INSA

The Future Directions of Dataflow-Based Reconfigurable Hardware Accelerators

Francesca Palumbo[1], Claudio Rubattu[1,2], **Carlo Sau**[3], Tiziana Fanni[3], Luigi Raffo[3]

[1]University of Sassari, PolComIng – Information Engineering Group

[2]University of Rennes, INSA Group

[3]University of Cagliari, Diee – Microelectronics and Bioengineering Group

Rennes, 12-14 December 2017

Outline

- MDC Tool Summary
  - Motivation and Approach
  - Current Functionalities and Future Directions
- Hardware-Software Partitioning
  - Co-Processing Support and Automated Characterization
- Enhancing the MDC High-Level Synthesis Support
  - Integration with the CAPH HLS engine
- Run-time Monitoring of CGR Accelerators
  - Extension of PAPI for dataflow in CGR hardware
- Providing Further Degrees of Reconfigurability
  - Mixed-Grain Reconfiguration Possibilities

Outline

- MDC Tool Summary
  - Motivation and Approach
  - Current Functionalities and Future Directions
- Hardware-Software Partitioning
  - Co-Processing Support and Automated Characterization
- Enhancing the MDC High-Level Synthesis Support
  - Integration with the CAPH HLS engine
- Run-time Monitoring of CGR Accelerators
  - Extension of PAPI for dataflow in CGR hardware
- Providing Further Degrees of Reconfigurability
  - Mixed-Grain Reconfiguration Possibilities

MDC tool Summary

Motivations

HIGH PERFORMANCES

real time, portability, long battery life

UP-TO-DATE SOLUTIONS

last audio/video codecs, file formats...

MORE INTEGRATED FEATURES

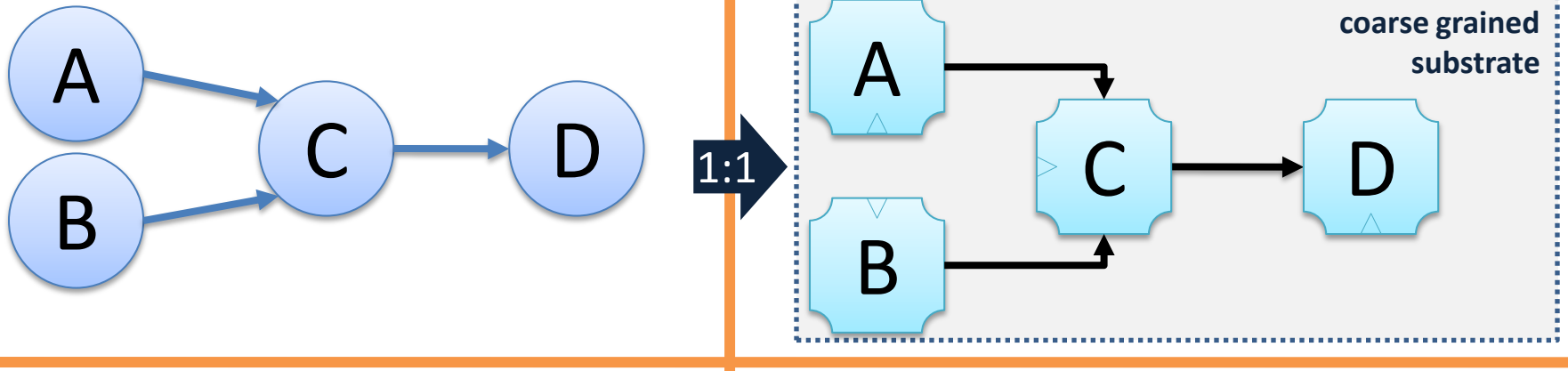
MP3, Camera, Video, GPS...

MARKET DEMAND

convenient form factor, affordable price, fashion

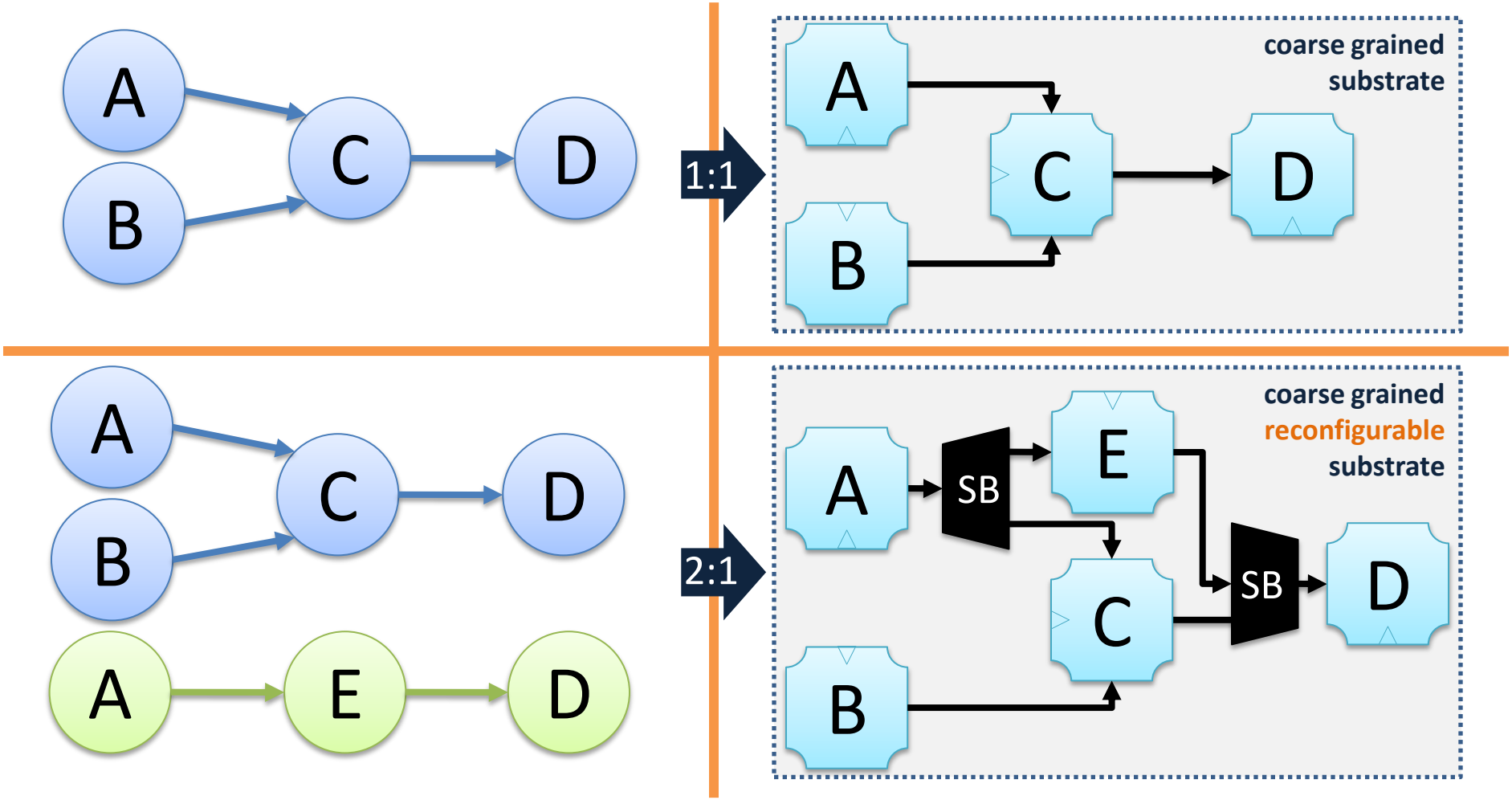
MDC tool Summary

Approach



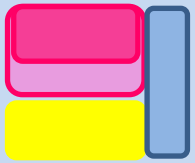
MDC tool Summary

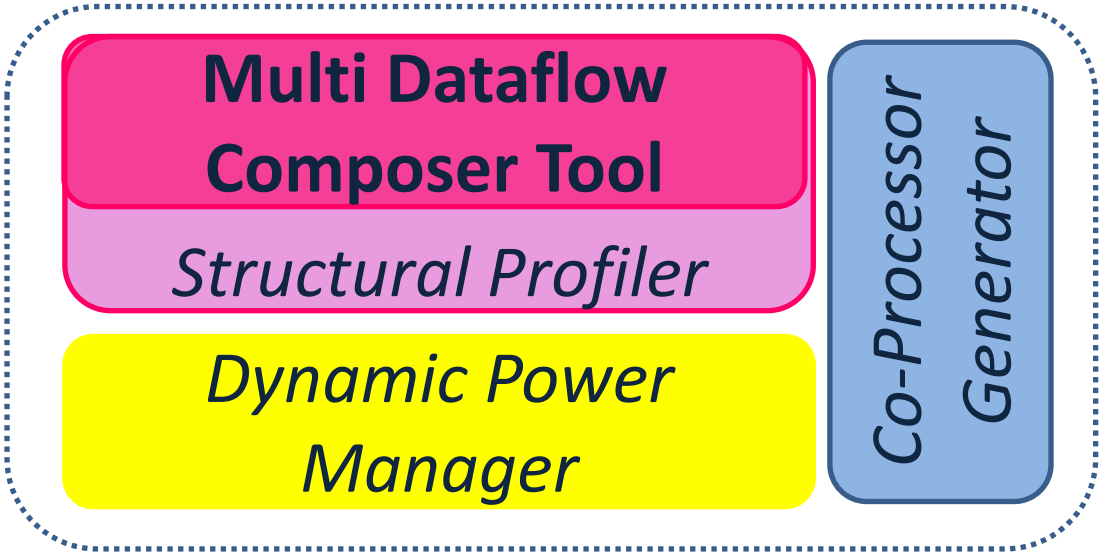
Approach



MDC tool Summary

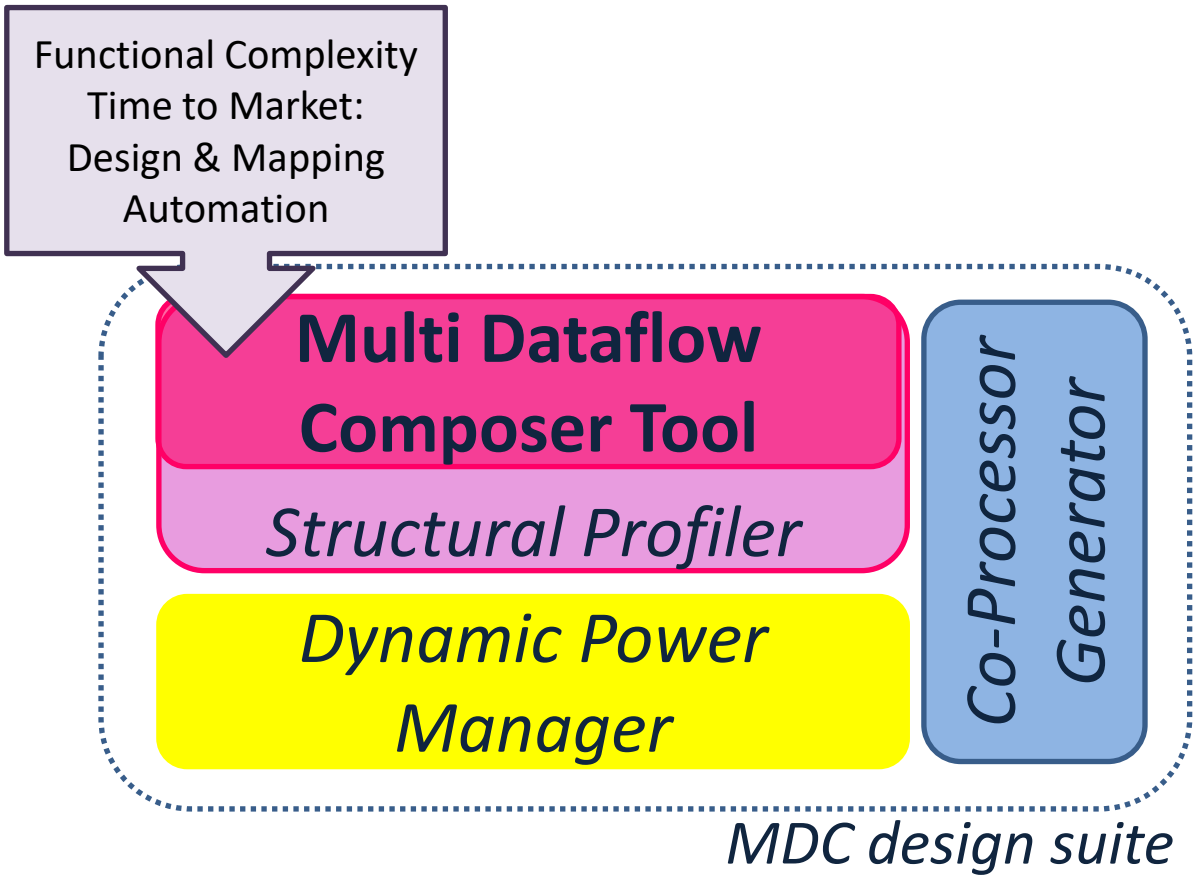
Current Functionalities





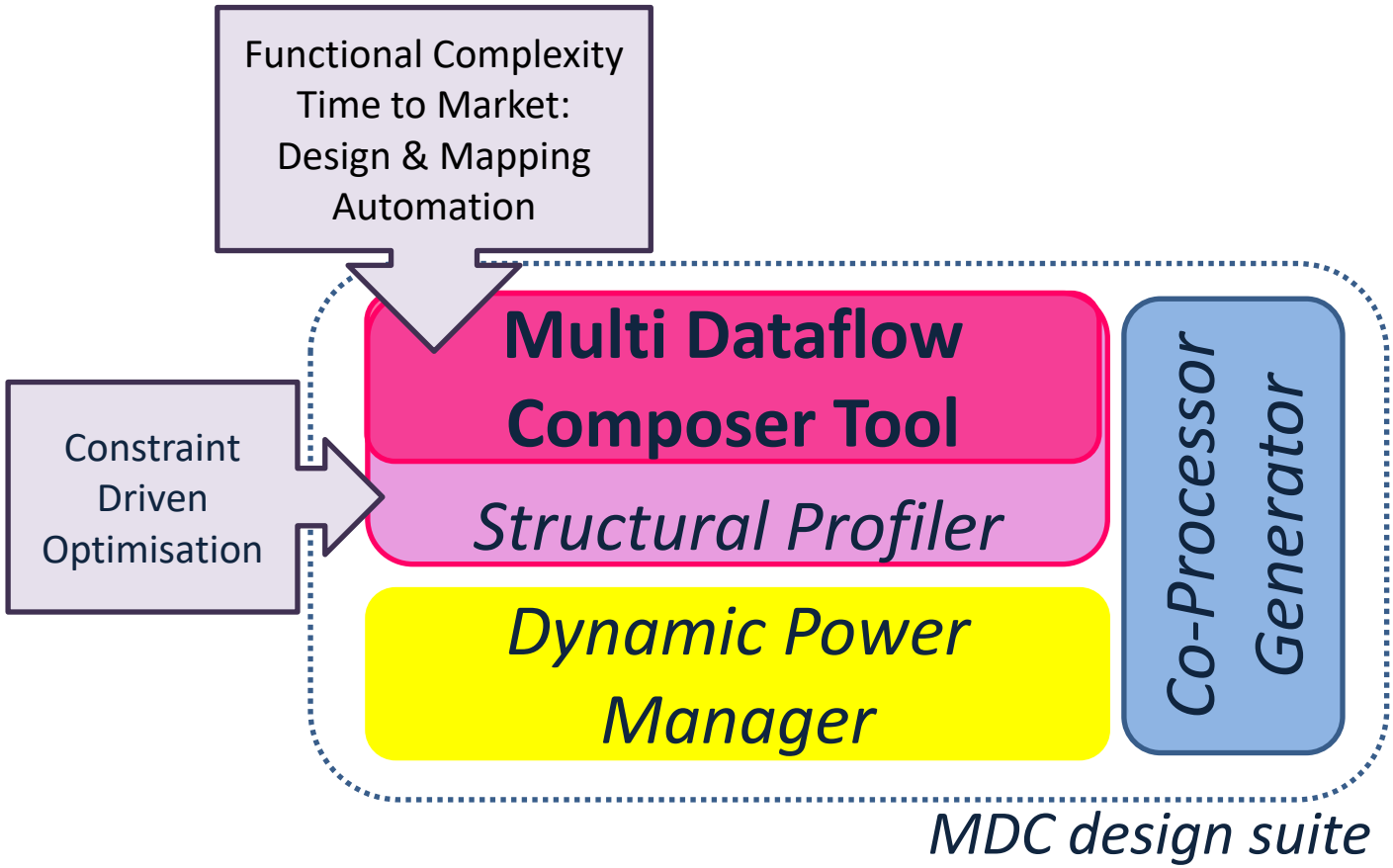
MDC tool Summary

Current Functionalities



MDC tool Summary

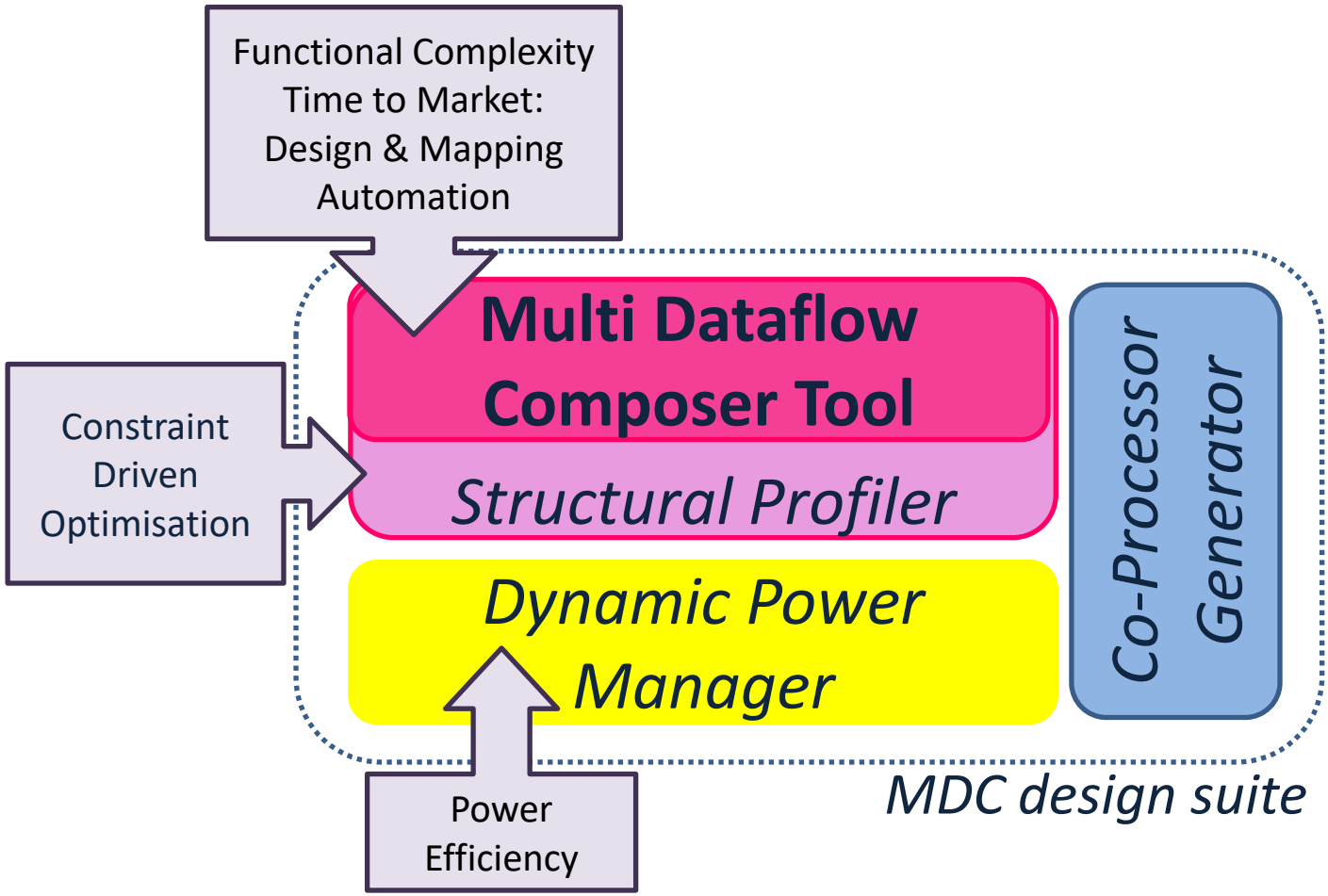
Current Functionalities



http://sites.unica.it/rpct/

MDC tool Summary

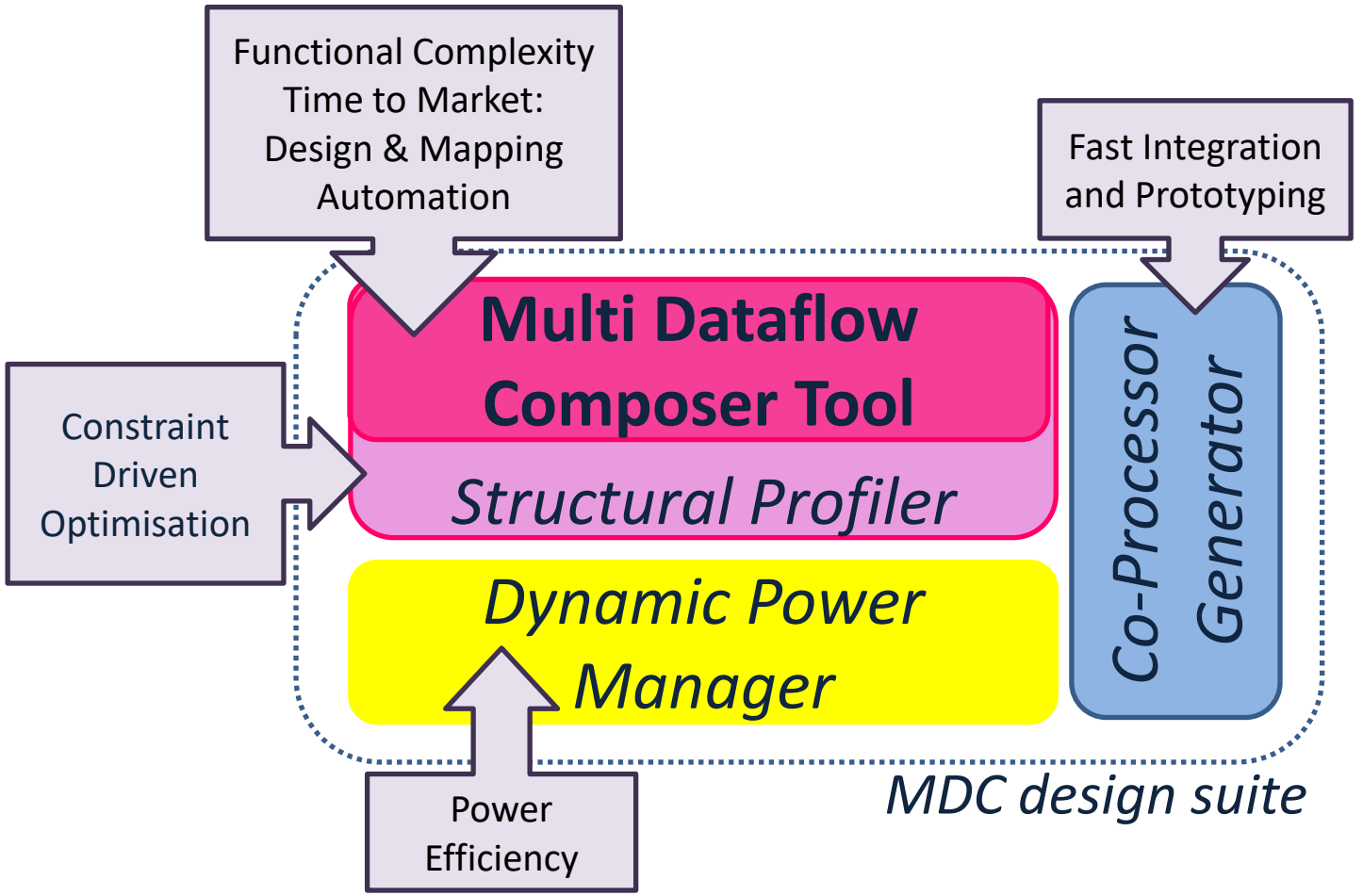
Current Functionalities



http://sites.unica.it/rpct/

MDC tool Summary

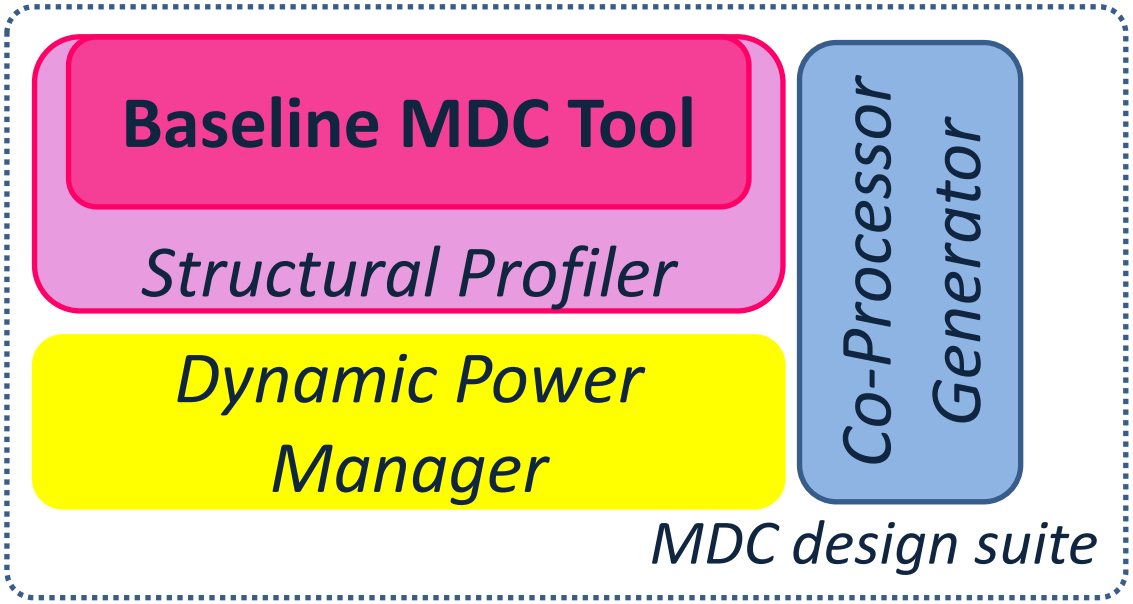
Current Functionalities



http://sites.unica.it/rpct/

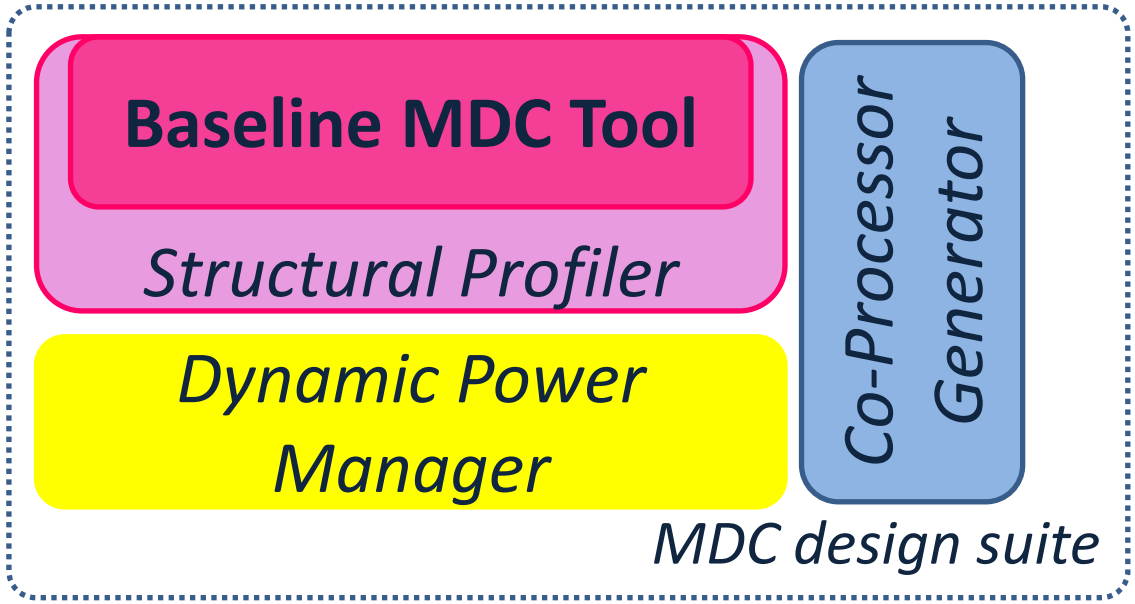
MDC tool Summary:

Future Directions



MDC tool Summary:

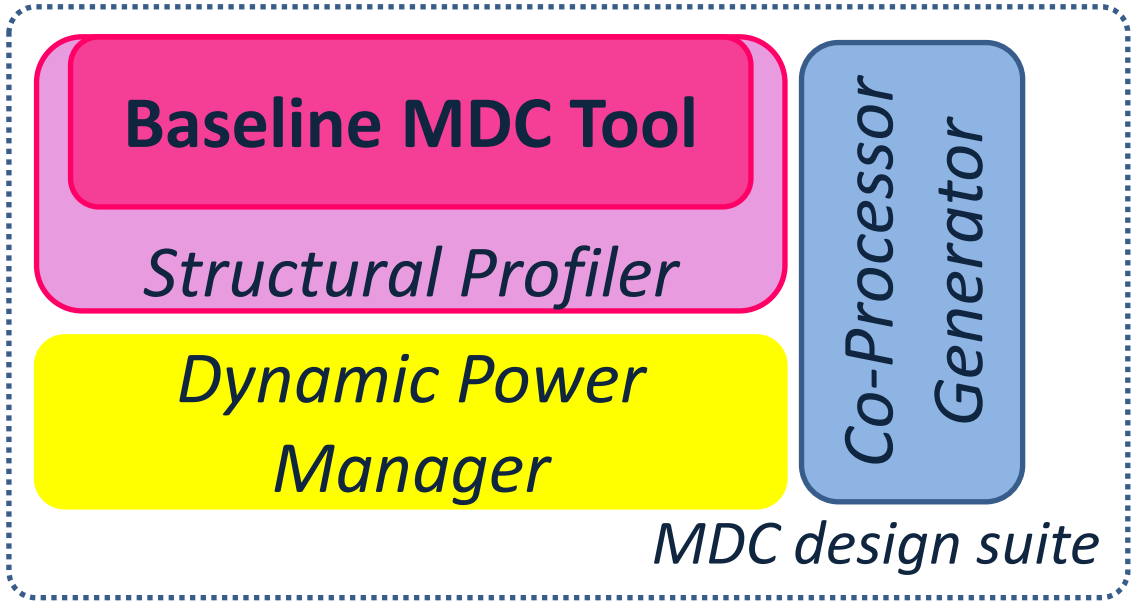
Future Directions



HW/SW Partitioning

MDC tool Summary:

Future Directions

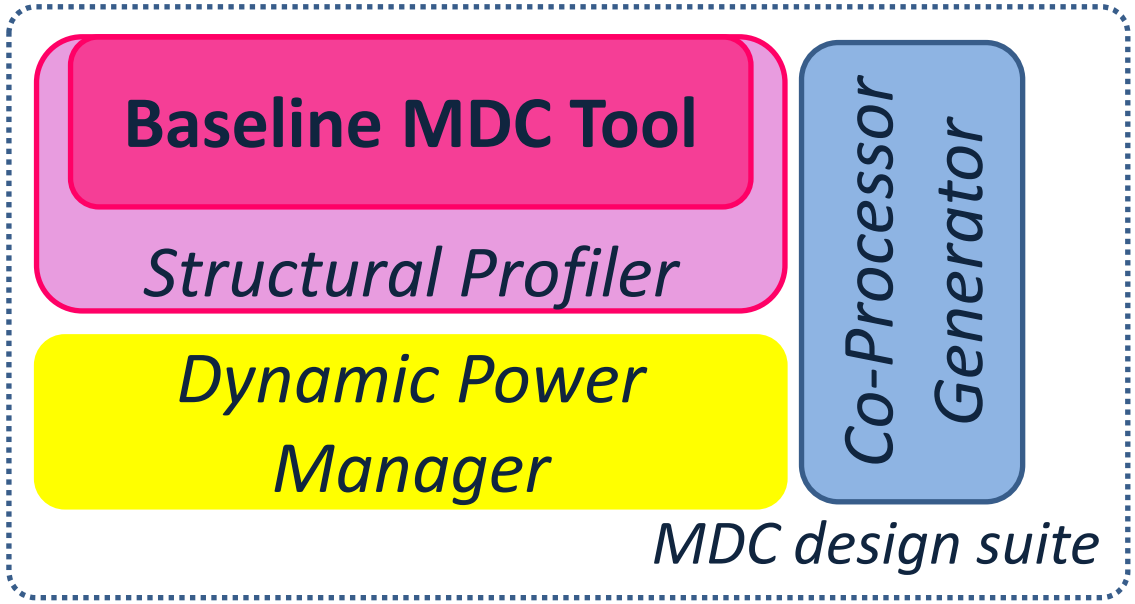


HW/SW Partitioning

Enhancing HLS

MDC tool Summary:

