Systems for Algorithmic Trading in FPGA

Milan Dvorak
dvorak@invea.com
Petr Kastovsky
kastovsky@invea.com
• Czech university spin-off company
• FPGA team formed in 2000
• Company established in 2007
• 40+ employees
• Key focus
  ▪ Hardware acceleration
  ▪ Electronic trading
  ▪ FPGA solutions
  ▪ Network monitoring and security
Motivation

• **Evolution of trading**
  - Open out cry
  - Electronic
  - High frequency
  - Ultra low latency
  - First come first serve

• **High Performance Computing**
  - Derivatives trading
  - Monte Carlo simulations
FPGA technology

- **Programmable hardware**
  - Flexibility of software
  - Performance of hardware

- **Key features**
  - Field programmable
  - Massive parallelism
  - Deterministic
  - Low latency (sub us)

- **Drawbacks**
  - Hardware designer expertise required
  - Time to market longer compared to software
FPGA cards

- NIC (Network Interface Card) + FPGA chip
- Plugged in to a commodity box
- Provided as a whole solution
- 1G/10G/40G/100G interfaces
How can financial applications and trading systems benefit from the FPGA technology?
Content aware filtering

- Message Filtering
  - According to message type

- Symbol filtering
  - According to symbol ID

- Data Distribution
  - Multiple CPU cores, multiple network interfaces

- Data decoding&normalization
  - Unified message format

- Book handling
  - Convert order updates to price level updates

- Reduced CPU load on machines
Fast order execution

- Complex price computation in SW (Options pricing)
- Fast order execution engine in FPGA
  - SW feeds current desired prices to the FPGA
  - FPGA sends order as soon as matching price is on the book
- Complete tick-to-trade in hardware
- Wire-to-wire sub-microsecond latency
• Goal: evaluate all derivatives upon change in underlying price (stock → options)

• In SW – sequential task (few cpu cores)

• In FPGA – hundreds of symbols evaluated in parallel
FPGA can be used to accelerate generic computation

Monte Carlo simulation:

1. Load desired function to the FPGA
2. Let FPGA evaluate hundreds of points in parallel
3. Get the results

Parallel execution – faster than CPU

Lower power consumption
How to program the FPGA with user-defined function?
FPGA programming

• SW code – HLL (High Level Language)
  ▪ C/C++, Java, Python ...
  ▪ Flexible, easy to use
  ▪ 😊

• HW code – HDL (Hardware Description Language)
  ▪ VHDL, Verilog
  ▪ Difficult to learn, longer time-to-market
  ▪ 😞
• **New ways to work with FPGAs**

• HLS – High Level synthesis
  - Use HLL to write code for FPGA
  - Convert C/C++ to VHDL
  - Several commercial tools

• MathWorks tool chain
  - Convert your Matlab function to FPGA design
  - Simulink, Fixed-Point Designer, HDL Coder and HDL Verifier

• Complete solution with FPGA
  - FPGA is abstracted from the user
  - SW API calls to configure&use the FPGA
TradeCOPE

- All-FPGA trading solution
  - Lowest latency possible
  - Easy-to-use
- Deliverables
  - FPGA card, box
  - Software API, firmware framework
- Complete tick-to-trade processing
- User defined trading strategy
  - MathWorks tools or HLS
Summary

• FPGA
  ▪ Flexible, high performance, deterministic low latency

• Use cases of FPGA + algorithmic trading
  ▪ Content aware filtering
  ▪ Low-latency order execution
  ▪ Highly parallel symbol evaluation (options)
  ▪ Computation acceleration (Monte Carlo)

• FPGA programmability
  ▪ HLS – High level synthesis (C/C++ → FPGA)
  ▪ MathWorks (Matlab → FPGA)
  ▪ Whole solution (TradeCOPE)
INVEA-TECH a.s.
U Vodárny 2965/2
616 00 Brno, Czech Republic
www.invea.com

High-Speed Networking Technology Partner

Milan Dvořák
dvorak@invea.com