Interval Modulation: A New Paradigm for the Design of High Speed Communication Systems

Thesis by

Saleem Mukhtar

In Partial Fulfillment of the Requirements

for the Degree of

Doctor of Philosophy

California Institute of Technology

Pasadena, California

2004

(Submitted July 7, 2004)

Acknowledgements

I am deeply indebted to many people who made by Caltech years enlightening, rewarding and a memorable experience. First and foremost, I must extend by gratitude to Prof. Ali Hajimiri, Prof. Jehoshua Bruck and Prof. Paul Sternberg for granting me the honor of being members of their research group. I have benefited vastly from their technical expertise and insights and their high standards in research, teaching and publication. The financial support received from them, as well as the Department of Defense is greatly appreciated.

Second, I must thank my colleagues, former and current members of the Paradise and Chic Lab for years of friendship. These include Mike Gibson, Matthew Cook, Suleyman Gokyigit, Marc Reidel, Kevin Foltz, Massimo Franceschetti, Anxiao (Andrew) Jiang, Yuval Cassuto, Hossein Hashemi, Roberto Aparicio, Xiang Guan, Chris White, Behnam Analui, Ehsan Afshari, Arun Natarajan, Jim Buckwalter, Abbas Komijani, Sam Mandegaran and Aydin Babakhani. Special thanks must be extended to Behnam Analui and Xiaofeng Li who taught me how to design a chip, use the design tools and taught me how to use the measurement equipment. Also I must thank Anne Shen and Niklas Wadefalk for help with soldering and wirebonding. I must also thank my CNS classmates for six wonderful years here at Caltech and one fun trip to Israel. These include Daniella Meeker, Ania Mitros, Adam Hayes, Ofer Mazor, Bjorn Christianson, Javier Perez-Orive and Jason Davis. Also I must thank my closest friends Kashif Alvi, Pururav Thoutireddy and Arella Karspeck for some extraordinary times.

Most of all I must thank my family – my parents, Parveen and Khalid, my younger brother Salman, my Aunt Farzana and my Uncle Tariq, my cousins Farzana and

Nafees, Shabana and Mansoor and my nieces, Zoya and Zenab for their love, support and encouragement. They always inspired me to follow the path of my heart as opposed to the path of convenience. I am specially indebted to my parents for the extraordinary sacrifices they made to make not only my education at Caltech, but also my undergraduate education at Carnegie Mellon University possible. I am extremely grateful to them for their love, support and encouragement. This dissertation is dedicated to them as a token of my gratitude.

Contents

List of Figures

List of Tables

Abstract

In this thesis we propose a new, biologically inspired, paradigm for the design of high speed communication systems. The paradigm consists of a new modulation format referred to as Interval Modulation (IM). In order to transmit data in an efficient manner using this format, new coding techniques are needed. In this thesis we propose a coding technique based on variable length to variable length prefix trees and code construction algorithms are outlined. These codes are referred to as Interval Modulation Codes (IMC). Furthermore, data encoded with this modulation format cannot be transmitted or received using conventional synchronous CDR based receivers. In this thesis we outline a new asynchronous circuit architecture for both the transmitter and receiver. The architecture is based on active delay lines and eliminates the need for clock recovery.

Chapter 1

Introduction

In this thesis we propose a biologically inspired paradigm for the design of high speed digital communication systems for fiber optic communication. The transmission of information in neural systems has been extensively studied. Neural systems are largely asynchronous in nature. The transmission of information from one neuron to another, happens by means of a spike train, or a sequence of action potentials. These action potentials are all or none events. Once a neuron spikes, it enters a refractory period, during which it cannot generate another action potential. The reasons underlying this phenomenon are related to the biophysics of action potential generation. This is the only constraint on the time difference between two adjacent spikes or action potentials. How information is encoded in spike trains is an area of active study and debate, and there are reasons to believe information is encoded differently in different neurons based on function. Two leading hypothesis are rate codes, in which information is encoded in the frequency of action potentials or spikes and temporal codes in which information is encoded in the time difference between two consecutive action potentials.

Communication systems have a wide array of applications, and the theory underlying them has also received much attention. Where as communication systems, can refer to systems that are used to transmit analog or digital information, in this thesis we will mainly discuss communication systems for the transmission of digital information. Unlike neural systems, most communication systems, and in general digital systems are synchronous in nature. Furthermore, unlike neural systems, gen-

erally information in digital systems is encoded in the amplitude of the signal, a high represents a 1, a low represents a 0. This simple modulation format is known as Non Return to Zero (NRZ) modulation and it is the modulation format of choice in most communication systems. In spite of the simplicity of the modulation format, sending data across a link requires more than just stuffing bits in one end and having them show up at the other. The bits arriving at the receiver must be sampled with a clock to return them to the digital domain. With slower serial interfaces, such as analog spectrum telecommunication modems, this clocking may often be done through an asynchronous over sampling of the received bit stream. At the faster data rates in the optical communication domain, this over sampling becomes impractical. At faster transfer rates the sample clock must operate at the same rate as the bits in the data stream. While the sample clock could be delivered on a separate link, this is generally poor practice. The time skew between the data and the clock is difficult to manage, and the cost of the second links makes this prohibitive. Since no sample clock is delivered to the receiver along with the data, a clock must be extracted from the data stream. This is accomplished through the use of a high performance PLL (Phase Locked Loop) that detects the transitions in the serial stream as illustrated in Figure 1.1. Numerous 10-Gb/s fiber optic receivers based on the paradigm described above have been implemented [4], [8], [42], [43], [44], [45], [73], [88] and [101]. Unfortunately, an NRZ data stream may contain few if any transitions, especially when sending data of mostly one or zero bits. To send data of this type, the data must be modified to force additional transitions into the data stream. Thus there is a need for coding. Numerous methods exist to force additional transitions into a data stream. These can be broken down into two categories, scrambling and run length limited coding.

Scrambling modifies a data stream by merging it with one or more randomizer polynomials. Scrambling, used in telecommunications interfaces such as SONET [89] and ATM [6], is 100% efficient. For every bit in the source data stream, a single bit is sent across the interface. While this would at first appear as the perfect solution, scrambling does have its drawbacks. The characteristics of a scrambler are such that the scrambler can be zeroed out by specific data patterns. In other words, scrambling

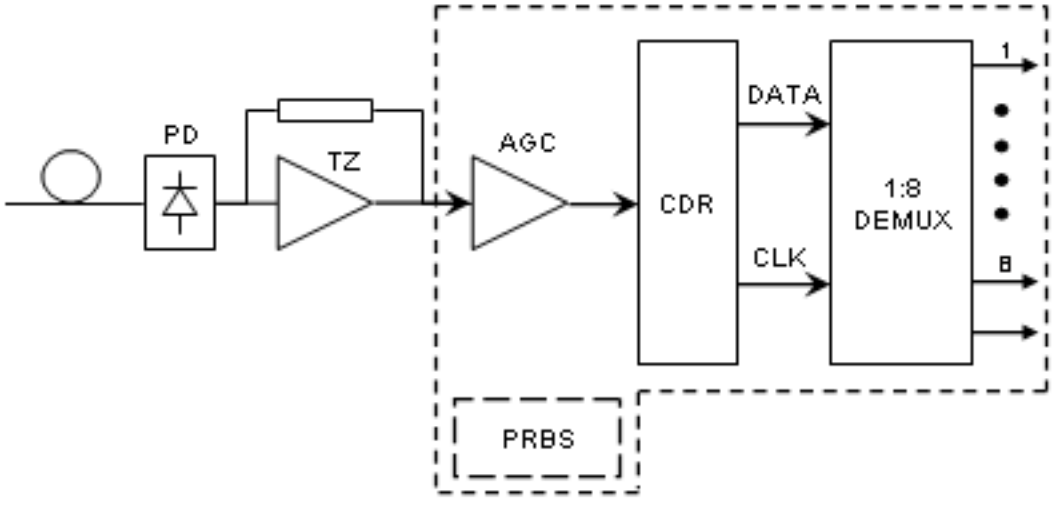


Figure 1.1: General System Level Architecture of a High Speed Optical Receiver

cannot guarantee transitions on the data stream, it can only make them more likely. For rogue bit sequences, there will be no data transitions. A scrambled interface is also somewhat limited in how link control information is moved across a link. Because the link efficiency is perfect, all combinations of bits are used to represent data. This requires all link information to be sent as combinations of data characters.

The other method of forcing transitions involves run length limited coding. In a coded interface, the source data is modified by mapping the source data into alternate bit combinations called symbols. These symbols are constructed with extra bits that guarantee a minimum transition density. This problem has been extensively studied in literature and numerous solutions are available [1], [30], [31], [49], [53], [65], [67], [68], [69] and [102]. Encoded interfaces generally have link efficiencies ranging from 50-95%. One of the most popular encodings is known as 8B/10B [102]. This encoding is used by popular high speed serial interfaces such as Fibre Channel [112], ATM [6], ESCON^{TM}, Gigabit Ethernet [34] and DVB-ASI. The 8B/10B code maps an 8-bit data character into a 10-bit symbol known as a transmission character. This code limits the maximum number of consecutive ones or zeros that can occur in the transmitted serial bit stream to five. The efficiency of the link is 80%. This code is optimized for transmission across optical media. In addition to guaranteeing transition density,

the code is DC balanced. Thus it limits the low frequency content in the data stream which in turn allows the use of low-cost AC-coupled optical modules. In addition since some of the bit sequences are unused, these can be included in the data stream as special control characters (such as start of frame, end of frame, etc.).

We will describe an alternate paradigm for the design of digital communication systems which is inspired by temporal codes that are used in the nervous system. The primary advantage of the proposed paradigm over conventional designs is a substantial improvement in data rate. The combination of amplitude modulation and run length limited coding can be thought of as interval modulation. In the 8B/10B code, there can be at most 5 consecutive zeros or ones. If B is the bit period, the time between voltage transitions is either $1B$, $2B$, $3B$, $4B$, or $5B$. Note that since the circuits used are synchronous, the time between voltage transitions are always a multiple of the bit period. The data rate is $0.8/B$ (compared to a data rate of $1/B$ if scrambling were used). The fundamental draw back of this scheme is also that the time intervals between voltage transitions must be multiples of a clock period. Consider what would happen if the time between voltage transitions were to be restricted to just two possibilities $1B$, $1.2B$. And data were transmitted using a very simple coding scheme where a pulse of duration $1B$ would be transmitted for every zero in the data stream and a pulse of $1.2B$ were transmitted for every one in the data stream. Even in the worst case (sequence of all ones), the data rate achieved by this simpler coding would be $1/1.2B$ which is greater than $0.8/B$. In addition to a higher data rate, the coding scheme is much simpler than the 8B/10B code and thus simpler to implement. The code also provides better error propagation properties. It must be noted that this simple scheme for coding is not optimal. Another scheme would first take a binary sequence of fixed length, and map it to a balanced binary sequence. Simple coding techniques for doing this exist and the number of extra bits is only logarithmic in the number of bits in the original sequence [61]. Thus if the original sequence is long enough, there is no loss in terms of data rate. Now the balanced code can be transmitted using the technique described above. Since the coded sequence contains an equal number of zeros and ones, the data rate would be $1/1.1B$. Also

the code has the interesting property that all binary sequences can be transmitted in the same amount of time. However, there is a loss in terms of error propagation properties. The above arguments illustrate, that if we drop the constraint that the time between voltage transitions must be a multiple of the bit period, then improvements in terms of data rate, DC balance and other properties are realizable. In the example above, we had restricted the time between voltage transitions to belong to one of two possibilities. This requirement is arbitrary. For example the time between voltage transitions could be restricted to belong to $1.0B$, $1.2B$, $1.4B$, $1.6B$, or $1.8B$. It is possible to show that in this case, the data rate can exceed even that achieved by scrambling. Furthermore, the time between voltage transitions does not have to be restricted to multiples of the clock period B or some other constant but can be a set of any arbitrary values, even though in the cases considered, the time between voltage transitions are multiple of $0.2B$. It must be noted that in the optical domain, data rates are very high and as mentioned the clock due to limitations in chip processes can operate at a frequency of at most $1/2B$ (this corresponds to a bit period of B). Given this limitation, it is not possible to implement a receiver which samples the amplitude using a clock with a frequency of $1/0.2B$. An alternate architecture is needed, a transceiver architecture which operates directly by measuring time between voltage transitions. Notice that if we were to sample the amplitude, we would need to sample it at a frequency of $1/0.2B$. However, if we are to measure time differences between adjacent transitions, the smallest measurable time interval only needs to be a bit period. And the maximum sampling frequency needs to be $1/2B$. The resolution of the time measurement circuit needs to be sub-bit period. Variants of Run Length Limited Codes can still be used to do the encoding in the case where time between transitions is a multiple of some number smaller than the bit period B, but since the architecture of the transceiver is different, conventional Run Length Limited Codes which are designed to be used with conventional synchronous transceivers, lose some of their desirable properties, like limiting the effects of error propagation. Formulating a new circuit architecture, will give rise to new coding constraints that must be satisfied by desirable codes. In this thesis will be address two fundamental questions

– one how is binary data to be encoded in a signal where time between transitions can take on arbitrary values. And second what circuit architecture can be used for transmission and reception of data using such a modulation format at high data rates.

We next review the organization of this thesis. In Chapter 2 we will discuss the two basic problem of transmission of analog and digital information through an optical channel. Next we will describe the modulation formats used for analog and digital transmission of information. We will then compare the proposed modulation format to the different modulation formats concluding with a hierarchical categorization of different modulation formats. In order to transmit data using the proposed modulation format one needs new and novel coding techniques. In this thesis we propose a coding technique based on variable length to variable length prefix free codes. In Chapter 3, we will introduce the concept of a prefix free code, and illustrate its primary advantage, instantaneous decodability without the need for look ahead. We will classify prefix free codes into four major categories and discuss and summarize the theoretical problems associated with prefix free coding that have been studied in literature. In Chapter 4 of this thesis we will propose the use of variable length to variable length prefix free codes for interval modulation and refer to these as interval modulation codes. Next we will formulate the problem of code construction. This problem will then we reduced to a large scale integer optimization problem which is well structured. We study the properties of the polyhedra defined by this structured system of inequalities. These properties lead to efficient algorithms for determining if the linear programming relaxation of the code construction problem is feasible and an efficient algorithm for solving the linear programming relaxation. In Chapter 5, we study a generalization of the Fibonacci numbers that arises in the context of rate analysis as well as code construction. In Chapter 6, we outline a novel asynchronous circuit architecture for high speed digital transceiver design. The transmitter is based on the concept of a reconfigurable ring oscillator whereas the receiver is based on the concept of multiplexed picosecond time digitizers. The proposed architecture does away with the need for clock recovery circuits. A prototype based on the ideas discussed in this chapter was fabricated and measurement results are discussed. In

Chapter 7 we briefly summarize the contributions of this thesis and outline exciting areas for future research.

Chapter 2

Modulation Formats

In this chapter we review different modulation formats that are commonly used for data transmission through fiber. Modulation formats can be classified into two categories, analog and digital, based on whether they are designed for analog or digital data transmission. The hierarchical classification of modulation formats used for analog and digital data transmission is presented in Figure 2.1.

2.1 Analog Modulation Formats

Modulation formats for analog data transmission can be grouped into two categories, Pulse Shape Modulation (PSM) Formats and Pulse Time Modulation (PTM) Formats. The simplest example of a PSM modulation format is Pulse Amplitude Modulation (PAM). In PAM the amplitude of individual, regularly spaced pulses in a pulse train is varied in accordance with the amplitude of the modulating signal. Such a scheme is both simple and bandwidth efficient but cannot deliver the signal to noise ratio. In addition PAM suffers to an extent from nonlinearity of the optical channel, the photodiode, the photodetector and associated circuitry, severely limiting the quality of information transmitted. Due to these reasons, real communication systems are based on alternate modulation formats discussed below.

The basic framework for research into PTM techniques was laid down around 50 years ago and reported in the late 1940s [23], [29], [54] and [64] but it is only recently that a revival of interest has been experienced with the development of fiber

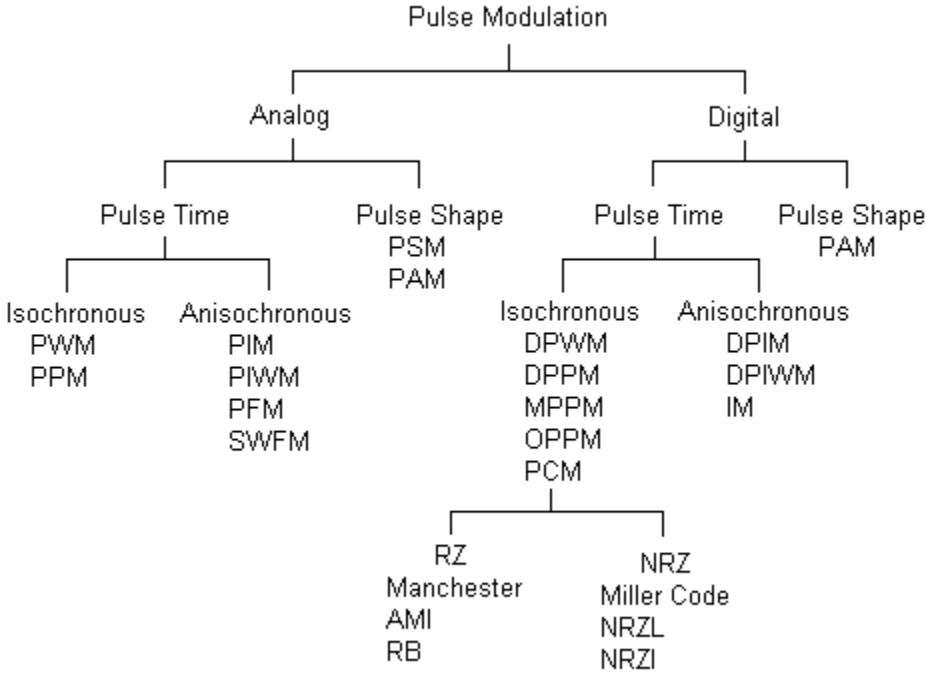


Figure 2.1: Hierarchical Categorization of Analog and Digital Modulation Formats

transmission systems [111]. In all PTM methods, one of a range of time-dependent features of a pulsed carrier is used to convey information in preference to the carrier amplitude. The fundamental advantages of these schemes are that modulation is simple, the signal is not quantized thus there is no need for digital coding, and the pulse format renders the scheme immune to device and channel nonlinearity. Furthermore, the signal can be routed through logic circuits and switching nodes in a network. PTM techniques can be further grouped into two categories – Isochronous PTM techniques and Anisochronous PTM techniques.

Isochronous PTM techniques are fixed rate schemes that require a fixed amount of time to transmit a sample of the modulating signal. Two common examples under this category are Pulse Width Modulation (PWM), sometimes also referred to as Pulse Duration Modulation (PDM) and Pulse Position Modulation (PPM). In PWM, the width of the pulsed carrier within a predetermined timeframe is changed according to the sampled value of the modulating signal. PPM can be considered as differentiated

PWM, and carries information by virtue of the continuously variable position of a narrow pulse within a fixed timeframe. Anisochronous PTM techniques variable rate schemes in which the time required to transmit a sample of the modulating signal varies and generally depends on the sampled value itself. This group consists of four different modulation formats, Pulse Interval Modulation (PIM), Pulse Interval and Width Modulation (PIWM), Pulse Frequency Modulation (PFM) and Square Wave Frequency Modulation (SWFM). As the name suggests in PIM, the variable intervals between adjacent narrow pulses is determined by the amplitude of the input signal. PIWM is derived directly from PIM to produce a waveform is which both mark and space convey information in alternating sequence. In both PIM and PIWM each successive timeframe commences immediately after the previous pulse unlike in PWM and PPM. In PFM, the instantaneous frequency of a train of narrow pulses is determined by the amplitude of the modulating signal. SWFM is closely related to PFM, consisting essentially of a series of square wave edge transitions occurring at the pulse positions of PFM.

The primary advantage of Isochronous PTM techniques over Anisochronous PTM techniques, is that Isochronous PTM techniques are far easier to multiplex in the time domain because of their fixed frame timing intervals and require only a simple demultiplexer at the receiving end. The primary advantage of Anisochronous PTM techniques over Isochronous PTM techniques is that Anisochronous PTM techniques essentially can transmit more samples in a given amount of time. PWM and PPM have been widely adapted for use in fiber optic applications [10], [11], [86], [96], [104], [105] and [106]. PFM has been used extensively for optical fiber transmission of video and broadcast quality TV signals [46], [47], [48], [50], [55] and [76] with SWFM being employed for the transmission of HDTV and other wideband instrumentation signals [66], [79], [82], [107], [109] and [108]. PIM and PIWM have found fewer wideband fiber optic applications [76], [77], [83], [84] and [110].

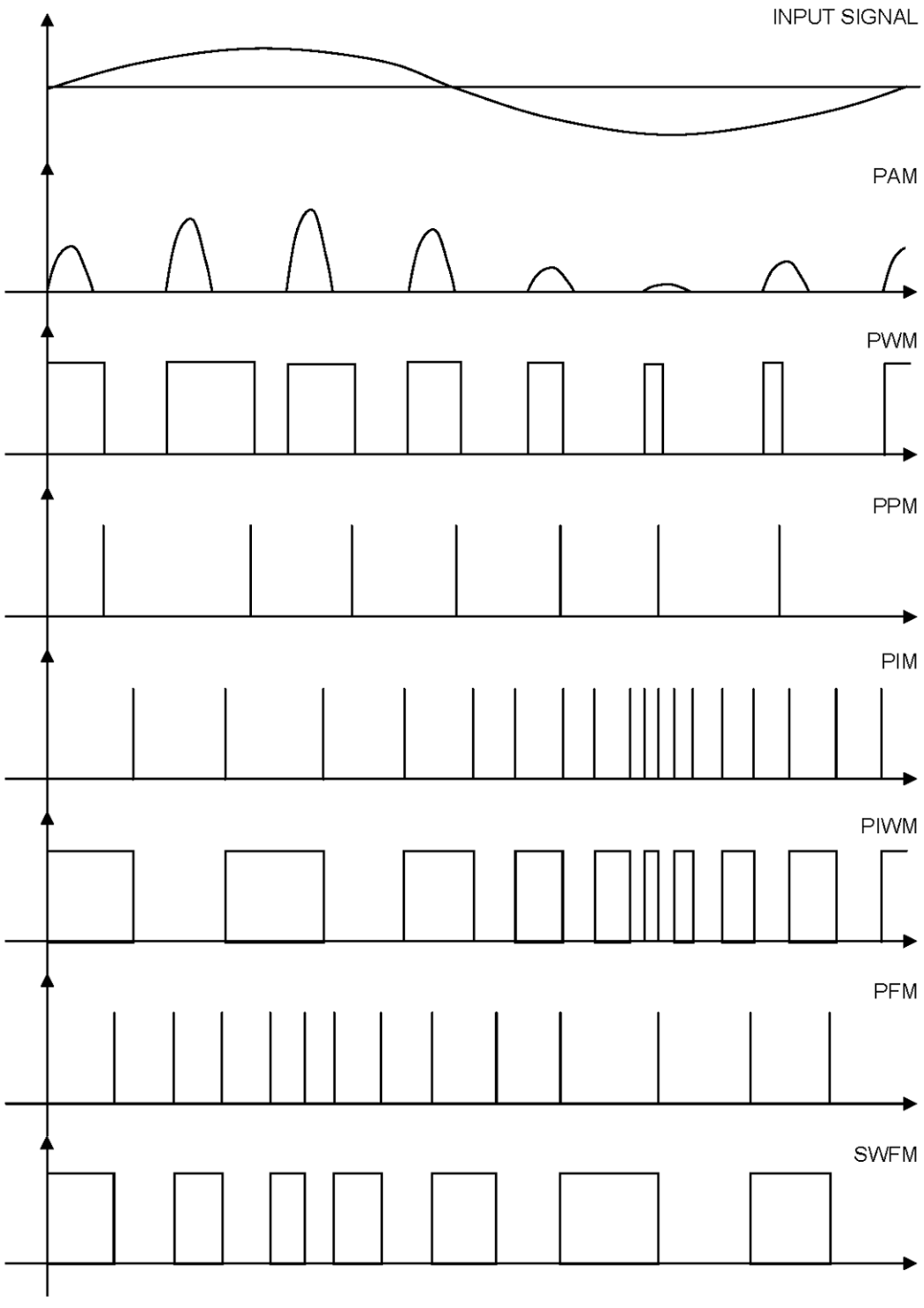


Figure 2.2: Analog Signal Represented Using Different Modulation Formats

2.2 Digital Modulation Formats

There are several modulation and encoding schemes that are suitable for optical communication systems. The simplest approach is based on intensity modulation with direct detection and is referred to as on-off keying (OOK). In this scheme a zero is represented by zero intensity and a one by positive intensity. The primary disadvantage of this scheme is the low power efficiency. Higher average power efficiency can be achieved by employing Pulse Time Modulation (PTM) schemes which will be discussed later. In these schemes a range of time dependent features of a pulse carrier can be used to convey information. OOK is also referred to as 2-PAM since there are two intensity levels, one corresponding to a logical 0 and the other corresponding to a logical 1. In an n-PAM modulation format, n distinct intensity levels are used, each of these can represent a unique binary combination of $\lfloor \log_2(n) \rfloor$ bits. In practical systems the number of intensity levels used is 2 or 4. The advantage of 4-PAM systems over 2-PAM systems is higher data rate, but 4-PAM systems suffer from the same limitations as PAM systems used for analog transmission. Namely, nonlinearities in the channel, photodiode and photodetector and associated circuits affect the threshold levels that must be used in the receiver. Furthermore, 4-PAM signals cannot be passed through digital logic and more complex processing circuitry is required.

As in analog modulation formats, an alternate is to use PTM techniques which can be grouped into two classes, Isochronous PTM techniques and Anisochronous PTM techniques. Isochronous PTM techniques, are fixed data rate techniques in which the time required to transmit a bit of information is constant, or equivalently, the number of bits transmitted in a constant amount of time is constant. Anisochronous PTM techniques are variable data rate techniques in which the time required to transmit a given number of bits varies depending on the bit sequence. The common Isochronous PTM techniques are Digital Pulse Width Modulation (DPWM) or Digital Pulse Duration Modulation (DPDM), Digital Pulse Position Modulation (DPPM), Multiple Pulse Position Modulation (MPPM), Overlapping Pulse Position Modula-

tion (OPPM) and Pulse Code Modulation (PCM). The common Anisochronous PTM techniques are Digital Pulse Interval Modulation (DPIM) and Digital Pulse Interval Width Modulation (DPIWM).

DPWM is comparable to PWM for analog transmission. In this modulation format the width of a pulsed carrier within a predetermined timeframe is changed according to the value of binary combination represented by that time frame. DPPM is a differentiated version of DPWM. In this modulation format, the position of a single pulse within a time frame encodes the binary combination represented by that time frame. MPPM is a generalization of DPPM. In this format the position of multiple, but a fixed number, of pulses within a time frame encodes the binary sequence represented by that time frame. OPPM is a modification of MPPM. In this modulation format, an added constraint is placed on the position of the multiple pulses within a time frame. The added constraint is that the multiple pulses occupy adjacent slots. PCM techniques can be further classified into two groups, Return to Zero (RZ) techniques and Non Return to Zeros (NRZ) techniques. Common RZ modulation techniques are RB Modulation, Alternate Mark Inversion Modulation (AMI) and Manchester Modulation. Common NRZ techniques are Non Return to Zero Level (NRZL) modulation, Non Return to Zero Inverted (NRZI) modulation and Miller Code Modulation. AMI modulation format is a pseudo ternary modulation format in which successive ones are represented by alternately positive and negative polarity and the absolute values of their amplitudes are normally equal and zeros are represented by zero amplitude. In Manchester Modulation, a zero is represented by a 0-1 transition whereas 1-0 transition encodes a 1. NRZL modulation is equivalent to OOK and 2-PAM modulation. A positive intensity level is used to encode a logical 1 and a zero intensity level is used to encode a 0. In NRZI modulation there is a change in amplitude level from one level to another, when a one is transmitted. The amplitude level remains unchanged when a zero is transmitted. This kind of encoding is also called differential encoding. In Miller Code Modulation, a logical one is encoded as a 01 and a zero is encoded as 10 if the preceding bit was a zero, and 00 if the preceding bit was a 1.

As mentioned the Anisochronous Modulation Techniques are DPIM and DPIWM.

In DPIM the time between two consecutive pulses encodes the binary sequence. Thus two binary sequences containing the same number of bits will be represented by different intervals. In DPIWM, the binary sequence is encoded in the width of pulses of alternating polarity.

2.3 Interval Modulation

Interval Modulation (IM) is an anisochronous modulation format for digital data transmission. Interval Modulation is similar to DPIWM in that the binary sequence in encoded in the width of pulses of alternating polarity. The fundamental difference between DPIWM and IM is that in DPIWM the widths of the pulses are constrained to be integer multiples of the bit period. In IM the widths must belong to set of arbitrary values which need not be integer multiples. It must be noted that the data rate achieved by NRZ, OOK and 2-PAM is 1. The data rate achieved by 4-PAM is 2. All other schemes achieve a data rate of less than 1. The disadvantages of 4-PAM have already been discussed. The threshold level used for thresholding intensity levels at the receiver becomes sensitive to nonlinearities of the channel, photodiode and photodetector and associated circuitry. Furthermore, 4-PAM signals are not compatible with conventional two state digital circuits. Like 4-PAM, IM can achieve a data rate in excess of 1. The primary advantages over 4-PAM are that the signal is binary, that is it uses only two intensity levels making it compatible with conventional digital circuitry and making it insensitive to nonlinearities in both the channel and other devices. The disadvantage of IM over 4-PAM are additional complexity since rate efficient coding of binary data is non trivial. Furthermore, since IM is Anisochronous whereas PAM is Isochronous some schemes must be deployed to mitigate the effects of error propagation. It must be noted that all Anisochronous schemes can be impacted by error propagation. So this disadvantage is not restricted to IM alone. Also since the pulse widths need not be multiples of the bit period but can take on arbitrary values, the transceiver design cannot be based on conventional clocked synchronous circuitry. In this thesis we will outline present a coding technique

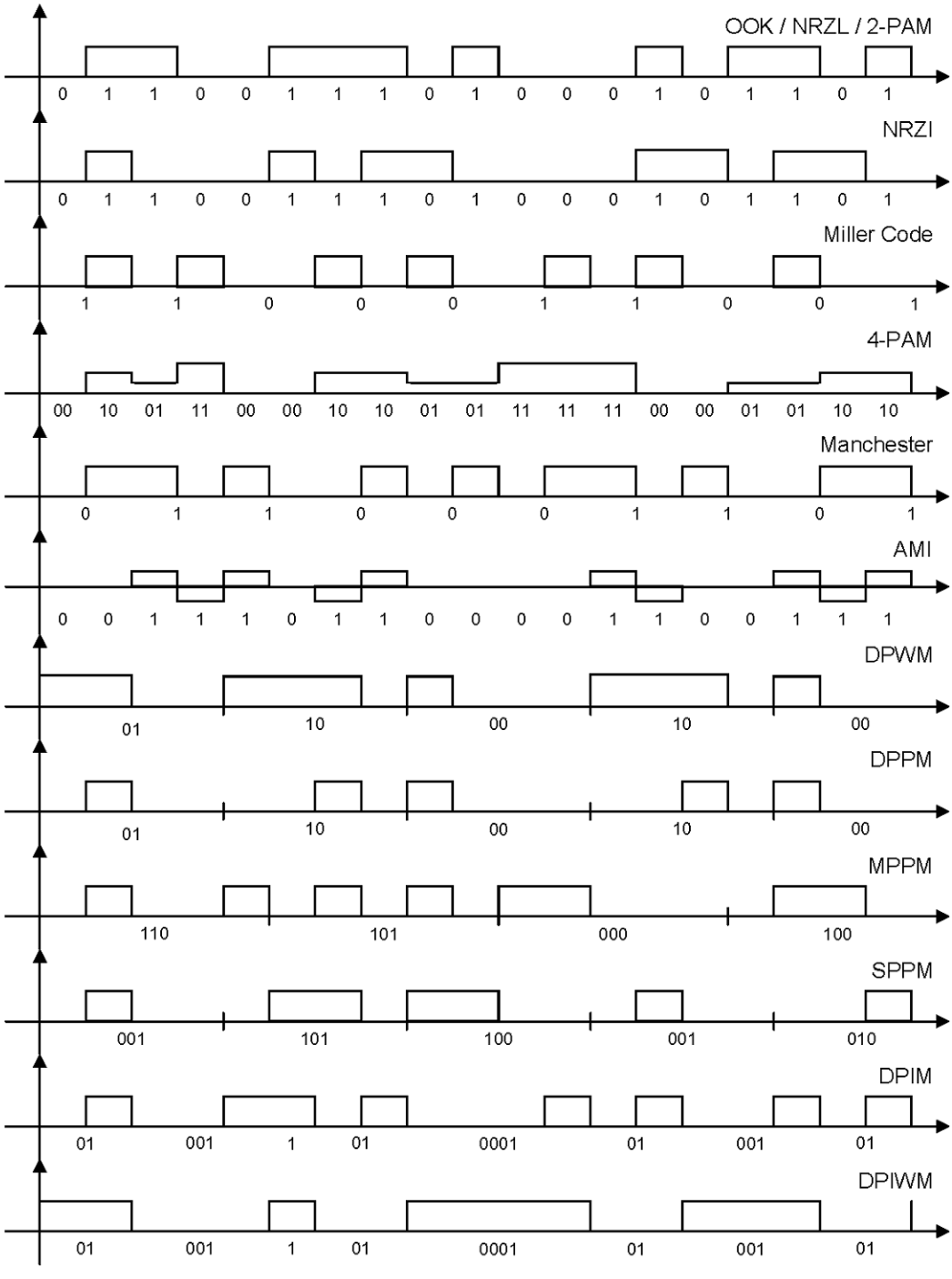


Figure 2.3: Digital Signal Represented Using Different Modulation Formats

based on variable length to variable length prefix-free codes and an asynchronous transceiver architecture.