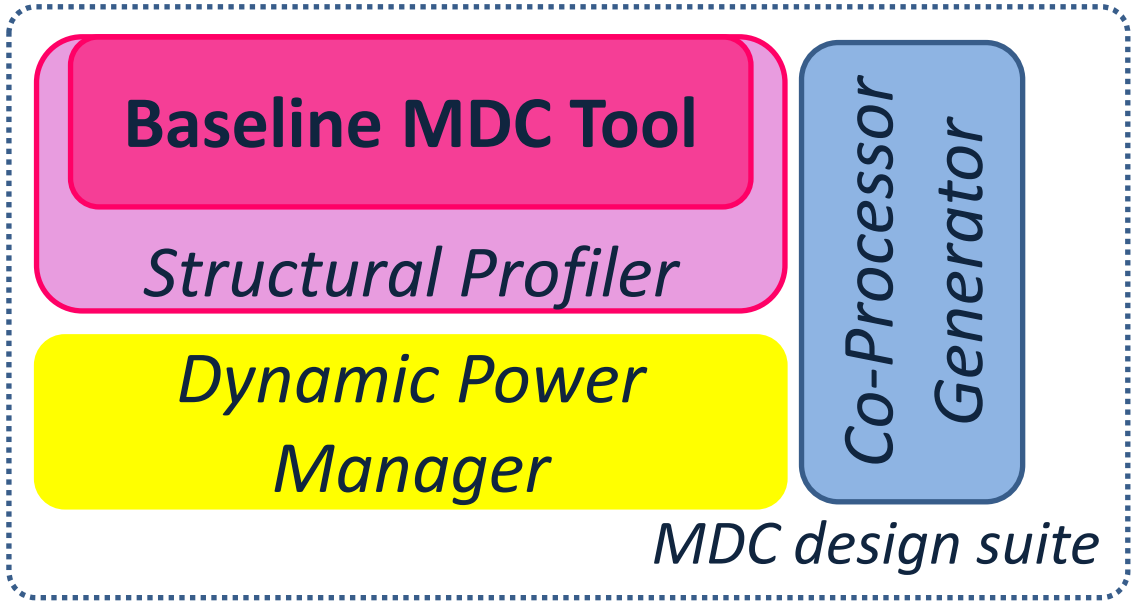
Future Directions



- HW/SW Partitioning
- Enhancing HLS
- Runtime Monitoring

MDC tool Summary:

Future Directions



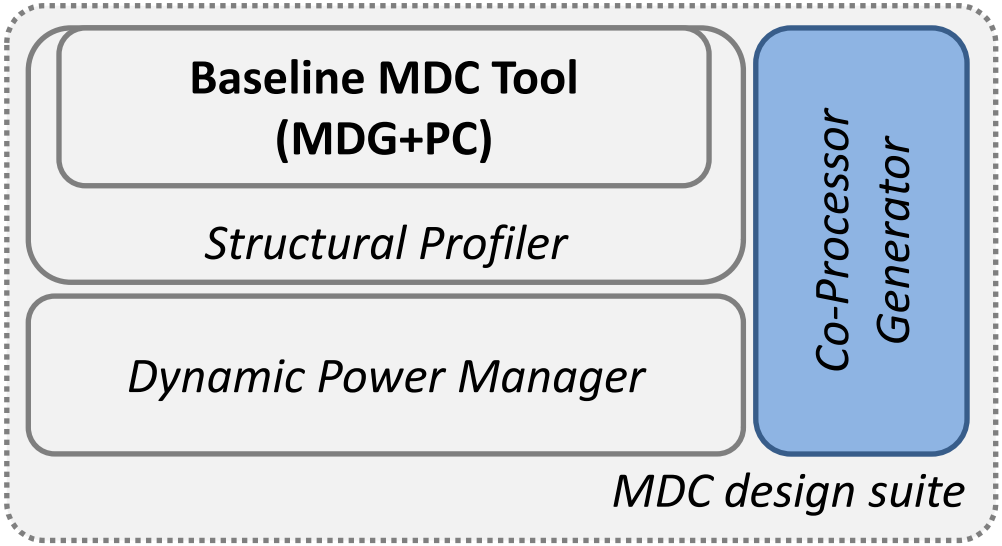
- HW/SW Partitioning
- Enhancing HLS
- Runtime Monitoring
- Reconfiguration Degrees

Outline

- MDC Tool Summary
  - Motivations and Approach
  - Current Functionalities and Future Directions
- Hardware-Software Partitioning
  - Co-Processing Support and Automated Characterization
- Enhancing the MDC High-Level Synthesis Support
  - Integration with the CAPH HLS engine
- Run-time Monitoring of CGR Accelerators
  - Extension of PAPI for dataflow in CGR hardware
- Providing Further Degrees of Reconfigurability
  - Mixed-Grain Reconfiguration Possibilities

Hardware-Software Partitioning

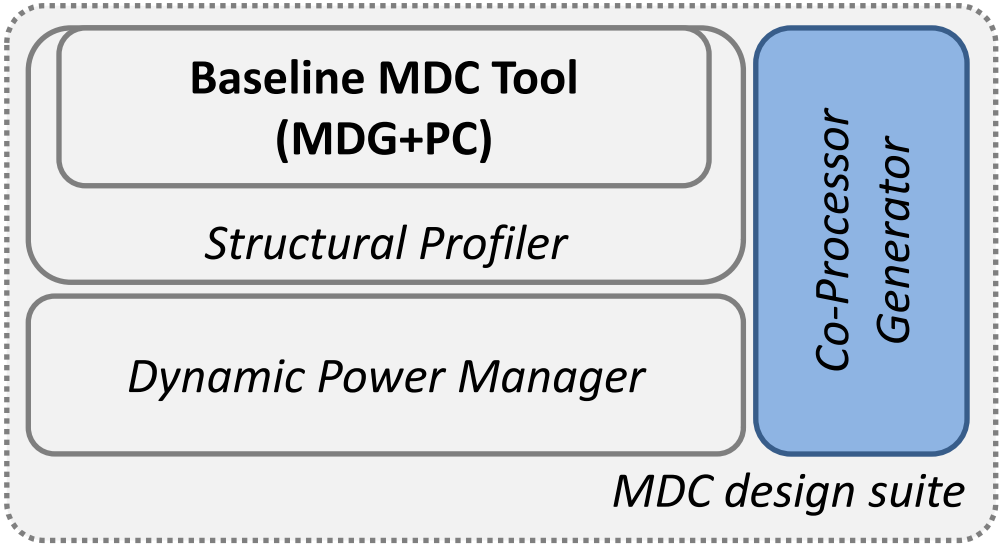
Co-Processing Support



MDC is a dataflow-based design suite for the development of coarse-grained reconfigurable systems with the capability of generating co-processing units.

Hardware-Software Partitioning

Co-Processing Support

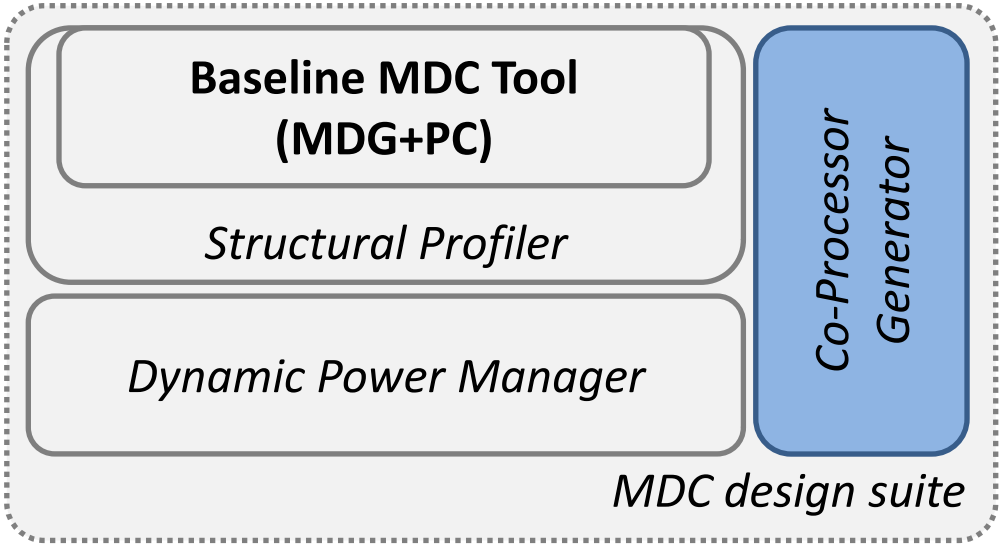


MDC is a dataflow-based design suite for the development of coarse-grained reconfigurable systems with the capability of generating co-processing units.

- MDC assembles **ready-to-use platform-dependent IPs**

Hardware-Software Partitioning

Co-Processing Support



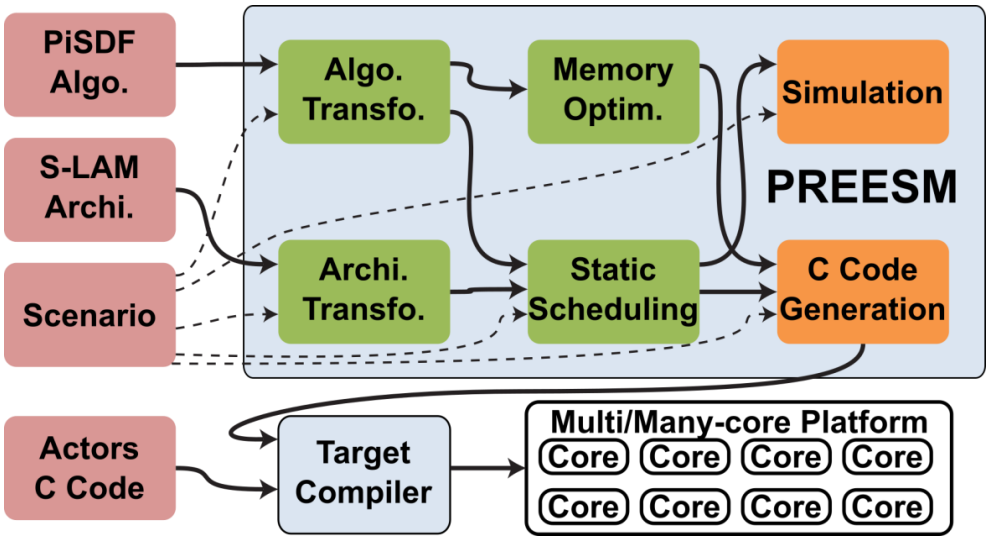
MDC is a dataflow-based design suite for the development of coarse-grained reconfigurable systems with the capability of generating co-processing units.

- MDC assembles **ready-to-use platform-dependent IPs**

- Designer can choose to opt for **memory-mapped** or **stream-based** coupling.

Hardware-Software Partitioning

Automated Characterization

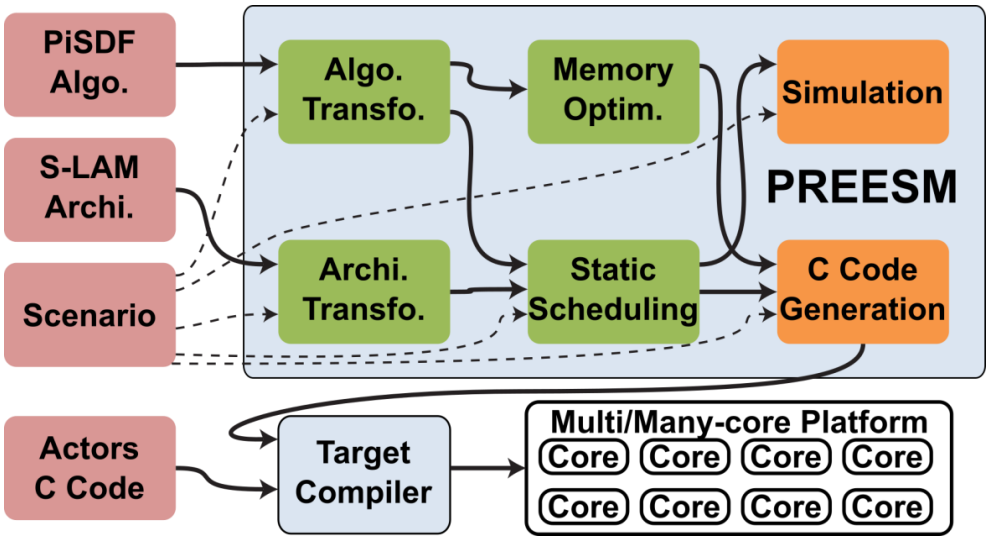


PREESM is rapid prototyping tool that generates code for heterogeneous multi/many-core embedded systems.

It provides mapping of actors to multiple processing cores, optimizing execution latency and balancing loads.

Hardware-Software Partitioning

Automated Characterization



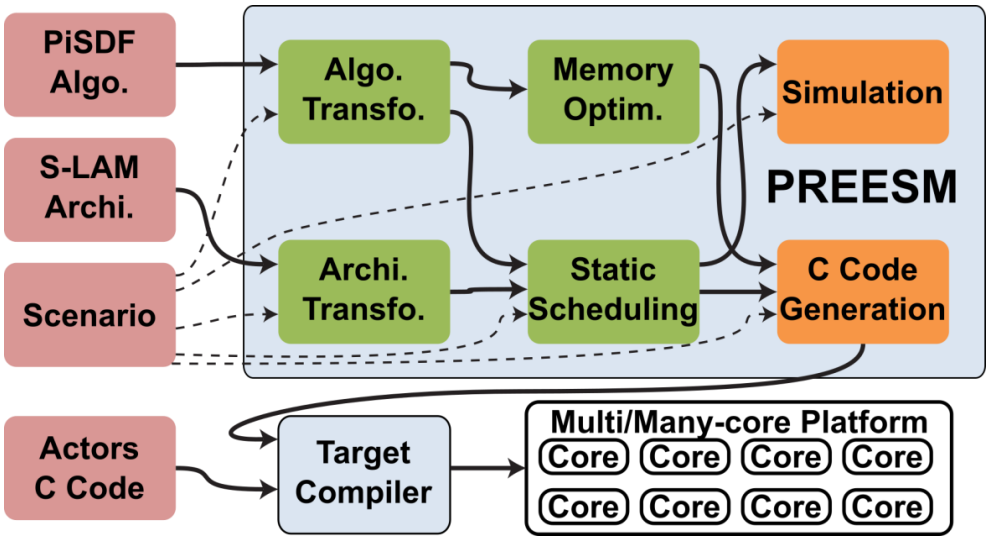
PREESM is rapid prototyping tool that generates code for heterogeneous multi/many-core embedded systems.

It provides mapping of actors to multiple processing cores, optimizing execution latency and balancing loads.

- Model the **costs** of the available **communication schemes and co-processing units**

Hardware-Software Partitioning

Automated Characterization



PREESM is rapid prototyping tool that generates code for heterogeneous multi/many-core embedded systems.

It provides mapping of actors to multiple processing cores, optimizing execution latency and balancing loads.

- Model the **costs** of the available **communication schemes and co-processing units**

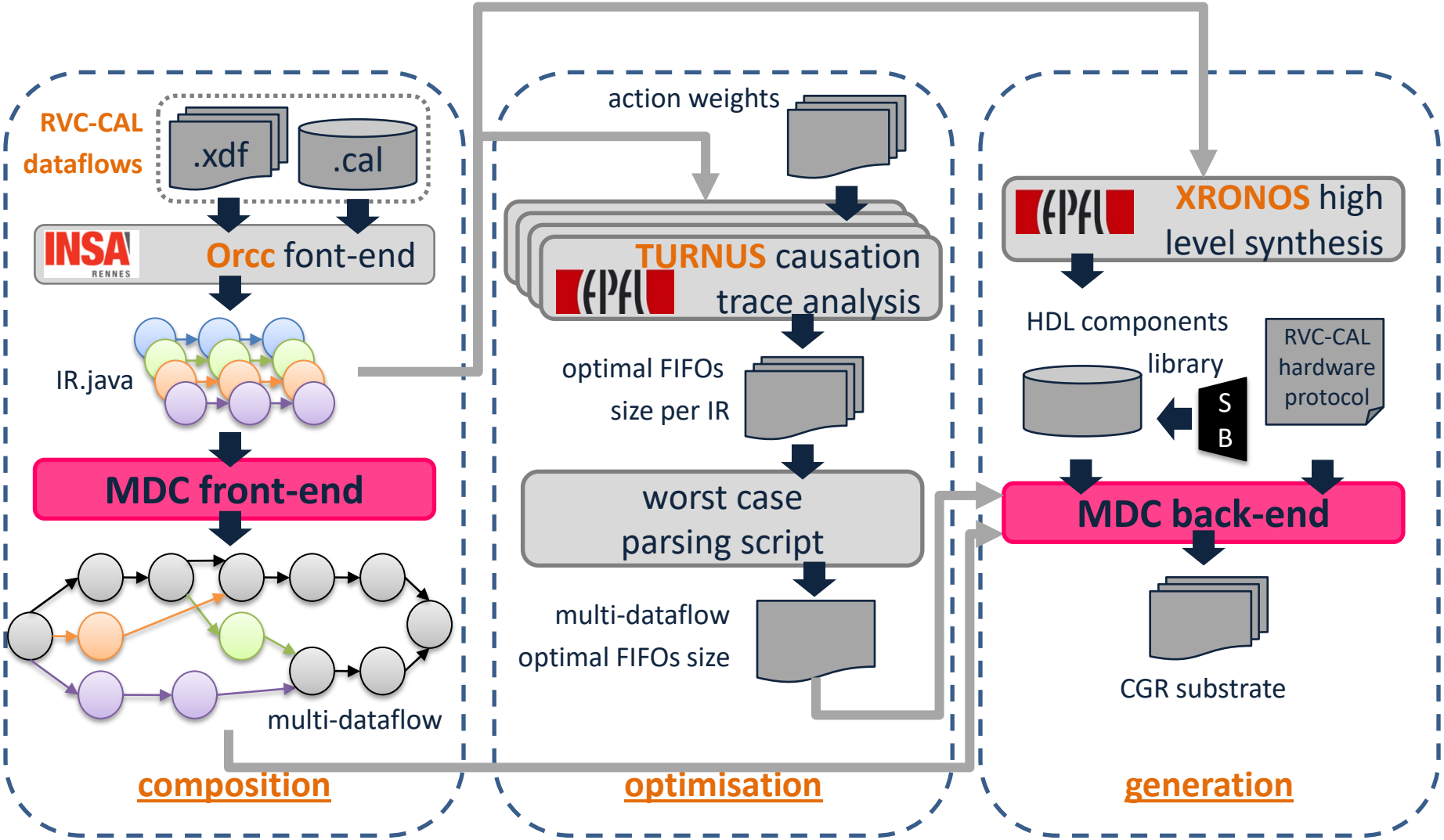
- **Connect PREESM and MDC** to delegate specific computations (an actor, a network of actors or a set of networks) to the most suitable co-processing units

Outline

- MDC Tool Summary
  - Approach
  - Baseline Functionality and Extensions
- Hardware-Software Partitioning
  - Co-Processing Support and Automated Characterization
- Enhancing the MDC High-Level Synthesis Support
  - Integration with the CAPH HLS engine
- Run-time Monitoring of CGR Accelerators
  - Extension of PAPI for dataflow in CGR hardware
- Providing Further Degrees of Reconfigurability
  - Mixed-Grain Reconfiguration Possibilities

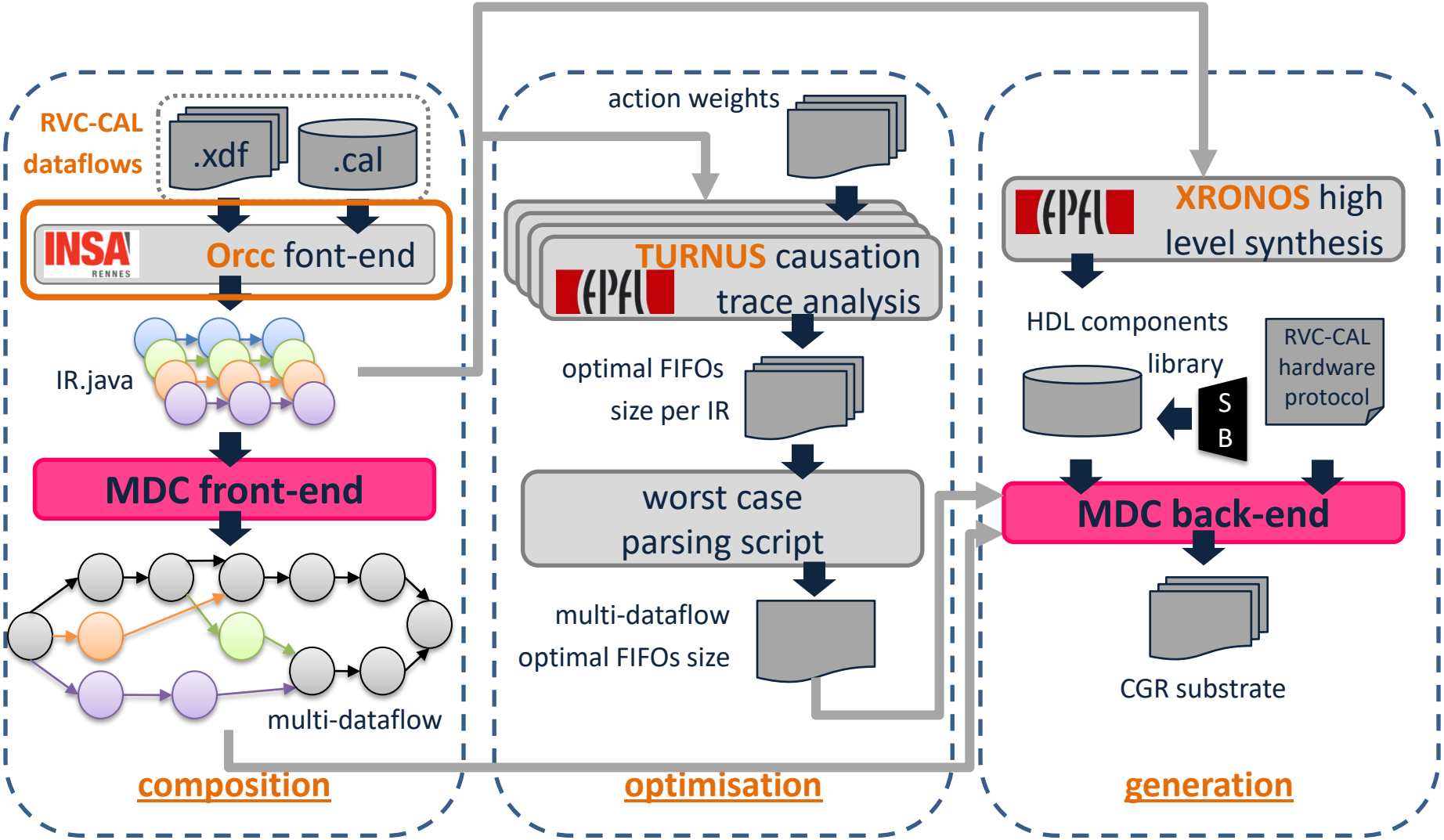
Enhancing MDC High-Level Synthesis Support

Previous Fully Automated Flow



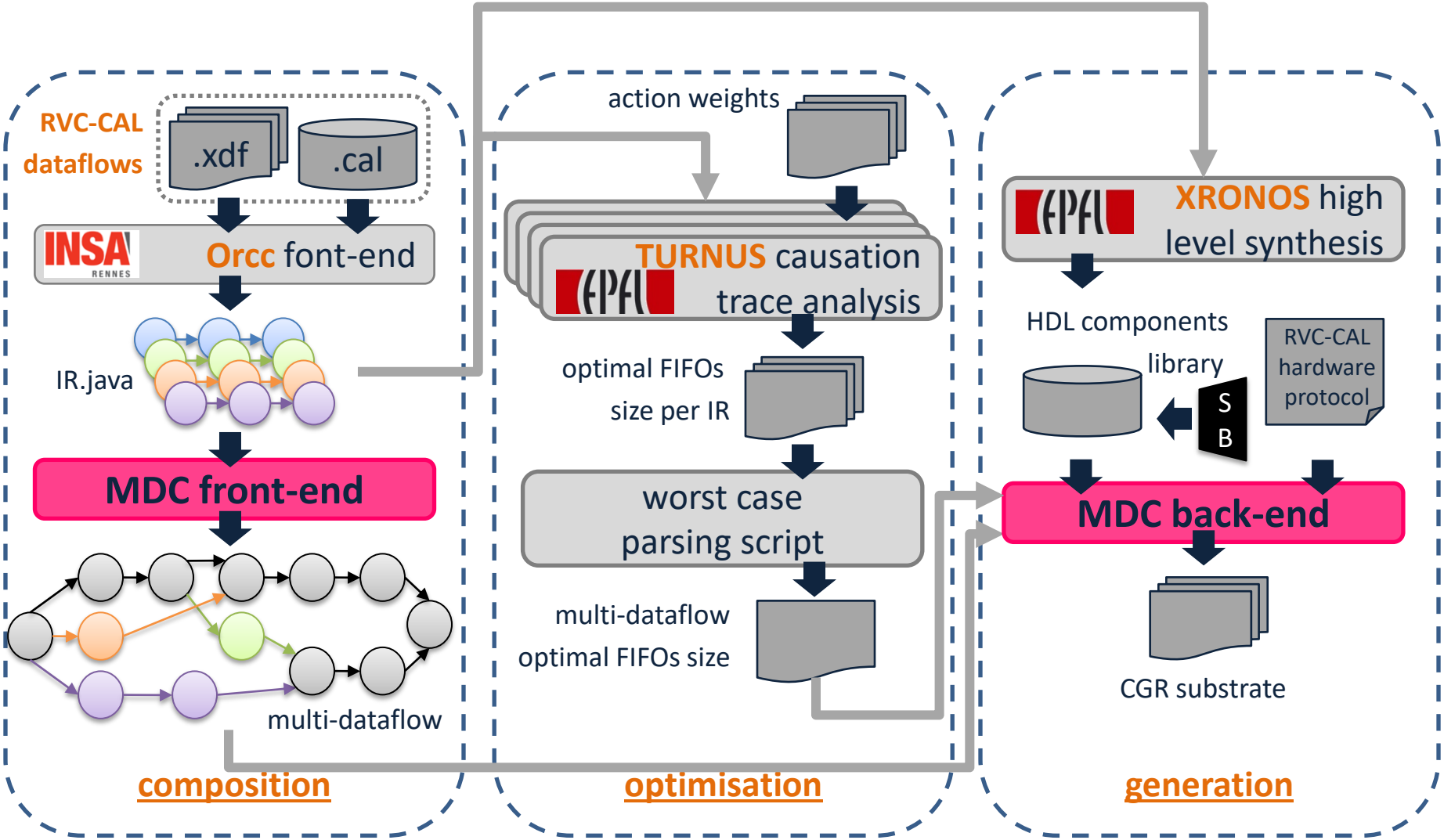
Enhancing MDC High-Level Synthesis Support

Previous Fully Automated Flow



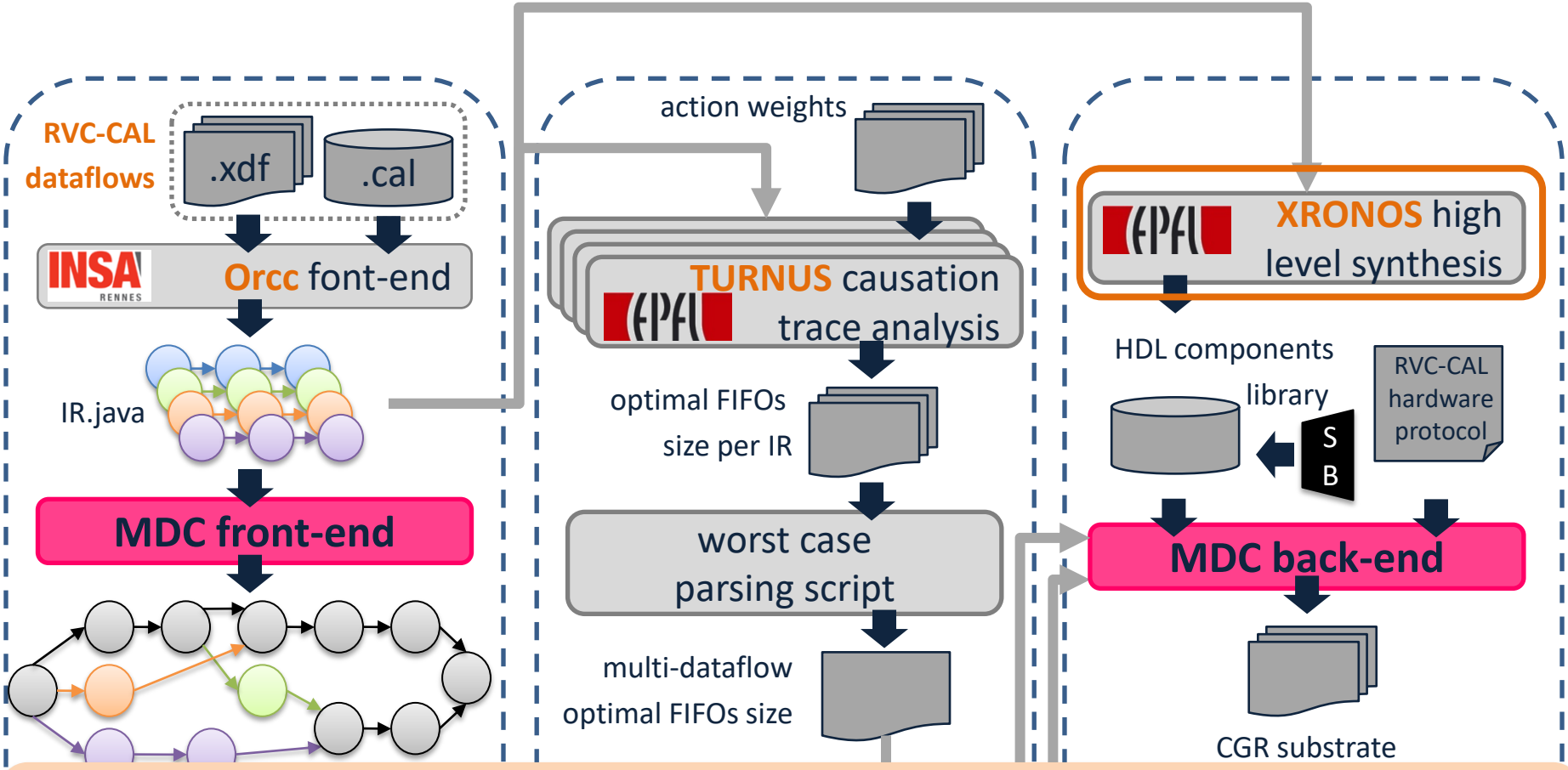
Enhancing MDC High-Level Synthesis Support

Previous Fully Automated Flow



Enhancing MDC High-Level Synthesis Support

Previous Fully Automated Flow



- High-Level Synthesis supports **only FPGAs from one specific FPGA vendor (Xilinx)**

Enhancing MDC High-Level Synthesis Support

CAPH

- CAPH is a domain-specific language for describing and implementing stream-processing applications.

INSTITUT PASCAL
sciences de l'ingénierie et des systèmes

CAPH SOURCE CODE

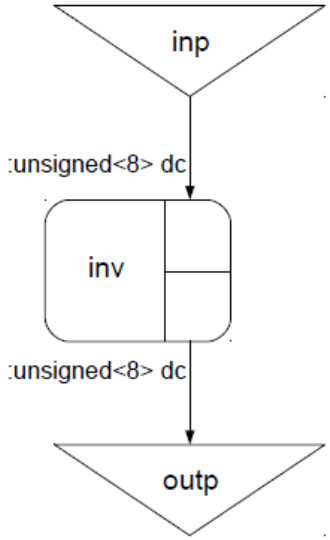
```
-- Global values declarations (libraries, variables, constants, functions, types)
#include "dc.cph"

-- Actor declarations
actor inv
in ( i : unsigned<8> dc )
out ( o : unsigned<8> dc )
rules
| i:'< -> o:'<
| i:'> -> o:'>
| i:' x -> o:'255-x ;

-- I/O declarations
stream inp: unsigned<8> dc from "sample.txt";
stream outp: unsigned<8> dc to "result.txt";

-- Network declarations
net outp = inv inp;
```

CAPH GRAPH



Enhancing MDC High-Level Synthesis Support

CAPH

- CAPH is a domain-specific language for describing and implementing stream-processing applications.

