



# High Performance Embedded Computing on the MPPA<sup>®</sup> Single Chip Manycore Processor

**CERN Seminar**

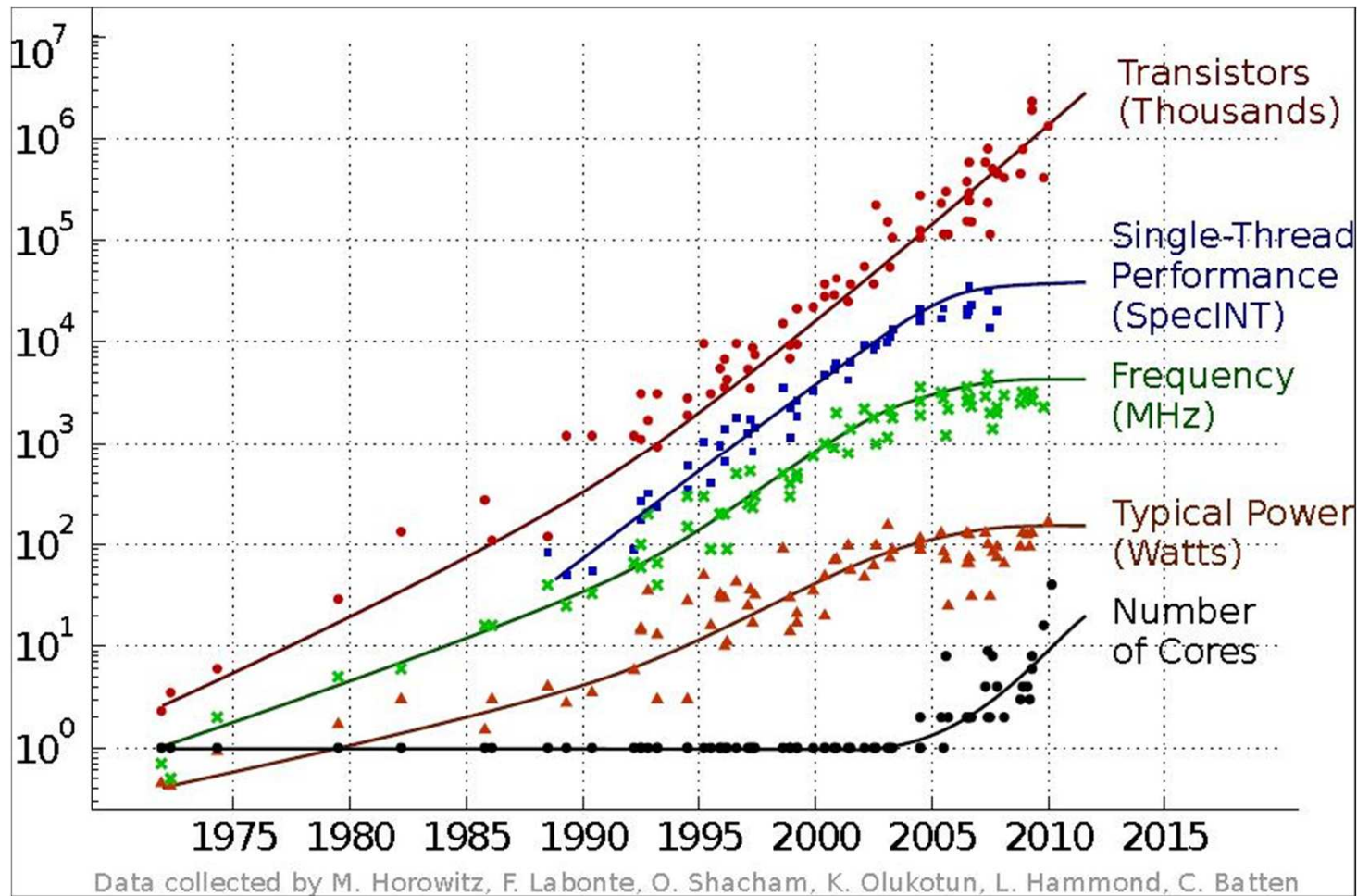
**Benoît Dupont de Dinechin**  
**[benoit.dinechin@kalray.eu](mailto:benoit.dinechin@kalray.eu)**

## Kalray Key facts

- Creation : 2008 – located in Paris, Grenoble (France) & Tokyo (Japan)
- Kalray people: 55+
- Joint laboratory with CEA engineers: 30
- Run by Joël Monnier, former VP STMicroelectronics
- Multi-Purpose Processing Array technology MPPA®
- Targeting the industrial and embedded computing market
- First product released Q4 2012, in 28 nm CMOS TSMC technology
- Independent technology – including core VLIW architecture and software tools, without any dependency on third party supplier
- Portfolio of 35 patents and 64 in progress

# The End of Dennard MOSFET Scaling Theory

After 2005 (90nm), frequency stagnates and power per area increases



## Manycore Challenges on Next Technology Nodes

- Dark Silicon Projection (Esmaeilzadeh et al. CACM 2013)
  - “At 8nm, over 50% of the chip will be dark and cannot be utilized”
  - Based on Device x Core x Multicore models
  - Multicore model assumes x86 CPU or GPU architecture
- Dally on “Future Challenges of Large-Scale Computing” (ISC 2013)
  - Exascale computing requires 1000x improvement in energy efficiency
  - By 2020: technology => 2.2x, circuit design => 3x, architecture => 4x
  - Power goes into moving data around – communication dominates power
- Not considered above: principles exploited by the MPPA<sup>®</sup>
  - Manycore platforms based on low-power CPUs and distributed memory
  - SoC nodes integrating high-speed networking and parallel computing

# MPPA<sup>®</sup>-256 Processor with 28nm CMOS TSMC

## 256 Processing Engine cores + 32 Resource Management cores



In production

- Processing performance
  - 700 GOPS – 230 GFLOPS (400MHz)
- Power efficiency
  - 5W to 15W (10W typical)
  - Advanced power management
- Timing predictability
- DDR3, PCI Gen3, Ethernet 10G
- Architecture scalability
  - Processor tiling through NoC extensions
- Software programmable
  - High level programming models
  - Advanced debugging and tracing

## KALRAY, a global solution

 **MPPA  
—  
MANYCORE**

High performance, low power and programmable massively parallel processors



 **MPPA  
—  
ACCESSCORE**

C/C++ based Software Development Kit (SDK) for massively parallel programming



 **MPPA  
—  
DEVELOPER**

Development platform  
“Ready to develop”



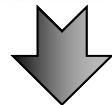
 **MPPA  
—  
BOARDS**

Reference design boards  
Application specific boards  
Single or Multi-MPPA boards



