MANGO

exploring Manycore Architectures for Next-GeneratiOn HPC systems

The MANGO Process for Designing and Programming Multi-Accelerator Multi-FPGA Systems

H2RC'18
Sunday, November 11th, 2018
Dallas, Texas, USA

Rafael Tornero, José Flich, José María Martínez, Tomás Picornell, Vincenzo Scotti
Email: ratorga@disca.upv.es

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 671668

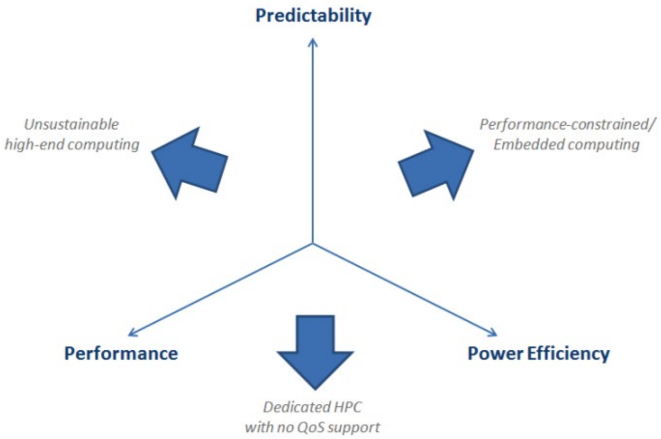
MANGO Context

- MANGO **FETHPC**-2014 project:
  - is about ***manycore architecture exploration in HPC***

Future and Emerging Technologies

- HPC quest for performance/power improvement
  - Trend in using heterogeneous components
    - GPUs, manycores, and even FPGAs
    - Goal is to get closer to the Intrinsic Computational Efficiency (ICE)
  - MANGO focuses on **heterogeneity**
    - How we combine heterogeneous components for the best achievement of computational efficiency
    - How to program/manage them for the best achievement of computational efficiency

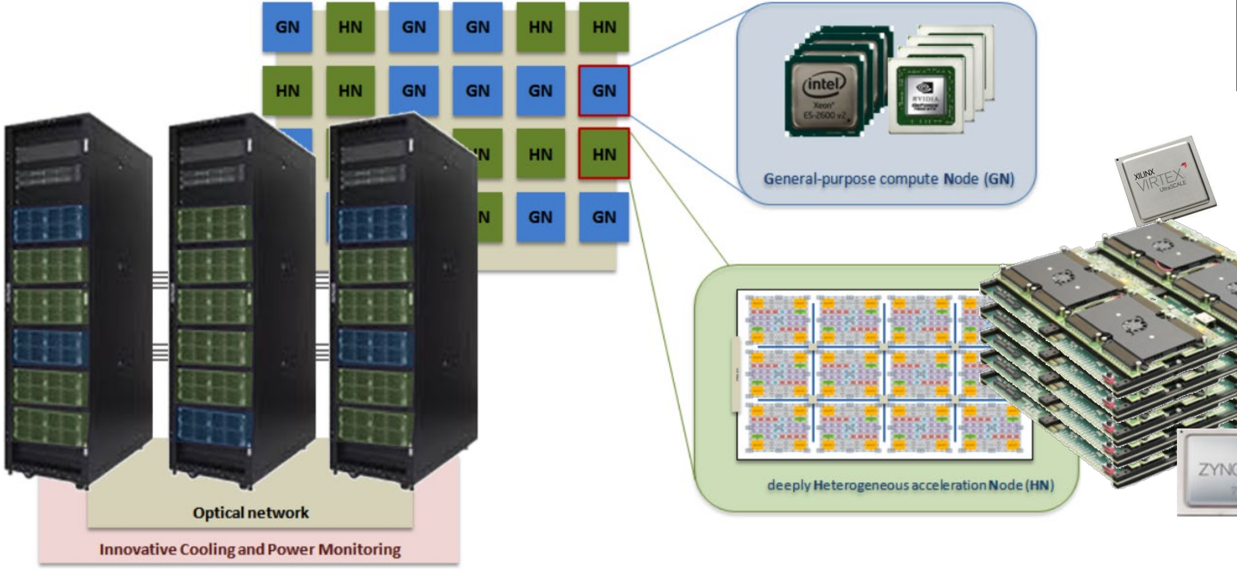
- Emerging requirements on HPC systems:
  - Predictability (QoS; time sensitivity)
    - Due to the merging of HPC with Big Data
  - Capacity computing
    - Run as many application instances as possible
  - MANGO addresses **predictability** and **capacity computing**
    - **3P model** (Performance/Power/Predictability)

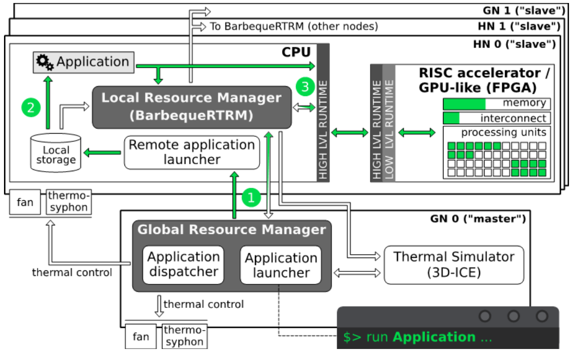

Predictability
Unsustainable high-end computing
Performance-constrained/ Embedded computing
Performance
Power Efficiency
Dedicated HPC with no QoS support

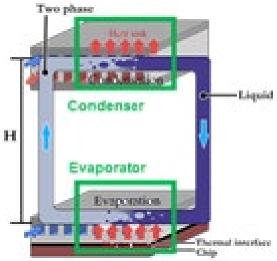
MANGO Context

- MANGO builds a **prototyping system** for 3P space exploration
- Goals:
  - Hardware
    - Develop a flexible prototype for rapid exploration of architectures
    - Explore new deeply heterogeneous manycore architectures
    - Real-time support exploring the PPP design space
    - Provide a unified and simple (homogeneous) access to the system via a smart interconnect
  - Software
    - Adapt programming models and compiler support to the new architectures
    - Develop the right resource manager to deal with the system
  - Infrastructure
    - Provide new monitoring tools to the system
    - Provide new cooling techniques to the system
  - Applications
    - Analyze impact of on a set of real applications
    - Support of video transcoding, medical imaging, security and surveillance applications

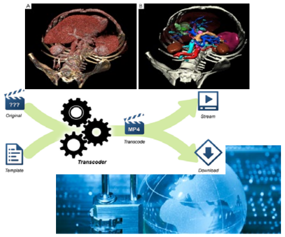
MANGO Prototype


GN HN GN GN HN HN
HN HN GN GN GN GN
HN HN HN
N GN GN
intel Xeon E5-2600 v2
NVIDIA
General-purpose compute Node (GN)
XILINX VIRTEX UltraSCALE
deeply Heterogeneous acceleration Node (HN)
Optical network
Innovative Cooling and Power Monitoring





- General-purpose nodes (Xeon+GPGPU) coupled with Heterogeneous nodes, HNs:
  - A large-scale cluster of high-capacity FPGAs
  - A robust, scalable interconnect for a **multi-FPGA manycore** system
  - Will enable FPGA acceleration *at scale*:
    - → a key ingredient for the EsD roadmap
  - A continuum from FPGA emulation to the final physical platform (might be an ASIC manycore, FPGA, mixed…)
    - → ***under a stable software environment***
  - Native isolation and partitioning mechanisms for **QoS-aware capacity computing** HPC applications
- Two-phase passive **energy-efficient cooling**
- Demonstrated applications with stringent high-performance and QoS requirements



Consortium

UNIVERSITAT POLITÈCNICA DE VALÈNCIA

prodesign electronic gmbh

CeRICT

FER

THALES

EATON Powering Business Worldwide

POLITECNICO MILANO

EPFL ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

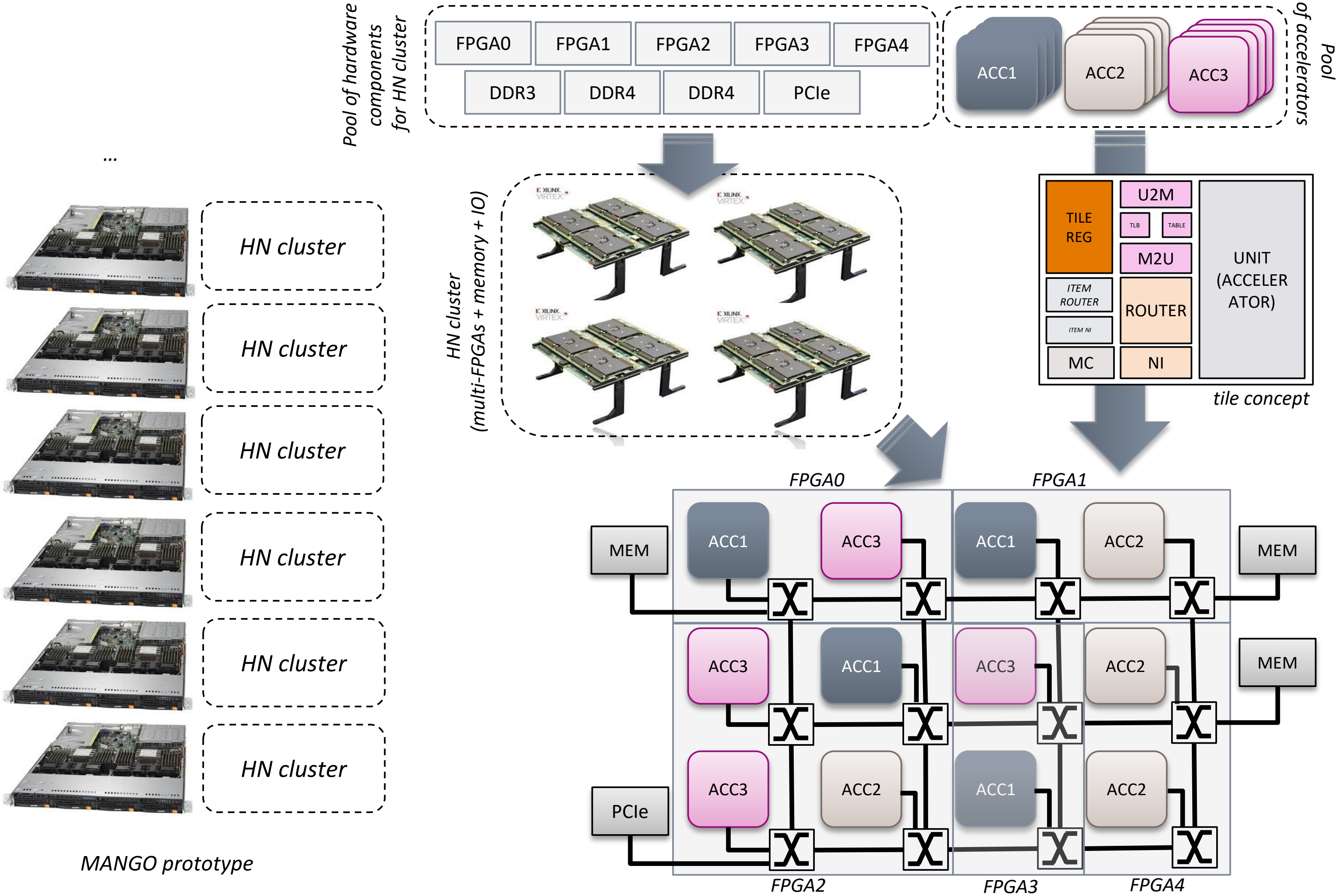
PHILIPS

MANGO: exploring Manycore Architectures for Next-GeneratiOn HPC systems

Agenda

- MANGO Architecture
  - HN Hardware and assembly
  - Heterogeneity
  - Network
  - Accelerator Interface
  - MANGO Design Flow
- FPGA resource utilization
- Conclusions

