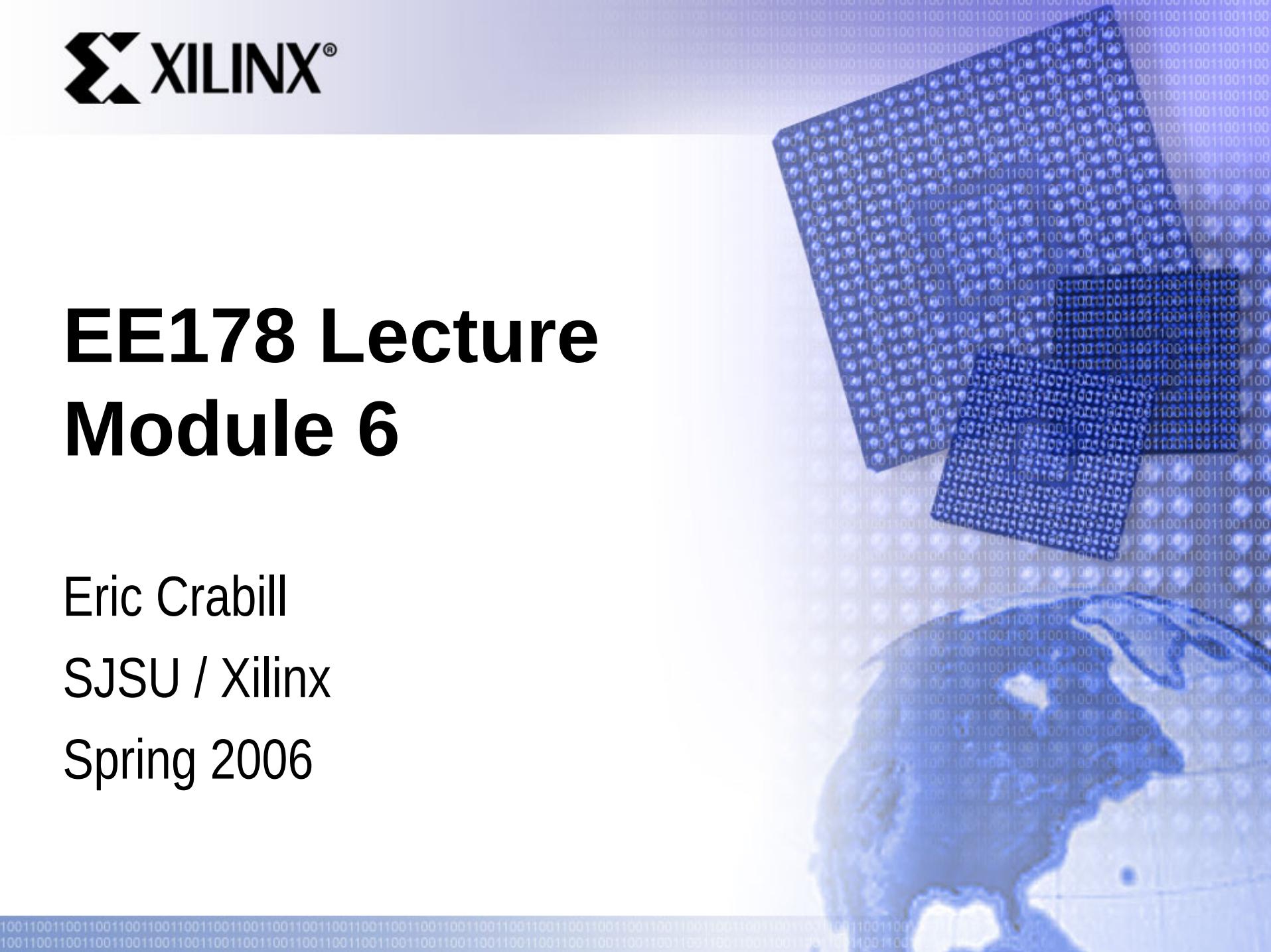


EE178 Lecture Module 6

Eric Crabill

SJSU / Xilinx

Spring 2006

Lecture #11 Agenda

- Suggested reading.
  - Spartan-3 Complete Datasheet.
- Overview of FPGA use at the PCB level.
  - General purpose pins.
  - Special and dedicated pins.
- Loading bitstreams -- FPGA configuration.
  - Configuration pins.
  - Configuration modes and examples.

Bigger Picture

- Your lab experience has been very convenient.
  - You are provided with a pre-designed board.
  - You are provided with mechanisms to program the FPGA device with your completed bitstream.
- In the future, you may need to design…
  - A mechanism to load bitstreams.
  - The board itself.

Spartan-3 FPGA Pins

- For other FPGAs, consult the device datasheet.
- As you would expect, the majority of the pins on the packaged device are general purpose I/O.
  - These pins are individually configured as input, output, or bidirectional by the configuration bitstream.
  - As a board designer, you may feel free to use these pins to interface to other devices on your board.
    - Use the pins “according to directions” in the datasheet.
    - Pay attention to FPGA device maximum ratings.

Spartan-3 FPGA Pins

- Additional things to keep in mind for general purpose I/O pins…
  - Some signaling standards may need special VREF.
  - Output voltages are banked and set by VCCO.
  - Differential signaling requires use of pin-pairs.
  - Observe simultaneous switching output guidelines.
  - Signal integrity and board-level simulations.

Spartan-3 FPGA Pins

- Quite a number of power and ground pins:
  - VCCINT pins are for a 1.2 volt power supply which powers the programmable logic.
  - VCCO pins are for the output driver power supplies and can be a range of voltages up to 3.3 volt; there are eight sets of VCCO, one per I/O bank.

Spartan-3 FPGA Pins

- Quite a number of power and ground pins:
  - VCCAUX pins are for a 2.5 volt power supply which powers the FPGA configuration interface as well as certain other internal functions.
  - VREF pins are input threshold references for certain I/O standards. Usable as general I/O if not needed.
  - GND pins, well… do these need explanation?

Spartan-3 FPGA Pins

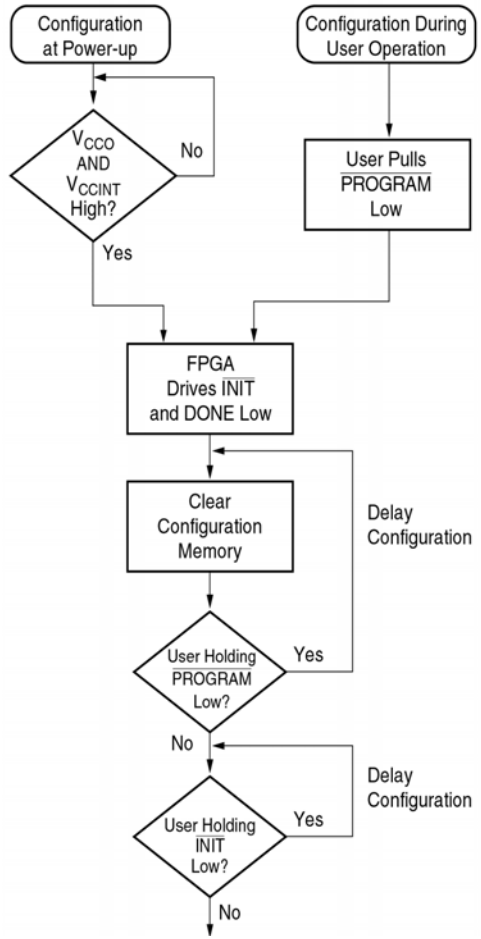
- There are also a few special clocking-related pins that have special properties:
  - GCLK0 through GCLK7 are special inputs for clocks.
  - These pins have special connections to global clock buffers and clock management circuits in the FPGA.
  - May be used as general purpose I/O pins if you don't need all of them for clock signals.