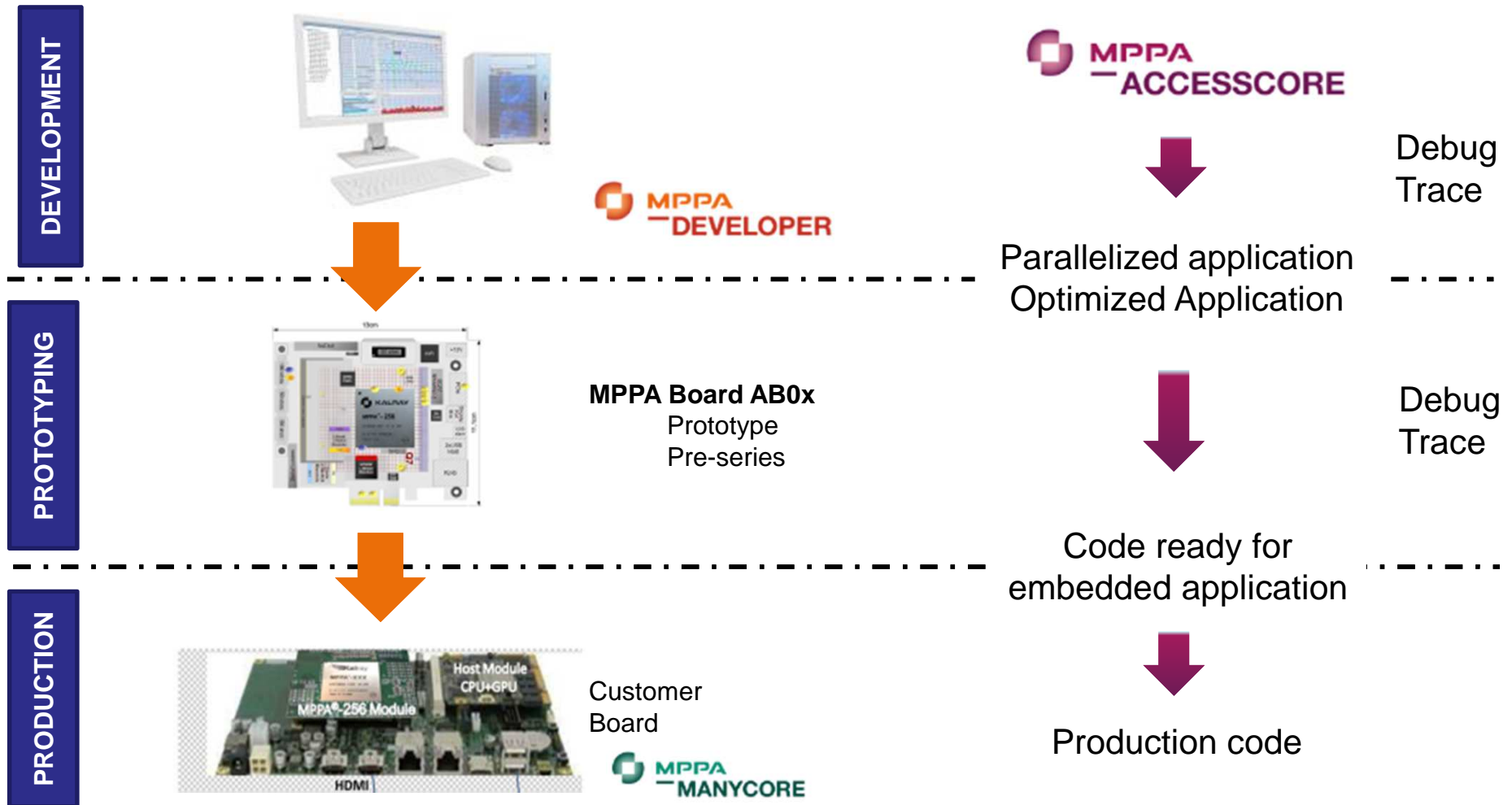
## MPPA® DEVELOPER Workstation

- Develop, optimize and evaluate your applications
- Exploit the computing power of the 256 VLIW cores
- “Ready to develop” configuration (no specific set-up)



- PCIe board MPPA®-256 Processor
- PCIe board for debug/probe
- Intel core I7 CPU 3.6GHz, Linux OS
- MPPA ACCESSCORE SDK installed
- Compatible with Multi MPPA board
- Additional services:
  - Extranet access
  - Support Team access
  - Getting started training
  - SDK maintenance

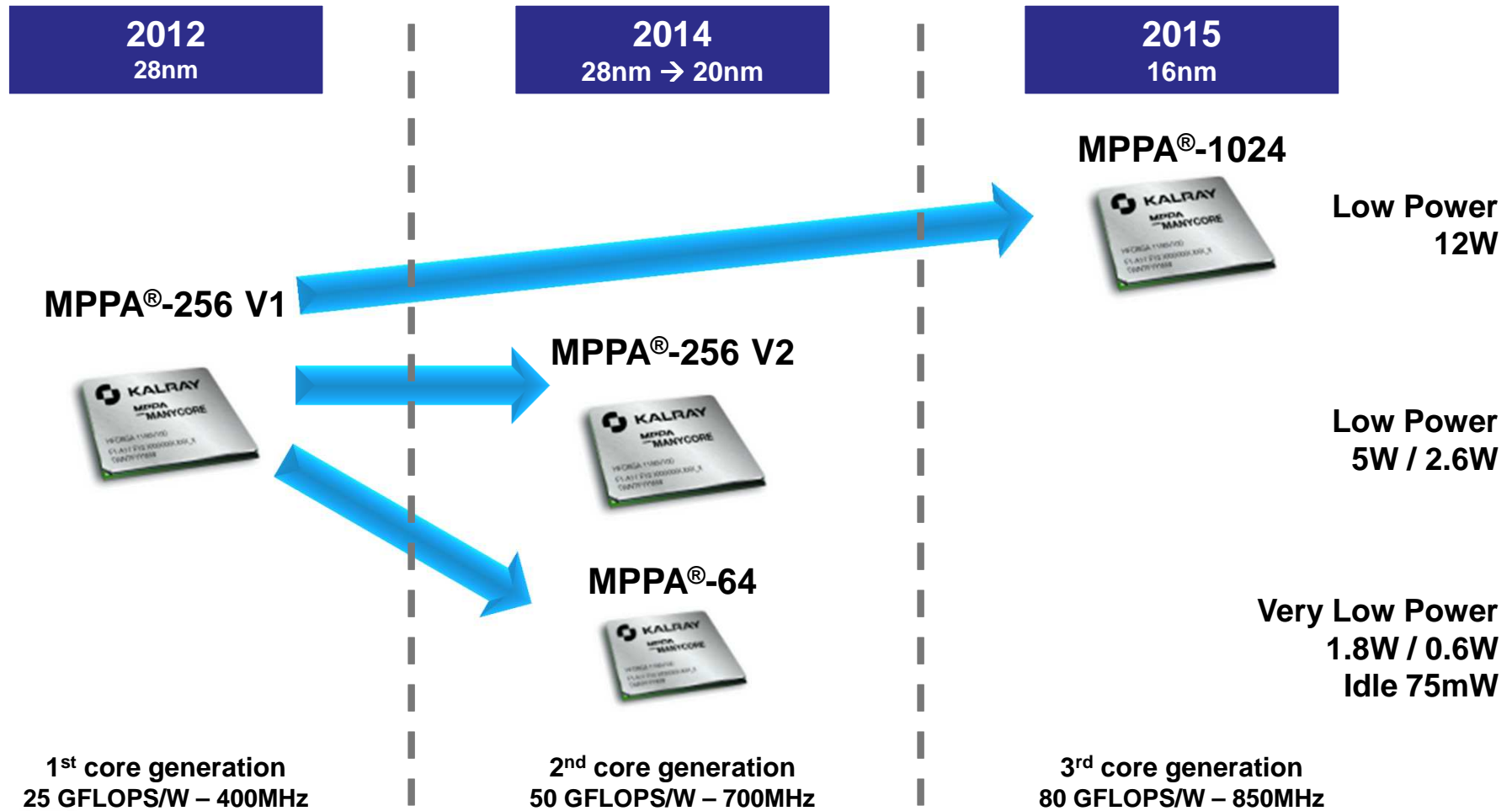
# From MPPA DEVELOPER to Customer Product





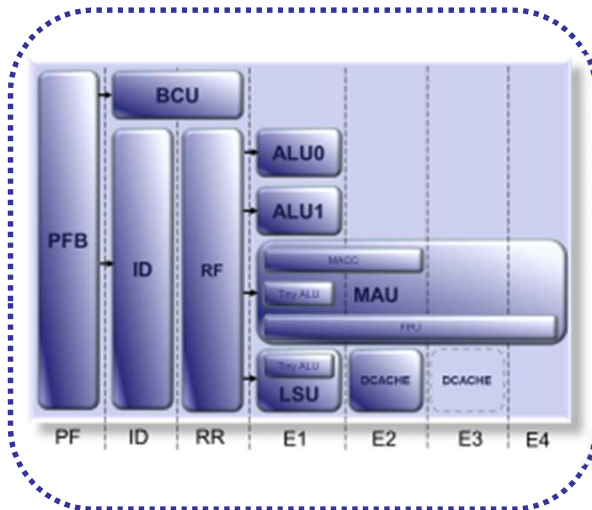
# MPPA MANYCORE Roadmap

Architecture scalability for high performances and low power



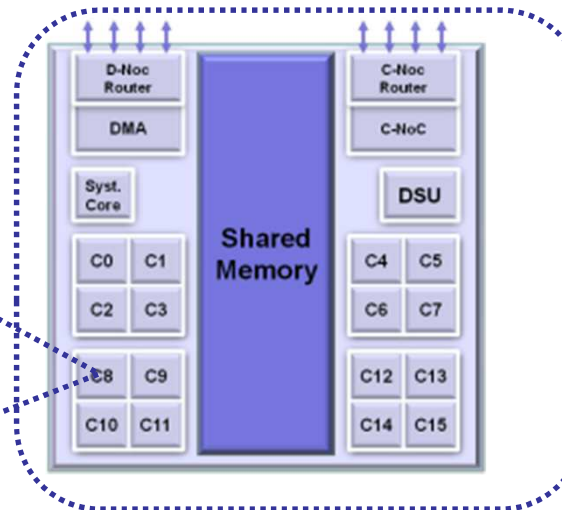
# MPPA<sup>®</sup>-256 Processor Hierarchical Architecture

