A Fast Radix-4 Algorithm and Architecture for DHT
Gautam Abhaychand Shah, Tejmal Saubhagyamal Rathore
Department of Electronics and Telecommunications, Mumbai University
St. Francis Institute of Technology, Borivali (West), Mumbai, India
E mail: gautamshah@ieee.org, tsrathor@ee.iitb.ac.in

Abstract—The radix-4 decimation-in-time fast Hartley transform and algorithm for DHT was introduced by Bracewell. A set of fast algorithms were further developed by Sorensen et al. In this paper, a fast radix-4 decimation-in-time algorithm that requires less number of multiplications and additions is proposed. It utilizes four different structures in the signal flow diagram. It exhibits a recursive pattern and is modular. The operational counts for the proposed algorithm are determined and verified by implementing the program in C. An analog architecture to implement the algorithm is proposed. The validity of the same is tested by simulating it with the help of the Orcad PSpice.

Keywords—Algorithm; Decimation-in-time; decimation-in-frequency; discrete Hartley transform; analog architecture; radix-4.

1. Introduction

Over the years, Discrete Hartley transform (DHT) has established as a potential tool for signal processing applications [1]-[3]. DHT is attractive mainly due to its real-valued kernel and the forward and inverse transforms are identical [4]. The seed for Hartley transform (HT) was sown by Hartley [5] in 1942. Bracewell [6] introduced a discretized version of HT and demonstrated that decimation-in-time (DIT) DHT resembles DFT. Several algorithms for its fast computation and opinions regarding them are reported. Meckelburg and Lipka presented decimation-in-frequency (DIF) FHT algorithm [7] claiming it to be faster than the one in [6]. Prado [8] presented an in-place version of FHT along with its operational complexity. The signal flow diagram originally proposed in [6] is restructured for clarity, and by applying the transposition theorem a DIF algorithm is obtained by Kwong and Shiu [9] having the same operational complexity. Hou [10] concluded that FHT algorithm, in essence, is a generalization of Cooley-Tukey FFT algorithm, but requires only real, as compared to complex, arithmetic operations in any standard FFT. Malvar [11] presents a new factorization of DHT which involves discrete cosine transform. His algorithms minimize the multiplications at the expense of an increased number of additions. Hao [12] examines both the pre- and post-permutation algorithms in [6] and [7] and suggests improvements to make them faster by use of fast rotation to reduce the multiplications and by incorporation of in-place or distributed permutation. Rathore [13] reports that, for both the DIT in [6] and the DIF in [7], the operational complexity involved is the same. He further utilizes the matrix approach, derives some properties of DHT [14], obtains the relations for computational complexity and presents DHT-based DFT and DFT-based DHT algorithms. The algorithms mentioned above are radix-2 algorithms.

Bracewell [15] further explored HT, its relation with FT, theorems, properties, matrix formulation, fast algorithms and suggested the radix-4 FHT algorithm which performs DHT in a time proportional to \( N \log N \) where \( N = 4^r \) and is computationally faster than radix-2 FHT. Sorensen et al. [16] further analyzed the radix-4 FHT algorithm having the same decomposition as [15], using the index mapping approach, implemented the algorithm and verified the operational complexities obtained by both the approaches are the same.

Various analog architectures have been reported in the literature to compute DHT. Culhane et al. [17] presented an analog circuit which utilizes a linear programming neural net to compute DHT. Raut et al. [18] presented basic switched capacitor building blocks in systolic array architecture to implement DFT. A two dimensional DCT structure proposed by Kawahito et al. [19] has been designed with fully differential switched-capacitor circuits. Digitally controlled analog circuits have been proposed by Chen et al. [20] which utilize the principle of charge scaling for computing DCT and DFT. Mal and Dhar [21] proposed an analog sampled data architecture for DHT. Analog circuits based on the current feedback operational amplifier (CFA) technique are suitable for
high frequency applications [22]-[25]. It combines high bandwidth and very fast large signal response. It can be used in place of traditional operational amplifiers (OA) and its current feedback architecture results in much better performance. It provides a closed-loop bandwidth determined primarily by the feedback resistor and almost independent of the closed-loop gain unlike OA-based circuits, which are limited by a constant gain-bandwidth product. It is free from the slew rate limitations inherent in traditional OAs. It can be used in ways similar to a conventional OA while providing performance advantages in wideband applications.

In this paper, a fast radix-4 FHT algorithm is proposed. It is a modified version of the algorithm in [15] and utilizes four different butterfly structures to implement the algorithm and reduces the operational complexity. The design of basic analog circuits based on CFAs and an architecture for the DHT are also proposed.

