Implementation of Vedic Multiplier for Digital Signal Processing

[1]S. S. Kerur, [1]Prakash Narchi, [1]Jayashree C N, [2]Harish M Kittur and [3]Girish V A

[1]Department of Electronics and Communication,
S.D.M.College of Engineering and Technology, Dharwad, Karnataka, India
[2] School of Electronics Engineering, V.I.T, Vellore, India.
[3] Department of ECE, D.S.Engineering College, Bangalore, India.

ABSTRACT

Digital signal processors (DSPs) are very important in various engineering disciplines. Fast multiplication is very important in DSPs for convolution, Fourier transforms, etc. A fast method for multiplication based on ancient Indian Vedic mathematics is proposed in this paper. The whole of Vedic mathematics is based on 16 sutras (word formulae) and manifests a unified structure of mathematics. Among the various methods of multiplication in Vedic mathematics, Urdhava tiryakbhyam is discussed in detail. Urdhava tiryakbhyam is the general multiplication formula applicable to all cases of multiplication. The coding is done in VHDL (very high speed integrated circuit hardware description language) and synthesis is done using Xilinx ISE series. The combinational delay obtained after the synthesis is compared with normal multiplier. Further, this Vedic multiplier is used in matrix multiplication. This Vedic multiplier can bring great improvement in the DSP performance.

Keywords: Vedic multiplier,Urdhava tiryakbhyam, VHDL, DSP.

1. INTRODUCTION

Vedic mathematics[2] is the name given to the ancient system of mathematics, or, to be precise, a unique technique of calculations based on simple rules and principles with which any mathematical problem can be solved – be it arithmetic, algebra, geometry or trigonometry. The system is based on 16 Vedic *sutras* or aphorisms, which are actually word formulae describing natural ways of solving a whole range of mathematical problems. Vedic mathematics was rediscovered from the ancient Indian scriptures between 1911 and 1918 by Sri Bharati Krishna Tirthaji (1884-1960), a scholar of Sanskrit, mathematics, history and philosophy [1]. He studied these ancient texts for years and, after careful investigation, was able to reconstruct a series of mathematical formulae called *sutras*.

Bharati Krishna Tirthaji, who was also the former Shankaracharya (major religious leader) of Puri, India, delved into the ancient Vedic texts and established the techniques of this system in his pioneering work, *Vedic Mathematics* (1965), which is considered the starting point for all work on Vedic mathematics. According to Mahesh Yogi, *The* sutras *of Vedic Mathematics are the software for the cosmic computer that runs this universe*. A great deal of research is also being carried out on how to develop more powerful and easy applications of the Vedic *sutras* in geometry, calculus and computing. Conventional mathematics is an integral part of engineering education since most engineering system designs are based on various mathematical approaches. The need for faster processing speed is continuously driving major improvements in processor technologies, as well as the search for new algorithms. A multiplier is one of the key hardware blocks in most digital signal processing systems. With advances in technology, many researchers have tried to design multipliers which offer either of the following- high speed, low power consumption, regularity of layout and hence less area or even combination of them in multiplier. The Vedic mathematics approach is totally different and considered very close to the way a human mind works. In this work, we try to present multiplication operations and the implementation of these using both conventional, as well as Vedic mathematical methods in VHDL language [1]. We highlight a comparative study of both approaches in terms of gate delays.

The paper is organized as follows. In section 2, Vedic multiplication method based on Urdhava Tiryakbhyam sutra is discussed. Section 3 deals with the design and implementation of the above said multiplier. Section 4 summarizes the experimental results obtained, while section 5 presents the conclusions of the work.

2. THE VEDIC MULTIPLICATION METHOD

The proposed Vedic multiplier is based on the Vedic multiplication formulae (Sutras). These Sutras have been traditionally used for the multiplication of two numbers in the decimal number system. In this work, we apply the same ideas to the binary number system to make the proposed algorithm compatible with the digital hardware. Vedic multiplication based on Urdhava Tiryakbhyam sutra is discussed below:

2.1 Urdhava Tiryakbhyam sutra:

The multiplier is based on an algorithm Urdhava Tiryakbhyam (Vertical & Crosswise) of ancient Indian Vedic Mathematics. Urdhava Tiryakbhyam Sutra is a general multiplication formula applicable to all cases of multiplication. It literally means "Vertically and crosswise". It is based on a novel concept through which the generation of all partial products can be done with the concurrent addition of these partial products. The parallelism in generation of partial products and their summation is obtained using Urdhava Tiryakbhyam explained in Figure 1. The algorithm can be generalized for n x n bit number. Since the partial products and their sums are calculated in parallel, the multiplier is independent of the clock frequency of the processor. Thus the multiplier will require the same amount of time to calculate the product and hence is independent of the clock frequency. The net advantage is that it reduces the need of microprocessors to operate at increasingly high clock frequencies. While a higher clock frequency generally results in increased processing power, its disadvantage is that it also increases power dissipation which results in higher device operating temperatures. By adopting the Vedic multiplier, microprocessors designers can easily circumvent these problems to avoid catastrophic device failures. The processing power of multiplier can easily be increased by increasing the input and output data bus widths since it has a quite a regular structure [7]. Due to its regular structure, it can be easily layout in a silicon chip [3]. The Multiplier has the advantage that as the number of bits increases, gate delay and area increases very slowly as compared to other multipliers. Therefore it is time, space and power efficient.

Multiplication of two decimal numbers- 325*738

To illustrate this multiplication scheme, let us consider the multiplication of two decimal numbers (325 * 738). Line diagram for the multiplication is shown in Figure 2. The digits on the both sides of the line are multiplied and added with the carry from the previous step. This generates one of the bits of the result and a carry. This carry is added in the next step and hence the process goes on. If more than one line are there in one step, all the results are added to the previous carry. In each step, least significant bit acts as the result bit and all other bits act as carry for the next step. Initially the carry is taken to be zero. To make the methodology more clear, an alternate illustration is given with the help of line diagrams in Figure 3 where the dots represent bit 0 or 1. [6]

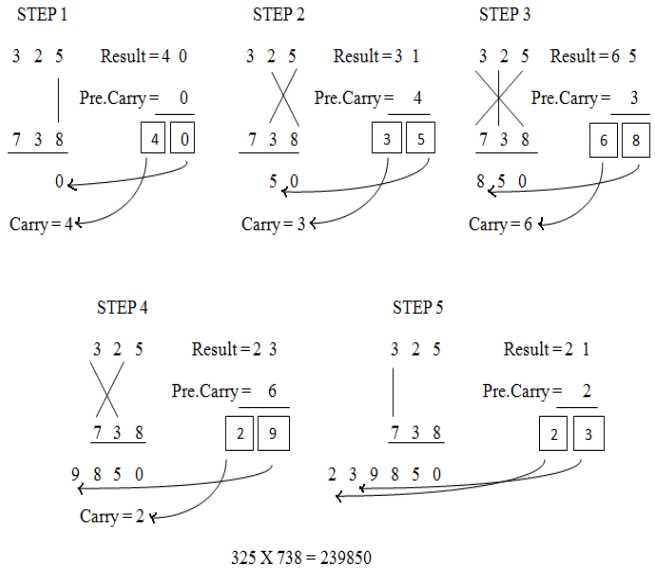


Figure 1: Multiplication of two decimal numbers by Urdhava Tiryakbhyam [4]

Algorithm for 8 X 8 Bit Multiplication Using Urdhava Tiryakbhyam (Vertically and crosswise) for two Binary numbers [8]-

A = A7A6A5A4 A3A2A1A0
X1 X0
B = B7B6B5B4 B3B2B1B0
Y1 Y0
X1 X0
* Y1 Y0

F E D C

CP = X0 * Y0 = C
CP = X1 * Y0 + X0 * Y1 = D
CP = X1 * Y1 = E
Where CP = Cross Product

To illustrate the multiplication algorithm, let us consider the multiplication of two binary numbers a3a2a1a0 and b3b2b1b0. As the result of this multiplication would be more than 4 bits, we express it as... r3r2r1r0. Line diagram for multiplication of two 4-bit numbers is shown in Figure 2 which is nothing but the mapping of the Figure 1 in binary system. For the simplicity, each bit is represented by a circle. Least significant bit r0 is obtained by multiplying the least significant bits of the multiplicand and the multiplier. The process is followed according to the steps shown in Figure 2.

