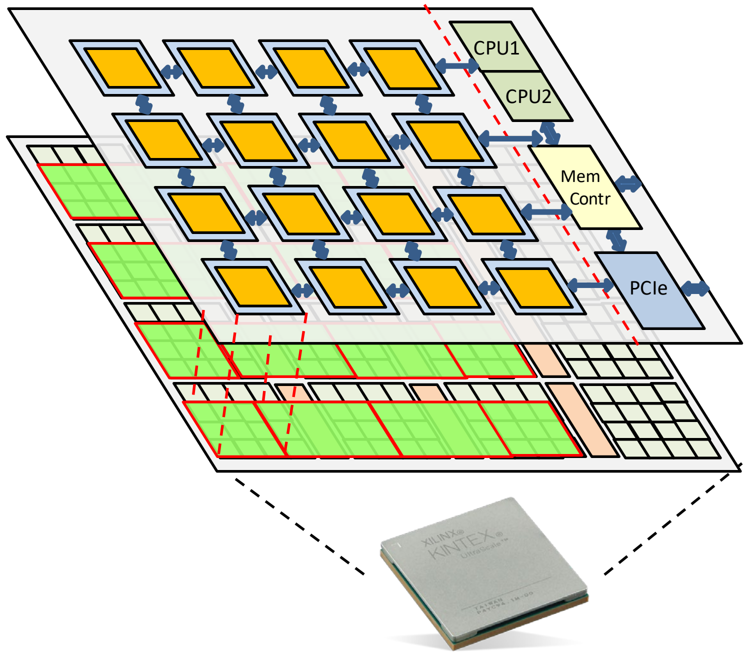
A Template for a Runtime Reconfigurable FPGA Overlay Architecture Targeting Dataflow Applications



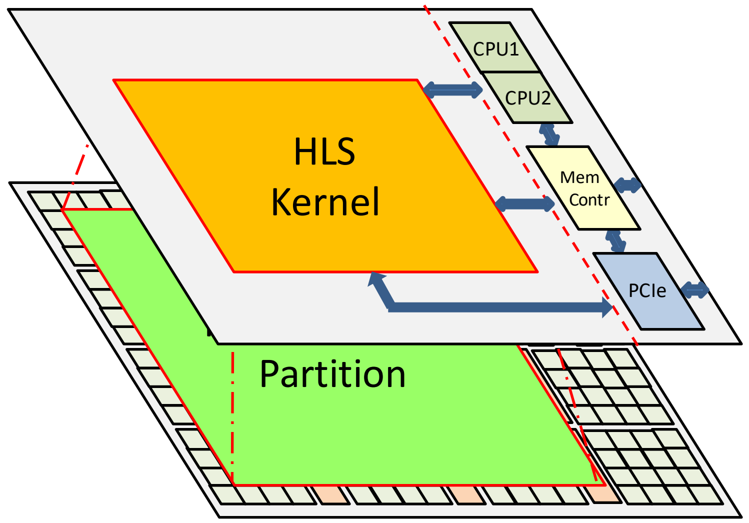
Professor Dr.-Ing. Thilo Pionteck
Chair Hardware-Oriented Technical Computer Science

Outline

1.: Motivation

Limitations of current design flows

- HDLs (Verilog, VHDL)
  - Long development times
  - Long synthesis times
- HLS (C, C++, OpenCL, oneAPI)
  - Reduced development times
    Limitations: HLS targets large, single-use kernels
  - Infrastructure provided by a shell
  - (still) long synthesis times



1.: Motivation

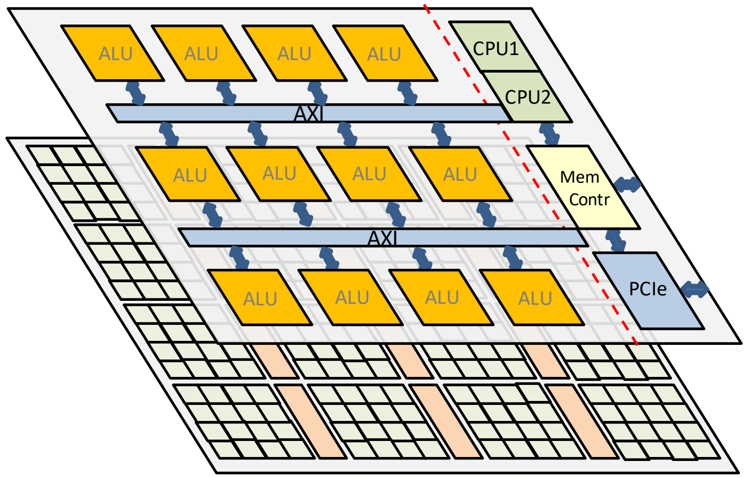
Overlay Architectures

- Abstraction of raw programming interface of FPGAs
- Structuring of hardware resources and partitioning into set of tiles
- Provides infrastructure
  - Interfaces to external memory and PCIe
  - Communication architecture between tiles

1.: Motivation

Overlay Architectures

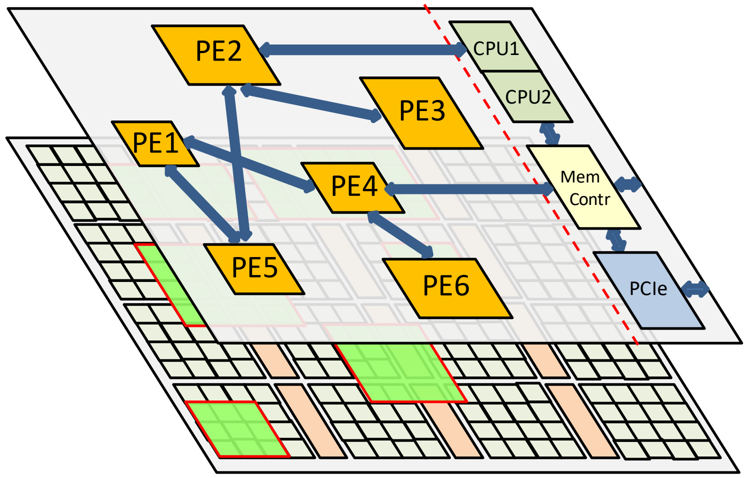
- Abstraction of raw programming interface of FPGAs
- Structuring of hardware resources and partitioning into set of tiles
- Provides infrastructure
  - Interfaces to external memory and PCIe
  - Communication architecture between tiles
- Categories:
  - Static overlay architectures (coarse grained ALUs)



