Kalray MPPA®

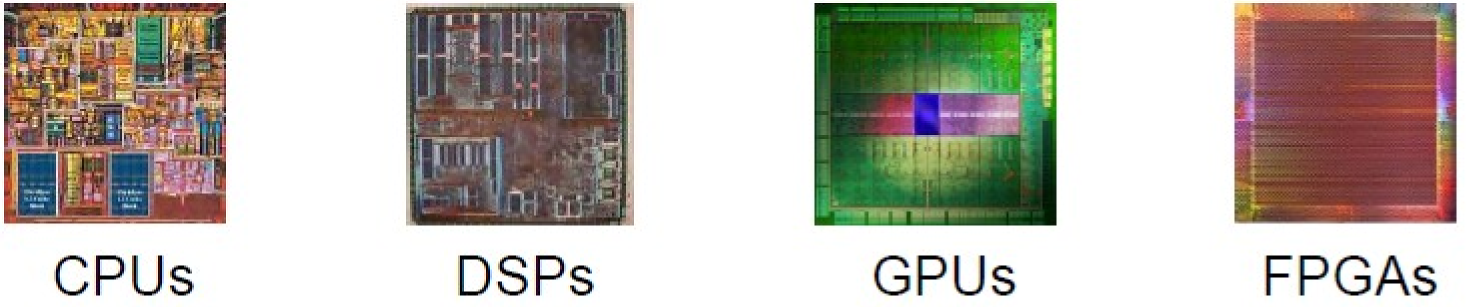
Massively **P**arallel **P**rocessor **A**rray

Synchronous Language Execution on the Kalray MPPA®-256 Bostan Manycore Processor

Amaury Graillat
Amaury.graillat@kalray.eu

Landscape of Computing Technologies

- Field-Programmable Gate Arrays (FPGA)
  - Most effective on bit-level computations
  - Require HDL programming
  - **Suitable for safety-critical computing**
- Digital Signal Processors (DSP)
  - Most effective on fixed-point arithmetic
  - Require low-level programming
  - **Suitable for safety-critical computing**
- Graphics Processing Units (GPU)
  - Most effective on regular computations
  - Require CUDA or OpenCL programming
  - **Unsuitable for safety-critical computing**
- Intel Many Integrated Core (MIC)
  - Require multicore programming + exploitation of SIMD instructions (AVX)
  - **Unsuitable for safety-critical computing**



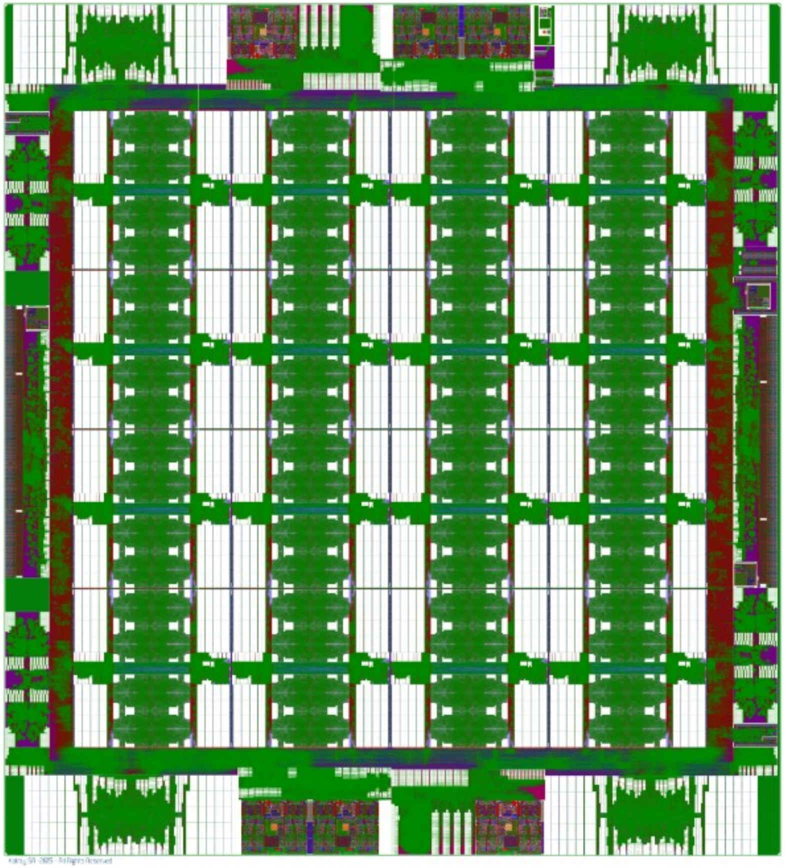
Courtesy Altera

MPPA® MANYCORE Architecture Highlights

- DSP type of acceleration
  - Energy efficiency
  - Timing predictability
  - Software programmability
- CPU ease of programming
  - C/C++ GNU programming environment
  - 32-bit or 64-bit addresses, little-endian
  - Rich operating system environment
- Integrated many-core processor
  - 32 management cores on chip
  - 256 application cores on chip
  - High-performance low-latency I/O
- Scalable massively parallel computing
  - MPPA® processors tiled together through NoC extensions

