CHAPTER 1

INTRODUCTION

A communication satellite functions as an overhead wireless repeater station that provides a microwave communication link between two geographically remote sites. Due to its high altitude, satellite transmissions can cover a wide area over the surface of the earth. What makes a communications satellite different from other satellites is its payload, usually a radio repeater consisting of a transmitter-receiver combination called a transponder. A transponder is a communications repeater for radio signals received on board the satellite, processed and then retransmitted to an earth station. A transponder is generally defined by its bandwidth capacity, its available effective isotropic radiated power and its on-board processing capabilities. Early transponders were simple and consisted of only a few subsystems: receiving and transmitting antennas, a Low-Noise Amplifier (LNA) receiver, a frequency converter and a High-Power Amplifier (HPA) transmitter. This basic type of transponder would only receive the signal through the Receiver antenna, amplify it in the LNA, change the frequency (down-convert) and feed it to the HPA, and then transmit it through the Transmitter antenna. Since no change is made to the signal, except for carrier frequency translation and power amplification, a satellite with this type of transponder is also known as repeater or bent-pipe satellite. A bent-pipe transponder is almost transparent to the user since it sends back basically the same information, and in the same sequence as it arrived, thus “repeating” the signal. That approach has been sufficient for the past and the most current satellite applications, but may not be enough for digital packet satellite networks. Fig. 1.1 shows the basic arrangement for a bent-pipe satellite link.
As mentioned earlier, a repeater satellite only changes frequency and power parameters on a digitally modulated signal. It makes no attempt to detect the digital data. This means that if there is distortion on the uplink signal, this distortion will be amplified and translated in frequency to the downlink, thus lowering the quality of the signal even more. This has a negative impact on the received downlink signal power and total carrier to noise (C/N) ratio and results in inferior bit error rate. Neither the user nor the repeater satellite can improve that signal, although coding may help to improve the Bit Error Rate (BER) performance.

Owing to the above mentioned limiting factors on basic bent-pipe satellites, new types of transponders were conceived, which allow a number of
different approaches to avoid these problems. None of these approaches is currently able to solve all of them, since the limitations are different for each application. These types of transponders are used on “smart” satellites called On-Board Processing (OBP) satellites, since they process in various ways the up-linked signals before repeating them on the downlink. One type of on-board processing, called Base-Band (BB) processing takes place by down-converting, demodulating, detecting the signal and reconstructing the binary information stream to correct any errors that might have been encountered during the uplink. A new waveform is generated with the clean binary sequence and, after that; the signal is again modulated and up-converted to be transmitted on the downlink, free from any uplink degradation, thus improving the Energy per bit per noise power Density (Eb/No) ratio and, therefore, the BER at the receiving earth station.

The main limitation for onboard signal processing is the power, since a satellite derives its power from solar panels or other expendable resources. Also, the onboard system should be able to support features such as extremely high data rates, Quality-of-service and multimedia applications. The algorithms proposed by researchers to provide these features are extremely sophisticated and computationally complex. Direct implementation of these algorithms on hardware platforms gives a limited performance in terms of maximum achievable data rates. Thus, all signal processing onboard the satellite must be done at high speed while consuming minimum power, which is a challenging task.

To increase the data processing rate without impairing the performance is the main focus of this work. This problem can be solved in two ways. First, by making the signal processing algorithms computationally efficient so that they need minimum hardware for their implementation, without significant loss in Bit Error Rate (BER) performance. The algorithmic modifications are
achieved by applying linear algebra techniques such as iterative, sub-optimal schemes or exploiting the structure of the algorithms. These algorithms, typically working on matrix data sets, have significant levels of parallelism. This inherent parallelism and bit-level nature of the computations can be exploited to achieve high data processing speeds by an efficient architectural design. Modifying the algorithm for a computationally efficient solution and exploiting its parallelism using suitable architectural designs are needed to achieve high speed data processing.

