

High Speed Serdes Devices and Applications

David R. Stauffer
Jeanne Trinko Mechler
Michael Sorna
Kent Dramstad
Clarence R. Ogilvie
Amanullah Mohammad
James Rockrohr

Springer

Marvell Semiconductor, Inc. - Ex. 1008, Page 000001
IPR2025-01220 (Marvell Semiconductor, Inc. v. Credo Technology Group Ltd.)

High Speed Serdes Devices and Applications

Marvell Semiconductor, Inc. - Ex. 1008, Page 000002

David R. Stauffer • Jeanne Trinko Mechler
Michael Sorna • Kent Dramstad
Clarence R. Ogilvie • Amanullah Mohammad
James Rockrohr

High Speed Serdes Devices and Applications

Springer

Marvell Semiconductor, Inc. - Ex. 1008, Page 000003

David R. Stauffer
IBM Corporation
Essex Junction, VT
USA

Kent Dramstad
IBM Corporation
Essex Junction, VT
USA

Amanullah Mohammad
IBM Corporation
Research Triangle Park, NC
USA

Michael A. Sorna
IBM Microelectronics
Hopewell Junction, NY
USA

Jeanne T. Mechler
IBM Corporation
Essex Junction, VT
USA

Clarence R. Ogilvie
IBM Corporation
Essex Junction, VT
USA

James D. Rockrohr
IBM Microelectronics
Hopewell Junction, NY
USA

ISBN 978-1-4419-4641-6 e-ISBN 978-0-387-79834-9

© 2009 Springer Science+Business Media, LLC
Softcover reprint of the hardcover 1st edition 2009
All rights reserved. This work may not be translated or copied in whole or in part without the written permission of the publisher (Springer Science+Business Media, LLC, 233 Spring Street, New York, NY 10013, USA), except for brief excerpts in connection with reviews or scholarly analysis. Use in connection with any form of information storage and retrieval, electronic adaptation, computer software, or by similar or dissimilar methodology now known or heareafter developed is forbidden. The use in this publication of trade names, trademarks, service marks, and similar terms, even if they are not identified as such, is not to be taken as an expression of opinion as to whether or not they are subject to proprietary rights.
While the advice and information in this book are believed to be true and accurate at the date of going to press, neither the authors nor the editors nor the publisher can accept any legal responsibility for any errors or omissions that may be made. The publisher makes no warranty, express or implied, with respect to the material contained herein.

9 8 7 6 5 4 3 2 1

springer.com

Marvell Semiconductor, Inc. - Ex. 1008, Page 000004

Table of Contents

Preface v
Acknowledgments vii

Chapter 1: Serdes Concepts 1
1.1 The Parallel Data Bus 1
1.2 Source Synchronous Interfaces 2
Reducing the Number of I/O Pins 2
Clock Forwarding 3
Higher Speed Source Synchronous Interfaces 4
1.3 High-Speed Serdes 8
Serializer / Deserializer Blocks 9
Equalizers 10
Clock and Data Recovery (CDR) 14
Differential Driver 15
Differential Receiver 17
Diagnostic Functions 17
Phase-Locked Loop 19
1.4 Signal Integrity 19
The Channel 19
Package Models 21
Jitter 21
Channel Analysis Tools 23
1.5 Signaling Methods 24
1.6 Exercises 27

Chapter 2: HSS Features and Functions 31
2.1 HSS Core Example: HSS EX10 10-Gbps Core 31
HSS EX10 Input/Output Pin Descriptions 32
HSS EX10 Register Descriptions 41
2.2 HSS EX10 Transmitter Slice Functions 53
Transmitter Parallel Data 54
Transmitter Signal Characteristics 56
Transmitter FFE Programming 58
Transmitter Power Control 59
Half-Rate/Quarter-Rate/Eighth-Rate Operation 60
JTAG 1149.1 and Bypass Mode Operation 62
PRBS / Loopback Diagnostic Features 64
Out of Band Signalling Mode (OBS) 65
Features to Support PCI Express 65
2.3 HSS EX10 Receiver Slice Functions 66
Receiver Data Interface 68
DFE and Non-DFE Receiver Modes 70

Marvell Semiconductor, Inc. - Ex. 1008, Page 000005

Serial Data Termination and AC/DC Coupling 71
Signal Detect 71
Receiver Power Control 72
JTAG 1149.1/1149.6 and Bypass Mode Operation 73
Half-Rate/Quarter-Rate/Eight-Rate Operation 76
PRBS / Loopback Diagnostic Features 77
Phase Rotator Control/Observation 78
Support for Spread Spectrum Clocking 78
Eye Quality 79
SONET Clock Output 80
Features to Support PCI Express 80
2.4 Phase-Locked Loop (PLL) Slice 80
Reference Clock 81
Clock Dividers 82
Power On Reset 82
VCO Coarse Calibration 83
PLL Lock Detection 83
Reset Sequencer 84
HSS Resynchronization 84
PCI Express Power States 87
2.5 Reset and Reconfiguration Sequences 87
Reset and Configuration 87
Changing the Transmitter Configuration 90
Changing the Receiver Configuration 92
2.6 References and Additional Reading 93
2.7 Exercises 94

Chapter 3: HSS Architecture and Design. 99
3.1 Phase Locked Loop (PLL) Slice 100
PLL Macro 101
Clock Distribution Macro 102
Reference Circuits 103
PLL Logic Overview 105
3.2 Transmitter Slice 107
Feed Forward Equalizer (FFE) Operation 109
Serializer Operation 112
3.3 Receiver Slice 114
Clock and Data Recovery (CDR) Operation 116
Decision Feedback Equalizer (DFE) Architectures 118
Data Alignment and Deserialization 121
3.4 References and Additional Reading 122
3.5 Exercises 123

Marvell Semiconductor, Inc. - Ex. 1008, Page 000006

Chapter 4: Protocol Logic and Specifications 125
4.1 Protocol Specifications 125
Protocol Layers 125
Serial Data Specifications 126
Basic Concepts 132
4.2 Protocol Logic Functions 134
Bit/Byte Order and Striping/Interleaving 134
Data Encoding and Scrambling 136
Error Detection and Correction 143
Parallel Data Interface 147
Bit Alignment 152
Deskewing Multiple Serial Data Links 153
4.3 References and Additional Reading 158
4.4 Exercises 159

Chapter 5: Overview of Protocol Standards 165
5.1 SONET/SDH Networks 168
System Reference Model 169
STS-1 Frame Format 170
STS-N Frame Format 174
Clock Distribution and Stratum Clocks 176
5.2 OIF Protocols 177
System Reference Model 177
SFI-5.2 Implementation Agreement 180
SPI-S Implementation Agreement 184
CEI-P Implementation Agreement 188
Electrical Layer Implementation Agreements 190
5.3 Ethernet Protocols 197
Physical Layer Reference Model 198
Media Access Control (MAC) Layer 201
XGMII Extender Sublayer (XGXS) 204
10-Gb Serial Electrical Interface (XFI) 207
Backplane Ethernet 213
PMD Sublayers for Electrical Variants 218
5.4 Fibre Channel (FC) Storage Area Networks 220
Storage Area Networks (SANs) 220
Fibre Channel Protocol Layers 222
Framing and Signaling 222
Physical Interfaces 229
10-Gbps Fibre Channel 236
5.5 PCI Express 237
PCI Express Architecture 238
Physical Layer Logic 241
Electrical Physical Layer 246
Power States 249
PCI Express Implementation Example 250

Marvell Semiconductor, Inc. - Ex. 1008, Page 000007

5.6 References and Additional Reading 251
5.7 Exercises 254

Chapter 6: Reference Clocks 263
6.1 Clock Distribution Network 263
Single-Ended vs. Differential Reference Clocks 263
Reference Clock Sources 265
Special Timing Requirements 268
Special Test Requirements 270
6.2 Clock Jitter 270
Jitter Definitions 271
Jitter Effects 276
PLL Jitter 277
6.3 Clock Floorplanning 281
Clock Tree Architecture 281
Clock Tree Wiring 282
6.4 Signal Integrity of the Clock Network 283
Analog Signal Levels and Slew Rates 283
Duty Cycle Distortion 286
Differential Clock Analysis Methodology 288
6.5 References and Additional Reading 293
6.6 Exercises 293

Chapter 7: Test and Diagnostics 297
7.1 IEEE JTAG 1149.1 and 1149.6 298
JTAG 1149.1 Overview 299
HSS Core Support for JTAG 1149.1 302
HSS Core Support for JTAG 1149.6 303
7.2 PRBS Testing and Loopback Paths 306
Loopback Paths 306
PRBS Circuits and Data Patterns 309
PRBS Test Sequence 314
7.3 Logic Built-In-Self-Test (LBIST) 317
LBIST Architecture 317
LBIST Considerations for HSS Cores 319
7.4 Manufacturing Test 320
Chip Level Test 320
HSS Macro Test 324
7.5 Characterization and Qualification Testing 327
Transmitter Tests 328
Receiver Tests 335
General Tests 338
7.6 References and Additional Reading 340
7.7 Exercises 340

Marvell Semiconductor, Inc. - Ex. 1008, Page 000008

Chapter 8: Signal Integrity 345
8.1 Probability Density Functions 345
Gaussian Distribution 345
Dual-Dirac Distribution 348
8.2 Jitter 349
Jitter Components 349
Deterministic Jitter 352
Random Jitter 356
Total Jitter and Mathematical Models 358
Jitter Budgets 362
Jitter Tolerance 364
8.3 Spice Models 365
Traditional Spice Models 365
Hybrid Spice/Behavioral Models 367
Spice Simulation Matrices 369
8.4 Statistical Approach to Signal Integrity 372
Analysis Approach 373
HSSCDR Software 388
8.5 References and Additional Reading 393
8.6 Exercises 394

Chapter 9: Power Analysis..... 397
9.1 Digital Logic Circuits 397
Digital Logic Active or AC Power 397
Digital Logic Leakage or DC Power 402
9.2 Non Digital Logic Circuits 410
AC (Active) Power 410
DC (Leakage) Power 410
Quiescent Power 410
9.3 HSS Power 411
HSS Power Equation 411
Multiple Power Supplies 412
Chip Fabrication Process 413
Mode-Dependent Power 414
Power Dissipation Breakdown 416
9.4 Reducing Power Dissipation 417
Power Concerns for the HSS Core Design 417
Power Dissipation Concerns for the Chip Designer 420
9.5 References and Additional Reading 421
9.6 Exercises 421

Marvell Semiconductor, Inc. - Ex. 1008, Page 000009

Chapter 10: Chip Integration 425
10.1 Simulation Models 427
Reset and Initialization Short Cuts 427
Simulation 'X' States 429
Modeled and Unmodeled Behavior 432
10.2 Test Synthesis 434
Scan Test Support 435
Macro Test Support 436
JTAG Logic Connections 440
Automation of Test Requirements 442
Running Macro Test using the JTAG Interface 444
10.3 Static Timing Analysis 445
Clock Timing 445
Receiver Parallel Data Outputs 450
Register Interface 452
Transmitter Synchronization 454
Serial Data Timing 456
Skew Management 457
Timing Backannotation for Simulation 458
10.4 Chip Floorplan and Package Considerations 459
Packages 459
Chip Physical Design 466
10.5 References 471
10.6 Exercises 472

Index..... 475

Marvell Semiconductor, Inc. - Ex. 1008, Page 000010

Chapter 1
Serdes Concepts

This chapter describes basic methods of transferring data from one chip to another chip, either on the same circuit board or across a cable or backplane to another circuit board. After reading this chapter, the reader should have a basic understanding of the rationale for using high-speed serializer/deserializer (Serdes) devices, and the inherent problems introduced by the high-speed operation of such devices.

1.1 The Parallel Data Bus

The simplest method of transferring data through the inputs or outputs of a silicon chip is to directly connect the datapath from one chip to the next chip (see Fig. 1.1). Since data often consists of more than one bit of information, the datapath is more than one-bit wide. In the figure, an *n*-bit datapath inside *Chip #1* is driven through chip outputs, across an *n* bit interconnect, through inputs of *Chip #2*, to an *n*-bit datapath inside the receiving chip. Synchronous data is transferred between the two chips since both chips are clocked by the same clock source.

There are two inherent problems of the parallel data bus shown in Fig. 1.1. The first problem is that *n* input/output (I/O) pins are required on each chip to transfer the data. At one point in history this was acceptable. However, Moore's Law has driven substantial increases in the number of circuits that can be manufactured on a chip compared to a few decades ago. The pin densities of chip packaging technologies have not increased at the same pace as silicon density. Therefore, I/O pins are substantially more expensive than silicon circuits, and dedicating *n* I/O pins for the above data bus is not acceptable for most chip applications.

The second inherent problem involves meeting timing requirements. The data is launched synchronously by *Chip #1* and is captured synchronously in *Chip #2* using the same clock. The data at the inputs of *Chip #2* must meet setup and hold times relative to the clock input of the chip. These setup and hold times must be calculated with sufficient margin to allow for differences in delay of the clock distribution path to the two chips, and through the chips to the launch and capture flip-flops. Delays may vary based on chip process, voltage, and temperature (PVT) conditions, and margin must be added to account for worst case variations. For higher clock frequencies, it may be necessary to use phase-locked loops (PLLs) in the chips to adjust the clock phase in order to compensate for the clock distribution delay within the chip and adapt to changing process, voltage, and temperature conditions. If the clock frequency is high enough, it will not be possible to build a system that will reliably transfer the data across this data bus.

D. R. Stauffer et al., *High Speed Serdes Devices and Applications*,

© Springer 2008

Marvell Semiconductor, Inc. - Ex. 1008, Page 000011

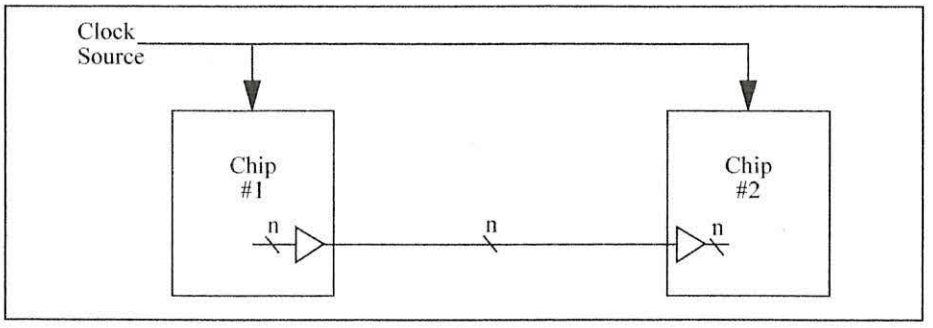


Fig. 1.1 Parallel data bus between two chips

1.2 Source Synchronous Interfaces

The two problems with the parallel data bus in Sect. 1.1 can be eliminated with the modifications to the system which are discussed in this section. These approaches are extensions of the parallel data bus. The parallel data bus and all of the extensions described in this section are considered to be *source synchronous interface* architectures. Such architectures include any interface where a clock input exists that can be used to capture the received data. This may be either a reference clock used by both the transmitting and receiving chip or the transmitting chip drives a clock to the receiving chip. In either case, clock recovery circuits are *not* required for source synchronous interfaces.

1.2.1 Reducing the Number of I/O Pins

The first issue to be addressed is reducing the number of I/O pins required to transfer the data between the chips. This is accomplished by multiplexing the n bits of data at the output of *Chip #1* onto k bits of interconnect ($k < n$), and then demultiplexing the k bits of interconnect at the input of *Chip #2* onto an n bit internal datapath. This is shown in Fig. 1.2. The resulting system only requires k I/O pins on each chip rather than the n pins previously required.

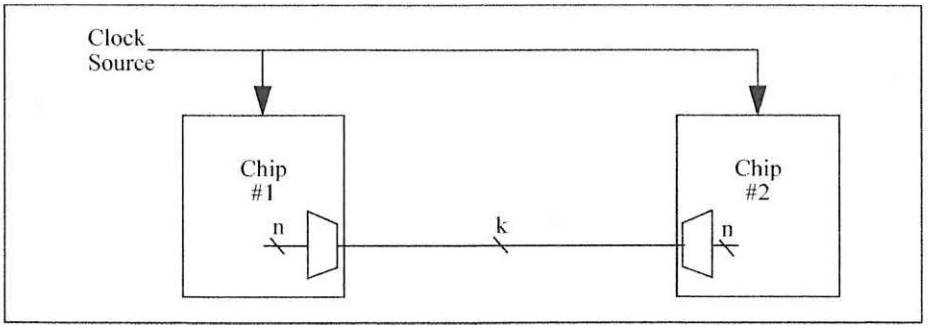


Fig. 1.2 Serializing the data to reduce pin counts

Marvell Semiconductor, Inc. - Ex. 1008, Page 000012

Of course, while the pin count requirements have been reduced by the ratio of $k : n$, the required frequency of the reference clock has increased by the inverse of this ratio. System designers generally do not like to distribute high-speed reference clocks within the system due to noise, electromagnetic interference (EMI), and power dissipation concerns. Often, a lower frequency clock is distributed, and PLLs in the chips are used to multiply this reference clock to a usable frequency. Variability of the phase of the resulting clock, along with the higher frequency of data transfer, tends to exacerbate the timing issues of the parallel data bus approach.

1.2.2 Clock Forwarding

In Fig. 1.3, a high-speed clock has been added to the datapath between the two chips. This clock source is assumed to supply a clock frequency somewhat lower than the frequency required to clock the data flip-flops on the chip interconnect. PLLs are used in each chip to generate clocks at a multiple of this frequency. The resulting clocks are used to launch and capture data in the respective chips. The output clock of the PLL in *Chip #1*, which is used to launch the data from this chip, is also an output of this chip. This clock is used by *Chip #2* to capture the data. This approach is called *clock forwarding*.

The advantage of this approach is that the high-speed clock used to launch the data at *Chip #1* is available to *Chip #2* as a reference to capture the data. Any variations in delays through clock distribution network driving the two chips does not need to be taken into account in timing analysis. Only delay variations between the clock path and the data bits are relevant. Variations between these paths due to process, voltage, and temperature track each other to some extent. The result is that timing analysis of the interface requires less margin and setup and hold times are therefore easier to meet.

So far we have not made any distinction or recommendations regarding the frequency of the high-speed clock relative to the bit rate of the interface. In general, the high-speed clock shown in the figure could be *single data rate* (SDR) or *double data rate* (DDR) (Fig. 1.4). The receiving chip captures data on every rising (or every falling) edge of an SDR clock; while the receiving chip captures data on every edge (both rising and falling edges) of a DDR clock.

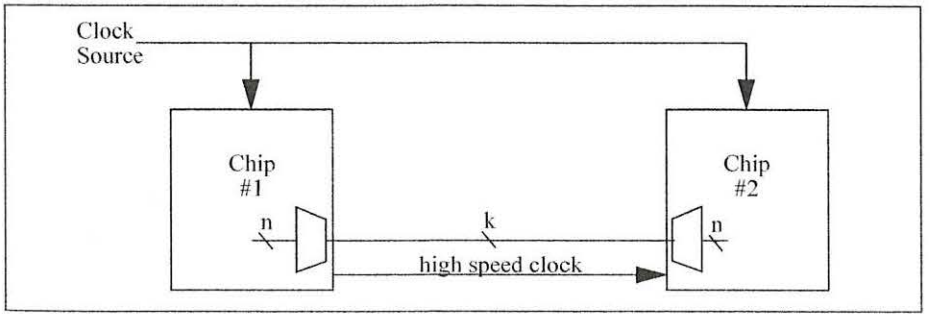


Fig. 1.3 High-speed clock forwarded with the data

Marvell Semiconductor, Inc. - Ex. 1008, Page 000013

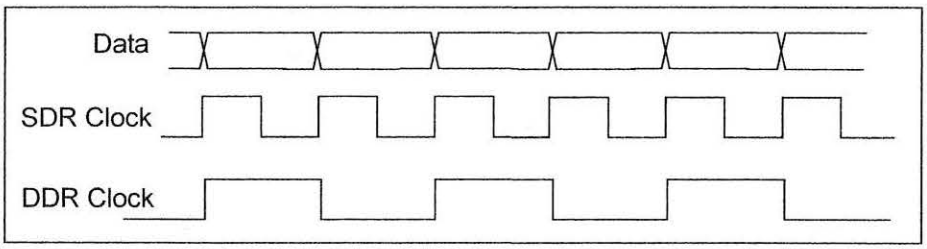


Fig. 1.4 Single data rate and double data rate clocks

The advantage of DDR clocks over SDR clocks lies in the bandwidth requirements for the corresponding I/O drivers and receivers. An I/O cell being used at a bit rate of *b* Mbits per second requires a bandwidth sufficient to transmit a 101010... data pattern. This corresponds to a frequency spectrum with an upper fundamental frequency limit of *b*/2 MHz. The corresponding frequency of an SDR clock is *b* MHz, twice the spectral limit of the data. However, the frequency of a DDR clock for the same interface is only *b*/2 MHz, consistent with the frequency spectrum of the data. Therefore, the same I/O drivers and receivers can be used to drive and receive both the data and the DDR clock.

Regardless of whether the high-speed clock is a SDR or a DDR clock, the receiving chip uses this clock to directly capture the data. This chip also uses the reference clock to generate an internal system clock at the same frequency. These clocks are mesochronous. While the frequency is the same (given that they share a common frequency reference), the phase relationship between the clocks is unknown and may vary due to PVT variations. Therefore, the receiving chip usually retimes the received data from the interface clock domain to the clock domain of the internal chip clock. FIFOs are used to perform this retiming function. It is desirable to minimize the number of flip-flops being clocked by the interface clock in order to minimize delay in the clock distribution network; otherwise timing issues will be exacerbated.

1.2.3 Higher Speed Source Synchronous Interfaces

The window of time during which data bits can be assumed to be valid is called the *eye*. This name originates from the shape of the waveform when the data signal is monitored on an oscilloscope that is continuously triggered. An example of a serial data eye is shown in Fig. 1.5a. *Eye closure* results from process, voltage, and temperature effects, as well as differences between signal rise and fall times, slew rates, etc. The more the eye is closed, the more difficult it is to find a point at which the signal can be reliability sampled to receive the data. The serial data eye shown in Fig. 1.5b is completely closed.

The largest possible *eye opening* is desirable. The width and height of an open eye can be measured as shown in Fig. 1.5c. The expected bit error rate (BER) of the link directly correlates to the amount of eye opening (both width and height). This section briefly describes some approaches to minimize eye closure of the data signal. Eye waveforms are discussed further in later chapters.