1.: Motivation

Overlay Architectures

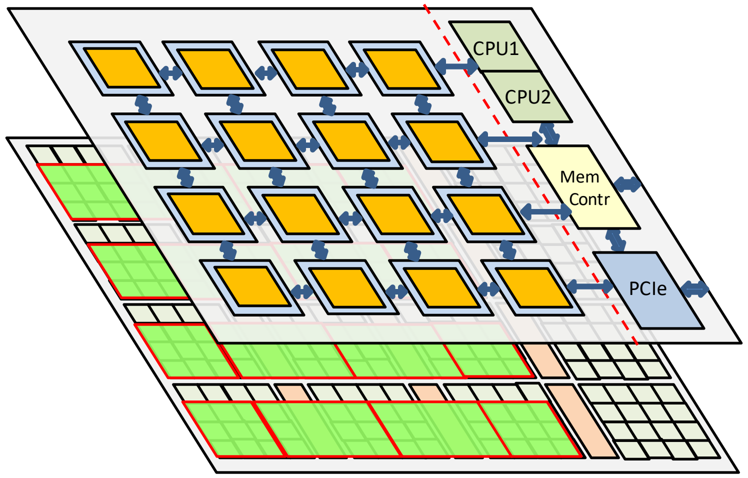
- Abstraction of raw programming interface of FPGAs
- Structuring of hardware resources and partitioning into set of tiles
- Provides infrastructure
  - Interfaces to external memory and PCIe
  - Communication architecture between tiles
- Categories:
  - Static overlay architectures (coarse grained ALUs)
  - Dynamic overlays for specific applications



1.: Motivation

Overlay Architectures

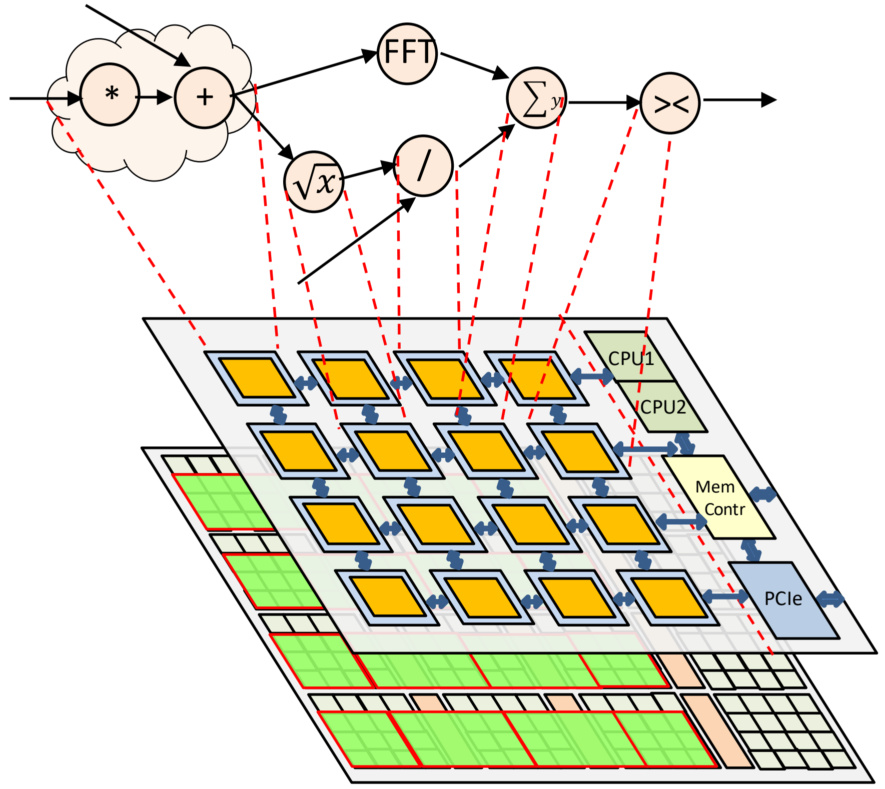
- Abstraction of raw programming interface of FPGAs
- Structuring of hardware resources and partitioning into set of tiles
- Provides infrastructure
  - Interfaces to external memory and PCIe
  - Communication architecture between tiles
- Categories:
  - Static overlay architectures (coarse grained ALUs)
  - Dynamic overlays for specific applications
  - Dynamic overlays with regular infrastructure for wide range of applications



1.: Motivation

Dynamic overlays with regular infrastructure

- Well suited for dataflow applications
- Application has to be partitioned into set of small tasks → dataflow graphs
  - Nodes (tasks) are mapped to tiles
  - Edges represent communication between nodes
- Dynamic reconfiguration allows to exchange tiles (tasks)
  - Overlay is static
- Disadvantages:
  Depending on applications, communication requirements between tasks is different
  - Requires different overlay architectures for different applications
  - Design is again time consuming