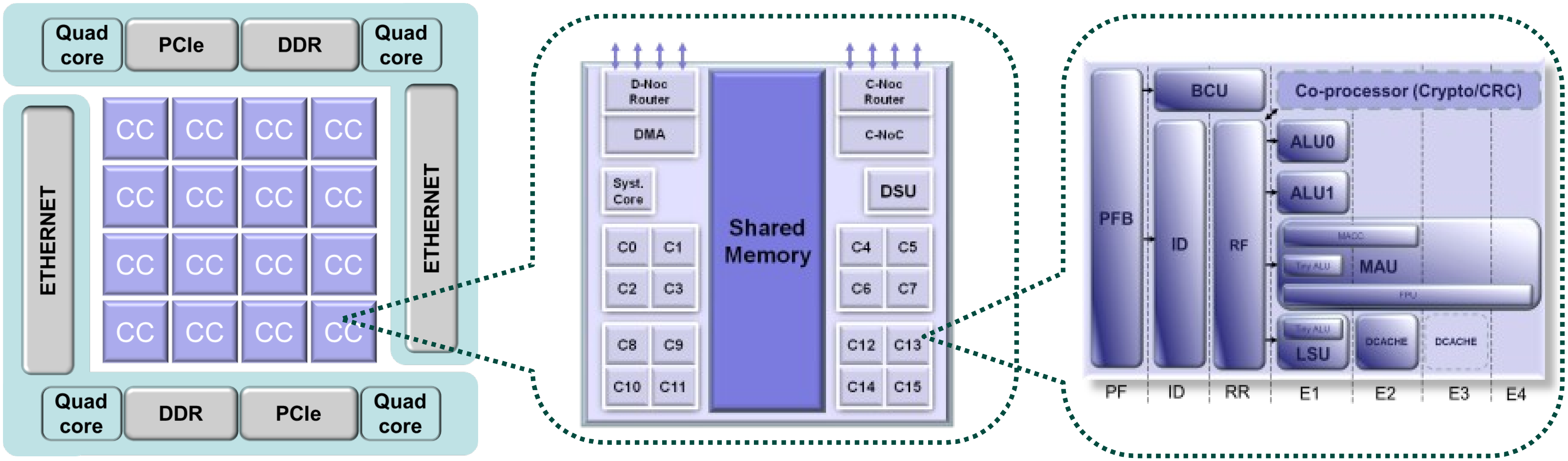
MPPA®-256 Bostan Processor

256 + 32 VLIW cores / 18 address spaces / 2D Torus dual NoC



- Physical characteristics
  - TSMC CMOS 28HP
  - 100µW/MHz per core + L1 caches
  - 2W to 3W leakage
- Processor interfaces
  - 2x DDR3 Memory interfaces
  - 2x PCIe Gen3 8-lane interface
  - 8x 1G/10G or 2x 40G Ethernet interfaces
  - SPI/I2C/UART interfaces
  - Universal Static Memory Controller (NAND/NOR/SRAM)
  - GPIOs with Direct NoC Access
  - NoC extension through Interlaken interface (NoCX)

MPPA®-256 Bostan Processor Architecture



Manycore Processor

- 16 compute clusters
- 2 I/O clusters each with quad-core CPUs, DDR3, 4 Ethernet 10G and 8 PCIe Gen3
- Data and control networks-on-chip
- Distributed memory architecture
- 634/317 GFLOPS SP/DP @ 600Mhz

Compute Cluster

- 16 user cores + 1 system core
- NoC Tx and Rx interfaces
- Debug & Support Unit (DSU)
- 2 MB multi-banked shared memory
- 77GB/s Shared Memory BW
- 16 cores SMP System

VLIW Core

- 32-bit or 64-bit addresses
- 5-issue VLIW architecture
- MMU + I&D cache (8KB+8KB)
- 32-bit/64-bit IEEE 754-2008 FMA FPU
- Tightly coupled crypto co-processor
- 2.4 GFLOPS SP per core @600Mhz

MPPA® Processor Co-Design for Avionics

- U. Saarland / AbsInt GMBH recommendations on VLIW core and cache micro-architecture design
  - AbsInt provides the aiT static timing analysis tool used to certify the flight control system of Airbus A380, Airbus A350 and Airbus A400M
  - AbsInt aiT tool also targets the Kalray VLIW cores
- Architecture with a focus on timing predictability
  - Core level: micro-architecture
    - ✓ Fully timing compositional core
    - ✓ LRU caches and write buffer
    - ✓ Cache bypass memory accesses
  - Cluster level: multi-banked shared memory
    - ✓ Core-private buses for memory bank access
  - Processor level: NoC with guaranteed services
    - ✓ Minimum bandwidth & maximum latency

CERTAINTY

Certification of Real Time Applications desIgNed for mixed criticaliTY

THALES

UNIVERSITE JOSEPH FOURIER

ARTTIC

Technische Universität Braunschweig

AbsInt

UPPSALA UNIVERSITET

ETH Eidgenössische Technische Hochschule Zürich Swiss Federal Institute of Technology Zurich

