Pergamon

0045-7906(94)00013-1

Computers Elect. Engng Vol. 21, No. 1, pp. 13–19, 1995

0045-7906/94 $7.00 + 0.00

MULTIPLIERLESS ARRAY ARCHITECTURE FOR COMPUTING DISCRETE COSINE TRANSFORM

MANIK CHANDRA MANDAL, ANINDYA SUNDAR DHAR and SWAPNA BANERJEE†
Department of Electronics & Electrical Communication Engineering, Indian Institute of Technology, Kharagpur-721 302, India

(Received for publication 21 July 1994)

Abstract—An array architecture for parallel computation of Discrete Cosine Transform (DCT) is presented. Multiplierless Givens rotors are employed as processing elements ensuring an economic usage of floor space. The interconnection pattern between the constituent linear arrays is determined by a permutation cycle involving a primitive root of the transform length that also governs the spatial sequence of the transform output points.

Key words: Givens rotor, array architecture, DCT.

I. INTRODUCTION

Orthogonal transforms find many important applications in digital signal processing and electromedical systems. Discrete Cosine Transform (DCT) is one of them, which has good data compression and filtering properties [1]. In many respects it closely resembles the statistically optimal Karhunen–Loeve transform (KLT). When the signal is real, use of DCT reduces the computation by a factor of 2 w.r. to conventional Discrete Fourier Transform (DFT) to evaluate the power spectrum of the signal.

In this paper an array architecture has been developed employing Givens rotor [2] as the processing element. The use of multiplierless Givens rotor reduces the hardware cost considerably compared to other functionally similar units involving multipliers. For the regular interconnection pattern the only prerequisite is that the transform length should be prime. The interconnection pattern between the processors of different linear arrays is governed by a permutation cycle involving the primitive root of the transform length.

Section II gives the theoretical considerations to develop the array architecture for the computation of DCT. In Section III, the time and hardware complexity of the implementation is discussed. Conclusions are drawn in Section IV.

II. THEORY

DCT of a data sequence $x(n)$, $n = 0, 1, 2, \ldots, N-1$ is defined [1,3] as

$$D(k) = \frac{2C(k)}{N} \sum_{n=0}^{N-1} x(n) \cos\left[\pi(2n+1)\frac{k}{2N}\right], \quad k = 0, 1, \ldots, N-1 \tag{1}$$

where

$$C(k) = \frac{1}{\sqrt{2}}, \quad k = 0$$
$$= 1, \qquad k = 1, 2, \ldots, N-1.$$

The inverse DCT (IDCT) is defined by

$$x(n) = \frac{1}{\sqrt{2}} D(0) + \sum_{k=0}^{N-1} D(k) \cos\left[(2n+1)\pi\frac{k}{2N}\right] \quad n = 0, 1, \ldots, N-1. \tag{2}$$

†Author for correspondence.

Since the constant factor $2C(k)/N$ in the definition of DCT can be lumped with other constants, hence without using it equation (1) is written as

$$D(k) = \sum_{n=0}^{N-1} f(n)\cos\left[k(2n+1)\frac{\pi}{2N}\right] \quad k = 0, 1, 2, \ldots, N-1. \tag{3}$$

It can be shown that the DCT is computable through DFT of the properly rearranged data sequence alongwith some plane rotation operations. The DFT itself can be computed through plane rotations [4], hence computation of the DCT can be carried out using only plane rotations.

Dropping the trivial $D(0)$ term, DCT can be decomposed into DFT [5] and plane rotation as shown below

$$[D(k) - D(N-k)] = [C(k)\ S(k)]Rot\left(\frac{\pi k}{2N}\right), \quad k = 0, 1, \ldots, N-1 \tag{4}$$

where

$$Rot(\theta) = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & -\cos\theta \end{bmatrix}$$

$C(k)$ and $S(k)$ are real and imaginary parts of the Fourier transform $F(k)$ of $x(n)$, which is obtained by rearranging $f(n)$.