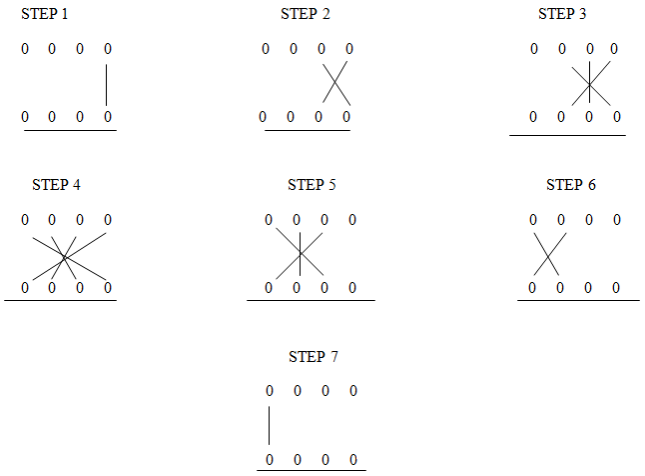


Figure 2: Line diagram for multiplication of two 4 - bit numbers.

Firstly, least significant bits are multiplied which gives the least significant bit of the product (vertical). Then, the LSB of the multiplicand is multiplied with the next higher bit of the multiplier and added with the product of LSB of multiplier and next higher bit of the multiplicand (crosswise). The sum gives second bit of the product and the carry is added in the output of next stage sum obtained by the crosswise and vertical multiplication and addition of three bits of the two numbers from least significant position. Next, all the four bits are processed with crosswise multiplication and addition to give the sum and carry. The sum is the corresponding bit of the product and the carry is again added to the next stage multiplication and addition of three bits except the LSB. The same operation continues until the multiplication of the two MSBs to give the MSB of the product. For example, if in some intermediate step, we get 110, then 0 will act as result bit (referred as rn) and 11 as the carry (referred as cn). It should be clearly noted that cn may be a multi-bit number.

Thus we get the following expressions:

$r0=a0b0;$ (1)

$c1r1=a1b0+a0b1;$ (2)

$c2r2=c1+a2b0+a1b1 + a0b2;$ (3)

$c3r3=c2+a3b0+a2b1 + a1b2 + a0b3;$ (4)

$c4r4=c3+a3b1+a2b2 + a1b3;$ (5)

$c5r5=c4+a3b2+a2b3;$ (6)

$c6r6=c5+a3b3$ (7)

With c6r6r5r4r3r2r1r0 being the final product. Hence this is the general mathematical formula applicable to all cases of multiplication.

The hardware realization of a 4-bit multiplier is shown in Figure 3. This hardware design is very similar to that of the famous array multiplier where an array of adders is required to arrive at the final product. All the partial products are calculated in parallel and the delay associated is mainly the time taken by the carry to propagate through the adders which form the multiplication array.

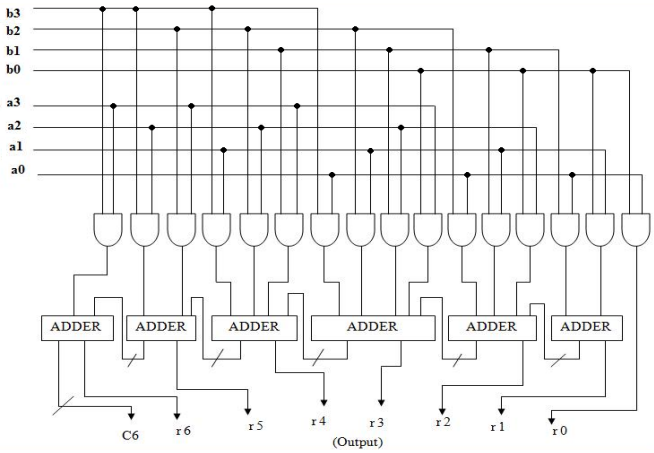


Figure 3: Hardware architecture of the Urdhava tiryakbhyam multiplier. [6]

3. DESIGN & IMPLEMENTATION

The Vedic multiplier is implemented using VHDL and also other multipliers like booth multiplier and array multiplier are also implemented. The functional verification through simulation of the VHDL code was carried out using ModelSim SE 6.0 simulator. The entire code is completely synthesizable. The synthesis is done using Xilinx Synthesis Tool (XST) available with Xilinx ISE 9.1i.

The design is optimized for speed and area using Xilinx, device family Spartan3, package pq208, and speed grade -4.

Table 1 and Table 2 indicate the device utilization summary of the array, booth and Vedic multiplier for 8 bit and 16 bit respectively. Table 3 gives the timing report of the implementation in terms of maximum combinational path delay. While Figure 4 indicates the RTL schematic of the 8 bit Vedic multiplier, that of 8 bit Normal multiplier is indicated illustrated in Figure 5.