Second method to increase the data processing speed is by employing high speed and low power computational engines to implement these algorithms. Most of the signal processing is done in digital domain than in analog, due to the inherent limitations of analog implementations, such as high temperature sensitivity, sensitivity to DC offsets, DC voltage drifts, amplifier/mixer non-linearity, etc. Digital design does not suffer from the above-mentioned problems of analog implementation. Digital Signal Processing (DSP) techniques are prime enablers for digital communications.

DSP algorithms can be implemented in software or hardware. In software implementation, a DSP processor such as ADSP-21020 is used. In hardware implementations, Application-Specific Integrated Circuits (ASICs) or Field-Programmable-Gate-Arrays (FPGAs) are used. Designers are often forced to find a compromise between flexibility and performance. DSP processors are relatively inexpensive and very flexible, but they have severe throughput limitations, particularly in case of Space/MIL qualified components. ASICs offer improved throughput, but are relatively inflexible. ASICs also have a disadvantage since they require long lead times for development and production. FPGA devices provide nearly the throughput of custom ASICs while maintaining flexibility of a DSP processor to a good extent. This is because FPGAs can be reprogrammed to reconfigure the entire DSP
implementation. However, not all functions can be implemented efficiently on FPGAs. Some functions work faster on Digital Signal Processors too. So, a proper implementation strategy is required to enhance the overall processing speed of the onboard processing system.

1.2 EXISTING SATELLITE RECEIVER

The existing system being used by the India Space Research Organization for digital demodulator, employing the DSP processor - 21020, is shown in Fig. 1.2. The received signal from the antenna is conditioned in the RF front end before the ADC. The input to the ADC is a modulated signal at a standard IF of 455 kHz and it is sampled at a rate of 1.536 MHz by the ADC. This sampled signal is then processed in ADSP-21020 to recover the data bits. The main drawback of this system is its low throughput (64 K Baud). Thus algorithmic and architectural techniques, as mentioned in chapter 1.1, have to be applied to increase the throughput, which is the main focus of this work.

![Figure 1.2 Existing System](image)
1.3 LITERATURE REVIEW

An extensive study has been carried out in order to understand the Demodulator for satellite data communication using FPGA. In this work, we review the basic theory of Distributed Arithmetic and its modified versions to achieve a trade off between chip area and throughput. Then we present new architectures for Sampling Rate Converter and Digital Frequency Synthesizer suitable for FPGA implementation, which results in the use of reduced memory and throughput. A number of demodulator algorithms for data communication have been reported by researchers [1-13].

1.3.1 Distributed Arithmetic

Distributed arithmetic (DA) is so named because the arithmetic operations that appear in signal processing such as addition, multiplication are not "lumped" as a monolithic functional entity but are distributed in an often unrecognizable fashion [14]. The most-often encountered form of computation in digital signal processing is a sum of products or in vector analysis parlance, dot-product or inner product generation. This is also the computation that is executed most efficiently by DA. Our motivation for using DA is its extreme computational efficiency. The advantages are best exploited in circuit design, but off-the-shelf hardware often can be configured effectively to perform DA. By careful design, one may reduce the total gate count in a signal processing arithmetic unit by a number seldom smaller than 50 percent and often as large as 80 percent.
1.3.2 Historical Perspective of DA

The first widely known description of DA was given at a presentation by Abraham Peled and Bede Liu on IIR digital filter mechanization in 1974 at the Arden House Workshop on Digital Signal Processing [14]. Their work on both FIR and IIR digital filter mechanization was also published in the IEEE ASSP Transactions [15, 16]. Earlier work on DA had been performed in France by Croisier et al. [17]. The earliest documented work in the U.S. was done by Zohar [18-20], who had independently invented DA in 1968. Other early work in the United States was reported by little [21]. From the Arden House description of DA, Bona and Terhan at Rockwell International designed an integrated-circuit DA compensator for control systems applications [22], and White generalized its control system application [23].

The April 1975 special digital-signal processing issue of IEEE Proceedings contained an article by Freeny on DA applications to the telephone system at Bell Laboratories [24] and an article by White on general vector dot-product formation using DA [25]. Later that year White et al. developed an AGM digital autopilot based on DA [26], Classen et al. at Philips in Netherlands described communications systems applications [27], and Buttner and Schilessler at the University of Erlangen in Germany showed how to reduce the memory requirements [28].