MANGO Architecture

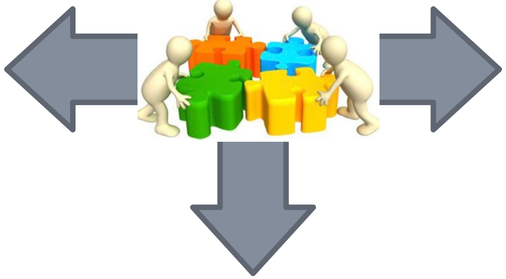


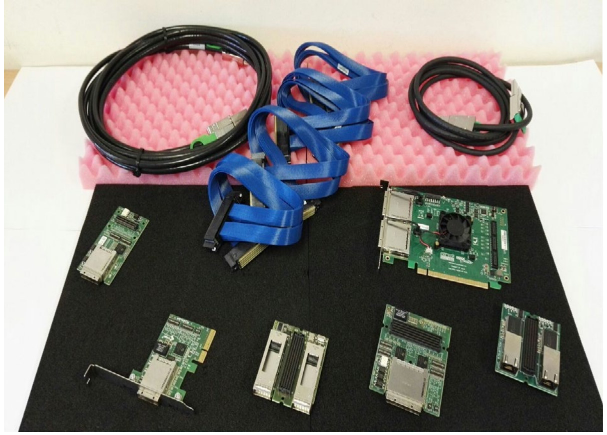
MANGO prototype

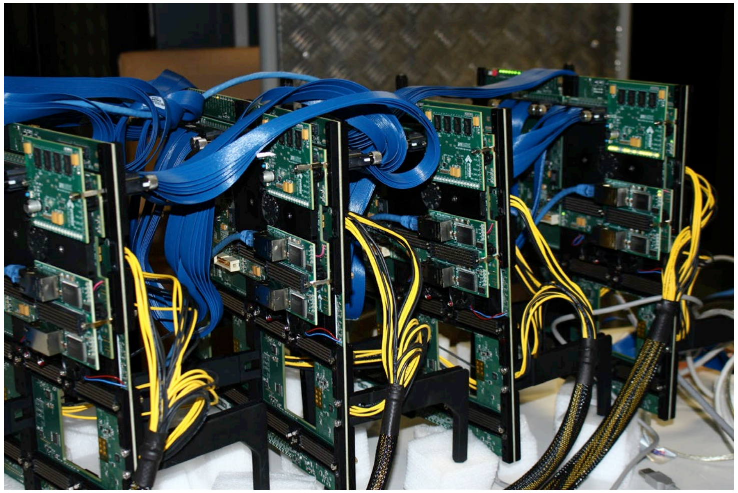
HN Hardware and Assembly

prodesign electronic gmbh

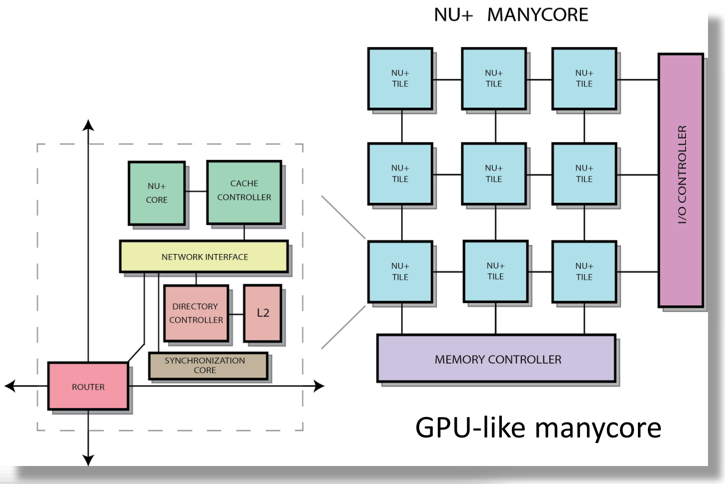
Lego-like excercise



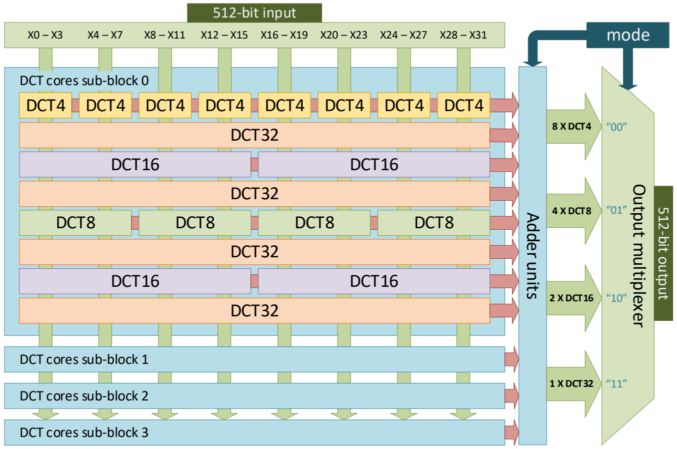


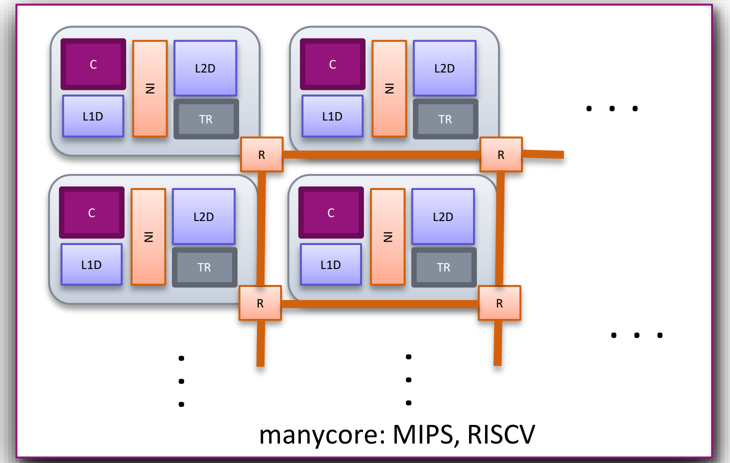


Heterogeneity: Pool of Accelerators

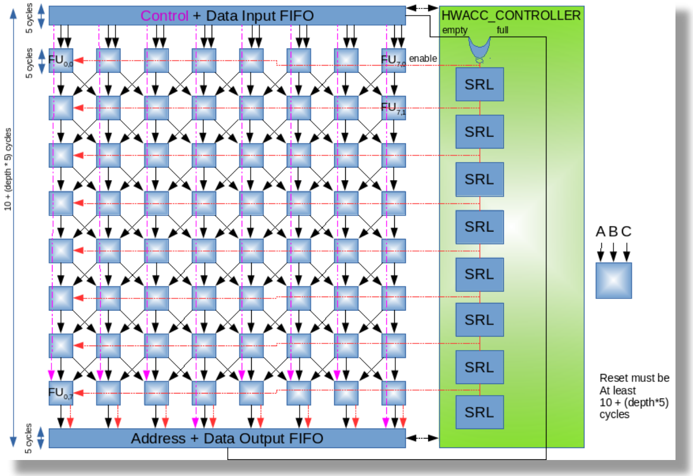


GPU-like manycore





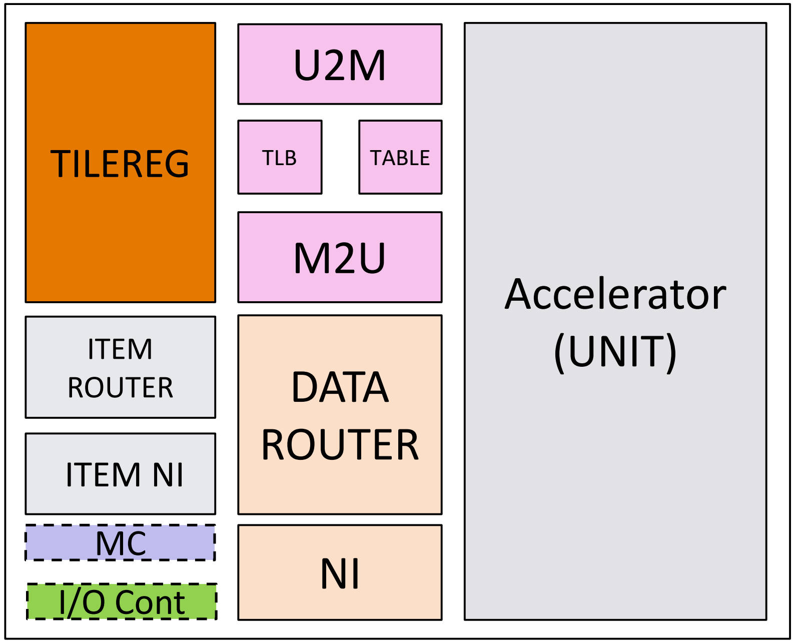
manycore: MIPS, RISCV



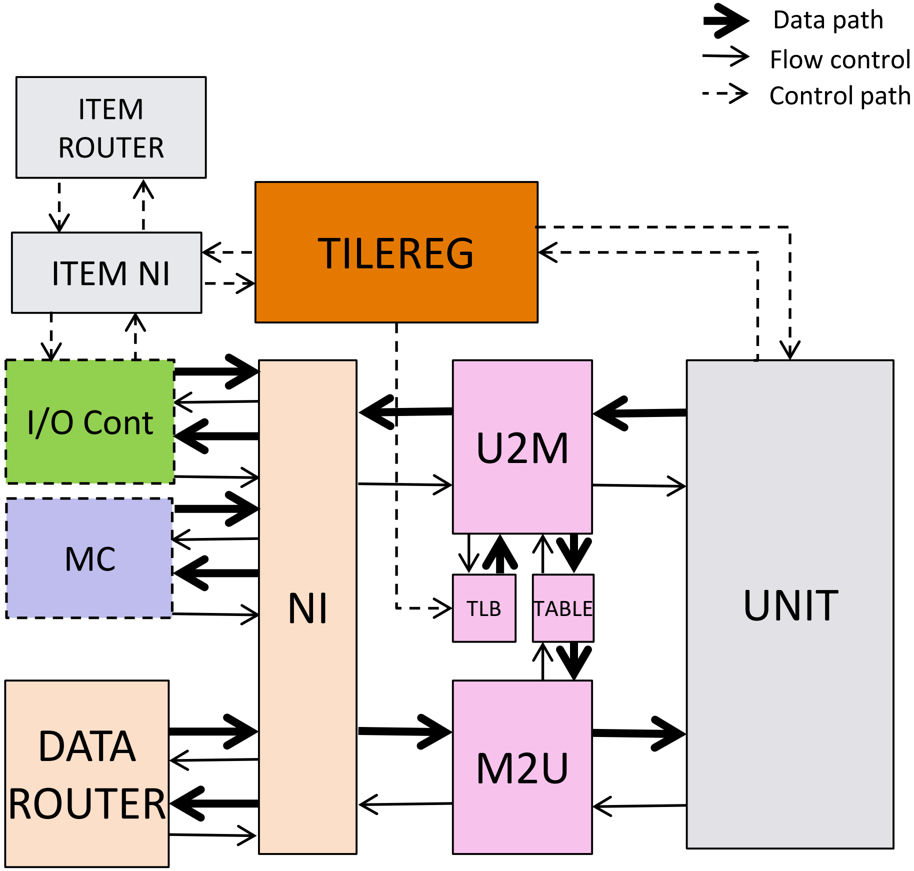
General purpose → Custom made

Heterogeneity

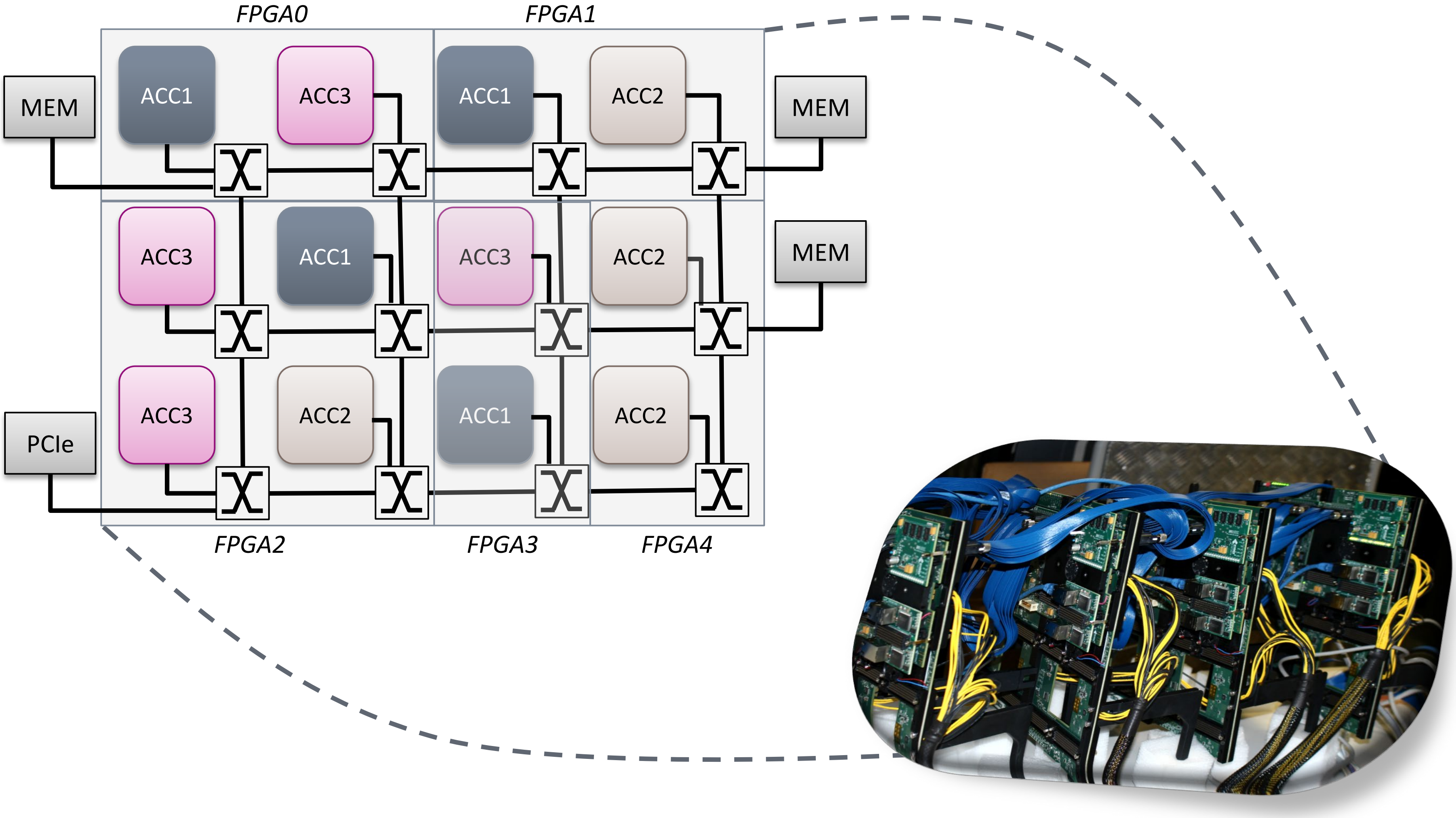
Tile concept



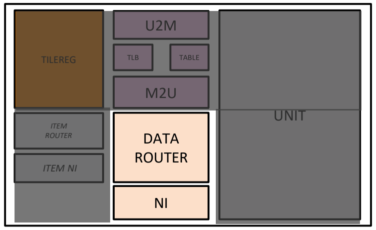
Element connection



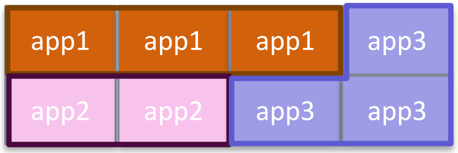
Heterogeneity, but regular layout



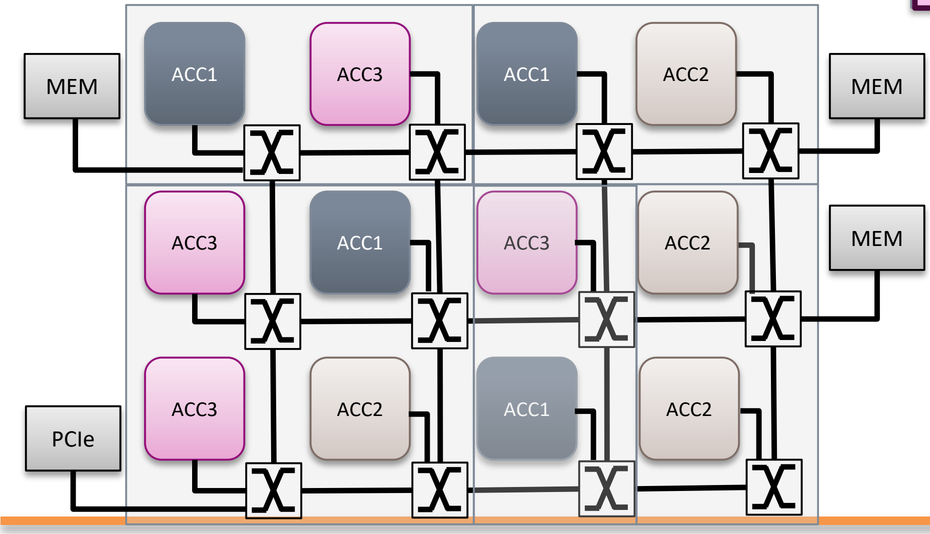
Data Network

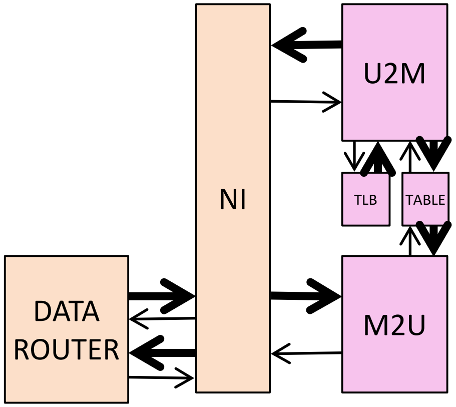


- DATA Routers connected among them in a 2D mesh layout.
- NI decouples network from tile components.
- Support for different Virtual networks (VN) with different number of Virtual channels (VC).
- Support for dynamic assignment of bandwidth per VN.
- Support for capacity computing



