Design of High Performance Low Power 16 Bit Arithmetic Units Using Kogge-Stone Parallel Prefix Adder Architectures

Shubhajit Roy Chowdhury, Aritra Banerjee, Aniruddha Roy and Hiranmay Saha

Abstract-- **The necessity of designing high speed and low power arithmetic circuits suitable for computationally intensive applications has motivated the design of a high performance arithmetic unit using high speed parallel prefix adder architectures. The present work comprises of designing high performance low power 16 bit arithmetic unit using Kogge Stone parallel prefix adder architectures. Using the adder, a multiplexer based design of the arithmetic unit has been achieved that can perform basic operations like addition, subtraction, increment, decrement and data transfer. For the present work, 0.15µm technology has been used. Using the designed circuit, a delay of 0.3 ns has been found. The average power dissipation of the system is found to be equal to 8.4µW at a supply voltage of 1.5V. The whole simulation has been done using TSPICE.**

Index Terms-- **Architectural Design, Arithmetic Unit, Kogge-Stone Adder, Parallel Prefix Structure.**

I. Introduction

THE arithmetic operations of two binary numbers are one of the most interesting problems in modern digital VLSI systems consuming a major design effort of digital signal processors and general purpose microprocessors. The maximum operating speed of these processors depends on how fast the main computation block can process data. For a large number of applications, the speed critical computation block includes adders and subtractors either as stand alone blocks or integrated into multiplier and divider architectures. As a result specially optimized adder and subtractor architectures are required for high performance systems. High speed arithmetic operation on two binary numbers is specifically required for developing high speed arithmetic units. Subtraction is achievable by means of addition of the minuend with the two's complement of the subtrahend. As a result high adders are required for high speed subtraction. The design of faster and more efficient adder architectures has been the focus of research efforts over the past few decades. Some architecture like carry skip adder, conditional sum adder and carry select adder rely on a basic ripple carry adder structure that has been modified to shorten carry propagation path [1]. The carry look-ahead adders are commonly used to pre-calculate the carry signals. For wide adders of bit width (N) greater than 16, the delay of carry look-ahead adders becomes dominated by the delay of passing the carry through the look-ahead stages. This delay can be reduced by looking ahead across the look-ahead blocks [2]. This concept has led to the emergence of multilevel tree of look-ahead structures to achieve delay that grows with log N. Such adders are variously referred to as tree adders, logarithmic adders, multilevel look-ahead adders and parallel prefix adders. Some well known tree adder architectures include Sklansky adder [3], Brent Kung adder [4] and the Kogge Stone adder [5]. Such types of adders are commonly used in several DSP applications where huge strings of binary numbers are to be added within a very short delay.

The present work comprises of designing high speed arithmetic units integrable into high speed arithmetic unit (AU) using Kogge Stone adder architectures. Compared to its standard counterparts like Brent Kung which require a greater number of stages and Sklansky adders which have higher fanouts at the intermediate levels, Kogge Stone has been used in our design as it achieves both $\log_2 N$ stages and fan out of 2 at each stage [5]. The proposed arithmetic unit uses radix-2 Kogge Stone adders. Although higher valence Kogge Stone adders having fewer stages of logic are also reported, they are not used in the present design because each stage has a greater delay and this becomes a poor tradeoff so far as CMOS logic design is concerned [6]. CMOS logic design has been used to achieve a low power implementation of the system. The designed arithmetic unit can perform basic operations like addition, subtraction, increment, decrement and data transfer.
The paper is organized as follows: Section II discusses on the notion of parallel prefix adder structure that is typically used to speed up the addition process. Section III focuses on the architectural design of the arithmetic unit. Section IV highlights the design of the Kogge Stone adder. Section V discusses on the design of the multiplexer that is used to multiplex several operations on a single arithmetic unit. Section VI presents the simulation results.

Shubhajit Roy Chowdhury is with IC Design and Fabrication Centre, Dept. of Electronics and Telecommunication Engineering, Jadavpur University, Kolkata-700032, India. (e-mail: shubhajit@juiccentre.res.in).

Aritra Banerjee is with IC Design and Fabrication Centre, Dept. of Electronics and Telecommunication Engineering, Jadavpur University, Kolkata-700032, India. (e-mail: aritraetce@gmail.com).

Aniruddha Roy is with IC Design and Fabrication Centre, Dept. of Electronics and Telecommunication Engineering, Jadavpur University, Kolkata-700032, India. (e-mail: anietce@gmail.com).