In 1975, Kai-Ping Yiu at Hewlett-Packard showed a convenient rule for handling the sign bit [29]; H. Schroder of Siemens in Munich [30] and C. S. Burrus of Rice University [31] have given some suggestions and insight for speeding up the algorithms, and K. D. Kammeyer wrote a survey paper to that effect [32]. Mechanization studies on application of DA to digital filters were discussed by Burrus [33], Jenkins and Leon [34], Zeman and Nagel [35], Tam
and Hawkins [36], Arjmand and Roberts [37] and White [38, 39]. Kammeyer [40] and Taylor [41] have presented error analyses, Smith and White [42] have considered DA for coordinate transformations, and Burleson and Scharf have applied it to a rotator array [43]. Taylor applied DA to a Gray-Markel filter [44], and Zohar discussed a VLSI implementation of a correlator/filter [45]. DA were also discussed by Taylor [46], Smith and Deyer [47] in their texts, and by Mintzer in Elliott’s handbook [48].

As Field Programmable Gate Array (FPGA) technology has steadily improved; FPGAs are now viable alternatives to other technology implementations for high-speed classes of Digital Signal Processing (DSP) applications [72]. In particular, radar front-end signal processing, an application formerly dominated by custom very large scale integration (VLSI) chips, may now be a prime candidate for migration to FPGA technology. As Tyler J Moeller (1999) demonstrates [72], current FPGA devices have the power and capacity to implement a FIR filter with the performance and specifications of an existing, in-system, front-end signal processing custom VLSI chip. A 512-tap, 18-bit FIR filter was built that could achieve sample rates of 5 MHz (with a clock rate of at least 40 MHz) using Xilinx Virtex FPGA technology, and was demonstrated through simulation. Distributed arithmetic was determined to be the most optimal structure for a FPGA FIR design, although future Research may show that fast FIR algorithms or filtering in the frequency domain might give better results.

A distributed arithmetic (DA) for highly efficient multiplier-less FIR filter designed on FPGA was reported [73]. In that work, the theory of the DA is described. Furthermore, a modification of the DA based on the look up table (LUT) and filter structure to implement the high-order filter and hardware-efficient design on FPGA is introduced. The filter has been designed and synthesized with Xilinx ISE 7.1 and implemented on a 4VLX40FF668 FPGA.
device. The results show that the DA architecture can implement FIR filters with the smaller resource usage and similar speed in comparison to the previous DA architecture.

An FIR adaptive filter implementation using a multiplier-free architecture was presented by Danil J. Allred [74]. The implementation is based on distributed arithmetic (DA) which substitutes multiply and accumulate operations with a series of look-up-table (LUT) accesses. This can be achieved at the cost of a moderate increase in memory usage. The design performs an LMS-type adaptation on a sample-by-sample basis. This is accomplished by an innovative LUT update using a matched auxiliary LUT. The system was implemented on an FPGA that enables rapid prototyping of digital circuits. Implementation results are provided to demonstrate that a high-speed LMS adaptive filter can be realized employing the proposed architecture.

A highly area-efficient multiplier-less FIR filter was presented by P. Longa [75]. Distributed Arithmetic has been used to implement a bit-serial scheme of a general asymmetric version of an FIR filter, taking optimal advantage of the 4-input LUT-based structure of FPGAs. Furthermore, they have introduced a modification in the accumulator stage to achieve further savings. The filter has been designed and synthesized with Altera Quartus II, and implemented on a Stratix FPGA device.

Daniel J Allred also presented an FIR adaptive filter implementation using a multiplier-free architecture [76]. The implementation was based on distributed arithmetic (DA) which substitutes multiplies and Accumulate operations with a series of look-up-table (LUT) accesses. This can be achieved at the cost of a moderate increase in memory usage. The design performs an LMS-type adaptation on a sample-by-sample basis. This is accomplished by an
innovative LUT update using a matched auxiliary LUT. The system is implemented on an FPGA that enables rapid prototyping of digital circuits.

