Conference on Design & Architectures for Signal & Image Processing
October 8-10, 2014
Madrid, Spain

Automatic Generation of Dataflow-Based Reconfigurable Co-processing Units

Francesca Palumbo
Università degli Studi di Sassari
PolComIng – Information Eng. Unit

Carlo Sau
Università degli Studi di Cagliari
DIEE – Dept. of Electrical and Electronics Eng.

OUTLINE

- Introduction:
  - Problem statement
  - Background
  - Goals
- Co-processing units generation:
  - Approach and baseline Multi-Dataflow Composer
  - Template Interface Layer: hardware and automatic composition
- Performance assessment
  - Use-case scenario
  - Results
- Final remarks and future directions

OUTLINE: INTRODUCTION

- Introduction:
  - Problem statement
  - Background
  - Goals
- Co-processing units generation:
  - Approach and baseline Multi-Dataflow Composer
  - Template Interface Layer: hardware and automatic composition
- Performance assessment
  - Use-case scenario
  - Results
- Final remarks and future directions

PROBLEM STATEMENT

CONSUMER NEEDS

- **HIGH PERFORMANCES** real time applications:
  - Media players, video calling...
- **UP-TO-DATE SOLUTIONS**
  - Support for the last audio/video codecs, file formats...
- **MORE INTEGRATED FEATURES** in mobile devices:
  - MP3, Camera, Video, GPS...
- **PORTABILITY**
- **LONG BATTERY LIFE**
  - Convenient form factor, affordable price...

PROBLEM STATEMENT

CONSUMER NEEDS

- **HIGH PERFORMANCES** real time applications:
  - Media players, video calling...
- **UP-TO-DATE SOLUTIONS**
  - Support for the last audio/video codecs, file formats...
- **MORE INTEGRATED FEATURES** in mobile devices:
  - MP3, Camera, Video, GPS...
- **PORTABILITY**
- **LONG BATTERY LIFE**
  - Convenient form factor, affordable price...

POSSIBLE SOLUTION

- **DATAFLOW MODEL OF COMPUTATION**
  - Modularity and parallelism → **EASIER INTEGRATION AND FAVOURED RE-USABILITY**
- **COARSE-GRAINED RECONFIGURABILITY**
  - Flexibility and resource sharing → **MULTI-APPLICATION PORTABLE DEVICES**

PROBLEM STATEMENT

CONSUMER NEEDS

- **HIGH PERFORMANCES** real time applications:
  - Media players, video calling...
- **UP-TO-DATE SOLUTIONS**
  - Support for the last audio/video codecs, file formats
  - Convenient form factor, affordable price...

Automated DESIGN FLOW are fundamental to guarantee SHORTER TIME-TO-MARKET. Dealing with APPLICATION SPECIFIC MULTI-CONTEXT systems, in particular for KERNEL ACCELERATORS, state of the art still lacks in providing a broadly accepted solution.

POSSIBLE SOLUTION

- **DATAFLOW MODEL OF COMPUTATION**
  - Modularity and parallelism → **EASIER INTEGRATION AND FAVOURED RE-USABILITY**
- **COARSE-GRAINED RECONFIGURABILITY**
  - Flexibility and resource sharing → **MULTI-APPLICATION PORTABLE DEVICES**

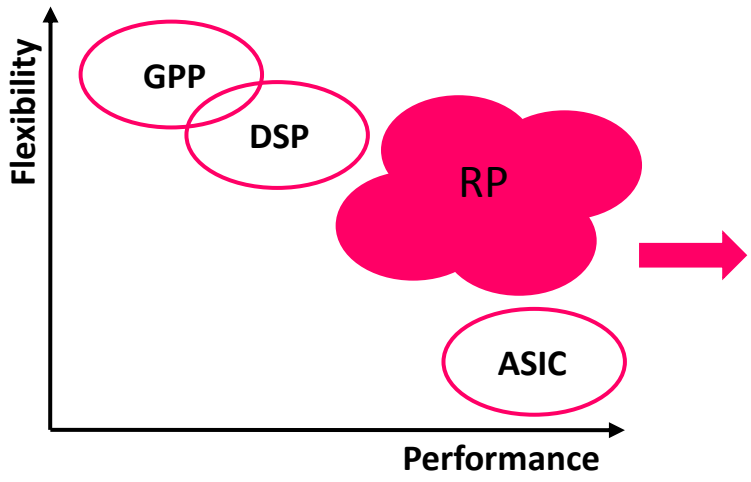
BACKGROUND: CG RECONFIGURABILITY

FINE- GRAINED (FG) ACCELERATORS

- High flexibility bit-level reconfiguration
- Slow and memory expensive configuration phase

COARSE-GRAINED (CG) ACCELERATORS

- Medium flexibility word-level reconfiguration
- Fast configuration phase



	FG	CG
	Bit-level	Word-level
Flexibility	☺	😐
Reconf. Speed	😐	☺
Config. Storage	☹	😐

BACKGROUND: CG RECONFIGURABILITY

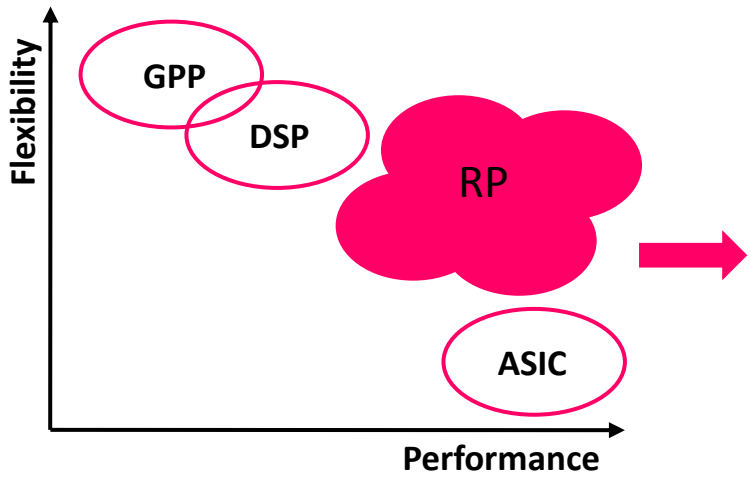
AUTOMATIC GENERATION

FINE- GRAINED (FG) ACCELERATORS → VIVADO (XILINX) / NIOS II (ALTERA)

- High flexibility bit-level reconfiguration
- Slow and memory expensive configuration phase

COARSE-GRAINED (CG) ACCELERATORS → ???

- Medium flexibility word-level reconfiguration
- Fast configuration phase



	FG	CG
	Bit-level	Word-level
Flexibility	☺	😐
Reconf. Speed	😐	☺
Config. Storage	☹	😐

BACKGROUND: CG RECONFIGURABILITY

AUTOMATIC GENERATION

FINE- GRAINED (FG) ACCELERATORS

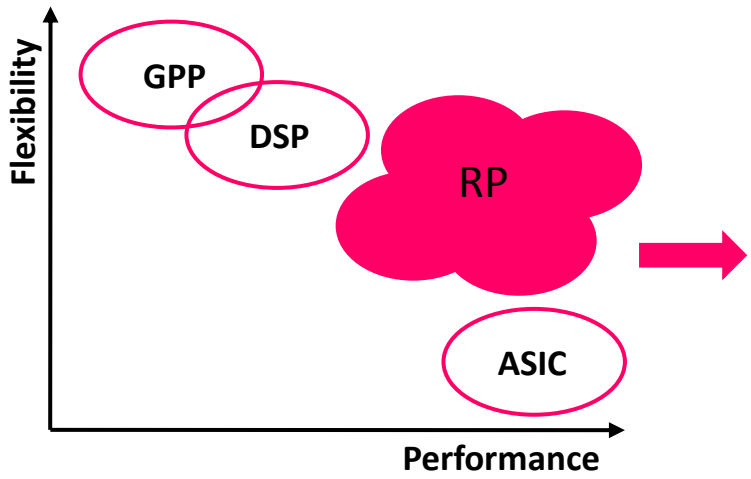
- High flexibility bit-level reconfiguration
- Slow and memory expensive configuration phase

→ VIVADO (XILINX)
NIOS II (ALTERA)

COARSE-GRAINED (CG) ACCELERATORS

- Medium flexibility word-level reconfiguration
- Fast configuration phase

→ MULTI-CONTEXT



	FG	CG
	Bit-level	Word-level
Flexibility	☺	😐
Reconf. Speed	😐	☺
Config. Storage	☹	😐

BACKGROUND: THE DATAFLOW MOC

DATAFLOW PROGRAM

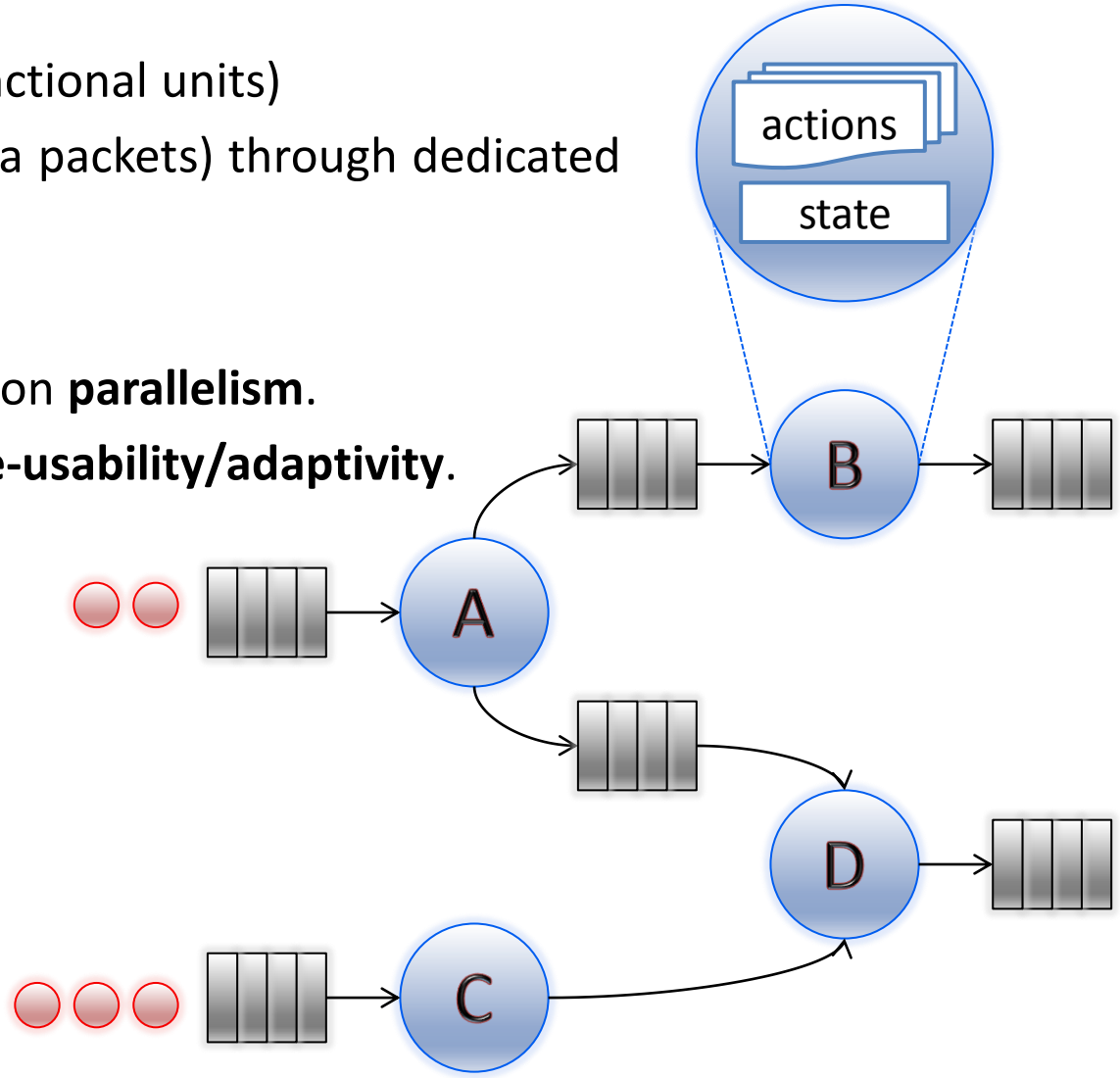
- Directed graph of **actors** (functional units)
- Actors exchange **tokens** (data packets) through dedicated channels

PECULIARITIES

- Explicit the intrinsic application **parallelism**.
- **Modularity** favours model **re-usability/adaptivity**.

EXTERNAL INTERFACE

- I/O ports **number**
- I/O ports **depth**
- I/O ports **burst** of tokens



BACKGROUND: THE DATAFLOW MOC

DATAFLOW PROGRAM

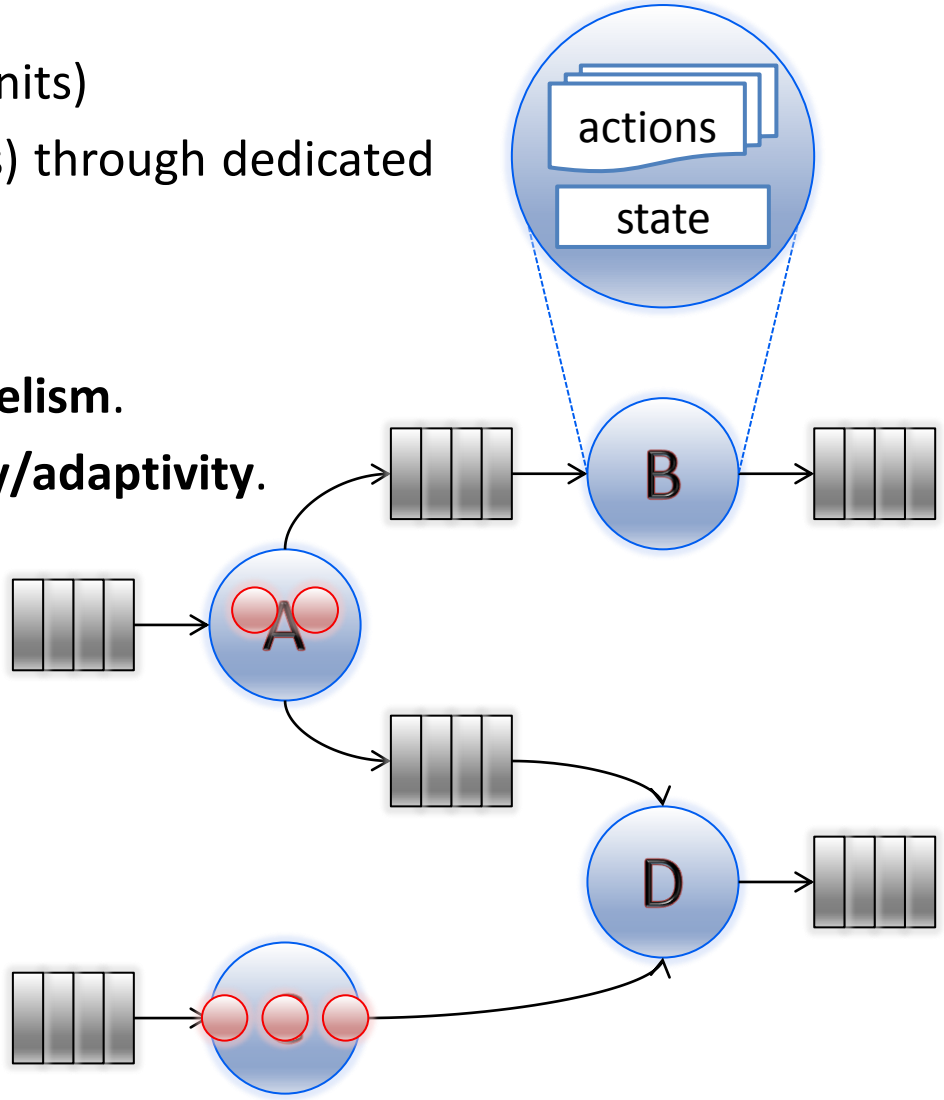
- Directed graph of **actors** (functional units)
- Actors exchange **tokens** (data packets) through dedicated channels

PECULIARITIES

- Explicit the intrinsic application **parallelism**.
- **Modularity** favours model **re-usability/adaptivity**.

EXTERNAL INTERFACE

- I/O ports **number**
- I/O ports **depth**
- I/O ports **burst** of tokens



BACKGROUND: THE DATAFLOW MOC

DATAFLOW PROGRAM

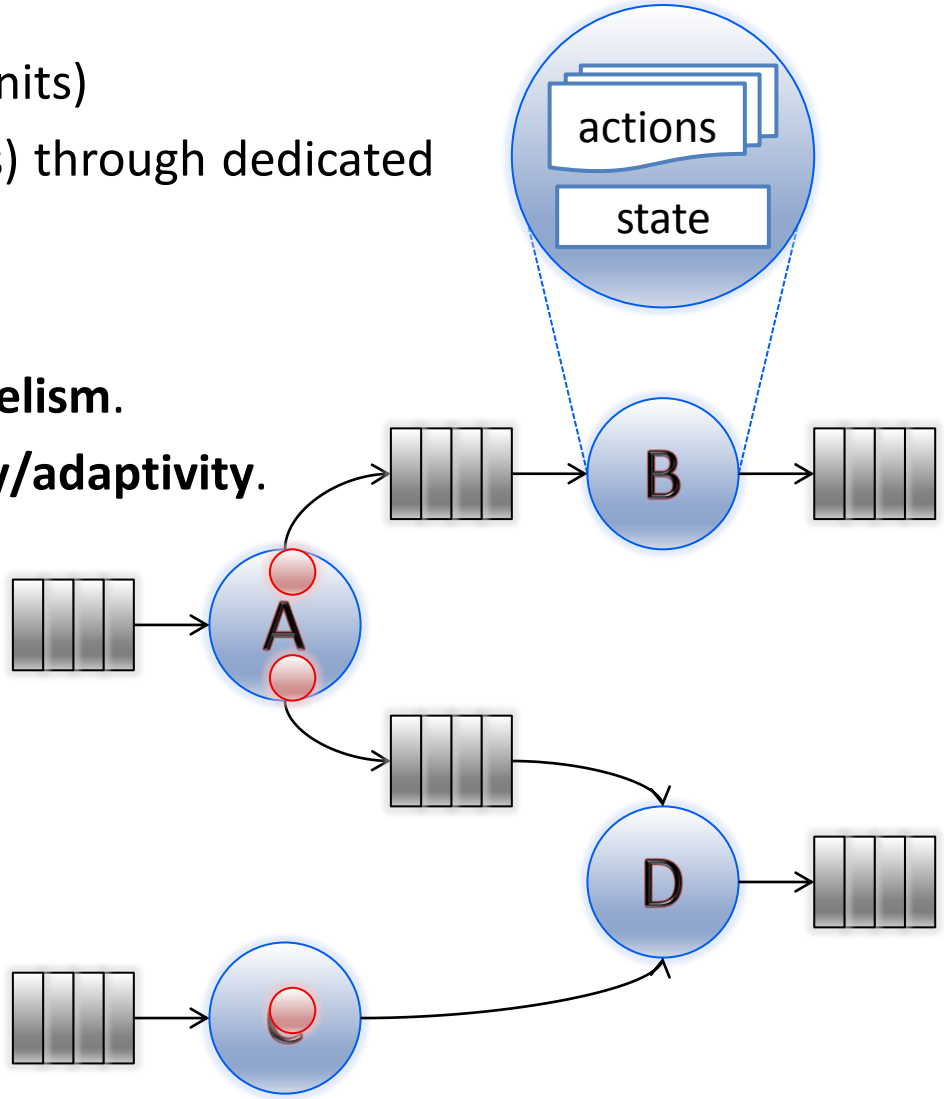
- Directed graph of **actors** (functional units)
- Actors exchange **tokens** (data packets) through dedicated channels

PECULIARITIES

- Explicit the intrinsic application **parallelism**.
- **Modularity** favours model **re-usability/adaptivity**.

EXTERNAL INTERFACE

- I/O ports **number**
- I/O ports **depth**
- I/O ports **burst** of tokens



BACKGROUND: THE DATAFLOW MOC

DATAFLOW PROGRAM

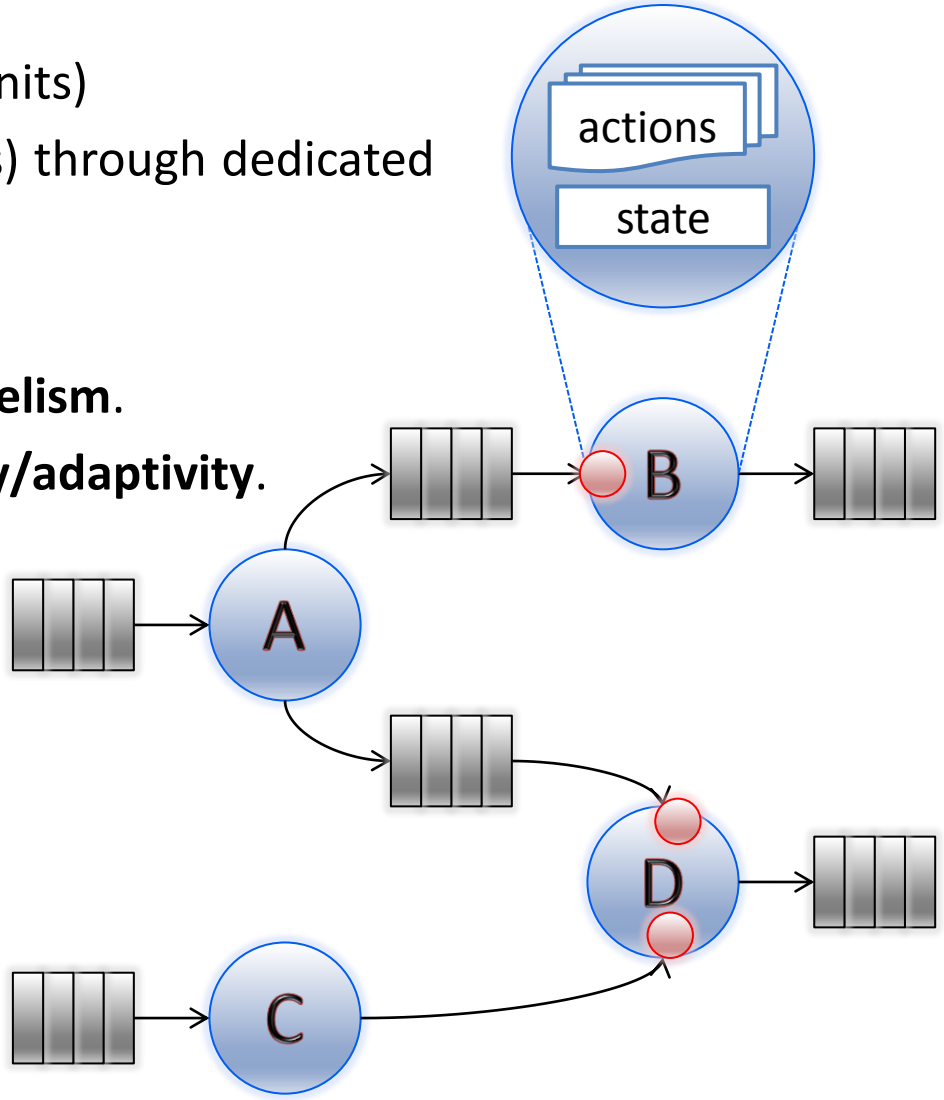
- Directed graph of **actors** (functional units)
- Actors exchange **tokens** (data packets) through dedicated channels

PECULIARITIES

- Explicit the intrinsic application **parallelism**.
- **Modularity** favours model **re-usability/adaptivity**.