Hiranmay Saha is with IC Design and Fabrication Centre, Dept. of Electronics and Telecommunication Engineering, Jadavpur University, Kolkata-700032, India. (e-mail: hsaha@juiccentre.res.in).

II. THE PARALLEL PREFIX PROBLEM

The carry look ahead and the tree adder architectures can be represented by a parallel prefix adder structure consisting of three main parts: preprocessing, carry look ahead network and post processing.

Let us consider two binary input vectors A and B, the preprocessing logic extract two special signals – propagate (P) and generate (G). The generate signal is high when a carry is generated at any stage. The propagate signal is high when there is a carry from a previous stage and the carry will be propagated into the next stage. The carry propagation problem can be expressed in terms of a prefix problem where for a set of binary inputs (x_i: i=0, 1, 2, …., n) the outputs (y_i: i=0, 1, 2, …., n) are defined by the help of an associative binary operator . as:

$$y_0 = x_0$$

$$y_i = x_i \bullet y_{i-1}$$

$$y_i = x_i \bullet x_{i-1} \bullet x_{i-2} \bullet \cdots\cdots\cdot \bullet x_0 \qquad (1)$$

Since the $\bullet$ operator is associative, it can be grouped in any order and computed in a number of levels. To express the sub products let us introduce the notation $Y_{i:j}^{k}$, where k is the level of the sub product and i:j represent a continuous range that this sub product covers. For the carry propagation problem, we define the sub product couple (G, P) such that:

$$(G,P)_{i:i}^{0} = (g_i, p_i) \qquad (2)$$

$$(G,P)_{i:j}^{k} = (G,P)_{i:q+1}^{k-1} \bullet (G,P)_{q:j}^{k-1} \qquad (3)$$

Where the desired

$$Carry_i = G_{i:0}$$

regardless of the number of levels necessary to cover the range i:0. Depending on the algorithm the carry propagation will have a different structure and shape. The maximum number of levels required to calculate the final Carry signal is referred to as the depth of the prefix graph is equal to the number of logic levels in the network. The depth of the carry propagate network is a function of the bit-width of the input. This number relates roughly to the delay of the network. The total numbers of binary associative operations within the network determine the active area required to compute the result. Secondary effects like the number of times a sub range is used in subsequent operations (fan out) and the distance between operators of an operation also contribute to the overall performance of the system.

III. ARCHITECTURAL DESIGN OF THE AU

The proposed AU can perform the following micro-operations shown in register transfer language (RTL) shown in table 1.

TABLE 1

Symbolic Notation	Description
$R3 \longleftarrow R1+R2$	Contents of R1 plus R2 is transferred to R3
$R3 \longleftarrow R1-R2$	Contents of R1 minus R2 is transferred to R3
$R2 \longleftarrow \overline{R2}$	1's Complements the contents of R2
$R2 \longleftarrow \overline{R2}+1$	2's Complements the contents of R2
$R2 \longleftarrow R1+\overline{R2}+1$	R1 plus 2's complement of R2
$R1 \longleftarrow R1+1$	Increment the content of R1
$R1 \longleftarrow R1-1$	Decrement the content of R1
$R1 \longleftarrow R2$	Transfer the content of register R2 to R1

The architectural design of the 16 bit arithmetic unit is shown in Fig. 1:

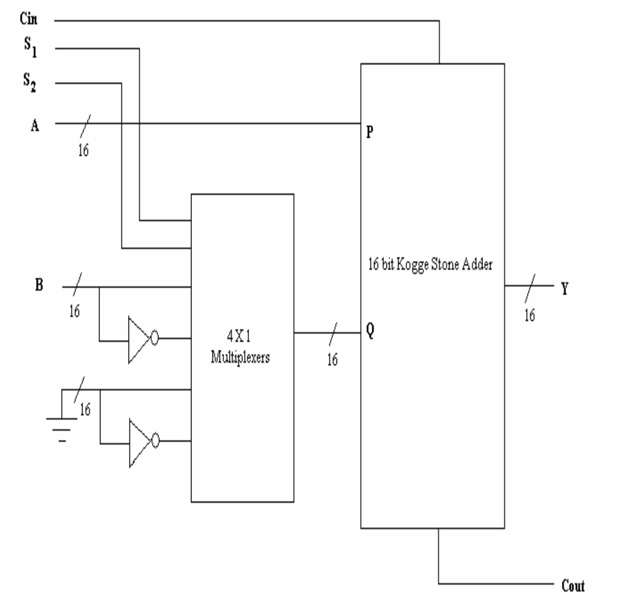

Fig. 1. Architectural design of the Arithmetic Unit.