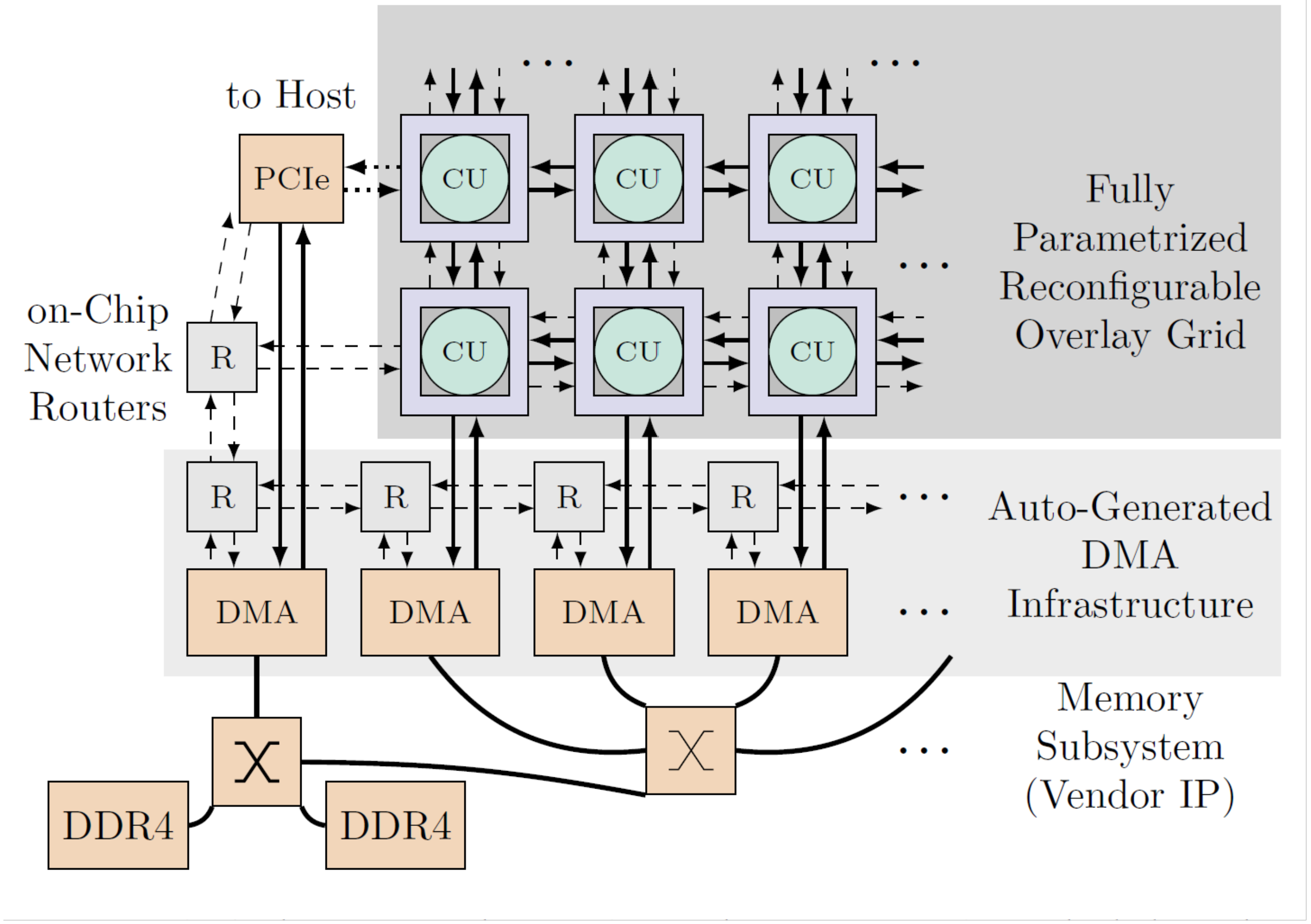
Idea: Template for domain specific overlay architectures

Outline

2.: Template for Overlay Architectures

Overlay Template

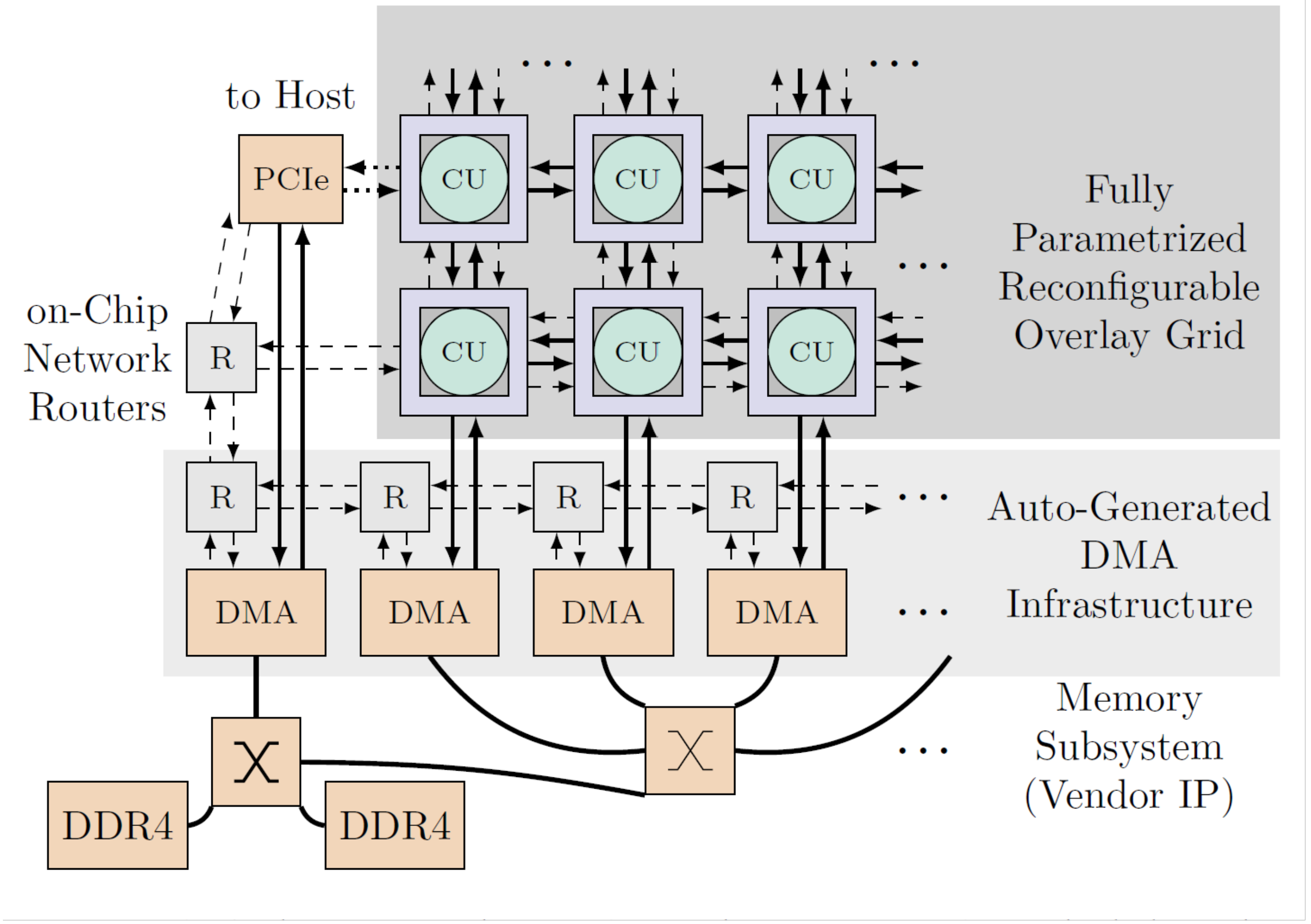
- Written in VHDL
- AXI stream interfaces
  support HDL, HSL, IP cores as kernels
- Template parameters (deployment time)
  - Number and size of tiles
  - Internal communication resources
  - External interfaces
  - Static hardware accelerators
- Runtime parameters
  - Kernels to be implemented
  - Communication paths



