INTERNATIONAL CAMPUS OF EXCELLENCE

From Dataflow Specifications to Customised Reconfigurable Datapaths Using HLS: the OpenCL Case for FPGAs

Rubén Salvador

[Kindly hosted by INSA: KDesnos, MPelcat, JFNezan, DMenard, LMorin…]

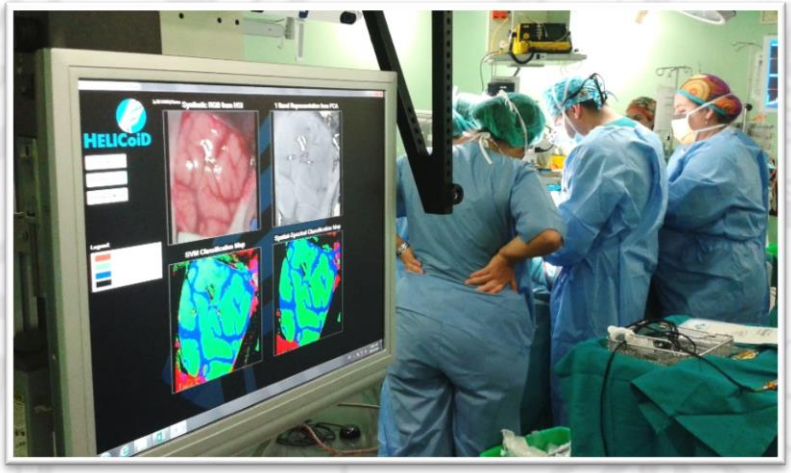
Universidad Politécnica de Madrid (UPM)
School of Telecommunications Systems and Engineering (ETSIST)
Research Center on Software Technologies and Multimedia Systems (CITSEM)

Dataflow Workshop
Rennes, 12-14 December 2017

CERBERO

Horizon 2020
European Union funding
for Research & Innovation

Context



HELICoiD

Co-funded by
the European Union

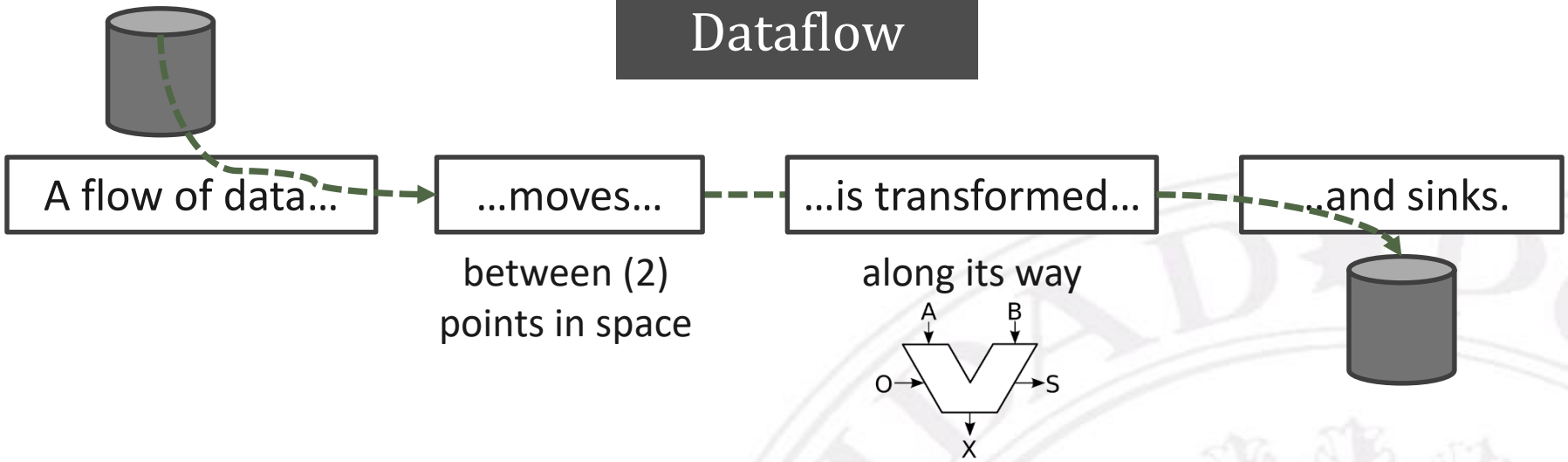
OUTLINE

- Motivation
- OpenCL FPGA
- Dataflow Mapping On Top Of OpenCL FPGA
- Implementation Steps

OUTLINE

- Motivation
- OpenCL FPGA
- Dataflow Mapping On Top Of OpenCL FPGA
- Implementation Steps

