AXIOM

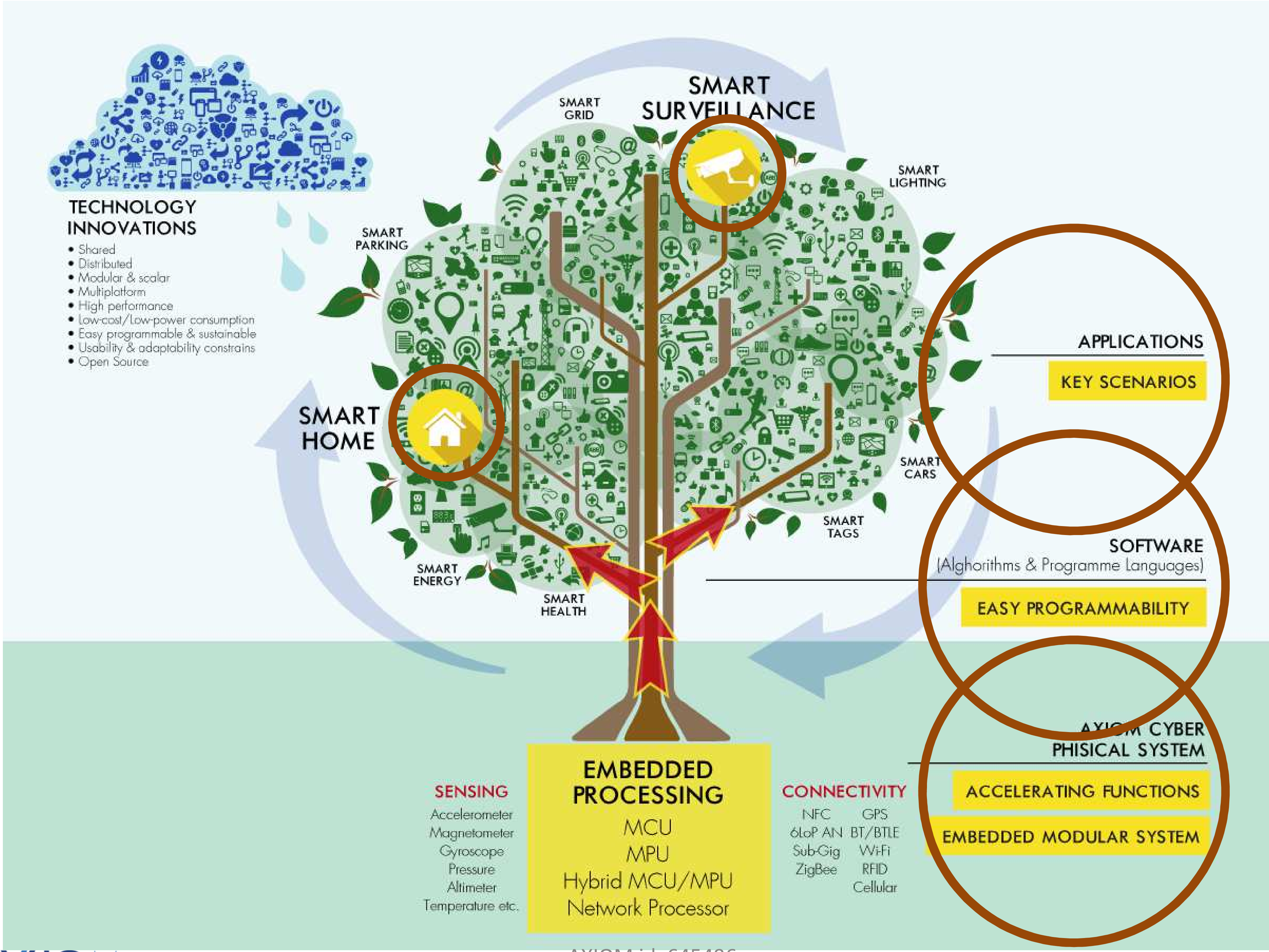
Agile, eXtensible, fast I/O Module for the cyber-physical era

De-CPS 2016: Workshop on Challenges and New Approaches
for Dependable and Cyber-Physical System Engineering
Pisa, Italy, 17 June 2016

Modeling Multi-Board Communication in the AXIOM Cyber-Physical System

Roberto Giorgi, Somnath Mazumdar, Stefano Viola, Paolo Gai, Stefano Garzarella, Bruno Morelli, Dionisios Pnevmatikatos, Dimitris Theodoropoulos, Javier Bueno, Antonio Filgueras, Daniel Jimenez-Gonzalez, Carlos Alvarez, Xavier Martorell, Eduard Ayguade

University of Siena
(Coordinator Partner)



TECHNOLOGY INNOVATIONS

- Shared
- Distributed
- Modular & scalar
- Multiplatform
- High performance
- Low-cost/Low-power consumption
- Easy programmable & sustainable
- Usability & adaptability constrains
- Open Source

APPLICATIONS

KEY SCENARIOS

SOFTWARE (Alghorithms & Programme Languages)

EASY PROGRAMMABILITY

AXIOM CYBER PHISICAL SYSTEM

ACCELERATING FUNCTIONS

EMBEDDED MODULAR SYSTEM

SENSING

Accelerometer
Magnetometer
Gyroscope
Pressure
Altimeter
Temperature etc.