VLIW Core



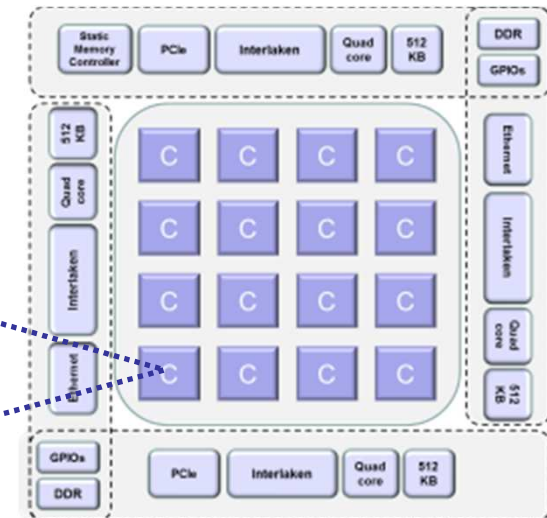
**Instruction Level  
Parallelism**

Compute Cluster



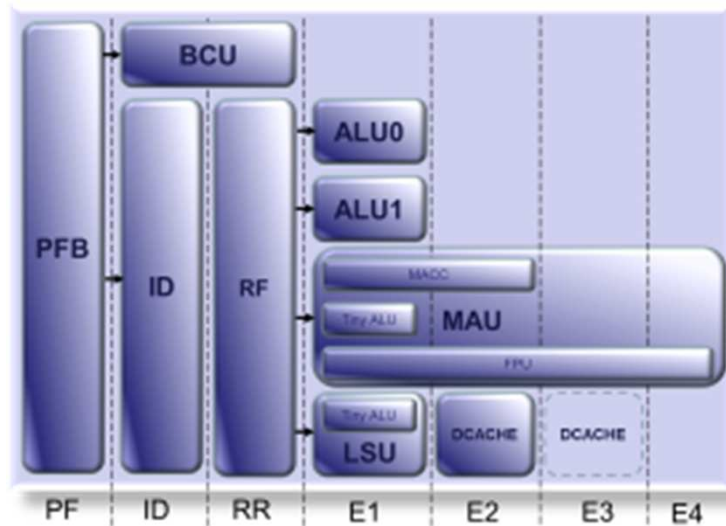
**Thread Level  
Parallelism**

Manycore Processor



**Process Level  
Parallelism**

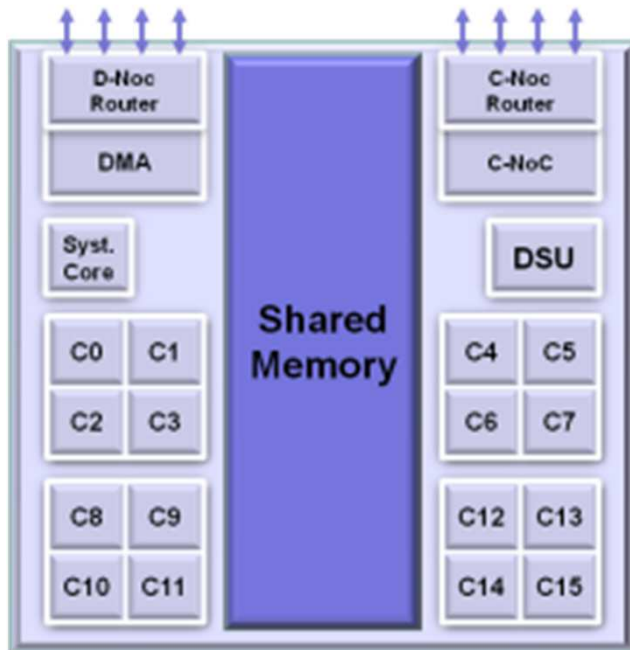
## MPPA<sup>®</sup>-256 VLIW Core Architecture



- 5-issue VLIW architecture
- Predictability & energy efficiency
- 32-bit/64-bit IEEE 754 FPU
- MMU for rich OS support

- Data processing code
  - Byte alignment for all memory accesses
  - Standard & effective FPU with FMA
  - Configurable bitwise logic unit
  - Hardware looping
- System & control code
  - MMU → single memory port → no function unit clustering
- Execution predictability
  - Fully timing compositional core
  - LRU caches, low miss penalty
- Energy and area efficiency
  - 7-stage instruction pipeline, 400MHz
  - Idle modes and wake-up on interrupt

## MPPA<sup>®</sup>-256 Compute Cluster

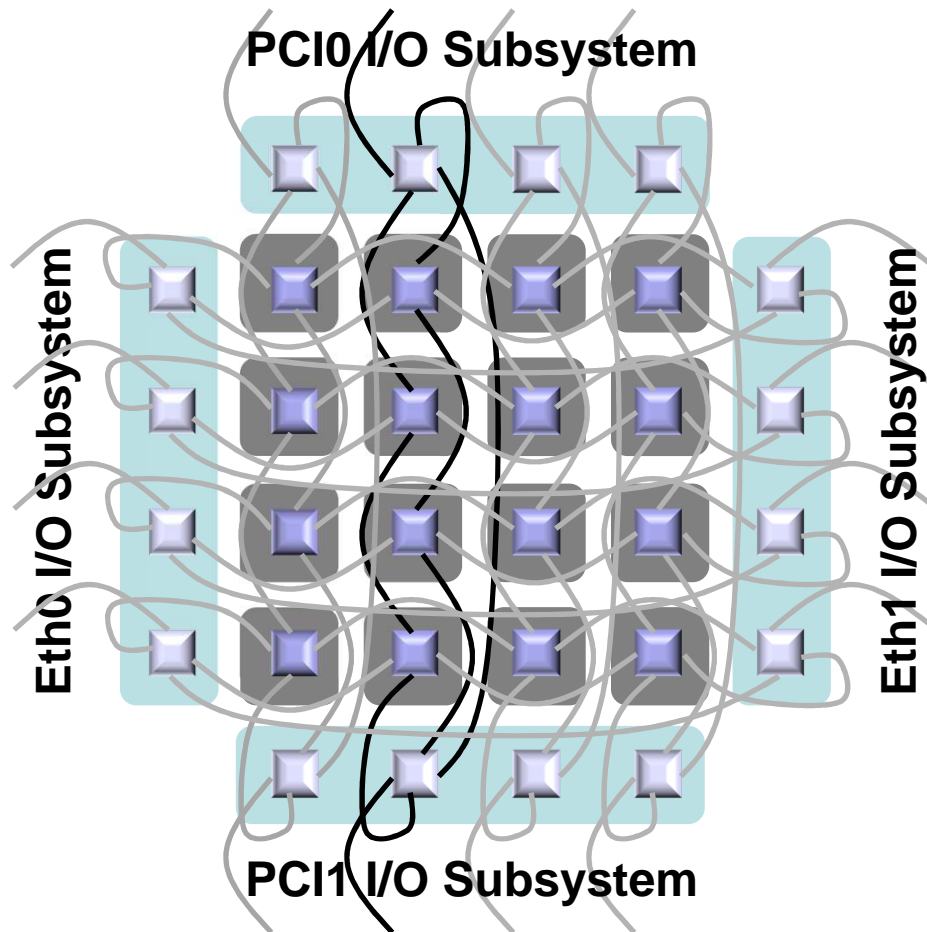


- 16 PE cores + 1 RM core
- NoC Tx and Rx interfaces
- Debug Support Unit (DSU)
- 2 MB of shared memory

- Multi-banked parallel memory
  - 16 banks with independent arbitrator
  - 38,4GB/s of bandwidth @400MHz
- Reliability
  - ECC in the shared memory
  - Parity check in the caches
  - Faulty cores can be switched off
- Predictability
  - Multi-banked address mapping either interleaved (64B) or blocked (128KB)
- Low power
  - Memory banks with low power mode
  - Voltage scaling

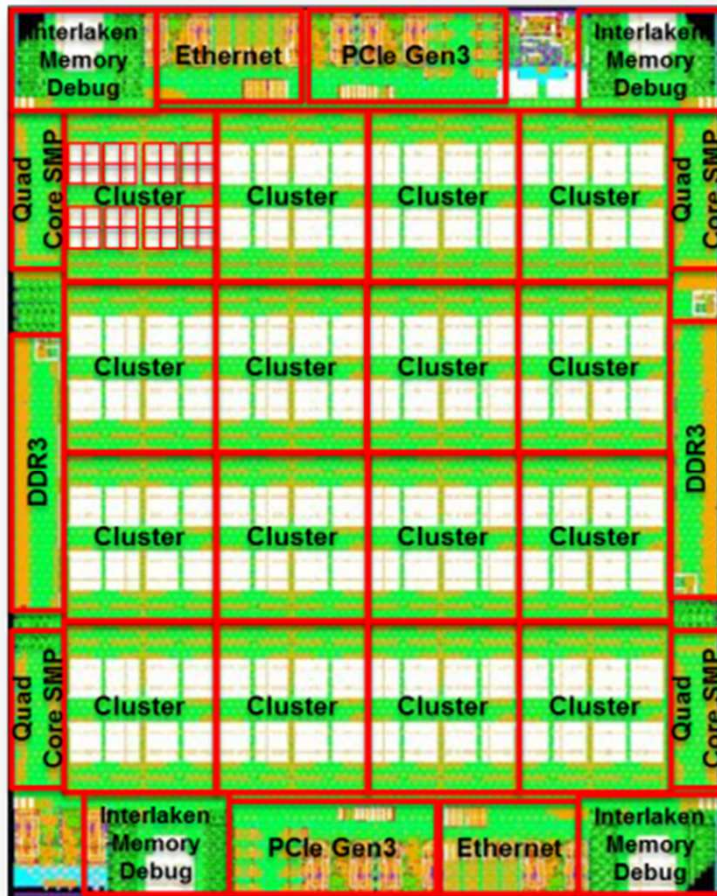
# MPPA<sup>®</sup>-256 Clustered Memory Architecture

