Purdue University Department of Physics

Physics 53600

Electronics Techniques for Research

Now in PowerPoint!

Spring 2020 Semester

Prof. Matthew Jones

The usual ANNOUNCEMENT

- Obvious changes to the course:
  - No in-person lectures: you'll have to read the lecture notes yourself
  - No more labs: don't worry about it – your grade will be based on work done so far
  - Remaining assignments will try to cover topics that would have been explored in the lab
  - Second mid-term: simplest to cancel it
  - Final exam: I think it will be a 24 hour exam with written responses that can be easily sent by e-mail.
- Changes to grading scheme:
  - Old scheme: Assignments (30%) exams (40%) lab (30%)
  - New scheme: Assignments (50%) exams (25%) lab (25%)

The usual ANNOUNCEMENT

- Because there won't be any in-person lectures, you will have to read the lecture notes yourself.
- To demonstrate that you have read them, you will be required to answer *one or two simple questions* before the next lecture is posted.
- The question will probably be at the beginning and you just have to e-mail me the answer

mjones@physics.purdue.edu

- To make this easy, please make your subject look like this:

"PHYS53600 Lecture xx questions Your Name"

- These will be part of your assignment grade, maybe contributing 10% of your total grade.

More ANNOUNCEMENTS

- Feel free to send me questions about the lecture material if there is anything you don't understand. I'm happy to give more explanation (and I'm soooo bored.)
- Send me e-mail if you think it would be useful to arrange a time as a class to have a time where you can ask questions by video.

LECTURE 23 QUESTIONS

1. *What are four problems that are solved by encoding data (eg, using 8b10b encoding)*

2. *On page 13, it shows how to terminate a transmission line using a resistor divider. Calculate the values of R_1 and R_2 that will work with the power supply voltages listed on that page. How much power does each termination dissipate?*

High Speed Serial Communications

- Communication standards like RS-232, RS-422, and RS-485 typically operate at less than about 100 kbps (kilo-bits per second)
- High speed communications like 1G or 10G Ethernet or PCIe operate at much higher rates over a wide range of distances
- What is the maximum speed that can be easily achieved? What are the limitations?

AC Coupling

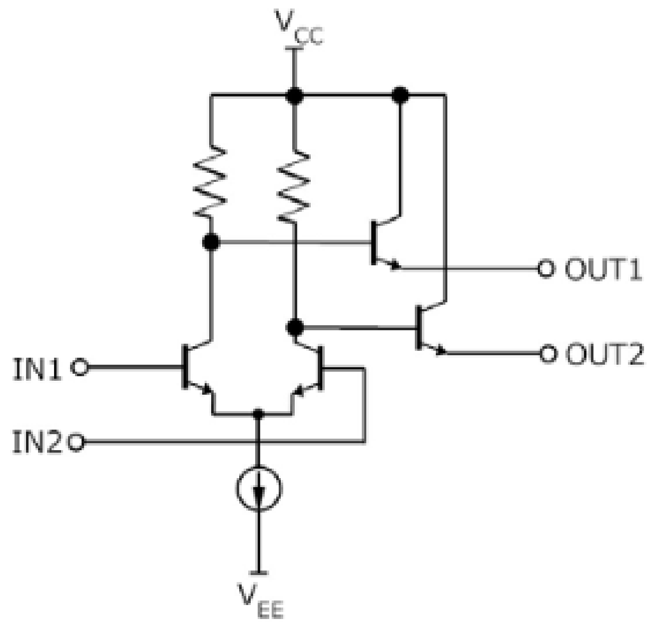
- Differential signaling eliminates the need for a common ground reference
- Differential pairs need to be terminated (somewhere) to prevent reflections
- However, the common-mode voltage on the pair must remain reasonably close to ground if the signals are coupled directly
- The DC current path can be broken by using capacitive coupling

Differential Drivers and Receivers

- To understand the limitations of differential signaling networks it is important to understand the driver and receiver architecture.
- We will start with ECL logic
  - ECL stands for Emitter Coupled Logic
  - The fundamental structure in the logic gates are NPN transistors configured as differential pairs
  - These are analog concepts applied to digital circuits

ECL Driver Architecture

- This is a typical ECL logic circuit (a buffer):



- For ECL logic, V_{CC}=ground and V_{EE}=-5.2 volts
- For PECL logic, V_{CC}=5V and V_{EE}=ground.