EXTERNAL INTERFACE

- I/O ports **number**
- I/O ports **depth**
- I/O ports **burst** of tokens



BACKGROUND: THE DATAFLOW MOC

DATAFLOW PROGRAM

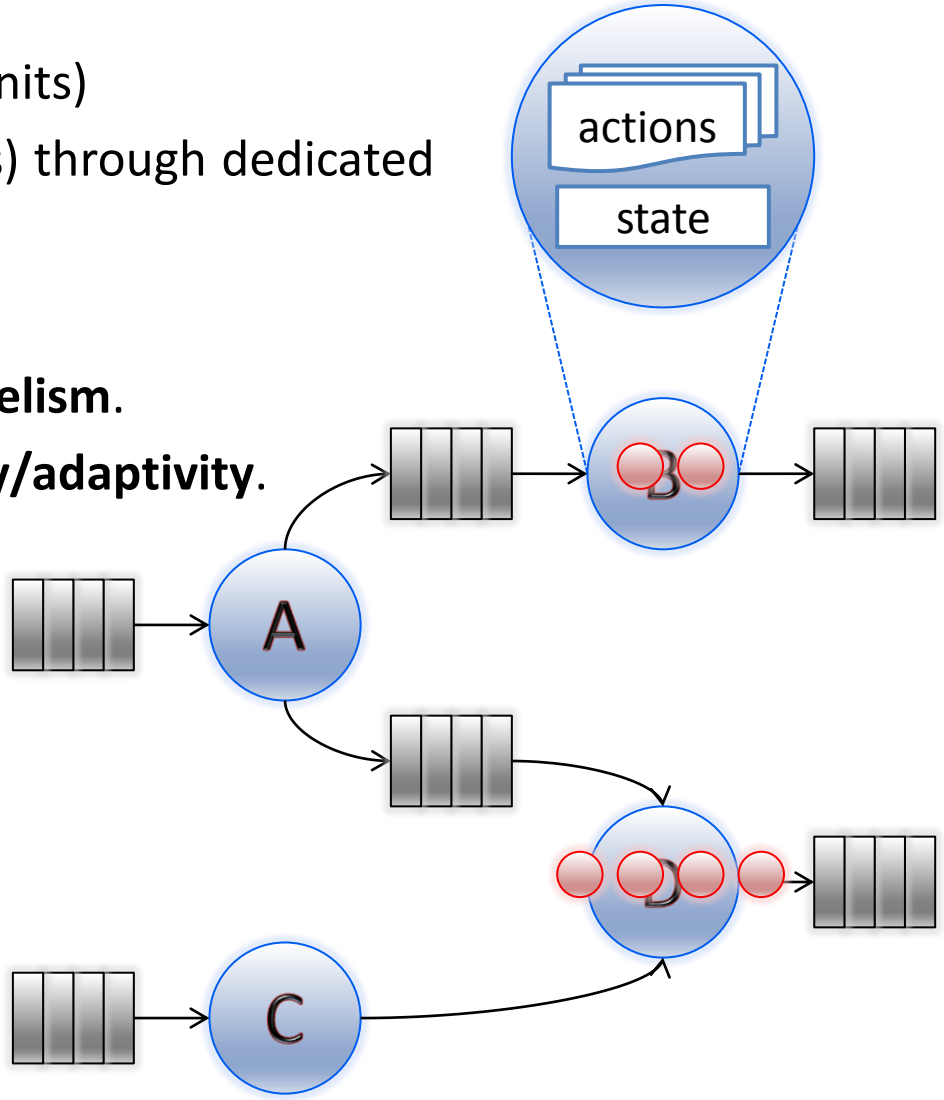
- Directed graph of **actors** (functional units)
- Actors exchange **tokens** (data packets) through dedicated channels

PECULIARITIES

- Explicit the intrinsic application **parallelism**.
- **Modularity** favours model **re-usability/adaptivity**.

EXTERNAL INTERFACE

- I/O ports **number**
- I/O ports **depth**
- I/O ports **burst** of tokens



BACKGROUND: THE DATAFLOW MOC

DATAFLOW PROGRAM

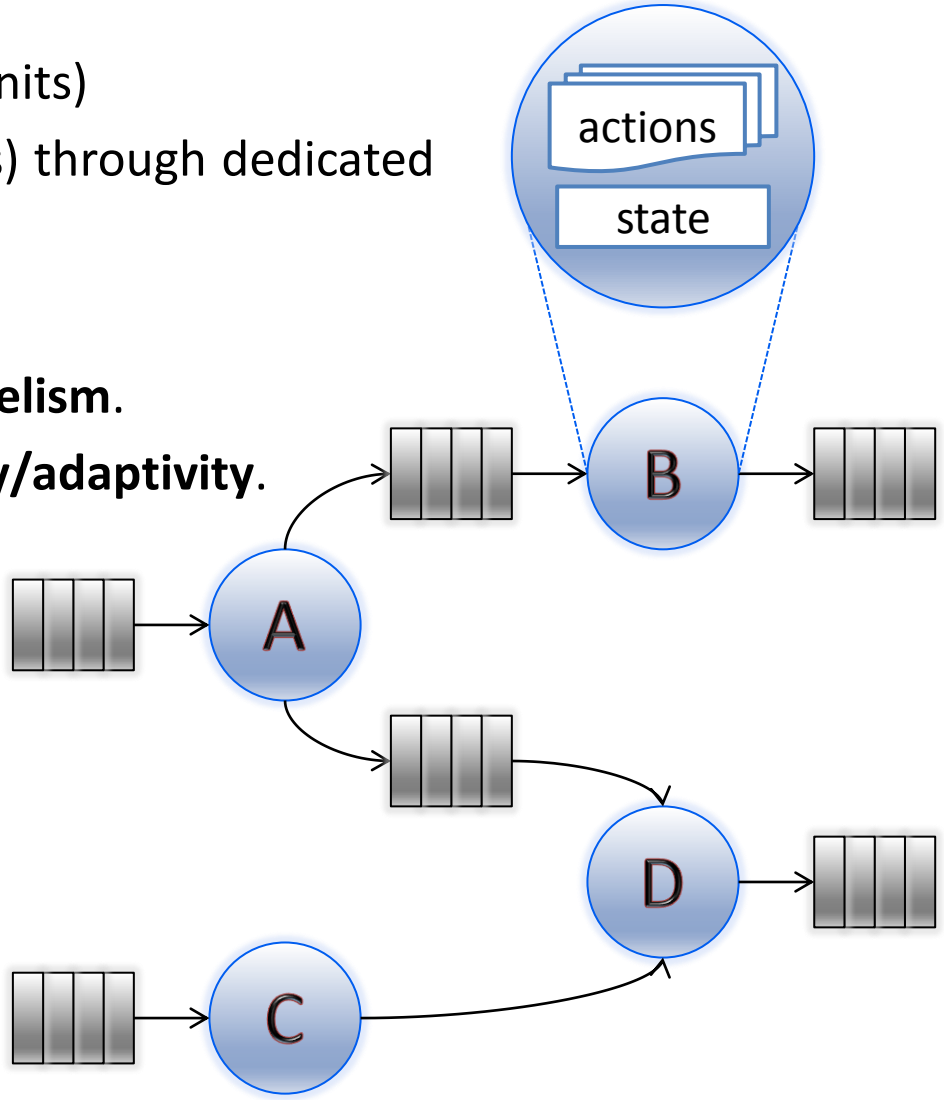
- Directed graph of **actors** (functional units)
- Actors exchange **tokens** (data packets) through dedicated channels

PECULIARITIES

- Explicit the intrinsic application **parallelism**.
- **Modularity** favours model **re-usability/adaptivity**.

EXTERNAL INTERFACE

- I/O ports **number**
- I/O ports **depth**
- I/O ports **burst** of tokens



GOALS AND WORK EVOLUTION

DASIP 2010:

- High-level dataflow combination tool, front-end of the Multi-Dataflow Composer tool.

DASIP 2011:

- Concrete definition of the hardware template and of the dataflow-based mapping strategy.

ISCAS 2012:

- Integration of the complete synthesis flow.

SAMOS 2014:

- Implementation of a coarse-grained multi-standard decoder.

GOALS AND WORK EVOLUTION

GOAL: AUTOMATIC deployment of EFFICIENT HARDWARE-ACCELERATORS, contemporarily tackling HIGH-PERFORMANCE and LONG-TERM ADAPTIVITY.

DASIP 2010:

- High-level dataflow combination tool, front-end of the Multi-Dataflow Composer tool.

DASIP 2011:

- Concrete definition of the hardware template and of the dataflow-based mapping strategy.

ISCAS 2012:

- Integration of the complete synthesis flow.

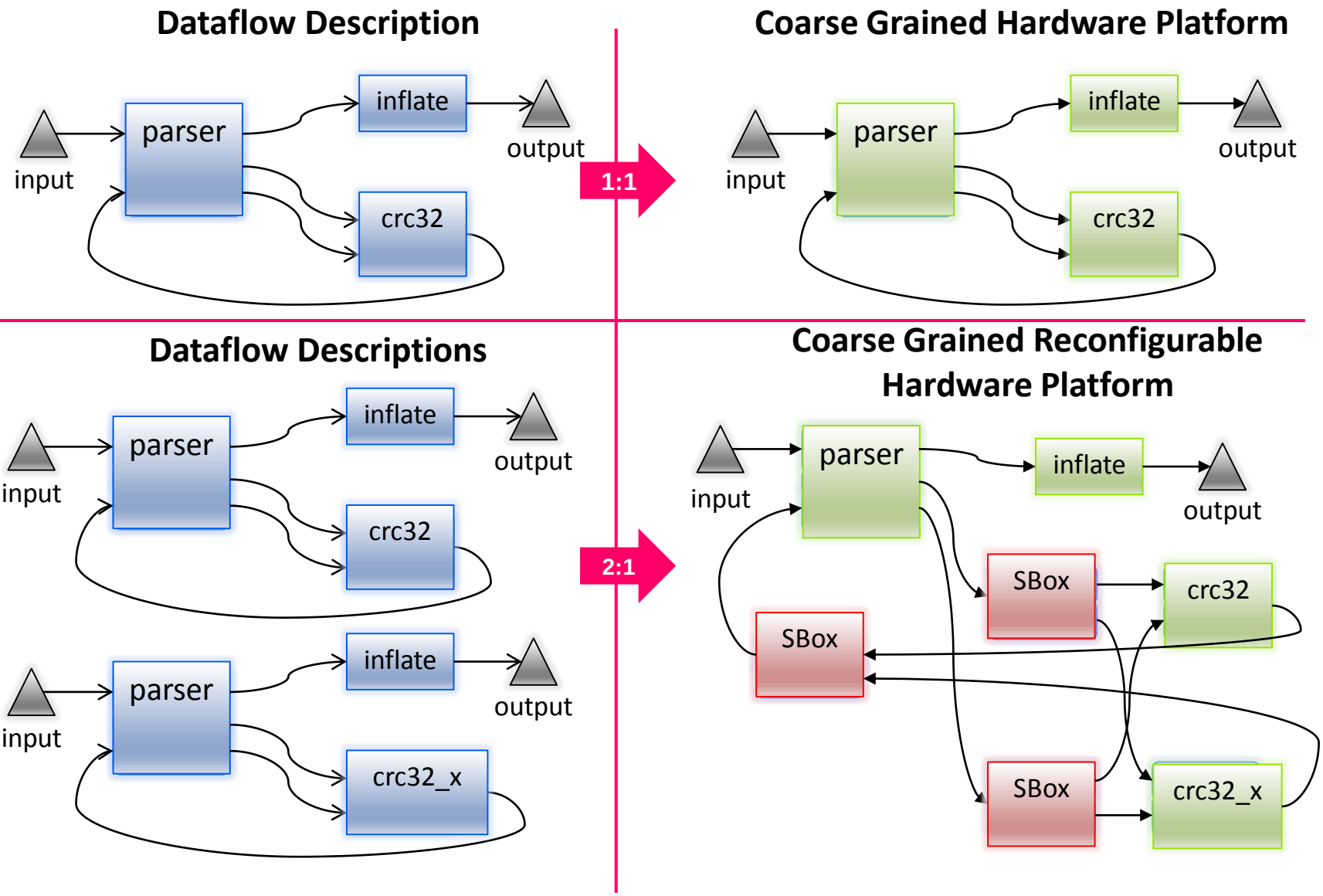
SAMOS 2014:

- Implementation of a coarse-grained multi-standard decoder.

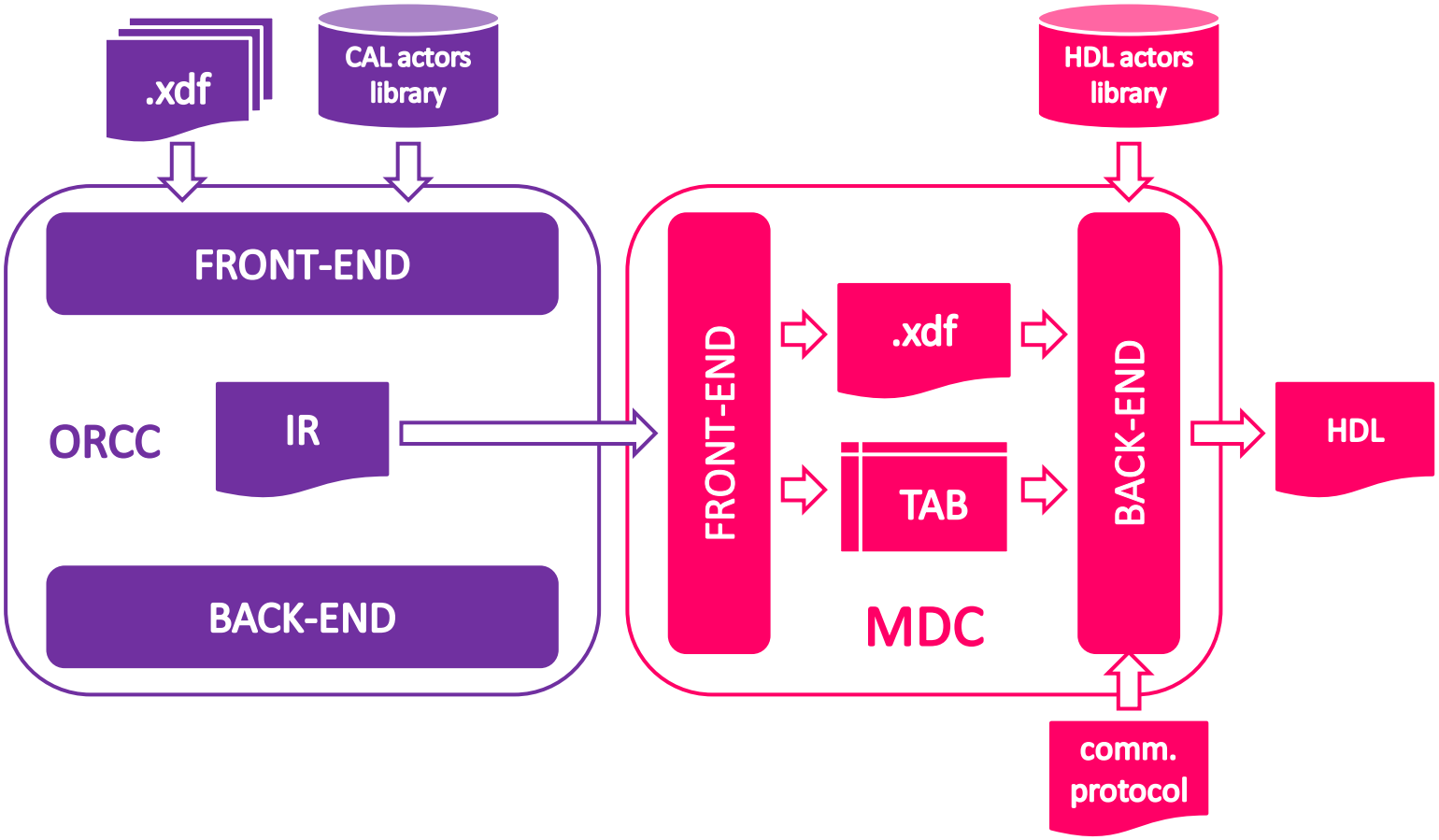
OUTLINE: CO-PROCESSOR GENERATION

- Introduction:
  - Problem statement
  - Background
  - Goals
- Co-processing units generation:
  - Approach and baseline Multi-Dataflow Composer
  - Template Interface Layer: hardware and automatic composition
- Performance assessment
  - Use-case scenario
  - Results
- Final remarks and future directions

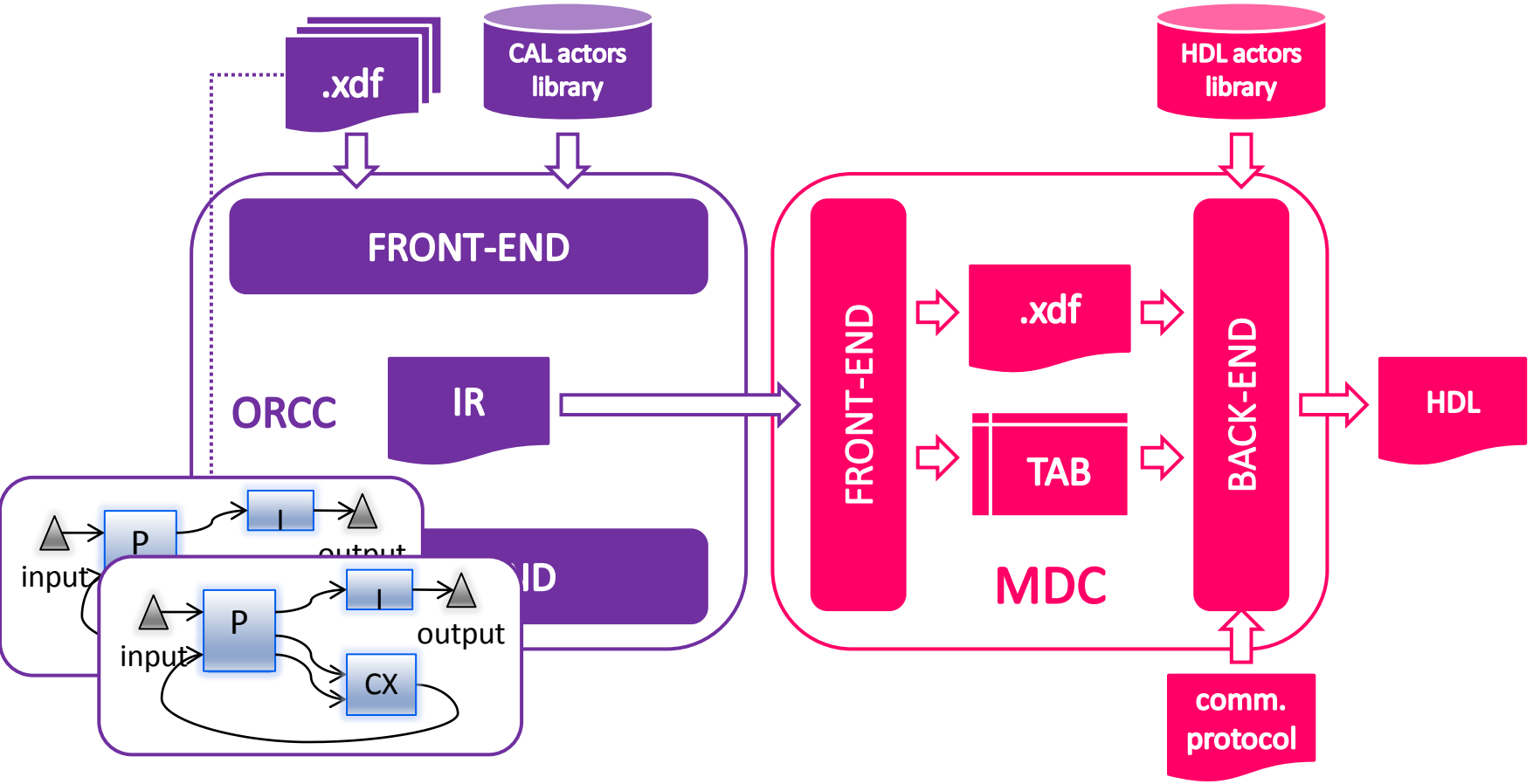
MDC: BASIC APPROACH



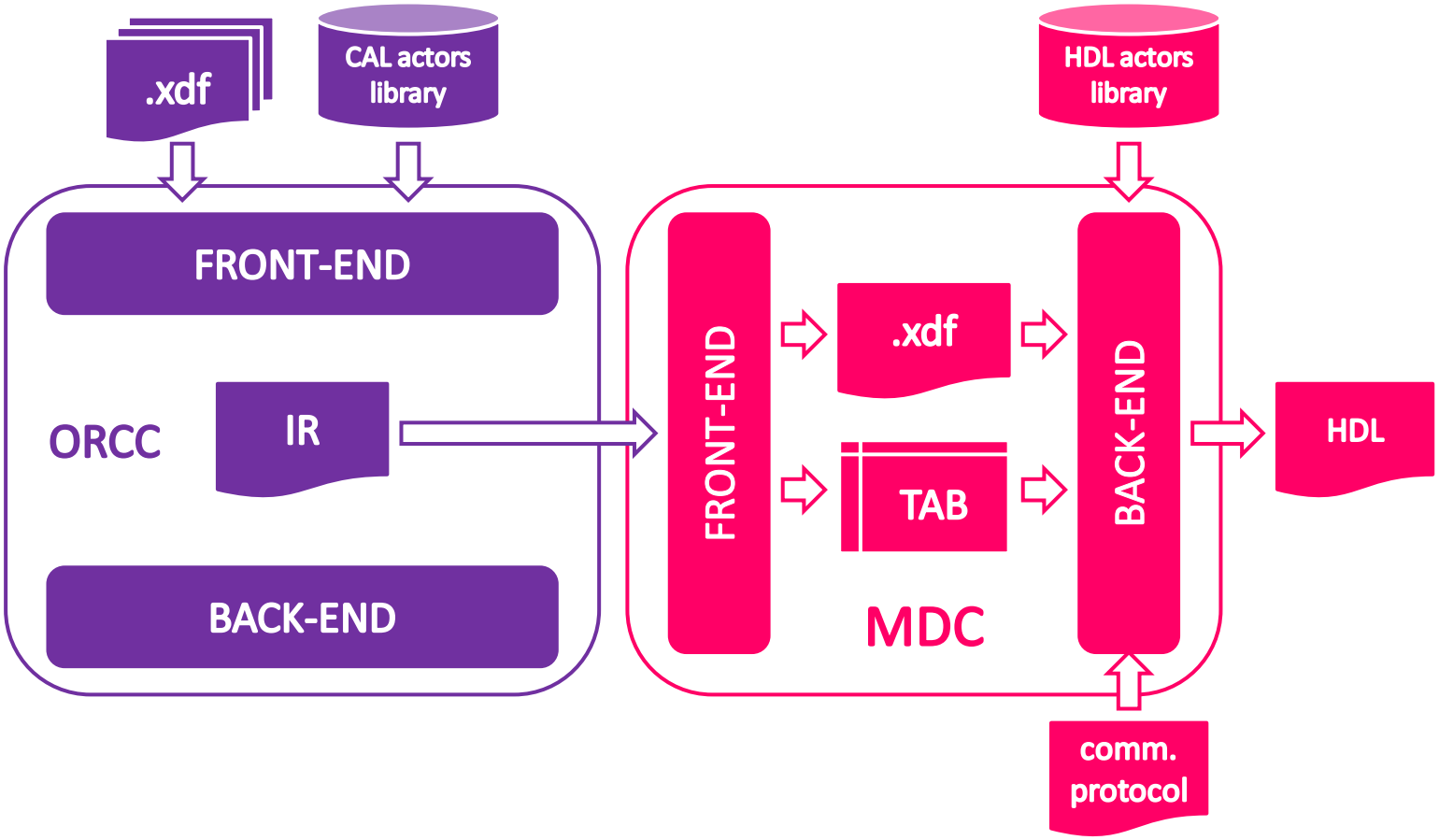
MDC: THE TOOL


.xdf
CAL actors library
HDL actors library
FRONT-END
ORCC
IR
BACK-END
FRONT-END
.xdf
TAB
BACK-END
HDL
MDC
comm. protocol

