

Field-Programmable Physics Processor (FP³)

Muiris Woulfe and Michael Manzke



Interaction, Simulation and Graphics Lab (ISG), Department of Computer Science, Trinity College Dublin

Abstract

We outline the design and implementation of an FPGA-based numerical integrator that will form the basis of our FPGA-based physics engine. Physics engines are considered one of the most important of a multitude of components requesting CPU time in a modern computer game, and so we propose offloading this physics computation to an FPGA-based physics engine. Moreover, we suggest inserting FPGAs into PCs, as the FPGAs' reconfigurability could be used to accelerate a wide range of applications; when the user is playing games, the FPGA could be configured to accelerate game physics, as outlined here. Our physics processor uses the fourth order Runge-Kutta numerical integration algorithm to solve the ordinary differential equations used in game physics. Our current analyses indicate that the performance of our physics processor should surpass that of the equivalent software executing on a CPU when several objects are simulated.

control signals. The register file supplies the relevant data to the arithmetic unit and stores the results of the arithmetic unit's operation. Data hazards are resolved inside the register file using a simple busy bit mechanism.

The arithmetic unit currently implements three floating-point operations – addition, subtraction and multiplication. These operations use IEEE 754 single-precision floating-point cores, which our analyses indicated were of sufficient precision for game physics.

Performance

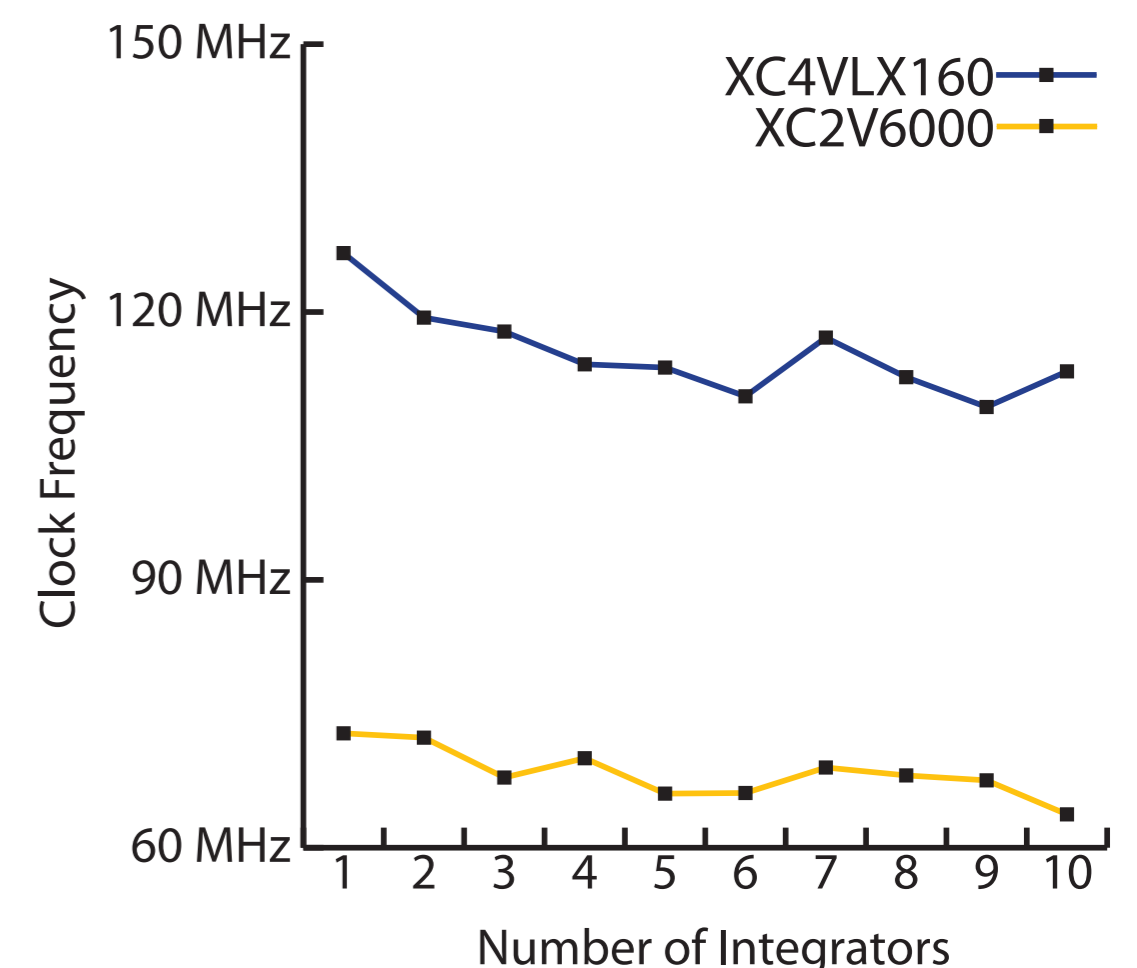
To evaluate the performance of our design, we implemented a simplified model of a ship. The first integration of this model consumes 408 clock cycles while each subsequent integration consumes 383 clock cycles. The reduction in timing is primarily due to instruction overlapping between integrations.

