A Low Latency Library in FPGA Hardware for High Frequency Trading (HFT)

Analysis and implementation

Lorenzo Borella Samuele Pio Lipani Marco Giunta

Physics and Astronomy department
University of Padua

Paper review, April 2021

Contents

Contents

What is HFT?

High frequency trading (HFT) is:

- Trading of equities, options, feautures at high speed in large volumes

What is HFT?

High frequency trading (HFT) is:

- Trading of equities, options, feautures at high speed in large volumes
- Earning money by exploiting the fleeting variation in stock price or demand

What is HFT?

High frequency trading (HFT) is:

- Trading of equities, options, feautures at high speed in large volumes
- Earning money by exploiting the fleeting variation in stock price or demand

What is HFT?

High frequency trading (HFT) is:

- Trading of equities, options, feautures at high speed in large volumes
- Earning money by exploiting the fleeting variation in stock price or demand

HFT accounts

- 70 % of all trades in US Markets in 2010

What is HFT?

High frequency trading (HFT) is:

- Trading of equities, options, feautures at high speed in large volumes
- Earning money by exploiting the fleeting variation in stock price or demand

HFT accounts

- 70 % of all trades in US Markets in 2010
- It continues to grow

What is HFT?

High frequency trading (HFT) is:

- Trading of equities, options, feautures at high speed in large volumes
- Earning money by exploiting the fleeting variation in stock price or demand

HFT accounts

- 70 % of all trades in US Markets in 2010
- It continues to grow

What is HFT?

High frequency trading (HFT) is:

- Trading of equities, options, feautures at high speed in large volumes
- Earning money by exploiting the fleeting variation in stock price or demand

HFT accounts

- 70 % of all trades in US Markets in 2010
- It continues to grow

What is HFT?

High frequency trading (HFT) is:

- Trading of equities, options, feautures at high speed in large volumes
- Earning money by exploiting the fleeting variation in stock price or demand

HFT accounts

- 70 % of all trades in US Markets in 2010
- It continues to grow

HFT involves

- Using computers to place orders based on pre-defined algorithms

Traders

Definition

A trader is an individual who engages in the buying and selling of financial assets in any financial market, either for himself or on behalf of another person or institution.

[1] Overnight positions refer to open trades that have not been liquidated by the end of the normal trading day, which are quite common in foreign exchange and futures markets.

Traders

Definition

A trader is an individual who engages in the buying and selling of financial assets in any financial market, either for himself or on behalf of another person or institution.

What's the difference between a trader and an investor?

[1] Overnight positions refer to open trades that have not been liquidated by the end of the normal trading day, which are quite common in foreign exchange and futures markets.

Traders

Definition

A trader is an individual who engages in the buying and selling of financial assets in any financial market, either for himself or on behalf of another person or institution.

What's the difference between a trader and an investor?

- A trader holds an asset for a shorter period of time to capitalize on short-term trends; investors use also overnight positions.[1]

[1] Overnight positions refer to open trades that have not been liquidated by the end of the normal trading day, which are quite common in foreign exchange and futures markets.

Traders

How do traders approach the stock exchange?

Traders

How do traders approach the stock exchange?

Trading venues make current pricing information available as continuous streams of electronic data then:

- Traders monitor these streams of data, asking for relevant stocks and assets to determine when and what to trade

Traders

How do traders approach the stock exchange?

Trading venues make current pricing information available as continuous streams of electronic data then:

- Traders monitor these streams of data, asking for relevant stocks and assets to determine when and what to trade
- Exploiting fleeting variations in stocks price they accumulate profit by making tiny gains on large numbers of transactions.

Traders

In practice:

[2]The term order book refers to an electronic list of buy and sell orders for a specific security or financial instrument organized by price level.

Traders

In practice:

- They receive multiple high-datarate UDP/IP market data streams over the network from trading venues (e.g Wall street, Piazza Affari)

[2]The term order book refers to an electronic list of buy and sell orders for a specific security or financial instrument organized by price level.

Traders

In practice:

- They receive multiple high-datarate UDP/IP market data streams over the network from trading venues (e.g Wall street, Piazza Affari)
- A first *line-handling* practice is to use redundancy over exchanges to reconstruct order books[2] from lost packets.

[2]The term order book refers to an electronic list of buy and sell orders for a specific security or financial instrument organized by price level.

Traders

In practice:

- They receive multiple high-datarate UDP/IP market data streams over the network from trading venues (e.g Wall street, Piazza Affari)
- A first *line-handling* practice is to use redundancy over exchanges to reconstruct order books[2] from lost packets.
- Traders employ a range of algorithms and heuristics to decide when and what to trade

[2]The term order book refers to an electronic list of buy and sell orders for a specific security or financial instrument organized by price level.

Traders

In practice:

- They receive multiple high-datarate UDP/IP market data streams over the network from trading venues (e.g Wall street, Piazza Affari)
- A first *line-handling* practice is to use redundancy over exchanges to reconstruct order books[2] from lost packets.
- Traders employ a range of algorithms and heuristics to decide when and what to trade
- The final step is to enter these orders into the exchange via a broker

[2]The term order book refers to an electronic list of buy and sell orders for a specific security or financial instrument organized by price level.

Brokers

Definition

The function of a Broker in the trading infrastructure is to be a market participant who takes orders from clients (traders) and passes them to the exchange. Brokers manage risk according to regulatory requirements

Brokers

Definition

The function of a Broker in the trading infrastructure is to be a market participant who takes orders from clients (traders) and passes them to the exchange. Brokers manage risk according to regulatory requirements

How do they make profit?

Brokers

Definition

The function of a Broker in the trading infrastructure is to be a market participant who takes orders from clients (traders) and passes them to the exchange. Brokers manage risk according to regulatory requirements

How do they make profit?

- Each venue charges a small *matching fee* per transaction

Brokers

Definition

The function of a Broker in the trading infrastructure is to be a market participant who takes orders from clients (traders) and passes them to the exchange. Brokers manage risk according to regulatory requirements

How do they make profit?

- Each venue charges a small *matching fee* per transaction
- The optimum execution venue must be chosen to ensure both the fastest execution and the lowest transaction fees

Brokers

Definition

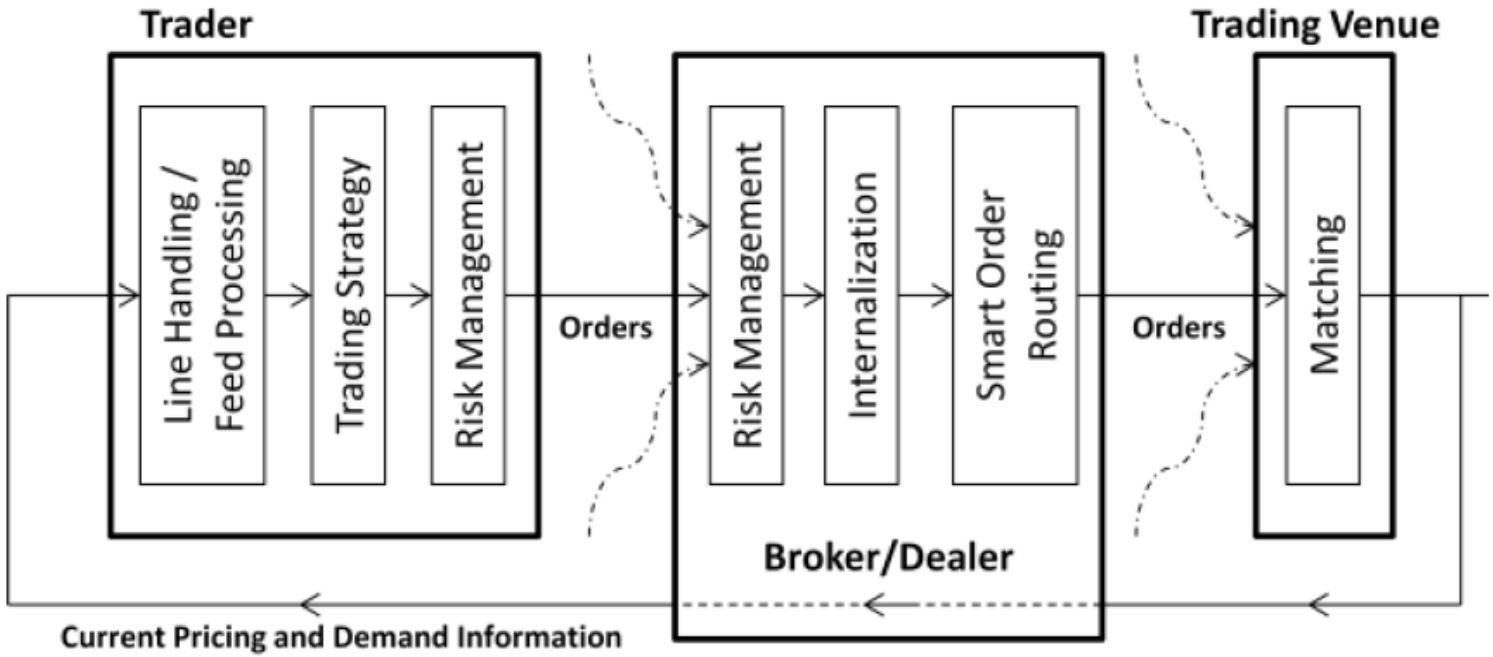
The function of a Broker in the trading infrastructure is to be a market participant who takes orders from clients (traders) and passes them to the exchange. Brokers manage risk according to regulatory requirements

How do they make profit?

- Each venue charges a small *matching fee* per transaction
- The optimum execution venue must be chosen to ensure both the fastest execution and the lowest transaction fees
- They match orders from different clients with each other internally, without sending them to the exchange

From traders to exchange path

Block diagram of traders, brokers and venues



Challenges in Financial Markets

Main challenges

Challenges in Financial Markets

Main challenges

- Latency: It's useful to execute orders faster than other investors to capture fleeting variations in price and demand in the markets

Challenges in Financial Markets

Main challenges

- Latency: It's useful to execute orders faster than other investors to capture fleeting variations in price and demand in the markets
- Jitter: it represents variations in latency while processing packets data from the market. Reducing jitter provides consistent and fair executions

Challenges in Financial Markets

Main challenges

- Latency: It's useful to execute orders faster than other investors to capture fleeting variations in price and demand in the markets
- Jitter: it represents variations in latency while processing packets data from the market. Reducing jitter provides consistent and fair executions

Challenges in Financial Markets

Main challenges

- Latency: It's useful to execute orders faster than other investors to capture fleeting variations in price and demand in the markets
- Jitter: it represents variations in latency while processing packets data from the market. Reducing jitter provides consistent and fair executions

Secondary challenges

Challenges in Financial Markets

Main challenges

- Latency: It's useful to execute orders faster than other investors to capture fleeting variations in price and demand in the markets
- Jitter: it represents variations in latency while processing packets data from the market. Reducing jitter provides consistent and fair executions

Secondary challenges

- Throughput: it implies handling large volume of data

Challenges in Financial Markets

Main challenges

- Latency: It's useful to execute orders faster than other investors to capture fleeting variations in price and demand in the markets
- Jitter: it represents variations in latency while processing packets data from the market. Reducing jitter provides consistent and fair executions

Secondary challenges

- Throughput: it implies handling large volume of data
- Flexibility: due to the sudden and persistent change of the market it allows to adapt to changing risks and trading strategies

Contents

Software based approach to HFT

Observation

Most of traders and brokers today implement HFT in software or commodity servers.