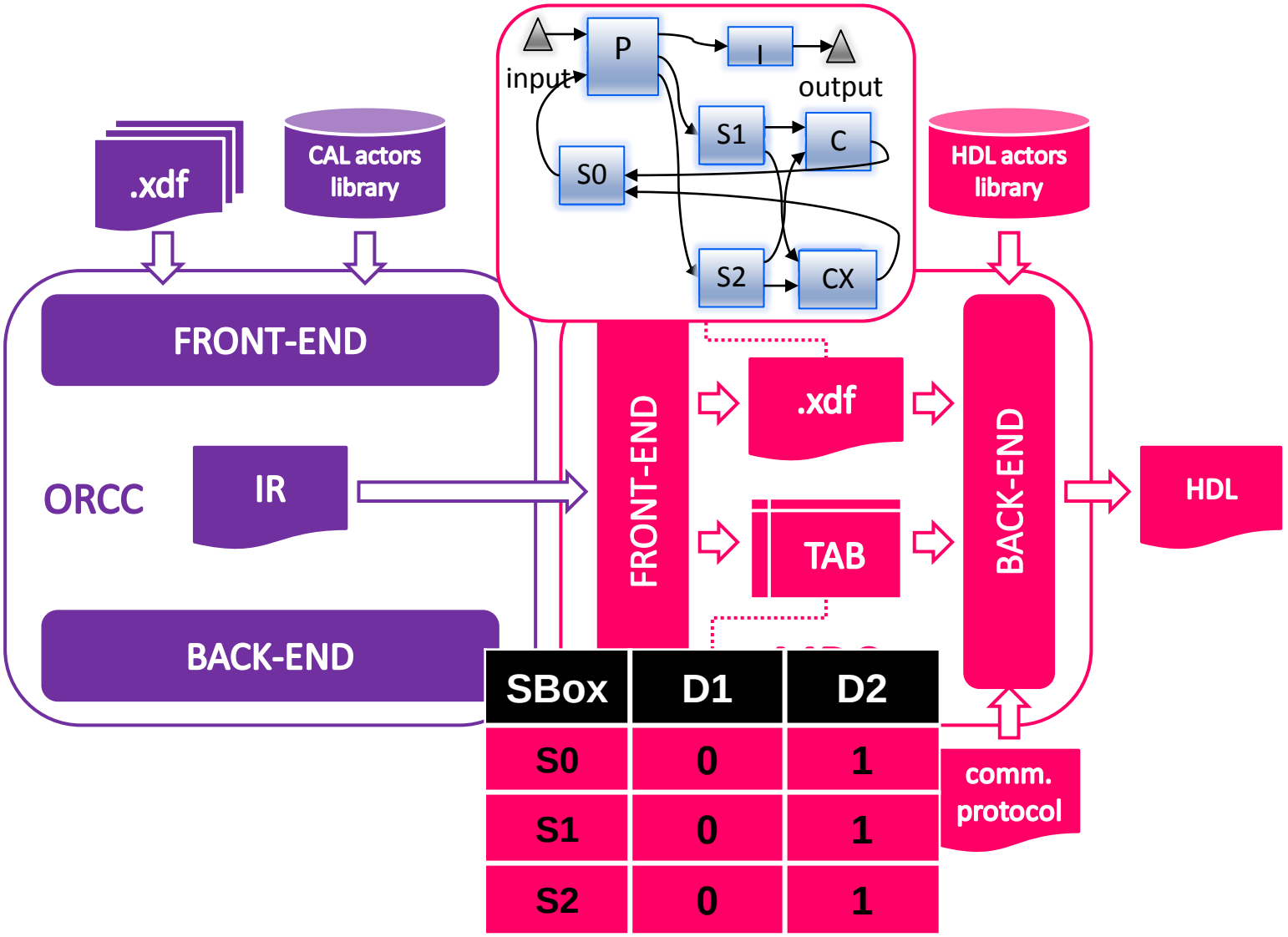
MDC: THE TOOL


.xdf
CAL actors library
HDL actors library
FRONT-END
ORCC
IR
ID
input
P
I
output
CX
FRONT-END
.xdf
TAB
BACK-END
HDL
MDC
comm. protocol

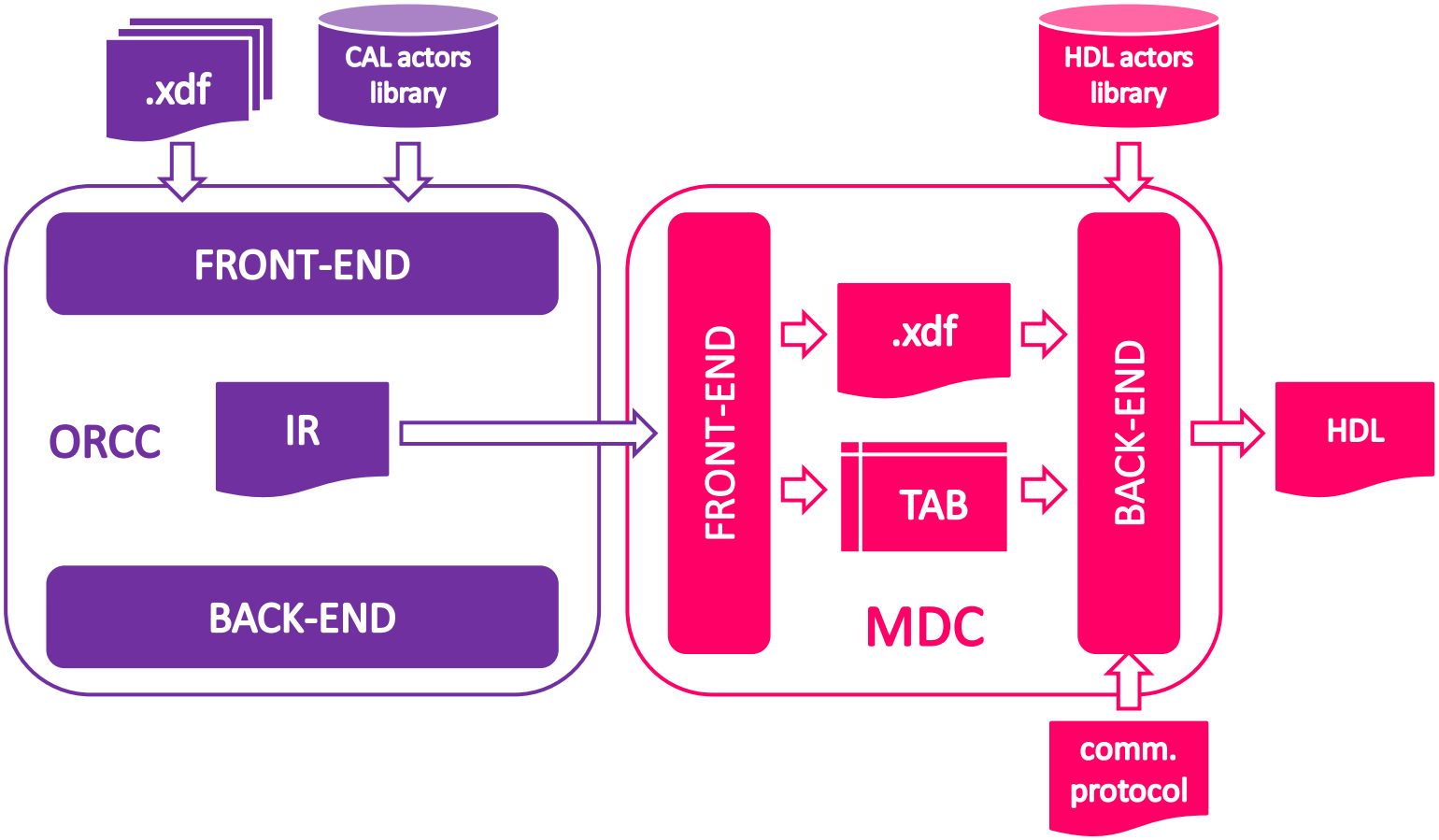
MDC: THE TOOL


.xdf
CAL actors library
HDL actors library
FRONT-END
ORCC
IR
BACK-END
FRONT-END
.xdf
TAB
BACK-END
HDL
MDC
comm. protocol

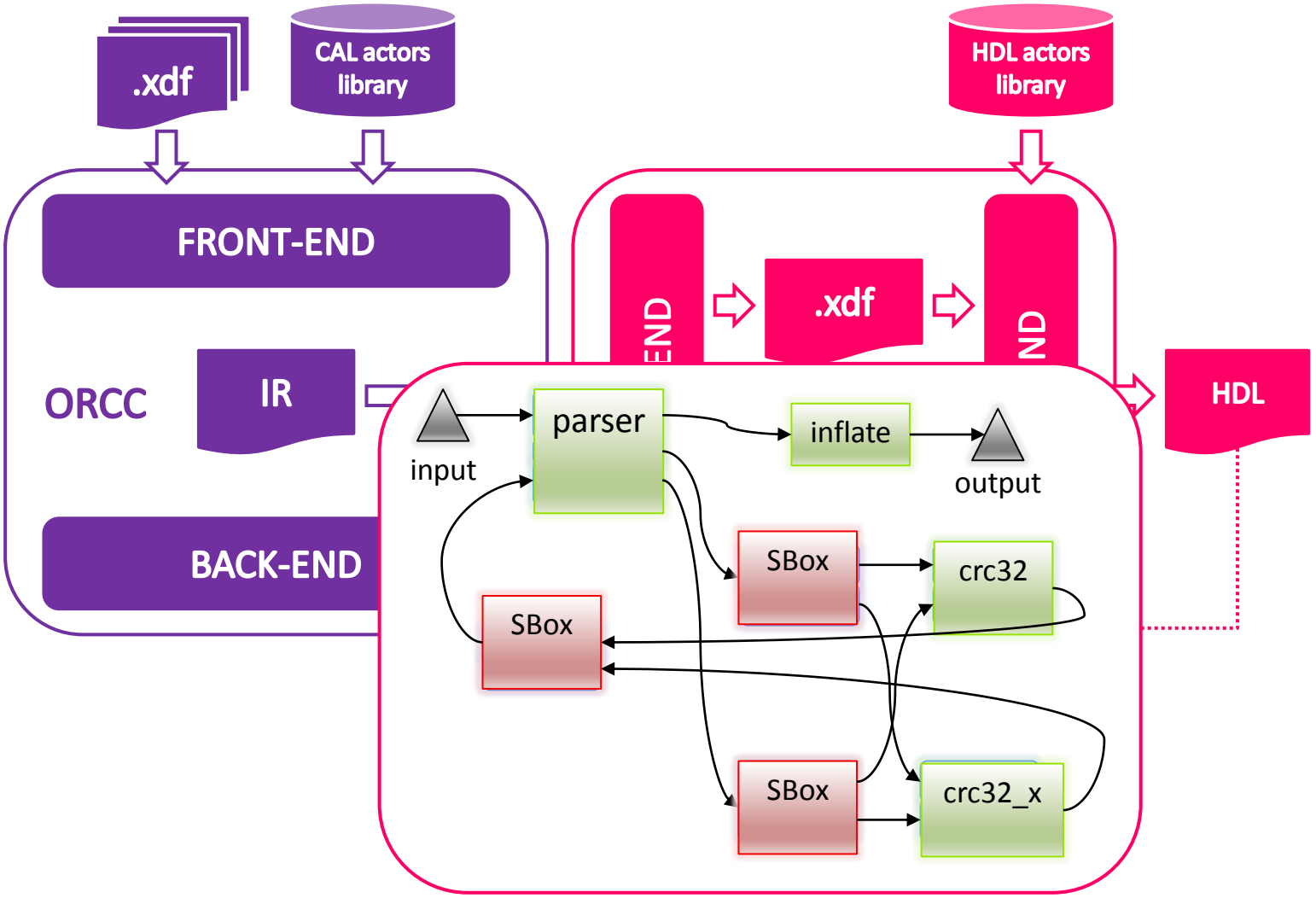
MDC: THE TOOL


P
I
input
output
S1
C
S0
S2
CX
.xdf
CAL actors library
HDL actors library
FRONT-END
ORCC
IR
BACK-END
FRONT-END
.xdf
TAB
BACK-END
HDL
comm. protocol
SBox
D1
D2
S0
0
1
S1
0
1
S2
0
1

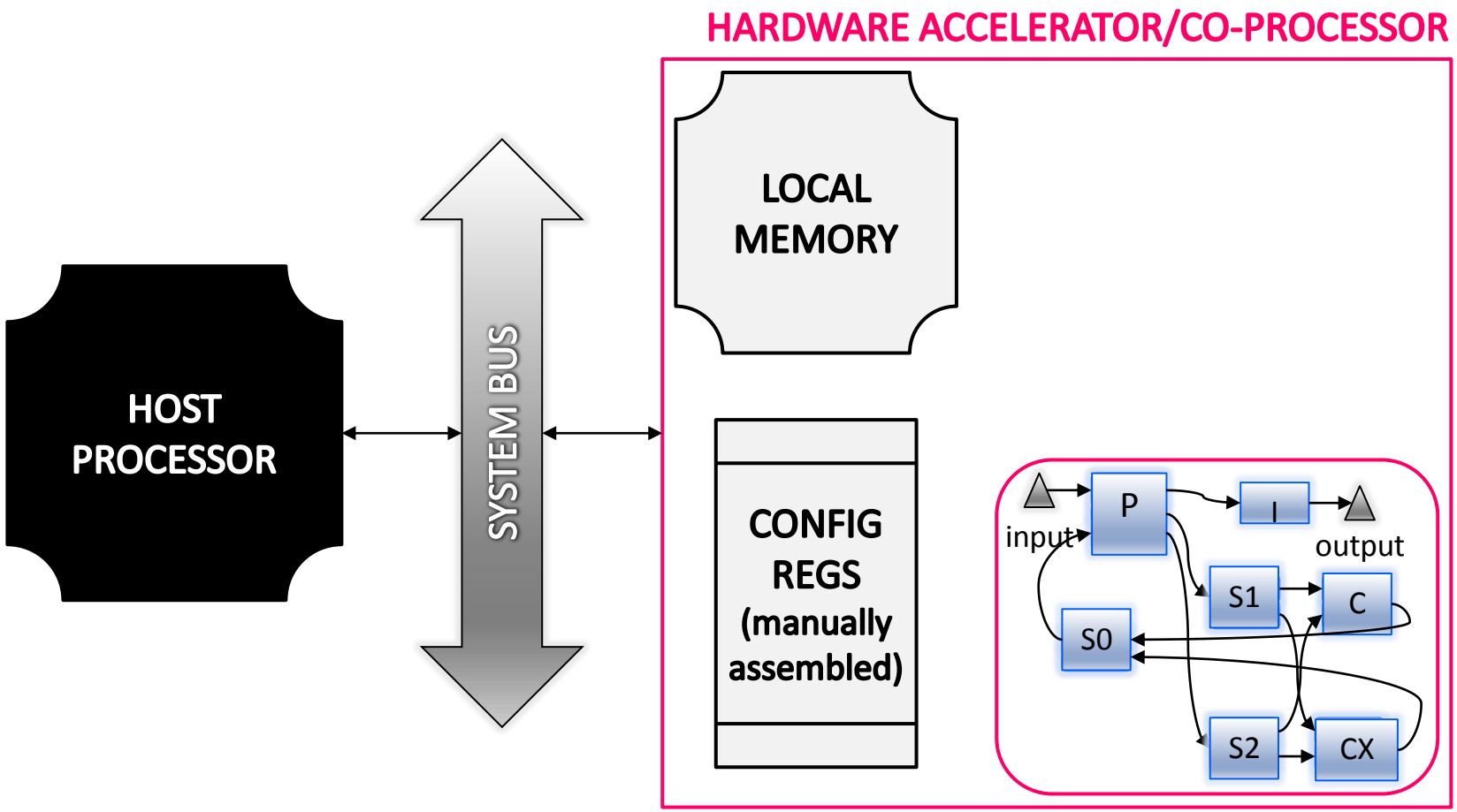
MDC: THE TOOL


.xdf
CAL actors library
HDL actors library
FRONT-END
ORCC
IR
BACK-END
FRONT-END
.xdf
TAB
BACK-END
HDL
MDC
comm. protocol

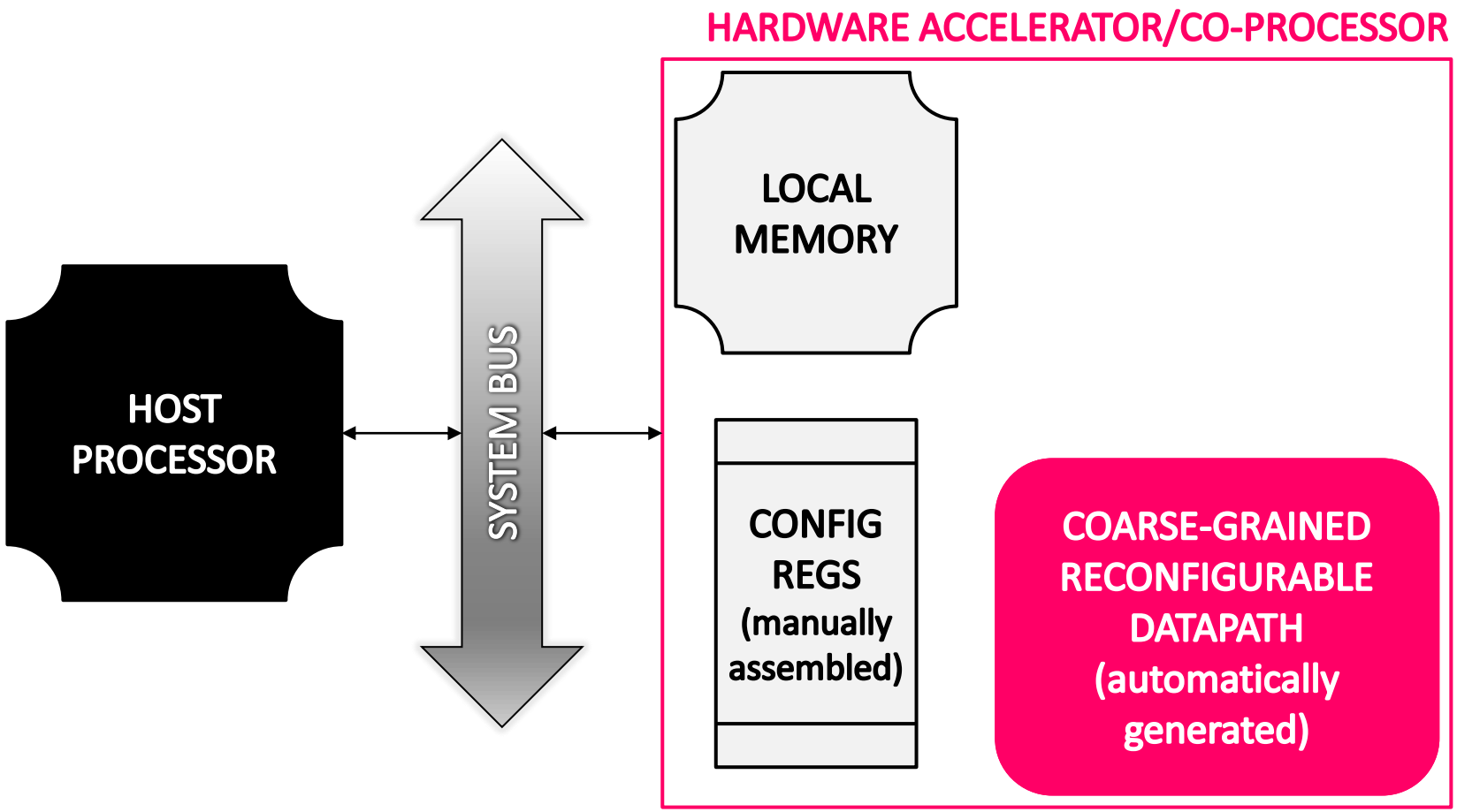
MDC: THE TOOL


.xdf
CAL actors library
HDL actors library
FRONT-END
ORCC
IR
BACK-END
.xdf
HDL
input
parser
inflate
output
SBox
crc32
SBox
SBox
crc32_x

