SCGPSim: A fast SystemC simulator on GPUs

Mahesh Nanjundappa[1], Hiren D. Patel[2], Bijoy A. Jose[1] and Sandeep K. Shukla[1]

[1]FERMAT Lab, Virginia Polytechnic Institute and State University, [2]University of Waterloo

FERMAT @ Virginia Tech
VIRGINIA POLYTECHNIC INSTITUTE AND STATE UNIVERSITY
Formal Engineering Research with Models, Abstractions and Transformations

ABSTRACT

SystemC promises an environment for faster hardware/ software design-space exploration.

With systems becoming more complex, the simulation performance of the corresponding SystemC RTL design deteriorates significantly as current SystemC standard makes no effort to parallelize its simulation kernel to take advantage of multi-core platforms that are becoming the norm nowadays.

In this work, we present a translator that transforms synthesizable SystemC designs into parallely executable programs targeting an NVIDIA graphics processing unit (GPU).

The translator retains the discrete-event semantics of the original designs by applying semantic preserving transformations. The resulting programs exploit the hardware parallelism for improved simulation efficiency.

Preliminary experiments show a simulation speed-up of approximately 30x-100x.

CONTACT

Mahesh Nanjundappa
FERMAT Lab, Virginia Tech
Email: knmahesh@vt.edu
Phone: 540-425-1151
Website: http://filebox.vt.edu/users/knmahesh/

INTRODUCTION

➤ SystemC, a modeling and simulation language, has been used in making early trade-off analysis and design-space exploration.

➤ OSCI implementation of SystemC simulation kernel, being a *Discrete Event Simulation Kernel* makes no attempts to take advantage of recent parallel architectures.

➤ We present a source-to-source (S2S) translator that transforms SystemC designs into parallely executable CUDA programs that run on NVIDA Graphics Processing Units (GPUs).

➤ Preliminary experiments show impressive results by giving speed-ups from 30-100x on certain benchmarks.

PARALLEL SYSTEMC KERNEL

➤ Parallelism is extracted by mapping each runnable process to a thread belonging to different warp and executing multiple threads parallely on GPU.

➤ To prevent *race* condition, **Double Buffering** mechanism is used.

