Research Report: The Arista 7124FX Switch as a High Performance Trade Execution Platform

Abstract:

Many groups are working on reducing trading execution latency - the time from a critical Ethernet frame (such as market feed or execution instruction message) to a corresponding market order being sent out. This report, an adjunct to the October 2012 Finteligent report titled "*Research Report: 10GbE Low Latency Networking Technology Review*" describes an approach where market feed analysis and order execution is performed directly on an Arista Application Switch to achieve an order of magnitude reduction in latency.

September 2013

1 Summary

This document is an adjunct to the report published by the Finteligent Trading Technology Community (FTTC) in October 2012 titled "*Research Report: 10GbE Low Latency Networking Technology Review*" and related reports released throughout 2012/13. These reports are available upon registration here.

FTTC is an informal community of technology vendors who collaborate to facilitate tests of architecture designs across the technology stack as it applies to high performance trading. The community has tested the impact of CPUs, servers, switches, virtualisation and FIX engines on overall performance as reflected in latency figures.

A standard and consistent test harness has been used across all the tests, such that comparison can be made between design concepts and products which are interchanged.

The majority of the tests have been conducted in the Intel fasterLAB facility in Winnersh near London, England using resources from OnX Enterprise Solutions Inc, Intel Corporation and collaborating vendors. This report presents the findings of tests conducted by Argon Designs in their Cambridge, England labs – using the same test harness and test rig configuration, with the objective of assessing the effect of heterogeneous hardware architectures (x86 + FPGA) to enhance trading systems' performance.

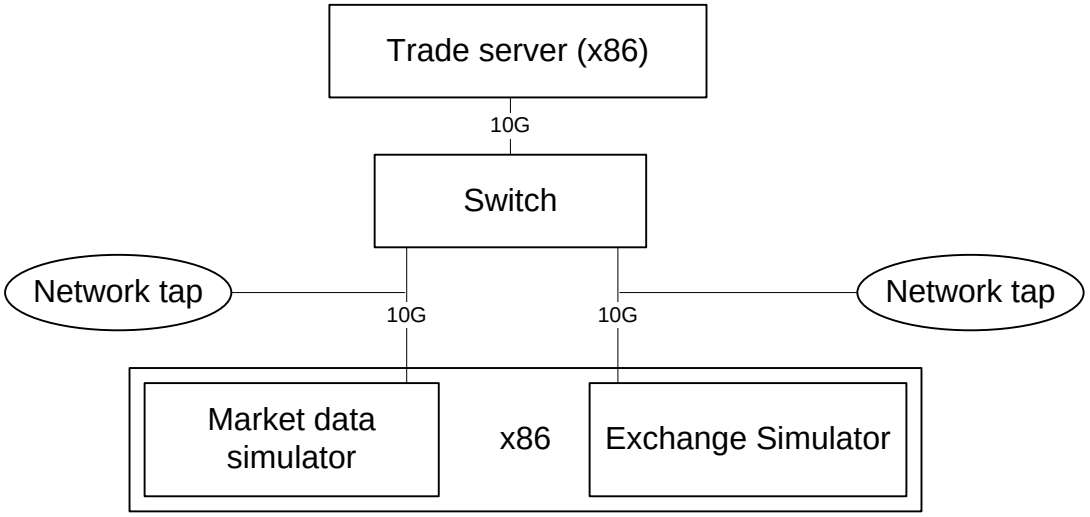
The best results achieved by FTTC for average latency from an all x86 environment was **4,663ns**. The Argon Design tests delivered latency of **170ns** for the same comparative measured leg of the trade – indicating that quantum gains can be achieved from hardware architecture design of the trading technology stack.

The design techniques, methodology and environment are described fully below.

2 The Finteligent setup

2.1 The hardware setup

The diagram below summarizes the hardware setup of the standard test rig developed by FTTC. An x86 trade server interfaces through a switch with a basic server which hosts a market data simulator and a venue/exchange simulator, both based on the FIX protocol. The switch is tapped on the ports connected to the simulators. The taps capture packets using a precise timestamping Endace card. The latency figures are calculated in a post-processing stage. (See full description at the end of section 4, page 9, in the Finteligent report.)