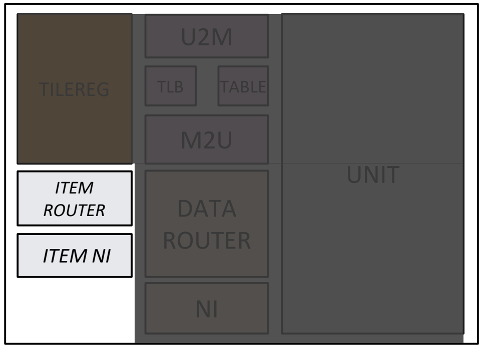
app1	app1	app1	app3
app2	app2	app3	app3

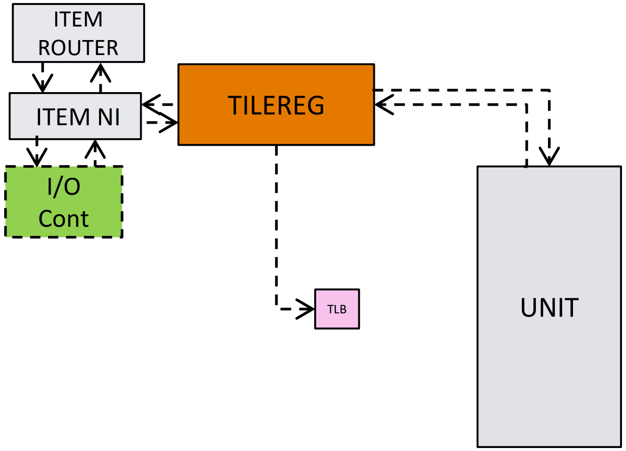




Control Network

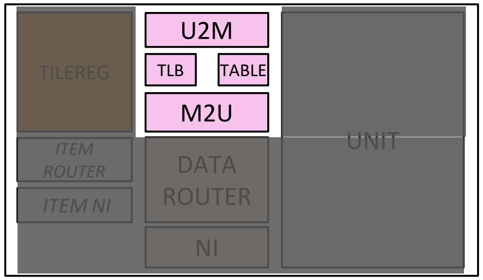
- Used for configuration and monitoring
  - MANGO Infrastructure
  - Accelerators or units
- Flexible and generic to let the accelerators be configured based on their complexities
  - Ad-hoc protocols

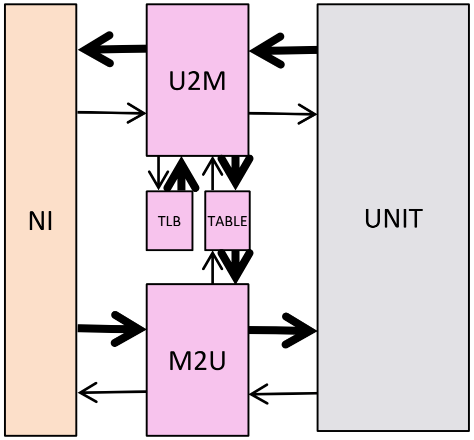




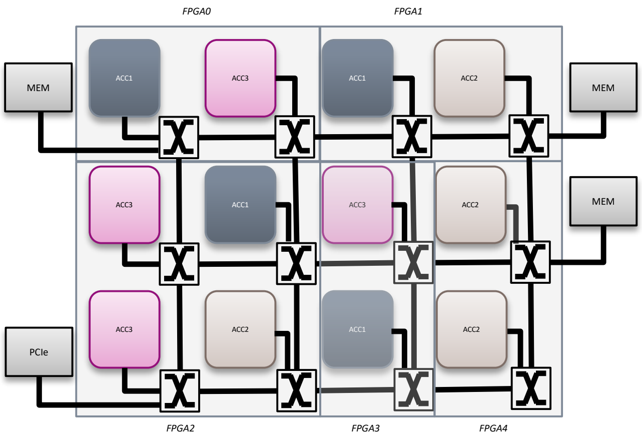
Accelerator interface

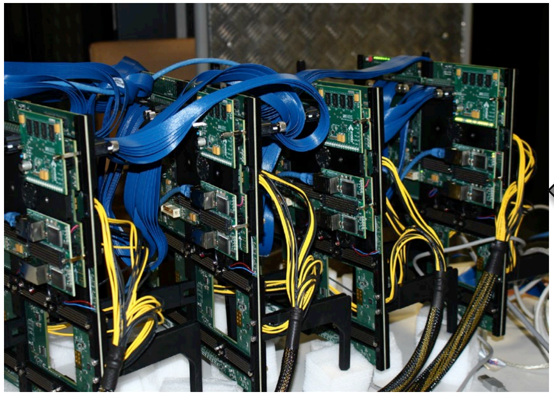
- Decouple the UNIT from the rest of the MANGO platform
- Allow the implementation of an unique interface for every UNIT
- Unify memory access
  - Byte, Half (16 bits), word (32 bits) & block (512 bits) memory access types
- Allow to map synchronization registers in the virtual memory address space

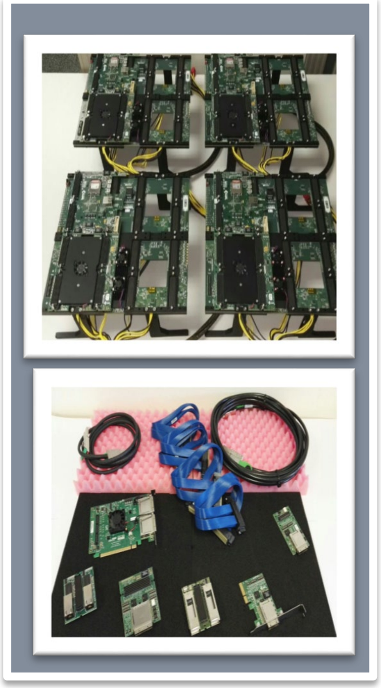


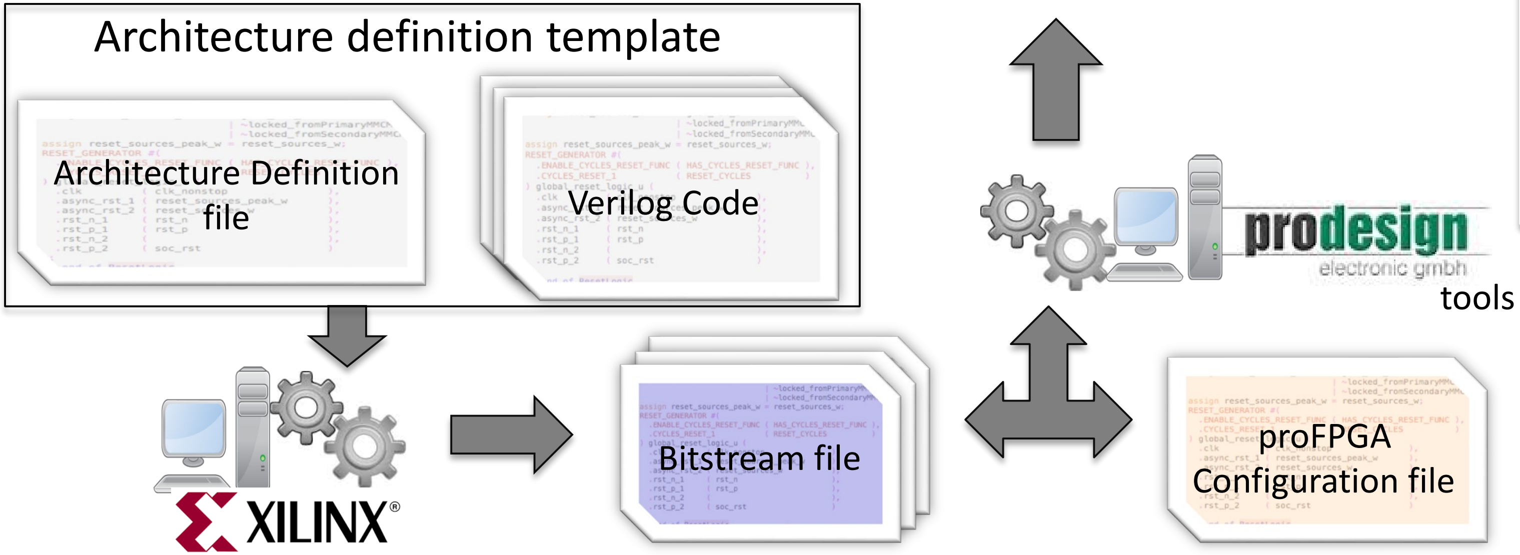


MANGO Design Flow









Architecture definition template

Architecture Definition file

Verilog Code

XILINX

Bitstream file

prodesign electronic gmbh tools

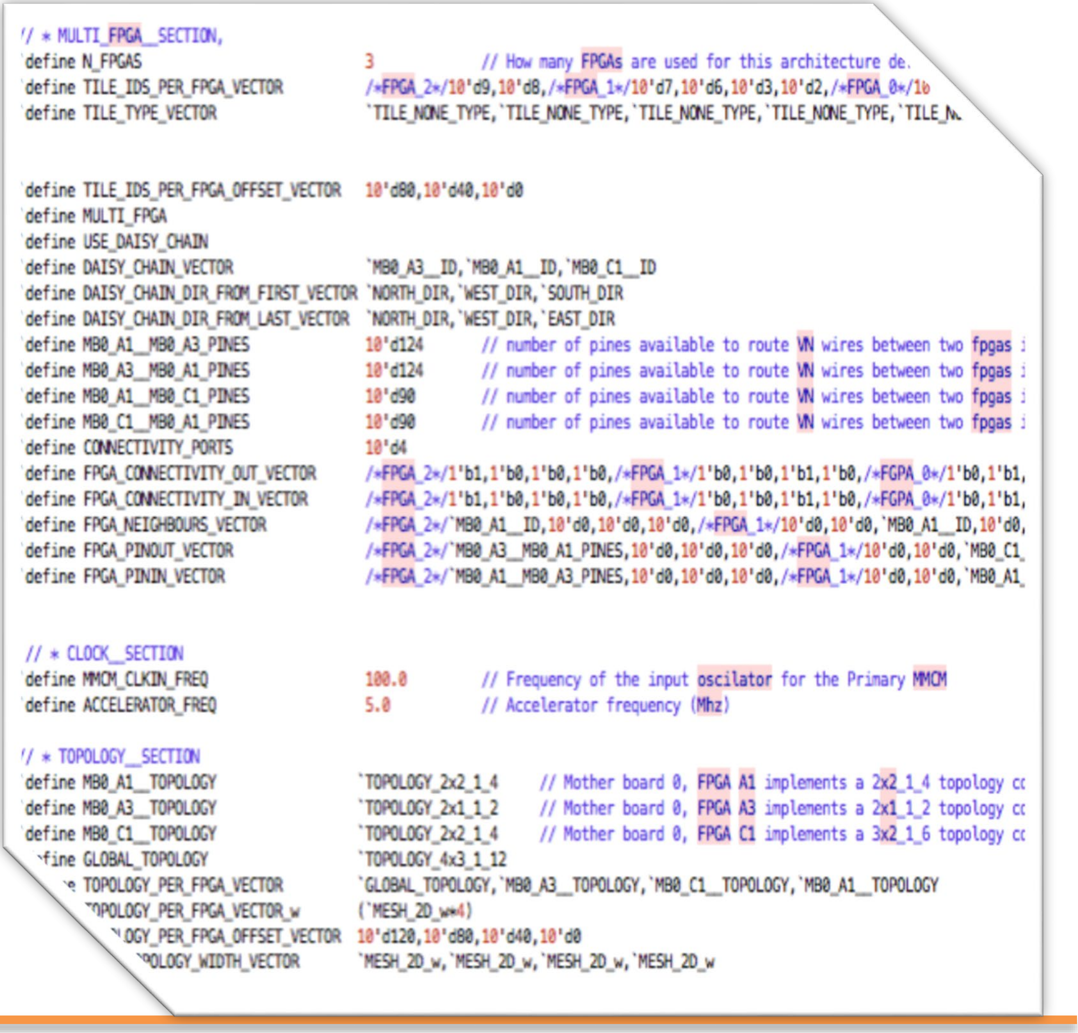
proFPGA Configuration file

Architecture Definition File

ARCHITECTURE ID

CONCEPT

- Sections
  - General
  - FPGA/multi-FPGA
  - Memory devices
  - I/O devices
  - Network

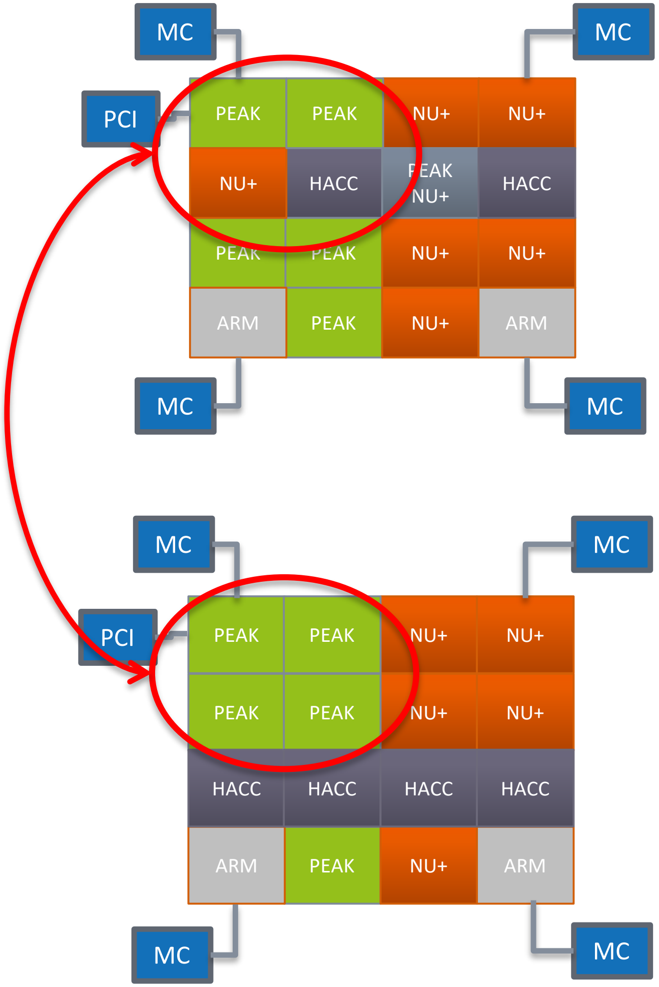


```
// * MULTI_FPGA__SECTION,
`define N_FPGAS                          3              // How many FPGAs are used for this architecture de.
`define TILE_IDS_PER_FPGA_VECTOR         /*FPGA_2*/10'd9,10'd8,/*FPGA_1*/10'd7,10'd6,10'd3,10'd2,/*FPGA_0*/10
`define TILE_TYPE_VECTOR                 `TILE_NONE_TYPE,`TILE_NONE_TYPE,`TILE_NONE_TYPE,`TILE_NONE_TYPE,`TILE_N


`define TILE_IDS_PER_FPGA_OFFSET_VECTOR  10'd80,10'd40,10'd0
`define MULTI_FPGA
`define USE_DAISY_CHAIN
`define DAISY_CHAIN_VECTOR               `MB0_A3__ID,`MB0_A1__ID,`MB0_C1__ID
`define DAISY_CHAIN_DIR_FROM_FIRST_VECTOR `NORTH_DIR,`WEST_DIR,`SOUTH_DIR
`define DAISY_CHAIN_DIR_FROM_LAST_VECTOR `NORTH_DIR,`WEST_DIR,`EAST_DIR
`define MB0_A1__MB0_A3_PINES             10'd124        // number of pines available to route VN wires between two fpgas i
`define MB0_A3__MB0_A1_PINES             10'd124        // number of pines available to route VN wires between two fpgas i
`define MB0_A1__MB0_C1_PINES             10'd90         // number of pines available to route VN wires between two fpgas i
`define MB0_C1__MB0_A1_PINES             10'd90         // number of pines available to route VN wires between two fpgas i
`define CONNECTIVITY_PORTS               10'd4
`define FPGA_CONNECTIVITY_OUT_VECTOR     /*FPGA_2*/1'b1,1'b0,1'b0,1'b0,/*FPGA_1*/1'b0,1'b0,1'b1,1'b0,/*FGPA_0*/1'b0,1'b1,
`define FPGA_CONNECTIVITY_IN_VECTOR      /*FPGA_2*/1'b1,1'b0,1'b0,1'b0,/*FPGA_1*/1'b0,1'b0,1'b1,1'b0,/*FGPA_0*/1'b0,1'b1,
`define FPGA_NEIGHBOURS_VECTOR           /*FPGA_2*/`MB0_A1__ID,10'd0,10'd0,10'd0,/*FPGA_1*/10'd0,10'd0,`MB0_A1__ID,10'd0,
`define FPGA_PINOUT_VECTOR               /*FPGA_2*/`MB0_A3__MB0_A1_PINES,10'd0,10'd0,10'd0,/*FPGA_1*/10'd0,10'd0,`MB0_C1_
`define FPGA_PININ_VECTOR                /*FPGA_2*/`MB0_A1__MB0_A3_PINES,10'd0,10'd0,10'd0,/*FPGA_1*/10'd0,10'd0,`MB0_A1_


// * CLOCK__SECTION
`define MMCM_CLKIN_FREQ                  100.0          // Frequency of the input oscilator for the Primary MMCM
`define ACCELERATOR_FREQ                 5.0            // Accelerator frequency (Mhz)

// * TOPOLOGY__SECTION
`define MB0_A1__TOPOLOGY                 `TOPOLOGY_2x2_1_4     // Mother board 0, FPGA A1 implements a 2x2_1_4 topology cc
`define MB0_A3__TOPOLOGY                 `TOPOLOGY_2x1_1_2     // Mother board 0, FPGA A3 implements a 2x1_1_2 topology cc
`define MB0_C1__TOPOLOGY                 `TOPOLOGY_2x2_1_4     // Mother board 0, FPGA C1 implements a 3x2_1_6 topology cc
 fine GLOBAL_TOPOLOGY                    `TOPOLOGY_4x3_1_12
   e TOPOLOGY_PER_FPGA_VECTOR            `GLOBAL_TOPOLOGY,`MB0_A3__TOPOLOGY,`MB0_C1__TOPOLOGY,`MB0_A1__TOPOLOGY
     OPOLOGY_PER_FPGA_VECTOR_w           (`MESH_2D_w*4)
       LOGY_PER_FPGA_OFFSET_VECTOR       10'd120,10'd80,10'd40,10'd0
        POLOGY_WIDTH_VECTOR              `MESH_2D_w,`MESH_2D_w,`MESH_2D_w,`MESH_2D_w
```