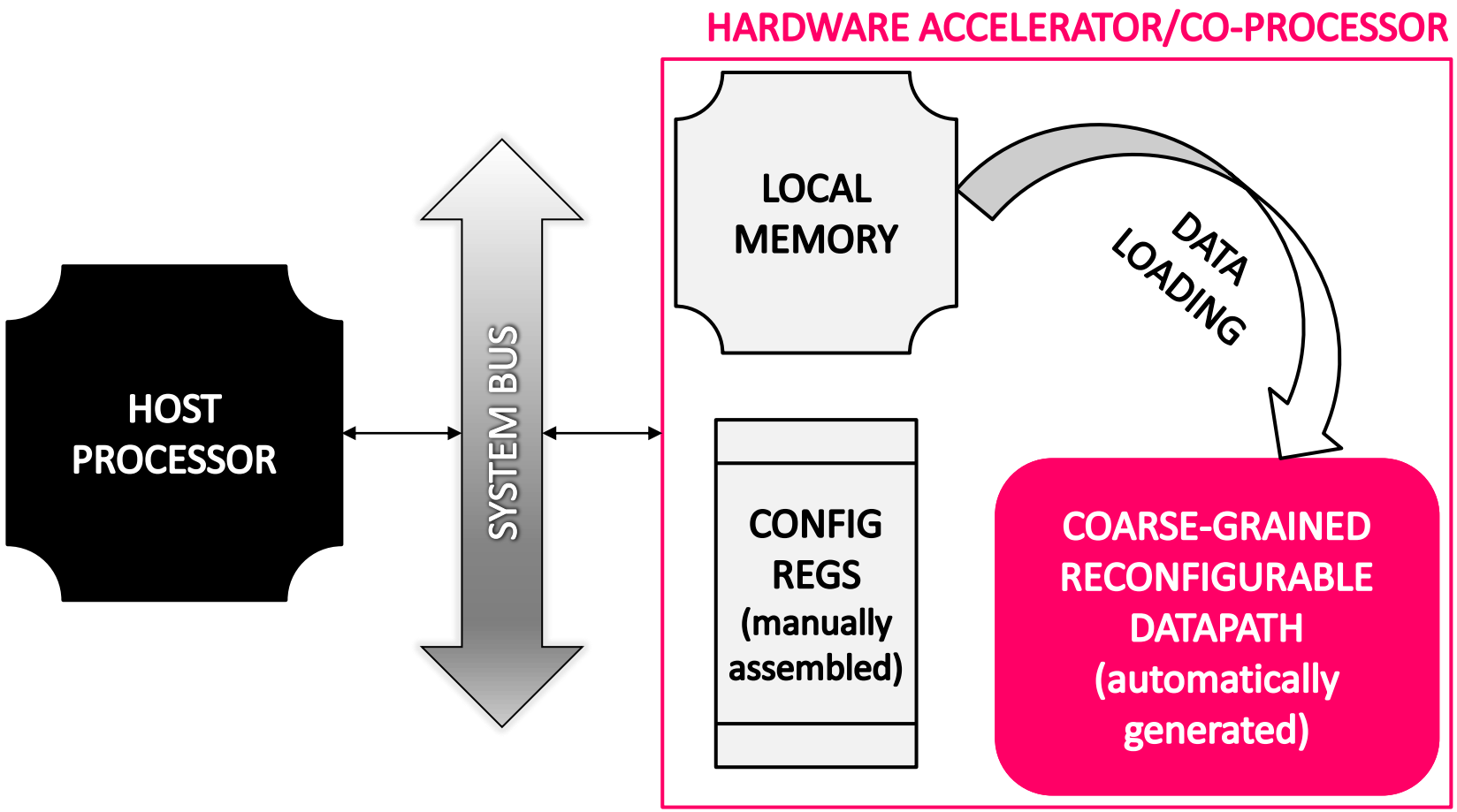
MDC: AUTOMATIC GENERATION



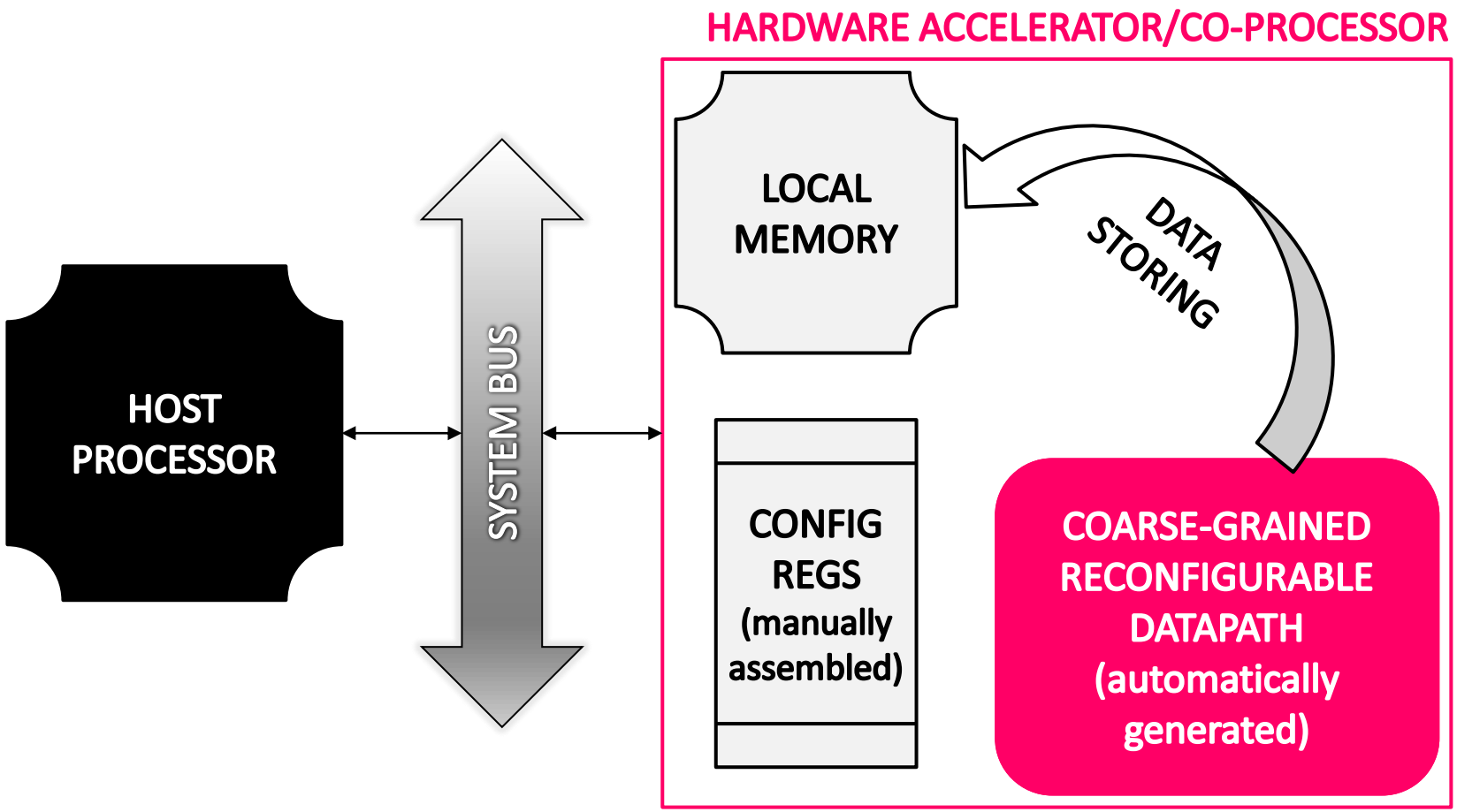
MDC: AUTOMATIC GENERATION



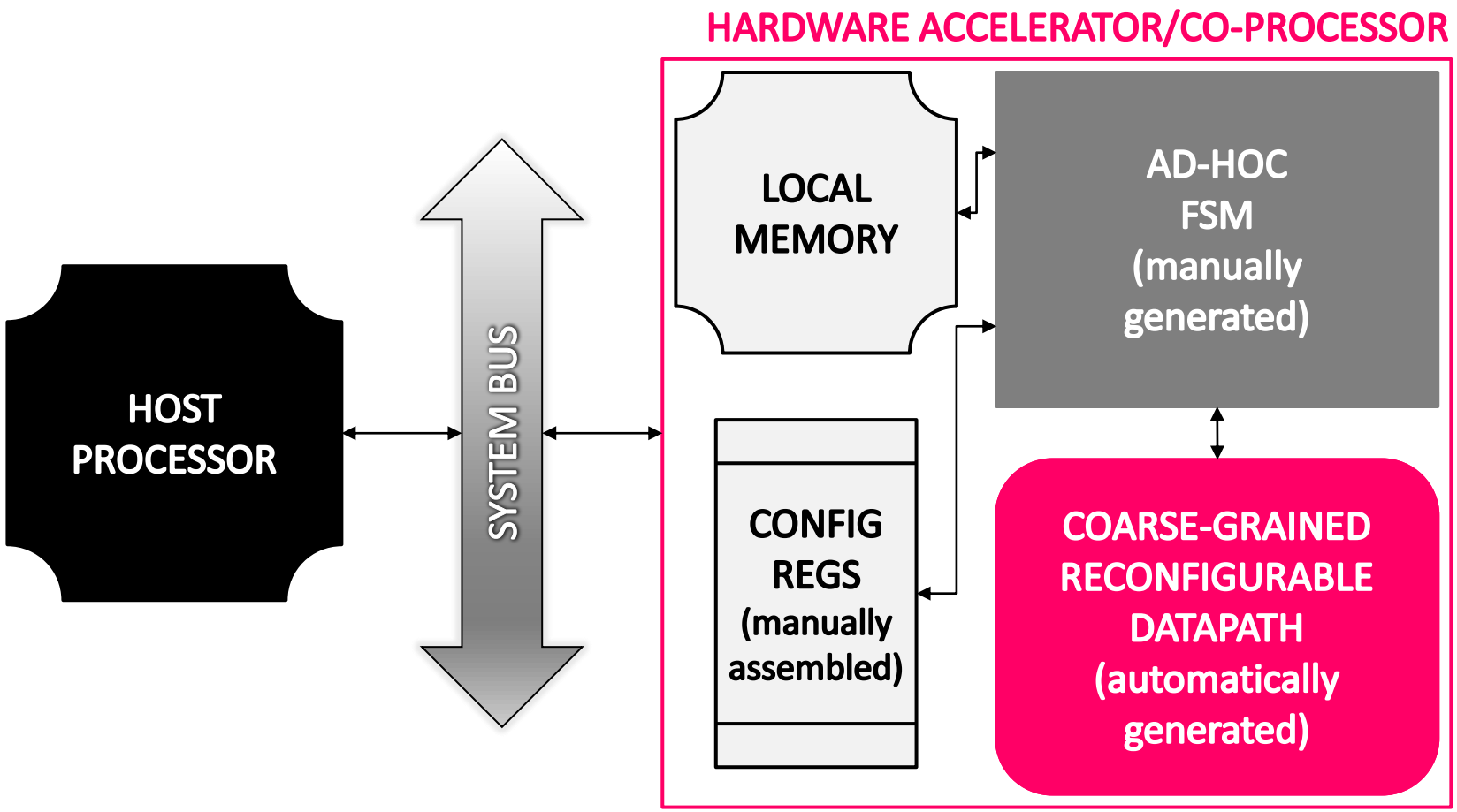
MDC: AUTOMATIC GENERATION



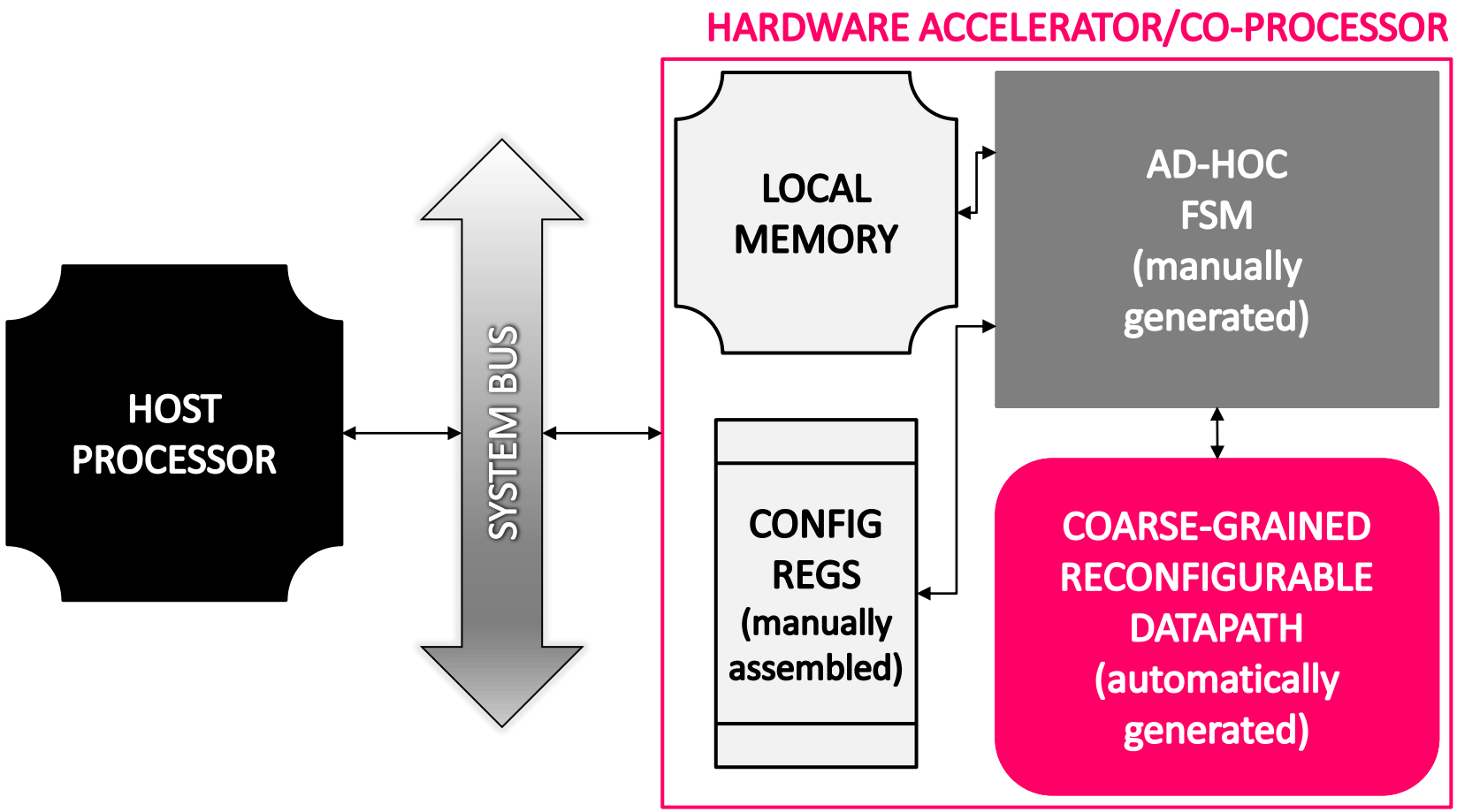
MDC: AUTOMATIC GENERATION



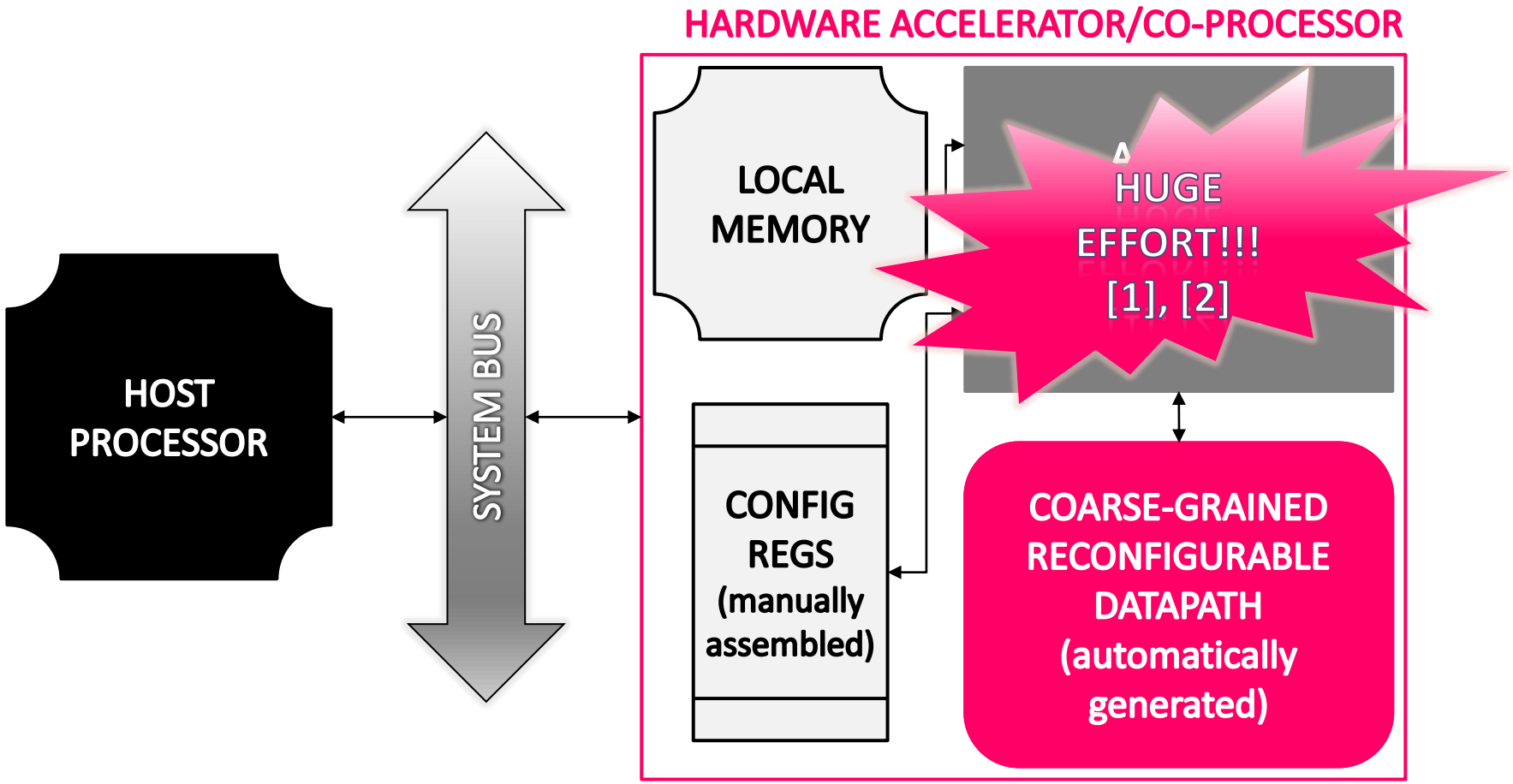
MDC: AUTOMATIC GENERATION



MDC: AUTOMATIC GENERATION



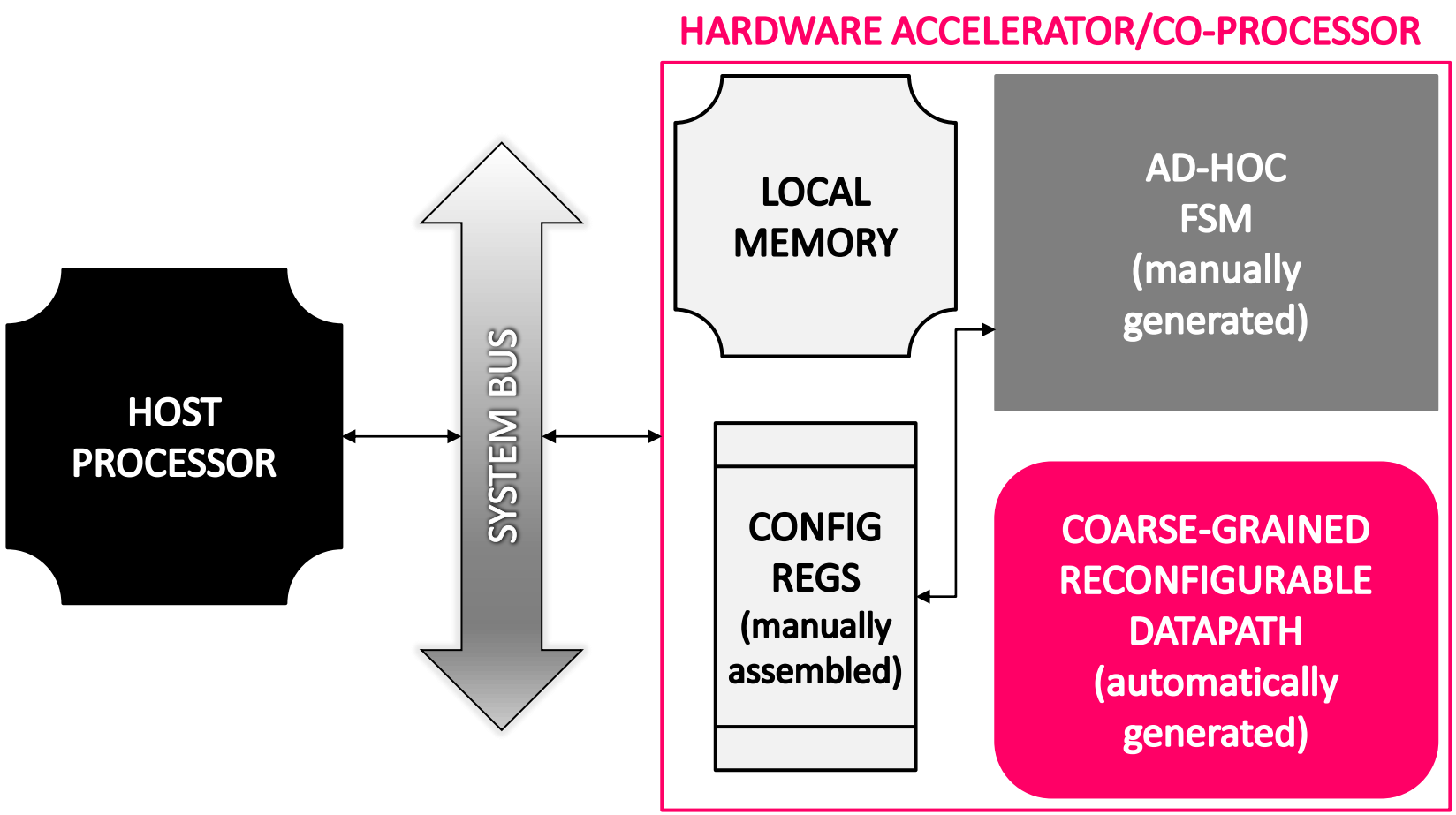
MDC: AUTOMATIC GENERATION



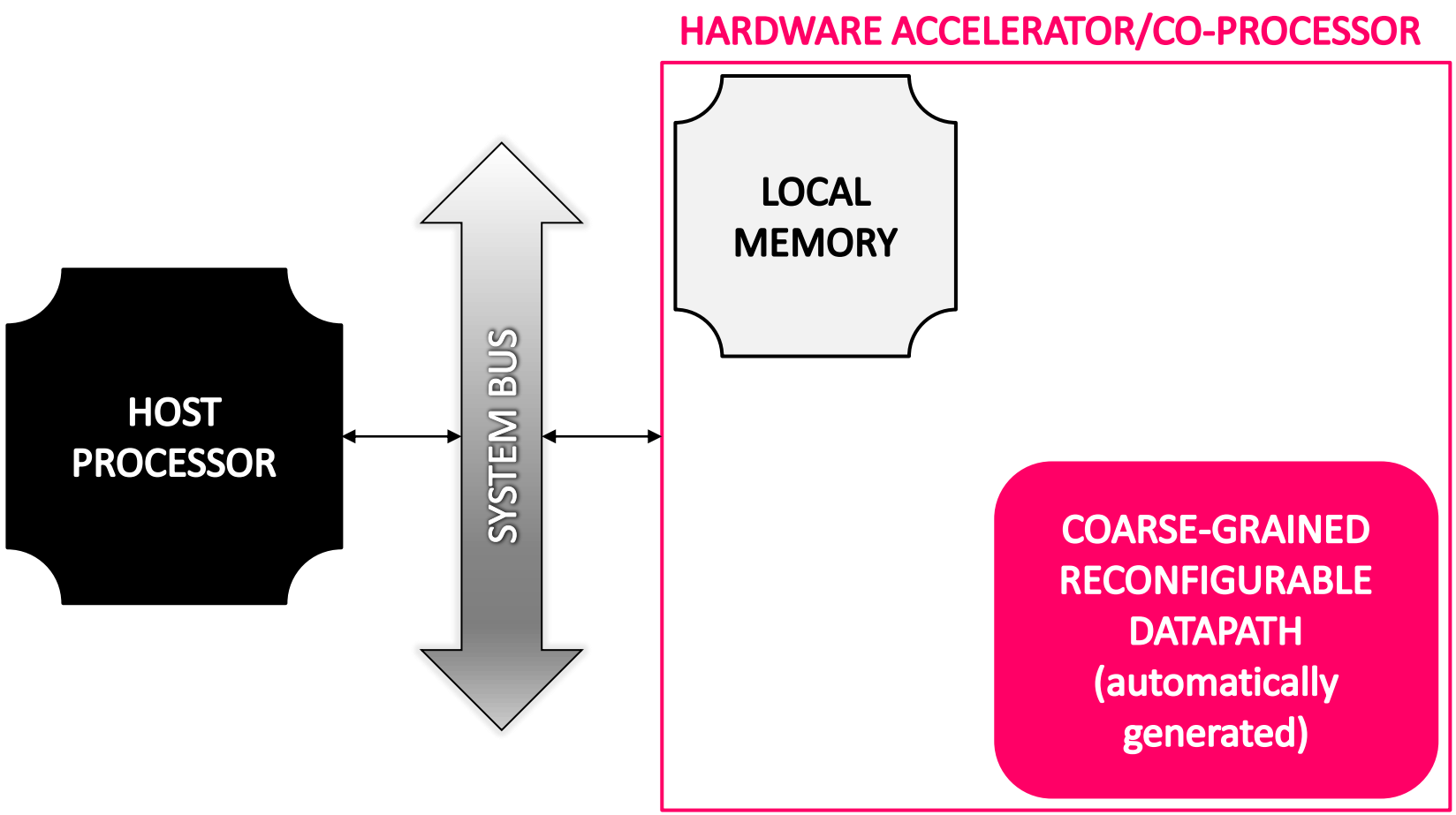
[1] F. Palumbo, N. Carta, D. Pani, P. Meloni and L. Raffo, *The multi-dataflow composer tool: generation of on-the-fly reconfigurable platforms*, Journal of Real-Time Image Processing, vol. 9, no. 1, pp 233-249, 2012.

