THE DESIGN AND IMPLEMENTATION OF A CHIPSET FOR GIGABIT/SECOND COMPUTER NETWORKS

A Thesis

Presented

to the Faculty of

California State University, Chico

In Partial Fulfillment

of the Requirements for the Degree

Master of Science

in

Computer Science

by

Summer 1992

THE DESIGN AND IMPLEMENTATION OF A CHIPSET FOR GIGABIT/SECOND COMPUTER NETWORKS

A Thesis

by

Richard C. Walker

Summer 1992

APPROVED BY THE DEAN OF THE GRADUATE SCHOOL:

Elaine G. Wangberg, Ph.D.

APPROVED BY THE GRADUATE ADVISORY COMMITTEE:

James L. Murphy, Ph.D., Chair

Seung Bae Im, Ph.D.

PUBLICATION RIGHTS

No portion of this thesis may be reprinted or reproduced in any manner unacceptable to the usual copyright restrictions without the written permission of the author.

Signed ______________________________

ACKNOWLEDGMENTS

I would like to thank Tom Hornak for his many contributions to this project, and for his guidance and inspiration throughout its development. Tom’s sharp intellect and clear thinking has helped me to avoid many pitfalls along the way. I also thank Chu Yen, my project manager during most of this effort. He has consistently been supportive of my academic endeavors, and his advice, guidance, and gentle prodding have been appreciated. I wish to thank the members of the G-Link design team: Cheryl Stout, Benny Lai, and Jieh-Tsorng Wu and Pat Petruno. These folks jumped right in, quickly absorbed the essence of the previous research, and did a first-class job of making the “G-Link” a commercial reality.

The G-Link chipset’s success in the Gigabit-rate computer standards arena has been due to the efforts of Bill McFarland, Hans Wiggers, Pat Petruno, David Cunningham, Steve Methley, and David Sears. I’ve especially appreciated the good discussions with Bill McFarland, and his advice and feedback about the G-Link design.

Packaging and assembly of the chipset were done by Jean Norman and Natalia McAfee for the early prototypes, and David Yoo and Lie Lian-Mueller for the commercial version. I would like to thank Don Pettengill, Shang-Yi Chiang, and the staff of the HP25 bipolar process team, without whose excellent work and cooperation, this research could not have been done. I also acknowledge the many early contributions to the “gigabit-link dream” made by Kent Springer, Bill Worley, Rasmus Nordby, Craig Corsetto and Doug Crandall.

Finally, I would like to thank the folks at Chico State University who have really made this whole process a pleasure. I thank Bob Britten for his early advice and encouragement, and also James Murphy and Seung Bae Im for their constructive comments and willingness to serve on my graduate advisory committee.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

FIGURE PAGE

ABSTRACT

THE DESIGN AND IMPLEMENTATION OF A CHIPSET FOR GIGBIT/SECOND COMPUTER NETWORKS

by

© Richard C. Walker 1992

Master of Science in Computer Science

California State University, Chico

Summer 1992

A Silicon bipolar Transmitter and Receiver Chip transfers parallel data across a 1.5 Gigabaud serial link. A new “Conditional-Invert, Master Transition” code and Phase-locked Loop are described and analyzed which provide adjustment-free clock recovery and frame synchronization. The packaged parts require no external components and operate over a range of 700 to 1800 Mbaud using an on-chip VCO. The line code and handshake protocol have been accepted by the Serial-HIPPI implementor’s group for serially transmitting 800 Mb/s HIPPI data, an ANSI standard, and by SCI-FI, an IEEE standard for 1 Gb/s interconnection of cooperating computer systems.

CHAPTER I

GENERAL BACKGROUND

The last decade has seen a tremendous increase in computing power with only modest advances in the bandwidth of the data-links used to interconnect these faster computers. Between 1982 and 1992, the speed of a typical engineering workstation has increased from half a MIPS (Million Instructions per Second) to 100 MIPS, an increase of over two orders-of-magnitude. In that same period of time, computer network bandwidths have gone from Ethernet at 10 Mbit/second to FDDI at 100 Mbit/second: an increase of only one order-of-magnitude. Other factors, such as the wide use of multimedia applications, have also increased the demand for network bandwidth, threatening to create an “I/O bottleneck” for modern computing systems.

Serial links, unlike computer systems, cannot exploit parallelism and must run at proportionally higher rates for each increment in performance. Below about 100 MHz clock rates, traditional PC-Board design techniques can be used to implement link circuitry with collections of “generic” packaged parts; but as link speeds approach the Gigabit/s range, inter-chip timing-skews make it impractical to build low-cost Gigabit links in this way.

Long-haul telephone networks have used Gigabit-rate data links for many years. Exotic technologies such as SAW filters [28], and bulky microwave pulse circuitry [29], are generally used to achieve the highest possible bandwidth and distance. It is common for the retiming circuits of a telecom link to require periodic adjustment by skilled technicians, and for the equipment to be large and bulky. Such systems, easily justified when the cost is amortized over millions of users, have been too costly and complex for computer use.

For high speed fiber-optic links to become widely used in a computer setting, it is necessary to rethink the traditional telecom design techniques from the viewpoint of the computer environment. Key differences between a typical telecom application and a computer application are highlighted in Table 1 below.

Table 1: Telecom Versus Computer Link Needs

Issue	Telecom	Computer
Cost per link	~ $20,000	< $2,000
Link Distance	20-50 km	10 meters -10 km
Data Rate	Maximum Possible	100M - 1.5Gbit/s
Total Size	Rack Mount	< 10 square inches
Total Power	< 200 Watts	< 5 Watts
Maintenance Interval	Monthly	None

As can be seen in Table 1, the computer applications have some strong requirements in terms of cost, size, power and maintenance, but are quite flexible in link distance and data rates.

For gigabit/sec links to become widely used, it is necessary to hide the implementational complexity from the user. In this sense, the chip design should be like an object-oriented software package, exhibiting good information hiding and encouraging easy use. With a single chip implementation, the user only interacts with the link through a well-defined, low-speed parallel interface. All the critical timing of the high-speed clocks, the complexity of physical layer link management, plus the line coding and clock extraction can be effectively hidden from the chip user.

There have been several chipsets built to provide parts of this functionality in the 100 Mbit/sec range [1][2], but none at Gbit/sec rates. Many of the published Gbit/sec prototype links fall into the category of “hero experiments” or lab curiosities that merely demonstrate the feasibility of various ideas, but which require components that are not directly suitable for monolithic IC implementation.

This thesis describes a link interface chipset based on my past research [3][4][5][6], that effectively hides link complexity from the user and is useable in many Gigabit/second network applications.

From the start, the link was designed with computer applications in mind. It provides a 16, 17, 20 or 21 bit parallel interface to the user's bus, and performs the necessary serialization, line coding, clock extraction and decoding. The chipset is known as the HP G-Link, or Gigabit link. I have been the principle researcher and designer on the G-Link project for the past six years.

To provide flexibility for the needs of modern protocols, a new line code [7] was designed which allows control and fill frames to be sent that are distinct from normal data frames. These frames may be used for packet headers and delimiters as well as link maintainance. The advantages of this new code are described and contrasted with other codes commonly used for high-speed links.

An end-to-end state machine is included in the chipset to negotiate the start-up process, so that the physical/data link is able to tell the network layer that a valid connection has been established. In addition, the use of a start-up state machine addresses the issue of laser eye safety, as regulated by international standard [8], in which the physical layer must detect broken fibers so that the laser may be disabled to prevent damaging the eye of a human observer.

A new, patented clock-extraction technique [9][5], and monolithic Voltage Controlled Oscillator (VCO) [4] are implemented in the G-Link Chipset which can operate at higher speeds than many previously proposed circuits. In addition, the phase alignment of the clock-extractor is self calibrated and temperature independant. This thesis includes a review of previous clock extraction techniques and demonstrates the advantages of these new circuits as applied to low-cost monolithic implementation.

In addition to documenting the various design features of the chipset, those parts of the HIPPI, HIPPI-Serial, Fiber Channel, and SCI-FI standards documents are described that help put this work in perspective.

CHAPTER II

GB/SEC STANDARDS ACTIVITIES

Currently, there are four main different groups working on standards for computer network systems that operate at bit rates approaching 1 Gbit/second. At first it may seem curious that there are so many different standards in progress, but a close look at the situation shows that each committee has a different set of goals and requirements.

Nearly all of the different groups have deviated from the OSI model to some extent in an attempt maintain bandwidth and reduce latency. In those areas where sophisticated algorithms are required, every attempt has been made to implement them in hardware rather than software. An amusing anecdote explains why a major computer manufacturer decided against introducing an FDDI card for their workstation. It turns out that even though the raw bandwidth of the FDDI link was 10 times that of the standard Ethernet card, that the effective throughput increase was less than doubled. This common experience shows how badly the software overhead can dominate the system performance unless special care is taken.

