fidus
innovate•design•deliver

Defining and Achieving Requirements for Low-latency, High-Bandwidth Data Movement

This white paper introduces the nuances latency and bandwidth carry along with guidance for defining appropriate requirements and achieving final goals for low-latency, high-bandwidth applications.

A FIDUS SYSTEMS **WHITE PAPER**

INTRODUCTION

Over the last decade, technology has shifted from information generation and sharing to information interpretation. With Web 2.0 and the deluge of content from mobile, users have generated, stored, and shared more content than any other time in history. At first, we were not very good at extracting insights from all that information. But we're getting smarter – at least, we are trying to.

Today, we are shifting towards contextualizing data. This shift places a lot more demand on the large networks of machine learning (ML) models that rely on massive amounts of data for the training and inference phases.

It is no surprise that engineers and designers consistently aim to create systems with minimal latency, so they can be as fast as possible.

But is that really what every system needs?

It is critical to understand exactly where latency and bandwidth specifications are coming from and why each value is important for the success of the product or system.

This white paper introduces the nuances latency and bandwidth carry along with guidance for defining appropriate requirements and achieving final goals for low-latency, high-bandwidth applications.

UNDERSTANDING LATENCY AND BANDWIDTH NUANCES

The affordances of low-latency, high-bandwidth systems are well documented. The combination of low latency and high bandwidth amounts to high-throughput, high-speed data movement. However, achieving goals related to latency and bandwidth usually requires challenging trade-offs elsewhere in the system.

Engineers typically think their systems need to be faster than they truly need to be, airing on the side of extreme. In pursuit of excellence, there is a tendency to over-constrain the problem.

For example: *"We need to ingest, process, and store data from our sensors in X clock cycles."*

Some designers take this statement and create a system with a CPU hosting the maximum number of cores and running at the highest clock frequencies to ensure that all sensors are properly serviced.

Others tend to focus on piecemeal solutions rather than looking at the problem from a system-level.

For example: *"I need X GBs of data going through this pipe, so I'll use a theoretical mathematical equation to understand my bandwidth requirements."*

Assuming the line rate, specified by the interface, can be used in data rate calculations is a common example of an error based in theory. Because line rate accounts for the data rate and the protocol overhead, the achievable data rate, in practice, may be lower than the theoretical calculation implied.

In terms of bandwidth, designers often calculate values simply based on the number of channels and the clock rate. For example, a PCIe Gen3 x16 interface can theoretically support 128 Gbps based on a straight calculation of 8 Gbps per lane over 16 lanes. In practice, because PCIe leverages 128B/130B encoding, the bandwidth is reduced to approximately 126 Gbps. Other items like read/write patterns, structures associated with memory, and protocol stack latency also contribute to additional overhead. These factors are not usually addressed appropriately. Limitations reveal themselves in the implementation details, so it is important to understand the nuances of each aspect of the technology stack when it comes to key system performance metrics.

Overall, effectively defining latency and bandwidth requirements lead development teams to the most appropriate and efficient solution. For example, imagine a scenario where a surgeon controls a robotic surgery video system from a theatre next to the operating suite. That surgeon needs to see the impact of their movements in real time. Humans unknowingly tolerate many frames of video delay, but as soon as the surgeon's hand motion disagrees with their eyes, their confidence in the system waivers. If they perceive a lag, they may try to compensate by changing their movements and end up over-adjusting. A 10-frame requirement might suffice for the average non-surgeon, but the surgeon themselves may need a 3-frame requirement in practice, narrowing the solution set.

DEFINING LOW-LATENCY, HIGH-BANDWIDTH GOALS

By understanding precise requirements, you can develop a final solution that meets exact demands. There are a couple of different ways to think about requirements, represented here:

1. SPECIFICITY

When defining target specifications related to low latency and high bandwidth, consider higher order outcomes before trying to arrive at: We want X seconds of latency.

For example, consider the following two statements:

I need the car to brake immediately.

I need to stop this car before someone gets hurt.

These are two very different requirements. The former does not account for external factors like occupant safety. If a car brakes immediately, the person inside the vehicle could be injured, even if a jaywalker or an oncoming driver is not. Fidus encourages customers to understand where their quantitative goals are coming from to truly understand the problem. Estimates or exaggerations can change problem definitions and, by extension, entire systems.

Peel back the onion to understand why low latency is important for the application. An algorithm expert might come to Fidus and say, "I want you to design a PCIe Gen 5 card to run my algorithm." Here, we take a step back and ask:

Why do you need a PCIe Gen 5?

Typically, it is to accommodate perceived bandwidth needs; however, we might look at the problem and conclude they do not need the PCIe Gen 5 because the required system bandwidth can be achieved using an earlier PCIe generation with a wider bus or different topology. This would come at the expense of more nets on the PCB, but the components and implementation would be more cost effective, often achieving a significant reduction in the final bill of materials (BOM) cost compared to the PCIe Gen 5 solution. Fidus designers have the objectivity to identify and recommend these types of tradeoffs.

