Evolution of FPGA Trading Systems

John Lockwood
CEO, Algo-Logic

"Using FPGA for financial analytics:
Has the programmability nut been cracked?"

at the

STAC® SUMMIT

High technology in finance

New York / June 13, 2018

Agenda & registration: www.STACresearch.com/events

ALGORITHMS IN LOGIC

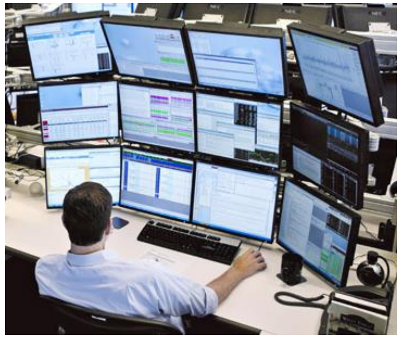
ALGO-LOGIC®

HTTP://ALGO-LOGIC.COM

Evolution of Trading Systems

- **From Specialists in the pit**
  - Trade on behalf of clients
  - Make markets by buying and selling
  - Trade in seconds to minutes (1 to 60 seconds)

- **To Software Trading**
  - Automated trading using software from the desktop
  - Trade in milliseconds to seconds (0.001 to 1 seconds)



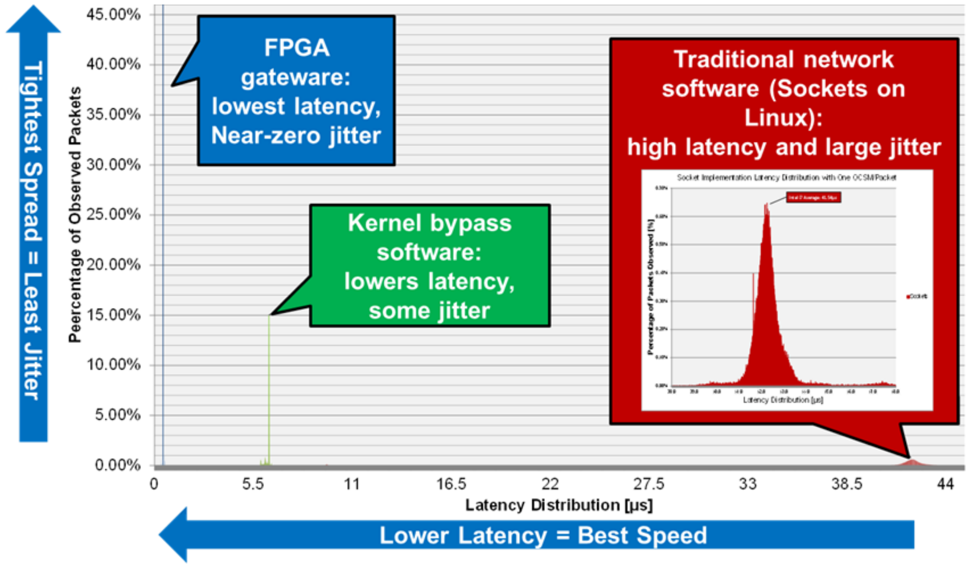
- **To Optimized Software**
  - Bypass the operating system kernel
  - Run in servers at market co-Location datacenters
  - Trade in microseconds to milliseconds (0.000001 to 0.001 seconds)

- **To FPGA-Accelerated Trading**
  - Implement algorithms in logic
  - Run on Field Programmable Gate Arrays (FPGA)
  - Trade in nanoseconds using logic (< 0.000001 seconds)

ALGO-LOGIC

FPGA Gateware vs. Software

– Gateware in FPGA is fast
  - Sub-microsecond Latency
– Gateware is deterministic
  - Near-zero jitter
  - Like hardware, unlike software



- **Firms with faster trading systems**
  – Use FPGAs for instant Tick-to-Trade
  – Instantly reacts to adverse situations
  – Win most of the HFT profits

- **Firms with slower trading systems**
  – Still rely on software to trade
  – Respond slowly to market changes
  – Lose alpha due to missed trades

From: Implementing Ultra Low Latency Data Center Services with Programmable Logic, *IEEE Micro* (Not a STAC Benchmark)