[2] N. Carta, C. Sau, D. Pani, F. Palumbo and L. Raffo, *A Coarse-Grained Reconfigurable Approach for Low-Power Spike Sorting Architectures*, IEEE/EMBS Conference on Neural Engineering, 2013.

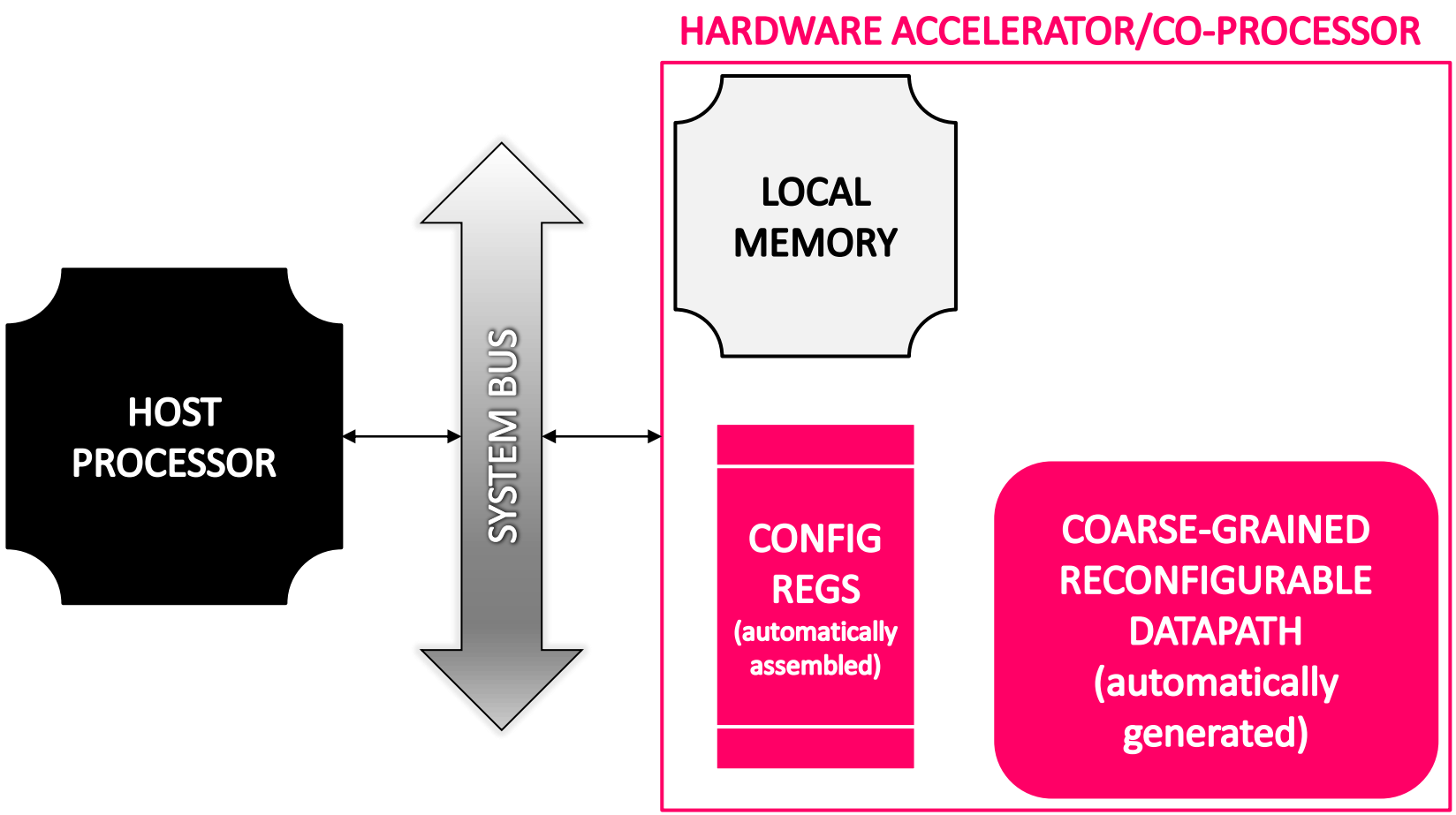
THE TEMPLATE INTERFACE LAYER (TIL)



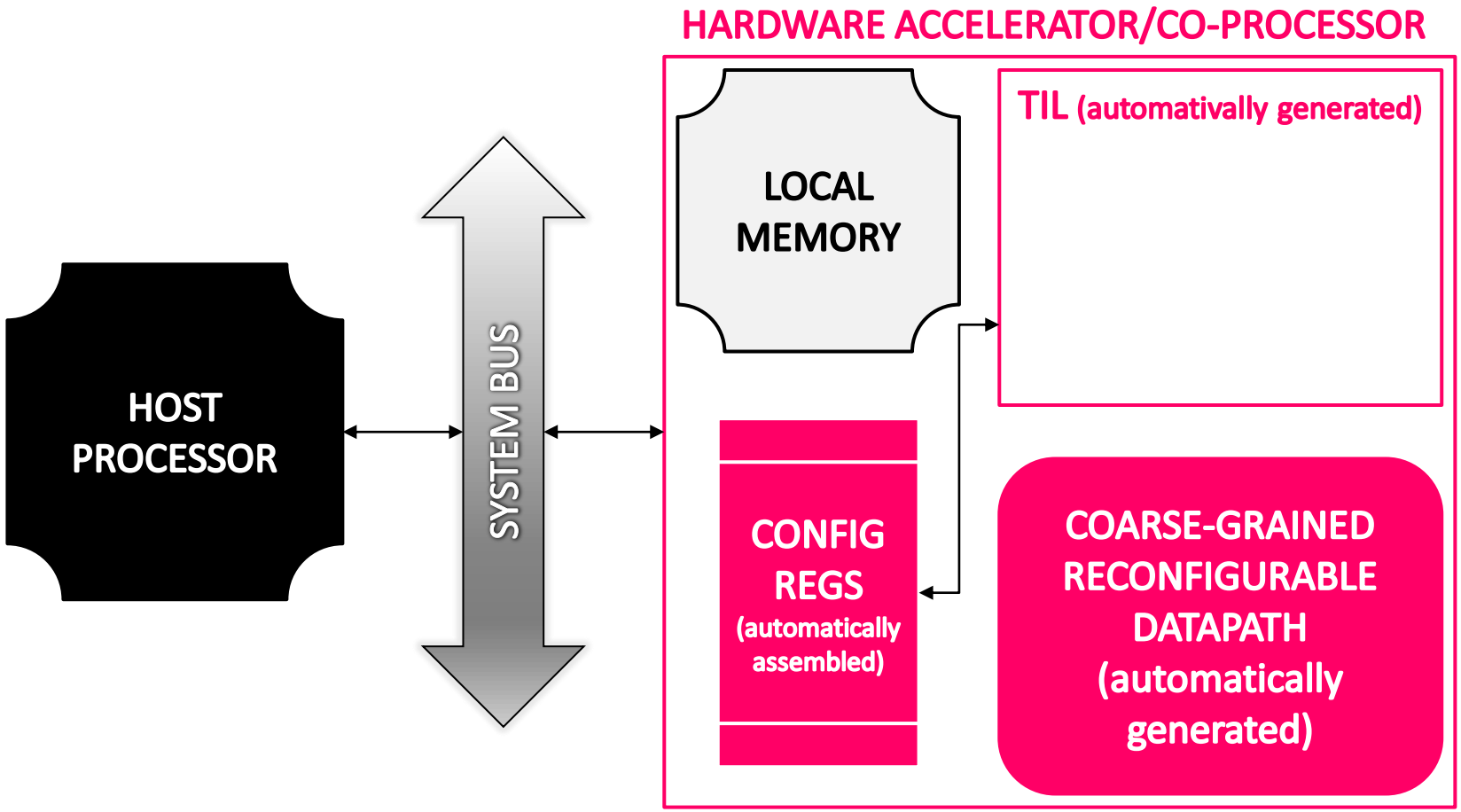
THE TEMPLATE INTERFACE LAYER (TIL)



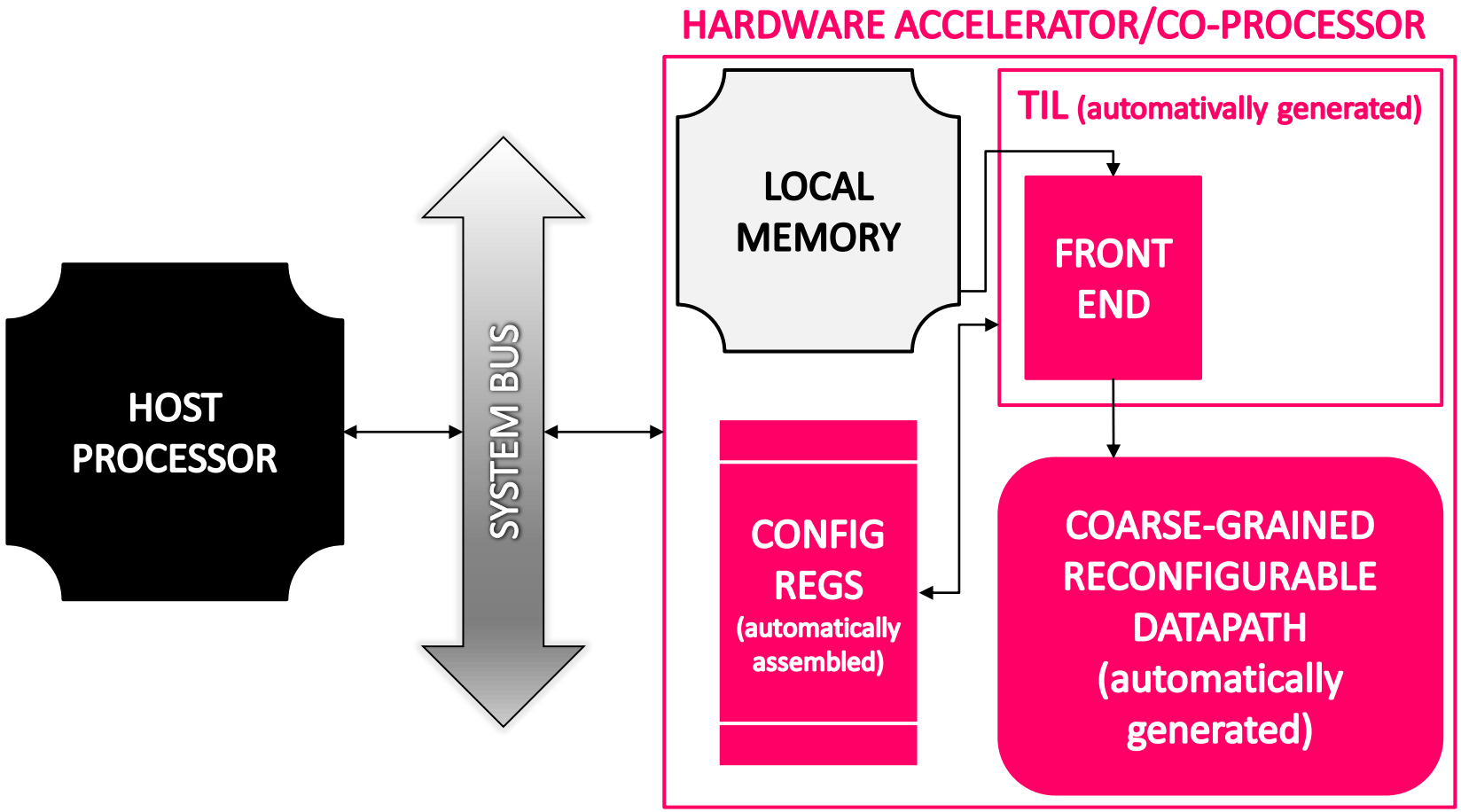
THE TEMPLATE INTERFACE LAYER (TIL)



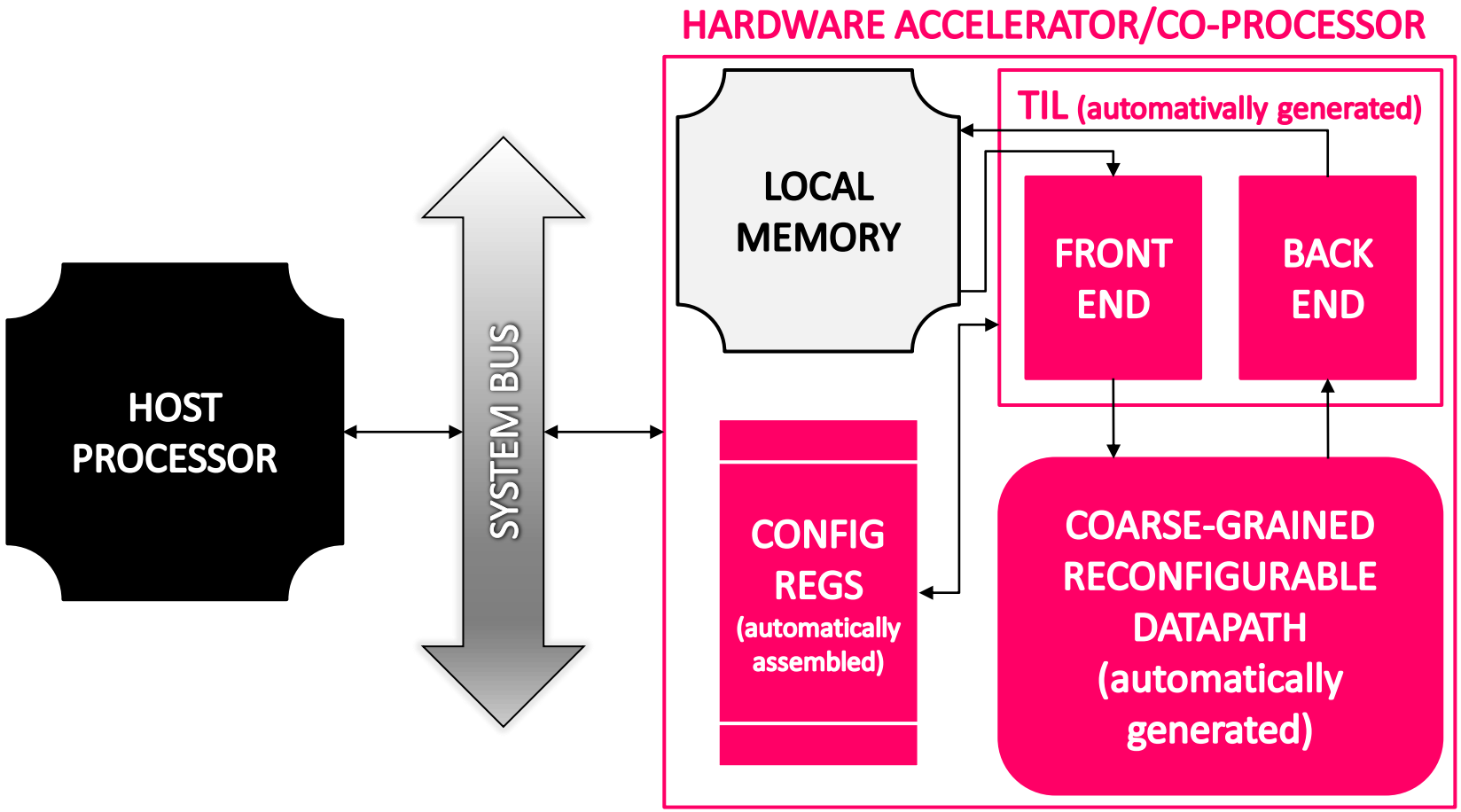
THE TEMPLATE INTERFACE LAYER (TIL)



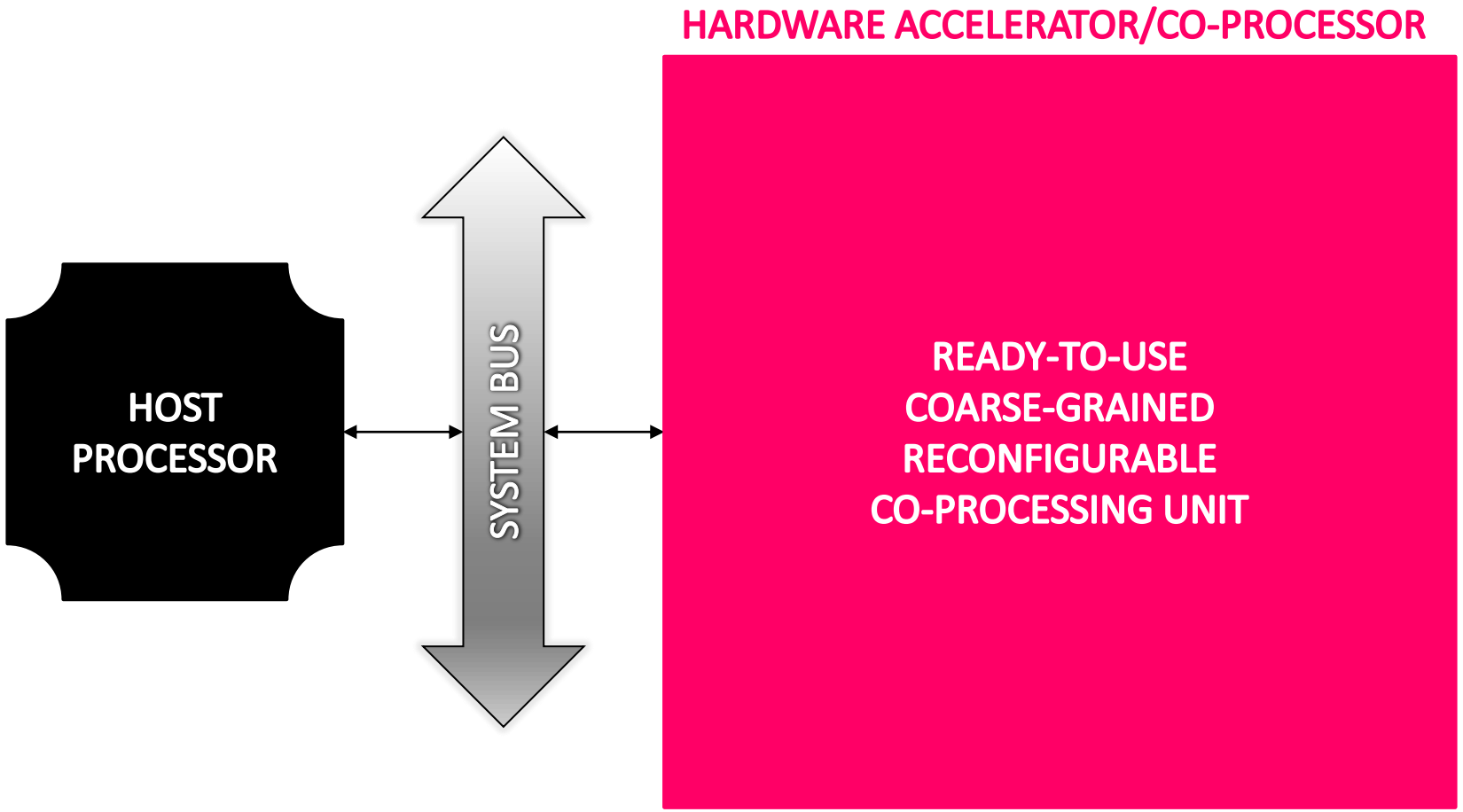
THE TEMPLATE INTERFACE LAYER (TIL)



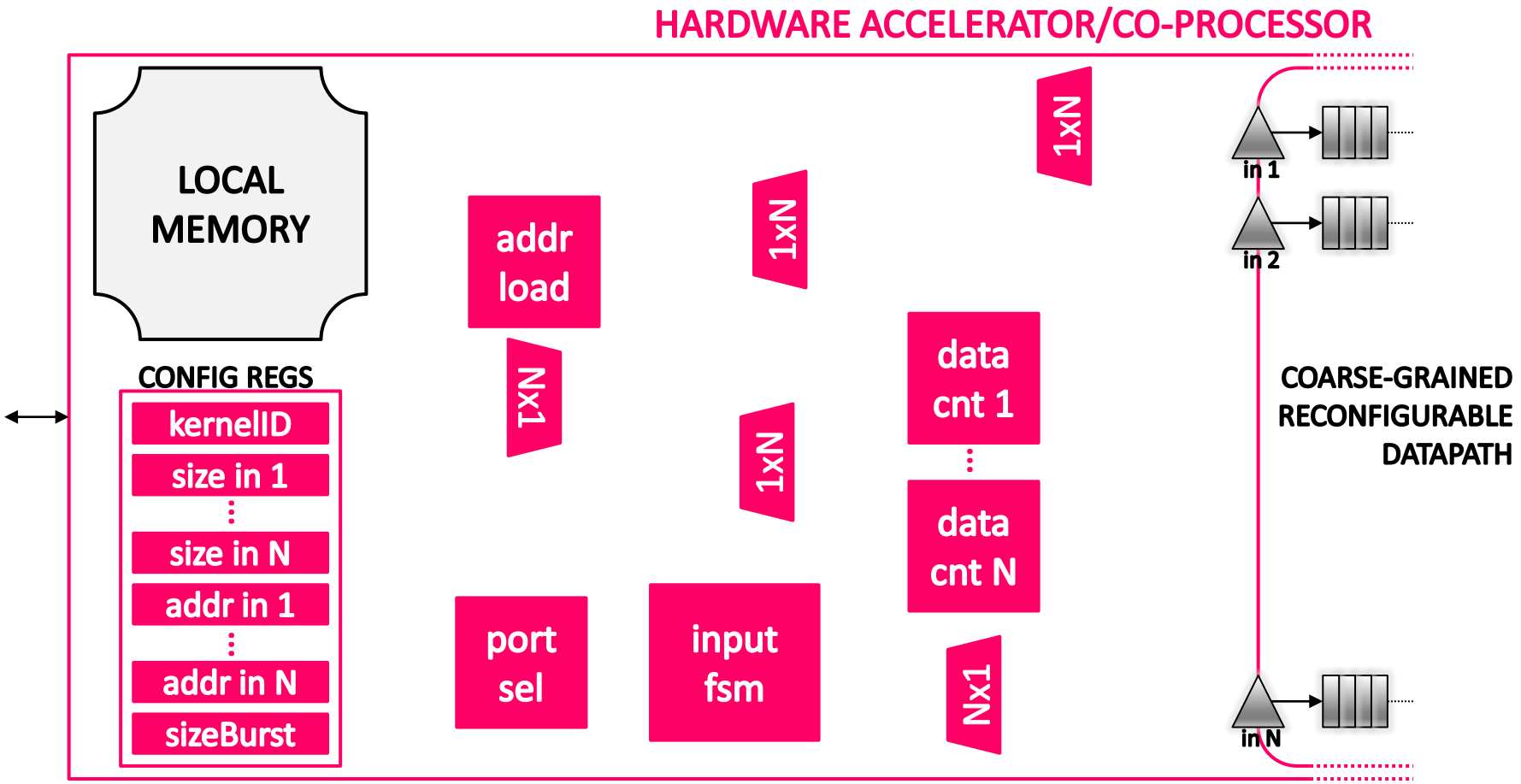
THE TEMPLATE INTERFACE LAYER (TIL)