2.: Template for Overlay Architectures

Communication Infrastructure

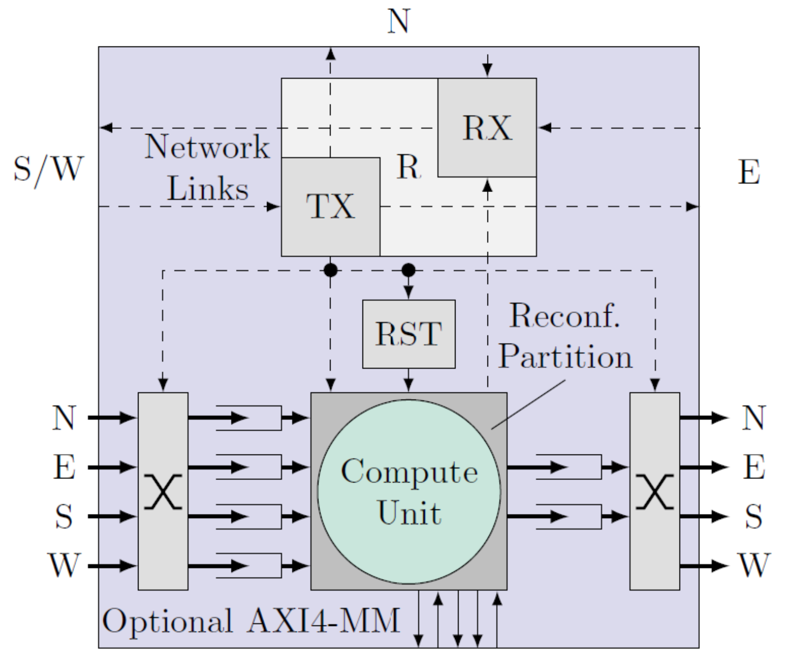
- Topology is defined at design time
  From 2D feed-forward to 8-neighborhood
- DMA engines for feeding in streams
- Lightweight packed-based NoC
  - Small data tansfers
  - Bases on AXI stream
  - Used to
    - Configure DMAs, Tiles, CUs
    - Transmit error codes
    - Transmit small results
  - Host-to-overlay, overlay to host

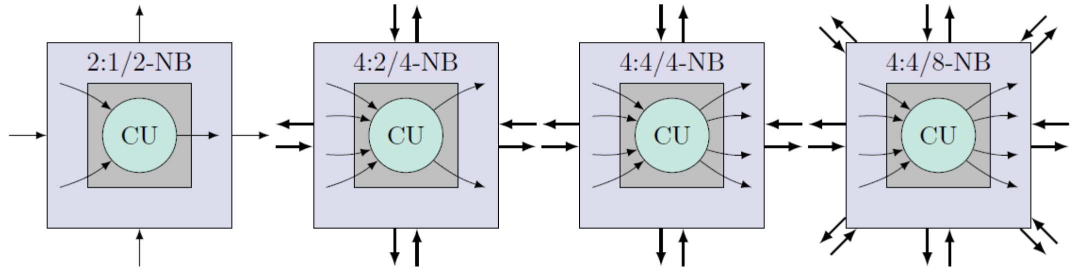

to Host
PCIe
CU
CU
CU
CU
CU
CU
Fully Parametrized Reconfigurable Overlay Grid
on-Chip Network Routers
R
R
R
R
R
DMA
DMA
DMA
DMA
Auto-Generated DMA Infrastructure
X
X
DDR4
DDR4
Memory Subsystem (Vendor IP)

2.: Template for Overlay Architectures

Tile Architectrue

- Consists of
  - Communication ressources
  - Reconfigurable partition
  - Compute Unit (CU) (supports dynamic reconfiguration)
  - Local reset controler
  - Optional: AXI memory-mapped interface
  - On-chip network
- AXI crossbars allow data stream to be routed freely
- CU interface is AXI stream complient
- Predefined topology configurations
- Buffers
  - Break timing paths
  - Registers or FIFOs

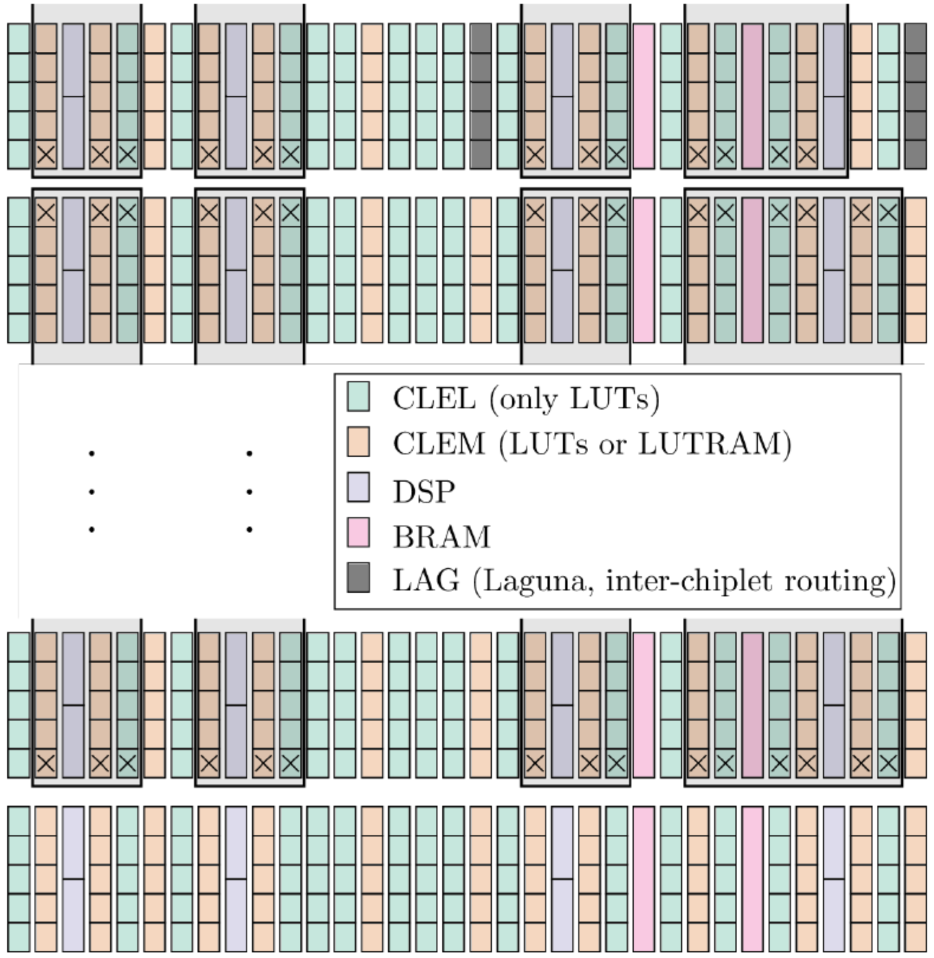




2.: Template for Overlay Architectures

Design Considerations

- “right“ size of the reconfigurable partition
- Spacing between tiles
- Spacing next to inter-chiplet links (LAG)
- Positioning of DMA engines
- Parallelisation within CU (SIMD)
- Custom floor planning tool on top of RapidWright



