SFI-4 Phase 2 Interface using the 64b/66b Line Code

Rick Walker

Agilent Laboratories, Palo Alto, CA

rick_walker@labs.agilent.com

Richard Dugan

Agilent Technologies, Semiconductor Products Group, San Jose, CA

richard_dugan@agilent.com

Contribution Number: OIF2001.586

Working Group: Physical and Link Layer (PLL) Working Group

Title: SFI-4 Phase 2 Interface using the 64b/66b Line Code

Source: Rick Walker

Agilent Labs

rick_walker@labs.agilent.com

Date: November 6, 2001

Abstract: Proposes a modified form of the 802.3ae 64b/66b code as a physical layer linecode for SFI-4 Phase 2: a protocol independent, 12.5 Gb/s, four wire, chip-to-chip interconnect interface.

For additional information contact:

The Optical Internetworking Forum, 39355 California Street,

Suite 307, Fremont CA 94538

510-608-5990 phone, info@oiforum.com

Topics

- Motivation
- Code Overview
- Coder Block Diagram
- Scrambling Principle
- Decoder Block Diagram
- Frame sync algorithm
- Summary

Motivation and Strategy

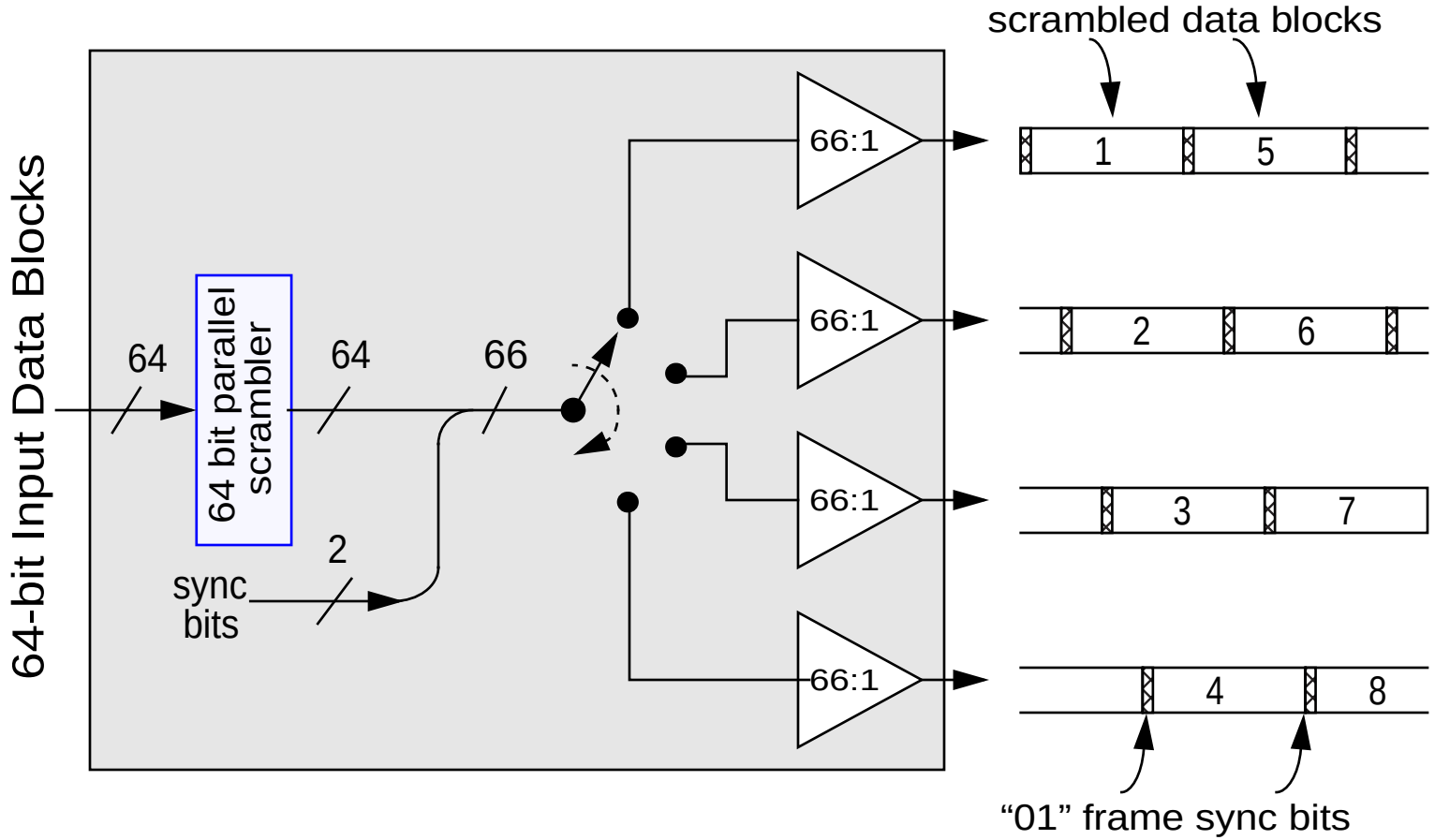
- Allow protocol-agnostic “virtual ribbon-cable” operation
- Leverage existing 64b/66b code to reduce design time and risk
- Reduce pin count, routing complexity, and power dissipation
- Eliminate deskew channel to reduce number of high-speed lines by 25%
- Minimize coding overhead to permit strong FEC