Software based approach to HFT

Observation

Most of traders and brokers today implement HFT in software or commodity servers.

Advantages of software based:

- Working on commodity servers allows algorithms to be expressed in familiar high-level programming languages and be re-compiled quickly to make improvements.

Software based approach to HFT

Observation

Most of traders and brokers today implement HFT in software or commodity servers.

Advantages of software based:

- Working on commodity servers allows algorithms to be expressed in familiar high-level programming languages and be re-compiled quickly to make improvements.

Software based approach to HFT

Observation

Most of traders and brokers today implement HFT in software or commodity servers.

Advantages of software based:

- Working on commodity servers allows algorithms to be expressed in familiar high-level programming languages and be re-compiled quickly to make improvements.

Disadvantages:

Software based approach to HFT

Observation

Most of traders and brokers today implement HFT in software or commodity servers.

Advantages of software based:

- Working on commodity servers allows algorithms to be expressed in familiar high-level programming languages and be re-compiled quickly to make improvements.

Disadvantages:

- Most of latency comes from the computers operating system networking stack

Software based approach to HFT

Observation

Most of traders and brokers today implement HFT in software or commodity servers.

Advantages of software based:

- Working on commodity servers allows algorithms to be expressed in familiar high-level programming languages and be re-compiled quickly to make improvements.

Disadvantages:

- Most of latency comes from the computers operating system networking stack

Software based approach to HFT

Observation

Most of traders and brokers today implement HFT in software or commodity servers.

Advantages of software based:

- Working on commodity servers allows algorithms to be expressed in familiar high-level programming languages and be re-compiled quickly to make improvements.

Disadvantages:

- Most of latency comes from the computers operating system networking stack

To reduce this source of latency, providers are implementing network interface cards that allow to bypass the operating system kernel.

Hands on latency data, current approaches

Software

1. Linux 10GE NICs
   - $15 - 20$ μs for Half-Round Trip Time through un-optimized kernel
   - TCP Offload: 2.9 μs Transmit $+6$ μs Receive for half round trip
2. Datagram Bypass Layer (DBL)
   - $3.5\mu s$ for UDP and $4.0\mu s$ for TCP
3. Infiniband MPI
   - $1\mu s$ (excluding application layer)

Hardware

1. Graphics Processor (GPU)
   - Optimized for throughput, but not optimized for low latency
   - Incurs additional overhead of passing data through PCIe bus
2. ASIC
   - Achieves sub-μs latency
   - But lacks flexibility to handle new protocols and features
3. FPGA
   - Provides $0.2\mu s$ latency w/TCP
   - Has the flexibility to support new protocols and features

Contents

Traditional hybrid computing with FPGAs

Both software and hardware are exploited; FPGA-based coprocessors accelerate CPUs. Processing time can be improved up to three orders of magnitude (e.g in risk management, option pricing and portfolio modelling).

Traditional hybrid computing with FPGAs

Both software and hardware are exploited; FPGA-based coprocessors accelerate CPUs. Processing time can be improved up to three orders of magnitude (e.g in risk management, option pricing and portfolio modelling).

Advantages

Traditional hybrid computing with FPGAs

Both software and hardware are exploited; FPGA-based coprocessors accelerate CPUs. Processing time can be improved up to three orders of magnitude (e.g in risk management, option pricing and portfolio modelling).

Advantages

- CPUs introduce power processing cores, large multi level caches and are best suited to control-intensive parts of an application

Traditional hybrid computing with FPGAs

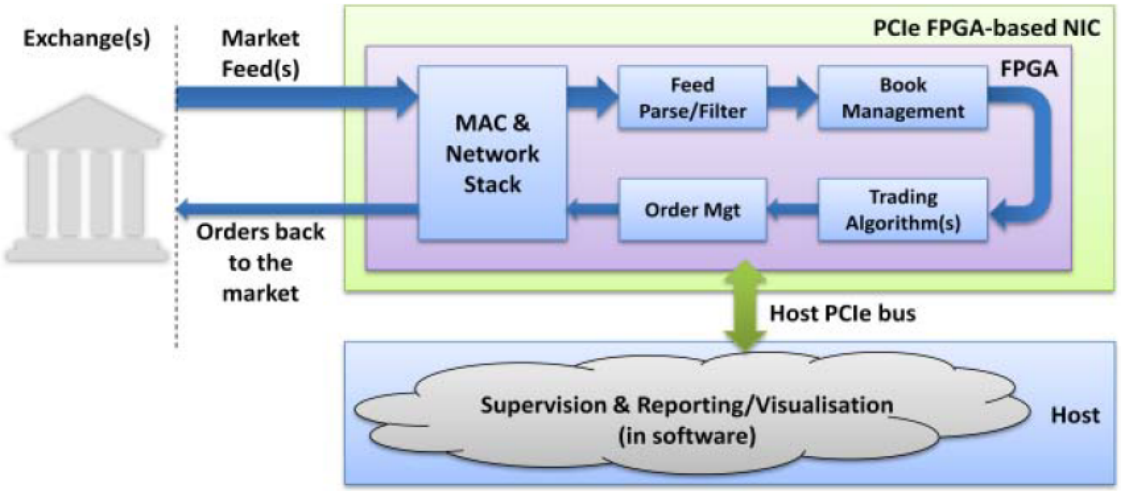
Both software and hardware are exploited; FPGA-based coprocessors accelerate CPUs. Processing time can be improved up to three orders of magnitude (e.g in risk management, option pricing and portfolio modelling).

Advantages

- CPUs introduce power processing cores, large multi level caches and are best suited to control-intensive parts of an application
- FPGAs perform best at non-floating-point tasks such as integer, binary, character or fixed-point data processing

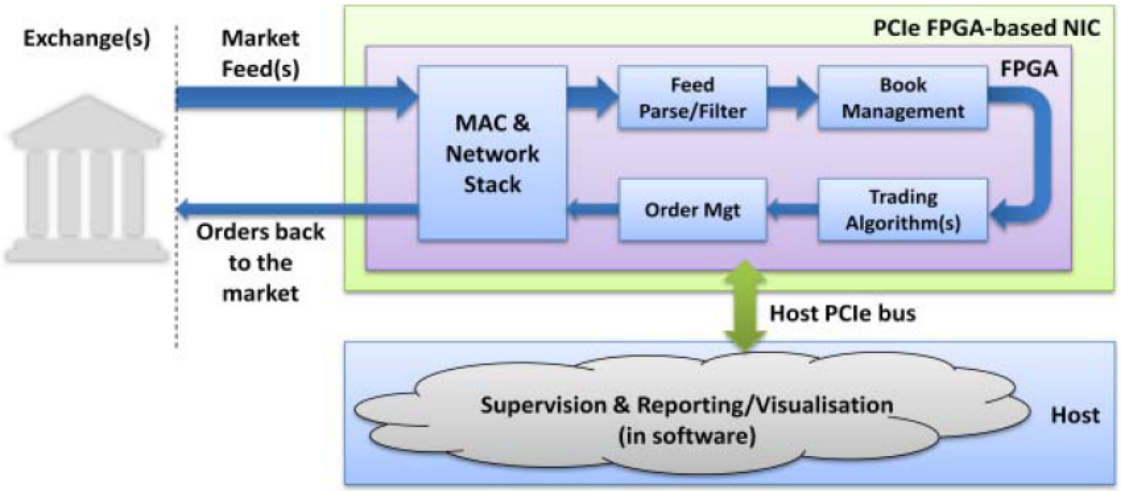
Smart-NICs based hybrid computing

- CPU is connected to one or more FPGA modules via PCIe bus



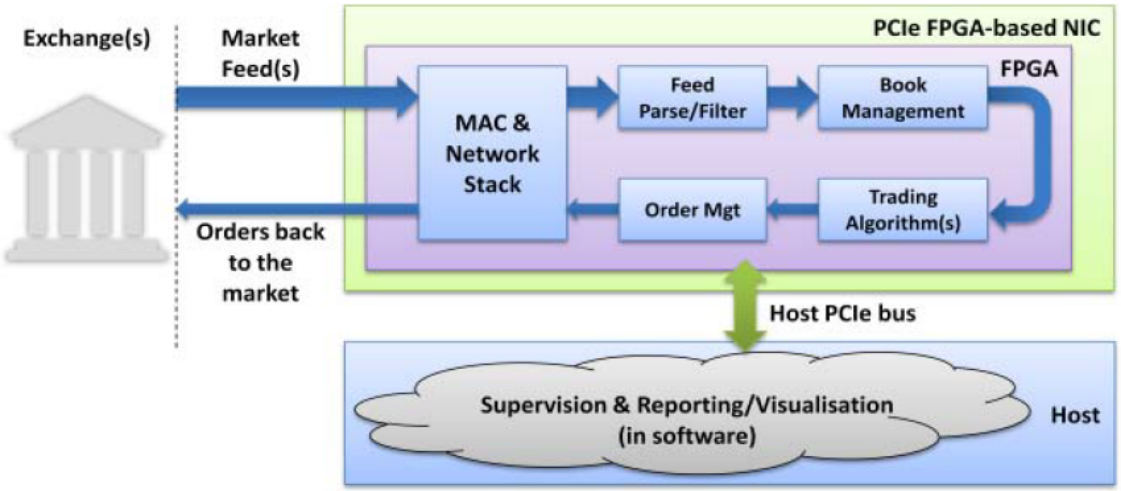
Smart-NICs based hybrid computing

- CPU is connected to one or more FPGA modules via PCIe bus
- FPGA-based NICs can turn around financial trades with a predictable latency of 2.7μs (considering the NASDAQ protocol)



Smart-NICs based hybrid computing

- CPU is connected to one or more FPGA modules via PCIe bus
- FPGA-based NICs can turn around financial trades with a predictable latency of 2.7μs (considering the NASDAQ protocol)
- The software counterpart without kernel bypass displays a value of $38 \pm 20\mu s$



