An Input Pattern Based Area Reduction Technique for Adder Structures in Low Power Applications

Krashna Nand Mishra, and Subash Chandra Bose

Abstract— **In this paper, we present a design methodology for the adder architectures that can be implemented in lesser area without sacrificing the performance in terms of speed and power. Based on the general idea for area reduction discussed in the design methodology, a new four-bit carry select adder has been designed and implemented. This new adder implements 1-bit addition using only six transistors considering the fact that the carry is "predetermined for one-bit block". Asynchronous nature of these cells has been exploited to reduce both static and dynamic power consumption upto 11-18% in comparison to conventional CMOS implementation. Critical path length is also reduced in comparison to other adder architectures; hence it shows significant improvement in speed with worst-case delay of 3.87 ns for four-bit slice. Both aspects of area modeling namely, number of transistors and layout regularity are being considered while integrating it into an area of 5927.55 μm^2 with almost 93% layout efficiency in 1.2 μm 5V SCL CMOS technology.**

Index Terms— **Adder; Low Power Design; Carry Select; Bit-slice;**

I. INTRODUCTION

Adder is one of the most widely used building blocks in all data processing (arithmetic) and digital signal processing architectures. In VLSI applications, Area and Power are very important factors that are taken into account while designing fast adders. Every aspect of the adder design directly relates to area optimization. Although, several things have already been tried out to minimize the area while designing such building blocks, we have approached this problem with a different perspective-

Since there are several combinations possible for the inputs to an adder, it is possible that some transistors are 'active' or switches only for a certain small set of critical combinations. So if the requirement is such as this set of combinations is not going to be used in some application, we can 'remove' the transistors corresponding to this critical set of combinations and save the area occupied by the transistors up to a great extent. In terms of number of input combinations possible for different architectures, silicon area can be reduced in terms of number of transistors used as well as the regularity in layout strategy to be followed, which has evolved after removing some transistors.

In this paper, novel architecture for improved CSA implementation will be presented and the circuit techniques for realizing different blocks used in four-bit CSA slice using transmission gate-pass transistor based logic will be described. Section III shows general architecture to implement carry select adder. The improved architecture for Four-bit slice has been described in section IV. The behavior and transistor level circuit implementation of different blocks used in this slice is discussed in its sub-sections. Slice characterization procedure has been given in section V. Section VI describes simulated results, and the conclusions are summarized in section VII

II. ALGORITHM

Inputs can appear either in serial or parallel manner. Based on the availability of inputs, the methodology can be stated under the following aspects:

- To freeze some of the primary input signals and replace the corresponding transistors (based on the value of these input signals) with appropriate connections to power supply or other nodes leading to area reduction in terms of number of transistors.
- To reduce excessive switching of nodes by implementing them in alternative architectural fashion resulting in the reduction of diffusion area (source and drain area of transistors) and thus glitching power also is reduced.
- Input driven circuit configurations that are used to reduce number of transistors do not always give proper voltage levels for some transition events. So by disallowing such transitions (and hence removal of corresponding transistors), area is reduced and voltage restoration logic can also be avoided.
- There is a subset of input patterns for which delay is more compared to other set of combinations. Hence such combinations can be avoided to improve speed performance.

Manuscript received June 21, 2006. This work was supported in part by the Central Electronics Engineering Research Institute, Rajasthan, India.

Krashna Nand Mishra is with the Dhirubhai Ambani Institute of Information & Communication Technology, Gujrat, India (e-mail: krashna_mishra@daiict.ac.in).

Subash Chandra Bose is with the IC Design Group, Central Electronics Engineering Research Institute, Rajasthan, India (e-mail: subash@ceeri.ernet.in).

In the architectural domain, Carry select adder (CSA) [1] uses two different chains of one bit adder cells corresponding to predetermined carry in values. Thus this algorithm can be exploited to reduce the number of transistors required to implement 1-bit cells. Consequently, reduced switching and critical path length adjustment also leads to higher speed and lesser power consumption. Although CSA shows significant improvement in speed [5] if carry look ahead carry is generated for LSB (Least Significant Bit), but at the cost of larger area and high power consumption.