IV. DESIGN OF THE KOGGE-STONE ADDER

The present work uses Kogge Stone adder to implement the arithmetic unit. The architecture of 16 bit Kogge-Stone adder consists of several cells. Fig. 2 shows the architecture of a 16 bit Kogge Stone adder employing 49 cells. It computes group generation terms and when necessary, group propagation terms from those calculated in the previous stages as shown in Fig. 5. Thus complete addition is performed in 4 stages which is $\log_2 16$ and delay is thus reduced logarithmically.

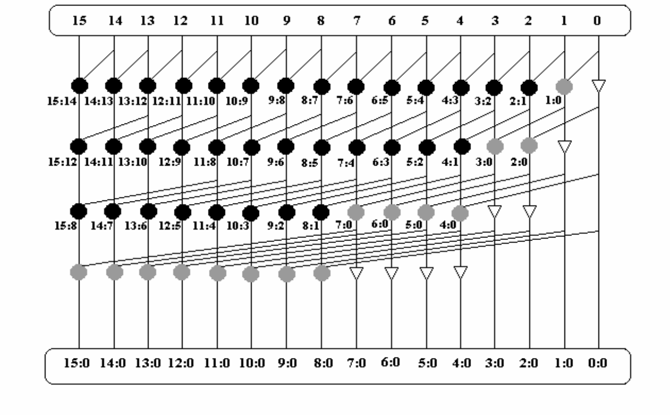

Fig. 2. Structure of the 16-bit Kogge-Stone adder.

The cells shown in figure 2 are categorized as black cells and grey cells. A black cell contains group generate and group propagate logic where a grey cell contains only the group generate logic. A group generates a carry if MSB part generates a carry output or, LSB part generates a carry and MSB part propagates the carry. When both MSB and LSB parts propagate the carry, then the overall group propagates the carry. The whole 16 bit adder comprises of 34 black cells and 15 grey cells. The valency-2 group generate signal ($G_{i:j}$) and group propagate signal ($P_{i:j}$) spanning bits i,…,j can be defined recursively for $i \geq k > j$ as:

$$G_{i:j} = G_{i:k} + P_{i:k}.G_{k-1:j} \tag{4}$$

$$P_{i:j} = P_{i:k}.P_{k-1:j} \tag{5}$$

Where,

$$G_{i:i} = G_i = A_i.B_i \tag{6}$$

$$P_{i:i} = P_i = A_i \oplus B_i \tag{7}$$

Then, the sum for bit i becomes:

$$S_i = P_i \oplus G_{i-1:0} \tag{8}$$

The carry input into bit i +1 is the carry output of bit i which is defined as,

$$C_i = G_{i:0} \tag{9}$$

A black cell computes group generation term $G_{i:j}$ and group propagation term $P_{i:j}$ from (4) and (5) respectively. The circuit of a single black cell is shown in Fig. 3. The circuit of the black cell is designed in such a way that complemented inputs arrive at the NAND gate simultaneously so that the OR function used for computing the group generate is done in just two gate delays instead of three gate delays as evident from (4).

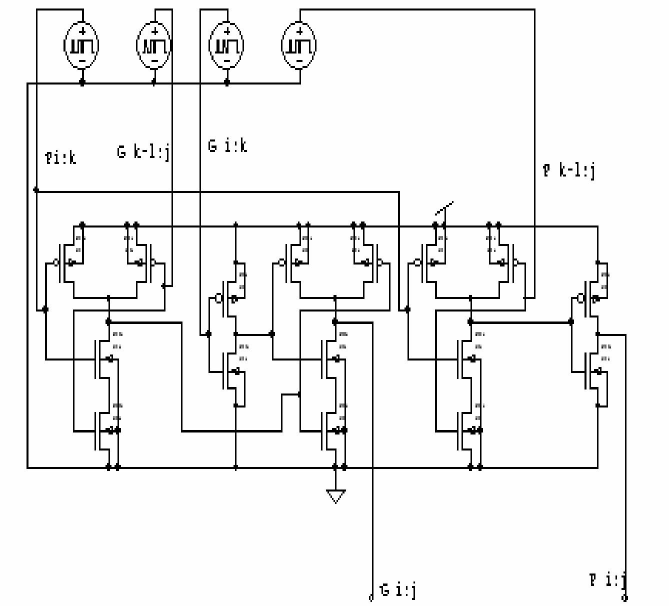

Fig. 3. Black Cell.