Alevo U280

Outline

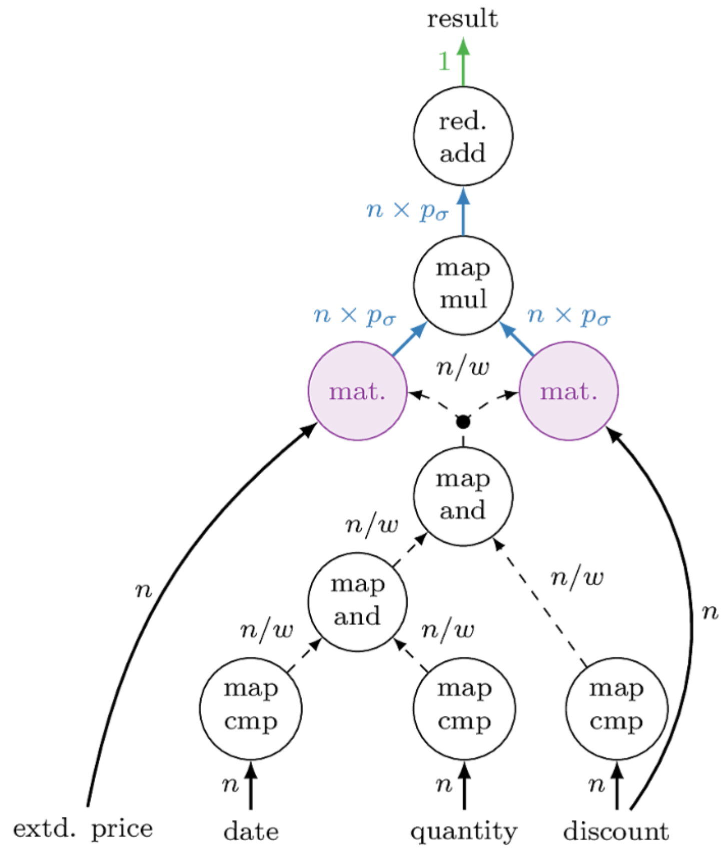
3.: Domain-Specific Prototype

Domain: Database Queries

- Queries may change at runtime
- Fixed set of database operators

```
SELECT
            sum (l_extendedprice * l_discount) as
revenue
FROM
            lineitem
WHERE
            l_shipdata >= date'[Date]'
      AND l_shipdate < date '[DATE]' + interval '1' year
      AND l_discount between [DISCOUNT] - 0.01
      AND [DISCOUNT] + 0.01
      AND l_quantity < [QUANTITY];
```

