Conference on Design and Architectures for Signal and Image Processing -2011
Electronic Chips & Systems design Initiative

dasip

ecsi

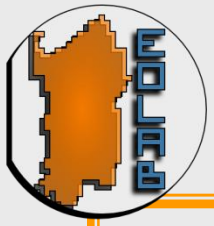
November 2nd-4th, 2011,
Tampere, Finland

# THE MULTI-DATAFLOW COMPOSER TOOL: A RUNTIME RECONFIGURABLE HDL PLATFORM COMPOSER

*Francesca Palumbo, Nicola Carta and Luigi Raffo*

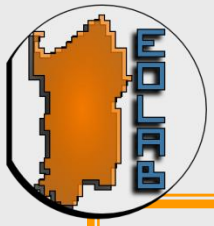
EOLAB - Microelectronics Lab
DIEE - Dept. of Electrical and Electronic Eng.
University of Cagliari - ITALY

Diee



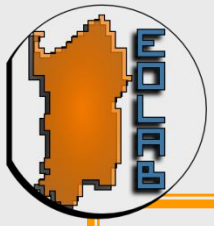
# Outline

- Introduction:
  - Problem formulation
  - Background
  - Goals



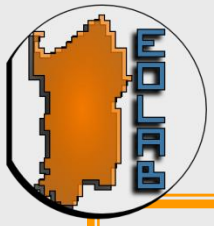
# Outline

- Introduction:
  - Problem formulation
  - Background
  - Goals
- The Multi-Dataflow Composer tool



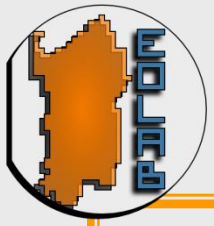
# Outline

- Introduction:
  - Problem formulation
  - Background
  - Goals
- The Multi-Dataflow Composer tool
- Performance assessment
  - Use-case scenario
  - Results



# Outline

- Introduction:
  - Problem formulation
  - Background
  - Goals
- The Multi-Dataflow Composer tool
- Performance assessment
  - Use-case scenario
  - Results
- Future research directions and conclusions
  - RVC extension
  - Applicable research hot topics
  - Final remarks



# Outline

- Introduction:
  - Problem formulation,
  - Background
  - Goals
- The Multi-Dataflow Composer tool
- Performance assessment
  - Use-case scenario
  - Results
- Future research directions and conclusions
  - RVC extension
  - Applicable research hot topics
  - Final remarks

# Scenario and Problem Statement

- Systems and applications on the market are becoming every day more ***complex***. We will be called to face the ***"the disappearing computer" phenomenon*** [Streit2005] [i.e. implicit interfaces, users could be un aware].

## APPLICATION TRENDS

### ICT TRENDS

- Ubiquitous access
- Personalized services
- Delocalized computing and storage
- Massive data processing systems
- High-quality virtual reality
- Intelligent sensing
- High-performance real-time embedded computing

### EXAMPLES

- Domestic robot
- Telepresence
- The car of the future
- Aerospace and avionics
- Human ++
- Computational science
- Realistic games
- Smart camera networks

SOURCE: http://www.hipeac.net/roadmap

# Scenario and Problem Statement

- Systems and applications on the market are becoming every day more ***complex***. We will be called to face the ***"the disappearing computer" phenomenon*** [Streit2005] [i.e. implicit interfaces, users could be un aware].

APPLICATION TRENDS

**INTEGRATION, SPECIALIZATION and HIGH PERFORMANCE REQUIREMENTS**

**in such**

**COMPLEX COMPUTATIONAL HUNGRY ENVIRONMENTS**

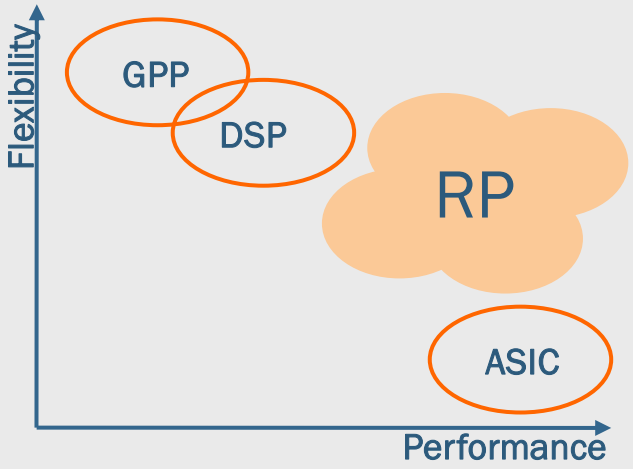
**threaten**

**TRADITIONAL DESIGN FLOW.**

SOURCE: http://www.hipeac.net/roadmap

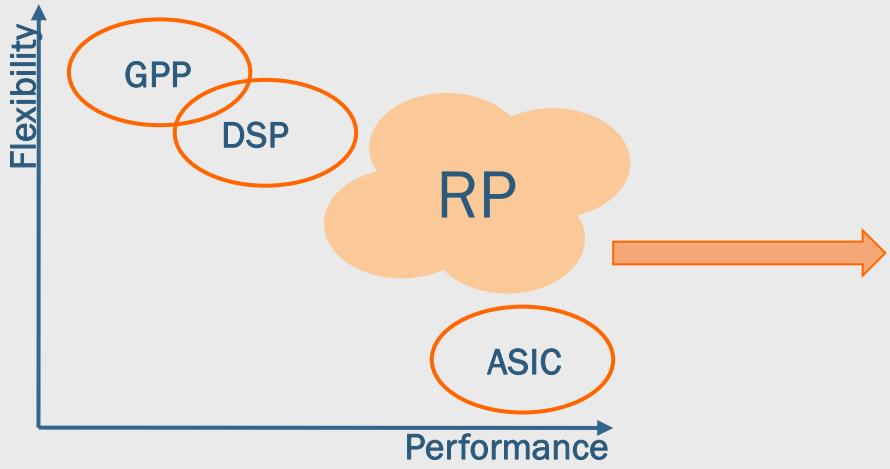
# STEP1: Reconfigurable Paradigm

- Systems are required to be ***flexible*** and ***efficient***.
- Reconfigurable Paradigm (RP) to hw design: specialized computing platforms, capable of changing configuration to serve the targeted computations.



# STEP1: Reconfigurable Paradigm

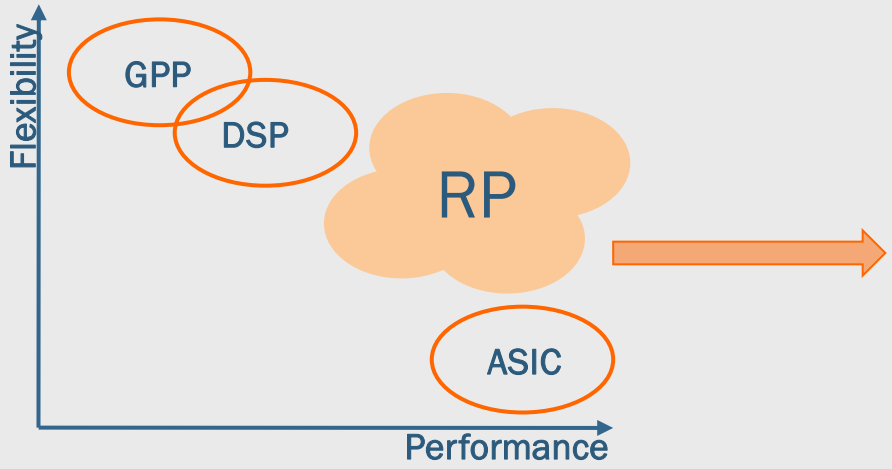
- Systems are required to be ***flexible*** and ***efficient***.
- Reconfigurable Paradigm (RP) to hw design: specialized computing platforms, capable of changing configuration to serve the targeted computations.




| | FINE-GRAINED | COARSE-GRAINED |
|---|---|---|
| | Bit-level | Word-level |
| Flexibility | ☺ | 😐 |
| Reconf. Speed | 😐 | ☺ |
| Config. Storage | ☹ | 😐 |

# STEP1: Reconfigurable Paradigm

- Systems are required to be ***flexible*** and ***efficient***.
- Reconfigurable Paradigm (RP) to hw design: specialized computing platforms, capable of changing configuration to serve the targeted computations.