INSTITUT PASCAL
sciences de l'ingénierie et des systèmes

CAPH SOURCE CODE

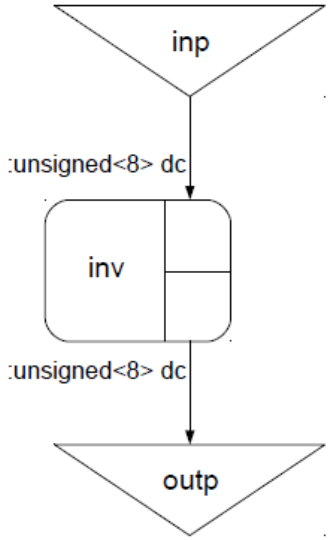
```
-- Global values declarations (libraries, variables, constants, functions, types)
#include "dc.cph"

-- Actor declarations
actor inv
in ( i : unsigned<8> dc )
out ( o : unsigned<8> dc )
rules
| i:'< -> o:'<
| i:'> -> o:'>
| i:' x -> o:'255-x ;

-- I/O declarations
stream inp: unsigned<8> dc from "sample.txt";
stream outp: unsigned<8> dc to "result.txt";

-- Network declarations
net outp = inv inp;
```

CAPH GRAPH



- It relies upon the **actor/dataflow model of computation**

Enhancing MDC High-Level Synthesis Support

CAPH

- CAPH is a domain-specific language for describing and implementing stream-processing applications.

INSTITUT PASCAL
sciences de l'ingénierie et des systèmes

CAPH SOURCE CODE

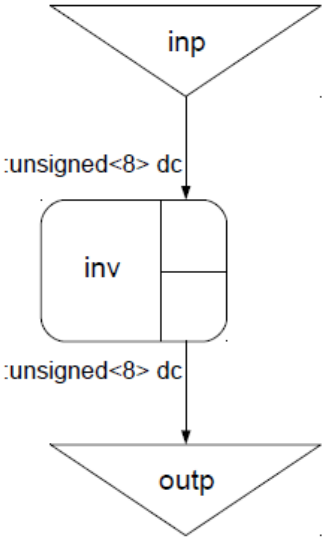
```
-- Global values declarations (libraries, variables, constants, functions, types)
#include "dc.cph"

-- Actor declarations
actor inv
in ( i : unsigned<8> dc )
out ( o : unsigned<8> dc )
rules
| i:'< -> o:'<
| i:'> -> o:'>
| i:' x -> o:'255-x ;

-- I/O declarations
stream inp: unsigned<8> dc from "sample.txt";
stream outp: unsigned<8> dc to "result.txt";

-- Network declarations
net outp = inv inp;
```

CAPH GRAPH



- It relies upon the **actor/dataflow model of computation**
- It is capable of **generating VHDL code**

Enhancing MDC High-Level Synthesis Support

CAPH

- CAPH is a domain-specific language for describing and implementing stream-processing applications.

INSTITUT PASCAL
sciences de l'ingénierie et des systèmes

CAPH SOURCE CODE

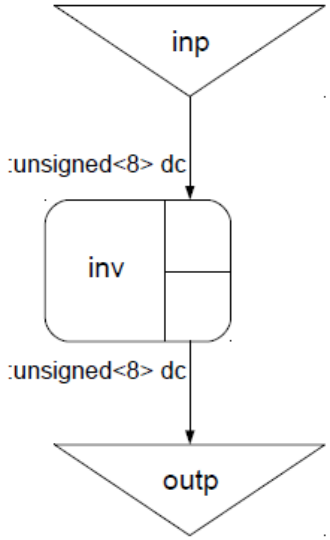
```
-- Global values declarations (libraries, variables, constants, functions, types)
#include "dc.cph"

-- Actor declarations
actor inv
in ( i : unsigned<8> dc )
out ( o : unsigned<8> dc )
rules
| i:'< -> o:'<
| i:'> -> o:'>
| i:' x -> o:'255-x ;

-- I/O declarations
stream inp: unsigned<8> dc from "sample.txt";
stream outp: unsigned<8> dc to "result.txt";

-- Network declarations
net outp = inv inp;
```

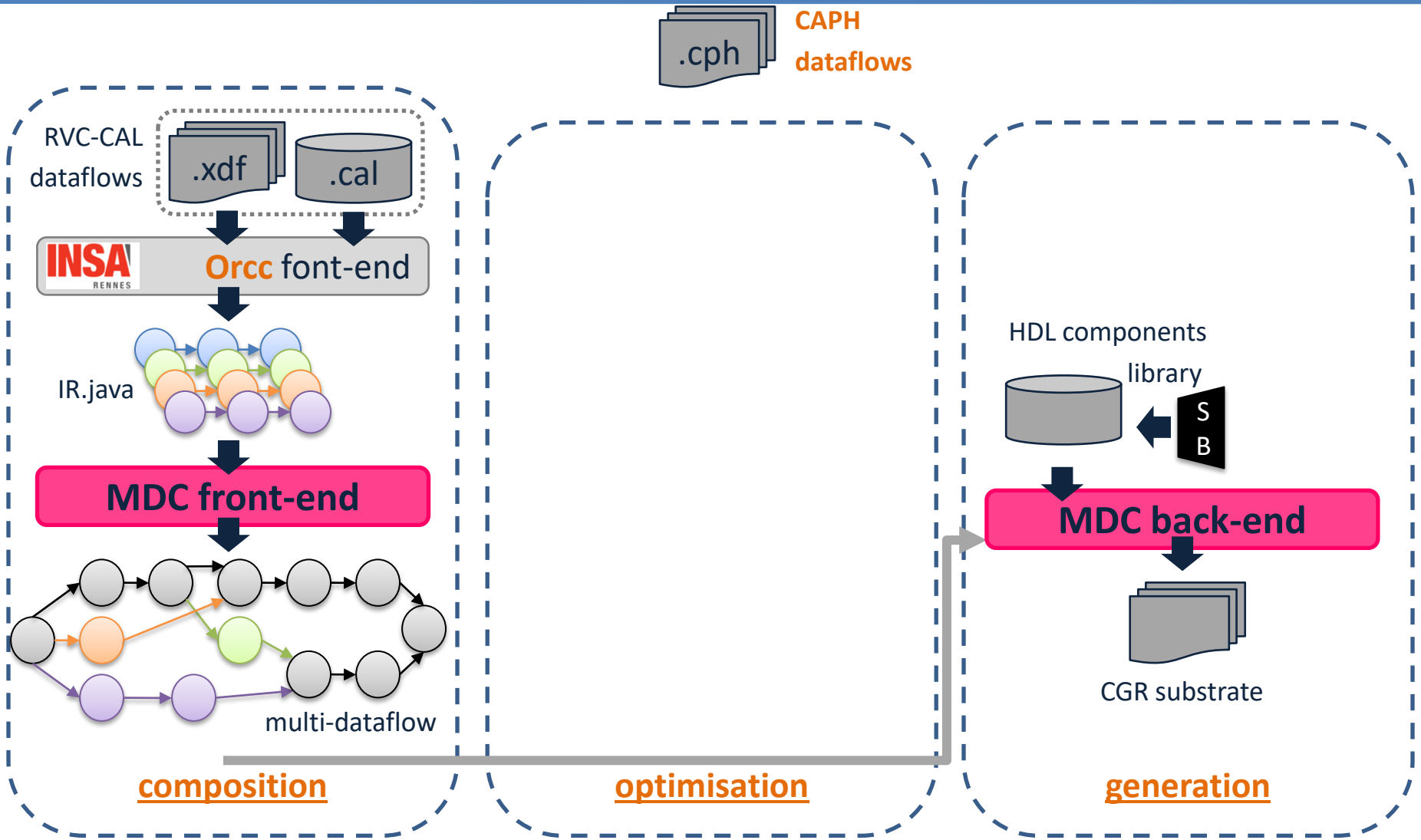
CAPH GRAPH



- It relies upon the **actor/dataflow model of computation**
- It is capable of **generating VHDL code**
- It is **platform agnostic**

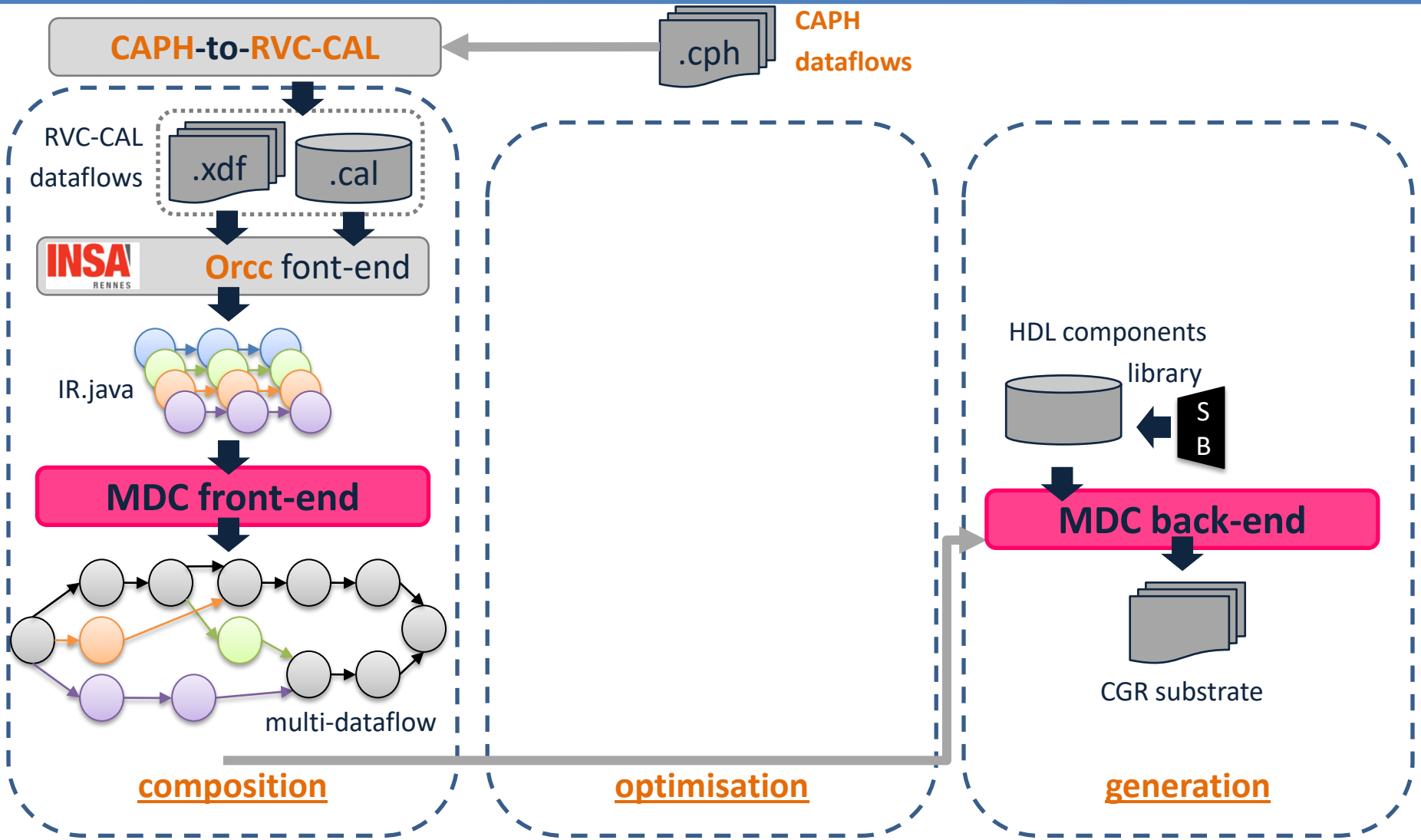
Enhancing MDC High-Level Synthesis Support

Fully Automated Flow



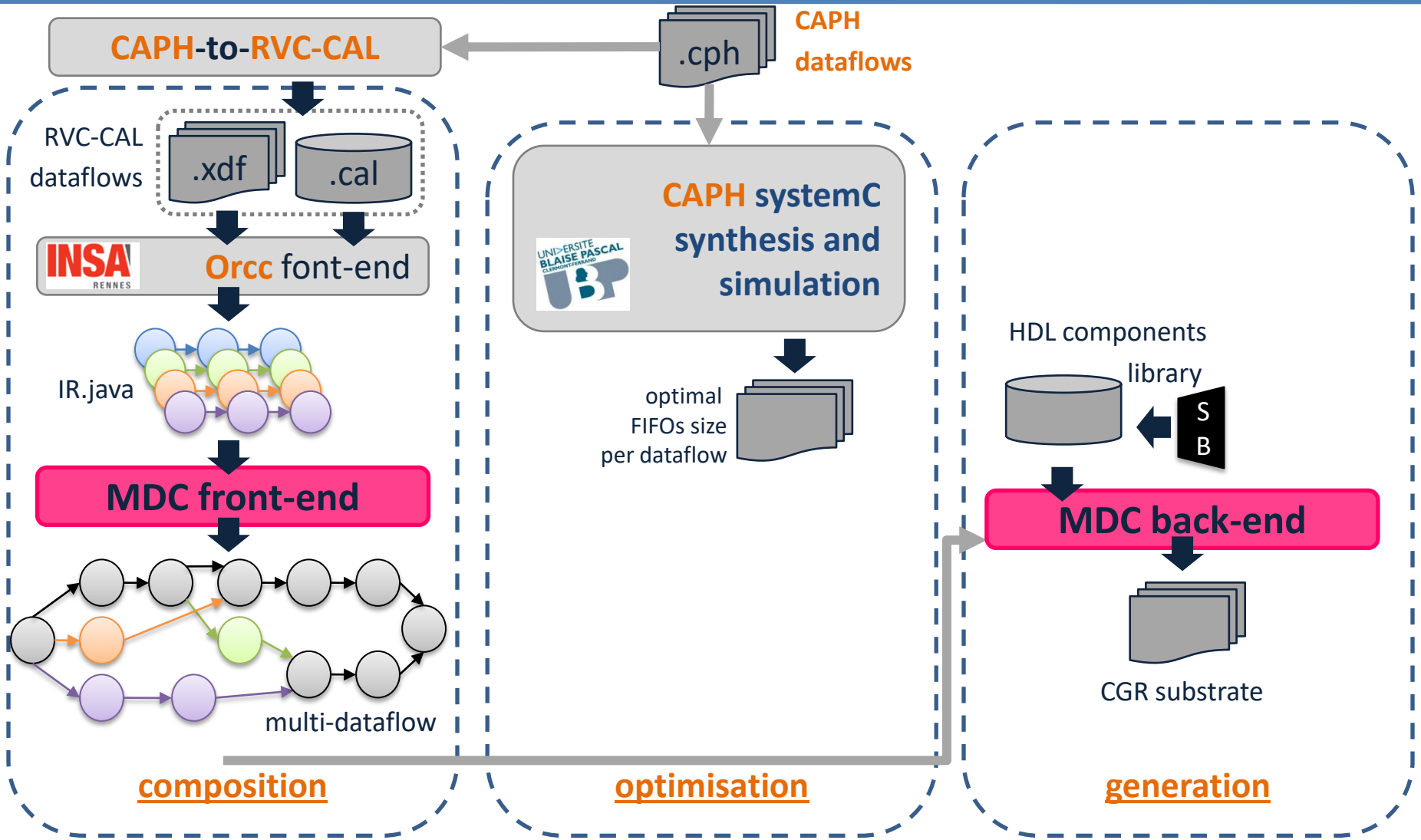
Enhancing MDC High-Level Synthesis Support

Fully Automated Flow



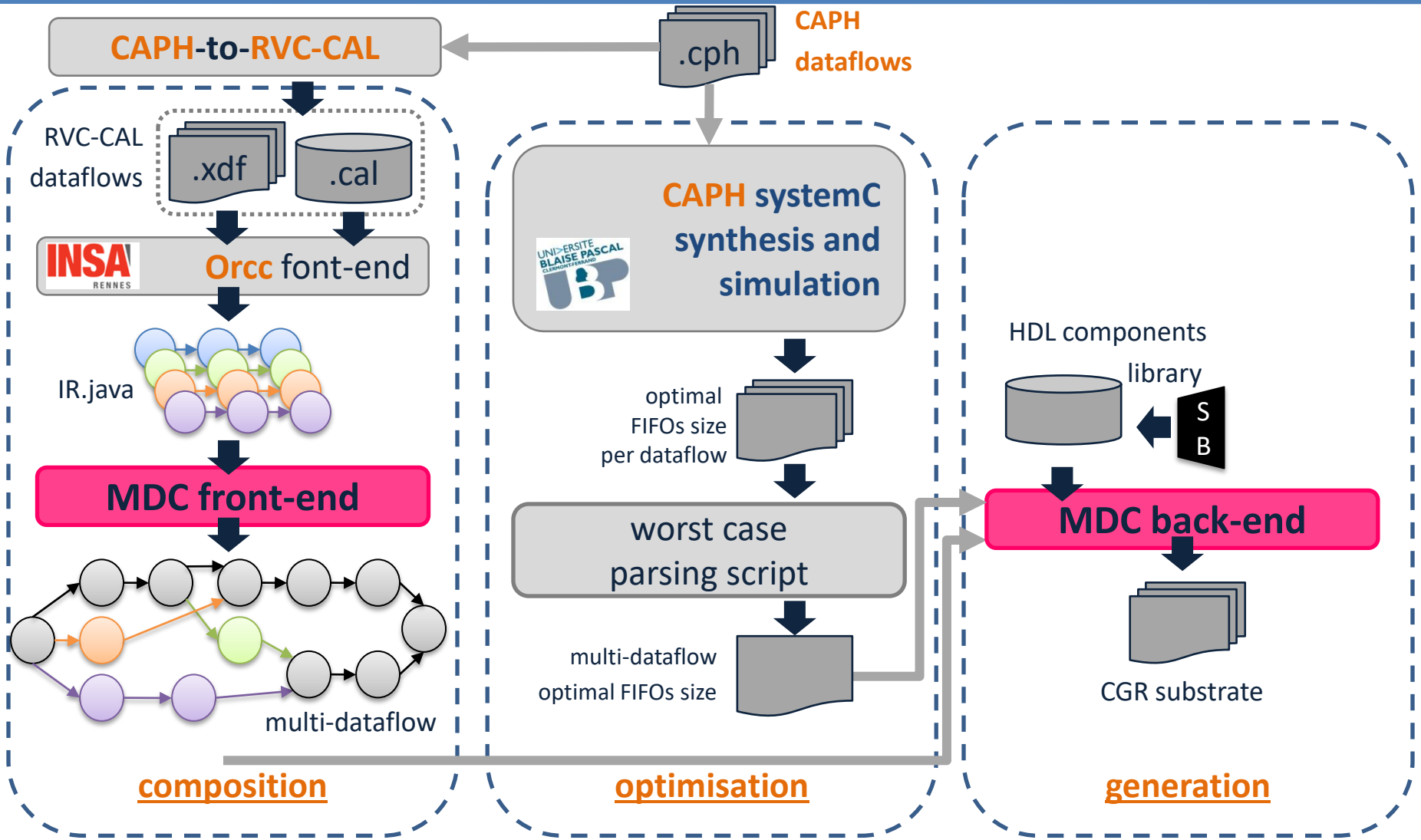
Enhancing MDC High-Level Synthesis Support

Fully Automated Flow



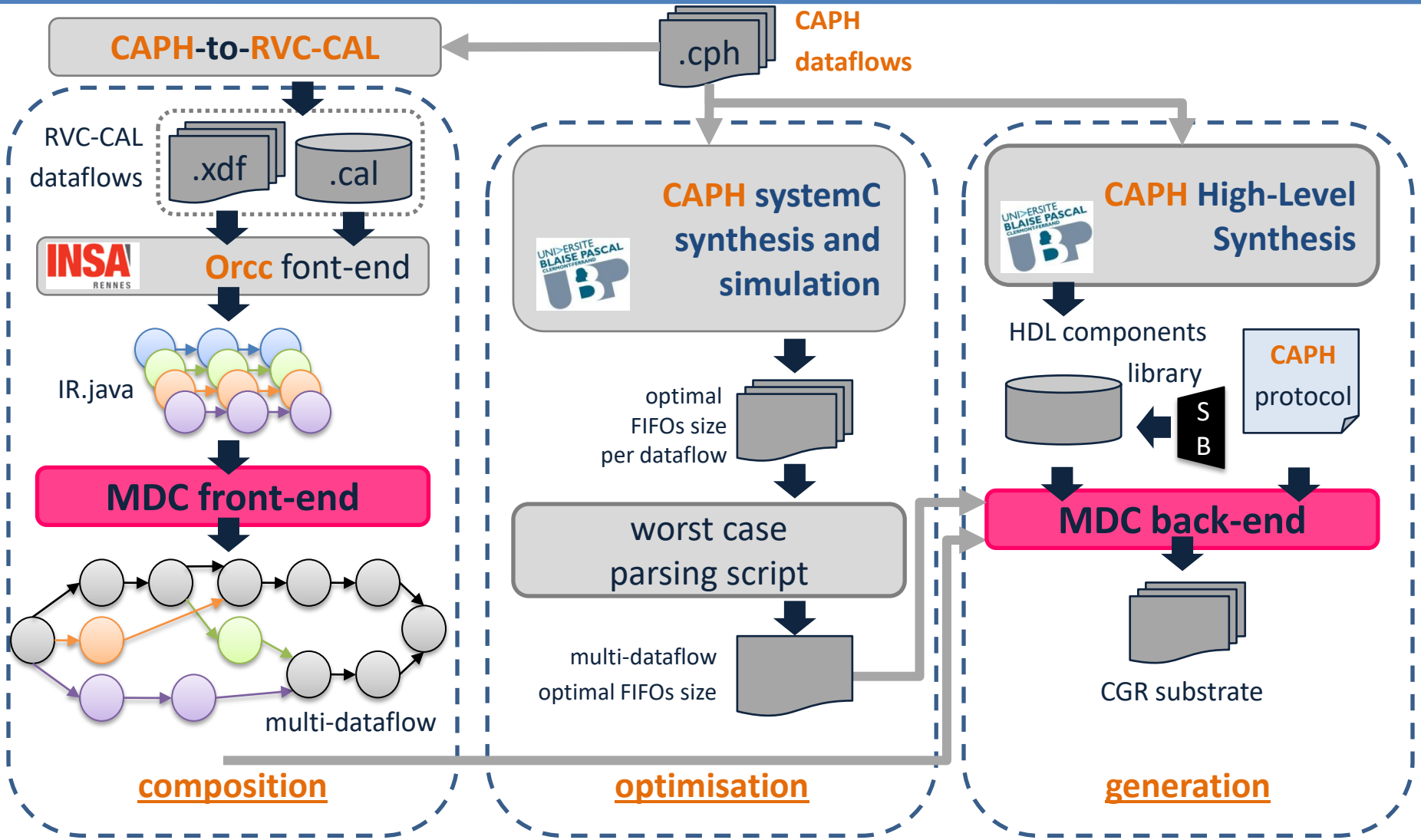
Enhancing MDC High-Level Synthesis Support

Fully Automated Flow



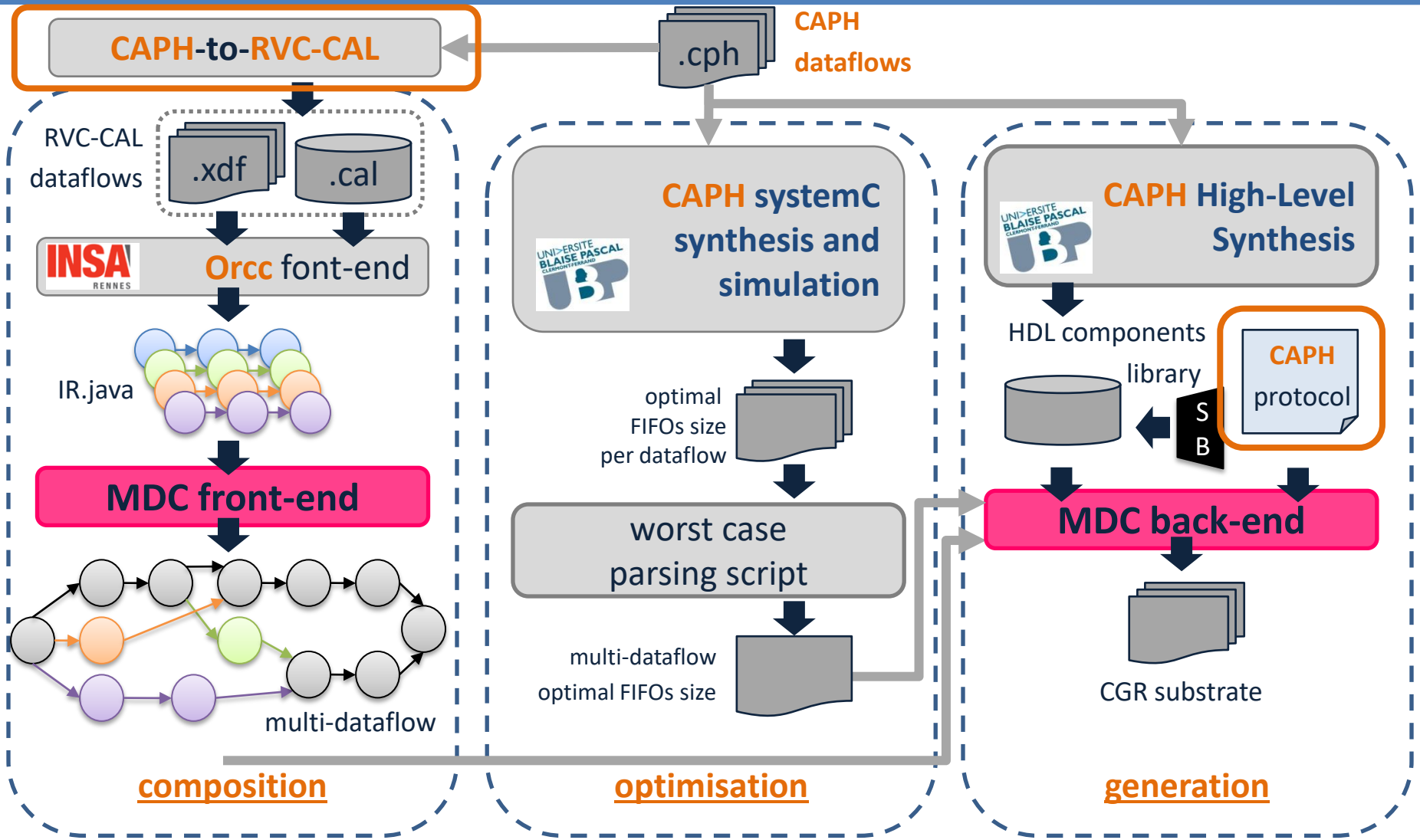
Enhancing MDC High-Level Synthesis Support

Fully Automated Flow



Enhancing MDC High-Level Synthesis Support

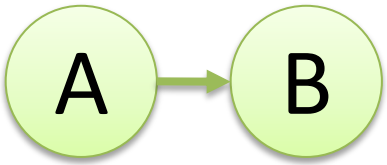
Fully Automated Flow

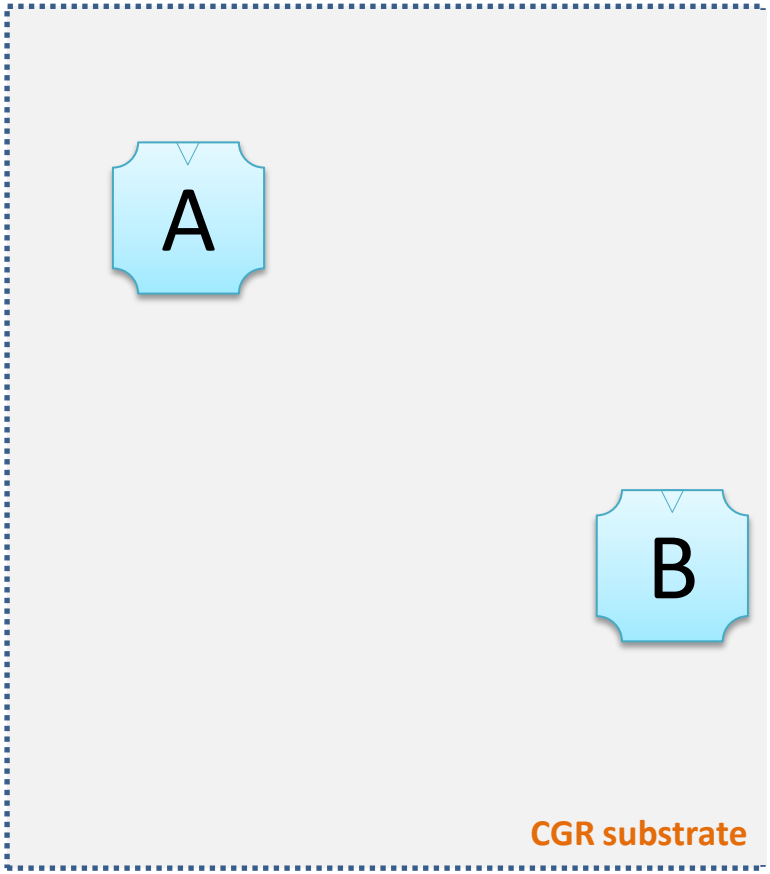


Enhancing MDC High-Level Synthesis Support

Protocol Generalization

```
<protocol>
  <sys_signals>
    <signal id="0" net_port="clock"  is_clock=""…></signal>
    …
  </sys_signals>
  <actor>
    <sys_signals>
      <signal id="0" port="clk" net_port="clock" …></signal>
      …
    </sys_signals>
    <comm_signals>
      <signal id="0" port="din" channel="data"…></signal>
      <signal id="1" port="dout" channel="data"…></signal>
      <signal id="2" port="wr" channel="en"…></signal>
      …
    <comm_signals>
  </actor>
 <predecessor>
    <sys_signals>…</sys_signals>
    <comm_signals>…<comm_signals>
  </predecessor>
  <successor>
    <sys_signals>…</sys_signals>
    <comm_signals>…<comm_signals>
  </successor>
</protocol>
```

