LEARNING HOW TO DESIGN A GRAPHICS PROCESSOR

Witold Kinsner, Dario Schor, Samantha Olson, and Mount-First Ng

Department of Electrical and Computer Engineering
University of Manitoba, Winnipeg, MB, Canada R3T 5V6
{kinsner@ece.umanitoba.ca | dario_schor@umanitoba.ca}

ABSTRACT

This paper describes the design and implementation of a rudimentary graphics processor, called GRAFIX, intended for use in a simple handheld gaming console. The processor is a part of a laboratory in an undergraduate course on Digital Systems Design 2 (DSD2) [1-2]. The GRAFIX processor can perform two different operations: (a) drawing an individual pixel, and (b) drawing a line using the Bresenham line drawing algorithm.

The DSD2 course provides foundational material on discrete mathematics and the theory of modern very-large switching circuits. It presents computer engineering students with a firm foundation in the modern theory of optimal logic design. It illustrates some applications through formal characterization of combinational functions and sequential machines, using contemporary techniques for the automatic synthesis and diagnosis of digital systems. It discusses (i) the design of VLSI systems with problems and approaches; (ii) ground-up development of algebraic structures, lattices, Boolean algebras for a generalized switching theory; (iii) exact optimization of two-level switching functions; (iv) heuristic techniques for the optimization of two-level logic circuits; (v) analysis, synthesis and optimization of complemented binary decision diagrams (BDDs) [3]; and (vi) provides design examples throughout the course.

This fairly high-level lab design of a graphics processor is possible because the students have acquired the necessary prerequisite knowledge from previous courses. None of the courses, however, has attempted to develop a complete processor of such scale. The machine specifications include: (a) interfacing of the host to the GRAFIX unit may be synchronous or asynchronous (one must be selected and the consequences of the choice must be discussed); (b) the frame buffer can be either a single port memory or a dual port memory (again, one must be selected, and the consequences of the choice must be discussed); (c) interfacing of the GRAFIX unit to the frame buffer may be synchronous or asynchronous. The GRAFIX lab is split into four sessions: (i) familiarization with tools and design of the arithmetic logic unit (ALU) for GRAFIX, (ii) design of its Data Path Unit (DPU), (iii) design of its Computing Control Unit (CCU), and (iv) integration and testing.

The objectives of those sessions are to learn how to (a) formulate an architecture of a simple graphics processor, (b) formulate an appropriate ALU, (c) formulate a DPU and CCU, (d) formulate a Command Interpreter, (e) formulate an Controller/Sequencer, (f) formulate the CCU interfacing with the GRAFIX I/O and with the DPU, (g) construct test procedures for each unit and incremental testing, (h) construct a supervisory module to provide the completed GRAFIX processor with test input, (i) integrate the system, (j) test operation of the GRAFIX processor, and (k) describe the system. VERILOG is used as the hardware-description language of choice.

The paper provides a detailed description of the architecture of the graphics processor, its design and implementation, as well as experience from running the laboratory many times.

References

- [1] Witold Kinsner, *Switching and Automata Theory*. Winnipeg, MB: University of Manitoba, 2011, 874 pp.
- [2] Witold Kinsner, *Laboratory Writeups*. Winnipeg, MB: University of Manitoba, 2000, 40 pp.
- [3] Gary D. Hachtel and Fabio Somenzi, *Logic Synthesis and Verification Algorithms*. New York, NY: Springer, 2006 (2nd ed.), 564 pp. {ISBN 978-0-387-31004-6}

Conference Topics: (1) Classroom innovation; (2) Undergraduate education.

Keywords: Undergraduate education; laboratory design; graphics processor.