III. A GENERAL ARCHITECTURE OF CSA

A typical carry-select adder breaks up the n bits into several stages, say k, with sizes s1, s2,…, sk such that ∑si = n. Most commonly used sizing scheme is an arithmetic progression: 1, 2,...,√(2n). In this case, two copies of a carry-evaluation block per select stage are needed. One copy evaluates the carry chain with the block-carry-in zero, while the other one assumes it to be one. But, still carry ripples inside the blocks putting 1-bit adder blocks in critical path to evaluate carry out of the stage. It has been depicted in Figure 1:

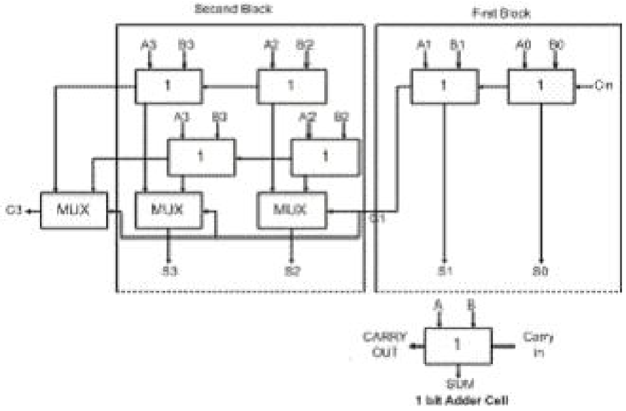


Fig. 1. Traditional CSA Block Diagram

Here, four bits are separated in two blocks of 2-bits each. Cin and C3 represent carry in and carry out respectively for the adder.

IV. REDUCED CARRY SELECT (RCS)-A NEW IMPLEMENTATION OF CSA

Figure 2 shows block diagram for four-bit CSA slice that consists of three types of blocks:

'14T' is a normal adder block that has been implemented using transmission gate based logic with pass transistors used near primary nodes. This circuit requires only 14 transistors to implement both sum and carry logic. These blocks are not the part of carry select logic. Its output provides carry for the next block.

Sizing for pass-transistor based circuits is not easily governed by logical effort technique [6]. Therefore, simulation based empirical approach has been followed to size transistors in this block for minimum glitches during transitions.

'0' and **'1'** are six transistor cells with predetermined carry inputs as 0 and 1 respectively. In this case, both sum and carry out are simultaneously available for each single bit block at time t1 and as soon as actual carry out is available from the previous block; it is being selected to the next block right after t1+Δ where Δ is single mux delay.

Here, **'m'** represents mux cells. Separate m-cells have been put for selecting sum and Cout in parallel evaluating both bits in single mux delay.

Thus, 1-bit adder blocks are no longer in critical path for output bits. So performance of m-cells would determine the performance of whole architecture under critical path condition. Since pass transistor based multiplexor is the fastest element in standard CMOS logic, performance can also be improved very much in speed. Although, adder operand bits have not been divided into blocks to implement select logic as original carry select adder architecture does [1], still it provides good performance in speed. It also avoids algorithmic constraints regarding optimal group distribution of bits. We further describe the operation of individual blocks in subsequent sub-sections.

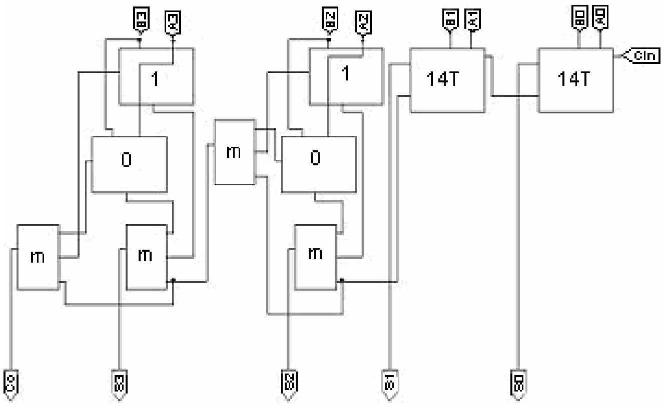


Fig. 2. Block Diagram of New Adder

A. Circuit Description: 14 transistor Adder Cell

Figure 3 shows schematic diagram of '14T' block. It uses only 14 transistors [2] and has been implemented using transmission gate based approach keeping in mind its fast propagation and reduced transistor count for implementing required function. Given the three 1-bit inputs A, B, and Cin, it calculates two 1-bit outputs S, for *sum* and Co, for *Carry out*, where

$$S = (A \oplus B) \oplus Cin$$

$$Co = A \cdot B + Cin \cdot (A \oplus B).$$

Power consumption is reduced by using only 4 transistors for XOR implementation [4] because there is no path for the output node to Vdd that results in no direct path from Vdd to Gnd. So there is no static power consumption.