A grey cell computes only the group generation term $G_{i:j}$ using (4). Total delay of a grey cell also equals 2 gate delays. Fig. 4 shows the circuit diagram of a grey cell.

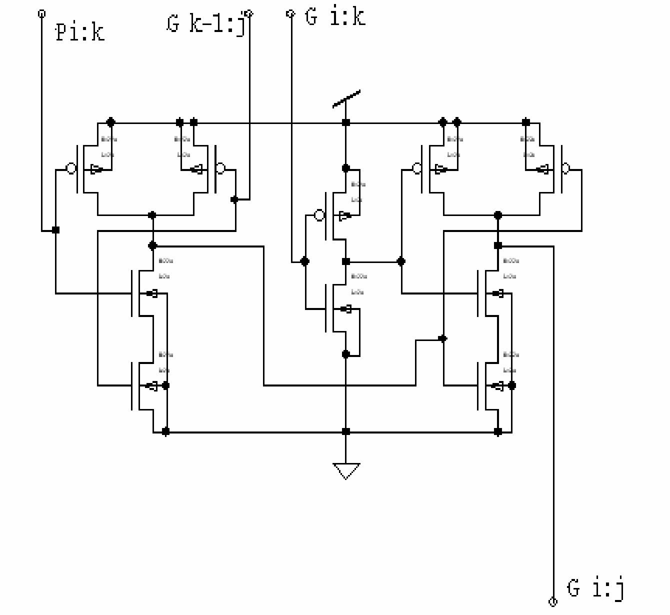

Fig. 4. Grey Cell.

For the 16-bit adder, $G_{i:i}$ and $P_{i:i}$ terms are calculated at the first stage from (6) and (7) respectively for each A_i , B_i pair. The circuit of a single block for this computation is shown in Fig. 5.

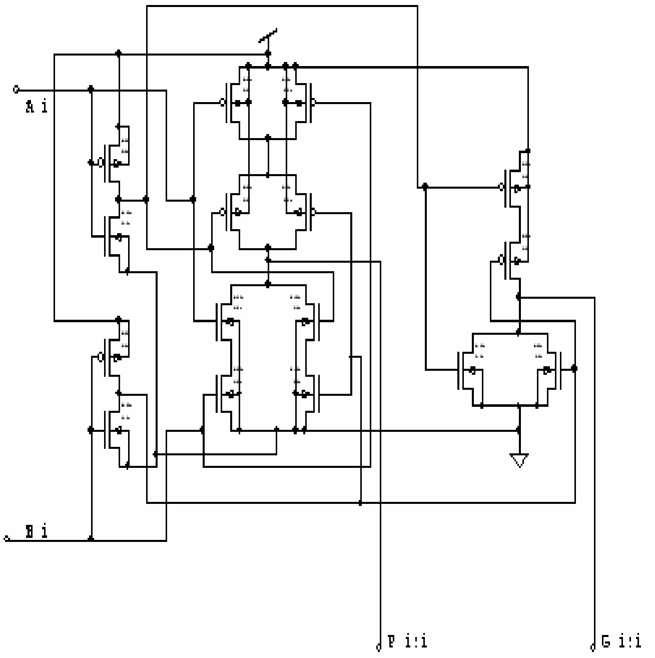

Fig. 5. Circuit for calculating $P_{i:i}$ and $G_{i:i}$.

At the last stage, sum bits are computed using (8) for all bit positions of the final 16 bit sum. The circuit for computing a single sum bit is shown in Fig. 6. The final C_{out} is the $G_{15:0}$ output term of last grey cell.