The x86 trade server (2x Intel Xeon E5-2690 @ 2.9GHz, with 192 GB of RAM) is a high end specification and has been optimised for low latency. The network cards implement kernel-bypass, and various power and networking settings have been tweaked in the BIOS and elsewhere. (See the start of section 5, page 9 of the Finteligent report.)

2.2 The trading algorithm

The standard test harness has a very simple trading algorithm which is used to generate the trades; its function is to get message data into the infrastructure allowing the focus of the benchmark to be on the hardware components under test. The algorithm consists of two parts, the “buy” part which is triggered by Ethernet frames from the market data simulator, and the “sell” part which is triggered by Ethernet frames from the trade matching function in the exchange/venue simulator.

The “buy” and “sell” parts are similar. Both involve sending market orders in response to FIX messages from simulators. The aim for high-frequency traders is to minimise the response latencies of the switch plus trade server system when sending those market orders.

(For more details, see page 8 of the Finteligent report.)

2.2.1 Test harness: The “buy” part

The market data simulator broadcasts Market Data Incremental Refresh messages where the stock symbols are taken at random from a list of stocks. Independently for each stock, the price is incremented in units of $0.001 on each update. Whenever such a feed update has a price with decimal part .000 the trade server should send a BUY market order to the exchange simulator, requesting to buy 100 shares of the given stock at the given price.

2.2.2 Test harness: The “sell” part

Whenever the exchange simulator receives a BUY market order from the trade server, it immediately sends an execution report confirming receipt of the BUY order. Some amount of time later, the exchange simulator sends another execution report informing the trade server

that the BUY order has been filled. When such an execution report is received, the trade server should send a SELL market order to the exchange simulator, requesting to sell the 100 shares it originally bought.

2.3 The results

As seen in the series of FTTC reports, a number of products and technologies were tested across the stack. In this optimised x86 hardware setup and simplified algorithmic setting, the best average latency achieved was **4,663ns**. This occurred with the Mellanox ConnectX-3 EN network card (specifically, for the "buy" strategy under a load of 71k market data updates per second).

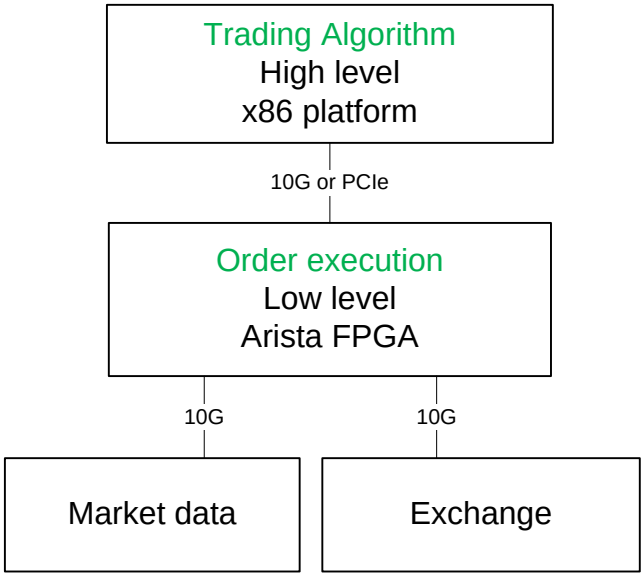
3 The Argon Design test scenario

In this test exercise the Argon Design trading stack is based on the Arista FX 7124 switch. This Arista switch is at the forefront of current design thinking and is unconventional because it contains a large FPGA (from Altera) with hardware-level access to 8 of its 24 10Gb Ethernet ports. By introducing an FPGA embedded alongside the switch ASIC and working in collaboration with x86 components, this design allows high end hardware technologies to be brought into the standard equipment mix in CoLo racks in exchanges and venues for high performance – HFT – trading strategy execution.