'14T' uses one inverter, which introduces unwanted delay between N4 to N2 leading to a 0-0 or 1-1 overlap. This overlap causes glitches or spurious transitions at the output node. It also introduces static power component at the inverter output. Hence proper sizing has been done to minimize these glitches up to a large extent.

Due to the reduced voltage swing at N4 when A=B=0, both the PMOS's are ON and NMOS will be weakly ON, which will force to draw current from the power supply although the circuit is in steady state. Still this is a good option for low power cell because of having only 14 transistors.

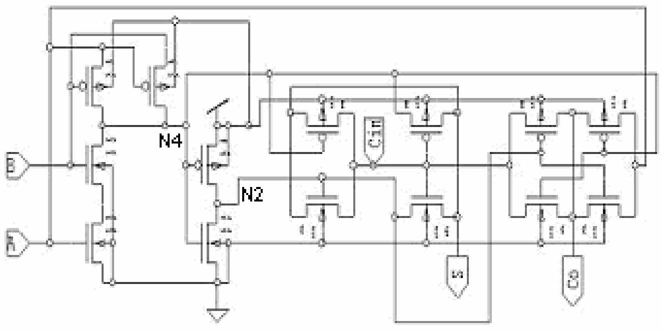
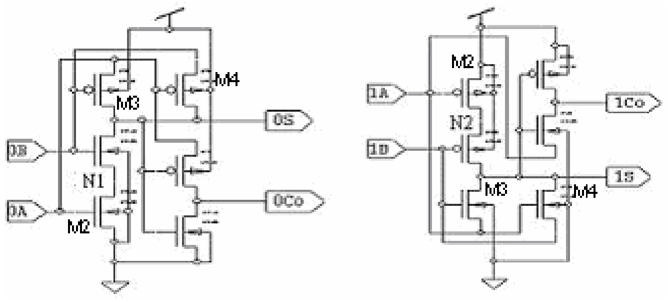


Fig. 3. Schematic Diagram of 14-transistor Transmission gate based Adder

B. *Circuit Description: 6-Transistor Adder Cell*

Figure 4 shows schematic diagram of '0' and '1' blocks. These six transistor cells operates as input-driven outputs except when both inputs are simultaneously "HIGH" for 0-cell and "LOW" for 1-cell, and output path is provided by Gnd and Vdd for 0 & 1-cell respectively.



(a) (b)

Fig. 4. Schematic Diagram of (a) 0-cell (b) 1-cell

Transistor M2 of both cells (the transistors, connected to either N1 or N2 at one end and to supply or Gnd at the other end) are of minimum size to minimize the glitches forced by charging or discharging of intrinsic capacitances at nodes N1 & N2 during switching. So other transistors in the chain are made wider to compensate for the delay imposed by these minimum-size transistors.

Proper sizing is being done considering the effective resistance values of R_n & R_p being 16.2kΩ and 31.3kΩ respectively in SCL Technology. The effective resistance is calculated as:

$$R = t_d/C_{load}$$

Where t_d is the delay time calculated at switching threshold (Vdd/2) between rising input and decaying output for NMOS, and similarly, between falling input and rising output for PMOS. Here, C_{load} is 0.1 pF for testing purpose. The test setup is shown in figure 5

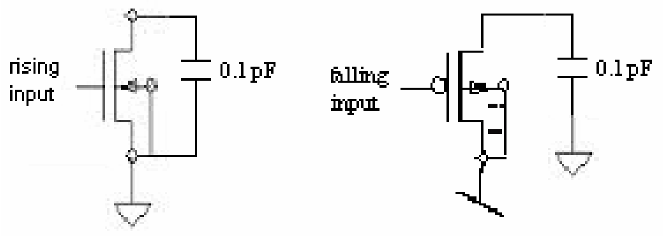


Fig. 5. Effective Resistance Calculation

When both inputs are simultaneously "LOW" for 0-cell and "HIGH" for 1-cell, voltage levels would be degraded by Vtn and Vtp respectively for 0 & 1 cells resulting in lower noise margin. Thus Transistors M3 & M4 of both cells are being kept as relatively low V_t transistors to improve noise margins.

Since output node path for both cells is not terminated by proper power supply either Vdd or Gnd for each combination, proper input buffering should be done restricting them to affect the performance of previous operating blocks in data-path architecture from where the desired inputs for adders are expected to come.