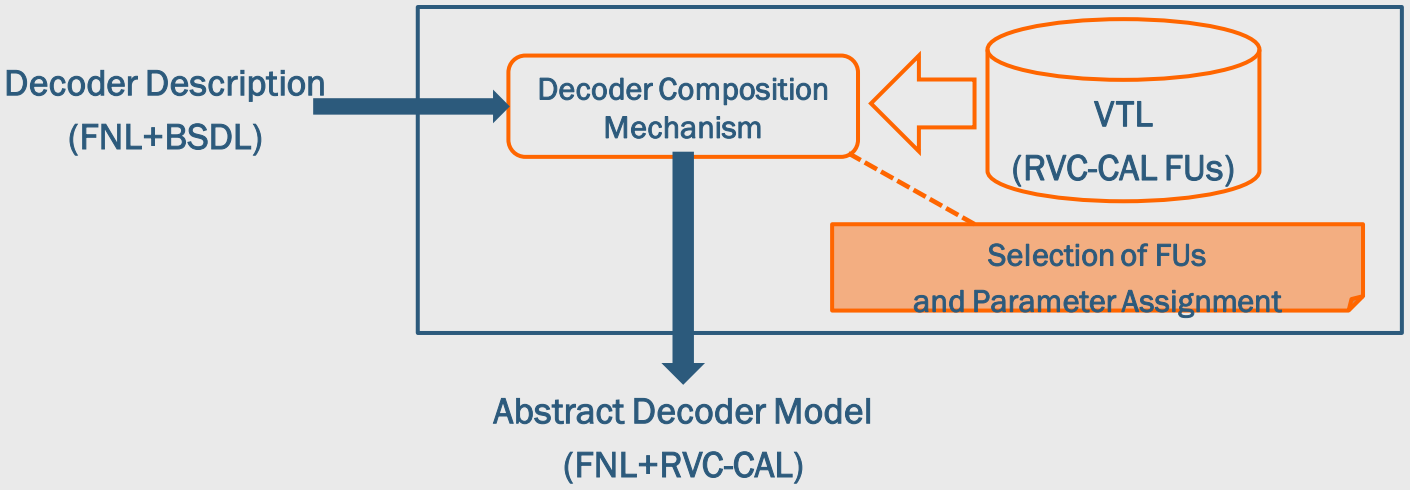

| | FINE-GRAINED | COARSE-GRAINED |
|---|---|---|
| | Bit-level | Word-level |
| Flexibility | ☺ | 😐 |
| Reconf. Speed | 😐 | ☺ |
| Config. Storage | ☹ | 😐 |

**HW-SW GAP:**

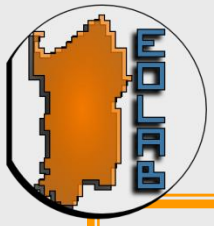
**The more the hw is specialized the more is difficult to program it.**

# STEP 2: RVC Standard

- The MPEG group has addressed the problem of defining an efficient formalism for codecs specification: the Reconfigurable Video Coding (RVC) framework is part of the MPEG standard since may 2010.




- Exploiting the Dataflow Model of Computation (D-MoC), specifications are provided in the form of dataflow programs: networks of Functional Units (FUs) belonging to a standard Video Tool Library (VTL).



# Goals and Research Evolution

RVC modularity can be coupled with a coarse-grained RP map on a unique hw substrate multiple D-MoC models.

# Goals and Research Evolution

**RVC modularity can be coupled with a coarse-grained RP map on a unique hw substrate multiple D-MoC models.**

- High-level dataflow combination tool, front-end of the actual MDC tool. [DASIP 2010]

# Goals and Research Evolution

**RVC modularity can be coupled with a coarse-grained RP map on a unique hw substrate multiple D-MoC models.**

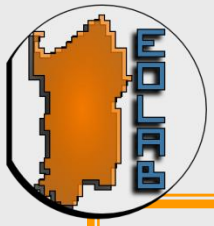
- High-level dataflow combination tool, front-end of the actual MDC tool. [DASIP 2010]

- Multi-Dataflow Composer (MDC) tool: concrete definition of the hardware template and of the D-MoC based mapping strategy. [DASIP 2011]

# Goals and Research Evolution

**RVC modularity can be coupled with a coarse-grained RP map on a unique hw substrate multiple D-MoC models.**

- High-level dataflow combination tool, front-end of the actual MDC tool. [DASIP 2010]
- Multi-Dataflow Composer (MDC) tool: concrete definition of the hardware template and of the D-MoC based mapping strategy. [DASIP 2011]
- Integration of the full high-level to hw composition and generation framework.

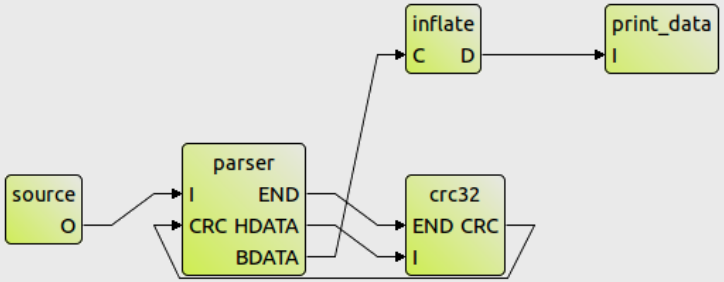


# Outline

- Introduction:
  - Problem formulation,
  - Background
  - Goals
- The Multi-Dataflow Composer tool
- Performance assessment
  - Use-case scenario
  - Results
- Future research directions and conclusions
  - RVC extension
  - Applicable research hot topics
  - Final remarks

# D-MoC and Coarse-Grained RP

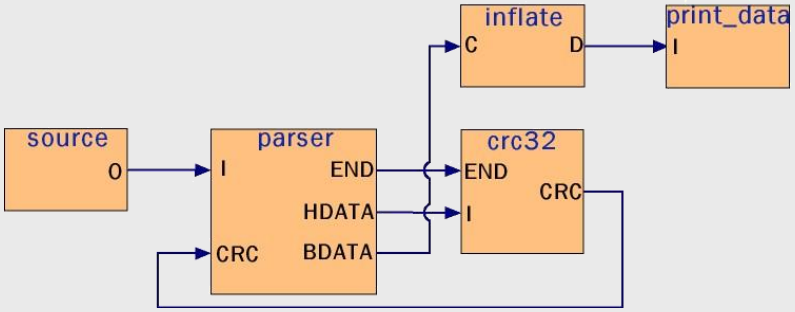
## D-MoC-based Formalism




[SOURCE: http://orcc.sourceforge.net/]

1:1

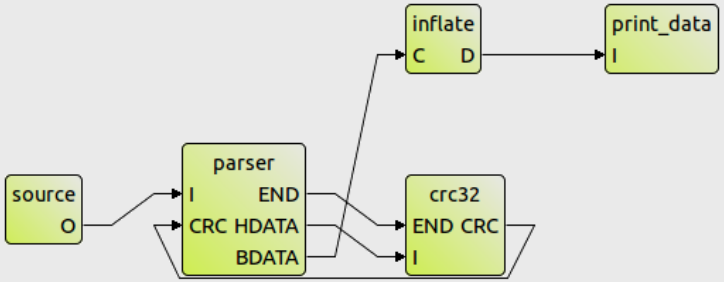
## HW Platform



# D-MoC and Coarse-Grained RP



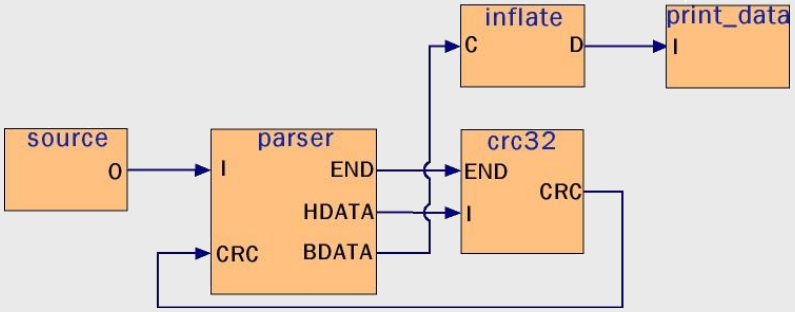
## D-MoC-based Formalism



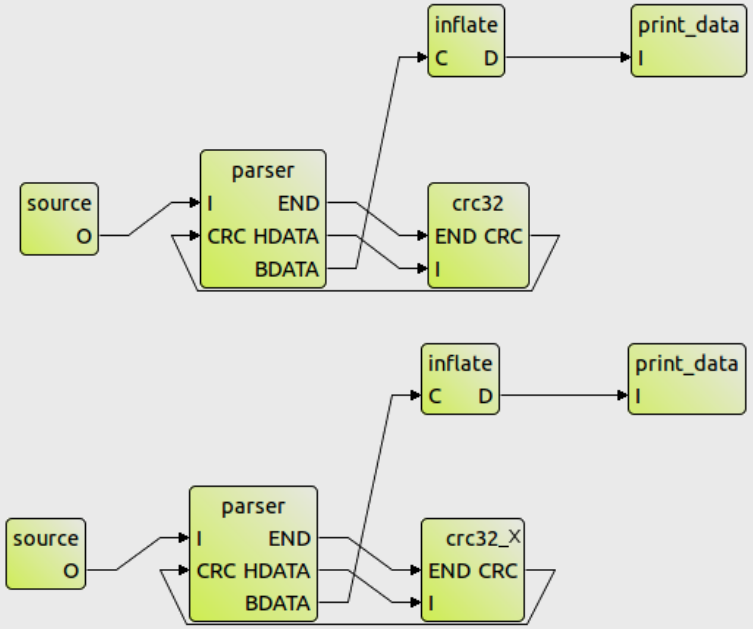

[SOURCE: http://orcc.sourceforge.net/]