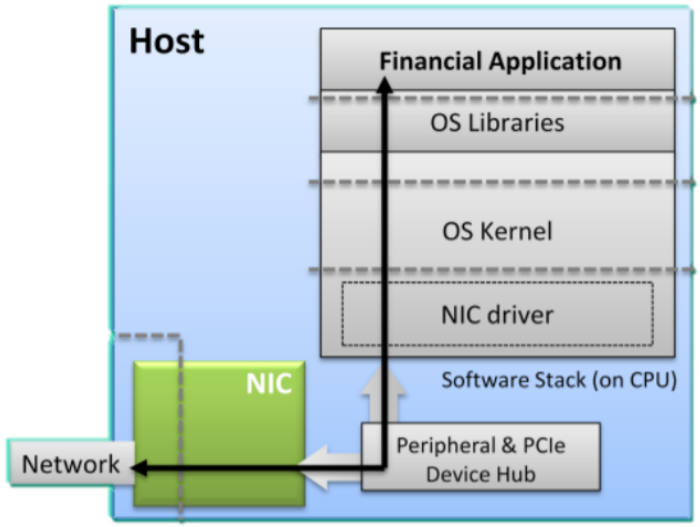
Smart-NICs based hybrid computing

Disadvantages

Requiring network traffic to be received on a network adapter implies:

- Unpredictable PCIe bus transfer: processes could be piled up before executed even if FPGA work in parallel pipelines, increasing execution time
- Memory copy
- Potential cache misses
- Amdahl's law

CPU + NIC + FPGA Offload



Pure FPGA computing

Observation

Some financial processing systems may be completely implemented on FPGA-based platforms without any host CPU yielding best possible latencies. Incoming market data can be processed directly on the FPGA without needing to travel over and back to a host CPU.

Pure FPGA computing

Observation

Some financial processing systems may be completely implemented on FPGA-based platforms without any host CPU yielding best possible latencies. Incoming market data can be processed directly on the FPGA without needing to travel over and back to a host CPU.

Packet reading: In contrast with software-based approach there are:

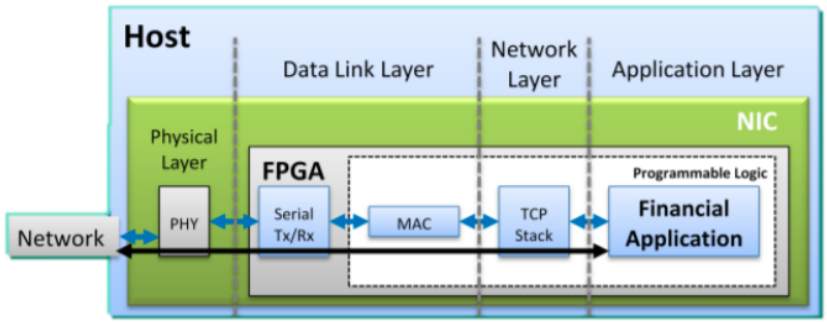
- No bus copy
- No memory copy
- No cache misses
- No parallel execution limitations
- No interrupt-driven software stacks

Pure FPGA computing

Advantages

- The incoming network data is fed directly into an application-specific processing pipeline via hardware PHY and MAC blocks.

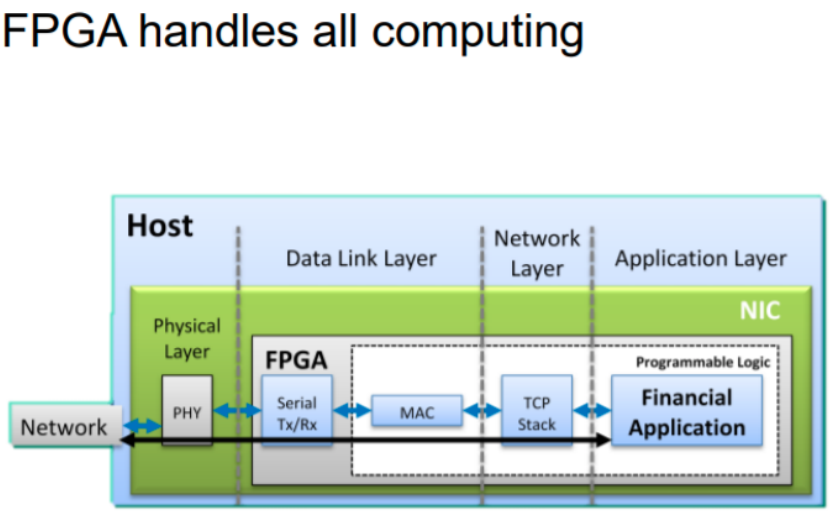
FPGA handles all computing



Pure FPGA computing

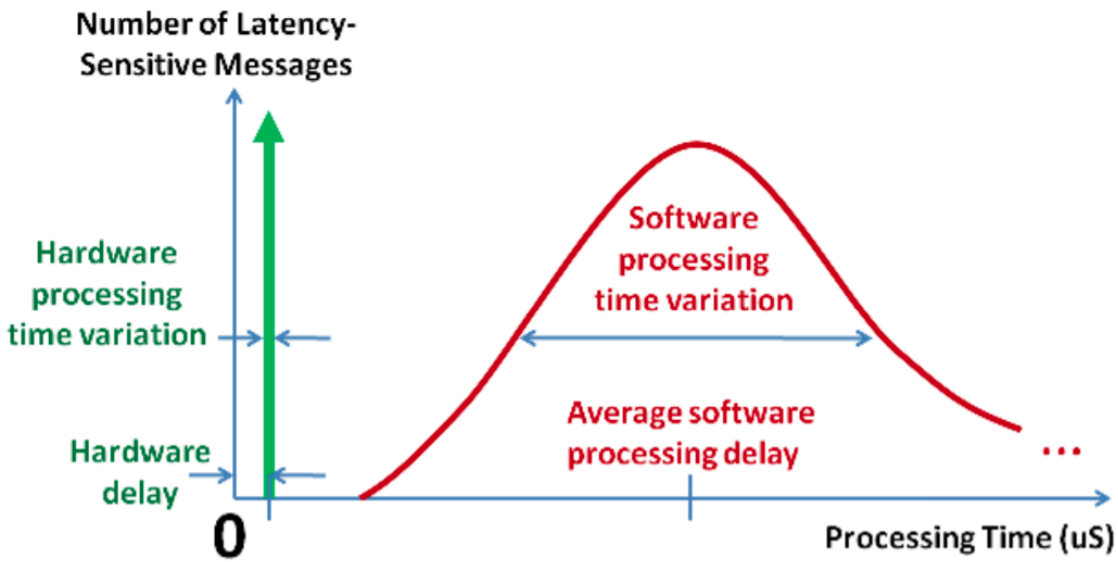
Advantages

- The incoming network data is fed directly into an application-specific processing pipeline via hardware PHY and MAC blocks.
- Relevant information within a packet can in fact be extracted before the complete packet is received, and the available bit-level access to incoming data greatly simplifies protocol parsing (a task tedious to describe using a standard programming language).



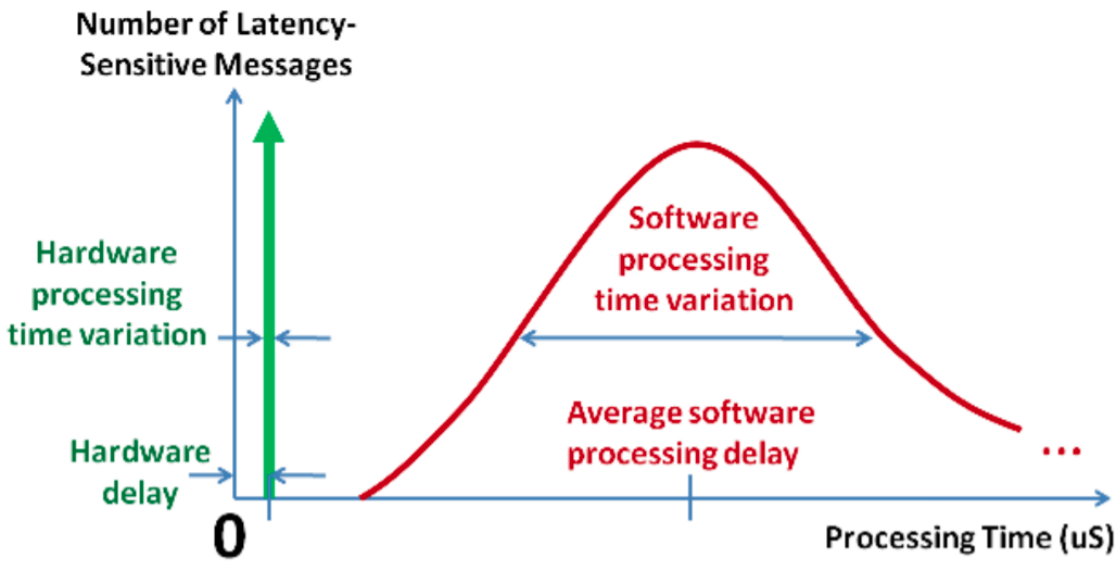
Pure FPGA computing

- The availability and the processing time for any piece of information is completely predictable down to a clock cycle



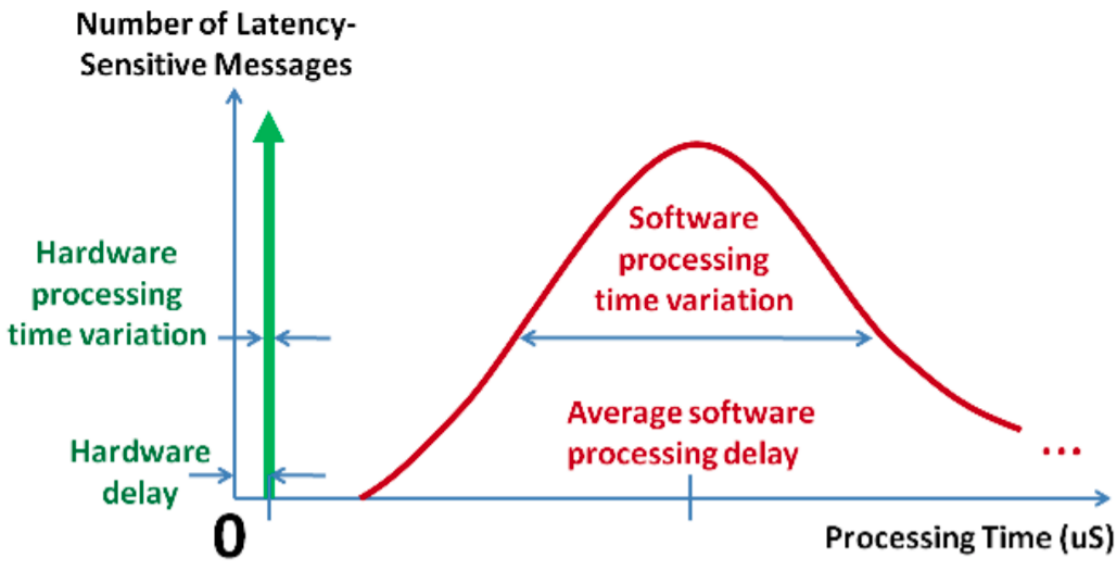
Pure FPGA computing

- The availability and the processing time for any piece of information is completely predictable down to a clock cycle
- Significantly lower latencies, high throughput with minimal jitter