Chapter 3

Prefix Free Coding

3.1 Prefix Free Codes

Definition 1 *A code is a mapping of source messages, words from the source alphabet, into codewords, words from the code alphabet.*

Consider the simple code shown in Figure 1 A. The source alphabet is $\{a, b, c, d\}$ and the code alphabet is $\{0, 1\}$. The code itself is a mapping of source messages or words from the source alphabet, $\{a, b, c, d\}$ to words from the code alphabet, $\{00, 01, 10, 11\}$. Thus the string *cabd* would be represented as 10 00 01 11. The source messages are the basic units into which the string to be represented is partitioned. These basic units may be single symbols of the source alphabet, as in the preceding example, or they may be strings of symbols from the source alphabet. When source messages of variable length are allowed, the question of how a message ensemble (sequence of messages) is parsed during encoding, into individual messages arises. Similarly, if variable length codewords are permitted, the question of how a code ensemble (sequence of codewords) is to be parsed into codewords during the decoding process arises.

Definition 2 *A code is distinct if each codeword is distinguishable from every other, that is the mapping from source messages to code words is one to one.*

Definition 3 *A distinct code is uniquely decodable if every codeword is identifiable when immersed in a sequence of codewords.*

Note that both the codes in Table 3.1 (A-D) are all distinct. The codes in Table 3.1 (A,C,D) are all uniquely decodable but the code in Table 3.1 (B) is not. If the code in Table 3.1 (B) were to be used 10 could be parsed as the codeword 10 or codeword 1 followed by codeword 0. Thus any message transmitted using this code cannot be uniquely decoded even though the code is distinct.

Message	Code
a	00
b	01
c	10
d	11

A

Message	Code
a	0
b	1
c	10
d	11

B

Message	Code
a	1
b	100000
c	00

C

Message	Code
a	0
b	10
c	110
d	111

D

Table 3.1: Examples of Distinct, Uniquely Decodable and Prefix Free Codes

Definition 4 *A uniquely decodable code is a prefix-free code if no codeword is prefix of any other codeword.*

Note that the codes in Table 3.1 (A,D) are prefix free. Prefix codes have the desirable property that they are instantaneously decodable. That is the code message can be parsed into codewords without the need for look ahead. If the source messages are also prefix free then the code is both instantaneously encodeable and decodable. There is no need for look ahead in either the coding or decoding process. However, the prefix property is not needed to ensure that a code be uniquely decodable. The code in Table 3.1 (C) is uniquely decodable but it is not prefix free. In order to decode a message encoded using the codeword set $\{1, 100000, 00\}$ look ahead is required. Note that the first codeword of the message 1000000001 is 1 but this cannot be determined until all ten symbols have been read. The algorithm for determining whether a 1 corresponds to the codeword 1 or the codeword 100000 is based on determining the parity of the number of zeros that follow the 1. Even though this code is not prefix free, it is uniquely decodable. However, decoding requires look ahead.

$aa \leftrightarrow 00$	$aa \leftrightarrow 0$	$a \leftrightarrow 00$	$a \leftrightarrow 0$
$ab \leftrightarrow 01$	$ab \leftrightarrow 10$	$ba \leftrightarrow 01$	$ba \leftrightarrow 10$
$ba \leftrightarrow 10$	$ba \leftrightarrow 110$	$bba \leftrightarrow 10$	$bba \leftrightarrow 110$
$bb \leftrightarrow 11$	$bb \leftrightarrow 111$	$bbb \leftrightarrow 11$	$bbb \leftrightarrow 111$
A	B	C	D

Table 3.2: Examples of Prefix Free Codes Belonging to Different Classes

3.2 Classification of Prefix Free Codes

Prefix free codes can be classified into four classes depending on the number of source alphabets in the source strings and the number of code alphabets in the code strings [63]. The first class consists of prefix free codes in which both the number of source alphabets in the source strings and the number of code alphabets in the code strings is fixed. These codes are referred to as the block to block prefix free codes. A common example is the ASCII representation of the alphanumeric characters. Another example is shown in Table 3.2 (A). Codes in which the number of source alphabets is fixed but the number of code alphabets is allowed to vary are referred to as block to variable length prefix free codes. A common example is Huffman codes [87, 27, 51], discussed later in this chapter. Another example is illustrated in Table 3.2 (B). Note that all the source messages have 2 symbols but the number of code alphabets in the codewords varies from one to three. Codes in which the number of source alphabets is allowed to vary but the number of code alphabets is fixed are referred to as variable length to block codes. An example is shown in Table 3.2 (C). Note that the number of source alphabets in the source messages varies from one to three but the number of code alphabets in the codewords is fixed to two. The most general class of codes consists of prefix free codes in which both the number of source and code alphabets is allowed to vary. A variable length to variable length prefix free code is shown in Table 3.2 (D).

Numerous variants of prefix free code construction problems can be defined. Generally, we are given the source and code alphabets. All source alphabets may have the same probability or the probabilities may differ. Furthermore, the code alphabets may have the same transmission times or the transmission times may differ.

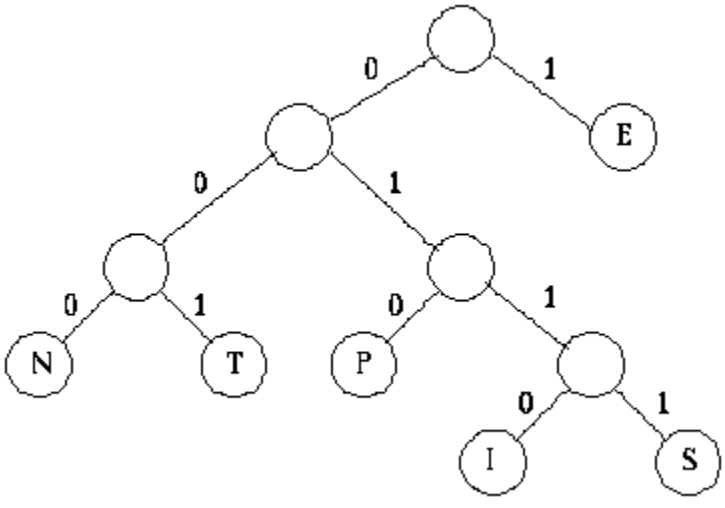


Figure 3.1: An Optimum Huffman Code for (E,29), (I,5), (N,7), (P,12), (S,4), (T,8)

Given the source and code alphabets, the probability of the source alphabets and the transmission times of the code alphabets, the problem is design a general code construction algorithm that minimizes or maximizes a given objective function. The objective function is generally the expected transmission time. Other variants of the code construction problem have been studied. In these variants in addition to the above, the code strings must satisfy some constraint, for example, all code strings must end in a given code alphabet.

3.3 Results on Prefix Free Coding

Prefix free codes that have been most extensively studied in literature are block to variable length prefix free codes. The problem where the probabilities of the source alphabet are different but the transmission costs of the code alphabet are the same was first studied in 1948 by Shannon [87] and in 1949 by Fano [27] who developed essentially identical methods for constructing near optimum codes. In 1952 Huffman [51] used an elegant combinatorial technique to obtain a strictly optimum solution to the problem. These codes are referred to as Huffman codes and an example is discussed next. Let the source alphabet be $\{E, I, N, P, S, T\}$ and the frequency of

occurrence be $(E, 29)$, $(I, 5)$, $(N, 7)$, $(P, 12)$, $(S, 4)$ and $(T, 8)$. The optimum Huffman Code is shown in Figure 3.1. In 1954 Blachman [12] generalized the Shannon-Fano approximation technique to treat the situation where both the probabilities of the source alphabet and the transmission costs of the code alphabet differ. In 1957 Marcus [70] improved on the Blachman technique by combining it further with the combinatorial results of Huffman. The algorithms of both Blachman and Marcus are approximate in nature and give near optimum solutions. Karp [57] developed an optimum algebraic solution by reducing the problem of code construction to an integer linear program which was solved using Gomory's integer programming algorithm [39, 40, 41]. Other approximation techniques have been developed by Krause [62], Cot [25, 26], Mehlhorn [72], Altenkamp and Mehlhorn [2] and Gilbert [35]. A dynamic programming algorithm for an exact solution has been developed by Golin and Rote [37] is $O(n^{C+2})$ where n is the number of source words and the transmission cost of the code alphabet belongs to integers from 1 to C. No polynomial time algorithm for arbitrary transmission costs is known. The special case of constructing block to variable length prefix free codes where probabilities of source messages is fixed but transmission costs of the code alphabets is allowed to vary was first studied by Varn [99] in 1971 and is also referred to as Varn coding. Exact solutions have also been proposed by Perl et.al. [78], Choy and Wong [21], Cot [24], Stanfel [93], Kapoor and Reingold [56] and Golin and Young [36]. In 1990 Berger and Yeung [9] introduced a new class of prefix free codes having the property that each codeword ends with a one. A useful application of 1-ended prefix codes is considered by Capocelli et. al. [18, 20], where is it shown how to construct from a given 1-ended prefix code a self synchronizing prefix code having the same codeword lengths. The alphabetic version of 1-ended prefix codes has been studied by Browning and Thomas [16] and Capocelli, et. al. [19]. Synchronizing codes deal with limiting error propagation when variable length prefix free codes are used. Work on variable length to block prefix free codes was pioneered by Tunstall [98]. Compared to block to variable length prefix-free codes and variable length to block prefix free codes, variable length to variable length prefix free codes have received very little attention in literature.

Chapter 4

Interval Modulation Codes

4.1 Review and Examples

Generally, prefix free codes are used for source coding. However, in our case we would like to use them for channel coding. The data stream to be transmitted consists of binary data, 0s and 1s. These are our source alphabets. Furthermore, the modulated signal can be represented by a sequence of code alphabets which have different transmission times. An Interval Modulation Code is a variable length to variable length prefix free code defined by the source alphabet, 0 and 1, and the code alphabet which consists of a finite set of symbols with different transmission times. In this thesis, we will formulate an algorithm to construct interval modulation codes to maximize worst case data transmission rate. That is minimize the time required to transmit the worst binary data stream. We define S to be the set of permissible time intervals. For expository purposes assume S is $\{1, 2\}$. A simple encoding strategy would be to map every binary 0 to a 1 and every binary 1 to a 2. This only achieves a worst case rate of 0.5 since a sequence of T 1s would require $2T$ time units to transmit. An alternate strategy would be to map $00 \rightarrow 111$, $01 \rightarrow 12$, $10 \rightarrow 21$, $110 \rightarrow 112$ and $111 \rightarrow 22$. Note that the sets $\{00, 01, 10, 110, 111\}$ and $\{111, 12, 21, 112, 22\}$ are both prefix-free. In the worst case this scheme would achieve a rate of 0.66 since a sequence of $2T$ 0s would require $3T$ time units to transmit. Since neither the number of bits nor the number of symbols is fixed, this is a variable length to variable length prefix-free code. Notice that the maximum number of bits that the encoder might

have to examine before it can encode part of the binary sequence is 3 (corresponding to 110 or 111). This is the delay associated with the encoder and is referred to as T_E. Similarly the decoder might have to wait up to 4 units of time before it can map part of the received symbol sequence back to bits (corresponding to 112 or 22). This is the delay associated with the decoder and is referred to as T_D. More complex variable length to variable length prefix-free codes are shown in Table 4.1 and Table 4.2.

4.2 Problem Formulation and Code Construction Using Integer Linear Programming

We have already defined the set of permissible time intervals or symbols to be S. We assume that all elements in S are positive integers[1]. Furthermore, we assume that S has at least 2 elements. Let m be the largest element in S. For $i \in \{1, 2, ..., m-1, m\}$, let

$$K_i = \begin{cases} 1 & i \in S \\ 0 & i \notin S \end{cases} \tag{4.1}$$

We define $N_K(T)$ to be the number of sequences of length T whose elements are in S.

$$N_K(T) = \begin{cases} \sum_{i=1}^{m} K_i N_K(T-i) & T > m \\ K_T + \sum_{i=1}^{T-1} K_i N_K(T-i) & 2 \le T \le m \\ K_1 & T = 1 \end{cases} \tag{4.2}$$

It is easy to show that $N_K(T)$ can be computed using the recurrence above. When $T = 1$, $N_K(T)$ is 1 if $K_1 = 1$ ($1 \in S$) and 0 if $K_1 = 0$ ($0 \notin S$). Hence, when $T = 1$, $N_K(T) = K_1$. Now consider the case when $2 \le T \le m$. The number of sequences of length T that end in T is K_T (1 if $T \in S$, 0 if $T \notin S$). The number of sequences of length T that end in i is $K_i N_K(T-i)$ ($N_K(T-i)$ if $T \in S$, 0 if $T \notin S$). Hence,

[1] If the symbols are not integers, they must be suitably scaled and truncated or rounded.

$T_E = 5$ $T_D = 2.73$ $R = 1.83$	$T_E = 6$ $T_D = 3.16$ $R = 1.87$	$T_E = 7$ $T_D = 3.63$ $R = 1.92$
$00 \leftrightarrow S_1$	$00 \leftrightarrow S_1$	$00 \leftrightarrow S_1$
$010 \leftrightarrow S_3$	$010 \leftrightarrow S_3$	$010 \leftrightarrow S_4$
$011 \leftrightarrow S_4$	$011 \leftrightarrow S_4$	$0110 \leftrightarrow S_6$
$100 \leftrightarrow S_5$	$100 \leftrightarrow S_5$	$0111 \leftrightarrow S_7$
$1010 \leftrightarrow S_6$	$1010 \leftrightarrow S_6$	$10000 \leftrightarrow S_2S_1$
$1011 \leftrightarrow S_7$	$1011 \leftrightarrow S_7$	$10001 \leftrightarrow S_2S_2$
$1100 \leftrightarrow S_2S_1$	$1100 \leftrightarrow S_2S_1$	$10010 \leftrightarrow S_3S_1$
$11010 \leftrightarrow S_2S_2$	$11010 \leftrightarrow S_2S_2$	$10011 \leftrightarrow S_8$
$11011 \leftrightarrow S_8$	$11011 \leftrightarrow S_8$	$10100 \leftrightarrow S_2S_3$
$11100 \leftrightarrow S_2S_3$	$11100 \leftrightarrow S_2S_3$	$10101 \leftrightarrow S_3S_2$
$11101 \leftrightarrow S_2S_4$	$11101 \leftrightarrow S_2S_4$	$10110 \leftrightarrow S_3S_3$
$11110 \leftrightarrow S_9$	$11110 \leftrightarrow S_9$	$10111 \leftrightarrow S_2S_4$
$11111 \leftrightarrow S_2S_5$	$111110 \leftrightarrow S_2S_5$	$11000 \leftrightarrow S_9$
	$111111 \leftrightarrow S_2S_7$	$11001 \leftrightarrow S_5S_1$
		$110100 \leftrightarrow S_3S_4$
		$110101 \leftrightarrow S_2S_5$
		$110110 \leftrightarrow S_5S_2$
		$110111 \leftrightarrow S_3S_5$
		$111000 \leftrightarrow S_5S_3$
		$111001 \leftrightarrow S_{10}$
		$111010 \leftrightarrow S_2S_6$
		$111011 \leftrightarrow S_5S_4$
		$111100 \leftrightarrow S_3S_6$
		$1111010 \leftrightarrow S_2S_7$
		$1111011 \leftrightarrow S_5S_5$
		$1111100 \leftrightarrow S_3S_7$
		$1111101 \leftrightarrow S_5S_6$
		$1111110 \leftrightarrow S_3S_8$
		$1111111 \leftrightarrow S_5S_7$

Table 4.1: Code with $S = \{1.00, 1.13, 1.27, 1.42, 1.60, 1.80, 2.03, 2.28, 2.57, 2.89\}$

$T_E = 4$ $T_D = 4.46$ $R = 0.87$	$T_E = 6$ $T_D = 6.75$ $R = 0.89$	$T_E = 7$ $T_D = 7.70$ $R = 0.91$
$00 \leftrightarrow S_1$	$00 \leftrightarrow S_1$	$000 \leftrightarrow S_4$
$01 \leftrightarrow S_2$	$010 \leftrightarrow S_4$	$0010 \leftrightarrow S_6$
$100 \leftrightarrow S_3$	$0110 \leftrightarrow S_6$	$0011 \leftrightarrow S_1S_1$
$101 \leftrightarrow S_4$	$0111 \leftrightarrow S_2S_1$	$0100 \leftrightarrow S_1S_2$
$110 \leftrightarrow S_5$	$1000 \leftrightarrow S_7$	$0101 \leftrightarrow S_2S_1$
$1110 \leftrightarrow S_6$	$10010 \leftrightarrow S_2S_2$	$01100 \leftrightarrow S_1S_3$
$1111 \leftrightarrow S_7$	$10011 \leftrightarrow S_3S_1$	$01101 \leftrightarrow S_3S_1$
	$10100 \leftrightarrow S_2S_3$	$01110 \leftrightarrow S_2S_3$
	$10101 \leftrightarrow S_3S_2$	$01111 \leftrightarrow S_3S_2$
	$10110 \leftrightarrow S_8$	$10000 \leftrightarrow S_1S_4$
	$10111 \leftrightarrow S_3S_3$	$10001 \leftrightarrow S_8$
	$11000 \leftrightarrow S_2S_4$	$10010 \leftrightarrow S_3S_3$
	$11001 \leftrightarrow S_5S_1$	$10011 \leftrightarrow S_2S_4$
	$11010 \leftrightarrow S_3S_4$	$10100 \leftrightarrow S_1S_5$
	$110110 \leftrightarrow S_2S_5$	$10101 \leftrightarrow S_5S_1$
	$110111 \leftrightarrow S_5S_2$	$101100 \leftrightarrow S_3S_4$
	$111000 \leftrightarrow S_9$	$101101 \leftrightarrow S_2S_5$
	$111001 \leftrightarrow S_3S_5$	$101110 \leftrightarrow S_5S_2$
	$111010 \leftrightarrow S_5S_3$	$101111 \leftrightarrow S_9$
	$111011 \leftrightarrow S_2S_6$	$110000 \leftrightarrow S_1S_6$
	$111100 \leftrightarrow S_5S_4$	$110001 \leftrightarrow S_3S_5$
	$111101 \leftrightarrow S_3S_6$	$110010 \leftrightarrow S_5S_3$
	$111110 \leftrightarrow S_{10}$	$110011 \leftrightarrow S_2S_6$
	$111111 \leftrightarrow S_2S_7$	$110100 \leftrightarrow S_5S_4$
		$110101 \leftrightarrow S_1S_7$
		$110110 \leftrightarrow S_7S_1$
		$110111 \leftrightarrow S_3S_6$
		$111000 \leftrightarrow S_2S_2S_1$
		$1110010 \leftrightarrow S_{10}$
		$1110011 \leftrightarrow S_2S_7$
		$1110100 \leftrightarrow S_7S_2$
		$1110101 \leftrightarrow S_5S_5$
		$1110110 \leftrightarrow S_2S_2S_2$
		$1110111 \leftrightarrow S_3S_7$
		$1111000 \leftrightarrow S_7S_3$
		$1111001 \leftrightarrow S_1S_8$
		$1111010 \leftrightarrow S_2S_2S_3$
		$1111011 \leftrightarrow S_5S_6$
		$1111100 \leftrightarrow S_2S_8$
		$1111101 \leftrightarrow S_7S_4$
		$1111110 \leftrightarrow S_2S_2S_4$
		$1111111 \leftrightarrow S_3S_8$

Table 4.2: Code with $S = \{2.00, 2.29, 2.61, 2.99, 3.41, 3.90, 4.46, 5.09, 5.82, 6.65\}$

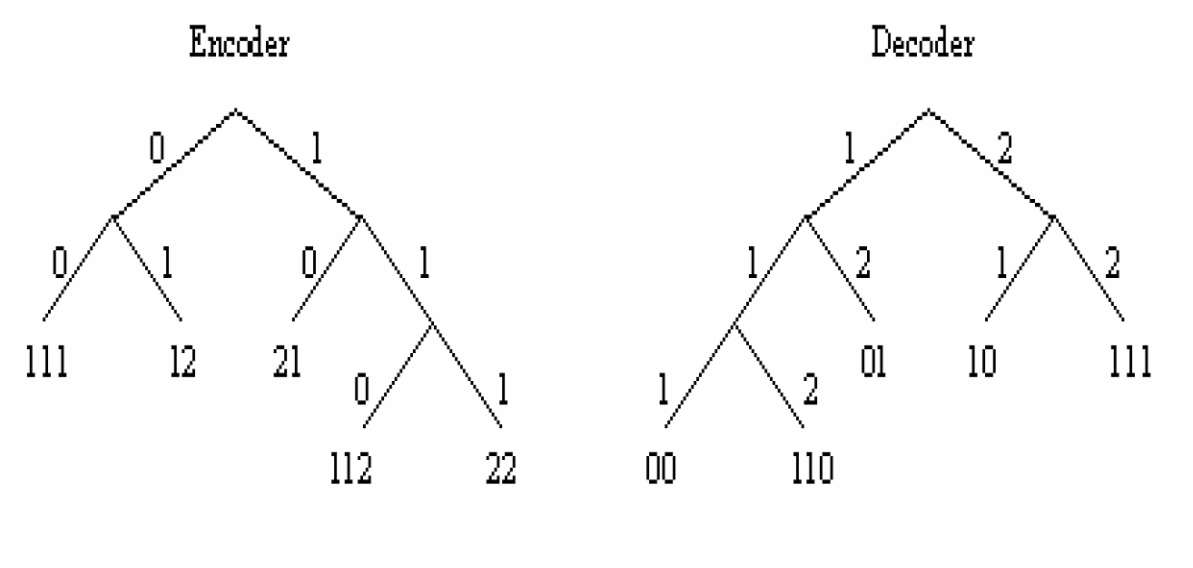


Figure 4.1: A Simple Interval Modulation Code Implemented Using Prefix Trees

$N_K(T) = K_T + \sum_{i=1}^{T-1} K_i N_K(T-i)$. The proof for when $T > m$ is similar and left as an exercise for the reader. It is easy to show that N_K is a generalization of the Fibonacci Numbers.

Example 5 *First consider the case when* $S = \{1, 2, 3\}$. *In this case* $m = 3$ *and* $K_1 = 1$, $K_2 = 1$ *and* $K_3 = 1$. *Using the recurrence above, we find* $N_K(5) = 13$. *The* 13 *sequences of length* 5 *are* $\{11111, 1112, 1121, 1211, 2111, 122, 212, 221, 113, 131, 311, 23, 32\}$. *Now consider the case when* $S = \{2, 3, 5\}$. *In this case* $m = 5$ *and* $K_1 = 0$, $K_2 = 1$, $K_3 = 1$, $K_4 = 0$ *and* $K_5 = 1$. *Using the recurrence above we find* $N_K(9) = 8$. *The* 8 *sequences of length* 9 *are* $\{2223, 2232, 2322, 3222, 333, 225, 252, 522\}$.

Note that variable length to variable length prefix-free codes can be implemented using prefix trees. The coding technique outlined in the previous section for $S = \{1, 2\}$ is implemented using prefix trees in Figure 4.1.

Definition 6 *For each leaf node* x *of the encoder tree* T, *let* $d_e(x)$ *denote the length of the path from the root to* x, *called the "encoder delay" of* x. *Also, let* $d_d(x) =$

$x_1 + ... + x_n$, *where* $x_1, ..., x_n$ *is in* S^n *is the label of* x*, called the "decoder delay" of* x*.*

Let $\max_x\ d_e(x)$ *and let* $\max_x\ d_d(x)$ *be called the "maximum encoder"and "maximum decoder" delays, respectively, taken over all leaf nodes* x *of the tree* T*. We will refer to these as* T_E *and* T_D *respectively. Also define the "rate" to be* $R = \min_x d_e(x)/d_d(x)$*.*

Before we define the code construction problem and present a reduction to integer linear programming, we would like to prove a theorem.

Notation 7 Σ_S^+ *is the set of non-null strings over* S

For all $s \in \Sigma_S^+$, $\|s\|$ *is the sum of the elements in* s

$\Pi_S^l = \{s | s \in \Sigma_S^+ \text{ and } \|s\| = l\}$

Assume $p \in \Sigma_S^+$ *and* $l \geq \|p\|$*. We define* $P_A(p, l) = \{s | s \in \Sigma_S^+ \text{ and } \|s\| = l \text{ and } p \text{ is a prefix of } s\}$

Theorem 8 *We are given* $M_S \subseteq \Sigma_S^+$ *such that* M_S *is a prefix-free set. For all* i *let* $y_i = |\{s | s \in M_S \text{ and } \|s\| = i\}|$.

For all n, $y_n \leq N_K(n) - \sum_{l=1}^{n-1} N_K(n-l) y_l$

Furthermore, if we are given y_i *such that they satisfy the constraints above, we can find* $M_S \subseteq \Sigma_S^+$ *such that* M_S *is a prefix-free set and* $|\{s | s \in M_S \text{ and } \|s\| = i\}| = y_i$*.*

Proof. We are given n. Define $\bar{M}_S(n) = \{s | s \in M_S \text{ and } \|s\| < n\}$ and $\bar{R}_S(n) = \{s | s \in M_S \text{ and } \|s\| = n\}$.

Note that, $\bar{R}_S(n) \cup \bigcup_{s \in \bar{M}_S(n)} P_S(s, n) \subseteq \Pi_S^n$

Hence, $\left| \bar{R}_S(n) \cup \bigcup_{s \in \bar{M}_S(n)} P_S(s, n) \right| \leq |\Pi_S^n|$

It is easy to show that $\bar{R}_S(n) \cap \bigcup_{s \in \bar{M}_S(n)} P_S(s,n) = \phi$ and since $\bar{M}_S(n)$ is a prefix-free set for all $s_1,\ s_2 \in \bar{M}_S(n)$, $P_S(s_1,n) \cap P_S(s_2,n) = \phi$.

Hence, $\left|\bar{R}_S(n)\right| + \sum_{s \in \bar{M}_S(n)} |P_S(s,n)| \leq |\Pi_S^n|$

Hence, $y_n \leq N_K(n) - \sum_{l=1}^{n-1} N_K(n-l) y_l$

The proof of the converse is constructive. Note that $y_1 \leq N_K(1) = K_1$. If $K_1 = 0$, let $M_S = \{\}$ and if $K_1 = 1$, let $M_S = \{1\}$. Now we will assume that we have constructed a prefix-free set such that for $i \in \{1, ..., n-1\}$, $|\{s | s \in M_S \text{ and } \|s\| = i\}| = y_i$.

Note that, $y_n \leq N_K(n) - \sum_{l=1}^{n-1} N_K(n-l) y_l$

Now let $U = \Pi_S^n - \bigcup_{s \in M_S} P_S(s,n)$

Hence, $|U| = \left|\Pi_S^n - \bigcup_{s \in M_S} P_S(s,n)\right|$

First note that $\bigcup_{s \in M_S} P_S(s,n) \subseteq \Pi_S^n$. Also note that since $M_S(n)$ is a prefix-free set for all $s_1,\ s_2 \in M_S(n)$, $P_S(s_1,n) \cap P_S(s_2,n) = \phi$.

Hence, $|U| = N_K(n) - \sum_{l=1}^{n-1} N_K(n-l) y_l$

Hence, $y_n \leq |U|$. It suffices to pick y_n elements from U and add them to M_S. ■

Problem 9 *Given numbers T_E, T_D, and R, and a set of positive integers S, find a tree T with the fewest possible leaves such that the maximum encoder delay is T_E, the maximum decoder delay is T_D, the rate is at least R, and the labels of the leaves form a prefix-free set over S.*

For $l \in \{1, 2, ..., T_D\}$ and $d \in \{1, 2, ..., T_E\}$, let x_d^l be the number of leafs (non negative integer) in the encoder tree at depth d which have a label of length l. Also, we define x^l to be the number of labels of length l and we define x_d to be the number of leafs in the encoder at depth d. Formally,

$$x^l = \sum_{i=1}^{T_E} x_i^l \text{ and } x_d = \sum_{i=1}^{T_D} x_d^i \tag{4.3}$$

The problem is to design the smallest encoder decoder pair. So we would like to,

$$\min \sum_{i=1}^{T_D} \sum_{j=1}^{T_E} x_j^i \tag{4.4}$$

Firstly, since the encoder tree is a full prefix tree, the Kraft Inequality for full trees, tells us that

$$\sum_{i=1}^{T_E} 2^{(T_E - i)} x_i = 2^{T_E} \tag{4.5}$$

Furthermore the labels attached to the leaf nodes of the encoder tree must form a prefix-free set over S. Equivalently, the decoder must be a prefix tree. From Theorem 8 we know that,

$$\text{For } 1 \leq L \leq T_D,\ x^L + \sum_{l=1}^{L-1} N_K(L-l) x^l \leq N_K(L) \tag{4.6}$$

Also the encoder and decoder must achieve a rate of R. Hence,

$$j/i < R \implies x_j^i = 0 \tag{4.7}$$

It is easy to verify that equations (4.5,4.6,4.7) are also sufficient. If we are given x_j^i which satisfy these constraints we can construct a code which achieves the desired rate and has the specified delays. Since equation (4.5) is satisfied, the Kraft inequality for full binary trees tells us that we can construct a full binary tree such that the number of leafs at depth i is x_i. Since equation (4.6) is satisfied, Theorem 8 tells us

that we can find a prefix-free set M_S over S such that the number of labels in M_S of length l is x^l. We can now arbitrarily assign x_j^i labels of length i from the set M_S to leaf nodes of the encoder tree at depth j in a one on one manner.

4.3 Linear Programming Relaxation

Equations (4.4,4.5,4.6,4.7) represent an integer linear program. Equations (4.4,4.5,4.6,4.7) without the integrality constraint are the linear programming relaxation of the integer linear program. Although the solution to the linear programming relaxation does not provide a code, it does provide a lower bound on the size of the optimal code. In this chapter we will present a polynomial time[2] algorithm for finding an optimal solution to the linear programming relaxation, if one exists. We next formulate a necessary and sufficient condition for the linear programming relaxation to have a solution.

Note that equation (4.5) represents a hyperplane and equations (4.6,4.7) represent a convex polyhedra. For a solution to exist, this hyperplane must intersect the convex polyhedra. It must be noted that this criteria is both necessary and sufficient for the linear programming relaxation of the integer linear program to have a solution. However, it is not sufficient, only necessary to ensure the integer linear program has a solution. The hyperplane intersects that polyhedra if and only if $\max \sum_{i=1}^{T_E} 2^{(T_E-i)} x_i$ subject to equations (4.6,4.7) is greater than or equal to 2^{T_E}. In this chapter we also present a polynomial time algorithm for performing this optimization.

4.4 Generalized Fibonacci Polyhedra and their Properties

First we will define a Generalized Fibonacci Polyhedra. This definition is motivated by the matrix representation of the system of inequalities in equation (4.6).

[2]We define a polynomial time algorithm to be one that requires a polynomial number of arithmetic operations, integer or floating point.

Notation 10 *We let I^n be the $n \times n$ identity matrix and we let A^n be the $n \times n$ matrix below,*

$$A^n = \begin{bmatrix} 1 & N_K(1) & . & N_K(n-2) & N_K(n-1) \\ 0 & 1 & . & N_K(n-3) & N_K(n-2) \\ . & . & . & . & . \\ 0 & 0 & . & 1 & N_K(1) \\ 0 & 0 & . & 0 & 1 \end{bmatrix}$$

We refer to the ith column of A^n as A^n_i and the ith column of I^n as I^n_i. Note that A^n_i can be computed recursively as follows,

$$A^n_i = \begin{cases} I^n_i + \sum_{j=1}^{m} K_j A^n_{i-j} & i > m \\ I^n_i + \sum_{j=1}^{i-1} K_j A^n_{i-j} & 2 \leq i \leq m \\ I^n_1 & i = 1 \end{cases}$$

Definition 11 *We are given $K \in \{0,1\}^m$ and $\alpha \in \{0, R^+\}^n$. Let $\Gamma(K, \alpha) = \{y | y \in \{0, R^+\}^n$ and $A^n y \leq A^n \alpha\}$. Equivalently if $s \in \{0, R^+\}^n$ is the vector of slack variables we have $\Gamma(K, \alpha) = \{y | y, s \in \{0, R^+\}^n$ and $A^n y + I^n s = A^n \alpha\}$. We will refer to $\Gamma(K, \alpha)$ as a Generalized Fibonacci Polyhedra of dimension n and order m.*

Before we state and prove properties satisfied by Generalized Fibonacci Polyhedra we will prove a lemma.