Dataflow

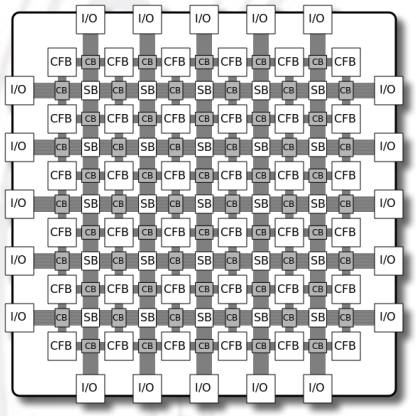


Spatial Computing

hardware

FPGA

point to point comms



datapath

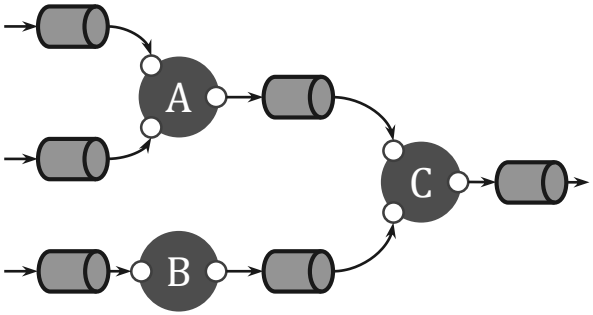
on-chip memory

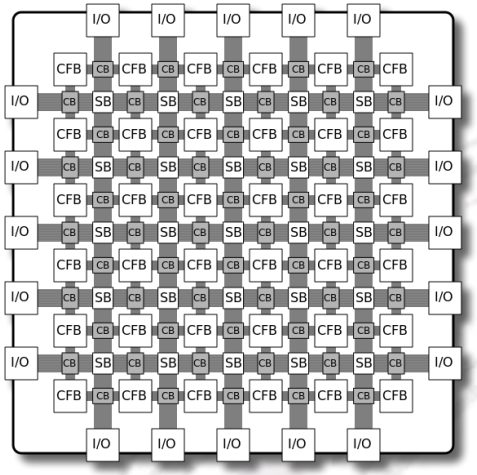
architecture

Dataflow

FPGA

HLS





OpenCL

Wide Community Embrace

System Level Integration

(SW) developers love

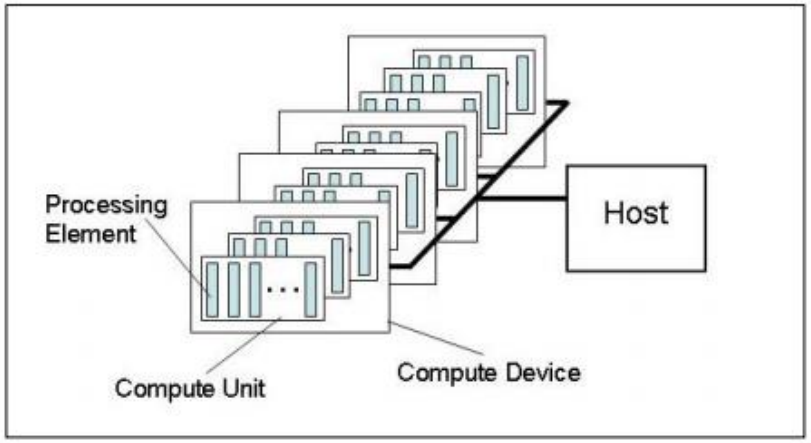
What can dataflow bring to the OpenCL community?

	OpenCL	Dataflow
Pros	• Functionally portable • Wide community acceptance • Support for HLS • …	• Graph analysis & Guarantees • Schedulability, deadlocks, FIFO sizing • Concurrent execution model • Comms interaction

Cons	• No dataflow (streaming) friendly
	  • Global memory comms
	• Compute accelerator model
	  • Data offload (writes/reads)
	  • Throughput oriented (vs latency)
		• Niche domain
		• Most work for multi/manycore

