

# Design Space Exploration to Accelerate Nelder-Mead Algorithm using FPGA

Nam Khanh Pham<sup>1,2</sup>, Amit Kumar Singh<sup>1</sup>, Akash Kumar<sup>1</sup>, Khin Mi Mi Aung<sup>2</sup>

<sup>1</sup>Department of Electrical and Computer Engineering, National University of Singapore, Singapore

<sup>2</sup>Data Storage Institute, A\*STAR, Singapore  
 {phamnamkhanh, amit.singh, akash}@nus.edu.sg  
 Mi\_Mi\_AUNG@dsi.a-star.edu.sg

Nelder-Mead algorithm (NMA) is the best-known algorithm for multidimensional optimization without involving derivative computations. Due to the simplicity in implementation and the fast convergent property of NMA, it is widely used in the fields of statistics, engineering, physics and medical sciences. In practice, when objective function is complicated, the optimization procedure requires a lot of computation efforts, leading to a time-consuming process. This work introduces a NMA solver engine fully implemented on FPGA hardware and performs design space exploration to provide various solutions suitable for FPGA device.

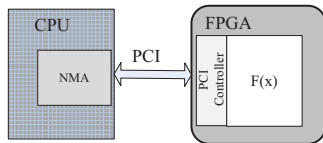


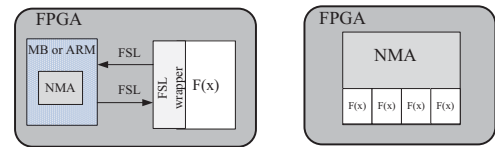
Figure 1. Hybrid solution

Traditional implementations of NMA [3], for example, PC-based solutions cannot cater for low latency requirements for high computation workloads. Therefore, there have been efforts to find FPGA-based solutions to accelerate the overall optimization process [1, 2]. However, existing solutions just focus on the hardware accelerator of the objective functions and leave the NMA solver on PC platform (Fig. 1). Obviously, this *Hybrid Solution* is an inefficient approach in term of performance and energy consumption since it introduces communication overheads between two platforms due to required execution on both PC and FPGA devices.

To address aforementioned issues, we explore various FPGA-based design spaces in order to accelerate the parameter estimation process using NMA algorithm and categorize the designs into two solutions: *Embedded System (ES) solution* and *Pure hardware (HW) solution*. In the ES solution (Fig. 2(a)), the NMA solver is implemented either on the Microblaze (MB) or ARM processor that is embedded on the FPGA device and the objective function using reconfigurable logic on FPGA. The communication between the NMA solver and the HW accelerator of the objective function happens inside the FPGA device; hence, the effort of data transfer through Ethernet link to PC is eliminated.

In the Pure HW solution (Fig. 2(b)), both the NMA solver and the objective functions are implemented on FPGA device. To the best of our knowledge, this is the first effort to develop a HW accelerator for NMA solver. The implementation of NMA solver on HW has several advantages over other solutions:

- Speed up of HW implementation over software imple-



(a) ES Solution (b) Pure HW Solution  
 Figure 2. Proposed solutions

mentation.

- Communication overhead between NMA solver and accelerators of objective functions is totally eliminated.
- Ease of parallelism exploitation when multiple solvers are needed.

Fig. 3 shows that our explored solutions can optimize different objective functions up to hundred times faster than the solutions obtained by existing approaches. The Pure HW approach outperforms the ES approaches in terms of execution time (latency). This is due to additional obtained speed up of the HW implementation over the SW implementation.

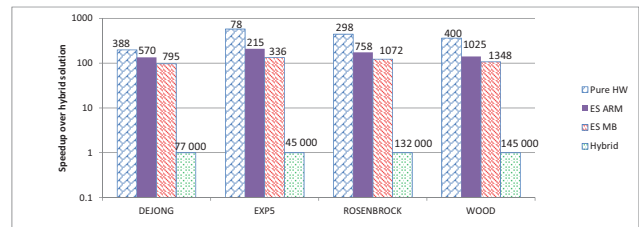


Figure 3. Speed up of Proposed Solutions over Hybrid Solution

Additionally, as shown in Fig. 4, our solutions are much more efficient in terms of energy consumption when compared with state-of-the-art. In ES approaches, since only the objective functions are implemented on HW, the resource utilization and the energy consumption is smaller than the Pure HW approach.

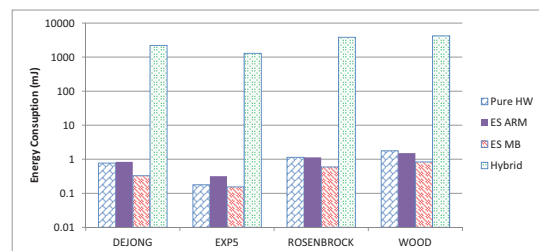


Figure 4. Energy consumption of different Solutions

## REFERENCES

- [1] I. Allugundu, P. Puranik, Y. P. Lo, and A. Kumar. Acceleration of distance-to-default with hardware-software co-design. In *FPL*, pages 338–344. IEEE, 2012.
- [2] C. Guo and W. Luk. Accelerating maximum likelihood estimation for hawks point processes. In *FPL*, pages 1–6. IEEE, 2013.
- [3] J. A. Nelder and R. Mead. A simplex method for function minimization. *The computer journal*, 7(4):308–313, 1965.