$$F(k) = C(k) + jS(k) = \sum_{n=0}^{N-1} x(n)\, e^{-j2\pi nk/N}. \tag{5}$$

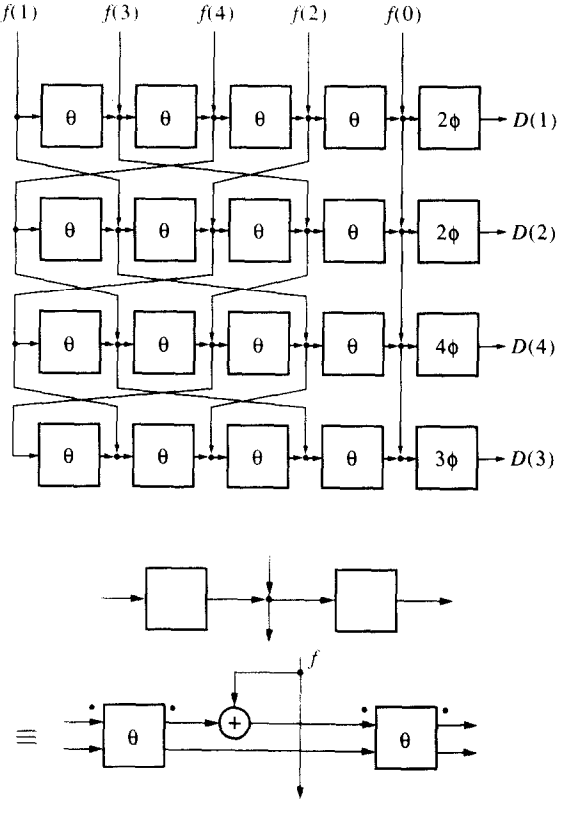

Fig. 1. Array architecture for DCT.

The sequence $x(n)$ is obtained from $f(n)$ as follows

$$x(n) = f(2n), \quad n = 0, 1, 2, \ldots, \frac{(N-1)}{2}$$

$$= f(2N - 2n - 1), \quad n = \frac{(N+1)}{2}, \ldots, N-1.$$

Since the transform length is prime and all prime numbers greater than 2 are odd, N is assumed to be odd.

The Fourier components are computed through plane rotation by

$$[C(k)\ S(k)] = \sum_{n=0}^{N-1} [x(n)\ 0] Rot(m\theta) \tag{6}$$

where $m = kn$ modulo N, $\theta = 2\pi/N$.

Substituting equation (5) into equation (4) we get

$$[D(k) - D(N-k)] = \left[\sum_{n=0}^{N-1} X(n) Rot(m\theta)\right] Rot(k\phi) \quad k = 1, 2, \ldots, \frac{N-1}{2} \tag{7}$$

where $X(n) = [x(n)\ 0]$ and $\phi = \theta/4 = \pi/2N$.

As the length of the transform is prime, the sequence $m \equiv [kn \text{ modulo } n]$ with n varying from 0 to $N-1$ is essentially a permutation of the natural numbers $0, 1, \ldots, N-1$. The permutation pattern changes with the value of $k = 1, 2, \ldots, N-1$. This ensures that the terms under the summation in equation (7) involves rotations through all multiples of θ [viz. $0, \theta, 2\theta, \ldots, (N-1)\theta$], once each.

By introducing transformation $k \to p_k$ in equation (7), where p_ks are the elements of a permutation cycle $P = \{p_k\}$, containing $N-1$ distinct integers in the range $[1, N-1]$ given by

