# I feel the need for speed: Exploiting latest generation FPGAs in providing new capabilities for high frequency trading

# I feel the need for speed: Exploiting latest generation FPGAs in providing new capabilities for high frequency trading

Mark Klaisoongnoen
EPCC at the University of Edinburgh
Edinburgh, United Kingdom
mark.klaisoongnoen@ed.ac.uk

Nick Brown
EPCC at the University of Edinburgh
Edinburgh, United Kingdom

Oliver Brown
EPCC at the University of Edinburgh
Edinburgh, United Kingdom

**ACM Reference Format:**
Mark Klaisoongnoen, Nick Brown, and Oliver Brown. 2021. I feel the need for speed: Exploiting latest generation FPGAs in providing new capabilities for high frequency trading. In *International Symposium on Highly Efficient Accelerators and Reconfigurable Technologies (HEART '21), June 21–23, 2021, Online, Germany.* ACM, New York, NY, USA, 2 pages. https://doi.org/10.1145/3468044.3468059

## 1 PROJECT VISION

Field Programmable Gate Arrays (FPGAs) have enjoyed significant popularity in financial algorithmic trading [5][2]. Such systems typically involve high velocity data, for instance arriving from markets, streaming through FPGAs which then undertake real-time transformations to deliver insights within tight time constraints. Such high bandwidth, low latency data processing approaches have proven highly successful in delivering important insights to financial trading floors.

However due to the real-time nature requirements there is only a small window in which such data manipulations can occur. Therefore these transformations are, by necessity, fairly simplistic as there is not time for more advanced workloads. However, the past few years have seen very significant improvements in both the hardware and software eco-system for FPGAs which is potentially a game changer in this regard. New, more advanced hardware technologies such as Xilinx's Alveo and Intel's Stratix range, provide far more capability than ever before, and with exciting developments such as the AI engines in Xilinx's Versal ACAP due for release later this year, open up significant possibilities. Furthermore, the investment in the software ecosystem not only improving the programmability of these devices [1] but also the growth of open source libraries [7], potentially significantly reduces programming time and enables the development of more complex codes.

Exploiting the concurrency within FPGAs requires one to rethink CPU-optimized codes to a dataflow style of computing. To target low latency in real-time settings, the fundamental question which this PhD research is looking to answer is, given the aforementioned advances in FPGAs, *what algorithmic techniques are most appropriate to enable a step change in capability in high frequency algorithmic trading through HLS?* This research will involve the development of new dataflow techniques enabling high-throughput and low latency processing of streaming network data on next-generation FPGAs, ultimately enabling an increase in the processing work that can be completed in an *acceptable* timeframe window.


*HEART '21, June 21–23, 2021, Online, Germany*

ACM ISBN 978-1-4503-8549-7/21/06.
https://doi.org/10.1145/3468044.3468059

## 2 WORKING WITH STAC RESEARCH

The Securities Technology Analysis Center (STAC) oversee and manage a world leading benchmark suite which comprises membership of over 350 financial institutions and more than 50 vendors. This diverse set of members, from the largest global banks, to hedge funds, to hardware companies, all benefit from the industry standard financial benchmarks that STAC provides. Such companies can then explore, test against, and optimise their technologies for these codes.

It is a fantastic opportunity to be able to work with STAC during this PhD project, and this will greatly benefit the research with the ultimate aim being to enable further understanding of the dataflow techniques required to exploit next-generation FPGAs for high frequency trading. Working with industry standard benchmark codes written in a mixture of C, C++ and Python, and associated specifications means that our FPGA-based research will be applicable to these real-world problems, and having access to the STAC community also increases our ability to disseminate research findings and gain feedback on research results from experts in the field of finance.

However it should be highlighted that we are using the benchmarks in a different way than STAC members themselves. Unlike the STAC members who adhere to the strict benchmarking guidelines to perform official and highly trustworthy audits of their hardware, software or techniques, we are leveraging components of the benchmarks in a more flexible manner. Focused primarily at the dataflow algorithmic level, the flexibility that research provides means we can focus on components that will closely suit the FPGA, or are especially interesting to answer our specific research questions. The idea is that such research findings can then be fed back and made freely available to the other STAC members who could then adopt these in their engineered solutions. In the short to medium term we are concentrating on two of the STAC benchmarks, STAC-A2™ and STAC-A3™, which are likely of most immediate interest to our project vision.

## 3 STAC-A2: RISK COMPUTATION

The STAC-A2 benchmark [4] focuses on the computation of market risk sensitivities. Market risk describes price movements in a market and their impact on the value of an investor's position in holding a financial instrument. In the bigger picture, financial derivatives such as options derive their value from that of an underlying asset, and the calculated market risk measures how much the option's

price will change given a change in the price of the underlying asset. Thus, the understanding of market risk and its management plays a central role for investors, traders and regulatory institutions in today's financial markets.

The STAC-A2 benchmark specification involves the pricing of options of varying problem sizes. It includes simulation via the Heston stochastic volatility model [3], and the Longstaff and Schwartz [6] model. Figure 1 illustrates the top level call graph of this benchmark and the percentage of time spent directly by each function, or through child functions, for the generic x86 implementation in C++, unoptimized for specific hardware. It can be seen that each of the seven variations of the benchmarks all call the *scenario* function with different data, and this function and its child functions account for over 97% of the code runtime.