3.1 The philosophy

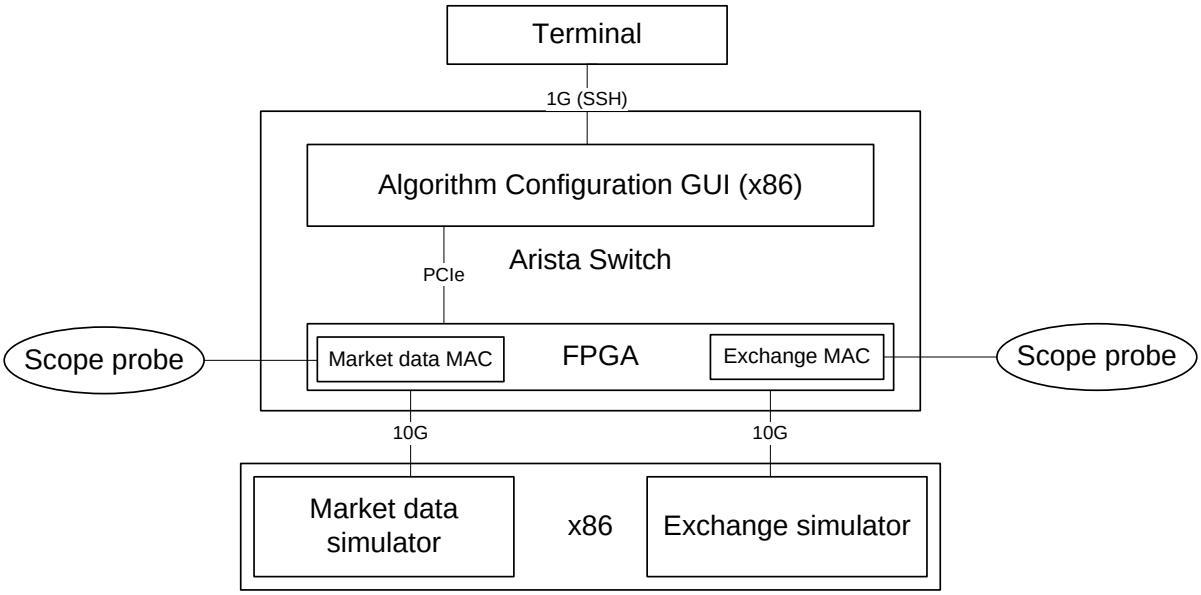
The following diagram shows the "fast-path" approach to making market orders. The high level trading algorithm, run on an x86 platform, delegates the low level order execution to the FPGA in the Arista switch. The insight is that by decoupling trading algorithm and order execution, execution times can be taken right down while retaining the flexibility and scale of the x86 platform as a high-level numerical analysis platform. This approach uses the best hardware technology for the task in hand – it is not "either-or".

For firms thinking about their technology strategy to keep them at the high end of performance, it allows the option of retaining x86 code libraries (and taking advantage of CPU innovation such as many core) while using FPGA coding techniques to handle discrete tasks in a complimentary manner. Working in tandem the results deliver more than an order of magnitude improvement.



3.2 Argon Design FPGA-x86 architecture set up

As a proof-of-concept demonstration the order execution functions have been decoupled and coded to the FPGA part of the Arista 7124FX switch. The diagram below shows the hardware setup.



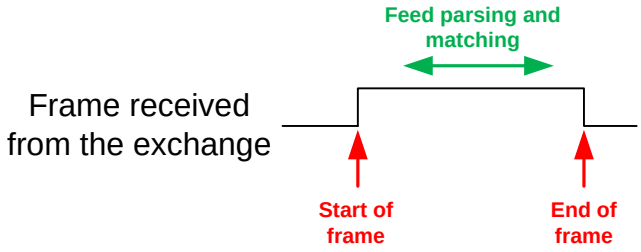
At the bottom of the above topology, the same market data and exchange simulator test harness from the Finteligent consortium has been used on a similar specification server. Moving up, the passive switch in the Finteligent setup is replaced by an active Arista FPGA Application Switch. The network taps in the Finteligent setting are replaced by scope probes at the FPGA MAC layer. At the highest level, the trade server is replaced by a terminal where high level trading strategies are configured by an operator of the test harness itself and executed in real time by the switch. This is discussed further in the last section.

3.3 Latency improvement techniques available to FPGAs

We use two techniques which are made possible with the use of an FPGA.