2. Discrete Hartley Transform

Set An N-point DHT $X_m$ of a sequence $x(n)$ is defined as

$$X_m(k) = \sum_{n=0}^{N-1} x(n) \cos \left( \frac{2\pi nk}{N} \right), \quad k = 0, 1, \ldots, N-1 \quad (1)$$

where $\cos(.) = \cos(.) + \sin(.)$.

The radix-4 DIT FHT is derived by Sorensen et al. as

$$X_m(k) = X_{4\alpha}(k) + \cos \left( \frac{2\pi k}{N} \right) X_{4\alpha+1}(k) + \sin \left( \frac{2\pi}{N} \right) X_{4\alpha+2}(N-k) + \cos \left( \frac{2\pi k}{N} \right) X_{4\alpha+3}(N-k), \quad k = 0, 1, \ldots, N-1 \quad (2)$$

where $X_{4\alpha}$ is a length $N/4$ DHT. Their program implements the FHT algorithm with an operation count of $N_a$ additions and $N_m$ multiplications given by

$$N_a = \frac{(11N\log_2 N - 5N + 8)}{4},$$

$$N_m = \frac{6N\log_2 N - 14N + 20}{6} \quad (3)$$

3. The Proposed Algorithm

The proposed algorithm utilizes four different butterfly structures as shown in Figs. 1 – 4.

![Figure 1. Structure A](image)

![Figure 2. Structure B](image)

![Figure 3. Structure C](image)

![Figure 4. Structure D](image)

Structure-A is a 4-point structure consisting of a summing structure which performs 8 additions. Structure-B is an 8-point structure consisting of a multiplying structure and a summing structure. The former structure performs 2 multiplications and the latter structure performs 14 additions. The multiplications by $\sqrt{2}$ are performed by the former structure. An 8-point structure-C consists of a multiplying structure which performs 10 multiplications and 6 additions and a summing structure which performs 16 additions. The multiplications by the sine (S) and cosine (C) coefficients and their related additions are performed by the multiplying structure. An
8-point structure-D is a modification of structure-C. The multiplying structure is modified and performs 12 multiplications and 6 additions whereas the summing structure is the same and performs 16 additions.

The proposed algorithm for an N-point DHT has \( P \) stages. The algorithm is viewed as a combination of the above structures connected in a recursive manner. The flow of the algorithm for various values of \( P \) is depicted in Fig. 5. For \( P = 1 \), structure-A is utilized. It is designated as E1. For \( P = 2 \), E1 is repeated four times in the first stage where as B and C are added to form the second stage. The input sequence is permuted to rearrange the data in a radix-4 permutation and applied as input \( x \). There is a reshuffle block which rearranges the data as it moves from the first stage to the second. The 0\(^{th}\) and 2\(^{nd}\) data element of each of the E1 blocks are applied as inputs to the B-block in consecutive, where as the 1\(^{st}\) and 3\(^{rd}\) data element of each of the E1 blocks are applied as inputs to the C-block in consecutive. The transformed output \( X \) is obtained as the even set at B and the odd set at C block, respectively.

**Figure 5.** Flow of the Algorithm for (a) \( P = 1 \), (b) \( P = 2 \)

As \( P \) increases, for \( N = 4^\ell \), EP is obtained by utilizing ‘E(P-1)’ four times to form the initial stages where as B, C and D are added to form the last stage. In the last stage, one B, one C and \( \left( \frac{4^\ell}{16} - 2 \right) \) D blocks are added. B is assigned 0, C is assigned \( \frac{4^\ell}{16} \), and D are assigned from 1 to \( \left( \frac{4^\ell}{16} - 1 \right) \), and from \( \left( \frac{4^\ell}{16} + 1 \right) \) to \( \left( \frac{4^\ell}{8} - 1 \right) \). These blocks are arranged in an ascending order. The flow of the algorithm is as depicted in Fig. 6. The input sequence is permuted to rearrange the data in a radix-4 permutation and applied as input \( x \) to the E(P-1) blocks. The reshuffle block rearranges the data as it moves from E(P-1) blocks to B, C and D blocks of the last stage. The sequence is indicated within the blocks representing the structures. The output sequence \( X \) is obtained as shown in B C and D blocks.