Lemma 12 *We are given $K \in \{0,1\}^m$, $\alpha \in \{0, R^+\}^n$ and $y \in \Gamma(K, \alpha)$. Let s be the vector of slack variables corresponding to y.*

If $2 \leq n \leq m$,
$\sum_{i=1}^{n-1} y_i A^{n-1}_i + \sum_{i=1}^{n-1} s_i I^{n-1}_i = \sum_{i=1}^{n-1} [\alpha_i + (\alpha_n - y_n) K_{n-i}] A^{n-1}_i$
If $n > m$,
$\sum_{i=1}^{n-1} y_i A^{n-1}_i + \sum_{i=1}^{n-1} s_i I^{n-1}_i$

$$= \sum_{i=1}^{n-m-1} \alpha_i A_i^{n-1} + \sum_{i=n-m}^{n-1} [\alpha_i + (\alpha_n - y_n) K_{n-i}] A_i^{n-1}$$

Proof. Since $y \in \Gamma(K, \alpha)$, $Ay + Is = A\alpha$. Hence,

$$\sum_{i=1}^{n} y_i A_i^n + \sum_{i=1}^{n} s_i I_i^n = \sum_{i=1}^{n} \alpha_i A_i^n$$

Note that $y_n + s_n = \alpha_n$. Hence,

$$\sum_{i=1}^{n-1} y_i A_i^n + \sum_{i=1}^{n-1} s_i I_i^n = \sum_{i=1}^{n-1} \alpha_i A_i^n + \alpha_n A_n^n - y_n A_n^n - s_n I_n^n$$
$$= \sum_{i=1}^{n-1} \alpha_i A_i^n + (\alpha_n - y_n)(A_n^n - I_n^n)$$

Now consider the case when $2 \leq n \leq m$. In this case $A_n^n = I_n^n + \sum_{i=1}^{n-1} K_i A_{n-i}^n$. Hence,

$$\sum_{i=1}^{n-1} y_i A_i^n + \sum_{i=1}^{n-1} s_i I_i^n$$
$$= \sum_{i=1}^{n-1} \alpha_i A_i^n + (\alpha_n - y_n)(I_n^n + \sum_{i=1}^{n-1} K_i A_{n-i}^n - I_n^n)$$
$$= \sum_{i=1}^{n-1} \alpha_i A_i^n + (\alpha_n - y_n) \sum_{i=1}^{n-1} K_{n-i} A_i^n$$
$$= \sum_{i=1}^{n-1} [\alpha_i + (\alpha_n - y_n) K_{n-i}] A_i^n$$

Hence,

$$\sum_{i=1}^{n-1} y_i A_i^{n-1} + \sum_{i=1}^{n-1} s_i I_i^{n-1} = \sum_{i=1}^{n-1} [\alpha_i + (\alpha_n - y_n) K_{n-i}] A_i^{n-1}$$

If $n > m$, $A_n^n = I_n^n + \sum_{i=1}^{m} K_i A_{n-i}^n$. We can show,

$$\sum_{i=1}^{n-1} y_i A_i^n + \sum_{i=1}^{n-1} s_i I_i^n$$

$$= \sum_{i=1}^{n-1} \alpha_i A_i^n + (\alpha_n - y_n) \sum_{i=n-m}^{n-1} K_{n-i} A_i^n$$
$$= \sum_{i=1}^{n-m-1} \alpha_i A_i^n + \sum_{i=n-m}^{n-1} [\alpha_i + (\alpha_n - y_n) K_{n-i}] A_i^n$$

Hence,

$$\sum_{i=1}^{n-1} y_i A_i^{n-1} + \sum_{i=1}^{n-1} s_i I_i^{n-1}$$
$$= \sum_{i=1}^{n-m-1} \alpha_i A_i^{n-1} + \sum_{i=n-m}^{n-1} [\alpha_i + (\alpha_n - y_n) K_{n-i}] A_i^{n-1} \quad \blacksquare$$

4.4.1 The 0-1 Principle

Before we prove the 0-1 Principle, we will introduce some notation and prove a lemma.

Definition 13 *Let $a, b \in \{0, R^+\}^n$. We say that $a \subseteq b$ iff for all $i \in \{1, ..., n\}$, $a_i > 0$ implies $b_i > 0$.*

Lemma 14 *We are given $K \in \{0,1\}^m$, $\alpha \in \{0, R^+\}^n$ and $y \in \Gamma(K, \alpha)$. If we are given $\bar{\alpha} \in \{0, R^+\}^n$ such that $\bar{\alpha} \subseteq \alpha$, then we can find $\bar{y} \in \Gamma(K, \bar{\alpha})$ such that $\bar{y} \subseteq y$ and $\bar{s} \subseteq s$, where s is the vector of slack variables corresponding to y and $\bar{s}$ is the vector of slack variables corresponding to $\bar{y}$.*

Proof. The proof is by induction.

Base Case: $n = 1$. We are given $y_1 + s_1 = \alpha_1$. If $\alpha_1 = 0$ then $y_1 = 0$, $s_1 = 0$ and $\bar{\alpha}_1 = 0$. We let $\bar{y}_1 = 0$ and $\bar{s}_1 = 0$. Note that $\bar{y}_1 + \bar{s}_1 = \bar{\alpha}_1$ and $\bar{y} \subseteq y$ and $\bar{s} \subseteq s$. If $\alpha_1 > 0$ then let $\bar{y}_1 = \bar{\alpha}_1 y_1/\alpha_1$ and $\bar{s}_1 = \bar{\alpha}_1 s_1/\alpha_1$. Note that $\bar{y}_1 + \bar{s}_1 = \bar{\alpha}_1 y_1/\alpha_1 + \bar{\alpha}_1 s_1/\alpha_1 = \bar{\alpha}_1(y_1 + s_1)/\alpha_1 = \bar{\alpha}_1$. Furthermore $\bar{y}_1 > 0$ implies $y_1 > 0$ and $\bar{s}_1 > 0$ implies $s_1 > 0$. Hence, $\bar{y} \subseteq y$ and $\bar{s} \subseteq s$.

Inductive Hypothesis: We are given $K \in \{0,1\}^m$, $\alpha \in \{0, R^+\}^{n-1}$ and $y \in \Gamma(K,\alpha)$. If we are given $\bar{\alpha} \in \{0, R^+\}^{n-1}$ such that $\bar{\alpha} \subseteq \alpha$, then we can find $\bar{y} \in \Gamma(K,\bar{\alpha})$ such that $\bar{y} \subseteq y$ and $\bar{s} \subseteq s$, where s is the vector of slack variables corresponding to y and $\bar{s}$ is the vector of slack variables corresponding to $\bar{y}$.

We are given $K \in \{0,1\}^m$, $\alpha \in \{0, R^+\}^n$ and $y \in \Gamma(K,\alpha)$. Let s be the vector of slack variables corresponding to y. We will assume that $\alpha_n > 0$. If $\alpha_n = 0$ then $y_n = 0$ and $s_n = 0$ and the result follows trivially from the induction hypothesis. There are two cases to consider.

Case 1 : $2 \leq n \leq m$. Since $y \in \Gamma(K,\alpha)$, by Lemma 12,

$$\sum_{i=1}^{n-1} y_i A_i^{n-1} + \sum_{i=1}^{n-1} s_i I_i^{n-1} = \sum_{i=1}^{n-1} [\alpha_i + (\alpha_n - y_n) K_{n-i}] A_i^{n-1}$$

We let $\bar{y}_n = \bar{\alpha}_n y_n / \alpha_n$ and $\bar{s}_n = \bar{\alpha}_n s_n / \alpha_n$ (Note that $\bar{y}_n > 0$ implies $y_n > 0$ and $\bar{s}_n > 0$ implies $s_n > 0$). Next we show that $\bar{\alpha}_i + (\bar{\alpha}_n - \bar{y}_n) K_{n-i} > 0$ implies $\alpha_i + (\alpha_n - y_n) K_{n-i} > 0$. If $\bar{\alpha}_n - \bar{y}_n > 0$ then $\alpha_n - y_n > 0$. Hence, if $(\bar{\alpha}_n - \bar{y}_n) K_{n-i} > 0$ then $(\alpha_n - y_n) K_{n-i} > 0$. Since $\bar{\alpha}_i > 0$ implies $\alpha_i > 0$, we have $\bar{\alpha}_i + (\bar{\alpha}_n - \bar{y}_n) K_{n-i} > 0$ implies $\alpha_i + (\alpha_n - y_n) K_{n-i} > 0$. By the inductive hypothesis we can find $\bar{y}_1, ..., \bar{y}_{n-1}$and $\bar{s}_1, ..., \bar{s}_{n-1}$ such that $\bar{y}_i > 0$ implies $y_i > 0$ and $\bar{s}_i > 0$ implies $s_i > 0$ and,

$$\sum_{i=1}^{n-1} \bar{y}_i A_i^{n-1} + \sum_{i=1}^{n-1} \bar{s}_i I_i^{n-1} = \sum_{i=1}^{n-1} [\bar{\alpha}_i + (\bar{\alpha}_n - \bar{y}_n) K_{n-i}] A_i^{n-1}$$

$$\begin{aligned}
&\sum_{i=1}^{n-1} \bar{y}_i A_i^n + \sum_{i=1}^{n-1} \bar{s}_i I_i^n = \sum_{i=1}^{n-1} [\bar{\alpha}_i + (\bar{\alpha}_n - \bar{y}_n) K_{n-i}] A_i^n \\
&= \sum_{i=1}^{n-1} \bar{\alpha}_i A_i^n + \sum_{i=1}^{n-1} [(\bar{\alpha}_n - \bar{y}_n) K_i] A_{n-i}^n \\
&= \sum_{i=1}^{n-1} \bar{\alpha}_i A_i^n + (\bar{\alpha}_n - \bar{y}_n)(A_n^n - I_n^n)
\end{aligned}$$

Hence,

$$\sum_{i=1}^{n} \bar{y}_i A_i^n + \sum_{i=1}^{n} \bar{s}_i I_i^n$$
$$= \sum_{i=1}^{n-1} \bar{\alpha}_i A_i^n + (\bar{\alpha}_n - \bar{y}_n)(A_n^n - I_n^n) + \bar{y}_n A_n^n + \bar{s}_n I_n^n$$
$$= \sum_{i=1}^{n-1} \bar{\alpha}_i A_i^n + \bar{s}_n (A_n^n - I_n^n) + \bar{y}_n A_n^n + \bar{s}_n I_n^n$$
$$= \sum_{i=1}^{n-1} \bar{\alpha}_i A_i^n + \bar{s}_n A_n^n + \bar{y}_n A_n^n$$
$$= \sum_{i=1}^{n-1} \bar{\alpha}_i A_i^n + \bar{\alpha}_n A_n^n = \sum_{i=1}^{n} \bar{\alpha}_i A_i^n$$

Case 2 : $n > m$. Since $y \in \Gamma(K, \alpha)$, by Lemma 12,

$$\sum_{i=1}^{n-1} y_i A_i^{n-1} + \sum_{i=1}^{n-1} s_i I_i^{n-1}$$
$$= \sum_{i=1}^{n-m-1} \alpha_i A_i^{n-1} + \sum_{i=n-m}^{n-1} [\alpha_i + (\alpha_n - y_n) K_{n-i}] A_i^{n-1}$$

We let $\bar{y}_n = \bar{\alpha}_n y_n / \alpha_n$ and $\bar{s}_n = \bar{\alpha}_n s_n / \alpha_n$ (Note that $\bar{y}_n > 0$ implies $y_n > 0$ and $\bar{s}_n > 0$ implies $s_n > 0$). For $i \in \{1, ..., n-m-1\}$ $\bar{\alpha}_i > 0$ implies $\alpha_i > 0$. Next we show that for $i \in \{n-m, ..., n\}$ $\bar{\alpha}_i + (\bar{\alpha}_n - \bar{y}_n) K_{n-i} > 0$ implies $\alpha_i + (\alpha_n - y_n) K_{n-i} > 0$. If $\bar{\alpha}_n - \bar{y}_n > 0$ then $\alpha_n - y_n > 0$. Hence, if $(\bar{\alpha}_n - \bar{y}_n) K_{n-i} > 0$ then $(\alpha_n - y_n) K_{n-i} > 0$. Since $\bar{\alpha}_i > 0$ implies $\alpha_i > 0$, we have $\bar{\alpha}_i + (\bar{\alpha}_n - \bar{y}_n) K_{n-i} > 0$ implies $\alpha_i + (\alpha_n - y_n) K_{n-i} > 0$. By the inductive hypothesis we can find $\bar{y}_1, ..., \bar{y}_{n-1}$and $\bar{s}_1, ..., \bar{s}_{n-1}$ such that for $i \in \{1, ..., n-1\}$ $\bar{y}_i > 0$ implies $y_i > 0$ and $\bar{s}_i > 0$ implies $s_i > 0$ and,

$$\sum_{i=1}^{n-1} \bar{y}_i A_i^{n-1} + \sum_{i=1}^{n-1} \bar{s}_i I_i^{n-1}$$
$$= \sum_{i=1}^{n-m-1} \bar{\alpha}_i A_i^{n-1} + \sum_{i=n-m}^{n-1} [\bar{\alpha}_i + (\bar{\alpha}_n - \bar{y}_n) K_{n-i}] A_i^{n-1}$$

$$\sum_{i=1}^{n-1} \bar{y}_i A_i^{n} + \sum_{i=1}^{n-1} \bar{s}_i I_i^{n}$$
$$= \sum_{i=1}^{n-m-1} \bar{\alpha}_i A_i^{n} + \sum_{i=n-m}^{n-1} [\bar{\alpha}_i + (\bar{\alpha}_n - \bar{y}_n) K_{n-i}] A_i^{n}$$
$$= \sum_{i=1}^{n-1} \bar{\alpha}_i A_i^{n} + \sum_{i=n-m}^{n-1} (\bar{\alpha}_n - \bar{y}_n) K_{n-i} A_i^{n}$$

$$= \sum_{i=1}^{n-1} \bar{\alpha}_i A_i^n + \sum_{i=1}^{m} [(\bar{\alpha}_n - \bar{y}_n) K_i] A_{n-i}^n$$

Hence,

$$\begin{aligned}
&\sum_{i=1}^{n} \bar{y}_i A_i^n + \sum_{i=1}^{n} \bar{s}_i I_i^n \\
&= \sum_{i=1}^{n-1} \bar{\alpha}_i A_i^n + \sum_{i=1}^{m} [(\bar{\alpha}_n - \bar{y}_n) K_i] A_{n-i}^n + \bar{y}_n A_n^n + \bar{s}_n I_n^n \\
&= \sum_{i=1}^{n-1} \bar{\alpha}_i A_i^n + (\bar{\alpha}_n - \bar{y}_n)(A_n^n - I_n^n) + \bar{y}_n A_n^n + \bar{s}_n I_n^n \\
&= \sum_{i=1}^{n-1} \bar{\alpha}_i A_i^n + \bar{s}_n (A_n^n - I_n^n) + \bar{y}_n A_n^n + \bar{s}_n I_n^n \\
&= \sum_{i=1}^{n-1} \bar{\alpha}_i A_i^n + \bar{s}_n A_n^n + \bar{y}_n A_n^n \\
&= \sum_{i=1}^{n-1} \bar{\alpha}_i A_i^n + \bar{\alpha}_n A_n^n = \sum_{i=1}^{n} \bar{\alpha}_i A_i^n \quad \blacksquare
\end{aligned}$$

Now we are ready to state and prove the 0-1 Principle.

Theorem 15 *We are given $K \in \{0,1\}^m$ and $\alpha \in \{0, R^+\}^n$. If $y \in \{0, R^+\}^n$ is an extreme point of $\Gamma(K, \alpha)$ and $s \in \{0, R^+\}^n$ is the vector of slack variables then $(y_n, s_n) = (0, \alpha_n)$ or $(y_n, s_n) = (\alpha_n, 0)$.*

Proof. The proof is by contradiction. Assume y is an extreme point of $\Gamma(K, \alpha)$ and $s \in \{0, R^+\}^n$ is the corresponding vector of slack variables. Since y is an extreme point of $\Gamma(K, \alpha)$, $y \in \Gamma(K, \alpha)$. Hence $A^n y \leq A^n \alpha$ or equivalently $A^n y + I^n s = A^n \alpha$. Let V be the set of vectors that are used by y, s. Specifically $A_i^n \in V$ iff $y_i > 0$ and $I_i^n \in V$ iff $s_i > 0$. Since $A^n y + I^n s = A^n \alpha$, $y_n + s_n = \alpha_n$. Furthermore, $y_n \geq 0$ and $s_n \geq 0$. Hence $0 \leq y_n \leq \alpha_n$. If $\alpha_n = 0$ then $(y_n, s_n) = (0, 0)$. So we will assume $\alpha_n > 0$. Now assume $(y_n, s_n) \neq (0, \alpha_n)$ and $(y_n, s_n) \neq (\alpha_n, 0)$. Hence $0 < y_n < \alpha_n$ and $s_n = \alpha_n - y_n > 0$. We will show that if this is the case then the vectors in V are not linearly independent and consequently y is not an extreme point of $\Gamma(K, \alpha)$.

Case 1 : $n = 1$. We have assumed $\alpha_1 > 0$, $0 < y_1 < \alpha_1$ and $s_1 = \alpha_1 - y_1 > 0$. Since $y_1 > 0$, $A_1^1 \in V$. Since $s_1 > 0$, $I_1^1 \in V$. Since $A_1^1 \in V$, $I_1^1 \in V$ and $A_1^1 = I_1^1$, V

cannot be a linearly independent set of vectors.

Case 2 : $2 \leq n \leq m$. By Lemma 12 we know that,

$$\sum_{i=1}^{n-1} y_i A_i^n + \sum_{i=1}^{n-1} s_i I_i^n = \sum_{i=1}^{n-1} [\alpha_i + (\alpha_n - y_n) K_{n-i}] A_i^n$$

We have assumed $\alpha_n > 0$, $0 < y_n < \alpha_n$ and $s_n = \alpha_n - y_n > 0$. Since $y_n > 0$, $A_n^n \in V$. Since $s_n > 0$, $I_n^n \in V$. Now note that if $K_{n-i} > 0$ then $\alpha_i + (\alpha_n - y_n) K_{n-i} > 0$. This is because $(\alpha_n - y_n) > 0$ and $\alpha_i \geq 0$. By Lemma 14 we can find $\bar{y} \in \{0, R^+\}^{n-1}$ and the vector of slack variables $\bar{s} \in \{0, R^+\}^{n-1}$ such that for $i \in \{1, ..., n-1\}$ $\bar{y}_i > 0$ implies $y_i > 0$ and $\bar{s}_i > 0$ implies $s_i > 0$. Furthermore,

$$\sum_{i=1}^{n-1} \bar{y}_i A_i^n + \sum_{i=1}^{n-1} \bar{s}_i I_i^n = \sum_{i=1}^{n-1} K_{n-i} A_i^n = \sum_{i=1}^{n-1} K_i A_{n-i}^n$$

Note that $I_n^n \in V$ and

$$I_n^n + \sum_{i=1}^{n-1} \bar{y}_i A_i^n + \sum_{i=1}^{n-1} \bar{s}_i I_i^n = I_n^n + \sum_{i=1}^{n-1} K_i A_{n-i}^n = A_n^n$$

Hence, by using a subset of the vectors in V excluding A_n^n we have been able to generate A_n^n which is also in V. Hence the vectors in V are not linearly independent.

Case 3 : $n > m$. By Lemma 12 we know that,

$$\sum_{i=1}^{n-1} y_i A_i^n + \sum_{i=1}^{n-1} s_i I_i^n$$
$$= \sum_{i=1}^{n-m-1} \alpha_i A_i^n + \sum_{i=n-m}^{n-1} [\alpha_i + (\alpha_n - y_n) K_{n-i}] A_i^n$$

We have assumed $\alpha_n > 0$, $0 < y_n < \alpha_n$ and $s_n = \alpha_n - y_n > 0$. Since $y_n > 0$, $A_n^n \in V$. Since $s_n > 0$, $I_n^n \in V$. Now note that if $K_{n-i} > 0$ then $\alpha_i + (\alpha_n - y_n)K_{n-i} > 0$. This is because $(\alpha_n - y_n) > 0$ and $\alpha_i \geq 0$. By Lemma 14 we can find $\bar{y} \in \{0, R^+\}^{n-1}$ and the vector of slack variables $\bar{s} \in \{0, R^+\}^{n-1}$ such that for $i \in \{1, ..., n-1\}$ $\bar{y}_i > 0$ implies $y_i > 0$ and $\bar{s}_i > 0$ implies $s_i > 0$. Furthermore,

$$\sum_{i=1}^{n-1} \bar{y}_i A_i^n + \sum_{i=1}^{n-1} \bar{s}_i I_i^n = \sum_{i=n-m}^{n-1} K_{n-i} A_i^n = \sum_{i=1}^{m} K_i A_{n-i}^n$$

Note that $I_n^n \in V$ and

$$I_n^n + \sum_{i=1}^{n-1} \bar{y}_i A_i^n + \sum_{i=1}^{n-1} \bar{s}_i I_i^n = I_n^n + \sum_{i=1}^{m} K_i A_{n-i}^n = A_n^n$$

Hence, by using a subset of the vectors in V excluding A_n^n we have been able to generate A_n^n which is also in V. Hence the vectors in V are not linearly independent. ■

4.4.2 The Decomposition Principle

Lemma 16 *We are given $K \in \{0,1\}^m$, $\alpha \in \{0, R^+\}^n$ and $y \in \Gamma(K, \alpha)$. For $i \in \{1, ..., k\}$, we are given $\alpha^i \in \{0, R^+\}^n$ such that $\sum_{i=1}^k \alpha^i = \alpha$ then we can find $y^i \in \Gamma(K, \alpha^i)$ such that $\sum_{i=1}^k y^i = y$ and $\sum_{i=1}^k s^i = s$ where s is the vector of slack variables corresponding to y and s^i is the vector of slack variables corresponding to y^i.*

Proof. The proof is by induction.

Base Case: $n = 1$. We are given $y_1 + s_1 = \alpha_1$. If $\alpha_1 = 0$ then $y_1 = 0$ and $s_1 = 0$. Furthermore for all $i \in \{1, ..., k\}$ $\alpha^i = 0$ since $\sum_{i=1}^k \alpha^i = \alpha = 0$ and

$\alpha^i \in \{0, R^+\}^n$. For all $i \in \{1, ..., k\}$ we let $y_1^i = 0$ and $s_1^i = 0$. Note that $y_1^i + s_1^i = \alpha_1^i$ and $\sum_{i=1}^k y_1^i = y_1$ and $\sum_{i=1}^k s_1^i = s_1$. If $\alpha_1 > 0$ then we let $y_1^i = \alpha_1^i y_1/\alpha_1$ and $s_1^i = \alpha_1^i s_1/\alpha_1$. Note that $y_1^i + s_1^i = \alpha_1^i y_1/\alpha_1 + \alpha_1^i s_1/\alpha_1 = \alpha_1^i(y_1 + s_1)/\alpha_1 = \alpha_1^i$. Furthermore, $\sum_{i=1}^k y_1^i = \sum_{i=1}^k \alpha_1^i y_1/\alpha_1 = (y_1/\alpha_1)\sum_{i=1}^k \alpha_1^i = (y_1/\alpha_1)\alpha_1 = y_1$ and $\sum_{i=1}^k s_1^i = \sum_{i=1}^k \alpha_1^i s_1/\alpha_1 = (s_1/\alpha_1)\sum_{i=1}^k \alpha_1^i = (s_1/\alpha_1)\alpha_1 = s_1$.

Inductive Hypothesis: We are given $K \in \{0,1\}^m$, $\alpha \in \{0, R^+\}^{n-1}$ and $y \in \Gamma(K, \alpha)$. For $i \in \{1, ..., k\}$, we are given $\alpha^i \in \{0, R^+\}^{n-1}$ such that $\sum_{i=1}^k \alpha^i = \alpha$ then we can find $y^i \in \Gamma(K, \alpha^i)$ such that $\sum_{i=1}^k y^i = y$ and $\sum_{i=1}^k s^i = s$ where s is the vector of slack variables corresponding to y and s^i is the vector of slack variables corresponding to y^i.

We are given $K \in \{0,1\}^m$, $\alpha \in \{0, R^+\}^n$ and $y \in \Gamma(K, \alpha)$. Let s be the vector of slack variables corresponding to y. We will assume that $\alpha_n > 0$. If $\alpha_n = 0$ then $y_n = 0$ and $s_n = 0$ and the result follows trivially from the induction hypothesis. There are two cases to consider.

Case 1 : $2 \leq n \leq m$. Since $y \in \Gamma(K, \alpha)$, by Lemma 12,

$$\sum_{i=1}^{n-1} y_i A_i^{n-1} + \sum_{i=1}^{n-1} s_i I_i^{n-1} = \sum_{i=1}^{n-1} [\alpha_i + (\alpha_n - y_n)K_{n-i}]A_i^{n-1}$$

We let $y_n^j = \alpha_n^j y_n/\alpha_n$ and $s_n^j = \alpha_n^j s_n/\alpha_n$ (Note that $\sum_{j=1}^k y_n^j = \sum_{j=1}^k \alpha_n^j y_n/\alpha_n = (y_n/\alpha_n)\sum_{j=1}^k \alpha_n^j = (y_n/\alpha_n)\alpha_n = y_n$ and $\sum_{j=1}^k s_n^j = \sum_{j=1}^k \alpha_n^j s_n/\alpha_n = (s_n/\alpha_n)\sum_{j=1}^k \alpha_n^j = (s_n/\alpha_n)\alpha_n = s_n$). Next note that for $i \in \{1, ..., n-1\}$ $\sum_{j=1}^k \alpha_i^j + (\alpha_n^j - y_n^j)K_{n-i} = \sum_{j=1}^k \alpha_i^j + K_{n-i}\sum_{j=1}^k \alpha_n^j - K_{n-i}\sum_{j=1}^k y_n^j = \alpha_i + (\alpha_n - y_n)K_{n-i}$. By the inductive hypothesis we can find $\bar{y}_1, ..., \bar{y}_{n-1}$and $\bar{s}_1, ..., \bar{s}_{n-1}$ such that for $i \in \{1, ..., n-1\}$ $\sum_{j=1}^k y_i^j = y_i$ and $\sum_{j=1}^k s_i^j = s_i$ and,

$$\sum_{i=1}^{n-1} y_i^j A_i^{n-1} + \sum_{i=1}^{n-1} s_i^j I_i^{n-1} = \sum_{i=1}^{n-1} [\alpha_i^j + (\alpha_n^j - y_n^j) K_{n-i}] A_i^{n-1}$$

$$\sum_{i=1}^{n-1} y_i^j A_i^n + \sum_{i=1}^{n-1} s_i^j I_i^n = \sum_{i=1}^{n-1} [\alpha_i^j + (\alpha_n^j - y_n^j) K_{n-i}] A_i^n$$
$$= \sum_{i=1}^{n-1} \alpha_i^j A_i^n + \sum_{i=1}^{n-1} [(\alpha_n^j - y_n^j) K_i] A_{n-i}^n$$
$$= \sum_{i=1}^{n-1} \alpha_i^j A_i^n + (\alpha_n^j - y_n^j)(A_n^n - I_n^n)$$

Hence,

$$\sum_{i=1}^{n} y_i^j A_i^n + \sum_{i=1}^{n} s_i^j I_i^n$$
$$= \sum_{i=1}^{n-1} \alpha_i^j A_i^n + (\alpha_n^j - y_n^j)(A_n^n - I_n^n) + y_n^j A_n^n + s_n^j I_n^n$$
$$= \sum_{i=1}^{n-1} \alpha_i^j A_i^n + s_n^j (A_n^n - I_n^n) + y_n^j A_n^n + s_n^j I_n^n$$
$$= \sum_{i=1}^{n-1} \alpha_i^j A_i^n + s_n^j A_n^n + y_n^j A_n^n$$
$$= \sum_{i=1}^{n-1} \alpha_i^j A_i^n + \alpha_n^j A_n^n = \sum_{i=1}^{n} \alpha_i^j A_i^n$$

Case 2 : $n > m$. Since $y \in \Gamma(K, \alpha)$, by Lemma 12,

$$\sum_{i=1}^{n-1} y_i A_i^{n-1} + \sum_{i=1}^{n-1} s_i I_i^{n-1}$$
$$= \sum_{i=1}^{n-m-1} \alpha_i A_i^{n-1} + \sum_{i=n-m}^{n-1} [\alpha_i + (\alpha_n - y_n) K_{n-i}] A_i^{n-1}$$

We let $y_n^j = \alpha_n^j y_n / \alpha_n$ and $s_n^j = \alpha_n^j s_n / \alpha_n$ (Note that $\sum_{j=1}^k y_n^j = \sum_{j=1}^k \alpha_n^j y_n / \alpha_n = (y_n/\alpha_n) \sum_{j=1}^k \alpha_n^j = (y_n/\alpha_n)\alpha_n = y_n$ and $\sum_{j=1}^k s_n^j = \sum_{j=1}^k \alpha_n^j s_n / \alpha_n = (s_n/\alpha_n) \sum_{j=1}^k \alpha_n^j = (s_n/\alpha_n)\alpha_n = s_n$). Next note that for $i \in \{1, ..., n-m-1\}$ $\sum_{j=1}^k \alpha_i^j = \alpha_i$. And for $i \in \{n-m, ..., n\}$ $\sum_{j=1}^k [\alpha_i^j + (\alpha_n^j - y_n^j) K_{n-i}] = \sum_{j=1}^k \alpha_i^j + K_{n-i} \sum_{j=1}^k \alpha_n^j - K_{n-i} \sum_{j=1}^k y_n^j = \sum_{j=1}^k \alpha_i^j + (\sum_{j=1}^k \alpha_n^j - \sum_{j=1}^k y_n^j) K_{n-i} = \alpha_i + (\alpha_n - y_n) K_{n-i}$. By the inductive hypothesis we can find $\bar{y}_1, ..., \bar{y}_{n-1}$and $\bar{s}_1, ..., \bar{s}_{n-1}$ such that for $i \in \{1, ..., n-1\}$ $\sum_{j=1}^k y_i^j = y_i$ and $\sum_{j=1}^k s_i^j = s_i$ and,

$$\sum_{i=1}^{n-1} y_i^j A_i^{n-1} + \sum_{i=1}^{n-1} s_i^j I_i^{n-1}$$
$$= \sum_{i=1}^{n-m-1} \alpha_i^j A_i^{n-1} + \sum_{i=n-m}^{n-1} [\alpha_i^j + (\alpha_n^j - y_n^j) K_{n-i}] A_i^{n-1}$$

$$\sum_{i=1}^{n-1} y_i^j A_i^n + \sum_{i=1}^{n-1} s_i^j I_i^n$$
$$= \sum_{i=1}^{n-m-1} \alpha_i^j A_i^n + \sum_{i=n-m}^{n-1} [\alpha_i^j + (\alpha_n^j - y_n^j) K_{n-i}] A_i^n$$
$$= \sum_{i=1}^{n-1} \alpha_i^j A_i^n + \sum_{i=n-m}^{n-1} (\alpha_n^j - y_n^j) K_{n-i} A_i^n$$
$$= \sum_{i=1}^{n-1} \alpha_i^j A_i^n + \sum_{i=1}^{m} [(\alpha_n^j - y_n^j) K_i] A_{n-i}^n$$

Hence,

$$\sum_{i=1}^{n} y_i^j A_i^n + \sum_{i=1}^{n} s_i^j I_i^n$$
$$= \sum_{i=1}^{n-1} \alpha_i^j A_i^n + \sum_{i=1}^{m} [(\alpha_n^j - y_n^j) K_i] A_{n-i}^n + y_n^j A_n^n + s_n^j I_n^n$$
$$= \sum_{i=1}^{n-1} \alpha_i^j A_i^n + (\alpha_n^j - y_n^j) \sum_{i=1}^{m} K_i A_{n-i}^n + y_n^j A_n^n + s_n^j I_n^n$$
$$= \sum_{i=1}^{n-1} \alpha_i^j A_i^n + s_n^j (A_n^n - I_n^n) + y_n^j A_n^n + s_n^j I_n^n$$
$$= \sum_{i=1}^{n-1} \alpha_i^j A_i^n + s_n^j A_n^n + y_n^j A_n^n$$
$$= \sum_{i=1}^{n-1} \alpha_i^j A_i^n + \alpha_n^j A_n^n = \sum_{i=1}^{n} \alpha_i^j A_i^n \quad \blacksquare$$

Theorem 17 *We are given $K \in \{0,1\}^m$, $\alpha \in \{0, R^+\}^n$ and $c \in R^n$. Let y_{opt} maximize cy subject to $y \in \Gamma(K, \alpha)$ and let C be the maximum of cy subject to $y \in \Gamma(K, \alpha)$. Let y_{opt}^i maximize cy subject to $y \in \Gamma(K, I_i^n)$ and let C^i be the maximum of cy subject to $y \in \Gamma(K, I_i^n)$. Equivalently, let C^i be the maximum of $[c_1, ..., c_i]y$ subject to $y \in \Gamma(K, I_i^i)$. We claim that $C = \sum_{i=1}^n \alpha_i C^i$.*

Proof. Let y_{opt} maximize cy subject to $y \in \Gamma(K, \alpha)$ and let s_{opt} be the slack variables corresponding to y_{opt}. For all $i \in \{1, ..., n\}$ let x_{opt}^i maximize cx subject to $x \in \Gamma(K, \alpha_i I_i^n)$ and let r_{opt}^i be the slack variables corresponding to x_{opt}^i. We know

that $\alpha = \sum_{i=1}^{n} \alpha_i I_i^n$. Hence, by Lemma 16 we can find $x^i \in \Gamma(K, \alpha_i I_i^n)$ such that $\sum_{i=1}^{n} x^i = y_{opt}$ and $\sum_{i=1}^{n} r^i = s_{opt}$. We will next show that $cx^i = cx_{opt}^i$. Let $C = cy_{opt}$. Since $\sum_{i=1}^{n} x^i = y$, $C = c\sum_{i=1}^{n} x^i = \sum_{i=1}^{n} cx^i$. Assume that there exists an i such that $cx_{opt}^i > cx^i$. Let $\bar{y} = (x_{opt}^i - x^i) + \sum_{j=1}^{n} x^j$ and let $\bar{s} = (r_{opt}^i - r^i) + \sum_{j=1}^{n} r^j$. Next we show that $\bar{y} \in \Gamma(K, \alpha)$. First note that both $\bar{y}, \bar{s} \in \{0, R^+\}^n$. Also,

$$A^n\bar{y} + I^n\bar{s} = A^n[(x_{opt}^i - x^i) + \sum_{j=1}^{n} x^j] + I^n[(r_{opt}^i - r^i) + \sum_{j=1}^{n} r^j]$$
$$= (A^n x_{opt}^i + I^n r_{opt}^i) - (A^n x^i + I^n r^i) + (A^n y_{opt} + I^n s_{opt})$$

Since $x_{opt}^i \in \Gamma(K, \alpha_i I_i^n)$, $A^n x_{opt}^i + I^n r_{opt}^i = A^n \alpha_i I_i^n$. Similarly, $A^n x^i + I^n r^i = A^n \alpha_i I_i^n$ and $A^n y_{opt} + I^n s_{opt} = A^n \alpha$. Hence,

$$A^n\bar{y} + I^n\bar{s} = A^n\alpha_i I_i^n - A^n\alpha_i I_i^n + A^n\alpha = A^n\alpha$$

Thus $\bar{y} \in \Gamma(K, \alpha)$. Now note that $c\bar{y} = (cx_{opt}^i - cx^i) + \sum_{j=1}^{n} cx^j = (cx_{opt}^i - cx^i) + C > C$. Hence, y_{opt} does not maximize cy subject to $y \in \Gamma(K, \alpha)$. This is a contradiction. Hence, for all $i \in \{1, ..., n\}$ $cx^i = cx_{opt}^i$. Thus, $C = cy_{opt} = \sum_{i=1}^{n} cx^i = \sum_{i=1}^{n} cx_{opt}^i$. It is easy to show that $cx_{opt}^i = \alpha_i cy_{opt}^i$ where y_{opt}^i maximizes cy subject to $y \in \Gamma(K, I_i^n)$. Hence, $C = \sum_{i=1}^{n} cx_{opt}^i = \sum_{i=1}^{n} \alpha_i cy_{opt}^i = \sum_{i=1}^{n} \alpha_i C^i$, where $C^i = cy_{opt}^i$. ■

4.5 Algorithm for Maximizing Linear Functions Over a Generalized Fibonacci Polyhedra

In this subsection we will state an algorithm for maximizing an arbitrary linear function over a Generalized Fibonacci Polyhedra. The algorithm we present is linear in the dimension of the polyhedra and is a consequence of the 0-1 Principle and the Decomposition Principle. Before we state the algorithm we will state the problem formally.