TABLE 1

DEVICE UTILIZATION SUMMARY

Name of the Multiplier (8 bit)	**Number of slices**	**No of 4 i/p LUTs**	**No of IOs**	**No of bounded IOBs**
Array multiplier	71 out of 768	123 out of 1536	33	32 out of 124

Booth multiplier	96 out of 768	178 out of 1536	33	32 out of 124
Vedic multiplier	8 out of 768	16 out of 1536	80	80 out of 124

TABLE 2

DEVICE UTILIZATION SUMMARY

Name of the Multiplier (16 bit)	**Number of slices**	**No of 4 i/p LUTs**	**No of IOs**	**No of bounded IOBs**
Array multiplier	290 out of 768	505 out of 1536	65	64 out of 124
Booth multiplier	499 out of 768	923 out of 1536	65	65 out of 124
Vedic multiplier	120 out of 768	240 out of 1536	90	90 out of 124

TABLE 3

MAXIMUM COMBINATIONAL PATH DELAY (ns)

Name of the Multiplier	**8x8 bit**	**16x16 bit**
Array multiplier	32.01ns	60.928 ns
Booth multiplier	29.549 ns	70.809 ns
Vedic multiplier	24.16ns	36.563 ns

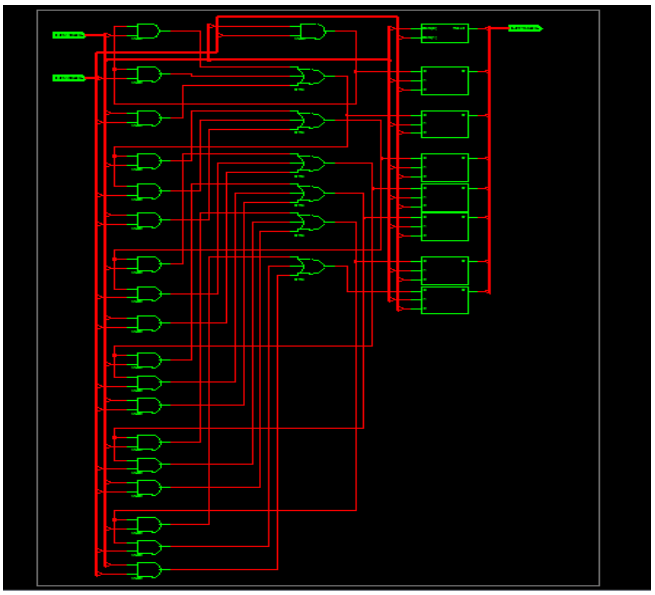


Figure 4: RTL schematic of 8 bit Vedic multiplier

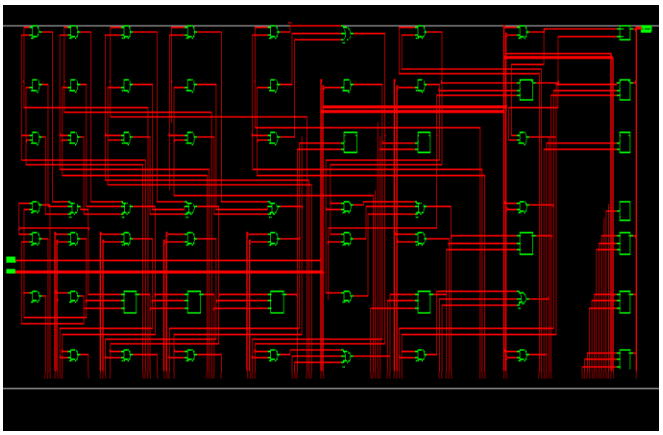


Figure 5: RTL schematic of 8 bit Normal multiplier

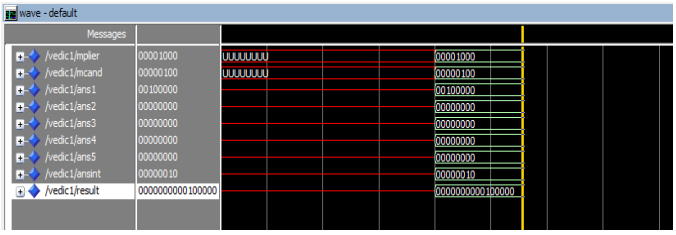
3.1 Simulation results:

[a] For Vedic multiplier(8x8 bit),multiplier a=00001000(8) and multiplicand b=00000100(4) and result=0000000000100000(32).

