

An Analog Architecture for Split-Radix DHT

Gautam A. Shah

Dept. of Electronics and Telecommunications
St. Francis Institute of Technology,
Borivali (West), Mumbai, India

Tejmal S. Rathore

Dept. of Electronics and Telecommunications
St. Francis Institute of Technology,
Borivali (West), Mumbai, India

ABSTRACT

The fast Hartley transform and algorithm for DHT was introduced by Bracewell. The split radix decimation-in-frequency algorithm that requires less number of operation counts as compared to the radix-2 and radix-4 algorithms was developed by Sorenson et al. In this paper, an analog architecture for a split radix decimation-in-time algorithm is proposed. It utilizes three different structures in the signal flow diagram. It exhibits a recursive pattern and is modular. The validity of the analog architecture is tested by simulating it with the help of the Orcad PSpice.

General Terms

Analog architecture, Discrete Hartley transform

Keywords

Decimation-in-time, Radix-2, Radix-4, Split-Radix

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 DIT in [6] and 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 $N \log_4 N$ where $N = 4^p$ and is computationally faster than radix-2 FHT. Sorenson et al. [16] further developed the decimation-in-frequency split-radix FHT algorithm using the index mapping approach, implemented the algorithm and verified the operational counts to be less than both radix-2 and radix-4 algorithms.

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 decimation-in-time split-radix FHT algorithm based on three different butterfly structures is proposed. The designs of basic analog circuits based on CFAs and an architecture for the split-radix FHT algorithm are also proposed.

2. PROPOSED ALGORITHM

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

$$X_H(k) = \sum_{n=0}^{N-1} x(n) \text{cas} \left(\frac{2\pi kn}{N} \right) \quad k = 0, 1, \dots, N-1 \quad (1)$$

where $\text{cas}(\cdot) = \cos(\cdot) + \sin(\cdot)$.

The split-radix DIF FHT for $N = 2^p$ is derived by Sorenson et al. as

$$X_H(2k) = \sum_{n=0}^{\frac{N}{2}-1} \left[x_n + x_{n+\frac{N}{2}} \right] \cos\left(\frac{2\pi}{N} 2kn\right), \quad 0 \leq k \leq \frac{N}{2}-1 \quad (2a)$$

$$X_H(4k+1) = \sum_{n=0}^{\frac{N}{4}-1} \left[\left(x_n - x_{n+\frac{N}{2}} + x_{\frac{N}{4}-n} - x_{\frac{3N}{4}-n} \right) \cos\left(\frac{2\pi}{N} n\right) + \left(x_{\frac{3N}{4}} - x_{\frac{N}{4}} + x_{\frac{N}{2}-n} - x_{N-n} \right) \sin\left(\frac{2\pi}{N} n\right) \right] \cos\left(\frac{2\pi}{N} 4kn\right), \quad 0 \leq k \leq \frac{N}{4}-1 \quad (2b)$$

$$X_H(4k+3) = \sum_{n=0}^{\frac{N}{4}-1} \left[\left(x_n - x_{n+\frac{N}{2}} + x_{\frac{3N}{4}-n} - x_{\frac{N}{4}-n} \right) \cos\left(\frac{2\pi}{N} 3n\right) - \left(x_{\frac{3N}{4}} - x_{\frac{N}{4}} - x_{\frac{N}{2}-n} + x_{N-n} \right) \sin\left(\frac{2\pi}{N} 3n\right) \right] \cos\left(\frac{2\pi}{N} 4kn\right), \quad 0 \leq k \leq \frac{N}{4}-1 \quad (2c)$$

In this paper, a split-radix DIT FHT is proposed. It utilizes three basic butterfly structures as shown in Figures 1(a) –(c) in a recursive manner. Structure- D_1 is a 2-point structure consisting of a summing structure (SS) which performs 2 additions. Structure- D_2 is an 4-point structure which utilizes 2 D_1 -structures and a COMBINE block consisting of an SS which performs 4 additions. Hence, structure- D_2 performs 8 additions. Structure- D_3 is an 8-point structure which utilizes one D_2 -structure, 2 D_1 -structures, a multiplication structure (MS) which performs 2 additions and 2 multiplications and a COMBINE block consisting of an SS which performs 8 additions. Hence, structure- D_3 performs 22 additions and 2 multiplications. The multiplications by the sine (S) and cosine (C) coefficients and their related additions are performed by MS. The proposed algorithm for an N -point DHT is viewed as a combination of the above structures connected in a recursive manner. The flow of the algorithm is depicted in Figure 2. D_p is obtained by utilizing ' $D_{(p-1)}$ ', 2 ' $D_{(p-2)}$ ', MS and COMBINE structures. The input sequence is permuted to rearrange the data in a radix-2 permutation and applied as input x . The outputs of the 2 $D_{(p-2)}$ structures are applied as input to the multiplying structure. The outputs of MS and $D_{(p-1)}$ structures are applied as input COMBINE block. The output sequence X is obtained at the output of the COMBINE block.

