Hardware Acceleration for Physical Modeling of Deformable Objects

Benjamin Bishop
Department of Computer Science
University of Georgia
Athens, GA 30602

Thomas P. Kelliher
Department of Mathematics and Computer Science
Goucher College
Baltimore, MD 21204

1 Introduction

Recently, there has been a great deal of interest in interactive physical modeling. Most current work has focused on rigid body simulation, since it is typically much faster than simulation of deformable objects.

Interactive simulation of scenes containing complex deformable objects on consumer-grade PCs appears to be several years away if we wait for Moores law.

2 Our Work

We believe that it is possible to use specialized hardware to bring complex interactive physical modeling to the consumer in the very near future, similar to what has happened with the specialized consumer 3d card market.

In order to prove this idea, we built a proof of concept system. This system involves a high density FPGA on a custom board connected to a host machine via parallel cable. It implements forward Euler integration of the spring equations and only simple collision detection in two dimensions. It is non-pipelined, and uses reduced precision functional units. An image of the board is shown in Figure 1. The architecture that was implemented in the FPGA is outlined in Figure 2.

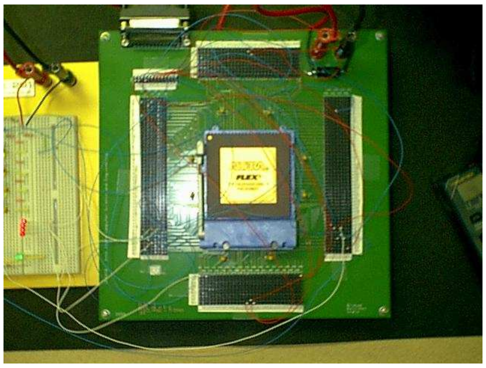

Figure 1: Custom circuit board with FPGA

These simplifications were necessary in order to deal with the area restrictions in the FPGA. The final FPGA utilization was 79% of the 250k gates. The high utilization was caused by the lack of interconnect available on the FPGA, and the fact that barrel shifting in floating point operations quickly consumes interconnect.

It is well known that forward Euler integration suffers from stability problems. We chose this scheme only for its simplicity. For later generations, a more suitable (stable/simple) integration technique must be found. This scheme should operate on local data, since global dependencies significantly complicate the hardware design.

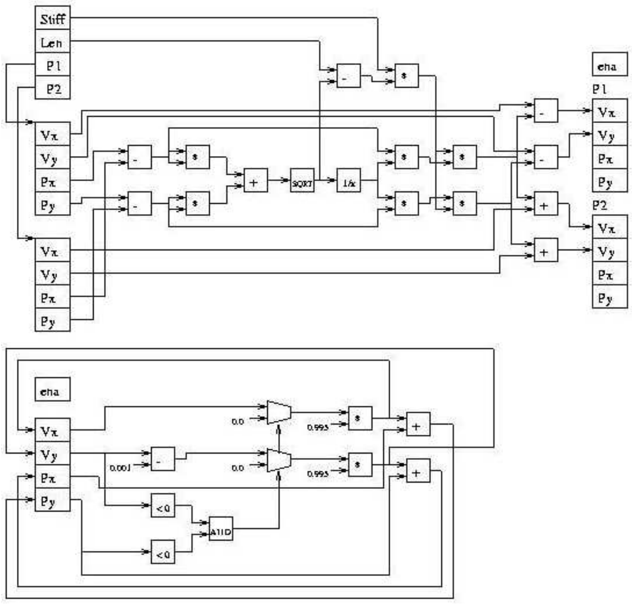

Figure 2: Architecture implemented in the FPGA. Spring solver at top, simple collision detection at bottom.

3 Results

Figure 3 shows an animation sequence that was generated on the board. It is a simple, 4 point, 6 spring cube. In comparison with our host machine, we estimated the performance improvement (assuming pipelining and equal clock rates, which would be easily attainable in an ASIC) to be 92 times, so the potential gain from hardware acceleration is clear.

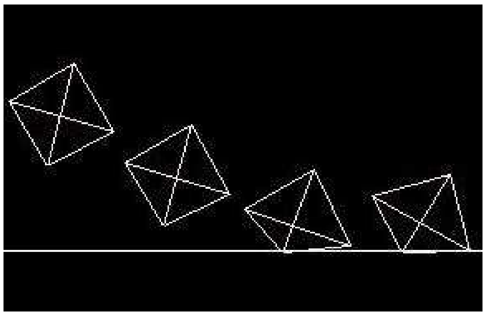

Figure 3: Animation generated in hardware