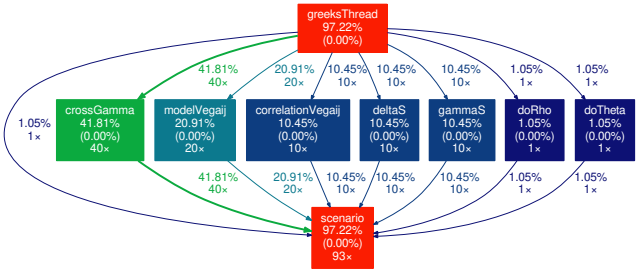


**Figure 1: STAC-A2: Partial call graph leaf with intermediate functions relying on the *scenario* function**

## 3.1 STAC-A2 on the FPGA

We have currently focused on porting the *scenario* function onto an Alveo U280 FPGA to accelerate this high workload by leveraging the massive amounts of concurrency that the FPGA provides. The current state is a fully working accelerated *scenario* function in HLS using Vitis which has been ported to the dataflow algorithmic style. The performance[1] of this function on a Xilinx Alveo U280 is illustrated in Table 1 against 26 cores of an Intel Xeon Platinum Skylake 8170 CPU. It can be seen that there is significant performance difference on the FPGA between the initial and optimised dataflow version, which involved techniques including reordering loops and caching results data in BRAM to best suit the architecture and keep all consistent parts of the kernel concurrently active and streaming data between them. Ultimately we would like to understand whether such risk analysis can be undertaken on streaming data in real-time in the high frequency context.

| Description | Runtime (ms) |
|---|---|
| CPU (26 cores of Skylake 8170) | 36.34 |
| Initial FPGA (Xilinx Alveo U280) | 1655.85 |
| Optimised FPGA (Xilinx Alveo U280) | 5.67 |

**Table 1: Performance comparison for STAC-A2 *scenario* function on the CPU and FPGA**

[1]The experiments conducted have not been designed to comply with official STAC benchmarking rules and regulations. Therefore the experimental results that we present are of a research nature and are not representative of official STAC audits.

# 4 STAC-A3: BACKTESTING

The STAC-A3 benchmark targets backtesting of trading strategies. A trading strategy generally covers the specification of a financial instrument which is to be traded, the time of the trade, the number of instruments, and the target market. Historical data provides a wealth of information about the likely profitability of trading strategies, and as such it is desirable to be used in evaluating the efficacy of new trading strategies before they are used in production. This is known as backtesting and involves a high workload.

Due to the high computational workload such investigations must currently be undertaken a priori, and it is our hypothesis that FPGAs could enable a more real-time approach, running backtesting automatically in response to market conditions to provide the traders with a bespoke prediction of how their trading strategies might perform. The STAC-A3 benchmark explores such backtesting by providing a number of trading algorithms which represent common workloads across a set of highly specialised applications. We are in the early phases of porting this benchmark to the FPGA, having focused more initially on STAC-A2.

# 5 FUTURE PLANS

This PhD is still currently in its early stage. At this point we are focusing on developing a solid FPGA port of the STAC-A2 and STAC-A3 benchmarks on the Alveo U280 and U250 to understand what performance the appropriate techniques on the FPGA can provide, but most importantly these are then a foundation for further exploration. One such area is the integration with Xilinx's Accelerated Algorithmic Trading reference design, which itself already enables the streaming of data via the Alveo's QSFP28 Ethernet ports, and exploring the performance of these two benchmarks on the FPGA from that perspective will be most applicable to the project vision that was explored in Section 1. Furthermore, the benchmarks are currently in double precision floating point, exploration of alternative number representations (e.g. single or half precision floating point, or arbitrary precision fixed point) will be very worthwhile. Whilst exploration of this will be interesting on the current generation FPGAs, it is likely to be especially interesting on the next generation AI engines, such as the Versal ACAP which can provide up to 400 AI engines each capable of highly vectorised arithmetic operations running at at-least 1 GHz.

# REFERENCES

[1] Nick Brown. 2020. Weighing Up the New Kid on the Block: Impressions of using Vitis for HPC Software Development. In *2020 30th International Conference on Field-Programmable Logic and Applications (FPL)*. 335–340.
[2] Milan Dvořák and Jan Kořenek. 2014. Low latency book handling in FPGA for high frequency trading. In *17th International Symposium on Design and Diagnostics of Electronic Circuits Systems*. 175–178.
[3] Steven L. Heston. 2015. A Closed-Form Solution for Options with Stochastic Volatility with Applications to Bond and Currency Options. *The Review of Financial Studies* 6, 2 (04 2015), 327–343.
[4] P. Lankford, L. Ericson, and A. Nikolaev. 2012. End-User Driven Technology Benchmarks Based on Market-Risk Workloads. In *2012 SC Companion: High Performance Computing, Networking Storage and Analysis*. 1171–1175.
[5] Christian Leber, Benjamin Geib, and Heiner Litz. 2011. High Frequency Trading Acceleration Using FPGAs. In *2011 21st International Conference on Field Programmable Logic and Applications*. 317–322.
[6] Francis A. Longstaff and Eduardo S. Schwartz. 2015. Valuing American Options by Simulation: A Simple Least-Squares Approach. *The Review of Financial Studies* 14, 1 (06 2015), 113–147.
[7] Xilinx. 2019. *Vitis Accelerated Libraries*. https://github.com/Xilinx/Vitis_Libraries