The radix-2 algorithm utilizes 2 previous level structures along with the MS and COMBINE structures to obtain the transform [26]. The split-radix utilizes 1 previous level and 2 'one before previous' level structures in the radix-2 and radix-4 decompositions respectively in a recursive manner along with the MS and COMBINE structures. This approach leads to a reduction in the operational counts over the radix-2 and radix-4 algorithms. The operational counts for structures D_1 and D_2 can be directly be obtained from Figures. 1 (a) and (b). The MS performs $(N-6)$ additions and multiplications. The COMBINE structure is a summing structure and performs N additions. The operational counts may thus be calculated in a recursive manner. For D_1 , $P=1$, $N_A=2$ and $N_M=0$ and for D_2 , $P=2$, $N_A=8$ and $N_M=0$. For structure- D_3 , $P=3$, $N_A=N_{A(p-1)}+2N_{A(p-2)}+(N-6)+N=8+(2 \times 2)+(8-6)+8=22$ and $N_M=N_{M(p-1)}+2N_{M(p-2)}+(N-6)=0+0+(8-6)=2$. In this manner, from the SFD recursive formulae for $P > 3$ are obtained as

$$N_A = N_{A(p-1)} + 2N_{A(p-2)} + 2N - 6 \quad (3)$$

$$N_M = N_{M(p-1)} + 2N_{M(p-2)} + N - 6 \quad (4)$$

It is found that both N_A and N_M for the proposed DIT split radix algorithm are the same as those reported for the DIF split-radix algorithm in [16] as expected. These operational counts are less than those obtained for the radix-2 and radix-4 algorithms. The proposed algorithm with $N=16$ is illustrated by the SFD as shown in Figure 3.

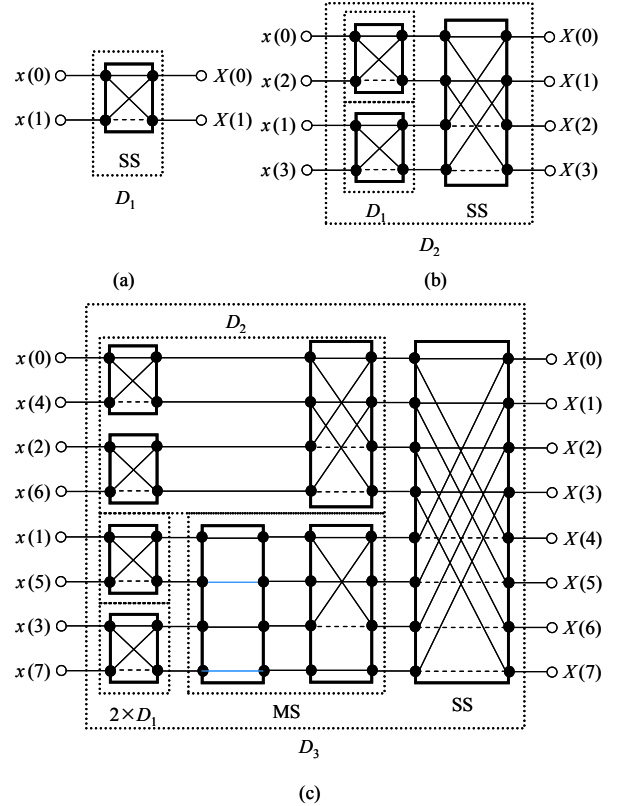


Fig 1: Structures (a) D_1 , (b) D_2 and (c) D_3

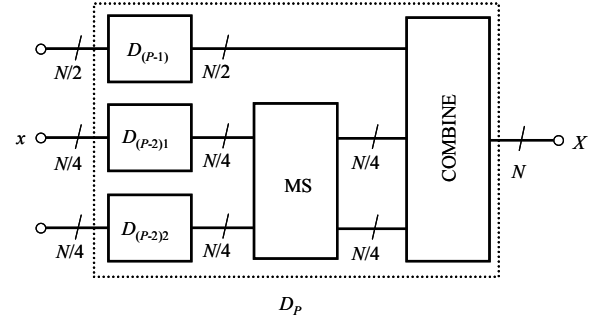


Fig 2: Flow of the algorithm

The input sequence is permuted to rearrange the data in a radix-2 permutation. It is then applied as input to the D_3 and 2 D_2 structures. The outputs of the 2 D_2 structures are applied as inputs to the multiplying structure. The outputs of D_3 and multiplying structure are applied as inputs to the COMBINE block. The transformed output X is obtained at the output of the COMBINE block.