## Explicitly addressed NoC with AFDX-like guaranteed services



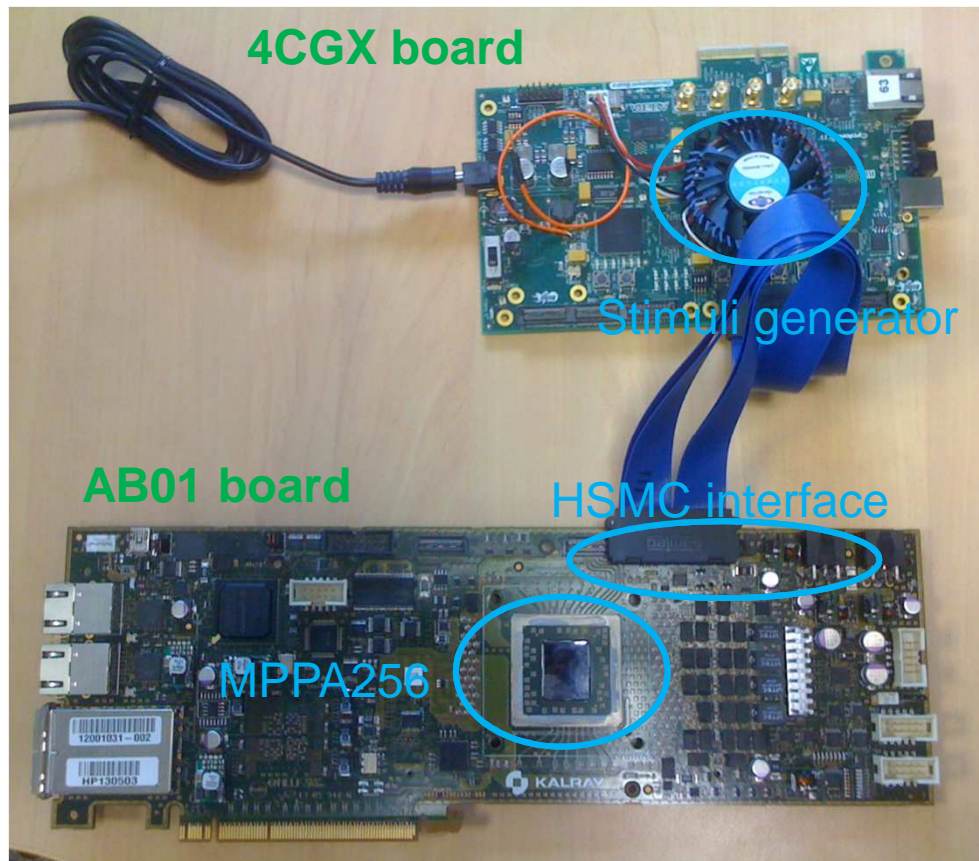
- 20 memory address spaces
  - 16 compute clusters
  - 4 I/O subsystems with direct access to external DDR3 memory
- Dual Network-on-Chip (NoC)
  - Data NoC & Control NoC
  - Full duplex links, 4B/cycle
  - 2D torus topology + extension links
  - Unicast and multicast transfers
- Data NoC QoS
  - Flow control and routing at source
  - Guaranteed services by application of network calculus
  - Oblivious synchronization

## MPPA<sup>®</sup>-256 Processor I/O Interfaces



- DDR3 Memory interfaces
- PCIe Gen3 interface
- 1G/10G/40G Ethernet interfaces
- SPI/I2C/UART interfaces
- Universal Static Memory Controller (NAND/NOR/SRAM)
- GPIOs with Direct NoC Access (DNA) mode
- NoC extension through Interlaken interface (NoC Express)

## MPPA<sup>®</sup>-256 Direct NoC Access (DNA)



### NoC connection to GPIO

- Full-duplex bus on 8/16/24 bits + notification + ready + full bits
- Maximum 600 MB/s @ 200 MHz
- Direct to the GPIO 1.8V pins
- Indirect through low-cost FPGA

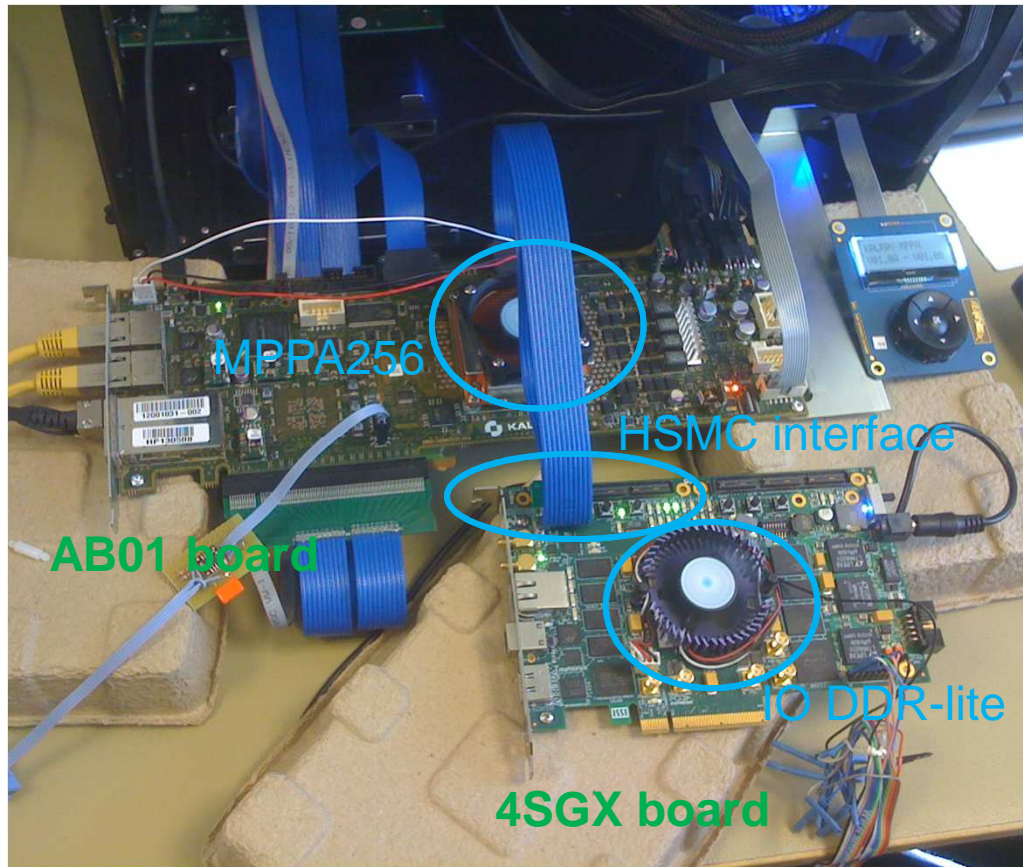
### Data sourcing

- Input directed to a Tx packet shaper on I/O subsystem 1
- Sequential Data NoC Tx configuration

### Data processing

- Standard data NoC Rx configuration
  - Application flow control to GPIO
  - Input data decoupling
  - Communication by sampling

## MPPA<sup>®</sup>-256 Sample Use of NoC Extensions (NoCX)



### Mapping of IO DDR-lite on FPGA

- Altera 4SGX530 development board
- Interlaken sub-system (x3 lanes up to 2.5-Gbit/sec)
- 300MHz DDR3
- x1 RM + x1 DMA + 512-Kbyte SMEM @ 62.5MHz
- Single NoC plug

### 4K video through HDMI emitters

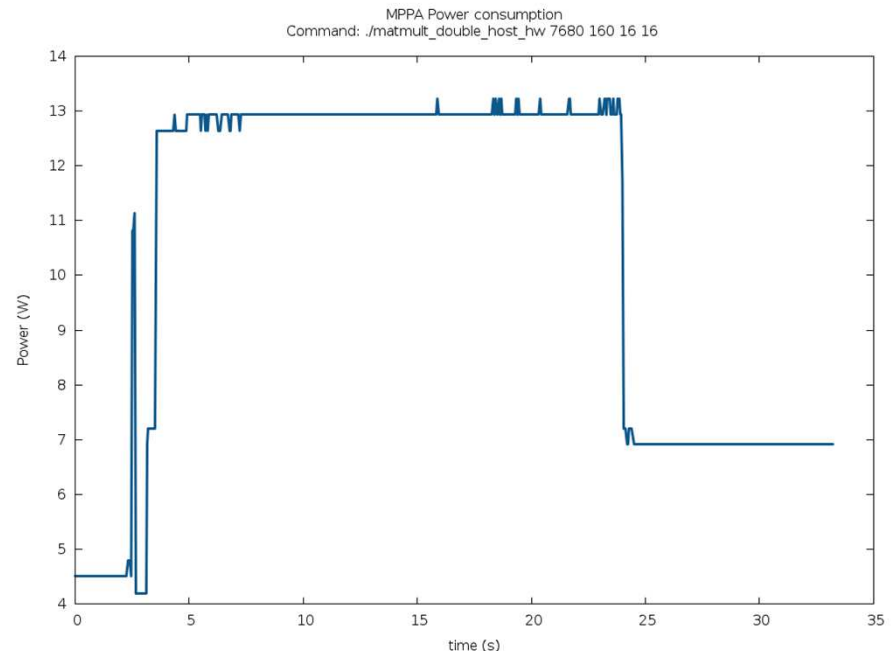
- Interlaken configured in Rx & Tx
- 1.6-Gbit/sec effective data NoC bandwidth reached
  - Limiting factor is the FPGA device internal frequency
  - Effective =  $62.5\text{MHz} * 32\text{-bit} * 80\%$
- Output of uncompressed 1080p video @ 60-frame/sec