Configuration Pins

- There are a number of special pins for loading your bitstream into the Spartan-3 device.
- Start by looking at pins that are common to most Xilinx FPGAs and how configuration begins:
  - PROG#
  - INIT#
  - DONE
  - M2, M1, M0

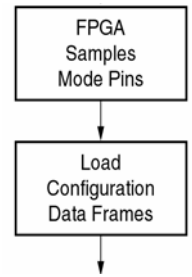
How Configuration Begins

- Configuration begins at power-on, or when PROG# is asserted.
- DONE is de-asserted.
- Configuration memory is cleared.
- Configuration can be delayed by holding PROG# or INIT#.



Loading the Bitstream

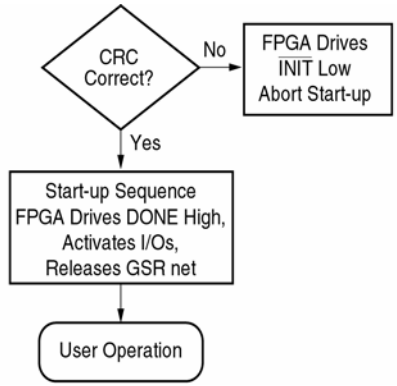
- Mode pins M2, M1, M0 tell the FPGA how the bitstream will be loaded.
- Bitstream is loaded…
- Bitstream size related to FPGA density; table is only for Spartan-3.

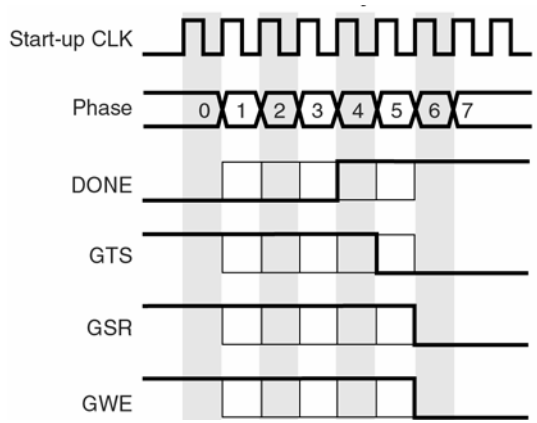


Device	File Sizes	Xilinx Platform Flash PROM	
		Serial Configuration	Parallel Configuration
XC3S50	439,264	XCF01S	XCF08P
XC3S200	1,047,616	XCF01S	XCF08P
XC3S400	1,699,136	XCF02S	XCF08P
XC3S1000	3,223,488	XCF04S	XCF08P
XC3S1500	5,214,784	XCF08P	XCF08P
XC3S2000	7,673,024	XCF08P	XCF08P
XC3S4000	11,316,864	XCF16P	XCF16P
XC3S5000	13,271,936	XCF16P	XCF16P

How Configuration Ends

- FPGA calculates CRC on bitstream while loading.
- After bitstream is loaded, if CRC is incorrect, the FPGA aborts… Why?
- If correct, the final start up sequence takes place.
- DONE is asserted, user operation begins.





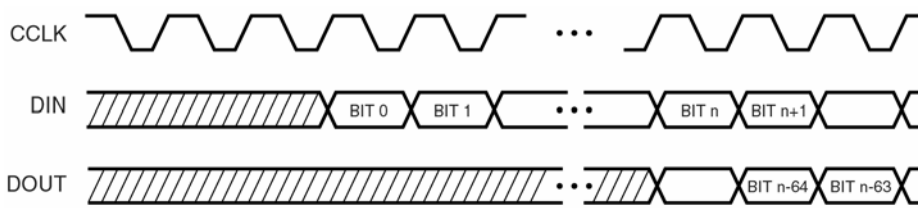
Configuration Modes

- For Spartan-3, there are five different modes.
- M2, M1, and M0 select the configuration mode.
- For other FPGAs, consult the device datasheet.
- Let's look at Master Serial and JTAG modes…

Configuration Mode	M0	M1	M2	Synchronizing Clock	Data Width
Master Serial	0	0	0	CCLK Output	1
Slave Serial	1	1	1	CCLK Input	1
Master Parallel	1	1	0	CCLK Output	8
Slave Parallel	0	1	1	CCLK Input	8
JTAG	1	0	1	TCK Input	1

Master Serial Mode

- Three pins are used in this serial mode.
  - FPGA outputs a configuration clock.
  - Every CCLK cycle, an external component provides the next bit of the bitstream via FPGA input DIN.
  - The FPGA mirrors the data on output DOUT.



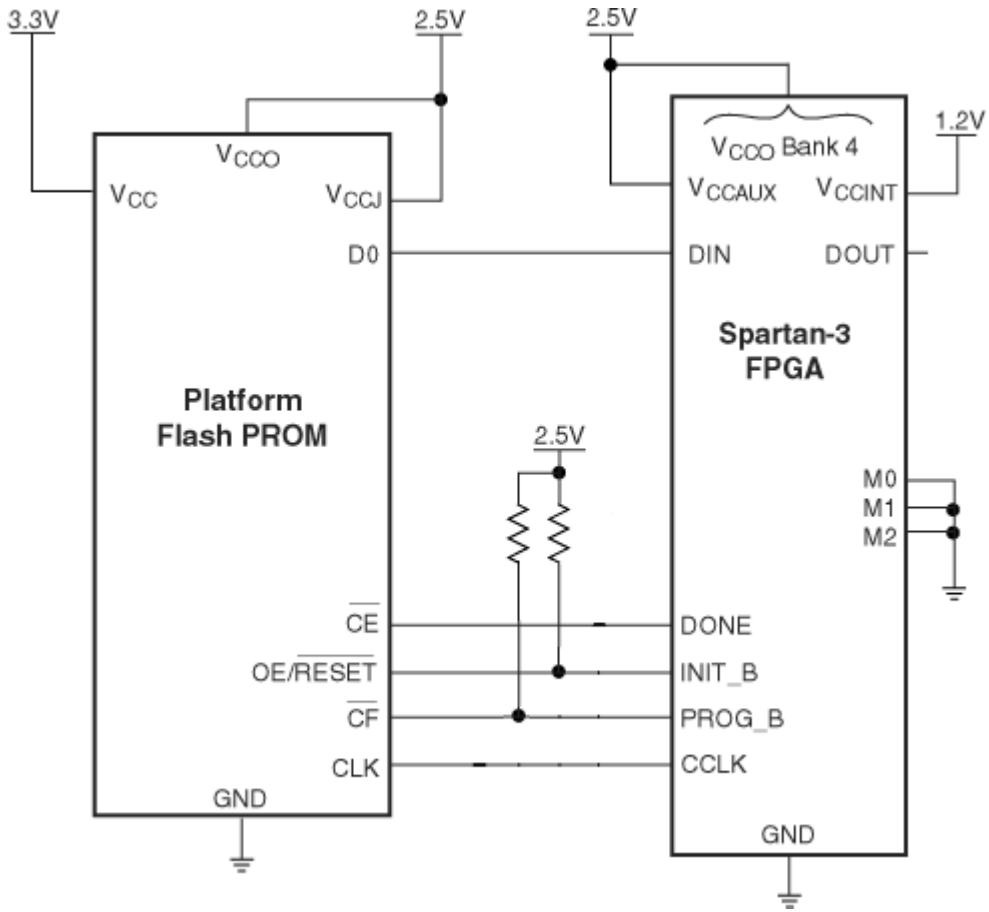
JTAG Mode

- Xilinx FPGAs and PROMs all have JTAG ports.
- IEEE 1149/1532 Test Access Port (JTAG).
  - TDI, test data in.
  - TDO, test data out.
  - TCK, test clock.
  - TMS, test mode select.
- This port is used for many things in addition to programming a device.