Code Overview

Use a simplified version of 64b/66b 10GbE linecode

- Data is transmitted in 8 byte (64-bit) blocks
- Each block is scrambled at the transmitter and descrambled at the receiver to ensure statistical DC-balance and transition density for RX PLL
- Block/byte alignment is provided by prepending each scrambled 64 bit block with two non-scrambled “01” sync bits, creating an overall 66 bit frame
- 802.3ae allows for non-data (control) frames indicated by a “10” sync pattern, but these codes are not needed, nor are proposed for SFI-4

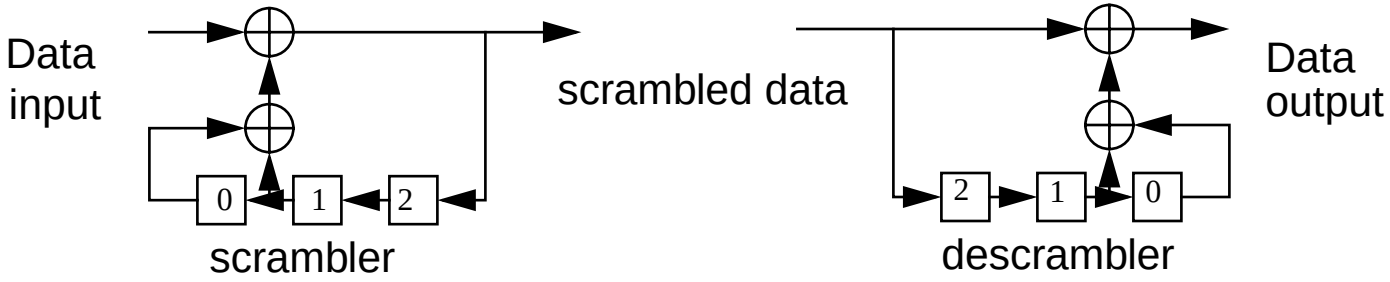
Coder Block Diagram



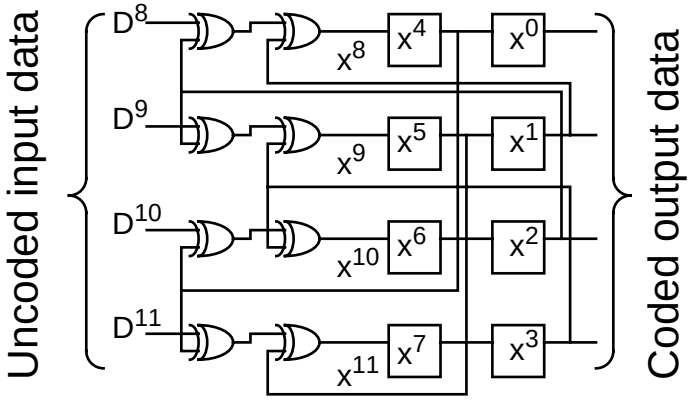
Coder is a simplified 64b/66b codec (data frames only), with 66 bit frames sent round-robin across the 4 output lanes.