$$p_k = g^{k-1} \text{ modulo } N \tag{8}$$

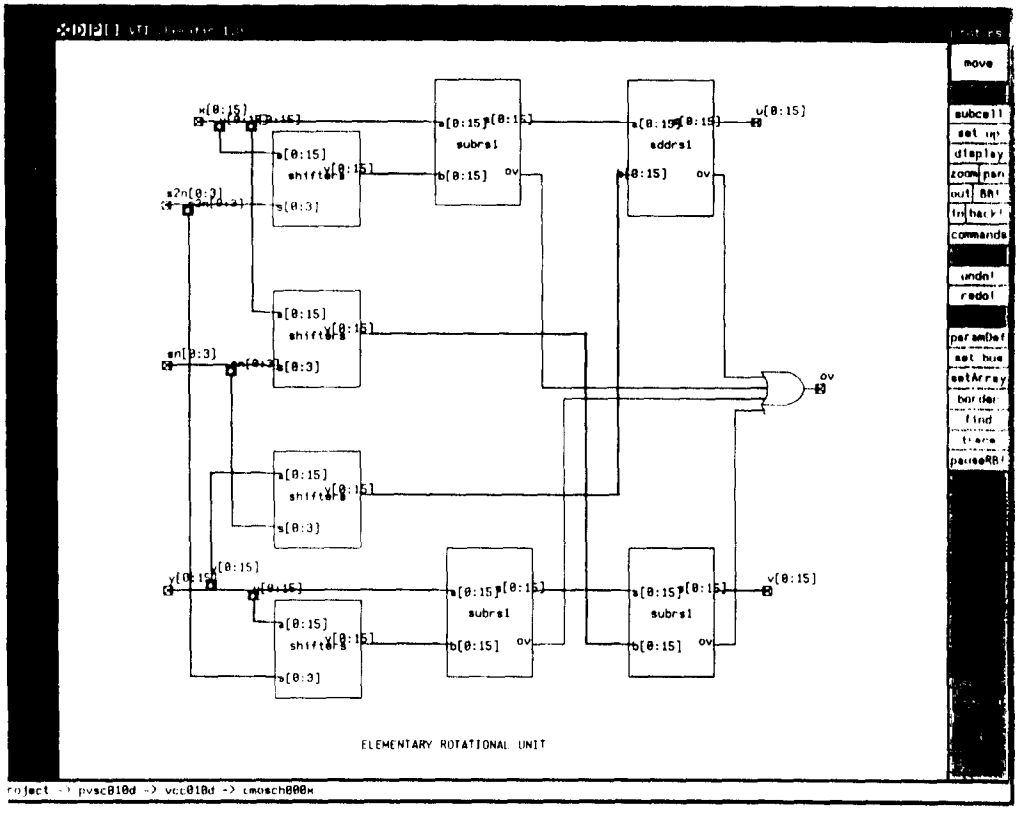

Fig. 2. Elementary rotational unit implemented using VTI CAD tools.

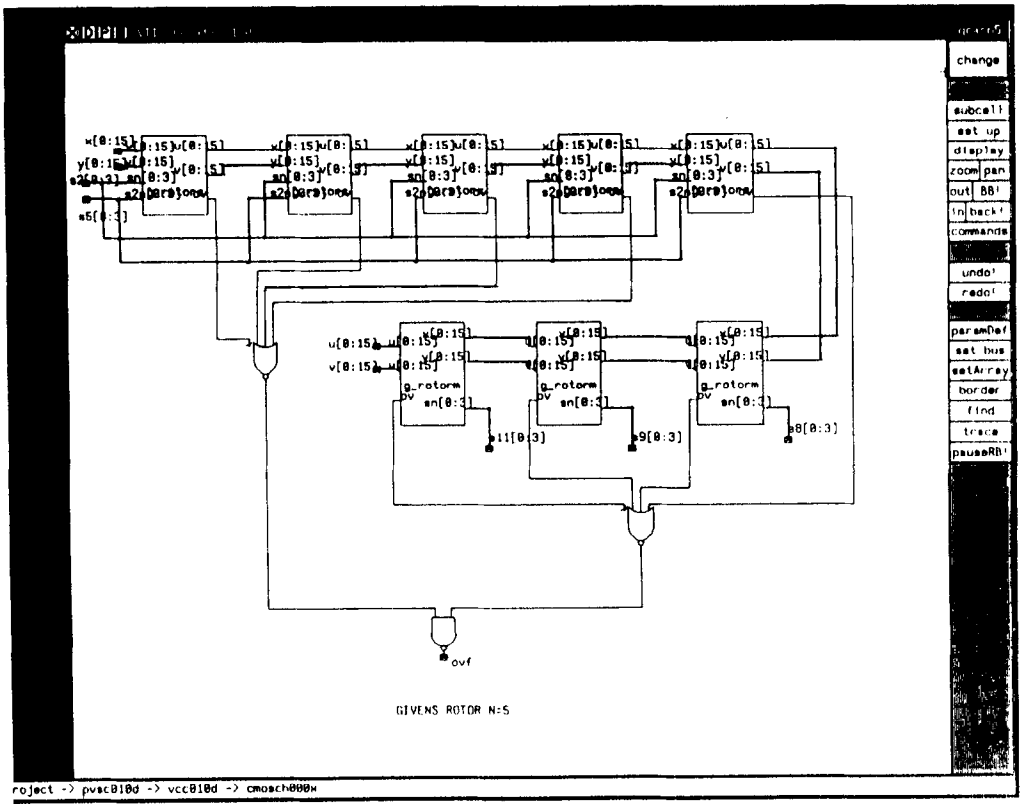

Fig. 3. Givens rotor implemented by cascading elementary rotational stages.