Pure FPGA computing

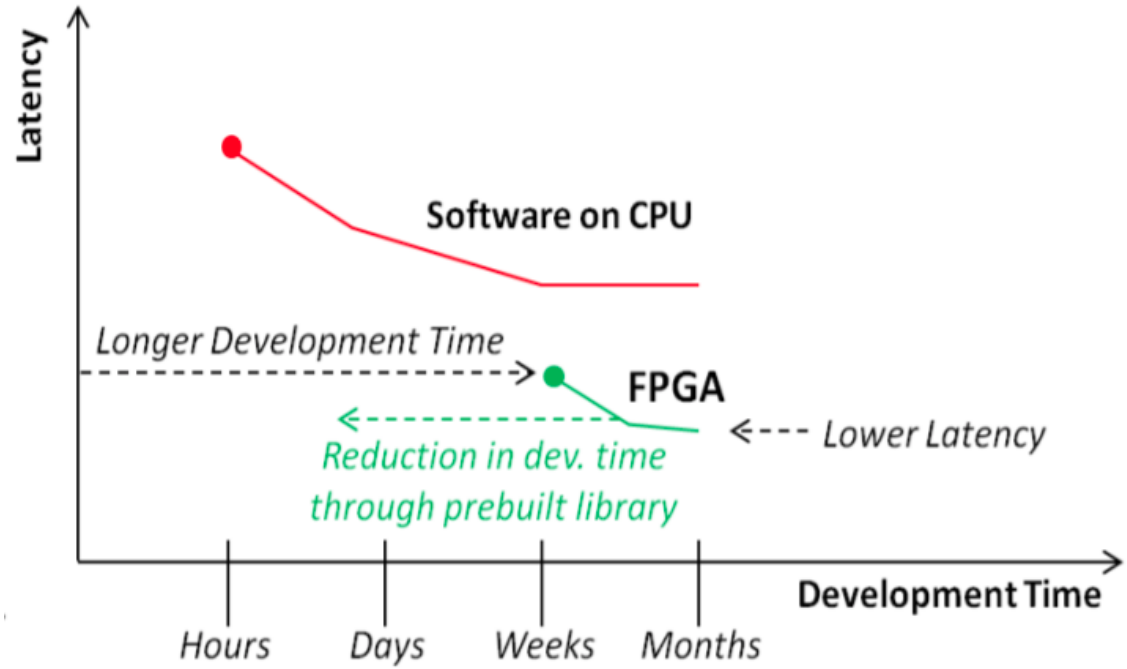
- The availability and the processing time for any piece of information is completely predictable down to a clock cycle
- Significantly lower latencies, high throughput with minimal jitter
- Flexibility: long-lived trading strategies



Pure FPGA computing

Disadvantages

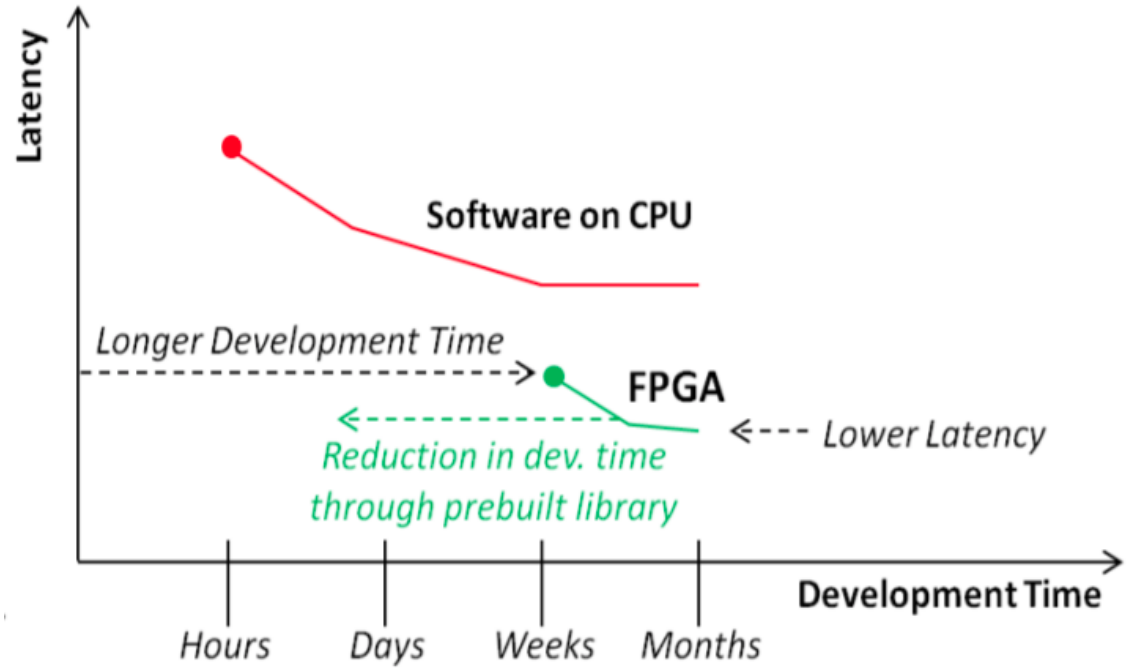
- Greater complexity of the development flow: building and verifying new hardware is more time consuming



Pure FPGA computing

Disadvantages

- Greater complexity of the development flow: building and verifying new hardware is more time consuming
- Many developers of financial applications are unfamiliar with FPGA technology and HDL



Contents

AL low latency library

Observation

The purpose of Algo-Logic team is to build a library of pre-built FPGA IP blocks for networking and financial protocol parsing. This allows end users to focus on their application without first constructing basic infrastructure

AL low latency library

Observation

The purpose of Algo-Logic team is to build a library of pre-built FPGA IP blocks for networking and financial protocol parsing. This allows end users to focus on their application without first constructing basic infrastructure

- This library is compatible with standard FPGA platforms. One supported platform is the NetFPGA-10G

AL low latency library

Observation

The purpose of Algo-Logic team is to build a library of pre-built FPGA IP blocks for networking and financial protocol parsing. This allows end users to focus on their application without first constructing basic infrastructure

- This library is compatible with standard FPGA platforms. One supported platform is the NetFPGA-10G
- Processes financial protocols extracting information from the packets as they flow through the FPGA

AL low latency library

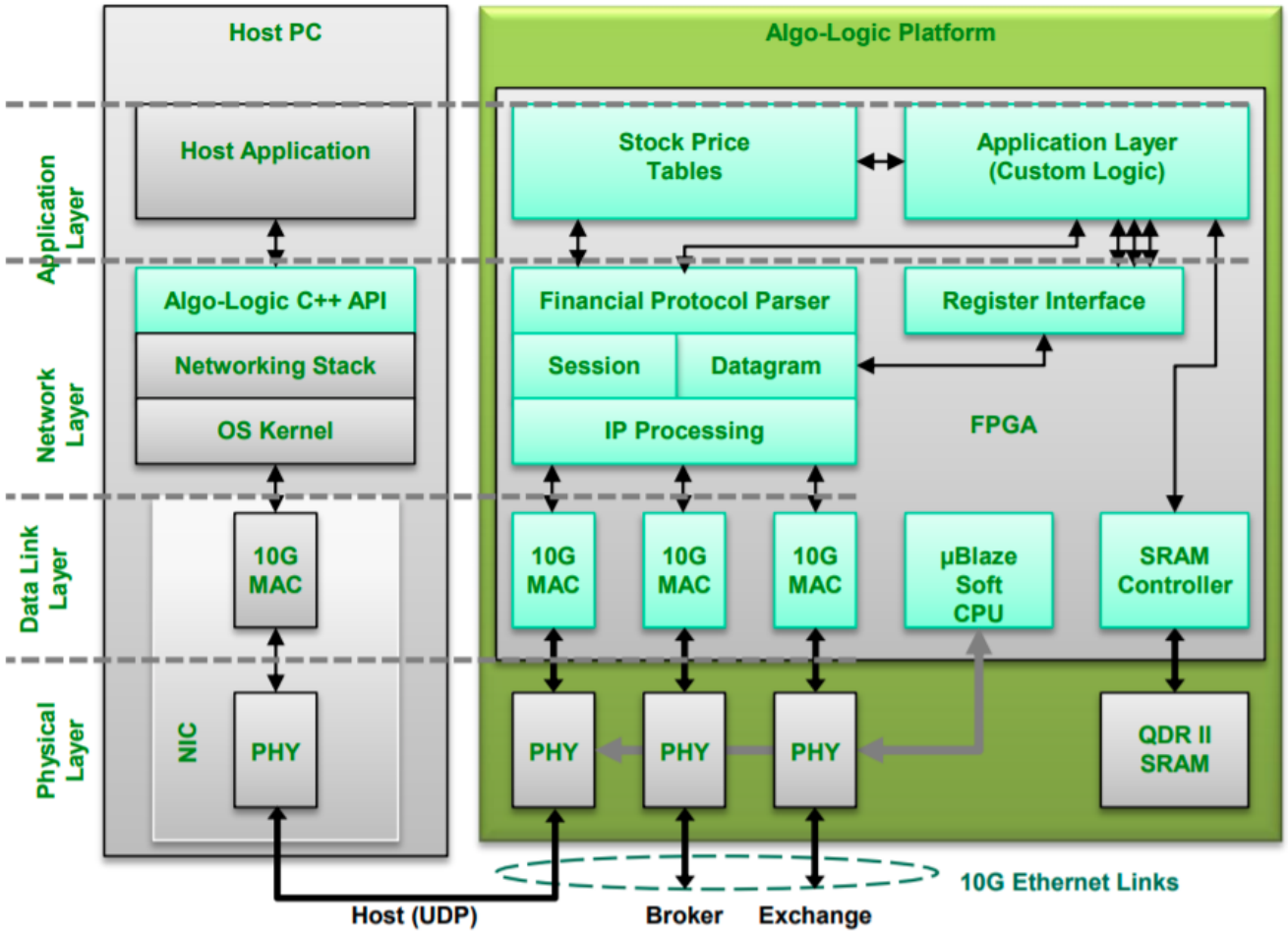
Observation

The purpose of Algo-Logic team is to build a library of pre-built FPGA IP blocks for networking and financial protocol parsing. This allows end users to focus on their application without first constructing basic infrastructure