# Measuring Power and Energy Consumption

- k1-power tool for power measurement from command shell

- `$ k1-power -h`  
`-oFILE` : output file  
`-gFORMAT` : plot in FORMAT  
`-s` : standalone mode
- `$ k1-power -oplot_double -gpdf \`  
`-- ./matmult_double_host_hw \`  
`7680 160 16 16`  
Time : 33.20 s  
Power (avg): 10.45 W  
Energy : 346.85 J
- `$ display plot_double.pdf`



- libk1power.so shared library, control measure from user code

- `int k1_measure_callback_function(int (*cb_function) (float time, double power));`
  - `int k1_measure_start(const char *output_filename);`
  - `int k1_measure_stop(k1_measure_t *measures);`
- ```
typedef struct {
    float time;
    double power;
    float energy;
} k1_measure_t;
```

## Manycore Technology Comparison

|                | Cores           | GFLOPS (SP) | Active Power | Real Time | DDR            | Ethernet |
|----------------|-----------------|-------------|--------------|-----------|----------------|----------|
| Intel Xeon Phi | 52 x86          | 2147        | 300W         | No        | GDDR5<br>1866  | No       |
| Tilera TileGx  | 72              | 80          | 60W          | No        | 4 DDR3<br>1600 | 8 10G    |
| NVIDIA Tegra4  | 4 A15<br>72 SC  | 45<br>75    | 8W           | No        | 2 DDR3<br>1866 | 1G       |
| TI Keystonell  | 4 A15<br>8 C66x | 45<br>154   | 25W          | Yes       | 2 DDR3<br>1600 | 10G      |
| Kalray MPPA    | 288 K1          | 230         | 10W          | Yes       | 2 DDR3<br>1600 | 8 10G    |

# Kalray Software Development Kit

## MPPA<sup>®</sup> ACCESSCORE – MPPA<sup>®</sup> ACCESSLIB



**Standard C/C++  
Programming  
Environment**

**Simulators,  
Profilers, Debuggers  
& System Trace**

**Operating Systems &  
Device Drivers**



**Dataflow  
Programming  
FPGA Style**

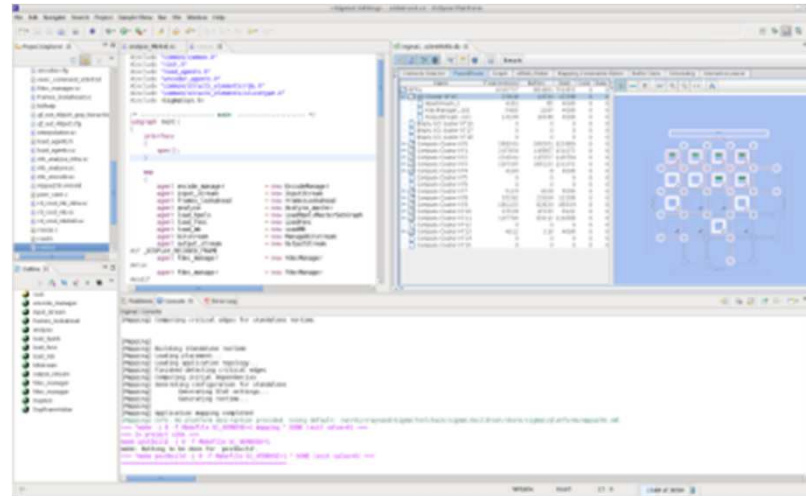
**POSIX-Level  
Programming  
DSP Style**

**Streaming  
Programming  
GPU Style**

# Standard C/C++ Programming Environment



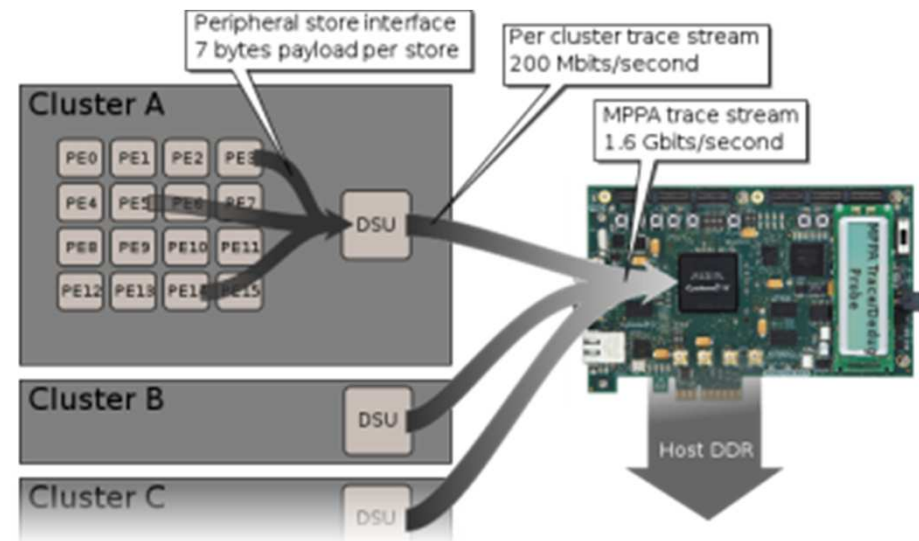
- Eclipse Based IDE and Linux-style command line
- Full GNU C/C++ development tools for the Kalray VLIW core
  - GCC 4.7 (GNU Compiler Collection), with C89, C99, C++ and Fortran
  - GNU Binary utilities 2011 (assembler, linker, objdump, objcopy, etc.)
  - GDB (GNU Debugger) version 7.3, with multi-threaded code debugging
  - Standard C libraries: uClibc, Newlib + optimized libm functions





# Simulators, Debuggers & System Trace

- Platform simulators
  - Cycle-accurate, 400KHz per core
  - Software trace visualization tool
  - Performance view using standard Linux kcachegrind & wireshark
- Platform debuggers
  - GDB-based, follow all the cores
  - Debug routines of each core activate the tap controller for JTAG output
- System trace acquisition
  - Each cluster DSU is able to generate 200Mb/s of trace
  - 8 simultaneous observable trace flows out of 20 sources
- System trace display
  - Based on Linux Trace Toolkit NG (low latency, on demand activation)
  - System trace viewer (customized view per programming model)

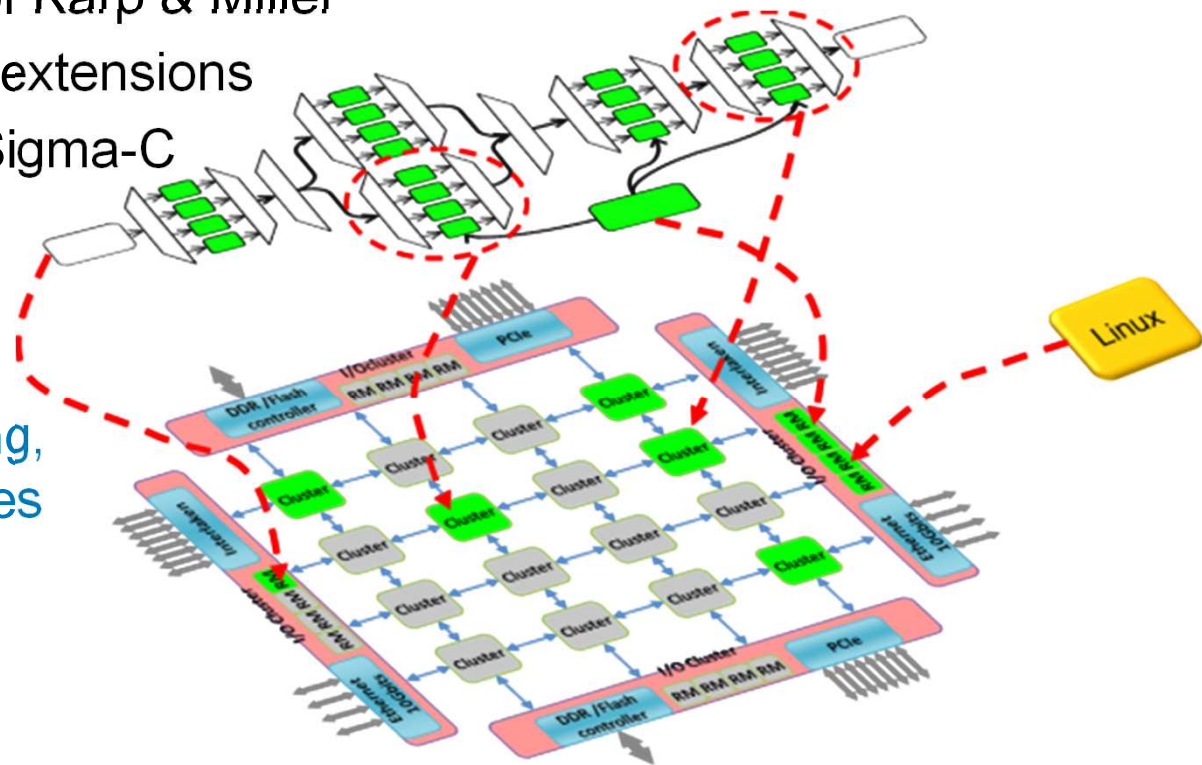




# Dataflow Programming Environment

- Computation blocks and communication graph written in C
- Cyclostatic data production & consumption
- Firing thresholds of Karp & Miller
- Dynamic dataflow extensions
- Language called Sigma-C