Evolution of FPGA Trading Systems

Conventional wisdom on building FPGA Trading

- Requires large staff of FPGA developers
- Takes years to get into the market (if at all)
- Difficult to program

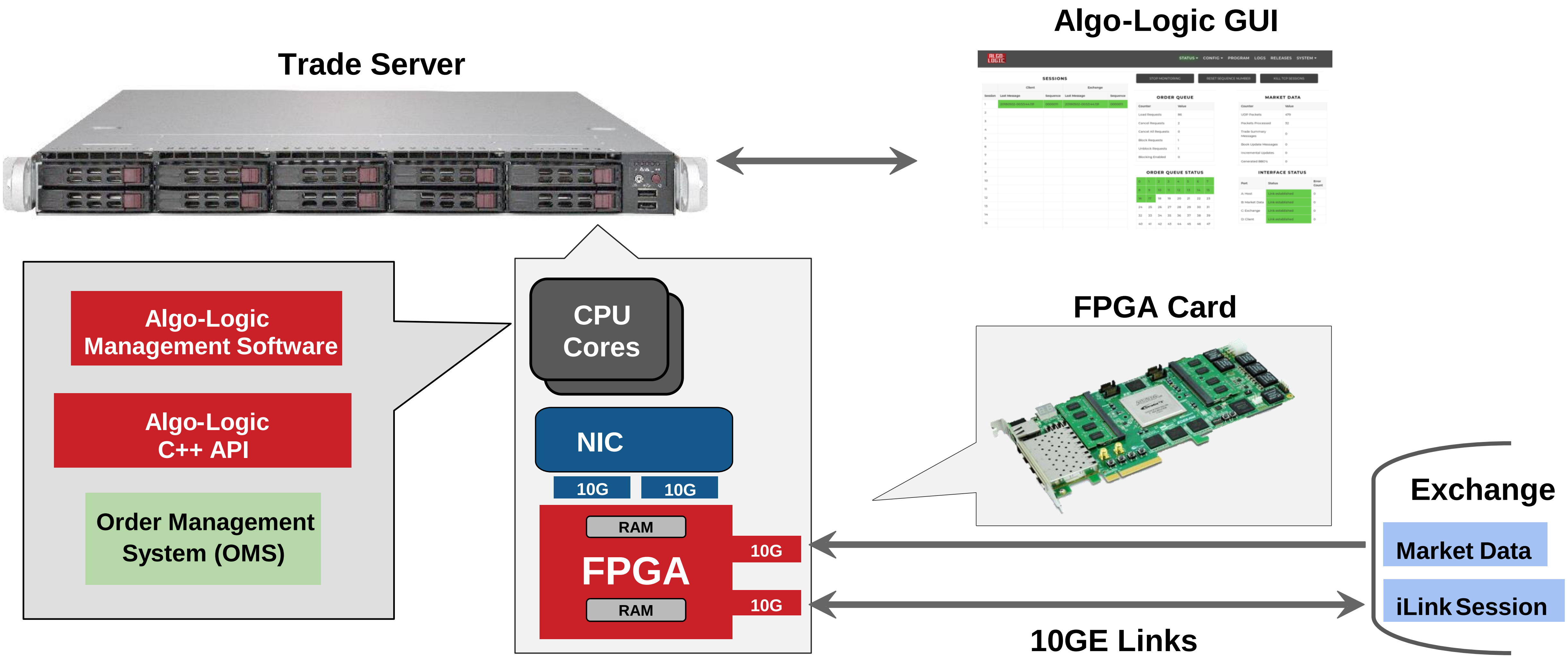
Leveraging pre-built FPGA Trading Systems

- Makes best use of staff
- Allows traders to come to market immediately
- Achieves best-in-class trading latency