Another approach taken by some of the committees is to make the Link look somewhat like a direct bus extender of the main CPU. This breaks with the traditional view of the network interface being a peripheral device.

Yet another watershed in link design is the issue of link latency. Those users interested in fine grain communication for parallel processing consider low latency to be paramount. Sometimes high link speeds are predicated, not by a need for large volumes of data transmission, but merely because it takes less time to send a packet. Usually, such users will not be interested in long link lengths because the time-of-flight starts to dominate the computational efficiency of their system.

The following sub-sections summarize the goals and approaches of the four main standards efforts at the Gigabit/second data rate.

High Performance Parallel Interface

The High Performance Parallel Interface (HIPPI) is a gigabit-rate interconnection standard sponsored by ANSI. As the name High Performance Parallel Interface implies, HIPPI is a parallel oriented interconnect. The physical medium of HIPPI is a fifty-conductor twisted-pair cable which is driven by differential ECL drivers [10][11]. There are two data rates supported, 800 Mbit/s, and 1600 Mbit/s. These two rates correspond to a 32 bit bus or a 64 bit bus at 25 MHz cycle times. The maximum length of the connection is 25 meters.

HIPPI is targeted at applications such as computer video distribution and supercomputer I/O needs. The standard defines a one-way simplex channel but many users plan on using two links back-to-back to provide full duplex operation.

A look ahead flow control mechanism is provided which can allow the pipelining of messages for efficient transmission on links with large propagation delays.

The HIPPI frame consists of packets which are composed of one or more 256 word bursts. In 32 bit mode, a burst is 1024 bytes, and in 64 bit mode a burst is 2048 bytes.

Error detection is provided by horizontal and vertical parity checks. Each byte sent on the link has a parity check bit, and at the end of each frame there is a length/ longitudinal redundancy checkword (LLRC). Because the link is defined to be one-way, no retry or recovery mechanism is proposed by the standard. It is expected that if the user needs better error protection, that he will provide his own detection/correction protocol external to the HIPPI definition.

Fiber Channel

HIPPI was initially implemented as copper-based interconnect due to the immaturity of fiber-optic technology. The long range goal has been to eventually support the HIPPI protocol on a fiber-optic physical layer. The fiber-optic implementation has many advantages such as smaller connector size and better immunity to interference. A decision was made early on in the HIPPI committed to standardize on the ANSI Fiber Channel as the fiber optic physical layer.

ANSI’s Fiber Channel specification is organized into four separate standards: FC0 through FC3. FC0 covers the physical and mechanical details of the interface. FC1 pertains to the transmission protocols, such as encoding and error control. FC2 covers frame structure, addresses and control signals, FC3 defines the mapping between the lower level standards and the logical control signals defined by HIPPI and other standards.

The scope of Fiber Channel is extremely ambitious. Some have called it an “everything but the kitchen sink” standard. It is the intention of Fiber Channel to support the data link layers of HIPPI, Small Computer Systems Interface (SCSI), and the Intelligent Peripherals Interface (IPI) all on top of the same physical definition.

Currently, the Fiber Channel is about 3-5 years from being finalized. Because of this long delay, an ad-hoc committee of HIPPI users was formed to develop a quick, interim Serial-HIPPI definition so that manufacturers could immediately start to produce inter-operable links. This effort is described in the following sub-section.

HIPPI-Serial

HIPPI-Serial is an ad-hoc consortium of users and manufacturers who felt that they could not wait for Fiber

Channel to be finalized. The group very quickly evaluated the currently available gigabit-rate hardware and finalized their agreement in March 1991. Because HIPPI-Serial-compliant hardware is now appearing on the market, there is some possibility that HIPPI-Serial could actually undercut Fiber Channel by becoming a de-facto standard.

The HIPPI-Serial group covered the design of a serial extender for the existing HIPPI 32-bit parallel, 800 Mbit/sec interface. The G-Link chipset and line code was approved as the physical layer for the HIPPI-Serial specification. I wrote and helped edit substantial portions of the HIPPI-Serial implementor’s agreement [12].

The committee considered four different coding proposals for the datalink layer: Digital Equipment Corporation’s 8B/10B code [13] with forward error correction, Gazelle’s 4B/5B code [14], Broadband Communication Product’s Scrambling Method [15], and Hewlett-Packard’s 21B/24B Conditional Invert/Master Transition (CIMT) code [16]. A comparison is made between the 8B/10B and the CIMT code in a later section.

Scalable Coherent Interface

The Scalable Coherent Interface standard [17][18] is an interconnect system that is more like a backplane replacement than a network interface. The goal of the SCI group was to provide a high-speed interconnect between computer systems that will allow fine-grain, low latency data transfers for implementing distributed shared-memory on large multi-processor systems (n <= 64K processors). The protocol keeps a directory of memory pages that are in use within the network to ensure that all memory references are coherent.

There are two versions of SCI: copper based (SCI) and fiber based (SCI-FI).

SCI-copper uses a physical layer of a 17-pair ribbon cable clocked at a 250 MHz data rate. 16 of the 17 signals are used for data, while transitions on the 17th bit are used as frame delimiters. The total data bandwidth is therefore 8 Gbit/second. The network topology is a single uni-directional ring. It was expected that SCI-copper would be used to interconnect parallel processors within a rack and SCI-FI would be used for inter-rack communication.

For the current state of the art, it is impractical to consider directly supporting the 8 Gbit/second data rate of SCI-copper on a fiber link. SCI-FI is therefore run at a lower speed of 1 Gbit/second. I proposed the use of the 17B/20B CIMT code [19], and it was subsequently adopted by the group.

Summary of Major Standards Efforts

Table 2 summarizes the main physical layer standards in the gigabit/sec area. All the proposed fiber standards are in the .8 to 1 Gbit/sec range. This corresponds to a coded baud rate of 1-1.5 Gbaud. The price of lasers grows very sharply when the baud rate starts to exceed 1.5 Gbaud. The standards committees have chosen their data rates based on this fact.

Also of interest is that all of the standards are defined as point-to-point connections. Only the SCI standard goes as far as defining a simplex ring built out of point-to-point links. The other standards leave the topology issue to be settled by higher level protocol needs. The clear message is that broadcast technology has run out of steam at gigabit rates.

Table 2: Summary of Gb/s Standards

Standard	Speed	Media	Coding	Topology	Status
SCI	8 Gbit/s	copper twisted pair	NRZ	Simplex Ring	Final
SCI-FI	1 Gbit/s	Single-mode Fiber	HP 17/20 CIMT code	Simplex Ring	Final
HIPPI	0.8-1.6 Gbit/s	copper twisted pair	NRZ	Simplex Link	Final
HIPPI-Serial	800 Mb/s	Single-mode Fiber	HP 20/24 CIMT code	Duplex Link	Final
Fiber Channel	0.1-1.0 Gbit/s	Multi/ single-mode Fiber	8B/10B	Simplex Link	due by ‘93-‘95

CHAPTER III

OVERVIEW OF CHIP DESIGN

Introduction

Parallel computers, high-resolution graphics and network backbones are among the many applications that could immediately benefit from inexpensive, compact, and easy-to-use gigabit-rate fiber-optic data links. Serial links have been widely used for telecom applications, however, parallel data interfaces are required for convenient connection to computer equipment.

The use of fiber media for gigabit-rate computer communication has been limited by the lack of low cost link interface chips. An earlier 4-chip chipset [6] established the feasibility of several integrable circuit techniques to achieve these data rates, but was difficult to use because of the high-speed chip interconnections and extra support circuitry required.

In this chapter, a monolithic Transmitter (TX) and Receiver (RX) chip pair is described that can be used for the transmission of parallel data, and that requires no external active components. From the user’s viewpoint, this chipset implements a full-duplex “virtual ribbon cable” interface (Figure. 1). For short distance applications, an on-chip

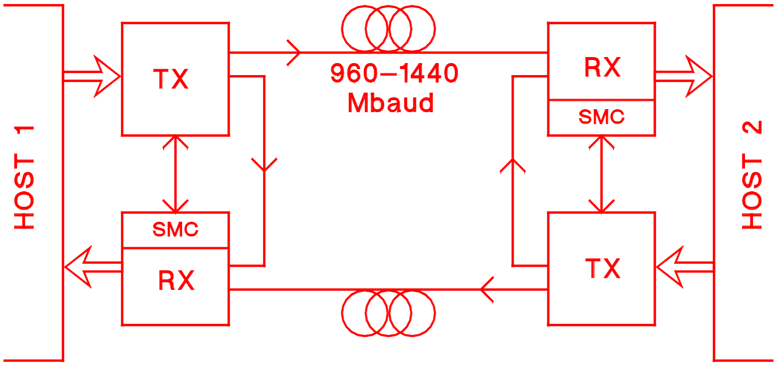

Figure 1. Block Diagram of a Full-duplex Link
Built From 2 Chipset Pairs.

equalizer is provided to allow the use of coaxial cables rather than a more costly fiber link. The chips require no external frequency determining elements or user adjustments and operate over a range of 700 to 1800 MHz using an on-chip VCO. A state machine controller (SMC) is also implemented on the RX chip to transparently handle a start-up handshake protocol. This work is the highest speed link interface chipset reported, to date, at this level of functionality and integration.