Numerical Integration

Ordinary differential equations (ODEs) describe the dynamic behaviour of objects, and so form the basis of game physics. To solve ODEs computationally, a numerical integration algorithm is typically employed. Due to the essentiality of ODEs to game physics, we began our implementation of the field-programmable physics processor with a numerical integrator.

Numerical integration algorithms approximate the solutions of ODEs. It is desirable to choose an algorithm that is as accurate as possible when considering the constraint that a physics engine is an interactive real-time system. Based on these observations, we decided to implement the fourth order Runge-Kutta algorithm, as it offers a satisfactory tradeoff between speed and accuracy.

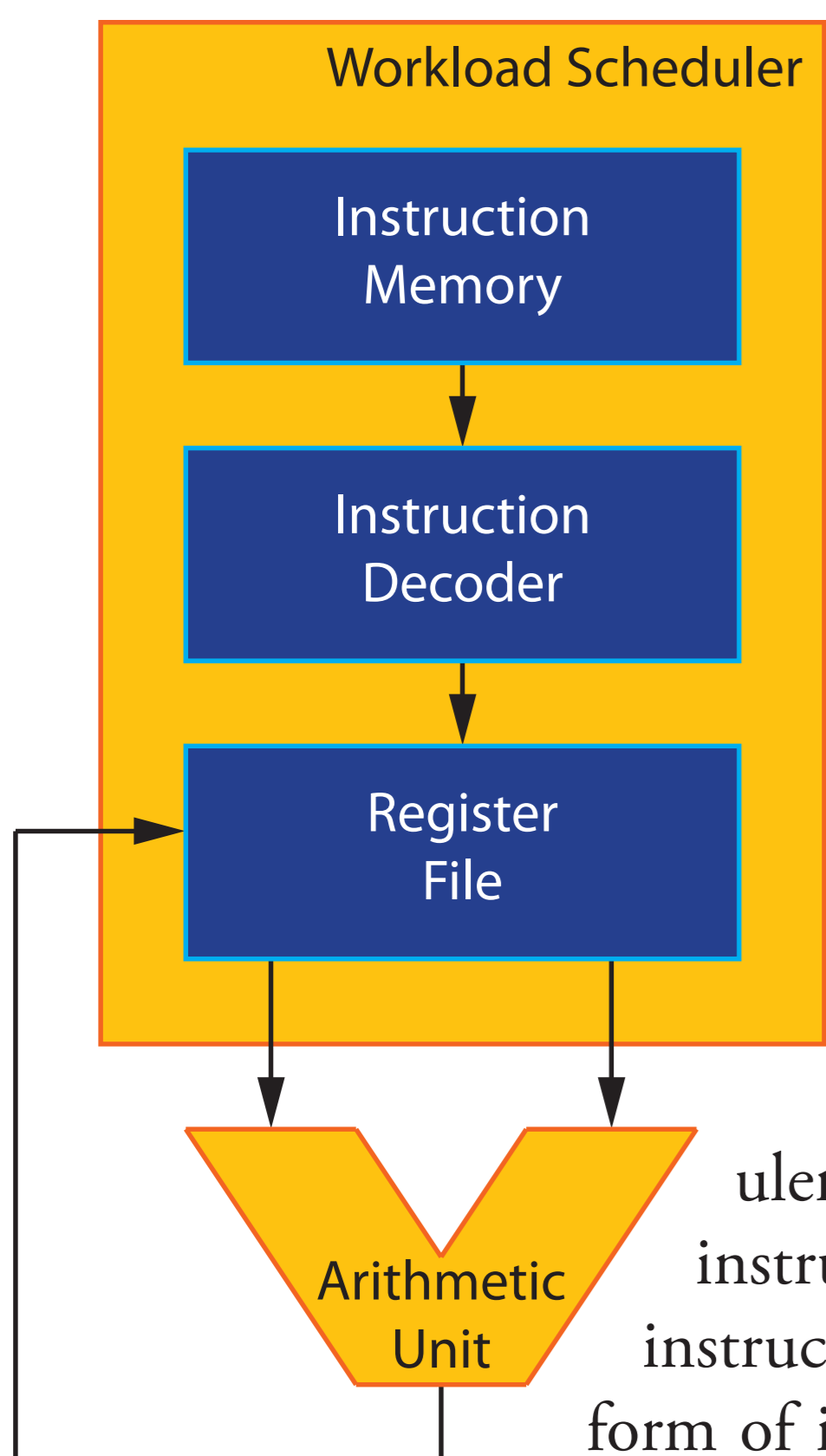
We tested our integrator on a Xilinx XC2V6000 FPGA of speed grade four and a Xilinx XC4VLX160 FPGA of speed grade twelve. The shortest period achieved on the XC2V6000 is 13.734 ns, which yields 5260.122 ns for the subsequent integrations. The shortest period achieved on the XC4VLX160 is 7.899 ns, which yields 3025.317 ns for the subsequent integrations. For comparison, the equivalent software algorithm executes in 56.11 ns on a 3.4 GHz Intel Pentium 4.