2. LIFECYCLE

Lifecycle (shorter or longer) also impacts the consideration set for low-latency, high-bandwidth solutions. For longer lifecycles of 10+ years, often seen in industrial applications, Fidus reduces the consideration set to focus on compatible implementation and enduring inventory. For products that have shorter lifecycles, there is a larger number of devices on the market, which provides more flexibility for implementation.

3. FUTUREPROOFING

Futureproofing not only applies to hardware and FPGAs, but also to embedded software development. Future-focused software design relies heavily on common best practices for modular code structure to accommodate future updates as library components are added and phased out. FPGA code needs to demonstrate thoughtful consideration of FPGA primitives and hardened IP blocks such that future updates are relatively seamless.

ACHIEVING LOW-LATENCY AND HIGH-BANDWIDTH GOALS

It is difficult to differentiate a leading-edge product on software alone, especially when it comes to low-latency and high-bandwidth requirements, so designers often turn to heterogeneous computing. Heterogeneous computing is a hybrid solution (where devices may include FPGA, ASICs, and CPUs or GPUs) relying on hardware when software cannot meet requirements and vice versa.

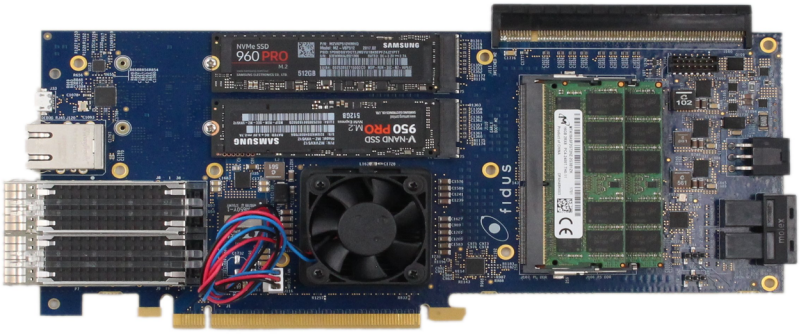
High performing low-latency, high-bandwidth solutions typically involve FPGA-based architectures due to parallelism and hardware-centric implementations that are furnished by FPGAs. The type of FPGA (i.e., size, speed, and cost) depends on the application area and system constraints.

SIDEWINDER AS A CUSTOMIZABLE SOLUTION

Fidus created Sidewinder, a leading-edge, FPGA-based networking and storage acceleration platform, to accommodate strict low-latency, high-bandwidth specifications. Achieving acceleration in networking and storage requires a combination of state-of-the-art technology selection in each domain and architectural decisions that provide access to the FPGA fabric.

Sidewinder is a PCIe card that is also well suited for any NVMe or NVMe Over Fabrics (NVMEoF, NVMF) workload acceleration. It features two 100 Gbps QSFP interfaces that designers can use to transmit and receive data. Having these interfaces tied to the FPGA fabric enables line rate processing. In today's market, a system designer could not otherwise find a processor that could keep up with 200 Gbps of data throughput.

With Sidewinder, Fidus can have software monitor and provision a system while the FPGA covers line rate processing on incoming data and stores it into drives, without software intervention. To achieve higher data transfer rates, we take a heterogeneous approach that allows us to use different compute elements to re-target a predominantly software-centric application with higher latency and lower throughput. Now, with this mixed software and hardware implementation, it is lightning fast.



Sidewinder-100TM is the world's first Xilinx® Zynq® UltraScale+TM ZU19EG Storage Accelerator PCIe card.

DESIGN WITH SIDEWINDER

ACCELERATE

Whether accelerating financial transactions or data center workload algorithms, Sidewinder's Zynq UltraScale+ can be optimally partitioned to accept, crunch, and return, both data and decisions, faster than any previous model.

DEVELOP

Even if you don't have a data center, put Sidewinder on a desktop and enjoy all of the benefits of having a Zynq UltraScale+ system.

LEARN

The Sidewinder is a great tool for university and college students to explore the latest technologies in programmable logic, heterogenous computing, networking, and storage.

CUSTOMIZE

Fidus Systems can customize Sidewinder to meet your needs. We have in-house experts in hardware, FPGA, software, signal integrity, and PCB layout. Let Sidewinder accelerate your workload, and let Fidus accelerate your product design.

ACHIEVING LOW-LATENCY, HIGH-BANDWIDTH DATA MOVEMENT

Efficient data movement requires hardware and software involvement. Each one adds a layer of complexity and latency that might not have been considered. When development teams start pulling all of the pieces together and understanding the way they interplay and operate, they are often looking at a different set of solutions than they were anticipating. Fidus designers are experienced to help engineering teams uncover and act on these insights.

With rigorous requirements definition and de-risking processes and the in-house knowledge base to access all major design disciplines, Fidus partners with clients early in development.

Interested in defining the requirements your project needs so you can move forward quickly? Contact Fidus to discuss your project today.

fidus.com/contact/

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