Configuration Example

- This example illustrates a circuit like that used on the Spartan-3 Starter Kit board.
  - Boundary scan to program FPGA by cable.
  - Boundary scan to program PROM by cable.
  - Master serial mode for PROM to program FPGA directly.

Configuration Example

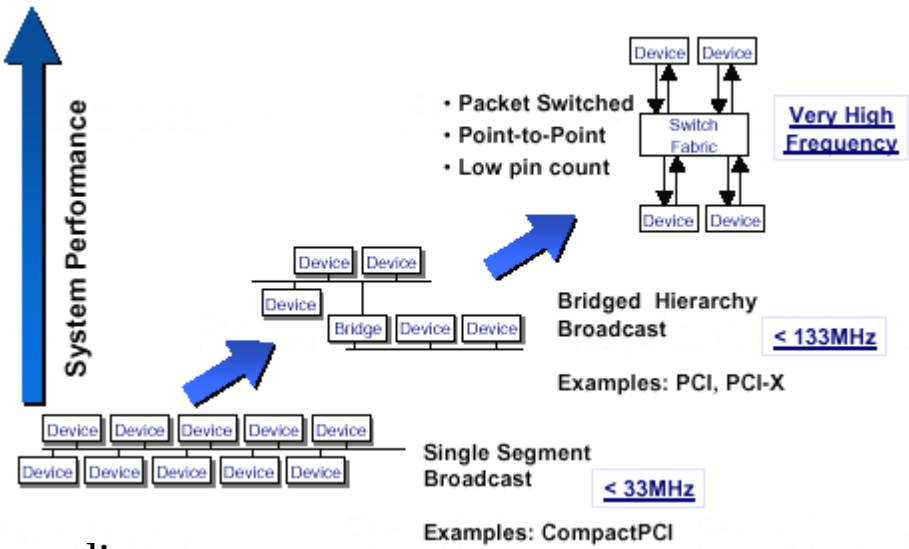


Lecture #12 Agenda

- Survey of gigabit serial transceivers.

Topology Trends

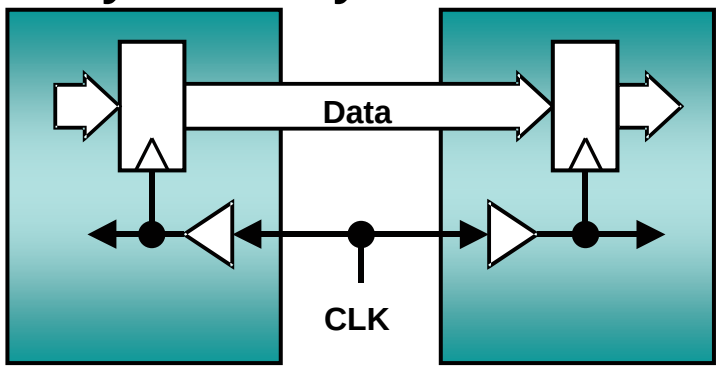
- Full duplex point to point
- Clock recovered from data
- Larger clock skew tolerance
- Typically differential signals
- Low pin counts, high frequency



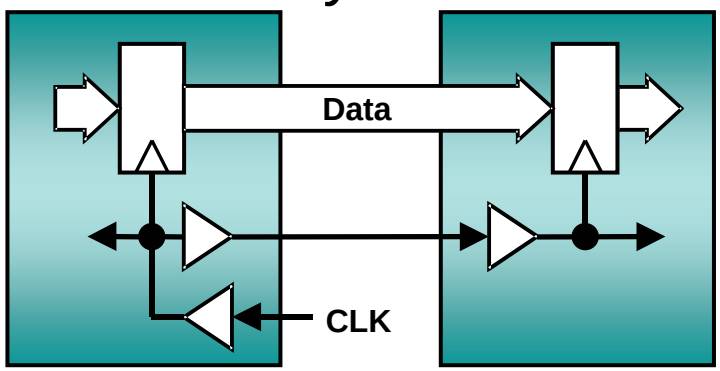
- Wide parallel busses
- System clock or clock forwarding
- Single or double data rate
- Typically single-ended
- High pin counts, low frequency

Traditional Parallel Links

System Synchronous

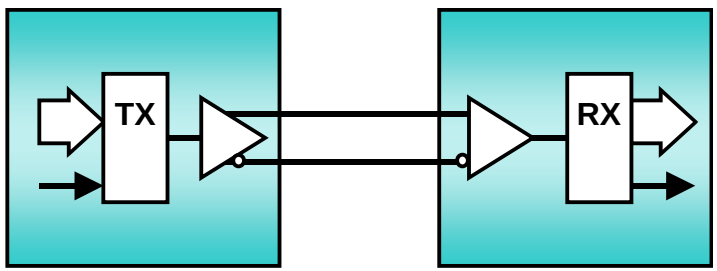


Source Synchronous



Gigabit Serial Link

- Serial transmission with embedded clock.
  - No clock trace.
  - No clock to data skew.
  - Reduced cross talk.



Speed Definitions

- Baud rate, bits per second.
  - Signaling rate at which bits are sent over a link.
  - One direction only!
- Data rate, useful bits per second.
  - Actual rate that data is being sent over a link after removing overhead.
  - One direction only!

Gigabit Serial Transmission

- Encode Clock with Data.
- Serialize Data and Transmit.
- Receive and Recover Clock.
- De-serialize and Decode Data.
- Rate Match.

Encode Clock with Data

- 8B/10B is a very popular encoding method.
- Represents 8-bit data words in 10-bit codes.
- Also has “special” characters for control.
- Encoding overhead 20% (8 bits become 10).

Encode Clock with Data

- Guarantees transition density for clock recovery at the receiving device PLL.
  - Maximum run length is five.
- Provides some error detection capability.
  - Can catch any single bit error.
  - Can catch some two bit errors.
- Running disparity to maintain DC balance.