Scrambling principle

An example 3-bit scrambler/descrambler in serial form:

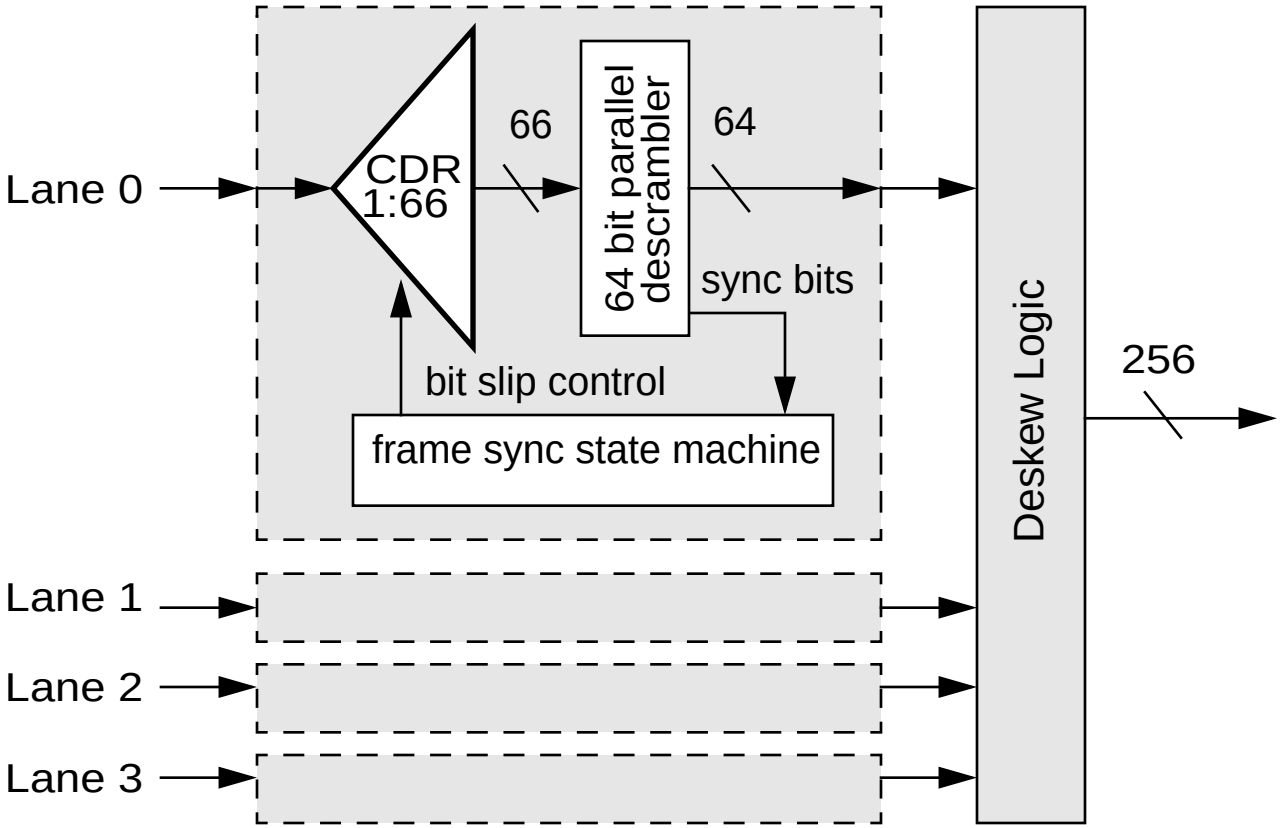


parallel form:



- Self synchronizing scrambler
- Can be parallelized for efficient implementation
- Using a long pattern length reduces possibility of jamming (eg: $x^{58}+x^{39}+1=0$)
- 64/66b's self-synchronous scrambler does not compromise Ethernet (802.3ae) CRC performance