ECL Driver Circuits

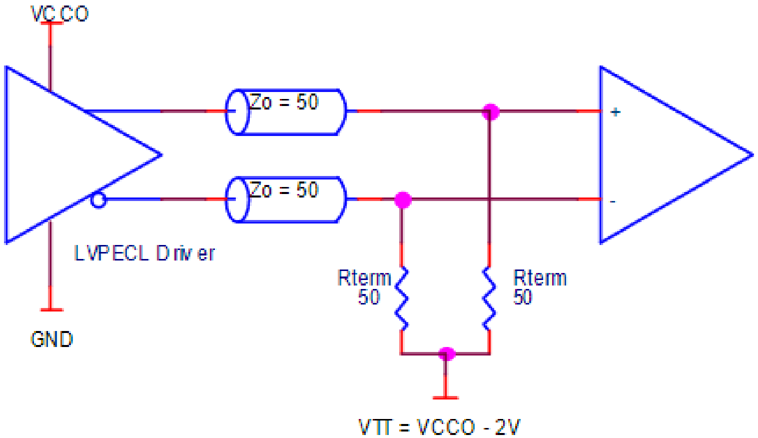
- The outputs of an ECL driver are the emitters of a pair of transistors
- These act as very low impedance current sources and can drive large capacitive loads (like long cables)
- The emitter must be at a lower voltage than the collector (and the base) for the transistor to be in the active region

ECL Receiver Circuits

- Cables must be terminated with their characteristic impedance to prevent reflections
- The termination voltage must be lower enough to put the driver transistors in the active region.
- Typical termination voltage: $V_{TT} = -2\ V$

ECL Receiver Circuits

- You can use a dedicated -2 volt power supply to provide the termination voltage

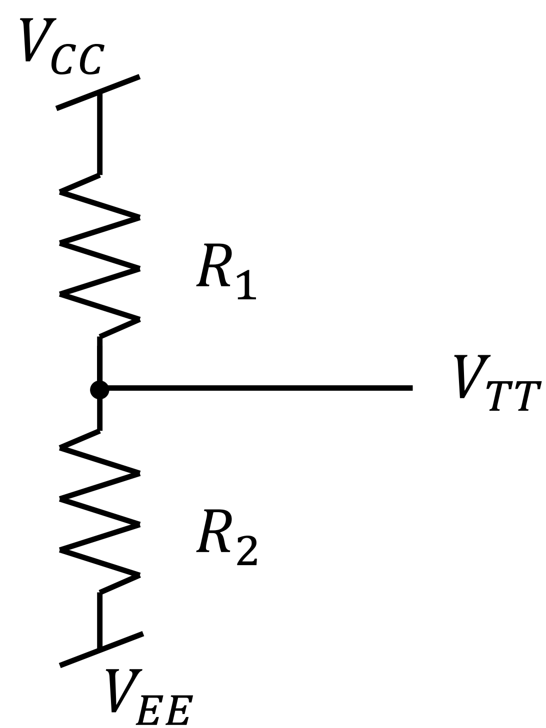


- In this example, each signal looks like a single 50 Ω transmission line
- Equivalent to a coupled pair of signals that act as a 100 Ω transmission line