- This library is compatible with standard FPGA platforms. One supported platform is the NetFPGA-10G
- Processes financial protocols extracting information from the packets as they flow through the FPGA
- The library components are used to construct custom trading applications, reducing time-to-market

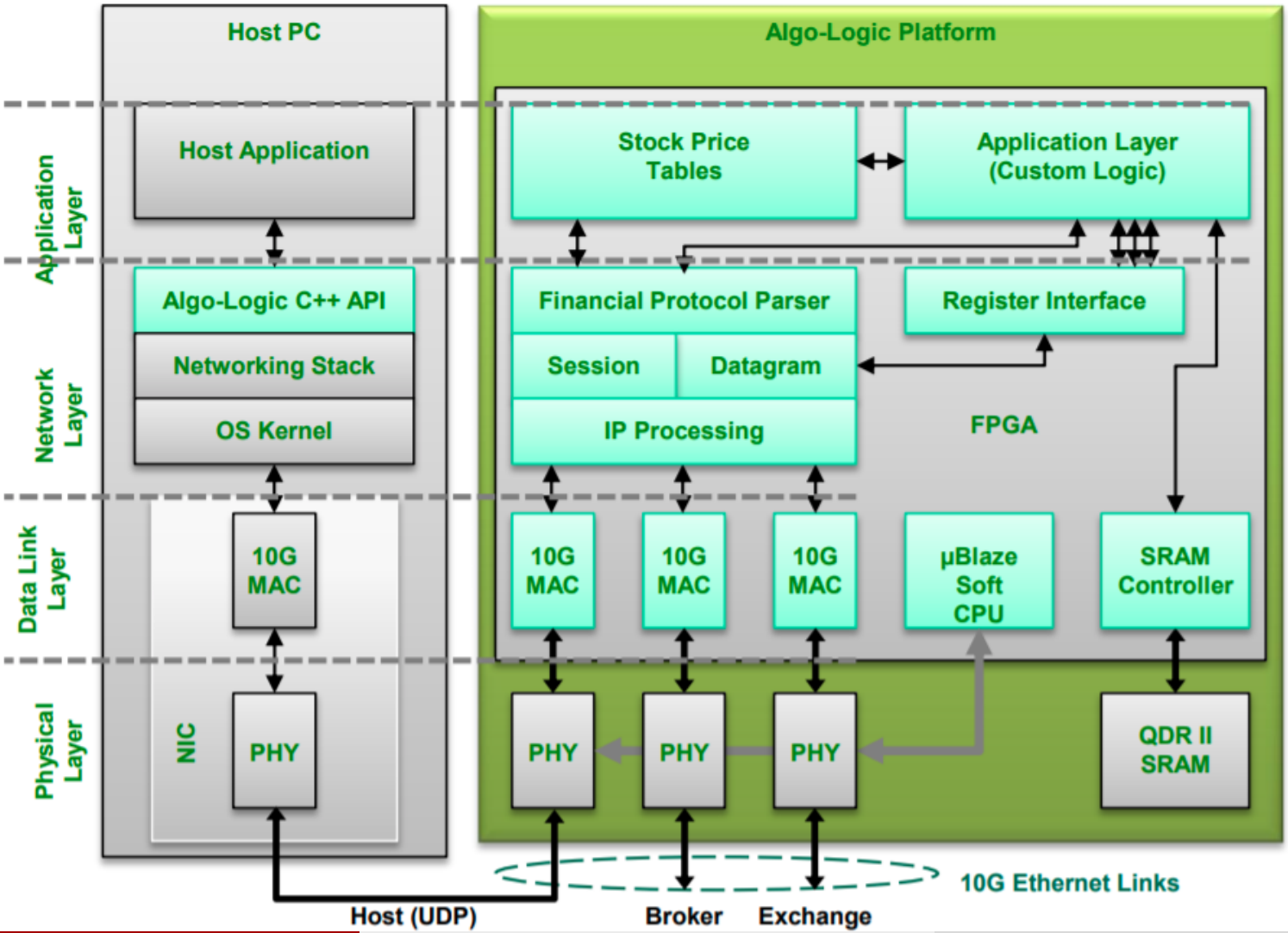
IP blocks categories

1. Infrastructure Components
2. Financial Processing Components



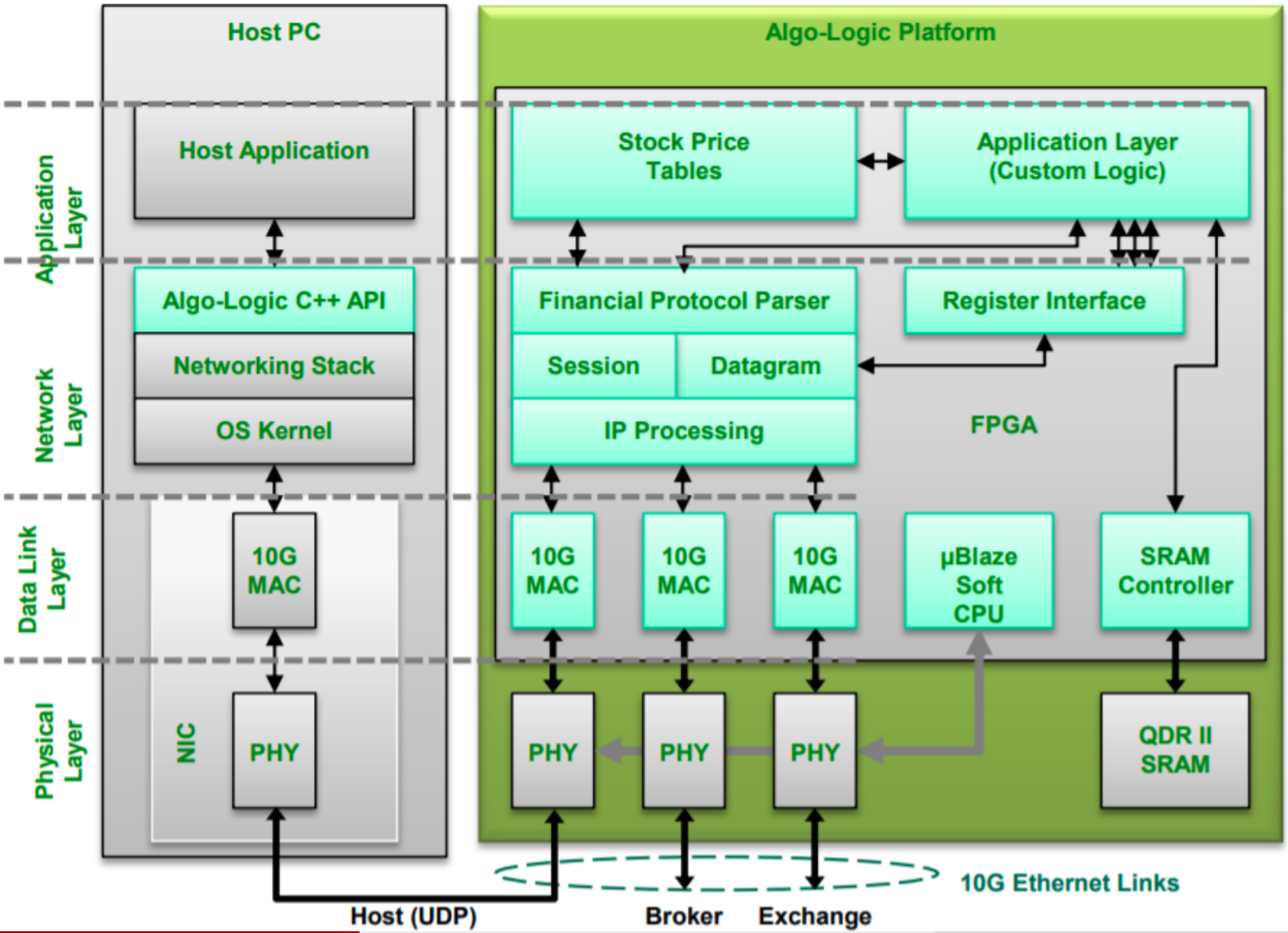
1-Infrastructure components

- All components share a common 64bit data word and AXI4 stream protocol