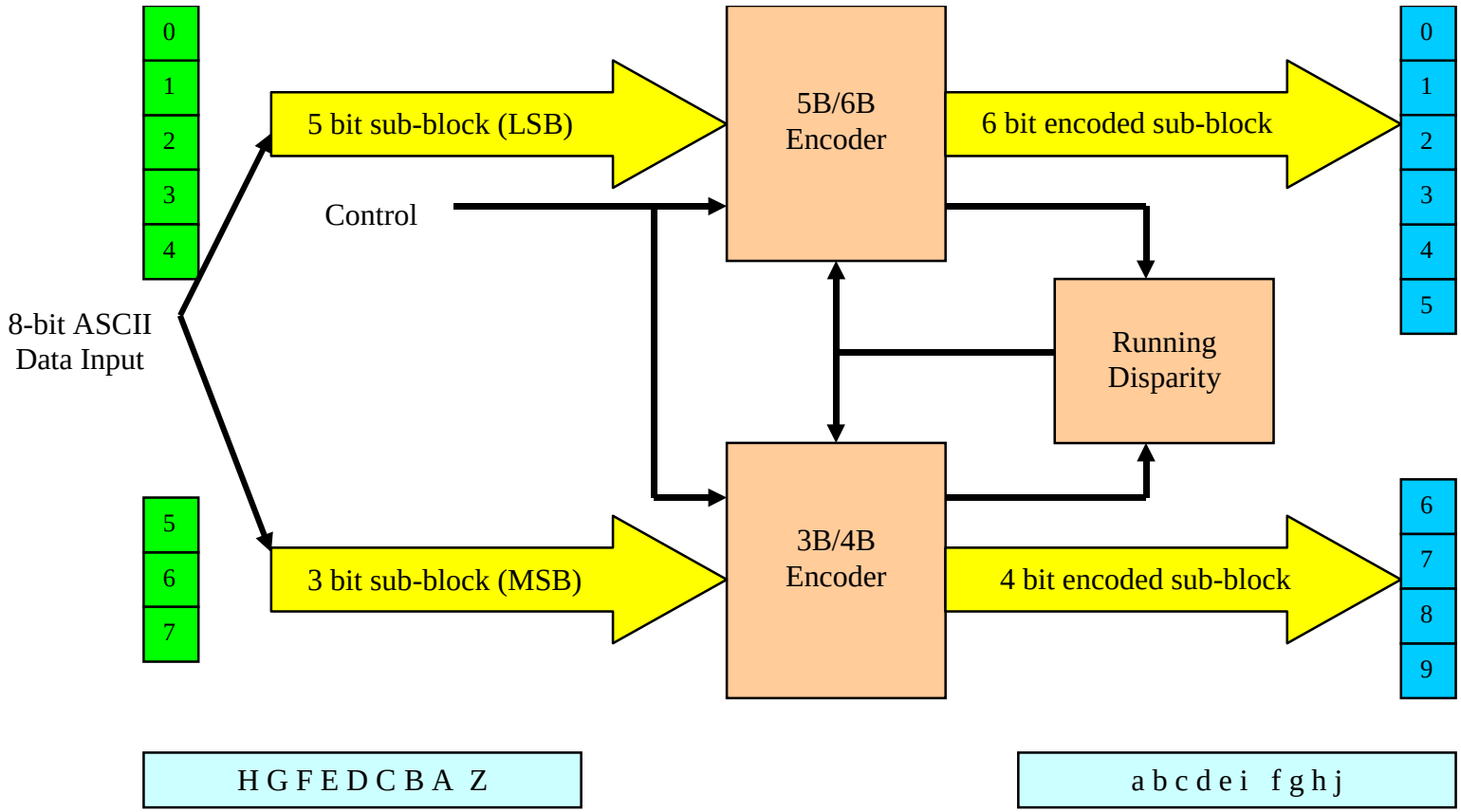
Encode Clock with Data

- Any code with a positive or negative disparity must be followed by one with neutral or the opposite disparity before one with the same disparity is used again.
- Two 10-bit codes exist for each 8-bit byte.
  - Original data byte 0x77 is code “D23.3”
  - D23.3 - => 000101 1100 (send if RD is +)
  - D23.3 + => 111010 0011 (send if RD is -)

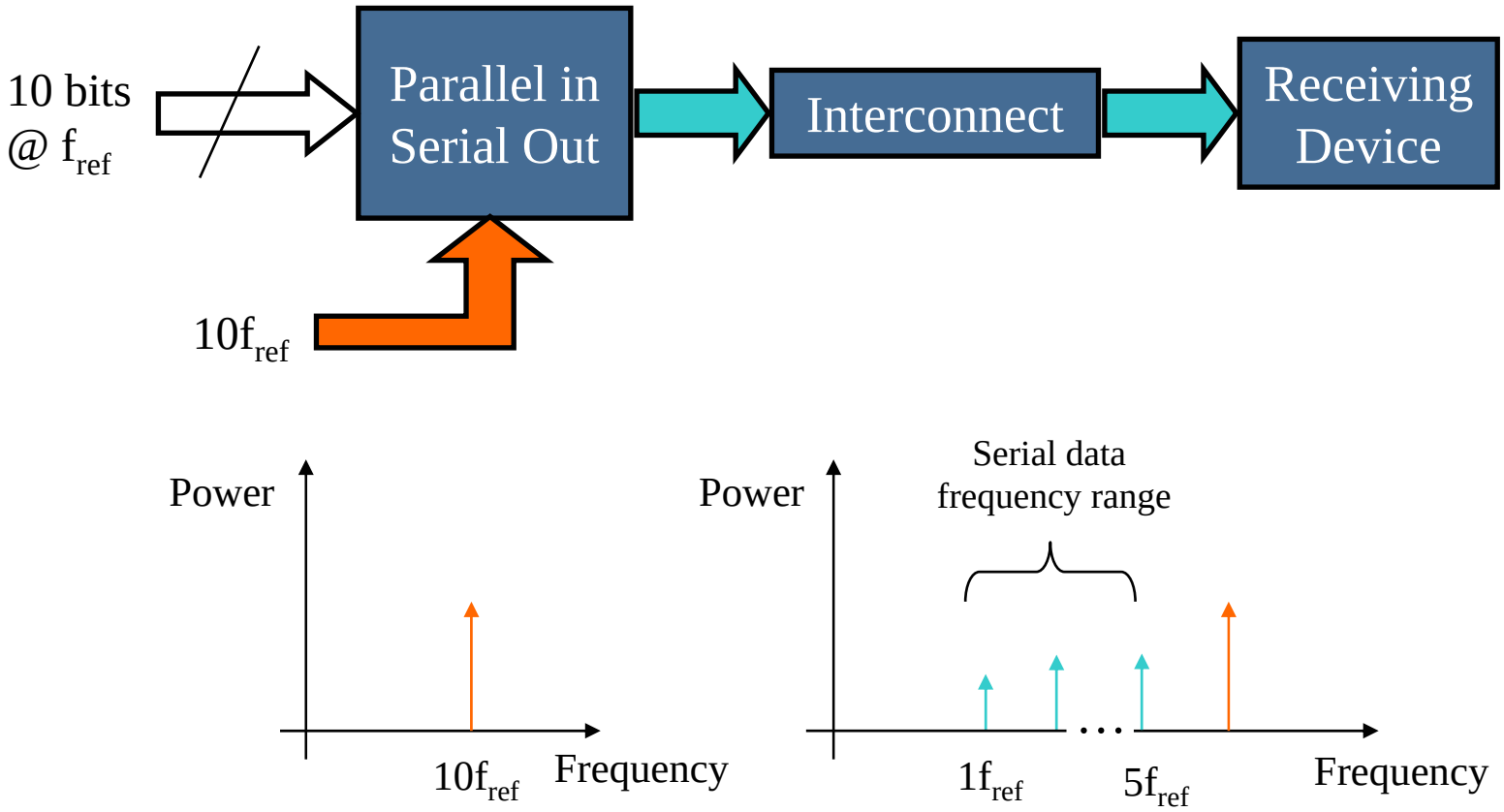
Encode Clock with Data

- Control characters are very important and many have special functions/names.
  - Delimit frames, part of protocol.
    - SOF (start of frame)
    - EOF (end of frame)
  - Serial link maintenance.
    - COM (comma for synchronization)
    - SKP (for data rate matching)