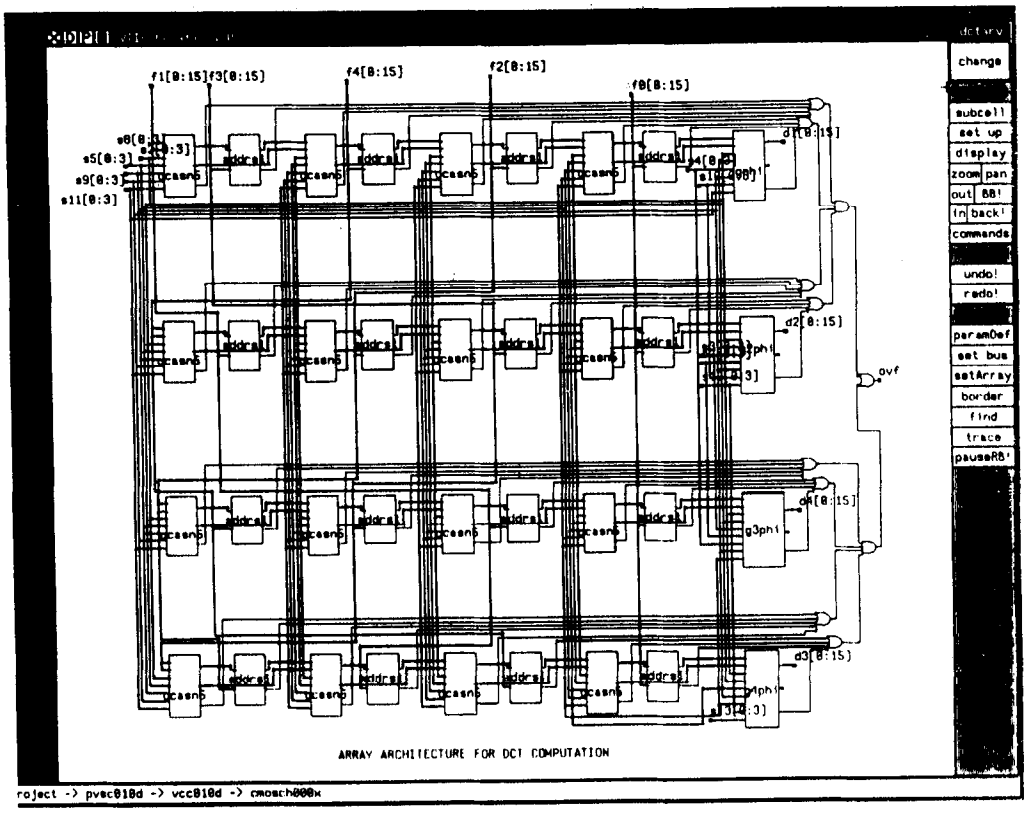

Fig. 4. Implementation of array architecture for DCT of transform length $N = 5$ using VTI CAD tools.

where g = primitive root of N [6]. By changing the index of summation in accordance to the multiples of θ and rearranging the data sequence, equation (7) is written as

$$[D(p_k) - D(N-p_k)]^T = \left[\sum_{m=0}^{N-1} (a_{k,m}) Rot(m\theta)\right] Rot(p_k\phi) \quad k = 1, 2, \ldots, \frac{(N-1)}{2}. \tag{9}$$

The rearranged data sequence indices can be found [6] from the relationship

$$a_{1,m} = m \quad m = 1, 2, \ldots, N-1 \tag{10a}$$

and

$$a_{i,P_d} = a_{i-1,P_{d-1}} \quad i, d = 2, \ldots, N-1. \tag{10b}$$

Equation (9) has the same form as N point Discrete Hartley Transform (DHT) [7] except that, it has another rotational stage ϕ. An array architecture is thus constructed with the Givens rotor as processing element for parallel computation of all the DCT points. The required computation depicted in equation (9) is directly mapped into the array architecture with the interconnection between various points of the array according to the relationship of equation (10b) [7] and is shown in Fig. 1.

III. COMPUTATIONAL COMPLEXITY

The elementary rotational stage is designed with shifters, adders and subtractors using VTI (VLSI Technology Inc., San Jose), CAD tools according to the principles laid down in the Appendix and is shown in Fig. 2. This is then cascaded with appropriate shifting to design the Givens rotor as depicted in Fig. 3.