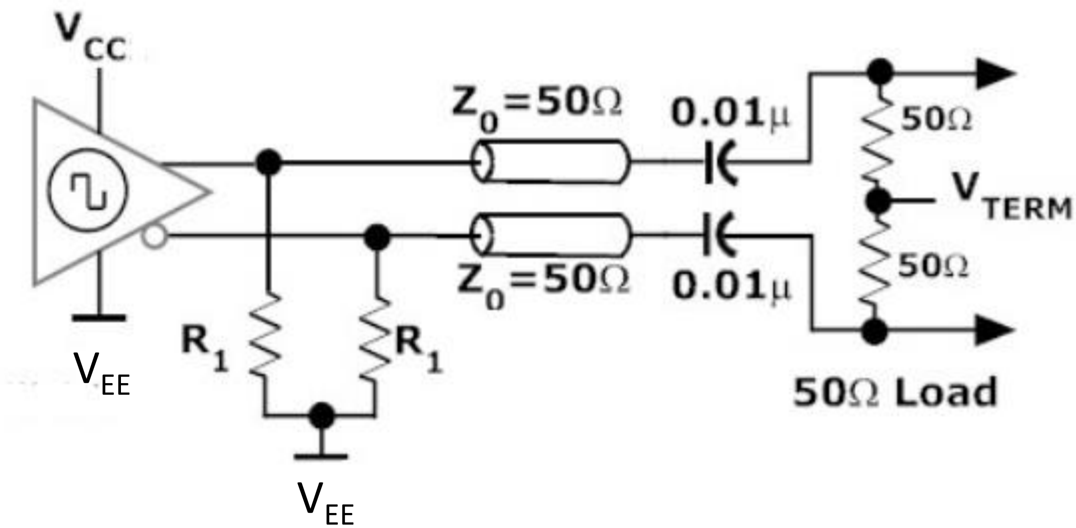
ECL Receiver Circuits

- If you don’t want to provide a dedicated -2 V power supply, you can make one using resistor dividers:
  - V_{CC}=0V, V_{EE}=-5.2V
  - V_{TT}=-2 volts
  - $R_{\|}$=50 Ω

  Now calculate R_1 and R_2…



Capacitive Coupling



The resistors at the driver end are needed to bias the output transistors.

The resistors at the receiving end eliminate reflections.

The capacitors break the DC current path, so now the receiver can be placed VERY far away from the driver.

Capacitive Coupling

- But now we have a problem… how do we transmit a long string of 0’s or 1’s?
- This would look like a constant DC voltage
- The capacitors would block this voltage and the receiver would gradually drift to V_{TT} at both inputs.
- Not likely to work very well…

Data Encoding

- Instead of sending the data directly, we can encode it so that we never send too many 0's or 1's in a row.
- Example: 3b4b encoding

3B Input (Decimal)	3B Input (Binary)	4B Output (Binary)
0	000	0100 or 1011
1	001	1001
2	010	0101
3	011	0011 or 1100
4	100	0010 or 1101
5	101	1010
6	110	0110
7	111	0001 or 1100 or 1000 or 0111

Data Encoding

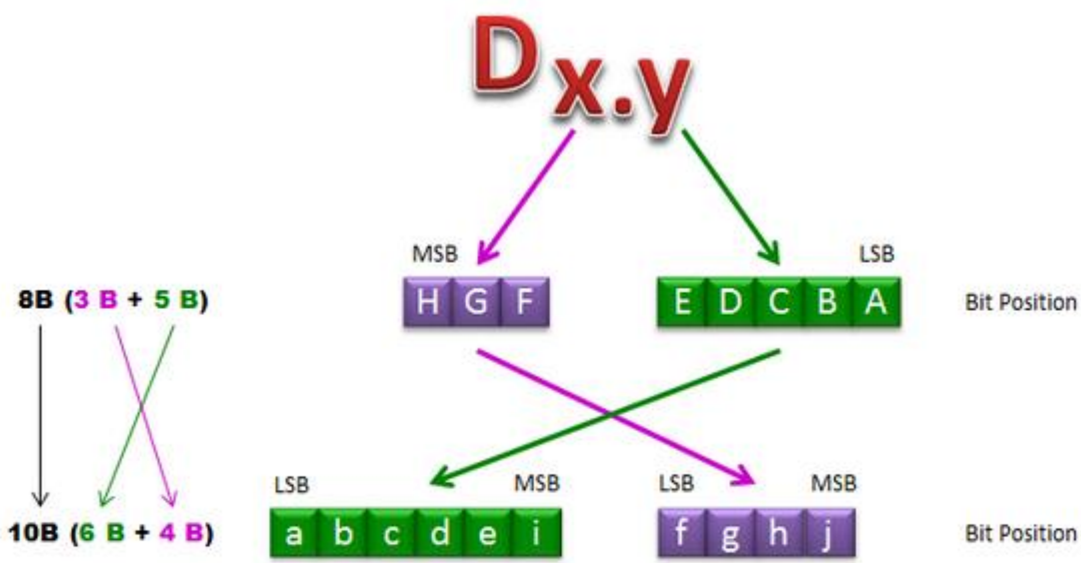
- 5b6b encoding:

This coding scheme uses only 46 out of the total possible 64 output codes.