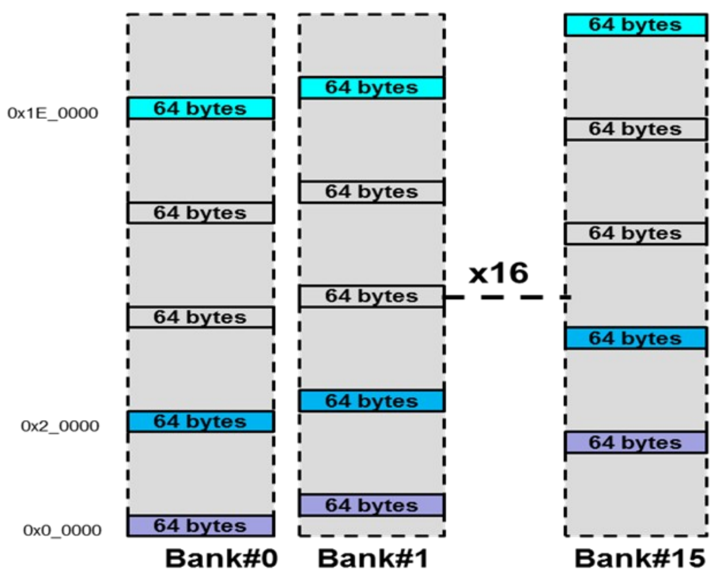
Kalray VLIW Architecture Compared to HP Labs Lx

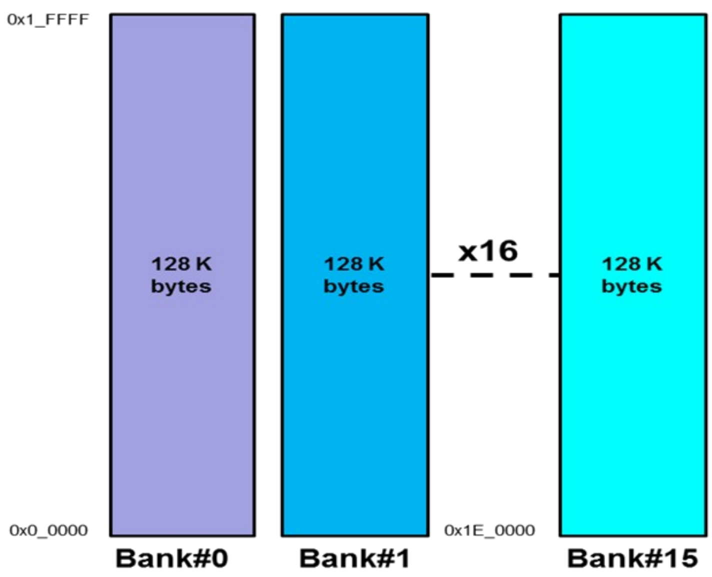
The Lx architecture begat the STMicroelectronics ST200 VLIW family

- Optimize use of the data memory bandwidth
  - Widening to 64-bit, no alignment restrictions
  - Enable large immediate values in instruction stream
  - All memory accesses may bypass the L1 data cache & write buffer
- Eliminate DLX ISA features and restrictions
  - Instructions with 3 or 4 source operands, 1 or 2 target operands
  - No aliasing between registers and special resources (LR, zero)
  - Memory addressing modes similar to those of PowerPC
  - Effective floating-point support with Fused Multiply Add
- Rework if-conversion support
  - Remove Boolean registers and SELECT instructions
  - Use CMOV and conditional load/store instructions
- Support hardware looping

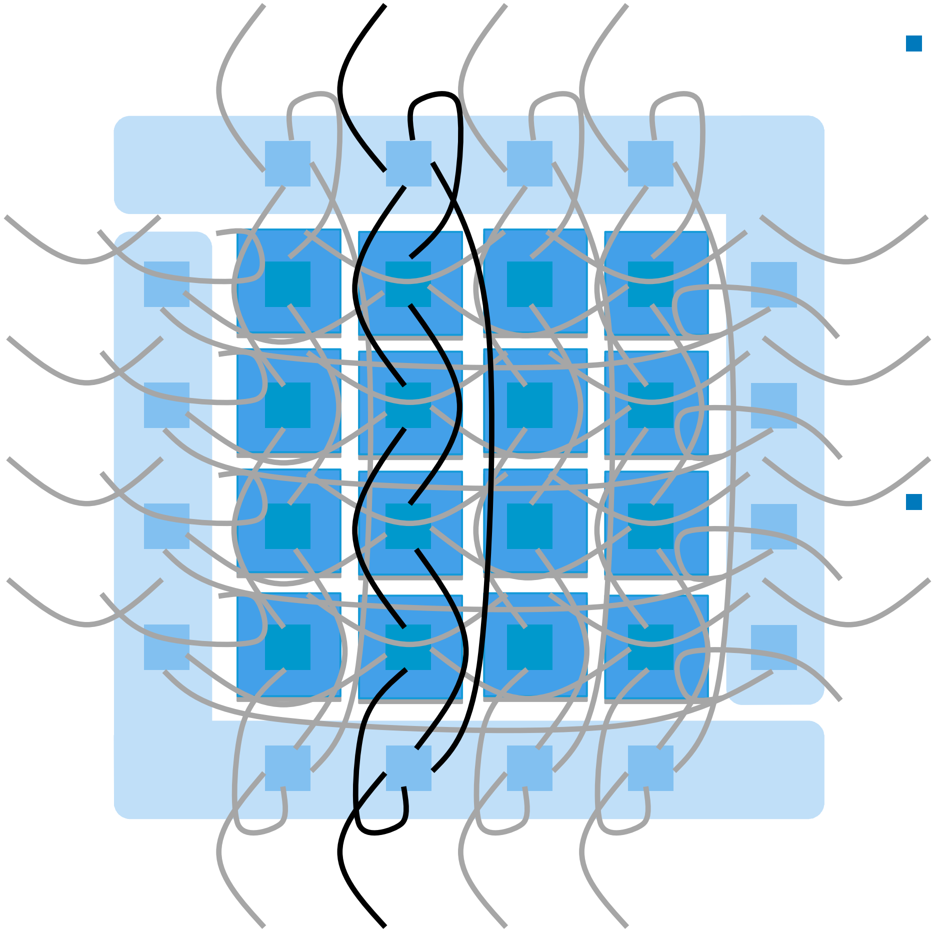
MPPA®-256 Bostan Compute Cluster

- 20 bus masters
  - 16 application cores
  - 1 management core
  - NoC Tx and Rx interfaces
  - Debug support unit (DSU)
- 16-banked shared memory
  - 2MB extensible to 4MB
  - No bus interferences between cores
  - RR arbitration between bus masters
  - Interleaved or blocked address map