EMBEDDED PROCESSING

MCU
MPU
Hybrid MCU/MPU
Network Processor

CONNECTIVITY

NFC	GPS
6LoP AN	BT/BTLE
Sub-Gig	Wi-Fi
ZigBee	RFID
	Cellular

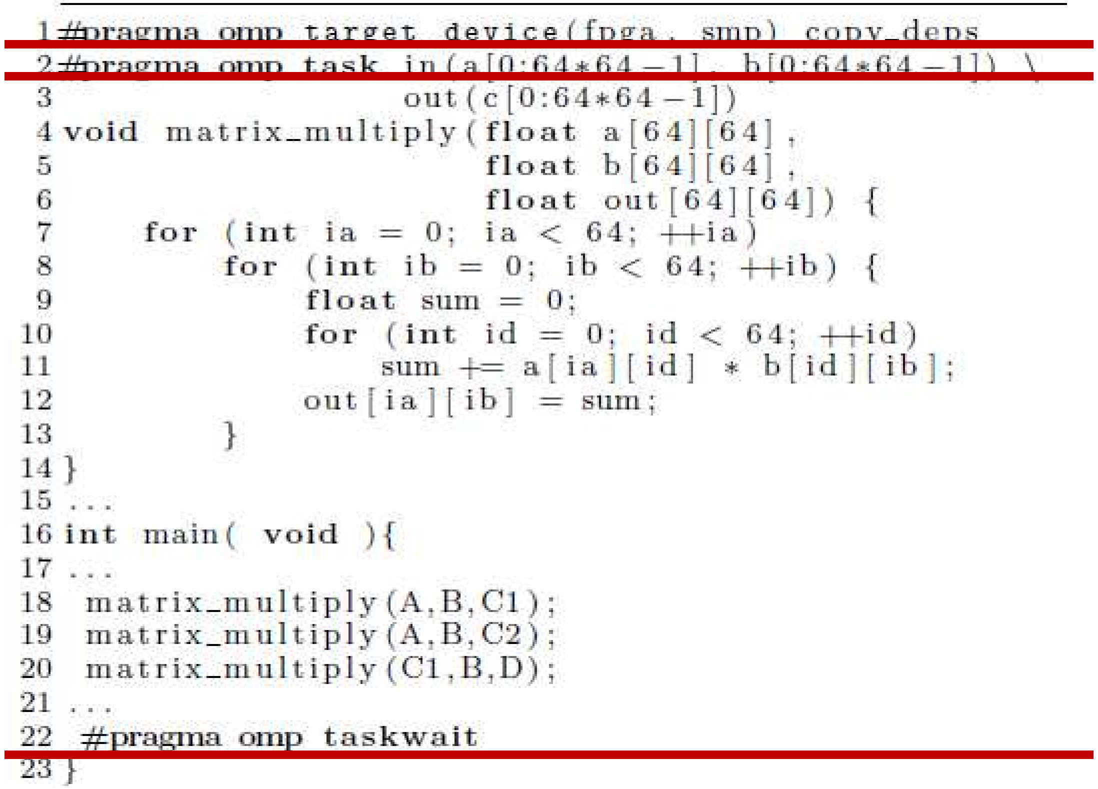
Agenda

1) Introduction on the AXIOM *IoT stack*
2) Importance of the high-speed interconnect
3) Details about the interfaces
4) Initial results