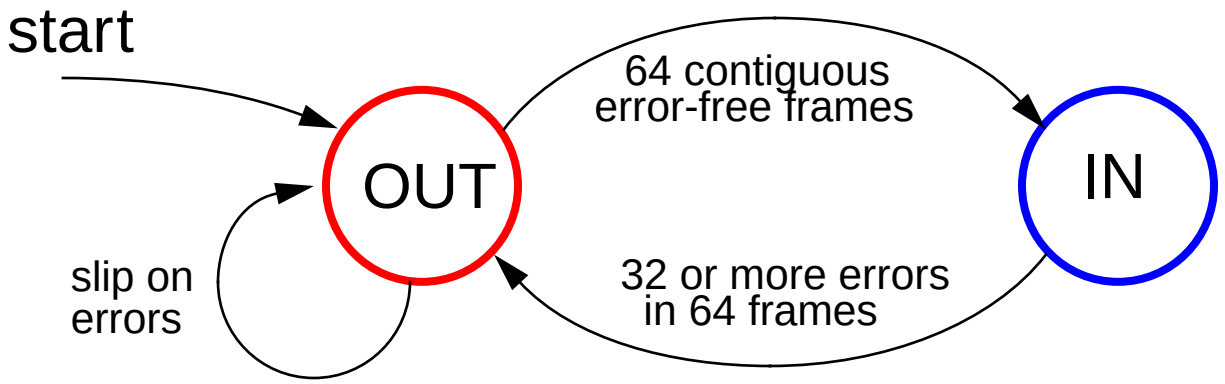
Decoder Block Diagram



Deskew logic uses recovered Lane 0 clock phases to optimally sample other lanes with +/- 16 bits of skew tolerance

Frame sync algorithm

- Look for the presence of “01” sync-bits every 66 bits. If bits 65 and 66 are not “01”, then slip the demux phase by one bit and try again
- Frame sync is acquired after 64 contiguous frames have been received with valid “01” headers
- Frame sync is declared lost if 32 invalid sync headers are received in any block of 64 frames



Comparison with SONET

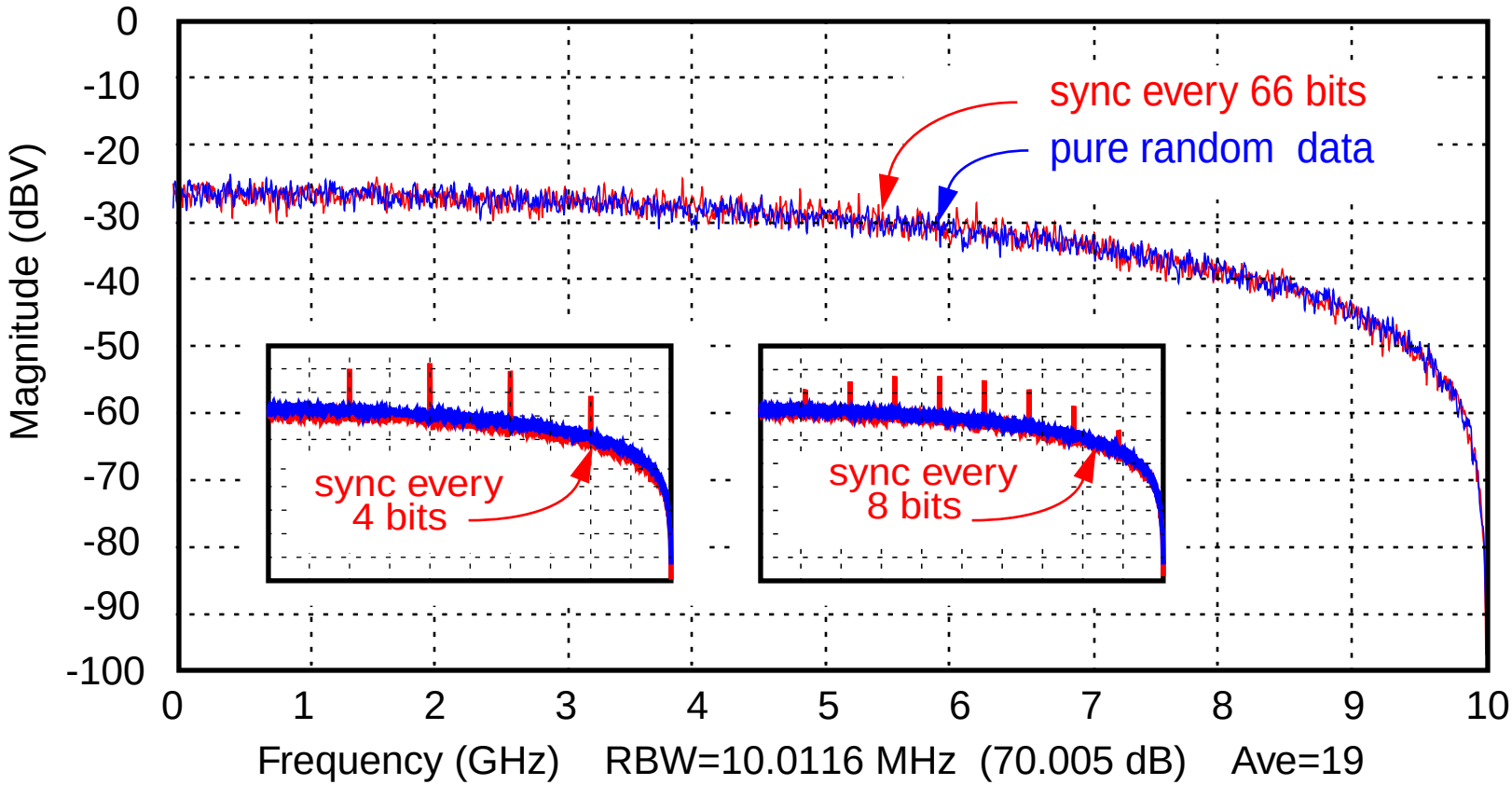
- two-bit preamble is analogous to SONET's A1/A2 sync bytes.
- SONET CDRs are designed to accommodate an 80 bit run length. This new code is deterministically limited to 64 bit run length due to the periodic preamble bits.
- Both codes use similar bit-slipping method to acquire frame sync.
- This code is similar in spirit to the SONET code. It is not in any way compatible, but inherits much of SONET properties while being much simpler to implement.
- Much less on-chip buffering required
- Lower latency