The architecture of the link is largely determined by the line code design which is discussed next.

Line Code, Clock and Frame Synchronization

Codes used for fiber-optic links are DC-balanced to permit the regulation of laser bias current by simply maintaining a fixed average optical power. Balanced data streams may also be conveniently AC-coupled at the receiver without incurring extra baseline wander or jitter. The “Conditional Invert, Master Transition” Code (CIMT) used in this chipset transmits the parallel data words in either true or complement form, as needed, to maintain DC-balance on the line.

To make the decision of whether or not to invert a data frame, the TX chip uses a majority gate, built from a DAC-like current summing circuit and comparator, to compute the polarity of the incoming frame. The frame polarity is compared against the sign of an up/down counter which keeps track of the total disparity of transmitted bits. If the two signs agree, the frame is sent inverted. Otherwise it is sent un-inverted.

As shown in Figure 2 four extra coding bits create a coding field (C-Field) which is appended to the data field (D-Field) during transmission. The chipset is programmable to allow the transmission of either 16 or 20 bits of data to produce a 20 or 24 bit line code frame. In addition, a FLAG bit is also available as an extra data bit, thereby increasing the data bits to 17 or 21, or can be internally toggled by the transmitter to allow enhanced receiver frame error detection. Table 3 shows the Data Frame bit definitions for the 20 Bit form of the code. The 17 bit form of the line code has been accepted as the standard code for the IEEE P-1596 Scalable Coherent Interface Group [20], and the 21 bit form

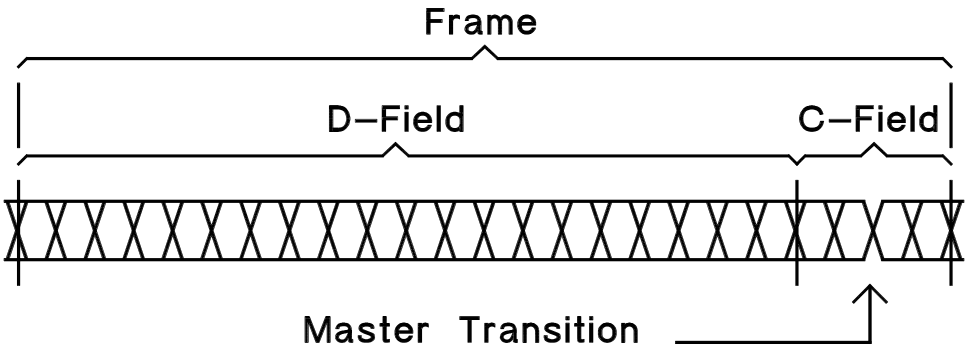

Figure 2. Structure of the CIMT Code

of the code has been accepted by the Ad-Hoc High Performance Parallel Interface (HIPPI) Serial Implementors Group [12].

Table 3: 20 Bit CIMT Data Frame Definition

Data Status	FLAG bit	D-Field	C-Field
True	0	D1-D20	1 1 0 1
Inverted	0	$\overline{\text{D1-D20}}$	0 0 1 0
True	1	D1-D20	1 0 1 1
Inverted	1	$\overline{\text{D1-D20}}$	0 1 0 0

The central pair of bits in the C-Field are always complementary and provide a “master transition” phase reference for the receiver Phase Locked Loop (PLL). This master transition is used by the PLL as the phase reference for both bit and frame clock recovery. The frame clock is used by the demultiplexer for frame alignment. Because each frame of the line code incorporates a reference transition, it is not necessary for the user to send any periodic frame-sync words, as is the case with 4b/5b and 8b/10b codes. This allows the link to be conveniently used in a synchronous environment where the insertion of extra frame-sync words is undesirable.

To implement distinctive packet headers, trailers, etc., it is necessary to support a set of control frames. These are described in Table 4 below. A new set of appended bits are used to distinguish between data and control.

Table 4: 20 Bit CIMT Control Frame Definitions

D-Field			C-Field
D1-D8	0 1	D9-D18	0 0 1 1
D1-D8	1 0	$\overline{\text{D9-D18}}$	1 1 0 0

When the user has no data to send, the link needs to transmit some type of “fill” character to keep the receiver PLL locked and to unambiguously indicate a lack of data. Table 5 gives the set of fill frames that are supported by the CIMT code. Fill Frame 0 is simply a 50% duty-cycle square wave at the frame rate which has its rising edge aligned with the Master Transition location.

During initial link startup, Fill Frame 0 is sent by the transmitter as a training sequence to allow the receiver PLL to acquire frequency and phase lock. The single rising edge is used to unambiguously locate the beginning of the frame.

In the startup of a full duplex link, it is necessary for each side of the link to signal its peer when it has achieved frequency and phase lock. This means that a second type of Fill Frame needs to be available for signalling that can still be used by the receiver for establishing frame lock. Fill Frame 1a and 1b are used for this purpose. Fill Frames 1a and 1b are square waves of slightly unequal duty cycles. They have the property that there is only a single rising edge per frame. To maintain cumulative DC balance, Fill Frame 1a is always used in alternation with Fill Frame 1b.

Table 5: 20 Bit CIMT Fill Frame Encoding

Name	D-Field			C-Field
Fill Frame 0	1 1 1 1 1 1 1 1 1	1 0	0 0 0 0 0 0 0 0 0	0 0 1 1
Fill Frame 1a	1 1 1 1 1 1 1 1 1	1 1	0 0 0 0 0 0 0 0 0	0 0 1 1
Fill Frame 1b	1 1 1 1 1 1 1 1 1	0 0	0 0 0 0 0 0 0 0 0	0 0 1 1

Finally, out of the set of all possible bit patterns, there exist several patterns that are not valid codes. Table 6 gives the disallowed states that can be used for error detection. When the receiver sees one of the bit patterns in Table 6, it flags the frame as being in error before passing it along to the user.

Table 6: 20 Bit CIMT Disallowed States

D-Field			C-Field
X X X X X X X X X	X X	X X X X X X X X X	X 0 0 X
X X X X X X X X X	X X	X X X X X X X X X	X 1 1 X
X X X X X X X X X	0 X	X X X X X X X X X	1 1 0 0
X X X X X X X X X	1 1	X X X X X X X X X	1 1 0 0
X X X X X X X X X	X X	X X X X X X X X	1 0 1 0
X X X X X X X X X	X X	X X X X X X X X	0 1 0 1

Coding schemes that satisfy the needs of clock recovery and DC-balance are a tradeoff between coder complexity and bandwidth utilization. Simple Manchester coders has an efficiency of only 50%: two symbols are sent for each received bit. Other codes such as 4b/5b used in FDDI [21], and 8b/10b proposed for Fiber-Channel [22], are more efficient than Manchester, achieving 80% efficiency, but are more complex to implement. Our code is only moderately complex to encode, very simple to decode, accomodates variable data widths, and achieves a high efficiency of 21/24, or 87.5%.

Chip Block Diagrams

A simplified block diagram of the TX chip is shown in Figure 3. The PLL/Clock Generator Block generates the high-speed serial clock by phase locking onto the incoming low-speed clock, which can be either at the full or one-half frame rate.