Marvell Semiconductor, Inc. - Ex. 1008, Page 000014

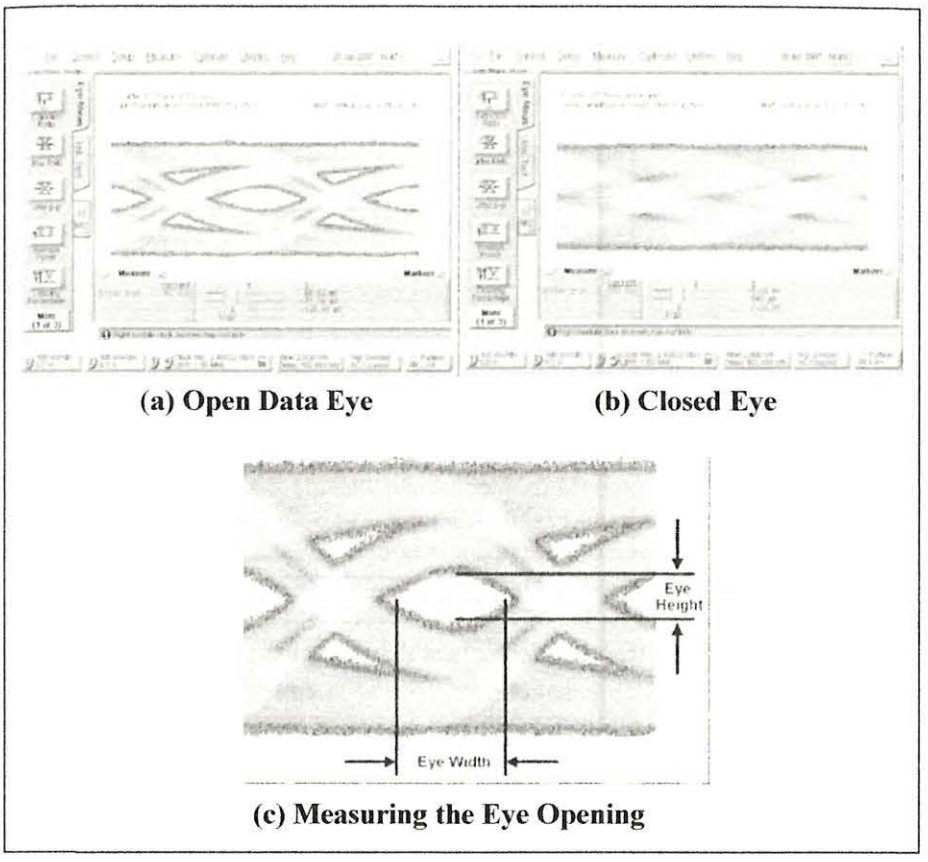


Fig. 1.5 Example of a serial data eye

1.2.3.1 Differential Signals

Unequal signal rise and fall times of nondifferential signals contribute to eye closure. Signals switching on the chip also create current variations on the power distribution grid of the chip, which in turn cause variations in voltage drop (noise) that can cause variation in delays of surrounding circuits. One method of reducing the effects of these phenomenon on the eye width is to drive differential signals between chips.

Differential signals represent the data bit using two electrical signals (*true* and *complement* signals). A logic "0" is represented by the *true* signal driven to its lower voltage limit, and the *complement* signal driven to its upper voltage limit; a logic "1" is represented by the *true* signal driven to its upper voltage limit, and the *complement* signal driven to its lower voltage limit. A differential receiver device interprets the logic bit value based on the difference between the two signals, and not based on the level of either signal individually.

Marvell Semiconductor, Inc. - Ex. 1008, Page 000015

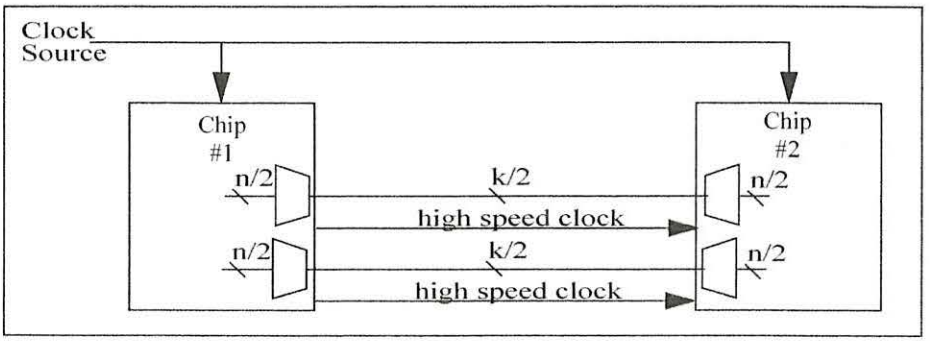


Fig. 1.6 Multiple sets of data with separate high-speed clocks

Differential driver circuits tend to have linear current draw and generate less noise on the power supply than equivalent single-ended drivers. Most noise sources induce voltage variation equally on both the true and complement signals; such *common mode noise* is ignored by the receiver. Also, since one leg of the differential signal is rising while the other is falling, or vice versa, unequal rise and fall time effects cancel.

The drawback of differential signals is that two chip pins are required for each data bit. However, this is offset by the higher speeds possible with differential signals that are not possible with single-ended signals.

1.2.3.2 Multiple Interface Clocks

The interface clock in Fig. 1.3 is the same clock as is used to launch the data, and in general is driven from a point in the clock distribution network as close to the actual flip-flops that launch the data as possible. Phase variation is introduced by any circuits which are not common to both the data path and clock path. Silicon process variables do vary from circuit to circuit on the same chip, the power distribution network may have unequal voltage drops to different circuits which may vary based on switching currents, and the temperature may vary from point to point on the chip. Tolerances and limits for all of these parameters must be taken into account when calculating delays, setup times, and hold times necessary for correct capture of the received data. At higher bit/baud rates, these parameters may significantly reduce the eye opening, and become the dominant mechanism for limiting the speed of the interface.

To maximize the eye width, the path through the clock tree to each of the data flip-flops and to the clock output should share as many circuits as possible, and the output driver for the clock should be similar to the output drivers for the data. Ideally, the same clock buffer should drive the clock to the output driver and should drive the clock input to all of the data flip-flops. The larger the number of bits in the data bus, the more difficult this becomes to implement. I/O drivers must be physically distributed based on the groundrules for connections to package pins. The greater the distance between circuits, the more process, voltage, and temperature variation, and the more circuits in the clock distribution network which cannot be shared due to lack of proximity.

Marvell Semiconductor, Inc. - Ex. 1008, Page 000016

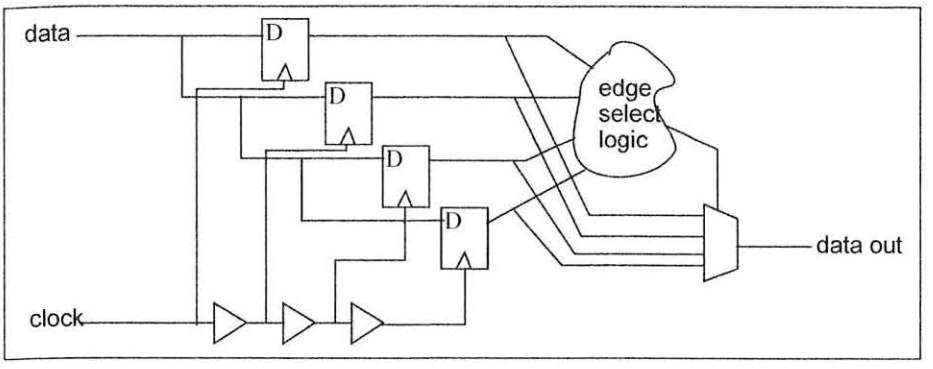


Fig. 1.7 Adapting the sampling clock phase in the receiver

One technique used to improve eye width is to limit the number of data bits associated with a given interface clock line. Wider data busses are built by using multiple interface clocks, each clock associated with a subset of the data bits. An example of this is shown in Fig. 1.6, where the *k* bit interconnect has been subdivided into two groups, each with its own high-speed interface clock. Note that the receiving chip must capture each group of data bits in separate clock domains, and needs to retime this data to the common clock domain internal to the chip.

1.2.3.3 Sample Edge Adaptation

Another technique used to permit higher speed operation of source synchronous interfaces is to process the data signal at the receiver and adapt the sampling phase of the clock on a per-bit basis. This is done by connecting the received interface clock signal to the input of a multitap delay line, and capturing the data signal in multiple flip-flops clocked by different clock phases. Logic can then be used to determine the clock phases between which data transitions are occurring, and select the optimal clock phase to be used to capture the data. This scheme is shown in Fig. 1.7.

Schemes, such as shown in Fig. 1.7, may require a training pattern either upon initialization of the interface or at regular intervals. If a training pattern is used, phase selections remain static between training periods. More complex implementations adjust dynamically based on the received data or based on training patterns embedded in the data stream. Alternative architectures which apply the data to the delay line are also possible. Note, however, that an inherent characteristic of most of these schemes is that the phase adjustment is less than plus/minus one bit time, and there must be a sufficient eye opening such that an optimal sampling phase exists.

Given the advanced schemes discussed above, data rates for source synchronous interfaces can be extended to several Gigabits per second (Gbps) per interconnect bit. However, PVT variations make further increases in interface speeds prohibitively complex. Beyond these speeds, High-Speed Serdes devices that extract the clock from edge transitions in the data stream become the preferred solution.

Marvell Semiconductor, Inc. - Ex. 1008, Page 000017

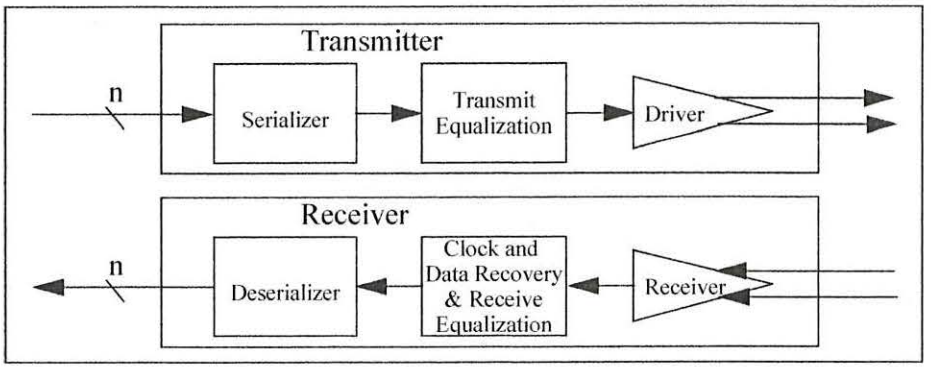


Fig. 1.8 Basic block diagram of typical high-speed serdes

1.3 High-Speed Serdes

High-Speed Serializer/Deserializer (HSS) devices are the dominant implementation of I/O interfaces at speeds of 2.5 Gbps and higher. Such devices are differentiated from source–synchronous interfaces in that the receiver device contains a clock and data recovery (CDR) circuit which dynamically determines the optimal sampling point of the data signal based upon the transition edges of the signal. In other words, clock information is extracted directly from the data rather than relying on a separate clock.

Figure 1.8 illustrates the basic block diagram of the transmit and receive channels of an HSS device. The transmitter serializes parallel data, equalizes it for reasons that will be explained shortly, and then drives the serial data onto a differential signal pair of interconnect wires. Feed forward equalizers (FFE) are commonly used in High-Speed Serdes devices, as discussed in Sect. 1.3.2. The receiver consists of a differential receiver, a CDR circuit which may also integrate an equalizer, and deserializes the data based upon the sample point established by the CDR. Peaking amplifiers and/or decision feedback equalizers (DFE) are commonly used for equalization in High-Speed Serdes receiver devices.

Note that Serdes cores are often designed to group multiple transmit and/or receive channels into a single device. The individual channels generally operate independently. Grouping channels allow some circuits to be shared across channels (for example the PLL noted below), and therefore the resulting block is more efficient in terms of chip area, cost, and power.

Serdes cores which contain only transmit or only receive channels are called *simplex* cores; Serdes cores which contain both transmit and receive channels are called *full duplex* cores. Note that the terminology "full duplex" does not imply that the electrical interface is bidirectional. Any given electrical interconnect channel has a fixed direction of data transmission. If a protocol application requires "full duplex" communication, then independent transmit and receive channels with independent interconnections are used to implement the interface. Rationale for using simplex vs. full duplex cores may include

Marvell Semiconductor, Inc. - Ex. 1008, Page 000018

(1) chip floorplan to minimize wiring crossings in the package design or circuit board design; (2) signal integrity concerns due to near-end crosstalk from transmit signals onto receive signals; (3) or applications where the number of transmit and receive channels is not equal.

The remainder of this section generically describes various circuits mentioned above in more detail, as well as providing generic descriptions of other circuits and functions commonly found in High-Speed Serdes cores.

1.3.1 Serializer/Deserializer Blocks

Conceptually, the input to the serializer transmit stage is an *n*-bit datapath which is serialized to a one-bit serial data signal for application to the FFE and Driver stages. Generally the value of *n* is a multiple of 8 or 10, and may be programmable on some implementations. Values of *n* which are multiples of 8 are useful for sending unencoded and/or scrambled data bytes; values of *n* which are multiples of 10 are useful for protocols which use 8B/10B coding, as discussed further in Sect. 4.2.2.1. (The 8B/10B encoder is generally implemented by logic outside the Serdes core.)

For simplicity, the block diagram in Fig. 1.8 illustrates the serializer feeding one-bit data into the transmit equalization block. Actual implementations may vary, and this datapath may be one or more bits wide. A wider datapath through the equalizer block results in a more complex design, but requires a lower operating frequency. Some implementations may initially multiplex the *n*-bit input to an *m*-bit datapath ($m < n$) prior to the equalizer, and perform the remainder of the serialization at the driver stage.

The serializer stage latches data on the *n*-bit input at the frequency of *baud rate*/*n*. The high-speed clock in the Serdes is divided down to generate a sample clock for the parallel data. Because the phase of this clock is determined by the internal state of the serializer, the Serdes channel generally provides this clock as an output for use by logic driving data to the transmit channel.

Conceptually, the deserializer receive block performs the inverse function of the serializer block. Serial data is deserialized onto an *n*-bit databus of similar width to the serializer. A sample clock is generated by dividing down the internal high-speed clock, and this clock is supplied as an output for use by logic latching the parallel data. In a similar manner to the serializer, actual implementations may perform partial deserialization in a prior stage.

Many Serdes receivers also include a feature to assist with data alignment of the output. Most applications organize data into bytes or words (groups of bytes). For 8B/10B encoded applications, data is organized into 10-bit encoded symbols. The initialization of the clock divider in the deserializer is arbitrary, and the data received on the parallel data bus will have an arbitrary alignment that is unlikely to match the byte or symbol boundaries of the protocol. This can be corrected by downstream logic to steer data onto the appropriate byte, symbol, or word boundary. Alternatively, many Serdes receivers provide an input which forces the deserializer to "slip" one bit. Downstream logic detects

Marvell Semiconductor, Inc. - Ex. 1008, Page 000019

that data is not aligned to the appropriate boundary, and repeatedly pulses the deserializer control until the data "slips" to the desired alignment.

1.3.2 Equalizers

The interconnect between the transmitter and receiver device (known as the *channel*) acts as a filter at typical baud rates, and distorts the serial data signal to varying extents. Figure 1.9 illustrates this distortion: The input waveform is a clean digital signal, but the output waveform is significantly distorted. The illustrated frequency response function for the channel is characteristic of a low-pass filter. Signal distortion occurs because the signal baud rate is above the cut-off frequency for this filter.

Signal integrity concerns frequently dictate that the data signal be equalized at the transmitter and/or receiver in order to counter the effects of the channel and decode the signal properly. Many variations on filter architectures are possible, all of which accomplish this. Fig. 1.10 illustrates the addition of an equalizer at the transmitter with a transfer function that is roughly the inverse of the channel's frequency response. This equalizer distorts the signal at the transmitter output such that the resulting signal at the receiver input is a clean waveform.

Most Serdes transmitter implementations include a FFE. The block diagram for a three-tap FFE is shown in Fig. 1.11. The serial data signal is delayed by several flip-flops which implement the *taps* for the filter. Each tap is multiplied by a *tap weight* value (also called a *filter coefficient*), and the results are summed and driven to the serial data output. FFE operation is described further in Sect. 3.2.1.

The number of FFE taps on the filter, the spacing of these taps relative to the baud rate, and the granularity of these tap weight values vary based on implementation. The terminology *preemphasis* or *deemphasis* refer to the FFE architecture, and indicate whether the data signal amplitude is increased or decreased as compared to the nonemphasized value by the FFE tap. The terminology *precursor taps* and *postcursor taps* refer to whether the FFE filter taps operate on an advanced or delayed signal (respectively) relative to the $t = 0$ tap. *Baud-spaced* taps are defined as taps where the delay from one filter tap to the adjacent tap is one-bit time interval; fractional spacing of the taps is also possible.

The FFE tap weights are selected to generate a filter with the inverse transfer function of the channel transfer function. Various algorithms exist for determining optimal FFE coefficient values; some select filter coefficients to maximize signal amplitude at the receiver, while others optimize eye width (i.e., minimizing jitter). More complex algorithms may search for an optimal trade-off between amplitude and jitter in order to optimize a more complex parameter (such as projected BER). FFE tap weights are determined for many applications by design and coded as fixed values within system software, however, there are some applications where FFE tap

Marvell Semiconductor, Inc. - Ex. 1008, Page 000020

weights are adjusted dynamically by the protocol based on signal characteristics at the receiver.