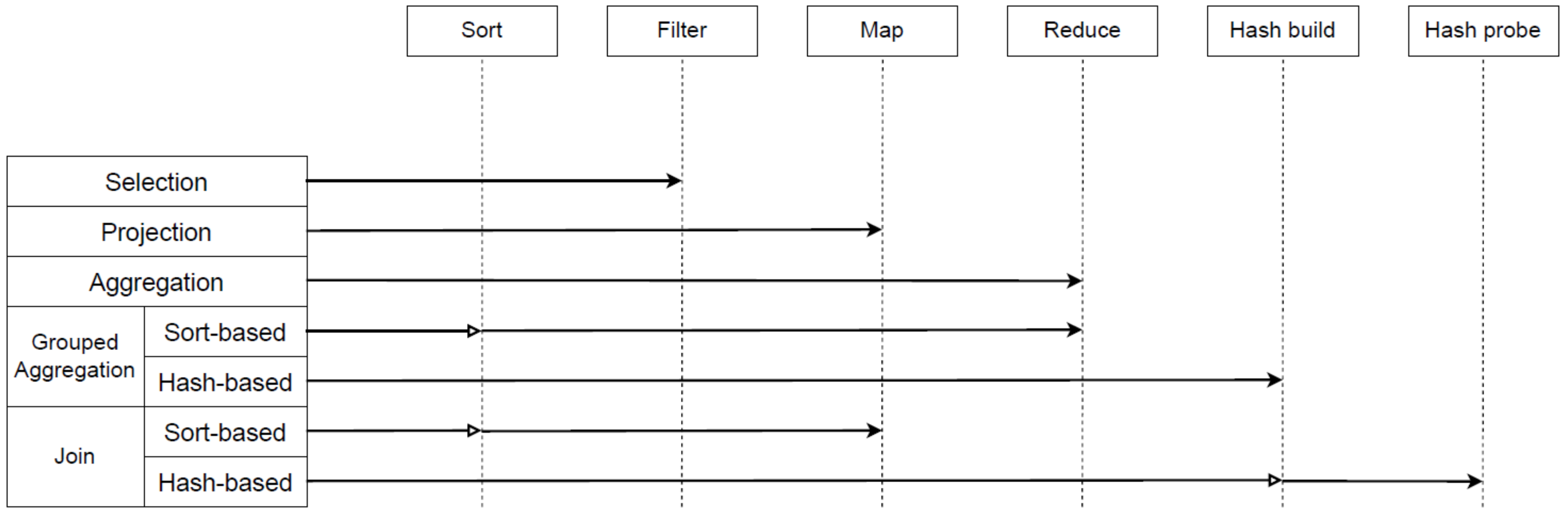
TPC-H Q6



TPC-H Q6

3.: Domain-Specific Prototype

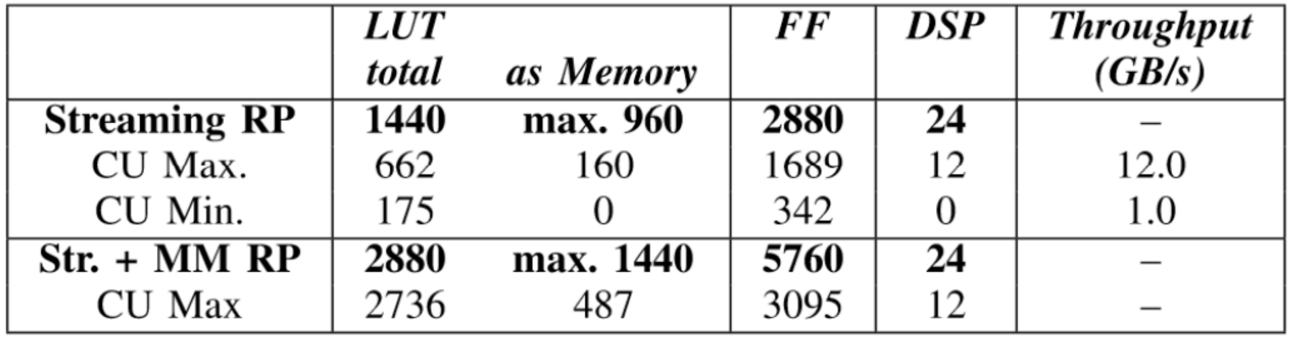
- Challenge: database operators extremely vary in complexity
  → “right” size for reconfigurable partition?
- Idea: partition database operators into set of basic functions
  → Primitives
- TPC-H: 31 primitives



3.: Domain-Specific Prototype

Configuration

- 4-Neighbour topology
- Streams: 4 input, 2 output
- 128 bits (4 SIMD)
- 11x4 tiles
- XILLYBUS (1.6 GB/s)
- 11 AXI Datamover IPs
- 2 DDR4 controllers with 512 bits *(theoretical maximum of 8 parallel data streams)*
- Size of Compute Unit
  - Largest primitive: data-parallel ordered aggregation
  - Smallest primitive:
    - count operation
    - Fifo when tile is not used

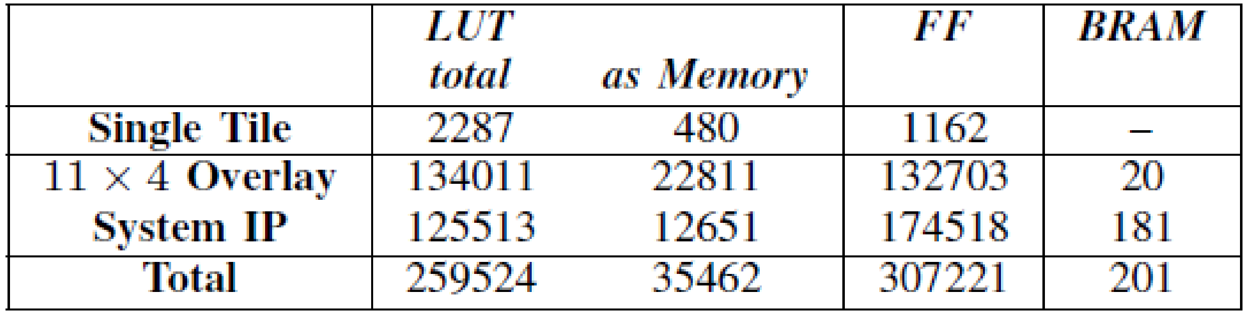


	LUT		*FF*	*DSP*	*Throughput*
	total	*as Memory*			*(GB/s)*
Streaming RP	**1440**	**max. 960**	**2880**	**24**	–
CU Max.	662	160	1689	12	12.0
CU Min.	175	0	342	0	1.0
Str. + MM RP	**2880**	**max. 1440**	**5760**	**24**	–
CU Max	2736	487	3095	12	–

3.: Domain-Specific Prototype

Results (Alveo U280)

- 250 MHz
- Synthesis results include dummy CUs



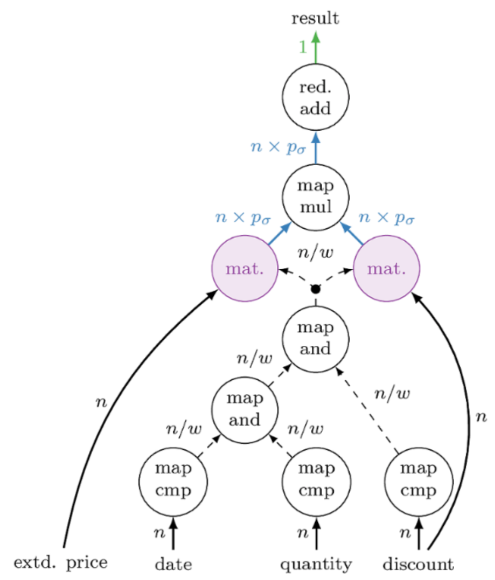
	LUT		*FF*	*BRAM*
	total	*as Memory*		
Single Tile	2287	480	1162	–
11 × 4 Overlay	134011	22811	132703	20
System IP	125513	12651	174518	181
Total	259524	35462	307221	201

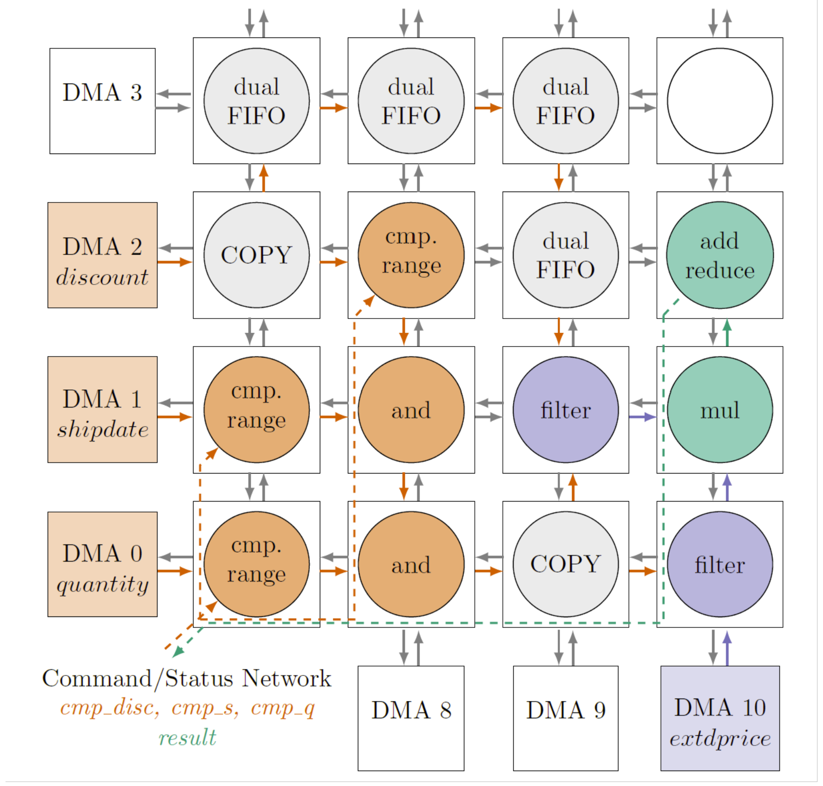
3.: Domain-Specific Prototype

Evaluation

- Query 1: many different grouped aggregation outputs, complex dataflow graph
- Query 6: computationally intensive and filters out a lot of data.