1:1

## HW Platform

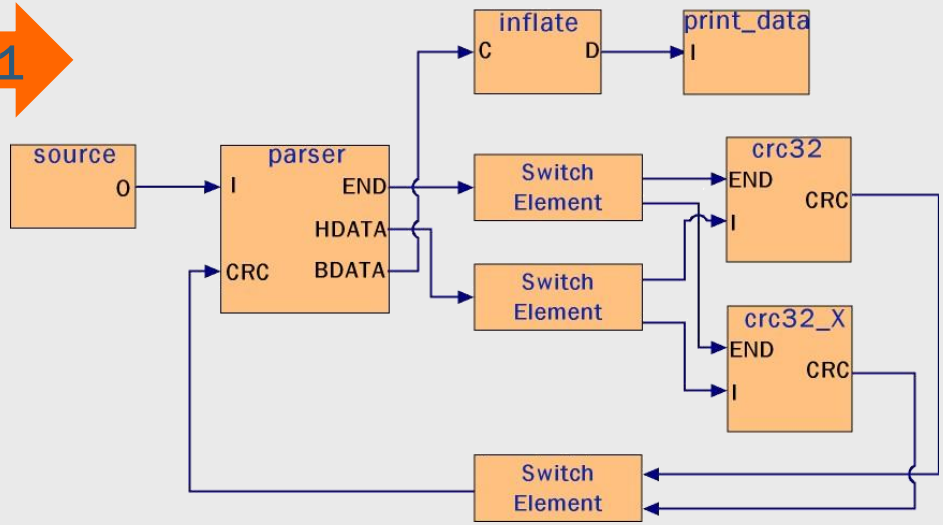



## D-MoC-based Formalism

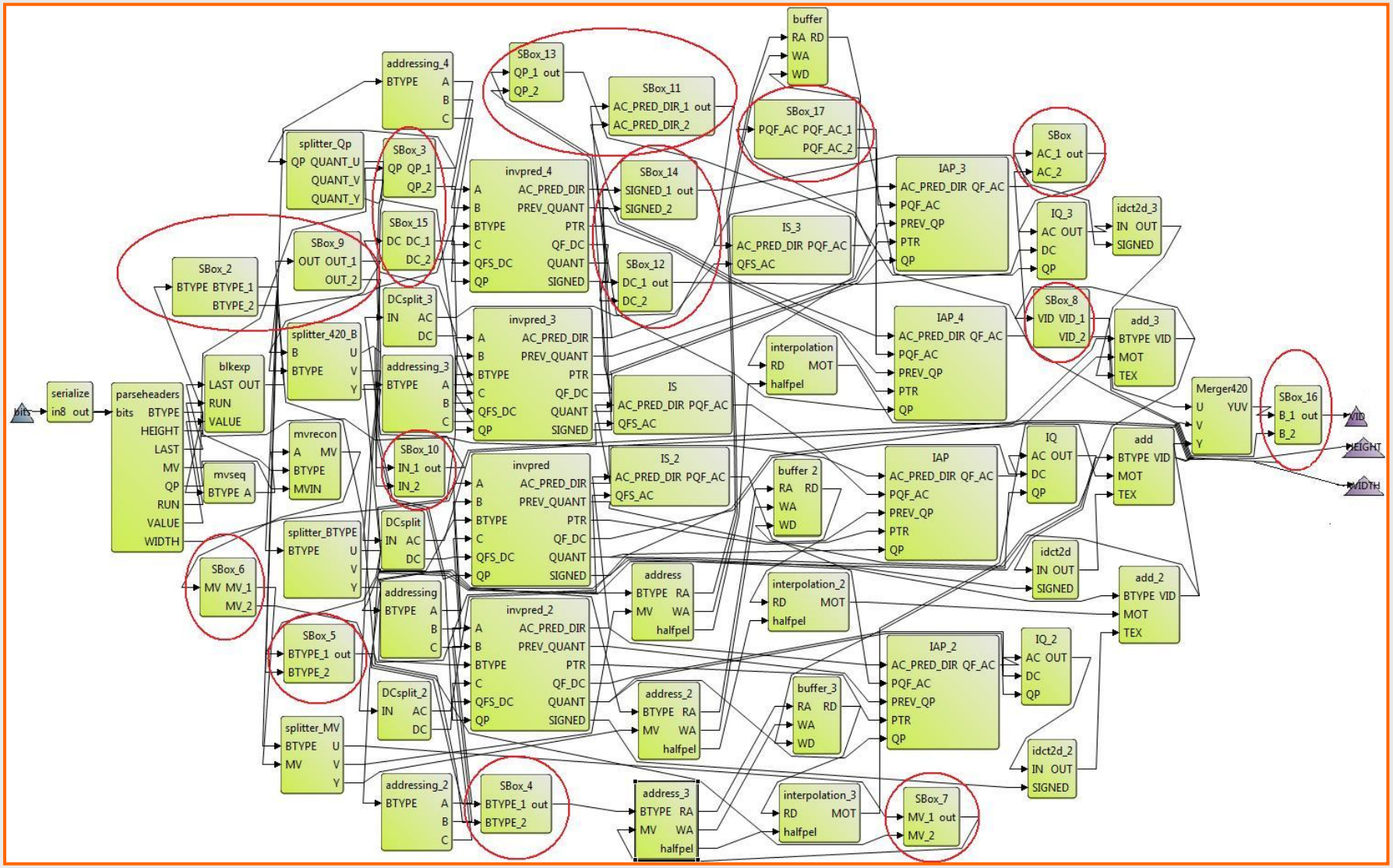



2:1

## Coarse Grained Reconfigurable HW Platform



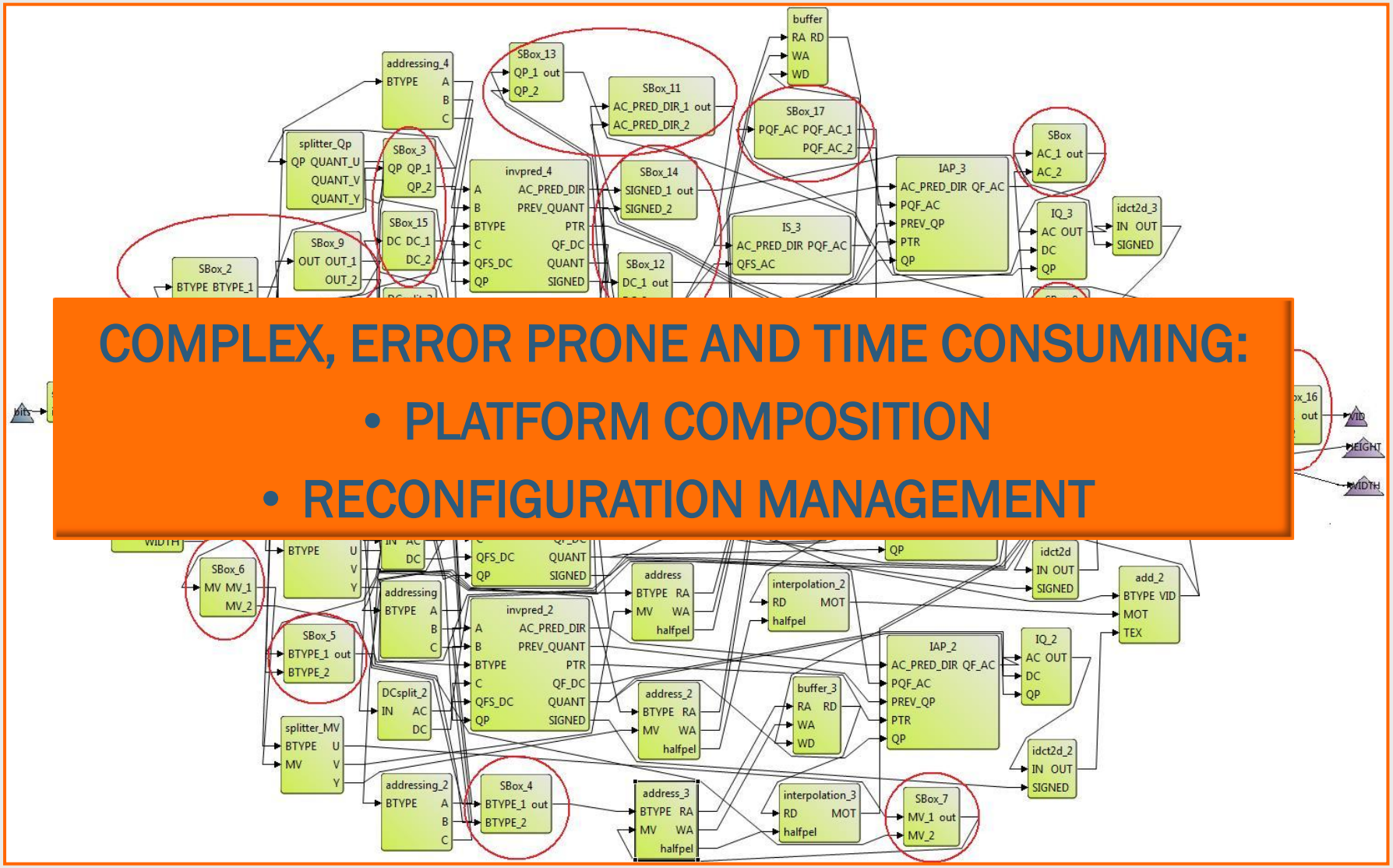
# Parallel and Serial MPEG-4 SP




DASIP 2011 – November 2nd-4th – Tampere, Finland

F. Palumbo et.al. , *"RVC: A multi-decoder CAL composer tool"*, in *Proc. DASIP 2010*]

# Parallel and Serial MPEG-4 SP



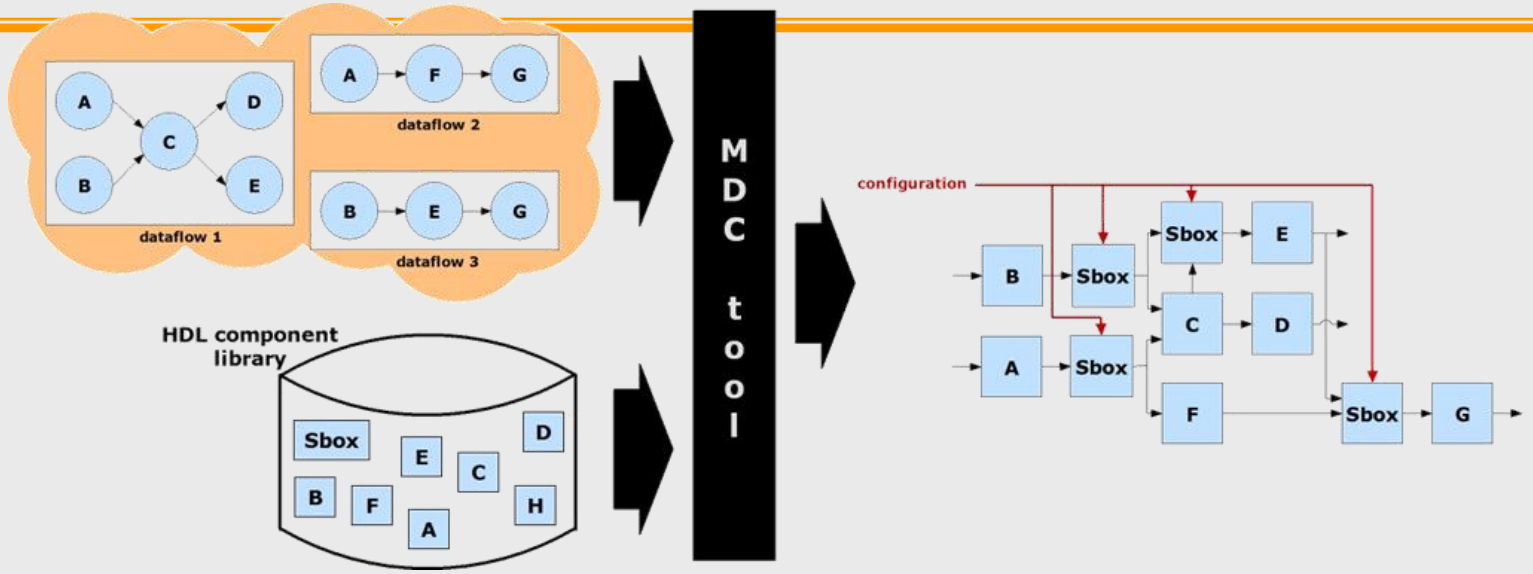

**COMPLEX, ERROR PRONE AND TIME CONSUMING:**

- **PLATFORM COMPOSITION**
- **RECONFIGURATION MANAGEMENT**



F. Palumbo et.al. , *"RVC: A multi-decoder CAL composer tool"*, in *Proc. DASIP 2010*]

# Multi-Dataflow Composer Tool



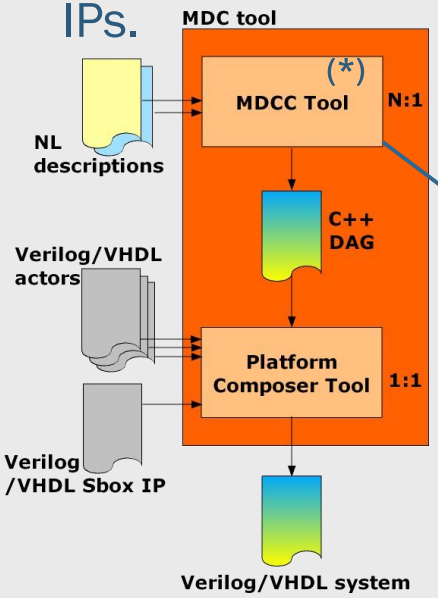

- The Multi-Dataflow Composer (MDC) tool **IS** an automatic platform constructor, composing different Functional Units (FUs) on a coarse-grained reconfigurable template.