Automatic mapping on MPPA<sup>®</sup> memory, computing, & communication resources



# Dataflow Models of Computation

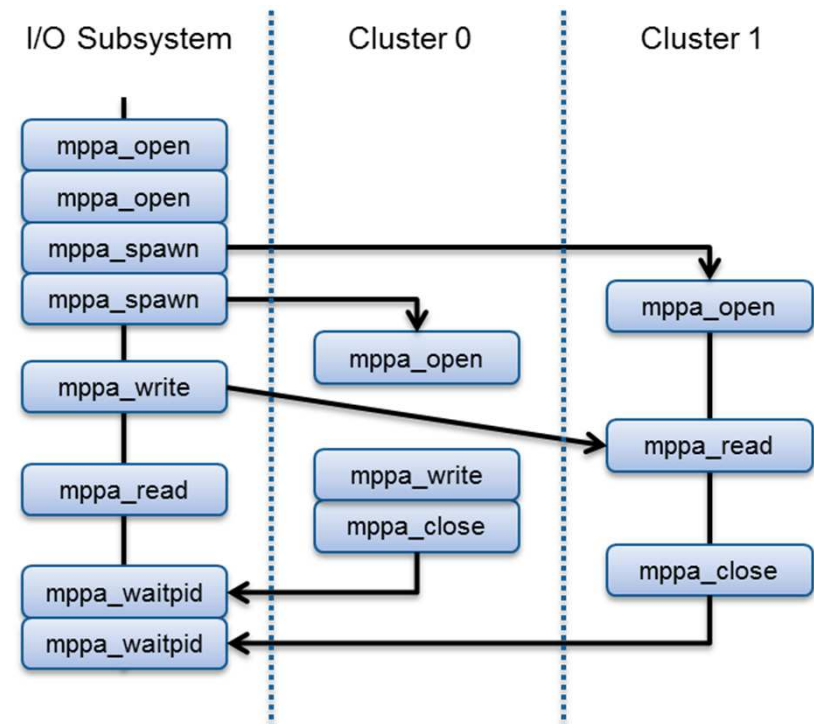


- Dataflow Process Networks (DPN) [Lee & Parks 1995]
  - Kahn Process Network with functional actors (no persistent agent state)
  - Kahn Process Network with sequential firing rules (can be tested in a pre-defined order using only blocking reads)
- Synchronous Dataflow [Benveniste et al. 1994]
  - Clocks are associated with tokens carried by the channels
- Static Dataflow (SDF) [Lee & Messerschmitt 1987]
  - Agents producing and consuming a constant number of tokens
  - Single-rate SDF is known as Homogenous SDF (HSDF)
- Cyclo-Static Dataflow (CSDF) [Lauwereins 1994]
  - A cyclic state machine unconditionally advances at each firing
  - Known number of tokens produced and consumed for each state

# POSIX-Level Programming Environment



- POSIX-like process management
  - Spawn 16 processes from the I/O subsystem
  - Process execution on the 16 clusters start with main(argc, argv) and environment
- Inter Process Communication (IPC)
  - POSIX file descriptor operations on 'NoC Connectors'
  - Inspired by supercomputer communication and synchronization primitives
- Multi-threading inside clusters
  - Standard GCC/G++ OpenMP support
    - #pragma for thread-level parallelism
    - Compiler automatically creates threads
  - POSIX threads interface
    - Explicit thread-level parallelism





# POSIX IPC with NoC Connectors



- Build on the ‘pipe & filters’ software component model
  - Processes are the atomic software components
  - NoC objects operated through file descriptors are the connectors:

| Connector | Purpose                      | Tx:Rx Endpoints      | Resources |
|-----------|------------------------------|----------------------|-----------|
| Sync      | Half synchronization barrier | N:1, N:M (multicast) | CNoC      |
| Portal    | Remote memory window         | N:1, N:M (multicast) | DNoC      |
| Sampler   | Remote circular buffer       | 1:1, 1:M (multicast) | DNoC      |
| RQueue    | Remote atomic enqueue        | N:1                  | DNoC+CNoC |
| Channel   | Zero-copy rendez-vous        | 1:1                  | DNoC+CNoC |

- Synchronous operations: `open()`, `close()`, `ioctl()`, `read()`, `write()`, `pwrite()`
- Asynchronous I/O operations on Portal, Sampler, RQueue
  - Based on `aio_read()`, `aio_error()`, `aio_return()`
  - NoC Tx DMA engine activated by `aio_write()`

# Streaming Programming Environment



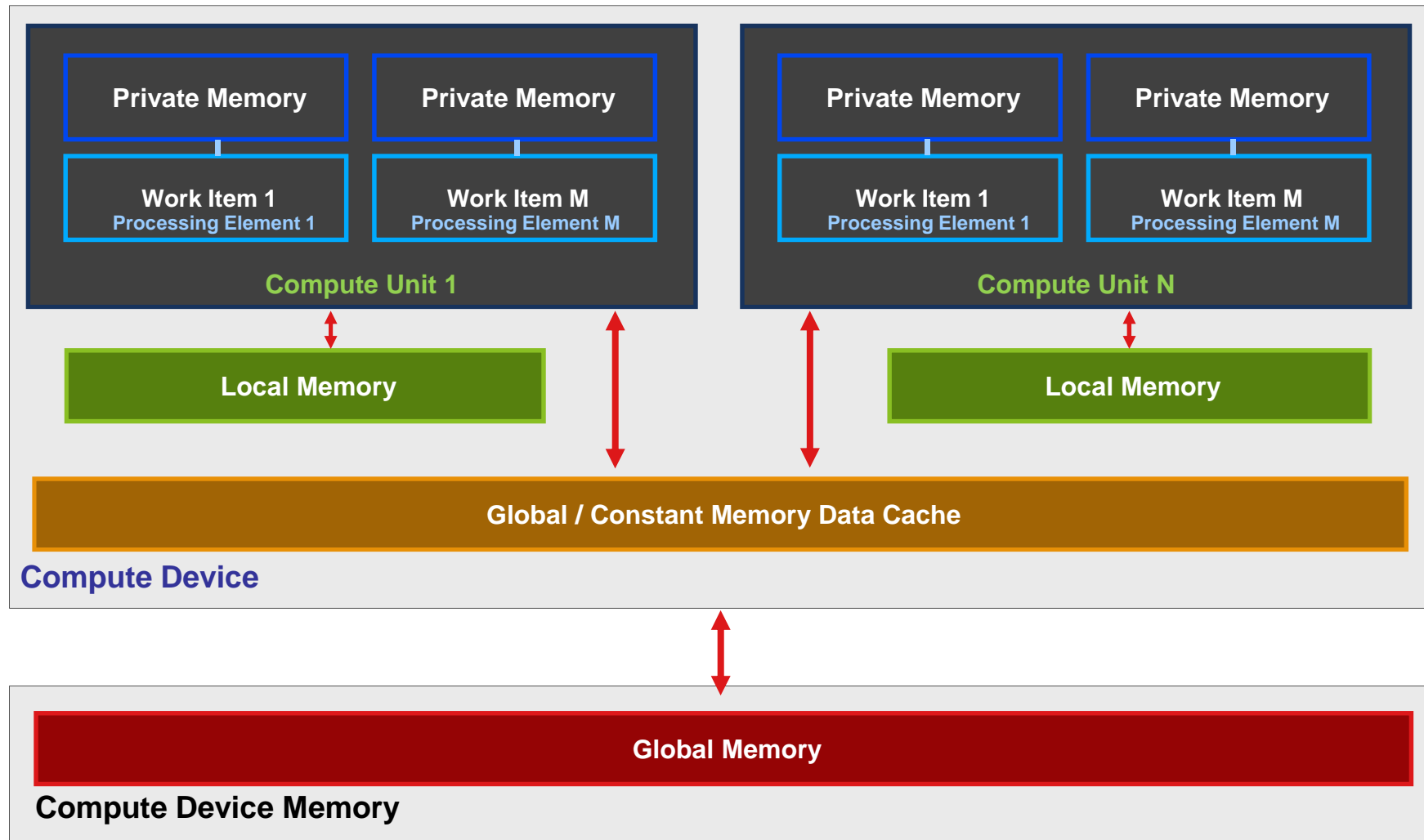
- MPPA<sup>®</sup> support of OpenCL
  - Task parallel model: one kernel per compute cluster
    - Native kernel mode: `clEnqueueNativeKernel()`
    - Standard task parallel mode: `clEnqueueTask()`
  - Emulate global memory with Distributed Shared Memory (DSM)
    - Use the MMU on each core, assume no false sharing
    - Use the MMU on each core, resolve false sharing like Treadmarks
  