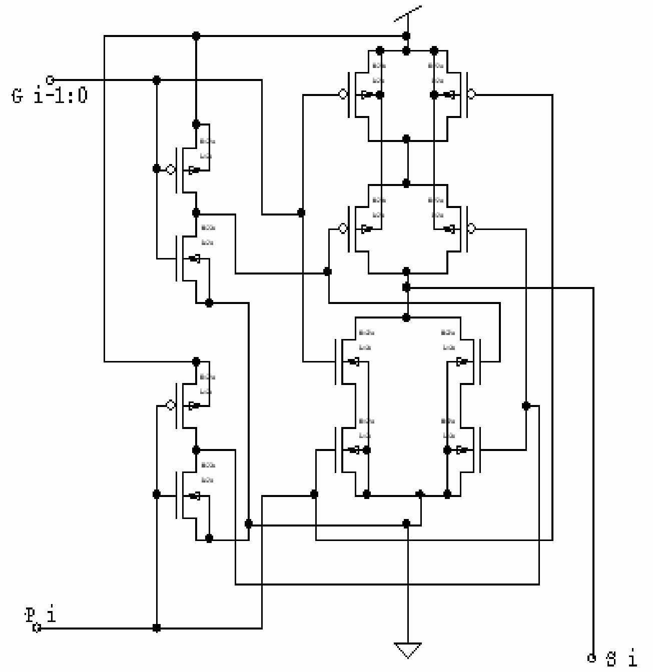

Fig. 6. Circuit for computing sum bit.

The circuit diagram of the whole Kogge Stone adder is shown in Fig. 7.

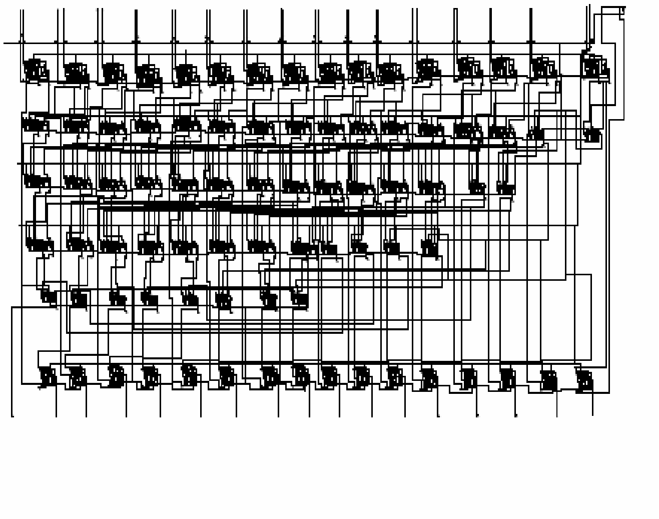

Fig. 7. Circuit diagram of the Kogge Stone adder.

V. DESIGN OF THE MULTIPLEXER

The proposed arithmetic unit can perform one of many operations and accordingly the operands may have to be presented to the adder in normal or complemented forms or bit strings of logic 0 or logic 1 only can be the one of the inputs to the adders as is evident from the discussion in section 3. The selection of a particular type of input to the adder is possible by employing a 4x1 multiplexer. The multiplexer has been designed using the CMOS logic design style. The circuit diagram of the multiplexer is shown in Fig. 8.

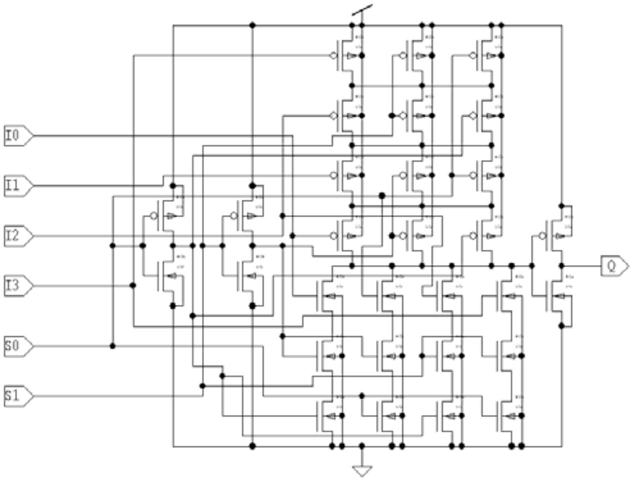

Fig. 8. Design of the 4X1 multiplexer.

S0 and S1 are its select inputs that selects between one of the four inputs I0, I1, I2 and I3 depending on its values. I0 and I1 are connected to B_i and $\overline{B_i}$ respectively. I2 and I3 are connected to logic 0 (ground) and logic 1 (i,e. Vdd) respectively. The output of the multiplexer is connected to the Q input of the Kogge Stone adder as shown in Fig. 1.

VI. SIMULATION RESULTS

SPICE Simulation results show that the average delay of the group generation and group propagation output of a black cell is equal to 0.045ns. The average delay of a single grey cell for computing group generation term is found to be 0.045ns. The average delay of computing final Carry output of the whole 16-bit Kogge-Stone adder circuit has been found 0.3ns. The average power dissipation of the whole structure is 8.4µ W. The supply voltage has been maintained at 1.5 Volts.