MPPA®-256 Bostan Network-on-Chip (NoC)



- Dual 2D-torus NoC
  - D-NoC: high-bandwidth RDMA
  - C-NoC: low-latency mailboxes
  - 4B/cycle per link direction per NoC
  - Nx10Gb/s NoC extensions for connection to FPGA or other MPPA®
- Predictability
  - Data NoC is configured by selecting routes and injection parameters
  - Injection parameters are the (σ,ρ) or (burst, rate) of Cruz network calculus
  - Guaranteed services rely on same methods as in AFDX Ethernet

Applications of MPPA® MANYCORE Processors

- Cloud and Data Center acceleration
  - Offloading of real-time or compute intensive functions from x86 applications
  - Domains of application: video, networking, storage, OHPC, data analytics, cybersecurity
  - MPPA® Compute Clusters seen as OpenCL Compute Units or pools of DSP processors

- High Performance Embedded Computing
  - Stand-alone computing enables increased integration of functions including those constrained by real-time
  - Domains of application: aerospace, automotive, transport, energy
  - MPPA® Compute Clusters seen as precision-timed multicore CPUs

MPPA® AccessCore™ SDK

Standard C/C++ Programming Environment

Simulators & Profilers, Debuggers & System Trace

Operating Systems & Device Drivers

AccessLib™ Optimized Libraries

C - Low Level/ Lib DSP Style Programming

C - POSIX-Level CPU Style Programming

OpenCL GPU Style Programming

OpenDataPlane Open API for networking

Linux

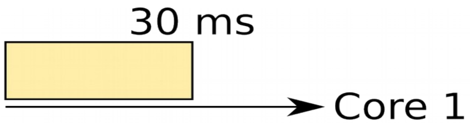
eSoL

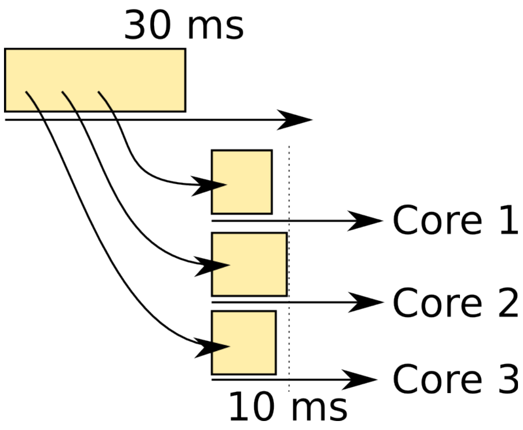
ANSYS

ESTEREL Technologies

SCADE Code Generation for the MPPA®

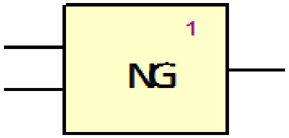
- Safety-critical control-command applications
  - Model-based programming using SCADE Suite® from Esterel Technologies
  - Complemented with static timing analysis of binary code (aiT from AbsInt)
- Motivations for multicore and manycore execution
  - Distribute the compute load across cores and reduce memory interferences
  - Effective implementation of multi-rate harmonic real-time applications
- Envision use of fast Model Predictive Control (MPC) techniques