Encode Clock with Data



Serialize and Transmit



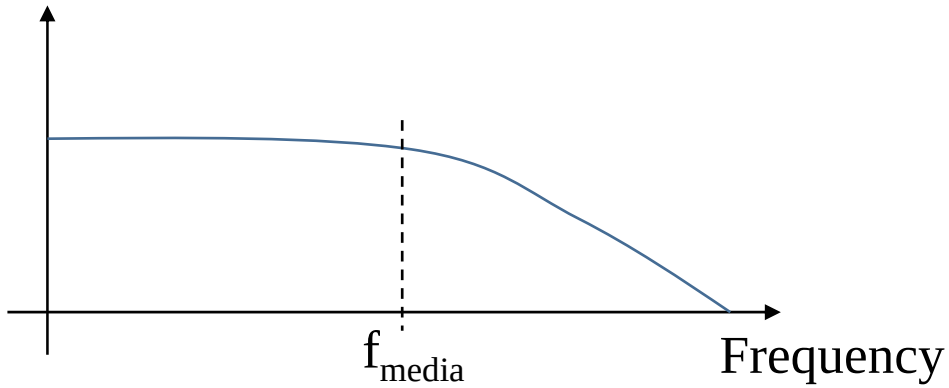
Serialize and Transmit

- Serial data transmitted at $10f_{ref}$
  - Data is 8b/10b encoded.
  - Maximum run-length of five.
- Examples of data and frequency content:
  - 0101010101… → $10f_{ref}/2 = 5f_{ref}$
  - 0011001100… → $10f_{ref}/4 = 2.5f_{ref}$
  - 0000011111… → $10f_{ref}/10 = 1f_{ref}$
  - Frequency range of $1f_{ref} – 5f_{ref}$

Serialize and Transmit

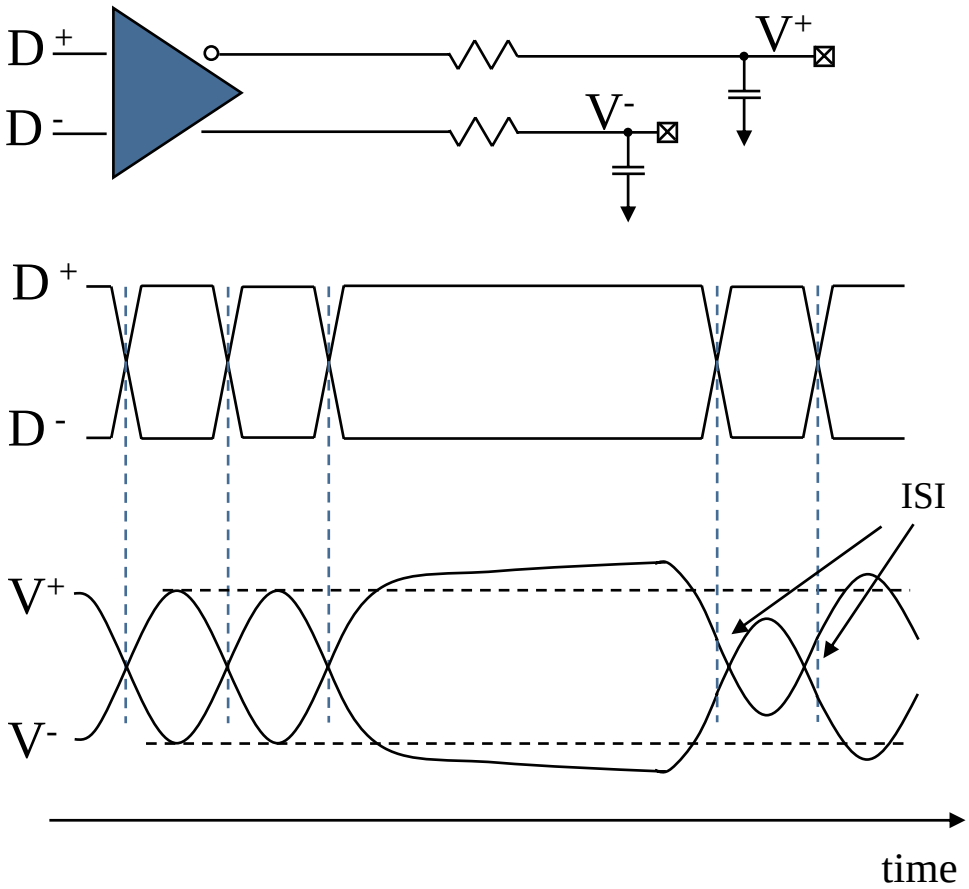
- All interconnect acts like a low pass filter.
  - LPF characteristics change based interconnect.
  - Reduces the performance of the system.
    - Inter-symbol interference, ISI.
    - Signal attenuation.
- It is possible to pre-compensate the output waveforms at the transmitter to reduce the effect of the interconnect.

Serialize and Transmit



- Any data at rates > f_{media} will be attenuated.
- Different frequencies are attenuated differently.
  - 0101010101… → attenuated the most.
  - 0000011111… → attenuated the least.
  - Leads to ISI.

Serialize and Transmit

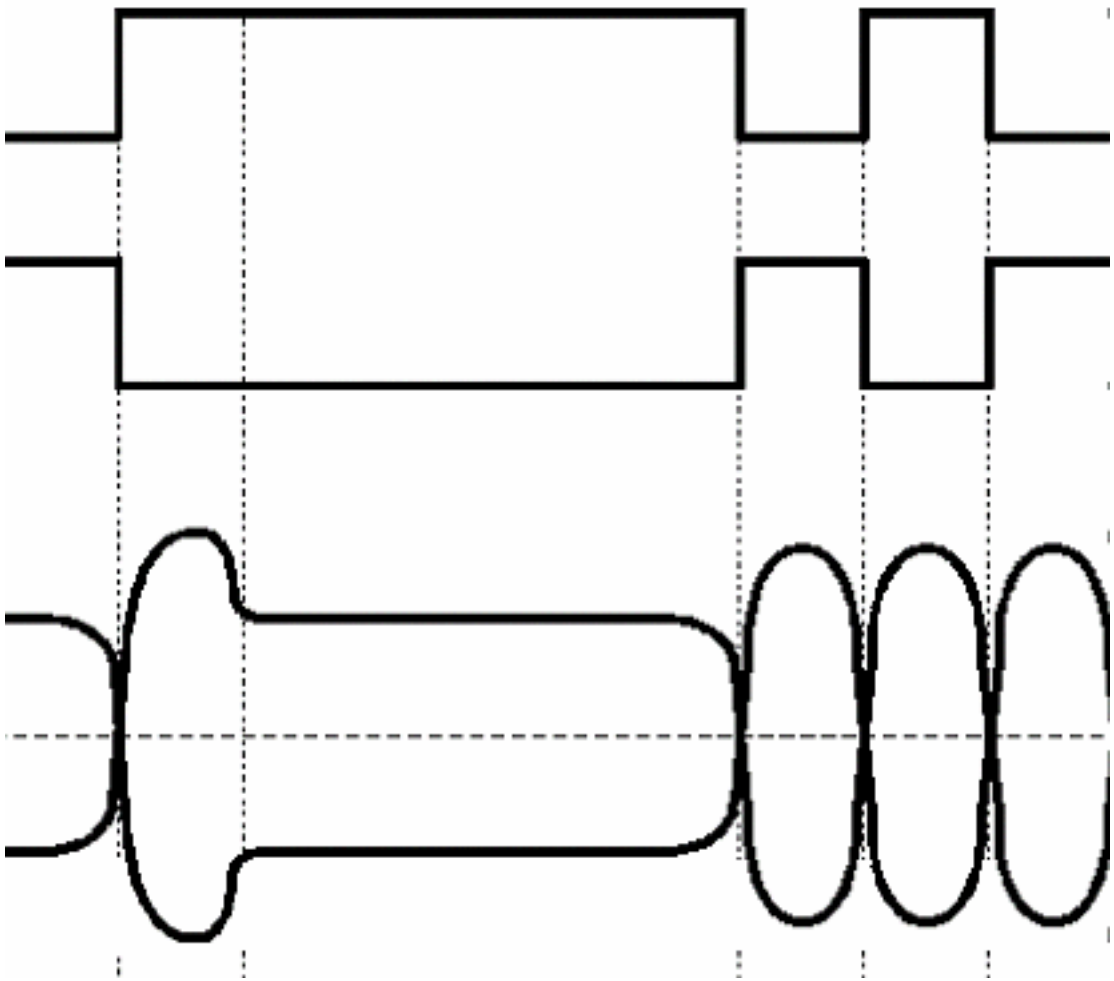


- Low frequency has more time to charge the media.
- High frequency data can not overcome the charge stored on the media.
- Waveform at receiver is less than ideal.