**Figure 6.** Flow of the Algorithm for \( N = 4^\ell \)

### 4. Operational Counts

An \( N \)-point DHT requires \( 4^{\ell-1} \) A-blocks each of which in turn perform 8 additions. The total number of additions performed by A-blocks is \( 4^{\ell-1} \times 8 = 2N \). The number of B-blocks required depends on the number of stages. For each stage \( L \) from 2 to \( P \), the number of blocks required are \( 4^{\ell-L} \). Each B-block performs 14 additions and 2 multiplications. The total number of additions and multiplications performed by B-blocks are \( \sum_{L=2}^{P} 4^{\ell-L} \times 14 \) and \( \sum_{L=2}^{P} 4^{\ell-L} \times 2 \) respectively. The number of C-blocks required depends on the number of stages. For each stage \( L \) from 2 to \( P \), the number of C-blocks required are \( 4^{\ell-L} \). Each C-block performs 22 additions and 10 multiplications. The total number of additions and multiplications performed by the C-blocks are \( \sum_{L=2}^{P} 4^{\ell-L} \times 22 \) and \( \sum_{L=2}^{P} 4^{\ell-L} \times 10 \) respectively. The number of D-blocks required depends on the number of stages. For each stage \( L \) from 3 to \( P \), the number of D-blocks required are \( \left( \frac{4^\ell}{8} - 2 \right) \times 4^{\ell-L} \). Each D-block performs 22 additions and 12 multiplications. The total number of additions and multiplications performed by the D-blocks are \( \sum_{L=3}^{P} \left( \frac{4^\ell}{8} - 2 \right) \times 4^{\ell-L} \times 22 \) and \( \sum_{L=3}^{P} \left( \frac{4^\ell}{8} - 2 \right) \times 4^{\ell-L} \times 12 \) respectively.

Thus, the operational counts considering all the structures are
\[ N_s = 2N + \sum_{i=2}^{N} (4^{r-1} \times 14) \left( \frac{4^i}{8} - 1 \right) \times 4^{r-2} \times 22 \]

\[ N_w = \sum_{i=2}^{N} (4^{r-1} \times 2) + \left( \frac{4^i}{8} - 2 \right) \times 12 \times (10) \times 4^{r-2} \quad (4) \]

On simplification, the operational counts are obtained as

\[ N_s = \frac{33N \log_2 N - 17N + 32}{12} \]

\[ N_w = \frac{3N \log_2 N - 5N + 8}{2} \quad (5) \]

It can be seen from (3) and (5) that both \( N_s \) and \( N_w \) are lesser for \( N > 4 \) for the proposed algorithm than those in [15, 16]. Both \( N_s \) and \( N_w \) are reduced by \( \left( \frac{N - 4}{6} \right) \) for \( N > 4 \). For \( N = 4 \), they are the same. Table I summarizes the comparison for different values of \( N \).

**TABLE I. Comparison of Operational Counts**

<table>
<thead>
<tr>
<th>Length</th>
<th>Radix-4 FHT algorithms [15][16]</th>
<th>Proposed Radix-4 algorithm</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>( N_s )</td>
<td>( N_w )</td>
</tr>
<tr>
<td>4</td>
<td>0</td>
<td>8</td>
</tr>
<tr>
<td>16</td>
<td>14</td>
<td>70</td>
</tr>
<tr>
<td>64</td>
<td>142</td>
<td>450</td>
</tr>
<tr>
<td>256</td>
<td>942</td>
<td>2498</td>
</tr>
<tr>
<td>1024</td>
<td>5294</td>
<td>12802</td>
</tr>
<tr>
<td>4096</td>
<td>27310</td>
<td>62466</td>
</tr>
</tbody>
</table>

The proposed algorithm with \( N = 16 \) is illustrated by the SFD as shown in Fig. 7. As \( P = 2 \) it has 2 stages. The first stage consists of \( 4^{r-1} \) A-blocks. The second stage consists of \( 4^{r-1} \) B-blocks, \( 4^{r-2} \) C-blocks and \( \left( \frac{4^i}{8} - 2 \right) \times 4^{r-2} \) D-blocks. These turn out to be \( 4^{r-1} = 4^{r-3} = 4 \) A-blocks in the first stage. The second stage consists of \( 4^{r-2} = 4^{r-3} = 4^6 \) B-block, \( 4^{r-1} = 4^{r-3} = 4^5 \) C-block and \( \left( \frac{4^i}{8} - 2 \right) \times 4^{r-3} = \left( \frac{4^2}{8} - 2 \right) \times 4^{r-3} = 0 \times 1 = 0 \) or no D-block.