5B Input (Decimal)	5B Input (Binary)	6B Output (Binary)
0	00000	100111 or 011000
1	00001	011101 or 100010
2	00010	101101 or 010010
3	00011	110001
4	00100	110101 or 001010
5	00101	101001
6	00110	011001
7	00111	111000 or 000111
8	01000	111001 or 000110
9	01001	100101
10	01010	010101
11	01011	110100
12	01100	001101
13	01101	101100
14	01110	011100
15	01111	010111 or 101000
16	10000	011011 or 100100
17	10001	100011
18	10010	010011
19	10011	110010
20	10100	001011
21	10101	101010
22	10110	011010
23	10111	111010 or 000101
24	11000	110011 or 001100
25	11001	100110
26	11010	010110
27	11011	110110 or 001001
28	11100	001110
29	11101	101110 or 010001
30	11110	011110 or 100001
31	11111	101011 or 010100

Data Encoding

- 8b10b encoding is useful when data is organized into 8-bit bytes.
  - It uses a combination of 3b4b and 5b6b encoding



Running Disparity

- The transmitter can also count the number of 0’s and 1’s transmitted and dynamically choose the output code so as to make them equal.
- If there are more 0’s than 1’s, then pick a code from the RD- table.
- If there are more 1’s than 0’s then pick a code from the RD+ table.
- This ensures that on average, there are an equal number of 0’s and 1’s so the decoupling capacitors will not drift to a net DC offset.

8b10b Encoding

- There are some left over codes that can aren't needed to encode the data
- They are very useful to transmit "non-data" control characters

Control symbols

Input				RD = −1	RD = +1
	DEC	HEX	HGF EDCBA	abcdei fghj	abcdei fghj
K.28.0	28	1C	000 11100	001111 0100	110000 1011
K.28.1 †	60	3C	001 11100	001111 1001	110000 0110
K.28.2	92	5C	010 11100	001111 0101	110000 1010
K.28.3	124	7C	011 11100	001111 0011	110000 1100
K.28.4	156	9C	100 11100	001111 0010	110000 1101
K.28.5 †	188	BC	101 11100	001111 1010	110000 0101
K.28.6	220	DC	110 11100	001111 0110	110000 1001
K.28.7 ‡	252	FC	111 11100	001111 1000	110000 0111
K.23.7	247	F7	111 10111	111010 1000	000101 0111
K.27.7	251	FB	111 11011	110110 1000	001001 0111
K.29.7	253	FD	111 11101	101110 1000	010001 0111
K.30.7	254	FE	111 11110	011110 1000	100001 0111

8b10b Control Characters

- When a link is idle (no data being transmitted) it still needs to send 0’s and 1’s to maintain the DC balance
- By convention, the transmitter can send a bunch of “idle” codes (K28.5)
- K28.5 is special because its pattern of bits cannot be the result of any other combination of codes
- This allows the word boundaries to be found in the serial data stream

Other Coding Schemes

- 64b66b encoding:
  - This doesn’t use a table of codes (it would be too large)
  - Instead, it calculates the codes dynamically using a simple binary polynomial
  - This effectively scrambles all the bits and makes them look like a random pattern of 0’s and 1’s
  - But they aren’t random and they can be unscrambled using a similar binary polynomial

Summary of Data Encoding

1. Eliminates long runs of 0’s and 1’s
2. Provides the ability to find word boundaries within the serial data stream
3. Error detection:
   - Several possible bit combinations don’t occur in the coding tables
   - If you receive such a combination then you know it must be an error in data transmission