Architecture Definition Template

Key aspects

- Ease to configure the Unit type for every tile



MC MC PCI

PEAK	PEAK	NU+	NU+
NU+	HACC	PEAK NU+	HACC
PEAK	PEAK	NU+	NU+
ARM	PEAK	NU+	ARM

MC MC

MC MC PCI

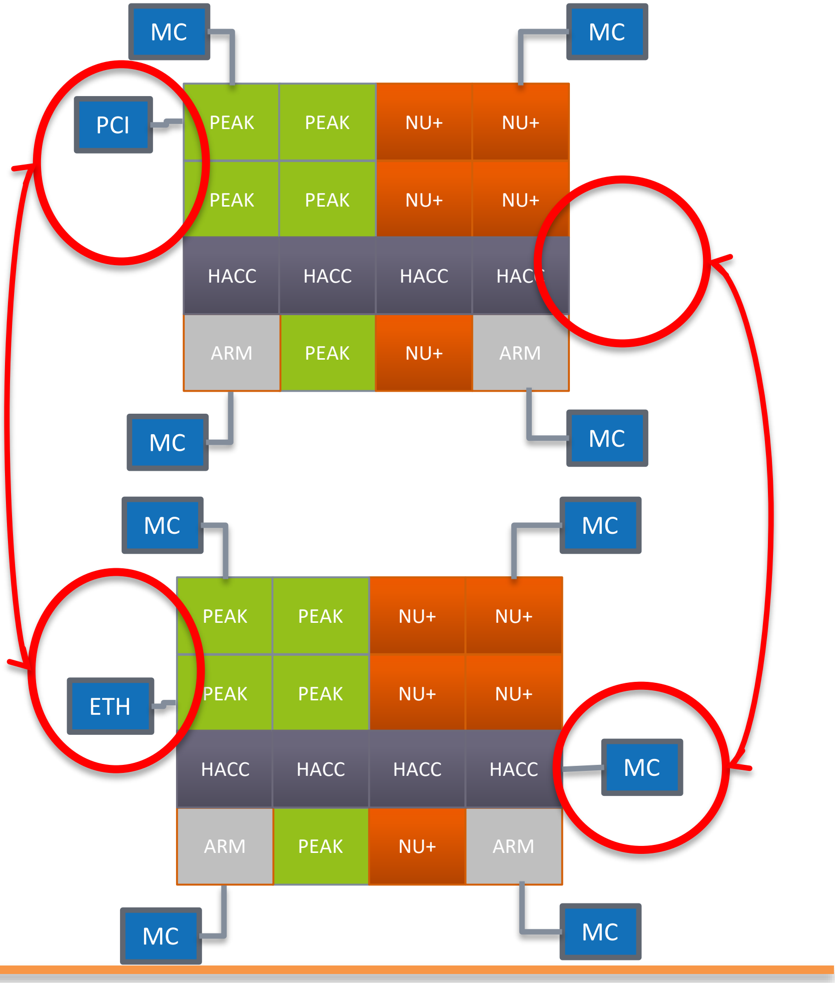
PEAK	PEAK	NU+	NU+
PEAK	PEAK	NU+	NU+
HACC	HACC	HACC	HACC
ARM	PEAK	NU+	ARM

MC MC

Architecture Definition Template

Key aspects

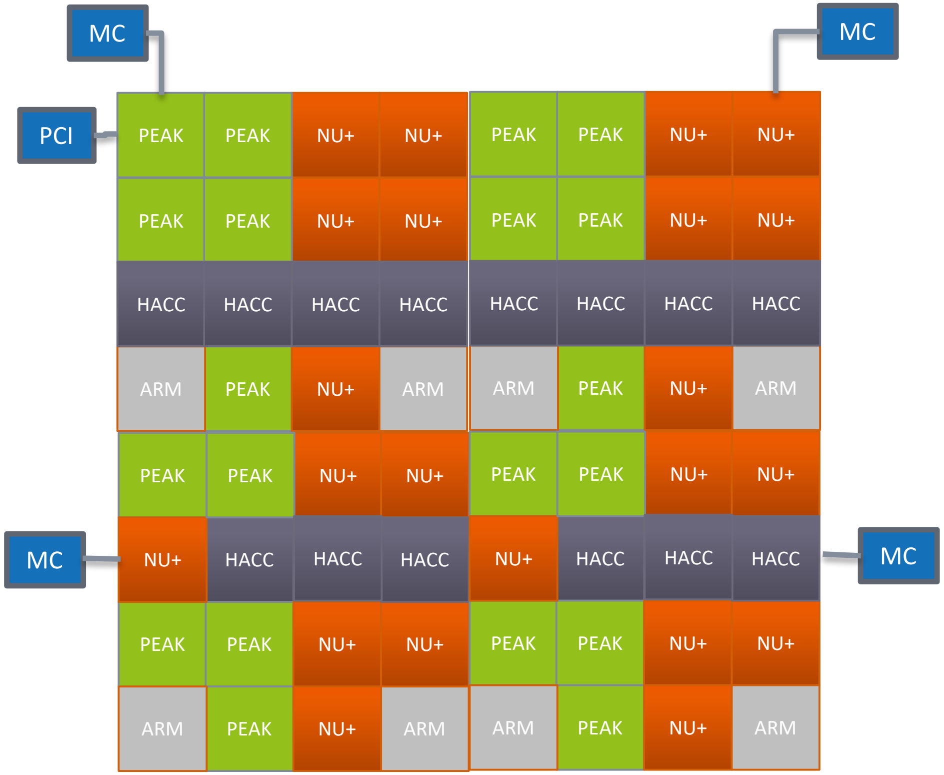
- Ease to configure the Unit type for every tile
- Ease to attach Memory Controllers (MC) and I/O devices (PCIe, Ethernet) to any tile



Architecture Definition Template

Key aspects

- Ease to configure the Unit type for every tile
- Ease to attach Memory Controllers (MC) and I/O devices (PCIe, Ethernet) to any tile
- Ease to configure the number of tiles

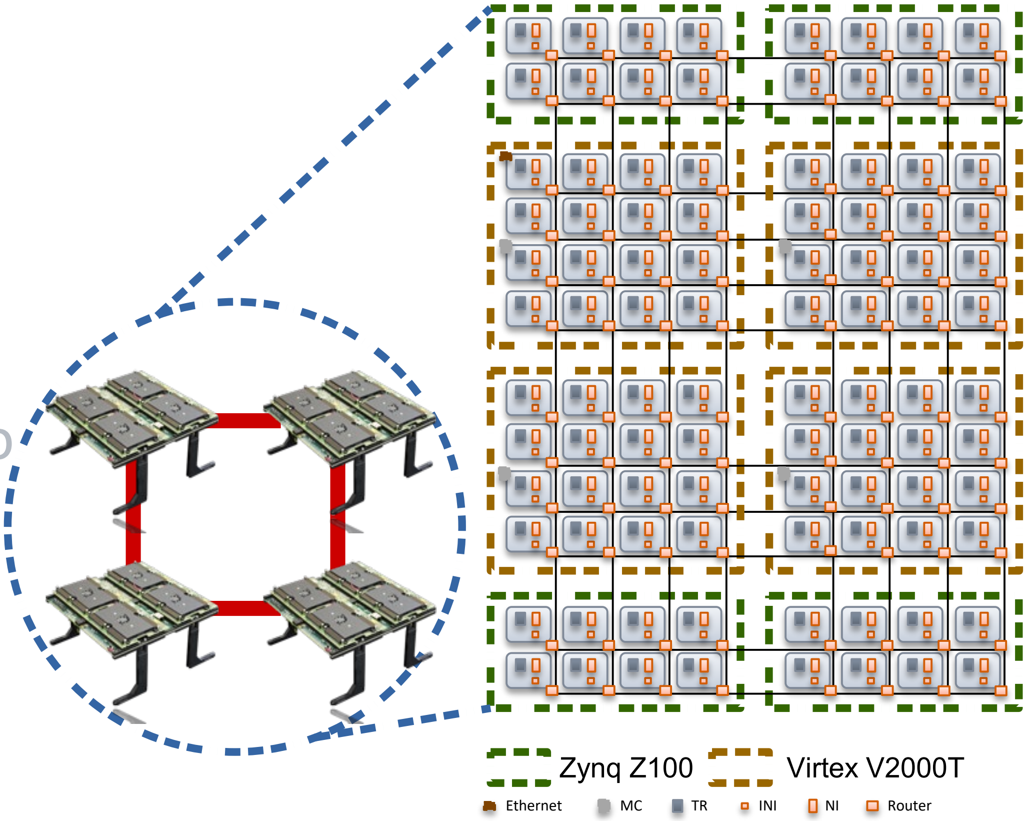


PEAK	PEAK	NU+	NU+	PEAK	PEAK	NU+	NU+
PEAK	PEAK	NU+	NU+	PEAK	PEAK	NU+	NU+
HACC	HACC	HACC	HACC	HACC	HACC	HACC	HACC
ARM	PEAK	NU+	ARM	ARM	PEAK	NU+	ARM
PEAK	PEAK	NU+	NU+	PEAK	PEAK	NU+	NU+
NU+	HACC	HACC	HACC	NU+	HACC	HACC	HACC
PEAK	PEAK	NU+	NU+	PEAK	PEAK	NU+	NU+
ARM	PEAK	NU+	ARM	ARM	PEAK	NU+	ARM

Architecture Definition Template

Key aspects

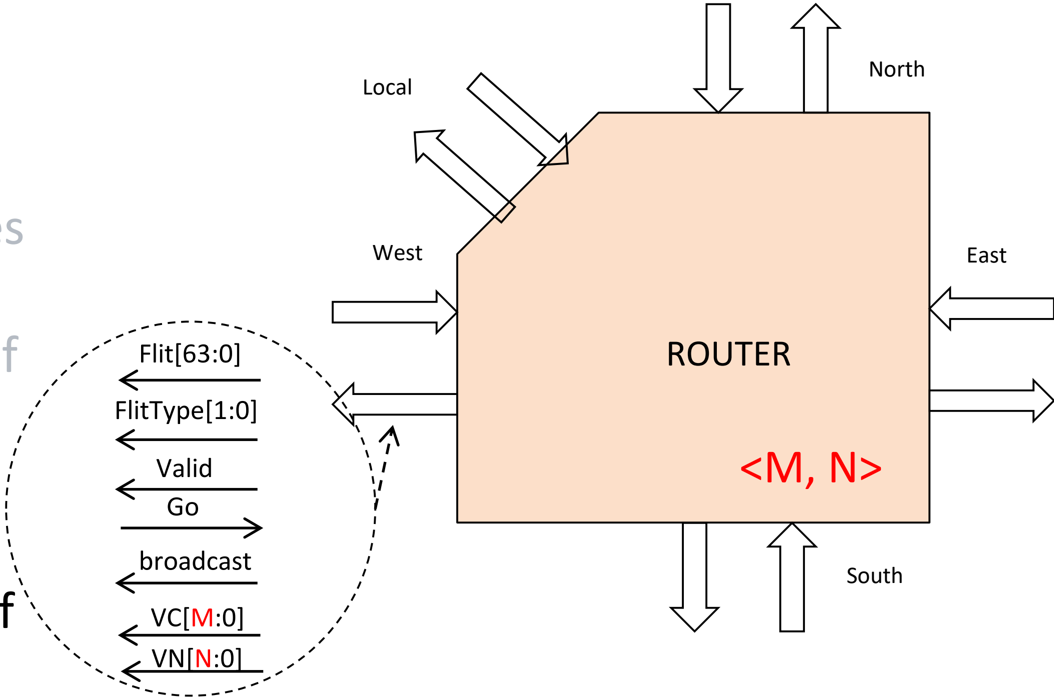
- Ease to configure the Unit type for every tile
- Ease to attach Memory Controllers (MC) and I/O devices (PCIe, Ethernet) to any tile
- Ease to configure the number of tiles
- Ease to implement multi-FPGA designs



Architecture Definition Template

Key aspects

- Ease to configure the Unit type for every tile
- Ease to attach Memory Controllers (MC) and I/O devices (PCIe, Ethernet) to any tile
- Ease to configure the number of tiles
- Ease to implement multi-FPGA designs
- Easy to configure the number of virtual networks and virtual channels to achieve QoS guarantee