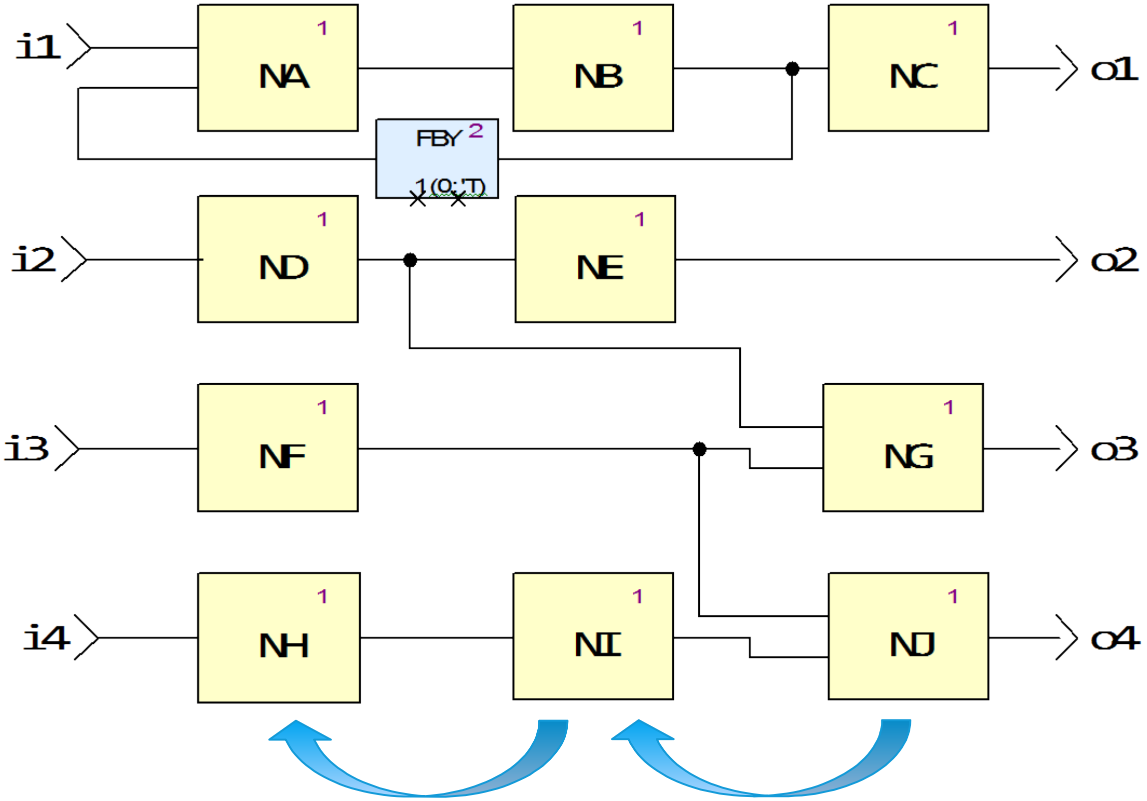
The SCADE Language

- A Dataflow language comparable to a circuit.
- Defines relation between inputs and outputs.



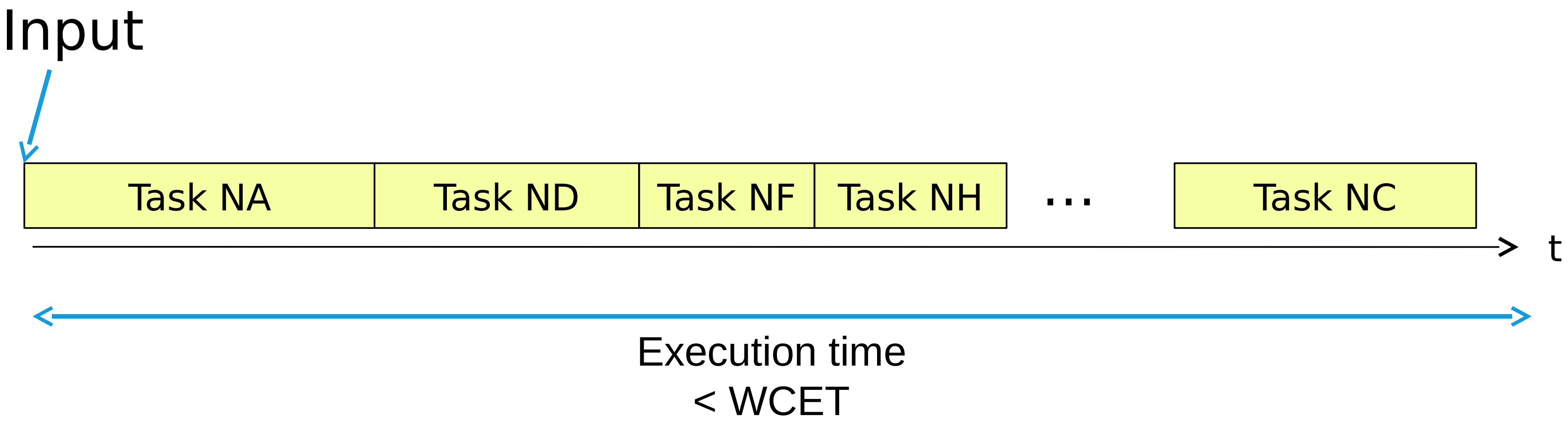
- A Synchronous language with zero-time execution semantics: A change on inputs instantaneously affect outputs
- **Designed for time-critical applications.**

Example of SCADE Program



Dependencies represented by wires.

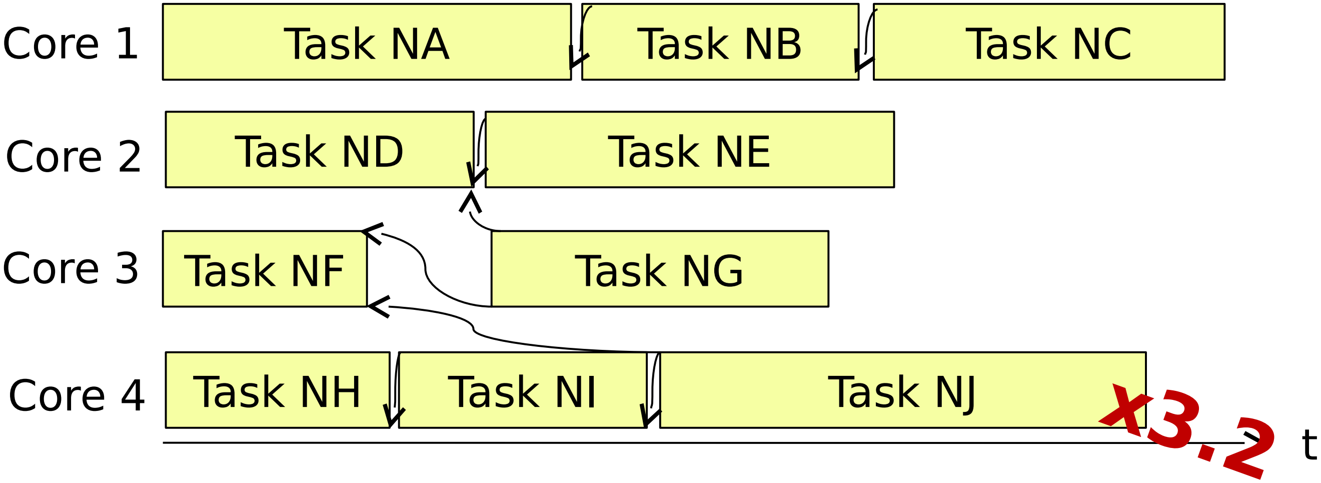
Motivations



- **Sequential execution**
  - **Time critical: Execution time < WCET < Latency**

