Chapter 1

Introduction

Signals are used to communicate among human beings, and human beings and machines. They are used to probe the environment to uncover details of structure and state not easily observable, and to control and utilize energy and information [1]. A signal is any physical quantity that is a function of time, space or any other independent variable [2]. Although signals can be represented in many ways, in all cases the information is contained in some pattern of variations. Signals are represented mathematically as functions of independent variables. The independent variable may be either continuous or discrete. Continuous-time signals are defined along a continuum of times and thus represented by a continuous independent variable. These are often referred to as analog signals. Discrete-time signals are defined at discrete times, and thus, the independent variable has discrete values. Discrete time signals are represented as a sequence of numbers. Besides the independent variable being either continuous or discrete, the signal amplitude may either be continuous or discrete. Digital signals are those signals for which both time and amplitude are discrete. [1]

Signal processing is concerned with the representation and transformation of signals and the information they contain. Signal processing plays a major role in diverse fields as speech and data communication, biomedical engineering, acoustics, instrumentation and many others [1]. It has always benefited from a close coupling between the theory, applications and technologies for implementing signal processing systems. Initially, signal processing was typically
done with analog systems that were implemented with electronic circuits or even with mechanical devices. The technology was almost exclusively continuous time analog technology. The rapid evolution of digital computers and microprocessors caused a major shift to digital technologies giving rise to the field of digital signal processing (DSP). DSP is based on processing of sequences of samples. The discrete-time nature of DSP technology is also the characteristic of other signal processing technologies such as charge-coupled and switched-capacitor technologies [2]. Most applications involve the use of discrete-time technology for processing continuous-time signals, in which a continuous-time technology is converted into a sequence of samples and after discrete-time processing the output signal is converted back to a continuous-time signal.

The utility of discrete-time signal processing was accelerated by the Cooley and Tukey algorithm [3] for computation of Fourier transform (FT) which is known as fast Fourier transform (FFT). FFT is significant because other signal processing algorithms which were developed till then required processing time of several orders of magnitude greater than the real time. Discrete Fourier transform (DFT) plays an important role in the analysis, design and implementation of discrete-time signal processing algorithms. FFT algorithms are computationally efficient for evaluating DFT using the ‘divide and conquer approach’ [1].

Signal processing systems may be classified along the same lines as signals. Continuous-time systems are systems for which both the input and output are continuous time signals. Discrete-time systems are those for which both the input and output are discrete time signals. Similarly a digital system is a system for which both the input and output are digital signals.[1] While many types of signal processing systems have moved into the digital domain, analog circuits have proved fundamentally necessary. Naturally occurring signals are analog, at least at the macroscopic level [4]. The role of analog integrated circuits in modern electronic systems remains important, even though digital circuits dominate the market for VLSI solutions. Analog systems play an essential role in interfacing
digital electronics to the real world. An important advantage of digital ICs is their relative ease of design over analog circuits. In particular, since digital circuit design is amenable to automation, several CAD-compatible digital integrated circuit design methodologies were developed, including design-for-testability, design optimization and rapid prototyping in the field-programmable gate arrays [5].

The growing computational demand for complex information processing has motivated significant research in the design of power efficient signal processing systems. One can achieve low-power designs by moving processing on system inputs from the digital processor to analog hardware [6]. However, for analog systems to be desirable, they need to provide a significant advantage in terms of size and power. They should be easy to use and integrate into a larger digital system. Field-programmable analog arrays can speed the transition of systems from digital to analog by providing the ability to rapidly implement advanced, low-power and reconfigurable signal processing systems [6].

As the device size shrinks, speed increases, fabrication techniques get better, supply voltage magnitudes drop, power dissipation reduces, and analog and digital circuits get fabricated on the same chip, there is a significant impact on the system design. Mixed-Mode signal processing is gaining importance. The designs involve analog circuits mixed with digital control to meet the application requirements.

## 1.1 Background

The seed for the discrete Hartley transform (DHT) was sown by Hartley [7] in 1942. Hartley recognized that the complex kernel of FT could be replaced by one involving the sum of a cosine and sine, a function called cas. Both transforms FT and Hartley transform (HT) convey information regarding the harmonic content, with the most important difference involving the method by which the
information is presented. FT uses a complex kernel where as HT a real one. Bracewell [8] introduces a discretized version of HT and demonstrates that DHT resembles DFT. Nevertheless, DFT is directly obtainable from DHT by simple additive operation. The properties of DHT commend themselves for application to numerical analysis and all the operations normally carried out using FT can also be performed using HT.