AXIOM OBJECTIVES

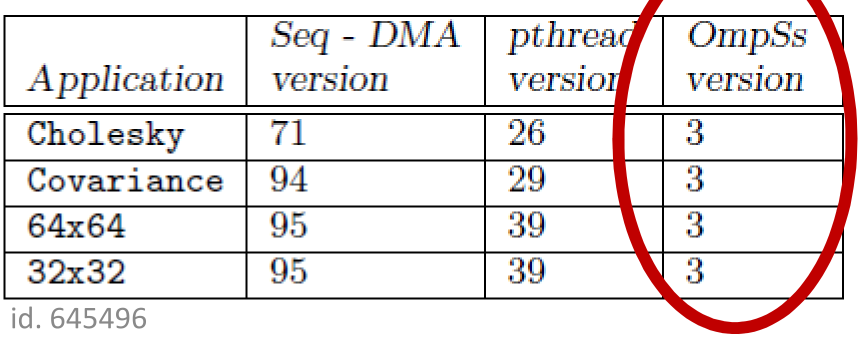
- **OBJ1) Realizing a small board that is flexible, energy efficient and modularly scalable**
  - Flexibility: FPGA provides acceleration, custom interconnects, ability to distribute threads across boards
  - Energy efficiency: low-power ARM, FPGA
  - Modularly scalable: fast+inexpensive interconnects based on SATA/USB-C, distributed shared memory across boards
- **OBJ2) Easy programmability of multi-core, multi-board, FPGA**
  - Programming model: Improved OmpSs
  - Runtime & OS: improved thread management
- **OBJ3) Leveraging Open-Source software to manage the board**
  - Compiler: BSC Mercurium
  - OS: Linux
  - Drivers: provided as open-source by partners
- **OBJ4) Easy Interfacing with the Cyber-Physical World**
  - Cyber-Physical World: integrating Arduino support for a plenty of pluggable board (so-called "shields")
  - Platform: building on the UDOO experience from SECO
- **OBJ5) Enabling real time movement of threads**
  - Runtime: will leverage the EVIDENCE's SCHED_DEADLINE scheduler (i.e. EDF) included Linux 3.14, UNISI's low-level dataflow-based thread management techniques
- **OBJ6) Contribution to Standards**
  - Hardware: SECO is founding member of the Standardization Group for Embedded Systems (SGET)
  - Software: BSC is member of the OpenMP consortium

EASY PROGRAMMABILITY VIA OPENMP-SS (OMPSS)



```
#pragma omp target device(fpga, smp) copy_deps
#pragma omp task in(a[0:64*64-1], b[0:64*64-1]) \
                 out(c[0:64*64-1])
void matrix_multiply(float a[64][64],
                     float b[64][64],
                     float out[64][64]) {
    for (int ia = 0; ia < 64; ++ia)
        for (int ib = 0; ib < 64; ++ib) {
            float sum = 0;
            for (int id = 0; id < 64; ++id)
                sum += a[ia][id] * b[id][ib];
            out[ia][ib] = sum;
        }
}
...
int main( void ){
...
 matrix_multiply(A,B,C1);
 matrix_multiply(A,B,C2);
 matrix_multiply(C1,B,D);
...
 #pragma omp taskwait
}
```

Only 3 lines of code to
- accelerate code on FPGAs
- distributed code across
  several AXIOM boards



Application	*Seq - DMA version*	*pthread version*	*OmpSs version*
Cholesky	71	26	3
Covariance	94	29	3
64x64	95	39	3
32x32	95	39	3

AXIOM – THE MODULE

- KEY ELEMENTS
  - K1: ZYNQ FPGA (INCLUDES 6 ARM CORES)
  - K2: ARM GP CORE(S)
  - K3: HIGH-SPEED & INEXPENSIVE INTERCONNECTS
  - K4: SW STACK – OMPSS+LINUX BASED
  - K5: OTHER I/F (ARDUINO, USB, ETH, WIFI, …)

CAN WE DO THAT ?

- SECO/UNISI achievements:
  - 2014: UDOO-ARM (99 $ PC+Arduino) → 600k$ on Kickstarter
  - 13th April 2016: UDOO-x86 (PC+Arduino, 10x faster than Raspberry-3) → 100k$ in 7 hours (!) on Kickstarter