- The MDC **IS** responsible of providing runtime programmability of the hw substrate to switch among given the dataflows.
- The MDC **IS NOT** capable of High Level Synthesis from dataflow to hw.

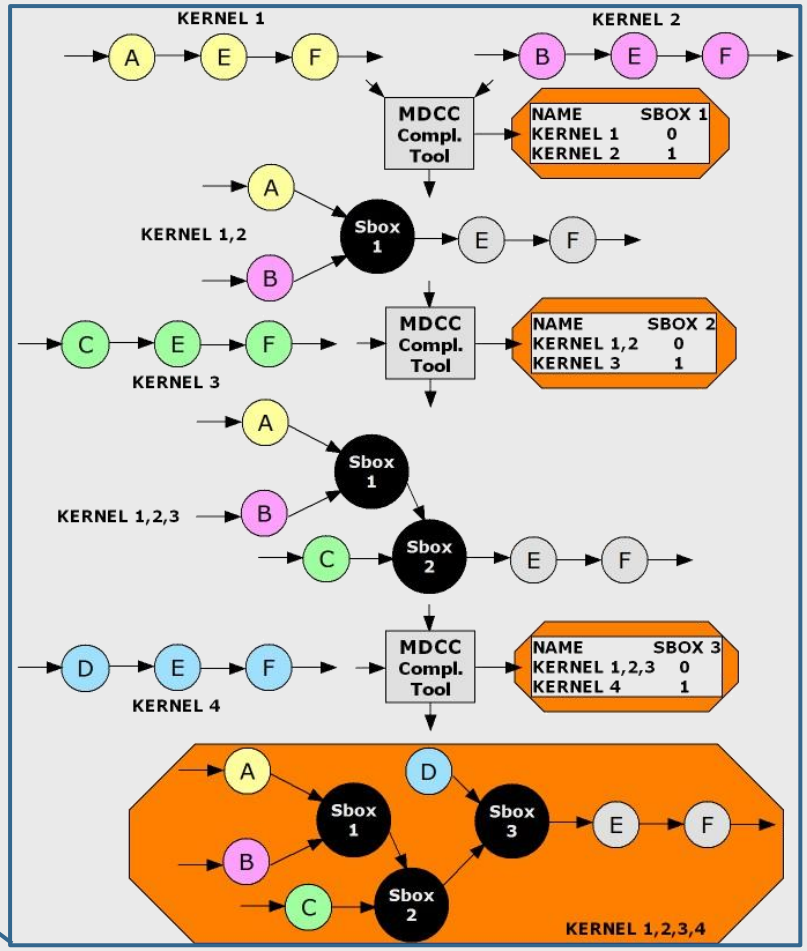
# MDC: Generalities

- The MDC tool, recognizing the similarities among different D-MoCs descriptions, automatically composes a unique reconfigurable multi-dataflow system:
  - exploiting heterogeneous blocks, the FUs in the input networks described according D-MoC formalism, with homogeneous interfaces;
  - integrating the minimum FUs set to correctly accomplish the provided dataflows.
- Reconfiguration is ensured by a couple of switching element, named switching box (Sbox):
  - inserted by the MDC tool at the crossroads among different dataflows to merge/separate the path of the processed data.
  - logically kept simple to provide high-speed reconfiguration (***one clock cycle is sufficient***).