The input sequence is permuted to rearrange the data in a radix-4 permutation. It is then applied as input to the A blocks also termed as E1 blocks. The outputs of these blocks are applied to the inputs of the B block and C block in accordance with the RESHUFFLE block as shown in Fig. 6. The 0th and \( \left( \frac{4^{r-1}}{2} \right) = \left( \frac{4^{10}}{2} \right) = 2^{nd} \) data element of each of the E1 blocks are applied as inputs to the B block where as the 1st and \( 4^{r-1} - 1 = 4^{7} - 1 = 3^{rd} \) data element are applied as inputs to the C block. The transformed output \( X \) is obtained as the even set at the B and the odd set at the C block respectively.

**Figure 7. SFD for proposed algorithm with \( N = 16, P = 2 \)**
5. Design of CFA Based Analog Circuits

An analog circuit for a summing structure is shown in Fig. 8.

![Figure 8](image)

**Figure 8.** Basic analog circuit for summing structure in A, B, C and D

The output is given by

\[
V_o = F \left[ 1 + \frac{R_3}{R_4} \right] - \left( \frac{V_{A1}}{R_{A1}} + \frac{V_{A2}}{R_{A2}} \right) R_f
\]

(6)

where

\[
F = \left[ R_{A1} \frac{R_{A1} \times R_3}{R_{A1} + R_{A1} \times R_3} \right] + \left[ R_{A2} \frac{R_{A2} \times R_3}{R_{A2} + R_{A2} \times R_3} \right] + \left[ R_{A3} \frac{R_{A3} \times R_3}{R_{A3} + R_{A3} \times R_3} \right] + \left[ R_{A4} \frac{R_{A4} \times R_3}{R_{A4} + R_{A4} \times R_3} \right]
\]

Thus, the circuit acts as a weighted summer and subtractor suitable for the summing structure in A, B, C and D. Structure A and B, C, D can be implemented using 4 and 8 such circuits, respectively. The combination of inputs and values of resistors for obtaining the various outputs required in the different structures with \(R_{A1} = R\) are as shown in Table II.

An analog circuit for a multiplying structure in structure-B is shown in Fig. 9.

![Figure 9](image)

**Figure 9.** Basic analog circuit for multiplying structure in B.

The output is given by

\[
V_o = \left( 1 + \frac{R_f}{R_3} \right) V_i
\]

(7)

Thus, it is suitable for the multiplying structure in B.

An analog circuit for a multiplying structure in structures C and D is shown in Fig. 10.

![Figure 10](image)

**Figure 10.** Basic analog circuit for multiplying structure in C and D.

The outputs are given by

\[
V_i = \left( \frac{R_f + R_3}{R_f + R_3} \right) \left[ \frac{R_f}{R_3} V_i + \frac{R_3}{R_f} V_s \right]
\]

### Table II. Combination of Inputs and Resistor values for Obtaining the Transform

<table>
<thead>
<tr>
<th>Inputs</th>
<th>Values of Resistors</th>
<th>Outputs</th>
</tr>
</thead>
<tbody>
<tr>
<td>(V_{A1})</td>
<td>(V_{A2})</td>
<td>(V_{A3})</td>
</tr>
<tr>
<td>✔ ✔ ✔ ✔ × ×</td>
<td>✔ ✔ ✔ ✔</td>
<td>R R R</td>
</tr>
<tr>
<td>✔ ✔ × × × ×</td>
<td>✔ ✔ ✔ ✔</td>
<td>R R ∞</td>
</tr>
<tr>
<td>✔ ✔ × × × ✔</td>
<td>✔ ✔ ✔ ✔</td>
<td>R ∞ ∞ 0.500 R</td>
</tr>
<tr>
<td>✔ × × × ✔ ✔</td>
<td>✔ ✔ ✔ ✔</td>
<td>∞ ∞ ∞ 0.333 R</td>
</tr>
<tr>
<td>✔ ✔ × × ✔ ×</td>
<td>✔ ✔ ✔ ✔</td>
<td>R ∞ ∞ R</td>
</tr>
</tbody>
</table>
\[
V_2 = \frac{R_o}{R_i} \left( 1 + \frac{R_o}{R_i} \right) V_i - \frac{R_o}{R_i} V_{y}
\]

Choosing \( R_i = R_o = R_2 = 1 \) and \( R_5 = \frac{R_1}{R_2} = 1 \)

\[
V_1 = V_4 + V_y, \quad V_2 = V_4 - V_y
\]

Further, choosing

\[
\frac{R_2 + R_4}{R_2 + R_4} = 1, \quad R_4 = R_5 = S, \quad R_6 = C \quad \text{and} \quad R_7 = \left( \frac{S + 1}{C} \right) - 1,
\]

\[
V_1 = SV_4 + CV_y, \quad V_2 = CV_4 - SV_y
\]

Thus, it is suitable for the multiplying structure in C and D.