Problem 18 *We are given $K \in \{0,1\}^m$, $\alpha \in \{0, R^+\}^n$ and $c \in R^n$. Find $C = max$ cy subject to $y \in \Gamma(K, \alpha)$.*

Algorithm 19 *For $i \in \{1, ..., n\}$ we compute C^i using the following recurrence,*

$$C^i = \begin{cases} \max(\sum_{j=1}^{m} K_j C^{i-j}, c_i) & m < i \leq n \\ \max(\sum_{j=1}^{i-1} K_j C^{i-j}, c_i) & 2 \leq i \leq m \\ \max(0, c_1) & i = 1 \end{cases}$$

Note that this can be accomplished in $O(nm)$ operations by first computing C^1, followed by C^2, followed by C^3, and so on and so forth. Now we let $C = \sum_{i=1}^{n} \alpha_i C^i$. This computation requires only $O(n)$ operations.

Proof. If we can show that for $i \in \{1, ..., n\}$ C^i is the maximum of cy subject to $y \in \Gamma(K, I_i^i)$ then from the Decomposition Principle it will follow that $C = \sum_{i=1}^{n} \alpha_i C^i$. To complete the proof it would suffice to establish that this is the case. Let y^i maximize cy subject to $y \in \Gamma(K, I_i^i)$ and let s^i be the corresponding vector of slack variables. Since y^i maximizes cy subject to $y \in \Gamma(K, I_i^i)$, y^i must be an extreme point of $\Gamma(K, I_i^i)$. There are three cases to consider.

Case 1 : $i = 1$. From the 0-1 Principle we know that $(y_1^1, s_1^1) = (1, 0)$ or $(y_1^1, s_1^1) = (0, 1)$. If $(y_1^1, s_1^1) = (1, 0)$ then cy^1 is c_1. If $(y_1^1, s_1^1) = (0, 1)$ then cy^1 is 0. Hence $C^1 = \max(0, c_1)$.

Case 2 : $2 \leq i \leq m$. From the 0-1 Principle, it follows that $(y_i^i, s_i^i) = (0, 1)$ or $(y_i^i, s_i^i) = (1, 0)$. If $(y_i^i, s_i^i) = (1, 0)$, it is easy to show that for $j \in \{1, ..., i-1\}$, $y_j^i = 0$ and $s_j^i = 0$. Hence, $cy^i = c_i$. Now consider the case when $(y_i^i, s_i^i) = (0, 1)$. We know that,

$$\sum_{j=1}^{i} y_j^i A_j^i + \sum_{j=1}^{i} s_j^i I_j^i = I_i^i + \sum_{j=1}^{i-1} y_j^i A_j^i + \sum_{j=1}^{i-1} s_j^i I_j^i = A_i^i$$

Hence,

$$\sum_{j=1}^{i-1} y_j^i A_j^i + \sum_{j=1}^{i-1} s_j^i I_j^i = A_i^i - I_i^i = I_i^i + \sum_{j=1}^{i-1} K_j A_{i-j}^i - I_i^i$$

Hence,

$$\sum_{j=1}^{i-1} y_j^i A_j^i + \sum_{j=1}^{i-1} s_j^i I_j^i = \sum_{j=1}^{i-1} K_j A_{i-j}^i$$

We let $\bar{c} = (c_1, ..., c_{i-1})$ and $\bar{y} = (y_1^i, ..., y_{i-1}^i)$. Also we define $\bar{\alpha} = (K_{i-1}, ..., K_1)$ and $\bar{C}$ to be the maximum of $\bar{c}\bar{y}$ subject to $\bar{y} \in \Gamma(K, \bar{\alpha})$. It is easy to show that the maximum of cy subject to $y \in \Gamma(K, I_i^i)$ and $(y_i, s_i) = (0, 1)$ is just $\bar{C}$. From the Decomposition Principle we know that $\bar{C} = \sum_{j=1}^{i-1} \bar{\alpha}_j C^j = \sum_{j=1}^{i-1} K_{i-j} C^j = \sum_{j=1}^{i-1} K_j C^{i-j}$. We have shown that the maximum of cy subject to $y \in \Gamma(K, I_i^i)$ and $(y_i, s_i) = (1, 0)$ is c_i and the maximum of cy subject to $y \in \Gamma(K, I_i^i)$ and $(y_i, s_i) = (0, 1)$ is $\sum_{j=1}^{i-1} K_j C^{i-j}$. Hence $C^i = \max(\sum_{j=1}^{i-1} K_j C^{i-j}, c_i)$.

Case 3 : $i > m$. This case is similar to Case 2. We will simply state that in this case $C^i = \max(\sum_{j=1}^{m} K_j C^{i-j}, c_i)$ and will leave the proof as an exercise for the reader.

■

4.6 Algorithm for Maximizing a Linear Function Over the Intersection of a Generalized Fibonacci Polyhedra and a Hyperplane

In this subsection we will state an algorithm for minimizing an arbitrary linear function over the intersection of a Generalized Fibonacci Polyhedra and a hyperplane. The algorithm we present is polynomial in the dimension of the polyhedra and is a

consequence of the 0-1 Principle and the Decomposition Principle. Before we state the algorithm we will state the problem formally.

Problem 20 *We are given* $K \in \{0,1\}^m$, $\alpha \in \{0, R^+\}^n$, $c \in R^n$ *and* $p \in R^n$ *and* $k \in R$. *Find* $C = \min cy$ *subject to* $py = k$ *and* $y \in \Gamma(K, \alpha)$. *Equivalently, find* $C = \min cy$ *subject to* $py = k$, $A^n y \leq A^n \alpha$ *and* $y \in \{0, R^+\}^n$ *where* A^n *is an* $n \times n$ *matrix such that* $y \in \Gamma(K, \alpha)$ *iff* $A^n y \leq A^n \alpha$ *and* $y \in \{0, R^+\}^n$.

Let π be the dual variable associated with the constraint $py = k$ and let μ be the vector of dual variables associated with the constraints $A^n y \leq A^n \alpha$. The dual of the linear program above is

$$\begin{array}{ll} \max & \pi k + \mu A^n \alpha \\ & \pi p + \mu A^n \leq c \\ \text{subject to} & \pi \text{ unrestricted}, \\ & \mu \in \{0, R^-\}^n \end{array}$$

We next decompose the above problem as follows

$$\max_{\pi \text{ unrestricted}} \left\{ \pi k + \begin{array}{ll} \max & \mu A^n \alpha \\ \text{subject to} & \mu A^n \leq c - \pi p \\ & \mu \in \{0, R^-\}^n \end{array} \right\}$$

Let $\bar{y}$ be the set of dual variables associated with the constraint $\mu A^n \leq c - \pi p$. We replace the inner linear program by its dual (instead of minimizing, we maximize the negative of the cost function)

$$\max_{\pi \text{ unrestricted}} \left\{ \pi k - \begin{array}{ll} \max & (\pi p - c)\bar{y} \\ \text{subject to} & A^n \bar{y} \leq A^n \alpha \\ & \bar{y} \in \{0, R^+\}^n \end{array} \right\}$$

Next we define $f(\pi) = \pi k$, $g(\pi) = \max\ (\pi p - c)\bar{y}$ subject to $A^n \bar{y} \leq A^n \alpha$, $\bar{y} \in \{0, R^+\}^n$ and $L(\pi) = f(\pi) - g(\pi)$.Using the results in the previous section, we know that for any given value of π, $g(\pi) = \sum_{i=1}^{n} \alpha_i g_i(\pi)$ where

$$g_i(\pi) = \begin{cases} \max(\sum_{j=1}^{m} K_j g_{i-j}(\pi), \pi p_i - c_i) & m < i \leq n \\ \max(\sum_{j=1}^{i-1} K_j g_{i-j}(\pi), \pi p_i - c_i) & 2 \leq i \leq m \\ \max(0, \pi p_1 - c_1) & i = 1 \end{cases}$$

We know that $g(\pi) = L(\pi) - f(\pi)$ where $f(\pi) = \pi k$. f is thus linear and continuous and L is the Lagrangian Dual of the linear program and is thus piecewise linear, concave and continuous. Therefore, g must be piecewise linear and convex. Using a similar argument it is possible to show that g_i will be piecewise linear, convex and continuous. Piecewise linear and continuous functions can be represented by a sequence of breakpoints (stored in sorted order) which partition the real axis into nonoverlapping intervals. Within each interval the piecewise linear function is linear and this line can be represented using two parameters describing the slope and the intercept. Addition of two piecewise linear functions results in a piecewise linear function whose parametric representation can be computed in linear time (linear in number of breakpoints). Similarly the maximum of a linear function and a piecewise linear function is a piecewise linear function whose parametric representation can be computed in linear time (linear in the number of breakpoints). Our proposal is to compute the parametric representation of g_1, g_2, ... , g_n and then compute the parametric representation of g and finally L. Given the parametric representation of L, it is easy to determine the value of π that maximizes L and thus the maximum value of L. For this process to be efficient we need to bound the number of breakpoints of g_1, g_2, ... , g_n and g and L. We do this next.

Let $b_i = \{\pi | \pi$ is a breakpoint in $g_i\}$ and let $B_i = \{\pi | \pi$ is a breakpoint in g_1 or g_2 or .... or $g_i\}$. Note that $b_i \subseteq B_i$ and hence $|b_i| \leq |B_i|$. We claim that $|B_i| \leq 2i - 1$. The proof is by induction. $|B_1|$ can be at most one since g_1 can have at most one breakpoint. Next we assume that $|B_{i-1}|$ can be at most $2(i-1) - 1 = 2i - 3$. First consider the case when $2 \leq i \leq m$. $\sum_{j=1}^{i-1} K_j g_{i-j}$ will be a piecewise linear and convex. Furthermore, if π is a breakpoint of $\sum_{j=1}^{i-1} K_j g_{i-j}$ then π is a breakpoint of g_1 or π is a breakpoint of g_2 or π is a breakpoint of g_3 or .... or π is a breakpoint of g_{i-1}. Hence the number of breakpoints of $\sum_{j=1}^{i-1} K_j g_{i-j}$ is less than or equal to $|B_{i-1}|$. Note that

$\sum_{j=1}^{i-1} K_j g_{i-j}$ is a piecewise linear, convex and continuous function and $\pi p_i - c_i$ is a straight line. A straight line can "intersect" a piecewise linear, convex and continuous function at at most two points (if the line overlaps a line segment, then the number of breakpoints is unaltered). Thus, $|B_i| \leq |B_{i-1}| + 2 = 2i - 3 + 2 = 2i - 1$. The proof for the case when $i > m$ is similar to the previous case and left as an exercise for the reader. Now note that $g(\pi) = \sum_{i=1}^{n} \alpha_i g_i(\pi)$. π is a breakpoint of g implies π is a breakpoint of g_1 or π is a breakpoint of g_2 or π is a breakpoint of g_3 or ... or π is a breakpoint of g_n. Hence, the number of breakpoints of g is less than or equal to $|B_n|$ which is equal to $2n - 1$. Since g has at most $2n - 1$ breakpoints and f is linear, L has at most $2n - 1$ breakpoints.

4.7 Applications and Examples

We will illustrate how the techniques developed in the previous section can be used to determine if the linear programming relaxation of the integer program has a solution, and find the optimal solution if one exists.

Example 21 *We are given $S = \{1, 2, 3\}$, $T_E = 7$, $T_D = 8$ and $R = 6/7 = .857$. We would like to determine if the linear programming relaxation of the integer program has a solution, and find an optimal solution if one exists. The linear programming relaxation of the integer program is given below.*

$$\min \sum_{i=1}^{T_D} \sum_{j=1}^{T_E} x_j^i \text{ , subject to}$$

$$\sum_{i=1}^{T_E} 2^{(T_E - i)} x_i = 2^{T_E}$$

$$1 \leq L \leq T_D, x^L + \sum_{l=1}^{L-1} N_K(L-l) x^l \leq N_K(L)$$

$$j/i < R \Longrightarrow x_j^i = 0,\ x_j^i \geq 0$$

Notice that without loss of generality we can assume that the only non zero variables are x_1^1, x_2^2, x_3^3, x_4^4, x_5^5, x_6^6, x_6^7 *and* x_7^8. *Consider an example – we claim that we can assume* $x_3^2 = 0$. *Assume* $x_3^2 > 0$. *We increase* x_2^2 *by* $x_3^2/2$ *and set* $x_3^2 = 0$. *Notice that* x^2 *decreased and thus the decoder inequalities will still be satisfied. However,* x_3 *decreased by* x_3^2 *and* x_2 *increased by* $x_3^2/2$. *Since the coefficient associated with* x_3 *is* 16 *and the coefficient associated with* x_2 *is* 32 *the encoder equality will still be satisfied. Also the value of the cost function has decreased. Thus, in an optimal solution* $x_3^2 = 0$. *The linear program above can be reduced to the linear program below.*

$$\begin{aligned} &\min && cx \\ &\textit{subject to} && px = k,\ A^n x \leq A^n \alpha,\ x \in \{0, R^+\}^n \end{aligned}$$

where $c = [1, 1, 1, 1, 1, 1, 1, 1]$, $x = [x_7^8, x_6^7, x_6^6, x_5^5, x_4^4, x_3^3, x_2^2, x_1^1]^T$, $p = [1, 2, 2, 4, 8, 16, 32, 64]$, $k = 128$, $\alpha = [0, 0, 0, 0, 0, 1, 1, 1]$ *and*

$$A^n = \begin{bmatrix} 1 & 1 & 2 & 4 & 7 & 13 & 24 & 44 \\ 0 & 1 & 1 & 2 & 4 & 7 & 13 & 24 \\ 0 & 0 & 1 & 1 & 2 & 4 & 7 & 13 \\ 0 & 0 & 0 & 1 & 1 & 2 & 4 & 7 \\ 0 & 0 & 0 & 0 & 1 & 1 & 2 & 4 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 2 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

A sufficient and necessary condition for the linear programming relaxation of the integer program to have a solution is that the maximum attained by the linear program below be greater than or equal to k.

$$\begin{aligned} &\max && px \\ &\textit{subject to} && A^n x \leq A^n \alpha,\ x \in \{0, R^+\}^n \end{aligned}$$

This is a linear programming problem over a Generalized Fibonacci Polyhedra and can be solved efficiently using the algorithm developed in this chapter. We solve the problem below.

$$
\begin{aligned}
C^1 &= \max(0, c_1) = \max(0, 1) = 1 \\
C^2 &= \max(\textstyle\sum_{j=1}^{1} K_j C_{2-j}, c_2) = \max(1, 2) = 2 \\
C^3 &= \max(\textstyle\sum_{j=1}^{2} K_j C_{3-j}, c_3) = \max(1 + 2, 2) = 3 \\
C^4 &= \max(\textstyle\sum_{j=1}^{3} K_j C_{4-j}, c_4) = \max(1 + 2 + 3, 4) = 6 \\
C^5 &= \max(\textstyle\sum_{j=1}^{3} K_j C_{5-j}, c_5) = \max(2 + 3 + 6, 8) = 11 \\
C^6 &= \max(\textstyle\sum_{j=1}^{3} K_j C_{6-j}, c_6) = \max(3 + 6 + 11, 16) = 20 \\
C^7 &= \max(\textstyle\sum_{j=1}^{3} K_j C_{7-j}, c_7) = \max(6 + 11 + 20, 32) = 37 \\
C^8 &= \max(\textstyle\sum_{j=1}^{3} K_j C_{8-j}, c_8) \\
&= \max(11 + 20 + 37, 64) = 68
\end{aligned}
$$

$$C = C^6 + C^7 + C^8 = 20 + 37 + 68 = 125$$

Note that the maximum that is attained is just 125 *which falls short of the desired* 128*. This means that the linear programming relaxation of the integer program does not have a solution.*

Example 22 *We are given* $S = \{1, 2, 3\}$*,* $T_E = 5$*,* $T_D = 6$ *and* $R = 4/5 = 0.80$*. We would like to determine if the linear programming relaxation of the integer program has a solution, and find an optimal solution if one exists. The linear programming relaxation of the integer program is given below.*

$$\min \quad \sum_{i=1}^{T_D} \sum_{j=1}^{T_E} x_j^i \ , \ subject\ to$$
$$\sum_{i=1}^{T_E} 2^{(T_E-i)} x_i = 2^{T_E}$$
$$1 \leq L \leq T_D, x^L + \sum_{l=1}^{L-1} N_K(L-l)x^l \leq N_K(L)$$
$$j/i < R \Longrightarrow x_j^i = 0,\ x_j^i \geq 0$$

Notice that without loss of generality we can assume that the only non zero variables are x_1^1, x_2^2, x_3^3, x_4^4, x_4^5 *and* x_5^6. *Consider an example – we claim that we can assume* $x_3^2 = 0$. *Assume* $x_3^2 > 0$. *We increase* x_2^2 *by* $x_3^2/2$ *and set* $x_3^2 = 0$. *Notice that* x^2 *decreased and thus the decoder inequalities will still be satisfied. However,* x_3 *decreased by* x_3^2 *and* x_2 *increased by* $x_3^2/2$. *Since the coefficient associated with* x_3 *is* 4 *and the coefficient associated with* x_2 *is* 8, *the encoder equality will still be satisfied. Also the value of the cost function has decreased. Thus, in an optimal solution* $x_3^2 = 0$. *The linear program above can be reduced to the linear program below.*

$$\begin{array}{ll} \min & cx \\ subject\ to & px = k,\ A^n x \leq A^n \alpha,\ x \in \{0, R^+\}^n \end{array}$$

where $c = [1,1,1,1,1,1]$, $x = [x_5^6, x_4^5, x_4^4, x_3^3, x_2^2, x_1^1]^T$, $p = [1,2,2,4,8,16]$, $k = 32$, $\alpha = [0,0,0,1,1,1]$ *and*

$$A^n = \begin{bmatrix} 1 & 1 & 2 & 4 & 7 & 13 \\ 0 & 1 & 1 & 2 & 4 & 7 \\ 0 & 0 & 1 & 1 & 2 & 4 \\ 0 & 0 & 0 & 1 & 1 & 2 \\ 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

A sufficient and necessary condition for the linear programming relaxation of the

integer program to have a solution is that the maximum attained by the linear program below be greater than or equal to k.

$$\begin{aligned} &\max && px \\ &\textit{subject to} && A^n x \leq A^n \alpha,\ x \in \{0, R^+\}^n \end{aligned}$$

This is a linear programming problem over a Generalized Fibonacci Polyhedra and can be solved efficiently using the algorithm developed in this chapter. We solve the problem below.

$$\begin{aligned} C^1 &= \max(0, p_1) = \max(0, 1) = 1 \\ C^2 &= \max(\textstyle\sum_{j=1}^{1} K_j C_{2-j}, p_2) = \max(1, 2) = 2 \\ C^3 &= \max(\textstyle\sum_{j=1}^{2} K_j C_{3-j}, p_3) = \max(1 + 2, 2) = 3 \\ C^4 &= \max(\textstyle\sum_{j=1}^{3} K_j C_{4-j}, p_4) = \max(1 + 2 + 3, 4) = 6 \\ C^5 &= \max(\textstyle\sum_{j=1}^{3} K_j C_{5-j}, p_5) = \max(2 + 3 + 6, 8) = 11 \\ C^6 &= \max(\textstyle\sum_{j=1}^{3} K_j C_{6-j}, p_6) = \max(3 + 6 + 11, 16) = 20 \end{aligned}$$

$$C = C^4 + C^5 + C^6 = 6 + 11 + 20 = 37$$

Note that the maximum that is attained is 37 *which exceeds the desired* 32*. This means that the linear programming relaxation of the integer program has a solution. We next illustrate how to compute the optimal solution to the linear programming relaxation of the integer program. The linear programming relaxation of the integer program is given below.*

$$\begin{aligned} &\min && cx \\ &\textit{subject to} && px = k,\ A^n x \leq A^n \alpha,\ x \in \{0, R^+\}^n \end{aligned}$$

This is equivalent to maximizing $L(\pi) = f(\pi) - g(\pi)$ *where* π *is unrestricted,*

$f(\pi) = \pi k$ *and*

$$g(\pi) = \left\{ \begin{array}{ll} \max & (\pi p - c)\bar{x} \\ \textit{subject to} & A^n \bar{x} \leq A^n \alpha,\ \bar{x} \in \{0, R^+\}^n \end{array} \right\}$$

Next, we compute $g_1, g_2, ..., g_5, g_6$.

$$g_1(\pi) = \max \left\{ \begin{array}{c} 0 \\ p_1\pi - c_1 \end{array} \right\} = \left\{ \begin{array}{ll} 0 & \pi \leq 1 \\ \pi - 1 & \pi > 1 \end{array} \right.$$

$$g_2(\pi) = \max \left\{ \begin{array}{c} \sum_{j=1}^{1} K_j g_{2-j}(\pi) \\ p_2\pi - c_2 \end{array} \right\} = \left\{ \begin{array}{ll} 0 & \pi \leq \frac{1}{2} \\ 2\pi - 1 & \pi > \frac{1}{2} \end{array} \right.$$

$$g_3(\pi) = \max \left\{ \begin{array}{c} \sum_{j=1}^{2} K_j g_{3-j}(\pi) \\ p_3\pi - c_3 \end{array} \right\} = \left\{ \begin{array}{ll} 0 & \pi \leq \frac{1}{2} \\ 2\pi - 1 & \frac{1}{2} < \pi \leq 1 \\ 3\pi - 2 & 1 < \pi \end{array} \right.$$

$$g_4(\pi) = \max \left\{ \begin{array}{c} \sum_{j=1}^{3} K_j g_{4-j}(\pi) \\ p_4\pi - c_4 \end{array} \right\} = \left\{ \begin{array}{ll} 0 & \pi \leq \frac{1}{4} \\ 4\pi - 1 & \frac{1}{4} < \pi \leq \frac{3}{2} \\ 6\pi - 4 & \frac{3}{2} < \pi \end{array} \right.$$

$$g_5(\pi) = \max \left\{ \begin{array}{c} \sum_{j=1}^{3} K_j g_{5-j}(\pi) \\ p_5\pi - c_5 \end{array} \right\} = \left\{ \begin{array}{ll} 0 & \pi \leq \frac{1}{8} \\ 8\pi - 1 & \frac{1}{8} < \pi \leq 2 \\ 11\pi - 7 & 2 < \pi \end{array} \right.$$

$$g_6(\pi) = \max \left\{ \begin{array}{c} \sum_{j=1}^{3} K_j g_{6-j}(\pi) \\ p_6\pi - c_6 \end{array} \right\} = \left\{ \begin{array}{ll} 0 & \pi \leq \frac{1}{16} \\ 16\pi - 1 & \frac{1}{16} < \pi \leq 3 \\ 20\pi - 13 & 3 < \pi \end{array} \right.$$

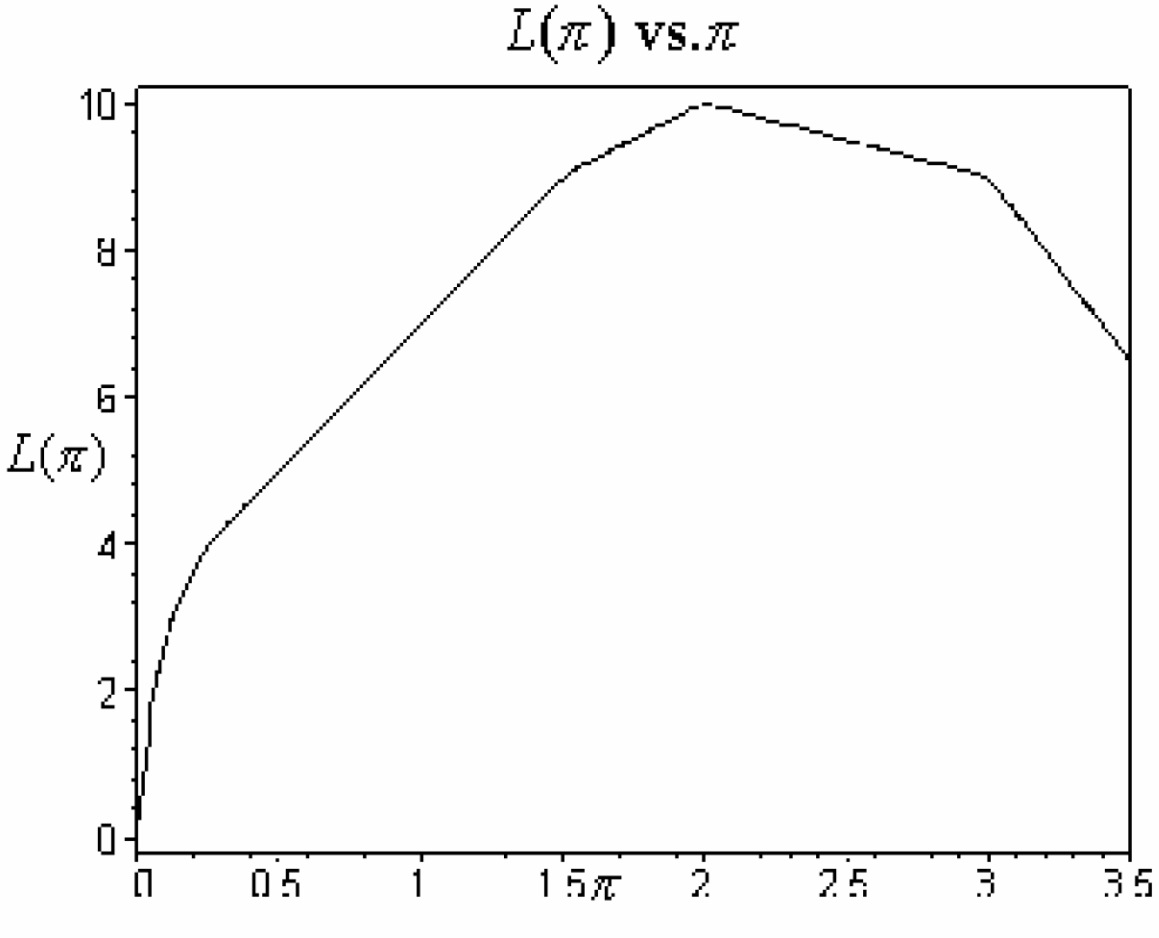
Hence,

$$g(\pi) = g_4(\pi) + g_5(\pi) + g_6(\pi) = \begin{cases} 0 & \pi \leq \frac{1}{16} \\ 16\pi - 1 & \frac{1}{16} < \pi \leq \frac{1}{8} \\ 24\pi - 2 & \frac{1}{8} < \pi \leq \frac{1}{4} \\ 28\pi - 3 & \frac{1}{4} < \pi \leq \frac{3}{2} \\ 30\pi - 6 & \frac{3}{2} < \pi \leq 2 \\ 33\pi - 12 & 2 < \pi \leq 3 \\ 37\pi - 24 & \pi > 3 \end{cases}$$

Hence,

$$L(\pi) = f(\pi) - g(\pi) = \begin{cases} 32\pi & \pi \leq \frac{1}{16} \\ 16\pi + 1 & \frac{1}{16} < \pi \leq \frac{1}{8} \\ 8\pi + 2 & \frac{1}{8} < \pi \leq \frac{1}{4} \\ 4\pi + 3 & \frac{1}{4} < \pi \leq \frac{3}{2} \\ 2\pi + 6 & \frac{3}{2} < \pi \leq 2 \\ -\pi + 12 & 2 < \pi \leq 3 \\ -5\pi + 24 & \pi > 3 \end{cases}$$

The maximum value taken on by $L(\pi)$ *is* 10 *and occurs at* $\pi = 2$. $L(\pi)$ *is plotted in the figure above. We next turn our attention to the integer program associated with this code construction problem. Using a standard ILP solver we find that the solution to the integer linear program is* $x_1^1 = 1$, $x_4^5 = 6$ *and* $x_5^6 = 4$ *(all other variables are* 0*). Hence the size of the code will be* 11. *Thus the solution to the linear program provides a lower bound on the size of the code. Next we illustrate how to construct the code itself. First we construct a prefix-free set using the source alphabet* $\{0, 1\}$ *such that the number of elements consisting of* 1 *bit is* 1, *the number of elements consisting of* 4 *bits is* 6 *and the number of elements consisting of* 5 *bits is* 4. *A valid set is* $\{0, 1000, 1001, 1010, 1011, 1100, 1101, 11100, 11101, 11110, 11111\}$. *We will call*


$L(\pi)$ vs.π
$L(\pi)$
10
8
6
4
2
0
0
0.5
1
1.5π
2
2.5
3
3.5

this set A. Next we construct a prefix-free set using the code alphabet $\{1, 2, 3\}$ *such that the number of elements having a duration of* 1 *unit is* 1*, the number of elements having a duration of* 5 *units is* 6 *and the number of elements having a duration of* 6 *units is* 4*. A valid set is* $\{1, 2111, 212, 221, 23, 311, 32, 2112, 213, 312, 33\}$*. We will call this set B. Now we arbitrarily map* 1 *element consisting of* 1 *bit from set A to* 1 *element of duration* 1 *in set B. We arbitrarily map* 6 *elements consisting of* 4 *bits in set A to* 6 *elements of duration* 5 *in set B in a one on one manner. Similarly, we arbitrarily map* 4 *elements consisting of* 5 *bits in set A to* 4 *elements of duration* 6 *in set B in a one on one manner. A valid mapping (code) is shown in the table below.*

$0 \leftrightarrow 1$	$1001 \leftrightarrow 212$	$1011 \leftrightarrow 23$	$1101 \leftrightarrow 32$	$11101 \leftrightarrow 213$	$11111 \leftrightarrow 33$
$1000 \leftrightarrow 2111$	$1010 \leftrightarrow 221$	$1100 \leftrightarrow 311$	$11100 \leftrightarrow 2112$	$11110 \leftrightarrow 312$	

4.8 Open Problems and Future Work

The algorithm presented in this chapter exploits the structure of Generalized Fibonacci Polyhedra represented by the set of inequalities in (4.6). However, equation (4.5) is well structured as well, and so is the cost function (4.4). A possible approach to developing a more efficient algorithm would be to study the properties of the polyhedra formed by the intersection of the Generalized Fibonacci Polyhedra, equation (4.6) and the hyperplane (4.5) and develop an algorithm for optimizing an arbitrary linear function over such a polyhedra. One could then extend this algorithm to solve the linear programming relaxation by treating the set of equations in (4.7) as a complicating constraint using Lagrangian Duality. Furthermore, although we have presented an algorithm for solving the linear programming relaxation, we have not derived bounds on the quality of the approximation obtained, for the case in which the integer linear program is feasible as well. Other open problems include deriving sufficient and necessary conditions for determining the feasibility of the integer linear program itself. As well as developing an efficient, polynomial time algorithm for solving the integer linear program. One can also study more complex structures,

like finite state machines for the purposes of developing more compact and realistic coding schemes.

Chapter 5

Generalized Fibonacci Numbers and their Sums

Two series that arise in both the context of rate analysis and code construction are the Generalized Fibonacci Numbers and their sums. Let m be a positive integer greater than or equal to 2. We define K_i for $i \in \{1, 2, ..., m-1, m\}$ as follows

$$K_i = \begin{cases} 1 & i \in S \\ 0 & i \notin S \end{cases}$$

Next we define $N_K(T)$ to be the number of sequences of length T whose elements are in S.

$$N_K(T) = \begin{cases} \sum_{i=1}^{m} K_i N_K(T-i) & T > m \\ K_T + \sum_{i=1}^{T-1} K_i N_K(T-i) & 2 \leq T \leq m \\ K_1 & T = 1 \end{cases}$$

If $m = 2$ and $K_1 = 1$, $K_2 = 1$ then

$$N_K(T) = \begin{cases} \sum_{i=1}^{2} N_K(T-i) & T > 2 \\ 2 & T = 2 \\ 1 & T = 1 \end{cases} = Fib(T)$$

Thus $N_K(T)$ is a generalization of the Fibonacci Numbers. Note that if for all $i \in \{1, 2, ..., m-1, m\}$, $K_i = 1$ then $N_K(T)$ would reduce to the generalization

studied by Capocelli and Cull [17].We next define $\bar{N}_K(T)$ as follows

$$\bar{N}_K(T) = \sum_{t=1}^{T} N_K(t)$$

In this chapter we will derive a closed form expression for the Generalized Fibonacci Numbers represented by $N_K(T)$. We will also derive recursive and closed form expressions for the sum of the Generalized Fibonacci Numbers represented by $\bar{N}_K(T)$. We will also show that if for all $i \in \{1, 2, ..., m-1, m\}$, $K_i = 1$ then sum of the Generalized Fibonacci Numbers are rounded powers much like the Generalization of the Fibonacci Numbers studied by Capocelli and Cull [17]. It must be noted that the recursive series studied by Spickerman [91] and Spickerman and Joyner [92] are also special cases of this generalization.