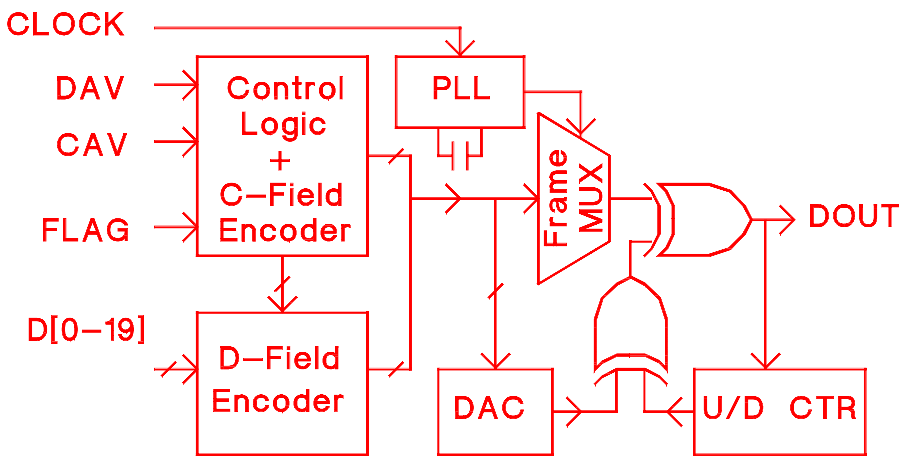

Figure 3. Simplified Block Diagram of TX chip

The receiver block diagram is shown in Figure 4. The data path consists of an input selector, two input sampling latches, a demultiplexer, a C-field decoder, and a D-field decoder. The input (DIN) is programmable to select data from either the normal data input, a loopback data input, or an equalized input. The equalized input provides a 3db boost at 600 MHz to compensate for skin-loss in long coaxial lines [23]. The improvement of Link BER with the equalizer used

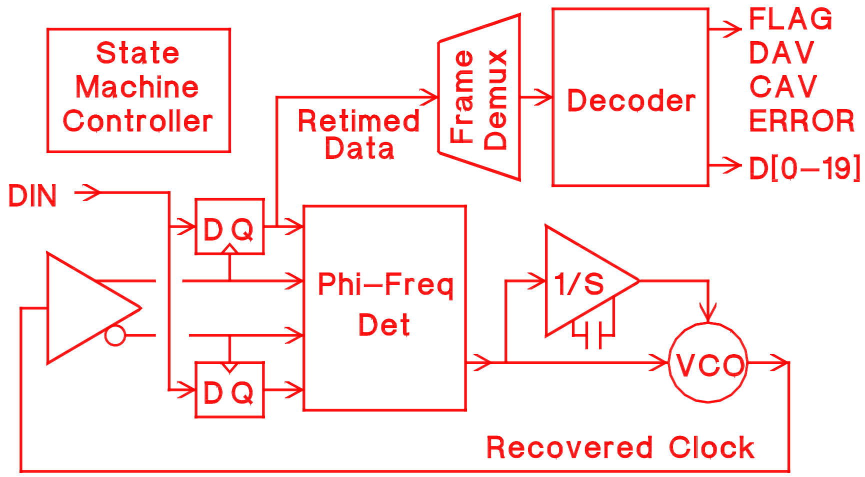

Figure 4. Simplifed Block Diagram of the RX Chip

with RG-58 coax is shown in Figure 5. For a given BER, the equalizer extends the usable link length by over 50%.

Phase Locked Loop Description

The on-chip PLLs used in both the TX and RX chips are nearly identical. For simplicity, this section only describes the implementation of the RX PLL.

Referring again to Figure 4, the incoming data stream is latched by two matched D-latches, one on the rising edge, and the other on the falling edge of the bit-rate VCO clock. When the loop is locked, the rising-edge retiming latch samples the center of each data bit and produces retimed data. The falling-edge phase detector latch samples the transitions between bits. The transition sample corresponding to the master transition is selected for use as a phase error

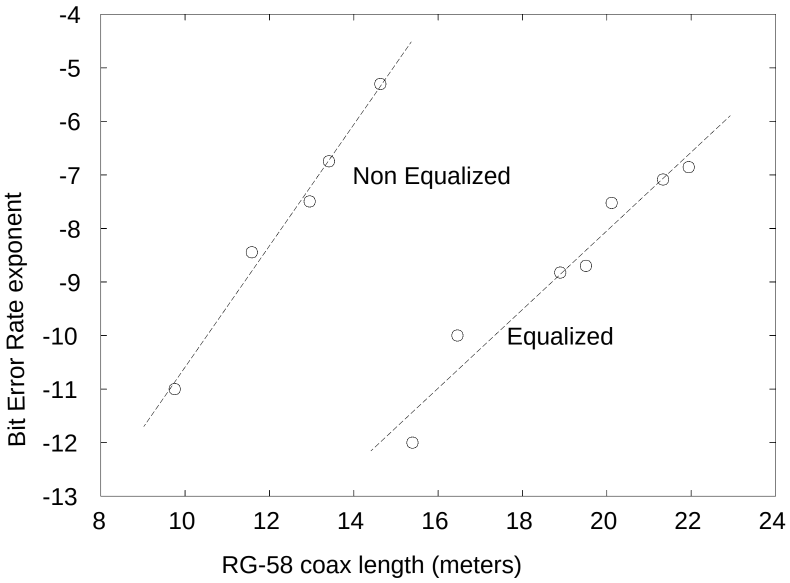

Figure 5. Bit Error Rate Performance With/Without Equalizer

indication. Since the code allows the master transition to be of either polarity, the sample is corrected for transition polarity by being XORed with the immediately preceding data bit to derive a binary-quantized (bang-bang) phase error indication. Because the phase detector and retiming latches are matched, assuming a 50% duty cycle VCO, the retiming clock phase is inherently aligned to the center of the bit cell over both process and temperature variation. In addition, the circuit can operate at the full speed at which a process is capable of building a functioning latch.

Phase-Locked Loop Dynamics

The phase detector described is non-linear, and conventional linear PLL theory is not useful for design or analysis. Precise loop behavior can be simulated efficiently with time-step simulators, but this is cumbersome to use for routine design. Fortunately a simple decomposition of the loop provides accurate closed-form expressions for both loop tracking jitter, and loop stability. An outline of this analysis is given in this section.

A simplified version of the clock recovery loop which assumes a fixed, rising, master transition is shown in Figure 6. The transition samples are decimated by the number of bits per frame, M, to isolate the one sample corresponding to the master transition.

If certain assumptions are met, as described in a later section, we can consider the system to be composed of two non-interacting loops. These are the loops labeled “bang-bang branch” and “integral branch” in Figure 6. The first loop includes the connection of the phase detector to the VCO input through the bang-bang branch of the loop filter, while the second loop includes the integral branch of the loop filter. The binary control, or “bang-bang” loop can be considered a phase tracking loop, while the integral branch can be viewed as a frequency tracking loop.

The fact that the phase detector output is quantized implies that the loop behavior will be oscillatory. In

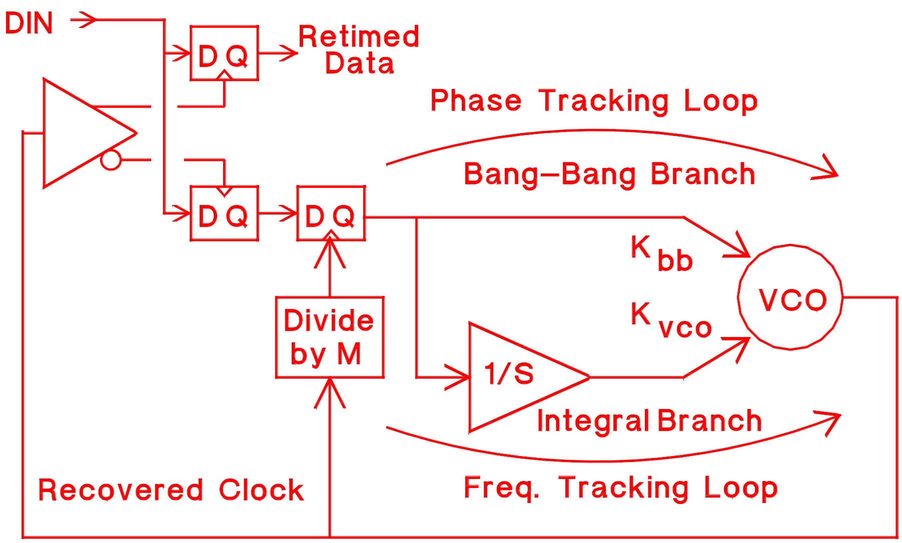

Figure 6. Simplified Clock Recovery Loop

steady-state conditions, the output of the phase detector (due to inevitable noise and jitter) will be a quasi-random string of “1’s” and “0’s”, which will program the VCO frequency to switch between two discrete frequencies, causing the VCO to ramp up and down in phase, thereby tracking the incoming signal phase. The phase detector output tends to alternate every frame, so that, other than the DC component, the bulk of the phase detector output spectrum falls outside the effective passband of the integrator branch of the loop, and can be practically neglected.

The integrator branch then operates on just the DC component of the phase detector output. Its job is to servo the center frequency of the VCO so that the two discrete VCO frequencies programmed by the bang-bang input will always bracket the frequency of the incoming data signal. This frequency adjustment occurs so slowly that it does not materially affect the operation of the high frequency bang-bang portion of the loop.

Proportional Branch of the Loop Filter

With a locked loop and assuming that the integrator output changes negligably during a single phase update, the VCO frequency step programmed by the bang-bang tuning input is $F_{step} = \pm\frac{\beta V_{\phi} K_{vco}}{2}$, where $\beta = \frac{K_{vco}}{K_{bb}}$; the ratio of the VCO wide range tuning gain to the bang-bang tuning gain; V_{ϕ} is the peak to peak voltage from the phase detector; K_{vco} and K_{bb} are the main and bang-bang VCO gain constants in Hz/volt.