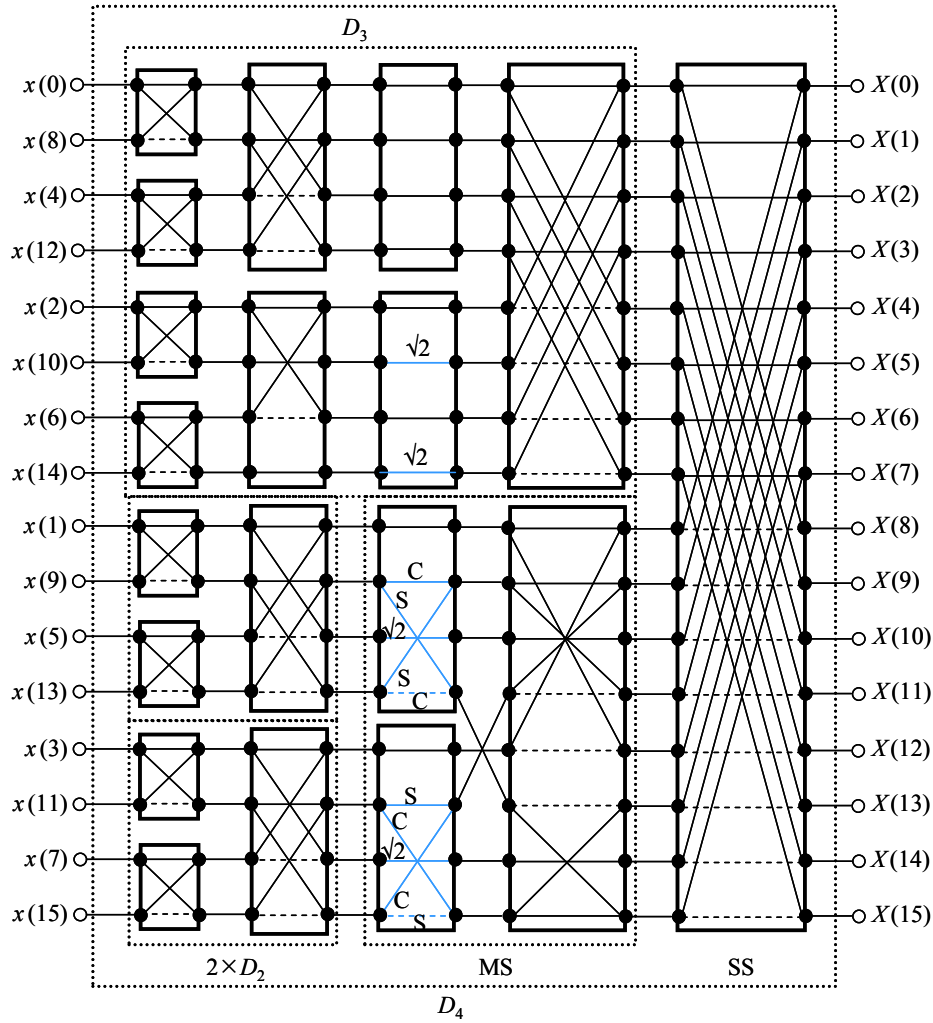


Fig 3: SFD for proposed algorithm with $N = 16, P = 4$

3. DESIGN OF CFA BASED ANALOG CIRCUITS

An analog circuit for a summing structure and multiplying structure that caters to the radix-2 decomposition is shown in Figure 4.

The outputs are given by

$$V_1 = \left(\frac{R_1 + R_2}{R_3 + R_4} \right) \left[\frac{R_2}{R_1} V_A + \frac{R_1}{R_1} V_B \right] \quad (5)$$

$$V_2 = \frac{R_6}{R_5} \left(\frac{1 + \frac{R_5}{R_6}}{\frac{R_7}{R_8} + 1 + \frac{R_7}{R_8}} \right) V_A - \frac{R_6}{R_5} V_B \quad (6)$$

Choosing $R_1 = R_2 = R_3 = R_4$ and $\frac{R_6}{R_5} = \frac{R_8}{R_7} = 1$,

$$V_1 = V_A + V_B, \quad V_2 = V_A - V_B \quad (7)$$

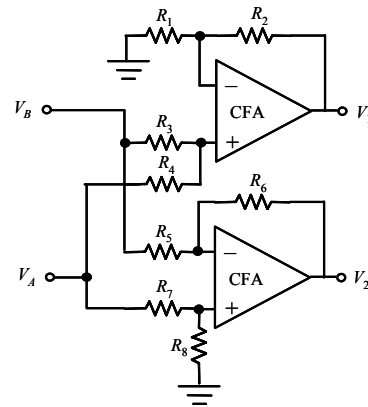


Fig 4: Basic analog circuit for summing and multiplying structure in radix-2 decomposition

Thus, it is suitable for the summing structure.

