ISSN(Online): 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Website: www.ijircce.com

Vol. 6, Issue 1, January 2018

Modulo 2^n±1 Adder/Subtractors for DSP Applications

Varsha Narasimha Murthy

Master's Student, Department of EEE, California State University, Sacramento, California, USA

ABSTRACT: In this paper, Modulo 2^n±1 individual andcombined adders/subtractors for 4-bit and 8-bit computations were proposed. These architectures were checked for working of operands in two widely known data representations, namelynormal and diminished-one representation. VLSI implementations for different simulation softwares (Mentor Graphics and Vivado) and Hardware devices (Artix 7) were tested for area and delay. The proposed architectures with operands in normal representation were designed and tested for parallel adders like Ripple Carry Adder (RCA) and Carry Look Ahead Adder(CLA) and parallel prefix adders. Brent Kung adder(BKA) and Kogge Stone adder(KSA) known as the delay efficient parallel prefix adders were used to design the modulo architectures for both normal and diminished-one data representations. BKA is shown to reduce the area by 27% for 4-bit computations and increase the performance by 51% for 8-bit computations. Radix-2 Modified Booth Multiplier architecture was used to test the efficiency of proposed modulo architectures.

KEYWORDS: Residue Number System, Modulo Architectures, Normal and Diminished representaions, Parallel adders, Parallel prefix adders, Modified Booth Multiplier.

I. INTRODUCTION

Residue Number System (RNS) is regarded as one of the carry free number system and is non-weighted. Here, a large integer is represented using a set of smaller integers, which provides a efficient way for computation of numbers. When a large integer is decomposed into smaller set of integers, large calculations can be performed independently and in parallel. Since the calculations are done on its corresponding residues in parallel, the hardware requirement is reduced and speed of the operations is improved. Addition and Subtraction operations including complex multiplication operations can be carried out in the same duration as that needed for addition operation. It makes use of Chinese remainder theorem of Modular Arithmetic for its operation. Modular arithmetic is a system of arithmetic integers where the numbers wrap around upon reaching an end value in the range. The main difficulty of the residue code relative to arithmetic operations is the determination of the relative magnitude of two numbersexpressed in residue code. Linear congruencies were used to develop the residue code easily.

It has been implemented in various applications like in fast number theoretic transforms, discrete Fourier transforms and differentiators and parallel FIR. However, not much work has been presented on the design of modulo 2^n ± 1 subtractors. Since the majority of DSP algorithms require a significant number of addition and subtraction operations, two solutions are attainable for implementing these algorithms over an RNS. They were, either to include both adder and subtractor circuits, or to include a single unit capable of performing either addition or subtraction depending on a mode signal. The constraint on the first solution is that it doubles the required hardware, whereas the second solution does not allow additions and subtractions to be executed in parallel.

Our main objective revolves around designing modulo architectures for 2^n±1 Subtractors and Adders for DSP applications. Building on the base of residue number system, IEAC architectures and parallel prefix adders, we anticipate that the proposed architecture will be area and power efficient.

In this paper the problem of designing efficient modulo 2^n + 1 subtractors and combined adders/subtractors in a unified way is being dealt. Considering both the cases of normal and diminished-one operands' representation 4 bit and 8 bit operand computations.

DOI: 10.15680/IJIRCCE.2018. 0601078

General architecture for computation of modular arithmetic consists of three main blocks. The IEAC (Inverted End Around Carry) adder stage, multiplexer stage and IEAC parallel adder stage. The further architectures are modifications of the general architectures depending on the application. End-around-carry is a type of carry that is required when a radix minus one compliment representation of integers is used and two integers so represented were summed. If a carry is generated at the most significant end of the two numbers, then this carry must be added to the digit at the least significant end of the result to give the radix-minus-one complement representation of the sum.

II. ARCHITECTURE

Throughout this paper, two modulo arithmetic architectures were represented. Section one introduces to a modulo architecture which performs the function of both adder and subtractor. For the sake of comparison for different efficiency parameters, possible implementation of adder architectures are used. Ripple carry adder and carry look ahead adder being the most widely used and simple non logarithmic adders. The comparison of parameters such as area and power are determined and presented.