Assuming a high DC gain in the loop integrator, the steady-state duty cycle from the phase detector output will be very close to 50%, usually alternating between “0” and “1” with an occasional doubling-up of bits to compensate for leakage in the integrator. The worst-case loop phase-error (in degrees) is then given by the phase walk-off of two successive update periods:

$$Jitter_{pp} = \frac{360\beta V_{\phi}K_{vco}M}{F_{vco}^2}$$

where $M$is the loop division ratio and F_{vco} is the nominal VCO frequency in Hz. In our loop, the hunting jitter is designed to be below 18 ps rms.

Before turning to the analysis of the integral branch of the loop, we need to derive the DC component, or duty cycle of the phase detector output stream. As already mentioned, in steady-state the duty cycle is 50%. Because the loop is phase-locked, the frequency of the VCO is, on the average, equal to the frequency of the serial data stream. If the incoming frequency is switched from F_{vco} to $F_{vco} + \Delta F$, with $-F_{step} \le \Delta F \le F_{step}$, the duty cycle, C, of the phase detector will necessarily shift such that

$$F_{vco} + \Delta F = C(F_{vco} + F_{step}) + (1 - C)(F_{vco} - F_{step})$$

Solving for the duty cycle,

$$C = \frac{\Delta F}{2F_{step}} + \frac{1}{2}$$

Unlike a traditional PLL, this result shows that the DC component of the phase detector output is proportional to frequency rather than phase. The effective gain constant of this “virtual” frequency detector, K_f, in volts/Hz is

$$K_f = \frac{V_\phi}{2F_{step}}$$

Integral Branch of the Loop Filter

To analyze the integral branch of the loop, both the binary-quantized phase detector and the bang-bang branch of the loop are replaced by an equivalent, linear “virtual” frequency detector with gain constant K_f. Standard linear feedback theory can then be easily used to determine the bandwidth and other salient characteristics of this loop. The unusual result is that the low frequency loop is only first order.

Because the phase-detector DC component is proportional to frequency rather than phase, an implicit integration does not appear in the loop transfer function. This means that there is no jitter buildup due to the action of the low frequency integrator. The jitter statistics are simply dominated by the hunting behavior of the high-frequency portion of the loop. However, unlike a normal first-order loop, the behavior of the bang-bang portion of the loop ensures that the average loop phase error remains zero with changes in input data frequency.

Loop Stability Criteria

The preceeding analysis assumed that the two branches of the loop were essentially non-interacting. For this to be true, it is important that the loop be set up so that, between phase sample update times, $(t_{update} = F_{vco}/M)$, the phase walkoff of the bang-bang branch of the loop, $\phi_{bb}(t)$, must dominate over the phase walkoff of the integral branch $\phi_{int}(t)$.

Taking the ratio of $\phi_{bb}(t)$ and $\phi_{int}(t)$ at the end of one frame update time, gives a figure of merit ξ for the loop stability:

$$\xi = \frac{2\beta\tau}{t_{update}}$$

ξ must be greater than one for the two loops to be considered non-interacting. In fact, if ξ becomes significantly less than 1, the “bang-bang” portion of loop will no longer stabilize the system and large low-frequency second-order phase oscillations will occur in the loop.

False-Locking and Frame Synchronization

During initial link startup, it is necessary to ensure that the PLL correctly determines the frequency of the incoming data, and also finds the location of the Master Transition.

In many clock-extraction circuits, the clock frequency is extracted from a coded, random data stream. A common difficulty with this approach is the problem of the PLL locking onto wrong frequencies which are harmonically related to the data rate. To avoid this problem, most systems limit the VCO range so that it can never be more than a few percent away from correct frequency.

Unfortunately, using a narrow-band VCO was not consistent with the goal of building a completely monolithic chipset. Integrated oscillators rely on low-tolerance IC components and are typically limited to +/- 30% tolerance on the center frequency. For customer flexibility, it was desired to extend the oscillator range to cover at least an octave. This range, in conjunction with digital dividers, allow the G-Link chipset to operate over a range of 120 to 1500 Mbaud, in four bands.

A second design problem is that of frame synchronization. At the receiver, some method must be employed to determine the boundaries between code frames so that they can be properly de-serialized back into the original parallel codewords. The G-Link chipset establishes and monitors frame synchronization by using the imbedded master transition. Unlike other links, the G-Link chipset allow the continuous transmission of unbroken streams of data, without the insertion of special synchronization words.

To eliminate the problems of false locking and frame synchronization, the G-Link chipset uses a start-up state machine and a frequency aquisition aid, as described in the next section.

State Machine Controller

Because the internal VCO is capable of operating over nearly a 3:1 range of frequencies, a frequency detector is necessary to avoid false locking problems. The frequency detector operates only when simple square-wave fill frames are being sent. A conventional sequential frequency detector, built of 2 resettable flip-flops [24], determines the sign of the frequency error. When the phase error is less than +/- 22.5°, the output of the phase detector is used. Otherwise, the loop filter is driven by the frequency detector output. Because the frequency detection circuit cannot operate on data frames, the State Machine Controller must disable the frequency detection circuit before data is allowed to be sent.

Neither node of a duplex link can achieve lock unless the opposite side is sending special Fill frames. Neither side of the link can stop sending Fill frames and start sending data unless it somehow knows that the other side has successfully achieved lock. The State machine uses the two distinct Fill frames: FF0 and FF1, to allow one side of the link to notify the other side of its current locking status. This guarantees that Fill frames will be sent whenever needed to restore lock, and only as long as necessary to achieve lock.

FF0 is a 50% balanced square wave with equal numbers of “0” and “1” bits. FF1 actually consists of two modified square-wave patterns: one which has the falling transition shifted on bit to the right, and the other shifted to the left. These two patterns are used in alternation to maintain balance on the link. Both FF0 and FF1 have a single, rising transition which is in the same position in the frame as the “master transition” of Data and Control frames. The rising edge of the Fill frames is used to initially establish an unambiguous frame reference. After initial lock, the master transition of the Data frames is used to maintain frame lock.

Figure 7 describes the state machine handshake procedure for a full duplex link in greater detail. Both the near and far ends of the link independently follow the state diagram of Figure 8. At power up, each end of the link enters the sequence at the arc marked “START”.

Each node in the state machine has three notations. The top notation is either “FDET” or “PHASE”. FDET stands for Frequency detect mode, and implies that the Frequency detector has been enabled in the RX Chip PLL. When the chip is in this mode, it is important that no data is being sent, as the frequency detector is only able to lock onto one of the spe-

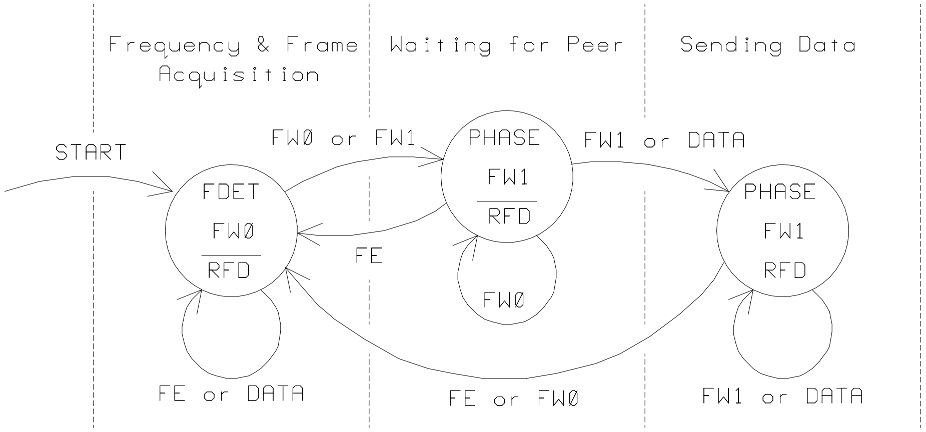

Figure 7. Link Startup State Machine

cial training Fill frames: FF0, or FF1. The PHASE notation means that the RX PLL is in phase detect mode and is ready for data transmission. The middle notation in each state is the type of Fill Frame which is to be sent by the node’s TX chip. The last notation is the Ready_For_Data (RFD) status on the TX chip. When RFD is low, the user holds off any incoming data while the TX chip is sending Fill frames. When RFD is high, data is sent if available, and otherwise Fill frames are sent to maintain link synchronization.

The consistent presence of the two Master Transition bits is monitored by the RX chip to detect a locked condition. If the RX chip detects an unlocked condition, this is flagged to the startup state machine as a Frame Error (FE). The RX chips at both ends of the link are able to detect the following frames: Data/Control, FF0, and FF1. Transitions are made from each of the states based on the current status condition received by the RX chip. Each of the subsequent arcs in the diagram are labeled with the relevant state that would cause a transition along that arc.

If either side of the full duplex link detects a Frame Error, it will notify the other side by sending FF0. When either side receives FF0, it follows the state machine arcs and reinitiates the handshake process. The user is notified of this action by the deasserting of RFD.