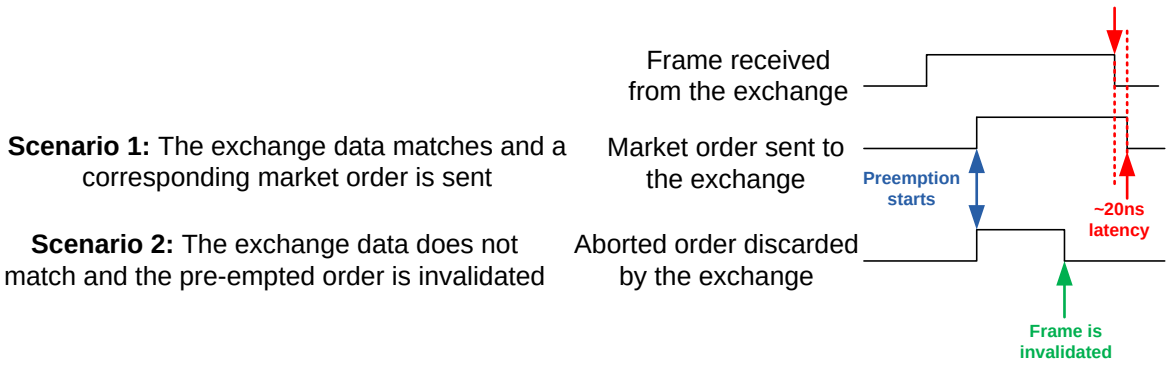
3.3.1 Inline parsing and matching

The first is inline parsing and matching of the FIX messages. As the Ethernet frames enter the FPGA, the FIX key/value pairs get parsed as soon as possible. This allows for partial information to be extracted before even the whole frame has been received.



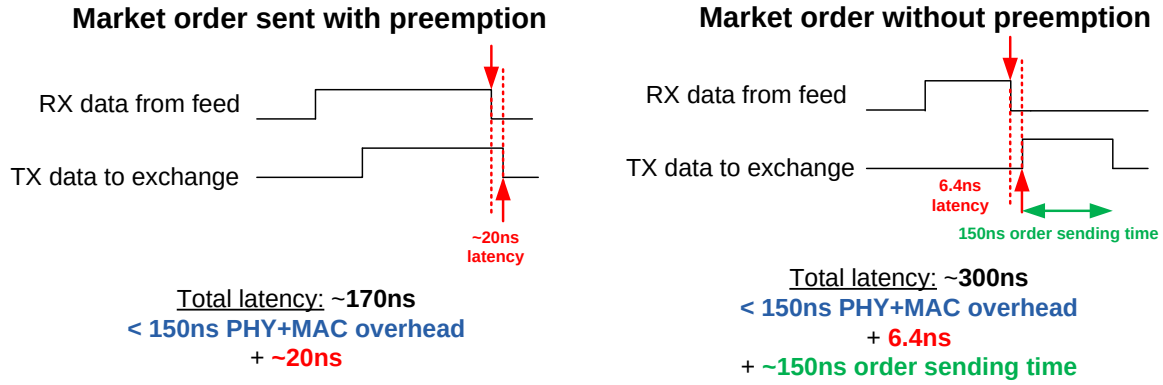
3.3.2 Pre-emption of market orders

The second technique is what we call “pre-emption”. Instead of waiting until the end of the FIX messages before acting upon them, we start sending the overhead part of the response which contains the Ethernet, IP, TCP and FIX headers. This allows for sending a populated response right after a match is made. If a match is not made, the Ethernet frame is “poisoned” by using an invalid Frame Checksum. This aborted Ethernet Frame will be discarded by the network device at the other end of the cable.