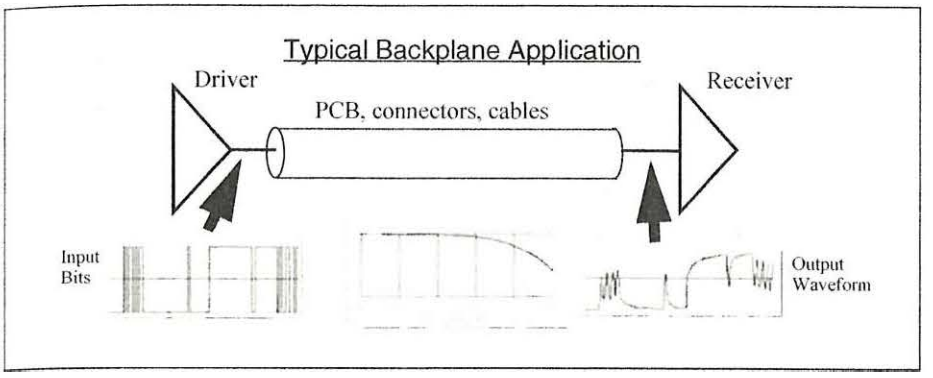


Fig. 1.9 Signal distortion for a typical channel application

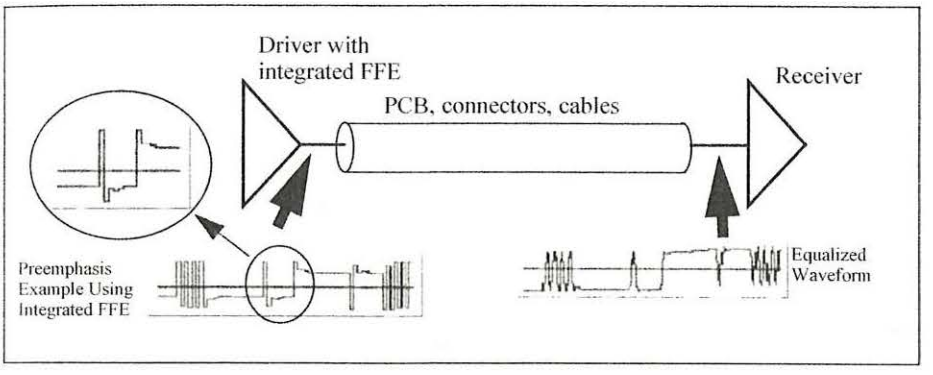


Fig. 1.10 Typical channel application with equalization at the transmitter

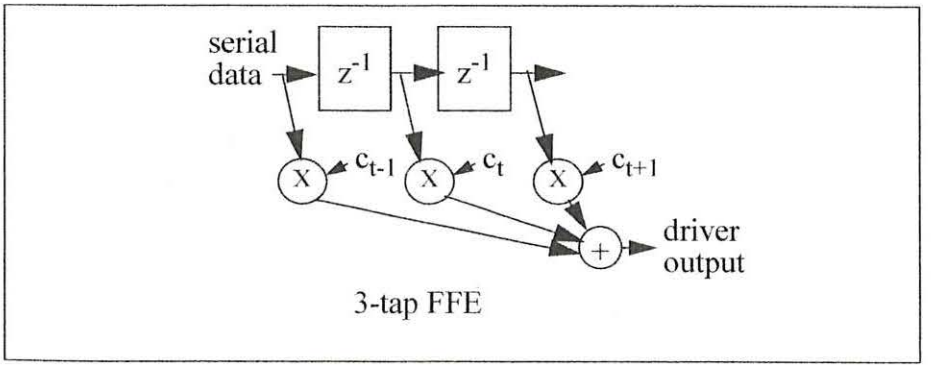


Fig. 1.11 Three-tap feed forward equalizer operation

Marvell Semiconductor, Inc. - Ex. 1008, Page 000021

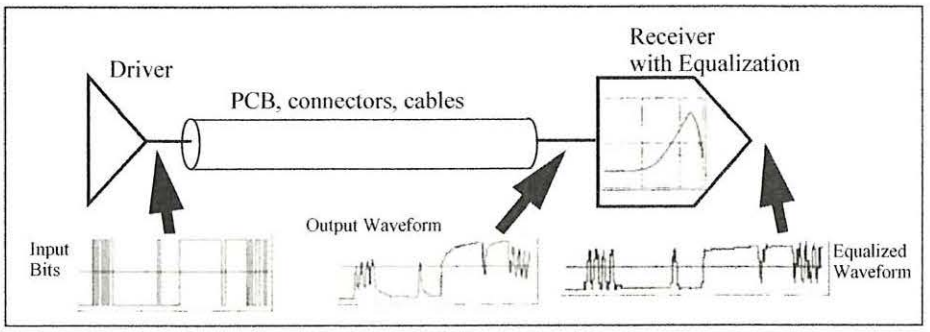


Fig. 1.12 Typical channel application with equalization at the receiver

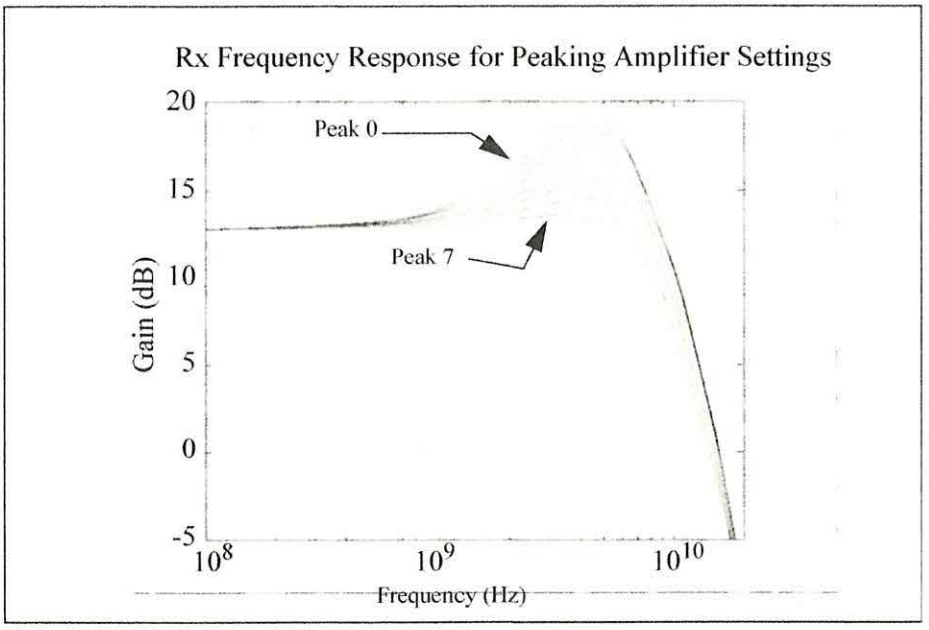


Fig. 1.13 Receiver frequency response for peaking amplifier settings

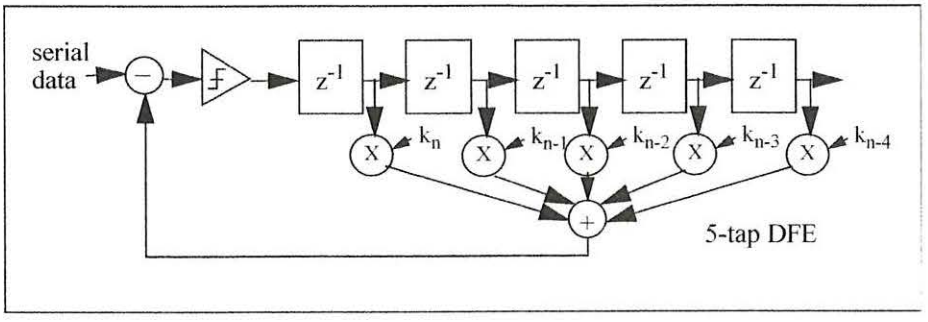


Fig. 1.14 Decision feedback equalizer architecture

Marvell Semiconductor, Inc. - Ex. 1008, Page 000022

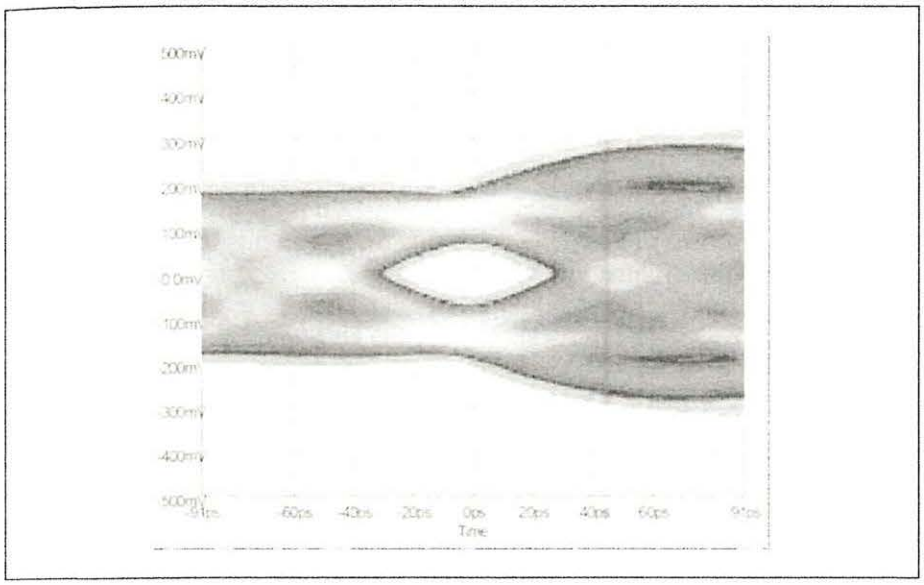


Fig. 1.15 Virtual eye after equalization

Equalization may also be performed at the receiver as illustrated in Fig. 1.12. Despite the signal distortion at the input of the receiver, this equalizer corrects for the distortion and produces a clean waveform. For lower speed or lower loss links, the most prevalent approach is to use some variant of a peaking amplifier. Peaking amplifier circuits amplify the higher frequency signal components more than the lower frequency components. If the peaking amount is matched to the high frequency loss (difference between high frequency and low frequency), then the channel is equalized and the eye is opened up. Some Serdes devices allow programmable peaking levels; the frequency response of such a peaking amplifier for various provisioned settings is shown in Fig. 1.13.

For higher baud rates, the transfer function of the channel can cause jitter exceeding the bit width of the data and significant loss of signal amplitude at higher frequencies. A DFE stage is often included in receivers for these baud rates in order to recover data despite the otherwise "closed" eye.

A conceptual block diagram of a DFE circuit is shown in Fig. 1.14. The serial data signal is applied to a slicer circuit which makes decisions as to whether the incoming signal is a "0" or a "1". The received serial data is then delayed by a number of flip-flops which implement the filter taps. Each tap is multiplied by a corresponding tap weight value, and the results are summed. This sum is then used to correct the amplitude of the incoming signal, affecting the decisions made by the slicer circuit. Slicer decisions are thus affected by feedback based on prior data received. Although not shown in Fig. 1.14, some DFE architectures use feedback to affect both the amplitude and the sample

Marvell Semiconductor, Inc. - Ex. 1008, Page 000023

time of the slicer circuit. Such architectures adjust the CDR sample time from bit to bit based on feedback regarding the last several bits received.

As was the case for FFE circuits, the number of DFE taps on the filter, the spacing of these taps relative to the baud rate, and the granularity of these tap weight values vary. DFE implementations usually also contain logic which trains the DFE and sets DFE tap weights to optimal values dynamically.

Using a DFE, the closed eye shown in Fig. 1.5b is cleaned up to produce the virtual eye shown in Fig. 1.15. Note that the eye in this illustration is produced by a DFE architecture which corrects the CDR sample time from bit to bit. The DFE correction is valid for only one instance in time (based on the history of the previous bits). As such, once the DFE makes a decision as to whether the bit is "0" or "1", the DFE then proceeds to make adjustments for the next bit time which are different for the various signal traces of the composite waveform. For this reason, the signal eye shown in the figure is open for the bit of interest, but does not appear open for adjacent bits.

Many variations on equalizer architectures exist. As the baud rate increases, equalizer architectures become increasing complex. In some cases, protocol standards specify a base level of required equalizer functionality.

1.3.3 Clock and Data Recovery (CDR)

Conceptually, CDR circuits monitor transitions of the data signal and select an optimal sampling phase for the data at the mid-point between edges. Since the timing of data transitions includes a jitter component, the CDR must perform some averaging to provide stability of this sampling point from one bit to the next. Intersymbol interference (ISI) and other components of deterministic jitter (DJ) are dependent on the spectral content of the data signal, and this frequency spectrum does change based on the data content. Shifts in this frequency spectrum sustained for hundreds of bits or more cause the CDR to adjust the optimal sampling phase dynamically.

CDR architecture is discussed further in Sect. 3.3.1. Features of the CDR may be of some significance to the Serdes user are discussed below.

1.3.3.1 Maximum Run Length

A significant parameter for the Serdes which is primarily the result of the CDR design is the maximum number of consecutive "0" or "1" bits which can be received before the sampling point of the CDR risks incorrectly sampling the bits. An excessively long run of consecutive bits of the same value means that the CDR is not detecting any data transitions, and therefore cannot recover any clock information to ensure the data continues to be sampled in the center of the eye. A small drift in the sampling point relative to the baud rate of the data may cause the CDR to sample more "0" or more "1" bits than were actually transmitted. Also, the sampling point may require recentering when data transitions resume, and additional bits may be sampled incorrectly as this adjustment occurs. Some CDR implementations drive the receive data to a PLL and use the output of the PLL as the sample clock; clock outputs of the

Marvell Semiconductor, Inc. - Ex. 1008, Page 000024

receiver may change frequency or stop when such CDRs do not receive data transitions for a sustained length of time.

The maximum run length of consecutive "0" or "1" bits which must be tolerated depends on the protocol application and the data encoding defined for a given protocol. For example, protocols which use 8B/10B encoding are guaranteed to have no more than 5 bit times between data transitions. Protocols using another common encoding, 64B/66B, are guaranteed to see run lengths no longer than 66 bit times. Scrambled protocols may encounter much longer run lengths, and must determine requirements using statistical analysis. For example, Sonet/SDH is a scrambled protocol which specifies systems must meet a BER of 1×10^{-12}. It is generally accepted that run lengths of scrambled Sonet/SDH data longer than 80 bits statistically occur less frequently than the specified BER. Therefore, a Serdes used to receive Sonet/SDH data must tolerate a run length of 80 bits.

The run length which can be tolerated by a CDR design is related to the frequency tolerance between the two clock sources. In a system using plesiosynchronous clocks, the reference clock used by the receiver (and the CDR circuit) may be running at a slightly different frequency from the reference clock used by the transmitter, as is described further in Sect. 4.1.3.1. The frequency tolerance between the two clock sources is generally specified in parts per million (ppm). In a plesiosynchronous system, the CDR must continually correct the phase of the sample clock to remain in the center of the data eye. During periods where no data transitions are being received, the error in phase position builds up. Therefore, as the frequency tolerance of the system is increased (corresponding to larger allowed frequency difference between the clock sources), the run length which can be tolerated by the CDR design is reduced for a given performance (BER) target.

1.3.3.2 Clock Operation During System Initialization

In the above discussion, it was noted that some CDR architectures derive the sample clock from the received data using a PLL. During system initialization or during system operation when cables are unplugged, etc., no data transitions are received for a substantial period of time. For some Serdes, this results in clock outputs of the receiver changing frequency or stopping. Any downstream logic clocked by these clock outputs must be designed to be tolerant of this frequency change or to assume logic is not clocked during these periods.

1.3.4 Differential Driver

The differential driver stage is an analog circuit which drives the *true* and *complement* legs of the differential signal. Output data must be driven such that jitter is minimized. In some architectures, data is latched in a flip-flop clocked at the baud rate, and the output of this flop is driven onto the differential output. Such implementations require an internal high-speed clock running at the baud rate. This is illustrated in Fig. 1.16.

Marvell Semiconductor, Inc. - Ex. 1008, Page 000025

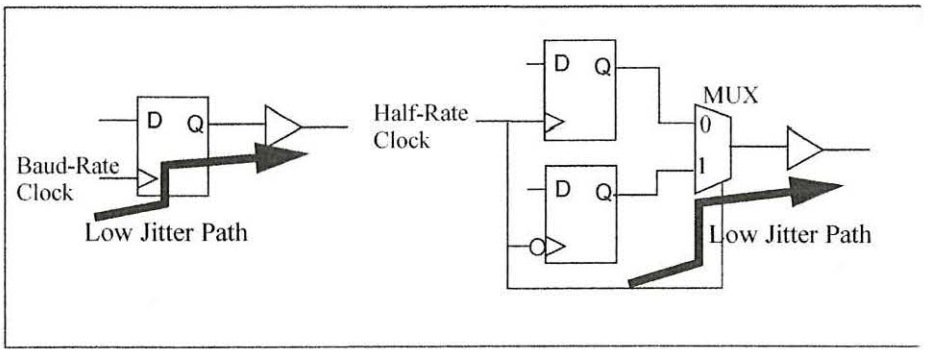


Fig. 1.16 Driver stage architectures

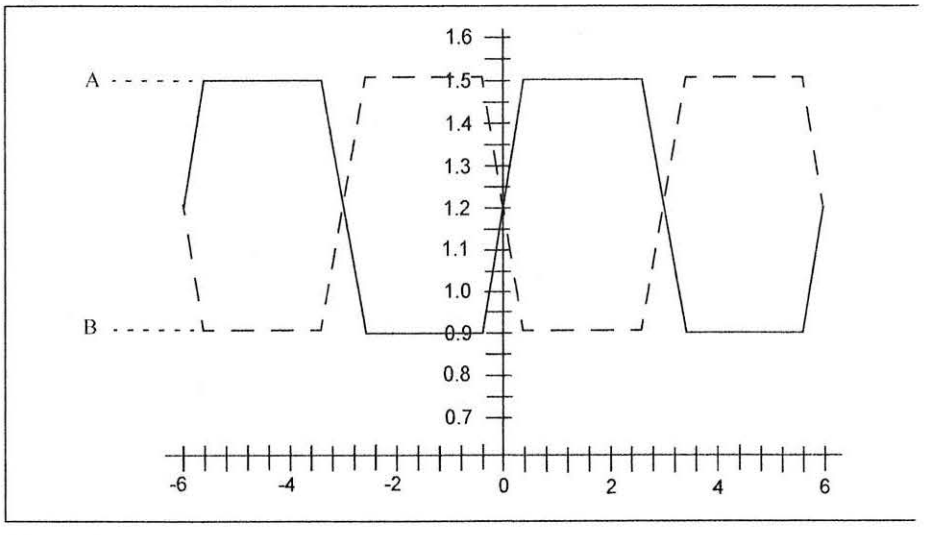


Fig. 1.17 Single-ended complementary signals

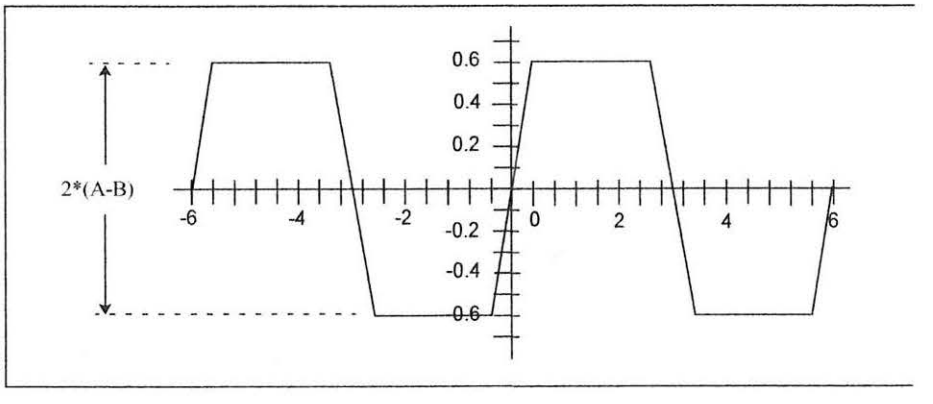


Fig. 1.18 Differential peak-to-peak signal

Marvell Semiconductor, Inc. - Ex. 1008, Page 000026

An alternative architecture, also shown in Fig. 1.16, uses an internal high-speed clock running at a frequency equal to half of the baud rate. Data is latched in two flip-flops on alternate edges of the high-speed clock. The high-speed clock also controls a multiplexor which alternately selects which of the flops drives the differential driver. Depending on the characteristics of the silicon technology, this architecture may result in lower jitter than the full-rate architecture.

Figure 1.17 illustrates typical voltage swings for the two legs of the differential signal, assuming a termination voltage of approximately 1.8 V. The average voltage on the signal is the *common mode voltage* (V_{cm}). For this example:

$V_{cm} = (1.5\,V + 0.9\,V) / 2 = 1.2\,V.$

The *differential voltage* (V_{diff}) is calculated by taking the voltage of the true leg and subtracting the voltage of the complement leg. Figure 1.18 illustrates the differential waveform corresponding to the single-ended signals from Fig. 1.17.

This differential voltage swings between the following limits:

$V_{diff} = 1.5\,V - \ \ 0.9\,V = +0.6\,V$

$V_{diff} = 0.9\,V - \ \ 1.5\,V = \ \ 0.6\,V$

This waveform has a total *peak-to-peak differential voltage* of $1.2\,V_{ppd}$. Note that the peak-to-peak voltage of the differential signal is twice the peak-to-peak voltage of either single-ended signal considered individually.

1.3.5 Differential Receiver

The differential receiver stage is an analog comparator circuit which compares the *true* and *complement* legs of the differential signal and outputs a "0" or "1" logic level based on the relative signal voltages. Differential receiver stages used with DFEs are linear amplifiers; the comparator circuit is incorporated into the DFE.

1.3.6 Diagnostic Functions

Additional logic is often incorporated into the transmitter and receiver designs to provide diagnostic capabilities for chip manufacturing test, circuit board manufacturing test, and system diagnostic tests. Typical functions include:

1. Pseudo random Bit Sequence (PRBS) Checker. PRBS sequences can be checked by comparing received data to the output of a local linear feedback shift register implementing the corresponding characteristic polynomial. Receiver devices often include a PRBS checker capable of checking one or more PRBS test patterns.

2. Loopback or Wrap Paths. Full duplex Serdes devices often provide the capability to wrap transmitter outputs to receiver inputs in order to self-check the functionality of the Serdes. Simplex cores do not have this capability,

Marvell Semiconductor, Inc. - Ex. 1008, Page 000027

although some simplex transmitters include a test receiver, and some simplex receivers include a test transmitter to perform self-test.

3. JTAG 1149.1 and JTAG 1149.6. These JTAG standards are used for manufacturing test of circuit boards, and require insertion of boundary scan cells on all chip I/O to support this testing. Since such logic cannot be inserted on high-speed I/O without impacting signal integrity, the Serdes core must provide appropriate hooks to drive differential outputs from boundary scan cells at the transmitter device, and sample inputs in boundary scan cells at the receiver device. JTAG 1149.6 expands the capabilities of JTAG 1149.1 to permit testing through decoupling capacitors and support independent testing of the true and complement legs of differential signals. JTAG 1149.1 and 1149.6 are covered in detail in Sect. 7.1.

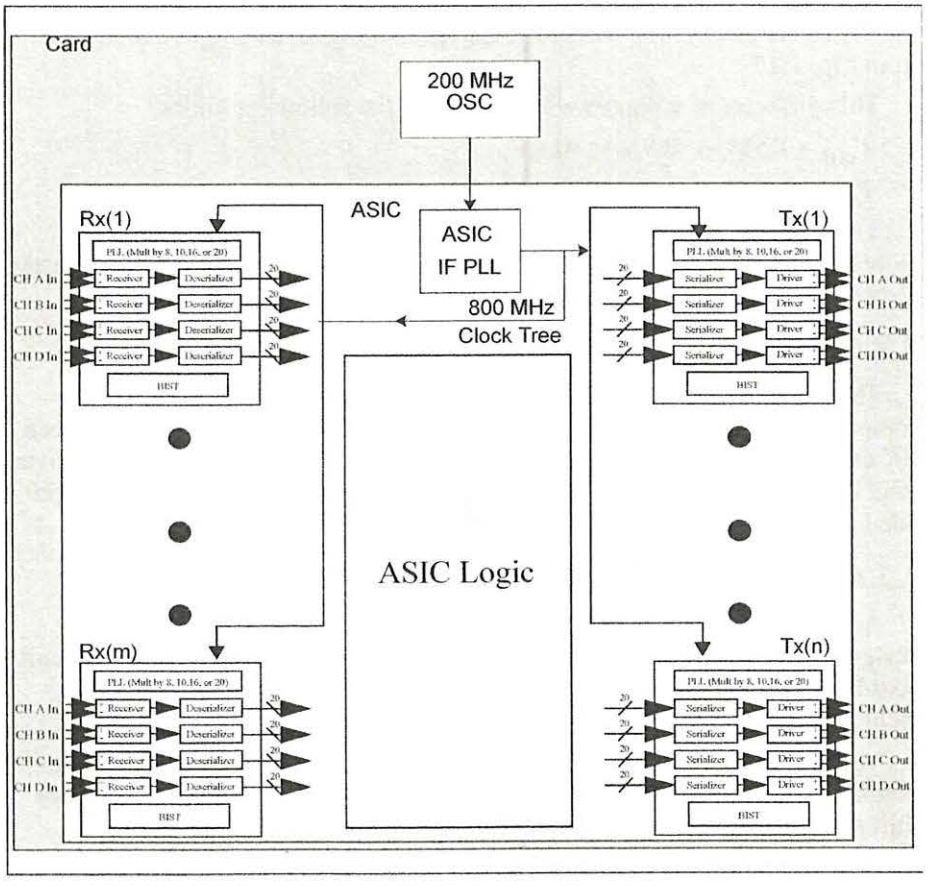


Fig. 1.19 Clock distribution example using an ASIC IF PLL

Marvell Semiconductor, Inc. - Ex. 1008, Page 000028

1.3.7 Phase-Locked Loop

The Serdes core requires an internal clock running at either the baud or half-baud rate depending on the architecture of the driver and receiver stages. Rather than distribute a high-speed clock throughout the chip, a lower frequency reference clock is distributed within the chip, and a PLL is used in the Serdes to multiply this clock to the appropriate frequency. A given Serdes implementation may contain multiple instances of transmitter and/or receiver channels. In such cases, it is common for a single PLL in the core to generate clocks for all channels within the core.

The off-chip clock source often operates at an even lower frequency than the on-chip reference clock. An additional PLL may be used in the chip to multiply the frequency of the off-chip reference clock to meet the desired on-chip reference clock frequency. Because the frequency of the on-chip reference clock is usually higher than the off-chip reference clock (but less than the internal clock in the Serdes core) the PLL which produces this clock is sometimes called an *intermediate frequency* (IF) *PLL*.

An example of clock distribution using such an IF PLL is shown in Fig. 1.19. An IF PLL is used to multiply the 200-MHz clock from an off-chip oscillator by four. The resulting 800-MHz reference clock is distributed on-chip to various Serdes cores. These Serdes cores each contain a PLL which additionally steps up the frequency of the 800-MHz reference clock to the desired baud rate.

1.4 Signal Integrity

This section provides an overview of the importance of signal integrity analysis to the design of successful systems using High-Speed Serdes.

1.4.1 The Channel

The *channel* is defined as the electrical path between the transmitter and the receiver, including printed circuit board traces, vias, cables, connectors, decoupling capacitors, etc. The channel may traverse the printed circuit board between two chips on the same card, or may traverse a system backplane connecting two printed circuit boards.

At frequencies of interest, the printed circuit board is not a perfect connection. Major channel impairments include insertion loss, reflections, and crosstalk. Channel frequency response, including these impairments, is typically measured using a vector network analyzer (VNA), and captured in a *Scattering Parameter* matrix format (commonly called *S-Parameters*) as is described in "Channel Response" under Sect. 8.4.1.2 in Chap. 8. Each of these impairments impacts the BER of the link. Interface standards typically require link BER performance in the range of 10^{-12} to 10^{-15}.