This startup protocol ensures that no user data is sent until the link connectivity is fully established. The use of a handshaked training sequence avoids the false lock problem inherent in PLL systems which attempt to lock onto random data with wide-range VCOs.

Implementation

The two chips were implemented in a 3-level metal, 25 GHz ft, silicon bipolar process [25] using full-custom, differential 4.5V ECL design. Both chips with their bypass and integrating capacitors are housed in a custom 68-pin surface-mount package. The 1.8W TX and 2.0W RX chips are each 3.5x3.5 mm in size and utilize 6100 and 6600 active devices, respectively. Both chips were fully functional at first silicon. Figure 8 shows the TX input clock at the parallel word rate, the transmitted frame, and the RX recovered clock at 1.5 Gbaud. Figure 9 shows a phase jitter histogram of RX recovered clock at 1.5 Gbaud demonstrating a loop hunting jitter of 8 ps RMS.

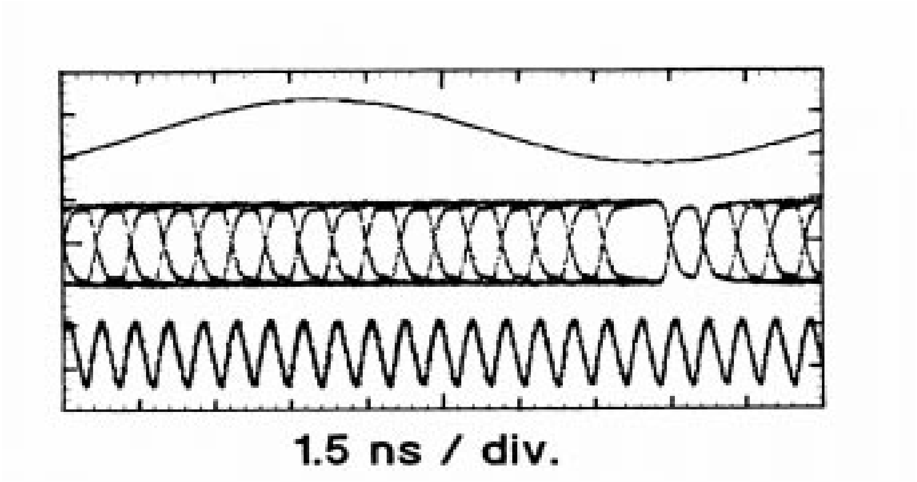

Figure 8. TX Input Clock, Transmitted Frame, and RX Recovered Clock at 1.5 Gbaud.

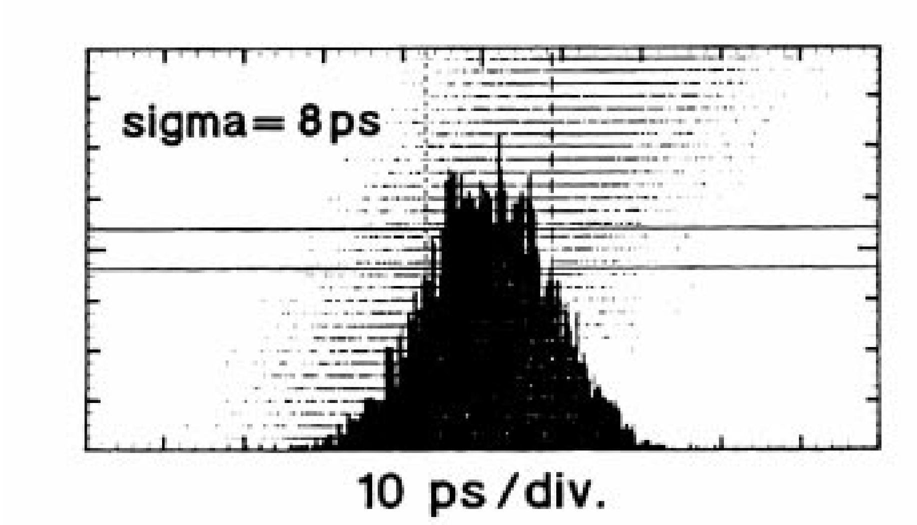

Figure 9. Phase Jitter Histogram of RX Recovered Clock at 1.5 Gbaud.

On-Chip Voltage Controlled Oscillator

The VCO is composed of a cascade of 3 variable delay blocks as shown in Figure 10. The low frequency signal from the integral branch of the loop drives the main tuning input, which is bandwidth limited to reduce its sensitivity to on-chip noise. The main tuning input adjusts the delay of each stage from one gate delay to three gate delays. The overall VCO frequency range is then approximately 3:1. This wide range allows the final system to be specified with a 2:1 range over both process and temperature. The “bang-bang” tuning input programs approximately +/- 0.1% steps in the VCO center frequency by modulating the base charge in Q1 and

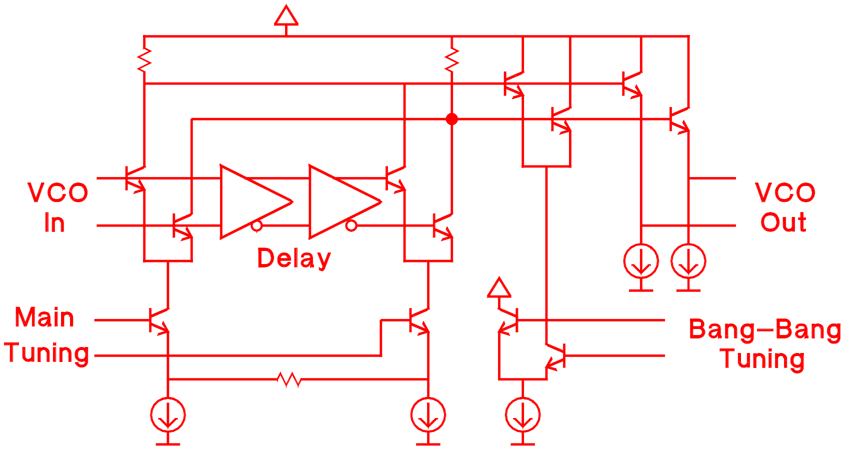

Figure 10. VCO Delay Cell

Q2, and is driven by the proportional branch of the loop filter.

Figure 11 shows the measured VCO tuning curve at three different power supply voltages: -4.5, 5.0, and -5.5.

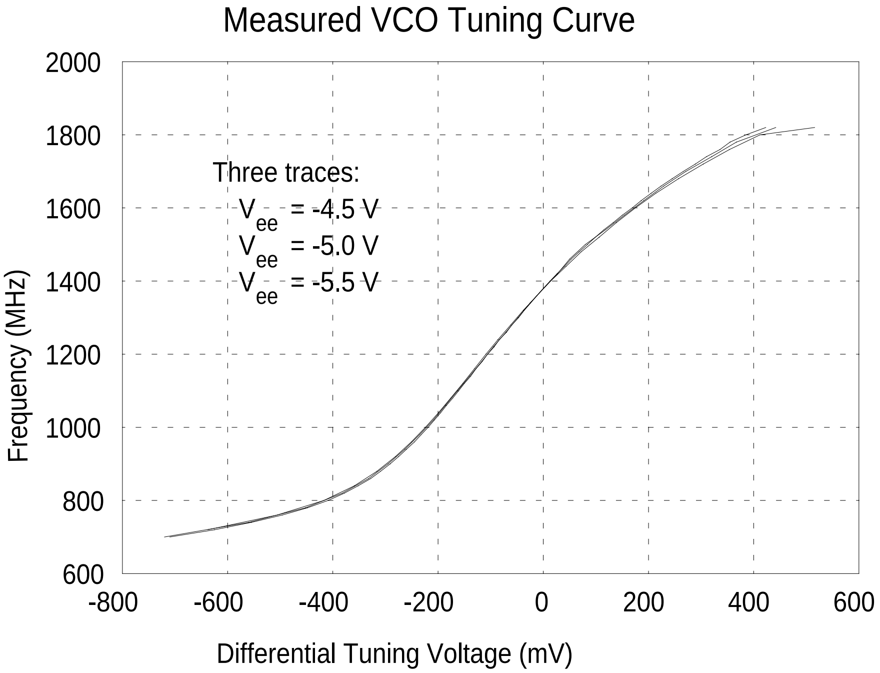

Figure 11. Measured VCO Tuning Curve at 4.5, 5.0 and 5.5 Volt Vee

Loop Filter

The loop filter is implemented with a charge pump integrator and a 0.1 uF external capacitor, which is housed within the package. The integrator is based on a unity-gain positive feedback technique (Figure 12) that cancels out the droop in the integrator filter capacitor [26]. The effective DC gain of this circuit approaches infinity as the feedback gain approaches unity. The unity-gain technique achieves high DC gain while avoiding the stability and noise sensitivity problems of on-chip high-gain op-amp designs.