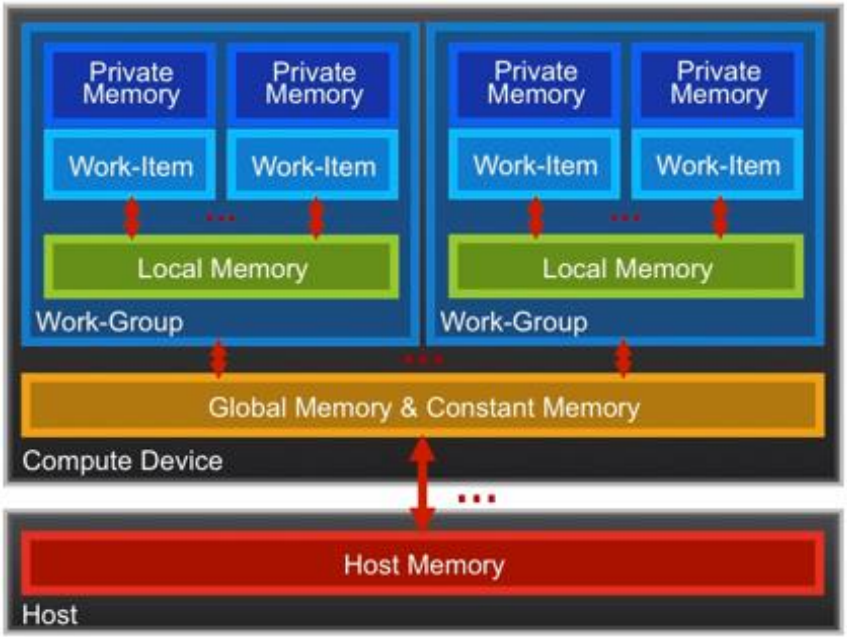
OUTLINE

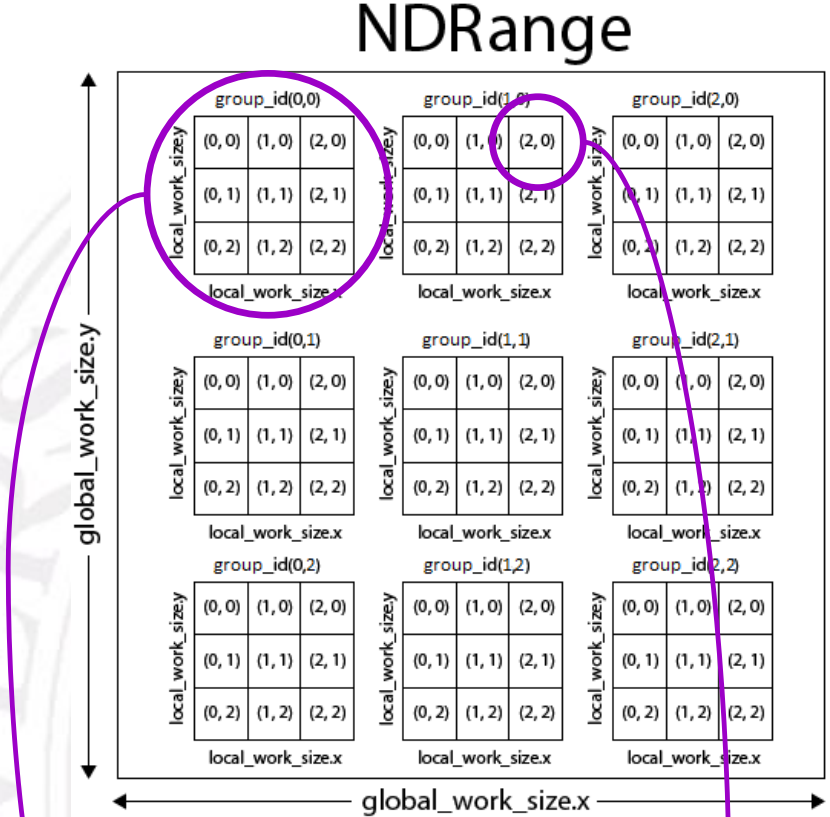
- Motivation
- **OpenCL FPGA**
- Dataflow Mapping On Top Of OpenCL FPGA
- Implementation Steps