Theorem 23 *Let* $p(\lambda) = \lambda^m - \sum_{i=1}^{m} K_i \lambda^{(m-i)}$ *and* $p'(\lambda) = m\lambda^{m-1} - \sum_{i=1}^{m} K_i(m-i)\lambda^{(m-i-1)}$. *For* $i \in \{1, 2, ..., m-1, m\}$ ϕ_i *be the roots of* $p(\lambda)$. *If the root of* $p(\lambda)$ *are distinct then*

$$N_K(T) = \sum_{i=1}^{m} \alpha_i \left[\phi_i\right]^T = \sum_{i=1}^{m} |\alpha_i| \left|\phi_i\right|^T \cos(\theta_i(T))$$

where $\alpha_i = \dfrac{\phi_i^{m-1}}{p'(\phi_i)}$ *and* $\theta_i(t) = \arg(\alpha_i) + t\arg(\phi_i)$

Proof. Let M be the companion matrix. It is easy to show that

$$M = \begin{bmatrix} K_1 & K_2 & . & K_{m-1} & K_m \\ 1 & 0 & . & 0 & 0 \\ . & . & . & . & . \\ 0 & 0 & . & 0 & 0 \\ 0 & 0 & . & 1 & 0 \end{bmatrix}$$

For $i \in \{1, 2, ..., m-1, m\}$, we define C_i and R_i as follows,

$$C_i = \begin{bmatrix} \phi_i^{m-1} & \phi_i^{m-2} & . & \phi_i^1 & 1 \end{bmatrix}^T \text{ and}$$

$$R_i = \begin{bmatrix} 1 & \phi_i^1 - \sum_{j=1}^{1} K_j \dfrac{\phi_i^1}{\phi_i^j} & . & . & \phi_i^{m-1} - \sum_{j=1}^{m-1} K_j \dfrac{\phi_i^{m-1}}{\phi_i^j} \end{bmatrix}^T$$

Since $p(\phi_i) = 0$, $\phi_i^m - \sum_{i=1}^{m-1} K_i \phi_i^{m-i} = K_m$ and $\phi_i^m = \sum_{i=1}^{m} K_i \phi_i^{m-i}$. Using these two identities, it is easy to show that $MC_i = \phi_i C_i$ and $M^T R_i = \phi_i R_i$. Thus C_i and

R_i are the column and row eigenvectors of M. Next we will establish that if $i \neq j$, then $R_i^T C_j = 0$. $(R_i^T M)C_j = R_i^T(MC_j)$. But $(R_i^T M)C_j = \phi_i R_i^T C_j$ and $R_i^T(MC_j) = \phi_j R_i^T C_j$. Hence, $\phi_i R_i^T C_j = \phi_j R_i^T C_j$. Hence, $\phi_i = \phi_j$. This is a contradiction, since the roots of the characteristic polynomial are distinct. Since, the roots of the characteristic polynomial are distinct and since $R_i C_j = 0$ if $i \neq j$, using an argument similar to the one by Capocelli and Cull [17], it is possible to show that

$$N_K(T) = \sum_{i=1}^{m} \alpha_i \left[\phi_i\right]^T \text{ where } \alpha_i = \frac{R_i^T I}{\phi_i R_i^T C_i} \text{ and}$$

$$I = \left[\begin{array}{ccccccc} N_K(m) & N_K(m-1) & . & . & . & N_K(2) & N_K(1) \end{array} \right]^T$$

To complete the derivation, we need to establish the exact value of $R_i^T I$ and $R_i^T C_i$. First let us derive an expression for $R_i^T I$.

$$R_i^T I = N_K(m) + \sum_{j=1}^{m-1} [\phi_i^j - \sum_{l=1}^{j} K_l \phi_i^{j-l}] N_K(m-j)$$

$$= N_K(m) + \sum_{j=1}^{m-1} \phi_i^j N_K(m-j) - \sum_{j=1}^{m-1} \sum_{l=1}^{j} K_l \phi_i^{j-l} N_K(m-j)$$

Collecting like powers of ϕ_i

$$= N_K(m) + \sum_{p=1}^{m-1} \phi_i^p N_K(m-p) - \sum_{p=0}^{m-2} \phi_i^p \sum_{j=1}^{m-p-1} K_j N_K(m-p-j)$$

$$= [N_K(m) - \sum_{j=1}^{m-1} K_j N_K(m-j)] + [\phi_i^{m-1} N_K(1)] + \sum_{p=1}^{m-2} \phi_i^p [N_K(m-p) - \sum_{j=1}^{m-p-1} K_j N_K(m-p-j)]$$

Now note that, $N_K(m) = K_m + \sum_{j=1}^{m-1} K_j N_K(m-j)$

Hence, $N_K(m) - \sum_{j=1}^{m-1} K_j N_K(m-j) = K_m$

Similarly, $N_K(m-p) - \sum_{j=1}^{m-p-1} K_j N_K(m-p-j) = K_{m-p}$

Also $N_K(1) = K_1$. Substituting these values into the formula for $R_i^T I$, we get

$$R_i^T I = K_m + \phi_i^{m-1} K_1 + \sum_{p=1}^{m-2} \phi_i^p K_{m-p} = \sum_{j=1}^{m} K_j \phi_i^{m-j}$$

But since $p(\phi_i) = 0$, $\phi_i^m = \sum_{j=1}^{m} K_j \phi_i^{m-j}$. Hence, $R_i^T I = \phi_i^m$

Now we will derive an expression for $R_i^T C_i$.

$$R_i^T C_i = \phi_i^{m-1} + \sum_{n=1}^{m-1} [\phi_i^n - \sum_{j=1}^{n} K_j \phi_i^{n-j}] \phi_i^{m-1-n}$$

$$= \phi_i^{m-1} + \sum_{n=1}^{m-1} \phi_i^{m-1} - \sum_{n=1}^{m-1} \sum_{j=1}^{n} K_j \phi_i^{m-1-j}$$
$$= m\phi_i^{m-1} - \sum_{j=1}^{m-1} K_j (m-j) \phi_i^{m-j-1}$$
$$= m\phi_i^{m-1} - \sum_{j=1}^{m} K_j (m-j) \phi_i^{m-j-1} = p'(\phi_i)$$

Now we are ready to derive the value of the multiplicative constant. It follows by substituting the value of $R_i^T I$ and $R_i^T C_i$ in the formula for α_i.

$$\alpha_i = \frac{R_i^T I}{\phi_i R_i^T C_i} = \frac{\phi_i^{m-1}}{p'(\phi_i)}$$

The characteristic polynomial can have complex roots. Thus we will derive the expression for $N_K(T)$ in polar form. First let us define

$$\theta_i(t) = \arg(\alpha_i) + t \arg(\phi_i)$$
$$N_K(T) = \sum_{i=1}^{m} |\alpha_i| \, |\phi_i|^T \left[\cos(\theta_i(T)) + \sqrt{-1} \sin(\theta_i(T))\right]$$

Note that the complex roots occur in conjugate pairs. It is easy to show that if ϕ_i and ϕ_j are conjugates, so are α_i and α_j. Hence, $\theta_i(T) = -\theta_j(T)$. Thus the complex sinusoid components of the i and j term in the summation will cancel. Thus,

$$N_K(T) = \sum_{i=1}^{m} |\alpha_i| \, |\phi_i|^T \cos(\theta_i(T)) \quad \blacksquare$$

Next we will derive a recursive expression for the sum of the Generalized Fibonacci Numbers.

Before we derive a closed form expression for the sum of the Generalized Fibonacci Numbers, we will prove a lemma.

Lemma 24 $\displaystyle\sum_{i=1}^{m} \frac{\alpha_i \phi_i}{\phi_i - 1} = \frac{\sum_{i=1}^{m} K_i}{[\sum_{i=1}^{m} K_i] - 1}$

Proof. First let us define a new function G as follows

$$G_K(n) = \begin{cases} \sum_{i=1}^{m} K_i G_K(n-i) & n > m \\ \sum_{i=n}^{m} K_i + \sum_{i=1}^{n-1} K_i G_K(n-i) & 2 \le n \le m \\ \sum_{i=1}^{m} K_i & n = 1 \end{cases}$$

Next we will prove $G_K(n) = \sum_{i=1}^{m} \gamma_i \phi_i^n$ where $\gamma_i = ([\sum_{j=1}^{m} K_j] - 1)\alpha_i \left(\phi_i - 1\right)^{-1}$

Since, G is a generalized Fibonacci type recurrence having the same characteristic polynomial, it can be written as,

$G_K(n) = \sum_{i=1}^{m} \gamma_i \phi_i^n$.

The only issue that needs to be addressed is the exact value of γ_i. Using an argument similar to that in Capocelli and Cull [17], it is easy to show that,

$$\gamma_i = \frac{R_i^T I}{\phi_i R_i^T C_i} = \frac{R_i^T I}{\phi_i p'(\phi_i)} \text{ where } I = \begin{bmatrix} G_K(m) & G_K(m-1) & . & G_K(2) & G_K(1) \end{bmatrix}^T$$

$$\left(\phi_i - 1\right) R_i^T I = \left(\phi_i - 1\right) \left[G_K(m) + \sum_{n=1}^{m-1} (\phi_i^n - \sum_{j=1}^{n} K_j \phi_i^{n-j}) G_K(m-n) \right]$$

By collecting like powers of ϕ_i we get,

$$= \left(\phi_i - 1\right) \sum_{p=0}^{m-1} \phi_i^p \left[G_K(m-p) - \sum_{j=1}^{m-p-1} K_j G_K(m-p-j) \right]$$

But we know that, $G_K(m-p) - \sum_{j=1}^{m-p-1} K_j G_K(m-p-j) = \sum_{j=m-p}^{m} K_j$

Substituting,

$$= \left(\phi_i - 1\right) \sum_{p=0}^{m-1} (\phi_i^p \sum_{j=m-p}^{m} K_j) = \sum_{p=0}^{m-1} (\phi_i^{p+1} \sum_{j=m-p}^{m} K_j) - \sum_{p=0}^{m-1} (\phi_i^p \sum_{j=m-p}^{m} K_j)$$

$$= \sum_{p=1}^{m} (\phi_i^p \sum_{j=m-p+1}^{m} K_j) - \sum_{p=0}^{m-1} (\phi_i^p \sum_{j=m-p}^{m} K_j)$$

$$= \phi_i^m \sum_{j=1}^{m} K_j + \sum_{p=1}^{m-1} \phi_i^p (\sum_{j=m-p+1}^{m} K_j - \sum_{j=m-p}^{m} K_j) - K_m$$

$$= \phi_i^m \sum_{j=1}^{m} K_j - \sum_{p=1}^{m-1} \phi_i^p K_{m-p} - K_m = \phi_i^m \sum_{j=1}^{m} K_j - \sum_{j=1}^{m} K_j \phi_i^{m-j} = \phi_i^m ([\sum_{j=1}^{m} K_j] - 1)$$

We have shown that, $\left(\phi_i - 1\right) R_i^T I = \phi_i^m ([\sum_{j=1}^{m} K_j] - 1)$

Hence, $R_i^T I = \frac{1}{(\phi_i - 1)} \phi_i^m ([\sum_{j=1}^{m} K_j] - 1)$

Substituting we get, $\gamma_i = \frac{R_i^T I}{\phi_i p'(\phi_i)} = ([\sum_{j=1}^{m} K_j] - 1) \frac{\phi_i^{m-1}}{\left(\phi_i - 1\right) p'(\phi_i)}$

Thus we have shown, $G_K(n) = \sum_{i=1}^{m} \gamma_i \phi_i^n$ where $\gamma_i = ([\sum_{j=1}^{m} K_j] - 1)\alpha_i \left(\phi_i - 1\right)^{-1}$

Now note that, $G_K(1) = \sum_{i=1}^{m} K_i$

Also, $G_K(1) = \sum_{i=1}^{m} \gamma_i \phi_i = ([\sum_{i=1}^{m} K_i] - 1)(\sum_{i=1}^{m} \frac{\alpha_i \phi_i}{\phi_i - 1})$

Hence, $\sum_{i=1}^{m} \frac{\alpha_i \phi_i}{\phi_i - 1} = \frac{\sum_{i=1}^{m} K_i}{[\sum_{i=1}^{m} K_i] - 1}$ ■

Next we will derive a closed form expression for the sum of the Generalized Fibonacci Numbers.

Theorem 25 *Let* $p(\lambda) = \lambda^m - \sum_{i=1}^{m} K_i \lambda^{(m-i)}$ *and* $p'(\lambda) = m\lambda^{m-1} - \sum_{i=1}^{m} K_i(m-i)\lambda^{(m-i-1)}$. *For* $i \in \{1, 2, ..., m-1, m\}$ *let* ϕ_i *be the roots of* $p(\lambda)$. *If the roots of* $p(\lambda)$ *are distinct then*

$$\bar{N}_K(T) = \kappa + \sum_{i=1}^{m} \beta_i [\phi_i]^{T+1} = \kappa + \sum_{i=1}^{m} |\beta_i| |\phi_i|^{T+1} \cos(\theta_i(T+1))$$

where $\beta_i = \frac{\phi_i^{m-1}}{(\phi_i - 1)p'(\phi_i)}$, $\theta_i(t) = \arg(\beta_i) + t \arg(\phi_i)$ *and* $\kappa = -\frac{\sum_{i=1}^{m} K_i}{[\sum_{i=1}^{m} K_i] - 1}$

Proof. $\bar{N}_K(T) = \sum_{t=1}^{T} N_K(T) = \sum_{t=1}^{T} \sum_{i=1}^{m} \alpha_i [\phi_i]^t$

$= \sum_{i=1}^{m} \alpha_i \sum_{t=1}^{T} [\phi_i]^t = \sum_{i=1}^{m} \alpha_i \frac{\phi_i^{T+1} - \phi_i}{\phi_i - 1} = -\sum_{i=1}^{m} \alpha_i \frac{\phi_i}{\phi_i - 1} + \sum_{i=1}^{m} \alpha_i \frac{\phi_i^{T+1}}{\phi_i - 1}$

But note that by Lemma 24,

$$\sum_{i=1}^{m} \alpha_i \frac{\phi_i}{\phi_i - 1} = \frac{\sum_{i=1}^{m} K_i}{[\sum_{i=1}^{m} K_i] - 1} = -\kappa$$

Hence, $\bar{N}_K(T) = \kappa + \sum_{i=1}^{m} \alpha_i \frac{\phi_i^{T+1}}{\phi_i - 1} = \kappa + \sum_{i=1}^{m} \beta_i \phi_i^{T+1}$

Now we convert this formula to polar form. First we define,

$\theta_i(t) = \arg(\beta_i) + t \arg(\phi_i)$

$\bar{N}_K(T) = \kappa + \sum_{i=1}^{m} |\beta_i| |\phi_i|^{T+1} [\cos(\theta_i(T+1)) + \sqrt{-1} \sin(\theta_i(T+1))]$

Again note that the complex roots occur in conjugate pairs. It is easy to show that if ϕ_i and ϕ_j are conjugates, so are β_i and β_j. Hence, $\theta_i(T) = -\theta_j(T)$. Thus

the complex sinusoid components of the i and j term in the summation will cancel. Thus,

$$\bar{N}_K(T) = \kappa + \sum_{i=1}^{m} |\beta_i| \, |\phi_i|^{T+1} \cos(\theta_i(T+1)) \quad \blacksquare$$

Next we define the generalization studied by Capocelli and Cull [17].

Definition 26 *We will formally define the mth order Fibonacci Numbers F and their sum $\bar{F}$ as*

$$F_m(n) = \begin{cases} \sum_{i=1}^{m} F(n-i) & n \geq m \\ 2^{n-2} & 2 \leq n \leq m-1 \\ 1 & n = 1 \\ 0 & n = 0 \end{cases} \quad \text{and } \bar{F}(n) = \sum_{i=1}^{n} F(i)$$

We will leave as an exercise for the reader to verify that when $S = \{1, 2, ..., m-1, m\}$ or equivalently $K_i = 1$ for all $i \in \{1, 2, ..., m-1, m\}$ then $F_m(T) = N_K(T-1)$ and $\bar{F}_m(T) = \bar{N}_K(T-1) + 1$. We let $p(\lambda) = \lambda^m - \sum_{i=1}^{m} \lambda^{(m-i)}$. Also let ϕ_i be the roots of $p(\lambda) = 0$ (it is easy to verify that the roots are distinct) and let ϕ_1 be the real positive root (it is easy to verify that the real positive root is unique). From Theorem 23 it follows that

$$F_m(T) = \sum_{i=1}^{m} \alpha_i \left[\phi_i\right]^{T-1} \text{ where } \alpha_i = \frac{\phi_i^{m-1}}{p'(\phi_i)}$$

Capocelli and Cull [17] further show that the mth order Fibonacci Numbers are rounded powers and can be computed from only the real positive root using the formula below.

$$F_m(T) = \left\langle \alpha_1 \phi_1^{T-1} \right\rangle$$

Next we will prove that the sum of the mth order Fibonacci Numbers, like the mth order Fibonacci Numbers are rounded powers.

Theorem 27 $\bar{F}_m(T) = \left\langle -\frac{1}{m-1} + \beta_1 \phi_1^T \right\rangle$ *where* $\beta_i = \frac{\phi_i^{m-1}}{(\phi_i - 1)p'(\phi_i)}$

Proof. From Theorem 25 it follows that $\bar{F}_m(T) = -\frac{1}{m-1} + \sum_{i=1}^{m} \beta_i [\phi_i]^T$ where $\beta_i = \frac{\phi_i^{m-1}}{(\phi_i - 1)p'(\phi_i)}$. We define $d_m(T) = \sum_{i=2}^{m} \beta_i [\phi_i]^T$. To complete the proof we need to show that for all $T \geqslant 0$, $|d_m(T)| \leqslant 1/2$. If $T \geqslant 2$ and if for all i, $2 \leqslant i \leqslant m$, $|d_m(T-i)| \leqslant 1/2$ then $|d_m(T)| < 1/2$. The proof of this statement is analogous to a proof presented by Capocelli and Cull [17] and thus omitted. Hence, it suffices to establish that for all m, if $2 - m \leqslant i \leqslant 1$, $|d_m(i)| \leqslant 1/2$.

Case 1: $m = 2$. We need to show that for i such that $0 \leqslant i \leqslant 1$, $|d_m(i)| \leqslant 1/2$. Direct computations show that $d_2(0) = -0.17$ and $d_2(1) = 0.10$.

Case 2: $m \geqslant 3$.We need to show that for i such that $2-m \leqslant i \leqslant 1$, $|d_m(i)| \leqslant 1/2$. We will first prove that this is true for $2-m \leqslant i \leqslant 0$ and then show that it also holds for $i = 1$.

(1) Consider that case when $2 - m \leqslant i \leqslant 0$. We need to show that $|d_m(i)| \leqslant 1/2$. Or equivalently, $-1/2 \leqslant d_m(i) \leqslant 1/2$. We know that $d_m(i) = \bar{F}_m(i) + 1/(m-1) - \beta_1 [\phi_1]^i$. For $i \leqslant 0$, $\bar{F}_m(i) = 0$. Thus, $d_m(i) = 1/(m-1) - \beta_1 [\phi_1]^i$. Hence, we need to prove, $-1/2 \leqslant 1/(m-1) - \beta_1 [\phi_1]^i \leqslant 1/2$. Or equivalently, $-1/2 - 1/(m-1) \leqslant -\beta_1 [\phi_1]^i \leqslant 1/2 - 1/(m-1)$. Now note that $-\beta_1 < -\beta_1 [\phi_1]^1 < .... < -\beta_1 [\phi_1]^{2-m}$. Hence, we need to show that $-1/2 - 1/(m-1) \leqslant -\beta_1$ and $-\beta_1 [\phi_1]^{2-m} \leqslant 1/2 - 1/(m-1)$.

(a) $-1/2 - 1/(m-1) \leqslant -\beta_1$ or $\beta_1 \leqslant 1/2 + 1/(m-1)$. Or equivalently, $((m+1)\phi_1 - 2m)^{-1} \leqslant (m+1)/(2(m-1))$. Capocelli and Cull [17] prove that $2 - 2/2^m < \phi_1$. Hence, $((m+1)\phi_1 - 2m)^{-1} < ((m+1)(2 - 2/2^m) - 2m)^{-1}$. Hence it would suffice to establish that $((m+1)(2 - 2/2^m) - 2m)^{-1} \leqslant (m+1)/(2(m-1))$. Simplifying the above inequality yields $(m+1)^2 \leqslant 2^{m+1}$. This is true for $m \geqslant 3$.

(b) $-\beta_1 [\phi_1]^{2-m} \leqslant 1/2 - 1/(m-1)$. Note that $-\beta_1 [\phi_1]^{2-m} < 0$ and since $m \geqslant 3$, $1/2 - 1/(m-1) \geqslant 0$.

(2) Consider the case when $i = 1$. We need to show that $-1/2 \leqslant d_m(1) \leqslant 1/2$ or $-1/2 \leqslant 1 - \beta_1 [\phi_1] + 1/(m-1) \leqslant 1/2$. Substituting the value of β_1 and simplifying reduces this inequality to, $2m(3m-1)/(3m^2+1) \leqslant \phi_1 \leqslant 2m(m+1)/(3+m^2)$.

(a) $2m(3m-1)/(3m^2+1) \leqslant \phi_1$. Capocelli and Cull [17] show that $2-2/2^m < \phi_1$. Hence, it would suffice to show that $2m(3m-1)/(3m^2+1) \leqslant 2-2/2^m$. Simplifying this inequality yields, $(3m^2+1)/(m+1) \leqslant 2^m$. This inequality holds for $m \geqslant 3$.

(b) $\phi_1 \leqslant 2m(m+1)/(3+m^2)$. Note that $\phi_1 < 2$. But since $m \geqslant 3$, $2m(m+1)/(3+m^2) \geqslant 2$. ■

Chapter 6

A 20Gbs Integrated Optical Transceiver in IBM BiCMOS 7HP Process Technology

In this chapter we will discuss our attempts to build a 20 Gbs Optical Transceiver based on the ideas outlined in the previous chapters. The contribution is a fundamentally new architecture for both the transmitter and receiver. Both these circuits are fully asynchronous. The transmitter is based on a reconfigurable ring oscillator and the receiver is based on picosecond time digitizers which directly measure the time between consecutive voltage transitions, unlike conventional receivers which sample the amplitude of the data signal. Thus there is no need for clock recovery. In the next section we will review the architecture of the transmitter and receiver.

6.1 Transmitter and Receiver Architecture

The transmitter consists of a single inversion ring oscillator formed by four reconfigurable delay lines. This is illustrated in Figure 6.1. A reconfigurable delay line is a tapped delay line connected to a multiplexer which allows selection of the appropriate tap output thereby effectively altering the delay between the input and output of the reconfigurable delay line. It must be noted that the outputs of the delay lines are shifted versions of each other as illustrated in Figure 6.2. The final output can be generated by simply computing the parity of the four outputs as illustrated. The

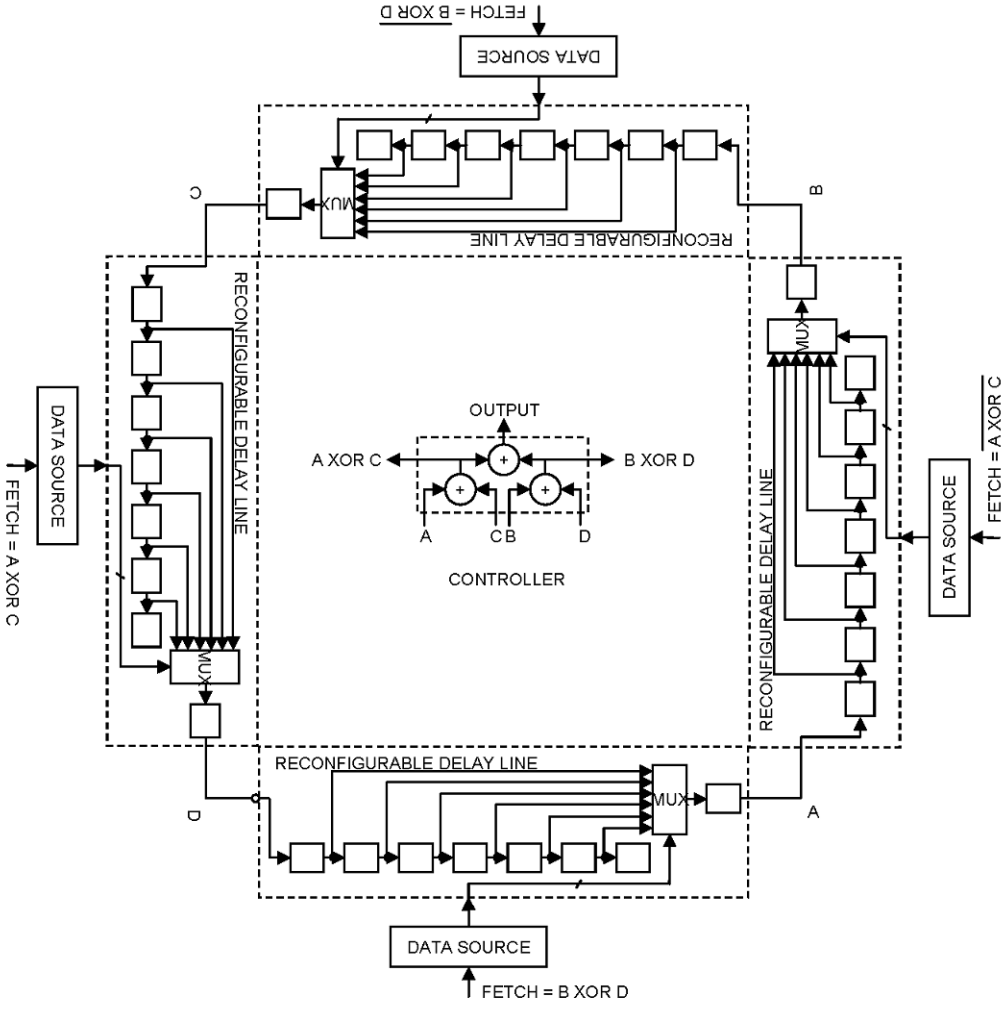


Figure 6.1: System Level Architecture of Asynchronous Transmitter

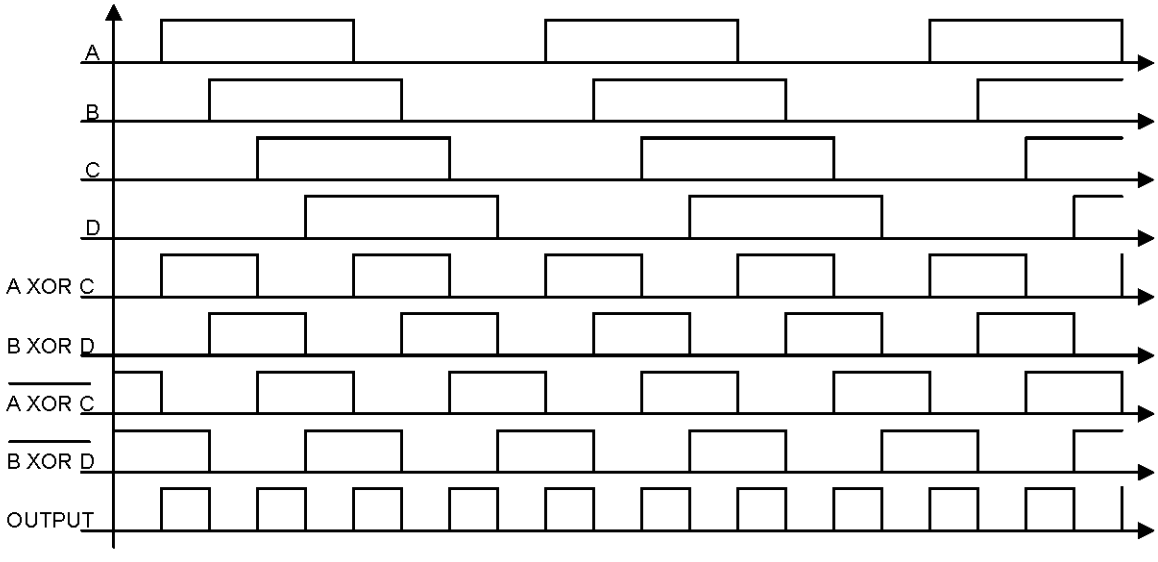


Figure 6.2: Transmitter Waveforms

width of the individual pulses can be adjusted by altering the delay of each reconfigurable delay line. In order to ensure proper functioning of the circuit it is important to ensure that all the delay elements in the delay line have stabilized before the delay line is reconfigured. Reconfiguring the delay line after its output has stabilized does not guarantee this condition will be satisfied. Consider the delay line connecting the nodes A and B. We will assume that each delay cell has a delay of $30ps$. The time required for a change in voltage at point A to propagate through is $6 \times 30ps = 180ps$ (it is not necessary that the change propagate thru the last delay element as it does not feed into the multiplexer). We assume the minimum delay between A and B is $100ps$, the size of the smallest symbol. Thus the signal can propagate through to node B before it has propagated through to the end of the delay line. However, the minimum delay between node A and C is $200ps$. Thus if a change in voltage at node A has propagated through to node C, it must have also propagated through the entire delay line in the reconfigurable delay connecting node A and node B. Thus the delay line can be reconfigured once the change has propagated through to node C. The data source is therefore triggered by $\overline{A \oplus C}$. Using a similar argument, logical expressions for triggering of data sources can be derived. It must be noted that all signals have a

bit period greater than or equal to twice the size of the smallest symbol, even though the difference between two adjacent symbols is smaller than the size of the smallest symbol. Other noteworthy features of the architecture are that the only component operating at full data rate is a single exclusive-or gate that is generating the final output and that the architecture naturally lends itself to time division multiplexing.

The architecture of the receiver is complementary to that of the transmitter. The input data signal drives a twisted ring counter which serves as a frequency divider. The outputs of the twisted ring counter are shown in Figure 6.4. These outputs are used as start and stop signals to control four independent time digitizers which measure the time between the rising edges of the start and stop signals thereby recovering the data. The architecture is illustrated in Figure 6.3. As in the transmitter the period of all signals is greater than or equal to twice the size of the smallest symbol. Furthermore, the only part of the circuit that operates at full data rate is the twisted ring counter which can be implemented using double edge triggered flip flops which themselves can be implemented using either latches or two-to-one multiplexers. These circuits are no more complex than an exclusive-or gate. In the proposed architecture there is no need for clock recovery as it functions by measuring the time between adjacent edges. The architecture is fully asynchronous, all logic gates are triggered by the data signal itself or signals derived from the data signal. Like the transmitter the architecture lends itself to time division multiplexing.