6. Analog Architecture and its Simulation

The basic analog circuits are appropriately combined to form structures A, B, C and D as shown in Figs. 11, 12 and 13.

**Figure 11. Circuit for structure A**

These structures can be directly mapped into the SFD shown in Fig. 7 and provide an analog architecture for easy implementation of the algorithm. These structures are interconnected to obtain radix-4 DHT for \( N = 16 \). The architectures for both the forward and inverse transformations are identical. They have been tested by simulating them with the help of Orcad PSPice. The original input sequence denoted as \( V(XN) \) is applied to the forward transformation and its output \( V(YN) \) is given as input to the inverse transformation. The original sequence is retrieved at the output of the inverse transformation as \( V(ZN) \). These sequences are shown in Figs. 14, 15 and 16 respectively.
7. Results

Programs for Bracewell’s and the proposed algorithm are executed in C to compute \( X_M \). They have been tested by applying different types of sequence patterns such as step, ramp, impulse and sinusoidal and are observed to give the desired output sequences. These output sequences are applied as input to the same program to obtain the inverse transformation and is observed to yield the original sequence pattern, thus verifying both the forward and inverse transformations.

The simulation of the analog architecture using the proposed radix-4 algorithm for \( N = 16 \) has been performed. The theoretically calculated and the outputs obtained by simulation for the forward and inverse transformations are tabulated in Table III.

**TABLE III.** Comparison of Theoretical Values and Simulation Results

<table>
<thead>
<tr>
<th>( n )</th>
<th>Input ( x(n) ) (mV)</th>
<th>Forward transformation output ( X_M ) (mV)</th>
<th>Inverse transformation output ( x(n) ) (mV)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Theoretical Values</td>
<td>Simulation Results</td>
</tr>
<tr>
<td>0</td>
<td>10</td>
<td>340.00</td>
<td>340.00</td>
</tr>
<tr>
<td>1</td>
<td>20</td>
<td>-120.55</td>
<td>-120.00</td>
</tr>
<tr>
<td>2</td>
<td>30</td>
<td>-68.28</td>
<td>-68.20</td>
</tr>
<tr>
<td>3</td>
<td>40</td>
<td>-49.93</td>
<td>-49.80</td>
</tr>
<tr>
<td>4</td>
<td>50</td>
<td>-40.00</td>
<td>-40.00</td>
</tr>
<tr>
<td>5</td>
<td>60</td>
<td>-33.36</td>
<td>-33.40</td>
</tr>
<tr>
<td>6</td>
<td>70</td>
<td>-28.28</td>
<td>-28.30</td>
</tr>
<tr>
<td>7</td>
<td>80</td>
<td>-23.98</td>
<td>-24.00</td>
</tr>
<tr>
<td>8</td>
<td>90</td>
<td>-20.00</td>
<td>-20.00</td>
</tr>
<tr>
<td>9</td>
<td>100</td>
<td>-16.02</td>
<td>-16.00</td>
</tr>
<tr>
<td>10</td>
<td>110</td>
<td>-11.72</td>
<td>-11.70</td>
</tr>
<tr>
<td>11</td>
<td>120</td>
<td>-66.64</td>
<td>-66.68</td>
</tr>
<tr>
<td>12</td>
<td>130</td>
<td>00.00</td>
<td>00.00</td>
</tr>
<tr>
<td>13</td>
<td>140</td>
<td>9.99</td>
<td>9.99</td>
</tr>
<tr>
<td>14</td>
<td>150</td>
<td>28.28</td>
<td>28.40</td>
</tr>
<tr>
<td>15</td>
<td>160</td>
<td>80.55</td>
<td>80.60</td>
</tr>
</tbody>
</table>
It can be seen that the output of the forward transformation is within 2% of the theoretical value and the retrieved sequence is within 2% of the original sequence.

8. Conclusions

The proposed algorithm has been obtained by modifying Bracewell’s algorithm. As a result it requires lesser number of additions and multiplications. Its recursive structure allows generating the next higher order transforms from the lower order ones. It can be directly mapped into the SFD and provides a regular structure for easy implementation using the proposed analog circuits. The analog architecture utilizing these circuits is modular and can be scaled for large values of \( N \) unlike the neural network approach in [17]. It processes the data simultaneously at each stage and speeds up the transformation as compared to those which employ a multiply and accumulate approach as in [21]. Both the forward and inverse transformations have been validated by performing the simulation on Orcad PSpice and the results are in good agreement with those obtained theoretically.