Section two introduces to modulo architecture which is a modification of the previous architecture and performs only subtraction. From previous results we obtained the fact the adder and subtractor combined architectures almost doubles the required hardware as in [1]. Here, we implement the arithmetic computation of bits in 2 ways. Firstly, normal representation wherein n+1 number of bits for modulo 2^n +1 were used, Ripple carry adder and carry look ahead adder were used for implementing this architecture.Secondly, Diminished one representation where we use n bits for modulo 2^n+1, Logarithmic adders such as Kogge stone adder and Brent kung adder were used for implementing this architecture.

Section three evaluates the comparison between the various architectures in terms of area and delay.Lastly section four introduces the applications of these architectures and one of them being modulo multiplier is implemented for the sake of completion. It is a Radix-2 modified booth multiplier which operates on the basis of the modulo architectures developed.

Section 1: Modulo 2^n+1 Adder/Subtractor for NormalRepresentation of Operands using Ripple Carry Adder

Modulo 2^n +1 subtractor utilizes ripple carry adder in the IEAC adder block. RCA is a most basic form of adder for multiple bit-inputs. Full adder cell is the fundamental building block of the ripple carry adder. The propagation of carry bit from one full adder cell to another adds to the delay in computation of the sum. RCA in Fig. 1 was replaced by Carry Look-ahead Adder for modulo 2^n +1 subtraction. CLA computes one or more bits ahead of the sum and reduces the amount of time required to calculate carry bits and results of larger valued operands.

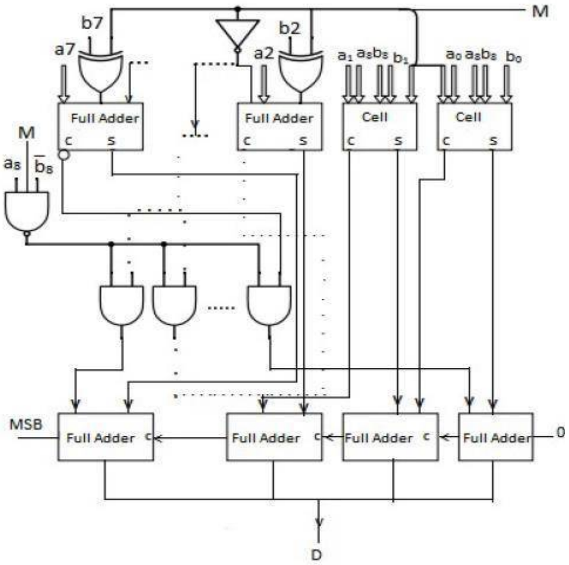


Fig 1. Proposed Modulo Adder/Subtractor for Normal Representation using RCA

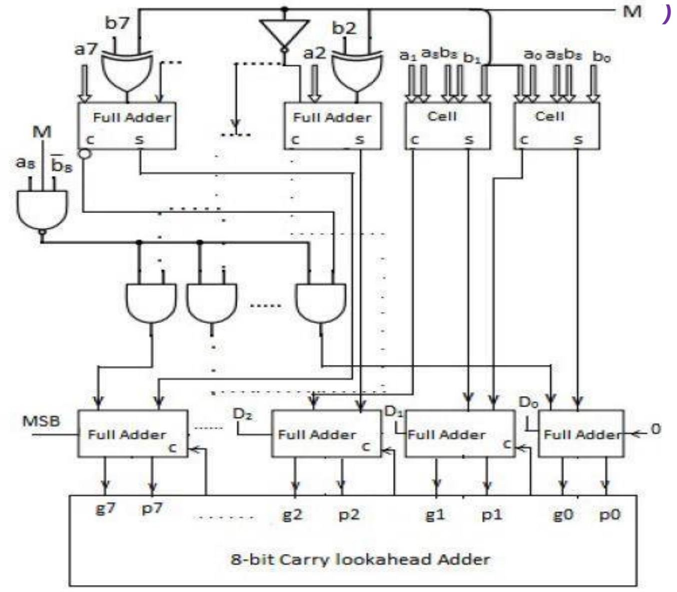


Fig 1.b Proposed Modulo Adder/Subtractor for Normal representation of operands using CLA

Section 2: *Modulo 2^n+1 Subtractor for Normal Representation using Ripple Carry Adder and Carry Look ahead adder.*

Assuming the fact that we consider two n+1 bit operands A=an …a0 and B= bn … b0 following Normal representation. Also, let |x|m denote the residue set of operand x when divided by m. The difference of A and B taken 2^n+1 modulo times can be realized as $|A+(2^{n+1}-1)-B+3|2\ ^{n}+1$ which further can be simplified as $|A+\overline{B}+3|2^{n}+1$ as in [1].