The equalization scheme used by Serdes devices must compensate for the channel loss and other impairments in order to achieve the desired BER. A common figure of merit is based on the evaluation of the insertion loss of the

Marvell Semiconductor, Inc. - Ex. 1008, Page 000029

channel at the Nyquist rate (two times the highest fundamental frequency of the signal). From this metric, a qualitative assessment as to the difficulty of signal propagation (and the necessary complexity of the equalization scheme) can be made.

Figure 1.20 illustrates the measured insertion loss curve for a number of examples of channels intended to support baud rates of 5 Gbps and higher. The frequency range of 5-Gbps data has a Nyquist rate of 5 GHz, and the loss of these channels at 5 GHz is substantially higher than at lower frequencies. Higher frequency components are therefore attenuated more than lower frequencies, resulting in varied signal amplitudes at the receiver. (In addition, the signal contains harmonic frequencies above 5 GHz, but these are typically filtered and are not critical to receiver operation.)

The channel transfer function is not strictly resistive, but also contains capacitive and inductive components. This results in frequency-dependent phase shift of the propagated signal. Such phase shift in effect causes the propagation delay of the signal to vary based on frequency, appearing as data-pattern-dependent jitter (discussed later in this section).

Although the above effects already contribute to significant signal degradation, the insertion loss and phase shift associated with the channel is usually not a straightforward linear function. The channel is an electrical transmission line which is terminated at the receiver, and has impedance discontinuities at each circuit board via, connector pin and abrupt bend in the circuit board trace. Each impedance discontinuity results in reflections of the electrical energy. As with any transmission line, reflected energy adds to or subtracts from the signal amplitude at various points along the transmission line, and results in resonances in the transfer function. This signal degradation is generally worse for shorter channel lengths; loss characteristics of longer channels tend to dampen reflections whereas signals may reflect between the transmitter and receiver multiple times on a short channel.

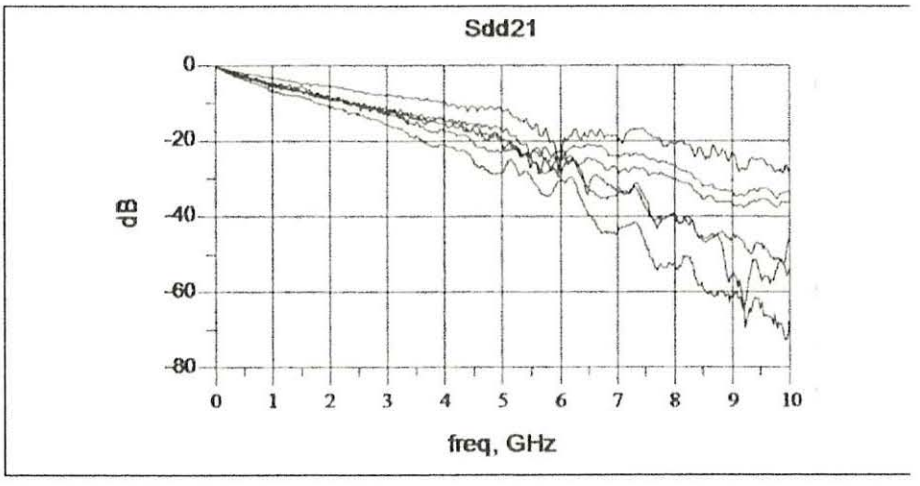


Fig. 1.20 Insertion loss for various channel examples

Marvell Semiconductor, Inc. - Ex. 1008, Page 000030

Crosstalk is energy coupled from an aggressor signal as noise on a victim signal, and is another significant characteristic of the channel which causes significant degradation of high-speed serial signals. While it would be nice to isolate each differential signal pair such that crosstalk was not significant, this is not practical in many real systems. The economics of most systems demand signal densities through package pins, connectors, and backplanes which result in neighboring signals causing crosstalk on the differential pair. Common-mode noise is ignored, but any differences in the noise on the two legs of the differential pair results in signal degradation.

1.4.2 Package Models

In addition to the channel as described above, the differential signal must also propagate through the chip package for the transmitting chip and the chip package for the receiving chip. While it would be convenient from an analysis viewpoint to consider chip packages as part of the channel interconnect, it is not practical to access the connections inside the package in order to measure channel response in the lab. For this reason, channel measurements are performed from package pads as described previously, and the transfer function of the package is considered separately. *Package models* are supplied by the chip manufacturer which model the transfer function of the package. Analysis of the overall interconnect is performed by cascading the transmitter package model, the channel transfer function, and the receiver package model.

Since trace lengths in the package substrate tend to be very short, insertion loss and phase shift are usually not the dominant source of signal degradation due to the package. Impedance mismatches and discontinuities tend to be a far greater concern. A measurement of the returned energy from a signal launched into the package ball at either the transmitter or the receiver is called *return loss*. Better impedance matching of connections to the silicon device within the package results in better return loss and less degradation of the signal.

1.4.3 Jitter

In an ideal world data bits would always transition at a fixed point relative to the clock which launches each bit. In the real world there are variations in the clock cycle time, data propagation delays, and signal slew times, etc.Variation in the timing of the bit transition relative to an ideal clock is called *jitter*. Jitter reduces the width of the data eye.

Any transmitter device has some amount of jitter in the launched data bit. *Jitter generation* is a measure of the timing variation in the transmitted data stream. Applications may include specifications for the maximum allowable jitter generation by a transmitter device, as measured under specified test conditions. Test conditions may include specification of test patterns to be used, and the load to which the transmitter device is connected.

The jitter generated by the transmitter device is amplified by the channel. The channel distorts the signal and introduces frequency-dependent phase

Marvell Semiconductor, Inc. - Ex. 1008, Page 000031

shift. As the data pattern of 1's and 0's shifts, the spectral content of the data changes. This results in varying delay which becomes additional jitter at the receiver. Crosstalk and reflections due to impedance discontinuities and return loss also contribute to shift the transition points of signal edges at the receiver.

The receiver must be able to tolerate the amount of jitter that occurs on its input. *Jitter tolerance* is a specification for the receiver device which defines the "worst case" signal that the receiver is expected to receive correctly with errors no greater than allowed by the specified BER. Applications may include specifications for how to test jitter tolerance compliance, including test patterns to be used and methods for generating a datastream with specified amounts of jitter.

Jitter and jitter tolerance may be specified in picoseconds, but are more often specified in terms of unit intervals (UI). Parameters are usually specified as either peak-to-peak or root-mean-square, depending on the type of jitter. One *Unit Interval* is the cycle time for the transmission of one bit on the interface at the speed at which the link is running. A jitter specification of 0.30 UIpp indicates that the total peak-to-peak jitter cannot exceed 30% of the bit time, leaving an eye opening of 70% of the bit time. A jitter tolerance specification of 0.70 UIpp indicates that the receiver cannot expect more than an eye opening of 30% of the bit time at its input.

Jitter transfer is the amount of the input jitter on the receiver input which is passed through to the output (also called *jitter gain*). This parameter is significant in applications which must retransmit data with the same timing as the receive data. Since each stage of retransmission contributes additional jitter to the overall system, a jitter transfer specification is required in such applications to allocate how much jitter can be added within any one stage.

Many standards subdivide jitter into various types, and may use varying terminology for the types of jitter. The following discussion describes a commonly used terminology, and is sufficient for the purposes of general understanding. More comprehensive definitions are found in Sect. 8.2.1:

1. Total Jitter (TJ). This is the total jitter of the signal as seen at the point of measurement. TJ can be measured directly on hardware and is the ideal bit time minus the actual eye width, specified as either a peak or peak-to-peak value.

2. Deterministic Jitter (DJ). This is the amount of the total jitter for which the jitter distribution is non-Gaussian. Several components of DJ are dependent on the data pattern being sent. The pattern of 1's and 0's which precedes the bit transition affects when the transition occurs. Jitter caused by variations in rise, fall, and slew times are also mostly deterministic. DJ is specified as either a peak or peak-to-peak value.

3. Random Jitter (RJ). This is the amount of the total jitter for which the jitter distribution is Gaussian. RJ does *not* correlate with the data pattern being sent. Some amount of jitter has nothing to do with the data pattern, and is simply the result of random processes. Because RJ is statistical in nature, it may be specified either as a peak, peak-to-peak , or as a root-mean-square value.

Marvell Semiconductor, Inc. - Ex. 1008, Page 000032

Generally interface standards specify two or three of the above jitter types. If two of the above types are specified (typically TJ and DJ, or TJ and RJ), then requirements for other type is implied. This approach allows trade-offs to be made by system and component designers, while still complying with the standard.

The rationale for specifying different types of jitter is that some of the jitter can be corrected. For instance, the FFE in the transmitter device alters the transitions of the data edges based on the pattern of 1's and 0's being sent. In effect, the FFE is used to inject DJ on the transmitter output. If the equalizer coefficients are set properly, this injected DJ is the opposite of the DJ that is created by the channel's transfer function. The result is that the two DJ contributors cancel and there is less jitter at the receiver.

The DFE in the receiver also operates to cancel the effects of DJ and thereby improve the jitter tolerance of the receiver. The DFE has the advantage of adapting to changing conditions of the channel, which may alter the characteristics of the DJ over time. On the other hand, RJ is not predictable, and therefore cannot be corrected by equalization. It is important to limit the amount of RJ on the serial link.

1.4.4 Channel Analysis Tools

At serial link speeds up to 4Gbps it is generally possible to follow reasonable design practices for the channel design. Spice™ or other equivalent circuit simulators are used to perform simulations at these baud rates. (In this text the term Spice is used generically as a designation for any of these simulators.) (Note: Spice is a trademark of Synopsys, Inc.) Spice simulations are used to verify signal integrity, and demonstrate that transmitter and receiver devices will interoperate across the channel. This approach is generally not sufficient at serial link speeds of 5Gbps and higher.

At higher serial links speeds, some care is required to design a channel which does not unduly degrade the signal. There are cases where small design changes can have substantial unexpected effects. Spice simulation can be used to simulate the channel design. However, remember that signal degradation is dependent on the data pattern and the crosstalk. Also, the more sophisticated DFE-based cores use complex algorithms to determine tap coefficients, etc., real time. This behavior must be accurately simulated to account for algorithmic errors and compensation techniques performed in logic. Spice simulation cannot capture this behavior and is therefore not generally used for the higher speed links. In addition, it is not sufficient to simulate a few thousand bits; simulations must be long enough and contain enough randomly generated data patterns on both the primary channel and the crosstalk channel to ensure the system operates within the specified BER limits. This may require a substantial amount of simulation.

Because of these simulation run-time requirements, statistical analysis of channel designs has become prevalent at higher speeds. First, the channel design is prototyped and lab measurements are made to determine the transfer function characteristics of the channel. Transfer functions are measured for both the channel of interest, and for noise coupling between the primary

Marvell Semiconductor, Inc. - Ex. 1008, Page 000033

channel and crosstalk channels. The resulting transfer functions are coded in a form called *S-Parameters*. Alternatively, S-Parameters for subcomponents of the channel (i.e., printed circuit boards, connectors, backplane, etc.) can be measured individually and cascaded.

Next, statistical simulation of the channel is performed using a transmitter model, the channel S-Parameters, package models for both transmitter and receiver chips, and a receiver model. The transmitter and receiver models may be models for specific vendor Serdes devices, or they may be ideal reference models specified by an interface standard. The resulting "eye" on the output of the receiver model is characterized by an eye width and height and is a function of BER. If the "eye" is sufficiently "open" for the desired BER, then the system will operate with no more than the specified number of bit errors.

The terminology "eye" is used loosely in the above paragraph, hence the quotation marks. The receiver device often contains a DFE. By convention, the "eye" is the signal at the sampling latch of a classical DFE design. However for some DFE designs this may not be a single node, or may be beyond the point where analog-to-digital conversion of the signal has been performed. The notion of an analog "eye" at this point is a virtual mathematical concept; this eye is not a measurable analog signal.

Note that signal integrity engineers often prefer to view analysis results in the form of a *bathtub curve* which graphs eye opening as a function of Q of the system (which relates to BER), as is described further in Sect. 8.2.4. The geometry of this curve results from the *dual-Dirac* probability density function, which models pattern-dependent jitter sources.

There are a number of implementations of tools that can be used to perform statistical signal integrity analysis. A number of EDA vendors supply software tools. Various silicon vendors have proprietary software to perform this analysis (i.e., IBM's HSSCDR software) which have built-in transmitter and receiver models for the vendor's Serdes. "StatEye" is an open source software tool which performs this analysis (see http://www.stateye.org). HSSCDR software is discussed in Sect. 8.4. Note that many of these software tools use MatLab™ as the underlying calculation engine.

1.5 Signaling Methods

This chapter has thus far assumed differential signals which use two signal levels that convey whether the bit being transmitted is a "0" or a "1". This is called *non-return-to-zero* (NRZ) signaling. (This name originally differentiated NRZ signaling from signals which always returned to their zero level between each transmitted bit.) The signal eye of an NRZ signal, shown in Fig. 1.5a, can be sliced at the midpoint of the waveform (0 mVppd) to determine whether each bit is a "0" or a "1". The baud rate of an NRZ signal is equivalent to the data bit rate (after any encoding), and the maximum fundamental frequency of the signal is half of this baud rate.

Multilevel signaling schemes are also possible. Figure 1.21 shows the signal eyes for *duobinary* and four-level *phase amplitude modulation* (PAM-4).

Marvell Semiconductor, Inc. - Ex. 1008, Page 000034

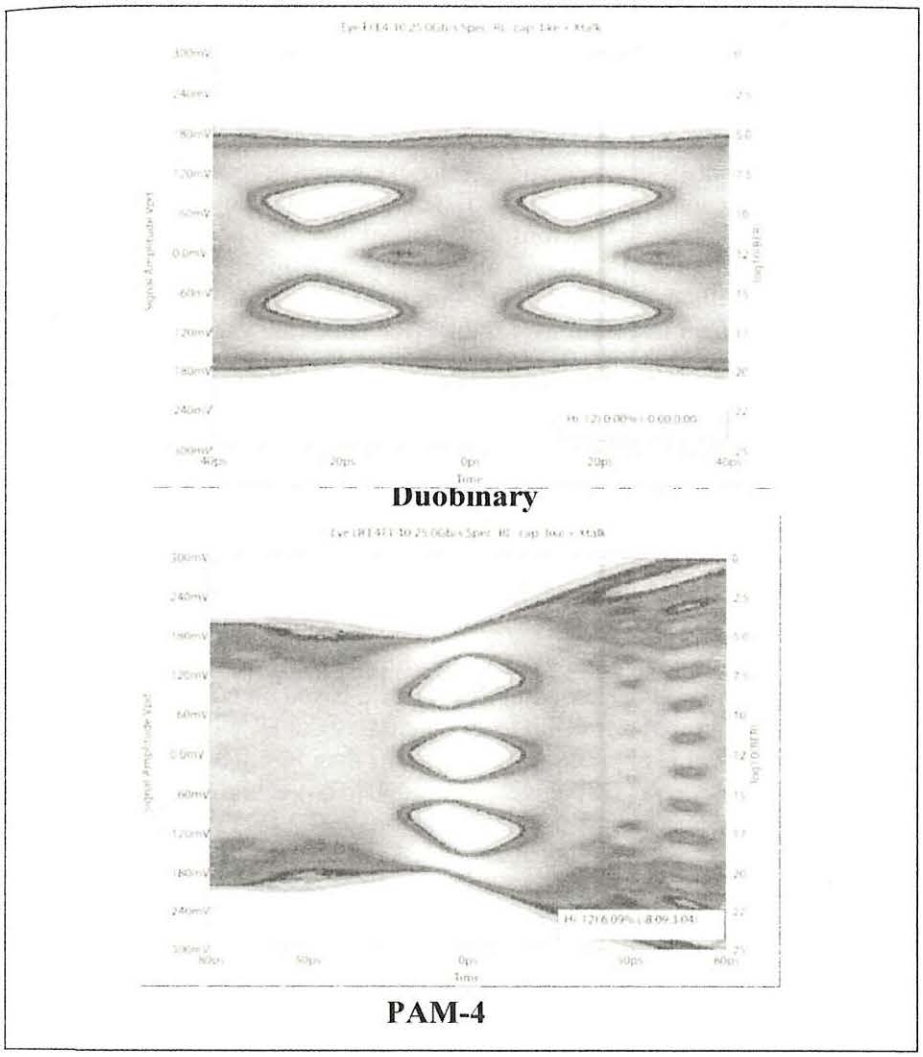


Fig. 1.21 Multilevel signaling eyes

Duobinary uses three signal levels, requiring two slicer circuits to determine the signal level. A "1" is indicated by any change in signal level, while a "0" is indicated by no change in signal level. The baud rate of a duobinary signal is the same as that of an equivalent NRZ signal. However, because the signal cannot transition from one extreme of the dynamic range to the other extreme in the same unit interval, the maximum fundamental frequency of the signal is one-quarter of this baud rate.

PAM-4 uses four signal levels, requiring three slicer circuits to determine the signal level. Each signal level represents the transmission of two bits, either "00," "01," "10," or "11." The baud rate of a PAM-4 signal is half that of an

Marvell Semiconductor, Inc. - Ex. 1008, Page 000035

equivalent NRZ signal, and the maximum fundamental frequency of this signal is 50% of this baud rate (25% of the baud rate of an equivalent NRZ signal).

Because the signal spectrum of a duobinary and PAM-4 signal is reduced from that of an NRZ signal, the insertion loss of the channel affects the signal less. This is offset by the fact that splitting the dynamic range of the signal into multiple levels results in a reduction of the eye height at the transmitter. If the slope of the insertion loss curve is sufficiently steep, then classical analysis indicates an advantage for the multilevel signal over that of the NRZ signal. The eye height of a PAM-4 signal, for example, is reduced by approximately 9dB at the transmitter from that of an equivalent NRZ signal. If the difference between the channel insertion loss at the NRZ baud rate vs. the PAM-4 baud rate is greater than 9dB, then PAM-4 would result in more eye opening at the receiver.

Note, however, that the classical analysis described above assumes no equalization is used in the HSS device. If equalizers are used, then the effects of these equalizers on channel response must be taken into account. Decision feedback equalization in the receiver has the effect of flattening the overall insertion loss of the system as shown in Fig. 1.22. While multilevel signaling may have an advantage for a given channel when equalization is not used, this advantage may not exist when NRZ is employed with a DFE in the receiver.

Most HSS devices integrated in ASIC chips use NRZ signaling, and all of the interface standards discussed in this text are based on NRZ signaling. Standards development efforts up to and including the 10–11 Gbps range have found NRZ with DFE to produce better results than multilevel signaling approaches. However, as HSS devices continue to target higher and higher data rates, this may or may not be true in the future.

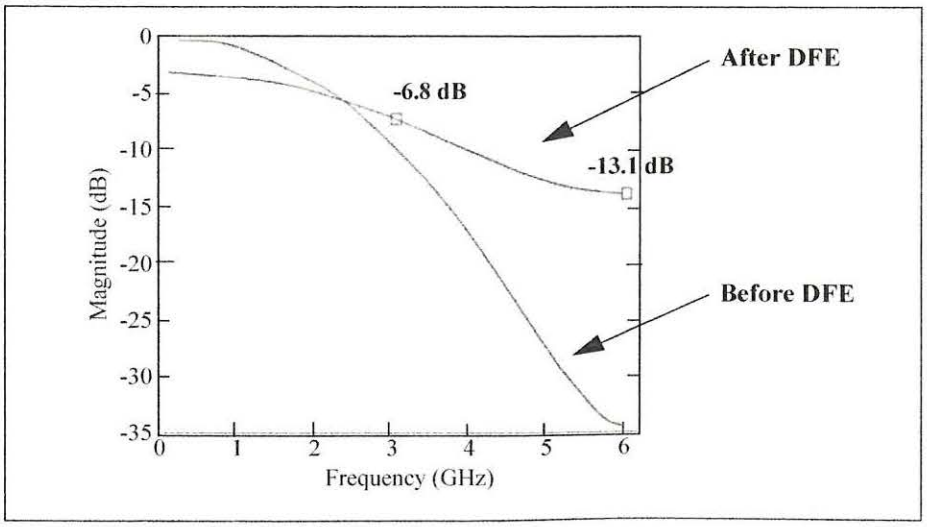


Fig. 1.22 Frequency response of a channel and considering DFE

Marvell Semiconductor, Inc. - Ex. 1008, Page 000036

Table 2.3 Receiver slice specific core pin definitions

Pin name	Type	Description
RXxSIGDETEN	In	Signal Detect Enable 0=Signal Detect power control using *Signal Detect Power Down* bit in *SIGDET Control Register*, 1=Signal Detect circuit powered on

2.1.2 HSS EX10 Register Descriptions

The HSSPRTADDR[7:0] inputs of the HSS EX10 core select the internal register to be written or read. The HSSPRTADDR[7:4] address bits select which slice is being addressed, and the HSSPRTADDR[3:0] bits select the particular register within the slice. All registers are 16 bits wide, although in many cases not all bits of the register have a defined function.

Table 2.4 describes how HSSPRTADDR[7:4] map to register space of the HSS EX10 slices. Note that values 0xC and 0xD are broadcast addresses which allow all TXx or RXx slice registers (respectively) to be written in parallel with one write cycle. Not all values of HSSPRTADDR[7:4] are used.

Table 2.4 HSS EX10 address map

Addr (7:3)	Maps to slice
0x0	TXA Slice Registers
0x1	TXB Slice Registers
0x2	RXA Slice Registers
0x3	RXB Slice Registers
0x4	TXC Slice Registers
0x5	TXD Slice Registers
0x6	RXC Slice Registers
0x7	RXD Slice Registers
0x8	PLL Slice Registers
0xC	Write all TXx Slice Registers in parallel.
0xD	Write all RXx Slice Registers in parallel.

Table 2.5 describes the registers and bit definitions for registers in the PLL slice. Table 2.6 describes the registers and bit definitions for registers in the transmitter slice. Table 2.7 describes the registers and bit definitions for registers in the receiver slice.