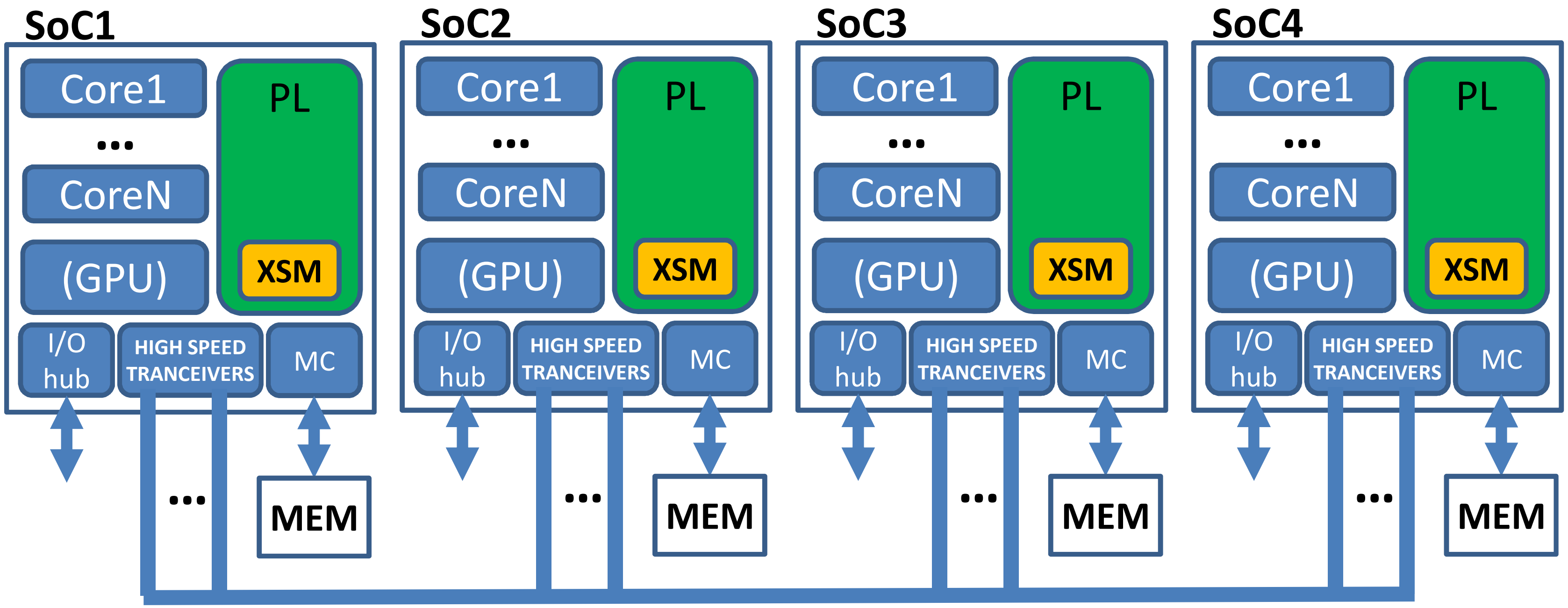
UDOO is a mini PC that could run either Android or Linux, with an Arduino - compatible board embedded

UDOO is an open hardware, low-cost computer equipped with an ARM i.MX6 Freescale processor for Android and Linux, alongside Arduino DUE's ARM on the same board!

UDOO X86 FOR CLUSTER

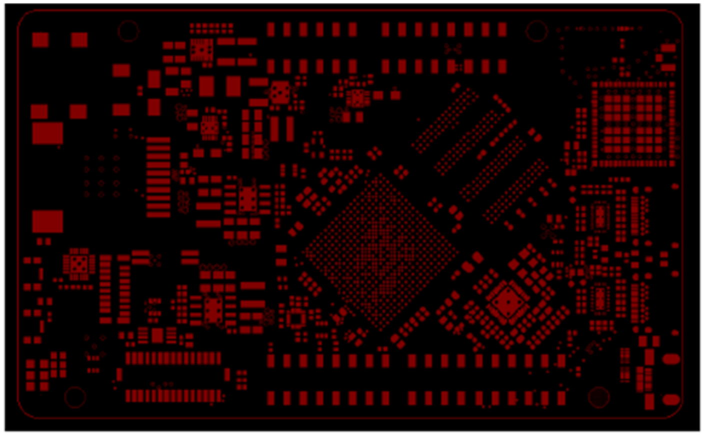
POWERED BY OmpSs
THANKS TO AXIOM

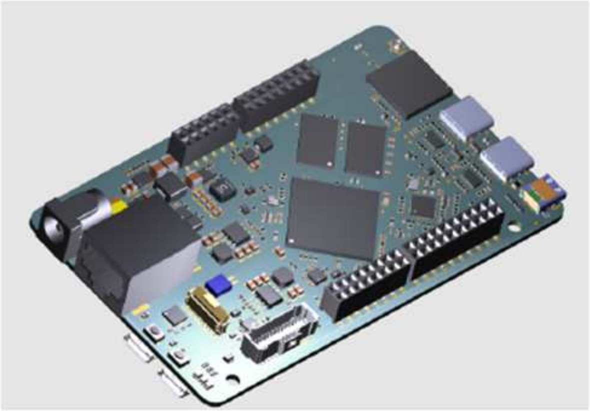
AXIOM Cluster Architecture



First AXIOM Board – under review

- Stackup definition
  - 10 layers PCB
  - HS/LS/power planes arrangement
- Placement
  - Achieve mechanical and electrical constraints
- Routing (WIP)
  - Design for power/signal integrity
- 3D model available