For an N point DCT the number of processing elements required with notation θ, is $(N-1) \times (N-1)$ and with rotation $k\phi, k = 1, 2, \ldots, N-1$ is $N-1$ as shown in Fig. 1. The number of adders is $(N-1) \times (N-1)$. So the total number of processing elements is $N(N-1)$, $[= (N-1) \times (N-1) + (N-1)]$. As an example, for the DCT of length $N = 5$ which is implemented using VTI CAD tools, shown in Fig. 4. The total number of processing elements and total number of adders are 20 and 16 respectively.

The latency is defined by the time required to complete the DCT computation and throughput is defined as the required time from the first output of data sequence to the first output of the following data sequence. In this architecture all the inputs are given simultaneously and computation is done parallely, latency and throughput are the same. Let the time required to complete one operation by each PE be T_p and the time required for addition is T_a, then latency and throughput are given by $T = NT_p + (N-1)T_a$.

The simulation result for the multiplierless Givens rotor and adder for $N = 5$ gives $T_p = 3293.9\ ns$ and $T_a = 126.4\ ns$. So the latency and throughput is 16975.1 *ns*.

IV. CONCLUSIONS

An array architecture capable of computing all the DCT points simultaneously is presented. Using the multiplierless Givens rotor as the basic processing element ensures the economic usage of floor space. The Givens rotor is realised using elementary rotational units which is implemented using shifters, adders and subtractors. The regular interconnection pattern between linear arrays facilitates its VLSI implementation efficiently. The architecture can be extended with similar interconnection pattern [8] to compute two dimensional DCT.

REFERENCES

1. N. Ahmed, T. Natarajan and K. R. Rao, Discrete Cosine Transform. *IEEE Trans. Computers* **C-23**, 90–93 (1974).
2. P. Dewilde, E. Deprettere and R. Nouta, Parallel and pipelined VLSI implementation of signal processing algorithm. In *VLSI and Modern Signal Processing* (Edited by S. Y. Kung, H. J. Whitehouse and T. Kailath), pp. 257–276. Prentice-Hall, Englewood Cliffs, N.J. (1985).

3. M. J. Narasimha and A. M. Peterson, On the comparison of the discrete cosine transform. *IEEE Trans. Commun.* 934–936 (1978).
4. A. S. Dhar and Swapna Banerjee, Array architecture for computing Fourier transform to assist device modelling. *Electron. Lett.* **28,** 697–698 (1992).
5. M. Vetterli and H. J. Nussbaumer, Simple FFT and DCT algorithms with reduced number of operations. *Signal Process.* **6,** 267–278 (1984).
6. C. M. Rader, Discrete Fourier transforms when the number of data samples are prime. *Proc. IEEE* **56,** 1107–1108 (1968).
7. A. S. Dhar and Swapna Banerjee, An array architecture for fast computation of discrete Hartley transform. *IEEE Trans. Cir. Syst.* **38,** 1095–1098 (1991).
8. A. S. Dhar and Swapna Banerjee, An array architecture for computing two dimensional discrete Hartley transform. *Computers Elect. Engng.* **17,** 23–29 (1991).

APPENDIX

The processing element, Givens rotor is considered as a functional block implementing the equation

$$\begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}. \tag{A1}$$

For an N point DCT, $\theta = 2\pi/N$
Let

$$\theta = \alpha_1 + \alpha_2 + \cdots + \alpha_m.$$

As

$$\begin{bmatrix} \cos(\alpha_1+\alpha_2+\cdots+\alpha_m) & \sin(\alpha_1+\alpha_2\cdots+\alpha_m) \\ -\sin(\alpha_1+\alpha_2+\cdots+\alpha_m) & \cos(\alpha_1+\alpha_2+\cdots+\alpha_m) \end{bmatrix} = \begin{bmatrix} \cos\alpha_1 & \sin\alpha_1 \\ -\sin\alpha_1 & \cos\alpha_1 \end{bmatrix} \begin{bmatrix} \cos\alpha_2 & \sin\alpha_2 \\ -\sin\alpha_2 & \cos\alpha_2 \end{bmatrix} \cdots \begin{bmatrix} \cos\alpha_m & \sin\alpha_m \\ -\sin\alpha_m & \cos\alpha_m \end{bmatrix}. \tag{A2}$$