Henry Samuel presented all-digital architecture for implementing the front-end signal processing functions in a quadrature modulator and demodulator for high bit-rate digital radio applications. A pair of CMOS chips has been designed and submitted for fabrication in a 1.25-µm process and is expected to accommodate symbol rates up to 35 M Baud. The modulator chip accepts a pair of 8 bits in-phase and quadrature data streams and generates a band limited IF output with an excess bandwidth factor of 35%. The demodulator chip accepts a digitized IF input signal and generates a pair of filtered in-phase and quadrature baseband signals. Clock and carrier recovery must be performed external to the demodulator chip. The modulator and demodulator chips each incorporate 40-tap multiplier less FIR square-root Nyquist matched filters and the cascade of the two chips achieves a peak inter symbol interference distortion of -54 dB (assuming ideal D/A and A/D conversion). The modulator chip can generate any arbitrary signal constellation within a rectangular grid of 256x256 points, thus, the all-digital implementation results in a generic chip set suitable for a wide variety of high bit-rate digital modem designs using various modulation formats such as M-ary PSK and Quadrature amplitude modulation [77].

Jung-Pal Choi Seung et al. in their distributed arithmetic-based architecture for an inner product between two length W vectors, the size of the ROM increases exponentially with N [78]. Moreover, the ROMs are generally the bottleneck of speed, especially when their size is large. In this paper, a ROM size reduction technique for DA was presented. This method is based on modified OBC (Offset Binary Coding) and control circuit reduction technique. By simulations, it was shown that the use of the presented technique can result in the reduction of the number of gates by up to 50%.
1.3.3 Digital Frequency Synthesizer

An extended form of the parabolic approximation, which was previously introduced by the authors, was presented by Amir M. Sodagar [95]. This form of the approximation is so close to the sine function that satisfies the accuracy requirements for a typical sine-output Digital Direct Frequency Synthesizer. Then, the approximation is used to develop a novel ROM-less architecture for sine-output direct Digital Frequency Synthesizers. Based on the optimized architecture, an experimental test system is developed. Experimental results for the implemented ROM-less sine-output parabolic Digital Direct Frequency Synthesizer show that the approximation error will cause harmonic levels below the spurious level associated with the output quantization.

Sung-Won Lee presents an area and power-efficient quadrature direct digital frequency synthesis technique called fine-grain angle rotation [96]. To reduce the large bit width requirement of the angle rotation, multiple start points are introduced and the angle of rotation is confined to the small angles. A prototype chip occupies 0.16mm$^2$ in 0.25 μm 1P5M CMOS technology and consumes 90 mW at 400MHz clock frequency, which significantly improved the performance compared to the previous state-of-the-art chips.

A new approach to design of a Digital Direct Frequency Synthesizer using Chebyshev polynomial approximation was presented in Ref. [97]. This technique was an improvement of previously designed DDFS circuits based on Taylor approximation. In that reference, a detailed description of the method used to compute the content of the ROM in Chebyshev DDFS was presented, analytically showing the improved accuracy of new DDFS with respect to Taylor one. Two new Chebyshev DDFS with 80dBSpurious free dynamic range have been designed up to the layout level, exploiting optimized
arithmetic circuits. Simulation results confirm improvement in performances with respect to Taylor DDFS [97].

Davide De Caro (2005). A detailed description of direct digital frequency synthesizers using an optimized piecewise linear approximation for phase to sine mapping, named dual-slope was presented in Ref. [98]. The dual-slope technique allows reducing ROM size with respect to previously proposed piecewise-linear approximation approaches, with beneficial effects on system performances. Two high-speed DDFS have been fabricated and characterized in 0.25 m CMOS technology. Both circuits produce two quadrature 12 bit outputs with a spectral purity of 80 dB. The first circuit reaches a maximum operating frequency of 600 MHz by using six pipelining stages. The second circuit operates up to 480 MHz clock speed while dissipating only 72 W/MHz. Analytical investigation of spectral performances achievable by using dual-slope approximation and detailed description of high speed flip-flop employed in 600 MHz DDFS were also presented in this paper.