Further, choosing

$$\frac{R_1 + R_2}{R_3 + R_4} = 1, \frac{R_3}{R_1} = \frac{R_6}{R_5} = S, \frac{R_4}{R_1} = C \text{ and } \frac{R_7}{R_8} = \left(\frac{S+1}{C}\right) - 1,$$

$$V_1 = SV_A + CV_B, V_2 = CV_A - SV_B. \quad (8)$$

or choosing

$$\frac{R_1 + R_2}{R_3 + R_4} = 1, \frac{R_3}{R_1} = \frac{R_6}{R_5} = C, \frac{R_4}{R_1} = S \text{ and } \frac{R_7}{R_8} = \left(\frac{C+1}{S}\right) - 1,$$

$$V_1 = CV_A + SV_B, V_2 = SV_A - CV_B. \quad (9)$$

Thus, it is suitable for the multiplying structures.

Another analog circuit for a summing structure that caters to the radix-4 decomposition is shown in Figure 5.

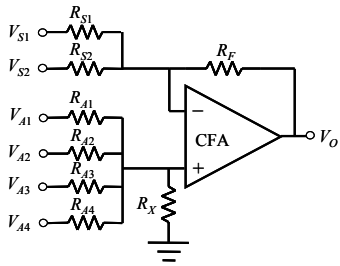


Fig 5: Basic analog circuit for summing structure in radix-4 decomposition

The output is given by

$$V_O = F \left[1 + \frac{R_F}{R_S} \right] - \left(\frac{V_{S1}}{R_{S1}} + \frac{V_{S2}}{R_{S2}} \right) R_F \quad (10)$$

where

$$F = \left[V_{A1} \frac{R_{A234X}}{R_{A1} + R_{A234X}} \right] + \left[V_{A2} \frac{R_{A134X}}{R_{A2} + R_{A134X}} \right] + \left[V_{A3} \frac{R_{A124X}}{R_{A3} + R_{A124X}} \right] +$$

$$\left[V_{A4} \frac{R_{A123X}}{R_{A4} + R_{A123X}} \right], R_S = R_{S1} \parallel R_{S2}, R_{A234X} = R_{A2} \parallel R_{A3} \parallel R_{A4} \parallel R_X,$$

$$R_{A134X} = R_{A1} \parallel R_{A3} \parallel R_{A4} \parallel R_X, R_{A124X} = R_{A1} \parallel R_{A2} \parallel R_{A4} \parallel R_X, \text{ and}$$

$$R_{A123X} = R_{A1} \parallel R_{A2} \parallel R_{A3} \parallel R_X.$$

Thus, the circuit acts as a weighted summer and subtractor suitable for the summing structure. The combination of inputs and values of resistors for obtaining the various outputs required in the different structures with $R_{A1} = R$ are as follows.

For obtaining the output $V_O = [V_{A1} + V_{A2} + V_{A3} + V_{A4}]/2$, the inputs are applied to V_{A1} , V_{A2} , V_{A3} and V_{A4} . The values of $R_{A2} = R_{A3} = R_{A4} = R$, $R_X = \infty$ and $R_S = R_{S1} \parallel R_{S2} = R_F$.

For obtaining the output $V_O = [V_{A1} + V_{A2} - (V_{S1} + V_{S2})]/2$, the inputs are applied to V_{A1} , V_{A2} , V_{S1} and V_{S2} . The values of $R_{A2} = R$, $R_{A3} = R_{A4} = \infty$, $R_X = 0.5R$, and $R_{S1} = R_{S2} = 2R_F$.

4. ANALOG ARCHITECTURE AND ITS SIMULATION

The basic analog circuit in Figure 4 forms the structure- D_1 . The basic analog circuits are appropriately combined to form structure- D_2 in radix-2 and radix-4 decompositions as shown in Figures. 6(a) and (b) respectively. Figure 7 depicts structure- D_3 in radix-2 decomposition and Figure 8 in a combined radix-2 and radix-4 decomposition. These structures can be directly mapped into the SFD and provide an analog architecture for easy implementation of the algorithm. These structures are interconnected to obtain split-radix 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 Figures. 9, 10 and 11 respectively.

5. RESULTS

Program for the proposed algorithm is executed in C to compute X_H . It has been tested by applying different types of sequence patterns such as step, ramp, impulse and sinusoidal and 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 split-radix algorithm for $N = 16$ has been performed. The theoretically calculated values and the outputs obtained by simulation for the forward and inverse transformations are tabulated in Table 1.

Table 1. Comparison of Theoretical Values and Simulation Results

| n | Input $x(n)$ (mV) | Forward transformation output X_H (mV) | | Inverse transformation output $x(n)$ (mV) |
|----|-------------------|--|--------------------|---|
| | | Theoretical Values | Simulation Results | |
| 0 | 0 | 0 | 0 | 0 |
| 1 | 383 | 2000 | 2000 | 384 |
| 2 | 707 | 0 | 0 | 709 |
| 3 | 924 | 0 | 0 | 928 |
| 4 | 1000 | 0 | 0 | 1000 |
| 5 | 924 | 0 | 0 | 928 |
| 6 | 707 | 0 | 0 | 709 |
| 7 | 383 | 0 | 0 | 384 |
| 8 | 0 | 0 | 0 | 001 |
| 9 | -383 | 0 | 0 | -381 |
| 10 | -707 | 0 | 0 | -705 |
| 11 | -924 | 0 | 0 | -924 |
| 12 | -1000 | 0 | 0 | -1000 |
| 13 | -924 | 0 | 0 | -924 |
| 14 | -707 | 0 | 0 | -705 |
| 15 | -383 | -2000 | -2000 | -381 |