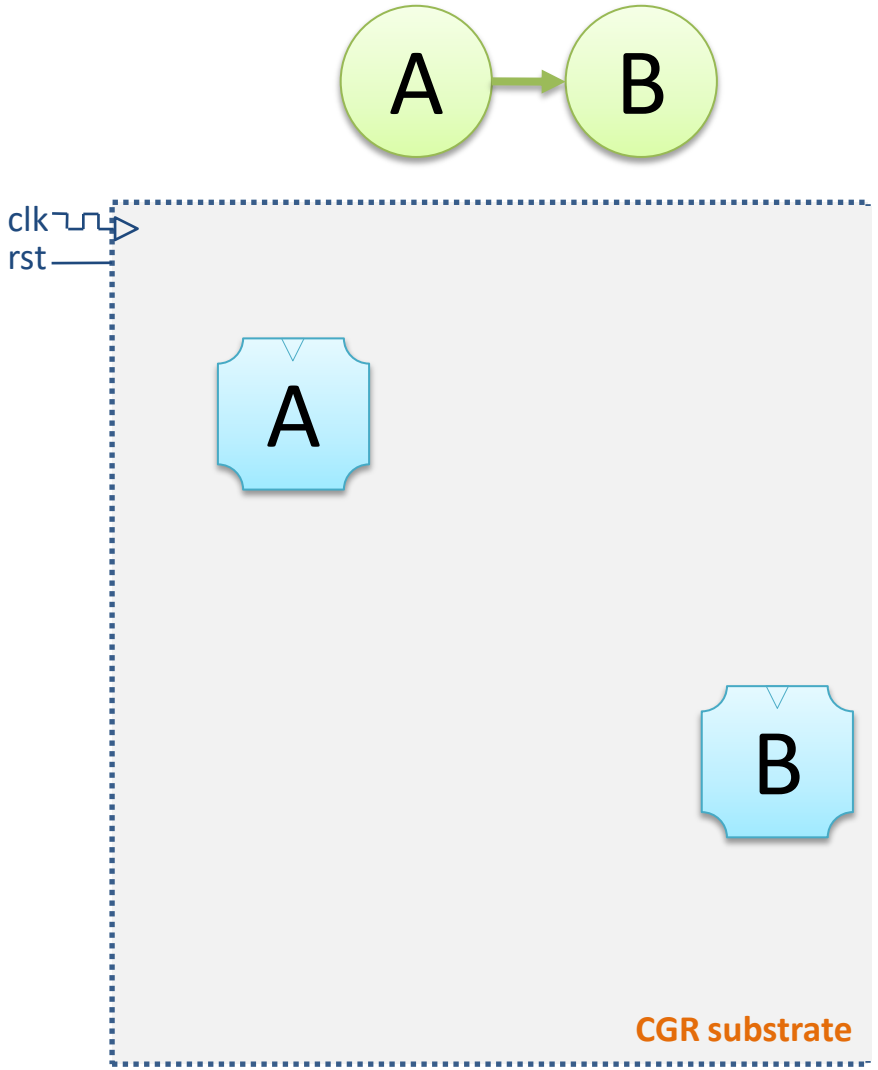




Enhancing MDC High-Level Synthesis Support

Protocol Generalization

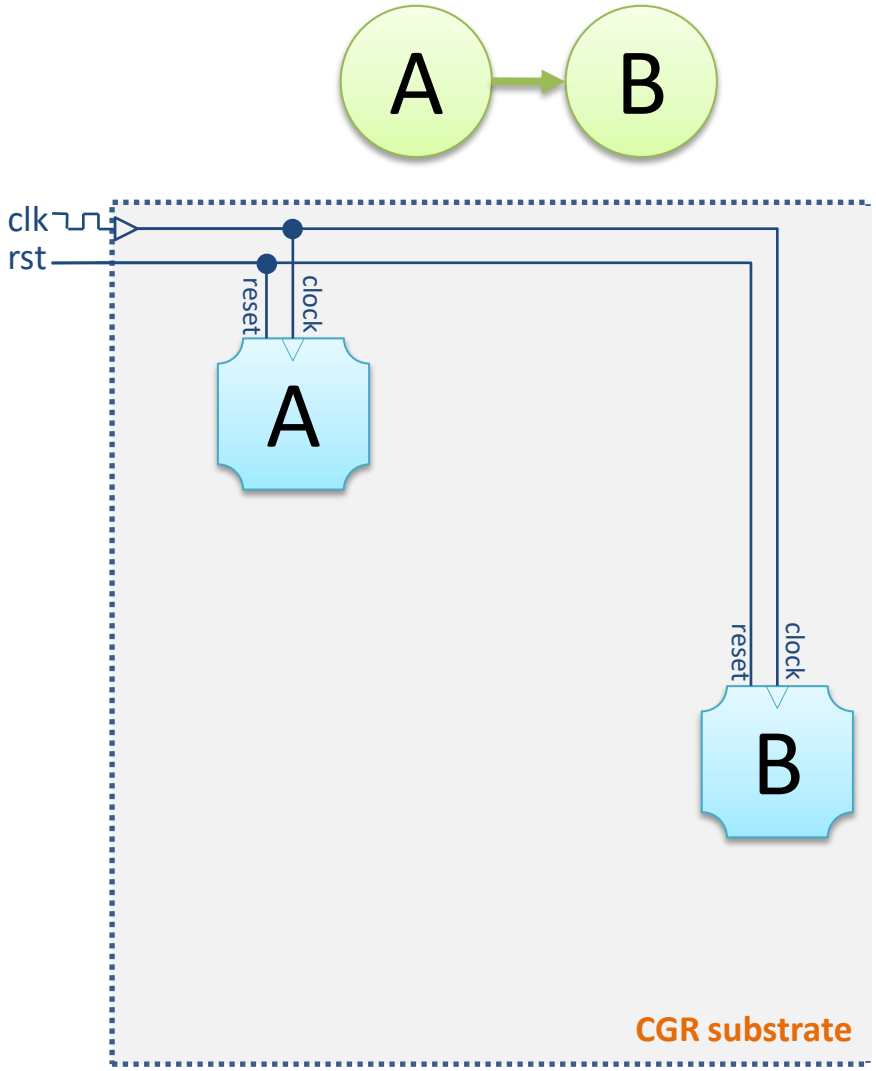
```
<protocol>
  <sys_signals>
    <signal id="0" net_port="clock"  is_clock=""…></signal>
    …
  </sys_signals>
  <actor>
    <sys_signals>
      <signal id="0" port="clk" net_port="clock" …></signal>
      …
    </sys_signals>
    <comm_signals>
      <signal id="0" port="din" channel="data"…></signal>
      <signal id="1" port="dout" channel="data"…></signal>
      <signal id="2" port="wr" channel="en"…></signal>
      …
    <comm_signals>
  </actor>
  <predecessor>
    <sys_signals>…</sys_signals>
    <comm_signals>…<comm_signals>
  </predecessor>
  <successor>
    <sys_signals>…</sys_signals>
    <comm_signals>…<comm_signals>
  </successor>
</protocol>
```



Enhancing MDC High-Level Synthesis Support

Protocol Generalization

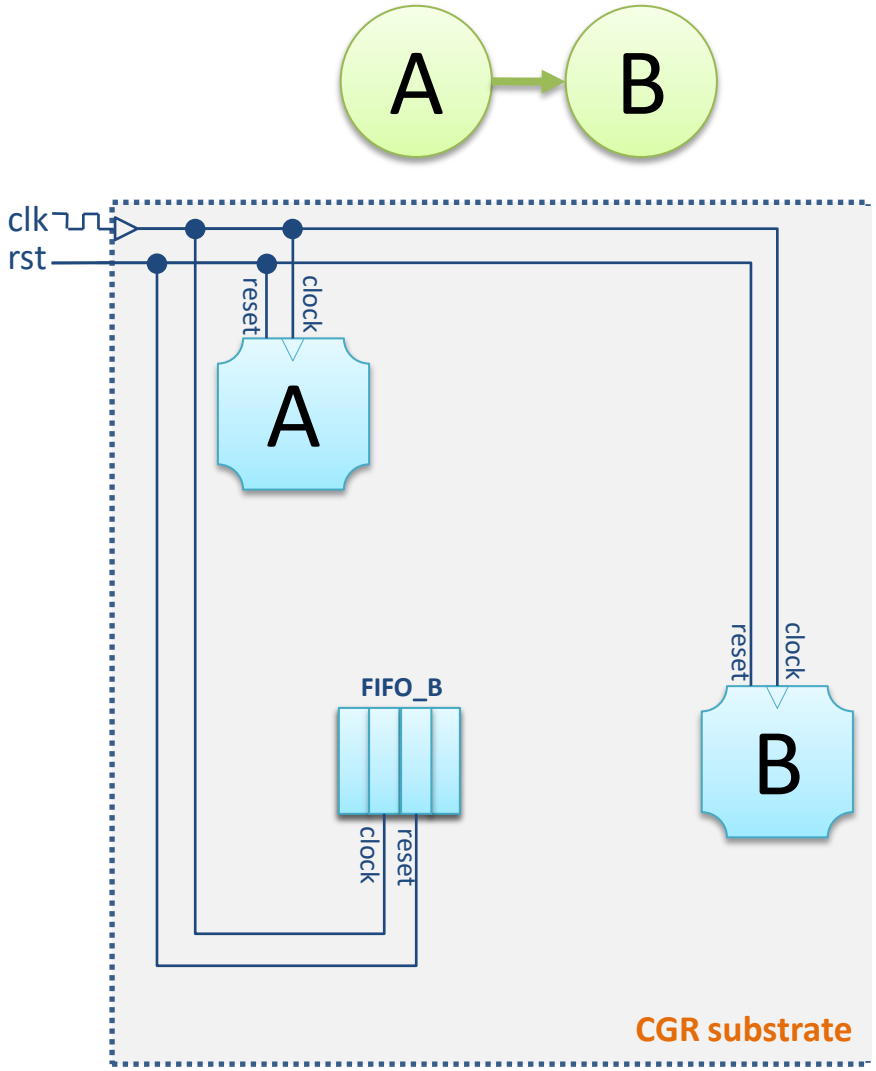
```
<protocol>
  <sys_signals>
    <signal id="0" net_port="clock"  is_clock=""…></signal>
    …
  </sys_signals>
  <actor>
    <sys_signals>
      <signal id="0" port="clk" net_port="clock" …></signal>
      …
    </sys_signals>
    <comm_signals>
      <signal id="0" port="din" channel="data"…></signal>
      <signal id="1" port="dout" channel="data"…></signal>
      <signal id="2" port="wr" channel="en"…></signal>
      …
    <comm_signals>
  </actor>
  <predecessor>
    <sys_signals>…</sys_signals>
    <comm_signals>…<comm_signals>
  </predecessor>
  <successor>
    <sys_signals>…</sys_signals>
    <comm_signals>…<comm_signals>
  </successor>
</protocol>
```



Enhancing MDC High-Level Synthesis Support

Protocol Generalization

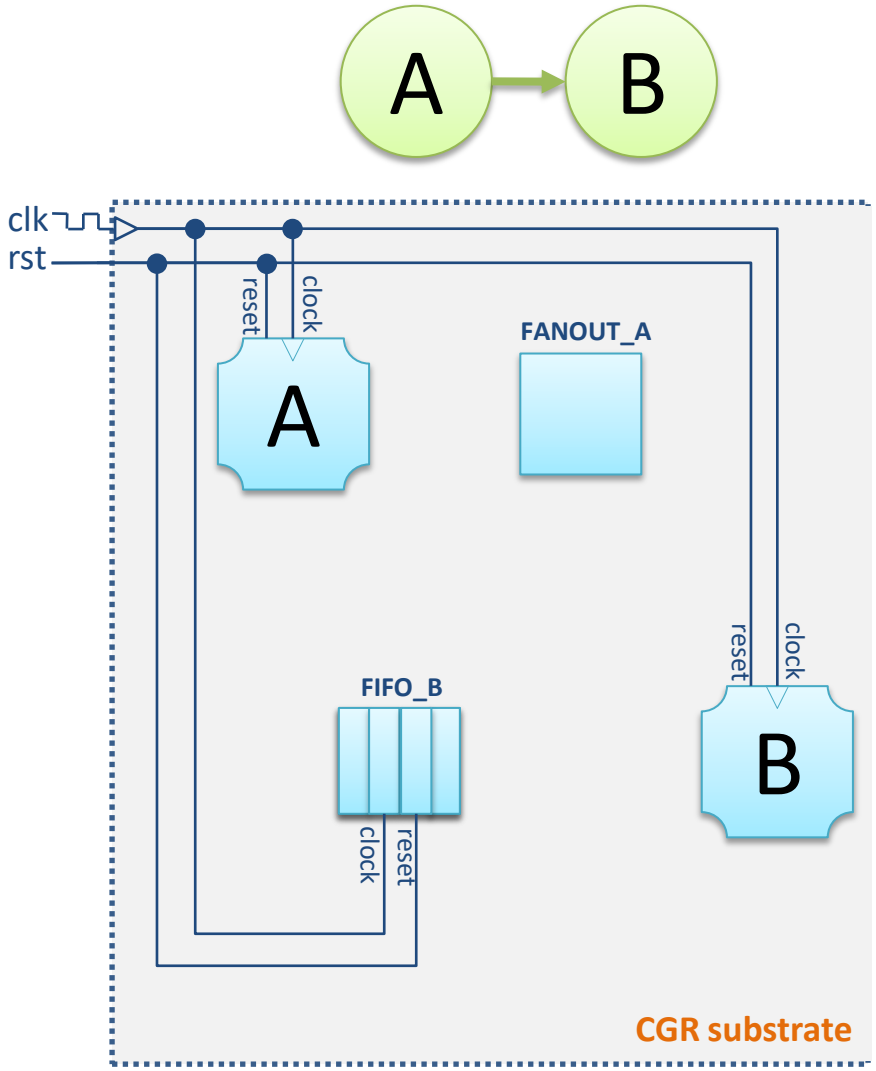
```
<protocol>
  <sys_signals>
    <signal id="0" net_port="clock"  is_clock=""…></signal>
    …
  </sys_signals>
  <actor>
    <sys_signals>
      <signal id="0" port="clk" net_port="clock" …></signal>
      …
    </sys_signals>
    <comm_signals>
      <signal id="0" port="din" channel="data"…></signal>
      <signal id="1" port="dout" channel="data"…></signal>
      <signal id="2" port="wr" channel="en"…></signal>
      …
    <comm_signals>
  </actor>
  <predecessor>
    <sys_signals>…</sys_signals>
    <comm_signals>…<comm_signals>
  </predecessor>
  <successor>
    <sys_signals>…</sys_signals>
    <comm_signals>…<comm_signals>
  </successor>
</protocol>
```



Enhancing MDC High-Level Synthesis Support

Protocol Generalization

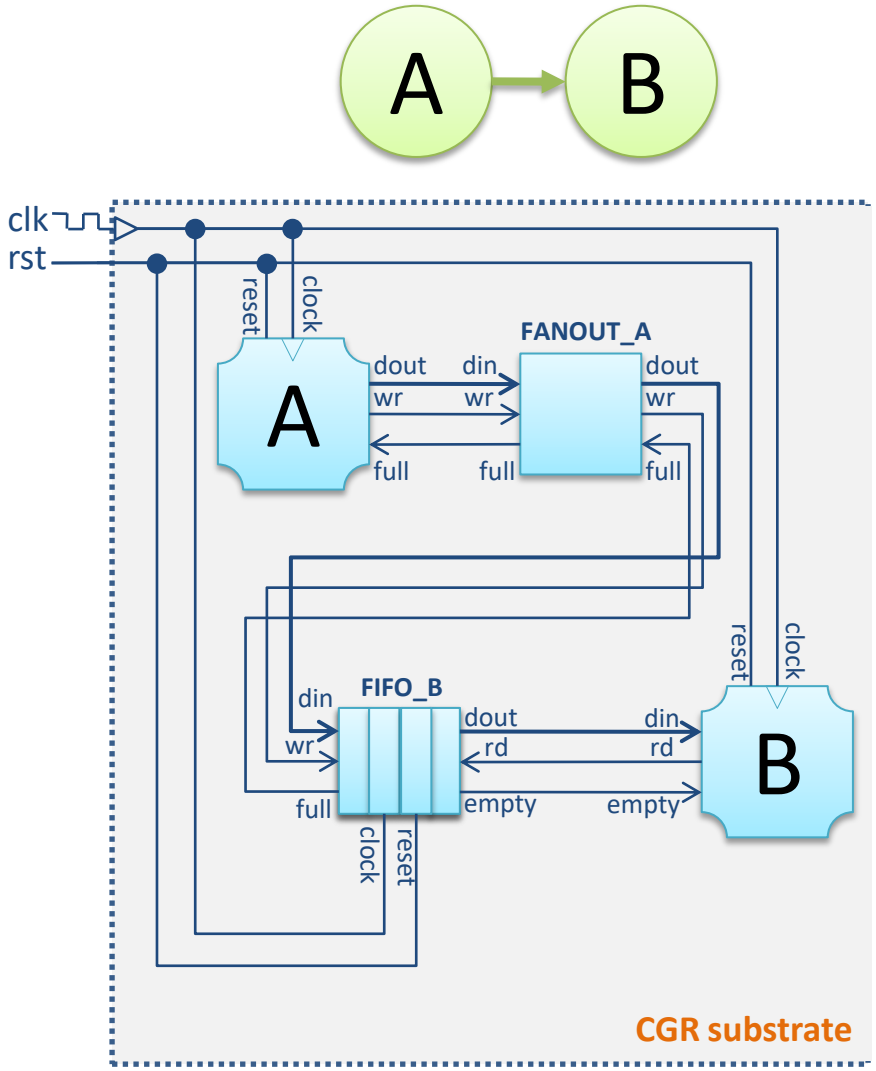
```
<protocol>
  <sys_signals>
    <signal id="0" net_port="clock"  is_clock=""…></signal>
    …
  </sys_signals>
  <actor>
    <sys_signals>
      <signal id="0" port="clk" net_port="clock" …></signal>
      …
    </sys_signals>
    <comm_signals>
      <signal id="0" port="din" channel="data"…></signal>
      <signal id="1" port="dout" channel="data"…></signal>
      <signal id="2" port="wr" channel="en"…></signal>
      …
    <comm_signals>
  </actor>
  <predecessor>
    <sys_signals>…</sys_signals>
    <comm_signals>…<comm_signals>
  </predecessor>
  <successor>
    <sys_signals>…</sys_signals>
    <comm_signals>…<comm_signals>
  </successor>
</protocol>
```



Enhancing MDC High-Level Synthesis Support

Protocol Generalization

```
<protocol>
  <sys_signals>
    <signal id="0" net_port="clock"  is_clock=""…></signal>
    …
  </sys_signals>
  <actor>
    <sys_signals>
      <signal id="0" port="clk" net_port="clock" …></signal>
      …
    </sys_signals>
    <comm_signals>
      <signal id="0" port="din" channel="data"…></signal>
      <signal id="1" port="dout" channel="data"…></signal>
      <signal id="2" port="wr" channel="en"…></signal>
      …
    <comm_signals>
  </actor>
  <predecessor>
    <sys_signals>…</sys_signals>
    <comm_signals>…<comm_signals>
  </predecessor>
  <successor>
    <sys_signals>…</sys_signals>
    <comm_signals>…<comm_signals>
  </successor>
</protocol>
```



Enhancing MDC High-Level Synthesis Support

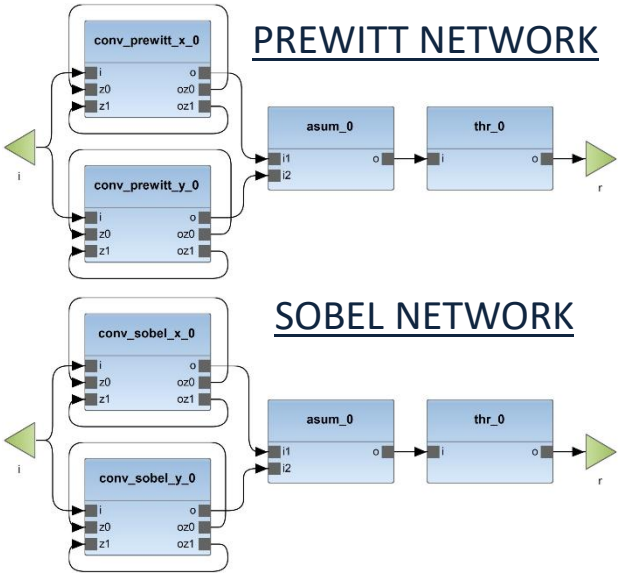
Prewitt/Sobel Multi-Flow Network

Enhancing MDC High-Level Synthesis Support

Prewitt/Sobel Multi-Flow Network

INPUT NETWORKS

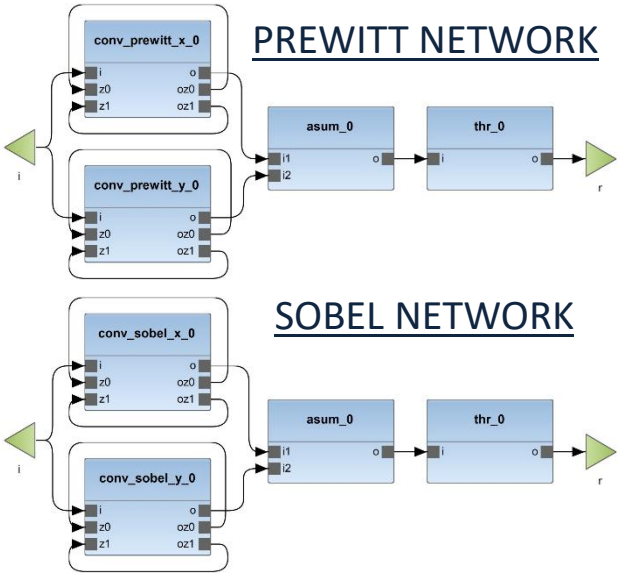
(provided by CAPH)



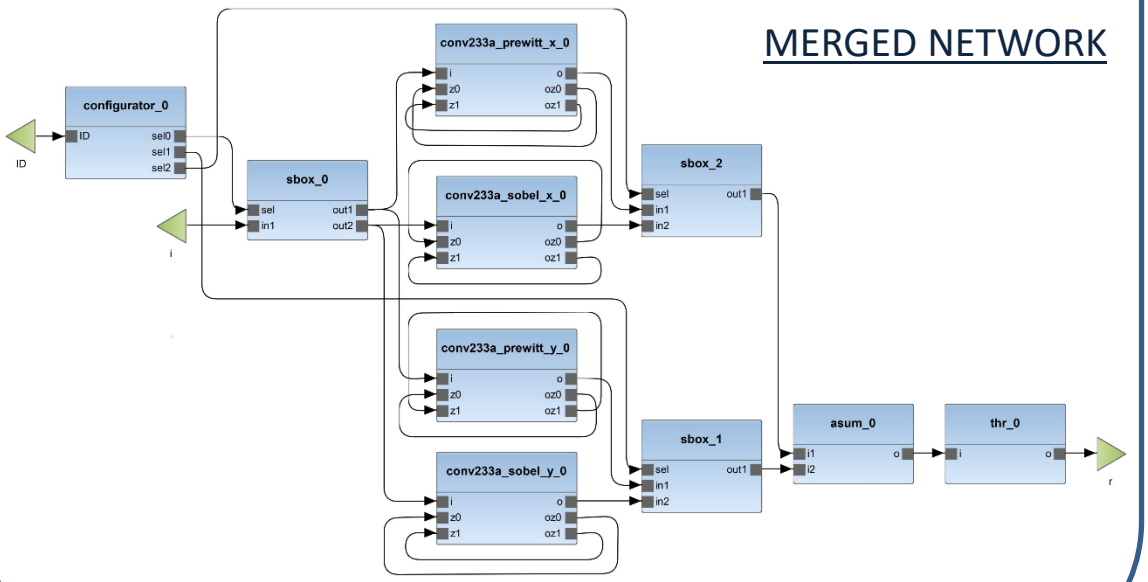
Enhancing MDC High-Level Synthesis Support

Prewitt/Sobel Multi-Flow Network

INPUT NETWORKS
(provided by CAPH)



OUTPUT NETWORK
(provided by MDC)



Enhancing MDC High-Level Synthesis Support

Preliminary Results

Enhancing MDC High-Level Synthesis Support

Preliminary Results

FPGA - Altera (5SGSMD5) and Xilinx (XC7VX485T)

RESOURCES	MDC+CAPH		MDC+XRONOS		XRONOS vs CAPH	
	Altera	Xilinx	Altera	Xilinx	Altera	Xilinx
REG	1484	780	-	632	-	-18,97%
LOGIC	1047	2347	-	1533	-	-34,68%
RAM	15	0	-	6.5	-	+100%
DSP	36	36	-	0	-	-100%
MAX FREQ [MHz]	105,80	93,69	-	142,86	-	+58,50%
EXEC TIME [cck]	15340	15340	-	15348	-	+0,05%

Enhancing MDC High-Level Synthesis Support

Preliminary Results

FPGA - Altera (5SGSMD5) and Xilinx (XC7VX485T)

RESOURCES	MDC+CAPH		MDC+XRONOS		XRONOS vs CAPH	
	Altera	Xilinx	Altera	Xilinx	Altera	Xilinx
REG	1484	780	-	632	-	-18,97%
LOGIC	1047	2347	-	1533	-	-34,68%
RAM	15	0	-	6.5	-	+100%
DSP	36	36	-	0	-	-100%
MAX FREQ [MHz]	105,80	93,69	-	142,86	-	+58,50%
EXEC TIME [cck]	15340	15340	-	15348	-	+0,05%

ASIC - TSMC 45 nm CMOS technology

	Prewitt/Sobel	Multi-Flow
AREA [kGE]	269,82	466,90 (+73%)
Max Freq [MHz]	417,36	399.04 (-4,4%)

Enhancing MDC High-Level Synthesis Support

Preliminary Results

FPGA - Altera (5SGSMD5) and Xilinx (XC7VX485T)

RESOURCES	MDC+CAPH		MDC+XRONOS		XRONOS vs CAPH	
	Altera	Xilinx	Altera	Xilinx	Altera	Xilinx

COMING SOON:

EXPLORATION ON THE BENEFITS OF DATAFLOW-BASED HLS IN CGR ARCHITECTURES ON THE ROAD

ASIC - TSMC 45 nm CMOS technology

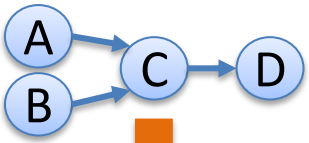
	Prewitt/Sobel	Multi-Flow
AREA [kGE]	269,82	466,90 (+73%)
Max Freq [MHz]	417,36	399.04 (-4,4%)

Outline

- MDC Tool Summary
  - Motivations and Approach
  - Current Functionalities and Future Directions
- Hardware-Software Partitioning
  - Co-Processing Support and Automated Characterization
- Enhancing the MDC High-Level Synthesis Support
  - Integration with the CAPH HLS engine
- Run-time Monitoring of CGR Accelerators
  - Extension of PAPI for dataflow in CGR hardware
- Providing Further Degrees of Reconfigurability
  - Mixed-Grain Reconfiguration Possibilities

Run-time Monitoring of CGR Accelerators

PAPI for dataflow in software



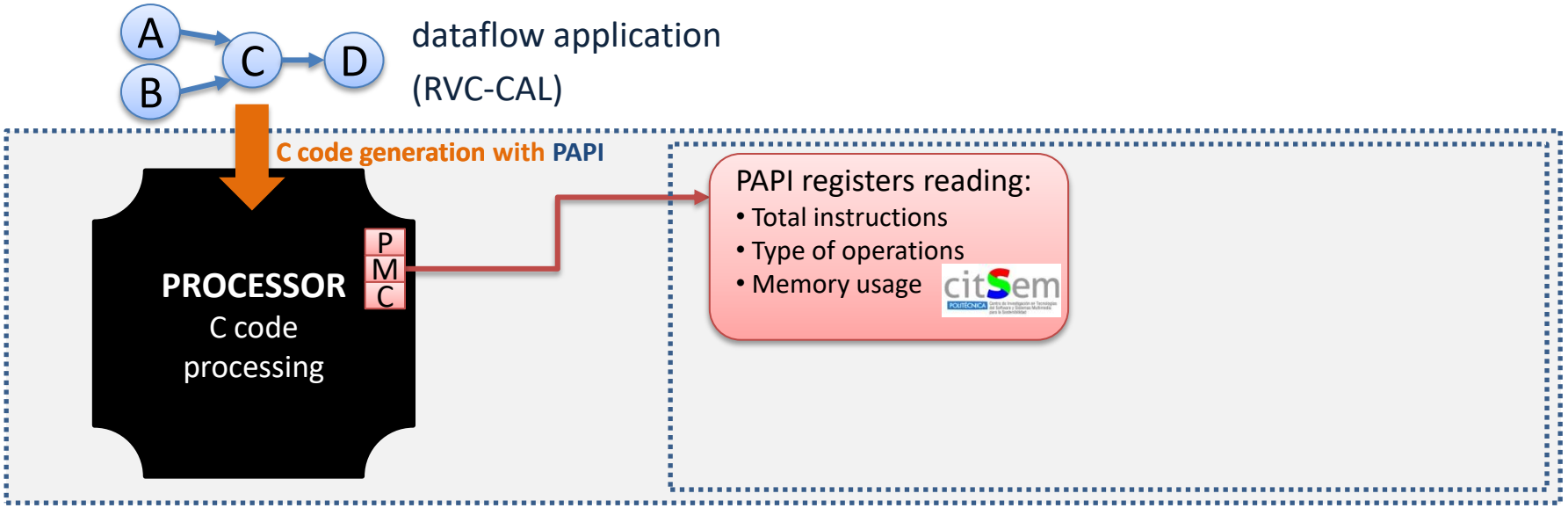
dataflow application
(RVC-CAL)

C code generation

PROCESSOR
C code
processing

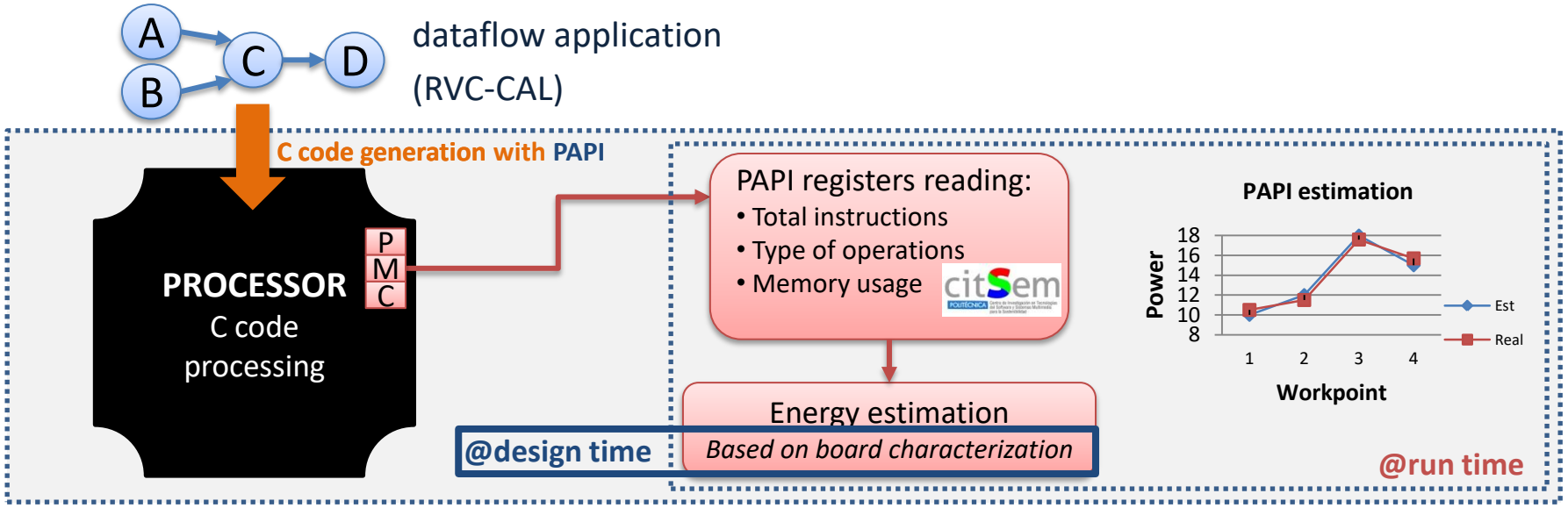
Run-time Monitoring of CGR Accelerators

PAPI for dataflow in software



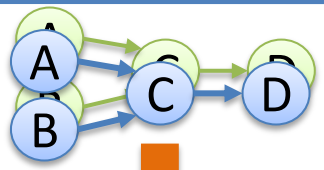
Run-time Monitoring of CGR Accelerators

PAPI for dataflow in software



Run-time Monitoring of CGR Accelerators

Extension of PAPI for dataflow in CGR hardware



dataflow applications
(RVC-CAL)

C code generation with PAPI

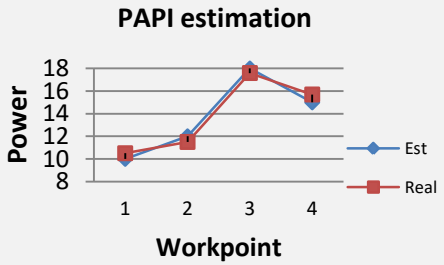
PROCESSOR
C code
processing

P
M
C

PAPI registers reading:
- Total instructions
- Type of operations
- Memory usage

Energy estimation
Based on board characterization

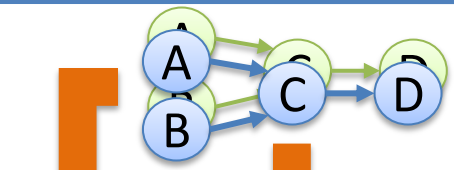
@design time



@run time

Run-time Monitoring of CGR Accelerators

Extension of PAPI for dataflow in CGR hardware



dataflow applications
(RVC-CAL)