Algo-Logic's Trading Systems

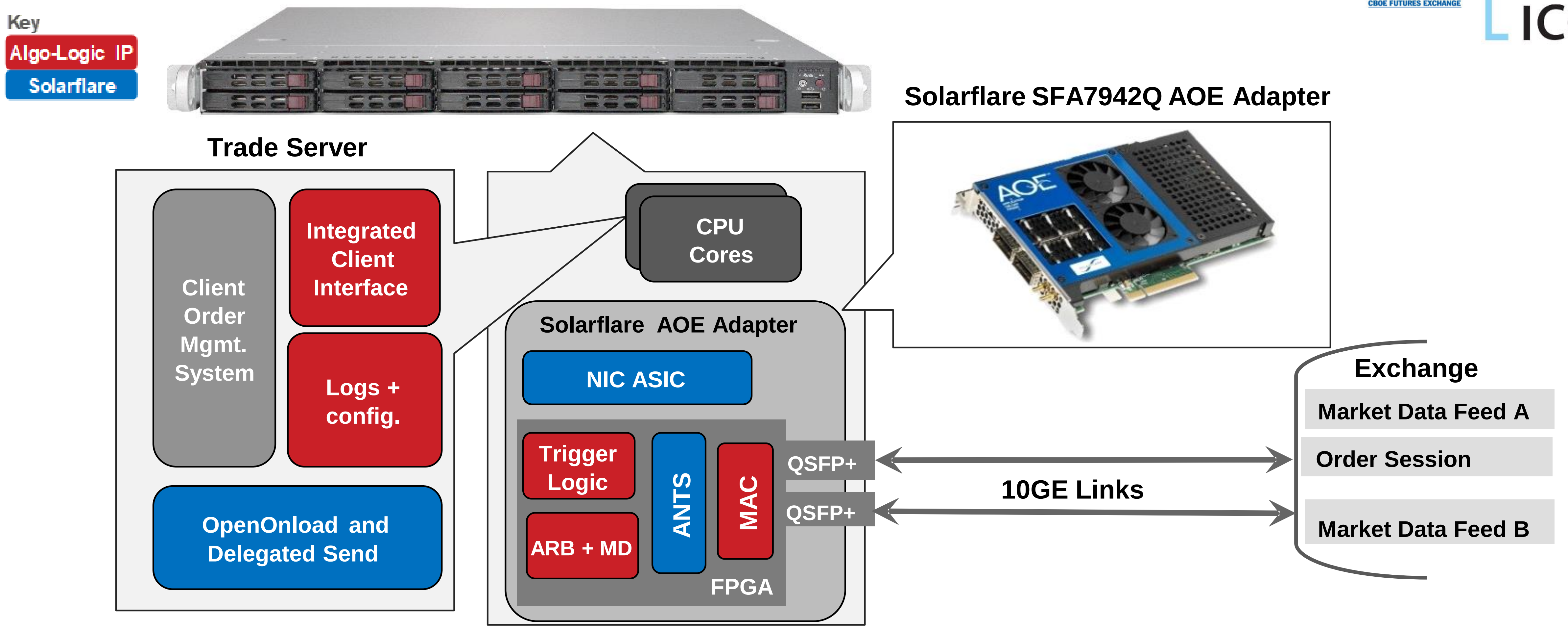
- Provides pre-built FPGA base systems for trading and risk control
- Developers interface with C/C++ Application Programming Interface (APIs)
- GUI provided for initial control and configuration