In both the transmitter and receiver all low frequency circuits including delay lines were implemented using differential CMOS circuits. The logic family used was Source Coupled Logic. This choice was made to minimize area and power consumption. A power supply of $1.8V$ was used. A logic level of $1.8V$ was used to represent a logical 1 and $1.2V$ was used to represent a logical 0. High frequency circuits which comprise the exclusive-or gates in the controller and the latches in the twisted ring counter were implemented using differential Bipolar circuits. The logic family used was Emitter Coupled Logic (ECL). A power supply of $3.6V$ was used. A logic level of $2.7V$ was used to represent a logical 1 and $2.1V$ was used to represent a logical 0. In IBM BiCMOS 7HP process technology, the V_{BE} of Bipolar transistors is $0.9V$. The

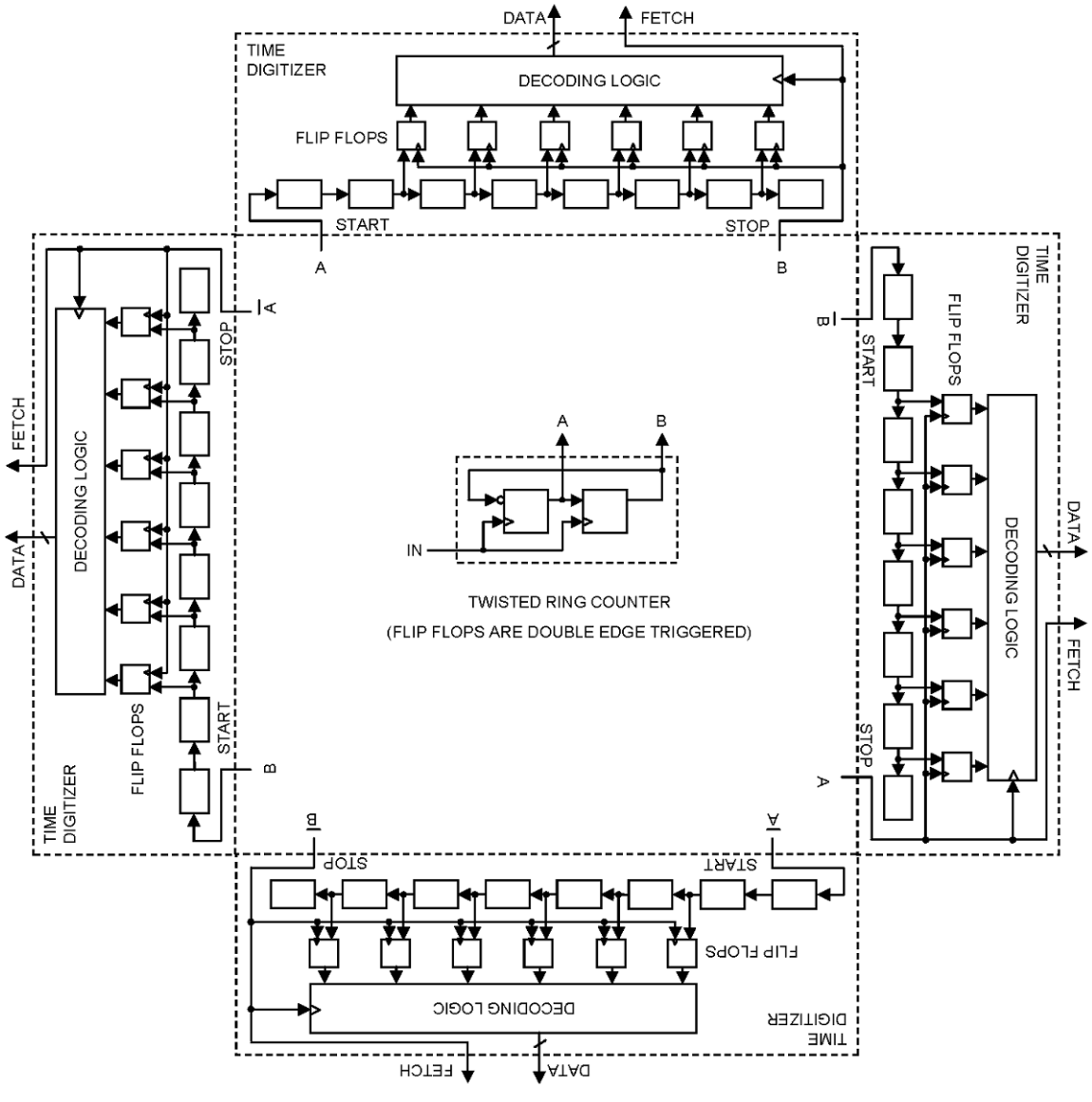


Figure 6.3: System Level Architecture of Asynchronous Receiver

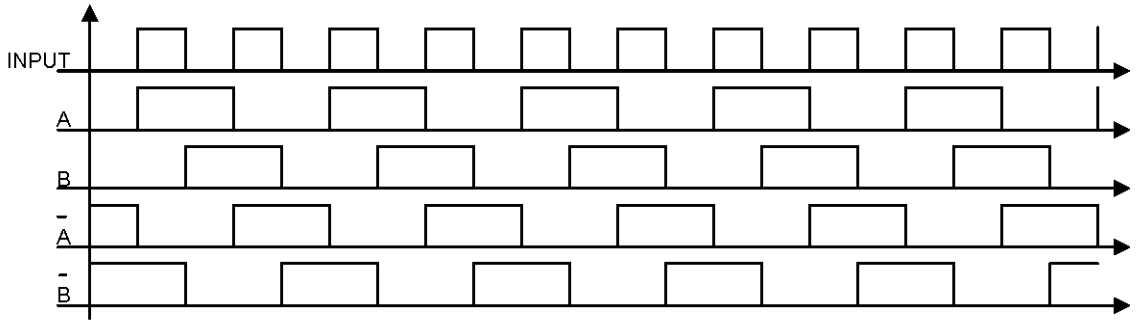


Figure 6.4: Receiver Waveforms

logic levels were chosen to simplify the design of circuits to interface SCL and ECL gates on chip. The transceiver was designed so that the symbols would be 100ps, 115ps, 130ps, 145ps, 160ps, 175ps, 190ps, 205ps, 250ps. The delay of a single delay element was 30ps whereas the intersymbol delay is 15ps. Interpolation circuits were incorporated in both the transmitter and the receiver to achieve delays smaller than the delay of a single delay element. In the next section we will discuss the design of the delay element and phase interpolation circuits. It will also be shown that the six to one multiplexer in the transmitter can also be used for phase interpolation. Also test circuits and pseudo random data sources must be incorporated in both the transmitter and receiver since a transceiver based on this architecture cannot be tested using off-the-shelf Pseudo Random Bit Sequence (PRBS) Generators and Bit Error Rate (BER) Testers. Design of pseudo random data sources and associated test circuitry will be discussed in a separate section. Finally, we will conclude with experimental results obtained from our first prototype.

6.2 Delay Elements and Phase Interpolation Circuits

Critical to a successful implementation of the architecture discussed in this thesis is the design of delay lines which are used in both the transmitter as well as the

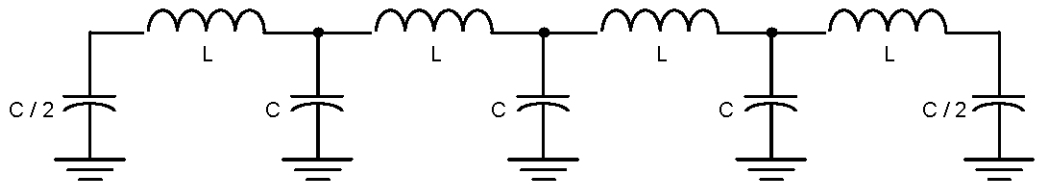


Figure 6.5: Constant-K LC Ladder Structure with 4 Stages (Termination Resistors are Not Shown)

receiver. Delay lines can be either active or passive. There are several passive LC (inductor and capacitor) structures that can be used as delay lines. The best known is the Bessel-Thomson filter that has maximally flat delay response [52], [60], [94] and [97]. However, it is not suitable for integration since it results in delay values that are small, and also its component values become unrealistic as filter order increases [22], [33], [81] and [103]. Most suitable for our application is the Constant-K ladder structure which consists of identical interconnected inductors and capacitors in ladder form as shown in Figure 6.5. The Constant-K ladder is a lumped approximation of a transmission line and hence can be used as a delay line [3] and [80]. It can be shown that the delay of the structure T_D and its characteristic impedance Z_0 are given by the equations below.

$$T_D = n\sqrt{LC} \quad \text{and} \quad Z_0 = \sqrt{\frac{L}{C}} \quad \text{where } n \text{ is the number of segments}$$

The statistical properties of integrated passive delay lines were studied by Analui and Hajimiri [3]. An oscillator with loop delay of 105.2ps was implemented in IBM BiCMOS 7HP process technology. The area of the chip was 900um x 560um. The size of the largest symbol and thus the total delay of the delay line should be $250ps$. In addition we need four delay lines in the transmitter and four delay lines in the receiver. This makes the use of passive Constant-K Delay Lines unfeasible. In our application active delay lines were used. Active delays are sensitive to process, temperature and supply voltage variations. Thus the delay element must be tunable and tuned against a known reference using a delay locked loop. The tunable element used in our design is shown in Figure 6.6 and a simplified circuit model is shown in Figure 6.7. The delay

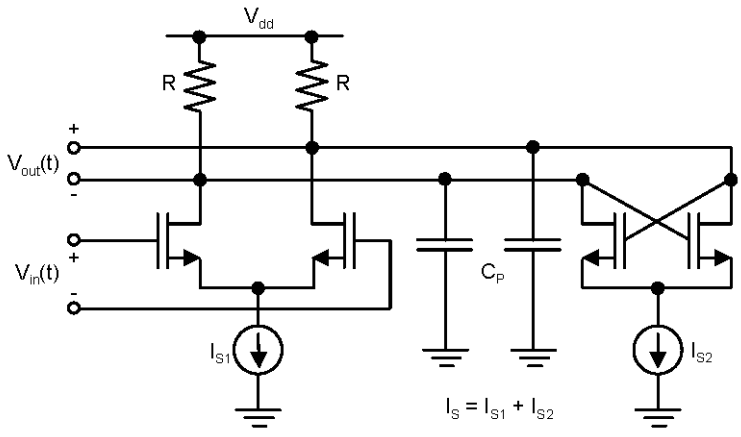


Figure 6.6: Differential Tunable Delay

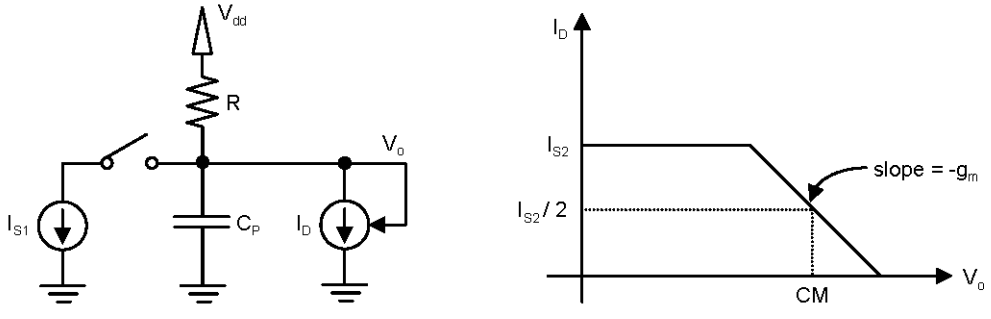


Figure 6.7: Simple Model of Tunable Delay

consists of a current starved differential inverter with a negative resistance placed in parallel with the resistive loads. The cross-connected transistors forming the negative resistance are modeled using a voltage controlled current source. The output current as a function of control voltage is also plotted in Figure 6.7.

It is easy to show that when the switch is closed the steady state solution is reached when $I_D = I_{S2}$. Thus $V_o = V_{dd} - R(I_{S1} + I_{S2})$. When the switch is opened the steady state solution is reached when $I_D = 0$ and thus $V_o = V_{dd}$. This simple model can also be used to explain how changing I_{S2} affects rise and fall time and thus alters the delay. Let the initial values of the tail currents be I_{S1} and I_{S2}. And the new values be I'_{S1} and I'_{S2}. Also $I_{S1} + I_{S2} = I'_{S1} + I'_{S2}$. Without loss of generality we will assume $I'_{S2} < I_{S2}$. We define $I_D(V_o)$ to correspond to the voltage current characteristic when

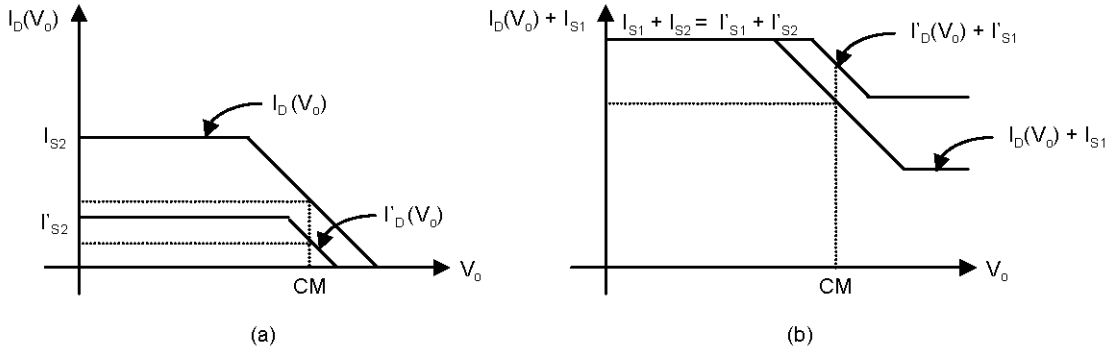


Figure 6.8: Effect of Changing Tail Currents on Tunable Delay

the tail currents are I_{S1} and I_{S2}. And $I'_D(V_o)$ to correspond to the voltage current characteristic when the tail currents are I'_{S1} and I'_{S2}.

First let us consider the case when the switch is opened. In this case no current flows into the current source. By Kirchoff's Current Law (KCL) it follows that

$$\frac{dV_o}{dt} = \frac{1}{C}\left[\frac{V_{dd} - V_o}{R} - I_D(V_o)\right]$$

Both $I_D(V_o)$ and $I'_D(V_o)$ are plotted in the Figure 6.8 (a). Note that for all values of $I_D(V_o) \geqslant I'_D(V_o)$. Thus, decreasing I_{S2} increases dV_o/dt for all values of V_o which decreases the rise time.

Next we consider the case when the switch is closed. In this case a current of I_S flows into the current source. Again by KCL it follows that

$$\frac{dV_o}{dt} = -\frac{1}{C}\left[(I_D(V_o) + I_{S1}) - \frac{V_{dd} - V_o}{R}\right]$$

We have plotted $I_D(V_o) + I_{S1}$ and $I'_D(V_o) + I'_{S1}$ in Figure 6.8 (b). It is evident that decreasing I_{S2} increases $I_D(V_o) + I_{S1}$ for all V_o. Thus the magnitude of dV_o/dt increases for all values of V_o. Thus the fall time will decrease as well. Given that the effect of decreasing I_{S2} is a decrease in both rise and fall times, it follows that its effect is to decrease the delay.

The delay of a single delay element is greater than the smallest required delay value. In order to generate delays smaller than the delay of the delay element phase interpolation was used [113]. As illustrated in Figure 6.9 the function of the phase

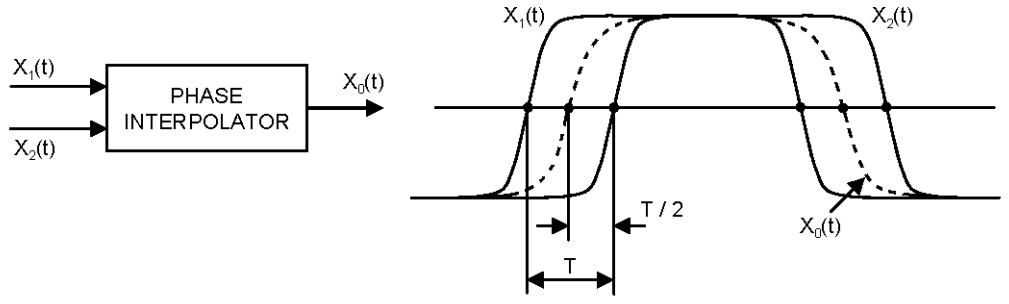


Figure 6.9: Phase Interpolation Concept

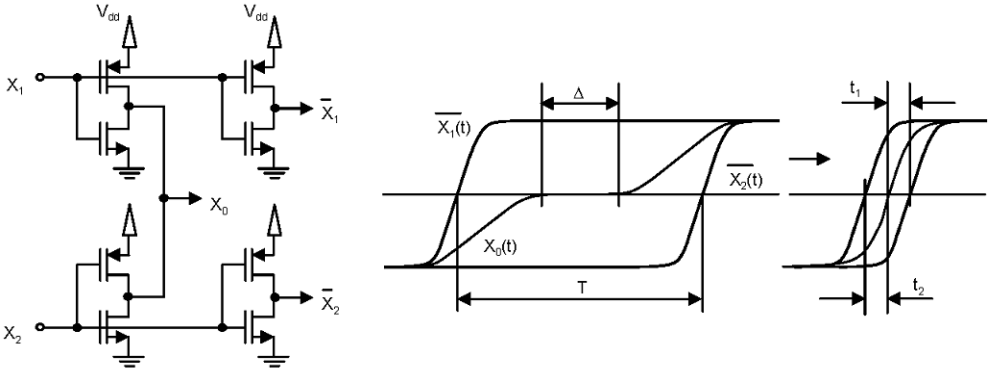


Figure 6.10: Static CMOS Phase Interpolator

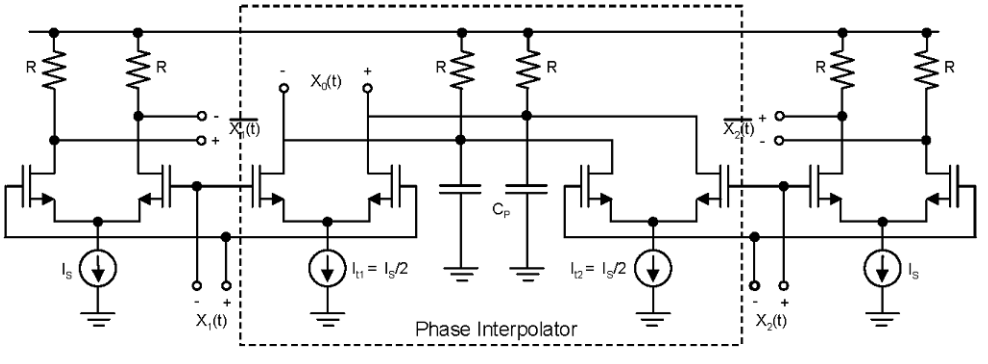


Figure 6.11: A Differential Current-Mode Phase Interpolator

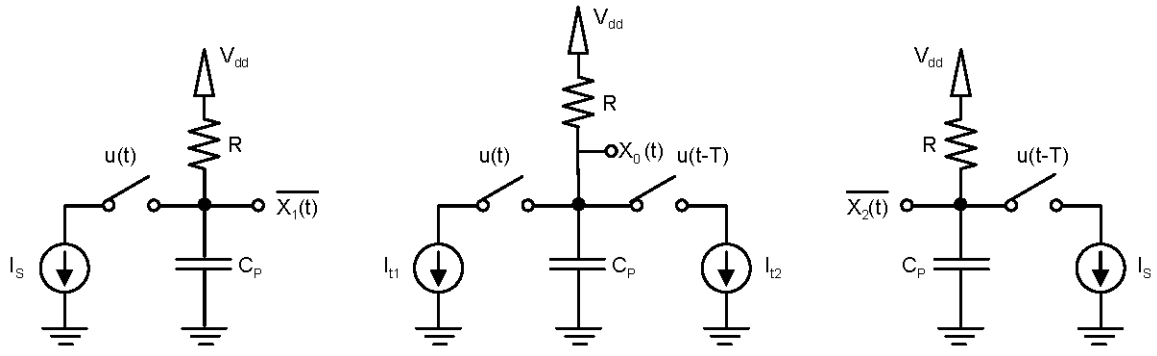


Figure 6.12: Simplified Model of Phase Interpolator

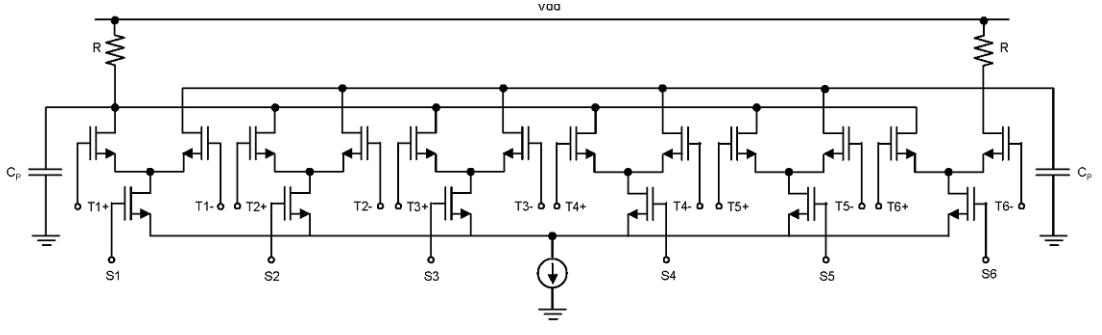


Figure 6.13: A Differential 6-1 Multiplexer and Phase Interpolator

interpolator is to generate an output signal X_0 for which the zero crossings are ideally placed halfway between the zero crossings of two input signals X_1 and X_2. Figure 6.10 shows a CMOS design of a phase interpolator. Interpolation is accomplished by shunting the outputs of two CMOS inverters. The output signal, X_0 is high when both inputs are low and low when both inputs are high. However, when one input is high and the other is low, X_0 becomes approximately $V_{dd}/2$ assuming the transistors are sized properly. As the delay between signal transitions in X_1 and X_2 is reduced, the intermediate region shown in Figure 6.10 shrinks and eventually disappears resulting in a smooth transition. In this case the circuit functions as a phase interpolator. For this circuit to function properly, the delay between X_1 and X_2 must be small, namely comparable to one gate delay, so that the transitions overlap to some extent. It is important to observe that the interpolated signal has a slower edge rate than the two original signals. Unfortunately this makes X_0 more sensitive to supply and substrate induced jitter. Supply noise rejection is one of the issues that must be addressed in the design of the phase interpolator as it can directly affect the location of the zero crossings of the interpolated signals. Although the single ended CMOS design of Figure 6.10 has the advantages of simplicity and low static power dissipation, its supply sensitivity is poor specially during signal transitions. Shown in Figure 6.11 is a current-mode implementation of a phase interpolator. The circuit operates on the same principal as the CMOS design of Figure 6.10. However, because of the differential implementation, it has much better supply rejection. Each differential pair is essentially an SCL inverter carrying half the tail current needed to provide a full logic swing at the gates outputs. Therefore, just as in the CMOS implementation, each input can only cause half a transition at the output. The resulting waveforms are similar to those shown in Figure 6.10. A simplified model of the phase interpolator in Figure 6.11 is shown in Figure 6.12. In this representation. switching functions of the transistors are modeled by applying current sources to the output nodes X_0, X_1 and X_2 at appropriate times. From the model it follows that

$$X_0(t) = V_{dd} + \frac{RI_S}{2}\left(\left(e^{-\frac{t}{RC_P}} - 1\right)u(t) + \left(e^{-\frac{t-T}{RC_P}} - 1\right)u(t-T)\right)$$

$$X_1(t) = V_{dd} + \frac{RI_S}{2}\left(e^{-\frac{t}{RC_P}} - 1\right)u(t)$$

$$X_2(t) = V_{dd} + \frac{RI_S}{2}\left(e^{-\frac{t-T}{RC_P}} - 1\right)u(t-T)$$

From these equations it follows that the transition point of X_0 is approximately (not exactly) midway between that of X_1 and X_2. The differential six to one multiplexer used in the transmitter is shown in Figure 6.13. In this multiplexer if two select lines were activated then the output would be an interpolated version of the corresponding outputs. In this case the circuit functions in much the same way as the phase interpolator in Figure 6.11.

6.3 Pseudo Random Data Sources

Conventional high speed transceivers can be tested using off-the-shelf test equipment. Since our architecture is radically different pseudo random data sources and test circuits must be integrated on chip. On chip test circuits should enable complete characterization of both the performance of the transmitter and receiver. Conventional transmitters and receivers are tested using PRBS testers which are based on Linear Shift Registers (LSR) [38]. LSRs can be implemented in two ways. The Fibonacci implementation, sometimes also referred to as a simple shift register generator (SSRG), consists of a simple shift register in which a binary weighted two modulo sum of the taps is fed back to the input. The modulo-2 sum of two 1-bit binary numbers yields 0 if the two numbers are identical, and 1 if they differ: 0+0=0, 0+1=1, 1+1=0. It is equivalent to XOR of the two 1-bit binary numbers. The general Fibonacci implementation is shown in Figure 6.14. For any given tap, weight g_i is either 0, meaning "no connection," or 1, meaning it is fed back. Two exceptions are g_0 and g_m, which are always 1 and thus always connected. Note that g_m is not really a feedback con-

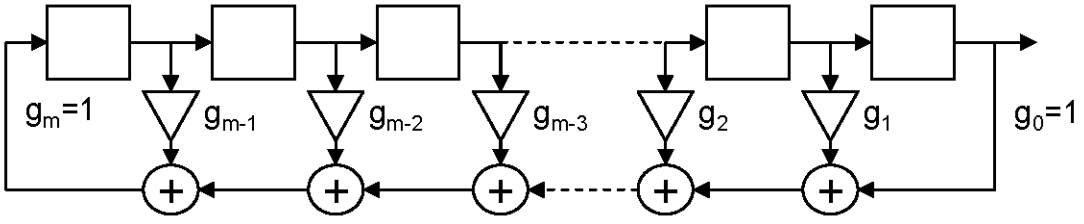


Figure 6.14: Fibonacci Implementation of a Linear Shift Register

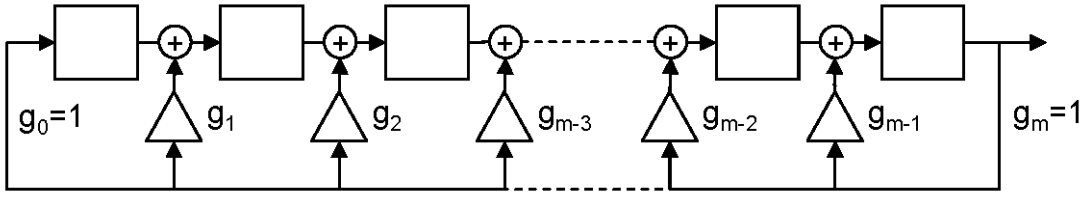


Figure 6.15: Galois Implementation of a Linear Shift Register

nection, but rather is the input of the shift register. The Galois implementation, also referred to as a multiple-return shift register generator (MRSRG) or modular shift register generator (MSRG), consists of a shift register, the contents of which are modified at every step by a binary-weighted value of the output stage. The general Galois implementation is shown in Figure 6.15. Careful inspection reveals that the order of the Galois weights is opposite that of the Fibonacci weights. Given identical feedback weights, the two LSR implementations will produce the same sequence. However, the initial states of the two implementations must necessarily be different for the two sequences to have identical phase. The initial state of the Fibonacci form is called the initial fill or initial vector, and this initial fill comprises the first m bits output from the generator. The initial state of the Galois generator must be adjusted appropriately to attain the equivalent initial fill.

LSR generators produce what are called linear recursive sequences (LRS) because all operations are linear. Generally speaking, the length of the sequence before repetition occurs depends upon two things, the feedback taps and the initial state. An LSR of a given size m (number of registers) is capable of producing a maximum of $2^m - 1$ states, but will do so only if proper feedback taps, or terms, have been chosen. Such

a sequence is called a maximal length sequence, maximal sequence, or less commonly, maximum length sequence. It is often abbreviated as m-sequence. These sequences are also referred to as a pseudonoise (PN) or pseudorandom sequences, due to their optimal noise-like characteristics. Maximal length generators can actually produce two sequences. One is the trivial one, of length one, that occurs when the initial state of the generator is all zeros. The other one, the useful one, has a length of $2^m - 1$. Together, these two sequences account for all 2^m states of an m-bit state register. Finite (Galois) field mathematics are used to derive m-sequence feedback taps. Any LFSR can be represented as a polynomial of variable X, referred to as the generator polynomial.

$$G(X) = g_m X^m + g_{m-1} X^{m-1} + g_{m-2} X^{m-2} + ... + g_2 X^2 + g_1 X^1 + g_0$$

The coefficients g_i represent the tap weights, as defined in Figure 6.14 and Figure 6.15, and are 1 for taps that are connected (fed back), and 0 otherwise. The order of the polynomial, m, represents the number of LSR stages. Rules of linear algebra apply to the polynomial, but all mathematical operations are performed modulo-2. The generator polynomial of the equation above is said to be primitive if it cannot be factored (i.e. it is prime), and if it is a factor of (i.e. can evenly divide) $X^n + 1$, where $n = 2^m - 1$ (the length of the m-sequence). It can be shown that an LSR represented by a primitive polynomial will produce a maximal length sequence. Alternately, for the purposes of design, one can find all prime factors of order m, of the polynomial $X^n + 1$. Consider a simple example. Let the number of registers m be 3. The length of the maximal sequence $n = 2^m - 1 = 7$. Now note that,

$$X^7 + 1 = (X + 1)(X^3 + X + 1)(X^3 + X^2 + 1)$$

Notice that $(X+1)(X^3+X+1)(X^3+X^2+1) = X^7+2X^6+2X^5+4X^4+4X^3+2X^2+2X+1$. When arithmetic is done modulo 2, $2X = 0$, $2X^2 = 0$ and so on and so forth. Thus $X^7 + 2X^6 + 2X^5 + 4X^4 + 4X^3 + 2X^2 + 2X + 1 = X^7 + 1$.

The two factors of order 3 are X^3+X+1 and X^3+X^2+1. The Fibonacci and Galois implementation corresponding to $X^3 + X + 1$ are shown in Figure 6.16. The sequence

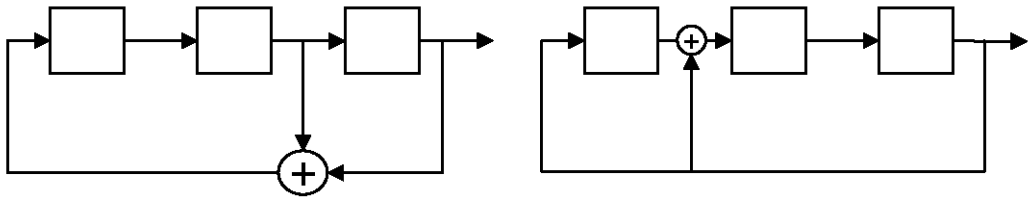


Figure 6.16: Fibonacci and Galois Implementation of Linear Shift Registers Corresponding to the Generating Polynomial $X^3 + X + 1$

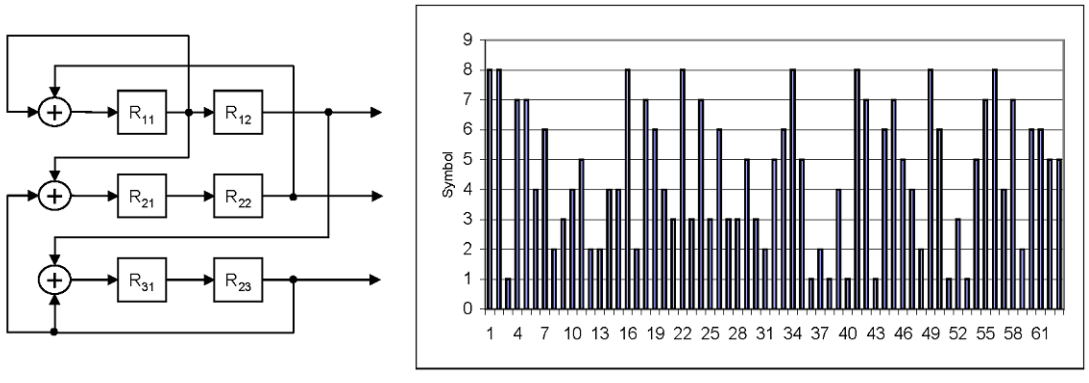


Figure 6.17: A Cross Coupled Two Dimensional Linear Shift Register and the Corresponding m-sequence.

generated by this LSR is $1, 1, 1, 0, 0, 1, 0$. It is left as an exercise for the reader to verify that the sequence generated by LSR corresponding to $X^3 + X^2 + 1$ is maximal. LSRs are used as PRBS circuits for testing conventional transceivers. In the context of the architecture outlined above, what is needed is a Pseudo Random Symbol Source (PRSS) as we have 8 different symbols excluding the blank symbol. Note that the 8 different symbols can be represented using different combination of 3 binary values. One technique to implement a PRSS would be to implement 3 independent LSRs. There are two primary disadvantages. First the length of the sequence would only be $2^m - 1$, whereas the total number of registers is $3m$ and therefore the shift registers can be in $2^{3m} - 1$ non-zero states. The second disadvantage is that all the symbols would not be generated with the same frequency. In fact if three LSRs shown in Figure 6.16 were implemented, the system would cycle through 7 symbols, and at least 1 symbol would not be generated at all.

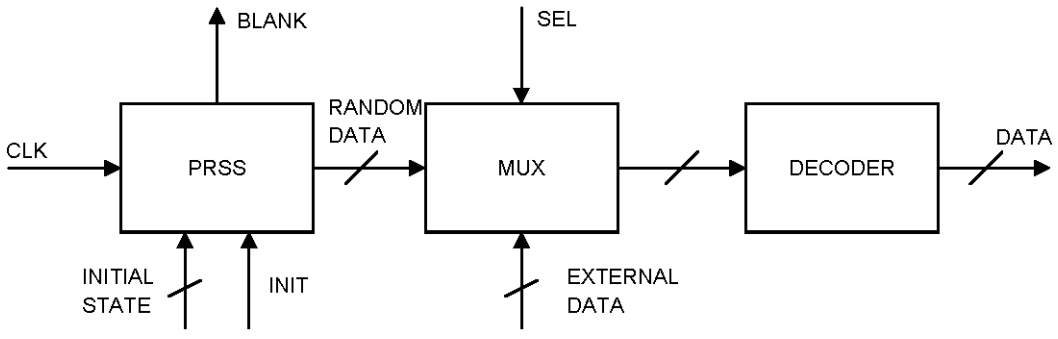


Figure 6.18: System Level Schematic of On Chip Datasource

The circuit used in our design is shown in Figure 6.17. It is inspired by Fibonacci Type LSRs. The fundamental difference between this and the previously proposed architecture is that the feedback terms are dependent on all registers. There is a cross coupling of feedback terms. The sequence generated by this circuit is also shown in Figure 6.17. This sequence has $63 = 2^{3*2} - 1$ elements and is thus maximal. Two immediate consequences of maximality are that all symbols will be generated and all symbols will be generated with approximately the same frequency. Specifically the symbol represented by 000 would be generated 7 times (corresponding to the 7 different states for registers R_{11}, R_{21} and R_{31}. When the output symbol is 000, these registers can be in any state excluding 000). All other symbols will be generated 8 times as the registers R_{11}, R_{21} and R_{31} can be in any 1 of 8 possible states.

In order to perform error measurements, the data sources in the receiver and transmitter must not only be identical, they must be synchronized to one another. Symbols from the data source in the transmitter are transmitted across the channel and processed by the receiver. In order to determine if the symbol was received without error it can be compared against the output of the data source in the receiver assuming that the data source in the receiver is generating the same pattern as the data source in the transmitter and is synchronized with it. In order to achieve synchronization, a "blank" symbol must be transmitted at the end of each cycle of the data source. When a blank is received, the data source in the receiver is reset to

the state after the blank state effectively synchronizing it with the data source in the transmitter. Every state in a maximal sequence of an LSR corresponds of a unique state of its registers. Thus a simple method to generate a blank symbol once during each cycle is to define a blank state. In our design we generate a blank every time all the registers R_{11}, R_{21}, R_{31}, R_{12}, R_{22} and R_{32} are set to 1.

Sym 1	Sym 2	Sym 3	Sym 4	Sym 5	Sym 6	Sym 7	Sym 8	Blank
0000	0001	0010	0011	0100	0101	0110	0111	1xxx
000001	000011	000010	000110	000100	001100	001000	011000	100000

The PRSS circuit described above uses a binary representation to represent the symbols. However, an alternate representation is used by the six to one multiplexer used in the reconfigurable delay lines. The two representations are summarized in the table above. A decoder circuit was implemented to transform the output of the PRSS circuit to the representation used by the reconfigurable delay lines. Also four two to one multiplexers were placed between the PRSS circuit and the decoder circuit. The multiplexers enabled selection of either pseudo random content from the internal data source or externally user specified data. The system level schematic of the datasource is shown in Figure 6.18.