Task parallelism

Data parallelism SIMD

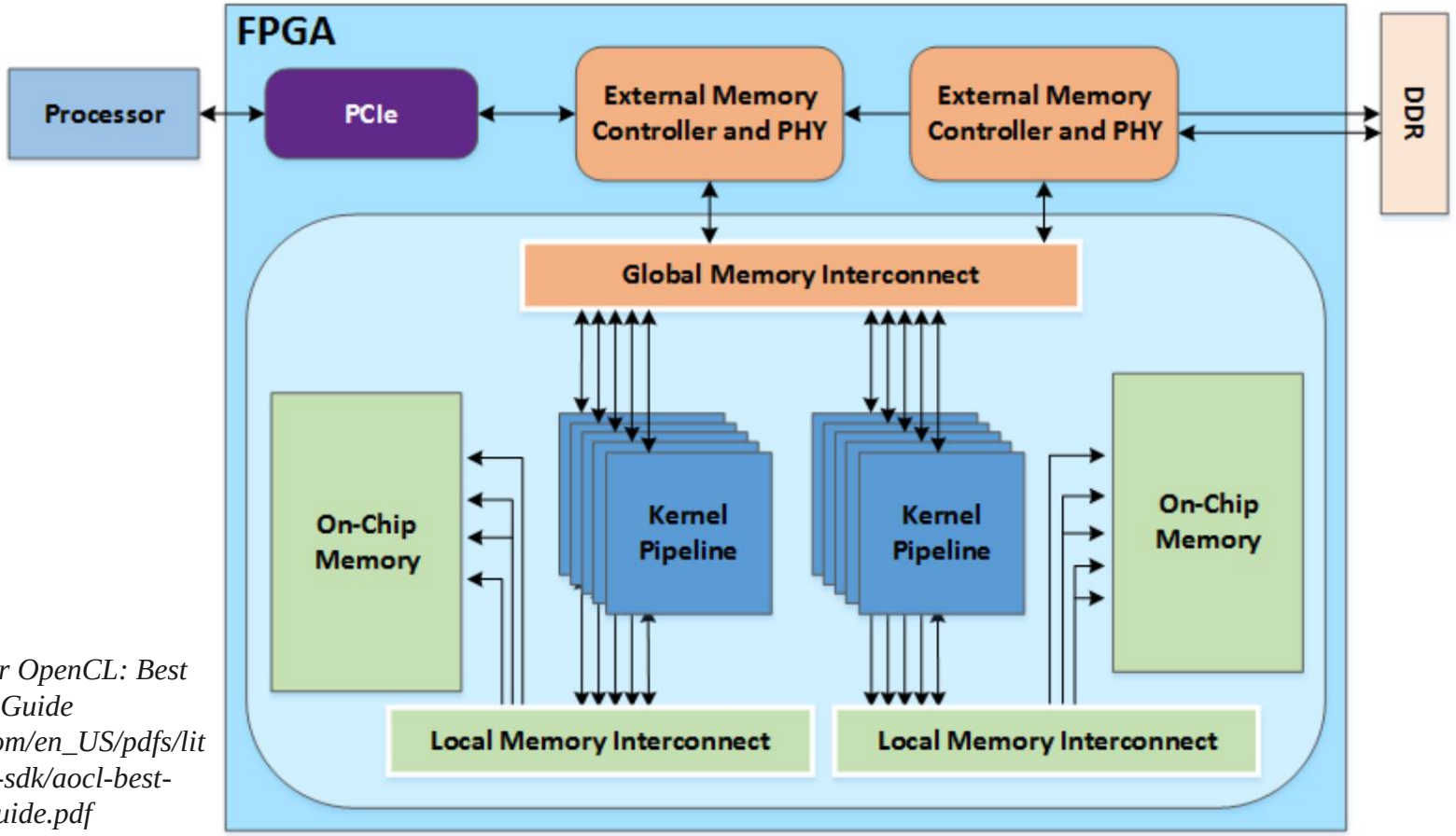




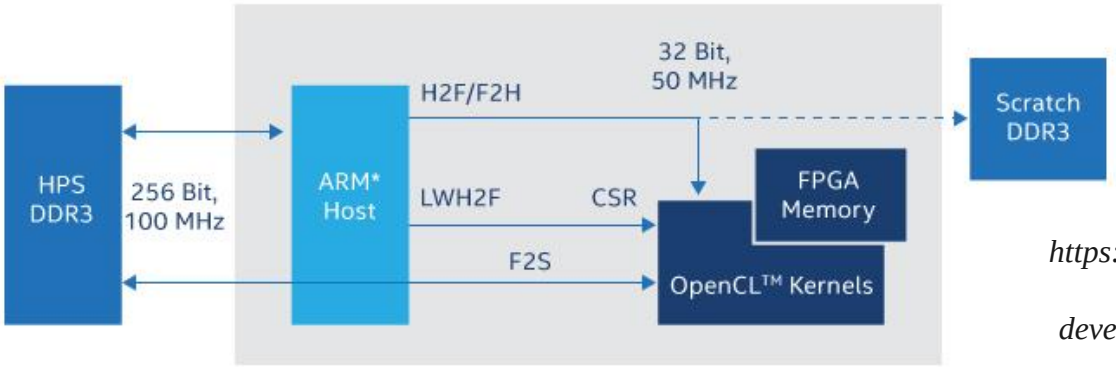
Work Group (WG)

Work Item (WI)

OpenCL FPGA Model



Intel FPGA SDK for OpenCL: Best Practices Guide
https://www.altera.com/en_US/pdfs/literature/hb/opencl-sdk/aocl-best-practices-guide.pdf



https://www.altera.com/products/design-software/embedded-software-developers/opencl/developer-zone.html

OUTLINE

- Motivation
- OpenCL FPGA
- **Dataflow Mapping On Top Of OpenCL FPGA**
- Implementation Steps