- MPPA<sup>®</sup> Bulk Synchronous Streaming
  - Adapt Bulk Synchronous Parallel (BSP) model to the MPPA<sup>®</sup>
    - Execute a number of cluster processes > number of clusters
    - Double buffering to overlap cluster process execution and swapping

# OpenCL Programming Environment



- Standard OpenCL has two programming models
  - Data parallel, with one work item per processing element (core)
  - Task parallel, with one work item per compute unit (multiprocessor)
    - In native kernels, may use standard C/C++ static compiler (GCC)
- MPPA<sup>®</sup> support of OpenCL
  - Task parallel model: one kernel per compute cluster
    - Native kernel mode: `clEnqueueNativeKernel()`
    - Standard task parallel mode: `clEnqueueTask()`
  - Emulate global memory with Distributed Shared Memory (DSM)
    - Use the MMU on each core, assume no false sharing
    - Use the MMU on each core, resolve false sharing like Treadmarks
  - Data parallel model once LLVM is targeted to the Kalray VLIW core
    - Currently use LLVM to generate C99, which is compiled with GCC

## OpenCL Target Compute Platform



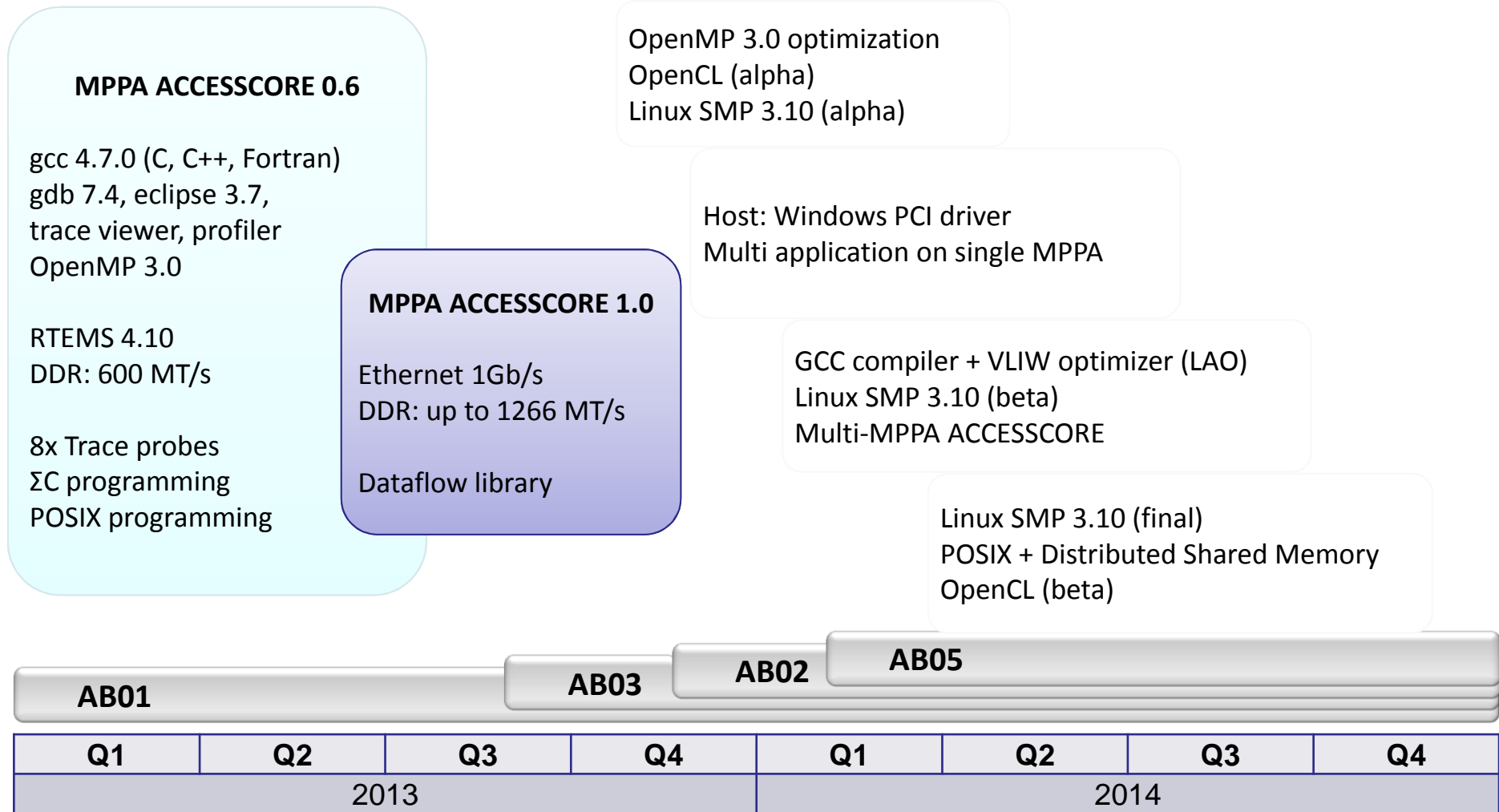
# MPPA<sup>®</sup> ACCESSLIB

## optimized application building blocks

- Application building blocks optimized at different scopes
  - MPPA Core – register file & cache
  - MPPA Cluster – shared memory
  - MPPA Partition – distributed memory
- Delivered as C libraries
  - Dataflow programming
  - POSIX-level programming
- Numerical and signal processing
  - FFT, Filtering and convolution
  - BLAS-level primitives
  - VSIPL (Vector Signal Image)
  - libm extensions with metalibm
- Video and image processing
  - H264, HEVC encode / decode
  - OpenCV Computer vision

| Scope     | 2013                   |              | 2014    |          |
|-----------|------------------------|--------------|---------|----------|
|           | Q3                     | Q4           | Q1      | Q2       |
| Core      | ★ DSPLIB<br>H264 codec | ★ HEVC Codec |         |          |
| Cluster   |                        | ★ DSPLIB     | ★ VSIPL |          |
| Partition | ★ Matmul, FFT          |              | ★ PBLAS | ★ OpenCV |

# MPPA Software Roadmap



# Addressable Market Segments

## INTENSIVE COMPUTING

- Oil and Gas
- Finance
- Numerical Simulation
- Life sciences



## IMAGE & VIDEO

- Video Broadcast
- Medical Imaging
- Digital Cinema
- Computer Vision



## EMBEDDED SYSTEMS

- Signal Processing
- Aerospace/Defence
- Transport
- Industrial Automation

(Opportunistic approach)



## TELECOM

- Network Traffic Analysis
- Security Services
- Software Defined Radio

(Market under investigation)

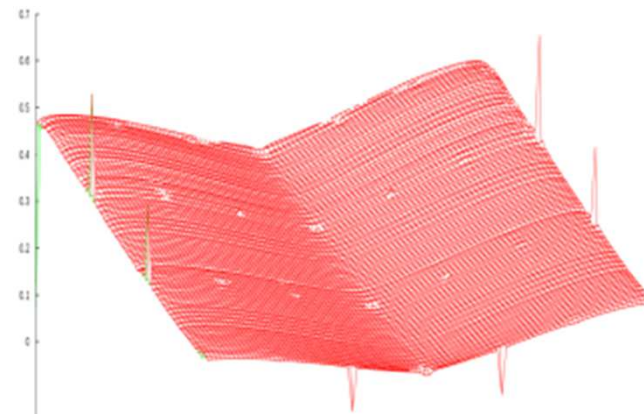
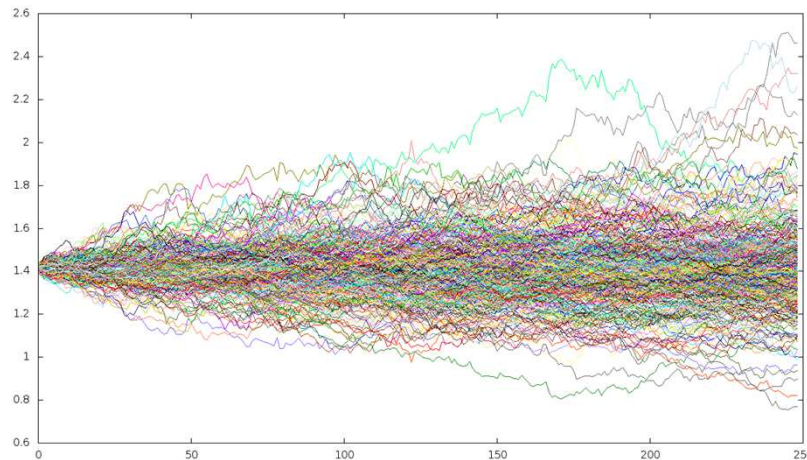