	Tiles (Compute, Support)	*DMA Streams*	*On-Chip Net. "Streams"*	*DPR (ms)*
Query 1 a	31 (20, 11)	11	17	20.2
Query 1 b	32 (22, 10)	4	28	12.1
Query 6	15 (11, 4)	4	4	25.7



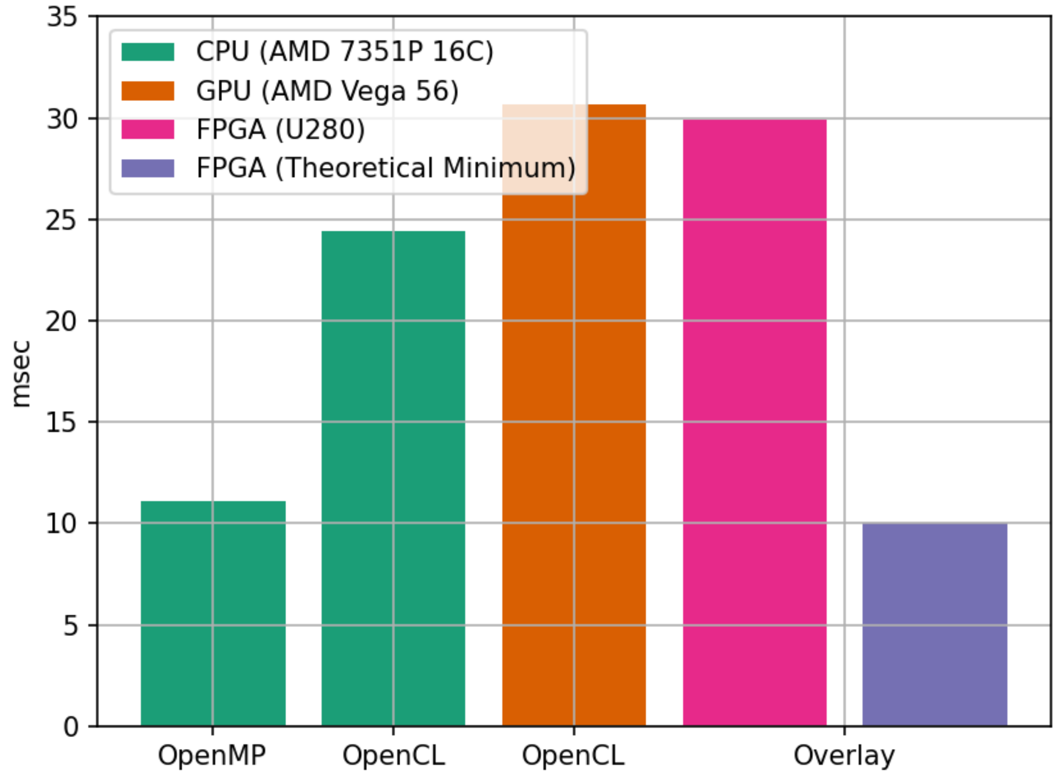


TPC-H Q6

3.: Domain-Specific Prototype

Performance

- TPC-H Q6, 19 million rows
- CPU: AMD 7315, 16 cores
  - 150 W
- GPU: AMD VEGA 56 GPU
  - 250 W
- FPGA: Alevo U280
  - 25 W
  - Design requires 1/6 of FPGA resources

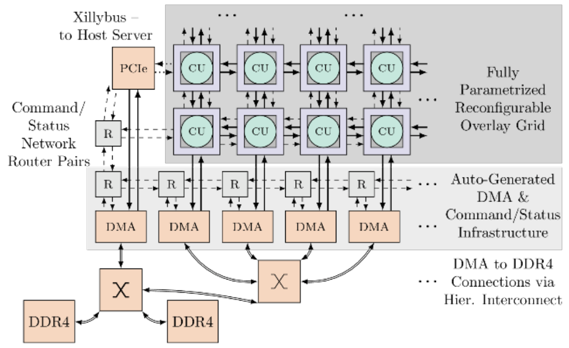


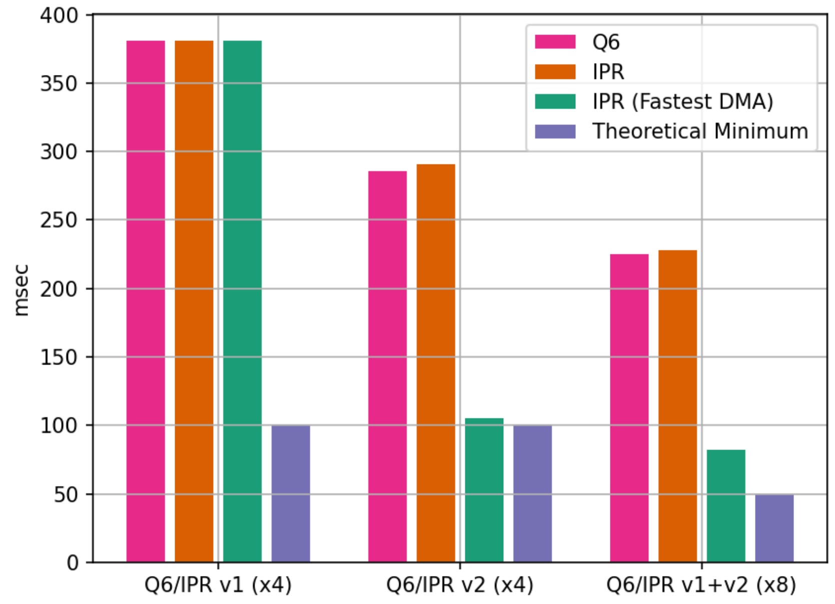
TPC-H Q6

3.: Domain-Specific Prototype

Evaluation of Memory Subsystem

- Ideal case:
  - No pipelining stall effects
  - One beat per clock cycle
- Test case: Independent parallel reduce (IPR) operation
  - Mapped to adjacent tiles of DMA channels