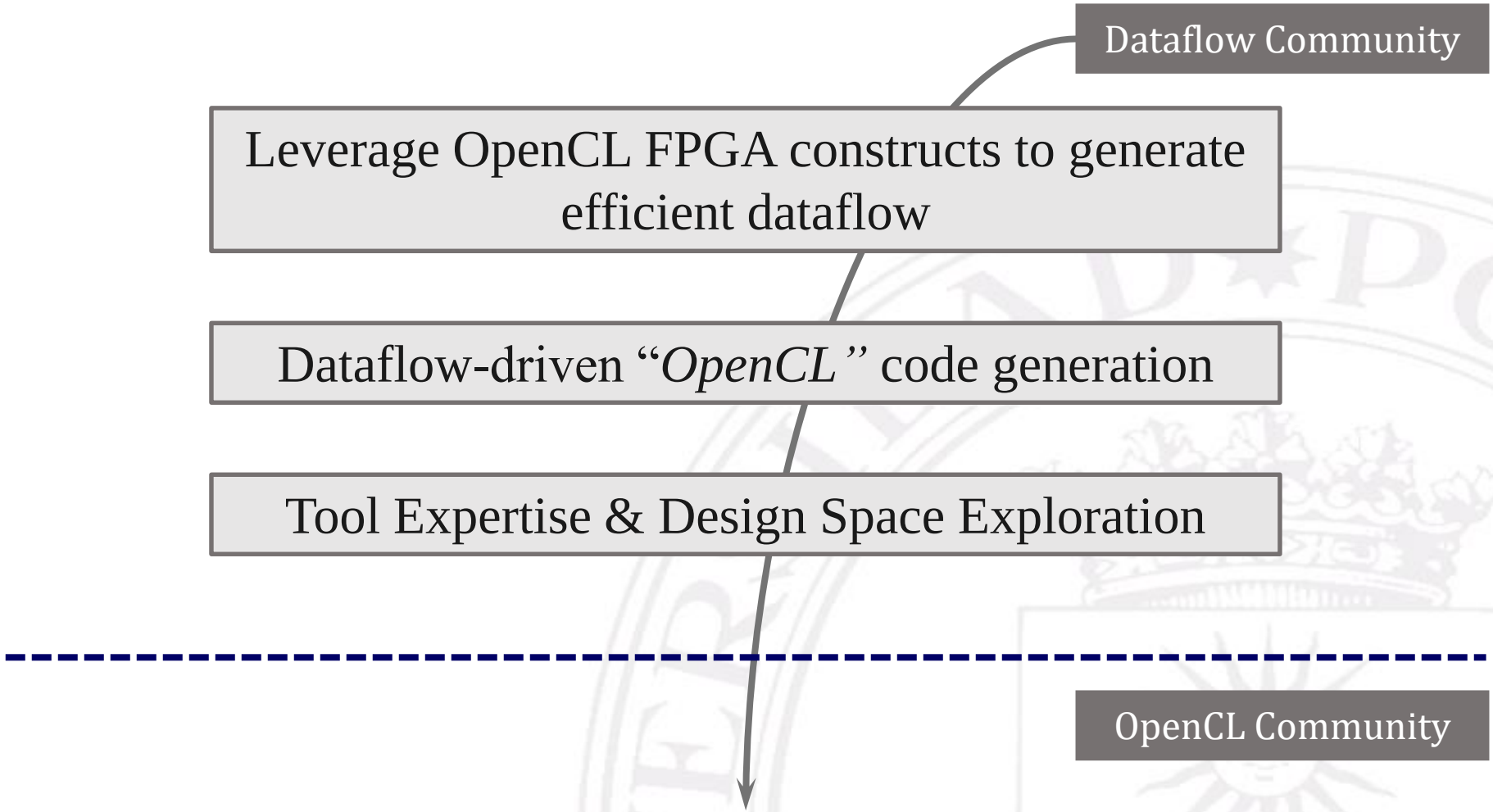
Desired Features & Expected Gains

Hardware acceleration (custom datapath)

Reduced processor (communication) overhead

Reduced memory transactions

Self-timed Execution



OpenCL Khronos Group Standard

Recent (2017) proposal: add dataflow semantics to OpenCL standard

Applying Models of Computation to OpenCL Pipes for FPGA Computing

Nachiket Kapre
University of Waterloo
200 University Ave W.
Waterloo, Ontario, Canada N2L 3G1
nachiket@uwaterloo.ca

Hiren Patel
University of Waterloo
200 University Ave W.
Waterloo, Ontario, Canada N2L 3G1
hiren.patel@uwaterloo.ca

ABSTRACT

OpenCL pipes offer a powerful construct for synthesizing multi-kernel FPGA applications with inter-kernel communication dependencies. The communication discipline between the FPGA kernels is restricted to producer-consumer style patterns supported with on-chip FPGA FIFOs. While this provides few restrictions on the compared to the alternatives. They can also be reconfigured as needed to support varying demands of user applications. However, FPGAs have traditionally been difficult to configure as programmers describe their applications as low-level *circuits* rather than high-level software programs. However, at a fundamental level, hardware circuits are specialized parallel programs. This means,

MoCs semantics to OpenCL

Kapre, Nachiket, and Hiren Patel. *Applying Models of Computation to OpenCL Pipes for FPGA Computing*. Proc. 5th **IWOCL**. ACM, **2017**.

OpenCL compute model + MoC Comms Schemes

Proposal for the OpenCL Standard

a.k.a.: compiler's job

Synchronous Dataflow (SDF)

Bulk Synchronous Parallel (BSP)

OpenCL Increasing Streaming Support

Pipes (OpenCL 2.0)

- Standard OpenCL Kernel-to-Kernel communication
- Overlap multi-kernel operation

Channels (Intel FPGA)

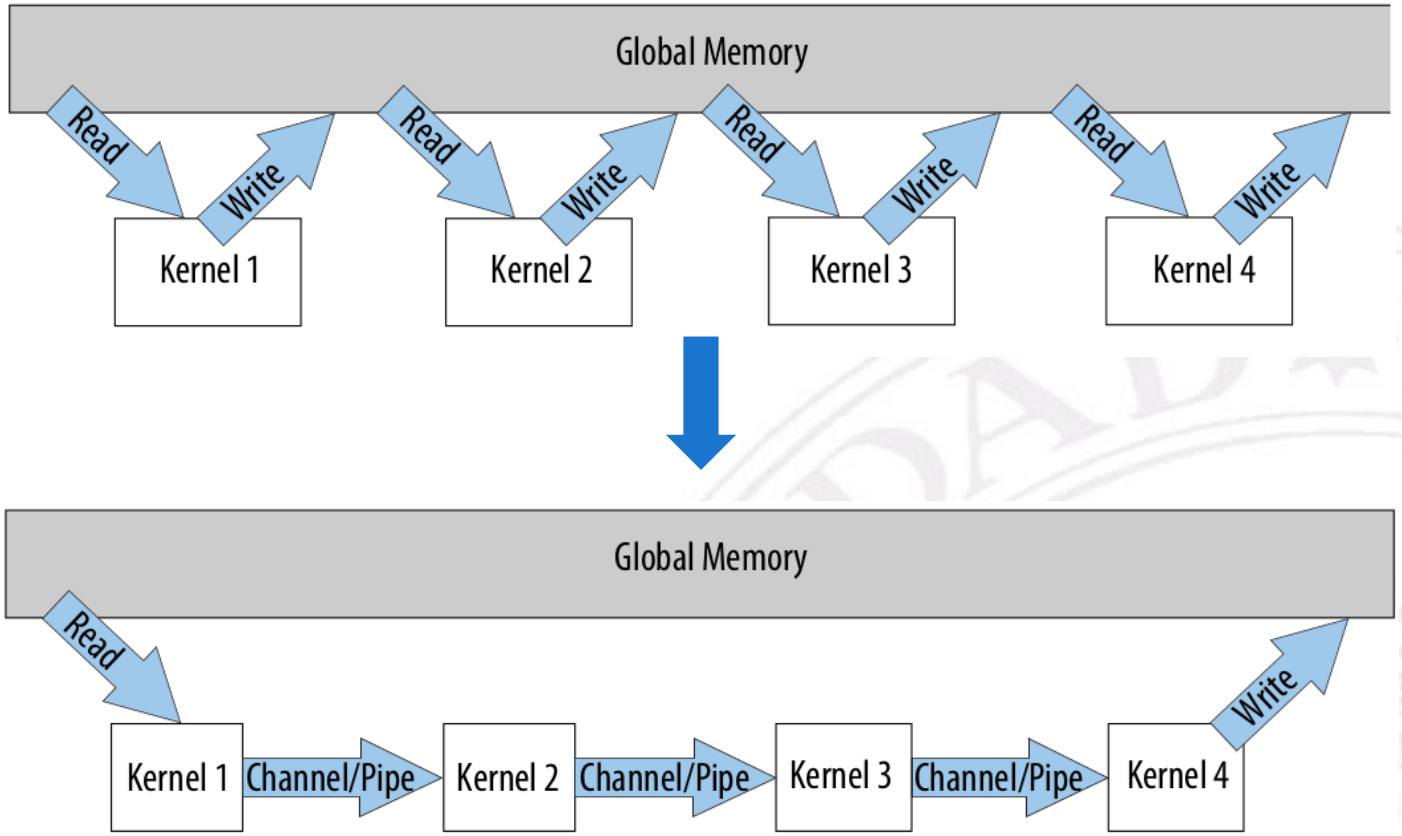
- Preferred Kernel-to-Kernel communication
- Self-triggered kernels (free run decoupled from host)