The above equation tells us that the difference of A and B is the sum of A and $\overline{B}$ taken modulo 2^n+1 times with a correction term equal to 3. As in [6] it is shown that sum of A and B operands can be carried out by Inverted End Around Carry (IEAC) n-bit parallel adder. In these n-bit parallel adders we implemented the four different kinds of it to compare the best results. As, mentioned before the Subtractors were implemented using RCA and CLA as parallel adder blocks with the AND gates. The subtractors were also implemented with MUXes as the replacement for AND gates with RCA and CLA as their parallel adder blocks as shown below.

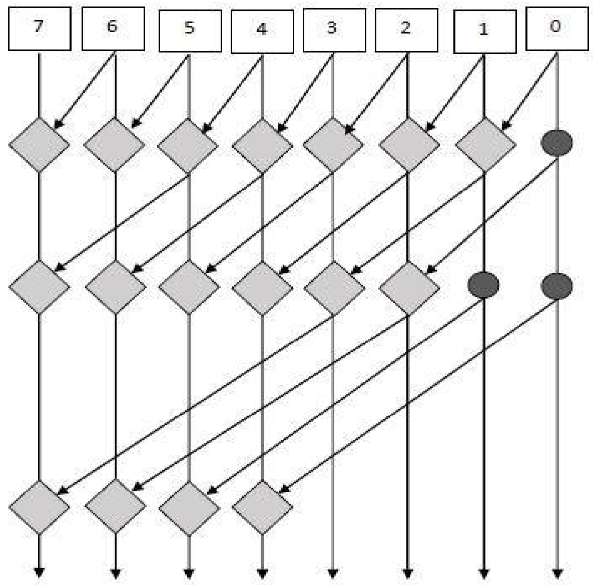


Fig 2.Kogge Stone Adder structure

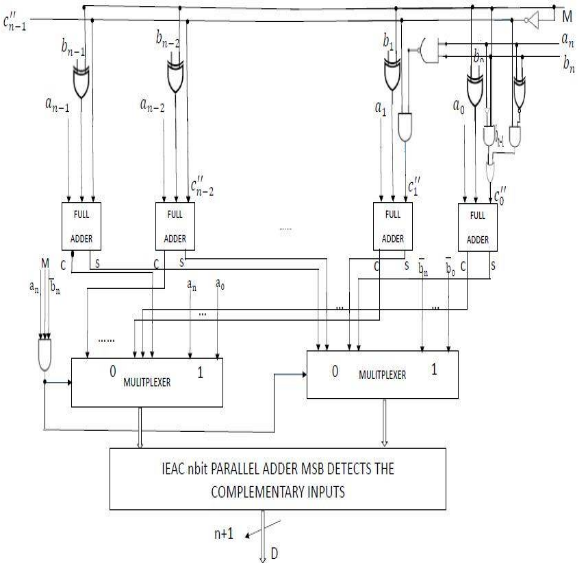


Fig 3. Modulo Subtractor Architecture

The parallel adder blocks were also implemented using parallel prefix adders such as Kogge stone adder and Brent Kung adder for better performance of the architecture.From [7] we derive that Kogge Stone adders are known for their low logic depths, high node count and minimal fan-out. The only disadvantage being the inclusion of wiring complexity. The *Kogge-Stone* tree achieves log2 *N* stages and fanout of 2 at each stage. This comes at the cost of many long wires that must be routed between stages. The tree also contains more PG cells; while this may not impact the area if the adder layout is on a regular grid, it will increase power consumption. Despite these costs, the Kogge-Stone tree is widely used in high-performance of adders. .