Marvell Semiconductor, Inc. - Ex. 1008, Page 000037

Table 2.5 HSS EX10 PLL slice register definitions

Addr (3:0)	Bits	R/W	Reset value	Description
0x0	**16**	**R**	**0x0000**	**VCO Coarse Calibration Status Register**
	0	R	0	CCALCOMP, calibration complete signal where: 0=calibration not complete, 1=calibration complete
	1	R	0	CCALERROR, calibration error occurred. 0=no errors, 1=calibration error occurred
	2	R	0	LOCK_DETECTED signal: 0=not locked, 1=locked
	15:3	R		Unused
0x1	**16**	**R/W**	**0x0000**	**VCO Coarse Calibration Control Register**
	0	R/W	0	Recalibrate signal, pulse high for minimum of eight reference clocks then return low to initiate an autocalibration sequence. 0 = normal (default), 1 = force PLL recalibration
	15:1	R		Unused
0x2	**16**	**R/W**	**0x00FF**	**Link Enable Register**
	0	R/W	1	Link enables to TXA. 0=disabled, 1=enabled
	1	R/W	1	Link enables to TXB. 0=disabled, 1=enabled
	2	R/W	1	Link enables to RXA. 0=disabled, 1=enabled
	3	R/W	1	Link enables to RXB. 0=disabled, 1=enabled
	4	R/W	1	Link enables to TXC. 0=disabled, 1=enabled
	5	R/W	1	Link enables to TXD. 0=disabled, 1=enabled
	6	R/W	1	Link enables to RXC. 0=disabled, 1=enabled
	7	R/W	1	Link enables to RXD. 0=disabled, 1=enabled
	15:8	R		Unused

Marvell Semiconductor, Inc. - Ex. 1008, Page 000038

Table 2.5 HSS EX10 PLL slice register definitions

Addr (3:0)	Bits	R/W	Reset value	Description
0x3	**16**	**R/W**	**0x0000**	**Link Reset Register**
	0	R/W	0	Link reset to TXA. 0=normal, 1=reset
	1	R/W	0	Link reset to TXB. 0=normal, 1=reset
	2	R/W	0	Link reset to RXA. 0=normal, 1=reset
	3	R/W	0	Link reset to RXB. 0=normal, 1=reset
	4	R/W	0	Link reset to TXC. 0=normal, 1=reset
	5	R/W	0	Link reset to TXD. 0=normal, 1=reset
	6	R/W	0	Link reset to RXC. 0=normal, 1=reset
	7	R/W	0	Link reset to RXD. 0=normal, 1=reset
	15:8	R		Unused

Table 2.6 HSS EX10 transmitter slice register definitions

Addr (3:0)	Bits	R/W	Reset value	Description
0x0	**16**	**R/W**	**0x0008**	**Transmit Configuration Mode Register**
	1:0	R/W	00	Rate Select 00=Full rate (default) 01=Half rate 10=Quarter rate 11=Eighth rate
	3:2	R/W	10	Parallel Data Bus Width 00=8 bit 01=10 bit 10=16 bit (default) 11=20 bit
	15:4	R		Unused

Marvell Semiconductor, Inc. - Ex. 1008, Page 000039

Table 2.6 HSS EX10 transmitter slice register definitions

Addr (3:0)	Bits	R/W	Reset value	Description
0x1	**16**	**R/W**	**0x0000**	**Transmit Test Control Register** Note: TXxPRBSEN pin = "1" overrides this register and forces PRBS7+ to be transmitted
	2:0	R/W	000	Test Pattern Selector 000 = PRBS7+ (noninverted) (default) 001 = PRBS7– (inverted) 010 = PRBS23+ (noninverted) 011 = PRBS23– (inverted) 100 = PRBS31+ (noninverted) 101 = PRBS31– (inverted) 110 = 1010101.... 111 = repeating pattern of 64 "1"s followed by 64 "0"s
	3	R/W	0	Test Pattern Generator Enable 0=disable generator and select Customer Parallel Data (default), 1=enable generator and select Test Pattern Data
	4	R/W	0	PRBS Reset. 0=normal (default), 1=reset applied to Test Pattern generator
	15:5	R		Unused
0x2	**16**	**R/W**	**0x0000**	**Transmit Coefficient Control Register**
	0	R/W	0	Apply Load This bit applies the register-loaded values of coefficients, power, polarity and FFE mode to the coefficient recalculation logic, and presents this new value to the analog circuits
	1	R/W	0	Reset Coefficient Logic 0=normal (default), 1=reset
	15:2	R		Unused

Marvell Semiconductor, Inc. - Ex. 1008, Page 000040

Table 2.6 HSS EX10 transmitter slice register definitions

Addr (3:0)	Bits	R/W	Reset value	Description
0x3	**16**	**R/W**	**0x0020**	**Transmit Driver Mode Control Register**
	1:0	R/W	00	FFE mode Select 00=FFE2 (default) 01=FFE3 10=reserved 11=Force Hi-Z
	4:2	R/W	000	Slow Slew Control Used to limit the minimum Transmitter output rise and fall time. 000 = 24ps min. (default) 001= 36ps min. 101= 50ps min. 011= 60ps min. 111=100ps min.
	15:5	R		Unused
0x4	**16**	**R/W**	**0x0000**	**Transmit Tap0 Coefficient Register**
	3:0	R/W	0000	FFE Tap 0 Coefficient This register's value is applied to the analog logic after "Apply Load" (Transmit Coefficient Control Register 0x02 bit 0) is pulsed. The value read from this register is the actual value being driven to the analog logic. Value is unsigned magnitude. See the Transmit Polarity Register 0x08 for sign values
	15:4	R		Unused
0x5	**16**	**R/W**	**0x003F**	**Transmit Tap1 Coefficient Register**
	5:0	R/W	111111	FFE Tap 1 Coefficient See the description for the Transmit Tap0 Coefficient Register (0x04)
	15:6	R		Unused
0x6	**16**	**R/W**	**0x0000**	**Transmit Tap2 Coefficient Register**
	4:0	R/W	00000	FFE Tap 2 Coefficient See the description for the Transmit Tap0 Coefficient Register (0x04)
	15:5	R		Unused

Marvell Semiconductor, Inc. - Ex. 1008, Page 000041

Table 2.6 HSS EX10 transmitter slice register definitions

Addr (3:0)	Bits	R/W	Reset value	Description
0x7	**16**	**R/W**	**0x007F**	**Transmit Power Register**
	6:0	R/W	0x7F	Transmit amplitude value (unsigned positive magnitude). Valid values are 0x20 minimum to 0x7F maximum.
	15:7	R		Unused
0x8	**16**	**R/W**	**0x0007**	**Transmit Polarity Register**
	0	R/W	1	Polarity (sign) value for FFE Tap 0 Coefficient. (0 = negative, 1 = positive) This register's value is applied to the analog logic after "Apply Load" (Transmit Coefficient Control Register 0x02 bit 0) is pulsed. The value read from this register is the actual value being driven to the analog logic
	1	R/W	1	Polarity (sign) value for FFE Tap 1 Coefficient. (0 = negative, 1 = positive) See this register's bit 0 description
	2	R/W	1	Polarity (sign) value for FFE Tap 2 Coefficient. (0 = negative, 1 = positive) See this register's bit 0 description
	15:3	R		Unused

Table 2.7 HSS EX10 receiver slice register definitions

Addr (3:0)	Bits	R/W	Reset value	Description
0x0	**16**	**R/W**	**0x0038**	**Receive Configuration Mode Register**
	1:0	R/W	00	Rate Select 00=Full rate (default) 01=Half rate 10=Quarter rate 11=Eighth rate
	3:2	R/W	10	Parallel Data Bus Width 00=8 bit 01=10 bit 10=16 bit (default) 11=20 bit
	5:4	R/W	11	DFE/non-DFE Mode Selector: 00=DFE5, 01=DFE3, 10 or 11 =non-DFE
	15:6	R		Unused

Marvell Semiconductor, Inc. - Ex. 1008, Page 000042

Table 2.7 HSS EX10 receiver slice register definitions

Addr (3:0)	Bits	R/W	Reset value	Description
0x1	**16**	**R/W**	**0x0000**	**Receive Test Control Register** Note: RXxPRBSEN pin = "1" overrides this register and enables checking of a PRBS7+ pattern
	2:0	R/W	000	Test Pattern Selector 000 = PRBS7+ (noninverted) (default) 001 = PRBS7– (inverted) 010 = PRBS23+ (noninverted) 011 = PRBS23– (inverted) 100 = PRBS31+ (noninverted) 101 = PRBS31– (inverted) 110 or 111 =Unused
	3	R/W	0	PRBS Check Enable 0=disabled (default), 1=enabled
	4	R/W	0	PRBS Reset. 0=normal (default), 1=reset applied to PRBS Checker
	5	R/W	0	Full Duplex wrap enable. 0=normal. Selects primary input to the RX, and disables the internal TX to RX wrap buffer (default), 1=wrap. Enables the wrap back driver in the TX to drive the internal wrap path to this RX
	6	R	0	State of RXxPRBSSYNC pin – PRBS checker status 0=PRBS checker not synchronized to incoming data, 1=PRBS checker synchronized and locked to incoming PRBS data
	7	R	0	State of RXxPRBSERR pin – PRBS checker status 0=PRBS pattern match or PRBS checker status = 0, 1=PRBS error detected after PRBS synchronized to incoming data
	8	R	0	State of RXxPRBSFRCERR pin – PRBS force error input signal status 0=PRBS normal operation, 1=PRBS error forced by opening wrap path
	15:10	R	0x00	Unused

Marvell Semiconductor, Inc. - Ex. 1008, Page 000043

Table 2.7 HSS EX10 receiver slice register definitions

Addr (3:0)	Bits	R/W	Reset value	Description
0x2	**16**	**R/W**	**0x0000**	**Sonet Clock Control Register** Enable and frequency selection for RXxRCVC16[T/C] clock output
	1:0	R/W	0x0	Sonet Clock rate selector 00=C4 (default), 01=C8, 10=C16, 11=C4
	2	R/W	0	Sonet Clock output enable 0=disabled, 1=enabled
	15:3	R		Unused
0x3	**16**	**R/W**	**0x0000**	**Phase Rotator Control Register**
	0	R/W	0	Spread Spectrum Clocking Enable: 0=Spread Spectrum Clocking support disabled, 1=Spread Spectrum Clocking support enabled. Should not be enabled unless SSC data is applied. RX Jitter tolerance is improved in non-SSC mode
	1	R/W	0	Reset Flywheel: 0=normal (default, the flywheel is enabled), 1=assert reset to the phase rotator flywheel (disable the flywheel)
	2	R/W	0	Freeze Flywheel: 0=normal (default), 1=freeze the phase rotator flywheel at its current update rate. This can be used to prevent periods of inactivity from altering the state of the flywheel
	15:3	R		Unused
0x4	**16**	**R**	**0xXXXX**	**Phase Rotator Position Register** These registers are continuously updated by the DFE algorithms. To accurately read the values in these registers, the DFE logic should be stopped by setting "DFE Stand By" (bit 2 of register 0x06) to "1."
	5:0	R	0xXX	Rotator Data channel Position: Snapshot sample of DATA channel phase rotator position. This is a six-bit vector indicating which of the 64 possible positions the phase rotator is in
	7:6	R		Unused
	13:8	R	0xXX	Rotator AMP channel Position: Snapshot sample of AMP channel phase rotator position. This is a six-bit vector indicating which of the 64 possible positions the phase rotator is in
	15:14	R		Unused

Marvell Semiconductor, Inc. - Ex. 1008, Page 000044

Table 2.7 HSS EX10 receiver slice register definitions

Addr (3:0)	Bits	R/W	Reset value	Description
0x5	**16**	**R/W**	**0x0000**	**Signal Detect Control Register**
	4:0	R/W	00000	Signal Detect Level: Unsigned value of comparator threshold used in SIGDET circuit.
	5	R/W	0	Signal Detect Power Down 0=enable (default – required for DFE mode), 1=power down the Signal Detect circuit
	15:6	R		Unused
0x6	**16**	**R/W**	**0x000X**	**DFE Control Register**
	0	R/W	0	DFE Control Logic Reset: 0=normal, 1=triggers a reset of the DFE logic
	1	R	X	Not Random Data Status: Proper training of the DFE engine requires sufficiently random data flow. In order to prevent the DFE engine from responding to periods of non-random data, a "random data detector" function is built into the logic. This bit is read to indicate current detection value of this logic. 0=Data is "random," 1=Data is not "random"
	2	R/W	0	DFE Stand By: 0=normal operation (default), 1=standby mode. All internal DFE operations are halted at the next internal break point. Clocks continue to run, but state machines are held idle
	3	R/W	0	Sample DFE request. 0=inactive (default), 1=a rising edge causes the pipeline sampling logic to capture a new snapshot, and makes the results available in registers 0x07 and 0x08.
	4	R	0	Sample DFE request completed. 0=inactive, or not ready yet (normal), 1=requested sample snapshot is now valid and available in registers 0x07 and 0x08
	15:5	R		Unused

Marvell Semiconductor, Inc. - Ex. 1008, Page 000045

Table 2.7 HSS EX10 receiver slice register definitions

Addr (3:0)	Bits	R/W	Reset value	Description
0x7	**16**	**R**	**0xXXXX**	**DFE Data and Edge Sample Register**
	7:0	R	0xXX	DFE Data Samples These are the latest samples captured from the pipeline registers in response to Sample DFE Request (reg 0x06 bit 4)
	15:8	R	0xXX	DFE Edge Samples These are the latest samples captured from the pipeline registers in response to Sample DFE Request (reg 0x06 bit 4)
0x8	**16**	**R**	**0x00XX**	**DFE Amplitude Sample Register**
	2:0	R	XXX	DFE Amplitude Samples These are the latest samples captured from the pipeline registers in response to Sample DFE Request (reg 0x06 bit 4)
	5:3	R	XXX	DFE Amplitude Sample Qualifiers These are the latest samples captured from the pipeline registers in response to Sample DFE Request (reg 0x06 bit 4)
	15:6	R		Unused
0x9	**16**	**R/W**	**0x0000**	**Digital Eye Control Register**
	4:0	R/W	0x00	Minimum eye height interrupt threshold This is an unsigned vector value (positive) defining the minimum acceptable eye amplitude, as measured by the DFE logic, before "Eye Amplitude Error Flag" bit is set in the Internal Status Register (reg 0x0F, bit 6) and interrupt is triggered. 0x0: 0 threshold (default, no interrupt set) 0x1 – 0xE: 1/16 of range per step 0xF: threshold set at 15/16 of full range
	9:5	R/W	0x00	Minimum eye width interrupt threshold This is an unsigned vector value (positive) defining the minimum acceptable eye width, as measured by the Dynamic Data Centering logic algorithm, before "Eye Width Error Flag" bit is set in the Internal Status Register (reg 0x0F, bit 5) and interrupt is triggered. 0x0: 0 threshold (default, no interrupt set) 0x1 – 0xF: eye width threshold in rotator steps, approximately 0.03UI per step
	14:10	R	0x00	EYE WIDTH: Latest available eye width measurement, in units of rotator steps
	15	R		Unused

Marvell Semiconductor, Inc. - Ex. 1008, Page 000046

Table 2.7 HSS EX10 receiver slice register definitions

Addr (3:0)	Bits	R/W	Reset value	Description
0xA	**16**	**R**	**0x00C0**	**DFE Tap 1 Register** This register is read to obtain the status of the DFE Tap 1 coefficient
	5:0	R	0x00	Dac Tap 1: magnitude of DFE Tap 1 coefficient
	7:6	R	11	Sign Tap 1: This sign represents the asserted sign of DFE Tap 1 as it is applied to the summer, and changes under normal operation in DFE mode. 10 = negative, 01 = positive, 11 = zero
	15:8	R		Unused
0xB	**16**	**R**	**0x0060**	**DFE Tap 2 Register** This register is read to obtain the status of the DFE Tap 2 coefficient
	4:0	R	0x00	Dac Tap 2: magnitude of DFE Tap 2 coefficient
	6:5	R	11	Sign Tap 2: This sign represents the asserted sign of DFE Tap 2 as it is applied to the summer, and changes under normal operation in DFE mode. 10 = negative, 01 = positive, 11 = zero
	15:7	R		Unused
0xC	**16**	**R**	**0x0030**	**DFE Tap 3 Register** This register is read to obtain the status of the DFE Tap 3 coefficient
	3:0	R	0x00	Dac Tap 3: magnitude of DFE Tap 3 coefficient
	5:4	R	11	Sign Tap 3: This sign represents the asserted sign of DFE Tap 3 as it is applied to the summer, and changes under normal operation in DFE mode. 10 = negative, 01 = positive, 11 = zero
	15:6	R		Unused
0xD	**16**	**R**	**0x0030**	**DFE Tap 4 Register** This register is read to obtain the status of the DFE Tap 4 coefficient
	3:0	R	0x00	Dac Tap 4: magnitude of DFE Tap 4 coefficient
	5:4	R	11	Sign Tap 4: This sign represents the asserted sign of DFE Tap 4 as it is applied to the summer, and changes under normal operation in DFE mode 10 = negative, 01 = positive, 11 = zero
	15:6	R		Unused

Marvell Semiconductor, Inc. - Ex. 1008, Page 000047

Table 2.7 HSS EX10 receiver slice register definitions

Addr (3:0)	Bits	R/W	Reset value	Description
0xE	**16**	**R**	**0x0030**	**DFE Tap 5 Register** This register is read to obtain the status of the DFE Tap 5 coefficient
	3:0	R	0x00	Dac Tap 5: magnitude of DFE Tap 5 coefficient
	5:4	R	11	Sign Tap 5: This sign represents the asserted sign of DFE Tap 5 as it is applied to the summer, and changes under normal operation in DFE mode. 10 = negative, 01 = positive, 11 = zero
	15:6	R		Unused
0xF	**16**	**R**	**0x0000**	**Internal Status Register** This register is used to report the status of certain internal operations. When status bits in this register change, HSSEYEQUALITY is asserted. A write to this register resets the HSSEYEQUALITY output.
	0	R	0	Phase Rotator Calibration Complete. 0=calibration not completed, 1=phase rotator offset calibration process is completed
	1	R	0	VGA locked First This register is set when the VGA achieves lock, and is cleared only by reset. It indicates lock was achieved at least once
	3:2	R	00	Unused
	4	R	0	DFE training complete This register is set when the DFE logic determines that its H coefficients have converged since reset. 0=initial DFE convergence not yet achieved, 1=initial DFE convergence achieved
	5	R	0	Eye Width Error Flag: 0=normal: Measured Data Eye Width at or above interrupt threshold set in the Digital Eye Control Register (reg 0x09); 1=error: Measured Data Eye Width below interrupt threshold set in the Digital Eye Control Register (reg 0x09)
	6	R	0	Eye Amplitude Error Flag: 0=normal: Measured Data Eye Height at or above interrupt threshold set in the Digital Eye Control Register (reg 0x09); 1=error: Measured Data Eye Height below interrupt threshold set in the Digital Eye Control Register (reg 0x09)
	15:7	R	0x000	Unused

Marvell Semiconductor, Inc. - Ex. 1008, Page 000048

2.2 HSS EX10 Transmitter Slice Functions

In this section, the functions of the transmitter slice of the HSS EX10 10-Gbps core are described. This core supports transmit bit rates as low as 8.5 Gbps and as high as 11.1 Gbps. Frequent references are made to the pin descriptions and register definitions found in Sect. 5.5.42.1. Although the HSS EX10 is only a tutorial example, the reader should compare the functions of the EX10 to real HSS cores with which the reader is familiar. Although implementations may vary, many similar functions will be found. In reading this chapter, the reader should acquire an understanding of the types of functions that may exist and some of the key features related to these functions.

A conceptual block diagram of the HSS EX10 transmitter is shown in Fig. 2.4. The parallel data input of the transmitter (TXxD[19:0]) has a 20-bit data width which may be programmed to capture 8, 10, 16, or 20 bits of user data (based on the setting of the *Parallel Data Bus Width* control bits in the *Transmit Configuration Mode Register*) on the rising edge of the word transmit clock (TXxDCLK). Other cores may support different data widths.

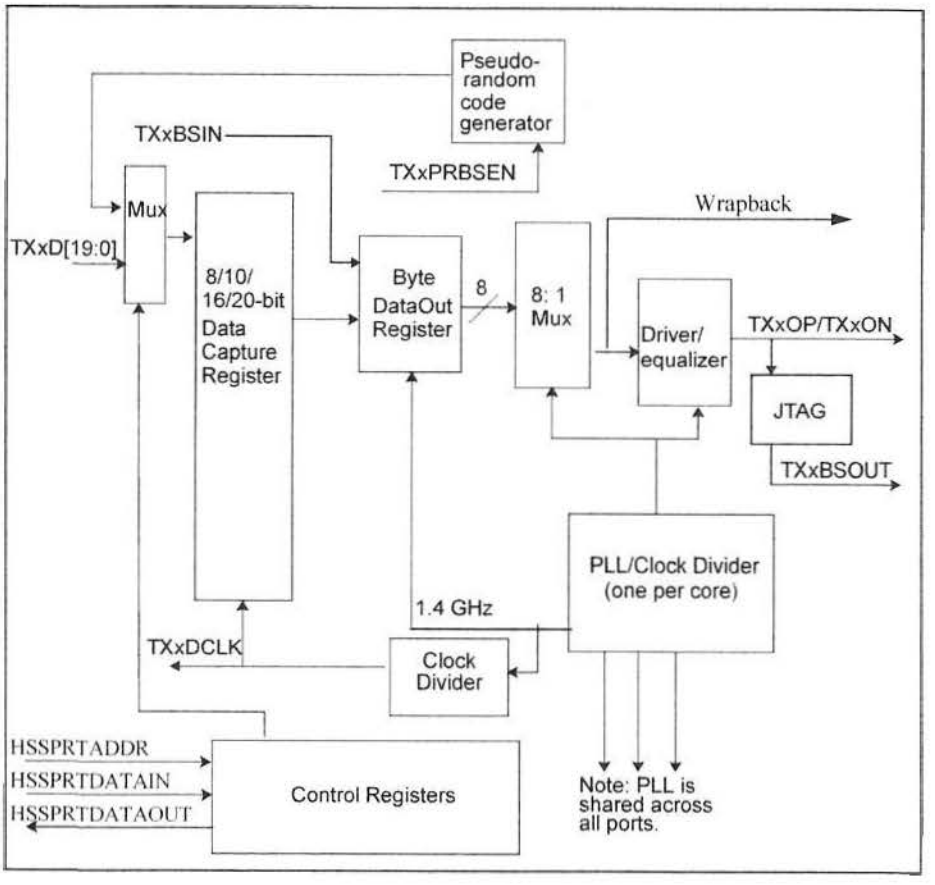


Fig. 2.4 Transmitter concept diagram

Marvell Semiconductor, Inc. - Ex. 1008, Page 000049

2.2.3 Transmitter FFE Programming

The HSS EX10 transmitter includes a feed forward equalization (FFE) to reduce intersymbol interference (ISI) at the receiver. This preemphasis technique uses a FIR filter to compensate for the high frequency roll-off of the transmission channel. Control inputs are provided to allow adjustment of the driver FFE filter coefficients on a per-port basis. The following equation describes the relationships of the FFE Coefficients.

$$H(Z) = K\,(C_0 z^{+1} + C_1 z^{0} + C_2 z^{-1}) \tag{2.1}$$

This Z-transform equation describes an FFE with three taps, including one pre- and one post tap. Other cores may have FFEs with fewer or more taps.

The driver amplitude (K) is adjustable in the range of 0 – 1,320 (nominal) mV peak-to-peak differential using the *Transmit Power Register* described previously in this chapter. The relative weights and polarities of C_0 to C_2 are configured using the three *Transmit TapX Coefficient Registers* ($X = 0, 1, 2$), and the *Transmit Polarity Register*. The actual range of relative coefficient weights is defined in Table 2.12.