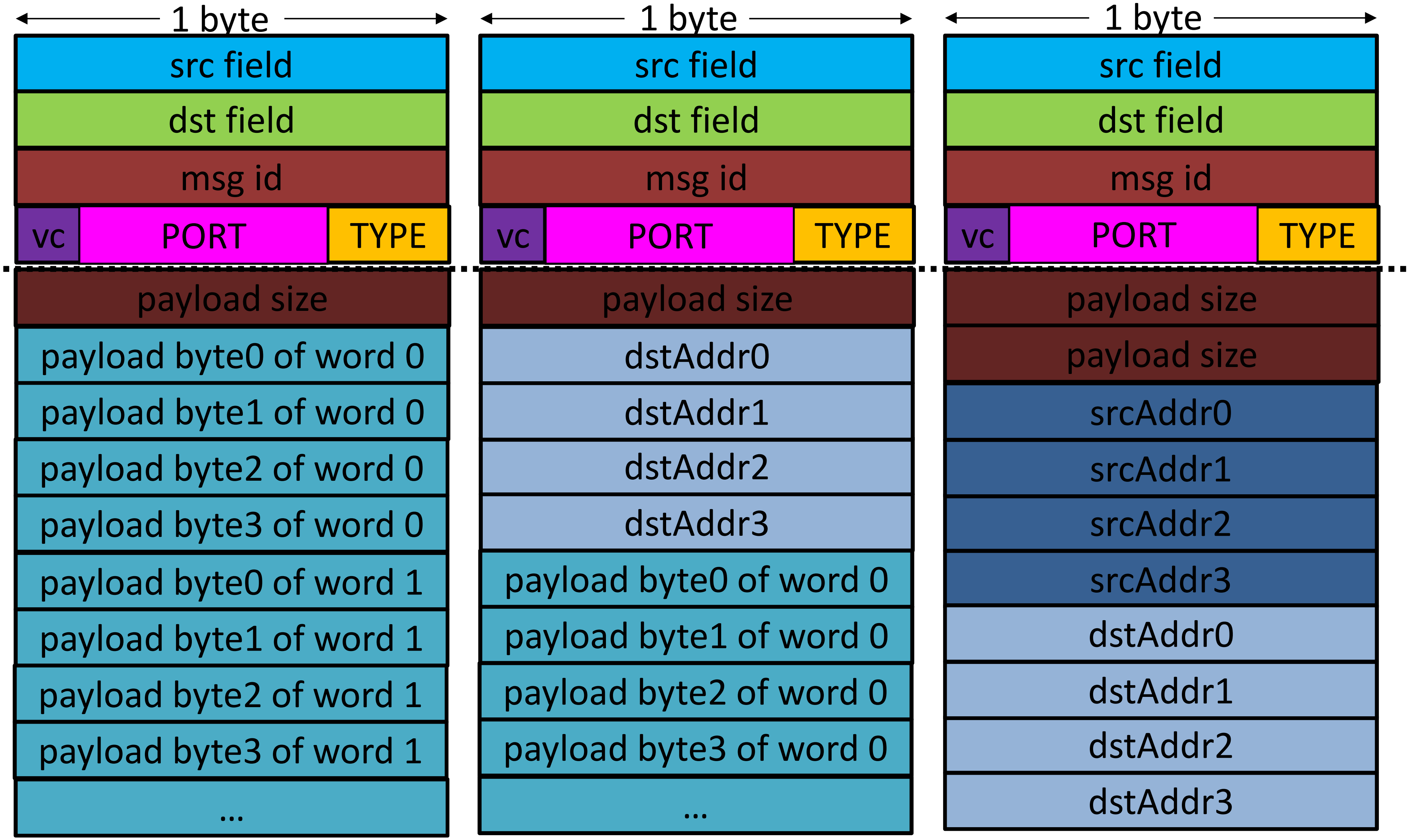
Axiom NIC: type of messages

- Multiple type of messages
  - Small messages (payload embedded in the descriptor)
    - RAW
      - Very short message (up to 128 bytes)
    - RAW NEIGHBOUR
      - RAW message to neighbour node
  - Big messages (payload as a pointer in the descriptor)
    - RDMA read/writes
      - Remote DMA transfer between two nodes
    - LONG
      - Based on RDMA, but without specify a destination address
      - Pool of buffers provided by the receiver node

Axiom NIC: queues

- Multiple queue available
  - TX/RX small messages
    - RAW and RAW NEIGHBOUR
  - TX/RX big messages
    - LONG and RDMA
- Port in each descriptor
  - To address process in the node
  - Process can bind one or more ports

