fidus
innovate•design•deliver

Best Practices for High-Speed Transceivers in Today's FPGA Designs

An overview of common transceiver design and cost challenges

A FIDUS SYSTEMS **WHITE PAPER**

Today's expanding data speed and volume requirements place a renewed emphasis on FPGAs and effective high-speed transceiver usage. Massive increases in data volume, driven by consumer internet and wireless demands, have led to faster communications protocols and the need for devices that support them. Ethernet speeds have increased from 10 Gbps to 400 Gbps in just a few years, while **PCIe speeds** have gone from 2.5 GT/s per lane in Gen1 up to 32 GT/s per lane with the recent Gen5 specification.

To satisfy these needs, FPGA designers must understand current design practices for high-speed transceivers (also known as multi-gigabit transceivers) to move large amounts of data while meeting requirements for power, cost, and time to market.

This paper explains key considerations for high-speed transceiver designs and recommendations for overcoming the most common development challenges. By reading the material presented here, FPGA designers will have a better understanding of how to approach and execute their next high-speed transceiver projects.

With over 20 years of FPGA design experience and over 400 customers in various markets, **designers at Fidus** have been exposed to a wide variety of applications across many different use cases. This paper shares some of our experiences in the methods required to achieve high speeds and power efficiency that won't burn a hole in the board or in the project's budget.

OVERVIEW OF HIGH-SPEED TRANSCEIVERS

At their most basic level, transceivers connect the internals of an FPGA to components outside the device using high-speed differential signals with embedded clocks to transfer data. This includes connections to HDMI and Ethernet ports, digital to analog converters (DAC) or analog to digital converters (ADCs), host PCs via PCIe and other FPGAs. A key consideration for transceivers is their ability to conform to industry standard interface protocols including attributes such as data rate, signal integrity and encoding.

For FPGA and Hardware designers, there's often a trade-off in implementing these standards. Trade-offs can include impacts on costs, development time, and complexity of the design.

Using external transceivers

Discrete, or external, transceivers are single function components that designers sometimes use to satisfy FPGA data transfer requirements. The benefits of external transceivers include the potential use of a lower-cost FPGA and a reduced effort in FPGA design if the device implements more of the protocol internally.

The downsides to using an external transceiver, however, include less flexibility in the overall design (assuming a suitable external device exists at all), increased device count on the PCB, additional manufacturing and test steps, as well as potentially higher BOM costs.

TOP CONSIDERATIONS FOR HIGH-SPEED TRANSCEIVER DESIGN

A high-speed transceiver is comprised of two channels, a transmitter and receiver, that connect between the FPGA fabric and external signals. The channels are usually supported by shared logic that supports requirements such as clock generation. The channels are composed of two layers that perform different functions:

1. Physical coding sublayer (PCS) to implement the digital logic required of the application protocol (e.g., Ethernet, PCIe, 8B10B, 64B66B).
2. Physical media attachment sublayer (PMA) to implement the analog circuits responsible for the actual transceiver functions (i.e., parallel to serial conversion, output drives and signal equalization).

When the transceiver operates, internal parallel data is sent through the transmitter PCS and PMA and converted into serial data to be sent out. Incoming serial data is processed by the receiver PMA and PCS for processing by internal FPGA components.

When selecting a transceiver, designers should consider how resources are organized, specified, and shared. The desired interface protocol and the FPGA itself determine the characteristics to consider, as shown in Table 1.

DRIVEN BY PROTOCOL	DRIVEN BY THE FPGA
Data rates	Hard blocks
Line coding (8B10B, 64B66B, raw)	Memory
Required lane count	Reference clocks
Protocol licensing	Available lane counts
Development effort	Power supplies
	Power consumption
	Temperature range
	Price

Table 1: Typical characteristics of a high-speed transceiver

The next sections discuss what FPGA designers should know to consider about these characteristics.

PCB considerations

With high-speed transceivers comes the need for high-speed signals, requiring careful attention to signal integrity and component layout.

Very high-speed requirements, such as 5 Gbps or more, may drive the need for expensive PCB transmission mediums to minimize perturbations to the signal's frequency content. High-frequency attenuation distorts signals, causing reduced signal amplitude at the receiver and increased interference between subsequent symbols. This requires trade-offs between the costs of circuit board materials to preserve signals and the potential for introducing errors.

Ensuring the correct layout can be a considerable challenge as high-speed components are especially sensitive to relative placement and operation.

Some of the more common challenges include:

1. **Working with adjacent transceivers requires the right clock signals going into the right transceiver banks** – this is where performing a trial FPGA compilation of all components before PCB layout can save costly PCB re-spins and debugging time.

2. **Ensuring the correct order of lanes into the device** – some IP and FPGAs have specific requirements about ascending or descending orders.

