Solution Brief

Financial Service Industrial

Low-Latency Data Mover Framework from Algo-Logic with Intel® FPGA PAC D5005

Significantly accelerate low-latency trade system development – while offering flexibility for user algorithm add-in.

Executive Summary

For many high-performance and data center applications, low latency is a critical success factor. For example, financial trading and investment companies rely on rapid transaction execution. In many cases, these trading opportunities must be so fast that transactions execution using programmable logic implemented in an FPGA can be a real competitive advantage. Firms that trade in this fashion will have a real market advantage over those that use software alone.

Challenges

Trading systems have evolved from the specialist who worked in a trading pit on the exchange floor to automated software trading systems running on desktop computers. These systems have now evolved further to run on co-located servers with multiple optimizations to reduce trading latency. Existing trading platforms typically employ CPUs running at the fastest clock rates and network interface cards (NICs) that bypass the operating system kernel. While this approach can reduce trading latency to microseconds, these systems are no longer the fastest and are therefore no longer competitive. To further reduce latency with deterministic response times, platforms that use FPGAs are now being deployed on the global markets. These FPGA platforms run ultra-low latency trading algorithms implemented with hardware logic instead of software.

Specialized skill sets are needed to create the logic that runs in FPGA hardware. Traditionally, these designs have been coded using low-level design languages such as VHDL and Verilog. Most trading firms have teams that can code algorithms in software, but few have teams with the skills required to implement algorithms in logic. This creates an advantage for trading firms that win trades using fast logic over those firms that miss trading opportunities because their algorithms run slower in software. This solution brief discusses how to bridge this competitive gap.

Solution

Algo-Logic and Intel have developed a high-speed reference framework design that offloads the network stack required for high-speed trading to logic that runs in an Intel® Stratix® 10 FPGA on the Intel® FPGA Programmable Acceleration Card (Intel® FPGA PAC) D5005 platform. The framework includes a fast PCIe* interface (the Algo-Logic Fast Data Mover), a C/C++ to FPGA (high level synthesis (HLS)) business logic implementation area, a TCP/IP offload engine, and an ultra-low-latency (ULL) 10GbE media access control (MAC).

ALGORITHMS IN LOGIC

ALGO-LOGIC®

HTTP://ALGO-LOGIC.COM

Quote from Algo-Logic

"Algo-Logic has been helping customers move algorithms from software to FPGA hardware for multiple markets over the last decade. The newest evolution is based on the Intel® FPGA PAC D5005. This platform, together with the Low-Latency Data Mover solution from Algo-Logic, is ideal for hybrid trading strategies where algorithms can run both in software and/or in FPGA logic combined"

- **John W. Lockwood, PhD**
  CEO, Algo-Logic

Algo-Logic Systems, Inc
https://Algo-Logic.com
(408) 707-3740
IntelSupport@Algo-Logic.com

Trading system developers can directly leverage this framework to offload the network stack to FPGA logic and replace the function of a NIC with this faster platform. In addition, if a trading system developer wants to move an algorithm into logic, they can leverage the Intel® High-Level Synthesis Compiler (Intel® HLS Compiler) in the Intel® Quartus® Prime software to map C/C++ code into the FPGA. Using the Intel HLS Compiler eliminates the need to write low-level VHDL or Verilog code for the core business logic that triggers on a trade or ensures compliance by performing Pre-Trade Risk Checks. This ultra-low-latency trading platform is now available for the Intel FPGA PAC D5005 board, which has been qualified for use in major original equipment manufacturer (OEM) servers.

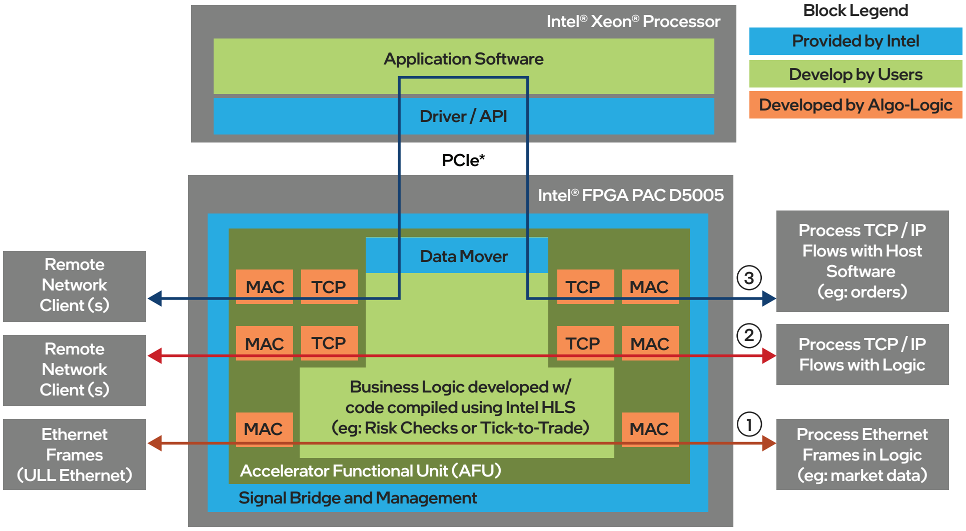
Intel® Xeon® Processor
Application Software
Driver / API
PCIe*
Intel® FPGA PAC D5005
Block Legend
Provided by Intel
Develop by Users
Developed by Algo-Logic
Remote Network Client (s)
Remote Network Client (s)
Ethernet Frames (ULL Ethernet)
Data Mover
MAC
TCP
TCP
MAC
MAC
TCP
TCP
MAC
Business Logic developed w/ code compiled using Intel HLS (eg: Risk Checks or Tick-to-Trade)
MAC
MAC
Accelerator Functional Unit (AFU)
Signal Bridge and Management
3
2
1
Process TCP / IP Flows with Host Software (eg: orders)
Process TCP / IP Flows with Logic
Process Ethernet Frames in Logic (eg: market data)