The *Brent-Kung* tree computes prefixes for 2-bit groups. These are used to find prefixes for 4-bit groups, which in turn are used to find prefixes for 8-bit groups, and so forth. The prefixes then fan back down to compute the carries-in to each bit. The tree requires 2log2*N* – 1 stages. The fanout is limited to 2 at each stage. The diagram shows buffers used to minimize the fanout and loading on the gates, but in practice, the buffers are generally omitted.Diminished-one representation involves operands with their values decreased by one. The representation uses only *n* bits.The architectures presented in [2] and [5] focus on handling zero values of operands and results.Parallel-prefix adders areused for computation of the difference. All parallel adder architectures consist of three main stages. Pre-processing stage computes propagate and generate signals. Carry generation network uses these signals to calculate intermediate carries. Post-processing stage computes the final sum and carry resultsA and B are n-bit operands. A* and B* are the diminished-one representations of the operands, which are reduced by a value of one. Az and Bz, when one, indicate that the respective operands are zero. A zero handling unit is necessary to indicate a zero result. The zero indication bit for the result is Dz. D* is the diminished-one representation of the difference of input operands

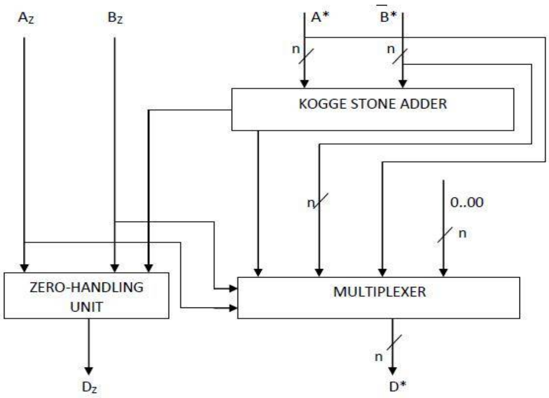


Fig 4 Modulo architecture for Diminished-one representation of operands using Kogge Stone Adder

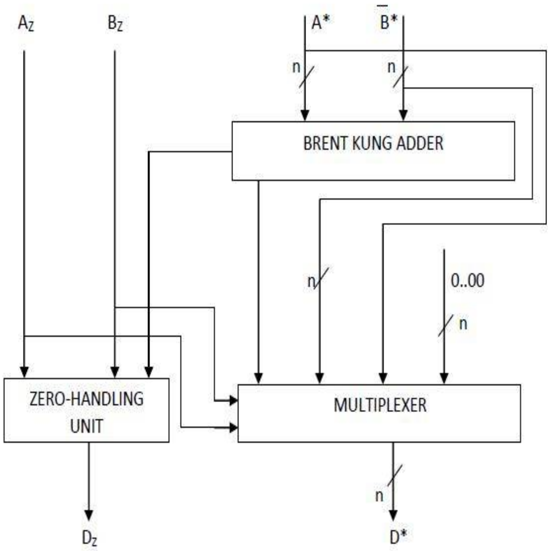


Fig 5 Modulo Architecture for Diminished-one representation of operands using Brent Kung adder

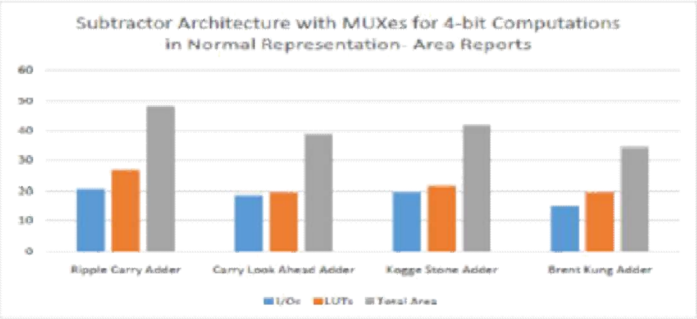
As mentioned earlier, Kogge Stone Adder is a parallel prefix form CLA Adder with low fan-out as inferred from [4]. Modulo subtractor using BKA affects circuit speed as they have logarithmic structures. Operands are combined in a tree-like structure as inferred from [4].

