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**

- 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)
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

- 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

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
HN Hardware and Assembly

Lego-like exercise
Heterogeneity: Pool of Accelerators

GPU-like manycore

manycore: MIPS, RISCV

General purpose

Custom made
Heterogeneity

Tile concept

Element connection

- Data path
- Flow control
- Control path
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.
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
Bitstream file

proFPGA Configuration file

XILINX tools

prodesign electronic gmbh

Nov 11th, 2018  H2RC’18
Architecture Definition File

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

• Ease to configure the Unit type for every 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
**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
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
#define FPGA_WITH_MEMORY_VECTOR
#define TYPE_MEMORY_VECTOR
#define TILES_CONNECTED_TO_MC_VECTOR
#define NUM_MEMORIES_PER_FPGA_VECTOR

1
'MB0_A1__ID
'FB_PPS_DDR3_R2_TYPE
10'd0
10'01

ARCHITECTURE ID      0
ARCHITECTURE ID      1
```

Nov 11th, 2018

H2RC’18 23
FPGA Resource utilization

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

Nov 11th, 2018
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
Architecture Definition

- 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,
  - MCGM_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
Architecture Definition File

- Sections
  - General
  - FPGA/multi-FPGA
  - Memory devices
  - I/O devices
  - Network
Architecture Definition File

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

```
DATA_NET__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
```
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.