Host-Kernel Pipes

Kang, K., and P. Yiannacouras. *Host Pipes: Direct Streaming Interface Between OpenCL Host and Kernel*. Proc. 5th **IWOCL**. ACM, **2017**.

- … only prototype demo so far

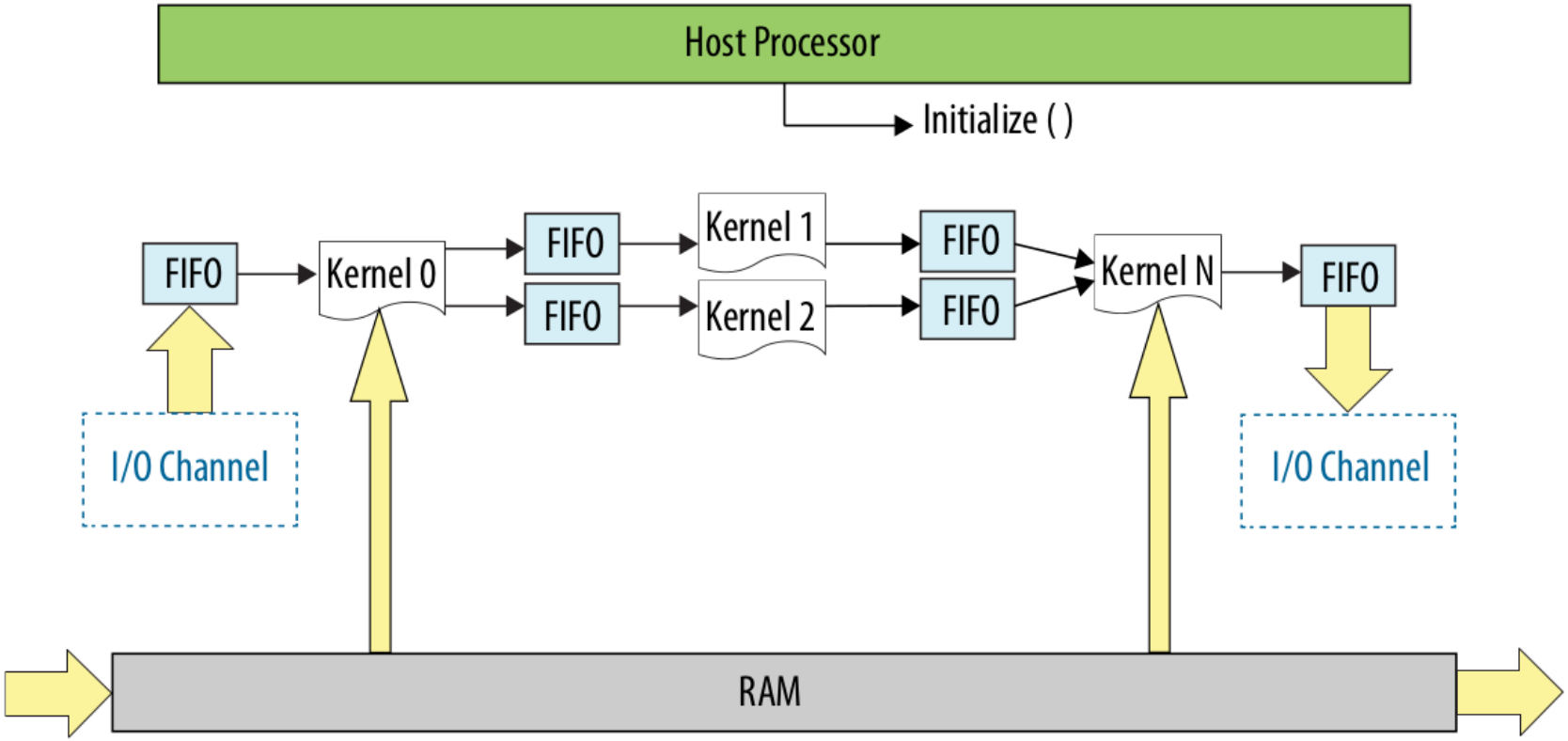
Kernel Operation Possibilities



Autorun kernels

- No host-kernel communication logic
- Autostart & Auto-restart
- Communicate through channels