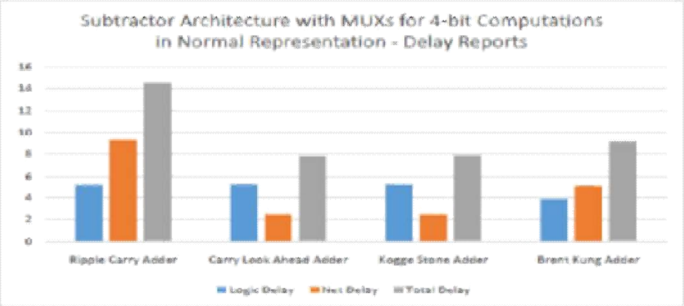
III. Simulation Results

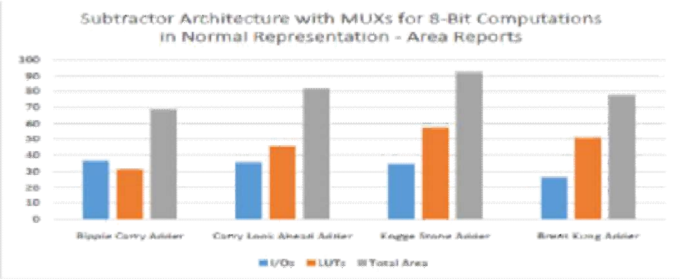
The different architectures above mentioned were implemented in different EDA platforms like Mentor Graphics and Vivado for getting accurate results. The designs were coded in Verilog and simulated and further synthesized to get the various area and delay reports. The codes were tested in Artix 7.

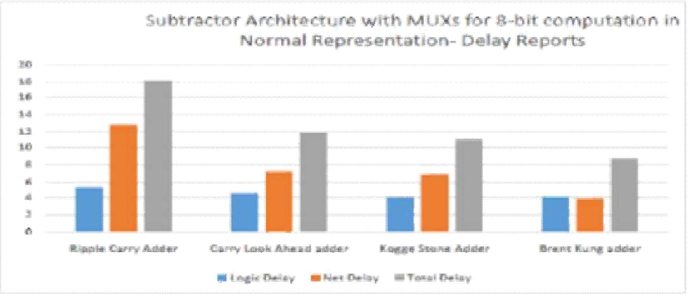
The number of input output (I/O) ports occupied by the architecture and the LUT reports were obtained. LUTs are the array that replaces runtime computation with a simpler array indexing operation. The total area occupied by the design on chip is also obtained.

Logic delay and Net delay reports were obtained where, Logic delay or commonly called as the propagation delay is the time length taken for the input applied, when it is stable and output obtained, when it's stable. Net delay is the difference between the time a signal is first applied to the net and the time it reaches other devices connected to the net. The total delay is obtained as the product of these two parameters. The architectures mentioned above were coded for 4-bit and 8-bit operands. The analysis of these various parameters allows us to explore different applications to which it can be implemented for.


Subtractor Architecture with MUXes for 4-bit Computations in Normal Representation- Area Reports


Subtractor Architecture with MUXs for 4-bit Computations in Normal Representation - Delay Reports


Subtractor Architecture with MUXs for 8-Bit Computations in Normal Representation - Area Reports


Subtractor Architecture with MUXs for 8-bit computation in Normal Representation- Delay Reports

IV. APPLICATION TO THE RADIX-2 MODIFIED BOOTH MULTIPLIER

The architecture presented as in [3] is said to speed up the implementation. It used the 2-bit recoding form of Modified Booth Algorithm for multiplication of two operands. The information encoded by each encoder block along with the multiplicand A is necessary for forming the partial product. The Selector block produces all the bits of the partial product. The result is then reduced to two by successive addition of their bits with equal weight. They are then added using the proposed modulo architectures. The results are accurate and found within the moduli range. As said the Modified Booth Multiplier is best used for Multiplication of two unsigned operands. The figure shows the Radix-2 Modified Booth