Algo-Logic's Tick-To-Trade System: Overview

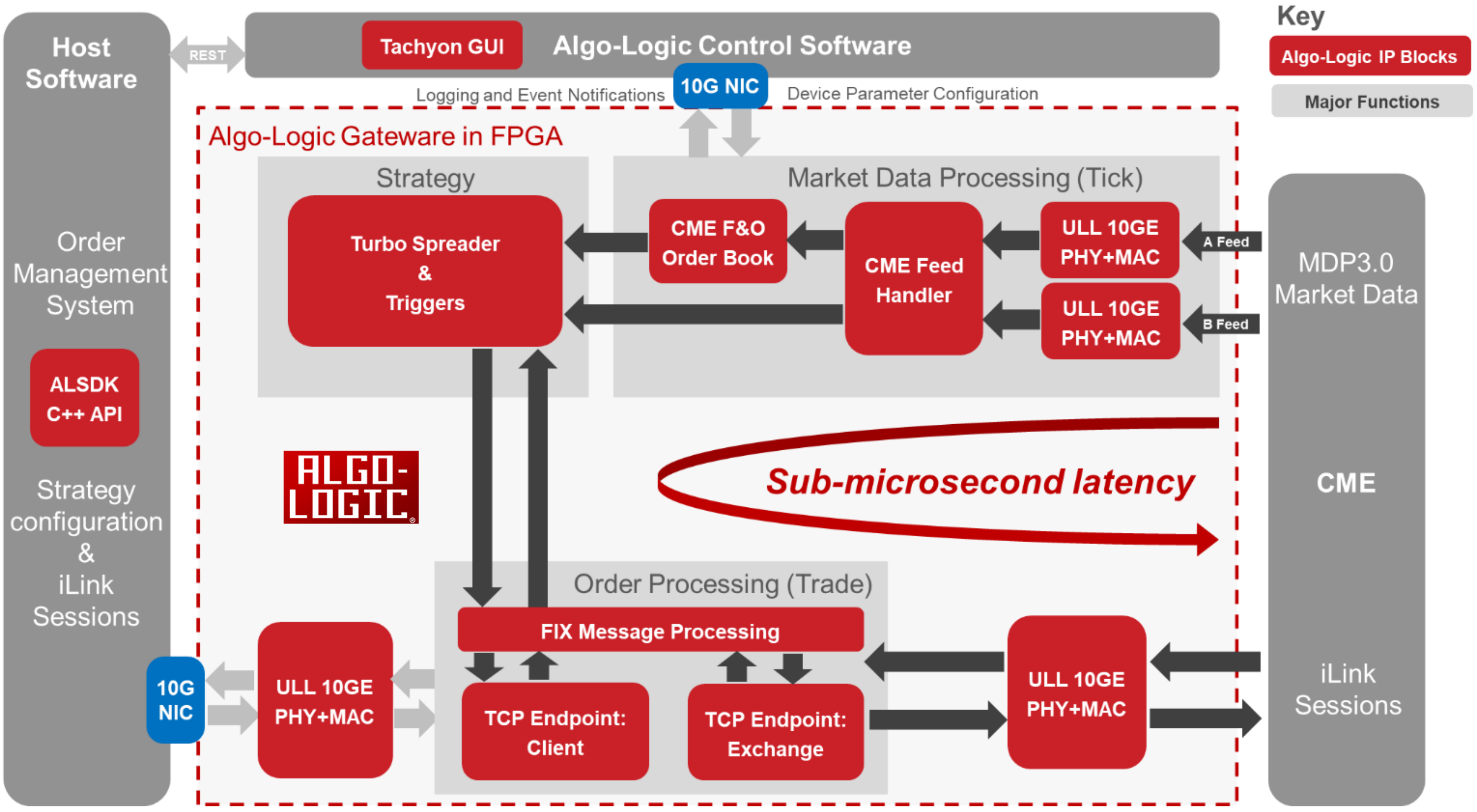


Algo-Logic's Trigger-Response Systems on AOE Platform

Available On: CME Group CFE ICE



Algo-Logic's Full Futures and Options Trading System in FPGA



Available for trading on all CME group markets (CBOT, COMEX & NYMEX). Supports futures trading of:

Example: Algo-Logic's API for Mass Quote Cancel in FPGA

Trigger parameters

```
arq::QuoteCancelTrigger trig;
// Take in input Trigger Parameters
trig.SetSecurityId(TriggerSecurityId);
trig.SetAllowableThreshold(AllowableThreshold);
```

FIX order to be injected

```
arq::OrderQueue queue(2048);  // maximum 2048 bytes
// Create FIX message
std::string fix_msg =
    "8=FIX.4.2\x01"
    "9=0157\x01"
    "35=Z\x01"
    "34=0000000\x01"
    "52=20160120-06:00:00\x01"
    "1028=N\x01"
    "298=3\x01"
    "49=SENDERCOMP\x01"
    "56=TARGETCOMP\x01"
    "57=TARSUB\x01"
    "142=US,IL\x01"
    "9771=ACCOUNT\x01"
    "50=UNIT_TEST_AUTO1\x01"
    "117=0000000BWO\x01"
    "295=1\x01"
    "55=ES\x01"
  "10=136\x01";
queue.AppendString(fix_msg);
```