C code generation with PAPI

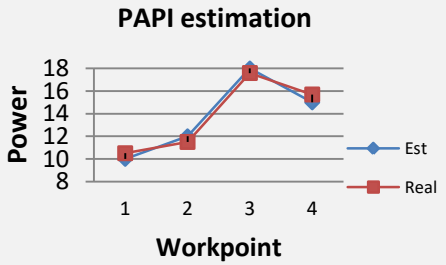
PROCESSOR
C code
processing

P
M
C

PAPI registers reading:
- Total instructions
- Type of operations
- Memory usage

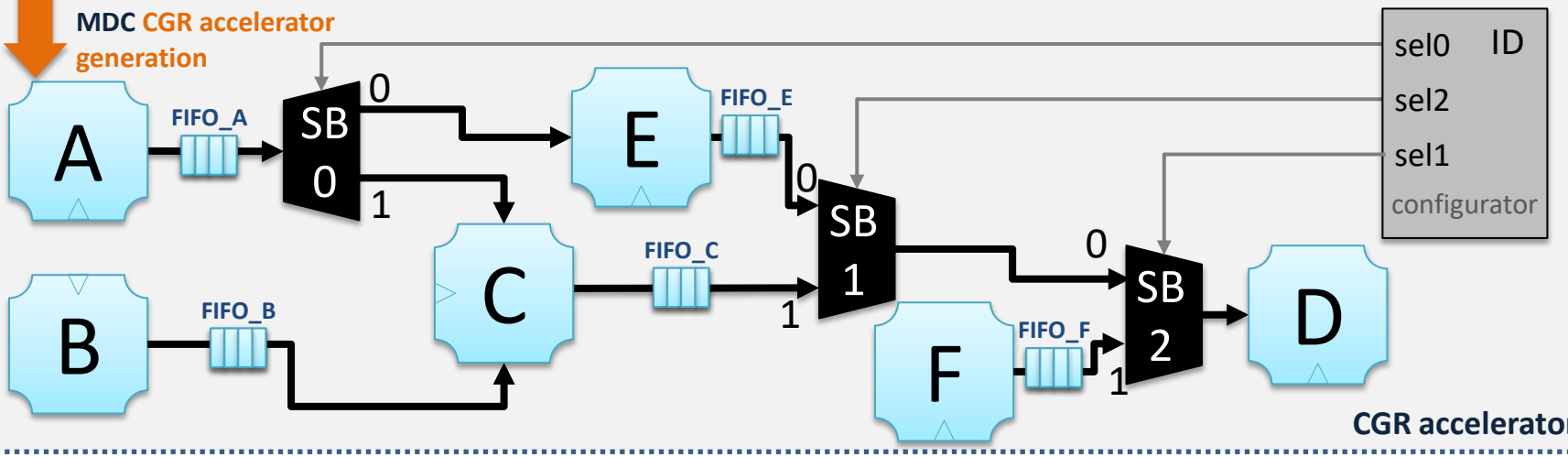
Energy estimation
Based on board characterization

@design time



@run time

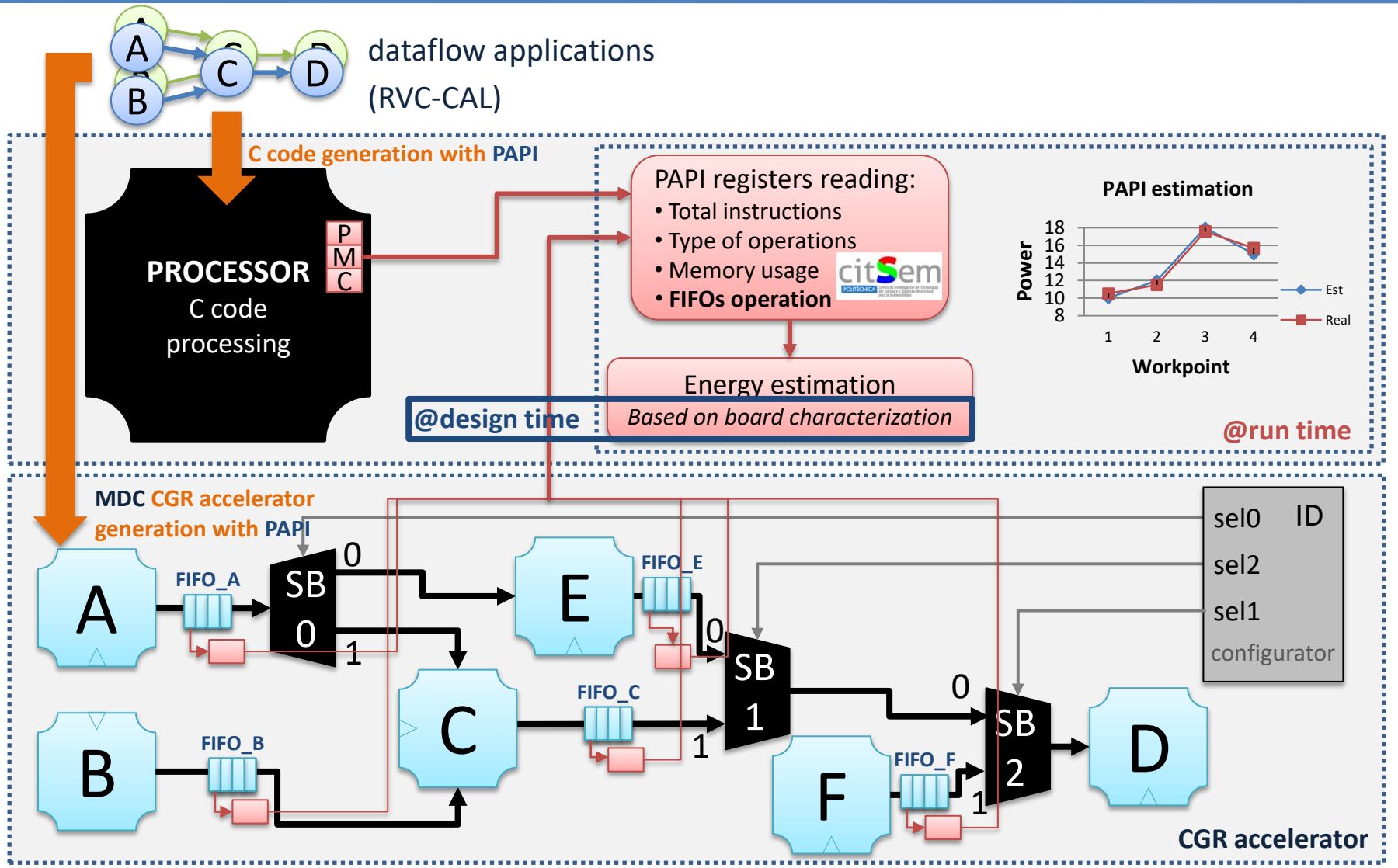
MDC CGR accelerator generation



CGR accelerator

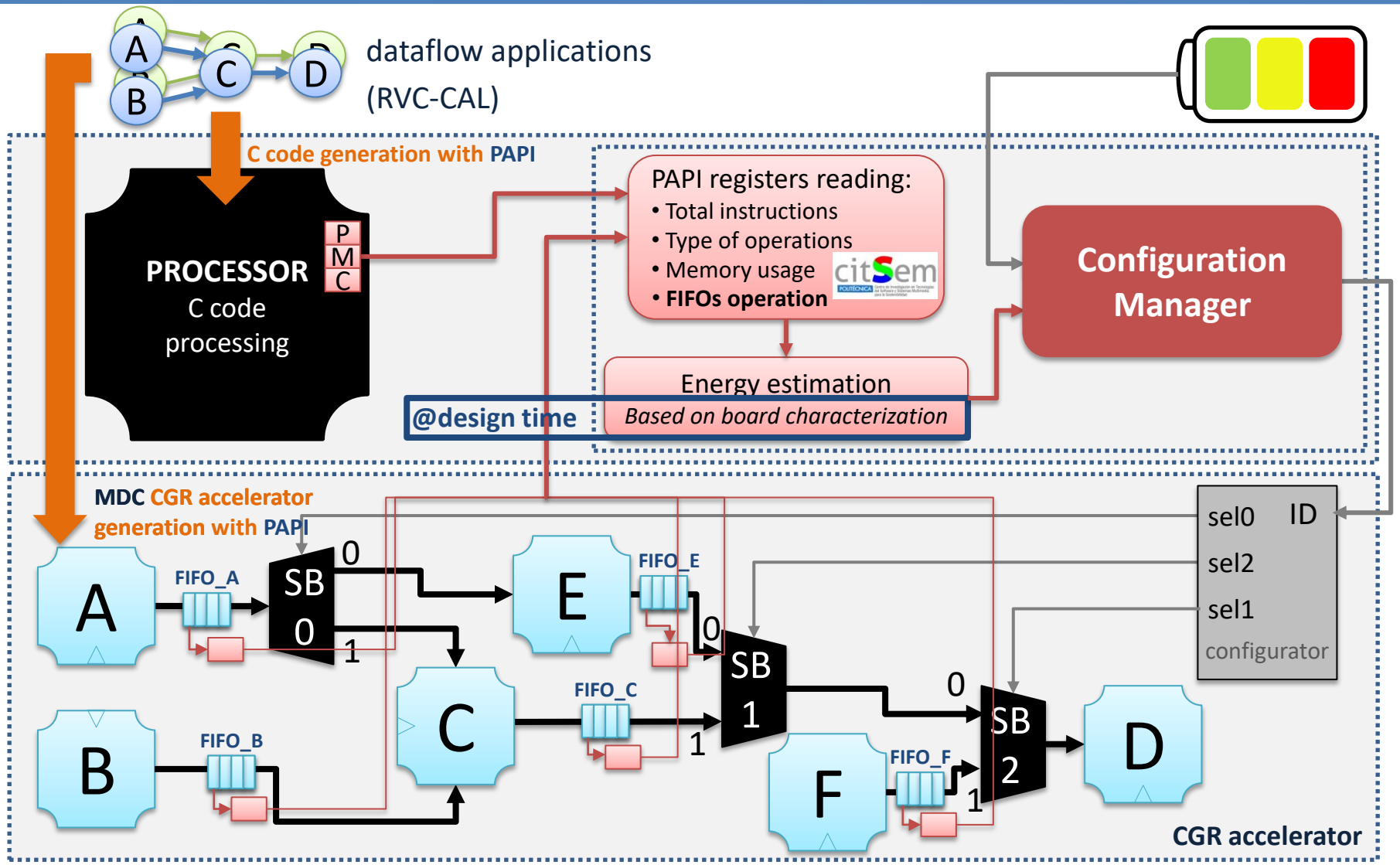
Run-time Monitoring of CGR Accelerators

Extension of PAPI for dataflow in CGR hardware



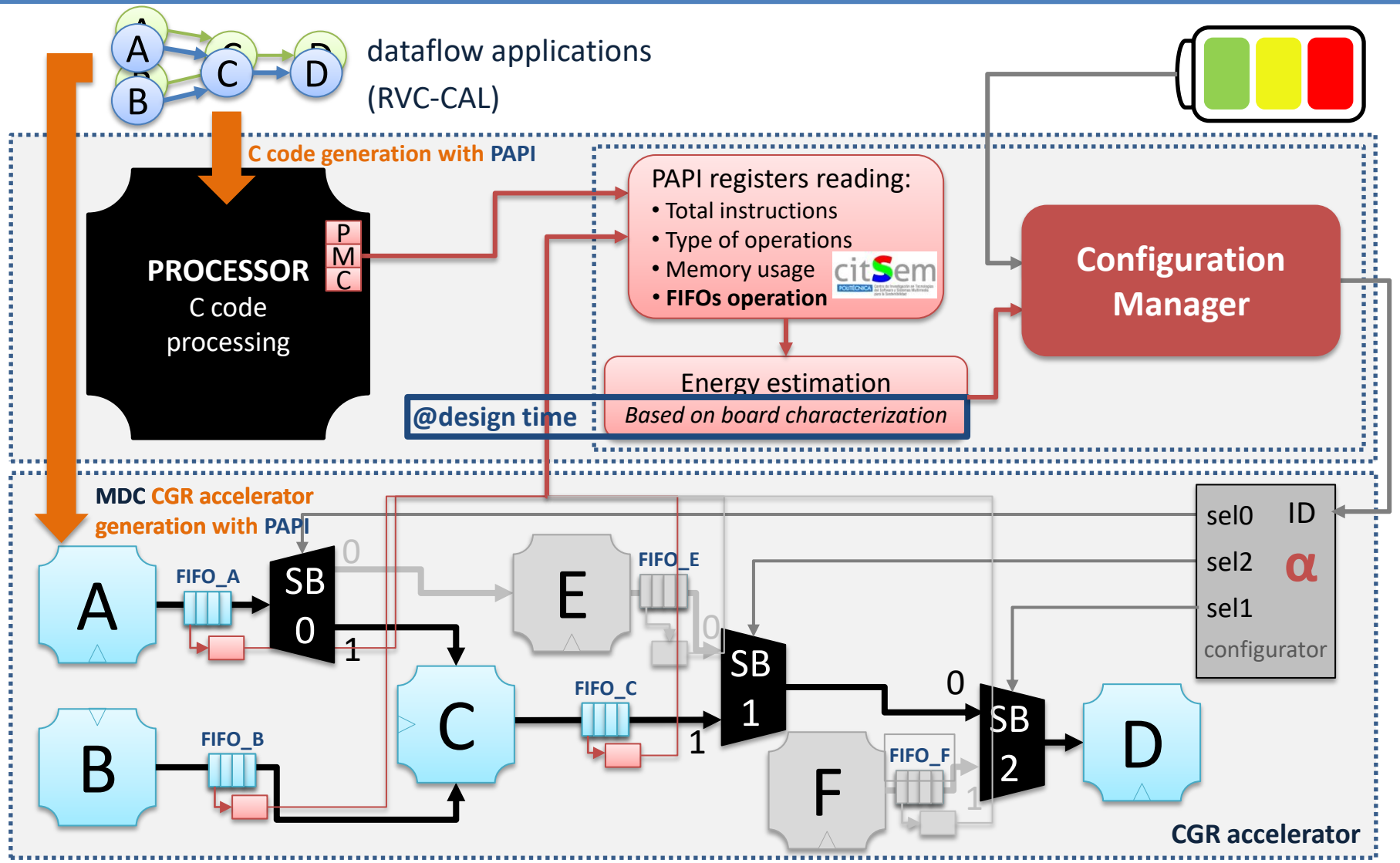
Run-time Monitoring of CGR Accelerators

Extension of PAPI for dataflow in CGR hardware



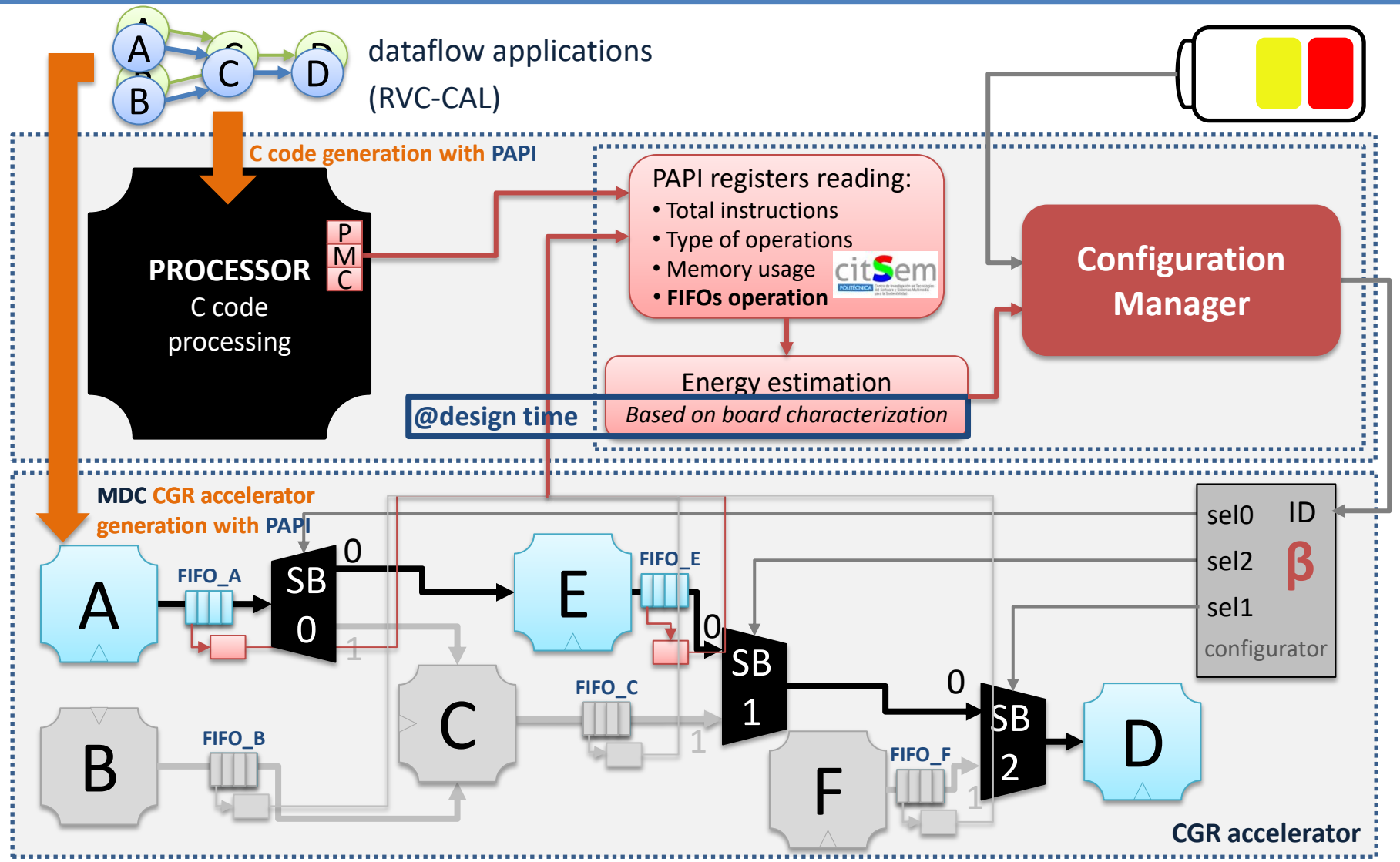
Run-time Monitoring of CGR Accelerators

Extension of PAPI for dataflow in CGR hardware



Run-time Monitoring of CGR Accelerators

Extension of PAPI for dataflow in CGR hardware



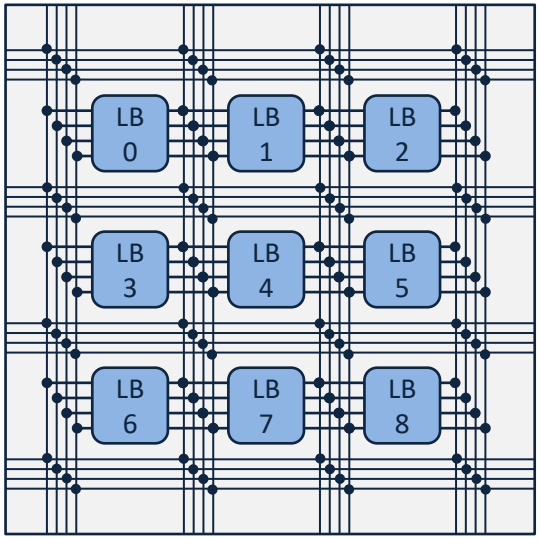
Outline

- MDC Tool Summary
  - Motivations and Approach
  - Current Functionalities and Future Directions
- Hardware-Software Partitioning
  - Co-Processing Support and Automated Characterization
- Enhancing the MDC High-Level Synthesis Support
  - Integration with the CAPH HLS engine
- Run-time Monitoring of CGR Accelerators
  - Extension of PAPI for dataflow in CGR hardware
- Providing Further Degrees of Reconfigurability
  - Mixed-Grain Reconfiguration Possibilities

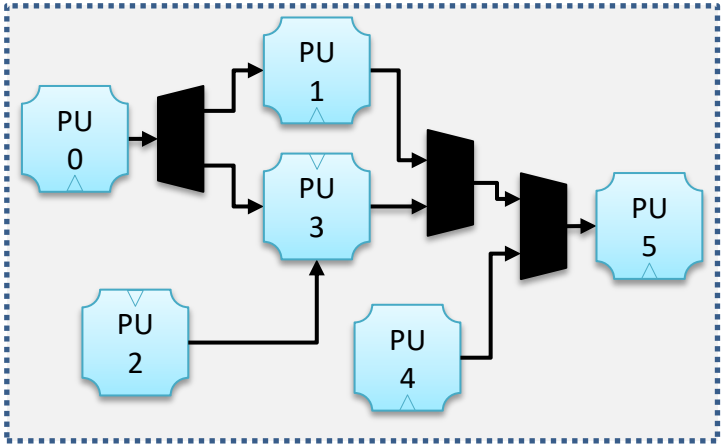
Providing Further Degrees of Reconfigurability

Fine-Grain and Partial Reconfiguration

FINE-GRAIN RECONFIGURATION (FPGA)



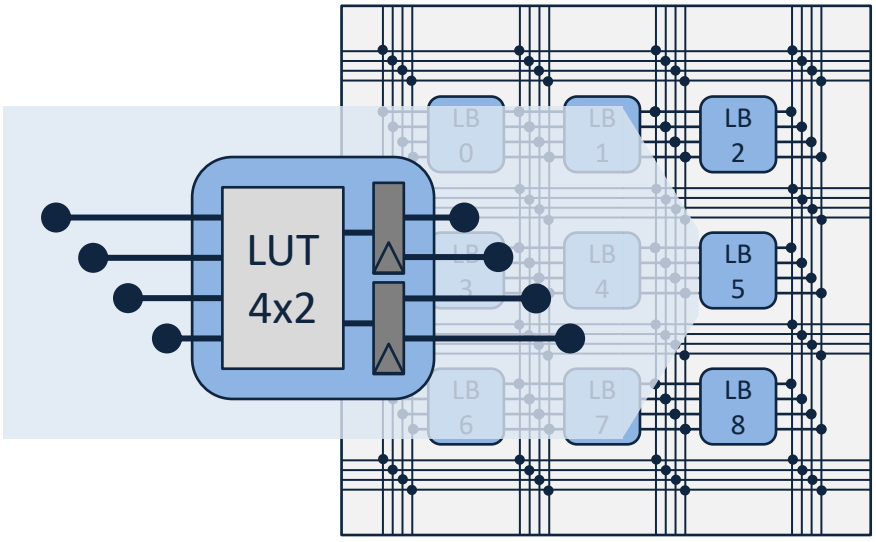
COARSE-GRAIN RECONFIGURATION (MDC ACCELERATOR)



Providing Further Degrees of Reconfigurability

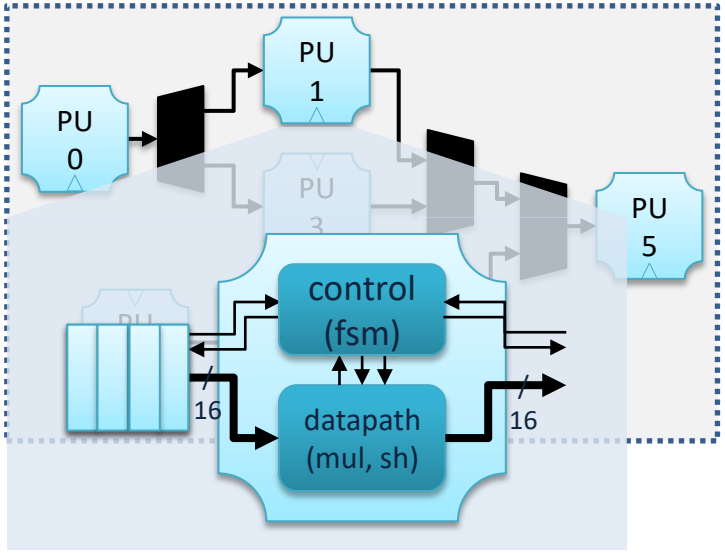
Fine-Grain and Partial Reconfiguration

FINE-GRAIN RECONFIGURATION (FPGA)



bit-level reconfiguration

COARSE-GRAIN RECONFIGURATION (MDC ACCELERATOR)

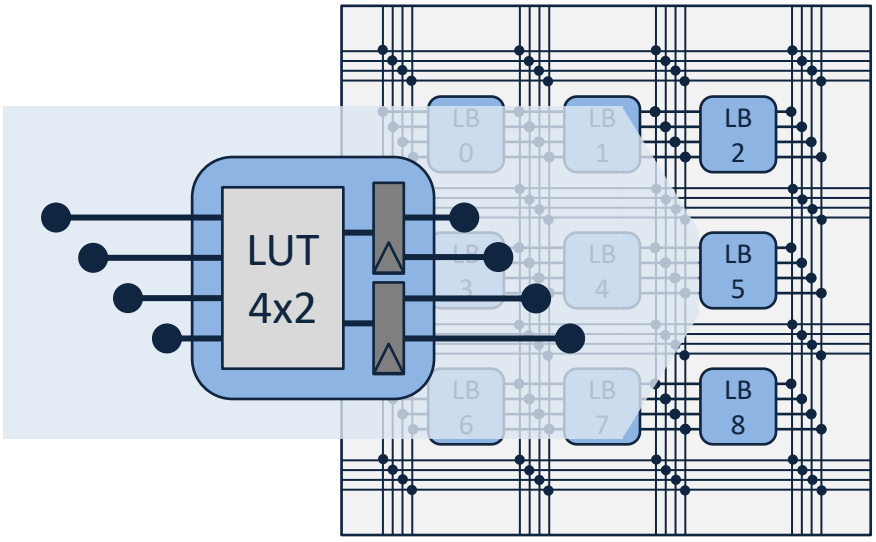


word-level reconfiguration

Providing Further Degrees of Reconfigurability

Fine-Grain and Partial Reconfiguration

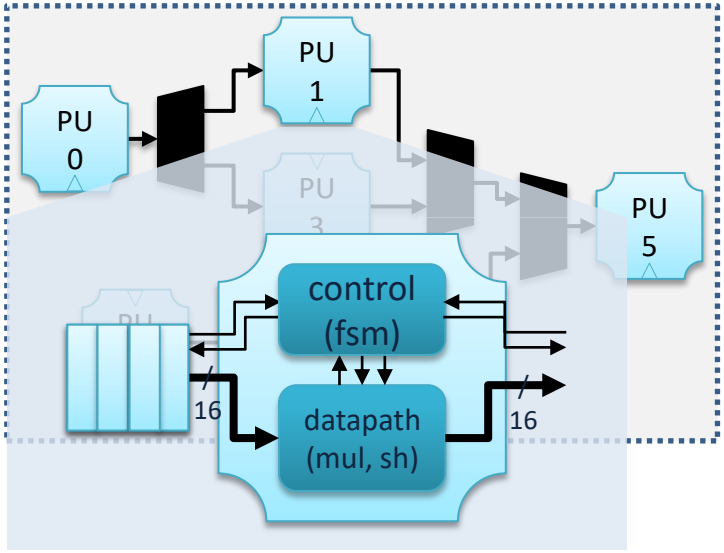
FINE-GRAIN RECONFIGURATION (FPGA)



bit-level reconfiguration

very flexible (any kind of HDL defined system)

COARSE-GRAIN RECONFIGURATION (MDC ACCELERATOR)



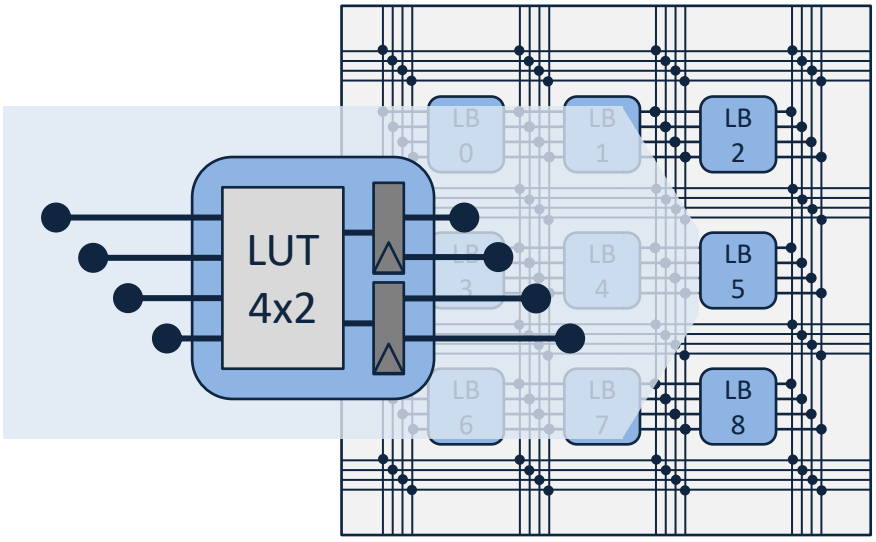
word-level reconfiguration

small flexibility (fixed set of predefined configuration)

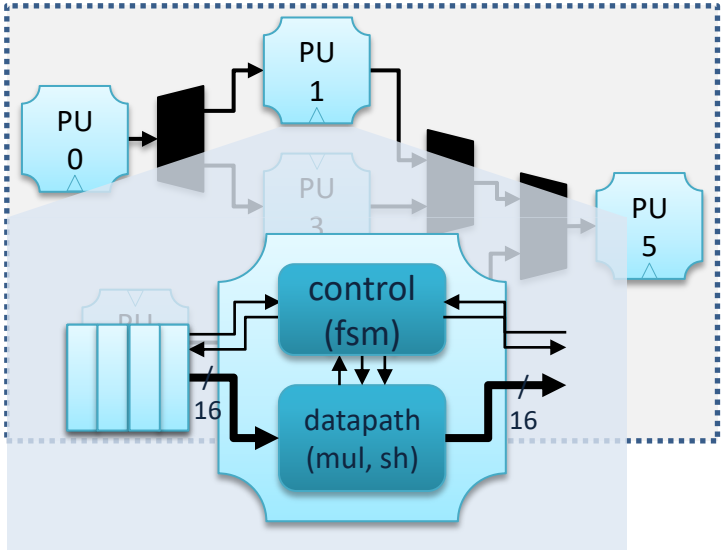
Providing Further Degrees of Reconfigurability

Fine-Grain and Partial Reconfiguration

FINE-GRAIN RECONFIGURATION (FPGA)



COARSE-GRAIN RECONFIGURATION (MDC ACCELERATOR)



bit-level reconfiguration

very flexible (any kind of HDL defined system)

slow to configure (lot of switches and LUTs)

big memory footprint (long configuration bitstream)

word-level reconfiguration

small flexibility (fixed set of predefined configuration)

fast to configure (small amount of switches)

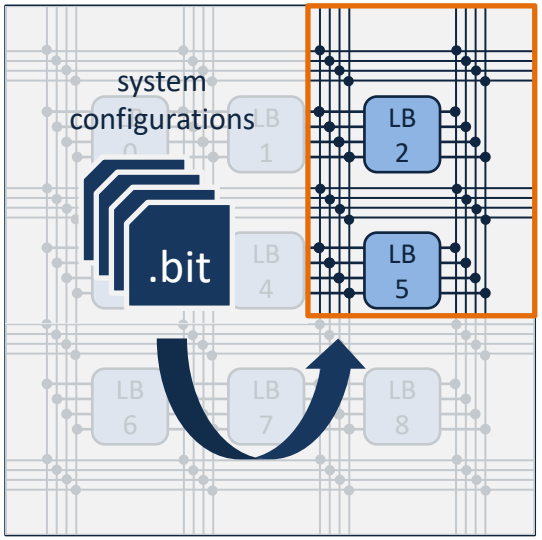
negligible memory footprint ($\log_2$(#config) bits)

Providing Further Degrees of Reconfigurability

Fine-Grain and Partial Reconfiguration

FINE-GRAIN RECONFIGURATION (FPGA)

DYNAMIC PARTIAL RECONFIGURATION (DPR)
runtime reconfiguration of only a well defined region of the FPGA



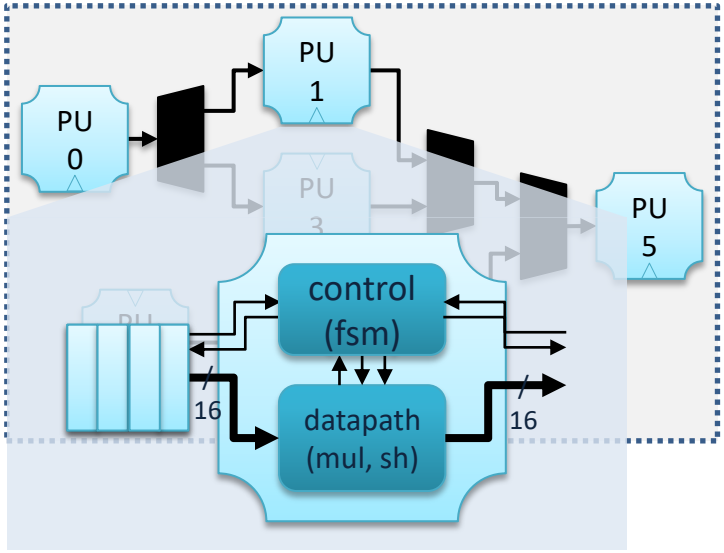
bit-level reconfiguration

very flexible (any kind of HDL defined system)

slow to configure (lot of switches and LUTs)

big memory footprint (long configuration bitstream)

COARSE-GRAIN RECONFIGURATION (MDC ACCELERATOR)



word-level reconfiguration

small flexibility (fixed set of predefined configuration)

fast to configure (small amount of switches)

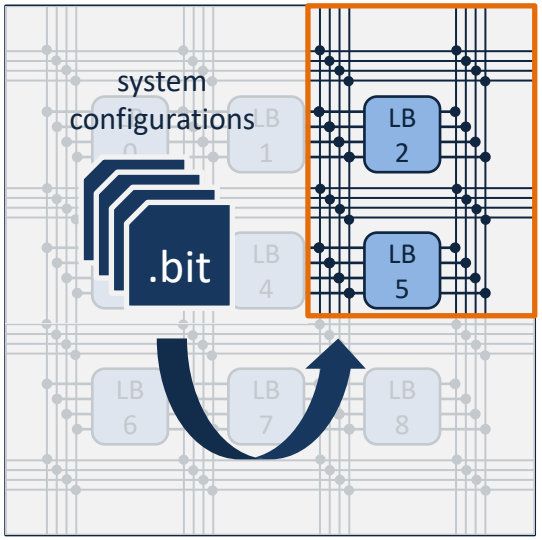
negligible memory footprint ($\log_2$(#config) bits)