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

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

(1.1)

where $\text{cas} (\cdot) = \cos (\cdot) + \sin (\cdot)$. The inverse relation is

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

(1.2)

### 1.1.1 Algorithms

An algorithm for HT analogous to FFT is the fast Hartley transform (FHT) algorithm [9]. This actually changed the way people looked at HT. This led to an opening for many researchers to develop algorithms for computing DHT. FHT performs DHT in a time proportional to $N \log_2 N$, utilizing decimation-in-time (DIT). DHT is a substitute for DFT; however, if the real and imaginary parts of DFT are explicitly required then they are directly obtainable as the even and odd parts of DHT. HT, its relation with FT, theorems, properties, matrix formulation, and fast algorithms are discussed in [10].

Over the years, DHT has established as a potential tool for signal processing applications [11]-[13]. Several algorithms for its fast computation and opinions regarding them are reported. Meckelburg and Lipka present a decimation-in-frequency (DIF) FHT algorithm [14] claiming it to be faster than the one in [9]. Sorenson et al. [15] further analyze FHT having the same decomposition as [9], using the index mapping approach, implement the algorithms for both DIT and
DIF, and verify their operational complexities to be the same. Prado [16] presents an in-place version of FHT along with its operational complexity. The signal flow diagram originally proposed in [9] is restructured for clarity, and by applying the transposition theorem Kwong and Shiu [17] obtain a DIF algorithm having the same operational complexity. The above approaches require computation of the cosine coefficients (CCs) and sine coefficients (SCs) which are stage-dependent. Hou [18] concludes that FHT algorithm, in essence, is a generalization of Cooley-Tukey FFT algorithm, but it requires only real, as compared to complex, arithmetic operations in any standard FFT. Malvar [19] presents a new factorization of DHT which involves discrete cosine transform (DCT). His algorithms minimize the multiplications at the expense of an increased number of additions. Hao [20] examines both the pre- and post-permutation algorithms in [9] and [14] 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 [21] reports that, for both the DIT in [9] and the DIF in [14], the operational complexity involved is the same. He further utilizes the matrix approach, derives properties of DHT [22], obtains the relations for computational complexity and presents DHT-based-DFT and DFT-based-DHT algorithms.

Rathore [23] presents a composite radix algorithm based on the matrix approach [22], applicable for any data length. Patwardhan [24] presents a mixed radix DIT DHT algorithm for an arbitrary data length. Further, Rathore [25] presents a general radix algorithm for DHT. Hu et al. [26] generalize DHT into four classes, odd DHT, inverse odd DHT, odd-squared DHT and inverse odd-squared DHT and derive fast algorithms for the resulting transforms. Zang [27] points out that these are similar to discrete $W$ transforms. Prabhu and Nagesh [28] present radix-3 and -6 DIF FHT algorithms which are derived by pairing the rotating factors with an appropriate reordering of the input sequence.
Pei and Wu [29] present the split-radix algorithm based on both even-term radix-2 decompositions and odd-term radix-4 decompositions simultaneously for the fast computation of the DHT. Bracewell [30] points out that the radix-4 transform can also be utilized as an alternative to split-radix when data lengths are powers of 2. This can be done by splitting the data sequence into two interleaved pairs and applying the radix-4 algorithm to each in turn simultaneously and combining the results. Bi and Yan [31]-[32] present split-radix algorithms which combine flexibility and regularity of various radix algorithms, allow for computations of DHT for various sequence lengths, and require a lesser operational count than the fixed radix algorithms. Bouguezel et al [33] present an algorithm using a mixture of radix-2 and radix-8 index maps in the computation of DHT of an arbitrary length $N = q \times 2^m$, where $q$ is an odd integer. The algorithm is expressed in a simple matrix form and it facilitates easy implementation and allows for an extension to multidimensional cases. Chiper et al [34] present a systolic algorithm that uses the advantages of cyclic convolution structure for the VLSI implementation of a prime length DHT. Meher et al [35] present a new formulation using cyclic convolutions that leads to modular structures consisting of simple and regular systolic arrays for concurrent pipelined realization of the DHT. Their structures for direct memory-based implementation offer more throughput than their distributed-arithmetic structures which offer less memory complexity.

Nevertheless, there is a strong need to compute the transform at a high speed to meet the requirements of real-time signal processing. This thesis presents a method to compute the elements of DHT matrix $H_N$. It identifies and proves the characteristics of $H_N$ [36]. It develops and implements the position-based method (PBM) [37]. PBM reduces the time required to compute the elements of $H_N$ as compared to the definition-based method (DBM). PBM is extended to compute the DHT utilizing simple matrix multiplication. However, it is found to be slower than the existing radix-2 FHT algorithm by Bracewell [9].
The existing radix-2 FHT algorithms in [9], [14]-[17] are studied with respect to their matrix formulation, signal flow diagram and operational complexities. 

This thesis presents Modified Radix-2 DIT [38] and DIF [39] algorithms which have a lesser operational complexity than those in [9], [14]-[17]. It presents the Modified Radix-4 Algorithm [40] which has a lesser operational complexity than those in [10] and [15]. It presents the signal flow diagram for a DIT split-radix algorithm which modifies the DIF split-radix algorithm in [15]. However, the operational complexity is the same. It finally presents the general-radix to cater to an arbitrary value of \( N \).