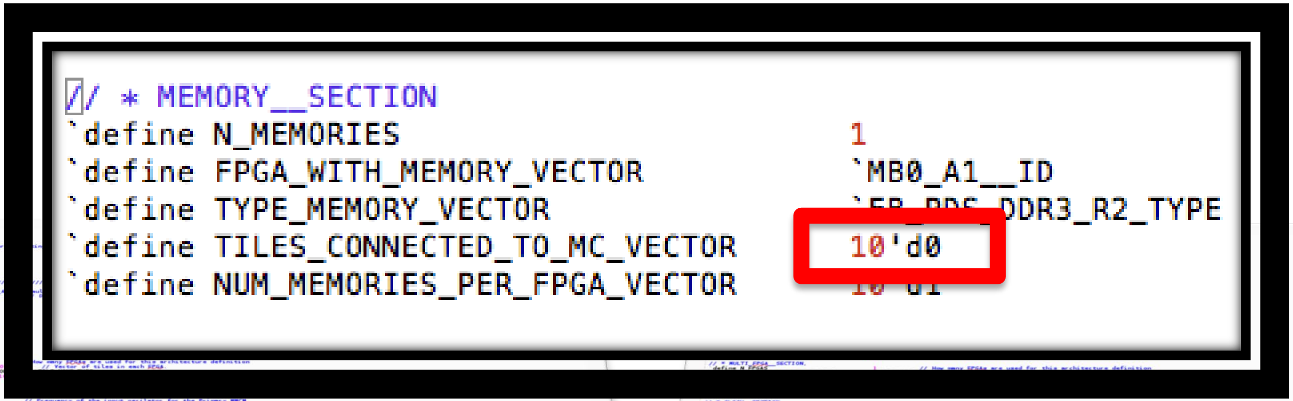


Architecture Definition Template

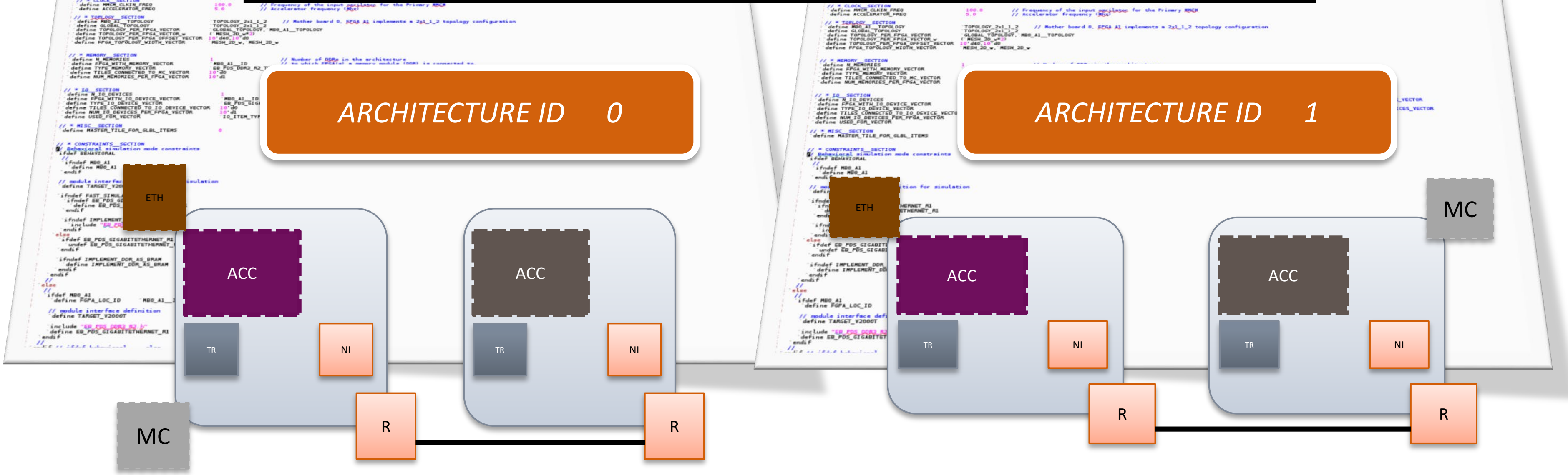
Key aspects

- Ease to configure the Unit type for every tile
- Ease to attach Memory Controllers (MC) and I/O devices (PCIe, Ethernet) to any tile
- Ease to configure the number of tiles
- Ease to implement multi-FPGA designs
- Easy to configure the number of virtual networks and virtual channels to achieve QoS guarantee

Architecture Definition File: Example

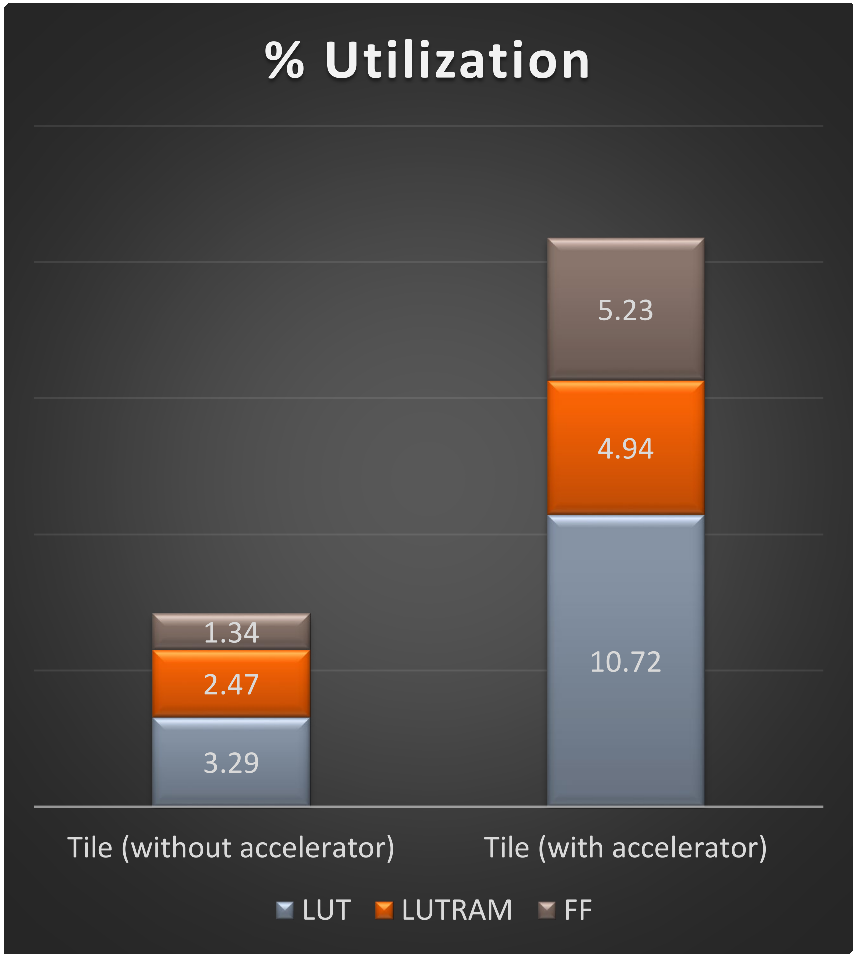


```
// * MEMORY__SECTION
`define N_MEMORIES                        1
`define FPGA_WITH_MEMORY_VECTOR           `MB0_A1__ID
`define TYPE_MEMORY_VECTOR                `EB_PDS_DDR3_R2_TYPE
`define TILES_CONNECTED_TO_MC_VECTOR      10'd0
`define NUM_MEMORIES_PER_FPGA_VECTOR      10'd1
```



FPGA Resource utilization

- Xilinx Virtex V2000T speedgrade -1
- Tile with Accelerator:
  - MIPS-based cache coherent 2-core accelerator
    - 32K L1I
    - 512K L1D
    - 1MB L2D



Conclusions

- The MANGO approach for supporting the implementation of multiple accelerators on a multi-FPGA platform
  - High customization of the cluster
    - Flexibility for architecture exploration
  - Flexibility in configuring an architecture
    - Rapid architecture exploration
  - Effectivity in the system communications
    - QoS guarantee
  - Effectivity in monitoring the system
- Percentage of resources needed per tile quite slow

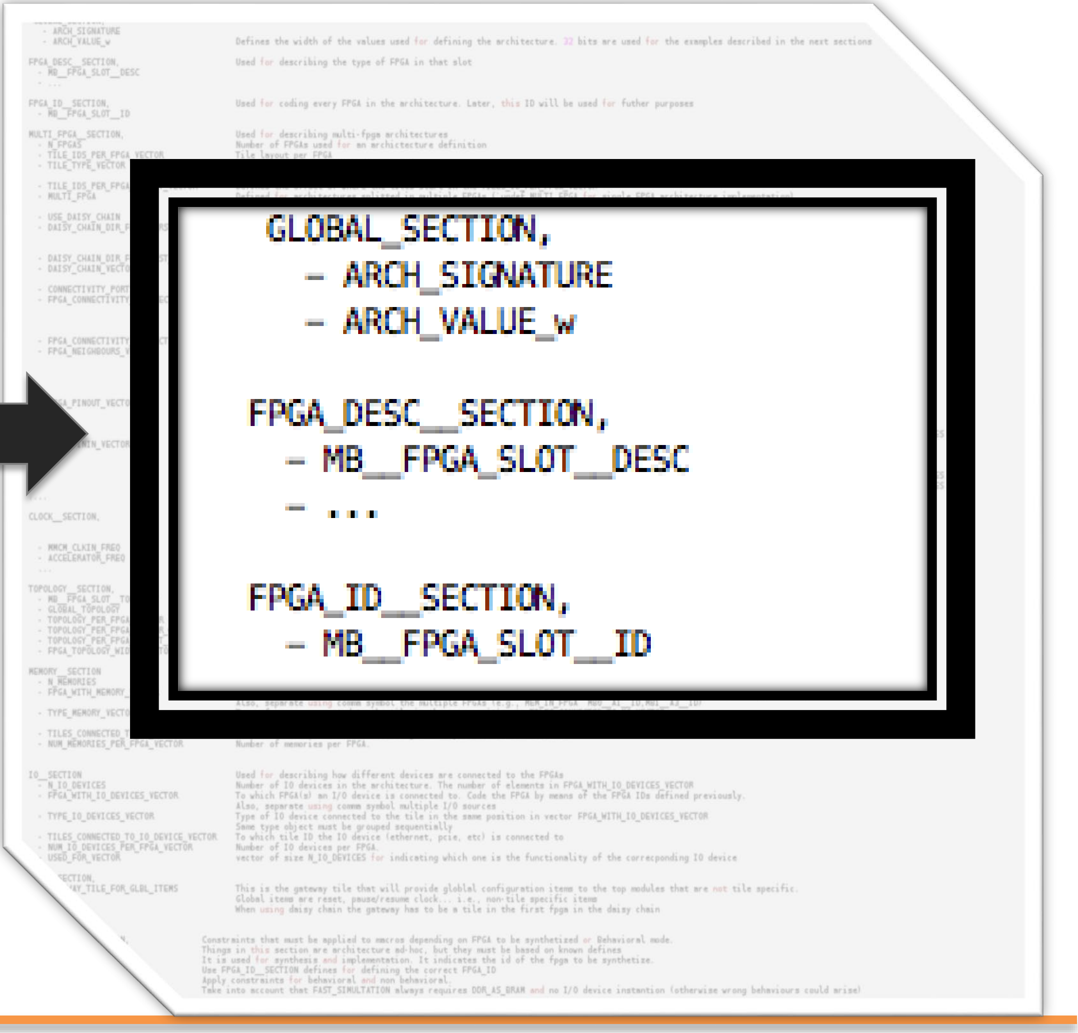
Thank you for your attention

- Contact us at
  - www.mango-project.eu
- Other directly related EU project
  - www.recipe-project.eu

- The MANGO project has received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreement No 671668.
- The content of this presentation reflects only the authors' views and the European Commission is not responsible for any use that may be made of the information it contains.

Architecture Definition File

- Sections
  - **General**
  - FPGA/multi-FPGA
  - Memory devices
  - I/O devices
  - Network

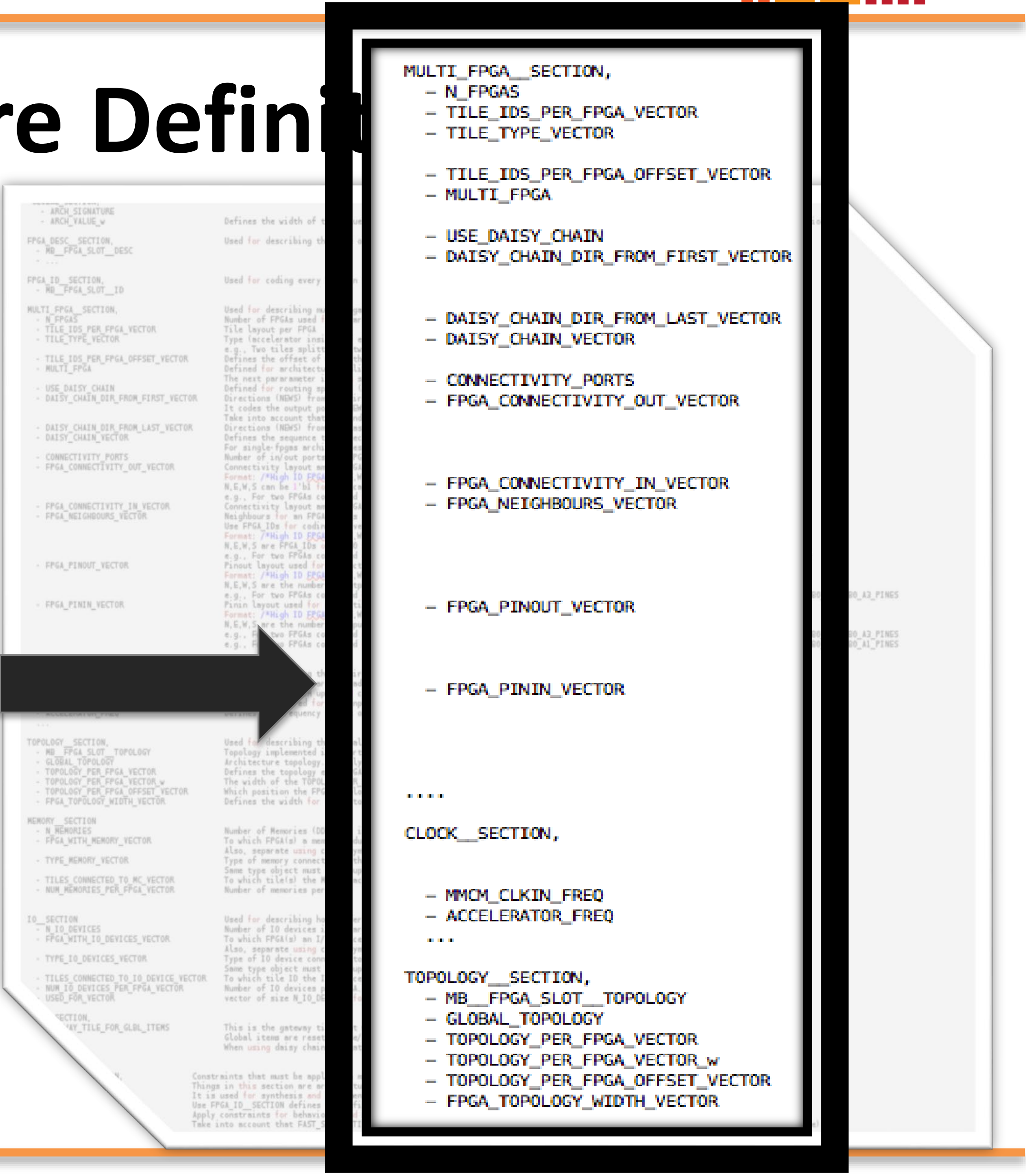


```
GLOBAL_SECTION,
  - ARCH_SIGNATURE
  - ARCH_VALUE_w

FPGA_DESC__SECTION,
  - MB__FPGA_SLOT__DESC
  - ...

FPGA_ID__SECTION,
  - MB__FPGA_SLOT__ID
```

```
  - ARCH_SIGNATURE
  - ARCH_VALUE_w                          Defines the width of the values used for defining the architecture. 32 bits are used for the examples described in the next sections

FPGA_DESC__SECTION,                       Used for describing the type of FPGA in that slot
  - MB__FPGA_SLOT__DESC
  - ...

FPGA_ID__SECTION,                         Used for coding every FPGA in the architecture. Later, this ID will be used for futher purposes
  - MB__FPGA_SLOT__ID

MULTI_FPGA__SECTION,                      Used for describing multi-fpga architectures
  - N_FPGAS                               Number of FPGAs used for an architecture definition
  - TILE_IDS_PER_FPGA_VECTOR              Tile layout per FPGA
  - TILE_TYPE_VECTOR

  - TILE_IDS_PER_FPGA
  - MULTI_FPGA

  - USE_DAISY_CHAIN
  - DAISY_CHAIN_DIR_F

  - DAISY_CHAIN_DIR_F
  - DAISY_CHAIN_VECTO

  - CONNECTIVITY_PORT
  - FPGA_CONNECTIVITY

  - FPGA_CONNECTIVITY
  - FPGA_NEIGHBOURS_V

CLOCK__SECTION,

  - MMCM_CLKIN_FREQ
  - ACCELERATOR_FREQ
  ...

TOPOLOGY__SECTION,
  - MB__FPGA_SLOT__TO
  - GLOBAL_TOPOLOGY
  - TOPOLOGY_PER_FPGA
  - TOPOLOGY_PER_FPGA
  - TOPOLOGY_PER_FPGA
  - FPGA_TOPOLOGY_WID

MEMORY__SECTION
  - N_MEMORIES
  - FPGA_WITH_MEMORY_

  - TYPE_MEMORY_VECTO

  - TILES_CONNECTED_T
  - NUM_MEMORIES_PER_FPGA_VECTOR          Number of memories per FPGA.

IO__SECTION                               Used for describing how different devices are connected to the FPGAs
  - N_IO_DEVICES                          Number of IO devices in the architecture. The number of elements in FPGA_WITH_IO_DEVICES_VECTOR
  - FPGA_WITH_IO_DEVICES_VECTOR           To which FPGA(s) an I/O device is connected to. Code the FPGA by means of the FPGA IDs defined previously.
                                          Also, separate using comma symbol multiple I/O sources
  - TYPE_IO_DEVICES_VECTOR                Type of IO device connected to the tile in the same position in vector FPGA_WITH_IO_DEVICES_VECTOR
                                          Same type object must be grouped sequentially
  - TILES_CONNECTED_TO_IO_DEVICE_VECTOR   To which tile ID the IO device (ethernet, pcie, etc) is connected to
  - NUM_IO_DEVICES_PER_FPGA_VECTOR        Number of IO devices per FPGA.
  - USED_FOR_VECTOR                       vector of size N_IO_DEVICES for indicating which one is the functionality of the correcponding IO device

  SECTION,
  AY_TILE_FOR_GLBL_ITEMS                  This is the gateway tile that will provide globlal configuration items to the top modules that are not tile specific.
                                          Global items are reset, pause/resume clock... i.e., non-tile specific items
                                          When using daisy chain the gateway has to be a tile in the first fpga in the daisy chain

     N,                                Constraints that must be applied to macros depending on FPGA to be synthetized or Behavioral mode.
                                       Things in this section are architecture ad-hoc, but they must be based on known defines
                                       It is used for synthesis and implementation. It indicates the id of the fpga to be synthetize.
                                       Use FPGA_ID__SECTION defines for defining the correct FPGA_ID
                                       Apply constraints for behavioral and non behavioral.
                                       Take into account that FAST_SIMULTATION always requires DDR_AS_BRAM and no I/O device instantion (otherwise wrong behaviours could arise)
```