Clock Recovery

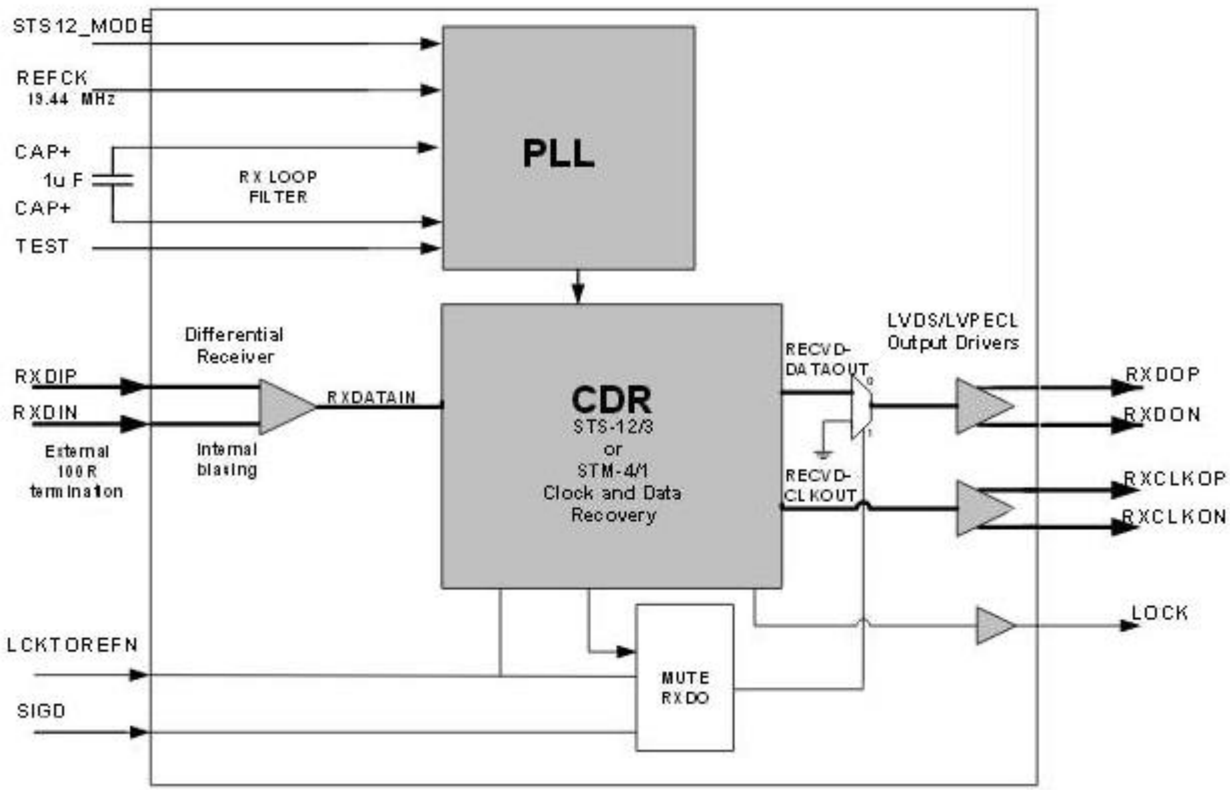
- How does the receiver know when to sample the data?
  - In the UART design, there was a high frequency master clock (1.8432 MHz) that got aligned using the START condition
- If we know that the data stream contains lots of 0→1 and 1→0 transitions, then we can dynamically adjust the frequency and phase of a receiving clock to match the bit boundaries

Phase Locked Loops

- The basic idea of a phase-locked loop is an oscillator in which the phase of a reference oscillator dynamically adjusted in a feedback loop
  - If the received transitions consistently arrive earlier than the reference clock edges, then decrease the phase
  - If the received transitions consistently arrive later than the reference clock edges, then increase the phase
- This means that the frequencies of the clocks used by the transmitter and receiver don’t have to be perfectly matched (but they have to close).

Clock/Data Recovery

- A typical high-speed serial data receiver contains a clock/data recovery circuit:



Clock/Data Recovery

- This is another benefit of data encoding:
  - It is beneficial to have lots of 0→1 and 1→0 transitions so that the clock phase can be precisely determined
- This whole scheme works extremely well provided drifts in the transmitter's clock phase/frequency are slow compared with the bit rate

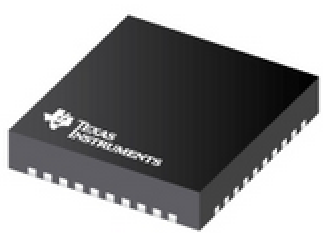
Implementation

- All of these features are usually provided by dedicated integrated circuit components.
- For example:

TLK1221 ACTIVE

In English | Alert me

Gigabit Ethernet Serdes in small 40-pin package

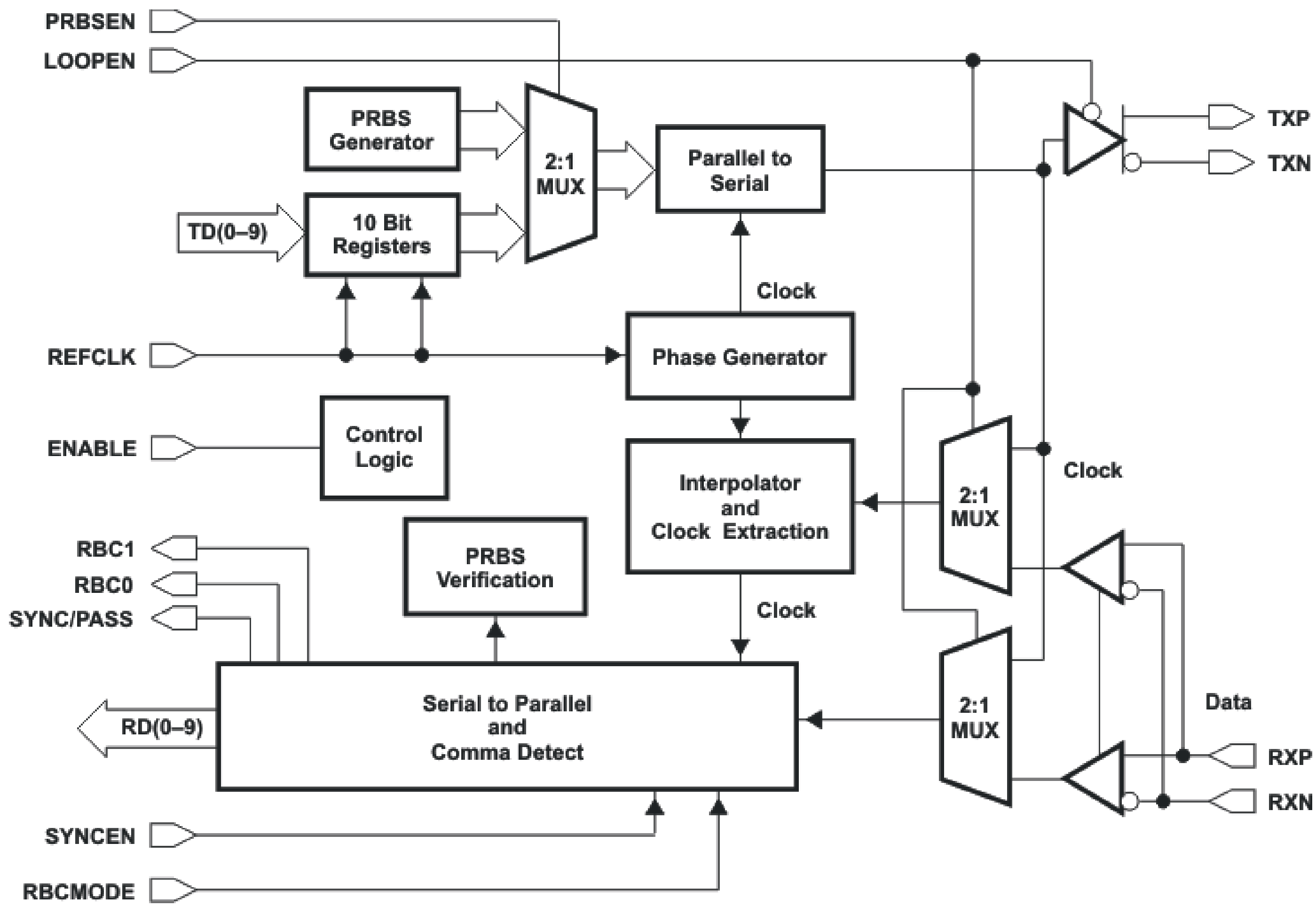


DATASHEET

Ethernet Transceivers datasheet (Rev. C)

Download

Example Implementation



Summary

- So those are more or less the principles behind high speed serial data transmission.
- They are the same principles used for several standards:
  - Ethernet (over cables or fiber optics)
  - XAUI (10 Gbps network over 4 separate serial links)
  - USB (over cables)
  - PCIe (over PCB backplanes)
  - JESD204B (predictable data latency necessary for high-speed data acquisition systems)