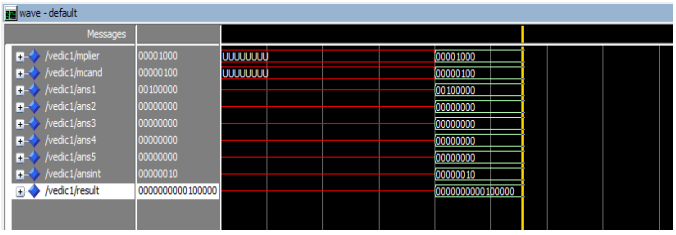
[b] For Vedic multiplier (16x16 bit), multiplier a=0000000000001000(8) and multiplicand b=0000000000000100(4) and result=00000000000000000000000000100000(32).

[c] For normal multiplier(8x8 bit), multiplier a=00001000(8) and multiplicand b=00000100(4) and result=0000000000100000(32).

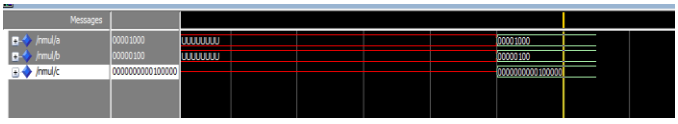
[d] For normal multiplier (16x16 bit),multiplier a=0000000000001000(8) and multiplicand b=0000000000000100(4) and result=00000000000000000000000000100000(32).



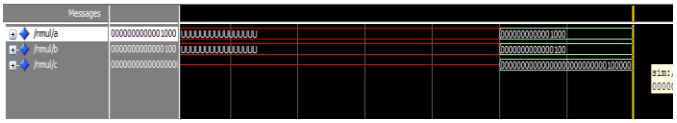
[a]



[b]



[c]



[d]

Also, this Vedic multiplier is used in the computation of 2x2 matrix multiplication. Synthesis results are given in Table IV for 2x2 matrix multiplication using normal (array) multiplier and Vedic multiplier.

TABLE 4

SYNTHESIS REPORT

Method	**No. of Slices**	**No of 4 i/p LUTs**	**No of IOs**	**No of bonded IOBs**	**No. of MULT 18x18s**	**Max Combinational path delay (ns)**
2x2 matrix using normal multiplier	176 out of 768	340 out of 1536	128	128 out of 124	4 out of 4	20.359 ns
2x2 matrix using Vedic multiplier	48 out of 768	88 out of 1536	96	64 of 124	4 out of 4	16.799 ns

4. EXPERIMENTAL RESULTS

In this paper, an attempt is made to explore the design space for optimal implementation of Vedic multiplier using VHDL. It is observed that 86.71% lesser slices and also around 88% lesser four input look-up are utilized for Vedic multiplier compared to other multipliers. The percentage utilization of bonded IOBs is 64.48%. The rest of the FPGA can be downloaded with the designs of varying complexities, since enough resources are still available.

From the above synthesis report and timing report, it can be inferred that 8 bit Vedic multiplier achieves higher speed by reducing gate delay by factor of 24% compared to array multiplier and around 18.2% compared to booth multiplier. Similarly, 16 bit Vedic multiplier achieves higher speed by reducing gate delay by factor of 39.9% compared to array multiplier and around 48.36% compared to booth multiplier.

Vedic multiplier has the greatest advantage as compared to other multipliers over gate delays and regularity of structures. The results also suggest that Vedic multiplier is faster than other multipliers and thus this is extremely advantageous.

5. CONCLUSION

Vedic Mathematics gives us a clue of symmetric computation. Vedic mathematics deals with various topics of mathematics such as basic arithmetic, geometry, trigonometry, calculus etc. All these methods are very efficient as far as manual calculations are concerned. The proposed Vedic multiplier proves to be highly efficient in terms of the speed. The main advantage is delay increases slowly as the input bits increases. Most of the important DSP algorithms, such as convolution, discrete Fourier transforms, fast Fourier transforms, digital filters, etc, incorporate multiply-accumulate computations. Since the multiplication time is generally far greater than the addition time, the total processing time for any DSP algorithm primarily depends upon the number of multiplications. Hence, this multiplier can be used to implement the above DSP algorithms.

6. ACKNOWLEDGMENT

We thank the Management, the Principal/Director and Staff of Sri Dharmasthala Manjunatheshwara College of Engineering and Technology, Dhavalgiri, Dharwad, Karnataka, India for encouraging us for this research work.

7. REFERENCES

[1] Charles. Roth Jr. "*Digital Systems Design using VHDL*," Thomson Brooks/Cole, 7th reprint, 2005.

[2] Jagadguru Swami Sri Bharati Krisna Tirthaji Maharaja, *Vedic Mathematics: Sixteen Simple Mathematical Formulae from the Veda*, Delhi (1965).

