donuts!

ECE 3375 Final Project, 2024

James Su, Shiv Patel, Khalid Zabalawi

March 22, 2024

ECE 3375 project

James Su, Shiv Patel, Khalid Zabalawi

Problem Definition

3D rendering is a common modern computing task, relevant in many different applications from animation to engineering. Graphics rendering is often assigned to a separate, specialized processor modern computers with a more parallelizable architecture.

Graphics cards are microcontrollers that are purpose built to handle parallelizable tasks, including graphics rendering. It contains its own microprocessor, the GPU, as well as its own memory and I/O.

In this project, we'll build a simple 3D graphics renderer on the DE10-SoC. It will show a 3D torus on a monitor using the VGA port, and provide controls to rotate the torus.

Effect on the user

3D rendering is a very generalizable technique with varying applications. Offloading graphics can allow the main processor to handle other tasks, increasing the efficiency of the device as a whole. Hardware architectures that excel at parallel tasks are also conveniently applicable in other computational tasks like machine learning, scientific computing, data mining and more. Real graphics cards provide this through CUDA and ROCm software tools.

Functional Description

When the program starts, a connected monitor will show a rendered torus.

Buttons 0-2 will correspond to rotations in the X, Y, and Z axis, and switches 0-2 will control the direction of these rotations respectively.

As far as input goes, it's relatively simple. However, the output space is much larger and the system must project the 3D object onto a 2D plane to form a z-buffer.

As far as making a GPU goes, this is more like writing a GPU *driver*. The general use ARM CPU is capable of doing anything a CUDA core can, but it's a lot slower for certain tasks (and vice versa).

Input/Output

- for input, we use the buttons and switches, and timer

- pressed buttons rotate the object, and the switches control the direction
- the timer is used to track the elapsed time since the last update
  * this way, we get more precise control over the rotation speed

Although the DE10-SoC could emulate GPU hardware with the FPGA, we'll stick to using the CPU for this project.