

# Accelerating 3D Convolution using Streaming Architectures on FPGAs

This is a summary of a paper presented at the 79th Society of Exploration Geophysicists (SEG) Meeting, Houston, October 2009.

## About the Authors

Haohuan Fu and Bob Clapp are with the Center for Computational Earth & Environmental Science (CEES) at Stanford University.

Oskar Mencer is CEO and Oliver Pell is VP of Engineering at Maxeler Technologies.

## Summary

We investigate FPGA architectures for accelerating applications whose dominant cost is 3D convolution, such as modeling and Reverse Time Migration (RTM). We explore design options such as: (1) using different stencils; (2) fitting multiple stencil operators into the FPGA; (3) processing multiple time steps in one pass; and (4) customizing the computation precisions.

Finite-difference based convolution operators perform multiplications and additions on a number of adjacent points. From a memory perspective, however, these points are often stored far apart inducing a large number of cache misses in software implementations.

In an FPGA implementation, a memory buffer stores the points located in memory between the first and last points of the stencil operator applied to a given data item. For a 7-point 3D convolution on a  $512 \times 512 \times 512$  array, the design requires a buffer for  $512 \times 512 \times 6$  data items.

We experiment with two different 3D stencils: a 7-point star stencil (Figure 1a) and a 3-by-3-by-3 cube stencil (Figure 1b) which perform respectively an 8th and 6th order finite difference. While the cube consumes 20% more arithmetic units in the FPGA than the star, the internal-FPGA-

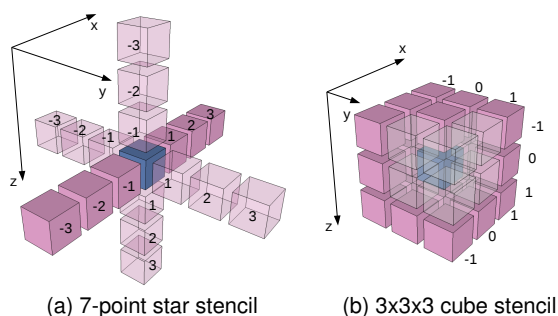


Figure 1: 2 Alternative stencil choices.

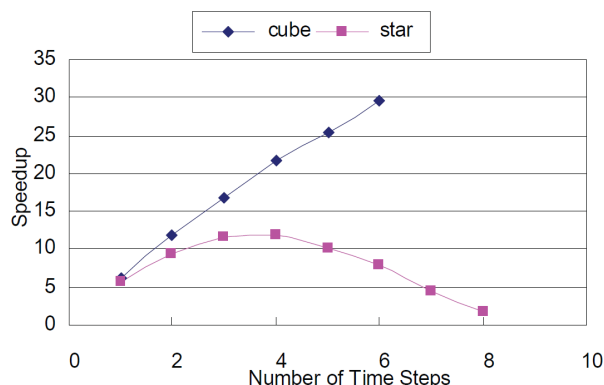


Figure 2: Speedups for processing different numbers of timesteps in each pass with each stencil type.

memory requirement of the cube is one third of the star. By exploiting the symmetry of the coefficients, the number of arithmetic operations reduces significantly for the cube.

To make a full utilization of all units on an FPGA, we have two options: (1) using multiple stencil operators to work on multiple data items in parallel; (2) processing multiple timesteps in one pass.

Increasing the number of stencil operators may not improve the performance when the input streams approach the memory bandwidth limit: both the cube and the star arrive at the saturation point of around 25x speedup with four stencil operators.

The other strategy is to process multiple timesteps in one pass, with the output of each unit as the input of the next unit. This solution will not be limited by memory bandwidth and will allow improving the order-of-time accuracy with relatively small costs. Figure 2 shows projected performance for the star and cube stencils over multiple timesteps. The cube stencil scales much better than the star stencil as the buffering requirements for the cube are a third those of the star.

We have implemented the 2nd order cube with 6 time steps onto the Maxeler MAX2 acceleration card and achieved a speedup of 29x. By using the two Virtex-5 LX330T FPGAs on a MAX2 card, we achieved up to 55x speedup in total. Projecting our design into the Virtex-6 SX475T FPGAs on the new MAX3 card, we can fit up to 13 time steps in the single FPGA on the card and achieve up to 110x speedup compared to a single-core CPU version.