- **Need actual hardware parallelism.**

Expected parallelization



- **Kahn process networks model**
  - **Node executes as soon all the inputs are available.**
  - **E.g. NG needs NF and ND**

Example of SCADE Program

- Original SCADE Program:

```
a = NA(i1, 0->pre(b));
h = NH(i4);

B = 1 + NB(a);
d = ND(i2);
f = NF(i3);
i = NI(h);

o1 = NC(b);
o2 = NE(d);
o3 = NG(f, d1);
o4 = NJ(f, i);
```

- Annotated SCADE Program:

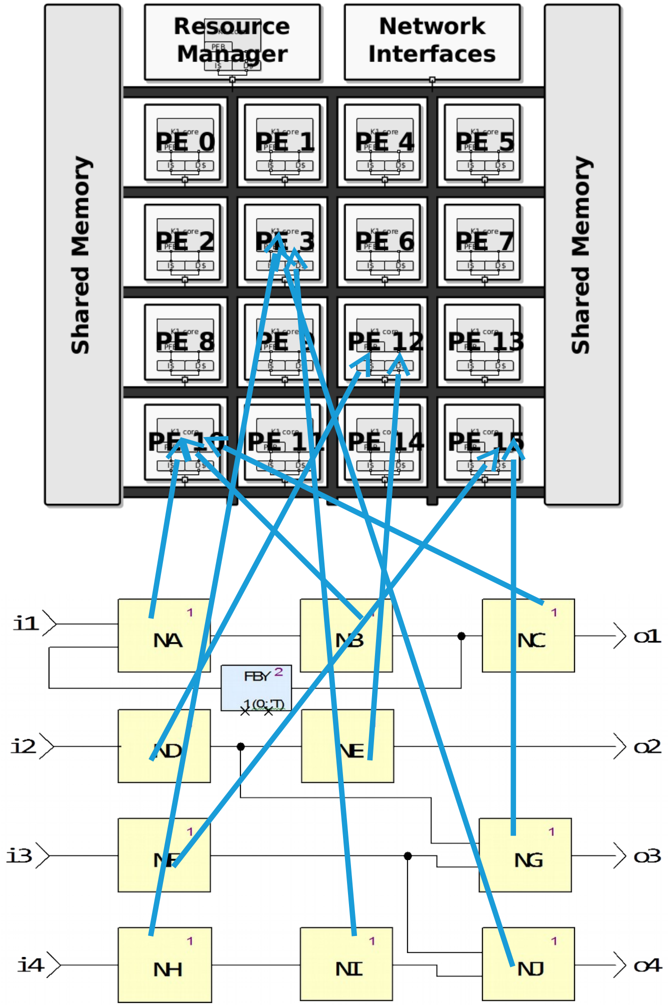
```
a = #par_1 NA(i1, 0->pre(b));
h = #par_1 NH(i4);

b = 1+ #par_2 NB(a);
d = #par_2 ND(i2);
f = #par_2 NF(i3);
i = #par_2 NI(h);

o1 = #par_3 NC(b);
o2 = #par_3 NE(d);
o3 = #par_3 NG(f, d1);
o4 = #par_3 NJ(f, i);
```