Serialize and Transmit

- Pre-emphasis is media compensation.
- Pre-emphasis boosts certain frequency bands to compensate for media loss.
- Helps signal attenuation and ISI problems caused by attenuation differences.
- Pre-emphasis is an open loop system.

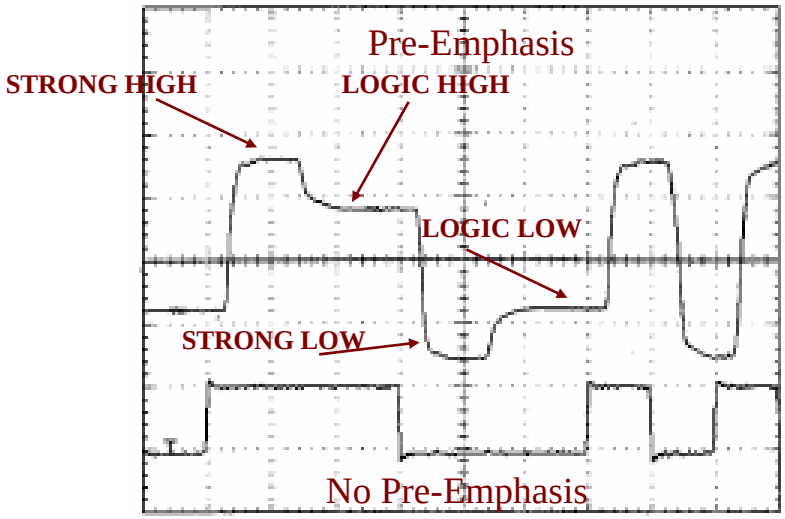
Serialize and Transmit

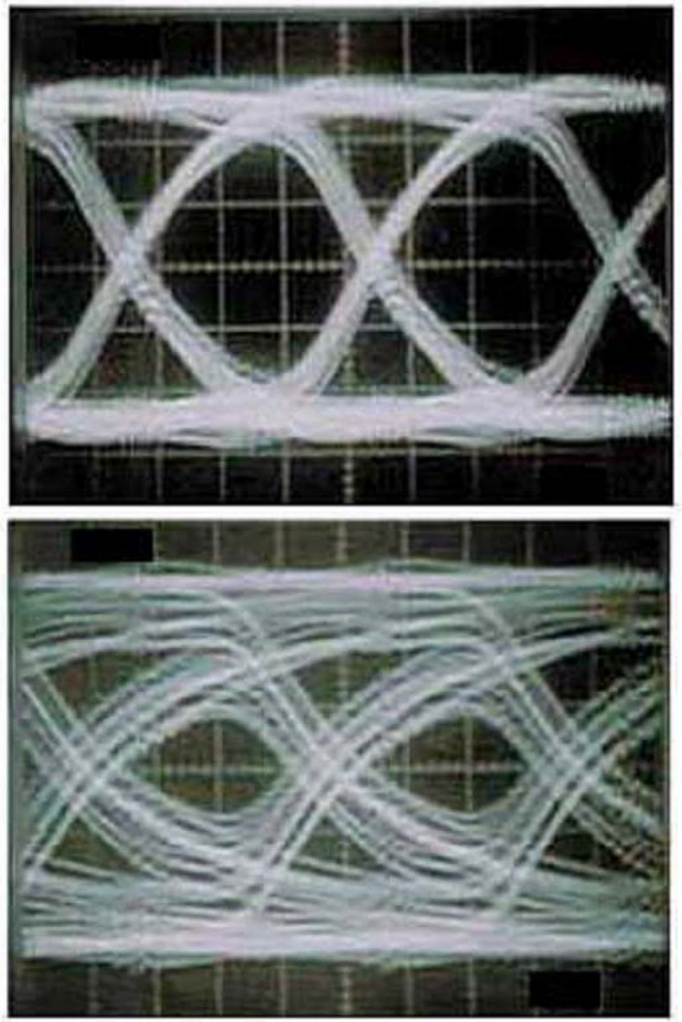


Serialize and Transmit

- No feedback is used to control pre-emphasis.
- How do you determine the right amount?
  - Use a media model and simulate; a quick and easy solution for a first guess.
  - Build a prototype of system; more accurate but more expensive and time-consuming.

Serialize and Transmit





Sample transmitter output and receiver eye diagrams.