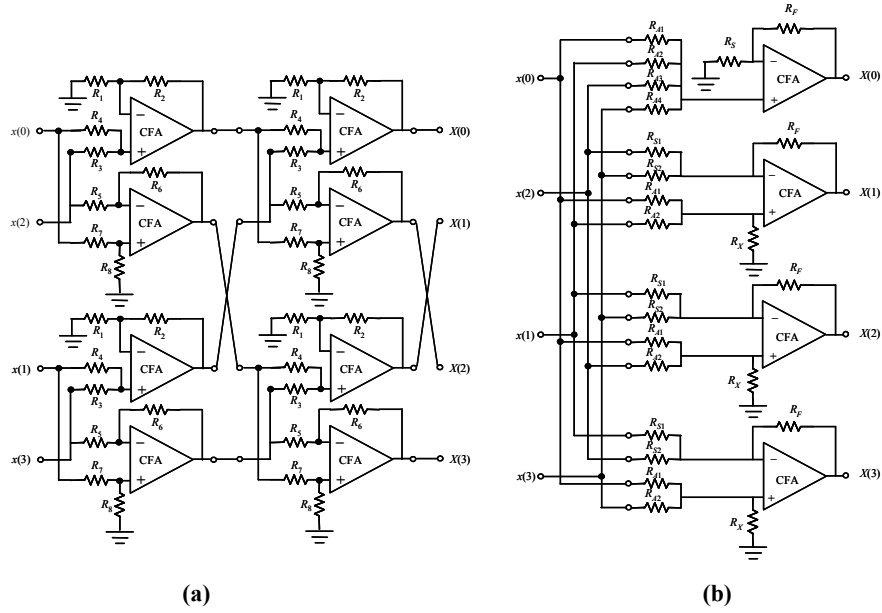


Fig 6: Circuits for structure- D_2 in radix-2 and radix-4 decompositions

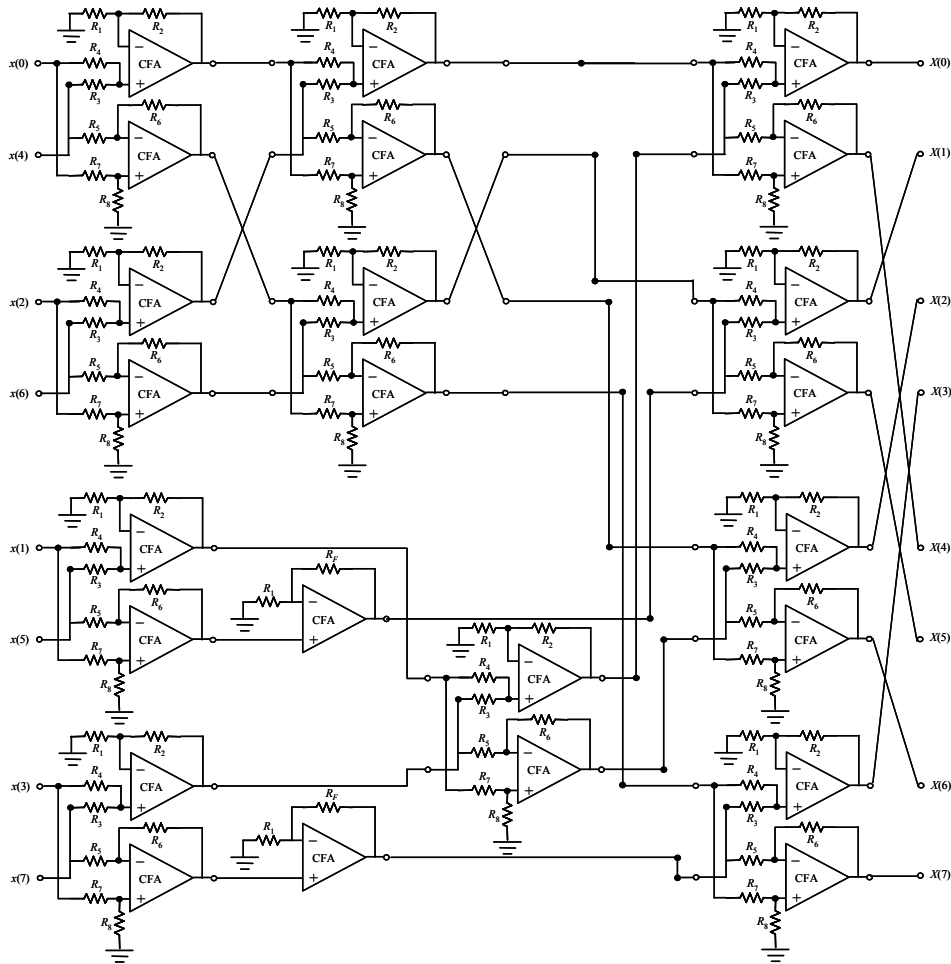


Fig 7: Circuit for structure- D_3 in radix-2 decomposition

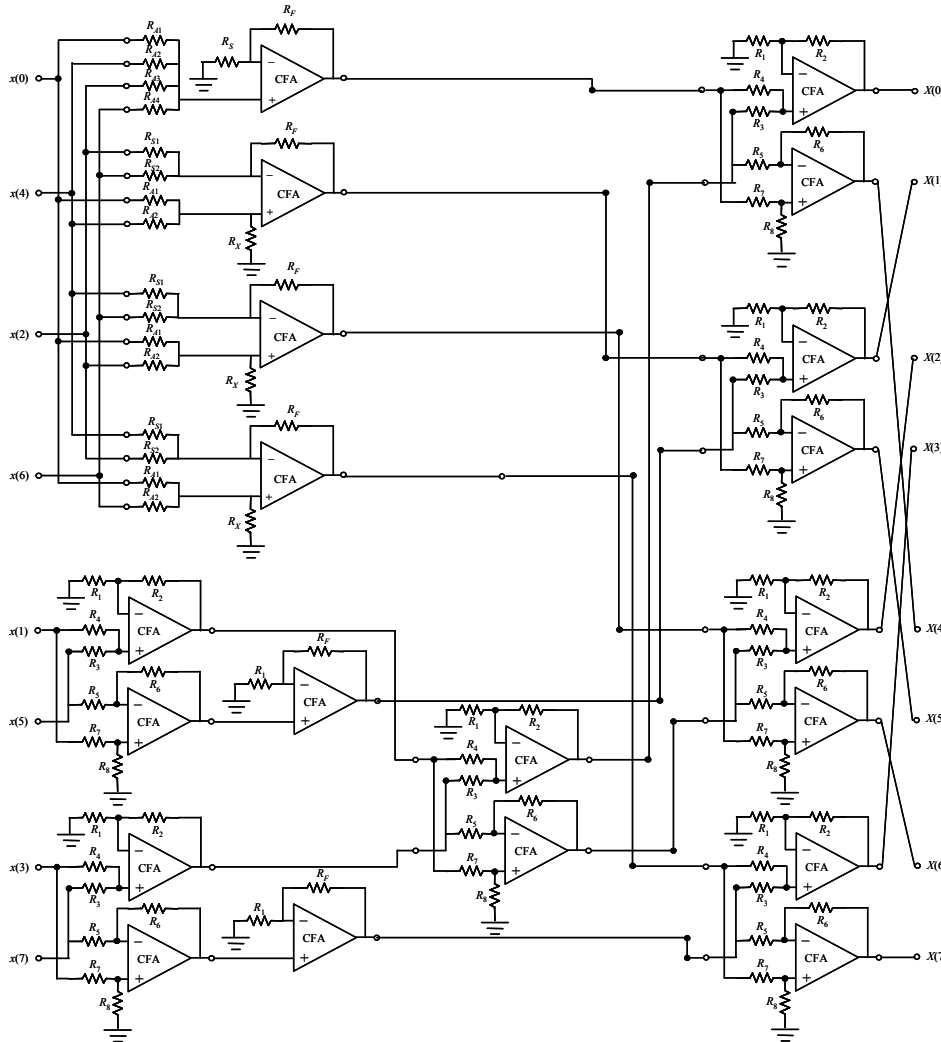


Fig 8: Circuit for structure- D_3 in combined radix-2 and radix-4 decomposition

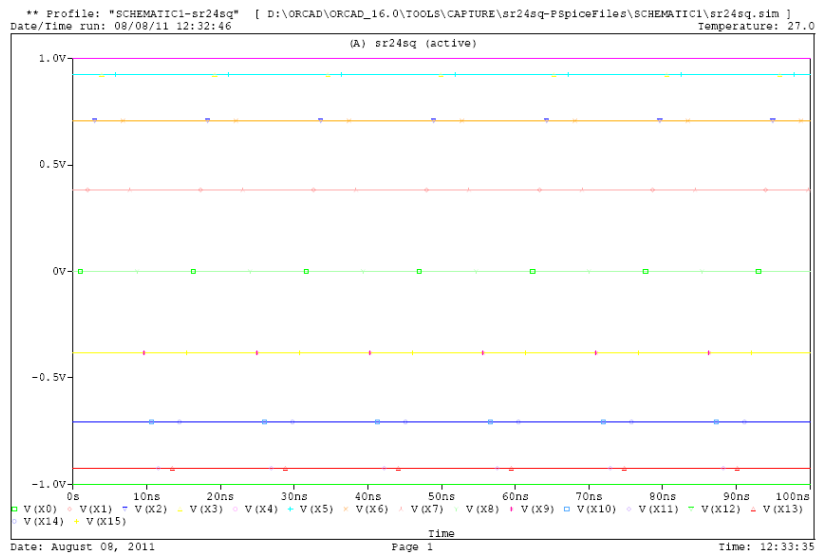


Fig 9: Input sequence

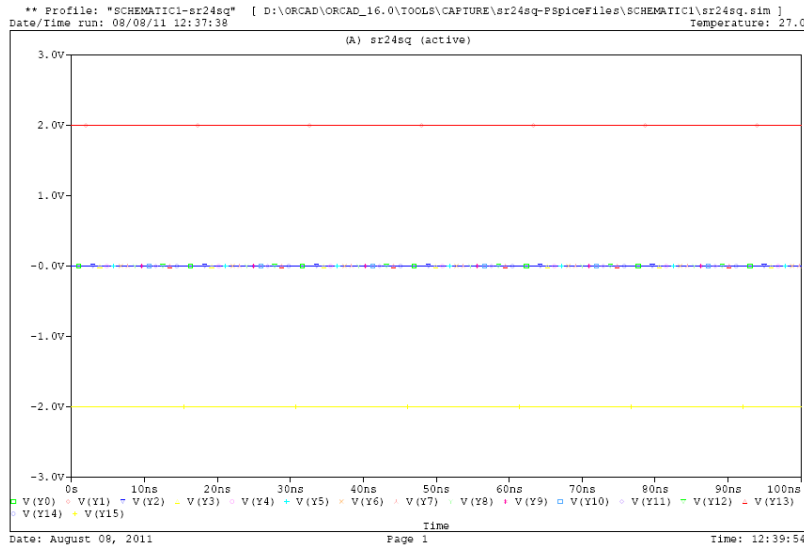


Fig 10: Ouput sequence after forward transformation

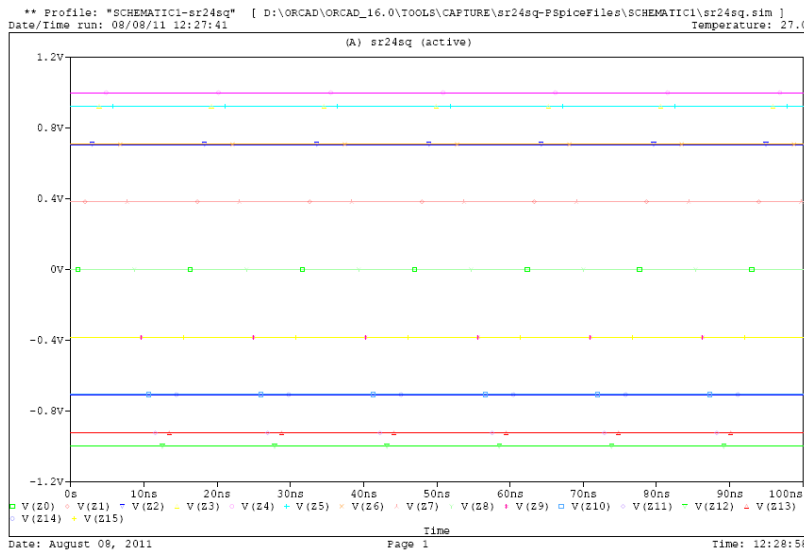


Fig 11: Retrived sequence after forward and inverse transformations

6. CONCLUSIONS

The proposed DIT split-radix algorithm has a recursive structure generates the next higher order transforms from the lower order ones. It requires less number of operational counts as compared to the radix-2 and radix-4 algorithms. 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 net 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.

7. REFERENCES

- [1] Paik, C. H. and Fox, M. D. 1988. Fast Hartley transform for image processing, IEEE Trans. Medical Imaging, vol. 7, no. 2, (June 1988), pp. 149-153.
- [2] Wu, J. L. and Shiu, J. 1991. Discrete Hartley transform in error control coding," IEEE Trans. Signal Processing, vol. 39, no. 10, (Oct. 1991), pp. 2356-2359.
- [3] Meher, P. K. and Panda, G. 1993. Unconstrained Hartley-domain least mean square adaptive filter," IEEE Trans. Circuits and Systems-II: Analog and Digital Signal Processing, vol. 40, no. 9, (Sep. 1993), pp. 582-585.
- [4] Meher, P. K., Srikanthan, T. and Patra, J. C. 2006. Scalable and modular memory-based systolic architectures for