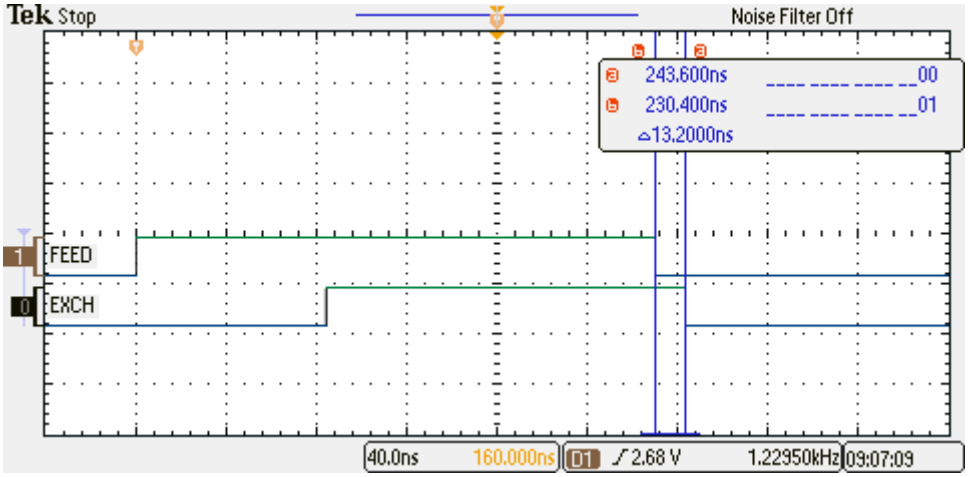


3.4 Latency results

For the PHY and MAC layers, the ultra-low latency solution by Tamba Networks was used, which has a roundtrip latency of 150ns. The diagram below explains the other latencies involved, both when preemption is enabled and disabled.



Latencies were already measured using a scope. We attach a screenshot below.



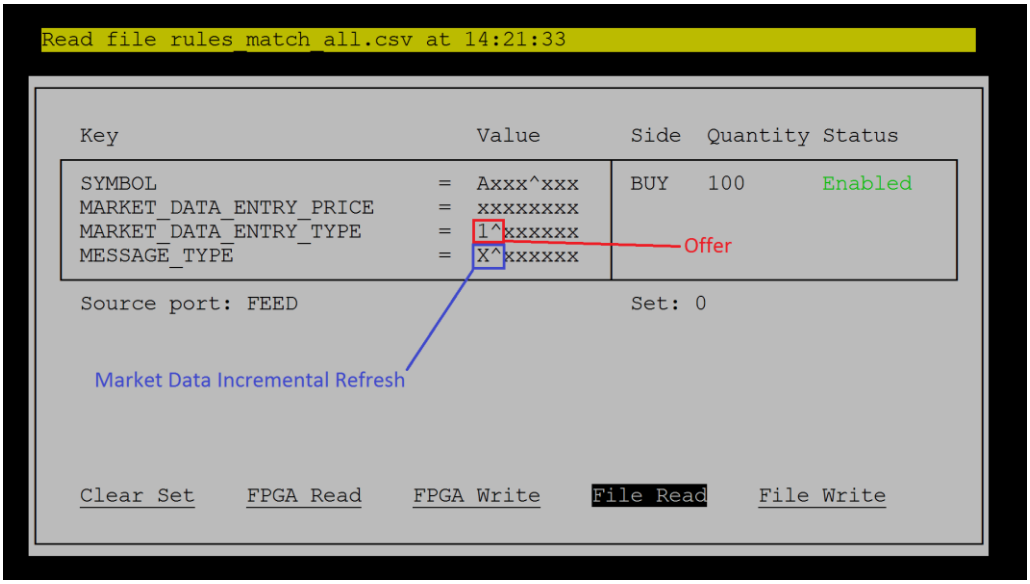
3.5 Matching of market data

The incoming market data feeds and execution message transports use the FIX protocol. FIX messages are a series of key/value pairs. For example, the keys could correspond to “price” and “symbol” and the values could correspond to “$492.120” and “AAPL”.

For each port with incoming market information a set of trigger events are defined, each with an associated action. Each trigger event is a set of key/value rules all of which are compared to the incoming FIX messages by a strategy matcher, and must all be true for the order to be completed. The comparison operators for the demo are equal, not equal, greater than and less than.

3.5.1 Real time configuration of trigger events

For the demonstration, a terminal is used to configure the trigger events in real time. The screenshot below shows an example trigger on the market feeds. The key/value rules are on the left, and the action on the right.



In a real system the terminal and algorithm configuration GUI would be replaced by a high performance x86 trading algorithm platform that issued rule sets to the FPGA. This would support the high trading volumes of a real world scenario.

4 Conclusion

The FTTC initiative has delivered very useful comparative data on performance over the 2012-2013 period. Argon Design's initiative in adding FPGA hardware has proved conclusively the advantages to be gained from such hybrid designs.

The potential of this kind of approach to technology design has been known to the market for some time, but the lack of widespread skills and the significant historical investment in x86 code libraries has meant that take up has been relatively slow and limited to niche ecosystems.

This exercise has proved the potential of treating x86 and FPGA as complimentary technologies that can work together. The key is the partitioning of tasks and the ability of companies like Argon Design to apply leading edge engineering skills to this specialist task. These proof points and the availability of skills aligned to trading workloads indicate that the technology is poised for prime time.