The function table of the arithmetic unit is shown in table 2.

TABLE 2

Select input	Inputs to adders	Output	Micro-operation
S1 S0 Cin	Q (P=A)	D=X+Y+Cin	
0 0 0	*B*	*D=A+B*	Add
0 0 1	*B*	*D=A+B+1*	Add with carry
0 1 0	$\overline{B}$	*D=A+* $\overline{B}$	Subtract with borrow
0 1 1	$\overline{B}$	*D=A+* $\overline{B}$ *+1*	Subtract
1 0 0	*0*	*D=A*	Data transfer
1 0 1	*0*	*D=A+1*	Increment
1 1 0	*1*	*D=A-1*	Decrement
1 1 1	*1*	*D=A*	Data transfer

The function table for the arithmetic unit through simulation conforms to the micro-operations that were intended to be implemented in the arithmetic unit.

VII. CONCLUSION

The present work comprises of the design of a low power high performance arithmetic unit. To increase the speed of computation of the arithmetic unit, Kogge Stone adder has been used instead of conventional carry propagation adders. The power and delay simulation of the whole circuit has also been done. In order to achieve a low power implementation of the system, CMOS logic design style has been used. Such type of arithmetic units can be suitably used in high performance processors typically for digital signal processing applications. Further work is going on.

VIII. REFERENCES

[1] S. Boyd, S.-J. Kim, D. Patil, and M. Horowitz, "Digital circuit optimization via geometric programming", *Operations Research*, 53(6):899–932, 2005.

[2] A.Weinberger and J.L. Smith, "A Logic for High Speed Addition", *National Bureau of Standards,* Circ. 591, pp. 3-12, 1958.

[3] J. Sklansky, "Conditional Sum Addition Logic", *IEEE Transactions on Electronic Computers,* Vol. 9, No. 6, pp.123-132, 1960.

[4] R.P. Brent and H.T. Kung, "A regular layout for parallel adders", *IEEE Transactions on Computers*, Vol. 31, No. 3, pp-64-65, 1982.

[5] P.M. Kogge and H. S. Stone, "A Parallel Algorithm for the Efficient Solution of a General Class of Recurrence Equations", IEEE Transactions on Computers, Vol. 22, No. 8, pp 224-228, 1973.

[6] N. Weste and D. Harris, "CMOS VLSI Design: A circuit and systems perspective", Prentice Hall of India, 2nd Edition, 2002.

IX. BIOGRAPHIES

Shubhajit Roy Chowdhury was born on August 27, 1981 in India. He received his B.E. degree in 2004 and M.E. degree in 2006. He is presently pursuing his Ph. D at the IC Design and Fabrication Centre, Department of Electronics and Telecommunication, Jadavpur University. He is a member of VLSI Society of India and a life member of the Indian Statistical Institute. His current research interests span around development of architectural design of ICs, FPGA based smart embedded systems for medical diagnosis and ASIC design of intelligent signal processing circuits. He has published over 18 papers in national and international journals and conferences.

Aritra Banerjee was born on April 28, 1987 in India. He is presently pursuing his B.E. in Electronics and Telecommunication Engineering from Jadavpur University. He is doing his B.E. research work at the IC Design and Fabrication Centre, Dept. of ETCE, Jadavpur University. His undergraduate research interests include power aware VLSI design of intelligent systems.

Aniruddha Roy was born on July 25, 1987 in India. He is presently pursuing his B.E. in Electronics and Telecommunication Engineering from Jadavpur University. He is doing his B.E. research work at the IC Design and Fabrication Centre, Dept. of ETCE, Jadavpur University. His undergraduate research interests include power aware VLSI design of intelligent systems.

Hiranmay Saha was born on January 4, 1946. He received his M. Tech degree in Radiophysics and Electronics from University of Calcutta in 1967 and Ph. D degree in Solar Cells and Systems from Jadavpur University in 1977. He is currently a Professor in the Department of Electronics and Telecommunication Engineering, Jadavpur University, Kolkata, India. He is a Fellow Member of Institution of Electronics and Telecommunication Engineers and was the Chairman of the same institution in the period 2004 – 2006. His present research interests include thin film silicon solar cells, smart sensors, VLSI based intelligent signal processing architectures and smart embedded systems.