Architecture Defini

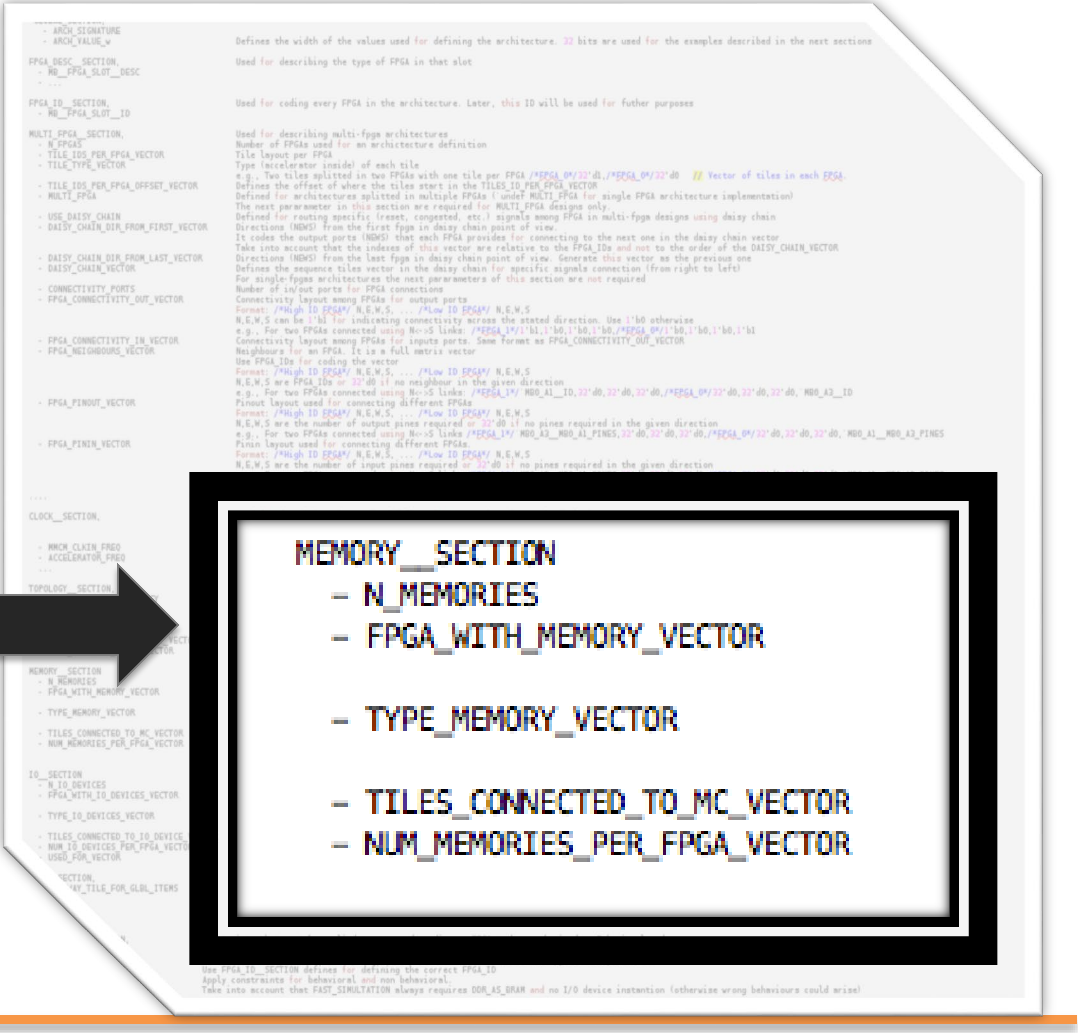
- ○Sections
  - ○General
  - ○**FPGA/multi-FPGA**
  - ○Memory devices
  - ○I/O devices
  - ○Network



```
MULTI_FPGA__SECTION,
  – N_FPGAS
  – TILE_IDS_PER_FPGA_VECTOR
  – TILE_TYPE_VECTOR

  – TILE_IDS_PER_FPGA_OFFSET_VECTOR
  – MULTI_FPGA

  – USE_DAISY_CHAIN
  – DAISY_CHAIN_DIR_FROM_FIRST_VECTOR


  – DAISY_CHAIN_DIR_FROM_LAST_VECTOR
  – DAISY_CHAIN_VECTOR

  – CONNECTIVITY_PORTS
  – FPGA_CONNECTIVITY_OUT_VECTOR



  – FPGA_CONNECTIVITY_IN_VECTOR
  – FPGA_NEIGHBOURS_VECTOR




  – FPGA_PINOUT_VECTOR



  – FPGA_PININ_VECTOR




....

CLOCK__SECTION,


  – MMCM_CLKIN_FREQ
  – ACCELERATOR_FREQ
  ...

TOPOLOGY__SECTION,
  – MB__FPGA_SLOT__TOPOLOGY
  – GLOBAL_TOPOLOGY
  – TOPOLOGY_PER_FPGA_VECTOR
  – TOPOLOGY_PER_FPGA_VECTOR_w
  – TOPOLOGY_PER_FPGA_OFFSET_VECTOR
  – FPGA_TOPOLOGY_WIDTH_VECTOR
```

Architecture Definition File

- Sections
  - General
  - FPGA/multi-FPGA
  - **Memory devices**
  - I/O devices
  - Network

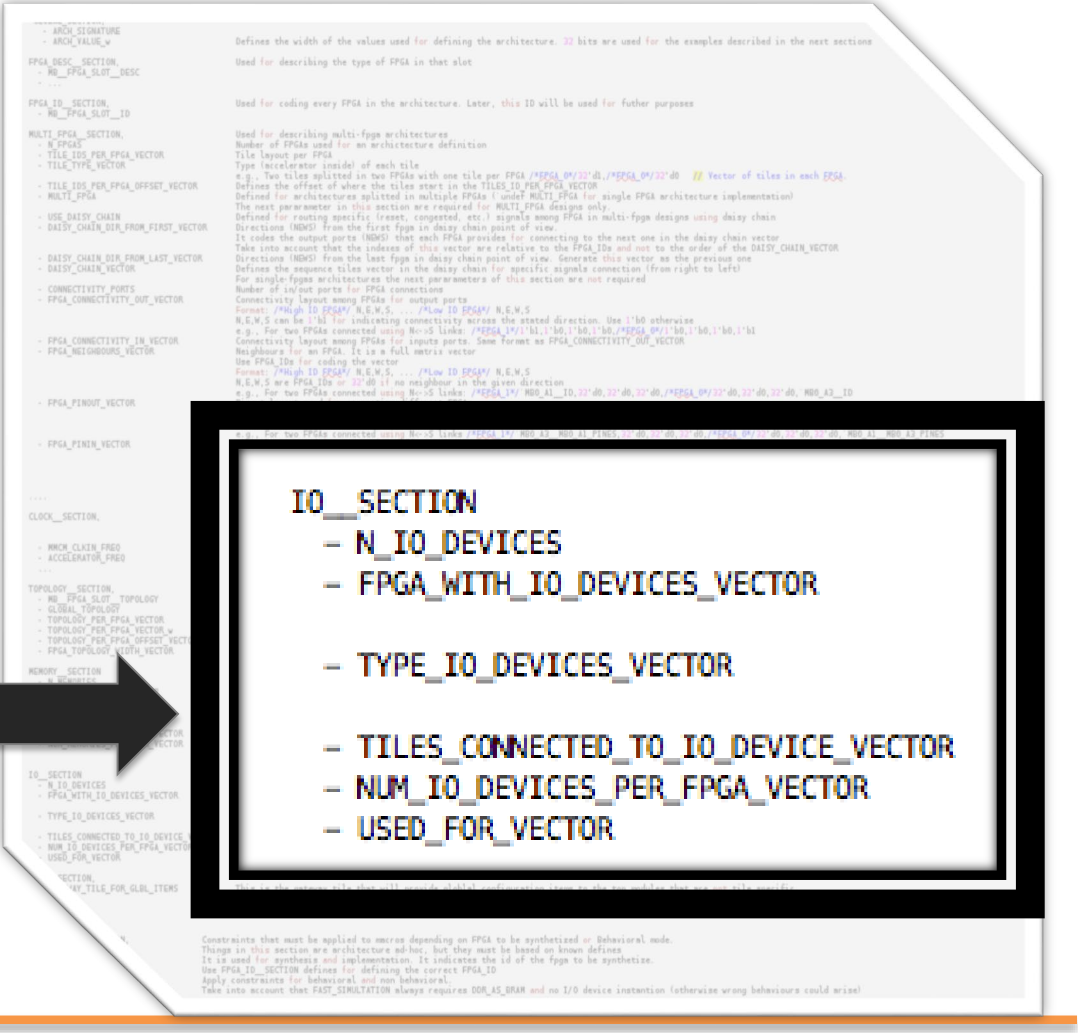


```
  - ARCH_SIGNATURE
  - ARCH_VALUE_w                        Defines the width of the values used for defining the architecture. 32 bits are used for the examples described in the next sections

FPGA_DESC__SECTION,                     Used for describing the type of FPGA in that slot
  - MB__FPGA_SLOT__DESC
  - ...

FPGA_ID__SECTION,                       Used for coding every FPGA in the architecture. Later, this ID will be used for futher purposes
  - MB__FPGA_SLOT__ID

MULTI_FPGA__SECTION,                    Used for describing multi-fpga architectures
  - N_FPGAS                             Number of FPGAs used for an archictecture definition
  - TILE_IDS_PER_FPGA_VECTOR            Tile layout per FPGA
  - TILE_TYPE_VECTOR                    Type (accelerator inside) of each tile
                                        e.g., Two tiles splitted in two FPGAs with one tile per FPGA /*FPGA_0*/32'd1,/*FPGA_0*/32'd0   // Vector of tiles in each FPGA.
  - TILE_IDS_PER_FPGA_OFFSET_VECTOR     Defines the offset of where the tiles start in the TILES_ID_PER_FPGA_VECTOR
  - MULTI_FPGA                          Defined for architectures splitted in multiple FPGAs (`undef MULTI_FPGA for single FPGA architecture implementation)
                                        The next parrameter in this section are required for MULTI_FPGA designs only.
  - USE_DAISY_CHAIN                     Defined for routing specific (reset, congested, etc.) signals among FPGA in multi-fpga designs using daisy chain
  - DAISY_CHAIN_DIR_FROM_FIRST_VECTOR   Directions (NEWS) from the first fpga in daisy chain point of view.
                                        It codes the output ports (NEWS) that each FPGA provides for connecting to the next one in the daisy chain vector
                                        Take into account that the indexes of this vector are relative to the FPGA_IDs and not to the order of the DAISY_CHAIN_VECTOR
  - DAISY_CHAIN_DIR_FROM_LAST_VECTOR    Directions (NEWS) from the last fpga in daisy chain point of view. Generate this vector as the previous one
  - DAISY_CHAIN_VECTOR                  Defines the sequence tiles vector in the daisy chain for specific signals connection (from right to left)
                                        For single-fpgas architectures the next parrameters of this section are not required
  - CONNECTIVITY_PORTS                  Number of in/out ports for FPGA connections
  - FPGA_CONNECTIVITY_OUT_VECTOR        Connectivity layout among FPGAs for output ports
                                        Format: /*High ID FPGA*/ N,E,W,S, ... /*Low ID FPGA*/ N,E,W,S
                                        N,E,W,S can be 1'b1 for indicating connectivity across the stated direction. Use 1'b0 otherwise
                                        e.g., For two FPGAs connected using N<->S links: /*FPGA_1*/1'b1,1'b0,1'b0,1'b0,/*FPGA_0*/1'b0,1'b0,1'b0,1'b1
  - FPGA_CONNECTIVITY_IN_VECTOR         Connectivity layout among FPGAs for inputs ports. Same format as FPGA_CONNECTIVITY_OUT_VECTOR
  - FPGA_NEIGHBOURS_VECTOR              Neighbours for an FPGA. It is a full matrix vector
                                        Use FPGA_IDs for coding the vector
                                        Format: /*High ID FPGA*/ N,E,W,S, ... /*Low ID FPGA*/ N,E,W,S
                                        N,E,W,S are FPGA_IDs or 32'd0 if no neighbour in the given direction
                                        e.g., For two FPGAs connected using N<->S links: /*FPGA_1*/ MB0_A1__ID,32'd0,32'd0,32'd0,/*FPGA_0*/32'd0,32'd0,32'd0, MB0_A3__ID
  - FPGA_PINOUT_VECTOR                  Pinout layout used for connecting different FPGAs
                                        Format: /*High ID FPGA*/ N,E,W,S, ... /*Low ID FPGA*/ N,E,W,S
                                        N,E,W,S are the number of output pines required or 32'd0 if no pines required in the given direction
                                        e.g., For two FPGAs connected using N<->S links /*FPGA_1*/ MB0_A3__MB0_A1_PINES,32'd0,32'd0,32'd0,/*FPGA_0*/32'd0,32'd0,32'd0, MB0_A1__MB0_A3_PINES
  - FPGA_PININ_VECTOR                   Pinin layout used for connecting different FPGAs.
                                        Format: /*High ID FPGA*/ N,E,W,S, ... /*Low ID FPGA*/ N,E,W,S
                                        N,E,W,S are the number of input pines required or 32'd0 if no pines required in the given direction

....

CLOCK__SECTION,

  - MMCM_CLKIN_FREQ
  - ACCELERATOR_FREQ
  - ...

TOPOLOGY__SECTION,

MEMORY__SECTION
  - N_MEMORIES
  - FPGA_WITH_MEMORY_VECTOR

  - TYPE_MEMORY_VECTOR

  - TILES_CONNECTED_TO_MC_VECTOR
  - NUM_MEMORIES_PER_FPGA_VECTOR

IO__SECTION
  - N_IO_DEVICES
  - FPGA_WITH_IO_DEVICES_VECTOR

  - TYPE_IO_DEVICES_VECTOR

  - TILES_CONNECTED_TO_IO_DEVICE_
  - NUM_IO_DEVICES_PER_FPGA_VECTO
  - USED_FOR_VECTOR

   ECTION,
    AY_TILE_FOR_GLBL_ITEMS

                                        Use FPGA_ID__SECTION defines for defining the correct FPGA_ID
                                        Apply constraints for behavioral and non behavioral.
                                        Take into account that FAST_SIMULATION always requires DDR_AS_BRAM and no I/O device instantion (otherwise wrong behaviours could arise)
```

```
MEMORY__SECTION
  - N_MEMORIES
  - FPGA_WITH_MEMORY_VECTOR

  - TYPE_MEMORY_VECTOR

  - TILES_CONNECTED_TO_MC_VECTOR
  - NUM_MEMORIES_PER_FPGA_VECTOR
```