Implementation on the Kalray MPPA-256

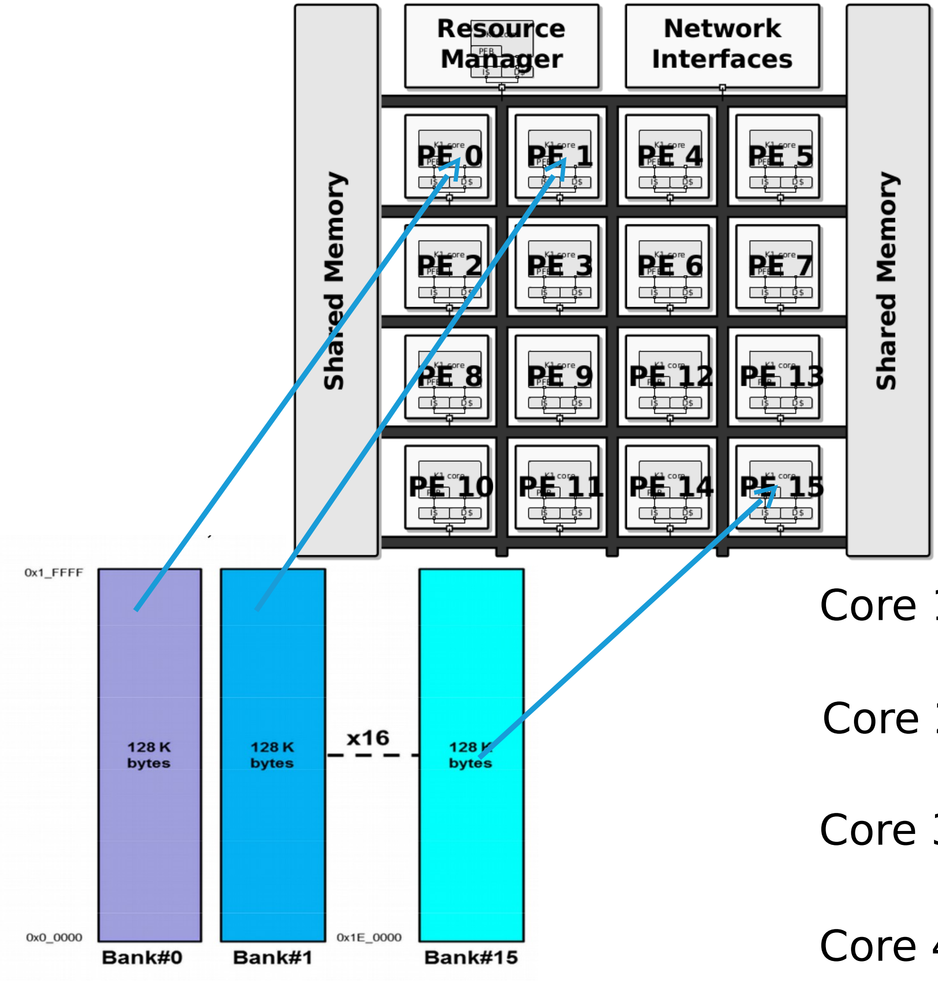
- Core-level parallelism: nodes on cores
- Sequential nodes on the same core

- Mapping specified by the developer:

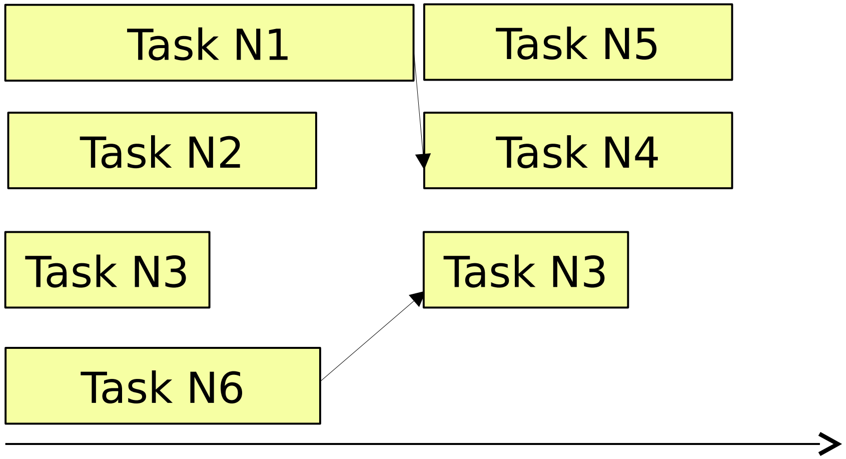
  Core 10 : NA ; NB ; NC
  Core 3 : NH ; NJ : NI
  Core 12 : ND ; NE
  Core 15 : NF ; NG



Memory interference reduction (SMEM)



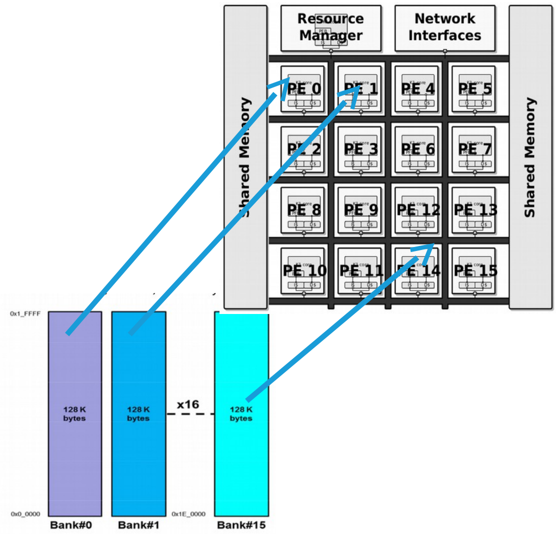
- One memory bank per core.
- Non-interferent memory access
- Parallel memory access.

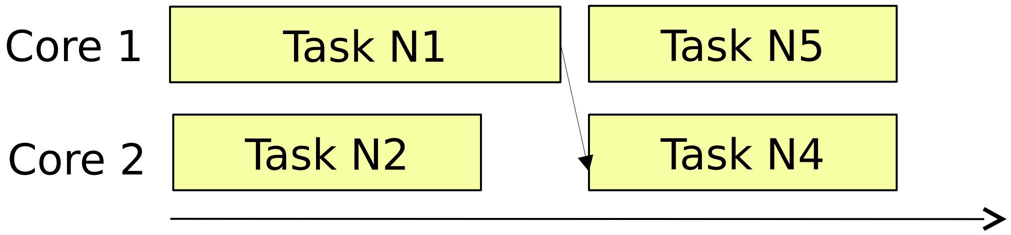


Memory interference reduction (SMEM)

- Remote write: a task sends its result.