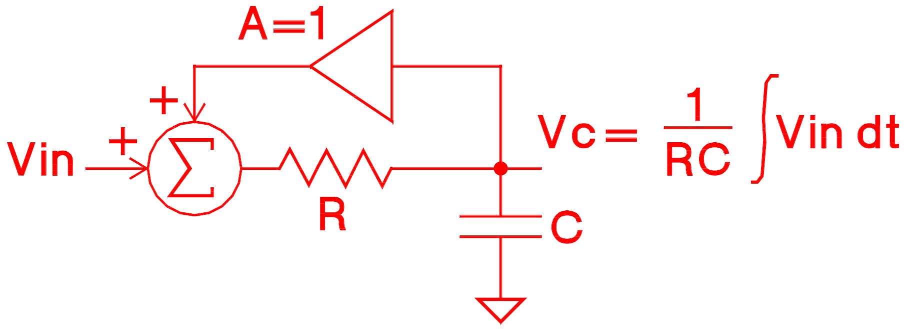

Figure 12. Block Diagram of Charge Pump Circuit

CHAPTER IV

COMPARISON BETWEEN 8B/10B AND CIMT CODE

DEC’s 8B/10B Code With Forward Error Correction

For HIPPI-Serial, Digital Equipment Corporation (DEC) proposed using an 8B/10B code similar to the IBM code that was used by the ANSI Fiber Channel group [27]. In addition to the overhead of the 8B/10B encoding, DEC also proposed the use of a forward error correction code (FEC).

The reason that an error correction code was considered is that many users planned to use the fiber link to extend existing HIPPI-copper connections. In a copper connection, the bit error rate (BER) is extremely low, usually better than 1e-15. Because of this low error rate, it is common to operate these links without any retry mechanism or other type of error control. In effect, the user is willing to have his computer crash once every decade or so as a trade-off for simple hardware. Unfortunately, fiber BER’s are often not much better than 1e-12 due to cable dispersion and modal noise.

The DEC proposal was to put error correction complexity into the code rather than complicate higher levels of the link protocol to support retries. The downside of this decision was that the overhead for an 8B/10B code with FEC is 50%, which is compared to the CIMT code overhead of 14%. However, with the FEC code all single-bit errors are corrected, and all double-bit errors are detected.

The DEC code breaks the data up into 8 byte or 64 bit chunks. Each byte is then coded with an 8B/10B code, expanding the number of bits to 80. Finally, 8 bits of FEC code are prepended to the frame and a complementary set of 8 bits is appended to the frame. By attaching the FEC bits to both the beginning and end of the frame in both true and complement form, the chance of an undetected error is minimized. After all overhead the final coded block size is 96 bits.

Comparison of DEC’s Code Versus HP’s Code

Deciding between DEC’s code and HP’s code comes down to a matter of philosophy. The use of laser links for short distance computer use is still fairly unproven technology. There was much disagreement as to what the worst case bit error rate would prove to be in production systems. There was also some disagreement on how error control of the laser link should be handled.

If the user intends to run without a higher level protocol performing error detection and retry, then a forward error correction code may be necessary to ensure data integrity. The price paid is higher coding overhead when compared to a more efficient code such as the 21B/24B CIMT code.

One way to look at this efficiency trade-off is to plot link utilization vs BER. For this analysis, we assume that retries do not cost extra bandwidth. This should be nearly achievable with the use of a modern windowing protocol, large packet sizes, and piggy-back acknowledgments.

For HIPPI-serial, data is sent in multiples of 256 word bursts. Each word can be 32 bits or 64 bits. For the 32 bit case, a burst is 8192 bits. A complete unencoded frame then consists of N*8192 bits.

For the HP code, the expected probability of transmitting a complete frame is equal to the probability of transmitting N*8192*(24/20) bits without an error. The link utilization is then the expected probability of error free transmission multiplied by the link code overhead.

$$Utilization_{CIMT} = \frac{20}{24}(1-p_e)^{\frac{N \bullet 8192 \bullet 24}{20}}$$

where p_e is the probability of a single bit to be in error.

For the DEC code, we must take into account the fact that all single bit errors within each 96 bit block are corrected. The inner term of the utilization equation is composed of the probability that no bits in a block are in error

$(1-p_e)^{96}$, plus the probability that exactly one bit is in error $96 \bullet p_e(1-p_e)^{96}$.

$$Utilization_{8B10B} = \frac{64}{96}[(1-p_e)^{96} + 96 \cdot p_e(1-p_e)^{95}]^{\frac{N \bullet 8192}{64}}$$

Figure 13 graphs both equations as a function of BER for N=1,10,100 and 1000. These values of N correspond to packet sizes of 8k, 80k 800k and 8 Mbits. In terms of raw utilization, the CIMT code with retry is about 20% more efficient than the DEC code up to quite high bit error rates. For 8 Mbit packets, the CIMT code is more efficient up to about 1e-8 BER.

There are a couple of interesting observations to be made by examining Figure 13. The first is that utilization is degraded by large packet sizes. This is because the probability of a packet having an unrecoverable error goes up with packet size. Another interesting observation is that in the rolloff region of Figure 13, above 1e-8 BER, the horizontal spacing between the curves in each utilization family are spaced twice as closely in the DEC case compared to the HP case. This is because the DEC code is sensitive to 2 bit errors rather than single bit errors.

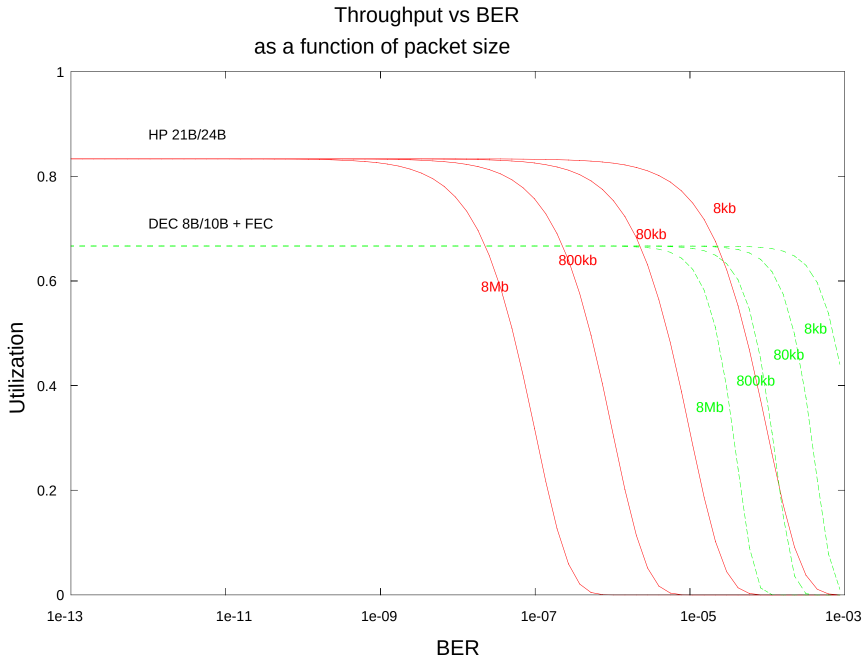

Figure 13. Efficiency of DEC and HP Code with Retries.

Although our assumption of transparent retry may seem unwarranted, the fact that the utilization of the HP code exceeds the DEC code up to 1e-9 BER even with 8Mbit packets is promising. With such a large packet size, you can achieve high retry efficiency by using a simple “stop and wait” acknowledgment system rather than a complex window protocol.

To be fair, there are some situations in which an FEC code may provide an advantage. An FEC code is likely to reduce the overall link latency. Any single-bit errors that occur, are immediately corrected and do not need to be retransmitted. This could be an important property for networks with long transmission delays that need to carry low-latency traffic.

CHAPTER V

COMPARISON OF G-LINK CLOCK RECOVERY CIRCUIT WITH TRADITIONAL TECHNIQUES

Typical Clock Extraction Circuits

Figure 14 shows a representative clock extraction circuit that is used for high bit-rate telecom systems. The incoming analog data stream is split into two parallel paths: the clock extraction chain, and the data retiming path.