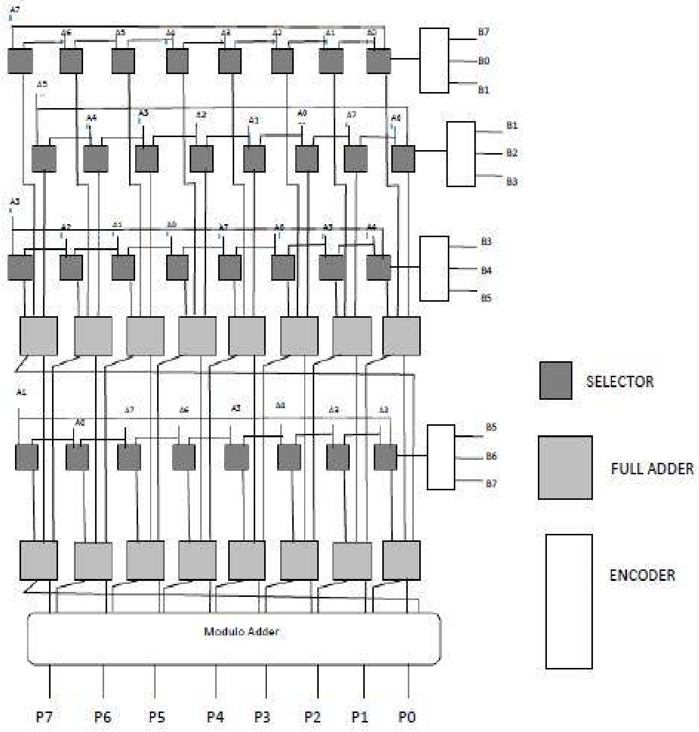


Fig 6 Multiplier with full adder blocks, encoder blocks selector blocks and modulo adder block

V. CONCLUSION

Several adder structures are replaced in conventional modulo adder architectures. These structures include Carry Look Ahead adder, Kogge Stone adder and Brent Kung adder in place of ripple carry adder. The performance of these subtractor and combined adder/subtractor architectures are analyzed in terms of area and delay. It is found that architecture implemented using Brent Kung Adder reduces the area by 27% for 4-bit computations and increase the performance by 51% for 8-bit computations. The adder architecture was further realized in a Modified Booth Modulo Multiplier

REFERENCES

1. E. Vassalos, D. Bakalis, H.T. Vergos, “On the Design odulo 2n±1 Subtractors and Adders/Subtractors”. Circuits Syst Signal Process. 1445–146 (2011)
2. C. Efstathiou, H.T. Vergos, D. Nikolos, Handling zero in diminished-one modulo 2n + 1 adders.Int. J. Electron. 90(2), 133–144 (2003)
3. C. Efstathiou, H.T. Vergos, D. Nikolos,Modified booth Modulo 2n −1 multipliers. IEEE Trans. Comput. 53(3), 370–374 (2004)
4. L. Kalampoukas, D. Nikolos, C. Efstathiou, H.T. Vergos, J. Kalamatianos, High-speed parallel prefix modulo 2n −1 Adders IEEE Trans. Computer 49(7), 673–680 (2000)
5. H.T. Vergos, C. Efstathiou, D. Nikolos, Diminished-one Modulo 2n + 1 adder design. IEEE Trans. Comput. 51(12), 1389–1399 (2002)
6. H.T. Vergos, D. Bakalis, C. Efstathiou, Fast modulo 2n + 1 multi-operand adders and residue generators. Integr. VLSI J. 43(1), 42–48 (2010)
7. Zahi Moudallal Ibrahim Issa Mohammad Mansour Ali Chehab Ayman Kayssi A Low power methodology for configurable wide Kogge
8. Stone adder Electrical and Computer Engineering Deparment American University of Beirut Beirut 1107 2020, Lebanon{zmm14, ihi02, mmansour, chehab, ayman}@aub.edu.lb
9. L. Kalampoukas, D. Nikolos, C. Efstathiou, H.T. Vergos, J. Kalamatianos, High-speed parallel prefix modulo 2n −1 Adders IEEE Trans. Computer 49(7), 673–680 (2000)
10. C. Efstathiou, H.T. Vergos, D. Nikolos,Modified booth Modulo 2n −1 multipliers. IEEE Trans. Comput.53(3), 370–374 (2004)
11. H.T. Vergos, D. Bakalis, C. Efstathiou, Fast modulo 2n + 1 multi-operand adders and residue generators.Integr. VLSI J. 43(1), 42–48 (2010)
12. E. Vassalos, D. Bakalis, H.T. Vergos, “On the Design odulo 2n±1 Subtractors and Adders/Subtractors”. Circuits Syst Signal Process. 1445 146 (2011)
13. H.T. Vergos, D. Bakalis, C. Efstathiou, Fast modulo 2n + 1 multi-operand adders and residue generators. Integr. VLSI J. 43(1), 42–48 (2010)

BIOGRAPHY

Varsha Narasimha Murthy is a Graduate Student at California State University, Sacramento, United States of America who pursued her Bachelor’s degree in Electronics and Communication at PESIT-Bangalore South Campus, Bangalore, Karnataka, India. Her interests lies in MicroElectronics and Digital VLSI Design.