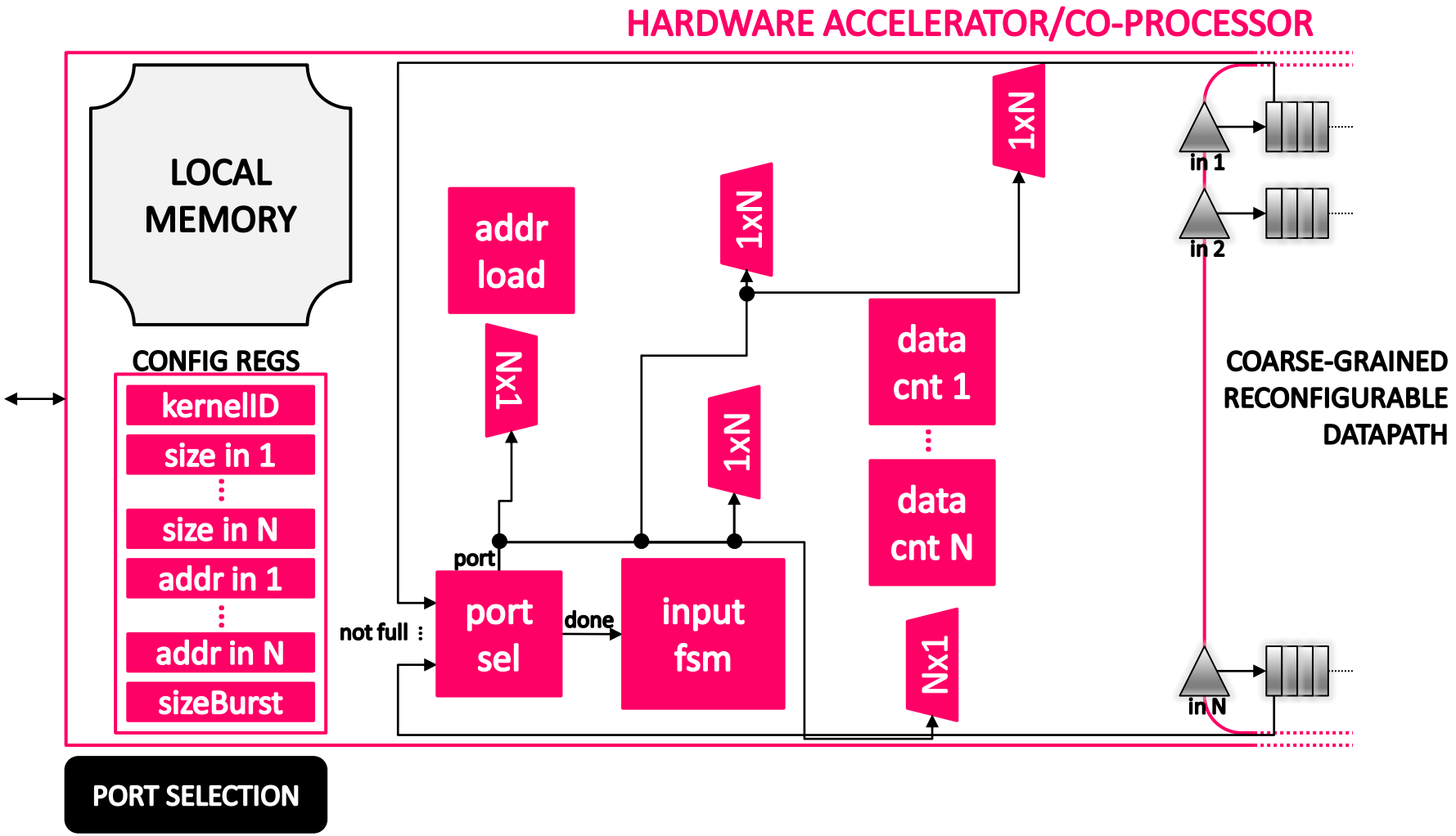
THE TEMPLATE INTERFACE LAYER (TIL)