Figure 1 Low-Latency Data Framework Block Diagram

Low-Latency Data Framework

There are three ways to process data with the low-latency data framework as shown in Figure 1:

1) As Ethernet frames with the Ultra-Low Latency (ULL) MAC

2) In-line using the MAC and TCP offload engine

3) In software using the MAC, TCP, and Intel's Low-Latency Data Mover

Algo-Logic's Ultra-Low-Latency MAC

Algo-Logic's ULL MAC reduces roundtrip network latency by several hundred nanoseconds as compared to MACs designed only for high throughput. Algo-Logic Systems' ULL MAC design implements 10GBASE-R MAC and physical coding sub-layer (PCS) functionality in FPGA logic to optimize latency and includes the following features:

- Ultra-low-latency round-trip packet transfers, as measured from fiber-to-fiber or from gate-to-gate at line rate for 10 Gigabit Ethernet (10GE)
- Local fault and remote fault detection
- Frame Check Sequence (FCS) insertion and verification at line rate
- Automatic transmit padding, jumbo frame support, transmit and receive statistics counters

Algo-Logic's TCP Offload Engine (TOE)

Algo-Logic's TCP Endpoint implements a reliable streaming network stack in FPGA logic. It allows applications implemented in logic to be directly connected to reliable network sockets by opening, maintaining, and closing TCP/IP Connections. The TCP/IP core was designed using RTL design for optimal performance and supports optional cut-through functions to receive (RX) and transmit (TX) data without extra store-and-forward delays.

Adjustable parameters include:

- Retransmission timeouts
- Size of shared on-chip retransmission buffer
- Fast retransmission
- Limits on retransmissions
- TX rewind

Low-Latency Data Mover

Intel's Low-latency Data Mover implements bidirectional data transfer between the Intel® Xeon® CPU host and the Intel® Stratix® 10 FPGA with an approximate latency of just 600 ns in each direction.

- Data and status/control messages are moved across the PCIe link using cut-through techniques
- The direct memory access (DMA) is optimized to move small packets
- The FPGA operates on the data while it arrives instead of waiting for receipt of an entire packet
- A low-level, user-space driver is provided for direct access to payload, status, and synchronization information.

Benefits of Trading with Logic

	ONLY OFFLOAD ETHERNET ENGINE	OFFLOAD TRADING ALGORITHM + ETHERNET ENGINE
Increase Performance	▪ Equivalent to 3rd party kernel bypass NIC	▪ 3.8X lower latency in pass through tests[1] ▪ Deterministic, jitter-free packet forwarding
Speed Time to Value	▪ Supported for deployment on a high-volume hardware platform ▪ Prebuilt logic saves development effort	

Learn More

Please visit **Algo-Logic** to see additional IP Cores and reference designs for Tick-to-Trade (T2T) and Pre-Trade Risk Checks (PTRCs) that can be integrated into the Intel FPGA PAC D5005 framework and reference designs for Financial Services applications.

[1] Testing by Algo-Logic on October 26, 2020. Server configuration: HPE DL380 G10, CPU = Intel® Xeon® Gold processor 6154 @ 3.00 GHz; DRAM = 128 GB total, and RHEL* 7.6. Production Intel FPGA PAC D5005. For more information, a benchmark report is available under NDA. Contact your sales rep for more information.

Intel does not control or audit third-party data. You should consult other sources to evaluate accuracy.

Performance varies by use, configuration and other factors. Learn more at **www.Intel.com/PerformanceIndex**.

Performance results are based on testing as of dates shown in configurations and may not reflect all publicly available updates. See backup for configuration details. No product or component can be absolutely secure.

Your costs and results may vary.

Intel technologies may require enabled hardware, software or service activation.