Kalray also serves the **Academic market**  
(universities and research institutions)



## Computational Finance Application

- Option pricing by Monte Carlo method
- Optimized pseudo random generator
- Parallel Map / Reduce scheme across multiple MPPA processors
- Optimized mathematical primitives for Kalray core

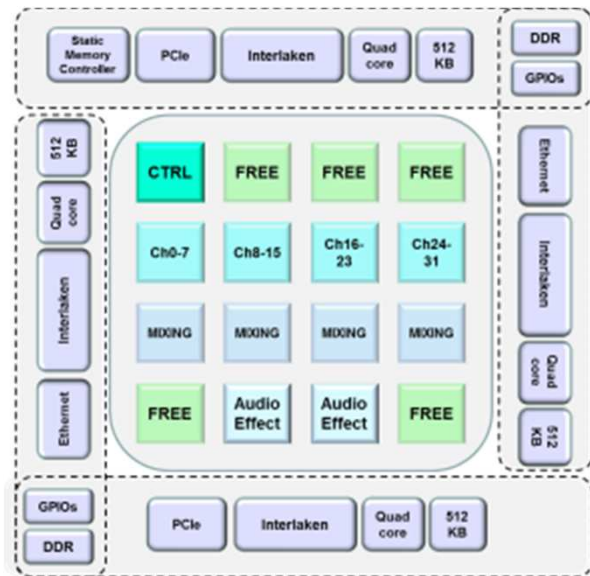


**Power efficiency 5x better than recent GPU**



# Audio Processing Application

*Increase performances and reduce total system cost*



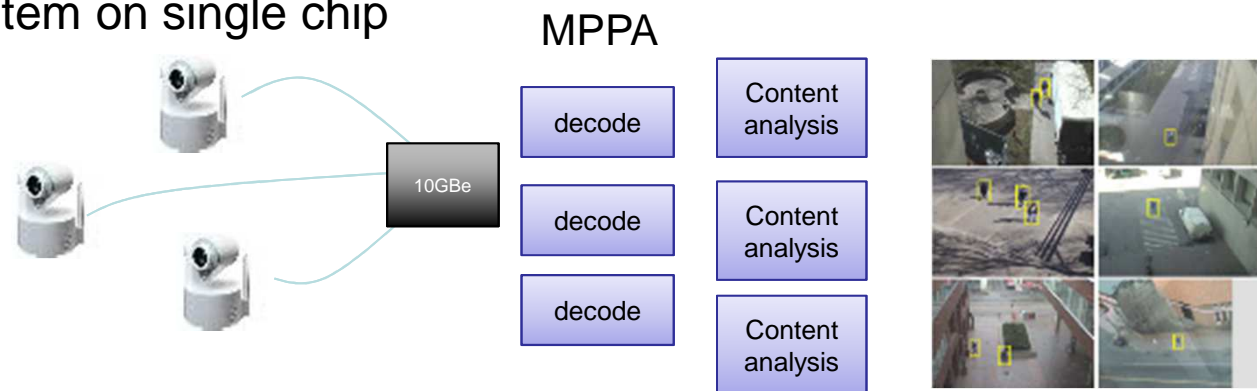
- Multi Channel processing
  - 256 VLIW cores ~ 500 Low End DSPs
- Channel routing and control
  - High performance NoC + 32 integrated DMAs
- System integration
  - Up to 8 x Ethernet 1GbE
- Low Latency audio processing
  - 500µs latency from input to output samples
- Cost effective
  - Equivalent to complex multi DSPs + FPGAs system



## Video Protection Example

- Improved Content analysis
  - High resolution camera / low false detection rate
  - Robust algorithms → high performance computing of MPPA
  - Real Time detection
  - More simple infrastructure → Compute power at the source
  
- System integration: Ethernet input / decode / content analysis / encode

- Multi camera system on single chip



## Augmented Reality Example

- Assisted operation & maintenance
  - ARMAR (Augmented Reality for Maintenance and Repair)

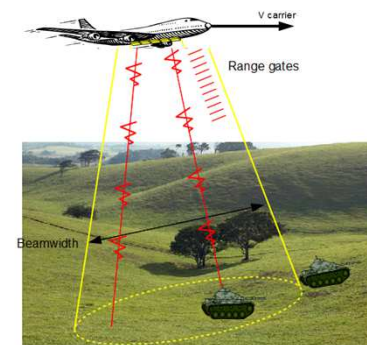
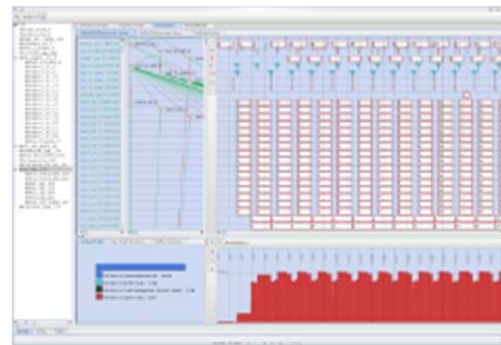
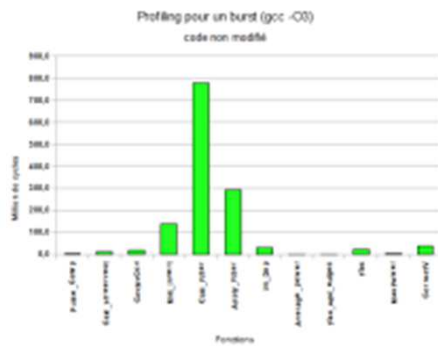


- Assisted conformity control



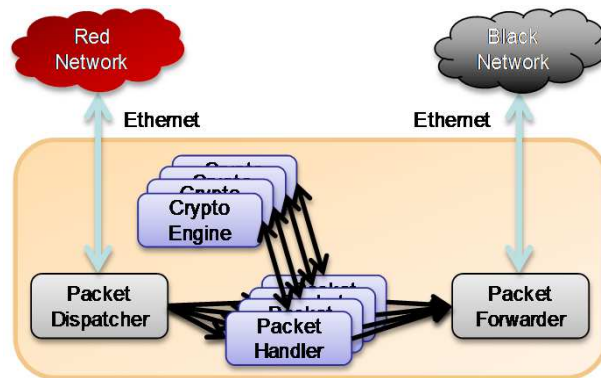
# Signal Processing Example

- Radar applications: STAP, ...
- Beam forming : Sonar, Echography
- Software Defined Radio (SDR)
- Dedicated libraries (FFT, FTFR, ... )



**Well suited to massively parallel architectures**  
**Alternative of embedded DSP + FPGA platforms**

# High-Speed VPN Gateway Example

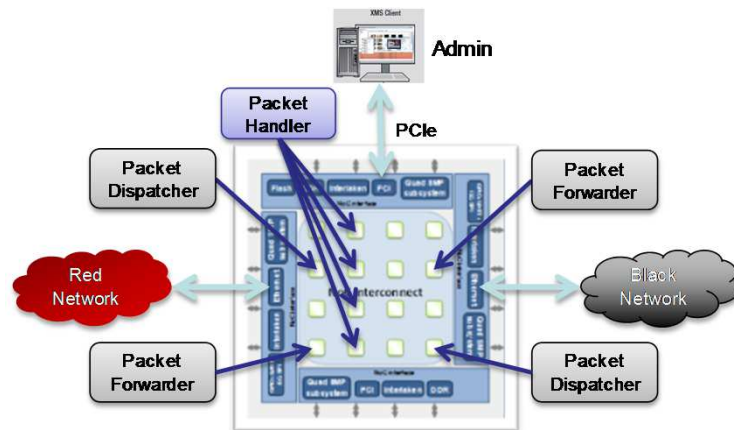


Evaluation for the implementation of a 20 to 40 Gbs VPN gateway

- IP packet processing
- AES cryptography

Exploit key features of the MPPA architecture

- 2 x 40 Gbs Ethernet interfaces (or 8 x 10 Gbs)
- PCIe Gen 3 for integration
- Optimized instructions for efficient cryptography
- NoC extension interface for multi-chip solutions



# Kalray Offices

## *Headquarters – Paris area*

86 rue de Paris,  
91 400 Orsay  
France

Tel: +33 (0)1 69 29 08 16  
email: [info@kalray.eu](mailto:info@kalray.eu)



## *Grenoble office*

445 rue Lavoisier,  
38 330 Montbonnot Saint Martin  
France

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



## *Japan office*

CVML, 3-22-1, Toranomom,  
Minato-ku, Tokyo 105-0001,  
Japan

Tel: 080-4660-2122  
email: [ksugiyama@kalray.eu](mailto:ksugiyama@kalray.eu)

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.