The resolution of the levels is also indicated in Table 2.12, with each tap having equal bit weighting. The driver circuit design enforces a constant driver output power for any combination of coefficients and polarities, provided the sum of coefficients $C_0 - C_2$ is 63 or higher (power decreases proportionally below this sum). As coefficients are initialized or updated, the logic calculates the appropriate internal amplitude (K) to maintain the overall output power at the level defined in the *Transmit Power register*.

Table 2.12 Transmitter FFE summary

Tap coefficient #	0	1	2
Max current (mA)	7.5	30	15
Relative max (%)	25	100	50
DAC resolution (bits)	6	8	7
Tap allocation	Precursor	Main tap	Postcursor

Table 2.13 Transmit driver modes

Mode	FFE mode select bits of "Transmit Driver Mode Control Register" for Port x	FFE taps activated
FFE2	"00"	0, 1
FFE3	"01"	0, 1, 2
(Reserved)	"10"	(Indeterminate)
Hi-Z	"11"	None (see Sect. 2.2.4.1)

Marvell Semiconductor, Inc. - Ex. 1008, Page 000050

To save power in certain applications, the FFE can be configured into several configurations using the *Transmit Driver Mode Control Register* (see Table 2.13). By using fewer taps, power is reduced.

2.2.3.1 Loading Transmit Coefficients

Many of the TX slice parameter control registers are not effective immediately after being written. Application of register values to the analog circuits is controlled by writing the *Apply Load* bit in the *Transmit Coefficient Control Register*. This allows the FFE and power level configuration to be completely loaded in the individual registers and then simultaneously applied to the analog circuits. For the HSS EX10, the values loaded into the *Transmit TapX Coefficient Registers*, the *Transmit Power Register*, the *Transmit Polarity Register*, and the *FFE Mode Select* field of the *Transmit Driver Mode Control Register* are not applied to the analog circuits until the *Apply Load* bit is pulsed. The read-back value of these registers reflects the value being applied to the analog circuits. If these registers are written and then read before the *Apply Load* bit is pulsed, the old values are read. (The value of the *Slow Slew Control* field of the *Transmit Driver Mode Control Register* is applied to the analog circuits immediately, and is not gated by the *Apply Load* bit.)

For cores which do not contain internal registers, filter coefficients may be set using input control pins. Sequencing updated values onto these pins is the responsibility of the chip designer.

FFE coefficient values are not intended to be changed dynamically while data is being transmitted (except as part of a speed negotiation or link optimization process). Significant changes to the TX slice configuration can cause loss of data at the receiver until such time as the receiver adapts to the new waveform characteristics.

2.2.3.2 FFE Coefficient Negotiation Support

Higher speed protocol standards sometimes support negotiation of transmitter FFE settings based on characteristics of the signal at the receiver. IEEE 802.3ap Backplane Ethernet is an example of such an application. Implementations of such applications must update FFE coefficients in response to a full-duplex training protocol by reading and writing the corresponding registers.

The training protocol used for Backplane Ethernet is introduced in Sect. 5.3.5.1. Some cores provide register control signals to more easily implement the actions defined by this protocol, which include incrementing/decrementing coefficient values. Such features are not defined for the HSS EX10 core.

2.2.4 Transmitter Power Control

Power management is becoming an increasingly important part of chip designs. Various features are incorporated into HSS EX10 core to facilitate turning off all or part of Transmitter Slice when the corresponding interfaces are not in use.

Marvell Semiconductor, Inc. - Ex. 1008, Page 000051

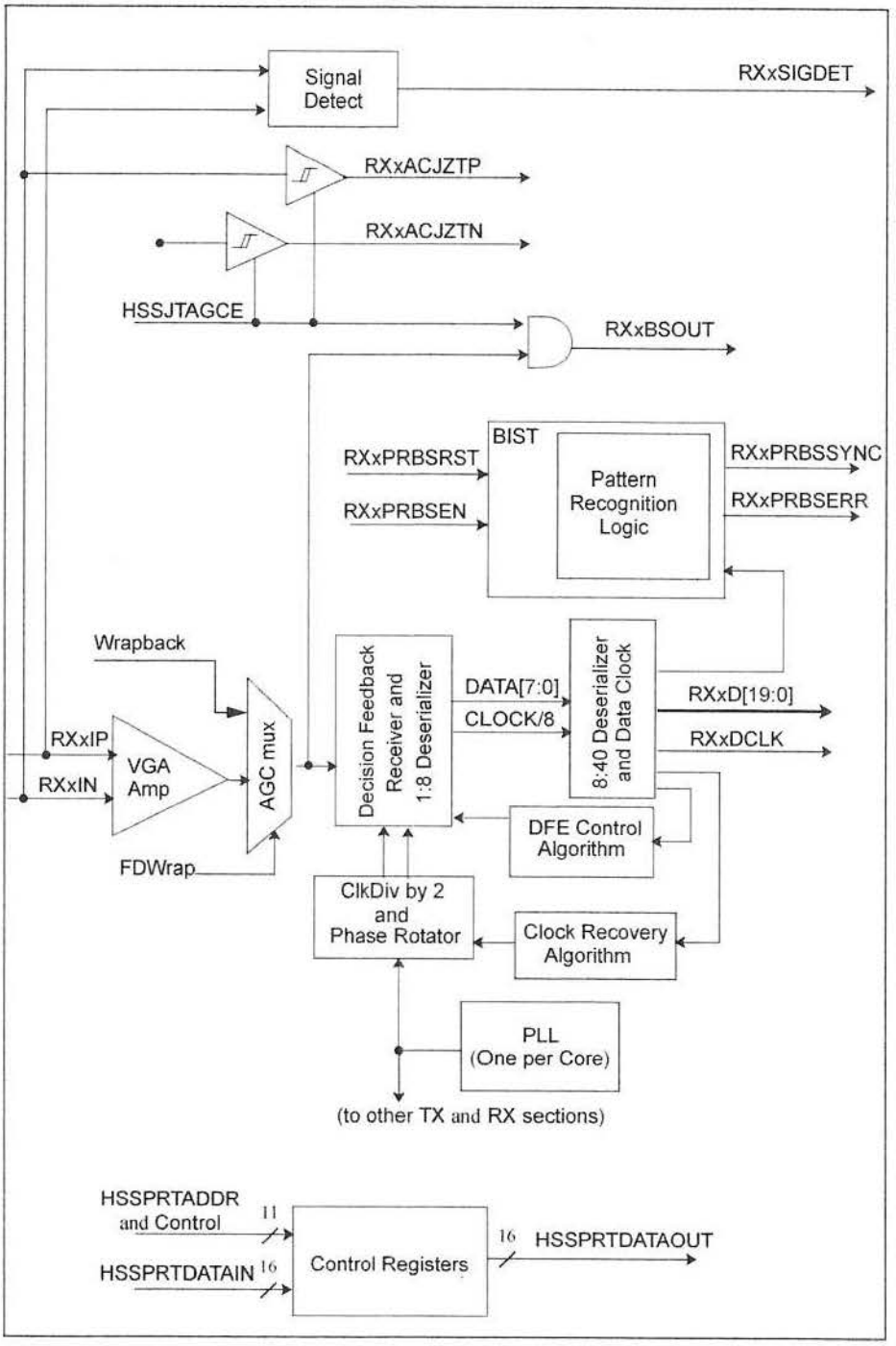


Fig. 2.7 Receiver concept diagram

Marvell Semiconductor, Inc. - Ex. 1008, Page 000052

Another consideration of Receiver Slice usage is the bit order in which bits of the parallel data bus are serially received. For the HSS EX10 core, the first bit received is always steered to bit 0 of the RXxD bus. The user must be cautious that this is consistent with the interface standard being implemented. If necessary the datapath connections to the RXxD inputs must be rearranged to obtain the desired bit order.

Receive data alignment may be adjusted through the use of the RXxDATASYNC control input, and may be independently adjusted on each receiver slice. When the RXxDATASYNC signal transitions from a "0" to a "1," the deserializer within the receive logic ignores the most recent serial input and holds the contents of the deserializer for one-bit clock cycle. The RXxDATASYNC operation has the effect of changing the deserializer alignment by one-bit position.

The RXxDATASYNC operation also has the effect of stretching the corresponding RXxDCLK cycle. This behavior is dependent on the rate mode selected. In full-and half-rate modes, the RXxDCLK is stretched 2 UI after every RXxDATASYNC operation. In quarter-and eight-rate modes, the RXxDCLK is stretched 1 UI after every RXxDATASYNC operation. Because the cycle is always stretched, the minimum pulse width of the RXxDCLK signal is never reduced. Figure 2.9 illustrates the timing requirements for the RXxDATASYNC input.

2.3.2 DFE and Non-DFE Receiver Modes

In full-rate and half-rate modes the HSS EX10 receiver operates in one of three equalization modes: non-DFE, DFE3, or DFE5. In quarter-rate and eighth-rate modes only non-DFE mode is supported. In non-DFE mode, the received input signal is processed through the variable gain amplifier (VGA) then captured in the front end logic and deserialized. In DFE3 and DFE5 modes, the deserialized samples are processed using DFE algorithms that automatically adjust the receiver threshold to better compensate for severe pattern-dependent distortions in the channel. These modes are selected using by the *DFE/non-DFE Mode Select* bits in the *Receive Configuration Mode Register* as shown in Table 2.17.

Table 2.17 DFE/non-DFE mode select

Receiver mode	DFE/non-DFE mode selector bits of "Receive Configuration Mode Register" for Port x	
DFE5	0	0
DFE3	0	1
non-DFE	1	0
non-DFE (default)	1	1

Marvell Semiconductor, Inc. - Ex. 1008, Page 000053

DFE5 and DFE3 modes use 5 and 3 filter taps, respectively for feedback. A DFE-*n* mode makes decisions regarding whether a bit is "1" or "0" based on the history of the last *n* bits received. DFE5 provides maximum performance in terms of BER, while DFE3 may be used in cases where signal degradation is not as severe. The non-DFE mode uses a fixed threshold receiver for cases where signal degradation is small enough not to require further equalization. The VGA amplifier is used in all modes of operation.

2.3.3 Serial Data Termination and AC/DC Coupling

The RX slice includes a termination resistance between the two legs of the differential input as well as biasing to the termination supply voltage. The common mode voltage that is seen on the differential signal tracks the termination supply voltage, and for this reason the termination supply may be applied through separate power pins on the HSS core. On some cores, additional inputs may be provided to select the voltage range for this supply.

The HSS EX10 uses the nomenclature AVTT to designate the termination supply at the transmitter and AVTR to designate the termination supply at the receiver. These supply voltages are usually set to the same voltage to prevent circulating DC currents and to insure proper operation and reliability of the receiver. Note that there may be power supply sequencing requirements between the AVTT/AVTR supplies and the main power supply.

Some applications and system designs use AC coupling on serial links to eliminate some of these restrictions. An AC coupled system includes decoupling capacitors inserted inline with the serial data signals such that there is no DC path between the transmitter and the receiver device. The decoupling capacitor must have sufficient capacitance to pass the lowest spectral frequency expected in the data. When AC coupling is employed, the common mode bias point of the transmitter and the receiver do not need to be the same.

In a DC coupled link, the current path through the transmitter, the channel, and the termination network in the receiver determines the resulting common mode voltage. The HSS EX10 receiver is designed using the assumption that the transmitter provides a termination which contributes to the voltage bias at the receiver. The current path through the transmitter does not exist when AC coupling is employed, and therefore the transmitter termination no longer contributes to the DC voltage bias for the circuit. The HSS EX10 core has an HSSRXACMODE control input on the PLL slice which enables biasing the receiver's termination network when the core is used in AC coupled applications. This pin enables an additional current path which biases the receiver inputs to a common mode voltage equal to around 80% of the AVTR voltage, which is optimal for operation of this receiver.

2.3.4 Signal Detect

Each individual receiver circuit includes signal detection circuitry. The function of signal detection is to continuously monitor the attached media channel and to provide feedback concerning link status. A "1" on the

Marvell Semiconductor, Inc. - Ex. 1008, Page 000054

The second DFE function is to determine the H1 to H5 tap coefficient values, which is accomplished via an algorithm in the DFE Engine (the DFE Engine block in Fig. 3.13).

Once the data has been equalized and captured, the data/edge/amplitude samples are aligned and deserialized in the *Data Alignment/2:8 Deserializer* block. This data is further processed in digital form by the three subfunctions in the synthesized logic macro: the *CDR Macro*, the *Data Deserializer*, and the *DFE Engine* functions. The data/edge/amplitude sample latches are clocked by three independent phase rotators. These phase rotators use a set of I/Q (0/90°) clocks and provide a digitally controlled phase interpolation function, producing 64-step granularity across a 2 UI range. This corresponds to a net time resolution of the phase rotator of 32 steps per UI.

Other blocks shown in Fig. 3.13 include the *SigDet* block which implements the signal detection features (see Sect. 2.3.4) and the *AC JTAG* block which provides low-speed board level testing (see Sect. 2.3.6).

3.3.1 Clock and Data Recovery (CDR) Operation

The CDR macro uses a twofold oversampling scheme, capturing four samples across two bits (two edge and two data samples). In Fig. 3.13, the data samples are captured in the *DFE* block and the edge samples are captured in the *DFE Edge* block. Note that the *DFE* block and the *DFE Edge* block use identical summers to maintain timing alignment between the two functions; this is critical for proper data centering. As illustrated in Fig. 3.14, a simplified *bang–bang phase detector* approach can be visualized. In practice, the HSS EX10 core uses a 16-way bang–bang phase detector, with 16 data and 16 edge samples implemented in the CDR logic macro. Digital filtering in the CDR logic results in a loop bandwidth of approximately 1/1,000th of the baud rate (depending upon incoming transition density).

There are four sets of latches in the bang–bang phase detector: the D0 and D1 data latches (also used extensively in the *DFE* block) and the E0 and E1 edge latches (specifically in the *DFE Edge* block). Samples are latched in these latches using clocks which are sequenced as follows: E0 is clocked first, followed by D0, E1, and finally D1. This sequencing is shown in Fig. 3.14. As shown in the figure, XOR gates are used to detect differences in phase. When a difference exists between adjacent values, the XOR output is asserted.

The simplified bang–bang phase detector scheme is examined for three cases of potential phase alignment in Fig. 3.14. In Fig. 3.14a, both the (E0, D0), and the (E1, D1) latches are sampling the same portion of the waveform. Since the digital values are the same, the DEC0 and DEC1 signals are not asserted. Likewise the (D0, E1), and the (D1, E0) latches are sampling differing logic levels, and therefore the INC0 and INC1 signals are asserted. The INC signal causes the phase of all four clocks to shift to the left. The net effect of this is to move the E0 and E1 clocks closer to the data transitions.

Figure 3.14b shows the opposite situation where the data edges are between the E0/D0 and the E1/D1 samples. In this case, the DEC0 and DEC1 signals

Marvell Semiconductor, Inc. - Ex. 1008, Page 000055

are asserted, and the clock phases shift to the right. The effect again is that the E0 and E1 clocks are moved toward the data transitions.

Finally, Fig. 3.14c shows the case where the signal is in lock. In the absence of noise, the lock condition for the phase detector is characterized by thrashing back and forth one step around the ideal sample phase.

The digital INC/DEC signals are then used to control the sampling phase of the clocks driving the Data/Edge latches. There are several popular schemes to digitally control the sampling phase, including phase rotators, PLLs, delay locked loops, and digital oversampling schemes. The HSS EX10 core utilizes a phase rotator- based CDR.

A phase rotator is essentially a digital mixer circuit which accepts two differential clocks with 90° separation, often referred to as In-phase and Quadrature-phase (I & Q). Shown in Fig. 3.15, the half-rate (C2) input clocks *CK_I* and *CK_Q* are mixed in various proportion to control the output phase of the rotator. The linearity (phase out vs. digital phase control in) or step size of a phase rotator is a strong function of the shape of the input clock waveforms. Optimal performance is achieved when the input signals are very close to sinusoidal waveforms. To that end, the *SLEWBUF* section controls the slew rate of the input signals and produces a sine wave on the *SCK_I* and *SCK_Q* outputs. Two 15-value IDACs (typically thermometer encoded) control the amount of *I* vs. *Q* contribution to the output, and the *POL* signals select the quadrant in which the output operates. The net result is that the digital control word INT + POL can produce any one of 64 phases at the ZP/ZN outputs.

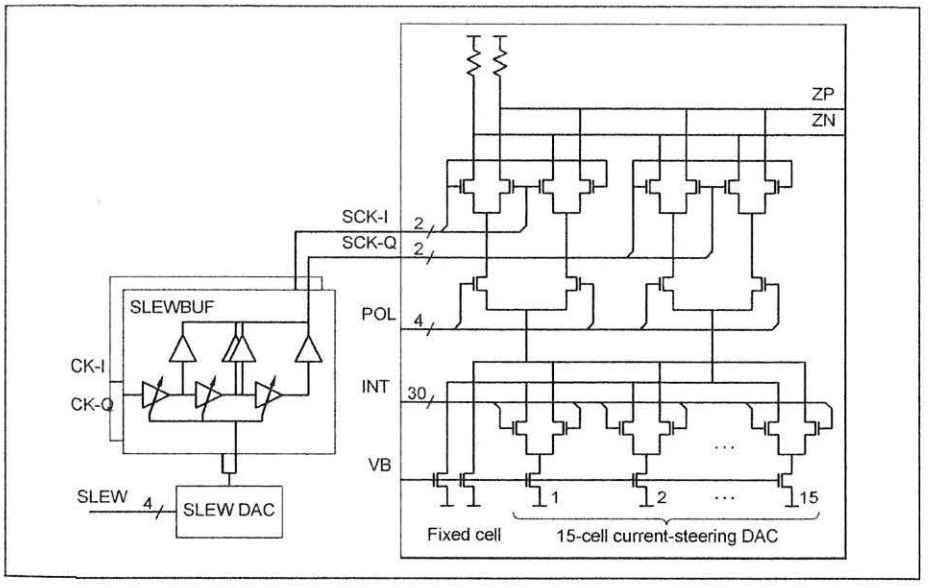


Fig. 3.15 Phase rotator conceptual diagram

Marvell Semiconductor, Inc. - Ex. 1008, Page 000056

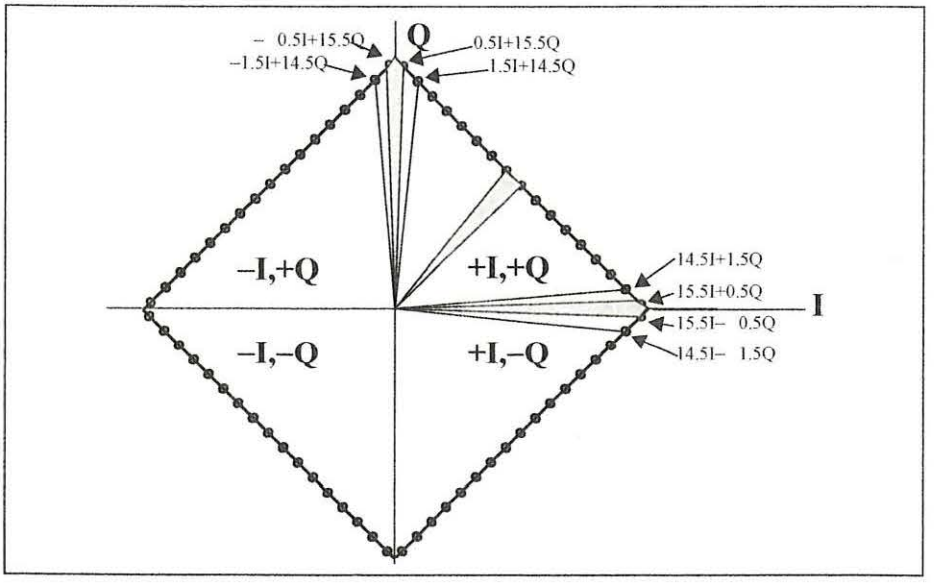


Fig. 3.16 Four quadrant phase rotator diagram

Fig. 3.16 illustrates a phase rotator quadrant diagram, showing the 64 possible combinations of phase. In order to track a frequency offset (in ppm) between the incoming signal stream and the local C2 clocks, the digital control word is periodically adjusted in a given direction. The greater the difference between the signal baud rate and the local C2 clock frequency, the more often the phase must be adjusted.

3.3.2 Decision Feedback Equalizer (DFE) Architectures

The heart of the DFE-based core is the high-speed analog function that varies the sample latch thresholds at the baud rate. Figure 3.17 illustrates how this DFE threshold movement improves the BER.

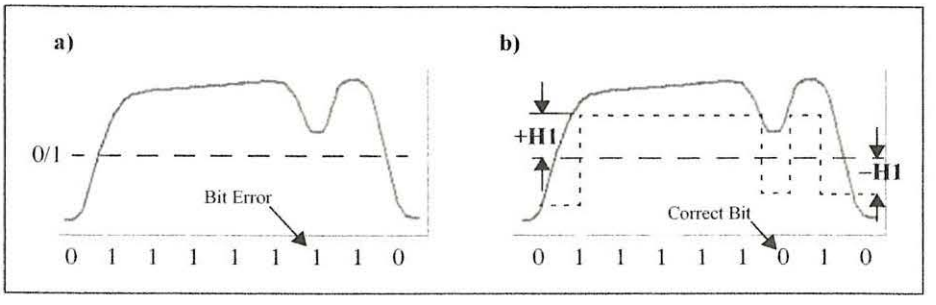


Fig. 3.17 Decision feedback equalizer threshold adjustment

Marvell Semiconductor, Inc. - Ex. 1008, Page 000057

Chapter 5
Overview of Protocol Standards

Older protocol standards defined electrical interfaces which used source synchronous bus approaches that were described in Chap. 1. As interface standards evolved to higher and higher data rates, most standards turned to specification of high-speed signals implemented with HSS cores. This chapter is by no means a complete summary of all of the protocol standards for which HSS cores are used, nor does this chapter contain sufficient information to design the logic associated with any given protocol standard. The goal is simply to provide the reader with a basic knowledge of some of the more popular standards.

Some of the protocol standards that are popular and relevant as of the time of this writing are listed below:

1. **Telecom Standards**

   (a) *International Standards*:

   - *SONET / SDH*: Telcordia GR-253-Core

     SONET (Synchronous Optical NETwork) is a serial protocol for optical telecom networks used within North America; SDH (Synchronous Digital Hierarchy) is the equivalent protocol used on other continents. The protocol is based on continuously sending frames containing multiple connection clients at the rate of 8,192 frames per second. Relevant baud rates are: 2.48832Gbps (OC-48), 9.95328Gbps (OC-192), and 39.81312Gbps (OC-768).