3. **Orienting the transceiver block correctly** such that the order of lines coming out aligns with the order on the other side, otherwise the designer may need to allocate extra board space to reorient the lines.

4. **Determining the number of lanes required to meet bandwidth and minimize costs** – 4-lane PCIe Gen 2 interfaces have the same theoretical bandwidth as one PCIe Gen 4 lane, but the PCIe Gen 4 implementation usually requires a more expensive FPGA and may require more expensive PCB material.

Clocking

As FPGA transceivers are timing sensitive, they require a dedicated reference clock with stringent specifications for noise and jitter performance. Controlling, reducing, and maintaining low jitter in the clock's signal path is critical, so FPGA designers should understand the required performance specifications of their clock sources. This includes phase noise and jitter and, if available, the protocol frequency tolerances for the different ends of the link.

FIDUS RECOMMENDS...
FPGA designers should understand the required performance specifications of their clock sources.

Protocols and data rates

With many options for protocols and speeds, FPGA designers must understand the requirements of the host and the associated implications on development. For example, if the host only supports PCIe Gen 3, then it likely doesn't make sense to incur the cost of implementing a transceiver that supports PCIe Gen 5 (future proofing considerations notwithstanding).

At the physical connectivity level, connection medium can also be dictated by the adopted protocol. With high-speed Ethernet transceivers, for example, there's usually implications between interfacing with an optical fiber connection versus copper Twinax. While a Twinax connection is usually cheaper and lower power than optical, development of a Twinax-capable transceiver may be more expensive as the required features for auto negotiation and forward error correction aren't always included in a base Ethernet protocol IP license.

Data width considerations

The most basic function of a high-speed transceiver is converting serial data to or from parallel data or an operation as a serializer/de-serializer (SerDes). In the basic conception, the parallel data width is sized to match a protocol dependent symbol width, such as 10-bits for 8B10B or 66 bits for 64B66B. Matching the parallel data width to the symbol width is conceptually simple and avoids any issues with inter-symbol alignment. However, this approach can lead to excessively fast clocks speeds or inefficiently wide data-paths depending on the line code and data rate.

Modern FPGA transceiver blocks include a "gearbox" function which allows the designer to manage these concerns, either widening a fast data rate signal to multiple symbols in parallel or narrowing a wide symbol and transmitting it over multiple clocks. This moderates transceiver interface clock speed to match what is comfortable for the FPGA fabric of specific

device. For example, sending one symbol per clock over a 5 Gbps PCIe interface would run at 500 MHz – much faster than most fabrics can handle. Using two symbols in parallel through with a gearbox would bring that rate down to a more manageable 250 MHz.

OVERCOMING TYPICAL DEVELOPMENT PROBLEMS

FPGA design and development practices have been evolving since the 1980s. While there is a lot of industry knowledge around what works and what doesn't work, it's worthwhile to understand what newer protocols and faster speeds demand in order to produce consistent, predictable results.

The next sections detail six best practices around the most challenging aspects of high-speed transceiver design.

1. Choosing the right FPGA family

Selecting the right FPGA vendor and family requires a thorough understanding of the project's requirements, however they may not always be fully specified. When researching vendors, the typical considerations to be aware of include:

- Number of transceivers and the speeds they run at (including different speed grades).
- Power consumption (especially important in space-constrained layouts where heat may be a concern).
- Protocols supported by hard IP (built into the device, highly efficient) and soft IP (created from the logic fabric of the FPGA, typically less efficient).
- Equalization options that affect reach (distance) and power consumption.
- Self-test features to allow checking of the FPGA and PCB without the need for external test equipment.

If possible, it's helpful to evaluate the performance of a potential transceiver before making the final decision. While they may look good on paper, it's critical to ensure that the actual device complies with specifications and that it operates under real world conditions without adverse effects.

FIDUS RECOMMENDS...
It's helpful to evaluate the performance of a potential transceiver before making the final decision.

2. Buying protocol IP licenses vs. build your own

In addition to the generic high-speed transceiver blocks which provide the physical layer, implementing a standard protocol requires protocol specific logic to implement the link and higher layers of the protocol. Primarily affecting cost and development time, a designer has the choice to license the IP for the required transceiver protocol or build it from scratch. Table 2 lists the trade-offs between each option.

There's also the choice of licensing IP from the FPGA transceiver vendor or a third-party company. It's always good to research and consider who can best support both the protocol and the device its run on. Typically, a third-party IP company has more protocol expertise, but they may lack experience in designing and verifying that protocol on the chosen device family.

FIDUS RECOMMENDS...
It's always good to research and consider who can best support both the protocol and the device its run on.