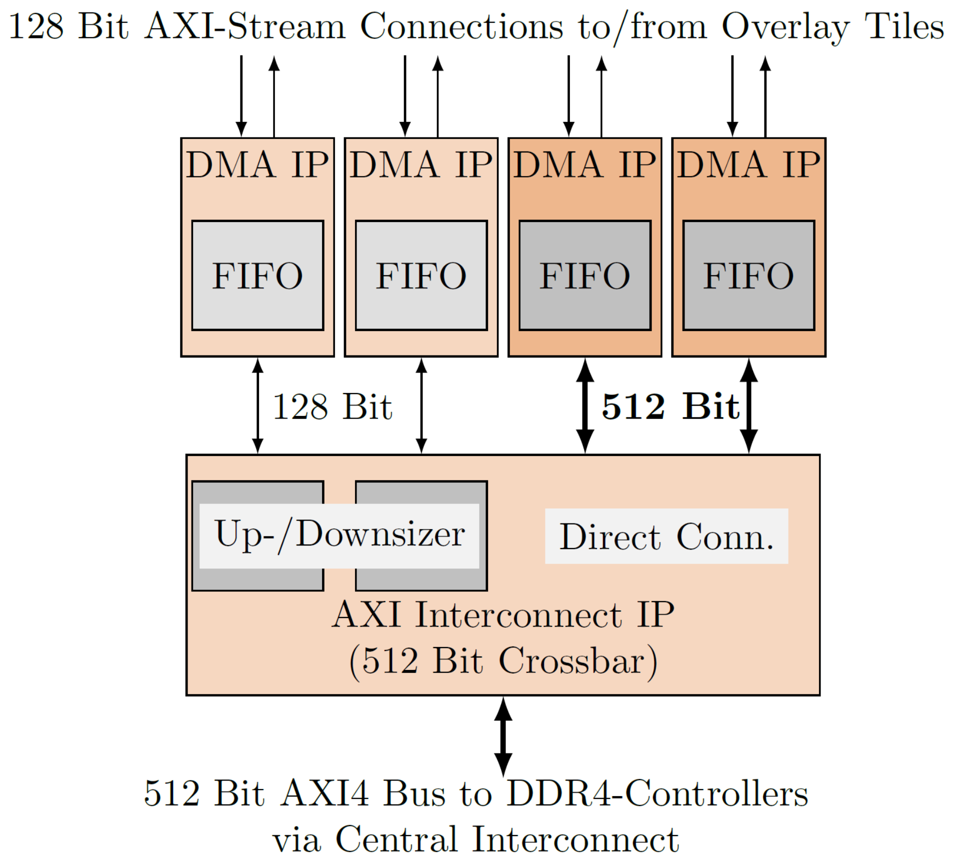


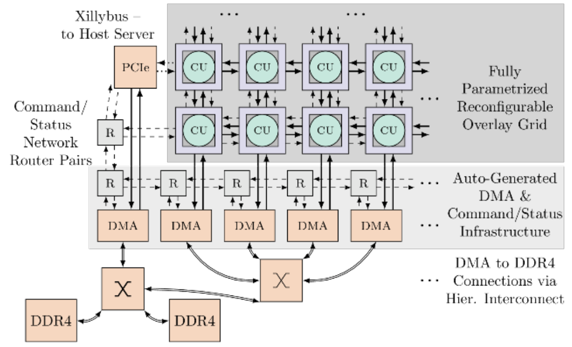
Two version of Q6 for test purpose

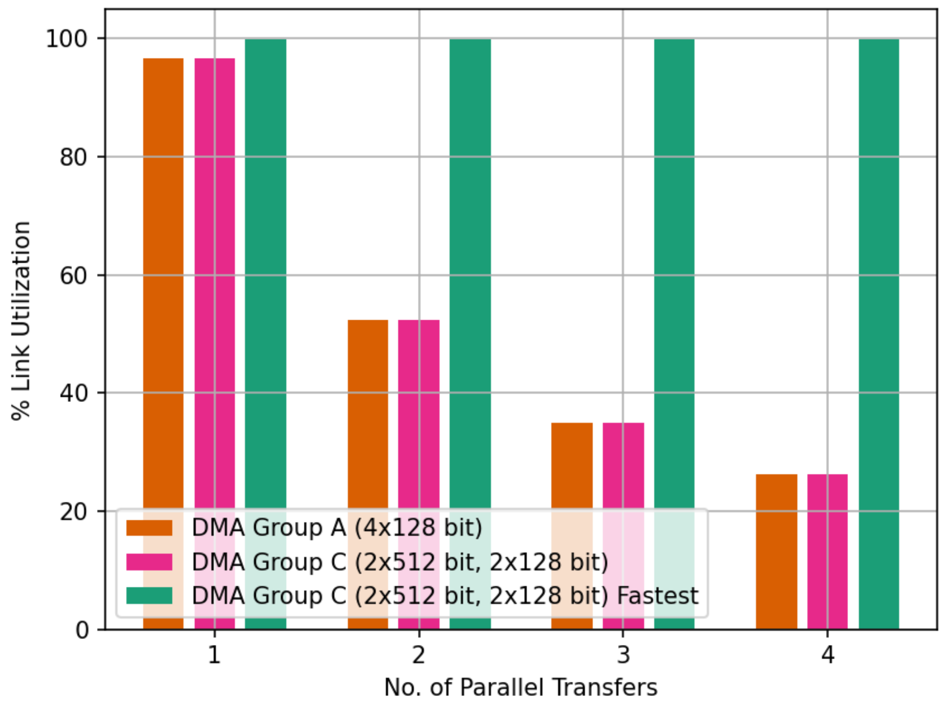
3.: Domain-Specific Prototype

Evaluation of Memory Subsystem

Effect of location of bit width conversion




Xillybus – to Host Server
PCIe
CU
Command/ Status Network Router Pairs
R
Fully Parametrized Reconfigurable Overlay Grid
Auto-Generated DMA & Command/Status Infrastructure
DMA
DMA to DDR4 Connections via Hier. Interconnect
X
DDR4



Outline

4.: Conclusion

- Introduced a template for runtime reconfigurable, domain-specific overlay architectures
  - Addresses design effort for overlay architectures
  - Allows dynamic exchange of compute units and runtime configuration of routing paths
  - AXI compliant to support HDL, IP cores, HLS kernels
- Demonstrated usefulness by means of example from database domain
  - Dynamic exchange of queris
- But:
  - Performance goals not reached
  - Caused by infrastructure IP cores outside overlay architecture
  - Several ways to circumvent this limitation

➡ Work in progress