   (b) *Optical Internetworking Forum* (OIF):

   OIF is an industry forum which develops *Implementation Agreements* that define interfaces internal to a system that are not within the scope of other standards bodies.

   - *OIF-SxI-5-01.0*: System Interface Level 5 (SxI-5): Common Electrical Characteristics for 2.488–3.125Gbps Parallel Interfaces

     SxI-5 is an electrical layer standard incorporated by reference in the SFI-5 and SPI-5 protocols. Relevant baud rates are from 2.488Gbps up to 3.125Gbps.

   - *OIF-SFI5-01.0*: Serdes Framer Interface Level 5 (SFI-5): 40Gbps Interface for Physical Layer Devices

     SFI-5 is a protocol layer standard defining the chip-to-chip or chip-to-module interface between a SONET/SDH Framer chip and an OC-768 40-Gbps Serdes chip. It is also used to interface between a Framer or Serdes chip and a Forward Error Correction (FEC) chip. Protocol uses

D. R. Stauffer et al., *High Speed Serdes Devices and Applications*,

© Springer 2008

Marvell Semiconductor, Inc. - Ex. 1008, Page 000058

16 SxI-5 electrical links at 2.48832Gbps per link to achieve 39.81312 Gbps, plus an additional link used as a deskew reference. Baud rates are higher on links between a FEC chip and a Serdes chip; baud rate in this application depends on the FEC implementation.

- *OIF-SPI5-01.1*: System Packet Interface Level 5 (SPI-5): OC-768 System Interface for Physical and Link Layer Devices

  SPI-5 is a protocol layer standard defining chip-to-chip or backplane interfaces between SONET/SDH Framer chips and other Network Processing Elements (NPE) of an OC-768 system. The protocol uses 16 SxI-5 electrical links to implement an interface with a total bandwidth of 39.81312–50.000Gbps. Additional links in the opposite direction are used to communicate queue status.

- *OIF-CEI-02.0*: Common Electrical I/O (CEI): Electrical and Jitter Interoperability agreements for 6G+ bps and 11G+ bps I/O

  CEI is an electrical layer standard for interfaces with baud rates in the range of 4.976–6.375Gbps, and in the range of 9.95–11.10Gbps. Multiple variants are defined to target various reach objectives (short reach chip-to-chip, and long reach backplane). This electrical layer specification is incorporated by reference in the CEI-P and SFI-5.2 protocols.

- *OIF-CEI-P-01.0*: Common Electrical I/O: Protocol (CEI-P): Implementation Agreement

  CEI-P is a protocol standard for a Physical coding sublayer for CEI electrical interfaces, specifying overhead bits using a variation on 64B/66B block encoding with data scrambling for encoding data on CEI electrical links. The specification includes FEC codes used for error checking, and optionally for error correction. This lower layer protocol is incorporated by reference in the SPI-S protocol.

- *OIF-SFI5-02.0*: Serdes Framer Interface Level 5 Phase 2 (SFI-5.2): Implementation Agreement for 40Gbps Interface for Physical Layer Devices

  SFI-5.2 is a protocol layer standard defining the chip-to-chip or chip-to-module interface with similar application to SFI-5. The protocol uses four CEI-11G-SR electrical links at 9.95328 Gbps per link to achieve 39.81312Gbps, plus an additional link used as a deskew reference.

- *OIF-SPI-S-01.0*: Scalable System Packet Interface Implementation Agreement: System Packet Interface Capable of Operating as an Adaption Layer for Serial Data Links

  The SPI-S protocol layer standard defines chip-to-chip or backplane interfaces between SONET/SDH Framer chips and other Network Processing Elements (NPE) of an OC-768 system. The protocol extends SPI-5 and other prior protocols defined by OIF, specifying a scalable interface (both in terms of baud rate and in terms of bit width) capable of

Marvell Semiconductor, Inc. - Ex. 1008, Page 000059

achieving any bandwidth. SPI-S builds on top of the CEI-P sublayer, although it may also be used with other 64B/66B encoded protocols.

2. Data Networking Standards

(a) *IEEE Std-802.3 Ethernet*: Ethernet is a packet protocol for data networks. The standard is written as a clause-based document which has expanded over time to include a multitude of interface variants. A sampling of relevant variants are listed below.

- *IEEE Std 802.3, 2000 Edition, Clause 38*: 1000Base-SX, LX

  This clause defines a Physical Medium Dependent (PMD) layer for 1,000 Mbps Ethernet. This protocol uses 8B/10B encoding, so the actual baud rate is 1.25 Gbps.

- *IEEE Std 802.3ae, 2002 Edition, Clause 47*: XGMII Extended Sublayer (XGXS) and 10-Gb Attachment Unit Interface (XAUI) Specification

  This clause defines a Physical Medium Dependent (PMD) layer for a 10-Gbps Ethernet interface between two chips, and is sometimes used across a backplane. The XAUI interface is specified as four links using 8B/10B encoding and running at a baud rate of 3.125 Gbps.

- *IEEE Std 802.3ak, 2004 Edition, Clause 54*: 10GBASE-CX4 Specification

  This clause defines a physical medium dependent (PMD) layer for a 10-Gbps Ethernet interface over an IBT cable. The protocol is similar to the XAUI specification in clause 47.

- *IEEE Std 802.3ap, Clauses 70, 71, and 72*: Backplane Ethernet

  These clauses define a physical medium dependent (PMD) layers for using the Ethernet protocol across a backplane. Clause 70 specifies a 1,000 Mbps PMD layer similar to clause 38. Clause 71 specifies a four link at 3.125 Gbps PMD layer similar to clause 47. Clause 72 specifies a serial 10.3 Gbps PMD layer.

3. Storage Networking Standards

(a) *INCITS T11 Fibre Channel*: Fibre Channel is a packet protocol for storage networks. Fibre Channel is used both for communication between systems in storage area networks and between host and disk drive devices. Fibre Channel is specified by a collection of documents which specify the physical layer, protocol, and other aspects of systems. Key documents are listed below.

- *FC-PI-4*: Fibre-Channel: Physical Interfaces: 4

  This standard specifies electrical and optical physical layer Fibre Channel variants defined by the T11.2 Task Group. The more popular serial variants utilize a single link at a baud rate of 1.0625, 2.125, 4.250, or 8.500 Gbps. There is also a 10 Gbps variant specified in a separate

Marvell Semiconductor, Inc. - Ex. 1008, Page 000060

document which utilizes an interface similar to XAUI (see prior Ethernet descriptions).

- *FC-FS-2*: Fibre-Channel: Framing and Signaling: 2

  This standard specifies the Fibre Channel protocol for use over any of the physical layer variants specified in FC-PI-x and is defined by the T11.3 Task Group. The protocol utilizes 8B/10B block encoding. Serial variants with baud rates of 8.500Gbps also scramble data prior to the block encoding stage.

(b) *INCITS T10 Serial Attached SCSI*: 2 (SAS-2) *T10/1760*: SAS is a packet protocol for storage networks. SAS is used for communication between host and disk drive devices. Current supported baud rates are 1.5 and 3.0Gbps; a 6.0-Gbps baud rate is expected in the future. The protocol utilizes 8B/10B block encoding with data scrambled prior to the block encoding stage.

(c) *Serial ATA International Organization: Serial ATA Revision 2.5* (SATA): SATA is a packet protocol for storage networks. SATA is used for communication between host and disk drive devices. Current supported baud rates are 1.5 and 3.0Gbps; a 6.0-Gbps baud rate is expected in the future. The protocol utilizes 8B/10B block encoding.

4. Transaction Protocols

(a) *Peripheral Component Interconnect Special Interest Group* (PCI-SIG): *PCI Express Baseline Specification, Version 2.0*: PCI Express is a transaction protocol for interconnecting peripheral devices in computing and communication platforms. Current supported baud rates are 2.5 and 5.0 Gbps. Ports use 1, 2, 4, 8, 12, 16, or 32 links in parallel to scale bandwidth as needed. The protocol utilizes 8B/10B block encoding with data scrambled prior to the block encoding stage.

(b) *InfiniBand Trade Association* (IBTA) *Infiniband Architecture Specification Volume 2, Release 1.2*: Infiniband is a transaction protocol for interconnecting host and peripheral devices, generally through a switch fabric. Current supported baud rates are 2.5, 5.0, and 10.0Gbps. Ports use 1, 4, or 12 links in parallel to scale bandwidth as needed. The protocol utilizes 8B/10B block encoding.

A selection of the above standards are described in more detail in the remainder of this chapter. Detailed references for these standards documents are provided in Sect. 5.6.

5.1 SONET/SDH Networks

The original work on the Synchronous Optical NETwork (SONET) standard was performed by Bell Communications Research (Bellcore), now Telcordia, which was founded as the result of the breakup of American Telephone and Telegraph (AT&T) in 1984. Prior to SONET, higher bandwidth telecommunications equipment tended to use proprietary specifications and

Marvell Semiconductor, Inc. - Ex. 1008, Page 000061

OIF has collaborated with the developers of an open source software tool called StatEye to provide an implementation of the mathematics specified in the CEI Implementation Agreement [12]. Certain releases of this tool include templates for the various CEI variants. StatEye performs the necessary statistical signal integrity analysis using these templates.

5.3 Ethernet Protocols

The Institute of Electrical and Electronics Engineers (IEEE) commissioned a committee to develop open network standards. Since this committee started work on this effort in February, 1980, the documents produced by this project were designated using the nomenclature IEEE 802.x. Several different network architectures were ultimately standardized, resulting in various ".x" extensions. The "Ethernet" system originally developed by Xerox was standardized and published as "IEEE 802.3 Carrier Sense Multiple Access with Collision Detection (CSMA/CD) Access Method and Physical Layer Specifications" in 1985. Although the IEEE document does not use the term, the "Ethernet" name continues to be commonly associated with this standard. While Ethernet was once one of several network standards with significant deployment (for example: the 802.5 Token Ring), over time it has emerged as the dominant network standard. The IEEE 802.3 document has gone through several revisions, adding additional clauses to support additional speeds, physical layers, etc., since its original publication.

The historical network topology for an Ethernet system consisted of multiple workstations connected to a common Ethernet bus using coaxial cable. Whenever one of the workstations wanted to send data, the network device would first listen to the bus to determine whether another workstation was using it. If the bus was in use, the device would wait. If the bus was not in use, the network device would transmit a data packet, while continuing to monitor the bus for any collisions resulting from two devices starting transmission at the same time. If a collision occurred, the device would abort the transmission and wait some amount of time before trying again. The CSMA/CD portion of the title for the IEEE 802.3 document results from this historical network access method.

Beginning in the early 1990s, "star-connected" network topologies became the network configuration of choice. In a "star-connected" topology, each workstation connects to a central network unit (either a network switch or a network hub), and each connection is a point-to-point link implemented using twisted-pair wire or optical fiber. Such topologies allowed full-duplex connections such that a network device can be both transmitting and receiving at the same time. Since each link had only one transmitter, collisions were no longer possible. Nevertheless, the CSMA/CD access methods were retained in the IEEE 802.3 standard through the 10 Mbps, 100 Mbps, and 1 Gbps generations. Clauses pertaining to 10 Gbps recognize that practical implementations are point-to-point links, and no longer support CSMA/CD access methods.

Marvell Semiconductor, Inc. - Ex. 1008, Page 000062

5.3.1 Physical Layer Reference Model

Figure 5.14 illustrates the Media Access Control (MAC) layer and the various components (with the associated Ethernet terminology) of the Physical Layer associated with 10-Gbps Ethernet. The Reconciliation Layer remaps signals between the 32-bit MAC/PLS service interface definition (clause 36) and the 10-Gbps Ethernet Media-Independent Interface (XGMII) definition. The MAC layer (via the Reconciliation Layer) and the Physical Layer are interconnected using the XGMII. The Physical Layer consists of the Physical Coding Sublayer (PCS), Physical Medium Attachment (PMA) sublayer, and the Physical Medium Dependent (PMD) sublayer. There are multiple clauses of the IEEE 802.3 document which apply to these sublayers, depending on the type of medium being used. HSS cores are used in the high-speed electrical implementation of these layers.

Table 5.10 describes the relationship between 10-Gbps Ethernet variants and normative clauses of the IEEE 802.3 document for the physical sublayers. The published clauses relevant to 10-Gb Ethernet are summarized below:

Clause 44. Introduction to the 10Gbps Baseband Network: This clause is a normative summary of applicable requirements for the 10-Gbps LAN/WAN variants described in Table 5.10. This clause references other 10Gbps clauses.

Clause 46. Reconciliation Sublayer (RS) and 10 Gigabit Media Independent Interface (XGMII): This clause defines both the Reconciliation Sublayer (RS) and the XGMII.

Clause 47. XGMII Extender Sublayer (XGXS) and 10 Gigabit Attachment Unit Interface (XAUI): This clause defines a method of extending the physical distance between the MAC layer device and the PCS/PMA layer device. An XGXS layer device is located near the MAC layer device (DTE XGXS), and a similar XGXS layer device is located near the PCS/PMA layer device (PHY XGXS). XAUI is the interface definition which interconnects the two XGXS layer devices.

Clause 48. Physical Coding Sublayer (PCS) and Physical Medium Attachment (PMA) sublayer, type 10GBASE-X: This clause defines the PCS and PMA layers for connecting an XGMII to a PMD using a 4 lanes by 3.125Gbps electrical interface. Block encoding/decoding (8B/10B), alignment, deskew, and clock rate compensation are components of this clause.

Clause 49. Physical Coding Sublayer (PCS) for 64b/66b, type 10GBASE-R: This clause defines the PCS layer for connecting an XGMII to a 10-Gbps serial PMA layer. Block encoding/decoding (64B/66B), alignment, and clock rate compensation are components of this clause.

Clause 50. WAN Interface Sublayer (WIS), type 10GBASE-W: This clause defines a sublayer which exists between the PCS and the PMA layer for Wide Area Network (WAN) variants. This sublayer maps Ethernet packets into the Synchronous Payload Envelope (SPE) of an STS-192c SONET frame.

Marvell Semiconductor, Inc. - Ex. 1008, Page 000063

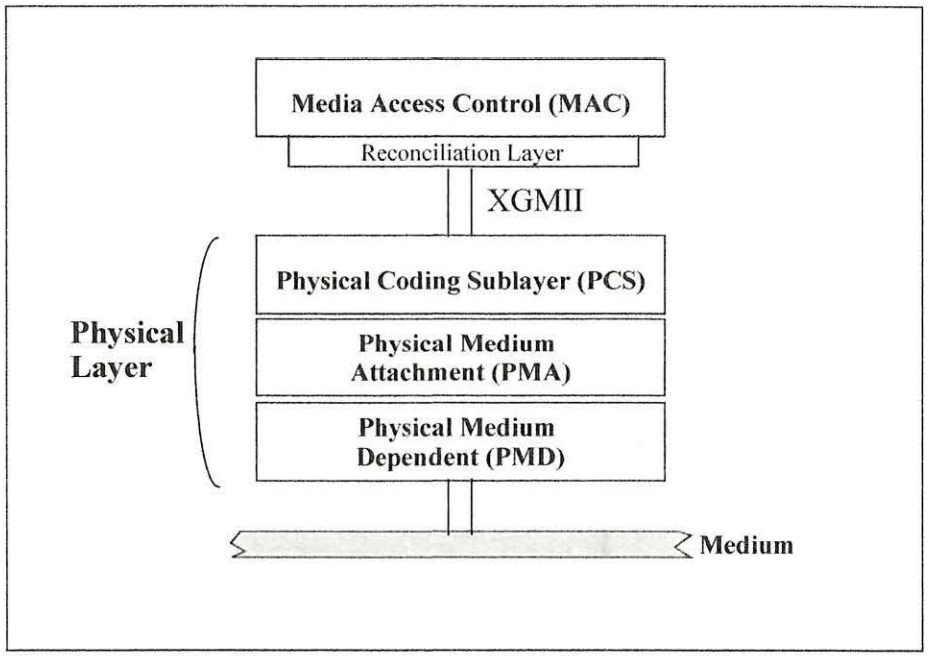


Fig. 5.14 LAN PHY/WAN PHY sublayers

Table 5.10 Relevant IEEE 802.3 clauses

Nomenclature	Type	PMD description	PCS layer clause	PMA layer clause	PMD layer clause	WIS (Clause 50)	Auto-Neg. (Clause 73)	FEC (Clause 74)
10GBASE-SR	LAN	850-nm serial optics	49	51	52			
10GBASE-LR	LAN	1,310-nm serial optics	49	51	52			
10GBASE-ER	LAN	1,550-nm serial optics	49	51	52			
10GBASE-LX4	LAN	4 x 1,310-nm CWDM optics	48	48	53			
10GBASE-CX4	LAN	4 x 3.125 Gbps electrical i/f	48	48	54			
10GBASE-SW	WAN	850-nm serial optics	49	51	52	X		
10GBASE-LW	WAN	1,310-nm serial optics	49	51	52	X		
10GBASE-EW	WAN	1,550-nm serial optics	49	51	52	X		
10GBASE-KX4	Backplane	4 x 3.125 Gbps electrical	48	48	71		X	
10GBASE-KR	Backplane	10 x Gbps serial electrical	49	51	72		X	O
Key: X = Mandatory, O = Optional								

Marvell Semiconductor, Inc. - Ex. 1008, Page 000064

Clause 51 Physical Medium Attachment (PMA) sublayer, serial: This clause defines a PMA sublayer for interfacing the PCS sublayer defined in clause 49 (or the WIS sublayer defined in clause 50) to a PMD using a serial 10.3125-Gbps electrical interface.

Clause 52 Physical Medium Dependent (PMD) sublayer and baseband medium, type 10GBASE-S (Short Wavelength Serial) and 10GBASE-L (Long Wavelength Serial) and 10GBASE-E (Extra Long Wavelength Serial): This clause defines the electrical-to-optical and optical-to-electrical PMD layer for 10-Gbps serial optical devices.

Clause 53 Physical Medium Dependent (PMD) sublayer and baseband medium, type 10GBASE-LX4: This clause defines the electrical-to-optical and optical-to-electrical PMD layer for a 10-Gbps optical device which employs Course Wave Division Multiplexing (CWDM) to transmit four 3.125Gbps data streams on the fiber.

Clause 54 Physical Medium Dependent (PMD) sublayer and baseband medium, type 10GBASE-CX4: This clause defines the electrical PMD layer for a 10-Gbps electrical device which uses four 3.125-Gbps electrical differential signals to transmit data across a cable.

The Ethernet variants also exist for communication across backplanes within host systems. The clauses relevant to 10-Gb Backplane Ethernet are summarized below:

Clause 69. Introduction to Ethernet operation over electrical backplanes: This clause is a normative summary of applicable requirements for the 1 Gbps and 10 Gbps backplane variants described in Table 5.10. This clause references other clauses applicable to these variants.

Clause 71. Physical Medium Dependent Sublayer and Baseband Medium, Type 10GBASE-KX4: This clause defines the electrical PMD layer for a 10-Gbps electrical device which uses four 3.125-Gbps electrical differential signals to transmit data across a backplane.

Clause 72. Physical Medium Dependent Sublayer and Baseband Medium, Type 10GBASE-KR: This clause defines the electrical PMD layer for a 10-Gbps serial electrical device which uses a 10.3125-Gbps electrical differential signal to transmit data across a backplane. This clause additionally defines a training frame structure and training state machine. This training protocol monitors the signal metrics at the receiver during link initialization and updates FFE coefficients in the transmitter in order to optimize performance of the link.

Clause 73. Auto-Negotiation for Backplane Ethernet: This clause defines autonegotiation protocols which are mandatory for PCS/PMA layer implementations for Backplane Ethernet. This protocol allows devices to negotiate which Backplane Ethernet variant is to be used, whether FEC is to be used, and other parameters.

Marvell Semiconductor, Inc. - Ex. 1008, Page 000065

Clause 74. Forward Error Correction (FEC) sublayer for 10GBASE-R: This clause defines an optional FEC sublayer for enhancing the BER performance of 10GBASE-KR.

5.3.2 Media Access Control (MAC) Layer

In this section the basic format of an Ethernet frame is discussed, as well as the XGMII interface definition. Other details of MAC layer function are beyond the scope of this text.

5.3.2.1 Ethernet Packet Format

Ethernet is a packet delivery system. The basic format of an Ethernet packet is shown in Fig. 5.15. Each packet consists of a preamble and start frame delimiter (SFD) to indicate the beginning of a packet, destination and source addresses to identify the recipient and the sender, a packet length field indicating the number of data bytes in the packet, a variable length data field, and a frame check sequence (FCS) field to facilitate error detection.

Each Ethernet network interface card (NIC) is assigned a unique MAC address by its manufacturer. The first 24 bits of the 48-bit MAC address identify the manufacturer and are assigned by the IEEE Registration Authority. The remaining bits are assigned by the manufacturer and are generally programmed into the hardware such that they cannot be changed. This MAC address uniquely identifies the NIC, even if the hardware moves to another location.

The data portion of the MAC frame generally carries data for a higher layer protocol, which also requires some control information to be transmitted. The data portion of the frame in Fig. 5.15 often consists of a logical link control (LLC) layer header followed by payload data. If there are fewer than 46 bytes of payload data, then the payload data is padded to achieve this minimum length.

Preamble (7 bytes)	SFD (1)	Destination Address (6 bytes)	Source Address (6 bytes)	Length (2)	Data (Up to 1500 bytes)	FCS (4 bytes)

Preamble: Idles which set timing.
Start Frame Delimiter (SFD): 'AB'h indicating start of frame.
Destination Address: Six byte unique MAC address of recipient device.
Source Address: Six byte unique MAC address of sending device.
Length: Two bytes indicating length of data field.
Data: Layer 2 Payload Data field. Generally contains a Logical Link Control (LLC) header and between 46 and 1500 bytes of payload data. Data is padded if shorter than 46 bytes.
Frame Check Sequence (FCS): Cyclic Redundancy Check (CRC) remainder for error detection.