Load trigger and FIX order in FPGA

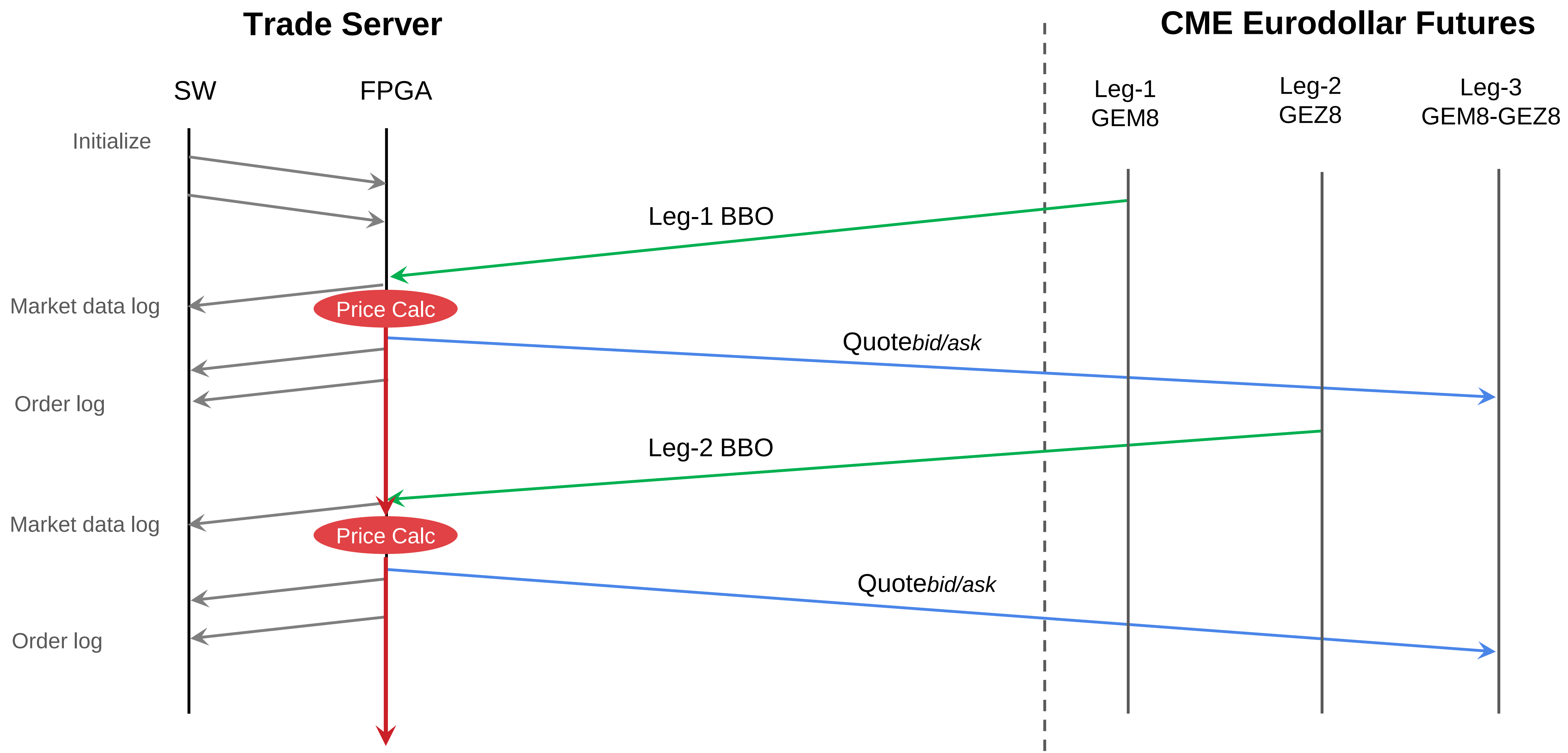
```
// Load Trigger and FIX Message
client.LoadQueue(qid, queue, trig, session);
```