To summarize, a number of demodulator algorithms for data communication have been reported by researchers [1-13]. Basics of DA and its filter application in Ref. [14-78]. Digital Frequency Synthesizer Algorithm and Architecture developed are available in the literature [79-131]. FPGA implementations of some of these architectures were also reported [132-134]. A novel, two stage estimation schemes for demodulator for processing satellite data was proposed in our work [135], where carrier frequency estimation is followed by timing recovery under training. Therein, the receiver algorithm has been partitioned into two parts, one to be implemented on FPGA and the other on a DSP. An overview of the whole system architecture was also presented and its performance evaluated. In this work, we review the basic theory of distributed arithmetic [136] and its modified versions to achieve a trade off between the chip area and the throughput. Thereafter, we present new architectures for sampling rate converter and digital frequency synthesizer
[136-138], which results in the use of far less memory than that of the conventional techniques. However, the DSP implementation of the remaining part, namely, frequency and timing offset, carrier recovery and LMS are out of scope of the present work.

1.4 MOTIVATION AND OBJECTIVE OF THIS WORK

1.4.1 Motivation

The draw back of the throughput in the existing Receiver is the seed motivation for this Research work. The DA based algorithm optimization and novel FPGA Architectural design yield the desired high throughput, and to reduce the size of the OBP, thus consuming less power. These were the prime motivating factors of this research.

1.4.2 Objectives

The following are the objectives of the present work.

- To investigate the shortfalls of the conventional satellite transponders and the advancements of smart satellites (On Board processing satellites). To analyze the scope for performance enhancements with respect to OBPs.

- To present new architectures for sampling rate converter by reviewing the basic theory of distributed arithmetic and its modified versions to achieve a trade off between the chip area and the throughput.
• To present New Architecture for Digital implementation of Digital Frequency Synthesizer, which generates the sinusoid of appropriate frequency. The main advantage of this architecture is that it does not depend upon the extensive use of ROM.

• Design of Architecture and simulated FPGA implementation of a new architecture for demodulator to increase the throughput and to reduce the size of the OBP.

1.5 ORGANIZATION OF THE THESIS

Quadrature Phase Shift Keying is the preferred modulation scheme for satellite data communication. This, being a coherent modulation scheme, mandates the knowledge of transmitted carrier frequency and phase at the receiver. Carrier frequency offset estimation is the process of estimating the offset between the frequency drift/change of the local oscillator at the receiver and the actual carrier frequency transmitted. In addition to that, the estimate of the exact sampling clock frequency and phase, to convert the continuous time received signal into a discrete time sequence of data symbols, must also be performed at the receiver and is called Timing Recovery.

In Chapter 2, a two stage estimation scheme is proposed, where carrier frequency estimation is followed by timing recovery under training. Frequency offset estimation is performed by finding the slope of the best fit line through the unwrapped phase. Since this carrier frequency estimation technique gives an approximate estimate of the frequency offset, there still remains some frequency offset to be corrected. This is done in the second stage, where least mean square algorithm is employed to track the variations in frequency and
estimating the symbols. Estimation of the correct sampling instant is based upon finding the minima of the absolute of the received training pattern and then calculating the maximum energy instant from it. An overview of the whole system is given and its performance is evaluated. BER curves for different frequency and timing offsets are presented. Some tasks in the system may serve as bottlenecks to higher data rates due to their computationally intensive nature. These tasks are identified in Chapter 3 and based upon that, the algorithm is partitioned into two parts. One part is to be implemented on FPGA and consists of high computational complexity modules and the other is to be coded on a DSP processor.

A number of DSP techniques can be applied to reduce the computational complexity of the selected tasks, thus easing their hardware implementation. In Chapter 4, a new VLSI architecture for the sampling rate converter, based upon Modified Distributed Arithmetic using Offset Binary Coding technique, is presented. Hardware implementation of the digital frequency synthesizer, which generates the sinusoid of appropriate frequency, is discussed in Chapter 5. In Chapter 6, we will describe the hardware implementation details of the FPGA part of the demodulator. Elaborate Results are presented in Chapter 7 in order to validate the proposed novel algorithms and architectures. Finally, conclusions drawn and scope of future work are presented in Chapter 8.