

GRAAL — A Development Framework for Embedded Graphics Accelerators

D. Crisu, S.D. Cotofana, S. Vassiliadis
Delft University of Technology
Computer Engineering Laboratory
Mekelweg 4, 2600 GA Delft, The Netherlands
{dan, sorin, stamatis}@ce.et.tudelft.nl

P. Liuha
Nokia Research Center
Visiokatu-1, SF-33720
Tampere, Finland
petri.liuha@nokia.com

Abstract

This paper presents a versatile hardware/software co-simulation and co-design environment for embedded 3D graphics accelerators. The GRAphics AccELerator design exploration framework (GRAAL) is an open system which offers a coherent development methodology based on an extensive library of SystemC RTL models of graphics pipeline components. GRAAL incorporates tools to assist in the visual debugging of the graphics algorithms implemented in hardware, and to estimate the performance in terms of throughput, power consumption, and area.

1. Introduction

With the advent of system-on-chip (SOC) design paradigm for embedded systems, 3D graphics hardware acceleration to support rich graphics interfaces and 3D entertainment environments on mobile platforms is becoming increasingly popular. To address the lack of specific tools to support the graphics architecture development process [2] and to leverage early decisions in the image quality/performance/power/cost design exploration space of embedded graphics accelerators, we designed GRAAL, a versatile hardware/software co-simulation and co-design tool. The GRAphics AccELerator design exploration framework (GRAAL) is an open system which offers a coherent development methodology based on an extensive library of graphics pipeline components modeled at RT-level in SystemC [3]. As a consequence, an entire system-on-chip can be simulated by integrating third-party SystemC models of components (microprocessors, memories, peripherals) along with our own parameterizable SystemC RTL model of the graphics hardware accelerator. GRAAL incorporates tools to assist in the visual debugging of the graphics algorithms implemented in hardware, and to estimate the performance in terms of throughput, power consumption, and area.

2. GRAAL design exploration framework

A pictorial representation of the design exploration framework we propose is presented in Figure 1. Within this framework, different graphics software applications can be run on the virtual system and various graphics hardware accelerator organizations can be developed, verified, evaluated, and optimized without building a physical prototype.

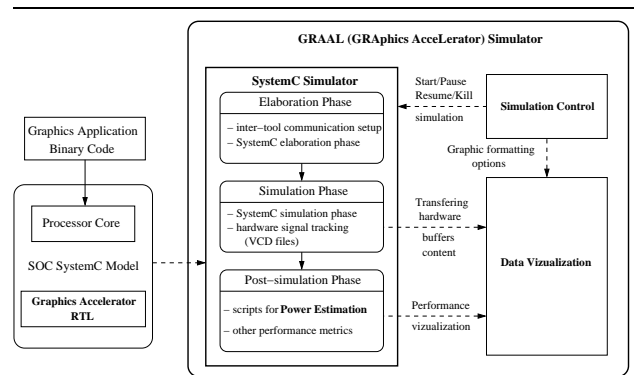


Figure 1. GRAAL tool framework.

GRAAL Simulator provides a graphical front-end for SystemC simulation control, data visualization, and performance estimation in our graphics hardware accelerator design exploration framework. Some of the implemented capabilities are depicted in Figure 2.

GRAAL offers two fast power estimation strategies, both based on SystemC RTL simulation. Both strategies are estimating the average power consumption over the entire simulation period, providing as a byproduct the energy drawn by the graphics accelerator from the battery. The first strategy is based on several Synopsys tools (CoCentric SystemC Compiler, Design Compiler, Power Compiler) [5], connected by custom-developed scripts, and provides estimates with a lower degree of accuracy (50%) being suitable for the initial micro-architecture exploration phase. For the tool inter-operation details the reader is referred to [5].