# MDC: Front-End

- The MDC front-end:
  - Elaborates the input D-MoC inputs to create atomic actors (only) networks;
  - Translates the flattened networks into C++ Directed Acyclic Graphs (DAGs);
  - Compares the DAGs and merges them into a unique C++ DAG;
  - With respect to (*), it stores the information for the runtime reconfiguration, producing the configuration tables (CTs) of the Sbox IPs.

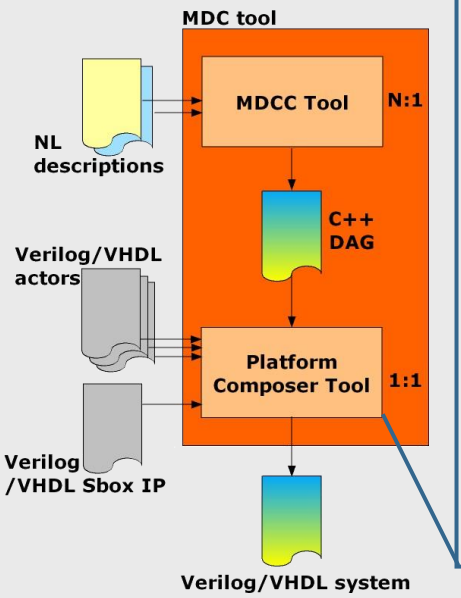


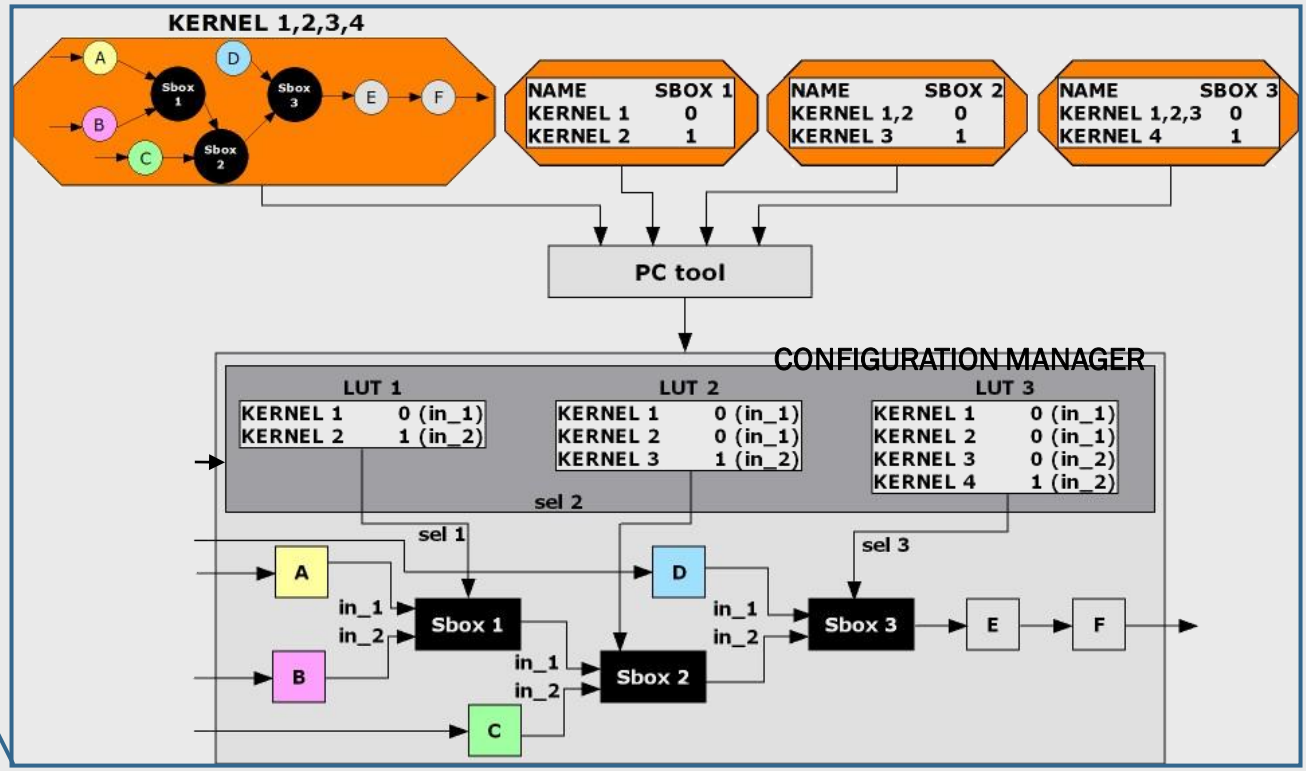



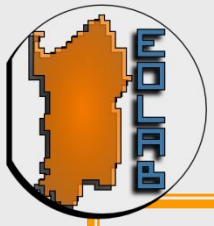

(*) [F. Palumbo et.al. , *"RVC: A multi-decoder CAL composer tool"*, in *Proc. DASIP 2010*]

# MDC: Back-End

- The MDC back-end is responsible of assembling the HDL Verilog coarse-grained reconfigurable hw, corresponding to the multi-dataflow C++ DAG produced by the MDC front-end.
- Having originally N different networks in input, N-1 LUTs are inserted in the final hw substrate, one for each CT created by the MDC front-end.



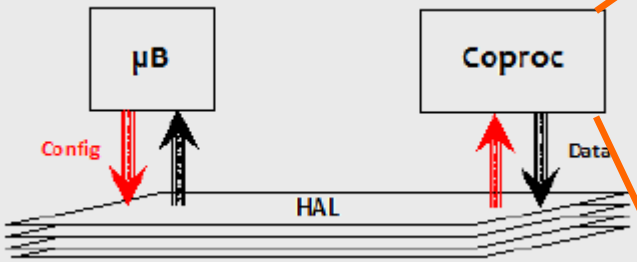



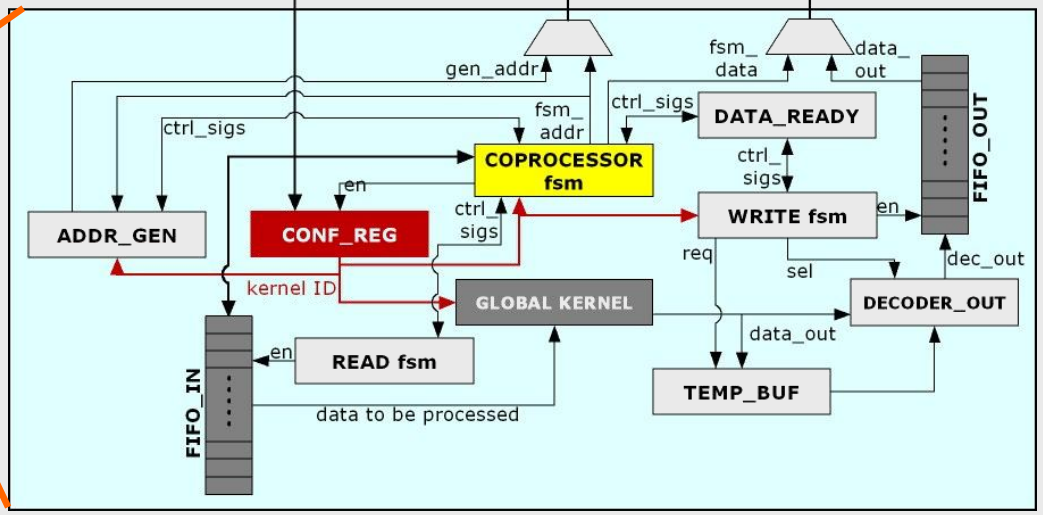


# Outline

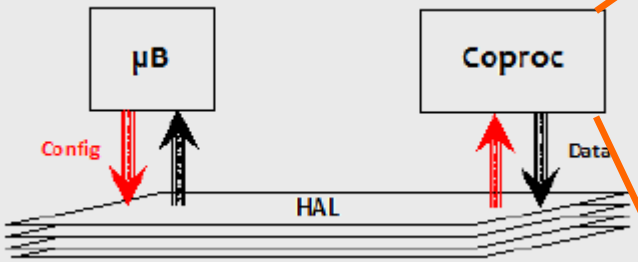
- Introduction:
  - Problem formulation,
  - Background
  - Goals
- The Multi-Dataflow Composer tool
- Performance assessment
  - Use-case scenario
  - Results
- Future research directions and conclusions
  - RVC extension
  - Applicable research hot topics
  - Final remarks