Intel FPGA SDK for OpenCL: Best Practices Guide
https://www.altera.com/en_US/pdfs/literature/hb/opencl-sdk/aocl-best-practices-guide.pdf



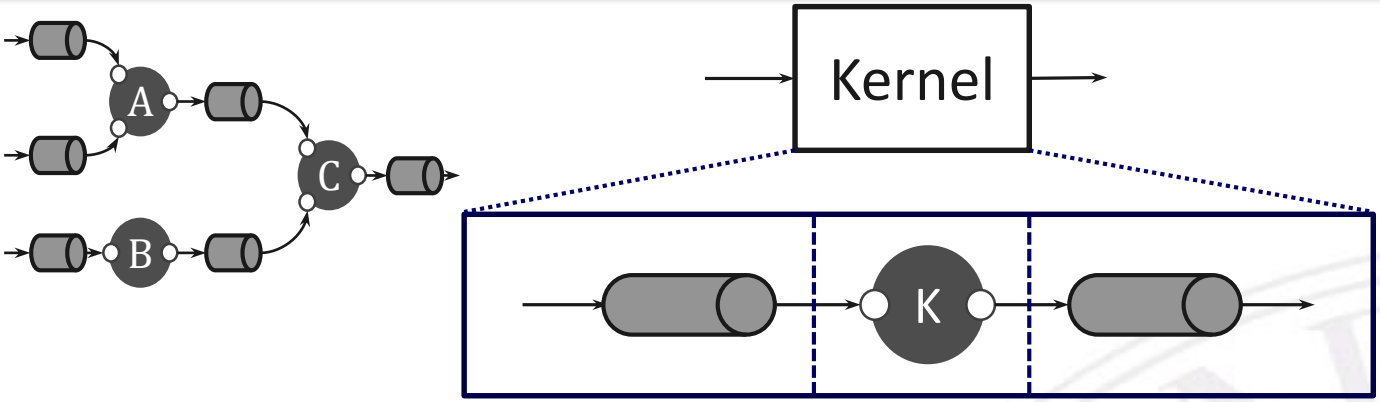
Channels

- Kernel execution decoupled from host
- Blocking/Non-blocking Read/Write API
- Synchronization mechanisms
- I/O Channels -> Streaming DSP

Intel FPGA SDK for OpenCL: Programming Guide
https://www.altera.com/content/dam/altera-www/global/en_US/pdfs/literature/hb/opencl-sdk/aocl_programming_guide.pdf

OUTLINE

- Motivation
- OpenCL FPGA
- Dataflow Mapping On Top Of OpenCL FPGA
- Implementation Steps



1.- Actor Firing Rules within OpenCL Kernels

Check overhead vs. performance

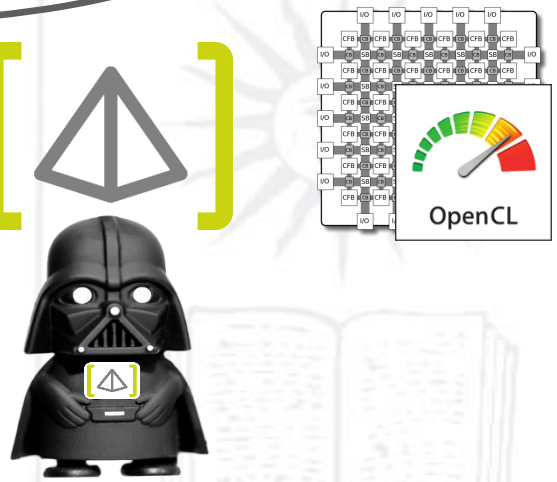
Acceptable?