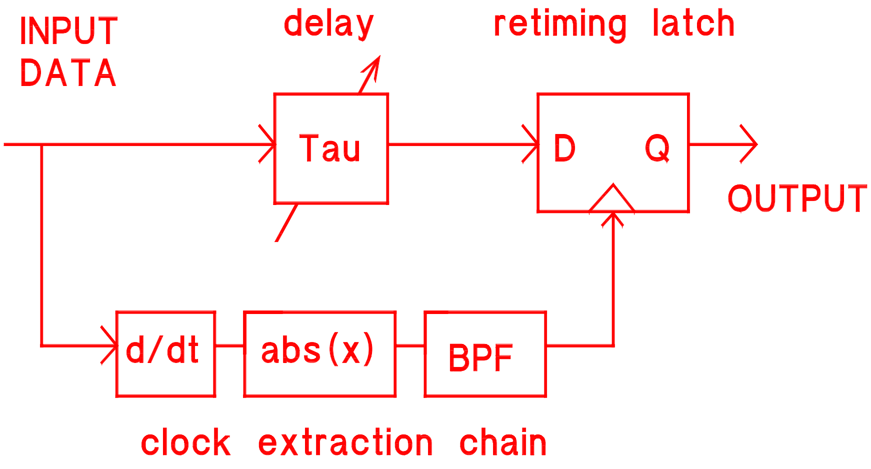

Figure 14. Traditional Clock Extraction Circuit

Because an NRZ data stream does not have a spectral component at the clock frequency, some non-linear process must be used to derive a clock signal from the data stream. In this example, a typical circuit is shown which consists of a time derivative followed by an absolute value function. This combination of elements creates a uni-directional pulse for every transition of the data. This new waveform contains a spectral component at the clock frequency. Once the clock component has been created, it can be isolated with a band-pass filter (BPF), implemented by either a passive filter or a phase-locked loop.

The problem with using either a filter or PLL in this configuration is that, although the circuit extracts the correct clock frequency, it does not extract the correct phase. For both approaches, there is a large phase shift between the input data and the recovered clock. The phase relationship between clock and data must then be somehow adjusted to compensate for variations due to process and temperature. This is commonly done by making an adjustment to a variable delay element in either the data or clock path. This variable delay adjustment can be integrated and temperature compensated, but the external high-Q filter is not suitable for low-cost monolithic implementation.

The new PLL as described in Chapter III avoids the phase mismatch problem because the phase detection circuit is inherently phase-matched to the retiming latch delay.

CHAPTER VI

CONCLUSIONS

Using a novel PLL, an adjustment-free, fully monolithic clock recovery circuit has been developed that can operate at speeds approaching the maximum flip-flop toggle rate of a given process.

The chipset embodies principles of information hiding and greatly simplifies the use of Gigabit-rate link technology in computer systems. Coding, Decoding, Multiplexing, Demultiplexing, Link Start-up, Frame Synchronization, and Link Monitoring are all handled by the chip with no intervention from the user.

Data, Fill, and Control Frames are supported by the chipset to make it useable with modern protocols. The chipset has been adopted, or is being considered, as a standard by several groups. These include IEEE P-1596 SCI-FI and the Ad-Hoc ANSI HIPPI-Serial groups.

From a survey of current gigabit standards, several trends are evident. Gigabit-rate networks will have to be simple and robust, probably doing as much in hardware as possible. Interconnects will be simplex only, with duplex connections built out of two separate simplex links. More complex topologies will be supported with switches and other building blocks.

There are two camps emerging, those who are willing to build a complex physical layer with the hope that the link can be made virtually error-free, and those who believe that the physical layer should be as simple as possible and that higher level retry mechanisms should handle error detection and correction. The final decision will be made on the basis of latency needs, and the sensitivity of laser system cost with bandwidth. The G-Link chipset design falls in the second camp, minizing gigabit-rate hardware cost and complexity by relying on upper level protocol error correction mechanisms.

REFERENCES CITED

REFERENCES CITED

[1] S. Vesuna and S. Price, “Digital Video Data Can Now Travel by Taxi,” *Computer Design*, pp. 80-83, Feb. 1, 1987.

[2] J. Tani, D. Crandall, J. Corcoran and T. Hornak, “Parallel Interface IC’s for 120Mb/s Fiber Optic Links,” *ISSCC Digest of Technical Papers*, vol. 30, pp. 190-191, Feb. 1987.

[3] R. C. Walker, T. Hornak, C. Yen and K. Springer, “A Chipset for Gigabit Rate Data Communication,” in *Proceedings of the 1989 Bipolar Circuits and Technology Meeting*, September 18-19, 1989, pp. 288-290.

[4] R. C. Walker, “A Fully Integrated High-Speed Voltage Controlled Ring Oscillator,” U.S. Patent 4 884 041, Nov. 28, 1989.

[5] R. C. Walker, and B. Lai, “A Method and Apparatus for Clock Recovery and Data Retiming for Random NRZ Data,” U.S. Patent 5 012 494, April 30, 1991.

[6] R. C. Walker, T. Hornak, C. Yen, J. Doernberg and K. H. Springer, “A 1.5 Gb/s Link Interface Chipset For Computer Data Transmission,” *IEEE Journal on Selected Areas in Communications*, vol. 9, no. 5, pp. 698-703, June 1991.

[7] D. Crandall, S. Hessel, T. Hornak, R. Nordby, K. Springer and R. C. Walker, “DC-Free Code for Arbitrary Data Transmission,” U.S. Patent 5 022 051, June 4, 1991.

[8] *IEC 825*, CENELEC, 1984. (HD 482 S1).

[9] C. Corsetto, T. Hornak, R. Nordby, R. C. Walker and C. Yen, “Phase Locked Loop for Clock Extraction in Gigabit Rate Data Communication Links,” U.S. Patent 4 926 447, May 15, 1990.

[10] D. Tolmie, “The High-Speed Channel (HSC) and Ultra-High-Speed Networks,” *Broadband (FOC/LAN) ‘89 Proceedings*, San Francisco, CA, October 30 - November 3, 1989, pp. 421-426.

[11] D. E. Tolmie, “The High-Speed Channel (HSC) Standard,” Digest of Papers. *COMPCON Spring ‘89*, San Francisco, CA, February 27 - March 3, 1989, pp. 314-317.

[12] *Serial-HIPPI Specification*, Revision 1.0, May 17, 1991.

[13] D. Knudson, B. Schofield, K. Springer, D. Byrne and B. Thomas, “Using a 1 Gbit/sec Fiber Optic Link with Forward Error Correction to Provide a 10-16 Bit Error Rate,” *Presented to Serial-HIPPI Working Group*, Nov. 7, 1990.

[14] *Preliminary Hot Rod High-Speed Serial Link (data sheet)*, Gazelle Microcircuits, pp. 6-9, September 1989.

[15] J. Toy, and P. Casper, “HIPPI Extender Requirements,” *Serial-HIPPI Implementors Group Minutes*, Nov. 7, 1990.

[16] R. Walker, “Hewlett-Packard’s 21b/24b Code Proposal for HIPPI-Serial,” *Serial-HIPPI Implementors Group Minutes*, February 19, 1990.

[17] D. B. Gustavson, “Scalable Coherent Interface,” *Digest of Papers COMPCON Spring ‘89,* San Francisco, CA, February 27 - March 3, 1989 , pp. 536-538.

[18] D. B. Gustavson, “IEEE P1596, a Scalable Coherent Interface for GigaByte/sec Multiprocessor Applications,” *IEEE transactions on Nuclear Science*, vol. 36, no. 1, pp. 811-812, 1989.

[19] R. Walker, “The HP 17b/20b Code: A Proposal for Sci-Fiber Link,” *IEEE P-1596 SCI-FI Meeting Minutes*, November 14, 1990.

[20] *SCI Scalable Coherent Interface (Draft)*, IEEE P1596, November 10, 1990.

[21] *FDDI Physical Layer Protocol (PHY)*, X3T9/85-39, Draft Proposed American National Standard, May 10, 1985.

[22] *Fiber Channel Physical Layer Medium Dependent (FC-0)*, X3T9.3, Working Draft Proposed American National Standard for Information Systems, March 5, 1990.

[23] C. S. Yen, Z. Fazarinc and R. Wheeler, “Time-Domain Skin-Effect Model for Transient Analysis of Lossy Transmission Lines,” *Proceedings of the IEEE* vol. 70, no. 7, pp. 750-757, July 1982.

[24] I. Shahiary, G. Des Brisay, S. Avery and P. Gibson, “GaAs Monolithic Digital Phase/Frequency Discriminator,” *1985 GaAs IC Symposium*, pp 183-186.

[25] W. M. Huang, et al.,”A High-Speed Bipolar Technology Featuring Self-Aligned Single-Poly Base and Sub-Micron Emitter Contacts,” *IEEE Electron Device Letters*, vol. 11, no. 9, pp. 412-414, September 1990.

[26] B. Lai, and R. Walker, “Unity Gain Positive Feedback Integrator with Programmable Charging Currents and Polarities,” U.S. Patent Pending, Filed April 30, 1992.

[27] 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, September 1983.

[28] C. B. Armitage, “SAW Filter Retiming in the AT&T 432 Mb/s Lightwave Regenerator,” *Xerox Copy - Missing Citatio*n, pp. 102-103, 1983.

[29] J. Gruber, M. Holz, P. Marten, R. Petschacher, P. Russer and E. Weidel, “A 1 Gbit/s Fibre Optic Communications Link, *4th European Conference on Optical Communication*,” Genua, pp. 556-563, September 12-15, 1978.