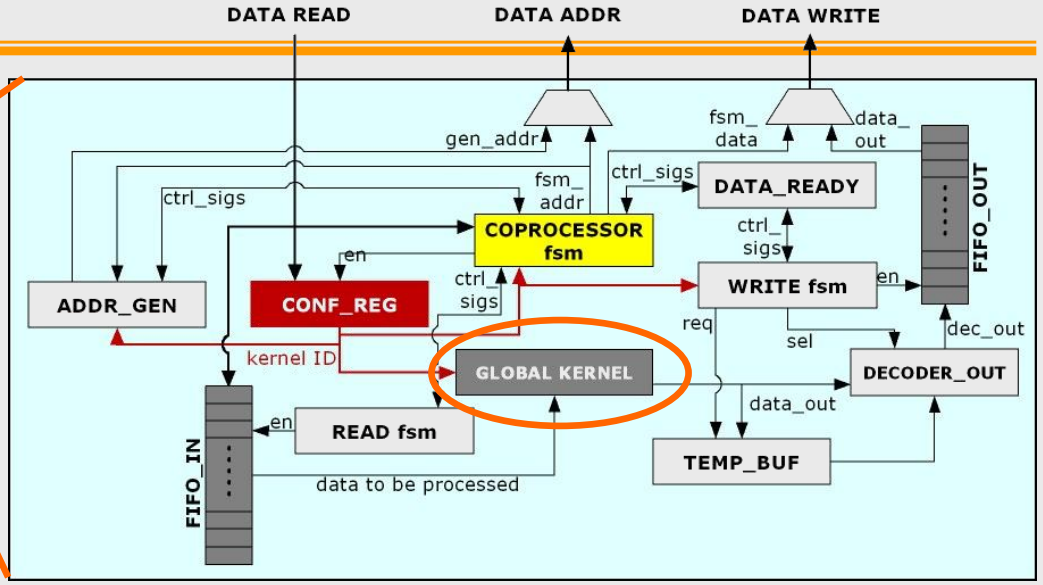
# Assessment: Design Under Test



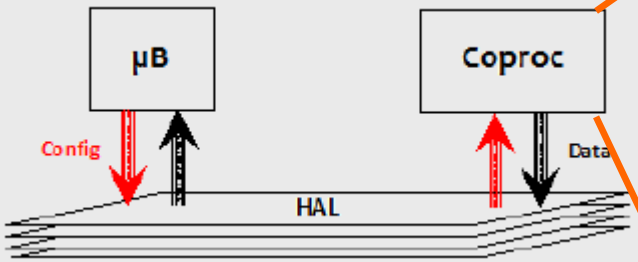
# Assessment: Design Under Test

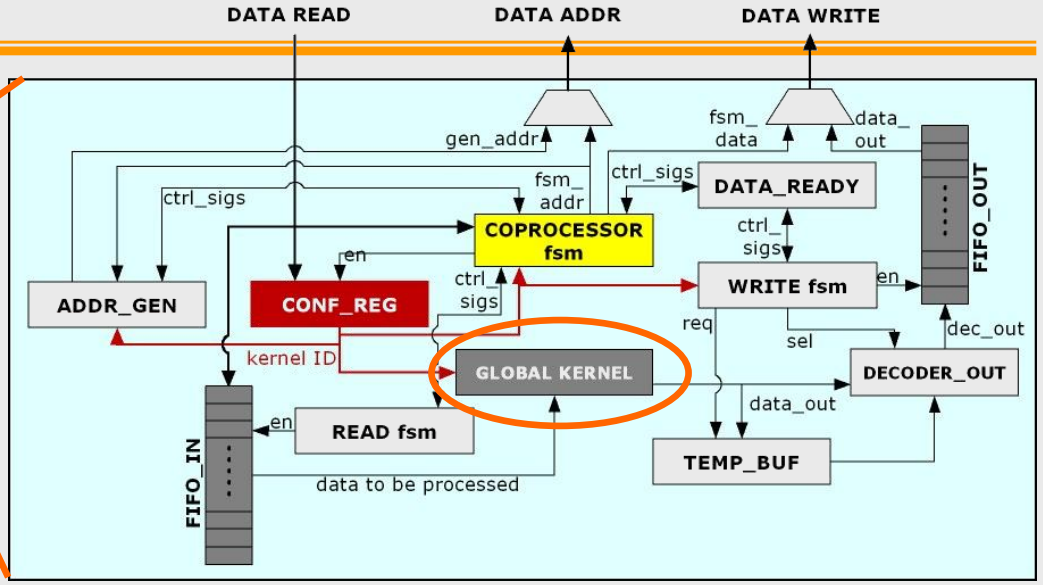


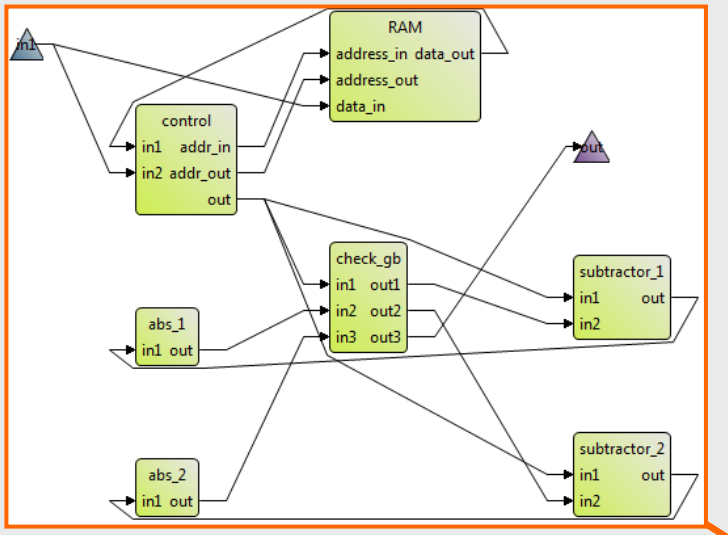




| | Zoom | Anti-Aliasing |
|---|---|---|
| Qsort | X | |
| Min_Max | X | X |
| Corr | X | |
| Abs | X | X |
| RGB2YCC | X | |
| YCC2RGB | X | |
| Sbwlabel | | X |
| Median | | X |
| Cubic | | X |
| Cubic_Conv | | X |
| Check_GeneralBilevel | | X |