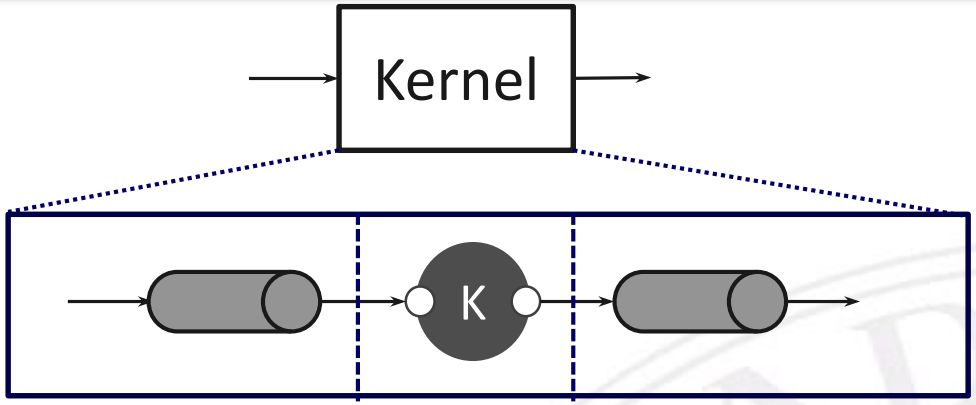
2.- Code Generatrion from PREESM

2.1.- Actor firing rules - scheduler

2.2.- FIFO analysis & Buffer generation

2.3.- Memory Access Optimization

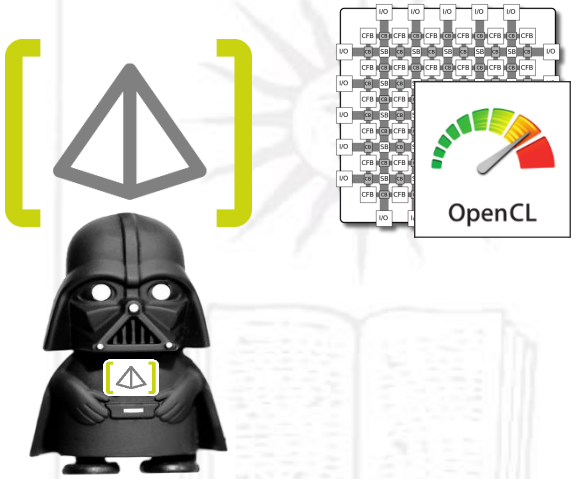


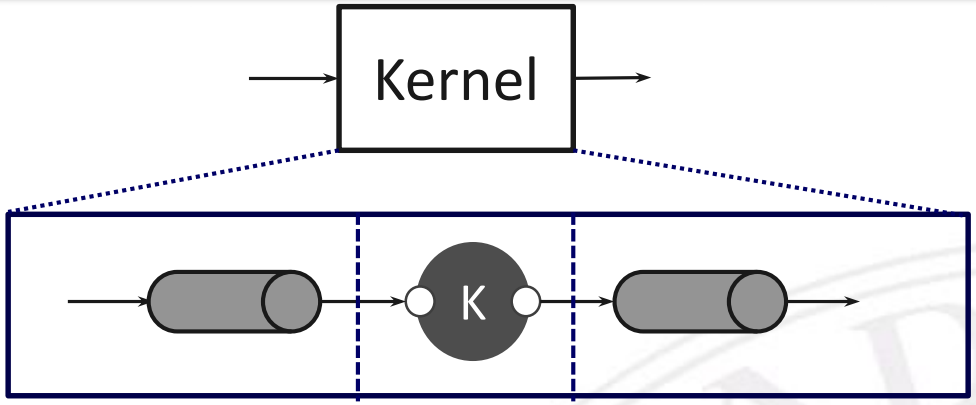


2.1.- Actor firing rules (scheduler)

Actor I/O IFs, firing rules, templates → Enough with channels sync? Borrow from *CAPH* ¿?

- **Host** code:
  - Platform initialization: *automatic*
  - Job management: *automatic*
    - input data & result data
    - "only" necessary for the host/device frontier
    - pointers mapped to device buffers
- **Kernel** code:
  - I/O interfaces (firing rules): *automatic*
  - Functionality: *manual* (provided by user)





2.1.- Actor firing rules (scheduler)

Actor I/O IFs, firing rules, templates

→ Enough with channels sync?

Borrow from *CAPH* ¿?

2.2.- Buffer generation

Leverage current PREESM buffer generation

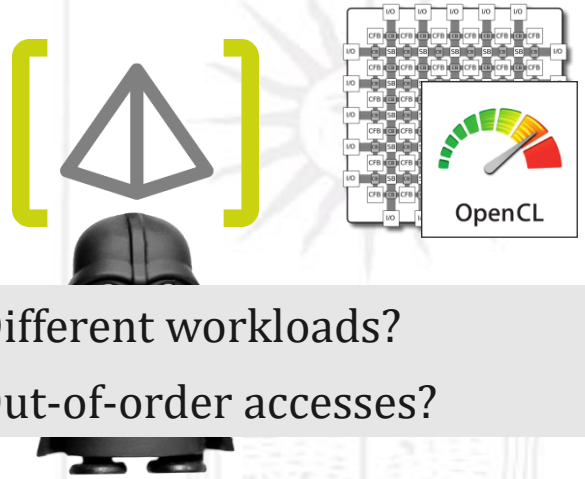
Pipes vs Channels vs Ad-hoc Buffer

2.3.- Memory Accesses Optimization

Streaming Dataflow

Shared/Global Memory

Local FPGA DDRs (kernel only)



Different workloads?

Out-of-order accesses?

3.- Hack the Flow

Kernel (actor) functionality → Component library

Plug HDL / CAPH

DSE: Predictability → Area / Latency / Throughput

Host-Device communications → Device Wrapper

DSE: Predictability → Compute upper bound?

Open Run Time ?¿ → pocl

Dynamic Reconfiguration ?¿ → Graph Reconfiguration

4.- New devices

Intel Xeon + FPGA → HPC community

VAADER
IETR research team

INSA RENNES

IETR
INSTITUT D'ÉLECTRONIQUE ET DE TÉLÉCOMMUNICATIONS DE RENNES

citSem
POLITÉCNICA
Centro de Investigación en Tecnologías del Software y Sistemas Multimedia para la Sostenibilidad

Thanks for your attention!!

ruben.salvador@upm.es
https://twitter.com/RubenSalvadorP
http://blogs.upm.es/rubensalvador/