FPGA Acceleration of Monte Carlo-based Financial Simulation: Design Challenges and Lessons Learnt

Pedro Echeverria, *Student Member, IEEE,* and Marisa Lopez-Vallejo, *Member, IEEE*
ETSIT Telecomunicación, Universidad Politecnica Madrid, Spain

The simulation of interest rate derivatives is a powerful tool to face the current market fluctuations. However, the complexity of the financial models and the way they are processed require exorbitant computation times, what is in clear conflict with the need of a processing time as short as possible to operate in the financial market. To shorten the computation time of financial derivatives the use of hardware accelerators becomes a must.

We have implemented an FPGA-based accelerator for financial simulation. More exactly, we have taken as reference the Libor Market Model (LMM) [1], one of the most well-known financial models, that defines the time evolution for the curve of interest rates according to no-arbitrage settings. This way, it models how the interest rates will evolve to price interest rate derivatives.

The solution for this model is obtained with stochastic methods relying on the Monte Carlo simulation and the use of gaussian random variables. To achieve the accuracy demanded by financial simulations, the Monte Carlo based Libor Market Model requires of a huge number of replications, each one implying the computation of a high number of variables. In addition, the equations needed to compute the model can reach enormous complexity and with strong data dependencies. The combination of these two facts makes the LMM a very computationally intensive model which requires a long execution time until the result is obtained.

Based on our experience, here we present the design challenges we have faced up and the lessons we have learned, not only to implement this kind of applications with a satisfying acceleration but also in a reasonable time to market.

I. Design challenges

A complete LMM model presents several design challenges and mainly due to its complex mathematical model and the existence of deep pipelines of numerous operators working in parallel. The most relevant ones are:

- **High quality Gaussian random number generation**. Conventional generators are not suitable for the LMM due to the special requirements of financial simulations (use of variance reduction techniques [2], need of variables with joint multivariate distribution).
- **The need of floating point arithmetic**. Highly accurate results are expected as any little error can mean the loss of a high amount of money. Thereby financial simulation mainly relies on double precision floating-point arithmetic. However, this arithmetic has a high cost on processing time, resources and complexity of the design solution. Precision and area must be traded-off while still obtaining highly accurate results.
- **The processing datapath**. The processing datapath presents strong data dependencies requiring a sofisticated control to obtain a throughput of one LIBOR calculated per clock cycle.
- **The need of an adaptable architecture**. Financial simulations are highly parameterizable depending on the inputs provided to tune the models to the real fluctuations of the markets. The datapath architecture must incorporate mechanisms to be flexible enough to work with different sets of parameters.
- **Efficient Communication**. The LMM can be part of a larger application, requiring the communication of large amounts of data between sw and hw. Efficient communication cores play a key role in the complete system.

II. Lessons learnt

The implementation of the above mentioned accelerator has taken more time than expected. This is not only due to the difficulty of the design, but also to the total absence of high level support tools and the inherent complexity of the connection between hardware and software worlds. Important lessons we have learnt can be summarized as:

- Need of reliable and efficient modules (real pre-designed, verified and documented IPcores), especially for floating-point operators and hw-sw communication interfaces.
- High level tools that free the designer from tedious low level details would be very welcome. The possible reduction on the acceleration is worth being paid in order to shorten the design time.
- Exhaustive validation is a must. Important strategies must be applied, such as divide and conquer, incremental design and verification. This must be done not only for the hardware, but also in the whole hw-sw system.
- Based on previous experiences, designers of these applications must follow a well stablished and tested design methodology.

We can conclude that current FPGAs allow the implementation of extremely complex applications, but we still need improved methodologies and tools to deal with this complexity in feasible design time. Otherwise, exterienced hardware designers would still be a must.

References

[1] A. Brace, D. Gatarek, and M. Musiela, “The market model of interest rate dynamics,” *Mathematical Finance*, no. 7, pp. 127–155, 1997.

[2] P. Echeverría, M. López-Vallejo, and J. M. Pesquero, “Variance reduction techniques for Monte Carlo simulations. a parameterizable FPGA approach,” in *Intl. Conf. on Electronics, Circuits and Systems*, 2008, pp. 1296–1299.