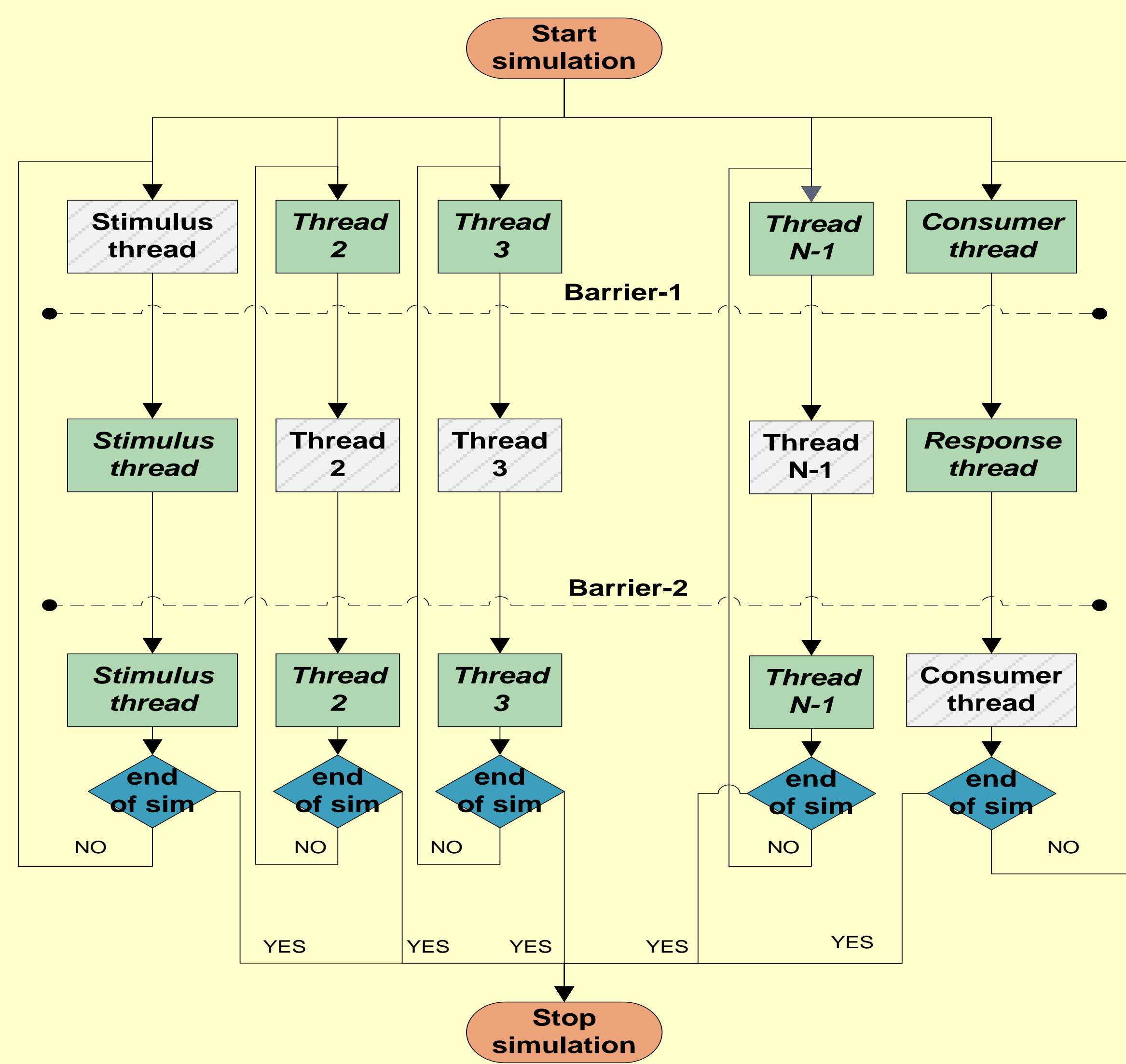➤ **Barrier Synchronization** is employed for synchronization of various threads.

Fig1: A Parallel SystemC Simulation Kernel

Thread doing simulation work
Thread Idle and waits for synchronization

SYSTEMC TO CUDA TRANSLATION

➤ XMLization provides an intermediate representation of the SystemC design.

➤ Intermediate format is parsed to extract structural information of the design.