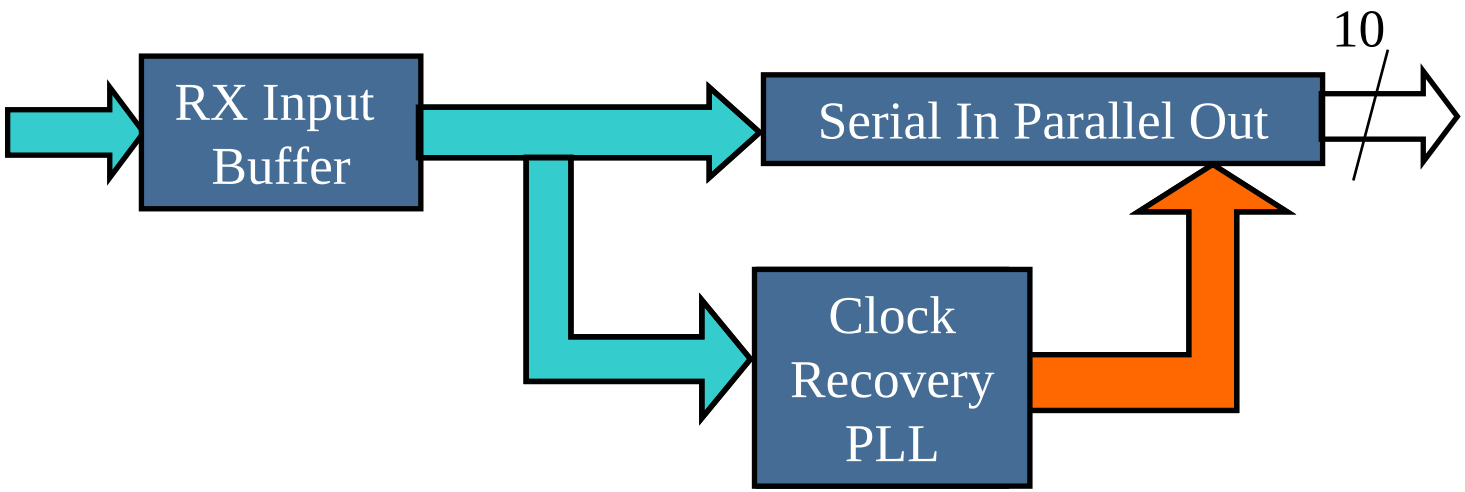
Receive and Recover Clock

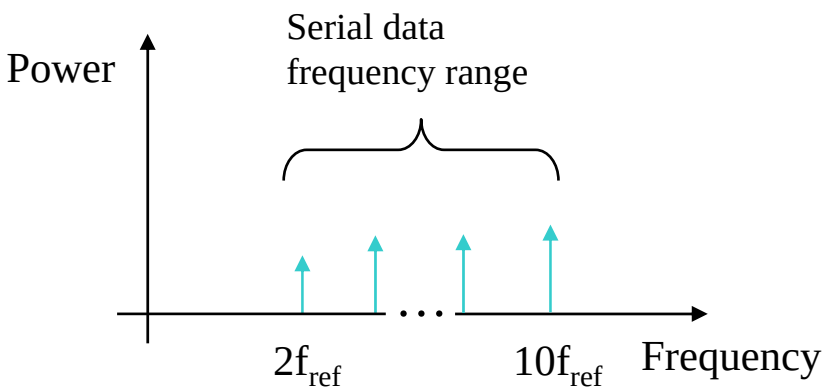
- The transmitter and receiver have different clocks.
  - Slightly different frequencies, e.g. ± 200ppm.
  - This forms a plesiochronous system.
- The transmitter uses its local reference clock to shift out the transmitted data.
- If the receiver were to use its own reference clock to receive the data, data errors would occur due to frequency mismatch.

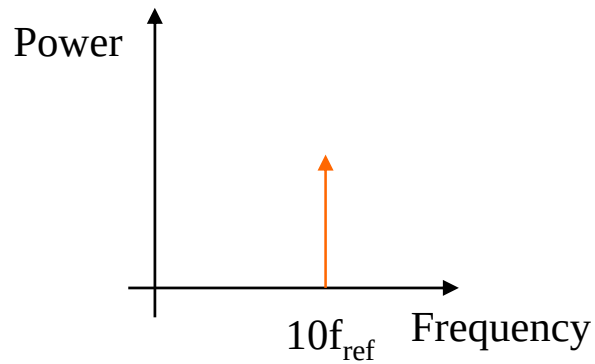
Receive and Recover Clock

- Need a method of matching the frequencies of the transmitter and receiver.
- Clock recovery is the regeneration of a clock at the receiver from the incoming serial data stream.
- Clock recovery units use a phase locked loop.
  - Phase 1: Coarse frequency detection.
  - Phase 2: Exact phase/frequency lock.

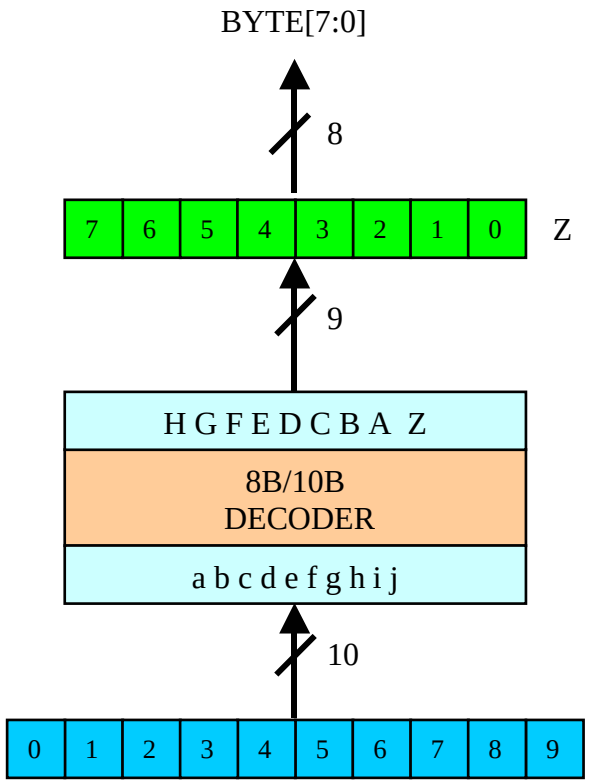
De-Serialize Data







Decode Data


BYTE[7:0]
8
7 6 5 4 3 2 1 0
Z
9
H G F E D C B A Z
8B/10B
DECODER
a b c d e f g h i j
10
0 1 2 3 4 5 6 7 8 9

Rate Match

- In practice, there is a receiver and transmitter on each side of a link.
  - This is called a “transceiver”.
  - Allows full duplex operation.
- Each side of the link has its own reference clock.
  - Used to clock logic for all TX and most RX paths.
  - Used to generate transmit clock of transceiver.
- Recovered data rate is not exactly the same!

Rate Match

- Reference clocks are specified to have a certain tolerance to form a plesiochronous system.
  - Example: clocks with maximum 200 ppm difference.
  - After every 5,000 clock events, one side may have had one more clock event than the other side.
- Eliminates need to distribute reference clock.
- This requires using an elastic buffer with “clock compensation” to synchronize the two systems.

Rate Match

- How does an elastic buffer work?
  - An elastic buffer is basically a FIFO.
  - Received data is written into the FIFO at the recovered clock rate.
  - Received data is read out of the FIFO at the local reference clock rate.
- Over time, the FIFO will eventually under/over run based on the frequency relationship of the two clocks.