6.4 Error Unit for Symbol Error Rate Measurement

As mentioned LSR based circuits are used to perform Bit Error Rate (BER) measurements on conventional transceivers. Of the shelf BER measurement equipment is available for testing high data rate transceivers. This equipment cannot be used to measure Symbol Error Rate (SER) for transceivers based on the architecture proposed in these thesis. Therefore, in conjunction with pseudo random data sources described in the previous section an error unit was incorporated on chip. The system level schematic of the error unit is shown in Figure 6.19. It consists of a digital comparator which compares data on two buses. Two outputs, specifically REC and ERR

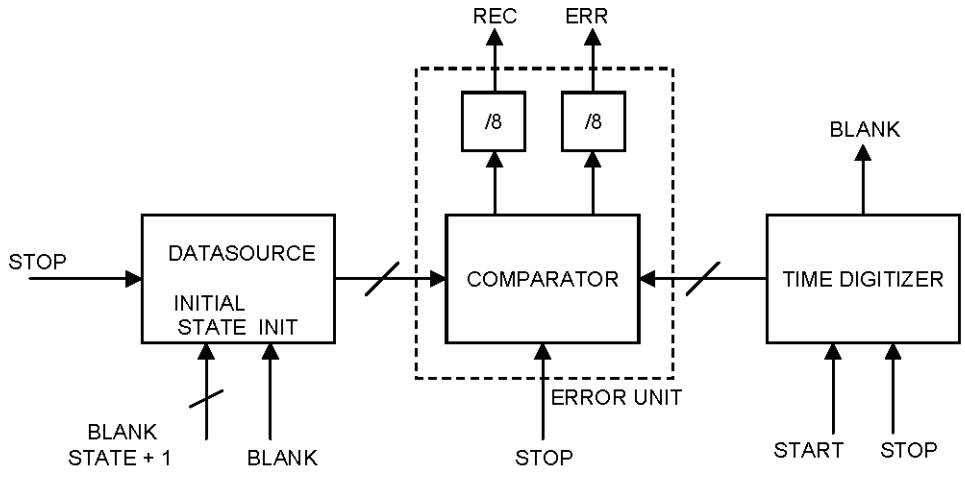


Figure 6.19: Error Unit Integrated with Data Source and Time Digitizer Circuit

are produced. REC toggles every time a symbol is received and ERR toggles every time an error is produced. These outputs are directed into divide by 8 circuits which reduce the frequency of these signals to allow interfacing to off-the-shelf low frequency counter circuits. Also shown in the figure, is how the error unit is integrated with the data source and the time digitizer. Note that the blank output of the time digitizer is used to reset the data source. This is the synchronization circuit used to ensure that the data sources in the transmitter and receiver and synchronized with one another. Also it must be noted that all circuits are "clocked" with the stop signal to the time digitizer.

These circuits can be used to perform a myriad of different tests. The transmitter can be configured to transmit data from the external data source. The receiver can also be configured to compare received data against data specified externally. By repeating this experiment for different symbol combinations, one can calibrate the receiver. By configuring the transmitter to transmit data from the pseudo random data source and configuring the receiver to compare received data against externally specified symbols, one can measure the frequency of different received symbols, and compare them against their expected frequency (given the properties of the pseudo random data source, the expected frequencies are fixed and known). By configuring

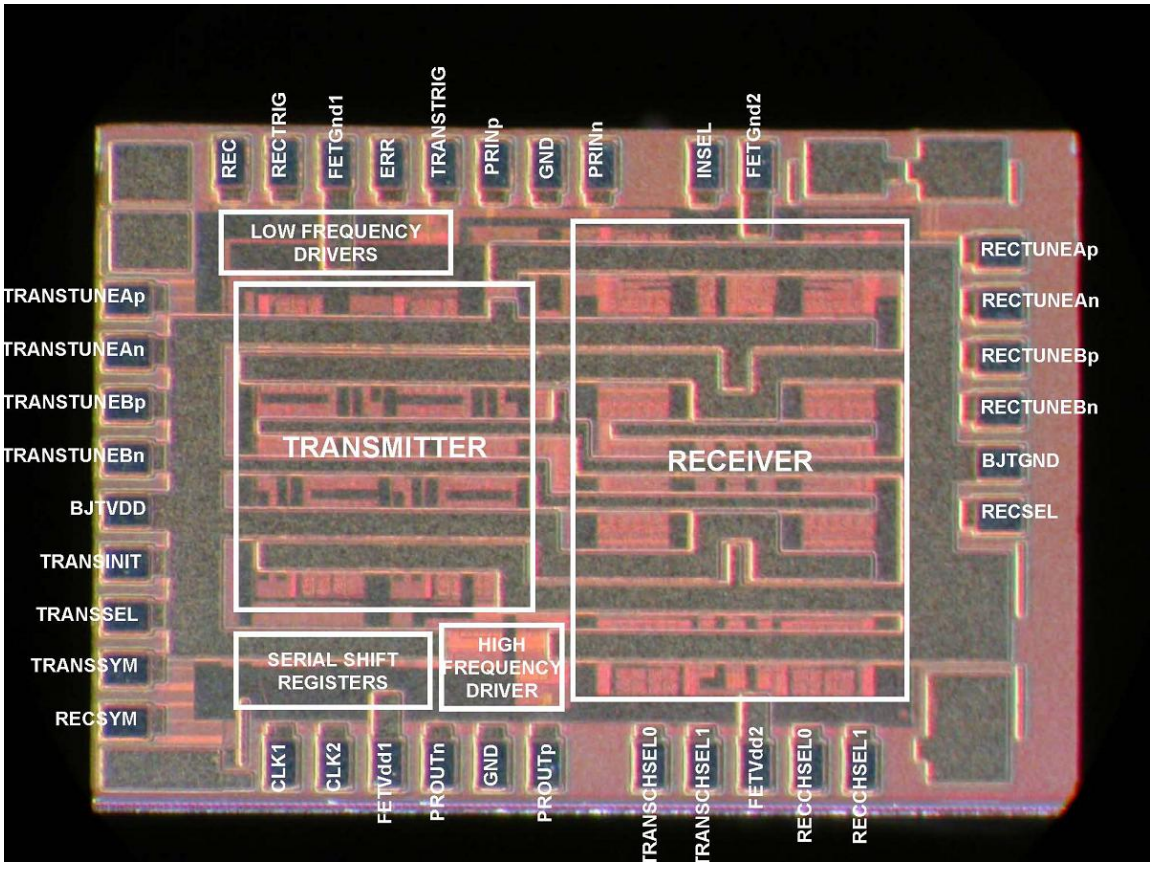


Figure 6.20: Die Micrograph of Experimental Prototype Fabricated in 0.18μm IBM BiCMOS 7HP Technology

the transmitter to transmit data from the pseudo random data source and configuring the receiver to measure received data against data from the pseudo random data sources one can measure the SER.

6.5 Experimental Results

An experimental circuit based on the architecture discussed in the previous section has been designed and integrated in 0.18μm IBM BiCMOS 7HP process technology. The circuit consists of a transmitter and receiver pair both of which have built-in pseudo random data (symbol) sources. Both the transmitter and receiver can also be supplied data (symbols) externally which must be loaded into two on-chip shift registers via a parallel port. The output of the transmitter connects to a high frequency

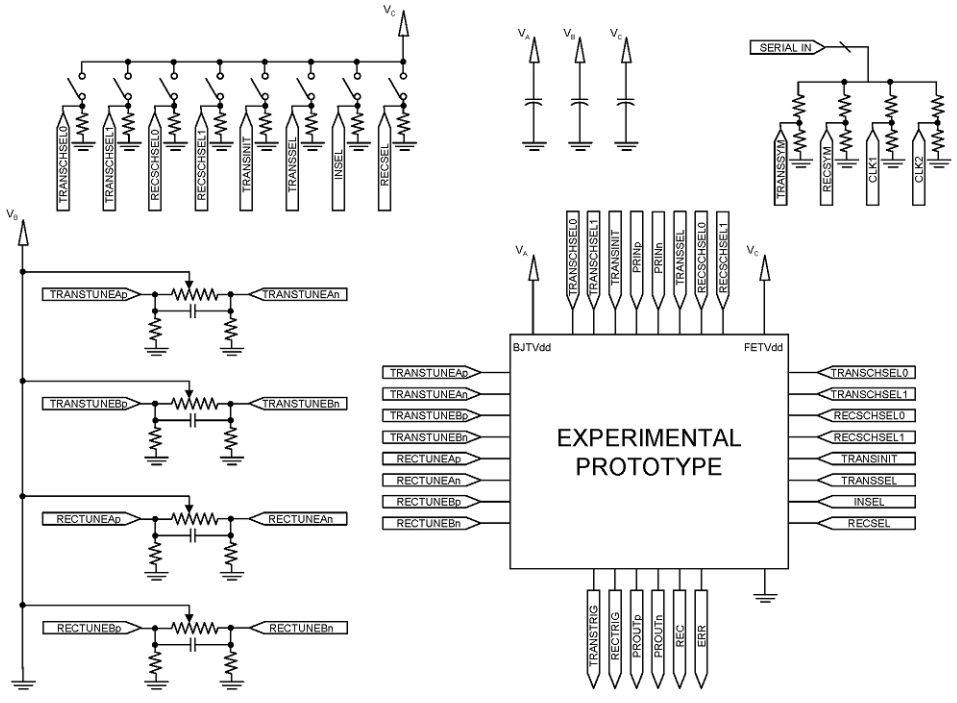


Figure 6.21: Control Circuit Schematic

driver circuit, the output of which can be probed to allow direct measurements on the transmitter output. The receiver input can be configured to connect to the transmitter output or it can be configured to connect to an external signal supplied by a high frequency pattern generator via a probe. This facilitates independent testing of the receiver circuit. The low frequency outputs of the transmitter and the receiver are connected to on chip low frequency driver circuits. These outputs facilitate testing of the transmitter and receiver using off-the-shelf low frequency test equipment without the need of performing high frequency measurements. A die micrograph of the prototype is shown in Figure 6.20. The delay locked loops for tuning of delay lines in the transmitter and receiver were not integrated in this prototype. Instead, these delay lines can be tuned using external control voltages. The die has an active area of 2mm x 1mm.

In order to characterize the experimental prototype an off-chip control circuit was designed. This circuit consists of a voltage divider circuit to level shift the output of the parallel port from 0-3.3V to 0-1.8V. This was necessary in order to interface

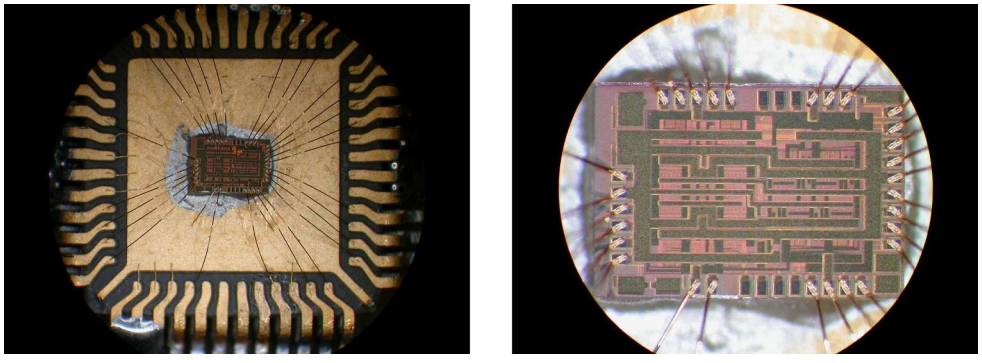


Figure 6.22: Experimental Prototype Wirebonded to Chip Carrier Package

the on chip shift registers which use a voltage level of 1.8V to represent a logical 1 and a voltage level of 0V to represent a logical 0 with the parallel port which uses a voltage level of 3.3V to represent a logical 1 and a voltage level of 0V to represent a logical 0. Also included in the circuit were tunable voltage dividers to generate the control voltages for tuning the delay lines in both the transmitter and receiver. The tuning circuits include a tapered cascade of surface mount capacitors for filtering noise which would translate into jitter on the transmitter output and measurement uncertainty in the receiver. The circuit also includes a DIP switch based circuit for generating the digital control voltages. A tapered cascade of electrolytic and surface mount capacitors is used to bypass all supply voltages. The schematic of the control circuit is shown in Figure 6.21.

A 3"x6" four plane copper printed circuit board (PCB) was designed to house the control circuit and the experimental prototype. The experimental prototype was mounted on a low inductance 48 pin chip carrier package using silver epoxy. This chip carrier package was soldered to the PCB and all pads on the experimental prototype, excluding the high frequency input and output pads were wirebonded to corresponding pads on the chip carrier package, as illustrated in Figure 6.22. Notice that due to the relative difference in size of the experimental prototype and the chip carrier package, the wirebonds used were long and thus had high inductance values. Due to high inductance values of the wirebond, the high frequency input and output were not wirebonded, but left exposed for probing. For the high frequency output

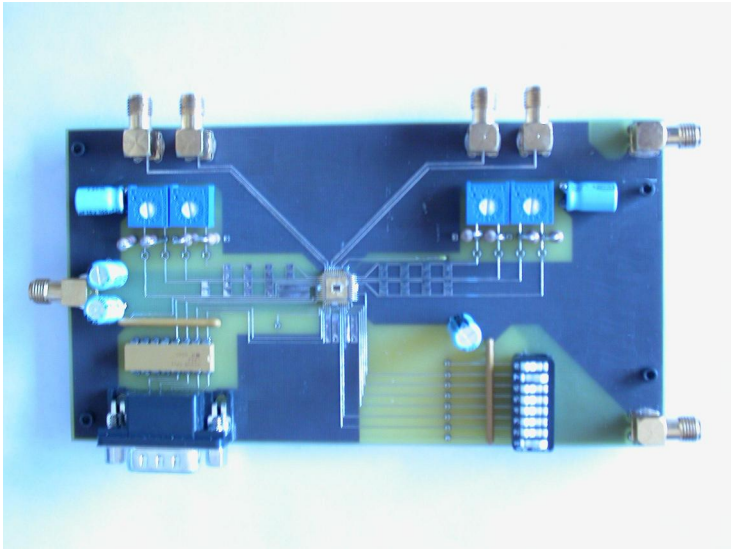


Figure 6.23: Printed Circuit Board with Control Circuitry, Chip Carrier and Experimental Prototype

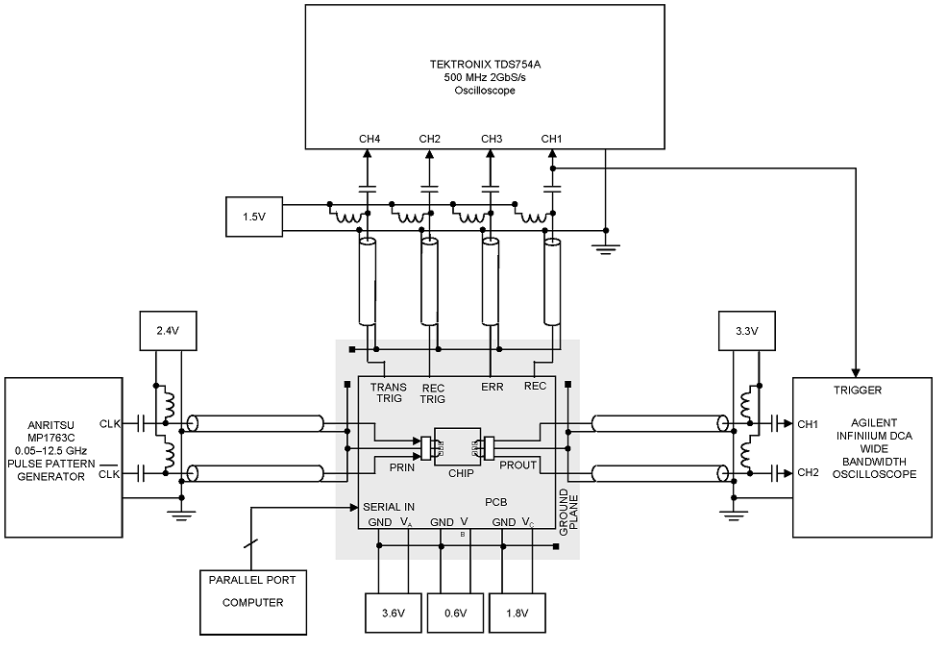


Figure 6.24: Measurement Setup

no on-board termination resistors were used since the reflections would be absorbed by the 50Ω impedance of the test equipment. For the high frequency input, there was no need for termination resistors, as these were built into the chip. The low frequency outputs were connected to SMA connectors via 50Ω transmission lines to allow connections to test equipment. No termination resistors were used on-board since signal reflections will be absorbed by the 50Ω input impedance of the test equipment. SMA connectors were also provided for the voltage supplies and a DB-9 connector was provided to interface the parallel port to the shift register circuitry. An assembled PCB with the experimental prototype mounted on a chip carrier package is shown in Figure 6.23. The clock output on an Anritsu MP1763C 0.05-12.5 GHz Pulse Pattern Generator was used to produce square waves of different frequencies for testing the receiver. This could be done by adjusting the data rate. An Agilent Infiniium DCA Wide Bandwidth Oscilloscope was used to directly monitor the high frequency output of the transmitter. The low frequency REC output was used as the trigger signal. The low frequency outputs were interfaced to a Tektronix TDS754A 500 MHz 2 GbS/s Oscilloscope. Bias-Ts were used to AC couple all input and output signals to test equipment and supply the appropriate bias voltage. The test setup is shown in Figure 6.24.

Two different methodologies were used to characterize the performance of the transmitter. The first was based on low frequency measurements and the second on high frequency measurements. First we will discuss the low frequency measurements. For these measurements the transmitter was configured to transmit symbols from the external data sources. Note that four symbols can be specified and they are transmitted in a round robin fashion. Let the durations of the four symbols be D_1, D_2, D_3 and D_4. Note that the period of receive signal, P_{REC} is given by the identity below.

$$P_{REC} = 16 * (D_1 + D_2 + D_3 + D_4)$$

The receive output of the error unit toggles every time a symbol is received on that channel. This happens after $D_1 + D_2 + D_3 + D_4$ time units. Thus the period

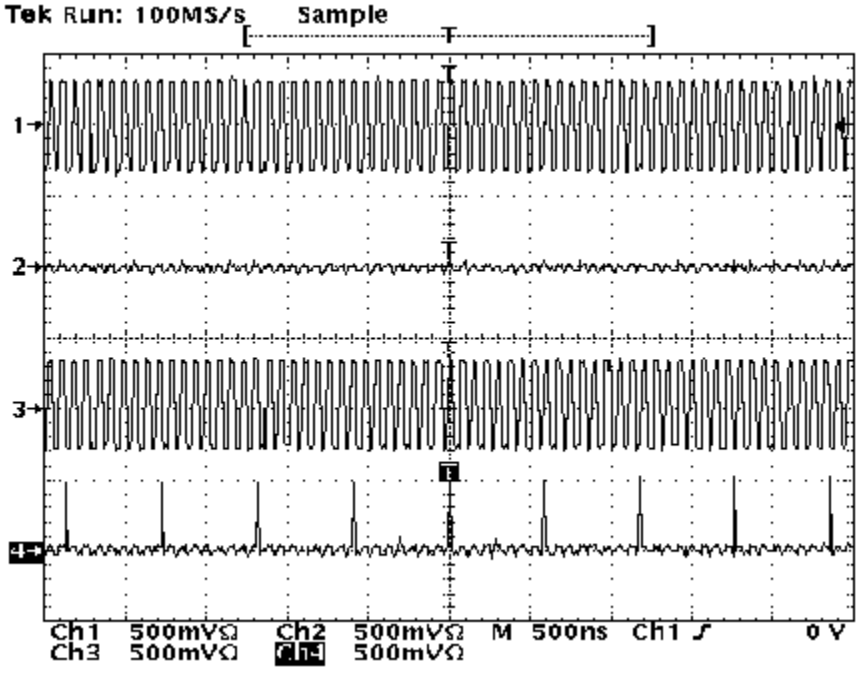


Figure 6.25: Low Frequency Transmitter Measurements

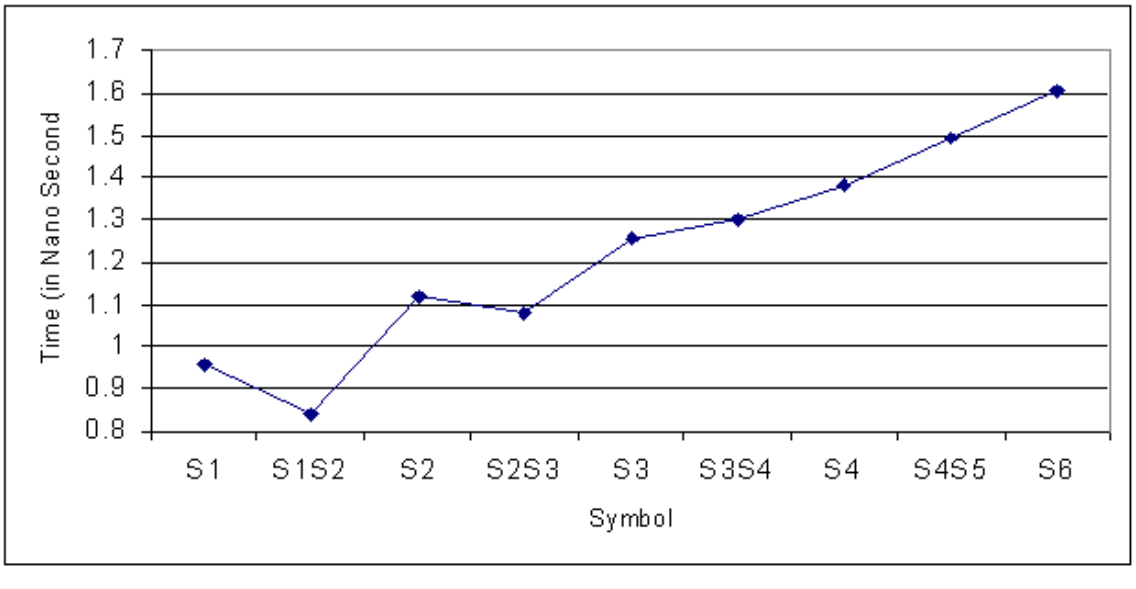


Figure 6.26: Delay Lines Characterized Using Low Frequency Measurements

of this signal is $2 * (D_1 + D_2 + D_3 + D_4)$. This output serves as the input to a frequency divider circuit which reduces the frequency by a factor of 8. The output of the frequency divider circuit is the receive signal that is measured. Note that if all the external data sources were to transmit the smallest symbol which has duration S_1, then

$$P_{REC} = 64 * S_1 \text{ or equivalently } S_1 = \frac{P_{REC}}{64}$$

Thus the duration of the smallest symbol can be measured by measuring the period of the REC signal. A similar methodology can be used to infer the sizes of all other symbols. An alternate method to measure the size of the symbol is to measure the time between pulses on the TRANSTRIG output. Note that the internal data sources cycle once after $2^n - 1$ symbols have been transmitted, where n is the number of flip flops in the data source. The time required to transmit $2^n - 1$ is $2^n - 1 * (D_1 + D_2 + D_3 + D_4)$ time units. Also the frequency is divided by a factor of 2. Thus

$$P_{TRANSTRIG} = 2 * 63 * (D_1 + D_2 + D_3 + D_4)$$

If all the external data sources were to transmit the smallest symbol which has a duration of S_1, then

$$P_{TRANSTRIG} = 504 * S_1 \text{ or equivalently } S_1 = \frac{P_{TRANSTRIG}}{504}$$

Hence, it follows that

$$P_{TRANSTRIG} = 504 * \frac{P_{REC}}{64} = 7.875 * P_{REC}$$

Sample output from the Tektronix TDS754A Oscilloscope is plotted in Figure 6.25. The REC output is on Channel 1 and the TRANSTRIG output is on Channel 4. Channel 2 is the RECTRIG output and Channel 3 is the ERR output. These will be discussed later. Notice that $P_{TRANSTRIG} \approx 8 * P_{REC}$. The experiment was repeated for different symbols and the symbol duration computed from time period measurements performed on the REC and TRANSTRIG outputs. The graph is plotted in

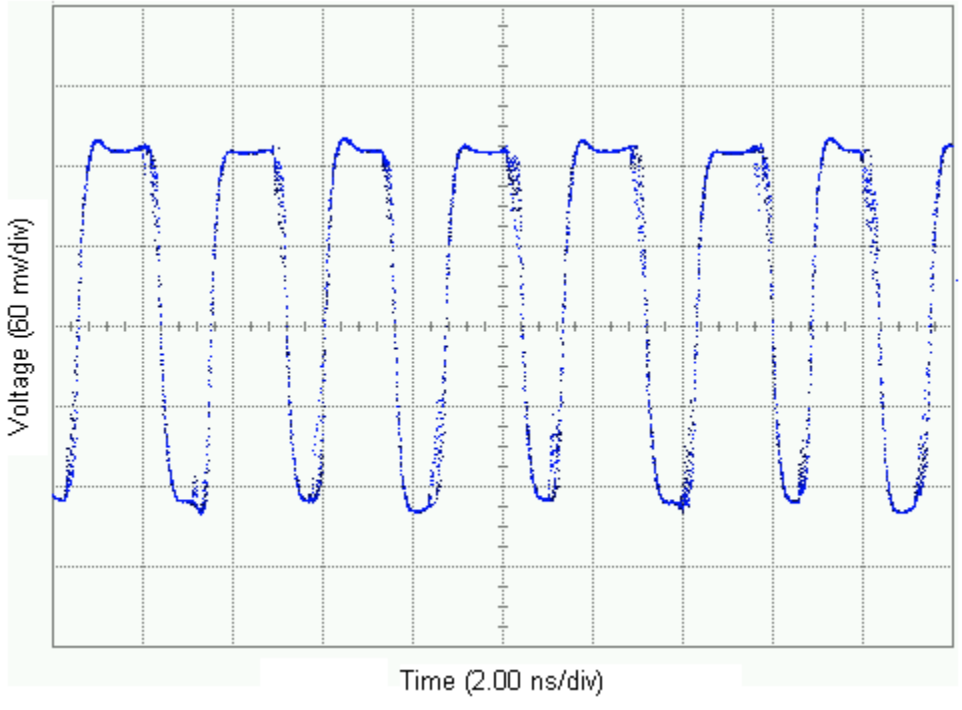


Figure 6.27: High Frequency Transmitter Measurements

Figure 6.26. As mentioned previously, the symbol durations can also be measured by performing high frequency measurements on the output of the transmitter. The transmitter was configured to transmit the smallest symbol S_1 on Channel 1 and Channel 3 and transmit the largest symbol S_{BLANK} on Channel 2 and Channel 4. The output on the Agilent Infiniium DCA Wide Bandwidth Oscilloscope is plotted in Figure 6.27. The transmitter was then configured to transmit the smallest symbol S_1 on all four channels and the time period of the output was measured. The duration of the symbol is just half the time period. This measurement was repeated for all the symbols. The results are plotted in Figure 6.28. They are in excellent agreement with those obtained by performing low frequency measurements on the REC and TRANSTRIG output and shown in Figure 6.26.

There are several issues that must be addressed. Firstly, the transmitter was designed so that the duration of the smallest symbol would be approximately $100ps$. The smallest measured symbol duration is approximately $850ps$. Secondly, the circuit was designed so that the time difference between two adjacent symbols would

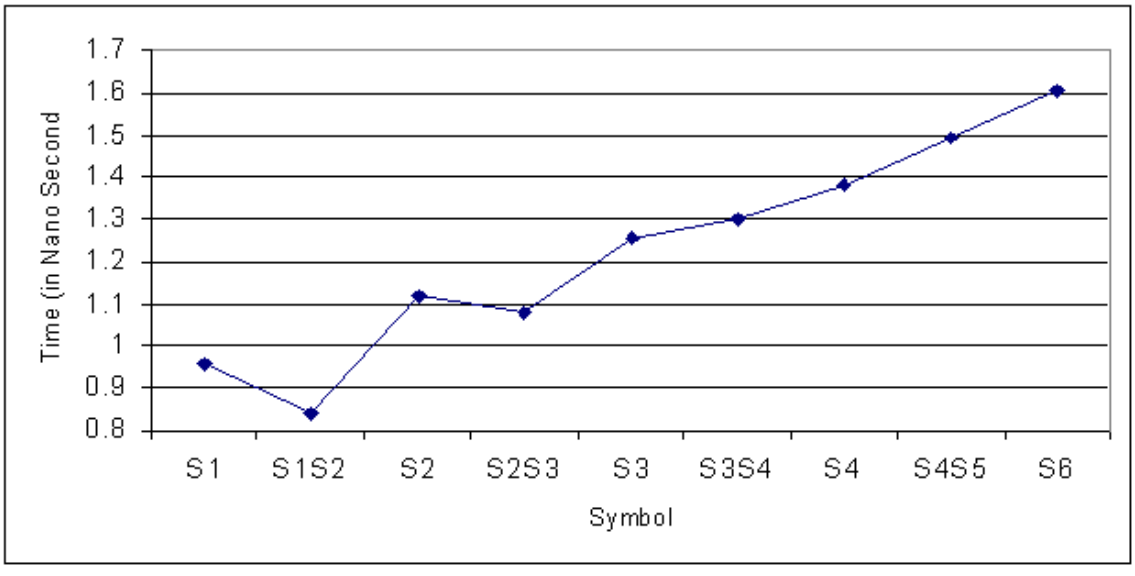


Figure 6.28: Delay Lines Characterized Using High Frequency Measurements

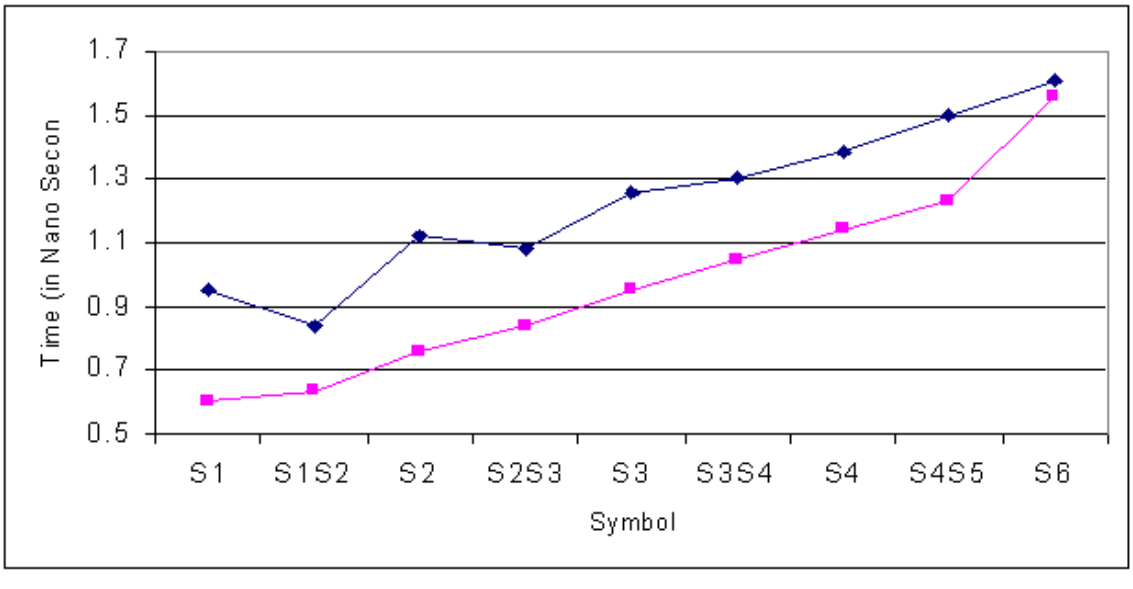


Figure 6.29: Comparison of Simulated Delays with Estimated Parasitics and High Frequency Measurements

be $105ps$. The additional delays can be accounted for by carefully estimation of the parasatic capacitances associated with the signal lines connecting the reconfigurable delay lines illustrated in Figure 6.30 and the parasitic capacitances associated with the signal lines connecting the delay cells to the multiplexer in the reconfigurable delay lines illustrated in Figure 6.31. First the lateral capacitance between the two differential signal lines connecting the output of a reconfigurable delay line to the next reconfigurable delay line in the oscillator was measured. Also the capacitance between the differential signal lines and the substrate was estimated. The capacitance was dominated by the lateral capacitance between the differential lines which was not accounted for during design. Also the lateral capacitance as well as the substrate capacitance was estimated for the signal lines connecting the outputs of the delay cells to the inputs of the multiplexer. Again the total capacitance was dominated by the lateral capacitance which was not account for in design. The oscillator was resimulated with these additional parasitic capacitances. The simulated and measured symbol sizes are plotted in Figure 6.29. It must be noted that the slopes of the two curves are in excellent agreement with one another. The size of the smallest symbol is a bit more than the simulated size suggesting that the capacitance was underestimated. Thirdly, the time differences do not increase linearly. It must be noted that in our simulation we have assumed that the parasitic capacitances are constant. This is true for the substrate capacitance. But the total parasitic capacitance is dominated by the lateral capacitances. In the reconfigurable delay line, the lateral capacitance is between adjacent signal carrying lines. Thus the capacitance is not constant but depends on how the signal in the adjacent lines changes with respect to the signal itself. If the signal in the adjacent signal line were identical to the signal in question, then the effective capacitance would be zero. On the other hand if the signal in the adjacent signal line is an inverted version of the signal in question, the effective capacitance would be twice the estimated value. We hypothesize that this effect results in the observed non-linearity. If the circuit were to be redesigned the layout needs to be modified taking into account lateral capacitances which have substantially affected the design discussed in this thesis.

