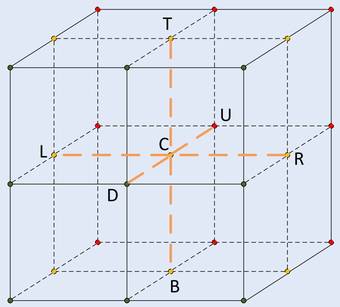
Accelerating 3D-Structured Grid on FPGA via OpenCL : A Case Study with OpenDwarfs

Anshuman Verma* and Wu-chun Feng* †
*Department of Electrical and Computer Engineering and †Department of Computer Science
Virginia Tech, Blacksburg, VA, USA

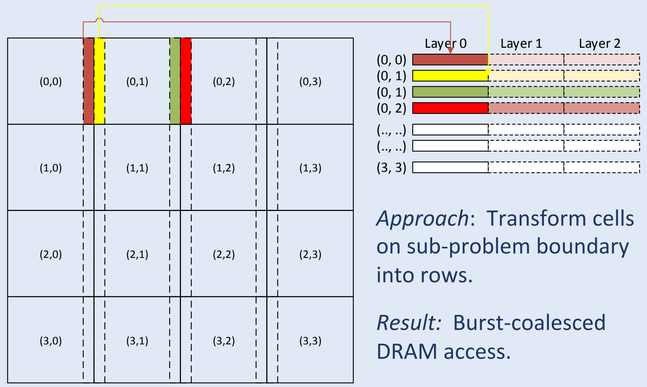
SyNeRGy
http://synergy.cs.vt.edu/

Introduction and Background

- **What?** ***3D Structured Grid (e.g.,. Stencil)***
  - A fundamental computation/communication pattern in structured grid algorithms.
- **How?**
  - Solving of partial differential equations (PDEs) in a 3D grid by applying a 7-point finite-difference algorithm
- **Characteristics?**
  - A memory-bound & low computational-intensity algorithm.
- **Goals**
  - Design of an FPGA accelerator for 3D structured-grid computation using OpenCL with different configuration parameters, e.g., different double-word plane sizes & variable height.
  - Performance evaluation of our approach with respect to a microkernel-based roofline model.



Fragmentation and Coalescing



Approach: Transform cells on sub-problem boundary into rows.

Result: Burst-coalesced DRAM access.

Dividing problem sizes larger than the stencil accelerator's plane size into smaller sub-problems

Experimental Set-Up

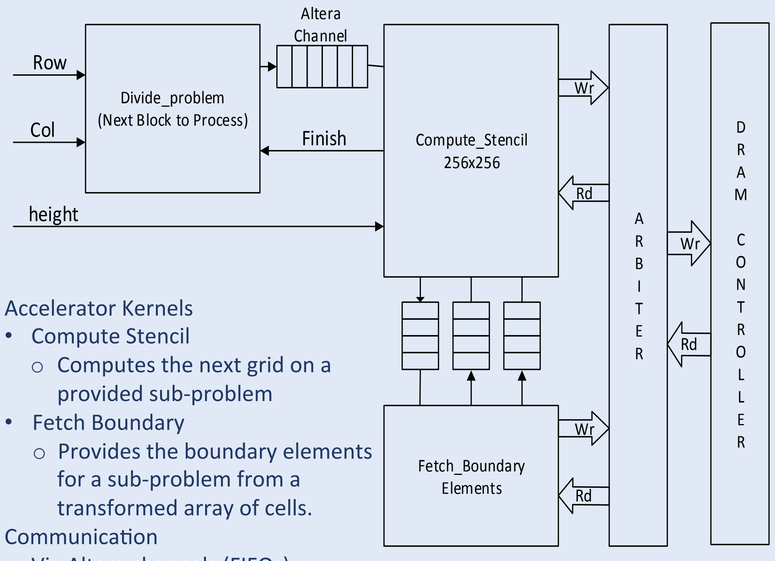
FPGA: Altera Stratix V (Bittware S5-PCIe-Hq-D8)
Tool: Altera OpenCL SDK V16.0 (Offline compiler)
Methodology:

- Performance evaluation with respect to a microkernel-based roofline model.

Problem: 3D grid of double-precision floating-point cells with each side being of 128 (p_128_c), 256 (p_256_c), or 512 (p_512_c)

Accelerator	Description
Acc_64x64	Plane size 64x64
Acc_128x128	Plane size 128x128
Acc_256x256	Plane size 256x256
Acc_nv	Naïve single work-item kernel

OpenCL Accelerator



Accelerator Kernels

- Compute Stencil
  - Computes the next grid on a provided sub-problem
- Fetch Boundary
  - Provides the boundary elements for a sub-problem from a transformed array of cells.

Communication

- Via Altera channels (FIFOs)

Kernels execute the stencil computation and fetch boundary elements as persistent kernels, i.e., they run forever.

On-Chip Memory Management

Data flow in on-chip memory for two consecutive fetch cycles.

- On-chip memory buffer consists of two plane memories (P0, P1) that are constituted from eight different memories. Size of P0 for the accelerator of a plane size A*A → (A+2)*A*sizeof(cell).
- Top two rows of the current computation layer are stored in a cyclic buffer constituted from two different memories (CB0, CB1), each of size A*sizeof(cell).
- This configuration requires at most two read-ports and one write port for each memory block.

Performance and Area Usage