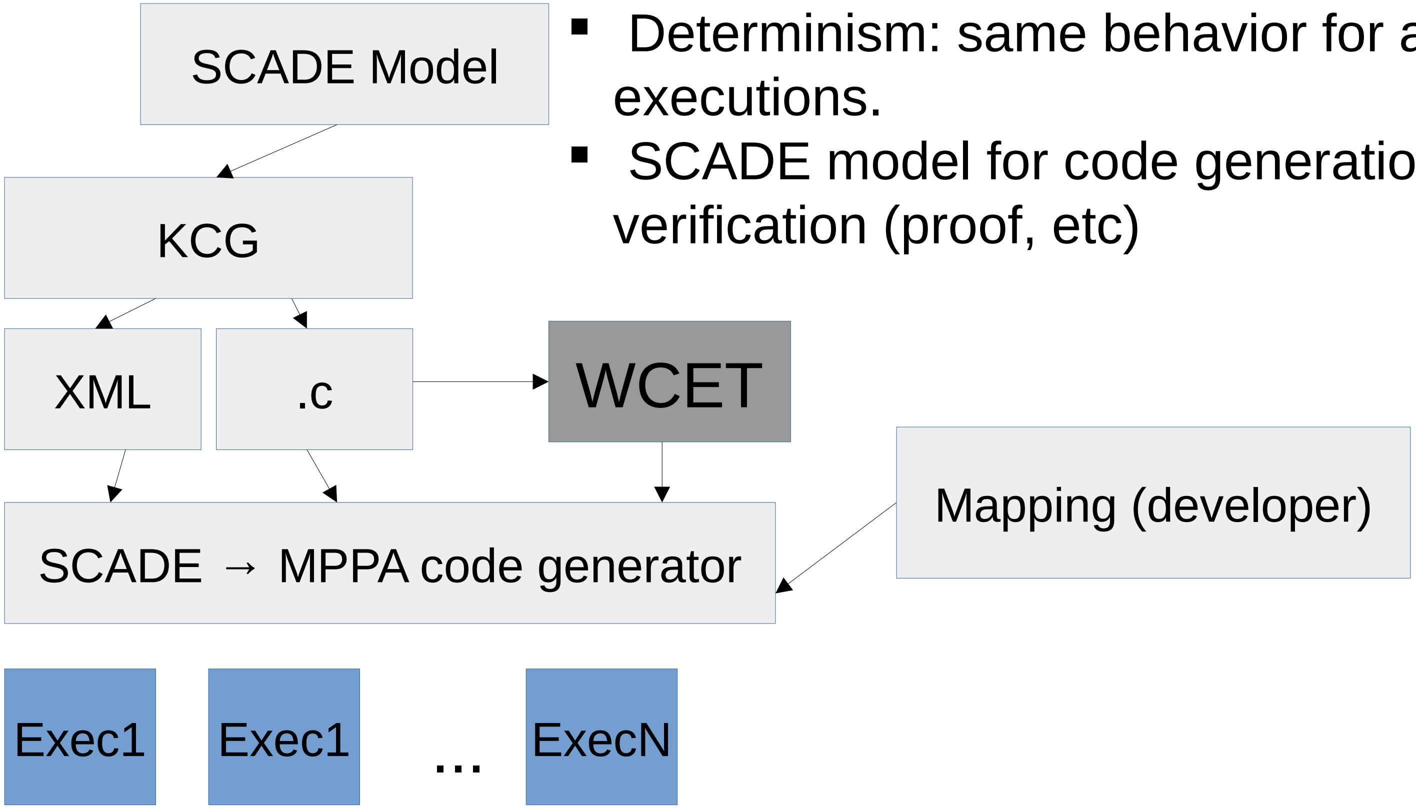
- Two implementations:
  - Signal when data is received (dynamic)
  - Receiver task starts when data is received **for sure**: (static) t(N4) >= t(N1)+WCET+WCTxT





A Semantics Preserving Approach

From SCADE program to MPPA.



- Determinism: same behavior for all executions.
- SCADE model for code generation and verification (proof, etc)

Conclusion

- Model-based code generation for MPPA.
- A semantics preserving approach.
- Time-critical constraints

Thank you!

KALRAY S.A.
Paris - France

86 rue de Paris,
91 400 Orsay
France

Tel: +33 (0) 184 00 00 45
email: info@kalray.eu

KALRAY S.A.
Grenoble - France

445 rue Lavoisier,
38 330 Montbonnot
France

Tel: +33 (0)4 76 18 09 18
email: info@kalray.eu

KALRAY INC.
Los Altos - USA

4962 El Camino Real
Los Altos, CA
USA

Tel: +1 (650) 469 3729
email: info@kalrayinc.com

MPPA, ACCESSCORE and the Kalray logo are trademarks or registered trademarks of Kalray in various countries.
All trademarks, service marks, and trade names are the marks of the respective owner(s), and any unauthorized use thereof is strictly prohibited. All terms and prices are indicatives and subject to any modification without notice.

PCIe Cards for the MPPA®-256 Bostan Processor

KONIC-80

- 1x MPPA®-256 Bostan processor
- 2x QSFP+ => 2 x 40GbE or 8 x 10GbE
- 2x DDR3 @ 2133 MT/s => 34 GB/s
- OpenDataPlane SDK
- Virtualization Offload
- First engineering samples: Q4-15
- Volume Production: Q1-16

TurboCard-3

- 4x MPPA®-256 Bostan processors
- 2x NoC Extension interfaces
- 2.5 TFLOPS SP / 1.25 TFLOPS DP
- 8x DDR3 @ 2133 MT/s => 136 GB/s
- OpenCL SDK
- First engineering samples: Q1-16
- Volume Production: Q2-16