### 1.1.2 Architectures

Various architectures are reported in the literature to compute DHT. Chakrabarti and Jaja [41] propose a modular bit-level systolic architecture. Dhar and Banerjee [42] employ a set of linear arrays of Givens rotors with a suitable implementation of the Givens rotor using add/subtract units and hard-wired shifters. Chang and Lee [43] derive two models of linear systolic arrays and suggest the use of cordic algorithms to make the systolic arrays more efficient in computation. Hsiao et al. [44] modify the above cordic processor and obtained a higher throughput and cost effective architecture. Kar and Rao [45] propose a unified systolic architecture for sliding window computation of discrete transforms. Nayak and Meher [46] implemented a bit-level systolic architecture for discrete orthogonal transforms using a serial–parallel vector–matrix multiplication scheme based on the Baugh–Wooley algorithm. Guo [47, 48] presents two architectures; one using parallel adders and the other using a distributed arithmetic based array that utilize identical ROM modules and eliminate the accumulation loop in the processing elements. Amira and Bouridane [49, 50] present architectures to implement DHT on field programmable gate arrays. Meher et al. [51] present a design framework for scalable and modular memory based implementation of DHT in systolic hardware. These architectures compute DHT using digital VLSI techniques.
There are architectures which compute DHT based on analog blocks. Culhane et al. [52] present an analog circuit which utilizes a linear programming neural net to compute DHT. The architecture is not modular and has a limited range of $N$. Raut et al. [53] present basic switched capacitor building blocks in systolic array architecture to implement DFT. The architecture is modular but utilizes a four phase clocking scheme. Kawahito et al. [54] present a two dimensional DCT based image compression structure designed with fully differential switched-capacitor circuits. It utilizes a variable quantization level analog-to-digital converter, where the compression ratio can be flexibly changed according to the desired image type and quality, however, with an increase in the complexity. Chen et al. [55] present digitally controlled weighted summation analog circuits which may be utilized for computing DFT, DCT and DWT. They carry out the weighted sum operations in the analog domain, work in the voltage mode and omit the AD conversion reducing the power dissipation. Mal and Dhar [56, 57] present analog sampled data architectures for DHT. They utilize a switched resistor or capacitor block, integrators and a cross-point switch array with a digital controller. The architecture is based on the multiply and accumulate approach and designed for sequential data samples. Its accuracy is dependent on the matching of resistors and capacitors responsible for setting the kernel coefficients.

Reconfigurable analog arrays and dubbed field-programmable analog arrays can speed the transition of systems from digital to analog by providing the ability to rapidly implement advanced, low-power signal processing systems [58]. The drive towards analog integrated circuits has demanded the development of high performance analog circuits that are reconfigurable and suitable for CAD methodologies. The architectures which compute DHT based on analog blocks [52]-[57] are mixed-mode signal processing architectures. The operation of analog circuits is controlled by digital signals to provide a good solution as they are simple, modular and easy to implement in real time [59].
This thesis presents new architectures to implement the modified algorithms in [38]-[40]. It presents basic analog circuits designed to perform both the summing structure and multiplying structure operations. Their sensitivities to passive component variations as defined in [60] are computed. Unlike the neural net approach in [52], the architectures are modular and can be scaled for large values of $N$. The developed architecture processes the data simultaneously at each stage and is therefore faster than those based on the multiply and accumulate approach [56, 57]. The architectures for both the radix-2 DITA and DIFA are tested for the forward and inverse DHT transformations using Orcad PSpice [61] – [63]. The architecture is further successfully extended to implement the radix-4 and split-radix DHT algorithms [64, 65]. The hardware implementations of the circuits for the architectures is done in the laboratory for small values of $N$.

1.2 Thesis Organization

In chapter 2, DHT matrix is studied from a different perspective. Interesting characteristics of the DHT matrix are identified and proved analytically. They are utilized to develop methods for fast computing the elements of the DHT matrix for an arbitrary value of $N$. A position-based method computationally fast compared to definition-based method is developed and implemented.

In chapter 3, FHTA by Bracewell is discussed. The algorithm by definition and modified radix-2 decimation-in-time and decimation-in-frequency algorithms are developed. Analytical expressions for the operation counts of these algorithms are derived. These are extended to radix-4, split-radix, and finally the general-radix algorithm for an arbitrary value of $N$.

In chapter 4, basic analog circuits are designed and utilized to implement architectures for radix-2, radix-4 and split-radix algorithms for computing DHT. Their sensitivities to variation in the passive components are obtained. The architectures are validated by software and hardware setups.
In chapter 5, a summary of the important contributions and conclusions are presented followed by the suggestions for further research in the area.