C. *Circuit Description: 5-Transistor MUX cell*

Figure 6 shows the implementation of MUX to select correct carry output based on carry-in available as carry out from the previous block. It is used as 'm' cell in the RCS architecture. M1 has been used as level restorer, which is sized so as not to beat the signal passing through the pass transistors during state transitions at the node that is to be restored. Also, it should be large enough to restore the voltage to Vdd quickly. Since pass transistor passes good '0', but a bad '1', the inverter has also been inversely ratioed to provide faster 1-> 0 transitions.

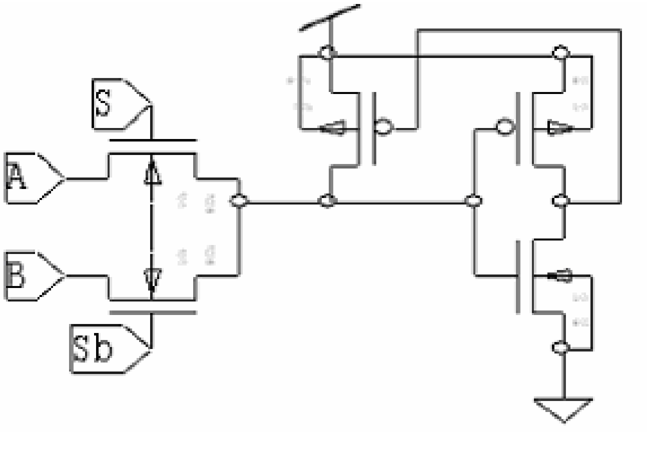


Fig. 6. Schematic Diagram of m-cell

D. *Asynchronous nature of 0 & 1-cells*

In CMOS circuits, gates dissipate energy only when they are switching. In synchronous circuit, gates switch just because they are connected to clock and not because they have new inputs to process. The clock driver that distributes clock signal evenly to all parts of a circuit switches to provide the timing reference even if a small part of the chip has something useful to do. For example, inputs may not change every time for one-bit block when clock switches. But arithmetic functions always have a fixed operation time independent of the inputs. It will take fixed number of clock cycles even for a simple addition like 1+1. Otherwise, it is difficult for the circuit to achieve pipelined operation. A synchronous circuit therefore wastes power when particular logic blocks are not utilized.

Asynchronous circuits, though inherently data driven, are active only when performing useful work. Since asynchronous logic block works with the help of 'request' and 'acknowledgement' signals, it avoids such complexity. Parts of the circuit that receives less data will automatically operate at a lower average frequency. The inherent nature of asynchronous circuits is input dependent computation, i.e., simple operands take less computation time, thus consuming less energy. Unfortunately, some additional logic is required to generate control signals. Thus, asynchronous circuits give power advantage at the cost of some additional area in comparison to synchronous circuits.

V. CHARACTERIZATION PROCEDURE

Figure 7 describes the adopted characterization procedure for the macrocell.

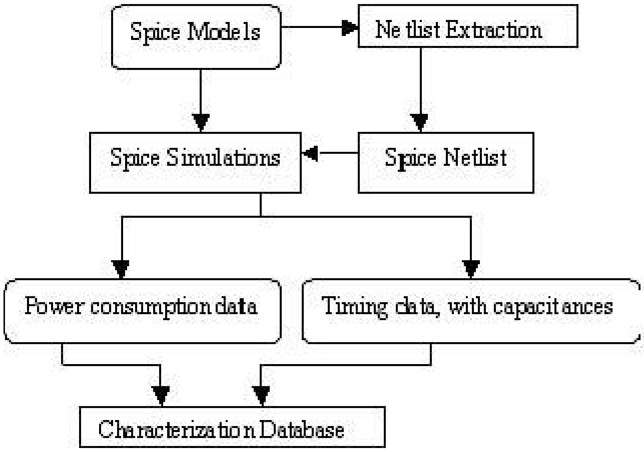


Fig. 7. Characterization Procedure

For characterization, SPICE netlist with distinct capacitances for each cell has been extracted from the cell layout using Tanner L-edit. The transistor models used are Level 13 model of 1.2µm CMOS Technology provided by Semiconductor Complex Limited (SCL), Chandigarh.

A. Power Modeling

To characterize the power consumption of a macrocell, a general method involves considering the power consumption of all possible input transition events in the look-up table. For larger adders, this approach is very time-consuming while calculating transition power for each entry in the look-up table.