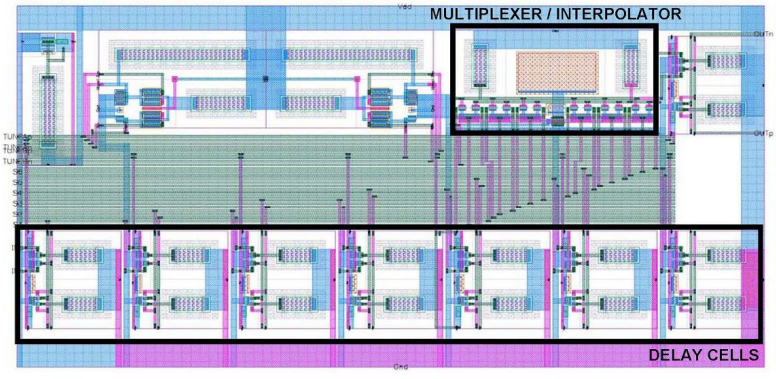


Figure 6.30: Layout of Reconfigurable Delay Lines

Figure 6.31: Transmitter Layout

It must be noted that the receiver is not affected by the corresponding parasitic effects. The high frequency data signal drives a twisted ring counter. The outputs of the twisted ring counter serve as the start and stop inputs to four independent picosecond time digitizers. The effect of parasitic capacitances is to delay both the start and stop signal but the time between these signals is not affected by the parasitic capacitance. Also notice that in the time digitizers the delay cells connect to six buffers as illustrated in Figure 6.32. The length of the signal lines is minimal when compared to the length of the corresponding lines in the reconfigurable delay line. Also the lateral capacitance are minimal due to spacing. The receiver was simulated with the extracted parasitics and it was found to function as designed. Thus it is designed to receive pulses with widths ranging from $100ps$ to $250ps$. Thus regardless of the symbol transmitted by the transmitter, the receiver will always detect a blank. When the receiver detects a blank it resets the internal pseudo random data sources preventing them from cycling. Thus no output is observed on RECTRIG as shown in Figure 6.26, Channel 2.

An alternate method to test the receiver is to supply an external data signal. The circuit to do this is shown in Figure 6.33. It must be noted that the resistance between the voltage sources used to bias the circuit and the termination resistors is 50Ω and the voltage drop across these resistors is $2.4V$. The DC current through the termination resistor is $48mA$. The DC current could be reduced to $0mA$ by connecting the termination resistors to $2.4V$. The widths of the metal lines connecting the pads to the termination resistors and the input of the buffer and their current carrying capacity is tabulated in the table below. From the table it is evident that there exists a problem of electro migration when the chip is connected to an external data source and this severely impacts reliability which prevents exhaustive testing of the receiver circuit using an external data source.

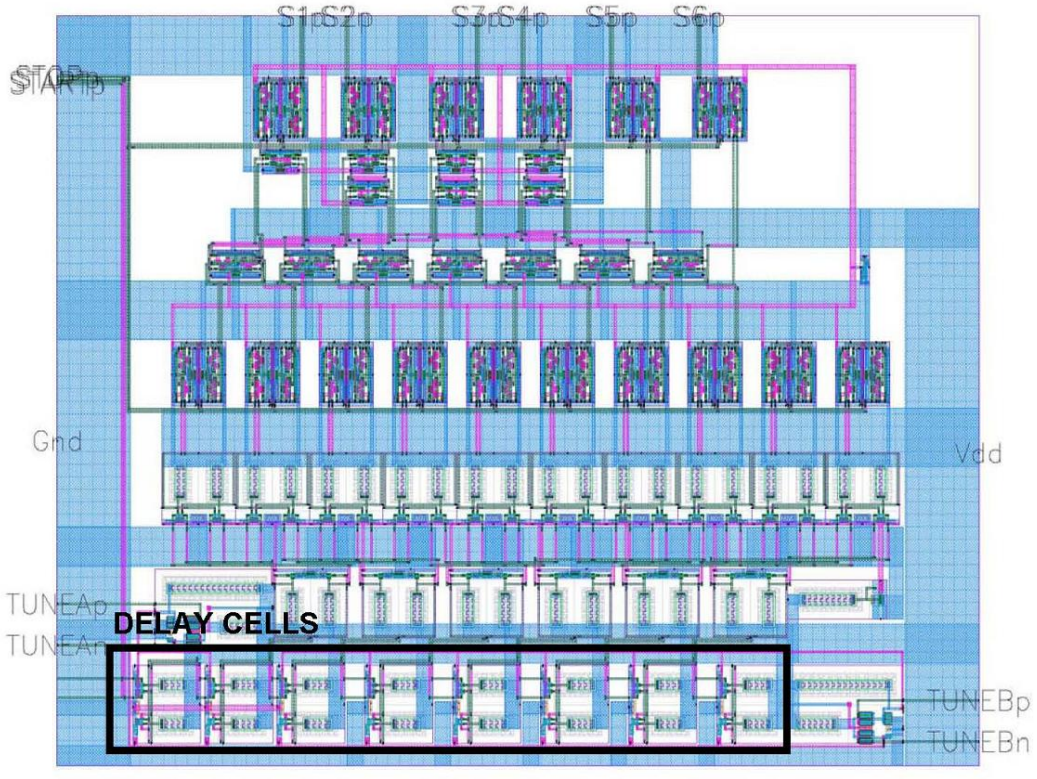


Figure 6.32: Layout of Time Digitizers

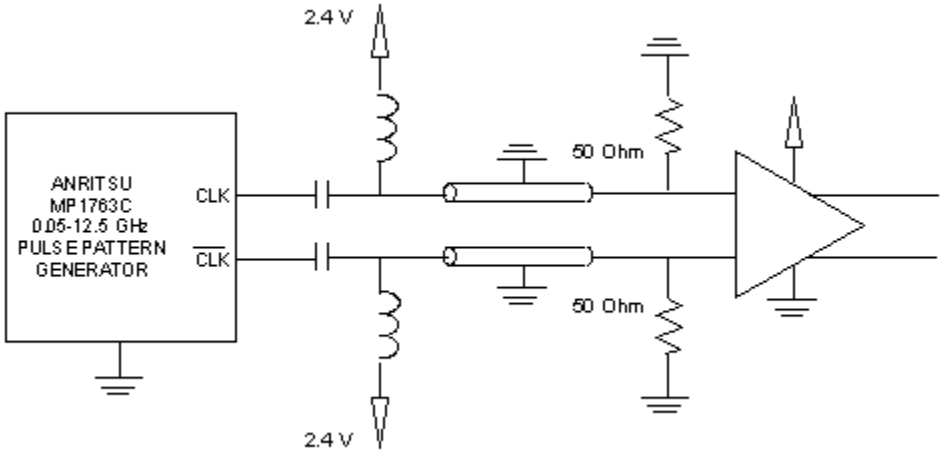


Figure 6.33: Circuit for Supplying an External Input to the Receiver

Metal Layer	Current Capacity	Width	Current Capacity	Actual Current
M1	$0.69(W - 0.0675)$	$20u$	$13.75mA$	$96mA$
M2	$0.69(W - 0.06)$	$15u$	$10.31mA$	$48mA$
M3	$0.69(W - 0.06)$	$15u$	$10.31mA$	$48mA$
M4	$0.69(W - 0.06)$	–	–	–
MT	$0.69(W - 0.06)$	–	–	–
LY	$0.84(W - 0.06)$	$55u$	$46.15mA$	$96mA$
AM	$2.59(W - 0.06)$	$50u$	$129.34mA$	$96mA$

Some preliminary measurements were performed on the receiver. The receiver was configured to receive data from the external data source. The error units were programmed to compare the received data against the smallest symbol. That is if the pulse width corresponded to the smallest symbol then the ERR output would not toggle. If however, the received symbols were larger the ERR output would toggle every time a symbol was received. Thus both the REC and ERR signal would have the same frequency. Initially a sequence of pulses having a width of $113.6ps$ was supplied and no output was observed on the ERR signal. The widths of the pulses was increased to $116.3ps$ causing the ERR signal to oscillate as the same frequency as the REC signal indicating the transition boundary between the first and the second symbol is between $113.6ps$ and $116.3ps$. The results are depicted in Figure 6.34. The REC signal is on Channel 1 and the ERR signal is plotted on Channel 4.

6.6 Summary

A new transceiver architecture was outlined. In this transceiver architecture, the transmitter is based on a reconfigurable ring oscillator and the receiver is based on picosecond time digitizers. Unlike conventional transceivers, the circuits are fully asynchronous and do not require clock recovery for sampling the data signal. Another fundamental advantage of the architecture is that it naturally lends itself to time division multiplexing, as a result only the controller in the transmitter and the twisted ring counter in the receiver need to be designed to operate at full data rate. Both

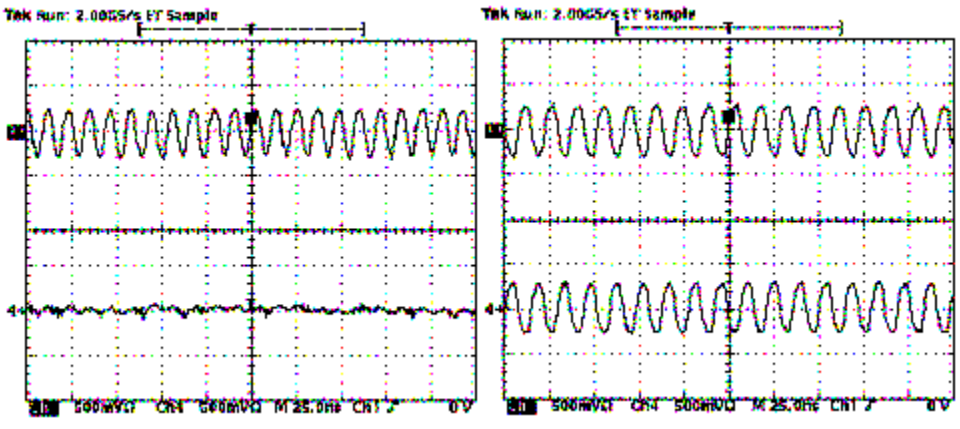


Figure 6.34: Low Frequency Receiver Measurements

these circuits are relatively simple to implement as they can be built out of two input exclusive-or gates and two-to-one multiplexers.

A first prototype was designed, implemented and fabricated in $0.18\mu m$ IBM BiCMOS 7HP process technology. A methodology for testing high speed asynchronous transceivers with off-the-shelf low frequency measurement equipment was presented. The fabricated prototype was extensively tested. Tests included both low and high frequency measurements, both of which were in excellent agreement with each other. It was shown that the performance of the transmitter was substantially mitigated by lateral electrical and magnetic cross-coupling of metal lines connecting the taps on the delay lines to the inputs of the multiplexer in the reconfigurable delay lines. Due to differences in design, the time digitizers were not affected by these parasitic effects. Thus the transmitter and receiver operated at two very different data rates, preventing us from exhaustively characterizing the performance of the receiver. Furthermore, when the receiver was supplied an external high frequency data signal, it did not function reliably due to an effect caused by electro migration. We briefly discussed how these problems could be addressed if the circuit were to be redesigned.

Chapter 7

Conclusions and Open Problems

The contribution of this thesis was a new modulation format suitable for high speed fiber optic communication where one of the limiting factors towards achieving higher data rates is high speed circuit design. In order to utilize this modulation format to transmit data in a rate efficient manner we showed that there is a need for new coding techniques. We proposed the use of variable length to variable length prefix free codes for rate efficient transmission of information using the proposed modulation format. We showed that the code construction problem can be reduced to a large scale integer optimization problem that is well structured. We studied properties of the system of inequalities that arise in this context and based on these properties we derived efficient algorithms for determining if the linear programming relaxation of the code construction problem is feasible. We also derived an efficient algorithm for solving the linear programming solution, if it is indeed feasible. In this thesis we have also proposed an alternate architecture at the circuit level for high speed transceiver design. Unlike conventional transceivers which are based on the idea of clock and data recovery we proposed a fully asynchronous, clock-less circuit that can be implemented at high speeds. The transmitter is based on the idea of a reconfigurable ring oscillator. The receiver is based on the concept of multiplexed picosecond time digitizers. A first prototype of the proposed circuit was implemented in IBM BiCMOS 7HP process technology and measurement results are discussed.

Before the proposed paradigm has an impact on how real world transceivers are designed numerous problems need to be addressed. The problem of solving the integer

program that arises in the context of code construction in polynomial time is unsolved. Also coding techniques based on variable length to variable length prefix free codes suffer from error propagation. Thus there is a need for self synchronizing codes and algorithms for constructing them. Furthermore, it might be possible to reduce the complexity of the codes by studying alternate data structures for coding like finite state machines or codes that unlike prefix free codes are not instantaneously decodable. It is not yet known if the complexity of these codes can be reduced by studying such data structures. Furthermore, systematic techniques for developing codes based on such data structures are not known. From a circuits standpoint, the first prototype was not fully functional and a new prototype needs to be developed which addresses some of the design issues that were discussed in an earlier chapter. Also a comparative study of transceivers based on active delay lines, passive delay lines and microstrips needs to be undertaken. And alternate techniques for delay generation such as integrate and dump, both in the context of the transmitter and receiver need to be explored and their effect on measurement error fully quantified. The paradigm could also benefit from circuit techniques for generation and detection of solitons. From a theoretical standpoint, the topic of generation of data dependent jitter and phase noise in reconfigurable ring oscillators needs to analytically examined. From a communication systems standpoint, the effect of noise in amplitude of the data signal and the effects of the channel like limited bandwidth and chromatic, polarization, multimode, waveguide and material dispersion on data dependent jitter, ISI and error rate need to quantified. Also a comparative study of various modulation formats in comparison to the one proposed needs to be undertaken.

In this thesis we studied the transmission of digital information within a biologically inspired framework. Specifically, we proposed a modulation format that was inspired by neural spike trains and studied the problem of efficient encoding of digital data and the design of high speed digital transceivers. The transmission of analog information through optical fiber is also challenging and of great commercial significance. Schemes which encode analog information in the amplitude of the signal suffer from distortion due to nonlinearities in the photo diode and photo detector. A variety

of modulation formats have been proposed but the transmission of analog information within a similar framework has not attracted much attention even though there is arguably a closer connection with the transmission of information between neurons and neural coding and there has been extensive research on these topics. Both the rate efficient encoding of analog information and design of high speed circuits for analog data transmission are exciting avenues for future work.

Bibliography

[1] R. L. Adler, D. Coppersmith and M. Hassner, "Algorithms for Sliding Block Codes – an Application of Symbolic Dynamics to Information Theory", IEEE Transactions on Information Theory, vol. 29, pp. 5-22, 1983.

[2] D. Altenkamp and K. Mehlhorn, "Codes: Unequal Probabilities, Unequal Letter Costs", J. ACM, vol. 27, no. 3, pp. 412-427, July 1980.

[3] B. Analui and A. Hajimiri, "Statistical Analysis of Integrated Passive Delay Lines", IEEE Custom Integrated Circuits Conference, pp. 107-110, Sep. 2003.

[4] L. I. Anderson, B. G. R. Rudberg, P. T. Lewin, M. D. Reed, S. M. Planer and S. L. Sundaram, "Silicon bipolar chipset for SONET/SDH 10-Gb/s Fiber Optic Communication Links", IEEE J. Solid State Circuits, vol. 30, pp. 210-218, Mar. 1995.

[5] E. Arikan, "An Implementation of Elias Coding for Input-Restricted Channel", IEEE Transactions on Information Theory, vol. 36, pp. 162-165, 1990.

[6] ATM Forum, ATM User-Network Interface Specification. Version 3.0, 1993.

[7] M. S. Bazaraa, J. Jarvis and H. D. Sherali, "Linear Programming and Network Flows", John Wiley & Sons, Inc., 1977, 1990.

[8] B. Beggs, "GaAs HBT 10-Gb/s Product", IEEE MTT-S International Microwave Symposium Workshop, Anaheim, CA, June 13-19, 1999.

[9] T. Berger and R. W. Yeung, "Optimum "1"-Ended Binary Prefix Codes", IEEE Transactions on Information Theory, vol. 36, no. 6, pp. 1435-1441, 1990.

[10] M. C. Berry, "Pulse Width Modulation for Optical Fiber Transmission", PhD. Thesis, Nottingham University, England. 1983.

[11] M. C. Berry and J. M. Arnold, "Pulse Width Modulation for Optical Fiber Transmission of Video", IEE International Conference on the Impact of VLSI Technology on Communication Systems, London, 1983.

[12] N. M. Blachman, "Minimum-Cost Encoding of Information", IRE Trans. Inform. Theory, vol. PGIT-3, pp. 139-149, 1954.

[13] M. Blaum and J. Bruck, "Coding for Skew Tolerant Parallel Asynchronous Communications", IEEE Transactions on Information Theory, vol. 39, no. 2, pp. 379-388, March 1993.

[14] M. Blaum, J. Bruck and L.H. Khachatrian, "Constructions of Skew-Tolerant and Skew-Detecting Codes", IEEE Transactions on Information Theory, vol. 39, no. 5, pp. 1752-1757, September 1993.

[15] M. Blaum and J. Bruck, "Coding for Delay-Insensitive Communication with Partial Synchronization", IEEE Transactions on Information Theory, vol. 40, no. 3, pp. 941-945, May 1994.

[16] D. J. Browning and J. B. Thomas, "Optimal Coding Schemes for Conflict-Free Channel Access", IEEE Trans. Commun., vol. 37, pp. 1004-1013, Oct. 1989.

[17] R. M. Capocelli and P. Cull, "Generalized Fibonacci Numbers are Rounded Powers", Third International Conference on Fibonacci Numbers and Their Applications, Pisa, Italy, pp. 57-62, 1988.

[18] R. M. Capocelli, A. De Santis, L. Gargano, and U. Vaccaro, "On the Structure of Statistically Synchronizable Codes", IEEE Transactions on Information Theory, vol. 38, pp. 407-414, Mar. 1992.

[19] R. M. Capocelli, A. De Santis and G. Persiano, "Efficient Algorithms for Conflict Free Channel Access", Tech. Rep., Univ. Salerno, July 1992.

[20] R. M. Capocelli, A. De Santis and G. Persiano, "Binary Prefix Codes Ending in a "1"", IEEE Transactions on Information Theory, vol. 40, no. 4, pp. 1296-1302, July 1994.

[21] D. M. Choy and C. K. Wong, "Bounds for Optimal α-β Binary Trees", BIT, vol. 17, pp. 1-15, 1977.

[22] E. Christian and E. Eisenmann, "Filter Design Tables and Graphs ", John Wiley and Sons, Inc., New York, 1966.

[23] D. Cooke, Z. Jelonek, A. J. Oxford and E. Fitch, "Pulse Communication", J. IEE, 94, Part IIIA, pp. 83-105, 1947.

[24] N. Cot, "A Linear-Time Ordering Procedure with Applications to Variable Length Encoding", Proc. 8th Princeton Conf. on Information Sciences and Systems, pp. 460-467, 1974.

[25] N. Cot, "Complexity of the variable length encoding problem", Proc. 6th Southeast Conference on Combinatorics, Graph Theory and Computing, Congressus Numerantium XIV, Utilitas Mathematica Publishing, Winnepeg, MB, Canada, pp. 211-244, 1975.

[26] N. Cot, "Characterization and Design of Optimal Prefix Codes", Ph.D. Dissertation, Stanford University, Stanford, CA., 1977.

[27] R. M. Fano, Res. Lab., for Electronics, Massachusetts Institute of Technology, Cambridge, Tech. Report no. 65, 1949.

[28] L. Farina and S. Rinaldi, "Positive Linear Systems – Theory and Applications", John Wiley & Sons, 2000.

[29] E. Fitch, "The Spectrum of Modulated Pulses", J. IEE, 94, Part IIIA, pp. 556-564, 1947.

[30] P. A. Franaszek, "Sequence-state Coding for Digital Transmission", Inform. Contr., vol. 1-J, pp. 155-164, 1969.

[31] P. Funk, "Run-length-limited Codes with Multiple Spacing", IEEE Transactions on Magnetics, vol. 18, pp. 772-775, 1982.

[32] F. R. Gantmacher, "Applications of the Theory of Matrices", Interscience Publishers, 1959.

[33] P. R. Geffe, "Simplified Modern Filter Design ", J. F. Rider Publisher Inc., New York, 1982.

[34] Gigabit Ethernet Alliance, "Gigabit Ethernet: Accelerating the Standard for Speed", Whitepapers, Mar. 1999.

[35] E. N. Gilbert, "Coding with Digits of Unequal Cost", IEEE Transactions on Information Theory, vol. 41, no. 2, pp. 596-600, 1995.

[36] M. J. Golin and N. Young, "Prefix Codes: Equiprobable Words, Unequal Letter Costs", SIAM J. Comput., vol. 25, no. 6, pp. 1281-1292, 1996.

[37] M. J. Golin and G. Rote, "A Dynamic Programming Algorithm for Constructing Optimal Prefix-Free Codes with Unequal Letter Costs", IEEE Transactions on Information Theory, vol. 44, no. 5, pp. 1770-1781, 1998.

[38] S. W. Golomb, L. R. Welch, R. M. Goldstein and A. W. Hales, "Shift Register Sequences", Aegean Park Press, 1982.

[39] R. E. Gomory, "Outline of an Algorithm for Integer Solutions to Linear Problems", Bull. Am. Math. Soc., vol. 64, pp. 275-278, Sep. 1958.

[40] R. E. Gomory, "An Algorithm for Integer Solutions to Linear Programs", Princeton IBM Math Research Project, Tech. Rept, Rept. No. 1, 1958.

[41] R. E. Gomory, "All-Integer Integer Programming Algorithm", IBM Res. Center, Yorktown Heights, N.Y., Res. Rep. RC-189, 1960.

[42] Y. M. Greshishchev and P. Schvan, "SiGe Clock and Data Recovery IC with Linear-Type PLL for 10-Gb/s SONET Application", Bipolar/BiCMOS Circuits and Technology Meeting, pp. 169-172, Sep. 1999.

[43] Y. M. Greshishchev, P. Schvan, J. L. Showell, M. Xu, J. J. Ojha and J. E. Rogers, "A Fully Integrated SiGe Receiver IC for 10-Gb/s Data Rate", ISSCC Dig. Tech. Papers, pp. 52-53, Feb. 2000.

[44] Y. M. Greshishchev, P. Schvan, J. L. Showell, M. Xu, J. J. Ojha and J. E. Rogers, "A Fully Integrated SiGe Receiver IC for 10-Gb/s Data Rate", IEEE Journal of Solid State Circuits, vol. 35, no. 12, pp. 1949-1957, Dec. 2000.

[45] J. Hauenschild et. al., "A Plastic Packaged 10-Gb/s BiCMOS Clock and Data Recovering 1:4 Demultiplexer with External VCO", IEEE J. Solid State Circuits, vol. 31, pp. 2056-2059, Dec. 1996.

[46] D. J. T. Heatley, "Unrepeatered Video Transmission using Pulse Frequency Modulation over 100 km of Monomode Optical Fiber", Electron. Lett., 18, pp. 369-371, 1982.

[47] D. J. T. Heatley, "Video Transmission in Optical Fiber Local Networks using Pulse Time Modulation", 9th European Conference on Optical Communication (ECOC), Geneva, pp. 343-346, Sep. 1983.

[48] D. J. T. Heatley and T. G. Hodgkinson, "Video Transmission over Cabled Monomode Fiber at 1.5μm using PFM with 2-PSK Heterodyne Detection", Electron. Lett., 20, pp. 110-112, 1984.

[49] C. D. Heegard, B. H. Marcus and P. H. Siegel, "Variable Length State Splitting with Applications to Average Runlength-constrained (ARC) Codes", IEEE Transactions on Information Theory, vol. 37, pp. 759-777, 1991.

[50] S. F. Heker, G. J. Herkovitz, H. Grebel and H. Wichansky, "Video Transmission in Optical Fiber Communication Systems using Pulse Frequency Modulation", IEEE Trans. Commun., 36 (2), pp. 191-194, 1988.

[51] D. H. Huffman, "A Method for the Construction of Minimum Redundancy Codes ", Proceedings of the IRE, vol. 40, no. 9, pp. 1098-1101, 1952.

[52] W. H. Huggins, "Network Approximation in the Time Domain ", Rep E 5048A, Air Force Cambridge Research Labs, Cambridge, Mass., Oct. 1949.

[53] K. A. S. Immink, "Codes for Mass Digital Storage", Shannon Foundation Publishers, 1999.

[54] Z. Jelonek, "Noise Problems in Pulse Communications", J. IEE, 94, Part IIIA, pp. 533-545, 1947.

[55] T. Kanada, K. Hakoda and E. Yoneda, "SNR Fluctuations and Non-Linear Distortion in PFM Optical NTSC Video Transmission Systems", IEEE Trans. COM-30 (8), pp. 1868-1875, 1982.

[56] S. Kapoor and E. M. Reingold, "Optimum Lopsided Binary Trees", J. ACM, vol. 36, pp. 573-590, 1989.

[57] R. M. Karp, "Minimum Redundancy Coding for the Discrete Noiseless Channel", IRE Trans. Inf. Theory", vol. 7, no. 1, pp. 27-38, 1961.

[58] W. H. Kautz, "Fibonacci Codes for Synchronization Control", IEEE Transactions on Information Theory, pp. 284-292, April 1965.

[59] K. J. Kerpez, "Runlength Codes from Source Codes ", IEEE Transactions on Information Theory, vol. 37, pp. 682-687, 1991.

[60] Z. Kiyasu, "On a Design Method of Delay Networks ", J. Inst. Elec. Comm. Eng. Japan, vol. 26, pp. 598-610, 1943.

[61] D. E. Knuth, "Efficient Balanced Codes ", IEEE Transactions on Information Theory, vol. 32, pp. 51-53, 1986.

[62] R. M. Krause, "Channels which transmit letters of unequal duration ", Inf. Control, vol. 5, no.1, pp. 13-24, 1962.

[63] D. A. Lelewer and D. S. Hirschberg, "Data Compression ", ACM Computing Surveys, vol. 19, no. 3, pp. 261-296, 1987.

[64] M. M. Levy, "Some Theoretical and Practical Considerations of Pulse Modulation ", J. IEE, 94, Part IIIA, pp. 565-572, 1947.

[65] D. Lind and B. Marcus, "An Introduction to Symbolic Dynamics and Coding ", Cambridge University Press, 1985.

[66] C. Lu, "Optical Transmission of Wideband Video Signals using SWFM ", PhD. Thesis, University of Manchester Institute of Science and Technology, Manchester, England, 1990.

[67] B. H. Marcus, P. H. Siegel and J. K. Wolf, "Finite-state Modulation Codes for Data Storage ", IEEE J. Sel. Areas Comm., vol. 10, pp. 5-37, 1992.

[68] B. H. Marcus, P. H. Siegel and J. K. Wolf, "Codes with a Multiple Spectral Null at Zero Frequency ", IEEE Transactions on Information Theory, vol. 35, pp. 463-472, 1989.

[69] B. H. Marcus, R. Roth and P. H. Siegel, "Handbook of Coding Theory ", Elsevier Press, 1998.

[70] R. S. Marcus, "Discrete Noiseless Coding ", M. S. Thesis, MIT, Electrical Engineering Dept., 1957.

[71] R. K. Martin, "Large Scale Linear and Integer Optimization ", Kluwer Academic Publishers, 1999.

[72] K. Mehlhorn, "An Efficient Algorithm for Constructing Nearly Optimal Prefix Codes ", IEEE Trans. Inf. Theory, vol. 26, no. 5, pp. 513-517, 1980.

[73] T. Morikawa et. al., "A SiGe single-chip 3.3V receiver IC for 10-Gb/s Optical Communication Systems", ISSCC Dig. Tech. Papers, pp. 380-381, Feb. 1999.

[74] S. Mukhtar and J. Bruck, "Frequency Modulation for Asynchronous Data Transfer", Electronic Technology Report, ETR036, April, 2001.

[75] K. Murty, "Linear Programming", John Wiley & Sons, Inc., 1983.

[76] A. Okazaki, "Still Picture Transmission by Pulse Interval Modulation", IEEE Trans. CATV-4, pp. 17-22, 1979.

[77] A. Okazaki, "Pulse Interval Modulation Applicable to Narrowband Transmission", IEEE Trans. CATV-3, pp. 155-164, 1978.

[78] Y. Perl, M. R. Garey and S. Even, "Efficient Generation of Optimal Prefix Code: Equiprobable Words using Unequal Cost Letters", J. ACM, vol. 22, no. 2, pp. 202-214, 1975.

[79] L. Pophillat, "Video Transmission using a 1.3 μm LED and Monomode Fiber", 10th European Conference on Optical Communications, Stuttgart, West Germany, pp. 238-239, 1984.

[80] D. M. Pozar, "Microwave Engineering ", John Wiley and Sons, Inc., New York, 1998.

[81] R. Saal, "Handbook of Filter Design ", AEG-Telefunken, Berlin, West Germany, 1979.

[82] K. Sato, S. Aoygai and T. Kitami, "Fiber Optic Video Transmission Employing Square Wave Frequency Modulation", IEEE Trans. COMM-33 (5), pp. 417-423, 1985.

[83] M. Sato, M. Murata and T. Namekawa, "Pulse Interval and Width Modulation for Video Transmission", IEEE Trans. CATV-3 (4), pp. 166-173, 1978.

[84] M. Sato, M. Murata and T. Namekawa, "A New Optical System Communication System using the Pulse Interval and Width Modulated Code", IEEE Trans. CATV-4 (1), pp. 1-9, 1979.

[85] A. Schrijver, "Theory of Linear and Integer Programming", John Wiley & Sons, Inc., 1986.

[86] C. B. Schrocks, "Proposal for a Hub Controlled Cable Television System using Optical Fiber", IEEE Trans., CATV-4, pp. 70-79, 1979.

[87] C. E. Shannon, "A Mathematical Theory of Communication", Bell. Sys. Tech. J., vol. 27, pp. 379-423, 623-656, July-October 1948.

[88] M. Soda, T. Suzaki and T. Morikawa, "A Si bipolar chip set for 10-Gb/s Optical Receiver", ISSCC Dig. Tech. Papers, pp.100-101, Feb. 1992.

[89] SONET OC-192, "Transport System Generic Criteria", Bellcore, GR-1377-CORE, no. 4, Mar. 1998.

[90] E. Sperner, "Ein Satz uber Untermengen einer endlichen Menge", Math. Z. vol. 27, pp. 544-548, 1928.

[91] W. R. Spickerman, "Binet's Formula for the Tribonacci Sequence", The Fibonacci Quarterly, no. 2, pp. 118-120, May 1982.

[92] W. R. Spickerman and R. N. Joyner, "Binet's Formula for the Recursive Sequence of Order K", The Fibonacci Quarterly, no. 4, pp. 327-331, 1984.

[93] L. E. Stanfel, "Tree Structure for Optimal Searching", J. ACM, vol. 17, pp. 508-517, 1970.

[94] L. Storch, "Synthesis of Constant-Time Delay Ladder Networks using Bessel Polynomials", Proc. IRE, vol. 42, pp. 1666-1675, 1954.

[95] G. Strang, "Introduction to Linear Algebra", Wellesley-Cambridge Press, 1993.

[96] S. Y. Suh, "Pulse Width Modulation for Analog Fiber Optic Communications", IEEE J., LT-5 (1), pp. 102-112, 1987.

[97] W. E. Thomson, "Delay Networks having Maximally Flat Frequency Characteristics", Proc. IEE., pt. 3, vol. 96, pp. 487-490, 1949.

[98] B. P. Tunstall, "Synthesis of Noiseless Compression Codes", Thesis Georgia Institute of Technology, 1967.

[99] B. Varn, "Optimal Variable Length Codes (Arbitrary Symbol Cost and Equal Codeword Probability)", Inf. Control. vol. 19, no. 4, pp. 289-301, 1971.

[100] T. Verhoeff, "Delay-insensitive codes – an overview", Distributed Computing, pp. 3:1-8, 1988.

[101] R. C. Walker et. al., "A 10-Gb/s Si-bipolar Tx/Rx Chipset for Computer Data Transmission", ISSCC Dig. Tech. Papers, pp. 302-303, Feb. 1998.

[102] A. X. Widmer and P. A. Franaszek, "A DC-Balanced, Partitioned Block, 8B/10B Transmission Code", IBM J. Res. Develop., vol. 27, no. 5, pp. 440-451, Sep. 1983.

[103] A. B. Williams, "Electronic Filter Design Handbook ", McGraw-Hill Book Co., New York, 1981.

[104] B. Wilson and Z. Ghassemlooy, "Optical Pulse Width Modulation for Electrically Isolated Analogue Transmission", J. Phys. (E), 18, pp. 954-958, 1985.

[105] B. Wilson and Z. Ghassemlooy, "Optical PWM Data Link for High Quality Analogue and Video Signals", J. Phys. (E), 20 (7), pp. 841-845, 1987.

[106] B. Wilson and Z. Ghassemlooy, "Optical Fiber Transmission of Multiplexed Video Signals using PWM", Int. J. Optoelectronics, 4, pp. 3-17, 1989.

[107] B. Wilson, Z. Ghassemlooy, I. Darwazeh, C. Lu and D. Chan, "Optical Squarewave Frequency Modulation for Wideband Instrumentation and Video Signals", IEE Colloquium on Analogue Optical Communications, London, Digest 1989, 165, Paper 9, 1989.

[108] B. Wilson, Z. Ghassemlooy and C. Lu, "Squarewave FM Optical Fiber Transmission for High Definition Television Signals", Proc. Int. Soc. Optical Eng., 1314, pp. 90-97, 1990.

[109] B. Wilson, Z. Ghassemlooy and C. Lu, "Optical Fiber Transmission of High-Definition Television Signals using Squarewave Frequency Modulation", Third Bangor Symposium on Communications, University of Wales, Bangor, pp. 258-262, May 1991.

[110] B. Wilson, Z. Ghassemlooy and J. C. S. Cheung, "Spectral Predictions for Pulse Interval and Width Modulation", Electron. Lett., 27 (7), pp. 580-581, 1991.

[111] B. Wilson and Z. Ghassemlooy, "Pulse Time Modulation Techniques for Optical Communications: A Review", IEE Proceedings-J., vol. 140, no. 6, pp. 346-357, Dec. 1993.

[112] X3T9.3 Task Group of ANSI, "Fiber Channel Physical and Signalling Interface (FC-PH)", rev. 4.2, Oct. 9, 1993.

[113] M. Zargari, "A BiCMOS Active Substrate Probe Card Technology for Digital Testing", Technical Report No. ICL97-070, Integrated Circuits Laboratory, Stanford University, Mar. 1997.