References


AUTHORS PROFILES

Gautam Abhaychand Shah was born in Dhule on Oct. 30, 1968. He received the B.E. (Electronics Engineering) from Walchand Institute of Technology, Shivaji University, Solapur, Maharashtra, India in 1990, M.E. (Digital Techniques and Instrumentation) from SGSITS, Devi Ahilya Vishwavidyala, Indore, Madya Pradesh, India in 1995 and is pursuing Ph.D at MPSTME, NMIMS University, Vile Parle, Mumbai.

He served the Electronics Department of SSVPS College of Engineering, Dhule from 1990 to 2001 in the capacity of Lecturer and Senior Lecturer. He then joined the Electronics & Telecommunication Department at St. Francis Institute of Technology, Borivali, as an Assistant Professor, and is currently working as an Associate Professor. He was In-charge Principal (July 2004 - February 2005) and In-charge Head of Electronics & Telecommunication Department (July 2005 - June 2006). He has published and presented 14 research papers in


[24] AD-844 data sheet, Monolithic operational amplifier, Analog Devices, Rev-C.

various national/international journals and conferences. His areas of teaching and research interest are Analog
Signal Processing, Filters, Digital Techniques, Microprocessors and Biomedical Engineering.

Mr. Shah is a Graduate Student Member of IEEE (USA), Fellow of IETE (India), Fellow of IE (India), Member of
ISTE (India). He has received the IETE K S Krishnan Memorial Award (2009) for his paper “Design,
Development and Applications of PC-Based Process Control Trainer for Automation” published in IETE
Technical Review.

T S Rathore was born in Jhabhua (M P, India) on Oct. 29, 1943. He received the B Sc (Electrical
Engineering), M E (Applied Electronics & Servomechanisms), and Ph D (by research on Passive
and Active Circuits) degrees in Electrical Engineering from Indore University, Indore, India in 1965, 1970 and 1975,
respectively.

He served SGSITS, Indore from 1965 to 1978 before joining the EE Department of IIT Bombay from where he
retired as a Professor on superannuation in June 2006. Currently, from July 2006, he is the Dean (R&D) and
Head of Electronics & Telecommunication Department at St. Francis Institute of Technology, Borivali.

He was a post-doctoral fellow (1983-85) at the Concordia University, Montreal, Canada and a visiting researcher at
the University of South Australia, Adelaide (March-June 1993). He was an ISTE visiting professor (2005-2007). He
has published and presented over 200 research papers in various national/international journals and conferences.
He has authored the book Digital Measurement

and Alpha Science International Pvt. Ltd., U K, 2003 and
translated in Russian language in 2004. He was the Guest
Editor of the special issue of Journal of IE on
Instrumentation Electronics (1992). He is a member on the
editorial boards of ISTE National Journal of Technical
Education and IETE Journal of Education. He has
witnessed, organized and chaired many national/ international conferences and in some he was also the
Chief Editor of the proceedings.

His areas of teaching and research interest are Analysis
and Synthesis of Networks, Electronic Circuit Design,
Switched-Capacitor Filters, Electronic-Aided Instrument-
ation, Hartley Transform, Signal Processing, Fault
Diagnosis and Knowledge-Based Systems.

Prof. Rathore is a Senior Member of IEEE (USA), Fellow
of IETE (India), Fellow of IE (India), Member of ISTE
(India), Member of Instrument Society of India, Member
of Computer Society of India. He has been listed in Asia’s
He has played a very active role as Fellow of IETE and
has served its Mumbai Centre as Volunteer member
(1997-98), Co-opted member (1998-99), Secretary (1999-
2001), Chairman (2001-02), Vice Chairman (2003-06) and
Chairman (2006-08).

He has received IETE M N Saha Memorial Award (1995),
IEEE Silver Jubilee Medal (2001), ISTE U P Government
National Award (2002), ISTE Maharashtra State National
Award (2003), IETE Prof S V C Aiya Memorial Award
(2004), IETE BR Batra Memorial Award (2005), IETE
Prof K Sreenivasan Memorial Award (2005), IETE K S
Krishnan Memorial Award (2009), IETE - Hari Ramji
Toshniwal Gold Medal Award (2010), and IETE best