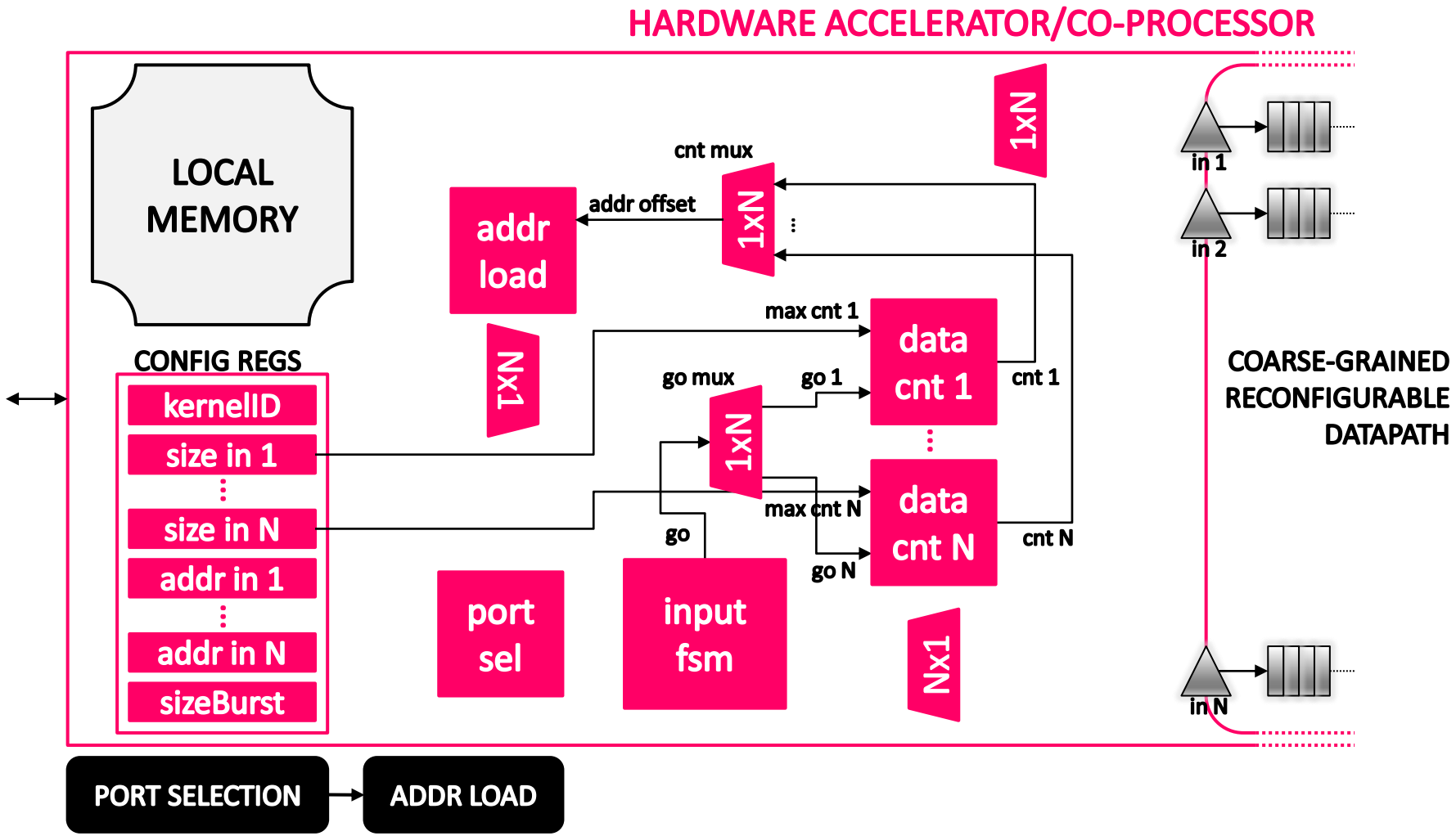
TIL FRONT-END



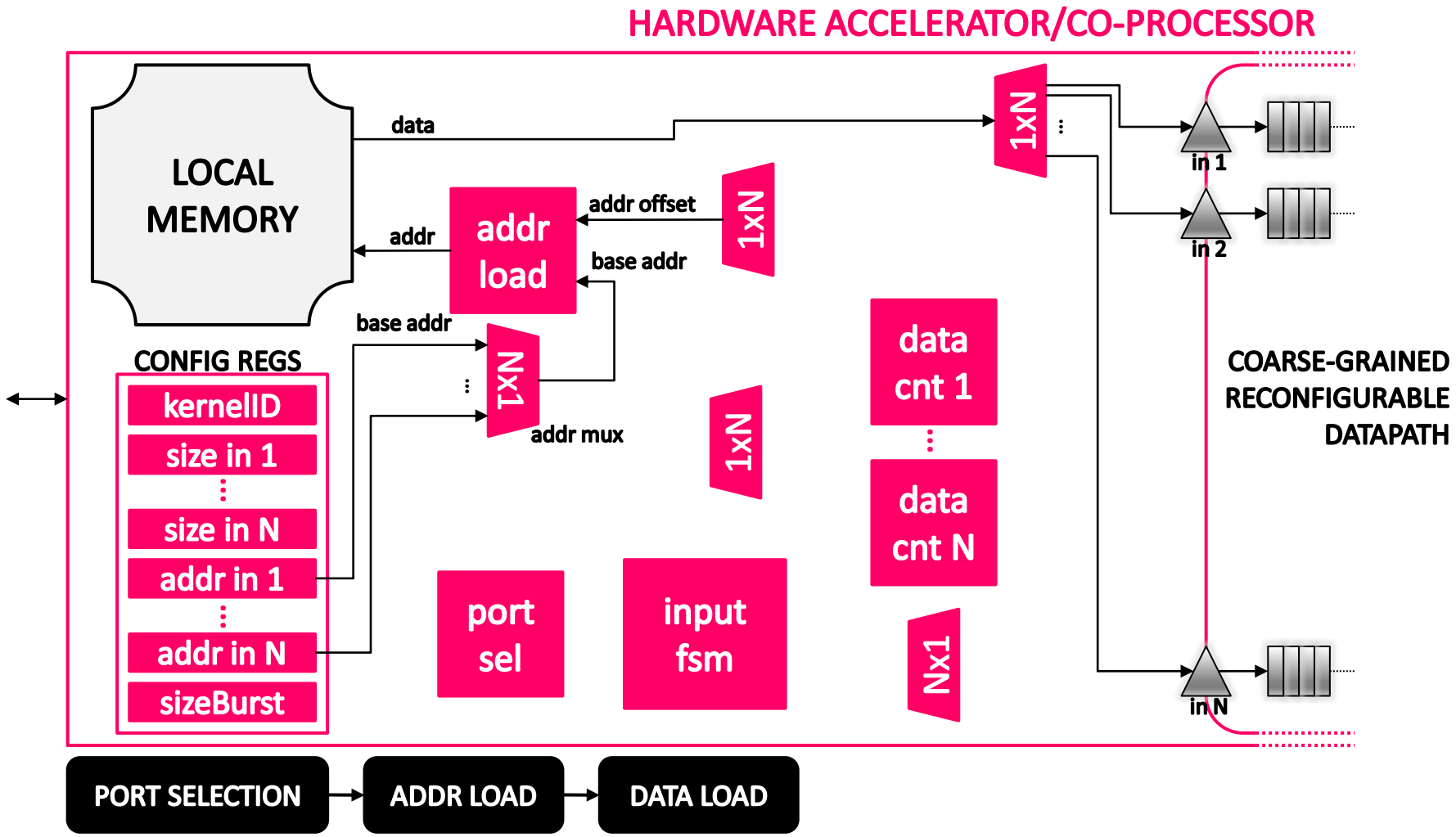
TIL FRONT-END



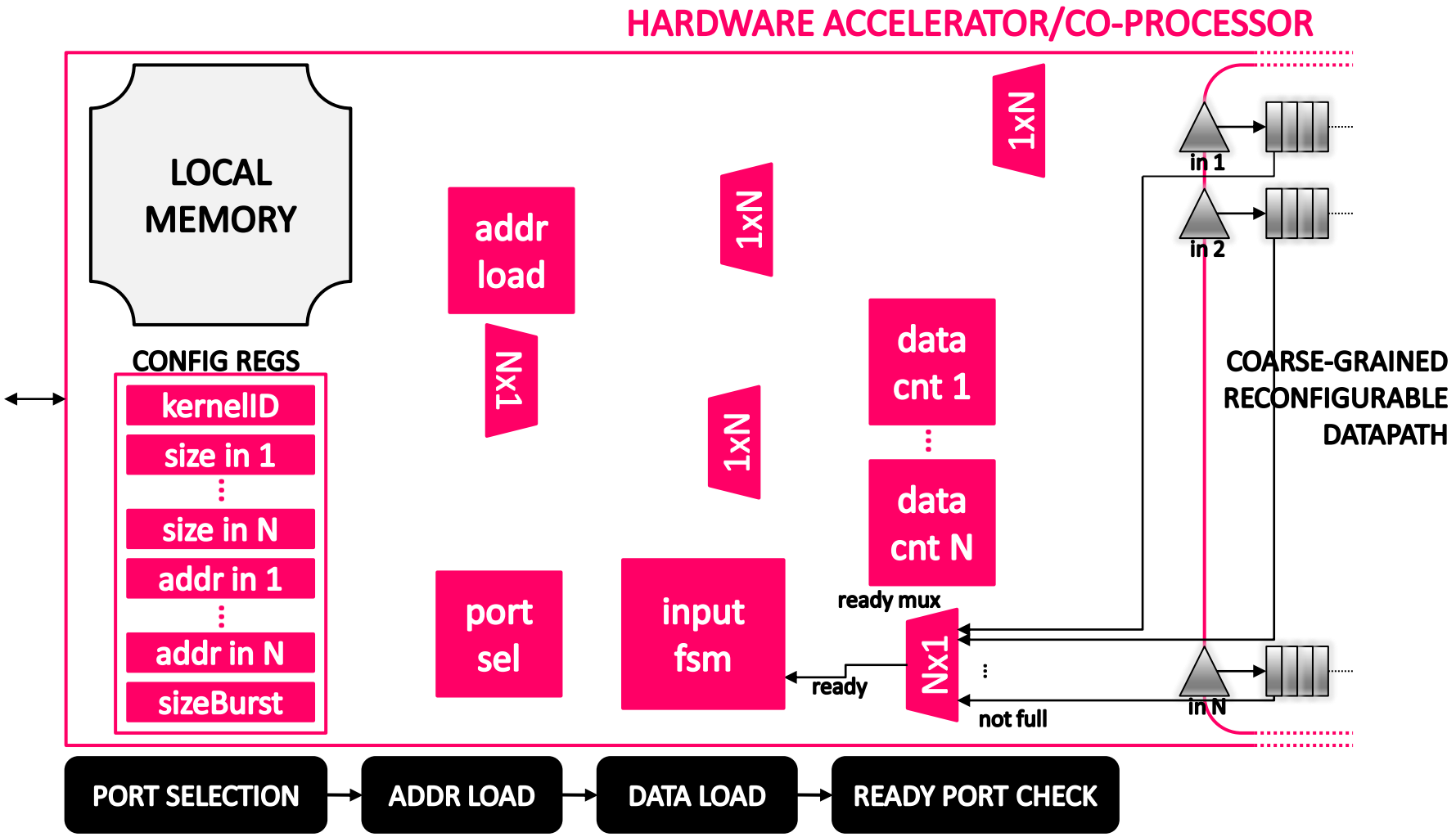
TIL FRONT-END



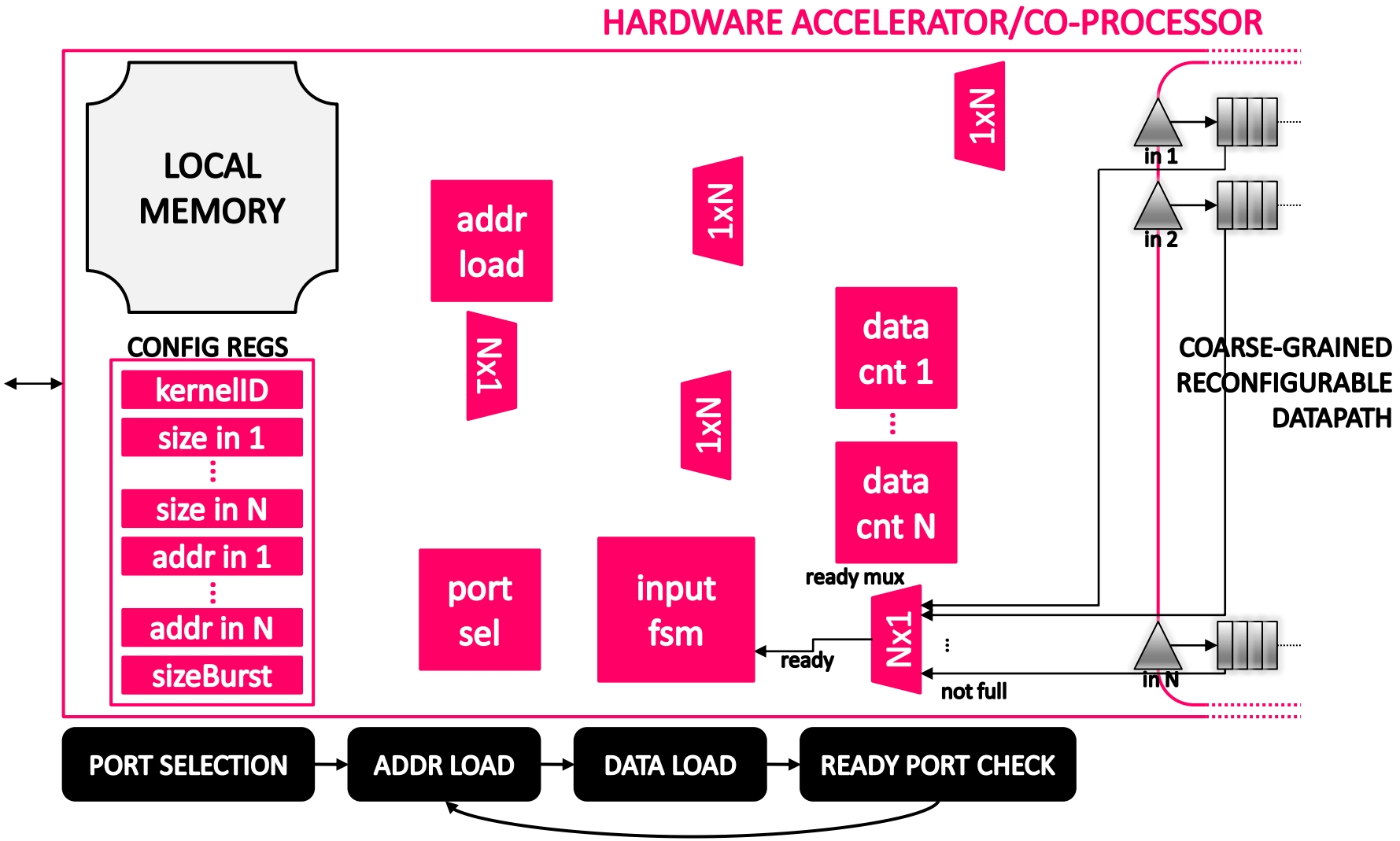
TIL FRONT-END



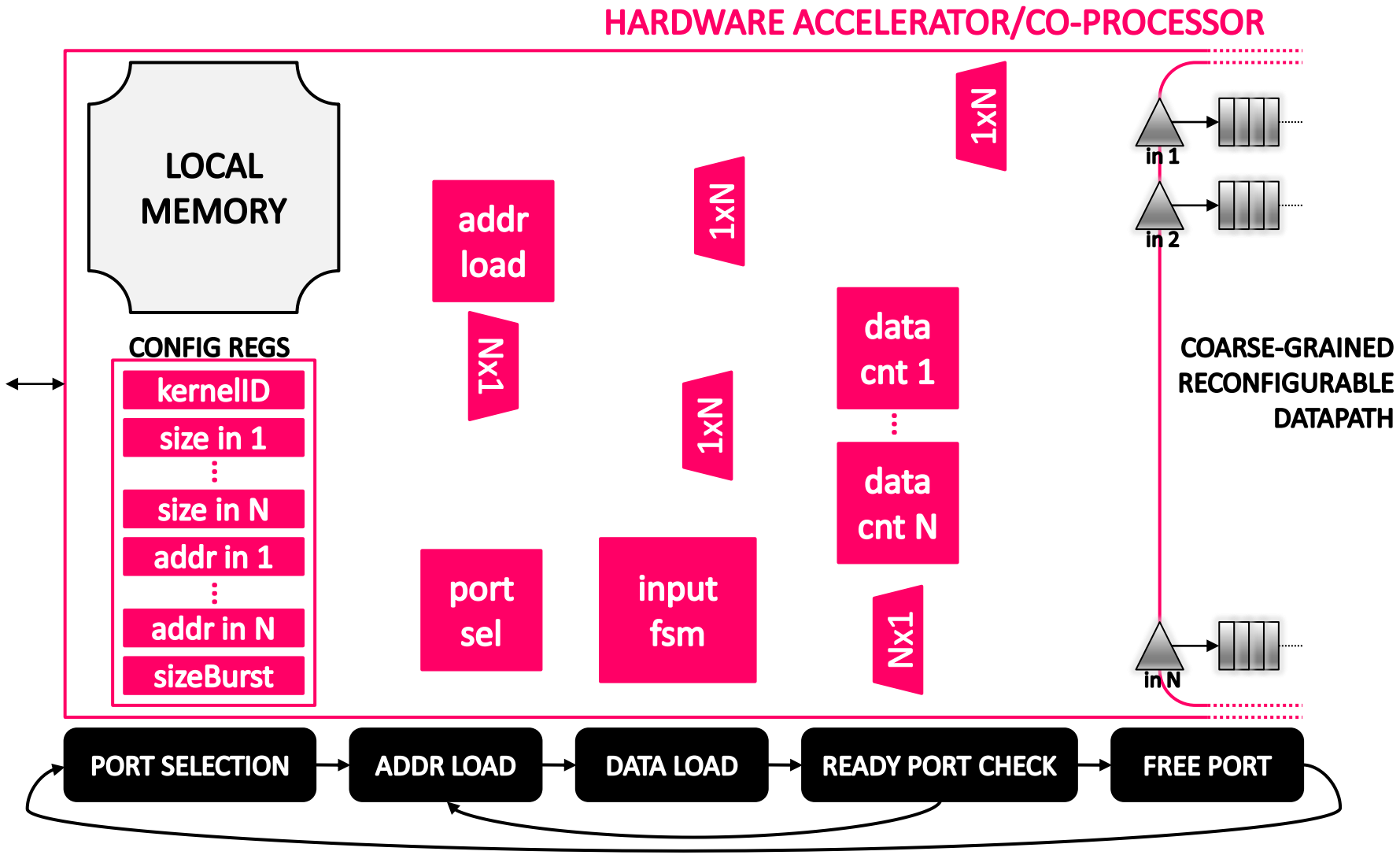
TIL FRONT-END



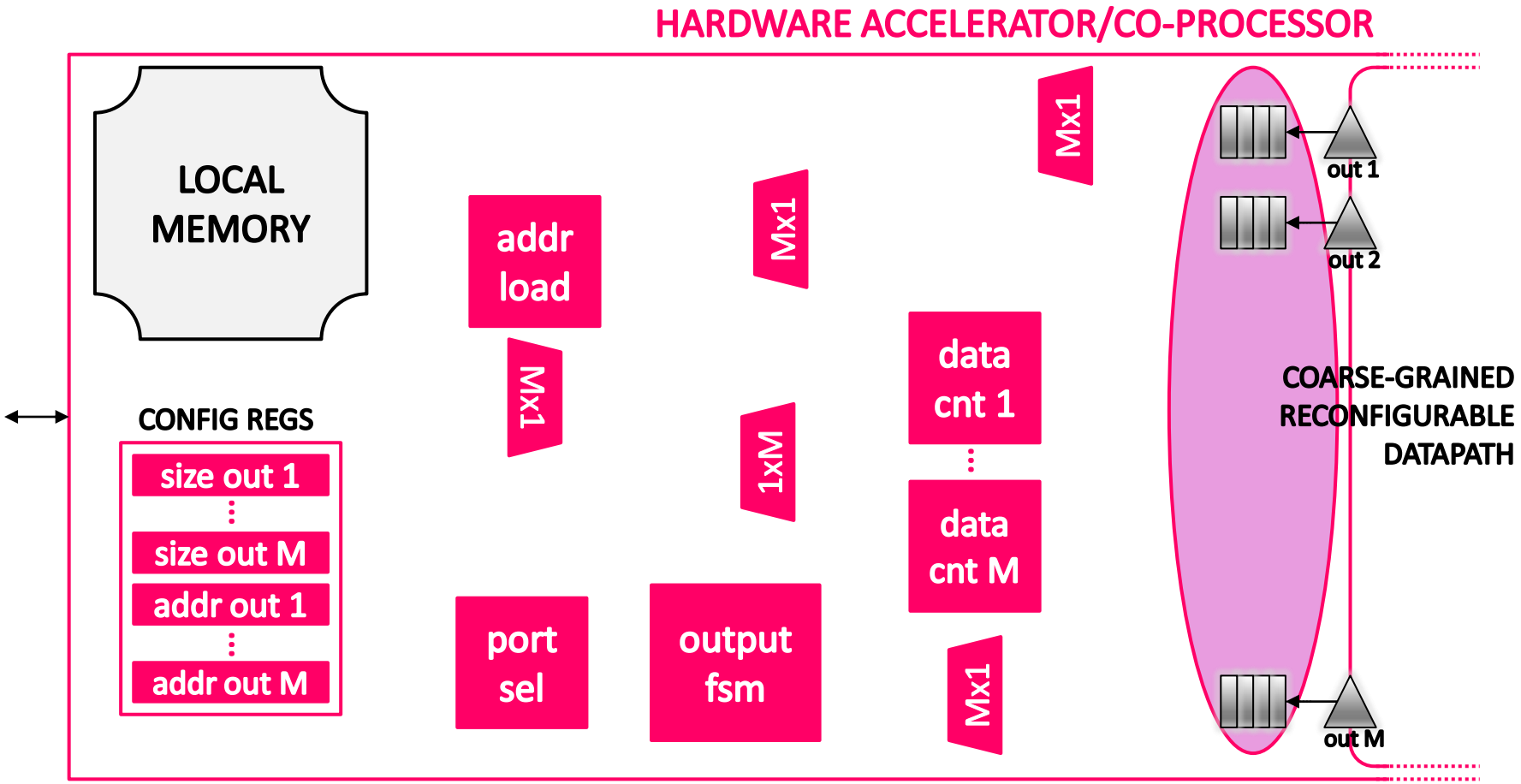
TIL FRONT-END



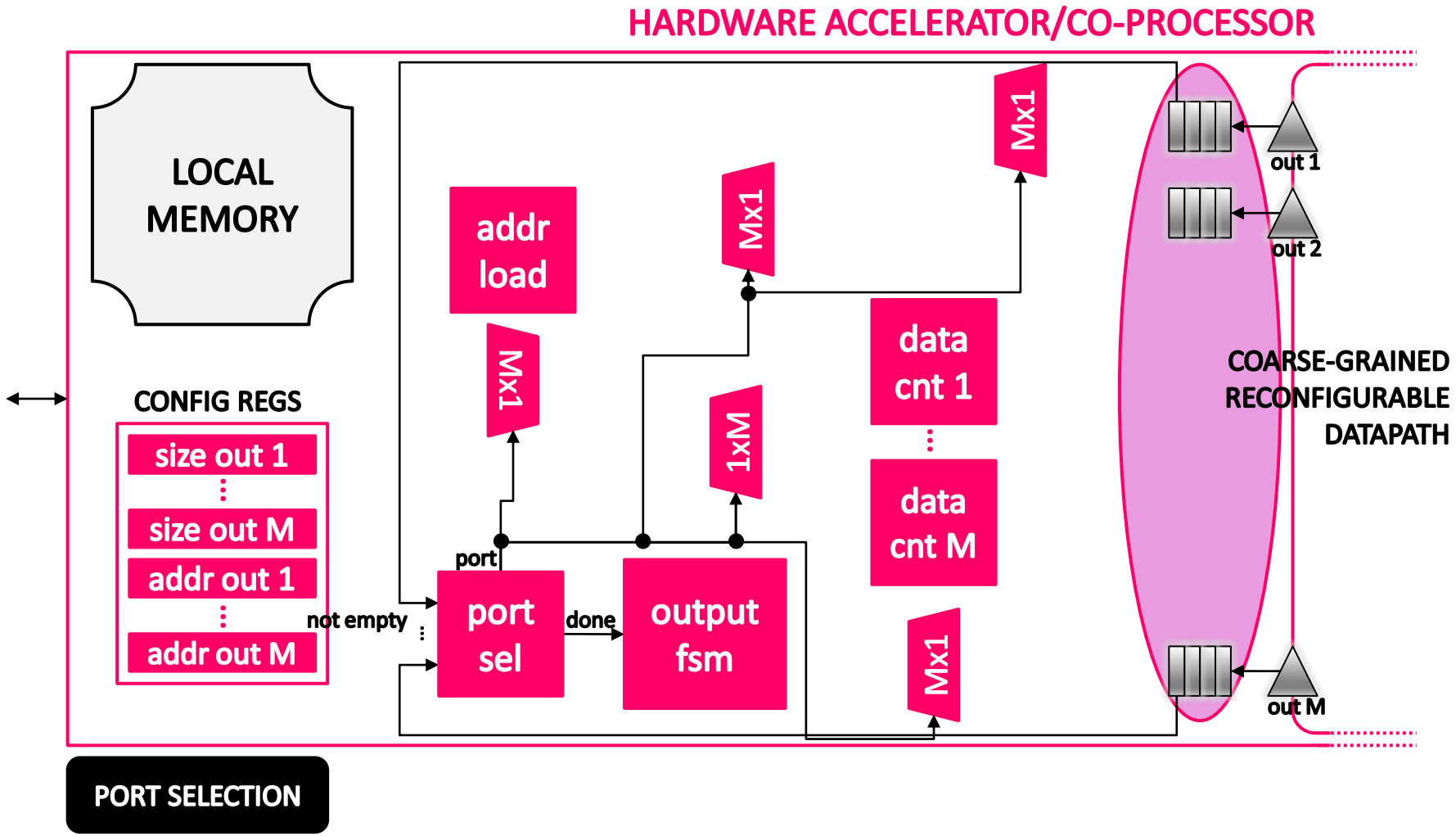
TIL FRONT-END



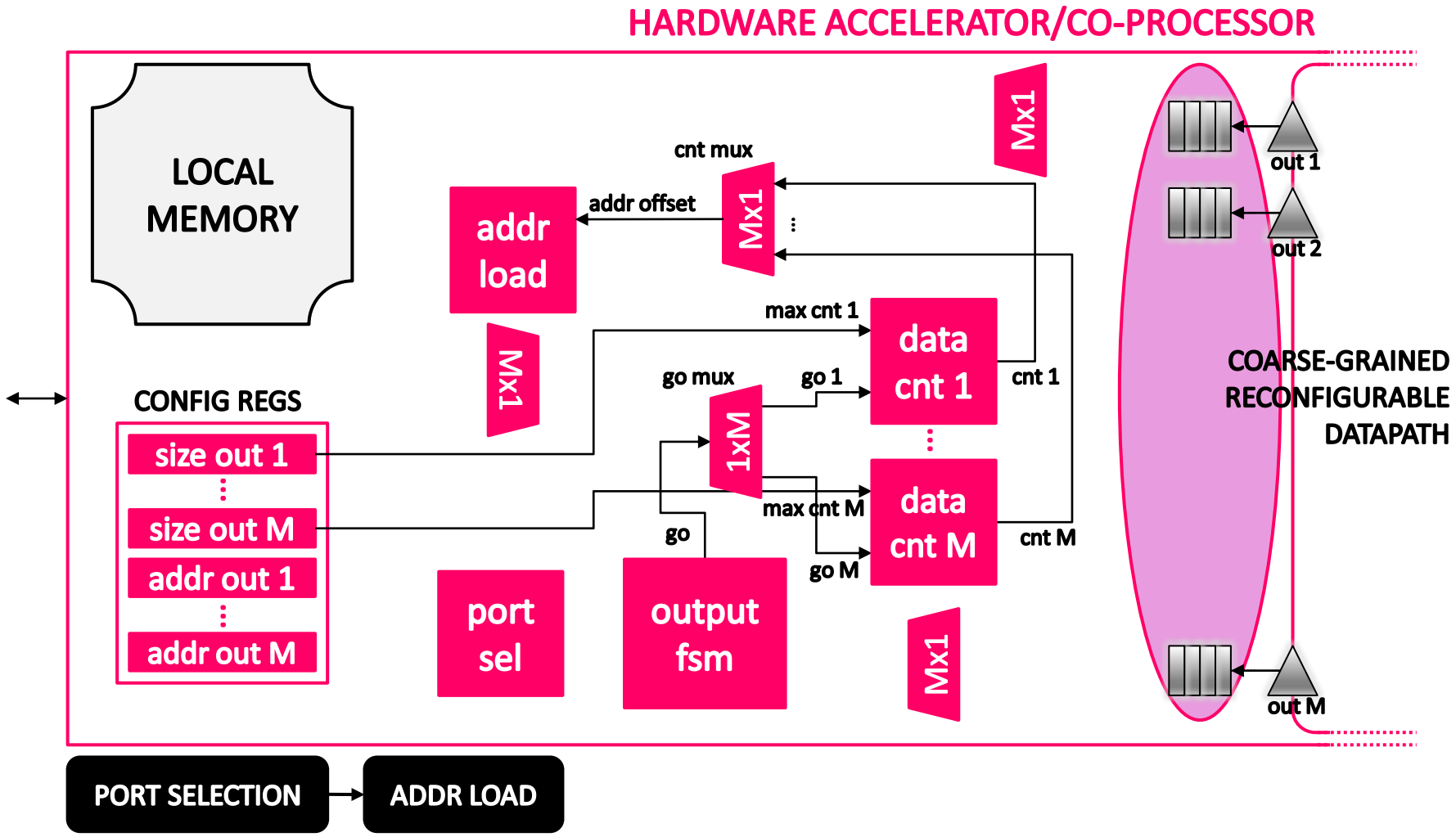
TIL BACK-END


HARDWARE ACCELERATOR/CO-PROCESSOR
LOCAL MEMORY
CONFIG REGS
size out 1
size out M
addr out 1
addr out M
addr load
Mx1
port sel
Mx1
1xM
output fsm
Mx1
data cnt 1
data cnt M
Mx1
out 1
out 2
out M
COARSE-GRAINED RECONFIGURABLE DATAPATH

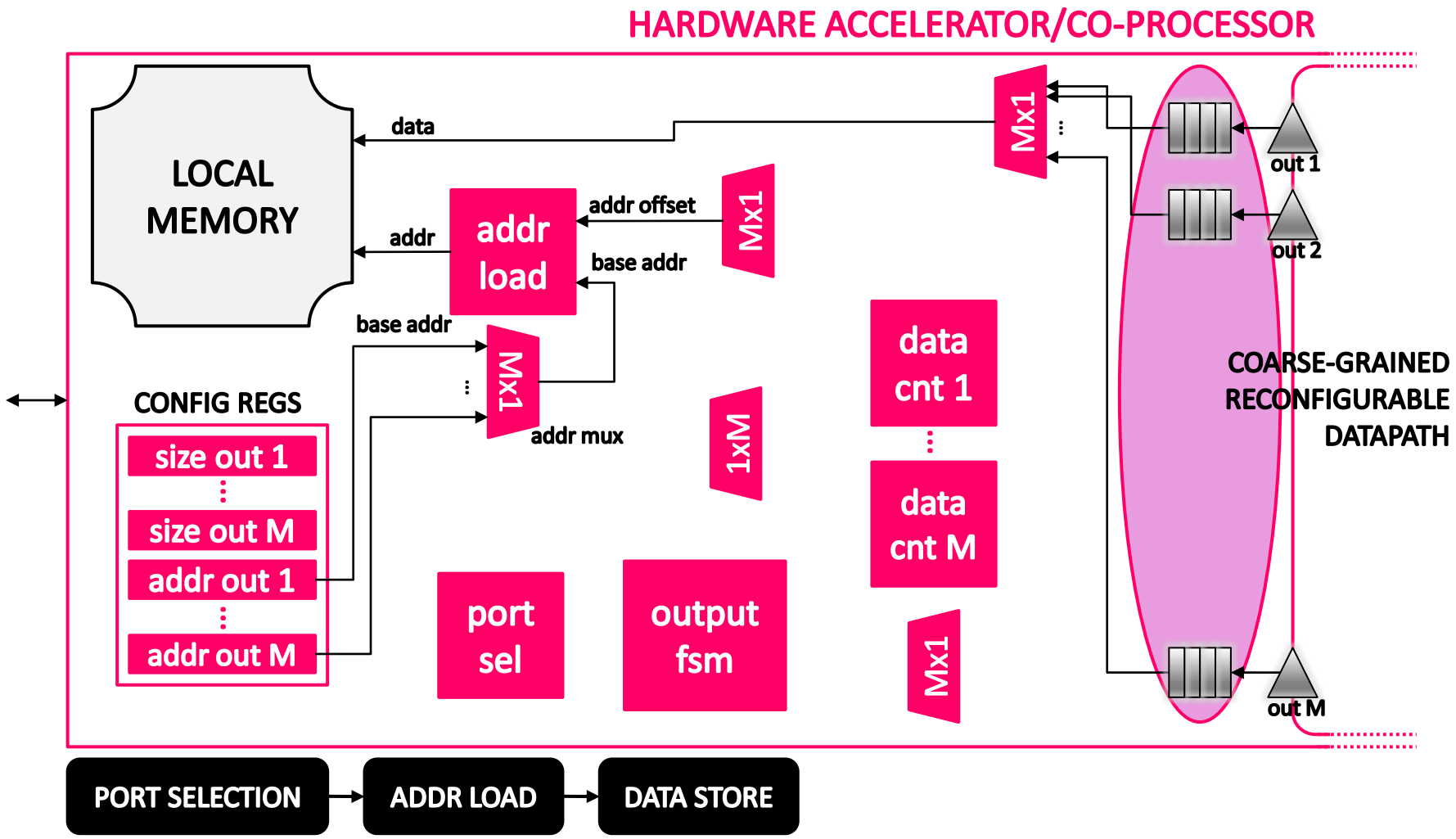
TIL BACK-END


HARDWARE ACCELERATOR/CO-PROCESSOR
LOCAL MEMORY
CONFIG REGS
size out 1
size out M
addr out 1
addr out M
not empty
addr load
Mx1
port
port sel
done
output fsm
1xM
Mx1
data cnt 1
data cnt M
Mx1
Mx1
out 1
out 2
out M
COARSE-GRAINED RECONFIGURABLE DATAPATH
PORT SELECTION

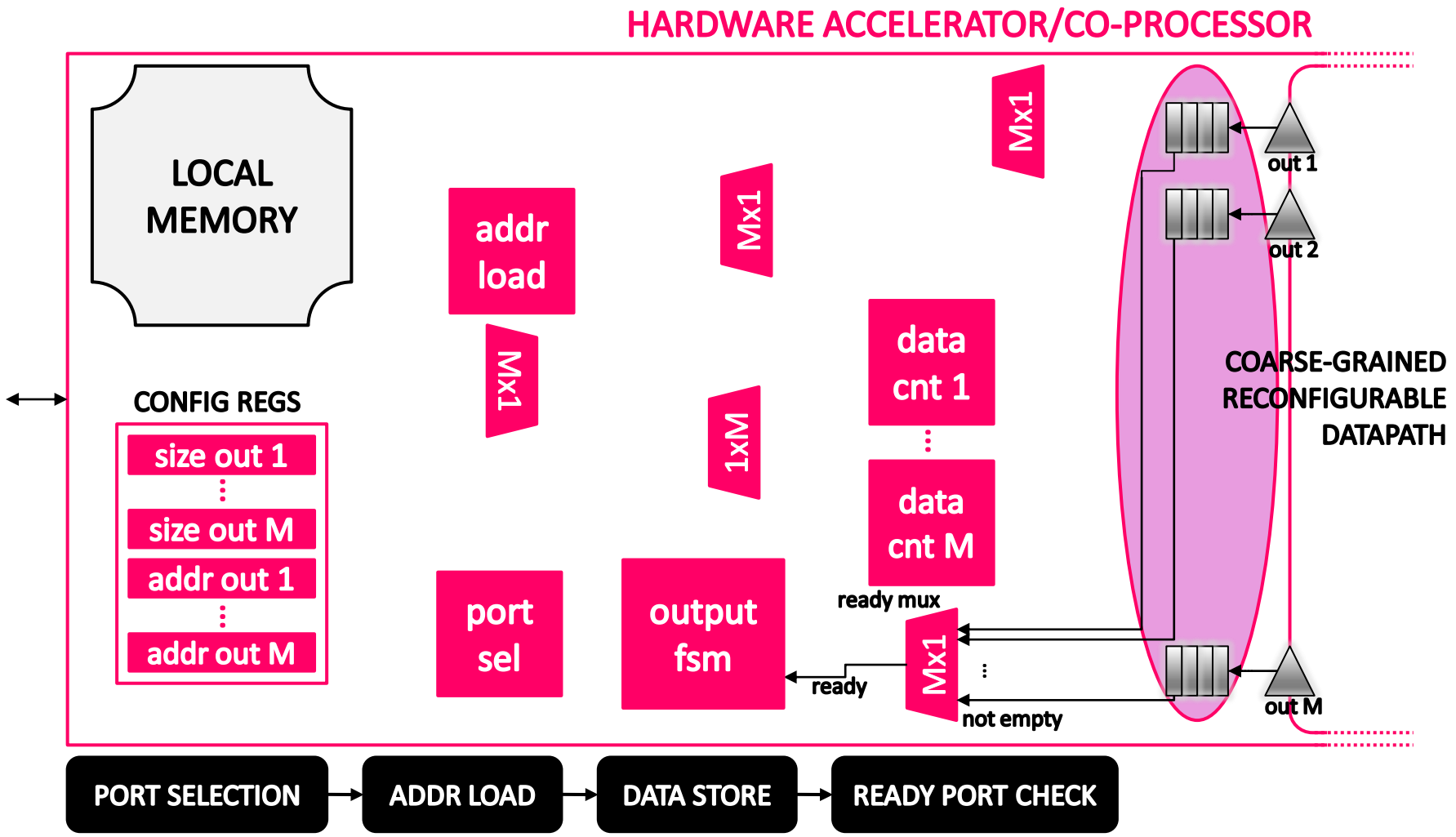
TIL BACK-END


HARDWARE ACCELERATOR/CO-PROCESSOR
LOCAL MEMORY
CONFIG REGS
size out 1
size out M
addr out 1
addr out M
addr load
addr offset
cnt mux
Mx1
Mx1
Mx1
Mx1
max cnt 1
data cnt 1
cnt 1
go mux
go 1
1xM
max cnt M
data cnt M
cnt M
go
go M
port sel
output fsm
COARSE-GRAINED RECONFIGURABLE DATAPATH
out 1
out 2
out M
PORT SELECTION
ADDR LOAD

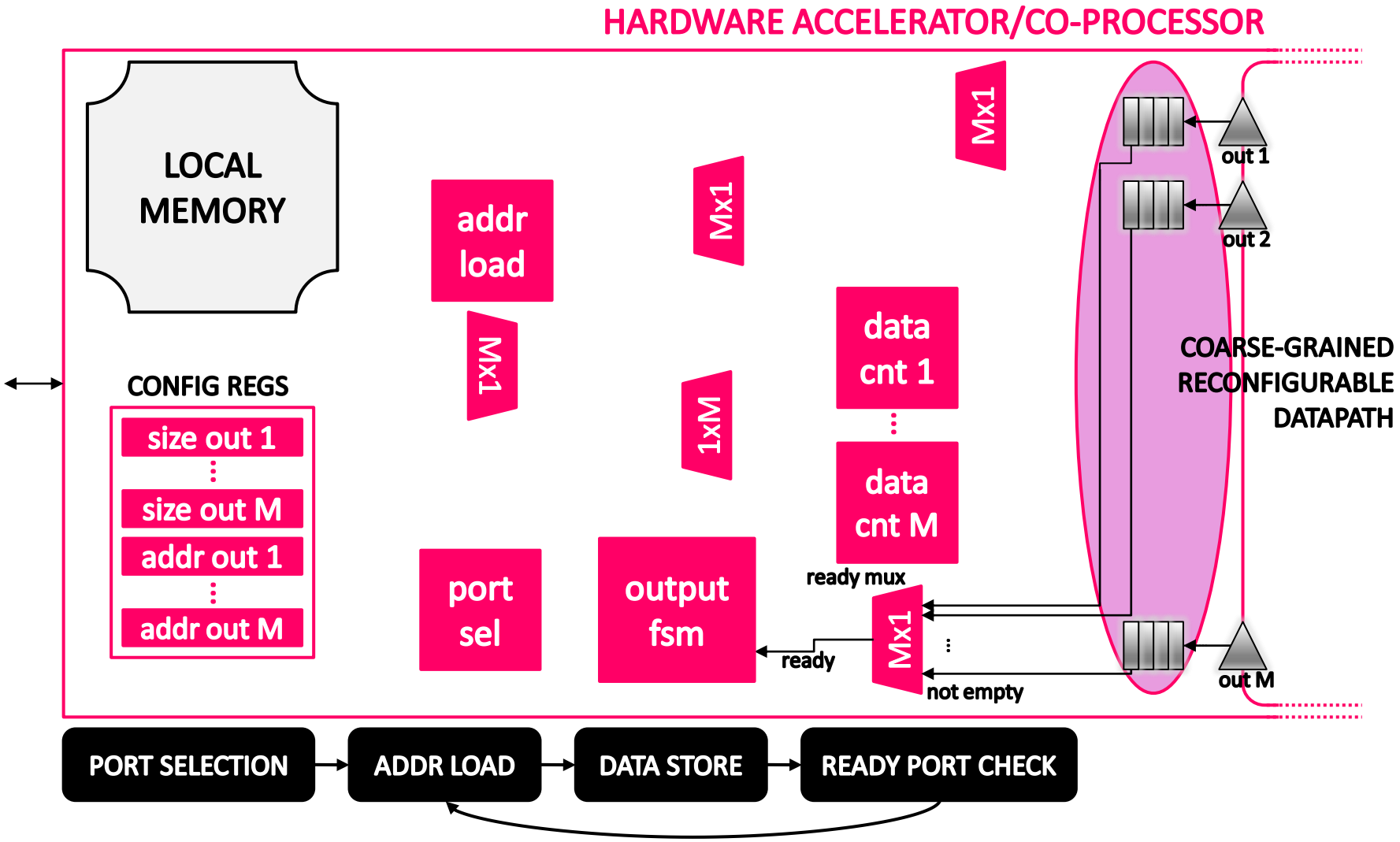
TIL BACK-END


HARDWARE ACCELERATOR/CO-PROCESSOR
LOCAL MEMORY
data
addr
addr load
addr offset
base addr
Mx1
base addr
Mx1
addr mux
CONFIG REGS
size out 1
size out M
addr out 1
addr out M
port sel
output fsm
1xM
data cnt 1
data cnt M
Mx1
Mx1
out 1
out 2
out M
COARSE-GRAINED RECONFIGURABLE DATAPATH
PORT SELECTION
ADDR LOAD
DATA STORE

