Evolved Market Access
nxAccess

What is nxAccess?
nxAccess is a fully featured execution engine with an FPGA algo sandbox allowing users to **preload** orders in hardware, receive market data then **trigger & update** those orders then **send** to the trading venue – with the speed and performance of hardware assisted by the flexibility of software.

Designed for **market making**, **high performance trading** and **arbitrage** strategies, nxAccess accelerates existing trading algorithms without the investment usually required for **FPGA level performance**.

- **Speed**: driven by the latest Xilinx VUS+ FPGA, nxAccess can fully process & normalize raw market data while preloading thousands of orders within the FPGA’s memory – offering **sub 500ns** latency
- **Efficiency**: with exchange connectivity and market data normalization fully offloaded to the FPGA, engineers can focus on the development of your business logic – **improving your overall time-to-market**
- **Resiliency**: our reliable and **ultra-lean C++ API** allows for seamless integration with your existing software stack and enables software to handle any resynchronization or reconnection with the exchange with ease.
- **Flexibility**: typical ULL solutions are limited by the rigidity of hardware implementation, while nxAccess supports **all available market data message & order types** – accelerating even the most complex trading strategies
- **Power**: you will get more out of your server as our best-in-class FPGA is designed to off-load more processing while **reducing your footprint**

What makes nxAccess unique?
nxAccess uses a hybrid implementation with a full FPGA data path for latency critical logic and an accelerated software data path for more complex and non-latency sensitive logic. The nxAccess architecture includes three main blocks:

- **Hardware feed handler** - processes raw market data and distributes normalized updates to both the software and hardware business logic
- **Hardware execution engine** - manages session level variables (sequence numbers, timestamps) and stores orders preloaded by the software application which the hardware business logic can trigger. The hardware execution stack also allows the software trading logic to trigger or send orders through the same execution sessions.
- **Hardware sandbox** - hosts client business logic and offers low latency interfaces with the feed handler, the execution engine and the software application.

The nxAccess solution offers added flexibility by now supporting trading algorithms developed using **Higher Level Synthesis** (HLS) technologies, in addition to traditional hardware native languages such as **VHDL or Verilog** – making **ultra low latency trading strategies** more accessible to trading firms without in-house hardware expertise. Latency critical strategies can now be migrated to FPGA in **a matter of weeks**, leading to an improved hit rate and increased trading margins.
Product overview:
This block diagram highlights the two principal data paths of the nxAccess solution:

- **The FPGA data path** – orders are triggered within the FPGA from either a market data or order fill event
- **The Hybrid data path** – the software application tracks the orders on the market & can adjust the hardware parameters as market data prices evolve over time

In order to ensure reliability and best performance, nxAccess uses a primary and secondary interface for both market data and order execution.

The nxAccess execution engine is a hybrid implementation that enables software to manage the order execution protocol and FPGA logic to trigger and send orders to the exchange in the following sequence:

1. **Preload** Orders are preloaded by software into the execution engine buffers.
2. **Trigger** The hardware business logic uses market data or return order messages to trigger preloaded buffers and provides fields to update the order.
3. **Update** Once triggered, the selected buffer is updated with fields from the hardware business logic and mandatory fields previously initialized by the software.
4. **Send** The updated order is sent to the exchange.

Specifications:
- Supports up to 8000 symbols
- Symbols / Message type / Book depth filtering
- A/B feed arbitration by message
- Book-based and/or order-based market data updates
- Market data & order execution exchange resynchronization
- Full Hardware TCP Stacks
- 2 MBytes of in-FPGA order storage
  - Up to 16,000+ order buffers
  - Orders can be stored one-by-one or in bulk
  - Order MTU up to 1504 Bytes
- Kill switch with software heartbeat & bandwidth watchdog
- Aggregated Rx/Tx audit trail
- Monthly performance & EDC updates

Evolve Past Latency.

www.enyx.com/nxAccess