# Assessment: Design Under Test

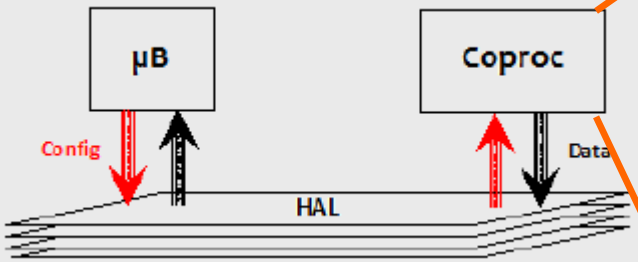


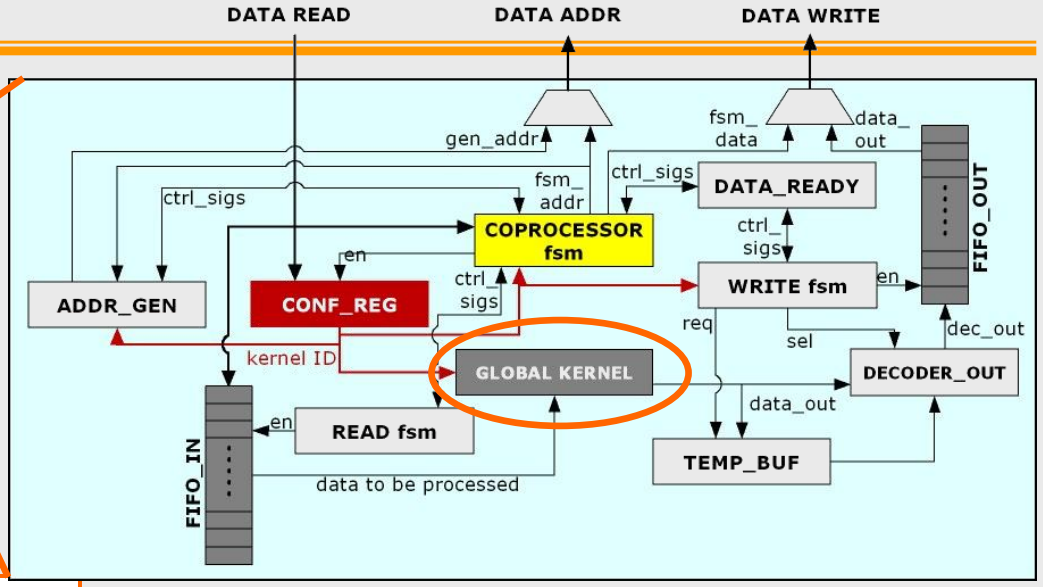



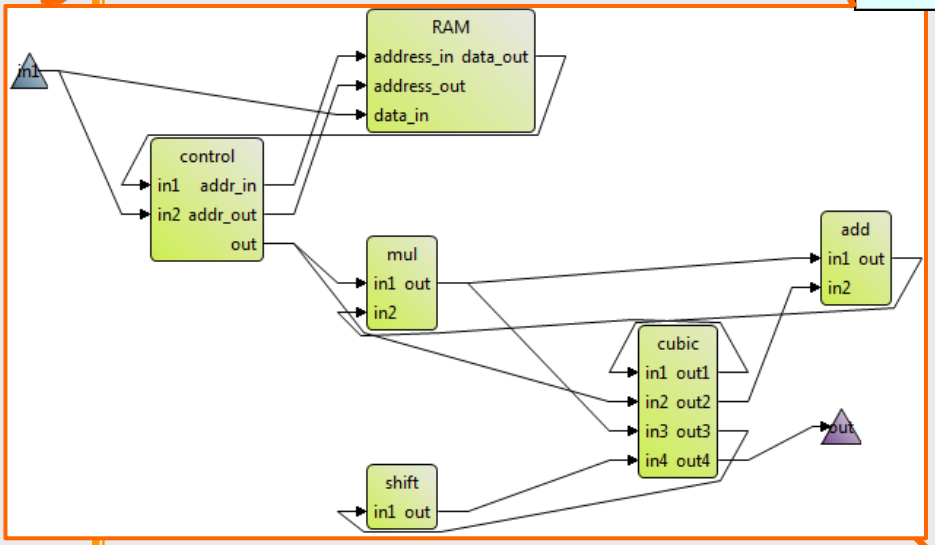




| | Zoom | Anti-Aliasing |
|---|---|---|
| Qsort | X | |
| Min_Max | X | X |
| Corr | X | |
| Abs | X | X |
| RGB2YCC | X | |
| YCC2RGB | X | |
| Sbwlabel | | X |
| Median | | X |
| Cubic | | X |
| Cubic_Conv | | X |
| Check_GeneralBilevel | | X |

# Assessment: Design Under Test

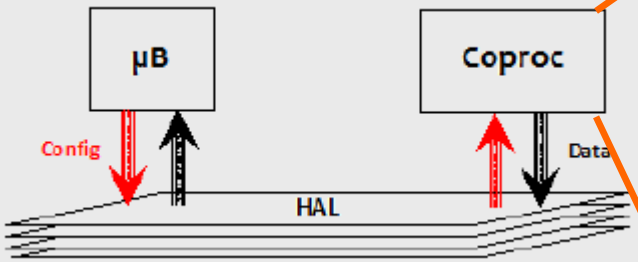


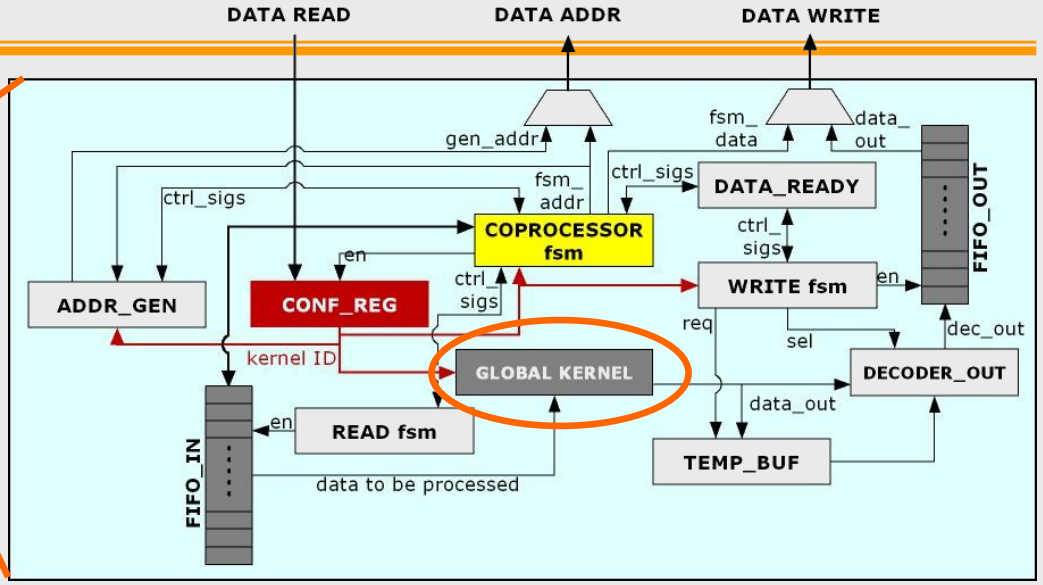




| | Zoom | Anti-Aliasing |
|---|---|---|
| Qsort | X | |
| Min_Max | X | X |
| Corr | X | |
| Abs | X | X |
| RGB2YCC | X | |
| YCC2RGB | X | |
| Sbwlabel | | X |
| Median | | X |
| Cubic | | X |
| Cubic_Conv | | X |
| Check_GeneralBilevel | | X |

# Assessment: Design Under Test




| | Zoom | Anti-Aliasing |
|---|---|---|
| Qsort | X | |
| Min_Max | X | X |
| Corr | X | |
| Abs | X | X |
| RGB2YCC | X | |
| YCC2RGB | X | |
| Sbwlabel | | X |
| Median | | X |
| Cubic | | X |
| Cubic_Conv | | X |
| Check_GeneralBilevel | | X |

# Assessment: Performance Results

[UC1: *Anti-Aliasing* application, UC2: *Zoom* application, UC3: *Anti-Aliasing* and *Zoom* applications together]

| | UC2 [No MDC] | UC2 [MDC] | UC1 [No MDC] | UC1 [MDC] | UC3 [No MDC] | UC3 [No MDC] |
|---|---|---|---|---|---|---|
| **Actor Sum** | 50 | 32 | 51 | 38 | 99 | 61 |
| LUT | 0 | 6 | 0 | 5 | 0 | 9 |
| cnv | 0 | 1 | 0 | 1 | 0 | 1 |
| Sbox | 0 | 68 | 0 | 47 | 0 | 137 |
| **Actor Sum** | 56 | 107 | 51 | 91 | 108 | 208 |

# Assessment: Performance Results

[UC1: *Anti-Aliasing* application, UC2: *Zoom* application, UC3: *Anti-Aliasing* and *Zoom* applications together]

| | UC2 [No MDC] | UC2 [MDC] | UC1 [No MDC] | UC1 [MDC] | UC3 [No MDC] | UC3 [No MDC] |
|---|---|---|---|---|---|---|
| **Actor Sum** | 50 | 32 | 51 | 38 | 99 | 61 |
| LUT | 0 | 6 | 0 | 5 | 0 | 9 |
| cnv | 0 | 1 | 0 | 1 | 0 | 1 |
| Sbox | 0 | 68 | 0 | 47 | 0 | 137 |
| **Actor Sum** | 56 | 107 | 51 | 91 | 108 | 208 |