- discrete Hartley transform, *IEEE Trans. Circuits Syst. I*, vol. 53, no. 5, (May 2006), pp. 1065–1077.
- [5] Hartley, R. V. L. 1942. A more symmetrical Fourier analysis applied to transmission problems, *Proc. IRE*, vol. 30, (Mar. 1942), pp. 144-150.
- [6] Bracewell, R. N. 1984. The fast Hartley transform,” *Proc. IEEE*, vol. 72, no. 8, (Aug. 1984), pp. 1010-1018.
- [7] Meckelburg, H. J. and Lipka, D. 1985. Fast Hartley transform algorithm, *Electronics Letters*, vol. 21, no. 8, (Apr. 1985), pp. 311-313.
- [8] Prado, J. 1985. Comments on “The fast Hartley transform,” *Proc. IEEE*, vol. 73, no. 12, (Dec. 1985), pp. 1862-1863.
- [9] Kwong, C. P. and Shiu, K. P. 1986. Structured fast Hartley transform algorithms, *IEEE Trans. Acoustics, Speech, and Signal Processing*, vol. ASSP-34, no. 4, (Aug. 1986), pp. 1000-1002.
- [10] Hou, H. S. 1987. The Fast Hartley Transform Algorithm, *IEEE Trans. Computers*, vol. C-36, no. 2, (Feb 1987), pp. 147-156.
- [11] Malvar, H. S. 1987. Fast computation of the discrete cosine transform and the discrete Hartley transform, *IEEE Trans. Acoustics, Speech, and Signal Processing*, vol. ASSP-35, no. 10, (Oct. 1987), pp. 1484-1485.
- [12] Hao, H. 1987. On fast Hartley transform algorithms, *Proc. IEEE*, vol. 75, no. 7, (July 1987), pp. 961-962.
- [13] Rathore, T. S. 1989. Recursive relations for complexities of Hartley transform algorithms, *IETE J. of Research*, vol. 35, no. 6, (Nov.-Dec 1989), pp. 357-359.
- [14] Rathore, T. S. 1990. Hartley transform – Properties and algorithms, In *Proceedings of the Nat. Conf. Real Time Systems*, Indore, (Nov. 1990), pp. 21-30.
- [15] Bracewell, R. N. 1986. *The Hartley transform*, New York: Oxford University Press.
- [16] Sorensen, H. V., Jones, D. L., Burrus, C. S. and Heideman, M. T. 1985. On computing the discrete Hartley transform, *IEEE Trans. Acoustics, Speech, and Signal Processing*, vol. ASSP-33, no. 4, (Oct. 1985), pp. 1231-1238.
- [17] Culhane, A. D., Peckerar, M. C. and Marrian, C. R. K. 1989. A neural net approach to discrete Hartley and Fourier transforms, *IEEE Trans. Circuits Syst.*, vol. 36, no. 5, (May 1989), pp. 695-703.
- [18] Raut, R., Bhattacharya, B. B. and Faruque, S. M. 1990. A discrete Fourier transform using switched-capacitor circuits in systolic array architecture, *IEEE Trans. Circuits Syst.*, vol. 37, no. 12, (Dec. 1990), pp. 1578-1580.
- [19] Kawahito, S. et al. 1997. A CMOS image sensor with analog two-dimensional DCT based compression circuits for on-chip cameras, *IEEE J. Solid-State Circuits*, vol. 32, no. 12, (Dec. 1997), pp. 2030-2041.
- [20] Chen, J., Shou, G., and Zhou, C. 1999. Digital-controlled analog circuits for weighted-sum operations, *IEICE Trans. Fundamentals*, vol. E82-A, no. 11, (Nov 1999), pp. 2505-2513.
- [21] Mal, A. K. and Dhar, A. S. 2003. Analog Sampled Data Architecture for Discrete Hartley Transform, In *Proceedings of the Tenth Int. Conf. on Convergent Technologies for Asia-Pacific Region*, Bangalore, India, vol. 3, (Oct. 2003), pp. 1035–1039.
- [22] Gift, S. J. G. and Maundy, B. 2005. Improving the Bandwidth Gain-Independence and Accuracy of the Current Feedback Amplifier, *IEEE Trans. Circuits Syst. II*, vol. 52, no. 3, (Mar. 2005), pp. 136-139.
- [23] Madian, A. H., Mahmoud, S. A. and Soliman, A. M. 2008. Field Programmable Analog Array based on CMOS CFOA and its Application, In *Proceedings of the IEEE Int. Conf. Electr. Circuits Syst.* (Aug.-Sept. 2008), pp. 1042–1046.
- [24] AD-844 data sheet, *Monolithic operational amplifier*, Analog Devices, Rev-C.
- [25] Rathore, T. S. and Khot, U. P. 2008. CFA-based grounded-capacitor operational simulation of ladder filters, *Int. J. Circ. Theor. Appl.*, vol. 36, no. 5-6, (July 2008), pp. 697-716.
- [26] Shah, G. A. and Rathore, T. S. 2011. A Mixed-Mode Signal Processing Architecture for Radix-2 DHT, *Int. J. Computer Science and Engineering*, vol. 3, no. 6, (June 2011), pp. 2554-2564.