But since these six-transistor 0 & 1-cells are going to be repeated further to implement large adders, we only need a systematic method to individually determine the contribution of each cell to the power consumption of whole circuit. Subsequently, the power consumption can be estimated for complete macrocell. We have estimated power consumption for these cells, using the following method.

Since total power in CMOS circuits is dominated by the dynamic switching of circuit elements (i.e., charging and discharging of capacitances), dynamic short-circuit (or overlap) currents and static leakage are of less importance. So we can find out average power consumption per million transitions at the output from spice simulations as given below [7].

Input patterns are applied in such a way that all possible combinations are covered.

```
vA A 0 bit({00111000110101100} pw=5ns on=5 off=0 rt=0.1ns ft=0.1ns)
vB B 0 bit({01010110111100000} pw=5ns on=5 off=0 rt=0.1ns ft=0.1ns)
.tran 5n 85n
.power vVdd 0n 85n
```

From output file, average power consumed -> p watts

Input pulse is making 1 transition per 5 ns, i,e, 1 million transition per 0.005 sec. so average power consumed per million transition= 0.005*p

B. Delay Modeling

Normally, delay models do not consider a realistic input waveform and consequently do not take into account the slope of input waveform in the propagation delay [3]. In real circuit applications, input waveform will depend on the fan-out and the driving capability of the preceding stage and therefore, the propagation delay will also depend on these parameters.

C. Area Modeling

Although transistor count is not a proper method for area estimation, it can provide suitable guideline for area estimation for different design structures. Although, in macrocell-based design techniques, two components-Cell area and Wiring area can be formulated for area modeling:

Total cell area is roughly proportional to the number of transistors or gate equivalents contained in a circuit. This can be precisely determined by netlist generated after physical layout.

Wiring area is proportional to the total wire length that can be estimated from the number of nodes and the average wire length of a node or, more accurate, from the sum of cell fan-out and the average wire length of cell-to cell connections (i.e. accounts for the longer wire length of nodes with higher fan-out). The wire lengths also depend on circuit size, circuit connectivity (i.e., locality of connections), and layout topology, which are known only after circuit partitioning and physical layout.

VI. SIMULATION RESULTS

Design has been implemented in 1.2 µm SCL technology using Tanner S-edit for schematic and Tanner L-edit for layout and spice simulations has been performed in Tanner T-spice considering worst-case NMOS & PMOS transistor models.

TABLE I
CHARACTERISTICS OF FOUR BIT TEST CIRCUIT

Attribute	Description
Process Technology	1.2 μm 5V SCL Level 13 Model
Architecture	Reduced Carry Select (RCS)
Worst Case Delay	3.87 ns
Area(Number of transistors/Colored area)	70/5927.55 μm^2
Power Consumption per million Transitions	2.7342 μW

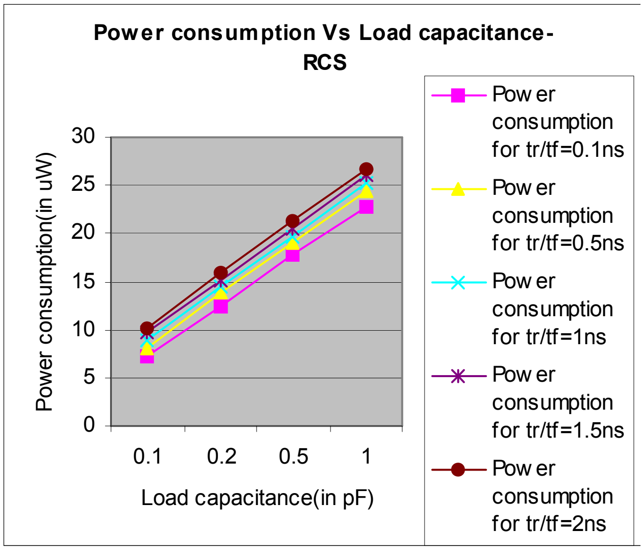


Fig. 8. Power consumption Vs Load Capacitance

VII. CONCLUSION AND FUTURE WORK

Critical path has been modified, resulting in improved timing performance. Required No. of transistors is also reduced to almost 33.3%, reducing area to a large extent.

Since all the cells except LSB cell are of identical nature in terms of regularity and signal activity, testing complexity is gracefully reduced while adding the constraint that same kind of testing block would be required for each bit block in fault detection and correction that would result in testing overhead.