[3] H. Thapliyal and M. B. Shrinivas and H. Arbania, "Design and Analysis of a VLSI Based High Performance Low Power Parallel Square Architecture", Int. Conf. Algo. Math.Comp. Sc., Las Vegas, June 2005, pp. 72-76.

[4] P. D. Chidgupkar and M. T. Karad, "The Implementation of Vedic Algorithms in Digital Signal Processing", Global J. of Engg. Edu, vol.8, no.2, 2004.

[5] Shamim Akhter,"VHDL Implementation Of Fast NXN Multiplier Based On Vedic Mathematics", Jaypee Institute of Information Technology University, Noida, 201307 UP, INDIA, 2007 IEEE.

[6] Harpreet Singh Dhillon and Abhijit Mitra, "A Reduced- Bit Multiplication Algorithm for Digital Arithmetics", International Journal of Computational and Mathematical Sciences 2;2 © www.waset.org Spring 2008.

[7] Himanshu Thapliyal and M.B Srinivas, "An Efficient Method of Elliptic Curve Encryption Using Ancient Indian Vedic Mathematics", IEEE, 2005.

[8] Himanshu Thapliyal and M.B Srinivas, "VLSI Implementation of RSA Encryption System Using Ancient Indian Vedic Mathematics", Center for VLSI and Embedded System Technologies, International Institute of Information Technology Hyderabad-500019, India.

Authors Institutional affiliations and Addresses

Mr. **S. S. Kerur** received Bachelor of Engineering in Electronics and Communication Engineering from Karnataka University, Dharwad in1995 and Master of Engineering in Digital Electronics from Karnataka University Dharwad, in 1998.

During the year 1998, he served as a faculty in Sir M Visveshwaraya Institute of Technology, Bangalore. Later he joined S. D. M. College of Engineering and Technology, Dharwad, Karnataka, India. Since 2007, he is serving as Assistant Professor in the department of Electronics and Communication Engineering, S. D. M. College of Engineering and Technology, Dharwad, Karnataka, India. Further, he is promoted as Asst. Professor (Senior Grade) from the year 2010.His main research interests include VLSI design, embedded systems and communication systems. He is pursuing his research in the area of VLSI architecture analysis and Embedded systems.

He is currently the life member of Indian Society for Technical Education (ISTE), fellow member of Institution of Electronics and Telecommunication Engineers (IETE), member of VLSI Society of India (VSI). He is also chairman of IETE Sub-center, Dharwad.

Mr. **Prakash Narchi** received Bachelor of Engineering in Electronics and Communication Engineering from Vishveshwaraya Technological University, Belgaum in 2009 and currently pursuing Master of Technology in Digital Electronics under Vishveshwaraya Technological University, Belgaum.

His main research interests include VLSI design, computer architecture and VHDL.

Mrs**. Jayashree C. Nidagundi** received Bachelor of Engineering in Electronics and Communication Engineering from Karnataka University, Dharwad in 1997. After completing the graduation, she joined as faculty in the Department of Electronics and Communication in B.V.B College of Engineering and Technology, Hubli, Karnataka, India. Later she completed her M.Tech in VLSI Design and Embedded Systems in the year 2004, from PG department Visvesvaraya Technological University, Belgaum in 2004, Karnataka, India.

Her main research interests include VLSI Design, Embedded Systems. Currently, she is Since 2008, she is serving as Assistant Professor in the Department of Electronics and Communication Engineering, S. D. M. College of Engineering and Technology, Dharwad, Karnataka, India.

She is currently the member of Institution of Electronics and Telecommunication Engineers. (IETE) and also member of Institution of Engineers (IE).

Dr. Harish M. Kittur received his Ph.D and M.Tech in the area of solid state electronics and devices. He has served as a faculty in the SDM College of Engg. And Technology, Dharwad from 2005 to 2009. Now he is now serving as a faculty in the department of Electronics and Communication, Vellore Institute of Technology, Vellore.

He is life member of Institution of Electronics and Telecommunication Engineers.

Dr. Girish V. A received his Ph.D and M.Tech in the 2005 and 1995 respectively. He received his Ph.D in the area of microwave and communication engineering. He has served as a faculty in the SDM Engineering College, Dharwad from 1995 to 2009. Since 2009 he is now working as a Professor in the Dept. of E&CE of Dayanand Sagar college of Engg., Bangalore.

He is life member of Institution of Electronics and Telecommunication Engineers and also life member of Institution of Engineers.