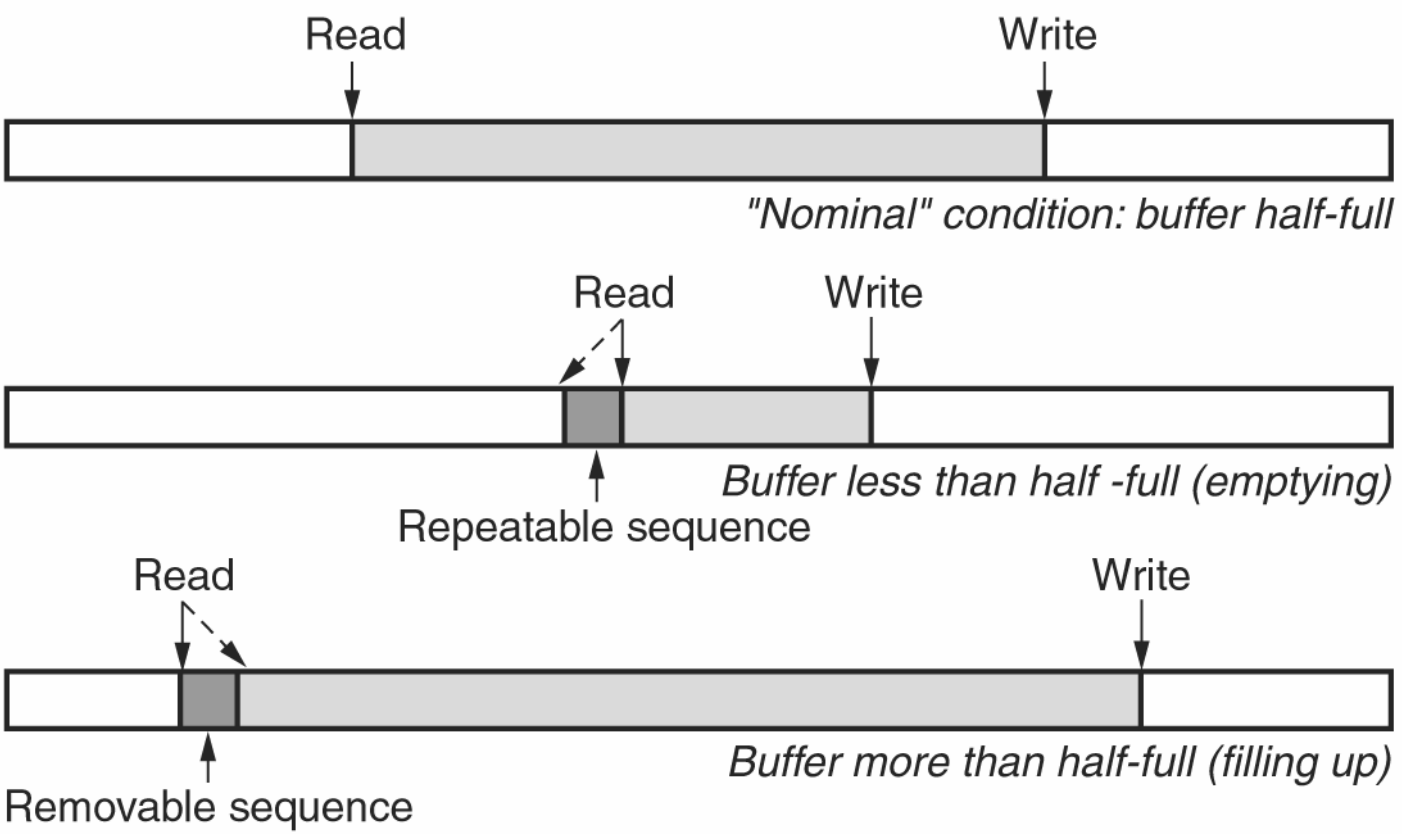
Rate Match

- By agreement, it is the responsibility of the transmitter to periodically insert “clock compensation sequences” into the transmitted data stream.
  - Must be an agreed upon special symbol or sequence of special symbols.
  - Rate of insertion is related to clock tolerances.
  - Link overhead, not meaningful user data.

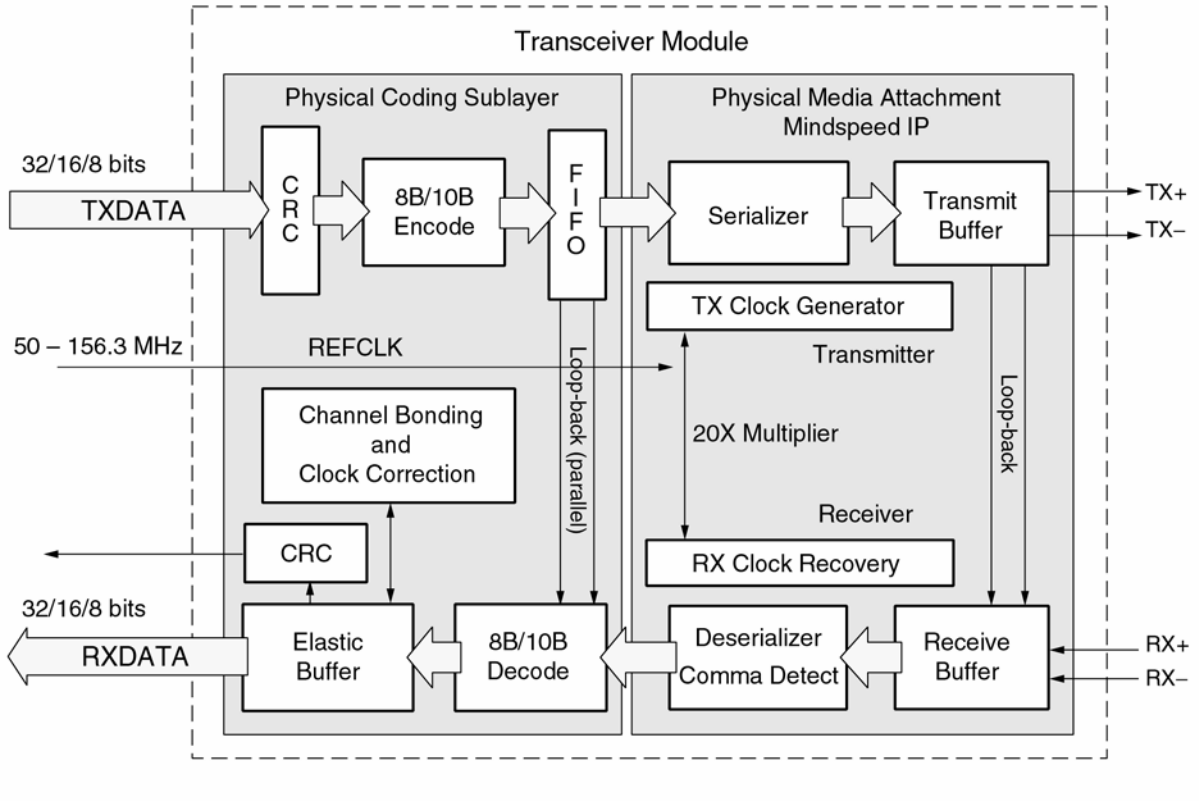
Rate Match

- At the receiver, the elastic buffer can identify clock compensation sequences.
  - If the buffer is getting dangerously full, the elastic buffer logic will delete a CCS to prevent overflow.
  - If the buffer is getting dangerously empty, the elastic buffer logic will duplicate a CCS to prevent underflow.
- User logic ignores CCS because it is not part of the data stream.

Rate Match


Read
Write
"Nominal" condition: buffer half-full
Read
Write
Repeatable sequence
Buffer less than half -full (emptying)
Read
Write
Removable sequence
Buffer more than half-full (filling up)

Gigabit Serial Transceivers



Gigabit Serial Transceivers

- Gigabit serial transceivers are available as off-the-shelf devices from many vendors.
- Xilinx offers some FPGA devices with integrated gigabit serial transceivers.
  - Virtex-II Pro, up to 3.125 Gbps
  - Virtex-II ProX, up to 10 Gbps
  - Virtex-4 FX, up to 11 Gbps
- FPGA integration is convenient!

Gigabit Serial Applications

- Fibre Channel
- 10 Gigabit Ethernet
- PCI Express
- Serial ATA

Fibre Channel

- Developed for Storage/System Area Networks.
- Specifications Controlled by ANSI.
- Three topologies:
  - Point-to-point.
  - Arbitrated loop.
  - Switched fabric.

Fibre Channel

- Baud rate 1.06 Gbps with data rate 848 Mbps.
- Baud rate 2.12 Gbps with data rate 1.7 Gbps.
- Higher baud rates of 4.25 and 10 Gbps are currently in development.

10 Gigabit Ethernet

- Implemented as four “lanes” at 2.5 Gbps each.
- Only optical connections; no copper cable.
- Uses either 8B/10B or 64B/66B data coding and optional data scramblers.

PCI Express

- Replacement for PCI, PCI-X, AGP, etc.
- Currently 2.5 Gbps per lane, with lane widths of 1x, 2x, 4x, 8x, 12x, 16x, and 32x possible.
- Uses 8B/10B data coding and data scramblers.
- PCI software model minimizes S/W impact.
- Supports hot-plugging and aggressive power management features.

Serial ATA

- Serial replacement of standard ATA in PCs.
  - Also related, Serially Attached SCSI...
- Thin cable replaces wide ribbon cable.
- 1.5 Gbps at data rate of 1.2 Gbps.
- Uses 8B/10B data coding.