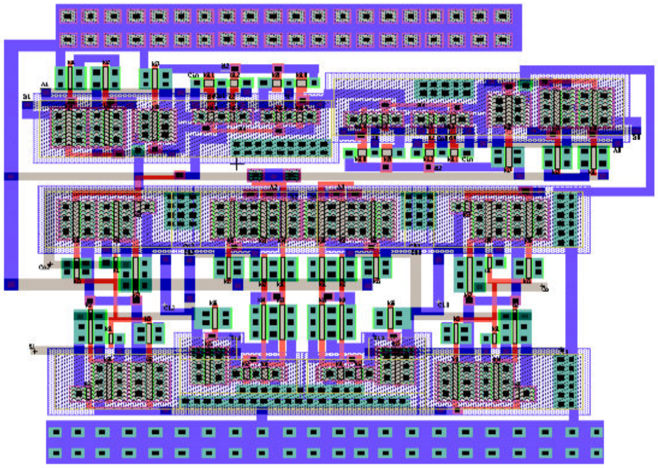


Fig. 9. Layout View of four bit RCS Slice

Since the evaluation of both carry and sum outputs of 0 & 1-cells is independent of other signal constraints, i.e., the cells are of asynchronous nature. Thus the performance can be improved while implemented through asynchronous mechanism.

Listed aspects of the algorithm can be focused on to derive several other hybrid macrocell architectures that may be very useful in small area domain.

ACKNOWLEDGMENT

I am greatly indebted to Dr. Dipankar NagChoudhuri who guided me to work in this direction and gave me an opportunity to work in one of the most prestigious research group, IC Design Group of CEERI.

I am also very much grateful to Mr. Raj Singh, Group Leader, IC Design Group, CEERI, Dr. A S Mandal, Mr. Sudhir Kumar and Dr. Abhijeet Karmakar for their time-to-time assistance during my work.

I want to thank Mr. Ravi Saini, Mr. Jay Gopal Pandey and Mr. Anil Saini who let me know various basic things about VLSI that turned out to be very useful in my work.

I would also thank Mr. P K Yadav, technical officer and Mr. Vinod Kumar for providing very good living facilities during my stay in CEERI.

I also want to express my gratitude to my fellow mates, Vaibhav, Neha, and Sirisha who were there in CEERI along with me from our VLSI student group. They were always a great source of inspiration and encouragement for me during my work.

REFERENCES

[1] O.J. Bedrij, “Carry-Select Adder”, IRE Transactions on Electronic Computers, pp.340-344, 1962.

[2] E. Abu-Shama and M. Bayoumi, “A new cell for low power adders,” in Proc. Int. Midwest Symp. Circuits and Systems, 1995, pp. 1014–1017.

[3] N. Hedenstierna and K. O. Jeppson, “CMOS circuit speed and buffer optimization,” IEEE Transactions on Computer-Aided Design, vol. CAD-6, pp.270–281, Mar. 1987.

[4] J Wang, S. Fang and W. Feng, “New Efficient Designs for XOR and XNOR functions on the Transistor Level,” in IEEE Journal of Solid State Circuits, Vol. 29, No. 7, pp. 780-786.

[5] R. P. Brent and H. T. Kung, “A regular layout for parallel adders,” in IEEE Transactions on Computers, vol. C31(3),pp. 260–264, March 1982.

[6] I. E. Sutherland, R. F. Sproull, and D. L. Harris. Logical Effort, Designing Fast CMOS Circuits. Morgan Kaufmann, 1999.

[7] Learning Material, “VLSI Subsystem Design”, prepared at CEERI as part of the project, “Special Manpower development for VLSI design and related software” under Ministry of Information & communication Technology, India

Krashna Nand Mishra received the Bachelor’s degree in Information & Communication Technogy in 2006 and currently working as the member of Technical staff for Telecom Competency center in the Persistent Systems Pvt.

Ltd, Pune, India. His research interests are VLSI analog & digital circuit design, particularly in Low power Design domain.

Dr. Subash Chandra Bose is a senior scientist in IC Design group at Central Electrinics Engineering Research Institute, Pilani, India. He has 18 years of experience in the field of VLSI Design, both Digital and Analog. He is also involved in teaching at various post graduate levels. He was awarded Senior DAAD fellowship in 1998. He has published more than 20 papers in conference proceedings and journals.