AXIOM Interconnect packets



1 byte

src field

dst field

msg id

vc PORT TYPE

payload size

payload byte0 of word 0

payload byte1 of word 0

payload byte2 of word 0

payload byte3 of word 0

payload byte0 of word 1

payload byte1 of word 1

payload byte2 of word 1

payload byte3 of word 1

…

send raw data

1 byte

src field

dst field

msg id

vc PORT TYPE

payload size

dstAddr0

dstAddr1

dstAddr2

dstAddr3

payload byte0 of word 0

payload byte1 of word 0

payload byte2 of word 0

payload byte3 of word 0

…

send data from local memory to remote address

1 byte

src field

dst field

msg id

vc PORT TYPE

payload size

payload size

srcAddr0

srcAddr1

srcAddr2

srcAddr3

dstAddr0

dstAddr1

dstAddr2

dstAddr3

read data from remote node

Initial network configuration

- Hypothesis
  - Makers will likely connect nodes in a random way
  - They will expect to connect to a single node to run an application on the cluster
- Prerequisite
  - One node used as an interface to the cluster:
    - Usually called *master* node
    - E.g.: connected with ethernet or Wi-Fi
- Initialization
  - The master node starts a discovery algorithm
    - Node_id assigned on each node
  - Then the routing table is computed and distribited
  - After that, the network is fully working

AXIOM Recursive Discovery Algorithm

```
int ax_discovery(node, next_id) {
        node.my_id = next_id++;
        for <each neighbour> {
                if <neighbour node already have an ID> {
                        <skip it>
                } else {
                        next_id = ax_discovery(neighbour, next_id);
                }
        }
        return next_id;
}

/* start the discovery algorithm on the master node */
next_id = 0;
ax_discovery(master, next_id);
```

Initial results

Programming Model Level

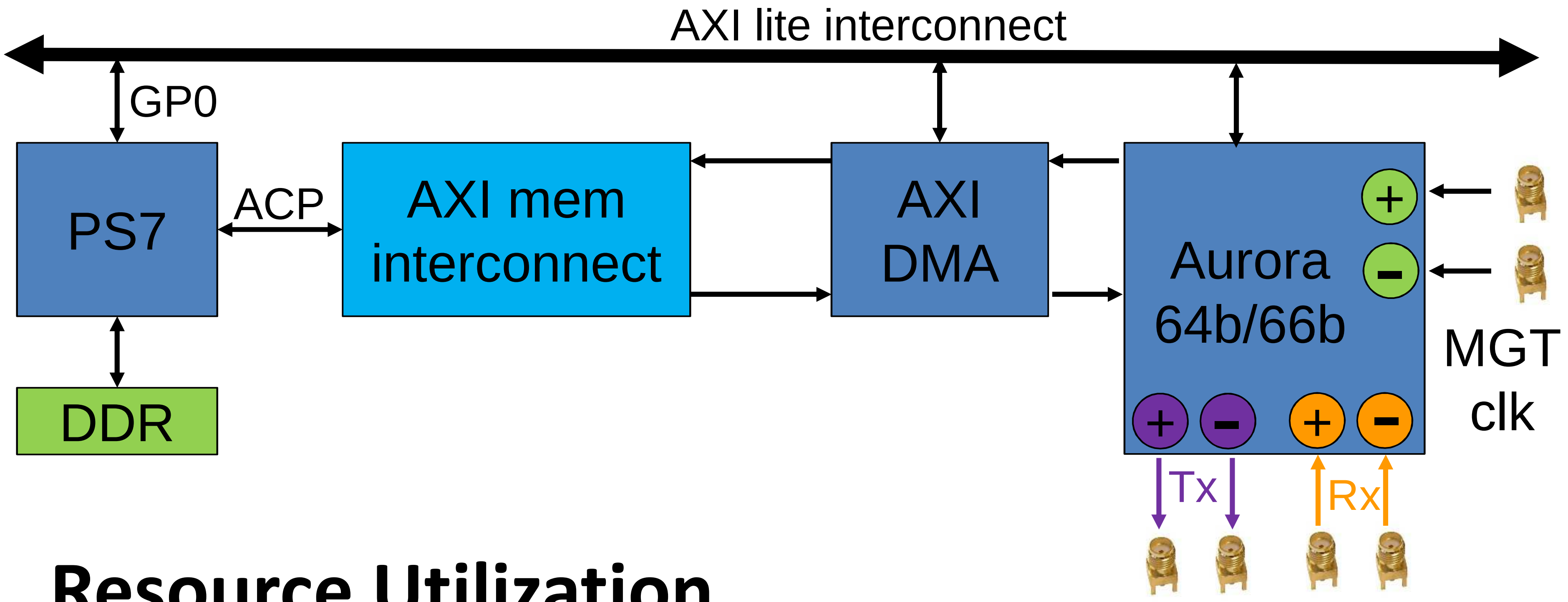
- Message Statistics

	Msg. size (bytes)	Avg. messages/s
Matmul	536 - 22000 - 32500	68 - 74
N-body	3072 - 5400 - 8192	62 - 107

- Application Performance (Gflops)