Fig. 5.15 IEEE 802.3 ethernet frame

Marvell Semiconductor, Inc. - Ex. 1008, Page 000066

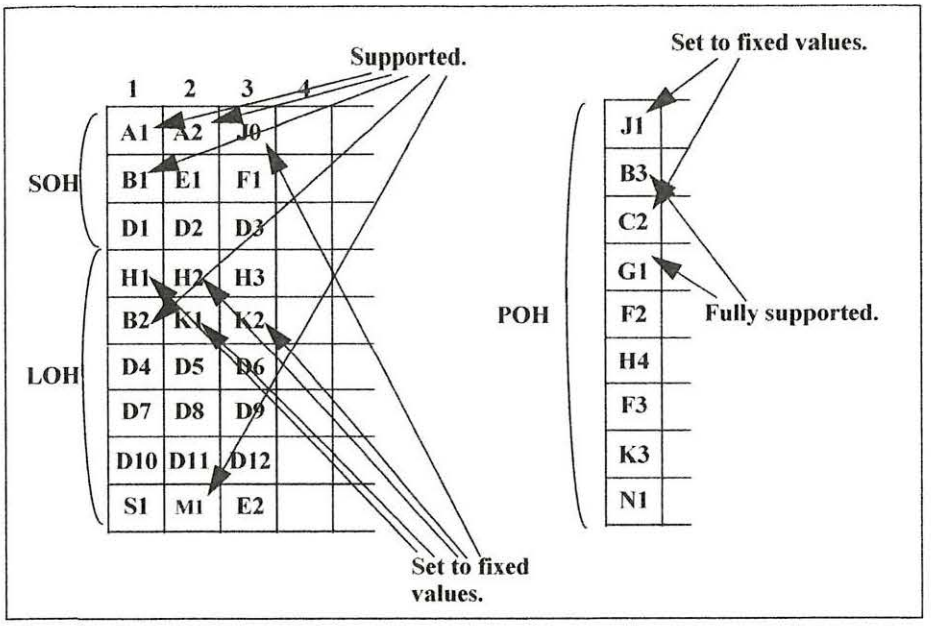


Fig. 5.20 WIS support for SONET/SDH overhead

Overhead bytes not listed above are not supported. They are set to fixed values and are not checked.

One last note on the WIS sublayer is in regard to the orientation of most significant bit and least significant bits on the interface between this layer and other sublayers. Ethernet bits are transmitted least significant bit first, while SONET/SDH octets are transmitted most significant bit first. For this reason, bit/octet ordering is interposed between the PCS and the PMA sublayers by the WIS sublayer.

5.3.5 Backplane Ethernet

There are three significant features associated with Backplane Ethernet which are not used in other Ethernet variants. These features are:

- Training protocol (defined in Clause 72 for 10GBASE-KR)
- Autonegotiation (defined in Clause 73 for all Backplane Ethernet variants)
- Forward error correction (FEC) (defined in Clause 74 for optional use with 10GBASE-KR)

5.3.5.1 Training Protocol

The training protocol is included in the IEEE 802.3 clause 72 specification for the 10GBASE-KR variant and must be executed upon start-up to initialize the PMD for this variant. This protocol continuously exchanges fixed-length training frames between the PMD and the link partner. The training frames

Marvell Semiconductor, Inc. - Ex. 1008, Page 000067

contain both a pseudorandom training pattern and an FFE coefficient control field. During the training process, each PMD monitors the quality of the received training pattern and generates FFE coefficient control commands to the PMD at the other end of the link. At the same time the PMD monitors received FFE coefficient control commands and updates the HSS configuration accordingly. It is important to note that Ethernet assumes full-duplex communication between PMDs, making it possible to perform two-way communication to negotiate link parameters.

Training Frames:

The training frame is a fixed length frame containing the following fields:

- *Frame Marker.* A four byte unique marker pattern denoting the start of the training frame. The marker pattern is "FFFF0000" hexadecimal. This marker is transmitted at 10.3125 Gbps.
- *Coefficient Update.* A 16-bit coefficient update field which allows the PMD to request changes to the FFE coefficients being used by the PMD at the other end of the link. This 16-bit coefficient field is transmitted using Manchester coding at a baud rate which is one-eighth of the normal 10.3125 Gbps baud rate. Transmission of this 16-bit field is the equivalent of transmitting 16 bytes at 10.3125 Gbps.
- *Status Report.* A 16-bit status field which indicates whether the PMD has finished its training process, as well as the status of any requested FFE coefficient updates. This 16-bit coefficient field is transmitted using Manchester coding at a baud rate which is one-eighth of the normal 10.3125 Gbps baud rate. Transmission of this 16-bit field is the equivalent of transmitting 16 bytes at 10.3125 Gbps.
- *Training Pattern.* Pseudorandom data generated with the polynomial: $G(x) = 1 + x^9 + x^{11}$
  This field is 512 bytes in length (4,094 bits PRBS bits followed by two zero bits), transmitted at 10.3125 Gbps.

The purpose of using low-speed Manchester encoding for the *Coefficient Update* and *Status Report* fields is to ensure reliable information is communicated even when the link performance is not optimized. These fields can be implemented by transmitting the parallel data byte "00000000" or "11111111" to represent a Manchester encoded "0" bit, and transmitting the parallel data byte "00001111" or "11110000" to represent a Manchester encoded "1" bit. (The byte pattern used to denote a Manchester encoded "0" or "1" is selected to ensure that a transition always occurs between bits.)

The *Coefficient Update* field contains the following subfields:

- *Preset.* Forces all FFE coefficients to be set to a state where equalization is turned off (i.e., coefficients for the main FFE tap is set to its maximum value, and coefficients for precursor and postcursor taps are set to zero).
- *Initialize.* Forces FFE to be turned on and coefficients to be set to predefined initialization values.

Marvell Semiconductor, Inc. - Ex. 1008, Page 000068

- *Coefficient* (+1) *Update.* Directs that the postcursor FFE coefficient shall be incremented, decremented, or held at current value. .
- *Coefficient* (0) *Update.* Directs that the main FFE coefficient shall be incremented, decremented, or held at current value.
- *Coefficient* (-1) *Update.* Directs that the precursor FFE coefficient shall be incremented, decremented, or held at current value.

Update requests to increment or decrement coefficients should only be sent if the received *Status Report* field corresponding to the coefficient indicates the coefficient is in *not updated* state.

The *Status Report* field contains the following subfields:

- *Receiver Ready.* Indicates that reporting node has finished training its receiver and is ready to receive data.
- *Coefficient* (+1) *Status.* Reports that the postcursor FFE coefficient is *not updated*, is *updated*, is at *maximum* value, or is at *minimum* value. The PMD initially reports all coefficient status as *not updated.* After receiving a Coefficient Update request to update the coefficient, the PMD updates the coefficient and then reports either *updated*, *maximum*, or *minimum* status. After receiving a Coefficient Update request to hold the previous value, the PMD returns status to *not updated.*
- *Coefficient* (0) *Status*. Reports that the main FFE coefficient is *not updated*, is *updated*, is at *maximum* value, or is at *minimum* value. Status transitions are similar to the description for the postcursor FFE coefficient.
- *Coefficient* (–1) *Status*. Reports that the precursor FFE coefficient is *not updated*, is *updated*, is at *maximum* value, or is at *minimum* value. Status transitions are similar to the description for the postcursor FFE coefficient.

The *coefficient status* and *coefficient update* fields associated with each FFE coefficient operate according to a simple two-way handshake as described above. Each PMD assesses the quality of the received signal (through analysis of the training pattern), and then instructs its partner node to update FFE coefficients in an attempt to improve the quality of the signal.

Training State Machine:

The training state machine is also defined by clause 72. When a node detects a signal on the link, it starts sending training frames and waits for reception of valid training frames from its link partner. Once training frames are being received, the node allows the receiver equalization to train, and determines whether updates are required to the transmitter equalization. FFE coefficient updates are requested via command bits in the transmitted training frame; the link partner's acknowledgement of these requests are communicated via the status bits in the received training frame. Simultaneously, the node monitors command bits in the received training frame to determine whether local FFE coefficient updates are being requested

Marvell Semiconductor, Inc. - Ex. 1008, Page 000069

by the link partner, and if applicable executes these requests and returns an acknowledgement via the transmitted training frame.

It should be noted that clause 72 does *not* specify the algorithms which should be employed to determine optimal equalizer settings. These algorithms are left as an exercise to the designer, and some implementations may even choose to use preset FFE coefficient values. (Although such implementations may not work well in some system configurations.)

The methods of determining signal quality and criteria for exiting training are also *not* specified by clause 72. These methods and criteria may be based on the PRBS pattern in the training frame, or may use other proprietary features of the HSS core.

HSS Features:

Some HSS cores may implement features which aid in signal quality assessment and allow faster updates to FFE coefficient values. Features such as the following are helpful to the implementation of the clause 72 Training Protocol:

- FFE coefficient negotiation requires assessment of the eye quality of the signal being received by the receive PMD layer. This signal quality is not directly observable by measurements at higher sublayers. The HSS EX10 digital eye feature assesses eye quality and provides training state machines with quick access to these measurements. This is useful for implementing the training protocol.
- FFE coefficient updates by a partner link during the training process should be made in a timely manner. HSS core features to quickly reset, load, increment, or decrement FFE coefficients speed up training.

5.3.5.2 Autonegotiation

The autonegotiation protocol is defined in IEEE 802.3 clause 73. Backplane Ethernet PHYs are required to implement the autonegotiation protocol, however, use of the protocol is under the control of the management interface and is optional. The protocol allows a device to advertise the modes of operation it supports to another device at the remote end of a backplane Ethernet link, and to detect the corresponding operational modes being advertised by the other device. The objective is to allow the devices to agree to a configuration that maximizes the performance of the link. The protocol additionally ensures that the PHY device is attached to its link partner at the remote end of the link, and is not responding to a crosstalk signal.

Autonegotiation is performed using low-speed Manchester encoding of the information to be exchanged. Clause 73 describes a number of Differential Manchester Encoding (DME) pages which are exchanged by the protocol. Autonegotiation supports the following partial list of functions:

- Advertise which backplane Ethernet variants are supported and negotiate which variant should be selected.
- Advertise whether the interface supports FEC and negotiate whether to enable FEC.

Marvell Semiconductor, Inc. - Ex. 1008, Page 000070

for the target BER. The measurement must also be based on sufficiently random data to ensure all possible pattern-dependent effects have been observed. A typical oscilloscope providing a ± 3 sigma histogram of eye width values is not an accurate portrayal of the eye width for a 10^{-12} BER, and using such a measurement understates the jitter of the system.

Traditionally, jitter is subdivided into the following two components: *deterministic jitter* (DJ), and *random jitter* (RJ). Each of these categories accumulates differently in the link and results in different requirements for compliance and budgeting schemes.

Deterministic jitter. This is the amount of the total jitter for which the jitter distribution is non-Gaussian. Deterministic jitter is always bounded in amplitude and is created by specific, identifiable causes. The terminology used in [1] for DJ is *high probability jitter* (HPJ). Four types of jitter are typically included as part of the DJ component:

- *Duty cycle distortion* (DCD) results from the difference in width between a logic "0" and a logic "1." This element of DJ is the result of a driver circuit that has rise and fall times that are not equal. Another cause of DCD results when a DC voltage offset is present between the true and complement legs of the differential signal. Pulse width shrinkage due to passive or active components of the channel may also be a factor. DCD is sometimes called *pulse width distortion.*
- *Data dependent jitter* (DDJ) includes timing variations that result from nonclocklike serial data waveforms as they propagate through a channel with bandwidth limitations. Given knowledge of the preceding and subsequent bits of the transmission, the DDJ component of the jitter is predictable, and therefore may be corrected through equalization. DDJ is also called *pattern dependent jitter* or *intersymbol interference* (ISI).
- *Periodic jitter* (PJ) is jitter which has a single fundamental harmonic plus possible multiple even and odd harmonics. PJ results from various electromagnetic noise sources in the system such as power supply noise and crosstalk from periodic signals. Clock signals are periodic signals which cause crosstalk that results in PJ on the victim signal.
- *Sinusoidal jitter* (SJ) is jitter which has a single fundamental harmonic and no additional harmonics. Sinusoidal jitter is generally defined in the context of applied SJ for jitter tolerance testing of a receiver device. For this reason it is generally considered separate from the periodic jitter which arises from sources in the system.
- *Bounded uncorrelated jitter* (BUJ) includes all components of non-Gaussian jitter which are not included in the various components listed above. Nonclock crosstalk aggressor signals operating at a baud rate that is synchronous to the baud rate of the victim signal produce jitter on the victim which is non-Gaussian and is also not correlated with data on the victim signal.

Marvell Semiconductor, Inc. - Ex. 1008, Page 000071

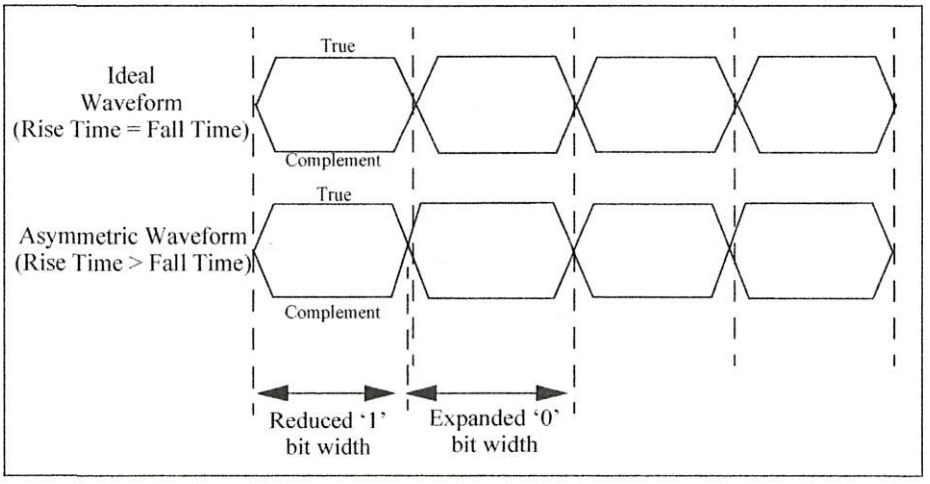


Fig. 8.4 DCD resulting from rise/fall time asymmetry

For any jitter distribution, the overall mean (μ) is zero, and therefore:

$$\mu = (\mu_1 + \mu_2)/2 = 0$$

$$A = (\mu_1 - \mu_2)/2 = \mathrm{DJ}_{\mathrm{DCD}}$$

where $\mathrm{DJ}_{\mathrm{DCD}}$ is the peak-to-peak value of the DCD jitter component, and μ and A are characteristics of the dual-Dirac PDF defined in (8.9). The resulting PDF for DCD is therefore defined by the following equation [5]:

$$\mathrm{DCD}(x) = 0.5\left[\delta\left(x - \frac{\mathrm{DJ}_{\mathrm{DCD}}}{2}\right) + \delta\left(x + \frac{\mathrm{DJ}_{\mathrm{DCD}}}{2}\right)\right] \tag{8.11}$$

Since DCD jitter is correlated with the data being transmitted, equalization circuits can compensate for this jitter component.

8.2.2.2 Data Dependent Jitter

Data dependent jitter refers to the timing variations caused by the bandwidth limitations of the channel being traversed by the signal. This type of jitter, also known as pattern dependent jitter or intersymbol interference (ISI), is observed on nonclocklike waveforms since the frequency spectrum of such signals is continually changing. DDJ leads to varying amounts of signal attenuation and phase delay when exposed to the frequency response characteristics of a typical transmission channel. Such effects are not observed on clock waveforms which have constant frequency components.

Virtually all data channels exhibit increasing insertion loss as the frequency of the signal increases. More expensive channel components (exotic materials and more costly design and fabrication techniques) can mitigate these effects to a degree, but high baud rates still expose these signals to a significant amount of signal attenuation resulting in DDJ contribution.

Marvell Semiconductor, Inc. - Ex. 1008, Page 000072

Table 8.3 CEI-11G-LR informative jitter budget [1]

Source	Uncorrelated jitter		Correlated jitter		Total jitter			
	Un-bounded Gaussian	Bounded high prob.	Bounded Gaussian	Bounded high prob.	Gaussian	Sinu-soidal	High prob.	Total
Abbreviation	UUGJ	UBHPJ	CBGJ	CBHPJ				
Unit	UIpp	UIpp	UIpp	UIpp	UIpp	UIpp	UIpp	UIpp
Transmitter	**0.150[a]**	**0.150[a]**			**0.150**		**0.150**	**0.300**
Channel			0.230	0.400				
Receiver input	**0.150**	**0.150**	**0.230**	**0.400[a]**	**0.275**		**0.550**	**0.825**
Equalizer				-0.300				
Post Equalizer	**0.150[a]**	**0.150[a]**	**0.230**	**0.100[a]**	**0.275**		**0.250**	**0.525**
DFE penalties				0.100				
Clock and Sampler	0.150	0.100		0.100				
Budget	**0.212**	**0.250**	**0.230**	**0.300**	**0.313**	**0.050**	**0.550**	**0.913**

Note:

[a]These values are normative values in [1]

[b]Due to receiver equalization, it reduces the ISI as seen inside the receiver. Thus this number is negative

[c]It is assumed that the eye is closed at the receiver, hence receiver equalization is required

©2008 Optical internetworking forum. All rights reserved. Used under permission.

Contributions of the UUGJ and CBGJ columns are combined in the *Gaussian* column of the *total jitter*, while contributions of the UBHPJ and CBHPJ columns are combined in the *high probability* column. Note that while the High Probability jitter is combined by adding the component jitter values, the Gaussian jitter must be combined using an RMS summation. This is because the probability of independent Gaussian events all having worst case values is extremely unlikely. On the other hand, the RMS summation of UUGJ and CBGJ produces a Gaussian jitter value corresponding to a BER consistent with the component numbers.

The SJ column reserves a portion of the jitter budget for applied SJ as part of jitter tolerance testing. This jitter component does not exist in an operational system.

Finally, the *Gaussian*, *sinusoidal*, and *high probability* jitter totals are summed to produce an overall *total jitter*.

The table assumes the jitter contribution of the transmitter device is entirely uncorrelated while distortion in the channel is entirely correlated. The sum of

Marvell Semiconductor, Inc. - Ex. 1008, Page 000073

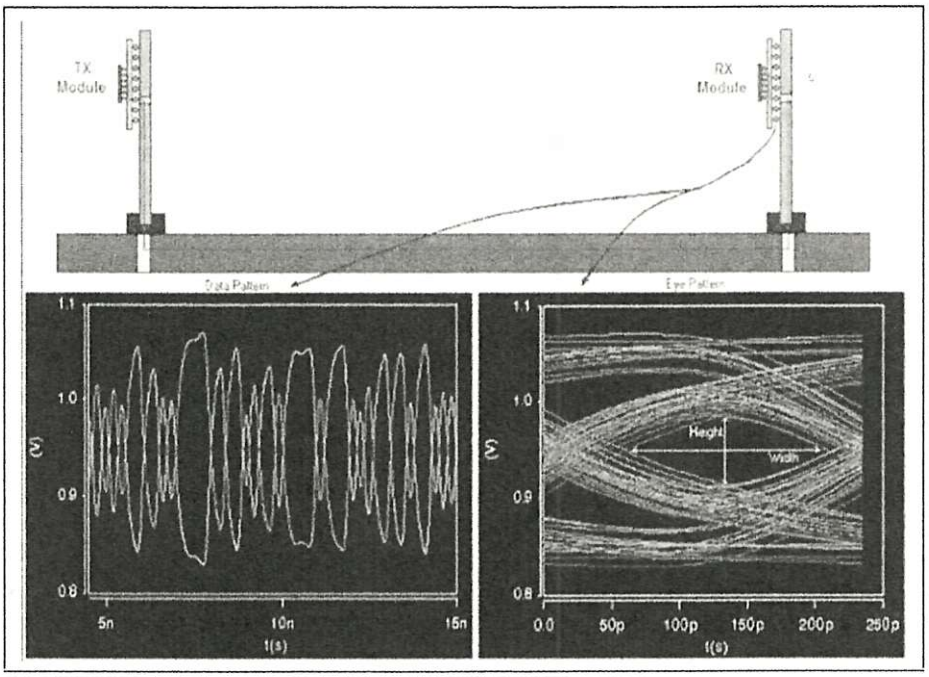


Fig. 8.14 Typical Spice simulation results

8.4 Statistical Approach to Signal Integrity

Circuit simulation using Spice models provides an accurate measure of the resulting signal waveform given the input conditions which are simulated. However, unless the input conditions are varied over a statistically representative range, Spice simulations do not guarantee that the BER of the system meets the specification. A BER of 10^{-b} implies that the signal may violate the specified eye mask no more often than once every 10^{b} bits. Many, many data patterns must be simulated under a variety of crosstalk and noise conditions to ensure the resulting signal meets this requirement. Lengthy simulations are required to guarantee a BER of 10^{-12} and simulations are prohibitively long for higher BER specifications.

At higher baud rates, intersymbol interference (ISI) becomes a key source of signal distortion. ISI is a component of data dependent jitter (DDJ), and is primarily caused by frequency response limitations of the channel. At higher baud rates, ISI effects stretch over multiple bit intervals, with the waveform of the current bit being affected by previous bits. In many cases the eye at the receiver is closed due to ISI, and receiver equalization is required to compensate. The receiver equalization must be included in the analysis in such cases to ensure the signal can be received properly and that the BER specification is met. In a Spice circuit simulation environment, this would be computationally prohibitive.

Marvell Semiconductor, Inc. - Ex. 1008, Page 000074

The OIF common electrical I/O (CEI) Implementation Agreement [1] was discussed in Sect. 5.2.5. This standard specifies normative channel requirements, and a statistical analysis approach was developed in conjunction with the development of this standard to verify compliance of the channel with these requirements. This approach is described in detail in [3], and was subsequently published in [1] as the normative method of determining channel compliance. An open source software tool called *StatEye* is available from [4], and implements this analysis. Other software tools also exist which implement similar statistical approaches, including the IBM HSSCDR tool described in Sect. 8.4.2.

8.4.1 Analysis Approach

Statistical approaches to signal integrity analysis can produce reliable results with significantly less computation than circuit simulation approaches. Transmitter jitter generation is statistically modeled using the dual-Dirac model for deterministic components, and using the Gaussian model for random components. Measured frequency response models are used for components of jitter due to the channel. The resulting analysis can project jitter behavior and the corresponding eye opening of the signal for a given BER at either the input to the receiver or at the output of the receiver equalization circuit. Statistical signal analysis can model complex receiver equalization circuits in the simulation without significant computational penalties. Additionally, software used to perform statistical analysis often includes algorithms which can determine the optimal settings for both transmitter and receiver equalization circuits.

The statistical analysis approach presented in this section is also described in [1] and [3], and is representative of the analysis performed by this class of software tools.

8.4.1.1 Pulse Response

Fig. 8.15 illustrates an example of input and output signals of a channel. The channel in this example is modeled by a simple RC network. The input data pattern is "011010011100," and an ideal input signal is assumed in the figure. The voltage level achieved by each bit of the output waveform depends on whether the values of prior bits were the same or different. This type of data dependent jitter is called *intersymbol interference*. While this example modeled the channel with an RC network, realistic channels generally also have inductance and impedance mismatches. The addition of inductance potentially causes ringing of the response signal, and impedance mismatches cause reflections which may lag the bit transition by up to several bit times.

The pulse response of a channel is defined as the received pulse for an ideal square wave launched into the channel, where the pulse width of the square wave is one unit interval. This response is calculated either by convolving the pulse with the impulse response of the channel, or by multiplying the Fourier spectrum of the ideal transmitted square wave with the channel

Marvell Semiconductor, Inc. - Ex. 1008, Page 000075