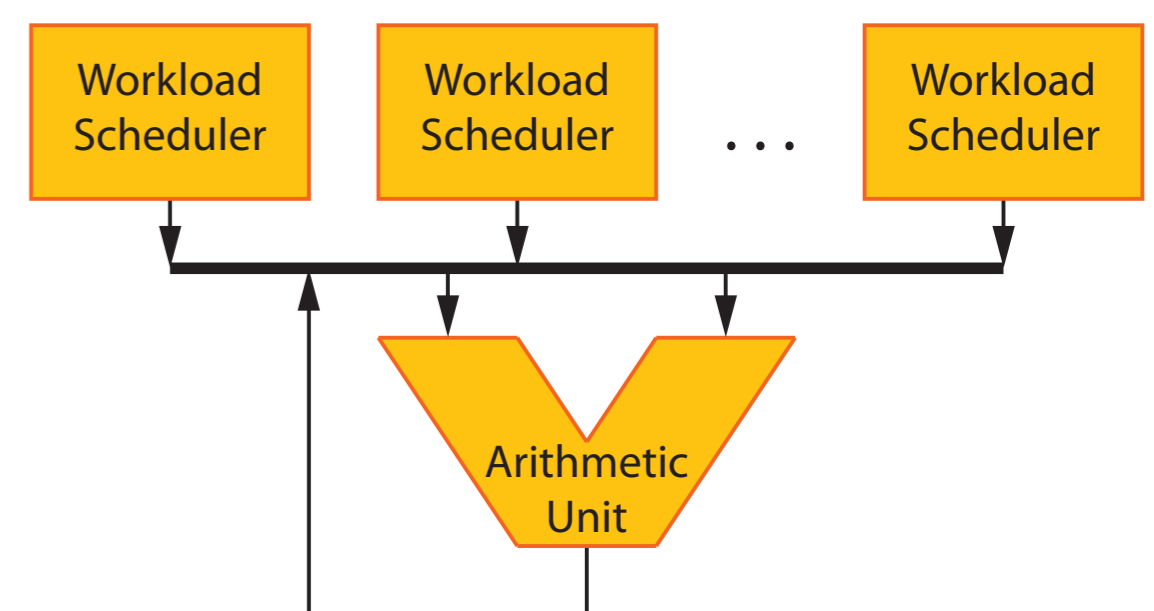
Design

We chose the architecture of a RISC CPU as the starting point for our design, because our numerical integrator demanded an adaptable architecture to facilitate a potentially limitless range of equations. Our design comprises two modular units – a workload scheduler and an arithmetic unit.

The workload scheduler decodes instructions and supplies data to the arithmetic unit. The scheduler comprises an instruction memory, instruction decoder and register file. The instruction memory stores equations in the form of instructions, which are decoded into



In the future, to improve the performance, we intend to increase the utilisation of the arithmetic unit since the algorithm consists of a chain of interdependent instructions that currently leave the arithmetic unit underutilised. We plan to connect a multitude of workload schedulers to a single arithmetic unit, which will enable good performance with minimal resource utilisation.



Acknowledgements

This research is supported by the Irish Research Council for Science, Engineering and Technology funded by the National Development Plan.

embarkinitiative

Investing in People and Ideas