Example: Algo-Logic's API to Accelerate Spreading in FPGA

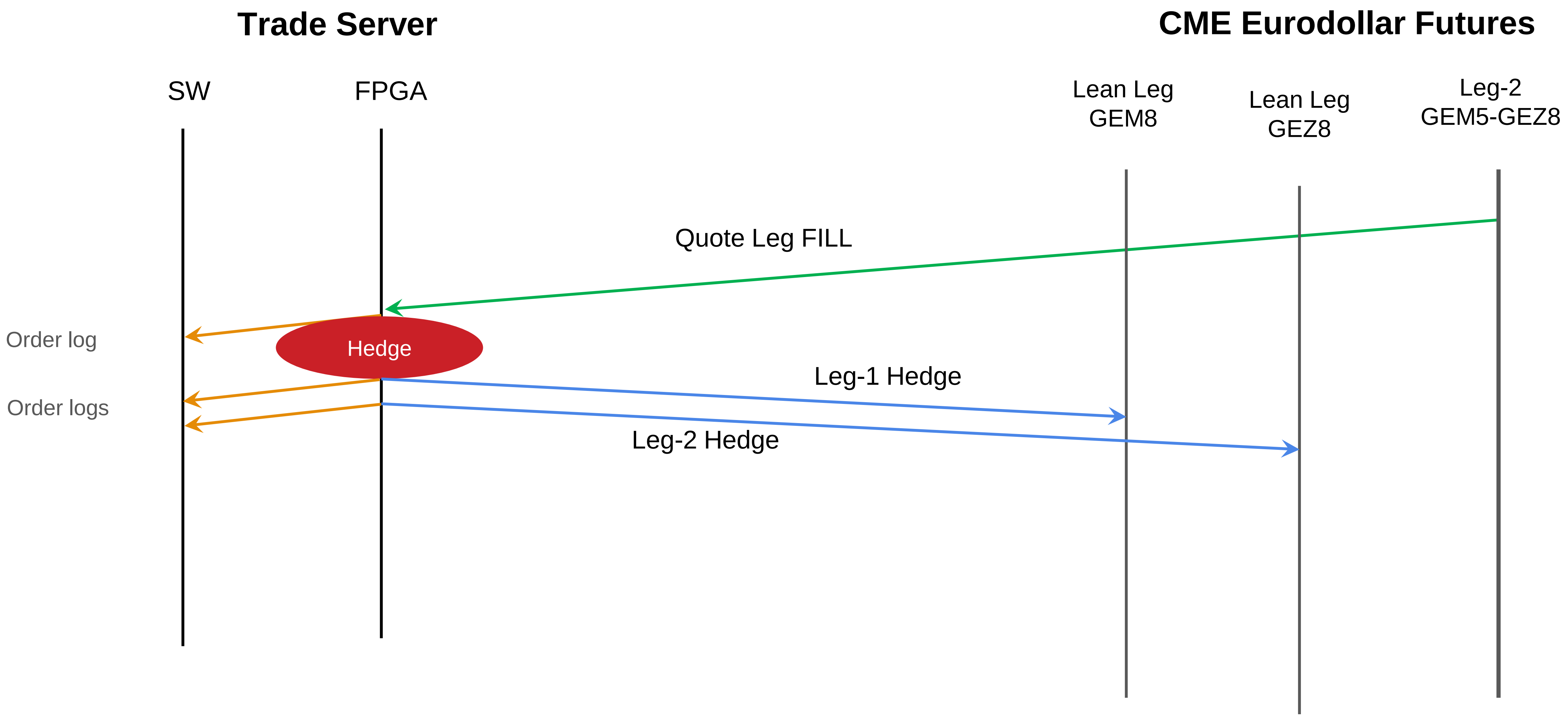
Legs of the spread	CME Eurodollar Futures
Lean leg	GEM8
Lean leg	GEZ8
Quote leg	GEM8-GEZ8

```
finance::Spreader spreader; // Create spreader object

spreader.LeanParam1(securityID, priceMultiplier); // 1st Lean Leg parameter
spreader.LeanParam2(securityID, priceMultiplier); // 2nd Lean parameter

spreader.setTickSize(TickSize); // Quoting parameter initialization
spreader.setBidEdge(BidEdge);
spreader.setAskEdge(AskEdge);

// Pre-load Quote Bid and Ask order queues
spreader.setBidQuoteOrder(bid.queue, client, bid.ClientOrderId, SessionId, quoteSecurityId);
spreader.setAskQuoteOrder(ask.queue, client, ask.ClientOrderId, SessionId, quote.SecurityId);
```