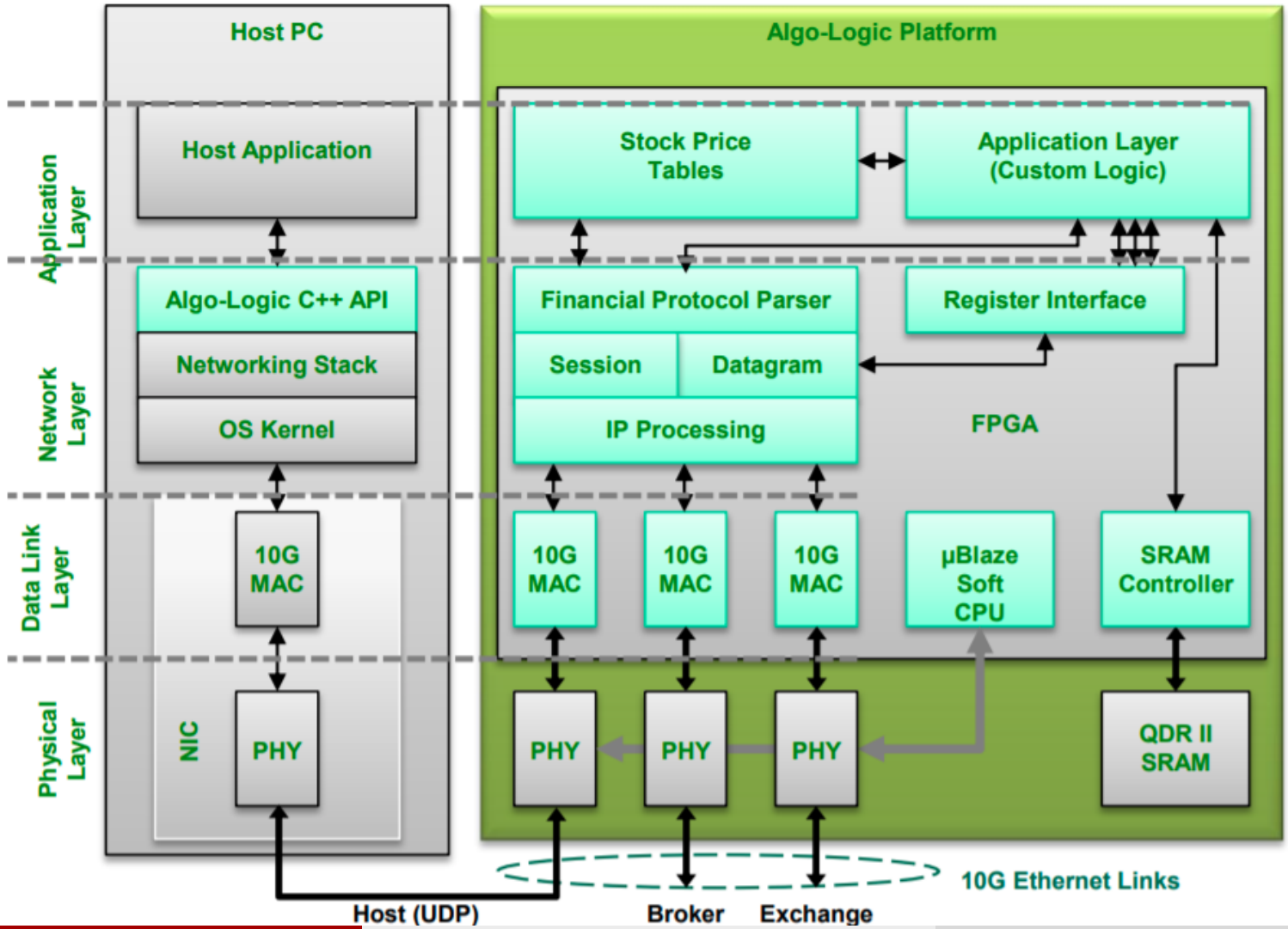
1-Infrastructure components

- A Register Interface module controls and monitors the status of registers written and read by host software



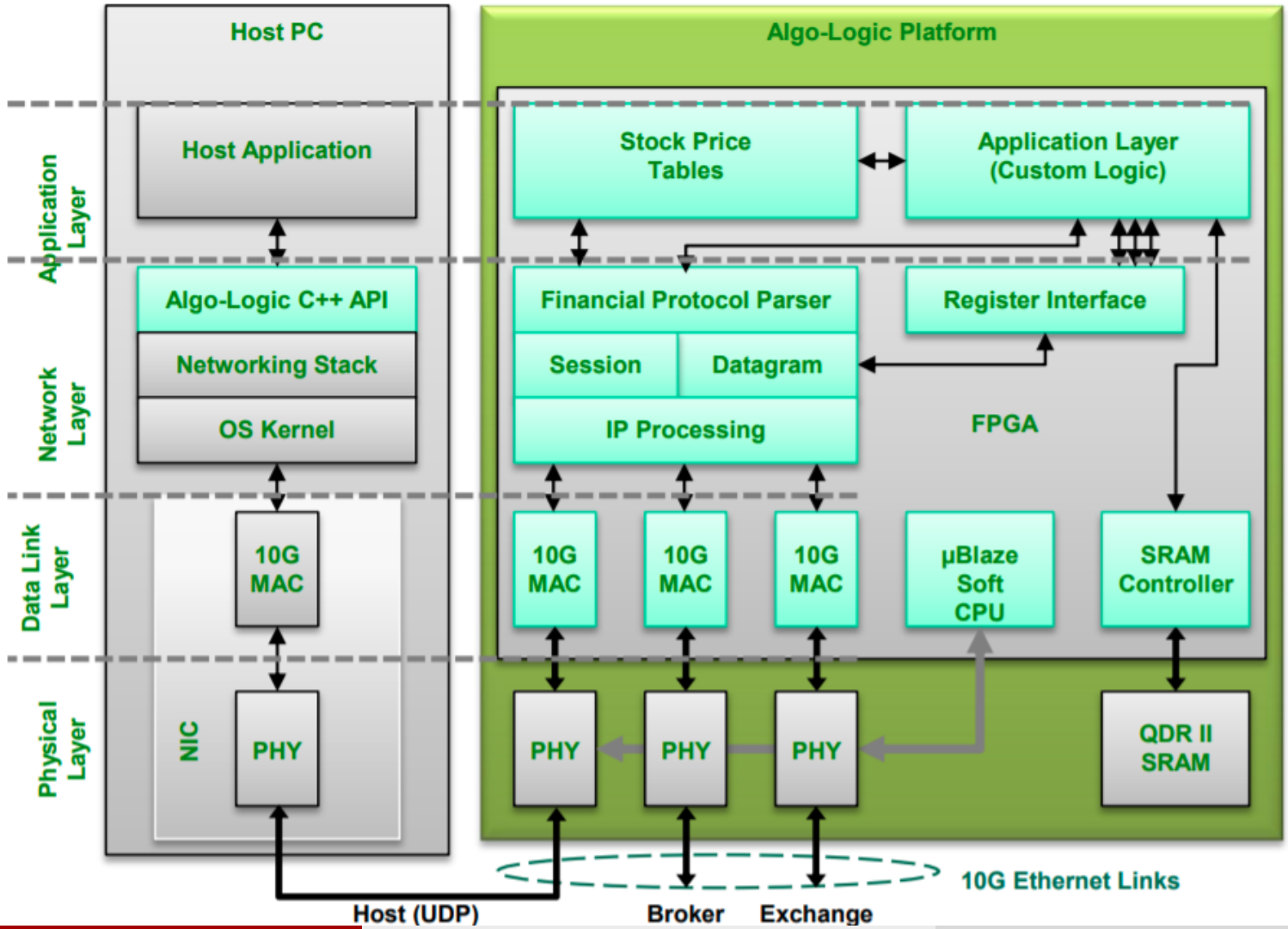
1-Infrastructure components

- TCP/IP separates headers from payloads(according to the financial protocol) and sends the payloads to the application layer



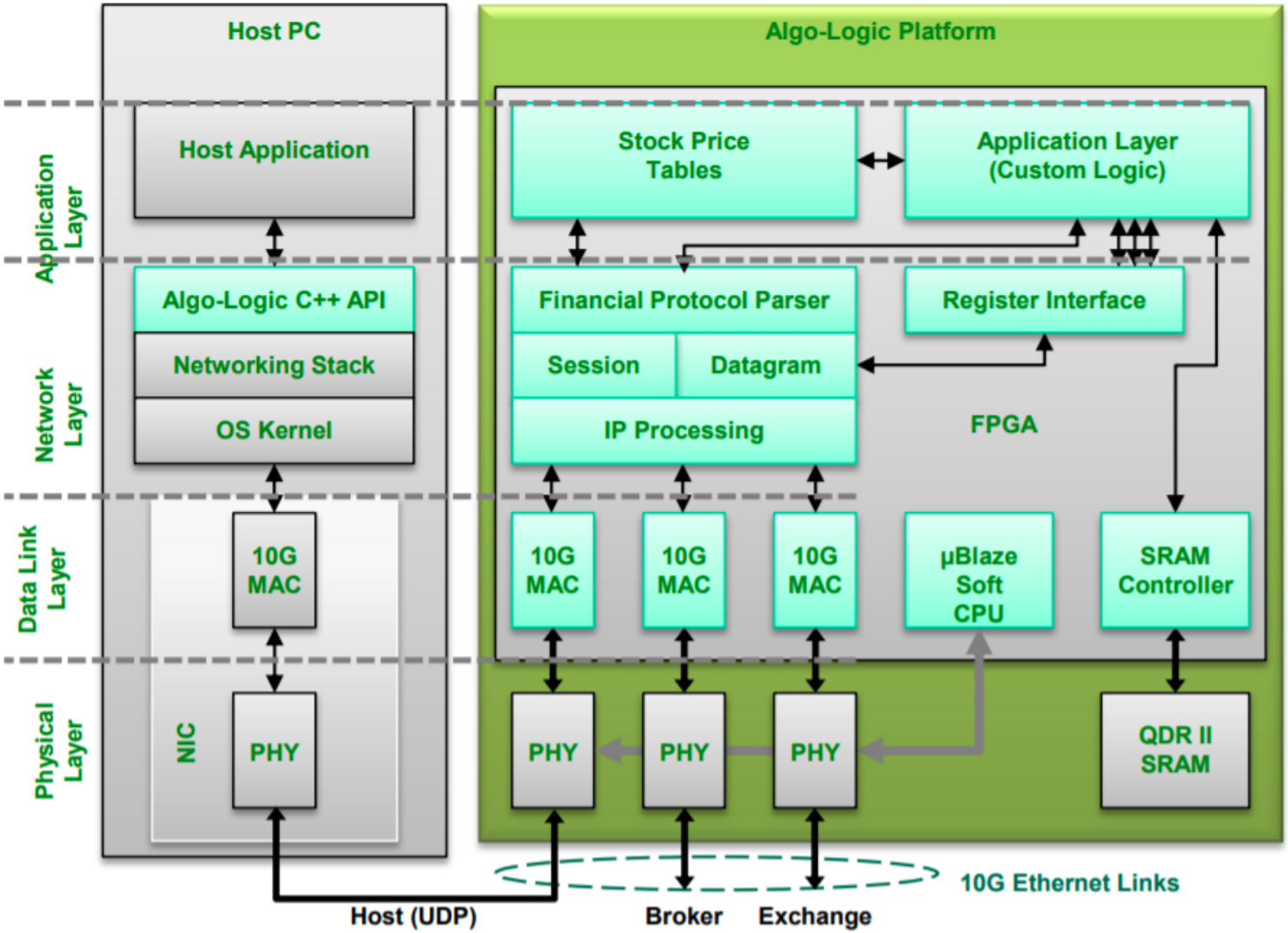
1-Infrastructure components

- Application layer processes packet payloads and sends them back to the TCP/IP layer