	LICENSED IP	BUILD YOUR OWN
Evaluation effort	Most vendors offer royalty-free evaluation kits, documentation, and samples	N/A
Development effort	Less effort	More effort; no documentation or samples to work from
Flexibility	Most licensed IP is encrypted, disallowing modifications by the user and limiting internal visibility for debugging	Total flexibility in modifications and debugging visibility
License model	Some licenses are pay-per-use and/or distinguish between development and production use	Self-owned IP
Release impact	More predictable impact on development time	Less predictable impact on development time

Table 2: Buy vs. build trade-offs for transceiver protocol IP

3. Implementing higher-level protocols

Many protocols require supporting software to run effectively, providing runtime control of the link operation. For example, the sources of consumer video protocols like HDMI and DisplayPort need to determine the characteristics of connected devices (display types, supported resolutions, etc.) and may need to negotiate link speeds and configurations. These requirements must be understood and planned for early enough in the design process to avoid expensive integration problems later.

FIDUS RECOMMENDS...
Requirements must be understood and planned for early enough in the design process...

4.Running loopback tests

If possible, FPGA designers should run loopback tests on their boards to ensure transceivers are receiving proper power, clocks are operating correctly, and signal integrity is within expected tolerances. While loopback boards may incur extra costs, they can easily pay back the investment by allowing a simple bit rate error test on the transceiver to ensure the hardware isn't faulty before proceeding with significant development, and more importantly, a lengthy debug effort.

FIDUS RECOMMENDS...
FPGA designers should run loopback tests on their boards...

5. Performing a trial compile

Trying to **understand all the FPGA design rules** is a nearly impossible task, as information is usually spread across several data sheets and user manuals between devices and protocol IP – often with no clarity on the interactions between different vendors. To avoid frustration and respin costs during integration with the PCB, it's best to perform early trial compilations of the FPGA project with the transceivers and all other IO included to ensure no warnings or errors.

FIDUS RECOMMENDS...
It's best to perform early trial compilations...

This provides confidence in the transceiver's operation and ensures the FPGA rules and constraints are met within the context of all connected components before sending the final PCB Gerber files out.

6. Running simulations

As transceiver simulations can take a long time to run and are often difficult to include in the project, many FPGA designers exclude them and simulate to the level of the protocol IP. When deciding whether to simulate transceivers or not, it's important to:

1. Understand the time the simulation will take and its impacts on tying up simulation licenses. Some FPGA and IP vendors offer a "speed up" option on their IP to minimize this cost.
2. Know that transceivers may take significant time and effort to include in the simulation project.
3. Know that part of this complexity comes from ensuring the protocol IP matches at either end of the link – it's sometimes difficult or costly to get both sides speaking the same language. This is particularly important for unidirectional protocols, where the IP for one direction is different (and may be licensed separately) from the IP for the other end of the link.

An option to reduce effort is to perform a minimal simulation that focuses on verifying basic transceiver and protocol operation with a full transceiver model, then stubbing out or bypassing the transceivers for more extensive simulations to verify the rest of your design.

FIDUS RECOMMENDS...
An option to reduce effort is to perform a minimal simulation...

CONCLUSION

To effectively design and implement high-speed transceivers requires an understanding of how to satisfy the requirements of their endpoints as well as their internal constraints. Designers should start with the rules and specifications provided by the vendors (both hardware and protocol IP) and consider their impacts on development efficiency and costs.

Fidus FPGA designers have the expertise and skills to **transform your project requirements into robust and cost-effective components**. With decades of experience, our team understands how to design and deliver successfully, eliminating the need for your team to research, experiment, rework FPGA designs, and, worse, incur costly PCB re-spins that delay product launches.

Interested in accelerating your next FPGA design project? **Contact Fidus today**.

20+
years experience
Collaborating with smart teams is what fuels us every day.

3,000+
successful projects
Your unique challenges are our obsession.

400+
customers
Extending your team with our expertise brings designs to market faster.

82%
repeat customers
Customers love to work with us, again and again.

ABOUT FIDUS

Fidus Systems, founded in 2001, specializes in leading-edge electronic product development with offices in Ottawa and Waterloo, Ontario, and San Jose, California. Our hardware, software, FPGA, verification, wireless, mechanical and signal integrity teams work to innovate, design and deliver next-generation products for customers in emerging technology markets. Fueled by 20+ years' experience and creativity, along with our collaborative and process driven approach, we turn complex challenges into well-designed solutions. And with over 400 customers and 3000+ completed projects, we have the expertise to be a seamless extension of your team, providing a clear focus and commitment to getting designs and prototypes to market faster. Once you start working with us, you'll trust us like one of your own. Our hallmark is transparency. Our guiding principle is first time right.

fidus
innovate•design•deliver

fidus.com