Providing Further Degrees of Reconfigurability

Fine-Grain and Partial Reconfiguration

FINE-GRAIN RECONFIGURATION (FPGA)

DYNAMIC PARTIAL RECONFIGURATION (DPR)
runtime reconfiguration of only a well defined region of the FPGA



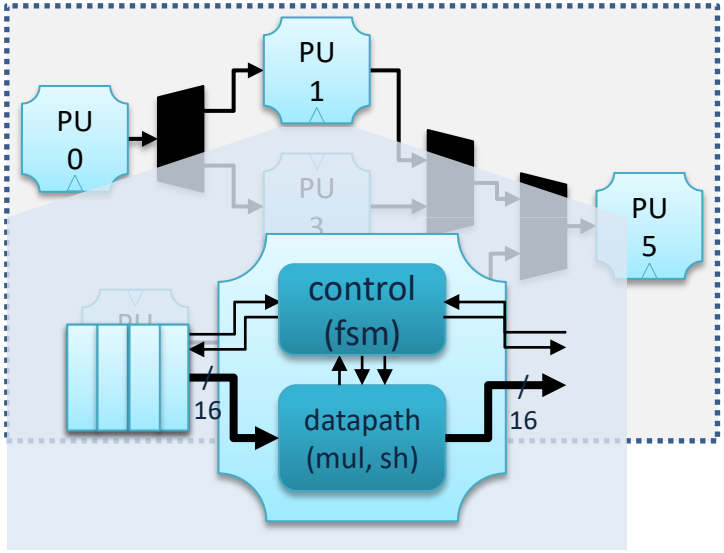
bit-level reconfiguration

flexible (HDL systems precedently implemented)

time to configure typically in terms of ms

memory footprint to be considered

COARSE-GRAIN RECONFIGURATION (MDC ACCELERATOR)



word-level reconfiguration

small flexibility (fixed set of predefined configuration)

fast to configure (small amount of switches)

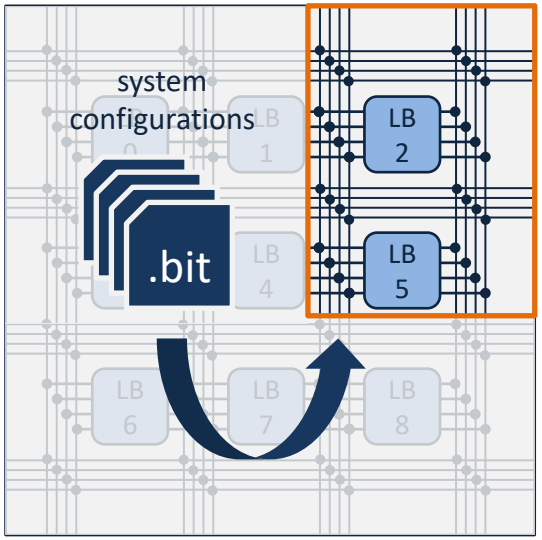
negligible memory footprint ($\log_2$(#config) bits)

Providing Further Degrees of Reconfigurability

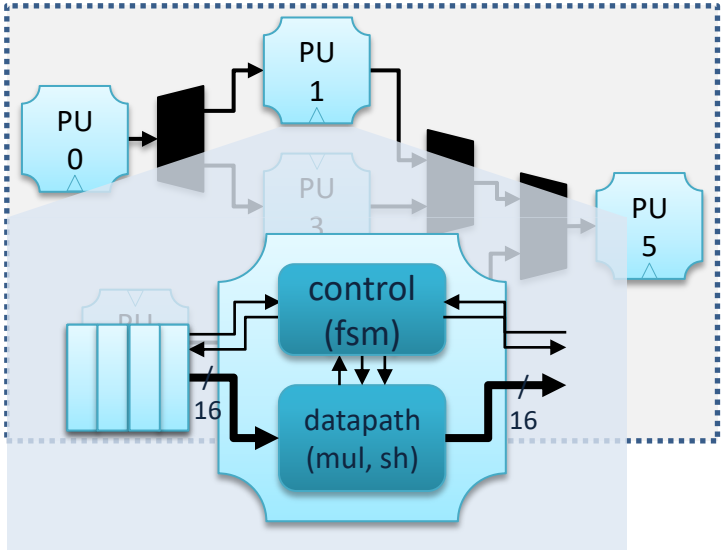
Fine-Grain and Partial Reconfiguration

FINE-GRAIN RECONFIGURATION (FPGA)

DYNAMIC PARTIAL RECONFIGURATION (DPR)
runtime reconfiguration of only a well defined region of the FPGA



COARSE-GRAIN RECONFIGURATION (MDC ACCELERATOR)



bit-level reconfiguration

flexible (HDL systems precedently implemented)

time to configure typically in terms of ms

memory footprint to be considered

power consumption peak during reconfiguration

word-level reconfiguration

small flexibility (fixed set of predefined configuration)

fast to configure (small amount of switches)

negligible memory footprint ($\log_2$(#config) bits)

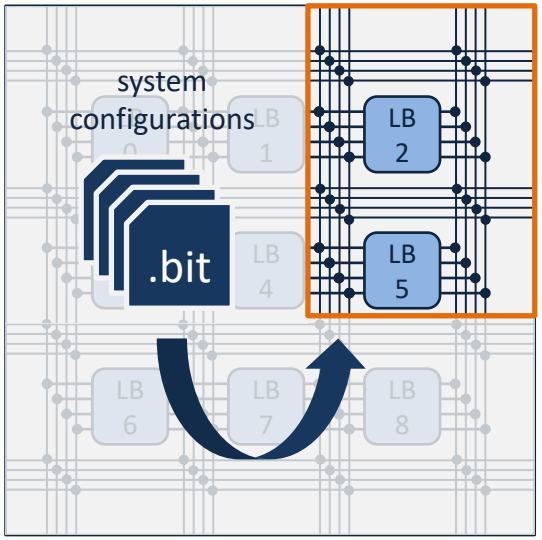
power consumption due to reconfiguration

Providing Further Degrees of Reconfigurability

Fine-Grain and Partial Reconfiguration

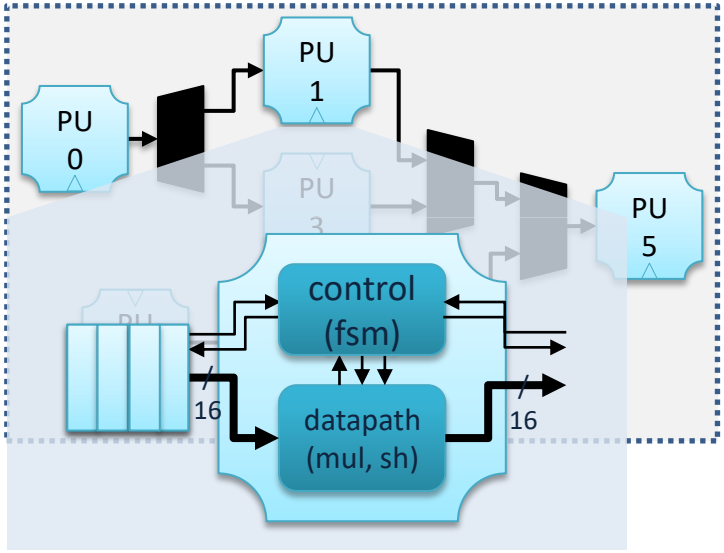
FINE-GRAIN RECONFIGURATION (FPGA)

DYNAMIC PARTIAL RECONFIGURATION (DPR)
runtime reconfiguration of only a well defined region of the FPGA



bit-level reconfiguration

COARSE-GRAIN RECONFIGURATION (MDC ACCELERATOR)



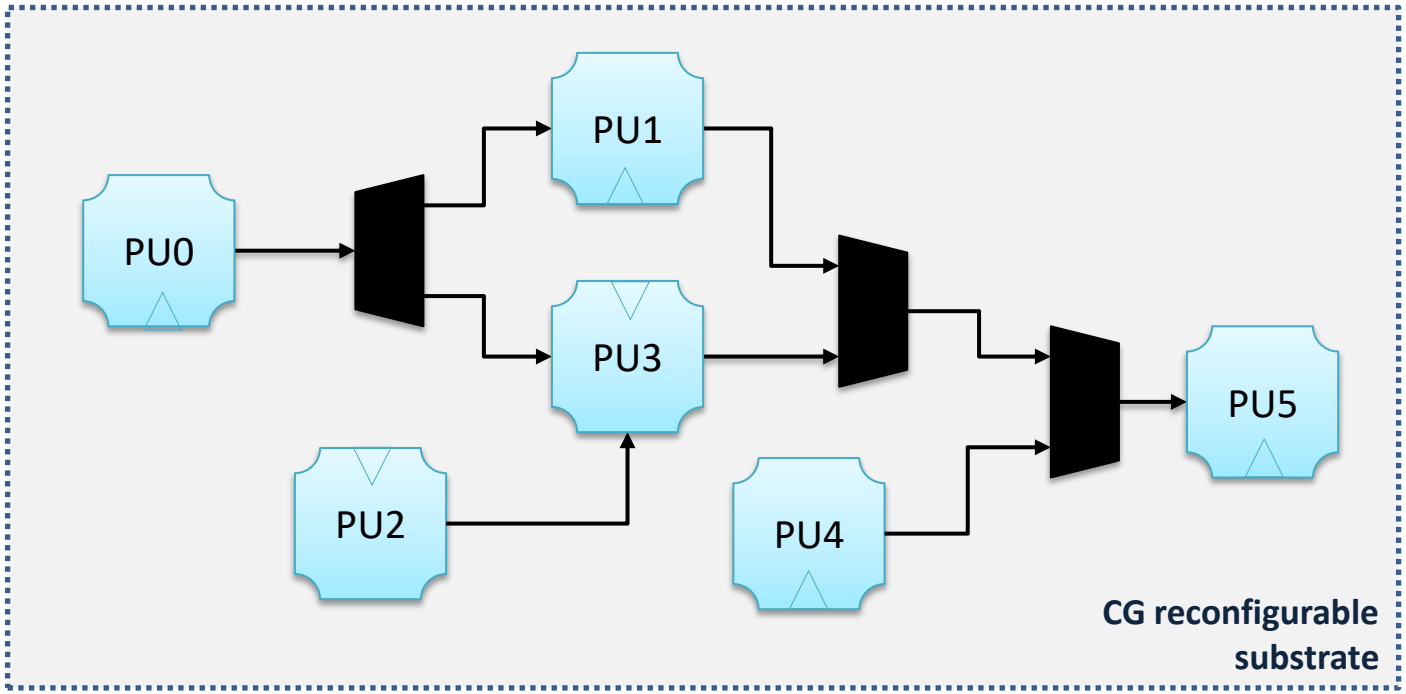
word-level reconfiguration

COMPLEMENTARITY

DPR	CGR
BIG change, BIG overhead	SMALL change, SMALL overhead

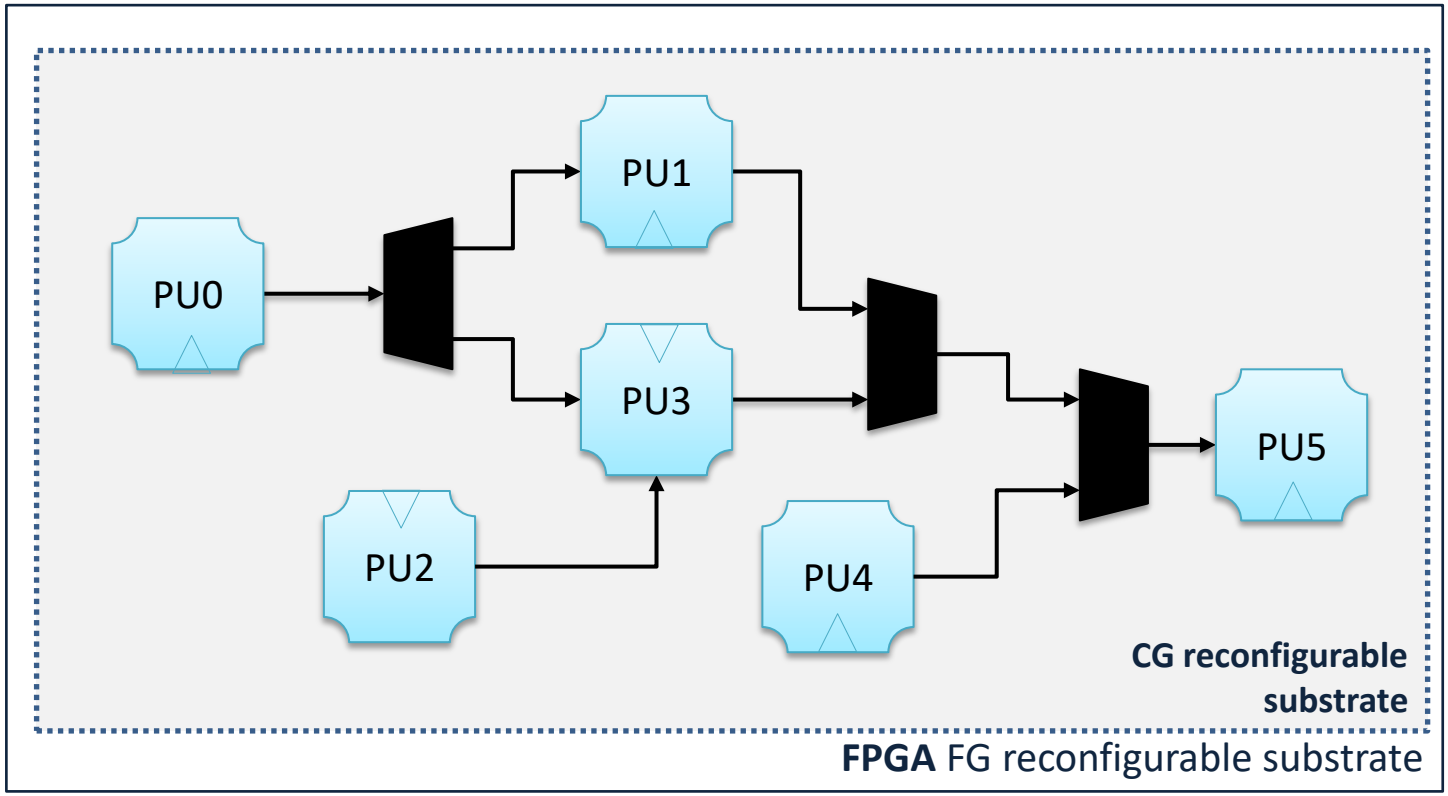
Providing Further Degrees of Reconfigurability

FG into CG reconfiguration



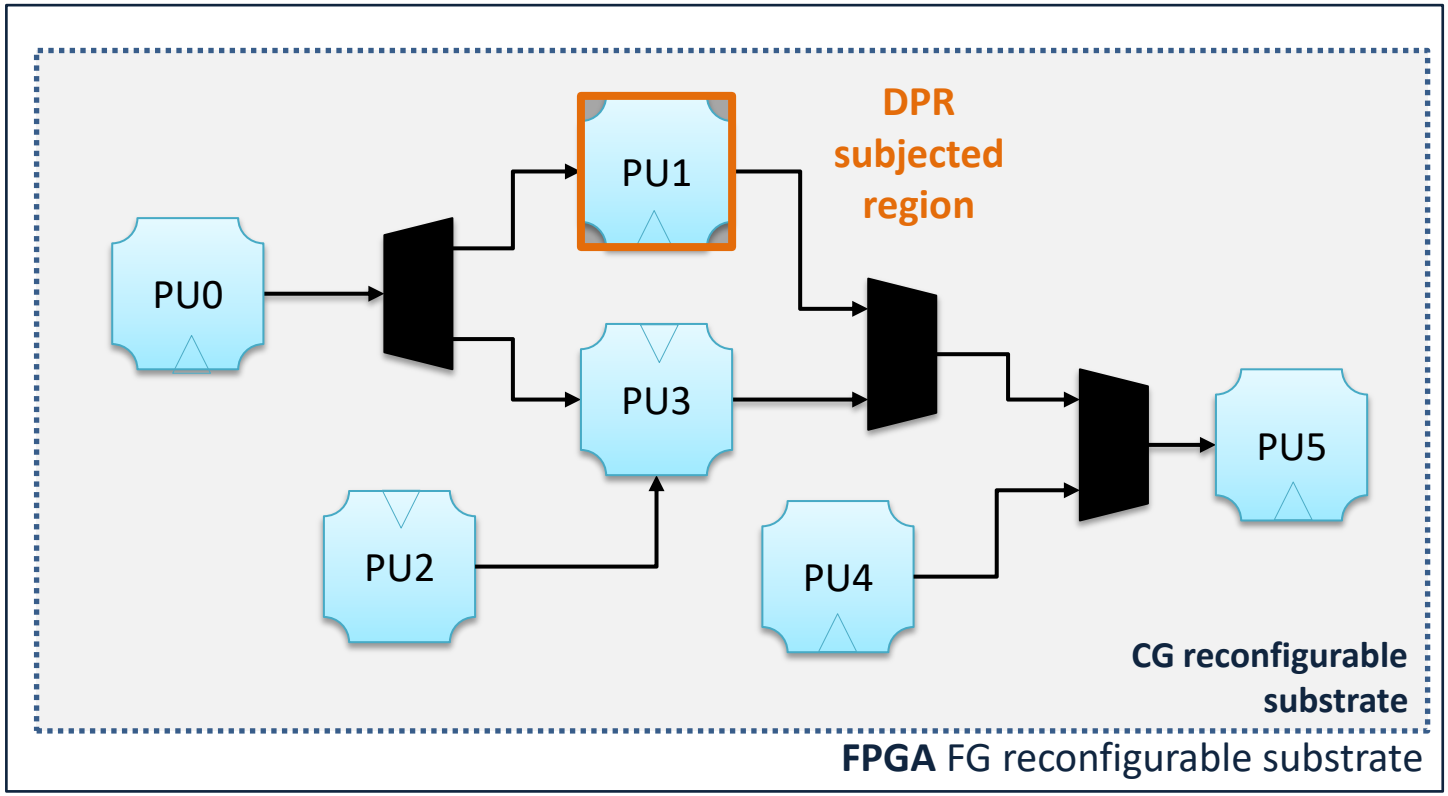
Providing Further Degrees of Reconfigurability

FG into CG reconfiguration



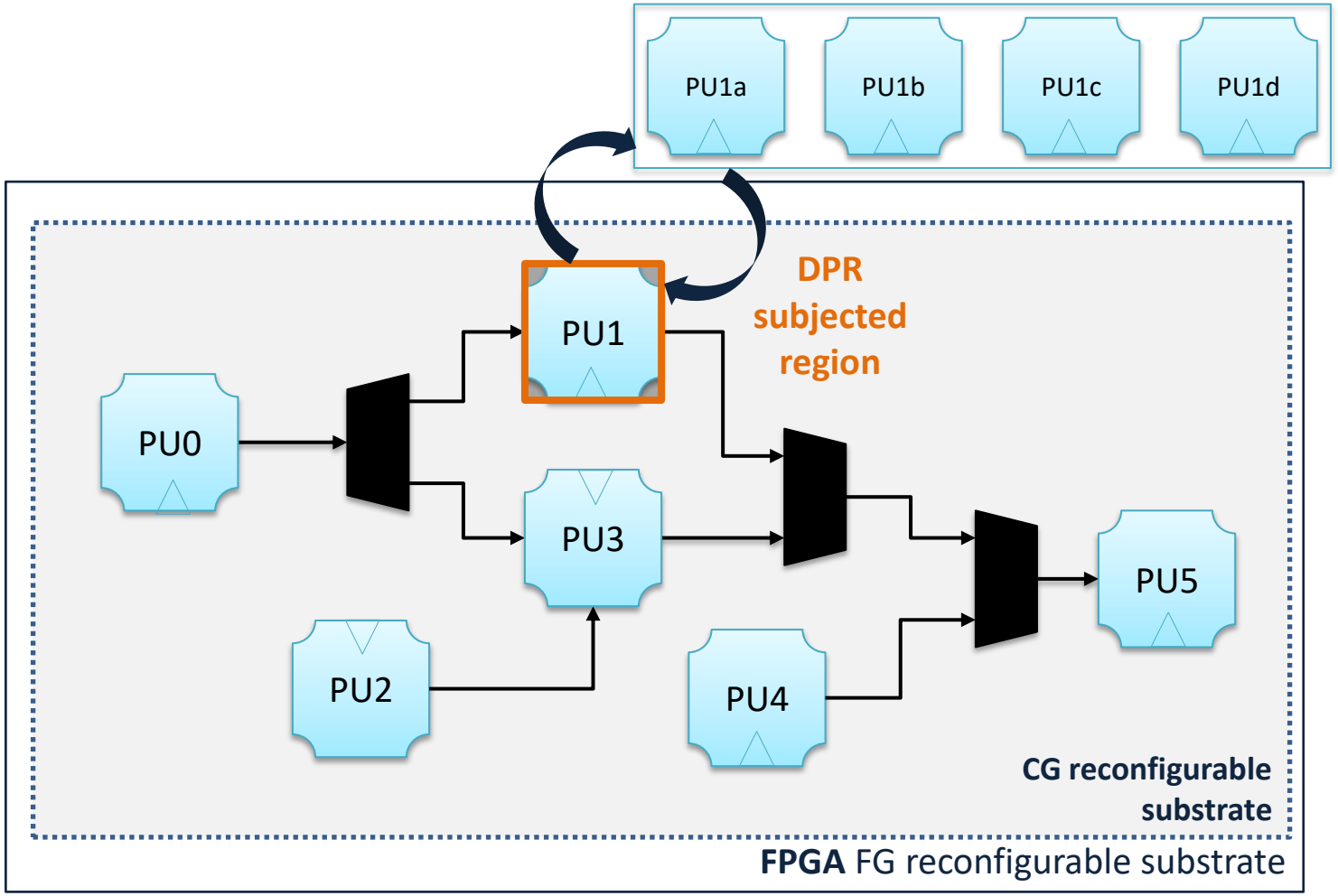
Providing Further Degrees of Reconfigurability

FG into CG reconfiguration



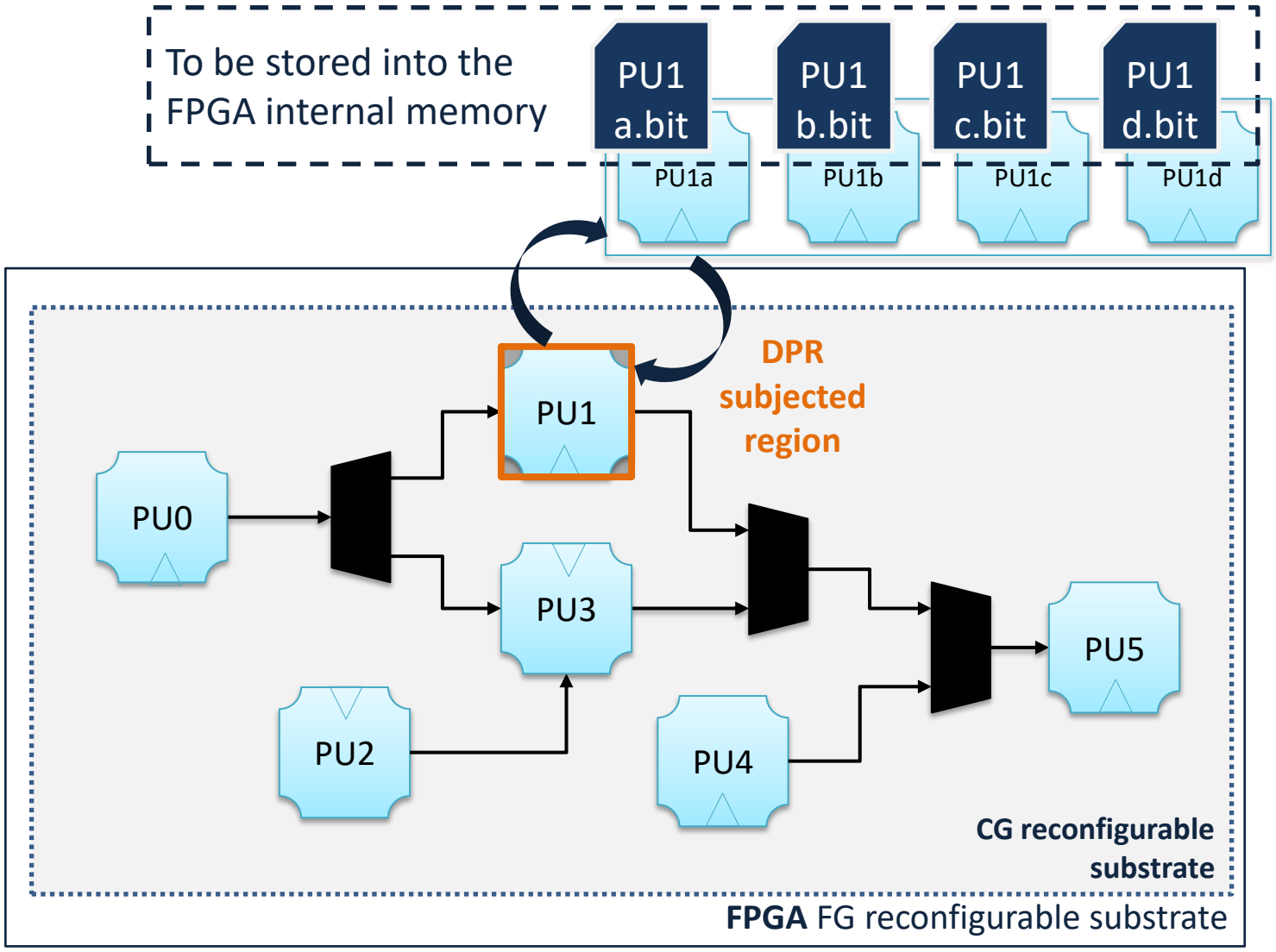
Providing Further Degrees of Reconfigurability

FG into CG reconfiguration



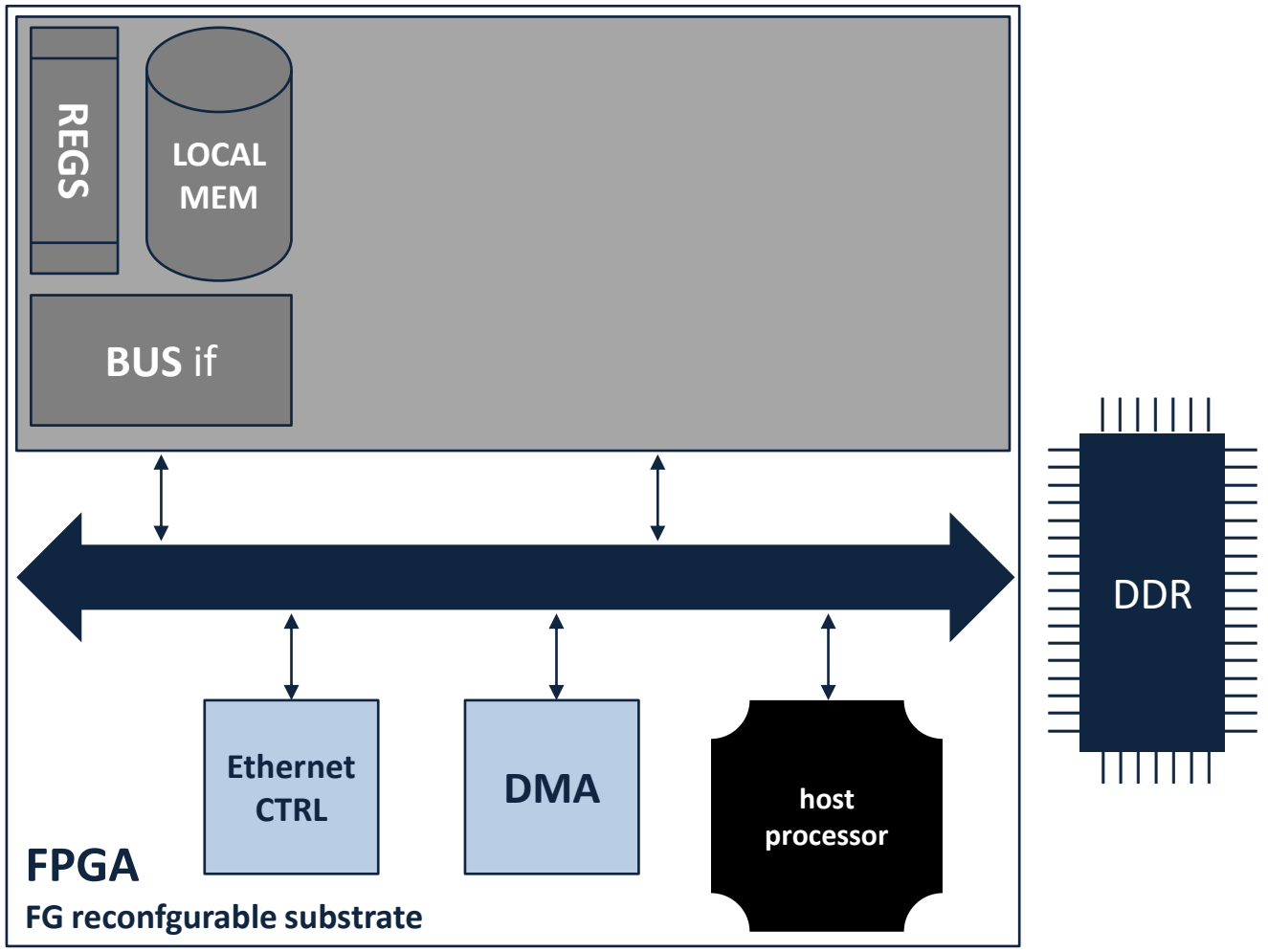
Providing Further Degrees of Reconfigurability

FG into CG reconfiguration



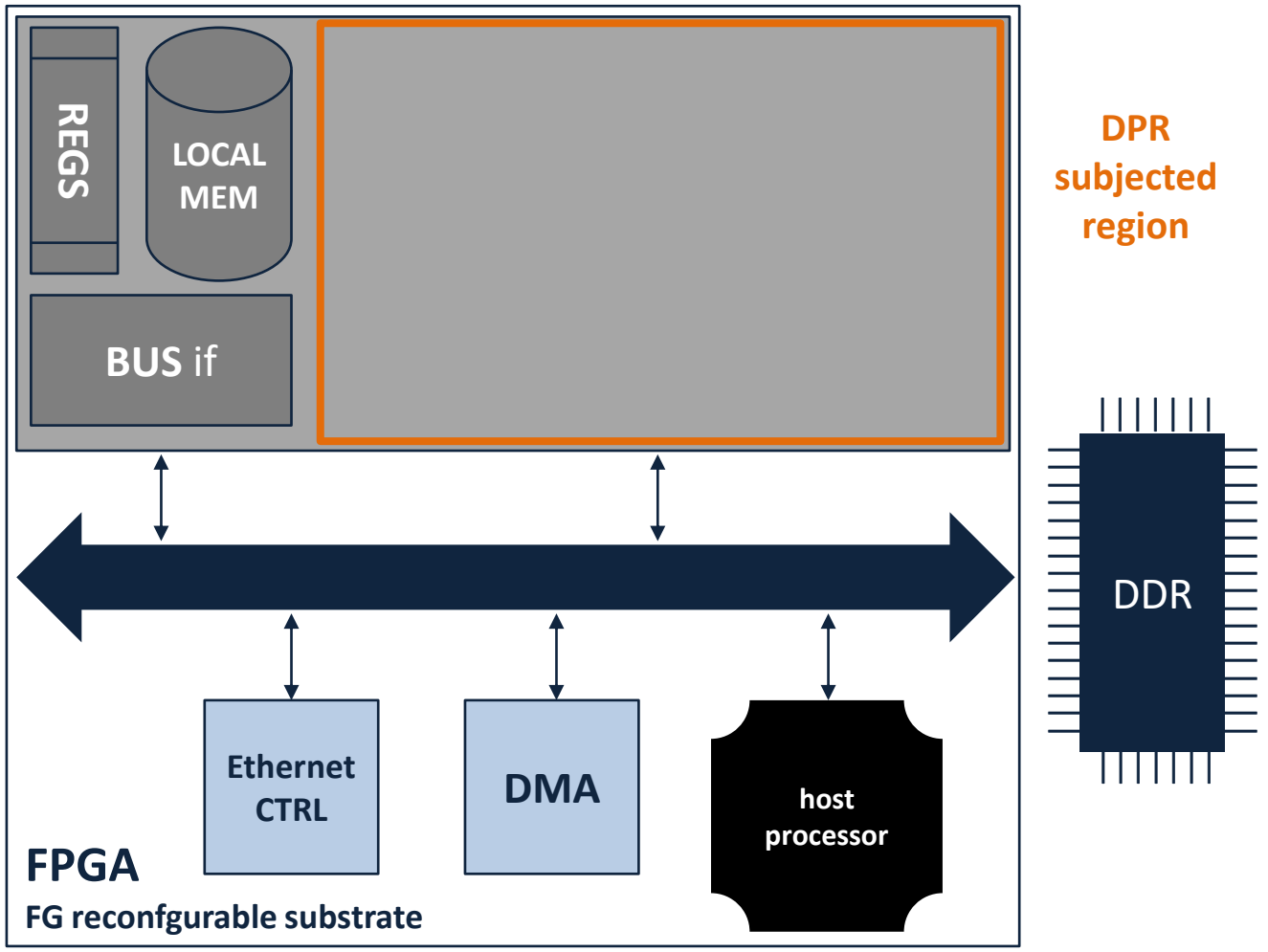
Providing Further Degrees of Reconfigurability