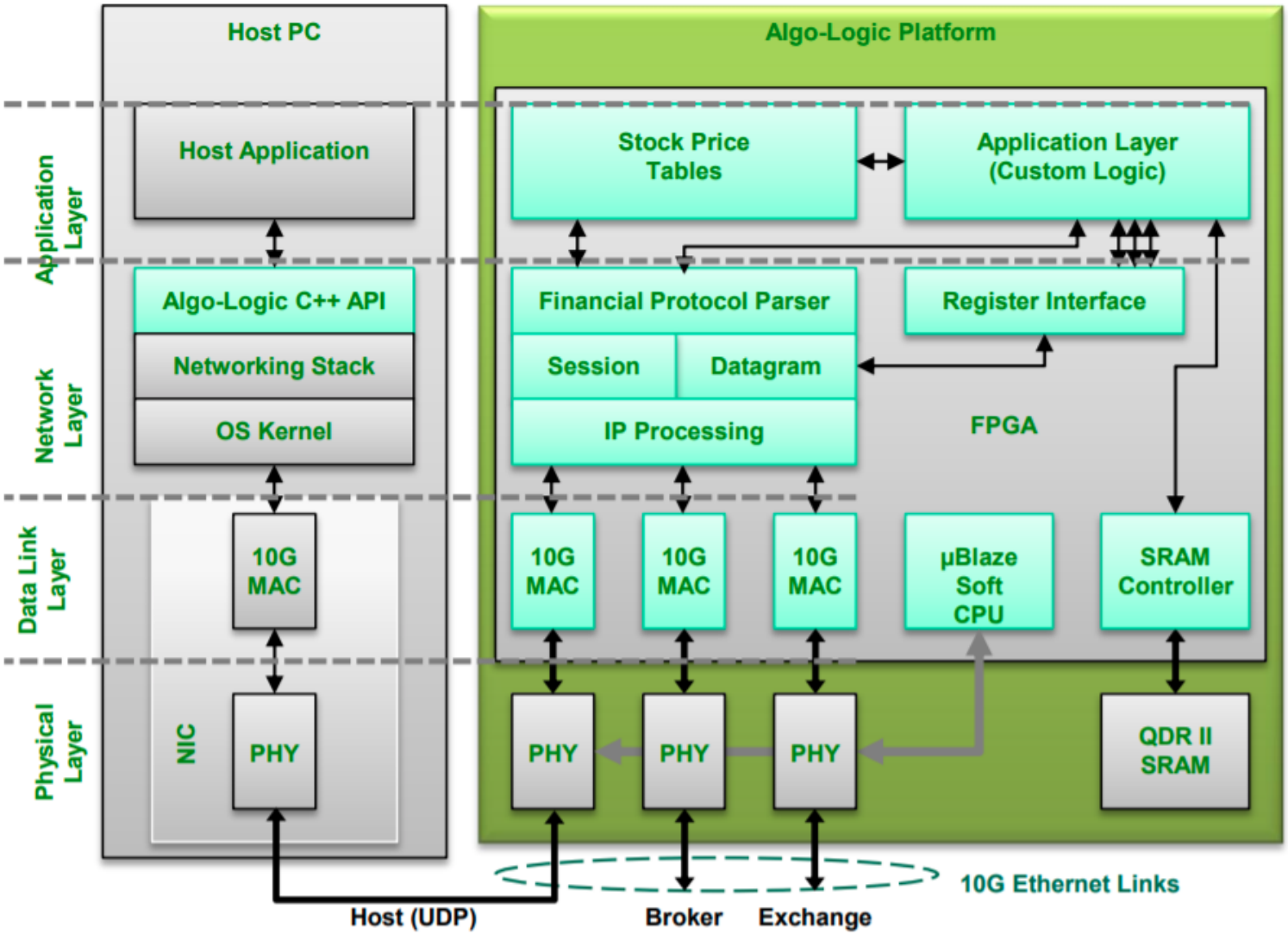
1-Infrastructure components

- TCP/IP reassembles headers and payloads and performs a checksum



1-Infrastructure components

- QDR (quad data rate) II SRAM is used in order to store prices of other stocks or historic data



2-Financial processing components

- Algo-Logics library includes IP blocks designed to process application layer messages for HFT.
- These messages (in form of packets) consist in a series of orders or regarding the order execution reports sent between clients, brokers and exchanges.

2-Financial processing components

- Algo-Logics library includes IP blocks designed to process application layer messages for HFT.
- These messages (in form of packets) consist in a series of orders or regarding the order execution reports sent between clients, brokers and exchanges.

There may be distinguished the following components:

1. Financial protocol parser
2. Market data parsing and on-chip storage for price data

Financial protocol parser

Definition

It receives a payload from the TCP/IP Processing Layer and identifies message boundaries for the data in the payload. The parser extract individual fields and raises a flag when one of them become valid.

Financial protocol parser

Definition

It receives a payload from the TCP/IP Processing Layer and identifies message boundaries for the data in the payload. The parser extract individual fields and raises a flag when one of them become valid.

Protocols

- **FIX** Financial Information eXchange
- **OUCH** NASDAQ
- **XPRS** DirectEdge
- **BOE** BATS BZX
- **ArcaDirect** NYSE Arca
- **Native Trading Gateway** LSE

Financial protocol parser, OUCH

- Preferred to TPC/IP

Financial protocol parser, OUCH

- Preferred to TPC/IP
- 8 bytes per clock cycle streaming through 64-bit AXI-Stream data path

Financial protocol parser, OUCH

- Preferred to TPC/IP
- 8 bytes per clock cycle streaming through 64-bit AXI-Stream data path
- As each field is extracted, the value is transferred to internal I/O pins and a valid flag is raised

Financial protocol parser, OUCH

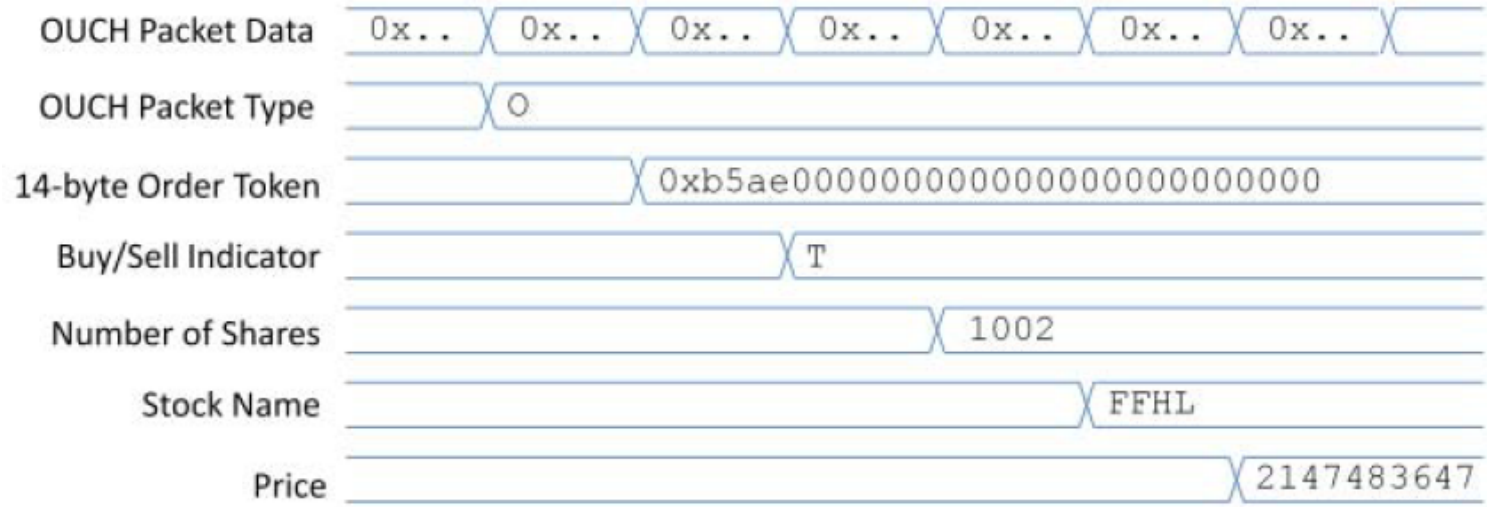
- Preferred to TPC/IP
- 8 bytes per clock cycle streaming through 64-bit AXI-Stream data path
- As each field is extracted, the value is transferred to internal I/O pins and a valid flag is raised
- the circuit extracts all OUCH packet fields within exactly 6.4 nanoseconds of the packets arrival with no jitter

Financial protocol parser, OUCH

- Preferred to TPC/IP
- 8 bytes per clock cycle streaming through 64-bit AXI-Stream data path
- As each field is extracted, the value is transferred to internal I/O pins and a valid flag is raised
- the circuit extracts all OUCH packet fields within exactly 6.4 nanoseconds of the packets arrival with no jitter

Financial protocol parser, OUCH

- Preferred to TPC/IP
- 8 bytes per clock cycle streaming through 64-bit AXI-Stream data path
- As each field is extracted, the value is transferred to internal I/O pins and a valid flag is raised
- the circuit extracts all OUCH packet fields within exactly 6.4 nanoseconds of the packets arrival with no jitter



Market data parsing and on-chip storage for price data

Observation

In financial processing applications need to know in real time the price of securities that appear in the orders.

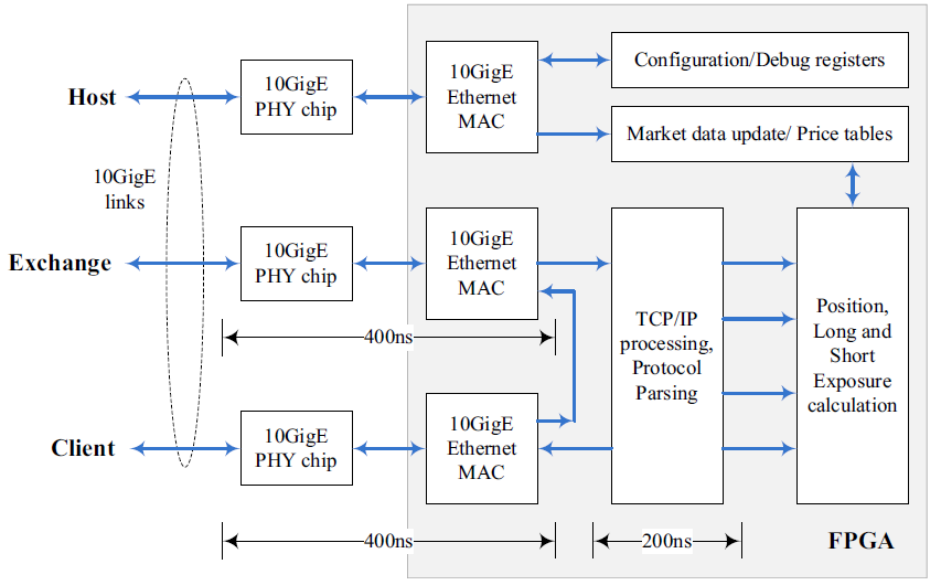
To track them:

- Market data is fed into the card via UDP/IP datagrams
- When they are exctracted they are stored on chip-in memory (8000 securities traded on U.S. Exchanges fit within the FPGAs on-chip memory)

FPGA implementation on NetFPGA-10G

Observation

AlgoLogics system acts as a bump in the wire between broker and exchange, processing the FIX execution reports coming from the exchange while passing on orders coming from the involved brokers. A third 10GE interface connects the appliance with the control and logging host.