Overhead

# Assessment: Performance Results

[UC1: *Anti-Aliasing* application, UC2: *Zoom* application, UC3: *Anti-Aliasing* and *Zoom* applications together]

| | UC2 [No MDC] | UC2 [MDC] | UC1 [No MDC] | UC1 [MDC] | UC3 [No MDC] | UC3 [No MDC] |
|---|---|---|---|---|---|---|
| **Actor Sum** | 50 | 32 | 51 | 38 | 99 | 61 |
| LUT | 0 | 6 | 0 | 5 | 0 | 9 |
| cnv | 0 | 1 | 0 | 1 | 0 | 1 |
| Sbox | 0 | 68 | 0 | 47 | 0 | 137 |
| **Actor Sum** | 56 | 107 | 51 | 91 | 108 | 208 |

CMOS 90 nm technology, Synopsys Design Compiler

| | Area [$\mu m^2$] | Saving % |
|---|---|---|
| UC1 [No MDC] | 3480468 | |
| UC1 [MDC] | 860816 | **75.27** |
| UC2 [No MDC] | 2198694 | |
| UC2 [MDC] | 809705 | **67.17** |
| UC3 [No MDC] | 5673164 | |
| UC3 [MDC] | 1018635 | **82.04** |

# Assessment: Performance Results

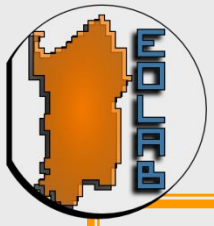
[UC1: *Anti-Aliasing* application, UC2: *Zoom* application, UC3: *Anti-Aliasing* and *Zoom* applications together]

| | UC2 [No MDC] | UC2 [MDC] | UC1 [No MDC] | UC1 [MDC] | UC3 [No MDC] | UC3 [No MDC] |
|---|---|---|---|---|---|---|
| Actor Sum | 50 | 32 | 51 | 38 | 99 | 61 |
| LUT | 0 | 6 | 0 | 5 | 0 | 9 |
| cnv | 0 | 1 | 0 | 1 | 0 | 1 |
| Sbox | 0 | 68 | 0 | 47 | 0 | 137 |
| Actor Sum | 56 | 107 | 51 | 91 | 108 | 208 |

CMOS 90 nm technology, Synopsys Design Compiler

| | Area [$\mu m^2$] | Saving % |
|---|---|---|
| UC1 [No MDC] | 3480468 | |
| UC1 [MDC] | 860816 | 75.27 |
| UC2 [No MDC] | 2198694 | |
| UC2 [MDC] | 809705 | 67.17 |
| UC3 [No MDC] | 5673164 | |
| UC3 [MDC] | 1018635 | 82.04 |

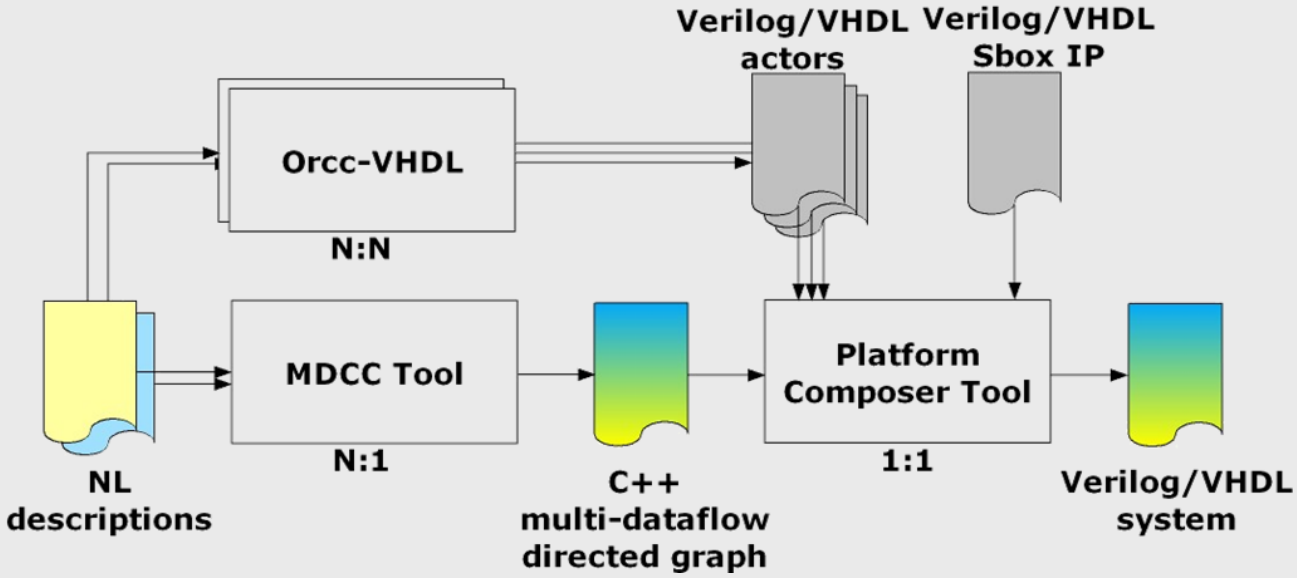
The more are the provided inputs, the more could be the actors overlap and thus the area saving.



# Outline

- Introduction:
  - Problem formulation,
  - Background
  - Goals
- The Multi-Dataflow Composer tool
- Performance assessment
  - Use-case scenario
  - Results
- Future research directions and conclusions
  - RVC extension
  - Applicable research hot topics
  - Final remarks

# EOLAB/INSA Cooperation: RVC Extension




- The MDC tool is a N:1 platform builder. Orcc-VHDL is a 1:1 high level hardware compiler. Therefore, they can be integrated to compose a ***complete multi-purpose systems generation and composition framework***.
- In the RVC domain, this integration will allow the creation of multi-standard codec platforms.

# Power Management and Profiler

- Power Management: trough the Sbox we foresee the possibility of switching off large portion of the substrate belonging to currently unused dataflows.

- Complexity Management: the MDC tool could be coupled with a high level profiler to allow moving additional steps toward the hw-sw gap closure. Such a profiler operating at the graph level, combining lower level back-annotated information and higher level functional information, will be able for example to provide important directives for reconfiguration.

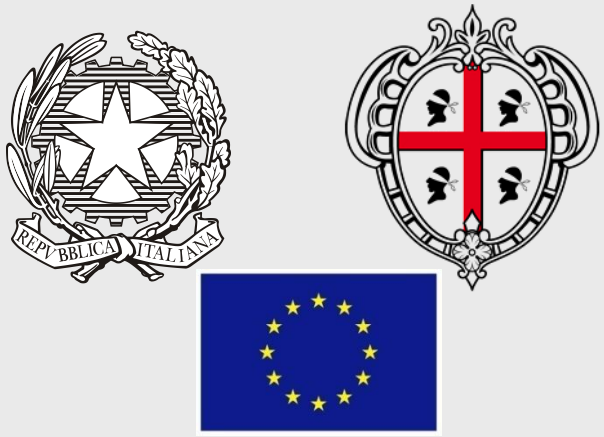
# Final Remarks

- The Multi-Dataflow Composer tool is intended to close the gap between complex multi-purpose heterogeneous hw platforms and their sw programming:
  - Leveraging on the combination of the Dataflow Model of Computation and the coarse-grained reconfigurable paradigm, it builds runtime reconfigurable multi-purpose systems, starting from the high level dataflow descriptions of the applications.
- Benefits:
  - Automatic derivation of complex hw platforms, with a very small users intervention.
  - Possibility of addressing any multi-purpose system, if described according to the RVC formalism.
  - Runtime reconfigurability is provided without neither hw shut-down nor suspension.
  - Concrete on-chip area saving.

# Acknowledgements

The research leading to these results has received funding from:


the European Community's Seventh Framework Programme (FP7/2007-2013) under grant agreement no. 248424, MADNESS Project




the Region of Sardinia, Young Researchers Grant, PO Sardegna FSE 2007-2013, L.R.7/2007 “Promotion of the scientific research and technological innovation in Sardinia” under grant agreement CRP-18324 RPCT Project

Conference on Design and Architectures for Signal and Image Processing -2011
Electronic Chips & Systems design Initiative
November 2nd-4th, 2011,
Tampere, Finland

# THE MULTI-DATAFLOW COMPOSER TOOL: A RUNTIME RECONFIGURABLE HDL PLATFORM COMPOSER

*Francesca Palumbo, Ph.D.*
francesca.palumbo@diee.unica.it
EOLAB - Microelectronics Lab
Dept. of Electrical and Electronics Eng.
University of Cagliari (ITALY)