Hence

$$\begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix} = \begin{bmatrix} \cos\alpha_1 & \sin\alpha_1 \\ -\sin\alpha_1 & \cos\alpha_1 \end{bmatrix} \begin{bmatrix} \cos\alpha_2 & \sin\alpha_2 \\ -\sin\alpha_2 & \cos\alpha_2 \end{bmatrix} \cdots \begin{bmatrix} \cos\alpha_m & \sin\alpha_m \\ -\sin\alpha_m & \cos\alpha_m \end{bmatrix}. \tag{A3}$$

If the angle α_i is small enough, such that $\sin\alpha_i = 2^{-n_i}$ where n_i is a positive integer then $\alpha_i = 2^{-n_i}$ and $\cos\alpha_i = 1 - 2^{-(2n_i+1)}$.

Equation (A1) is rewritten for small angle α_i

$$\begin{aligned} u &= x[1 - 2^{-(2n_i+1)}] + y2^{-n_i} \\ v &= -x^{-n_i} + y[1 - 2^{-(2n_i+1)}] \end{aligned} \tag{A4}$$

so multiplication by $\sin\theta$ is obtained by right shifting the data by n_i bits and by $\cos\theta$ is obtained by right shifting the data by $(2n_i + 1)$ bits and subtracting it from the original data.

The block diagram implementing equation (A4) is shown in Fig. A1. From equation (A3) it is seen that the Givens rotor is obtained by cascading m such elementary blocks, as shown in Fig. A2.

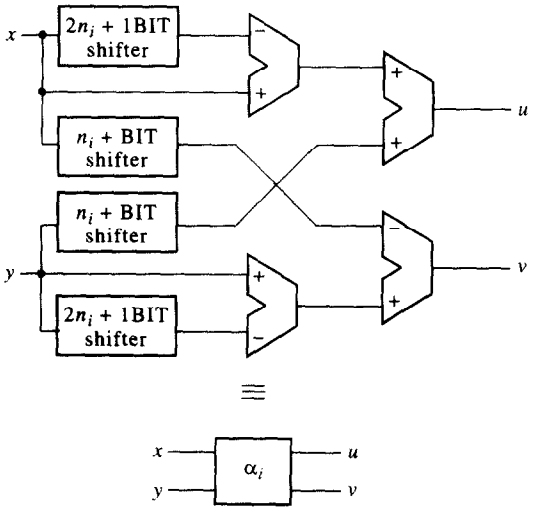

Fig. A1. Block diagram of elementary rotational stage.

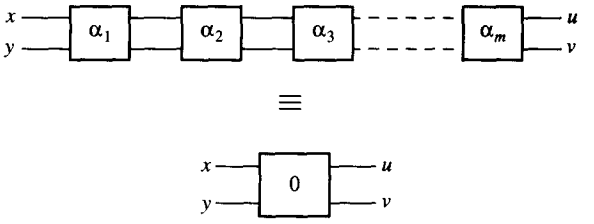

Fig. A2. Givens rotor.

AUTHORS' BIOGRAPHIES

Manik Chandra Mondal received his B.E. degree in Electronics & Tele-Communication Engineering in 1991 from Bengal Engineering College, University of Calcutta, India. In 1994 he received his M. Tech. degree in Integrated Circuits and Systems Engineering from the Indian Institute of Technology, Kharagpur. At present he is working as a design engineer in SGS Thompson Microelectronic (India) Pvt Ltd.

Anindya Sundar Dhar received his B.E. degree in Electronics and Telecommunication Engineering in 1987 from Bengal Engineering College, University of Calcutta, India. In 1988 he received his M. Tech. degree in Integrated Circuits and Systems Engineering from the Indian Institute of Technology, Kharagpur where he is working for his Ph.D. degree. His research interests include signal processing for biomedical applications and array architectures.

Swapna Banerjee received her B.E. and M.E. degree in Electronics and Telecommunication Engineering from Jadavpur University, India in 1971 and 1974 respectively. In 1982 she received her Ph.D. degree from the Indian Institute of Technology, Kharagpur. Since 1981 she has been with the Department of Electronics & Electrical Communication Engineering in the same Institute. At present she is an Associate Professor. Her research interests include device modelling, CAD and array architecture.