CG into FG reconfiguration



Providing Further Degrees of Reconfigurability

CG into FG reconfiguration

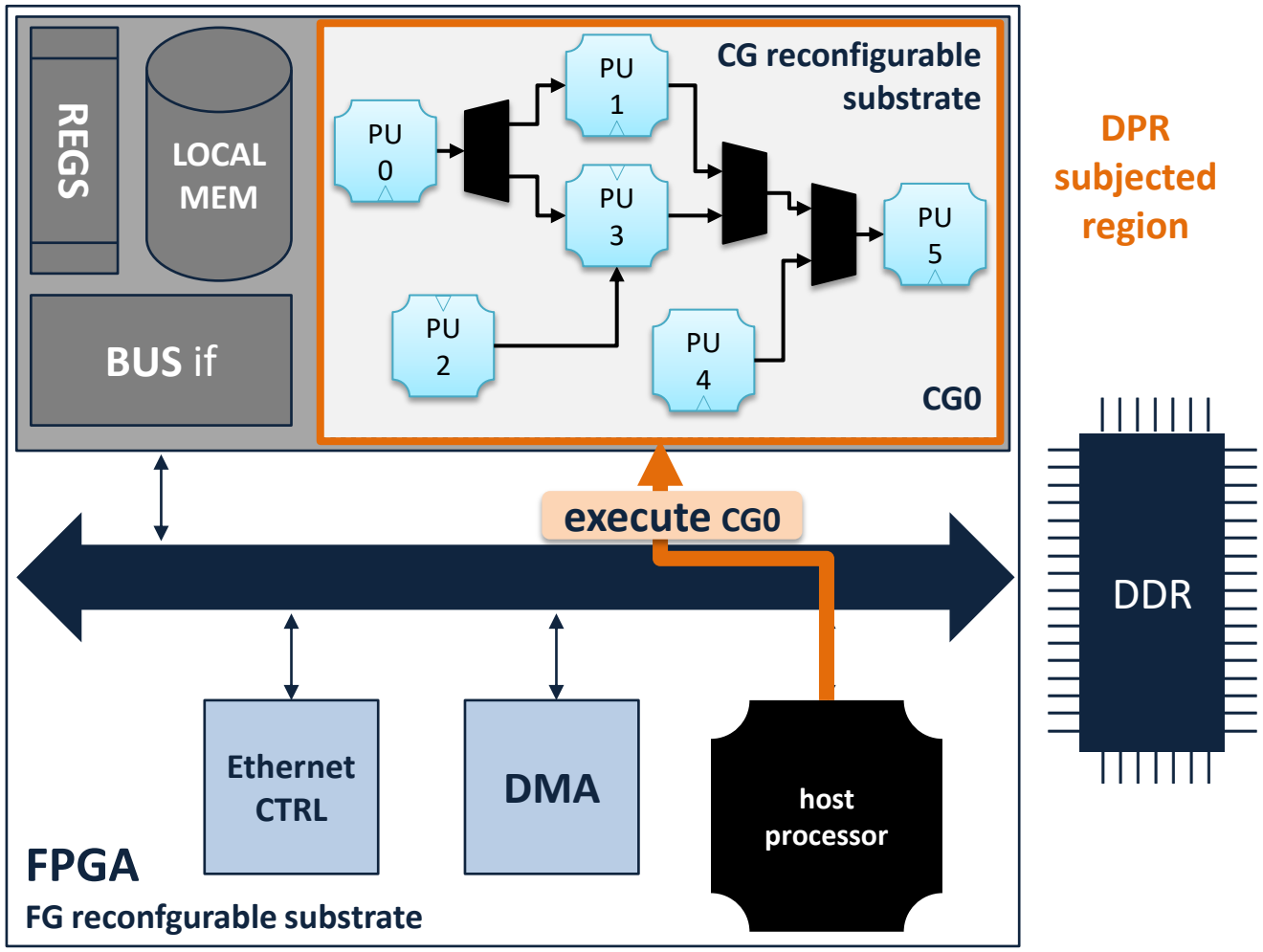


Providing Further Degrees of Reconfigurability

CG into FG reconfiguration

RUNTIME:

t0: config FG = CG0



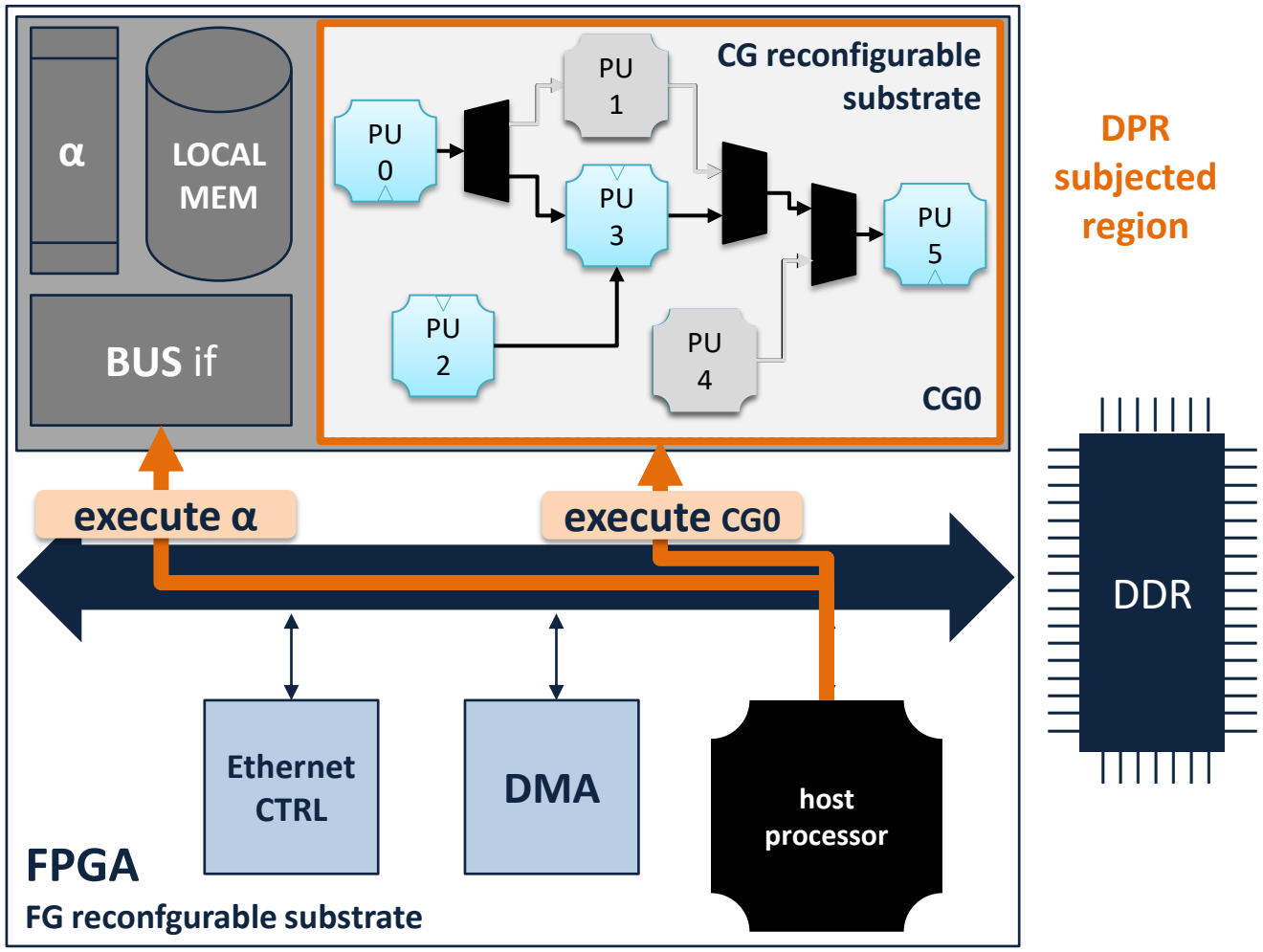
Providing Further Degrees of Reconfigurability

CG into FG reconfiguration

RUNTIME:

t0: config FG = CG0

t1: config CG = α



Providing Further Degrees of Reconfigurability

CG into FG reconfiguration

RUNTIME:

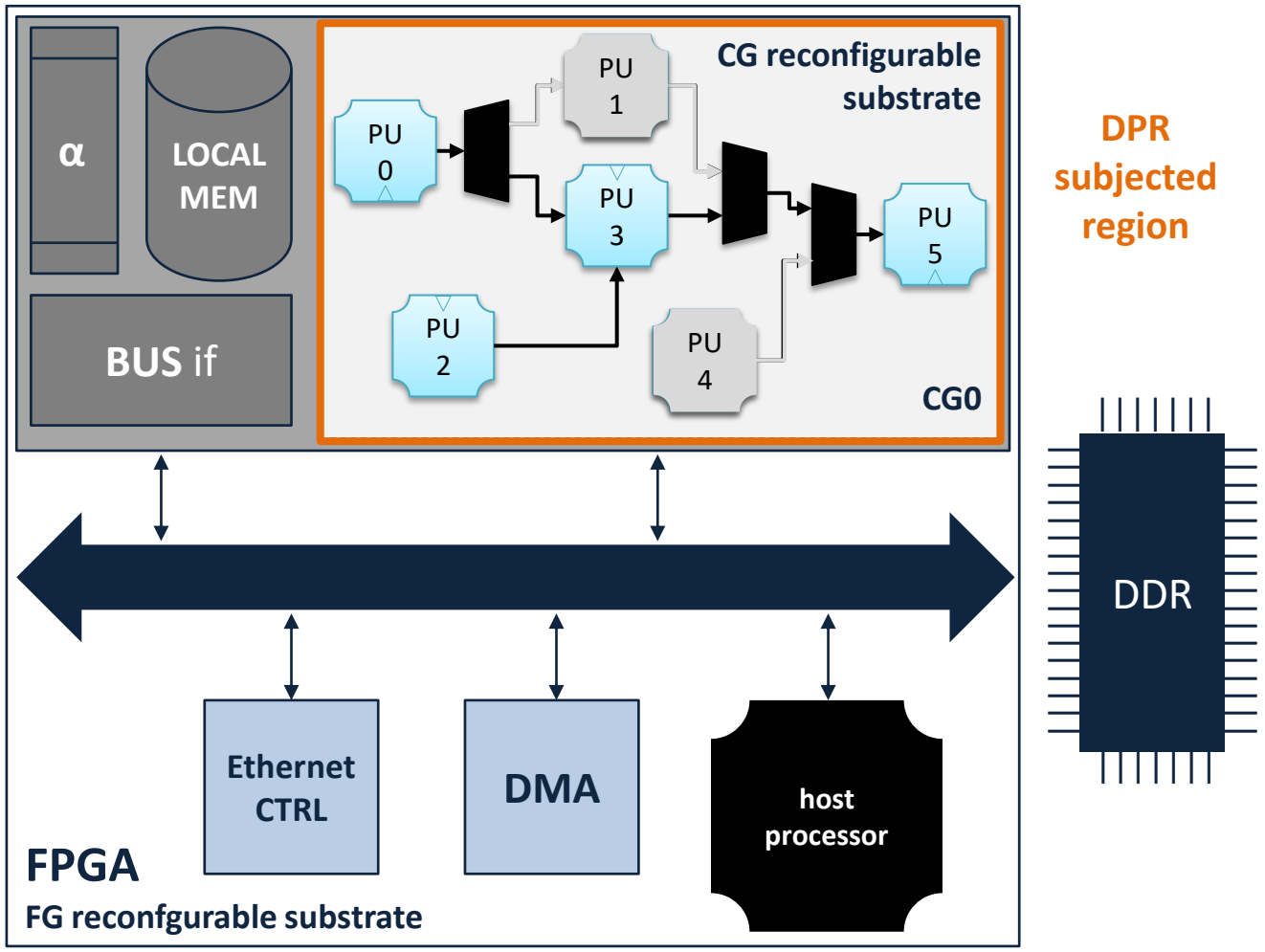
t0: config FG = CG0

t1: config CG = α

…

SMALL context change

…



Providing Further Degrees of Reconfigurability

CG into FG reconfiguration

RUNTIME:

t0: config FG = CG0

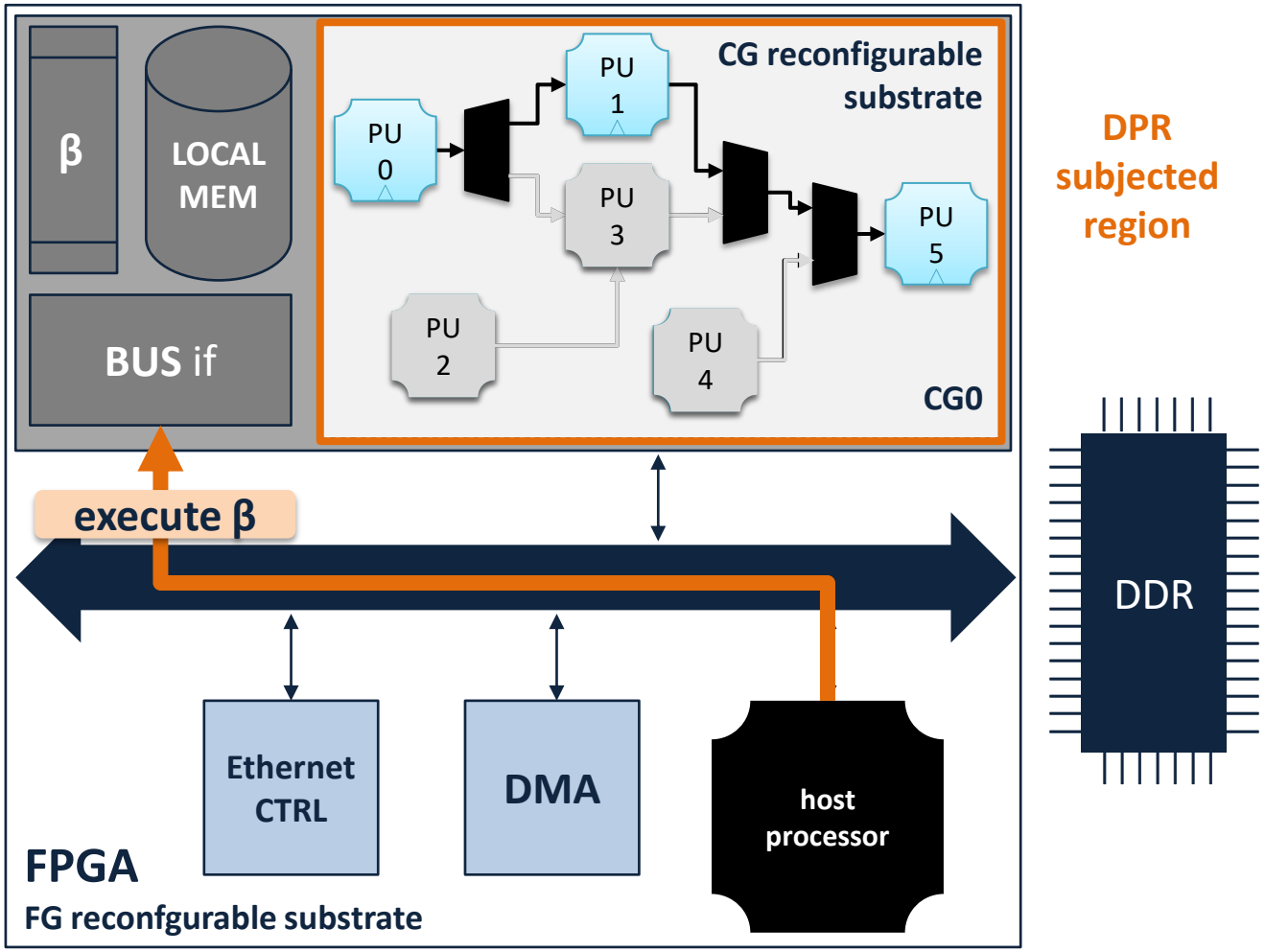
t1: config CG = α

…

SMALL context change

…

t2: config CG = β



Providing Further Degrees of Reconfigurability

CG into FG reconfiguration

RUNTIME:

t0: config FG = CG0

t1: config CG = α

…

SMALL context change

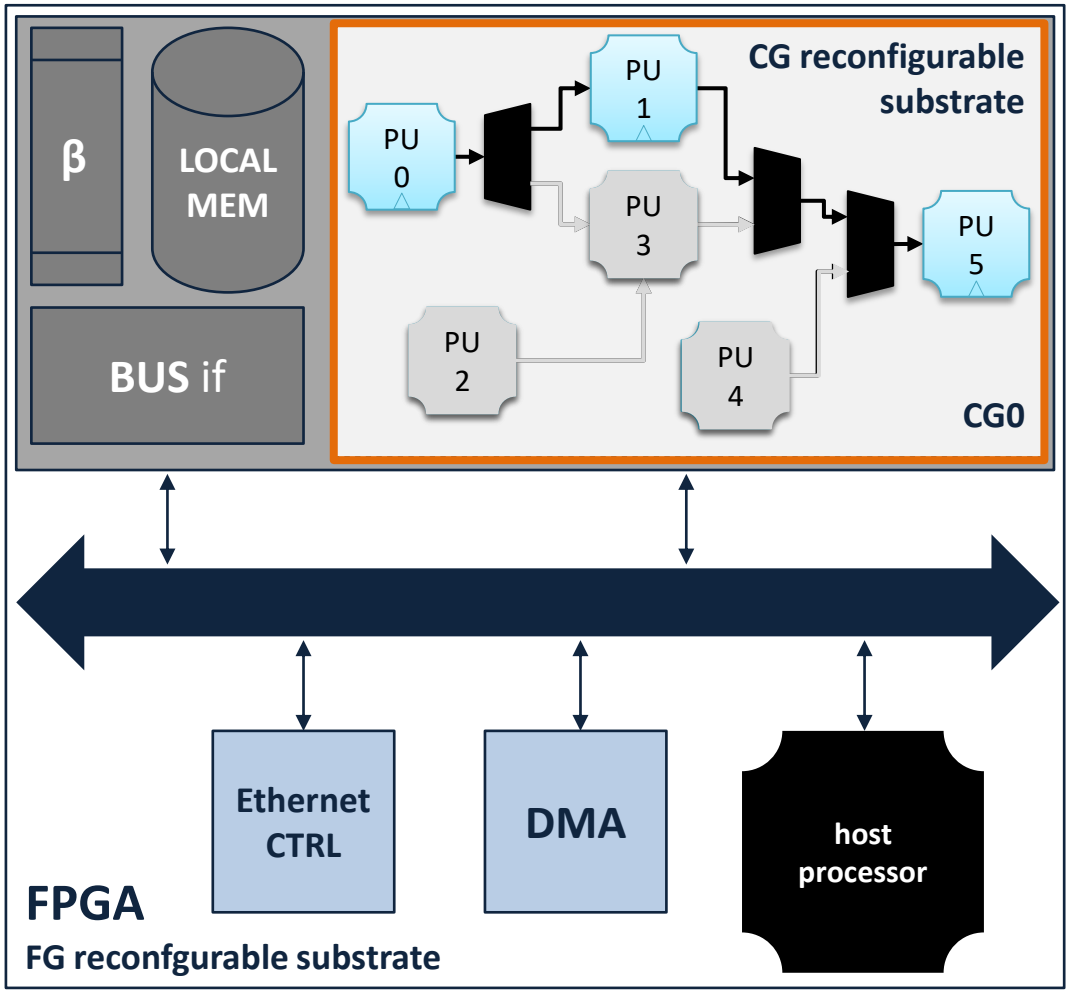
…

t2: config CG = β

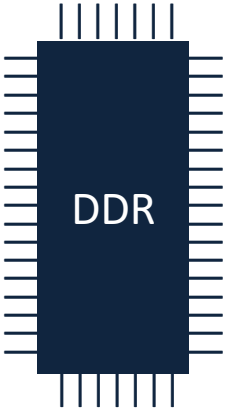
…

SMALL context change

…



DPR subjected region



Providing Further Degrees of Reconfigurability

CG into FG reconfiguration

RUNTIME:

t0: config FG = CG0

t1: config CG = α

…

SMALL context change

…

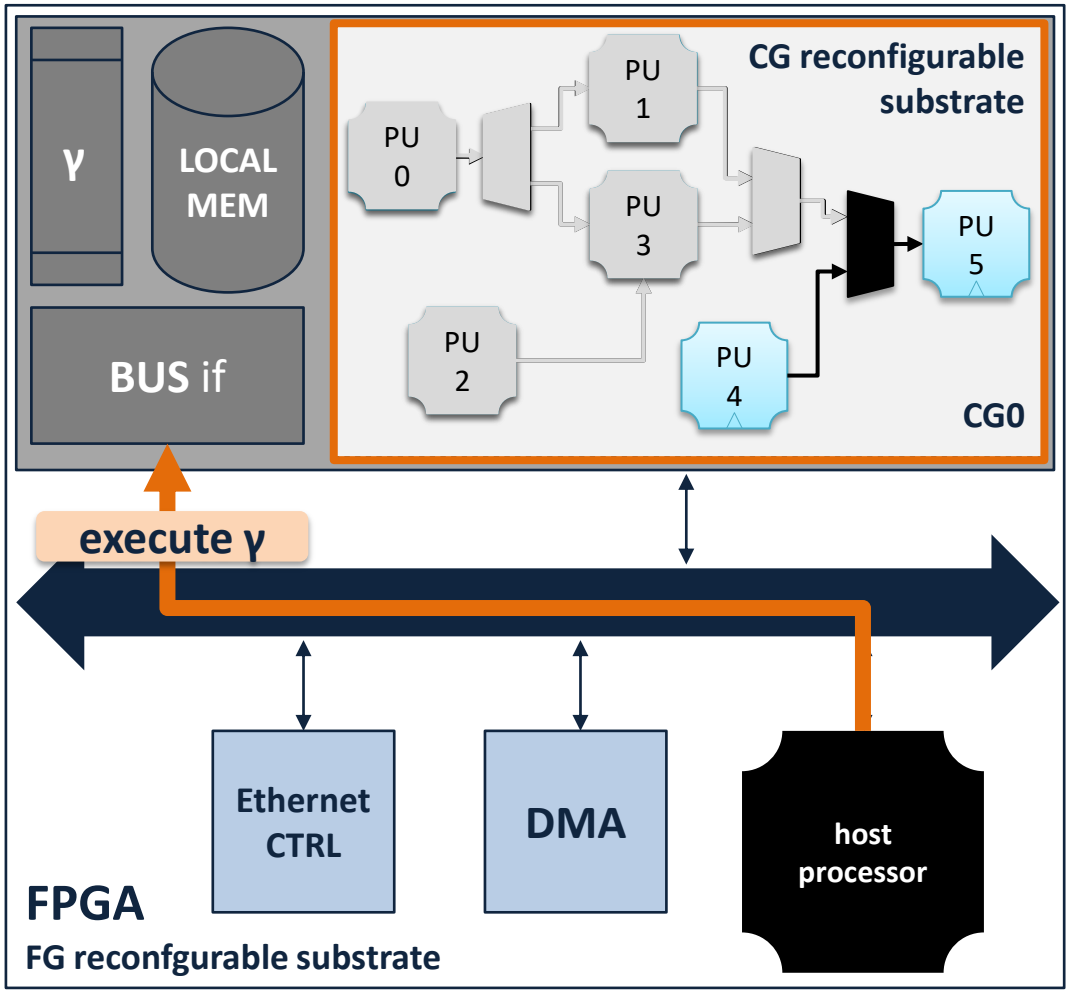
t2: config CG = β

…

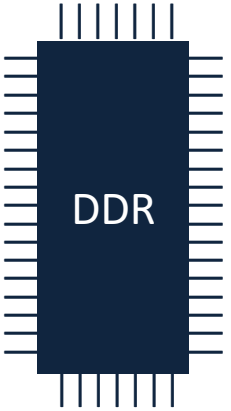
SMALL context change

…

t3: config CG = γ



DPR subjected region



Providing Further Degrees of Reconfigurability

CG into FG reconfiguration

RUNTIME:

t0: config FG = CG0

t1: config CG = α

…

SMALL context change

…

t2: config CG = β

…

SMALL context change

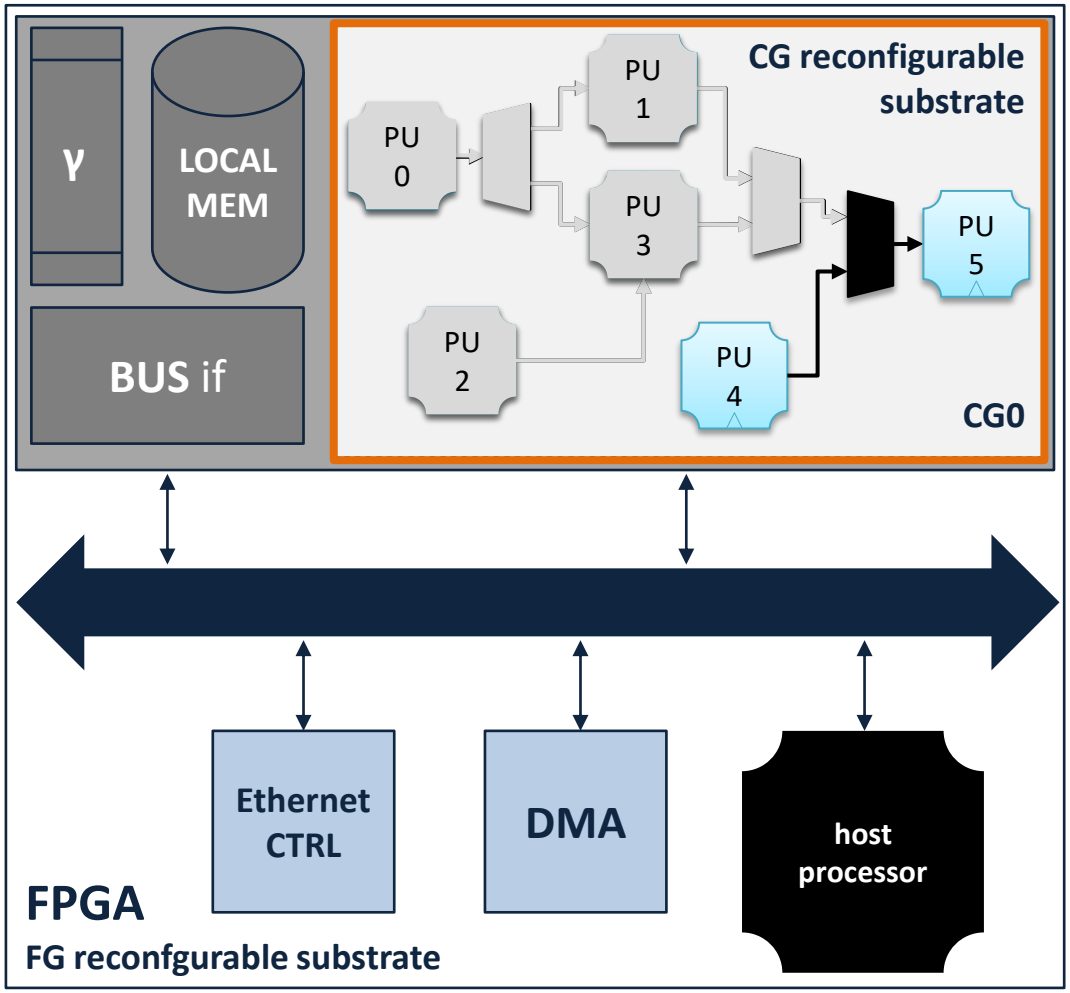
…

t3: config CG = γ

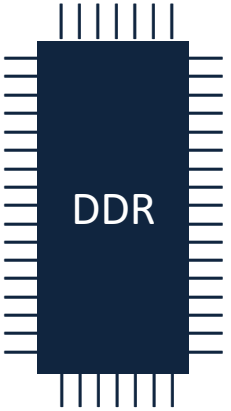
…

BIG context change

…



DPR subjected region



Providing Further Degrees of Reconfigurability

CG into FG reconfiguration

RUNTIME:

t0: config FG = CG0

t1: config CG = α

…

SMALL context change

…

t2: config CG = β

…

SMALL context change

…

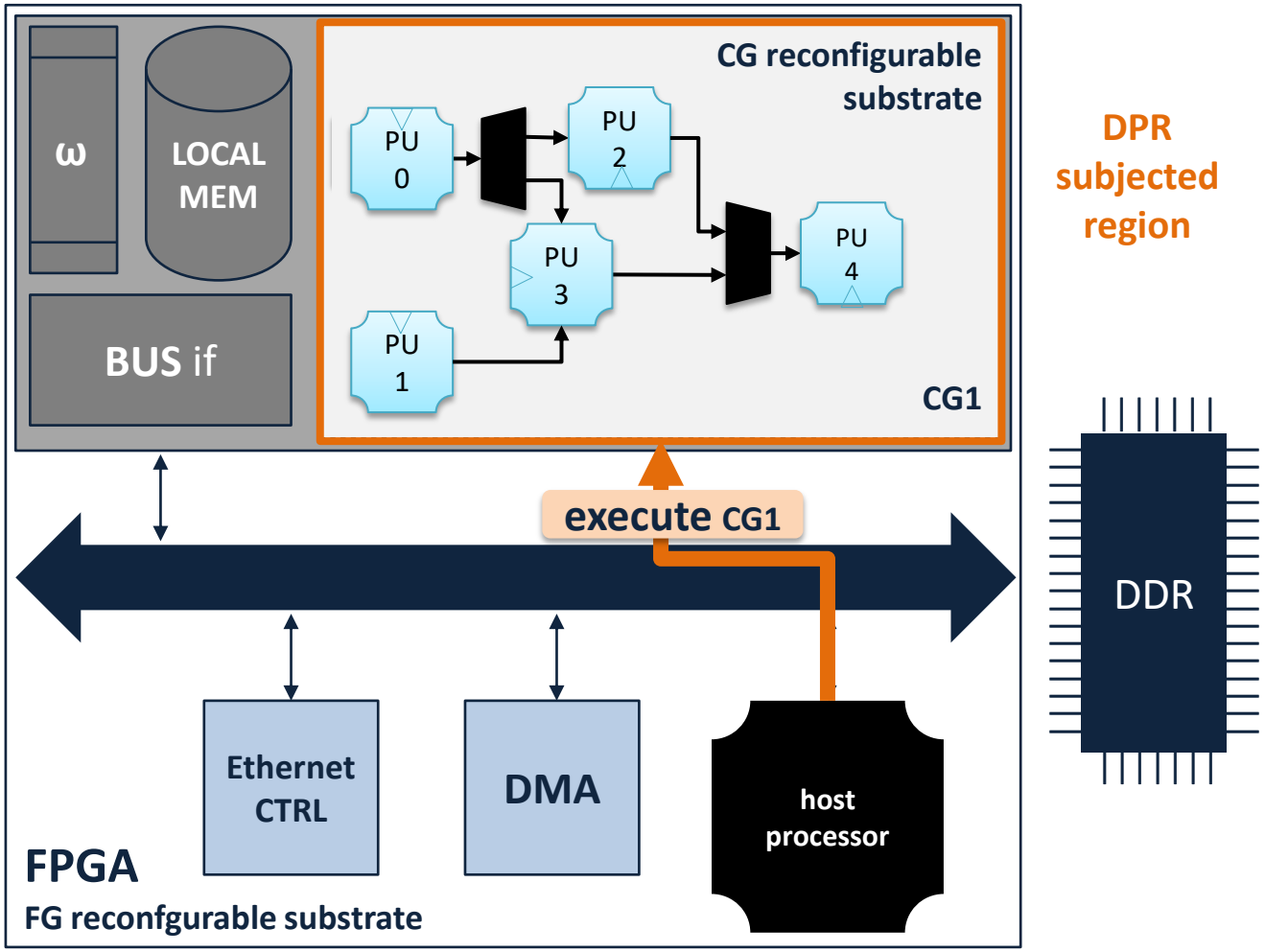
t3: config CG = γ

…

BIG context change

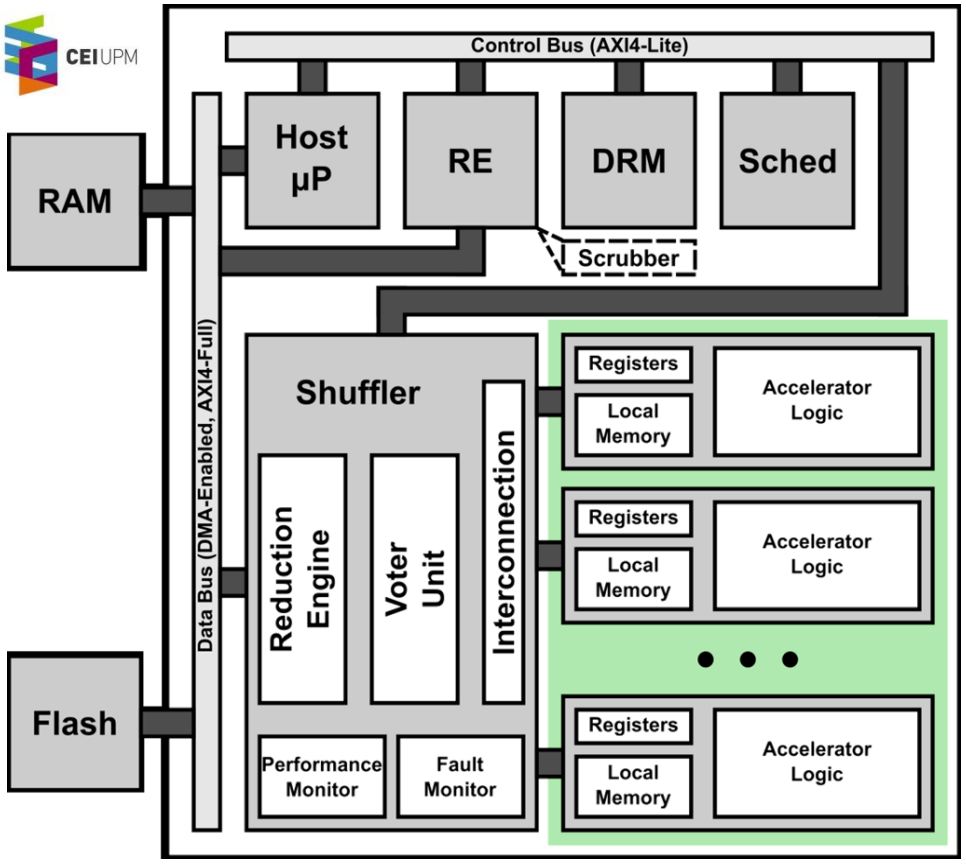
…

t4: config FG = CG1



Providing Further Degrees of Reconfigurability

CG into Artico^3

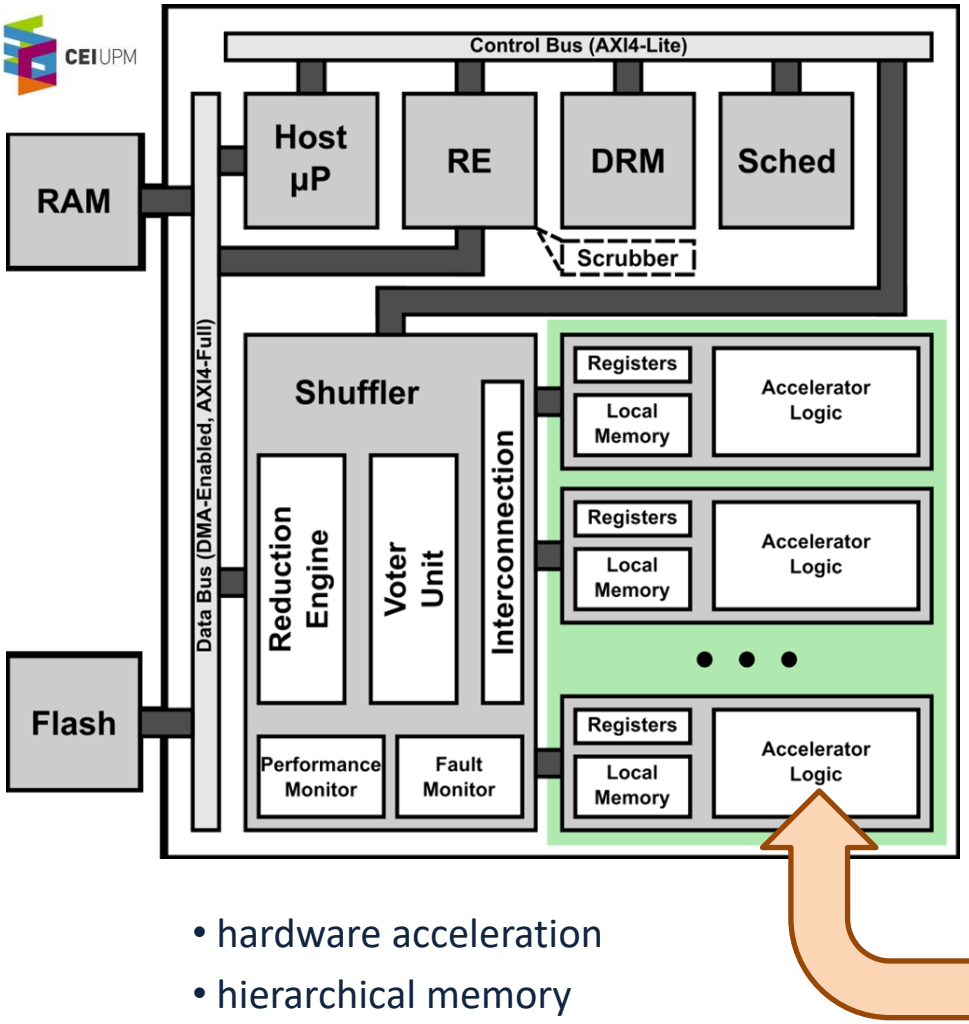


Artico3 is a **DPR** supporting architecture in charge of smartly manage **performance**, **consumption** and **dependability**.

- hardware acceleration
- hierarchical memory
- bus based, DMA enabled communication

Providing Further Degrees of Reconfigurability

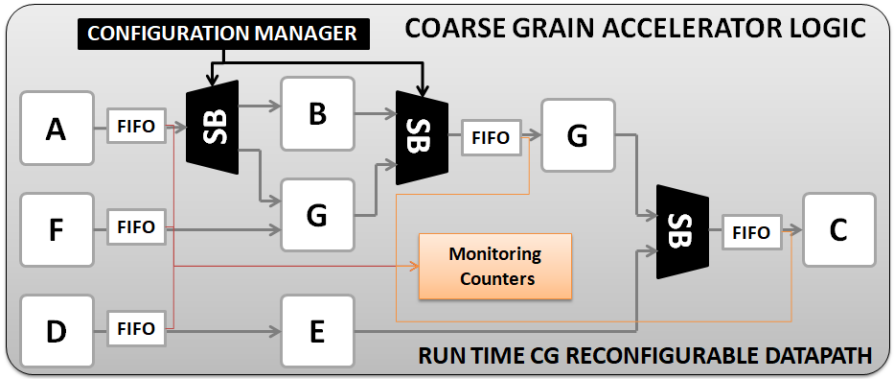
CG into Artico^3



Artico3 is a **DPR** supporting architecture in charge of smartly manage **performance**, **consumption** and **dependability**.

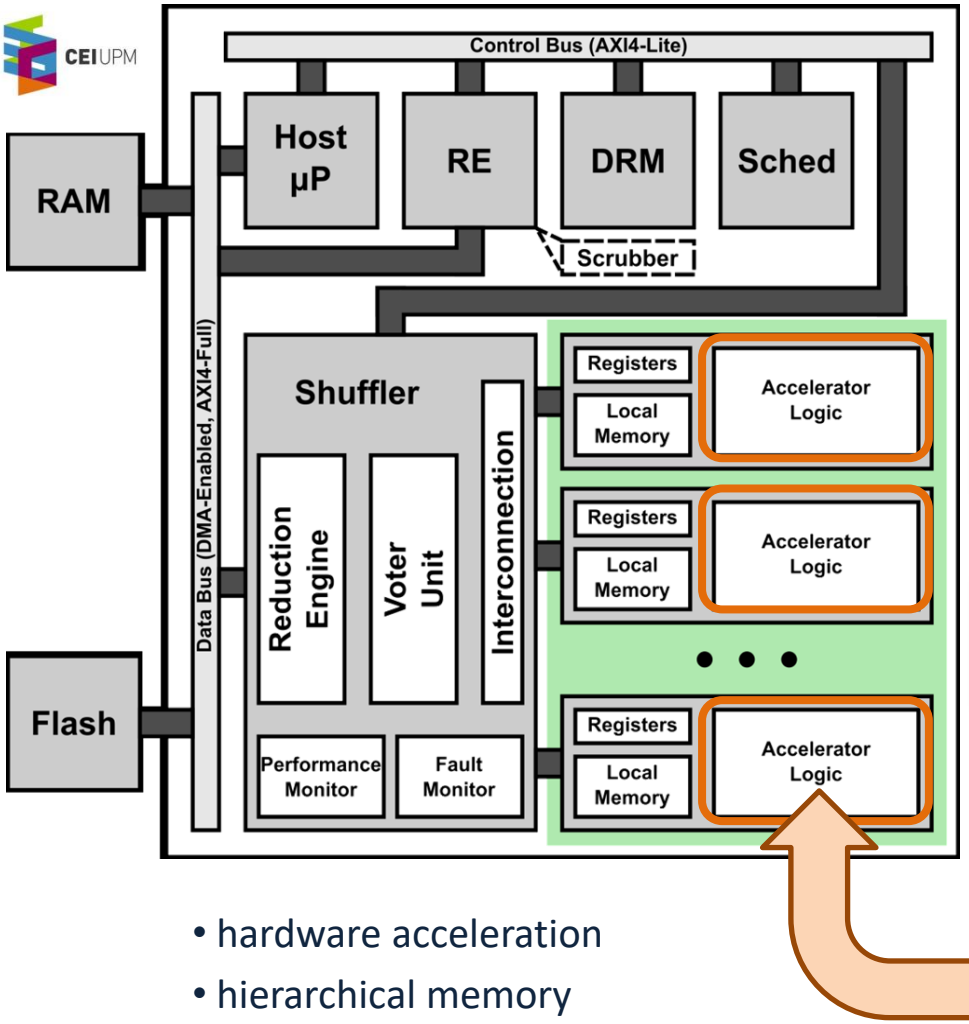
- enhance **flexibility** by enabling CGR within Artico3 slots

- hardware acceleration
- hierarchical memory
- bus based, DMA enabled communication



Providing Further Degrees of Reconfigurability

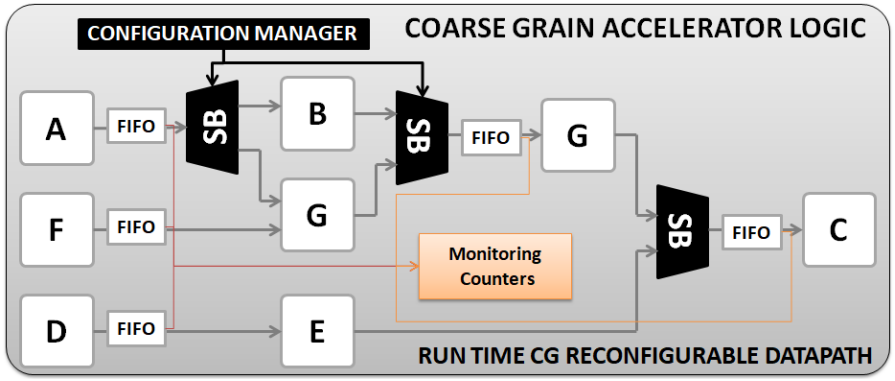
CG into Artico^3



- hardware acceleration
- hierarchical memory
- bus based, DMA enabled communication

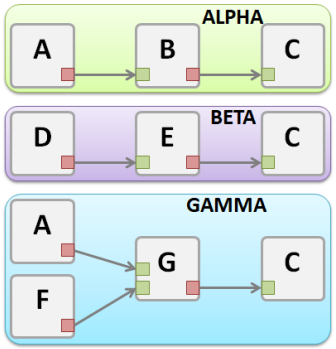
Artico3 is a **DPR** supporting architecture in charge of smartly manage **performance**, **consumption** and **dependability**.

- enhance **flexibility** by enabling CGR within Artico3 slots
- exploit dataflow to **facilitate/ automate programmability**



Providing Further Degrees of Reconfigurability

The big picture within CERBERO



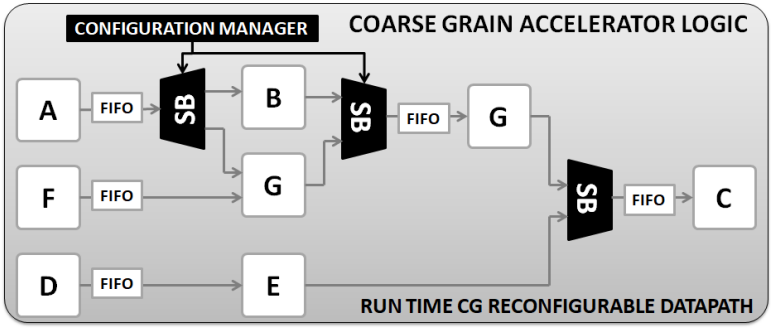
N inputs

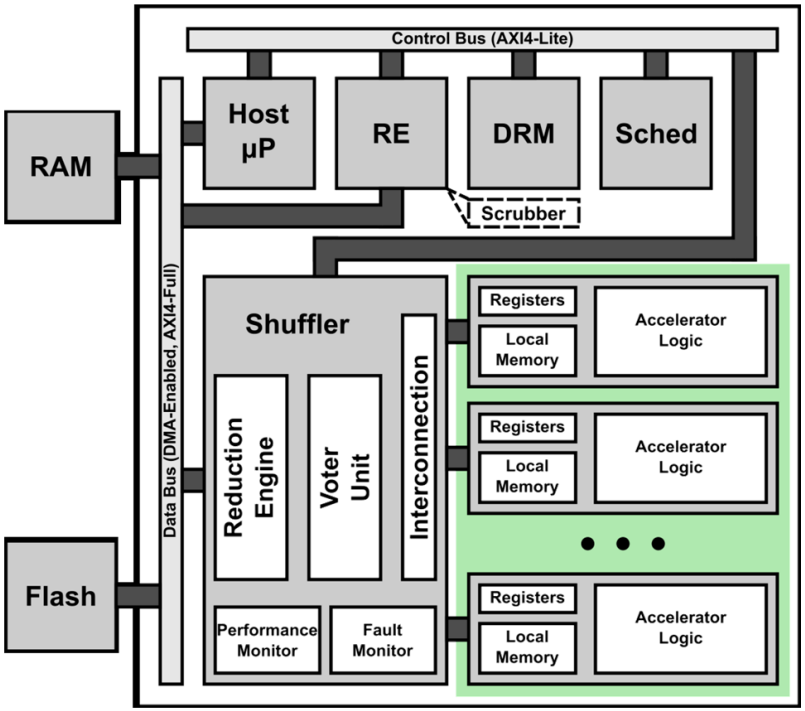
PREESM - N:N

+

MDC - N:1

1 output

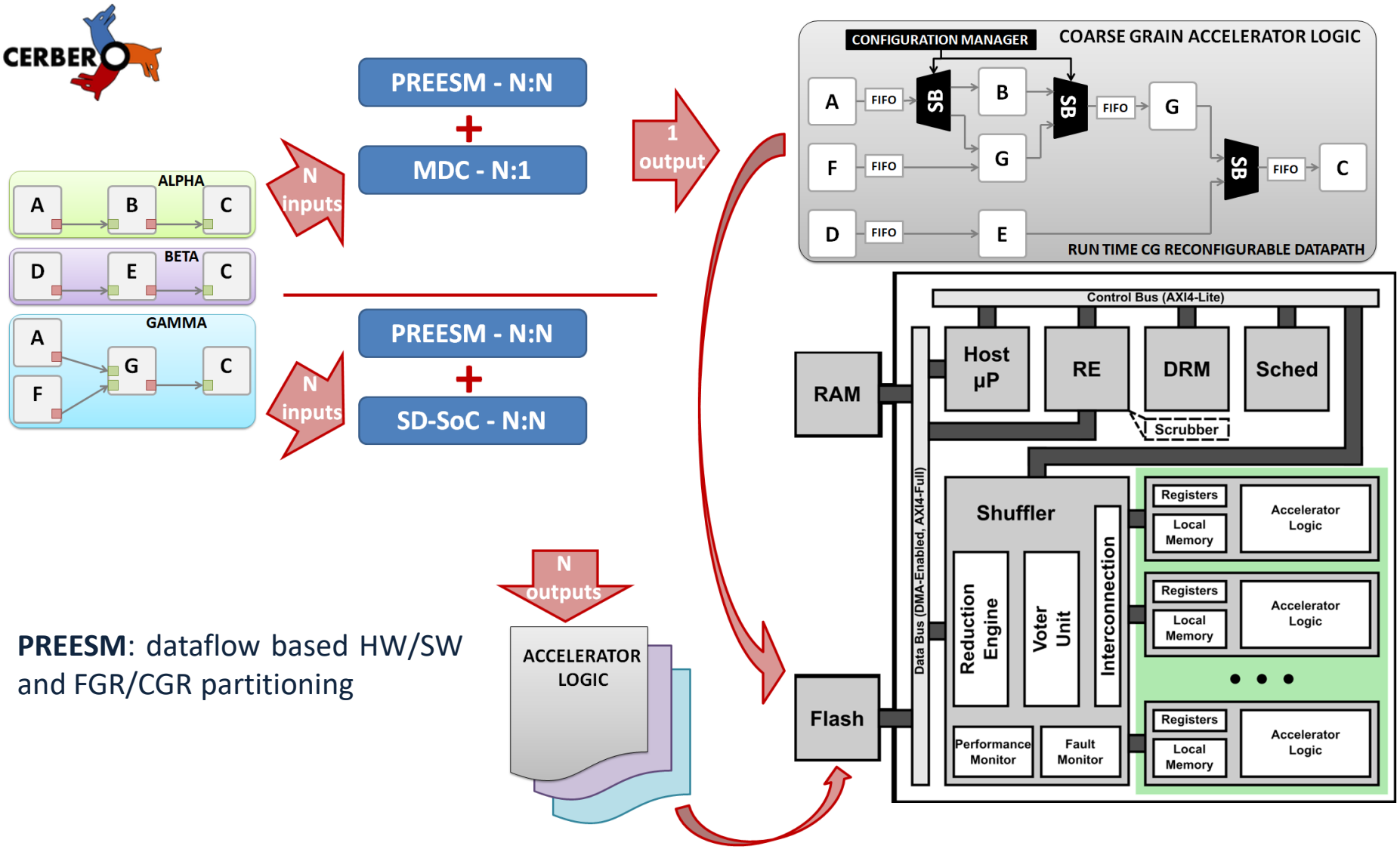




PREESM: dataflow based HW/SW and FGR/CGR partitioning

Providing Further Degrees of Reconfigurability

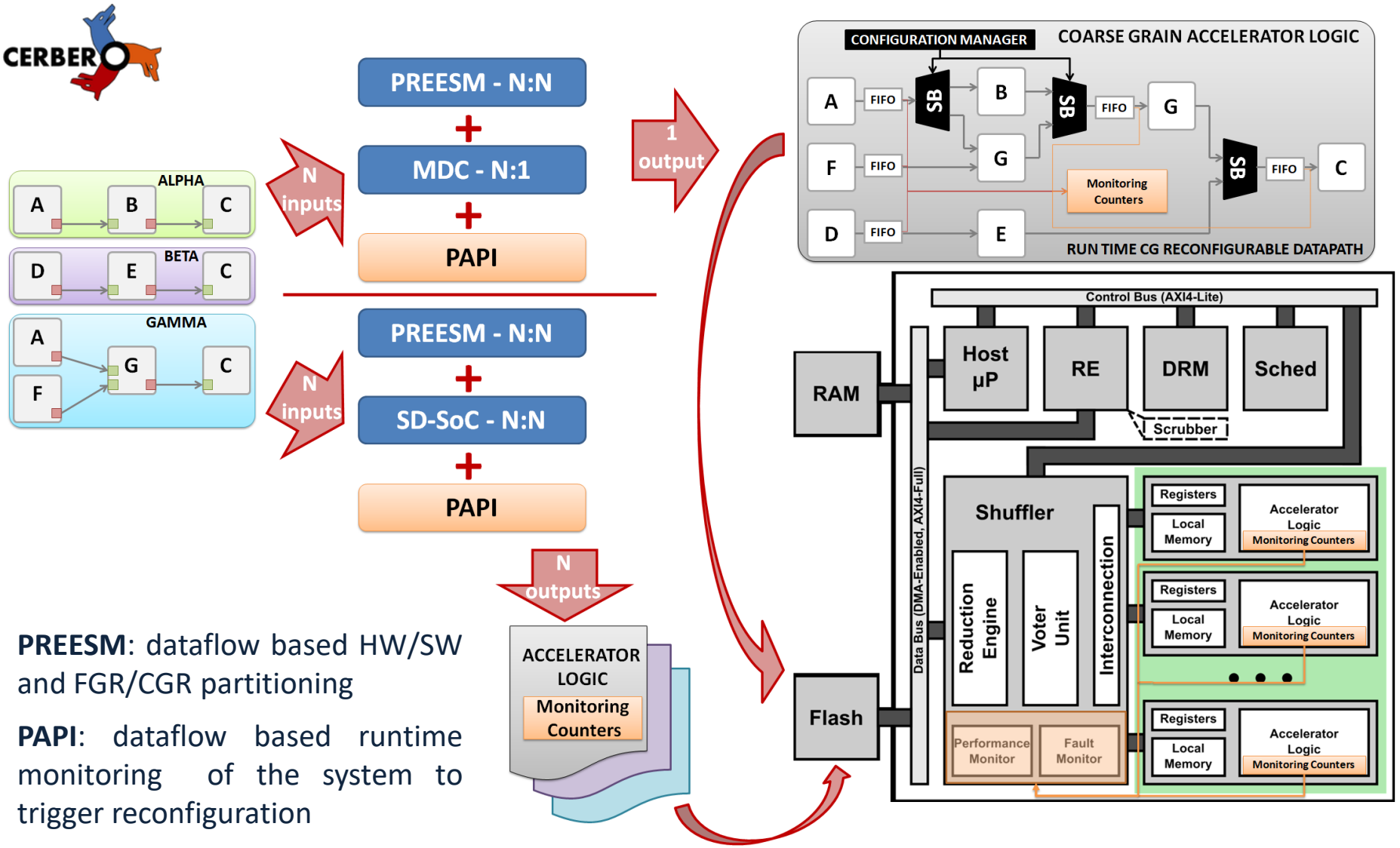
The big picture within CERBERO



PREESM: dataflow based HW/SW and FGR/CGR partitioning

Providing Further Degrees of Reconfigurability

The big picture within CERBERO

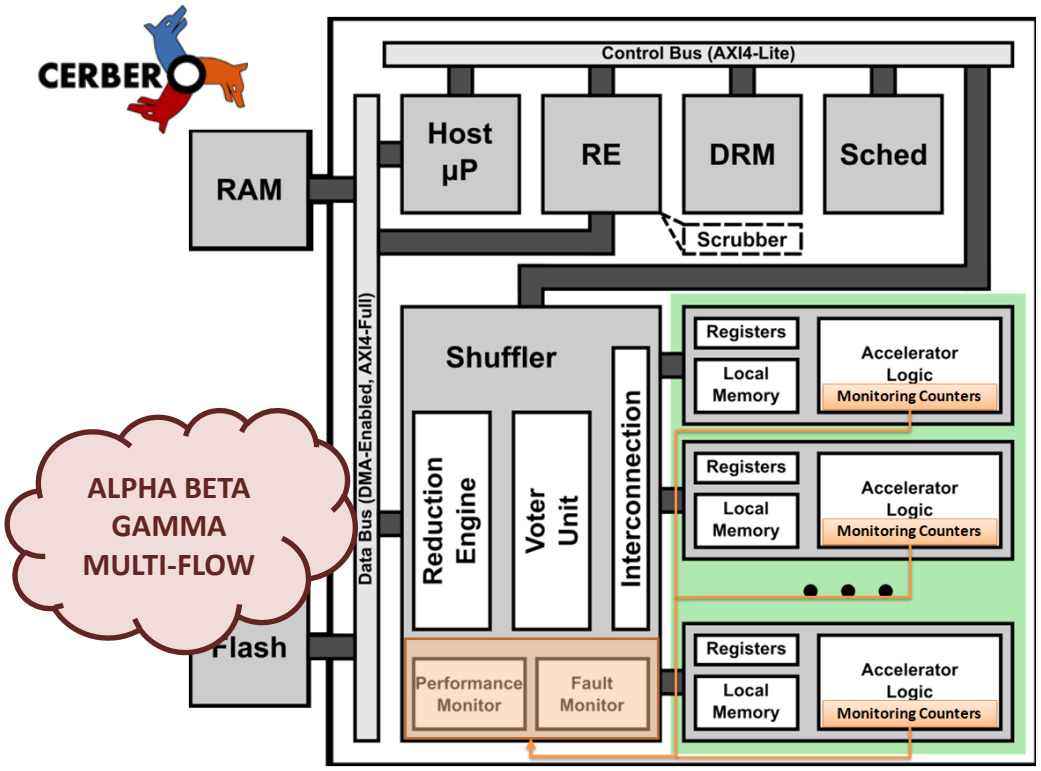


PREESM: dataflow based HW/SW and FGR/CGR partitioning

PAPI: dataflow based runtime monitoring of the system to trigger reconfiguration

Providing Further Degrees of Reconfigurability

The big picture within CERBERO

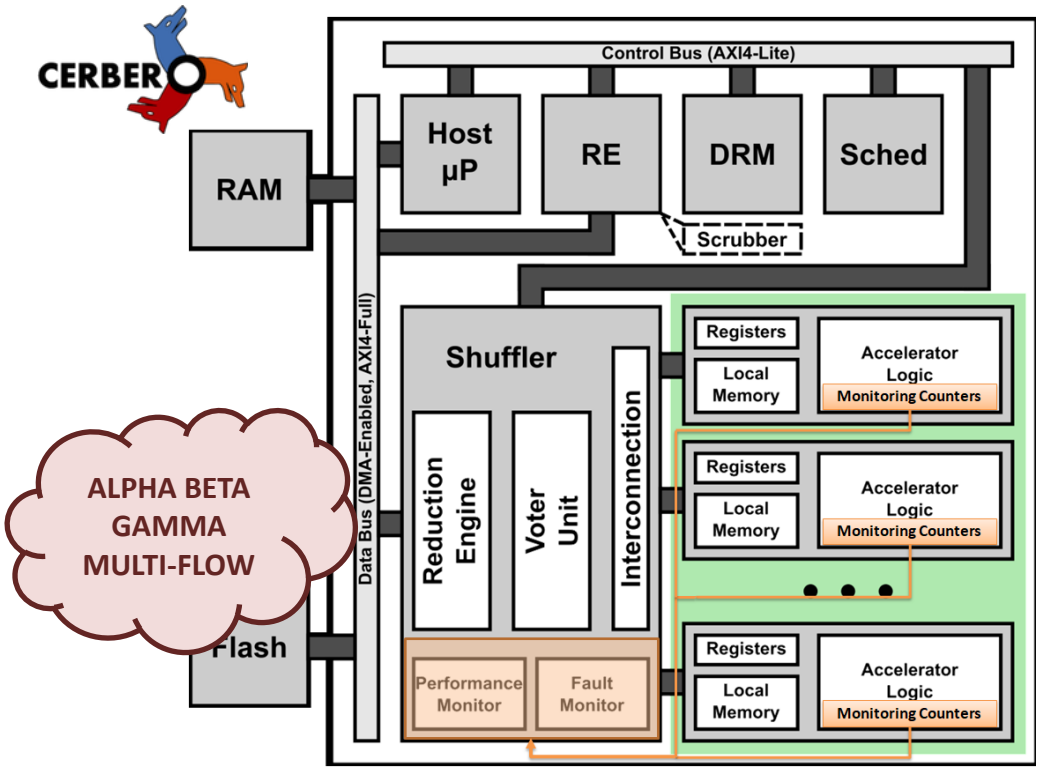


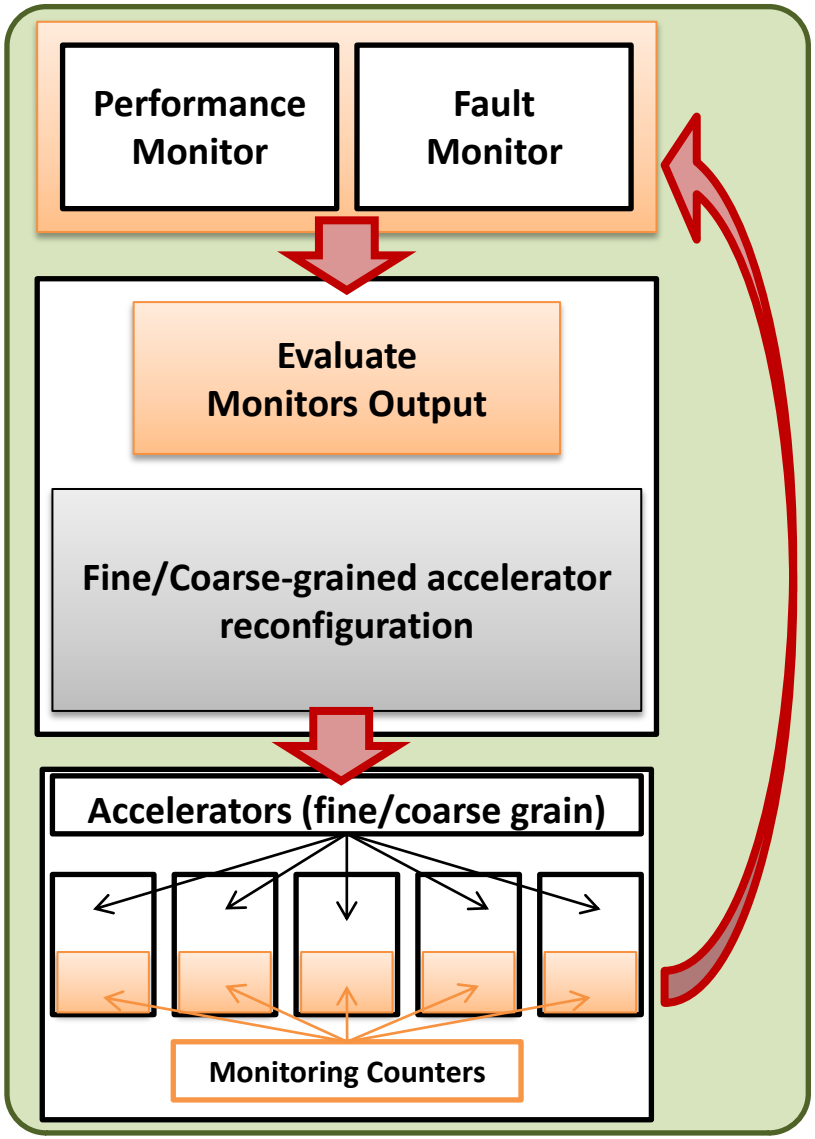
PREESM: dataflow based HW/SW and FGR/CGR partitioning

PAPI: dataflow based runtime monitoring of the system to trigger reconfiguration

Providing Further Degrees of Reconfigurability

The big picture within CERBERO





PREESM: dataflow based HW/SW and FGR/CGR partitioning

PAPI: dataflow based runtime monitoring of the system to trigger reconfiguration

Thanks To …

EU Commission for funding the ***CERBERO*** (*Cross-layer modEl-based fRamework for multi-oBjective dEsign of Reconfigurable systems in unceRtain hybRid envirOnments*) project as part of the H2020 Programme under grant agreement No 732105.

Coordinator:

Michal Masin (IBM), michaelm@il.ibm.com

Scientific Coordinator:

Francesca Palumbo (UniSS), fpalumbo@uniss.it

Innovation Manager:

Katiuscia Zedda (Abinsula), katiuscia.zedda@abinsula.com

Dissemination-Communication Manager:

Francesco Regazzoni (USI), francesco.regazzoni@usi.ch

www.cerbero-h2020.eu

info@cerbero-h2020.eu

@CERBERO_h2020

CERBERO

The Future Directions of Dataflow-Based Reconfigurable Hardware Accelerators

Francesca Palumbo, Claudio Rubattu,
Carlo Sau, Tiziana Fanni, Luigi Raffo

INSA

Rennes, 12-14 December 2017