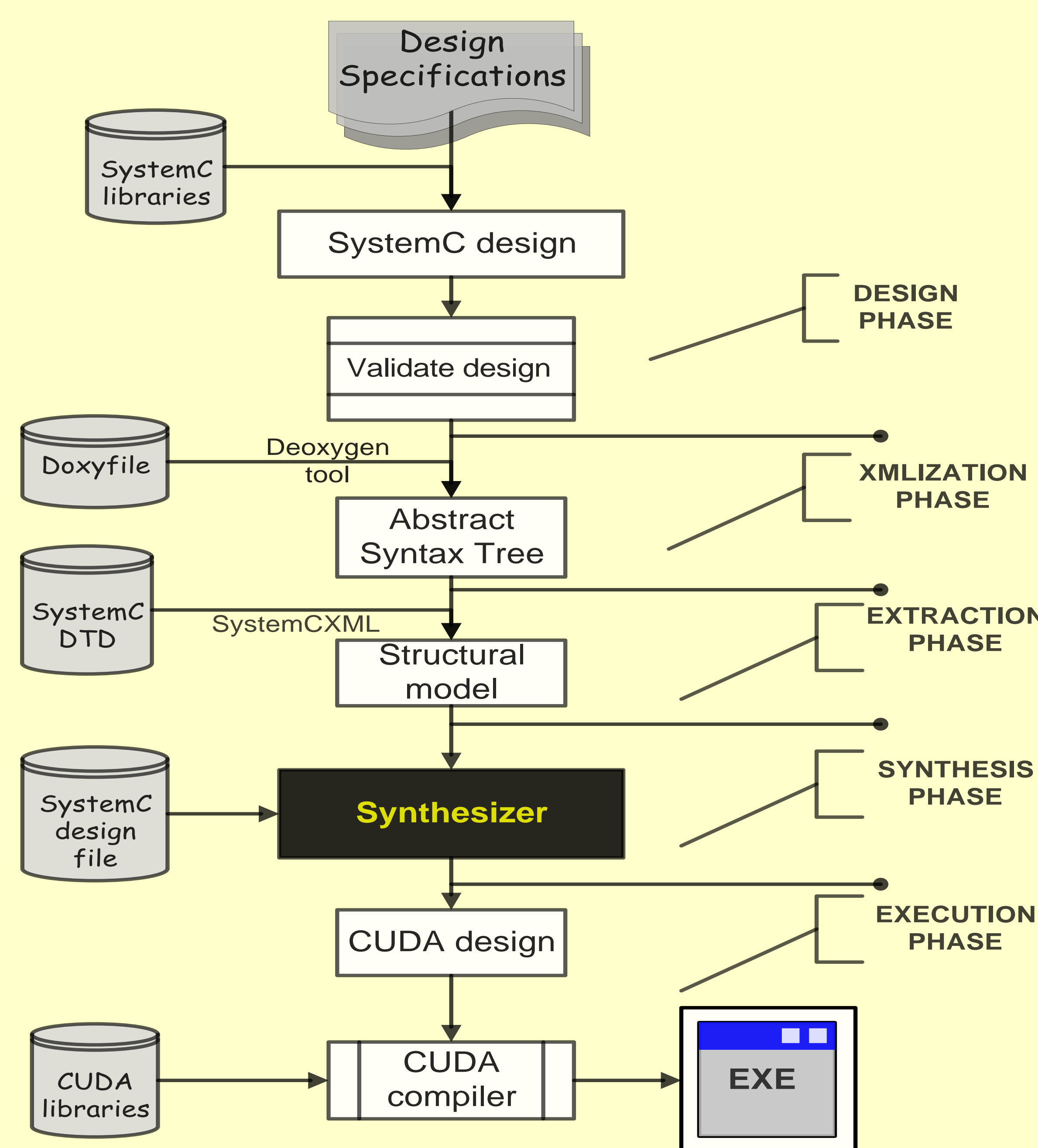➤ Synthesizer/ Translator uses semantic preserving transformations to automatically translate SystemC to CUDA.

Fig2: Design Flow of SystemC to CUDA Translator

EXPERIMENTAL RESULTS

➤ Experiments conducted for the benchmarks listed in Table 1 and the corresponding time taken and speed-up obtained is notified in the table.

➤ SystemC CPU simulation was done on Intel Core2Duo processor running at 1.83GHz with 1.5GB of RAM.

➤ SystemC GPU simulation was done on NVIDIA's TESLA D870 platform.

➤ Experiments were ran for 100,000 cycles and the results of both the methods were found to be same.

Design	CPU (sec)	CPU+GPU (sec)	Speed Up
Pipeline AES	120.1	3.916	30.66
FIR	51.9	1.37	37.88
10 Stage Buffer	28	0.277	101.083
Simple ALU	13	0.146	89.041
3 Stage Buffer	11	0.276	39.85

