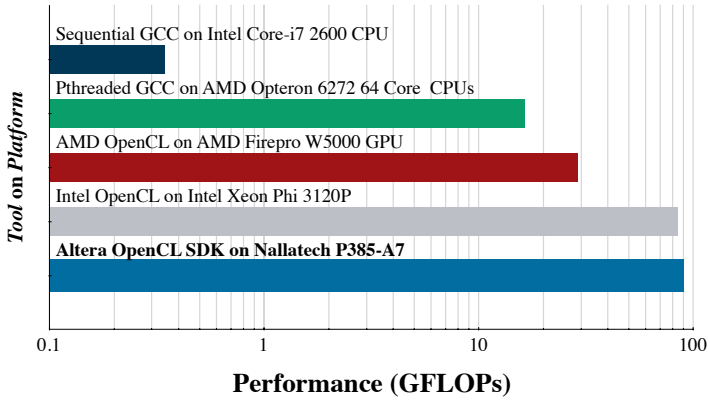
Is Altera's OpenCL SDK ready for business?

Gordon Inggs, Shane Fleming, David Thomas and Wayne Luk
Imperial College London

Software engineers can now add FPGAs to their HPC arsenal with Altera's OpenCL SDK along with compatible FPGA platforms from vendors such as Nallatech.

We have used a case study of challenging problems from Computational Finance to verify this claim. Our OpenCL FPGA implementations achieved performance 256 times faster than a sequential CPU implementation, which is better than the 240 times speedup achieved by the Intel Xeon Phi, the 70 times achieved by a GPU and 30 times by a 64-core CPU.

Fig. 1. Performance of Option Pricing Benchmark Implementations



I. Introduction

The Field Programmable Gate Array (FPGA) research community uses the term "Programability" to describe the challenge of using these high throughput, low energy devices effectively. The awkwardness of the term mirrors the difficulties faced by software engineers using FPGAs without specialist embedded computing knowledge. However this is set to change due to Altera's OpenCL SDK for FPGAs, a High Level Synthesis (HLS) tool based upon the mature, Open Compute Language standard. Beyond ease of adoption offered by the standard, the SDK maps the OpenCL task parallel programming model into highly efficient, pipeline parallel FPGA designs. By providing an easily understood programming model, we believe that Altera's tool finally makes FPGA computing accessible to a wide audience.

Our assertion of this accessibility is based upon a use case of a software developer, who has a limited understanding of FPGA hardware yet has been tasked with using the Altera OpenCL SDK to accelerate computational financial applications. Due to OpenCL's portability, we believe that a developer in this position would be considering competing acceleration technologies such as Multicore CPUs, Graphics Processing Units (GPUs) and hybrid architectures such as Intel's Xeon Phi.

II. Use Case Application Background

To assess the flexibility of the Altera OpenCL SDK, we have considered the valuation of a wide range of financial derivative products with European exercise properties using the Monte Carlo technique. This computationally intensive approach uses the average value of a large number of randomly generated simulations of the underlying asset prices' evolution to find the value of derivative products [1]. The C code in Listing 1 describes the Monte Carlo algorithm as our hypothetical software developer might in terms of the MapReduce computational design pattern [2].

Listing 1. Monte Carlo Option Pricing as MapReduce

```
//Map behaviour
for(i=0;i<PATHS;++i){
    state = path_init(seed++);
    for(j=0;j<PATH_POINTS;++j)
        state = path(state);
    value[i] = payoff(state);
}
//Reduce Behaviour
for(i=0;i<PATHS;++i)
    result += value[i]/PATHS;
```

Option Pricing by FPGA Experts

Imperial College London has published work on the FPGA acceleration of the pricing of exotic options using the Black-Scholes model [3]. Their work demonstrated that the use of FPGAs minimises both the latency and energy utilisation of option pricing computations. However, their implementations were written in a Hardware Description Language (HDL) and incorporated embedded computing expertise.