App / cores per node	**1**	**2**	**3**	**4**
Matmul (1 node)	0.28	0.57	0.84	1.11
Matmul (2 nodes)	0.52	1.01	1.52	1.54
N-body (1 node)	0.15	0.30	0.46	0.58
N-body (2 nodes)	0.17	0.35	0.61	0.72

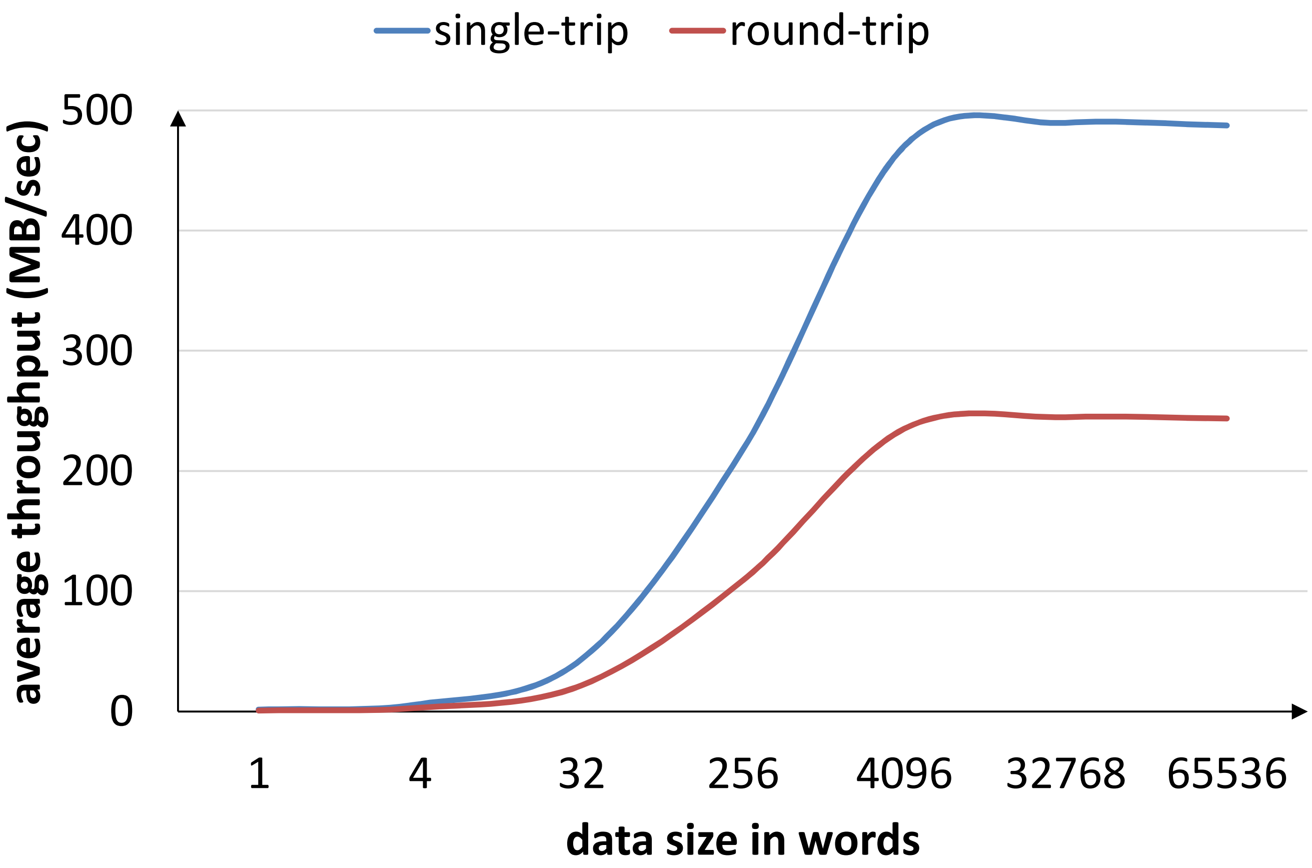
Block Diagram of Two ZC706 FPGA Boards Connected With SMA Cables