Table1: Experimental results for various examples
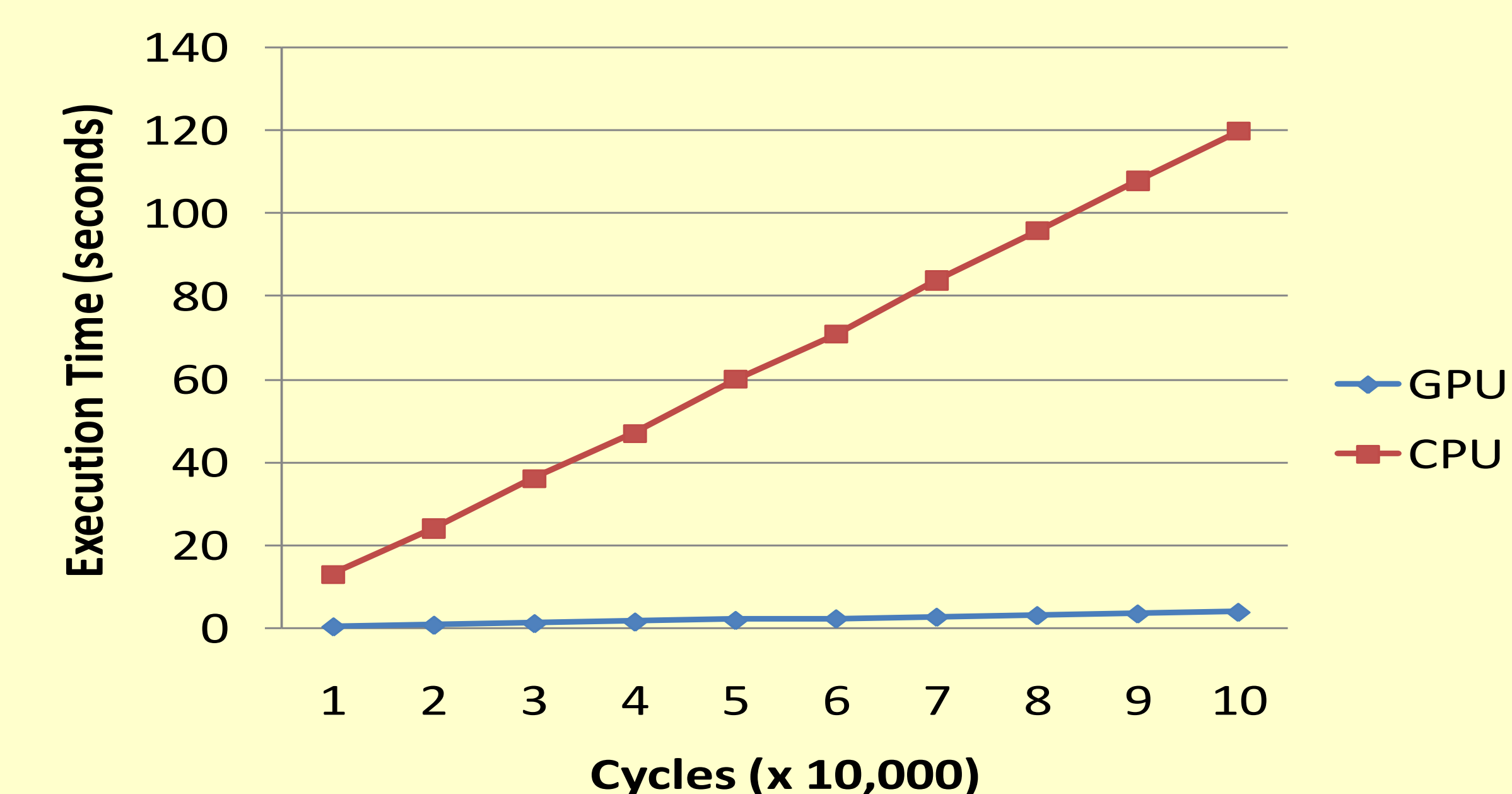
Graph1: Time taken versus Number of cycles for pipeline AES example

CONCLUSIONS

➤ An automatic translation of synthesizable subset of SystemC designs into parallely executable CUDA designs.

➤ Experimental results reveal an immense potential of GPU's being used in EDA applications to increase the performance of the EDA tools.

➤ Future work on expanding the translator to be able to handle translations of designs beyond synthesizable subset of SystemC.

REFERENCES

1. Nick Savoiu, Sandeep K. Shukla, and Rajesh K. Gupta. "Design for synthesis, transform for simulation: Automatic transformation of threading structures in high-level system models". UC Irvine Technical Report, 2001.

2. Mahesh Nanjundappa, Hiren D. Patel, Bijoy A. Jose and Sandeep K. Shukla. Technical Report on SCGPSim: Alternate SystemC Simulator, FERMAT Technical report 2009-04.

3. David Berner, Jean-Pierre Talpin, Hiren D. Patel, Deepak Mathaikutty, and Sandeep K. Shukla. Systemxml: An extensible SystemC front end using xml. In FDL, pages 405{409. ECSI, 2005.

4. NVIDIA Corp. NVIDIA CUDA. http://www.nvidia.com/cuda