Architecture Definition File

- Sections
  - General
  - FPGA/multi-FPGA
  - Memory devices
  - **I/O devices**
  - Network

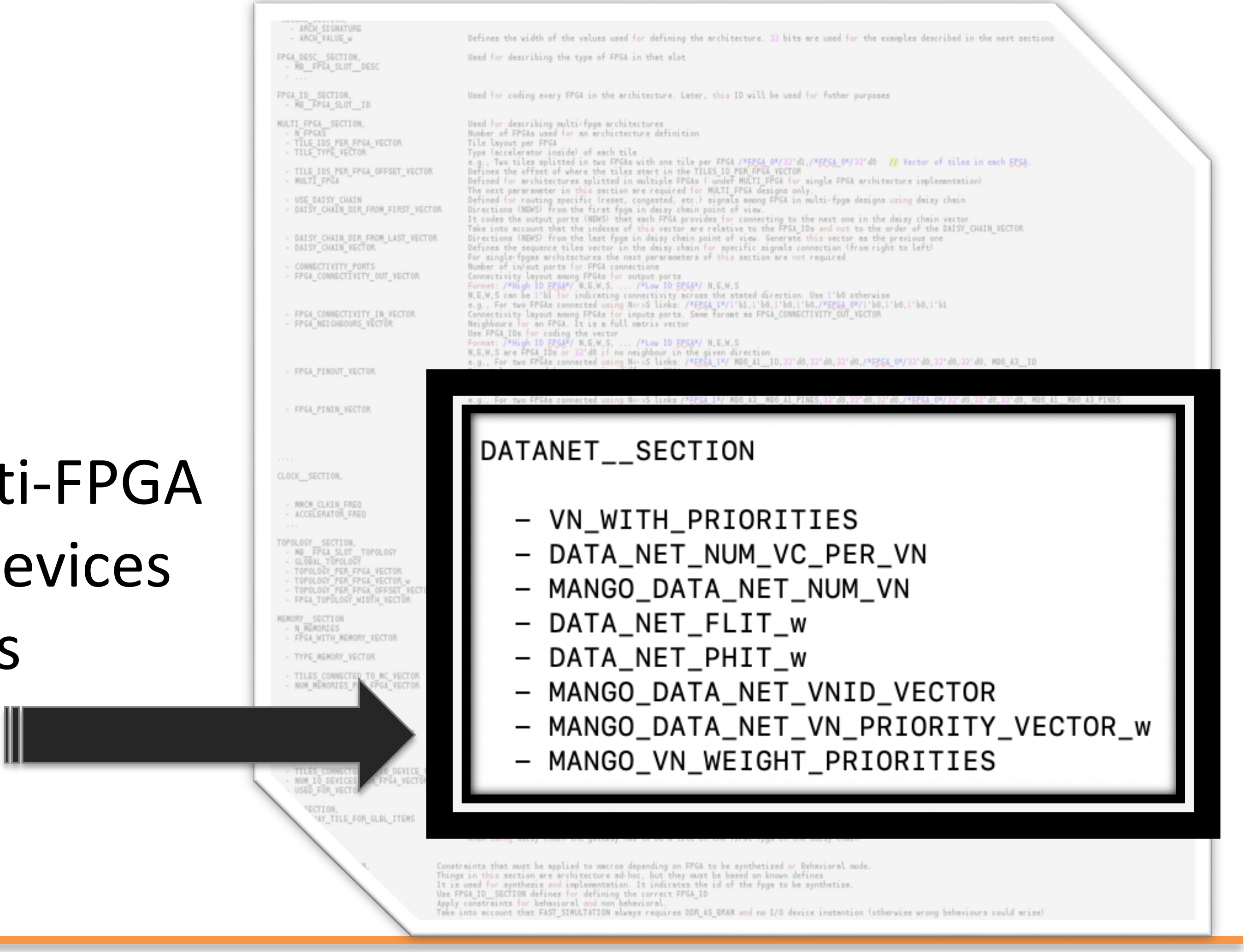


```
  - ARCH_SIGNATURE
  - ARCH_VALUE_w                      Defines the width of the values used for defining the architecture. 32 bits are used for the examples described in the next sections

FPGA_DESC__SECTION,                   Used for describing the type of FPGA in that slot
  - MB__FPGA_SLOT__DESC
  - ...

FPGA_ID__SECTION,                     Used for coding every FPGA in the architecture. Later, this ID will be used for futher purposes
  - MB__FPGA_SLOT__ID

MULTI_FPGA__SECTION,                  Used for describing multi-fpga architectures
  - N_FPGAS                           Number of FPGAs used for an architecture definition
  - TILE_IDS_PER_FPGA_VECTOR          Tile layout per FPGA
  - TILE_TYPE_VECTOR                  Type (accelerator inside) of each tile
                                      e.g., Two tiles splitted in two FPGAs with one tile per FPGA /*FPGA_0*/32'd1,/*FPGA_0*/32'd0   // Vector of tiles in each FPGA.
  - TILE_IDS_PER_FPGA_OFFSET_VECTOR   Defines the offset of where the tiles start in the TILES_ID_PER_FPGA_VECTOR
  - MULTI_FPGA                        Defined for architectures splitted in multiple FPGAs (`undef MULTI_FPGA for single FPGA architecture implementation)
                                      The next pararameter in this section are required for MULTI_FPGA designs only.
  - USE_DAISY_CHAIN                   Defined for routing specific (reset, congested, etc.) signals among FPGA in multi-fpga designs using daisy chain
  - DAISY_CHAIN_DIR_FROM_FIRST_VECTOR Directions (NEWS) from the first fpga in daisy chain point of view.
                                      It codes the output ports (NEWS) that each FPGA provides for connecting to the next one in the daisy chain vector
                                      Take into account that the indexes of this vector are relative to the FPGA_IDs and not to the order of the DAISY_CHAIN_VECTOR
  - DAISY_CHAIN_DIR_FROM_LAST_VECTOR  Directions (NEWS) from the last fpga in daisy chain point of view. Generate this vector as the previous one
  - DAISY_CHAIN_VECTOR                Defines the sequence tiles vector in the daisy chain for specific signals connection (from right to left)
                                      For single-fpgas architectures the next pararameters of this section are not required
  - CONNECTIVITY_PORTS                Number of in/out ports for FPGA connections
  - FPGA_CONNECTIVITY_OUT_VECTOR      Connectivity layout among FPGAs for output ports
                                      Format: /*High ID FPGA*/ N,E,W,S, ... /*Low ID FPGA*/ N,E,W,S
                                      N,E,W,S can be 1'b1 for indicating connectivity across the stated direction. Use 1'b0 otherwise
                                      e.g., For two FPGAs connected using N<->S links: /*FPGA_1*/1'b1,1'b0,1'b0,1'b0,/*FPGA_0*/1'b0,1'b0,1'b0,1'b1
  - FPGA_CONNECTIVITY_IN_VECTOR       Connectivity layout among FPGAs for inputs ports. Same format as FPGA_CONNECTIVITY_OUT_VECTOR
  - FPGA_NEIGHBOURS_VECTOR            Neighbours for an FPGA. It is a full matrix vector
                                      Use FPGA_IDs for coding the vector
                                      Format: /*High ID FPGA*/ N,E,W,S, ... /*Low ID FPGA*/ N,E,W,S
                                      N,E,W,S are FPGA_IDs or 32'd0 if no neighbour in the given direction
                                      e.g., For two FPGAs connected using N<->S links: /*FPGA_1*/ MB0_A1__ID,32'd0,32'd0,32'd0,/*FPGA_0*/32'd0,32'd0,32'd0, MB0_A3__ID

  - FPGA_PINOUT_VECTOR

                                      e.g., For two FPGAs connected using N<->S links /*FPGA_1*/ MB0_A3__MB0_A1_PINES,32'd0,32'd0,32'd0,/*FPGA_0*/32'd0,32'd0,32'd0, MB0_A1__MB0_A3_PINES
  - FPGA_PININ_VECTOR

....

CLOCK__SECTION,

  - MMCM_CLKIN_FREQ
  - ACCELERATOR_FREQ
  ...

TOPOLOGY__SECTION,
  - MB__FPGA_SLOT__TOPOLOGY
  - GLOBAL_TOPOLOGY
  - TOPOLOGY_PER_FPGA_VECTOR
  - TOPOLOGY_PER_FPGA_VECTOR_w
  - TOPOLOGY_PER_FPGA_OFFSET_VECTO
  - FPGA_TOPOLOGY_WIDTH_VECTOR

MEMORY__SECTION
  - N_MEMORIES
                  ECTOR
                  VECTOR

IO__SECTION
  - N_IO_DEVICES
  - FPGA_WITH_IO_DEVICES_VECTOR

  - TYPE_IO_DEVICES_VECTOR

  - TILES_CONNECTED_TO_IO_DEVICE_
  - NUM_IO_DEVICES_PER_FPGA_VECTOR
  - USED_FOR_VECTOR

   ECTION,
   AY_TILE_FOR_GLBL_ITEMS             This is the gateway tile that will provide glohlal configuration items to the top modules that are not tile specific

       N,                             Constraints that must be applied to macros depending on FPGA to be synthetized or Behavioral mode.
                                      Things in this section are architecture ad-hoc, but they must be based on known defines
                                      It is used for synthesis and implementation. It indicates the id of the fpga to be synthetize.
                                      Use FPGA_ID__SECTION defines for defining the correct FPGA_ID
                                      Apply constraints for behavioral and non behavioral.
                                      Take into account that FAST_SIMULATION always requires DDR_AS_BRAM and no I/O device instantion (otherwise wrong behaviours could arise)
```

```
IO__SECTION
  - N_IO_DEVICES
  - FPGA_WITH_IO_DEVICES_VECTOR

  - TYPE_IO_DEVICES_VECTOR

  - TILES_CONNECTED_TO_IO_DEVICE_VECTOR
  - NUM_IO_DEVICES_PER_FPGA_VECTOR
  - USED_FOR_VECTOR
```

Architecture Definition File

- Sections
  - General
  - FPGA/multi-FPGA
  - Memory devices
  - I/O devices
  - **Network**

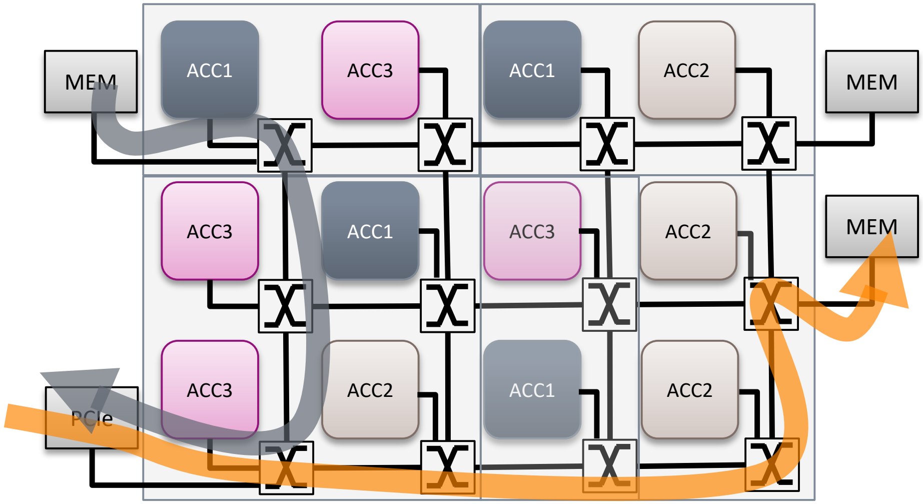


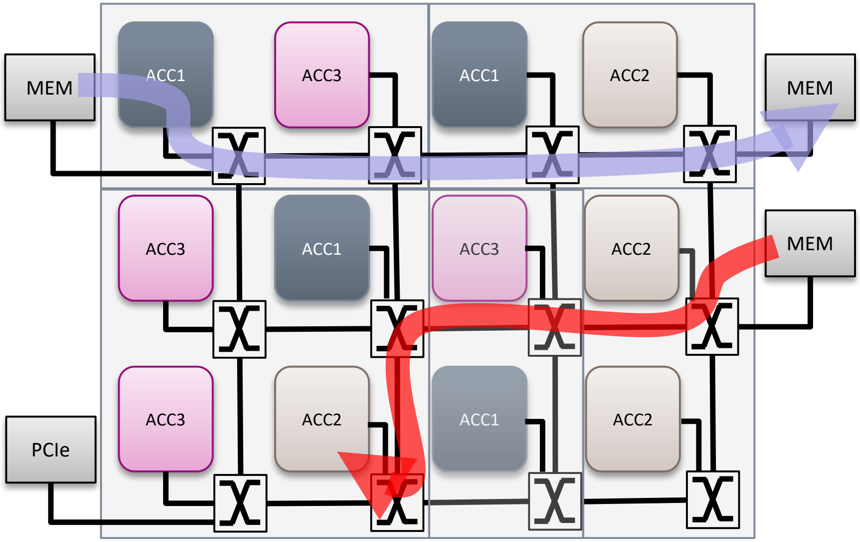
```
DATANET__SECTION

  – VN_WITH_PRIORITIES
  – DATA_NET_NUM_VC_PER_VN
  – MANGO_DATA_NET_NUM_VN
  – DATA_NET_FLIT_w
  – DATA_NET_PHIT_w
  – MANGO_DATA_NET_VNID_VECTOR
  – MANGO_DATA_NET_VN_PRIORITY_VECTOR_w
  – MANGO_VN_WEIGHT_PRIORITIES
```