Resource Utilization

LUTs	LUTs (%)	FFs	FFs (%)	BRAMs	BRAMs (%)
10583	4.8%	12452	2.8%	5	0.9%

Achieved Throughput

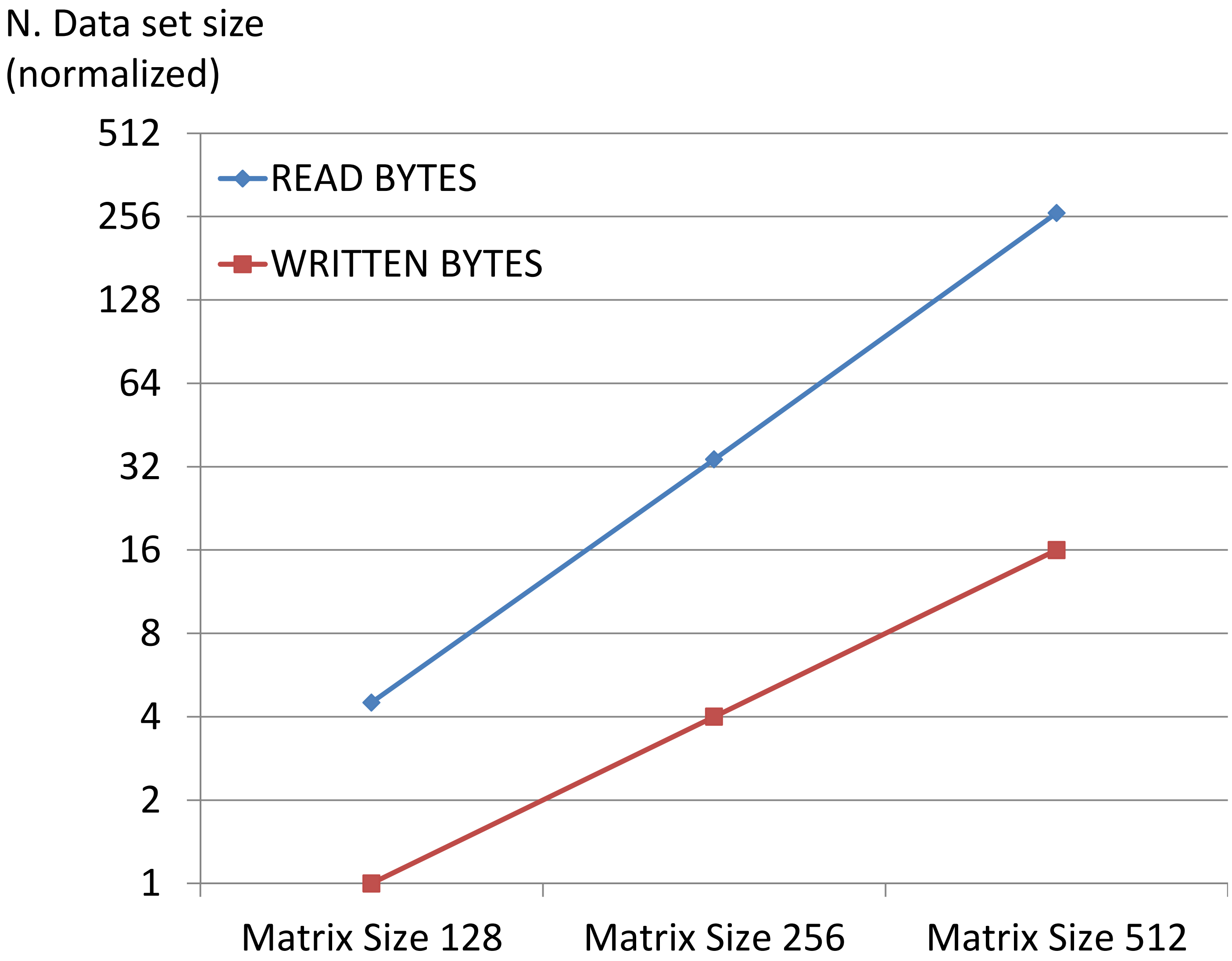


Network throughput when using the Aurora IP to exchange data between two ZC706 FPGA boards with a single MGT transceiver.

AXIOM NIC registers

- Have been emulated using QEMU on ARM64bit
  - Communication between virtual machine emulated using sockets on teh QEMU backend
- An initial implementation of the network in user space has been done
  - Discovery algorithm
  - Routing Algorithm
  - The nodes are able to deliver small messages using user space programs

DF-Threads Initial Results



Conclusions

- AXIOM is going to develop an innovative board communication infrastructure that will power next generation IoT boards
- The key message is:
  - Cluster of small boards
  - With uniform programming language based on OpenMP
  - Optimized thanks to FPGA and custom network with RDMA

European Commission

The EU Framework Programme for Research and Innovation

HORIZON 2020

AXIOM

Agile, eXtensible, fast I/O Module for the cyber-physical era

PROJECT ID: 645496

AXIOM project --- http://www.axiom-project.eu

S: UNIVERSITATIS SENARUM MCCXXXX

University of Siena
(Coordinator Partner)

BSC Barcelona Supercomputing Center
Centro Nacional de Supercomputación

UPC

EVIDENCE® EMBEDDING TECHNOLOGY

herta security

FORTH Institute of Computer Science

SECO

VIMAR energia positiva