Similarly, Technise Universität Kaiserslatuarn has published a benchmark of forward looking pricing problems[1] based upon the Heston model [4]. Along with this bench-

TABLE I
OVERVIEW OF EXPERIMENTAL PLATFORMS

Platform	Processor	CMOS Size (nm)	Clockrate (GHz)	Memory (GBs)	OpenCL Processing Elements
Sequential CPU	Intel Core i7-2600S	32	2.8	16	1
Multicore CPU	AMD Opteron 6272	32	2.1	128	64
GPU	AMD Firepro W5000	28	0.825	2	768
Intel Xeon Phi	Intel Xeon Phi 3120P	22	1.1	6	256
Nallatech P385NA7	Altera Stratix V GXA7	28	0.25	8	2-6

TABLE II
LATENCY PERFORMANCE OF OPTION PRICING TASKS(S)

Platform	Heston European	Heston Barrier	Heston Double Barrier	Heston Double Digital Barrier	Black-Scholes Asian
Sequential CPU	14220.39	8378.79	6784.38	6790.95	5747.82
Multicore CPU	307.63	278.04	253.81	277.77	226.69
GPU	123.54	129.99	131.23	131.19	67.09
Intel Xeon Phi	18.08	47.94	57.09	58.55	13.63
FPGA Expert References [3], [5]	287.00	287.00	287.00	287.00	31.68
Nallatech P385NA7	**32.49**	**29.70**	**32.67**	**30.50**	**29.50**

mark, they have published work on the FPGA acceleration of these problems, also demonstrating latency and energy optimisations [5], but again using HDLs and embedded computing expertise to realise these benefits.

III. EVALUATION

For our experimentals, we used the financial engineering problems from the work by FPGA experts described in the previous section: the 12 Kaiserslautern Heston Model-based options, many of which do not have closed form solutions, as well as Imperial College's Black-Scholes Model-based Asian option. Similar to the expert work, we used a large number of simulations, 10 Million, with a high number of discrete time steps, 4096, in each simulation.

Thanks to the large subset of C supported by OpenCL, our source code of the Monte Carlo pricing algorithm for the 13 problems was close to the map behaviour in Listing 1[2], across the experimental platforms detailed in Table I. We used the Combined-Tausworthe random number generator with the Box-Muller transformation to produce the Gaussian random numbers required [6], [7]. The reduction operation to calculate the average option value upon the host CPU. In all cases single precision floating point arithmetic was used.

We used latency to asses performance, i.e. the wall clock time from when execution on the host system is initiated until the price result is returned to the user. Hence the communication time, initialisation overhead and computation time were all captured.

IV. CONCLUSION

From our case study of a wide range of option pricing problems using both the Heston and Black-Scholes models, we conclude:

- Our OpenCL FPGA implementations compare favourably to expert FPGA implementations as well as competing accelerator technologies, as illustrated by the results in Figure 1 and Table II.
- The Altera OpenCL SDK enables the successful exploitation of fine-grained, pipeline parallelism in FPGAs without specialist embedded computing knowledge.
- The long compile times of designs offset the relative ease with which designs may be created.

We thus confirm that Altera's OpenCL SDK would be applicable in a computational finance context, and is indeed ready for business.

[1] http://www.uni-kl.de/en/benchmarking/option-pricing/

[2] https://github.com/Gordonei/ForwardFinancialFramework

REFERENCES

[1] J. C. Hull, *Options, Futures and Other Derivatives*, 8th ed. Pearson, 2011.

[2] J. Dean and S. Ghemawat, "Mapreduce: Simplified data processing on large clusters," in *Proceedings of the 6th Conference on Symposium on Opearting Systems Design & Implementation - Volume 6*, ser. OSDI'04. Berkeley, CA, USA: USENIX Association, 2004, pp. 10–10.

[3] A. H. Tse, D. B. Thomas, K. H. Tsoi, and W. Luk, "Efficient reconfigurable design for pricing asian options," *SIGARCH Comput. Archit. News*, vol. 38, no. 4, pp. 14–20, Jan. 2011. [Online]. Available: http://doi.acm.org/10.1145/1926367.1926371

[4] S. L. Heston, "A Closed-Form Solution for Options with Stochastic Volatility with Applications to Bond and Currency Options," *The Review of Financial Studies*, vol. 6, no. 2, pp. pp. 327–343, 1993. [Online]. Available: http://www.jstor.org/stable/2962057

[5] C. de Schryver, I. Shcherbakov, F. Kienle, N. Wehn, H. Marxen, A. Kostiuk, and R. Korn, "An Energy Efficient FPGA Accelerator for Monte Carlo Option Pricing with the Heston Model," in *2011 International Conference on Reconfigurable Computing and FPGAs (ReConFig)*, 30 2011-dec. 2 2011, pp. 468 –474.

[6] P. L'ecuyer, "Maximally equidistributed combined tausworthe generators," *Mathematics of Computation of the American Mathematical Society*, vol. 65, no. 213, pp. 203–213, 1996.

[7] G. Box and M. E. Muller, "A Note on the Generation of Random Normal Deviates," *The Annals of Mathematical Statistics*, vol. 29, no. 2, pp. 610–611, 1958.