Total wire-to-wire delay = 1µsec.
(400ns (PHY+MAC) + 200ns (Logic processing) + 400ns (PHY+MAC))

FPGA implementation on NetFPGA-10G

1. The design includes three instances of the 10GE MAC to interface to the host, the broker and the exchange

FPGA implementation on NetFPGA-10G

1. The design includes three instances of the 10GE MAC to interface to the host, the broker and the exchange
2. The host interacts via the configuration/debug register module and can access the market data update/price table module

FPGA implementation on NetFPGA-10G

1. The design includes three instances of the 10GE MAC to interface to the host, the broker and the exchange
2. The host interacts via the configuration/debug register module and can access the market data update/price table module
3. Current market prices for each security are received via UDP from the exchange and stored within on-chip memory

FPGA implementation on NetFPGA-10G

1. The design includes three instances of the 10GE MAC to interface to the host, the broker and the exchange
2. The host interacts via the configuration/debug register module and can access the market data update/price table module
3. Current market prices for each security are received via UDP from the exchange and stored within on-chip memory
4. Traffic from brokers and FIX execution reports from the exchange are passed from the MACs to the TCP/IP processing module

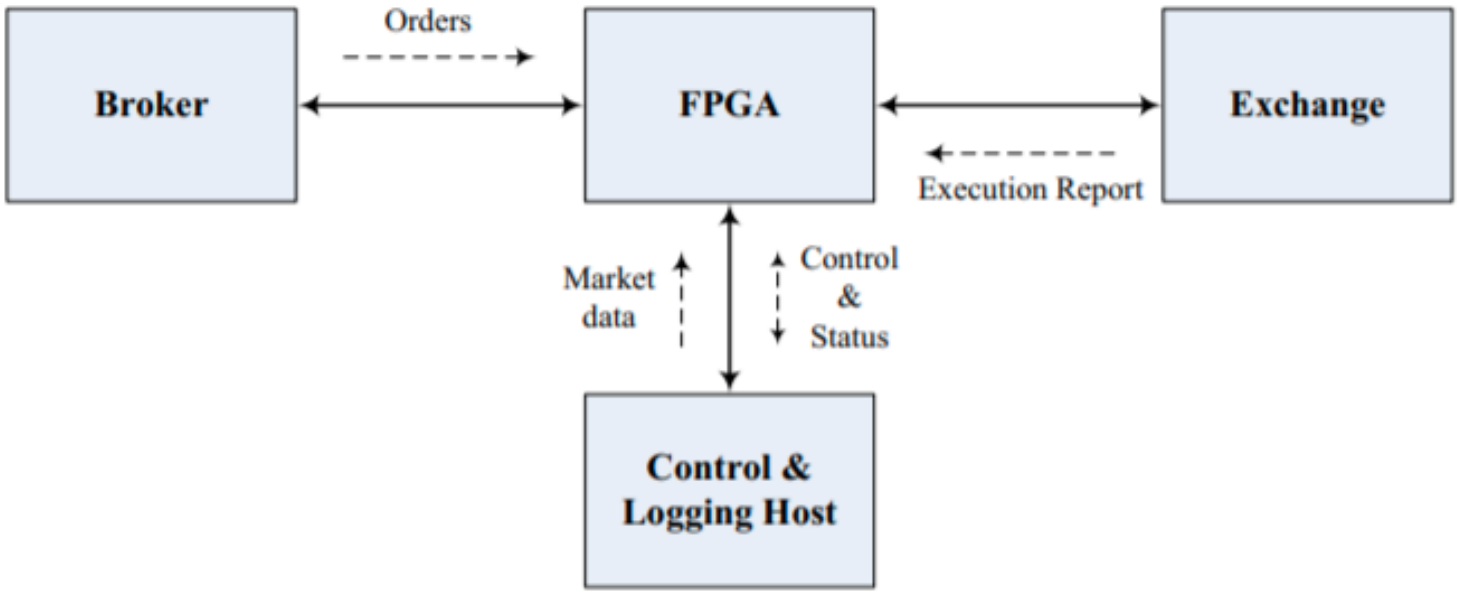
FPGA implementation on NetFPGA-10G

1. The design includes three instances of the 10GE MAC to interface to the host, the broker and the exchange
2. The host interacts via the configuration/debug register module and can access the market data update/price table module
3. Current market prices for each security are received via UDP from the exchange and stored within on-chip memory
4. Traffic from brokers and FIX execution reports from the exchange are passed from the MACs to the TCP/IP processing module
5. The Exposure and Position calculation module receives the traffic from the TCP/IP module and updates the market data/price tables

FPGA implementation on NetFPGA-10G

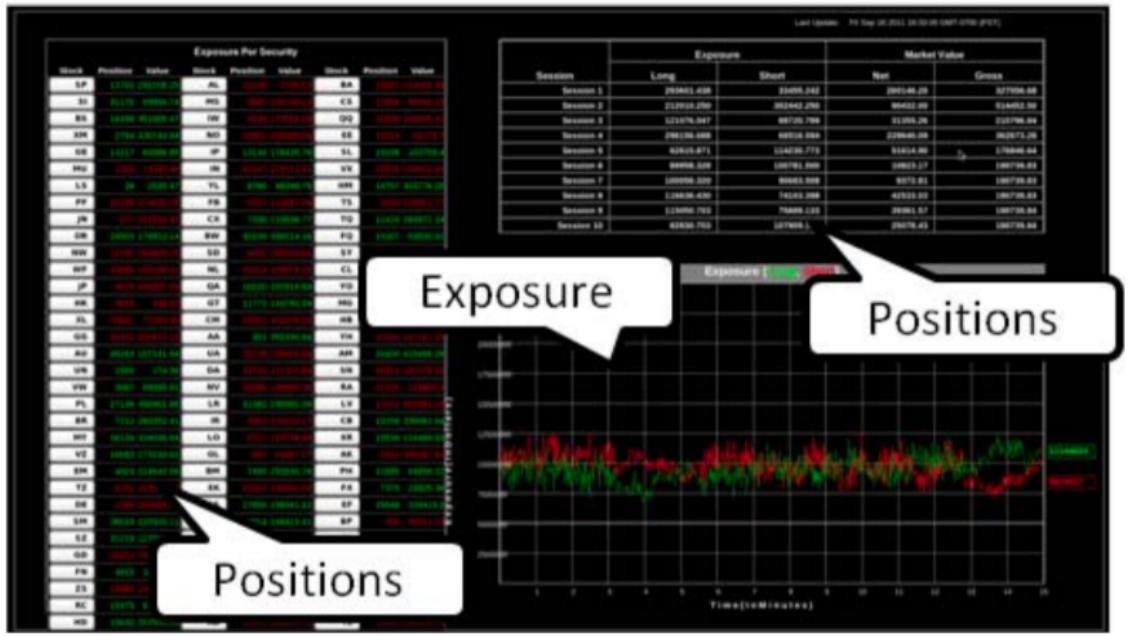
1. The design includes three instances of the 10GE MAC to interface to the host, the broker and the exchange
2. The host interacts via the configuration/debug register module and can access the market data update/price table module
3. Current market prices for each security are received via UDP from the exchange and stored within on-chip memory
4. Traffic from brokers and FIX execution reports from the exchange are passed from the MACs to the TCP/IP processing module
5. The Exposure and Position calculation module receives the traffic from the TCP/IP module and updates the market data/price tables
6. Finally, the FIX execution reports are forwarded to the clients providing details of order execution

Implementation block diagram



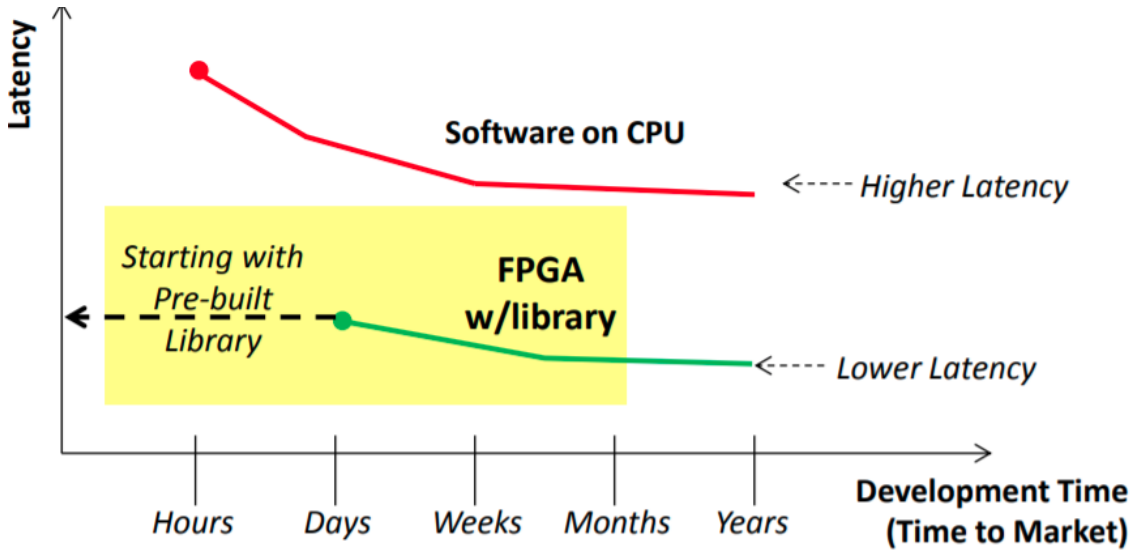
GUI

Web-based GUI showing Exposure and Position



Contents

Results



- The total wire-to-wire latency in this application is $1\mu s$ with liertally no jitter
- It is worth noting that the $1\mu s$ round-trip latency through this design is between one and two orders of magnitude lower than a recent software implementation
- The throughput is roughly $6.1M$ FIX messages/second

Bibliography I

Basic information about stock exchange. URL: https://www.investopedia.com/stock-trading-4689660.

John W.Lockwood Nishit Mehta, Michaela Blott. "A low-latency library in FPGA hardware for High Frequency Trading HFT". In: *2012 IEEE 20th Annual Symposium on High-Performance Interconnects* (2012). DOI: 10.1109/HOTI.2012.15.

What is OUCH protocol. URL: https://www.nasdaqtrader.com/content/technicalsupport/specifications/TradingProducts/ouch1.05.pdf.