Summary

- Eliminates deskew lanes
- Fits within 70 pin connector
- Ensures DC balance on all lanes
- Limits run-length to a maximum of 66 bits
- Single 64b/66b codec required
- Simple deskew accomodates 16 bits of skew
- 3.125% coding overhead
- Protocol Agnostic (virtual ribbon cable)
- Leverages existing 802.3ae 64b/66b code

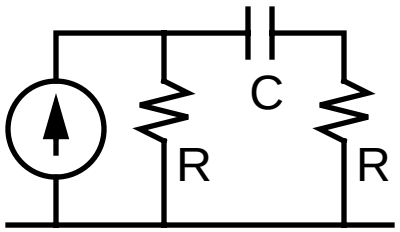
Supplemental slides

Spectral Properties



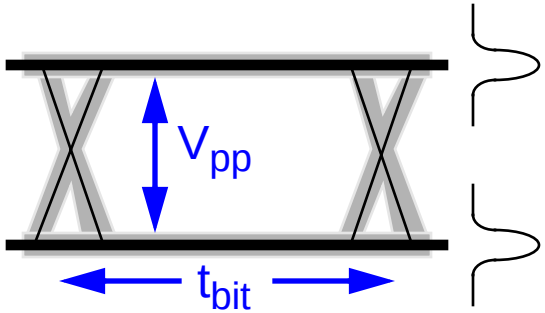
Spacing and power of spurs decrease with increasing frame size, becoming negligable by N=66.

Baseline Wander



A typical laser system will require a coupling capacitor at both TX and RX

time constant (tau) = 2*R*C



Eye diagram shows Gaussian wander (with stddev of sigma) due to statistical of 1's and 0's in charging coupling capacitor C.

The average value of N random bits is 1/2 with a sigma of 1/sqrt(N)
A coupling capacitor can be approximately considered to be performing a moving average over N bits, where N ~= 8*tau/(tbit)

$$\sigma(offset) = V_{pp}\sqrt{\frac{t_{bit}}{8\tau}}$$

Simulated Baseline Wander

1Tb of 10.3 Gb/s scrambled data through 0.01uf and 100ohms