# AccSim: A Flexible Hardware Accelerator Simulation Framework

xxx, xxx and xxx
School of Computing
National University of Singapore
Email: {xxx}@xxx

***Abstract*—xxx**

## I. Introduction

Improving general-purpose processing system is getting extremely difficult. More and more computer architects believe that the major improvements in cost-energy-performance will come from domain-specific hardware accelerators. Recent years have already seen a number of successful demonstrations utilizing domain specific hardware accelerators for critical domains of applications such as deep neural network [1], [2] database operations [3] and graph processing [4], [5]. In order to explore the hardware accelerator design, a hardware accelerator simulator is usually required. Indeed there are already many exisitng tools [**?**], [**?**] and models [**?**], [**?**] that can be used to help with the hardware accelerator design, it is non-trivial to develop a hardware accelerator on top of these work. For instance, there is a lack of general public cycle-accurate memory models available in [**?**], [**?**] while [**?**], [**?**] expose only primitive memory access interface and need to be further wrapped for an accelerator simulator. And a general accelerator simulator framework is highly desired for the hardware accelerator simulator development.

Despite the difference of the accelerator simulators, we argue that a general accelerator simulator design framework should have three common yet important features. First of all, it should provide memory models of various memory architectures. Basically memory is usually critical to the hardware accelerator and greatly affects the accelerator design. At the same time, memory techniques evolve rapidly over the years and novel memory architectures with distinct features emerge. In order to explore hardware accelerator design, various memory architectures needs to be evaluated. Secondly, it should provide abstract user-frinedly memory interfaces. Hardware accelerators usually have complex memory access patterns such as stream access, burst access as well as random access. Thus higher abstract memory access interface instead of primitive memory access interface should be provided. Thirdly, it should provide trade-off between simulation speed and precision. Hardware accelerators may have distinct simulation speed and precision requirements while exploring the hardware accelerator. For instance, some of the applications such as graph accelerators may process on a big data set. Low-level accurate memory model may result in extremely long simulation. Thus a simplified memory model should be used to obtain the general performance of the accelerators. For applications that are sensitive to the memory access latency, more accurate memory models are preferred.

There is still a lack of general accelerator simulator framework that fullfills all the three features mentioned above. To that end, we proposed a flexible hardware accelerator simulation framework to be reused for general hardware accelerator simulator development. Basically, it integrates ramulator supporting various memory architectures as the underlying memory model and thus allows hardware accelerator exploration over a broad range of memory architectures. In addition, abstract memory interfaces as well as memory content management are provided to faciliate the accelerator accessing the memory model. Finally, it also provides a mix of cycle-accurate memory model and simiplified analytical memory model obtained though sampling to compromise on simulation speed and accuracy.

The rest of the paper is organized as follows. Section 2 briefly reviews the related work. Section 3 presents the proposed accelerator simulation framework. Section 4 provides the experimental results and Section 5 concludes this paper.

## II. Related Work

Simulator is of vital importance for accelerator design and exploration especially in early design stage. While many accelerator may involve considerable memory access, a memory model should be included in the simulator as well. Although there have been a number of different open-source high-level memory models developed, these memory models are usually developed targeting general computer architectures or memory architectures and can not be easily adapted to a specific accelerator. As a result, developing an accelerator simulator with integrated memory model still takes quite a lot of design efforts.

EDA vendors such as Xilinx, Altera and Synopsys also provide accurate memory models as well as ample IPs which are convenient for accelerator development. However, these facilities are mostly used together with a hardware description language (HDL) model for prototyping or fabrication on ASICs. And it takes long time to develop the accelerator and the resulting design is usually not flexible for accelerator architecture exploration. In addition, they typically support only mature DDRx and LPDDRx models and don't cover novel and emerging memory architectures. As a result, this also limits the exploration of the accelerator architectural exploration.

In this work, we developed a flexible accelerator simulation framework for accelerator simulator design and exploration. It wrapps up the ramulator such that various memory models are supported and frequent memory interfaces are provided. In addition, it also allows trade-off between the simulation speed and precision. With this feature, accelerators operating on large data set can also be simulated with moderate accuracy and simulation time.

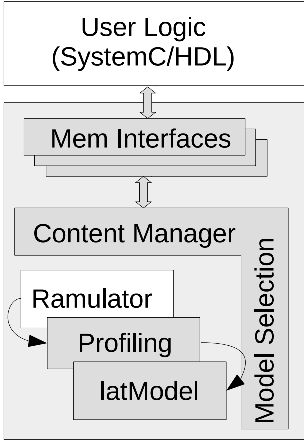
## III. AccSim Design Framework

In this work, a flexible accelerator simulator framework AccSim is developed for rapid accelerator design and exploration. The framework overview and supported features will be presented in the following subsections.

### A. AccSim Overview

AccSim overview is shown in Figure III-A. It can be seen that an accelerator simulator roughly consists of user logic descrbing the application and memory models simulating the memory access.

Fig. 1. AccSim Framework



### B. Major AccSim Features

memory interfaces
content management
co-simulation
simulation precision and speed trade-off

## IV. Experiments and results

### A. Experiment Setup

environment and benchmark applications

### B. Performance on Different Memory Models

### C. Simulation Speed and Precision

## V. conclusion

In this work, we have presented an automatic nested loop acceleration framework that is based on a soft coarse-grained reconfigurable array overlay. We have demonstrated that by taking advantage of the regularity of the overlay, intensive system customization specific to the given user application can be performed efficiently, resulting in up to 5 times performance improvement over solutions without customization at the cost of 10 to 20 minutes additional tools run time. Overall, the framework is able to generate accelerators that achieve up to 10 times speed up over software running on the host processor, resulting in a high design productivity experience for software programmers.

## Acknowledgment

This work was supported in part by the Research Grants Council of Hong Kong project ECS 720012E and the Croucher Innovation Award 2013.

## References

[1] N. P. Jouppi *et al.*, “In-Datacenter Performance Analysis of a Tensor Processing Unit TM,” *Isca*, pp. 1–17, 2017.

[2] Z. Li *et al.*, “A survey of neural network accelerators,” *Frontiers of Computer Science*, May 2017. [Online]. Available: http://dx.doi.org/10.1007/s11704-016-6159-1

[3] L. Wu *et al.*, “Q100: The Architecture and Design of a Database Processing Unit,” *SIGARCH Comput. Archit. News*, vol. 42, no. 1, pp. 255–268, 2014. [Online]. Available: http://doi.acm.org/10.1145/2654822.2541961

[4] T. Jun *et al.*, “Graphicionado : A High-Performance and Energy-Efficient Accelerator for Graph Analytics,” *49th International Symposium on Microarchitecture*, vol. To Appear, 2016.

[5] M. M. Ozdal *et al.*, “Energy Efficient Architecture for Graph Analytics Accelerators,” *Proceedings - 2016 43rd International Symposium on Computer Architecture, ISCA 2016*, pp. 166–177, 2016.