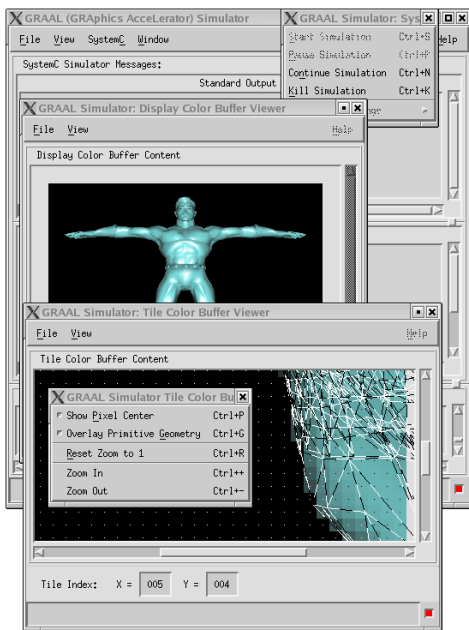


Figure 2. SystemC simulation control and graphical visualization in GRAAL Simulator program.

The second strategy utilizes the approach described in [1], requiring more technology-dependent data setup from the user, and is capable to deliver estimates within 25% of circuit-level simulations.

To facilitate design exploration, a library of OpenGL-compatible hardware modules has been modeled in SystemC that include all the Rasterizer Stage functionality and that can be plugged together to build a full-fledged graphics hardware accelerator. These models can be used as micro-architectural templates to support further refinement. All modules that implement OpenGL functionality have a fully parameterizable datapath, and can be configured to support a tile-based rasterization approach [6]. In addition, miscellaneous modules for interfacing in SOC, pseudo-modules for performance monitoring, signal activity recording, graphical visualization, etc. are provided.

3. A case study

To assess the effectiveness of the GRAAL development framework, we have designed, using the SystemC module library described in Section 2, an OpenGL 1.2 compliant 3D graphics hardware accelerator to be embed in an ARM based SOC platform. The accelerator was designed for QVGA displays and adopts a tile-based rasterization approach with the display conceptually split in 10×15 tiles, each with a 32×16 pixel resolution. One frame of the AWadvs-04 component of the OpenGL bench-

Processed Triangles	Fragments		Frame duration (clock cycles)
	processed	passed to color TFB	
15518	9510877	8759339	12168491

Table 1. Frame workload.

IC Technology		Std. Cell Library
UMC <i>Logic18-1.8V/3.3V-1P6M</i>		VST <i>eSi-Route/11</i>
Target Clk. Freq.	Std. Cell No.	Total Cell Area
200MHz	106k	2.44mm ²
Frame Estimated Average Power		Frame Estimated Energy
206mW		12.53mJ

Table 2. Graphics hardware estimation results.

mark SpecViewperf 6.12 [4] was generated on our virtual SOC platform using the GRAAL tool framework. The frame image obtained is presented in Figure 2. A few characteristics of the frame workload on the graphics accelerator obtained by SystemC RTL simulation are presented in Table 1. The results of the hardware synthesis on the graphics accelerator and estimated average power/energy drawn from the battery per frame duration are presented in Table 2.

4. Conclusions

In this paper was presented GRAAL, a versatile hardware/software co-simulation and co-design tool for embedded 3D graphics accelerators. An assessment of the effectiveness of the development framework has indicated that a candidate implementation of an embedded graphics accelerator can be developed in one working day and, on a typical PC, data with a simulated throughput of approximately 100 frames a day can be acquired from the simulated hardware for performance estimation purposes.

References

- [1] D. Crisu, S. Cotofana, and S. Vassiliadis. An Energy-Aware Architectural Exploration Tool for ARM-Based SOCs. In *Proceedings of 12th Annual Workshop on Circuits, Systems and Signal Processing, ProRISC 2001*, Nov. 2001.
- [2] Z. S. Hakura and A. Gupta. The Design and Analysis of a Cache Architecture for Texture Mapping. In *Proceedings of the 24th International Symposium on Computer Architecture*, pages 108–120. ACM Press, 1997.
- [3] The Open SystemC Initiative (OSCI), URL: <http://www.systemc.org>.
- [4] SPECviewperf 6.1.2, URL: <http://www.specbench.org>.
- [5] Synopsys Inc., URL: <http://www.synopsys.com>.
- [6] J. Torborg and J. Kajiya. Talisman: Commodity Realtime 3D Graphics for the PC. In *SIGGRAPH 96 Conference Proceedings*, pages 353–364, 1996.