Turbo Spreader: Quoting a Calendar Spread from two Outright Lean Legs



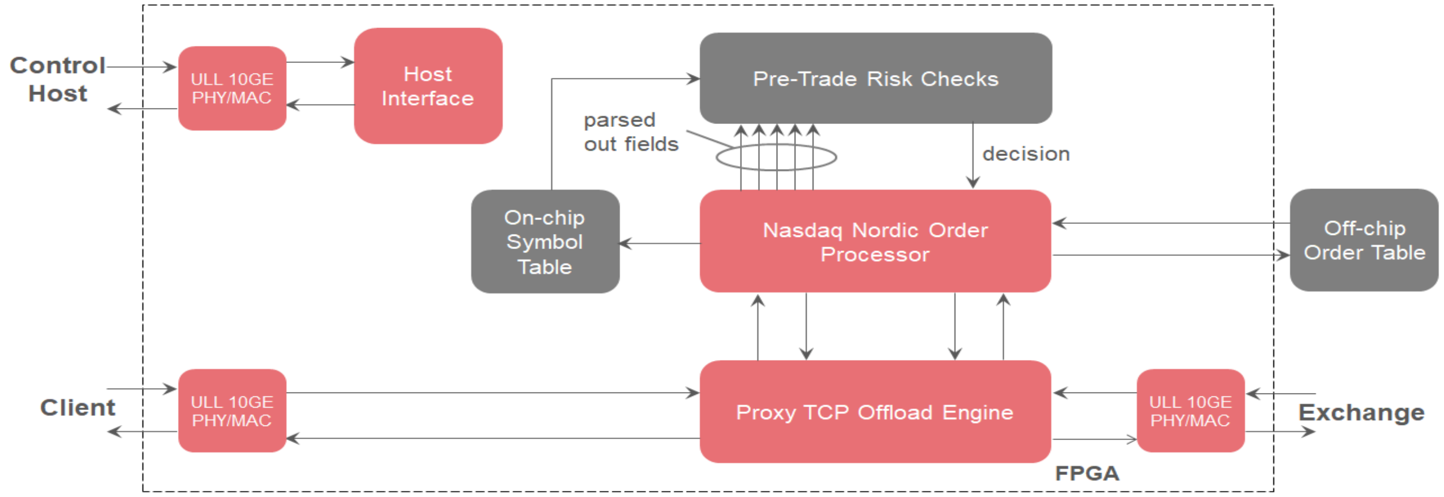
Hedging on a Filled Quote



Pre-Trade Risk Checks (PTRCs) in FPGA Logic

- **Algo-Logic's Original Ultra-Low-Latency PTRCs**
  - Deployed globally in equity markets
  - PTRCs coded by Algo-Logic in Verilog

- **Algo-Logic's new PTRC**
  - Includes IP cores for MAC, full-stack TCP proxy, order processors, and host interface in Logic
  - Allows brokers to specify PTRCs in RTL or with High-Level Synthesis (HLS)



Conclusions

- **Functionality of FPGA Trading has evolved**
  - Instant Trigger/Response with AOE
  - Parametrized order placement with trading in FPGA
  - Complex automated trade offload for spreading
  - Customer-provided RTL and HLS for PTRCs

- **Pre-built Trading systems come with software interfaces**
  - C/C++ API to set up to set up parameters of trades
  - Full GUI for control and configuration
  - Well defined interfaces to 3rd party modules

- **Algo-Logic's FPGA-Accelerated Trading Systems are available today**
  - Fast Trigger/Responses for AOE Card on multiple markets
  - Complete Tick-to-Trade in FPGA Logic for futures and options
  - Pre-trade Risk Checks (PTRCs) for multiple equity markets

info@algo-logic.com