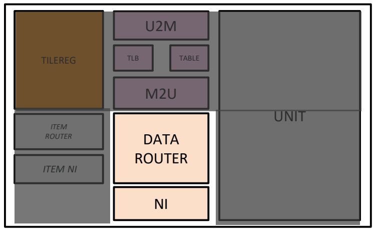
```
  - ARCH_SIGNATURE
  - ARCH_VALUE_w                      Defines the width of the values used for defining the architecture. 32 bits are used for the examples described in the next sections

FPGA_DESC__SECTION,                   Used for describing the type of FPGA in that slot
  - MB__FPGA_SLOT__DESC
  - ...

FPGA_ID__SECTION,                     Used for coding every FPGA in the architecture. Later, this ID will be used for futher purposes
  - MB__FPGA_SLOT__ID

MULTI_FPGA__SECTION,                  Used for describing multi-fpga architectures
  - N_FPGAS                           Number of FPGAs used for an archictecture definition
  - TILE_IDS_PER_FPGA_VECTOR          Tile layout per FPGA
  - TILE_TYPE_VECTOR                  Type (accelerator inside) of each tile
                                      e.g., Two tiles splitted in two FPGAs with one tile per FPGA /*FPGA_0*/32'd1,/*FPGA_0*/32'd0   // Vector of tiles in each FPGA.
  - TILE_IDS_PER_FPGA_OFFSET_VECTOR   Defines the offset of where the tiles start in the TILES_ID_PER_FPGA_VECTOR
  - MULTI_FPGA                        Defined for architectures splitted in multiple FPGAs (`undef MULTI_FPGA for single FPGA architecture implementation)
                                      The next pararameter in this section are required for MULTI_FPGA designs only.
  - USE_DAISY_CHAIN                   Defined for routing specific (reset, congested, etc.) signals among FPGA in multi-fpga designs using daisy chain
  - DAISY_CHAIN_DIR_FROM_FIRST_VECTOR Directions (NEWS) from the first fpga in daisy chain point of view.
                                      It codes the output ports (NEWS) that each FPGA provides for connecting to the next one in the daisy chain vector
                                      Take into account that the indexes of this vector are relative to the FPGA_IDs and not to the order of the DAISY_CHAIN_VECTOR
  - DAISY_CHAIN_DIR_FROM_LAST_VECTOR  Directions (NEWS) from the last fpga in daisy chain point of view. Generate this vector as the previous one
  - DAISY_CHAIN_VECTOR                Defines the sequence tiles vector in the daisy chain for specific signals connection (from right to left)
                                      For single-fpgas architectures the next pararameters of this section are not required
  - CONNECTIVITY_PORTS                Number of in/out ports for FPGA connections
  - FPGA_CONNECTIVITY_OUT_VECTOR      Connectivity layout among FPGAs for output ports
                                      Format: /*High ID FPGA*/ N,E,W,S, ... /*Low ID FPGA*/ N,E,W,S
                                      N,E,W,S can be 1'b1 for indicating connectivity across the stated direction. Use 1'b0 otherwise
                                      e.g., For two FPGAs connected using N<->S links: /*FPGA_1*/1'b1,1'b0,1'b0,1'b0,/*FPGA_0*/1'b0,1'b0,1'b0,1'b1
  - FPGA_CONNECTIVITY_IN_VECTOR       Connectivity layout among FPGAs for inputs ports. Same format as FPGA_CONNECTIVITY_OUT_VECTOR
  - FPGA_NEIGHBOURS_VECTOR            Neighbours for an FPGA. It is a full matrix vector
                                      Use FPGA_IDs for coding the vector
                                      Format: /*High ID FPGA*/ N,E,W,S, ... /*Low ID FPGA*/ N,E,W,S
                                      N,E,W,S are FPGA_IDs or 32'd0 if no neighbour in the given direction
                                      e.g., For two FPGAs connected using N<->S links: /*FPGA_1*/`MB0_A1__ID,32'd0,32'd0,32'd0,/*FPGA_0*/32'd0,32'd0,32'd0,`MB0_A3__ID
  - FPGA_PINOUT_VECTOR

                                      e.g., For two FPGAs connected using N<->S links /*FPGA_1*/`MB0_A3__MB0_A1_PINES,32'd0,32'd0,32'd0,/*FPGA_0*/32'd0,32'd0,32'd0,`MB0_A1__MB0_A3_P
  - FPGA_PININ_VECTOR

....

CLOCK__SECTION,

  - MMCM_CLKIN_FREQ
  - ACCELERATOR_FREQ
  ...

TOPOLOGY__SECTION,
  - MB__FPGA_SLOT__TOPOLOGY
  - GLOBAL_TOPOLOGY
  - TOPOLOGY_PER_FPGA_VECTOR
  - TOPOLOGY_PER_FPGA_VECTOR_w
  - TOPOLOGY_PER_FPGA_OFFSET_VECTO
  - FPGA_TOPOLOGY_WIDTH_VECTOR

MEMORY__SECTION
  - N_MEMORIES
  - FPGA_WITH_MEMORY_VECTOR

  - TYPE_MEMORY_VECTOR

  - TILES_CONNECTED_TO_MC_VECTOR
  - NUM_MEMORIES_P    FPGA_VECTOR

  - TILES_CONNECTE    IO_DEVICE_
  - NUM_IO_DEVICES    FPGA_VECTO
  - USED_FOR_VECTO

  SECTION,
    AY_TILE_FOR_GLBL_ITEMS

                                      Constraints that must be applied to macros depending on FPGA to be synthetized or Behavioral mode.
                                      Things in this section are architecture ad-hoc, but they must be based on known defines
                                      It is used for synthesis and implementation. It indicates the id of the fpga to be synthetize.
                                      Use FPGA_ID__SECTION defines for defining the correct FPGA_ID
                                      Apply constraints for behavioral and non behavioral.
                                      Take into account that FAST_SIMULTATION always requires DDR_AS_BRAM and no I/O device instantion (otherwise wrong behaviours could arise)
```

DMA Transfers

- DMA controller attached to every memory controller.
- Programming parameters: base address, source tile, size of the transfer, destination of the transfer, target address.
- Possibilities:

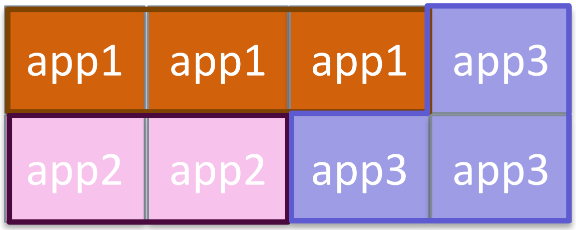




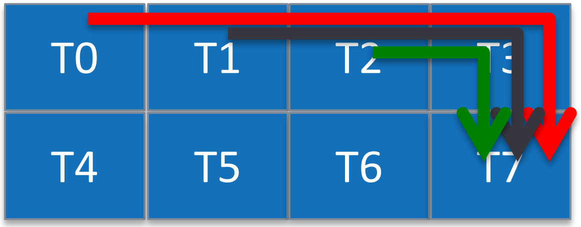


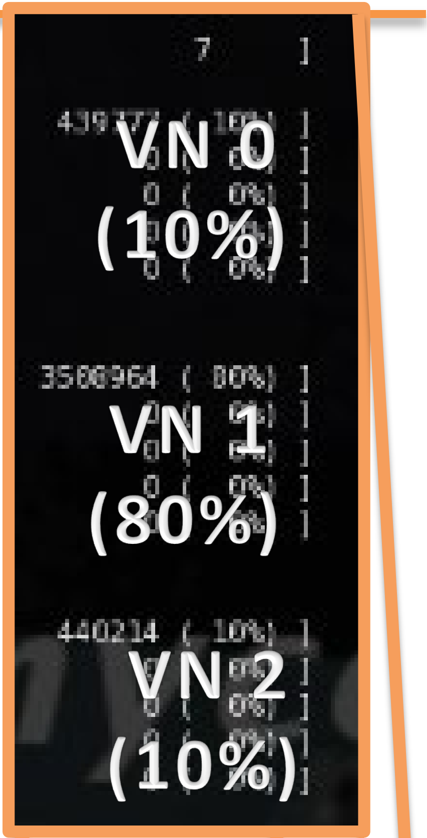
Data Network

- DATA Routers connected among them in a 2D mesh layout.
- NI decouples network from tile components.
- Supported different Virtual networks (VN) with different number of Virtual channels (VC).
- Supported dynamic assignment of bandwidth per VN.

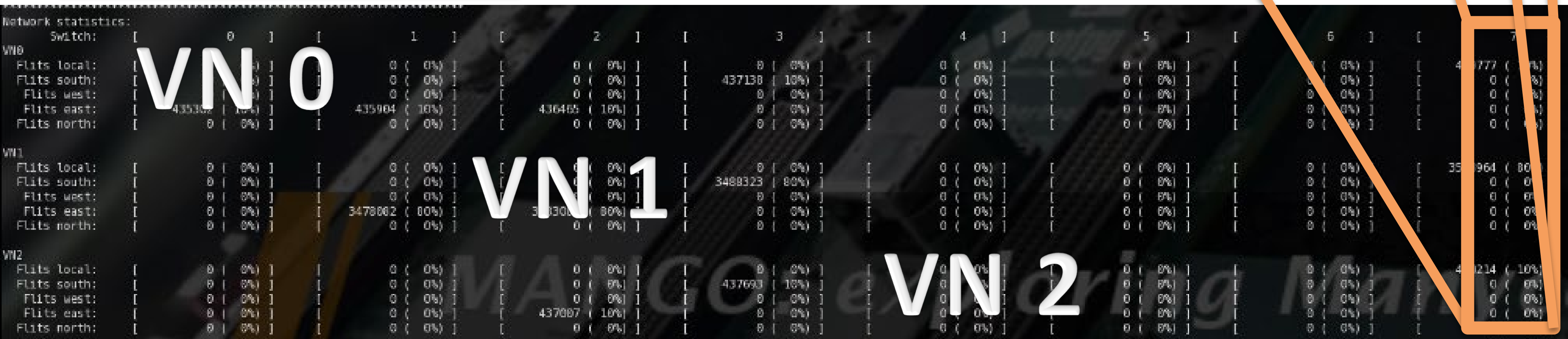


app1	app1	app1	app3
app2	app2	app3	app3





```
      7       ]

439777 ( 10%) ]
     0 (  0%) ]
     0 (  0%) ]
     0 (  0%) ]
     0 (  0%) ]

3506964 ( 80%) ]
     0 (  0%) ]
     0 (  0%) ]
     0 (  0%) ]
     0 (  0%) ]

440214 ( 10%) ]
     0 (  0%) ]
     0 (  0%) ]
     0 (  0%) ]
     0 (  0%) ]
```



```
Network statistics:
      Switch:   [        0     ]   [        1     ]   [        2     ]   [        3     ]   [        4     ]   [        5     ]   [        6     ]   [        7     ]
VN0
  Flits local:  [  [illegible] ]   [       0 (  0%) ]   [       0 (  0%) ]   [       0 (  0%) ]   [       0 (  0%) ]   [       0 (  0%) ]   [       0 (  0%) ]   [  439777 ( 10%) ]
  Flits south:  [  [illegible] ]   [       0 (  0%) ]   [       0 (  0%) ]   [  437138 ( 10%) ]   [       0 (  0%) ]   [       0 (  0%) ]   [       0 (  0%) ]   [       0 (  0%) ]
   Flits west:  [  [illegible] ]   [       0 (  0%) ]   [       0 (  0%) ]   [       0 (  0%) ]   [       0 (  0%) ]   [       0 (  0%) ]   [       0 (  0%) ]   [       0 (  0%) ]
   Flits east:  [  435302 ( 10%) ]   [  435904 ( 10%) ]   [  436465 ( 10%) ]   [       0 (  0%) ]   [       0 (  0%) ]   [       0 (  0%) ]   [       0 (  0%) ]   [       0 (  0%) ]
  Flits north:  [       0 (  0%) ]   [       0 (  0%) ]   [       0 (  0%) ]   [       0 (  0%) ]   [       0 (  0%) ]   [       0 (  0%) ]   [       0 (  0%) ]   [       0 (  0%) ]

VN1
  Flits local:  [       0 (  0%) ]   [       0 (  0%) ]   [       0 (  0%) ]   [       0 (  0%) ]   [       0 (  0%) ]   [       0 (  0%) ]   [       0 (  0%) ]   [ 3506964 ( 80%) ]
  Flits south:  [       0 (  0%) ]   [       0 (  0%) ]   [       0 (  0%) ]   [ 3488323 ( 80%) ]   [       0 (  0%) ]   [       0 (  0%) ]   [       0 (  0%) ]   [       0 (  0%) ]
   Flits west:  [       0 (  0%) ]   [       0 (  0%) ]   [       0 (  0%) ]   [       0 (  0%) ]   [       0 (  0%) ]   [       0 (  0%) ]   [       0 (  0%) ]   [       0 (  0%) ]
   Flits east:  [       0 (  0%) ]   [ 3478002 ( 80%) ]   [ [illegible] ( 80%) ]   [       0 (  0%) ]   [       0 (  0%) ]   [       0 (  0%) ]   [       0 (  0%) ]   [       0 (  0%) ]
  Flits north:  [       0 (  0%) ]   [       0 (  0%) ]   [       0 (  0%) ]   [       0 (  0%) ]   [       0 (  0%) ]   [       0 (  0%) ]   [       0 (  0%) ]   [       0 (  0%) ]

VN2
  Flits local:  [       0 (  0%) ]   [       0 (  0%) ]   [       0 (  0%) ]   [       0 (  0%) ]   [       0 (  0%) ]   [       0 (  0%) ]   [       0 (  0%) ]   [  440214 ( 10%) ]
  Flits south:  [       0 (  0%) ]   [       0 (  0%) ]   [       0 (  0%) ]   [  437693 ( 10%) ]   [       0 (  0%) ]   [       0 (  0%) ]   [       0 (  0%) ]   [       0 (  0%) ]
   Flits west:  [       0 (  0%) ]   [       0 (  0%) ]   [       0 (  0%) ]   [       0 (  0%) ]   [       0 (  0%) ]   [       0 (  0%) ]   [       0 (  0%) ]   [       0 (  0%) ]
   Flits east:  [       0 (  0%) ]   [       0 (  0%) ]   [  437007 ( 10%) ]   [       0 (  0%) ]   [       0 (  0%) ]   [       0 (  0%) ]   [       0 (  0%) ]   [       0 (  0%) ]
  Flits north:  [       0 (  0%) ]   [       0 (  0%) ]   [       0 (  0%) ]   [       0 (  0%) ]   [       0 (  0%) ]   [       0 (  0%) ]   [       0 (  0%) ]   [
```