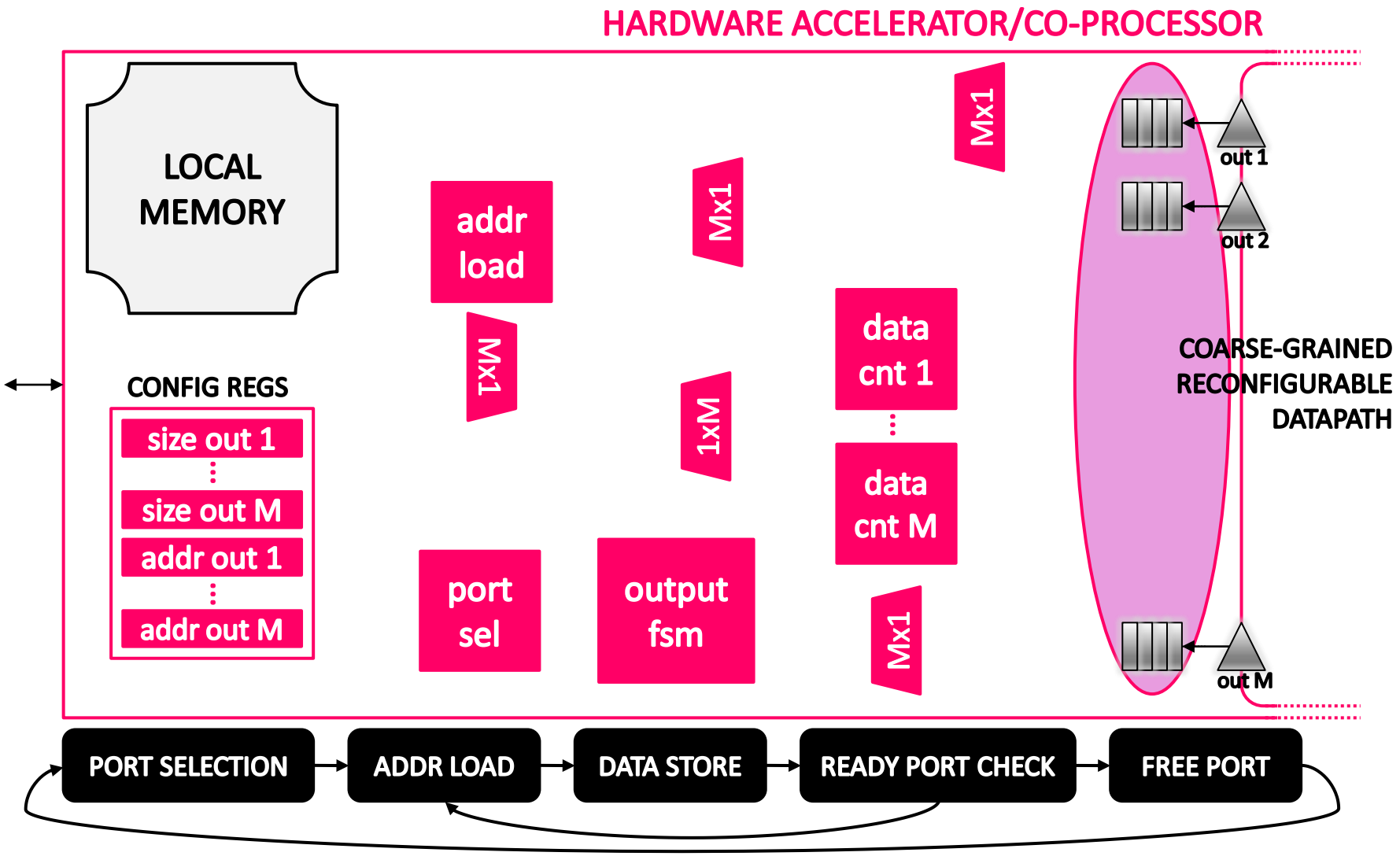
TIL BACK-END


HARDWARE ACCELERATOR/CO-PROCESSOR
LOCAL MEMORY
CONFIG REGS
size out 1
size out M
addr out 1
addr out M
addr load
Mx1
port sel
Mx1
1xM
output fsm
data cnt 1
data cnt M
ready mux
Mx1
ready
not empty
Mx1
out 1
out 2
out M
COARSE-GRAINED RECONFIGURABLE DATAPATH
PORT SELECTION
ADDR LOAD
DATA STORE
READY PORT CHECK

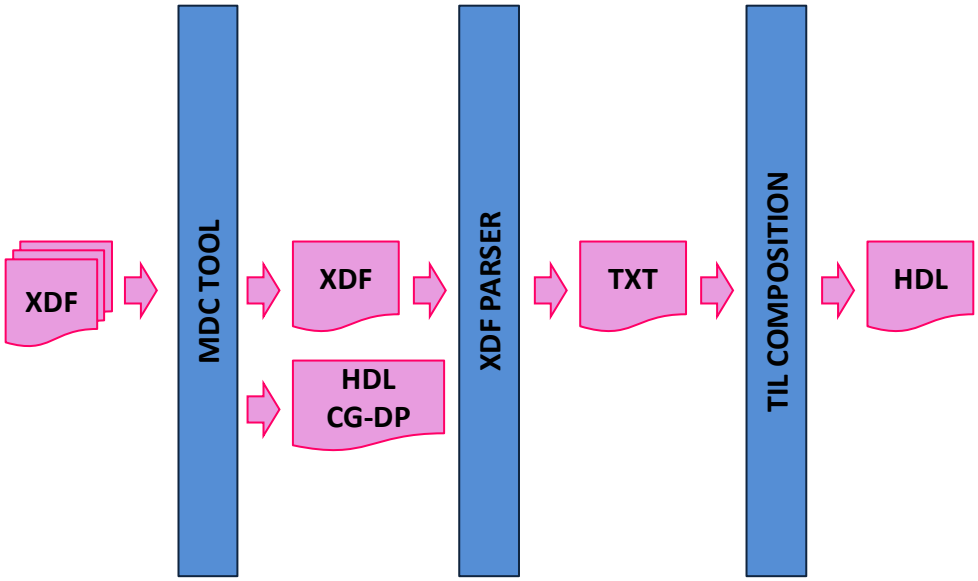
TIL BACK-END


HARDWARE ACCELERATOR/CO-PROCESSOR
LOCAL MEMORY
CONFIG REGS
size out 1
size out M
addr out 1
addr out M
addr load
Mx1
Mx1
1xM
Mx1
data cnt 1
data cnt M
port sel
output fsm
ready mux
ready
not empty
out 1
out 2
out M
COARSE-GRAINED RECONFIGURABLE DATAPATH
PORT SELECTION
ADDR LOAD
DATA STORE
READY PORT CHECK

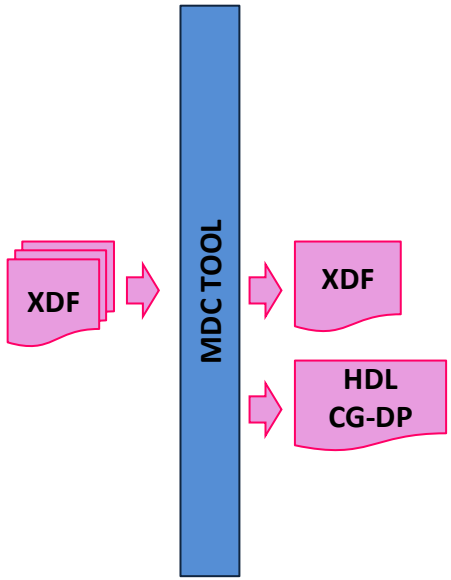
TIL BACK-END

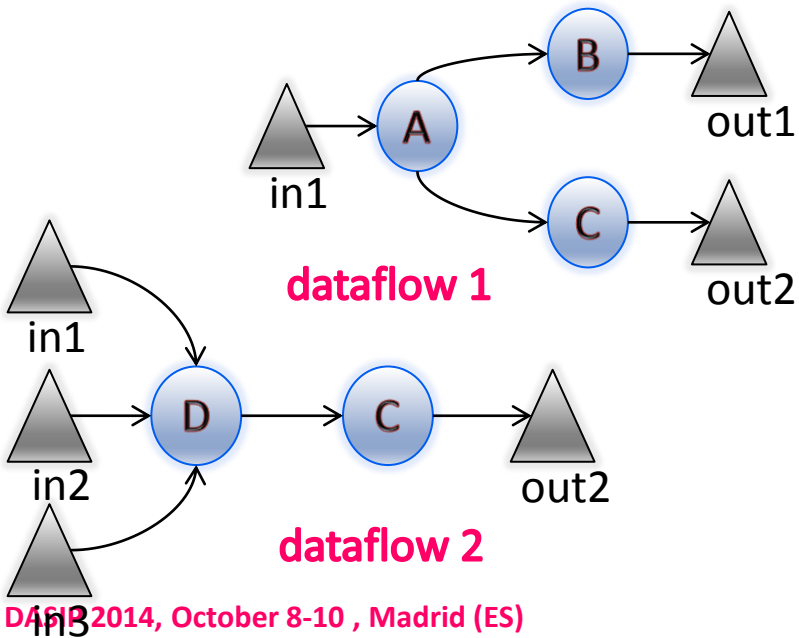

HARDWARE ACCELERATOR/CO-PROCESSOR
LOCAL MEMORY
CONFIG REGS
size out 1
size out M
addr out 1
addr out M
addr load
Mx1
port sel
Mx1
1xM
output fsm
Mx1
data cnt 1
data cnt M
Mx1
out 1
out 2
out M
COARSE-GRAINED RECONFIGURABLE DATAPATH
PORT SELECTION
ADDR LOAD
DATA STORE
READY PORT CHECK
FREE PORT

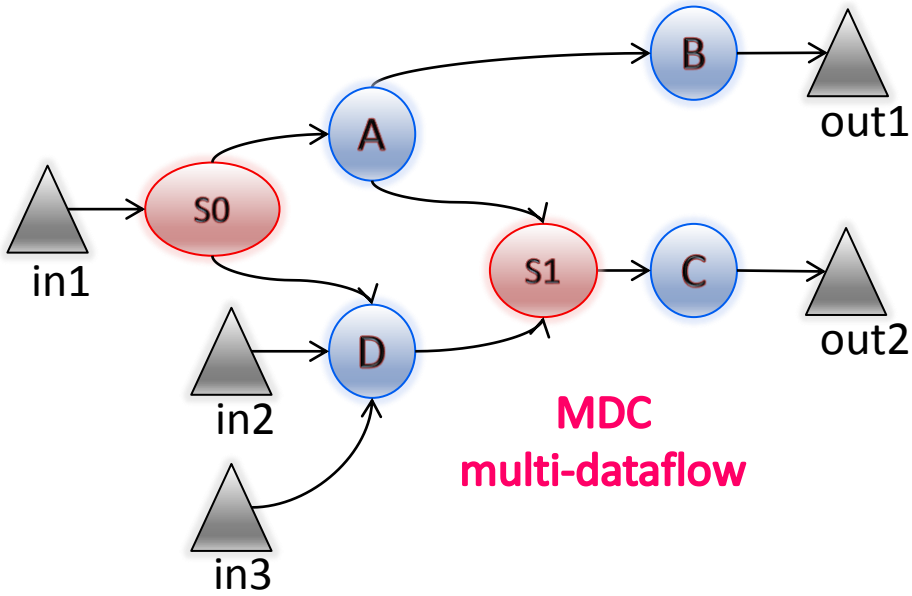
TIL:AUTOMATIC GENERATION



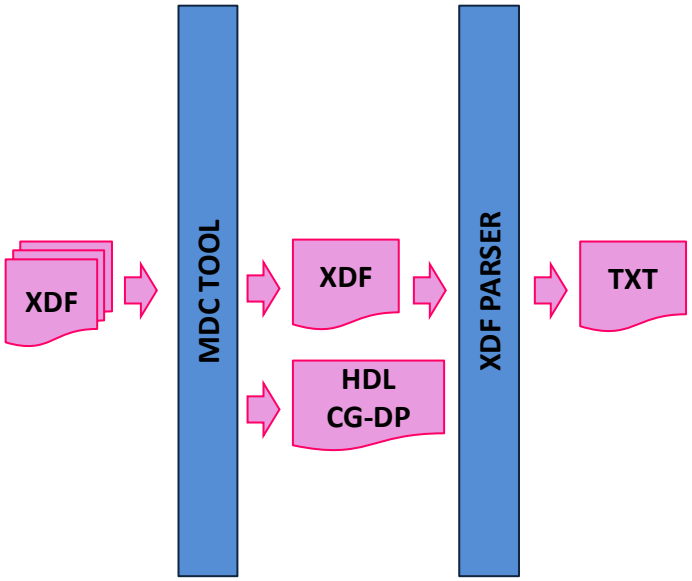
TIL:AUTOMATIC GENERATION

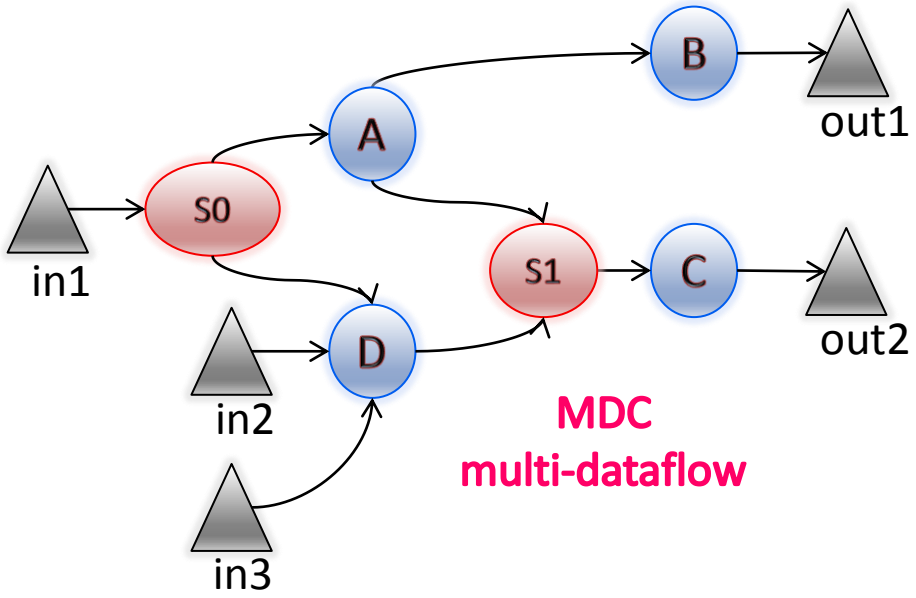






TIL:AUTOMATIC GENERATION





MDC
multi-dataflow

EXTERNAL INTERFACE

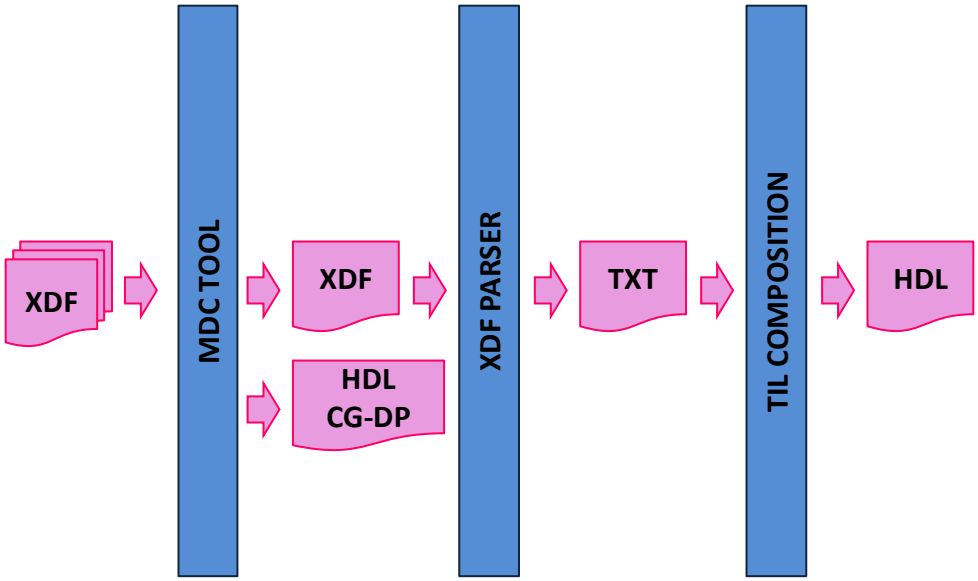
I ports **number** = 3

O ports **number** = 2

I/O ports **depth** = X

I/O ports **burst** of tokens = N

TIL:AUTOMATIC GENERATION



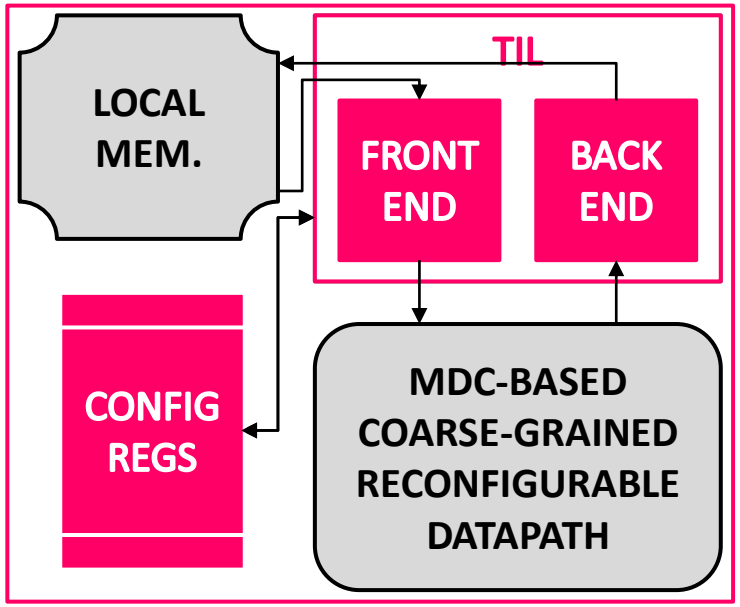
EXTERNAL INTERFACE

I ports **number** = 3

I ports **number** = 2

I/O ports **depth** = X

I/O ports **burst** of tokens = N



TIL:AUTOMATIC GENERATION

resource	number of resource instances		type
	3 inputs 2 outputs	N inputs M outputs	
register	12	2+N*2+M*2	port-dependent
counter	6	1+N+M	port-dependent
mux 2x1	4	4	extendable
mux Nx1	2	2	extendable
mux Mx1	3	3	extendable
demux 1xN	3	3	extendable
demux 1xM	3	3	extendable
FIFO	2	M	port-dependent
port selector	2	2	extendable
addr generator	2	2	extendable
FSM	2	2	fixed

OUTLINE: PERFORMANCE ASSESSMENT

- Introduction:
  - Problem statement
  - Background
  - Goals
- Co-processing units generation:
  - Approach and baseline Multi-Dataflow Composer
  - Template Interface Layer: hardware and automatic composition
- Performance assessment
  - Use-case scenario
  - Results
- Final remarks and future directions

ACHIEVED RESULTS: TIL ADAPTIVITY

number of I/O ports (value=M=N)	resource (% on available)				
	Slice Regs (207360)	Slice LUTs (207360)	BUFGs (32)	BRAMs (288)	frequency [MHz]
1	153 (0,1)	277 (0,1)	1 (3,1)	65 (22,6)	243,8
2	261 (0,1)	430 (0,2)	1 (3,1)	65 (22,6)	243,8
4	475 (0,2)	751 (0,4)	1 (3,1)	65 (22,6)	239,0
8	901 (0,4)	1558 (0,8)	1 (3,1)	65 (22,6)	208,9
16	1757 (0,9)	2760 (1,3)	1 (3,1)	65 (22,6)	197,6
32	4353 (1,7)	5339 (2,6)	2 (6,3)	65 (22,6)	158,4

Results have been retrieved through the Xilinx Synthesis Technology tool targeting a Virtex 5 330 FPGA board. Only the co-processing unit without any coarse-grained reconfigurable datapath has been considered.

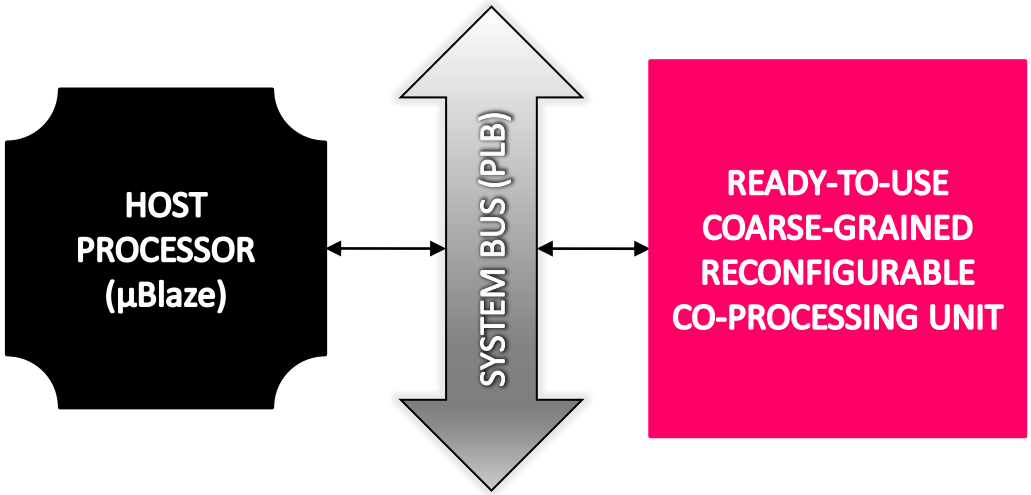
ACHIEVED RESULTS: TIL ADAPTIVITY

number of I/O ports (value=M=N)	resource (% on available)				
	Slice Regs (207360)	Slice LUTs (207360)	BUFGs (32)	BRAMs (288)	frequency [MHz]
1	153 (0,1)	277 (0,1)	1 (3,1)	65 (22,6)	243,8
2	261 (0,1)	430 (0,2)	1 (3,1)	65 (22,6)	243,8
4	475 (0,2)	751 (0,4)	1 (3,1)	65 (22,6)	239,0
8	901 (0,4)	1558 (0,8)	1 (3,1)	65 (22,6)	208,9
16	1757 (0,9)	2760 (1,3)	1 (3,1)	65 (22,6)	197,6
32	4353 (1,7)	5339 (2,6)	2 (6,3)	65 (22,6)	158,4

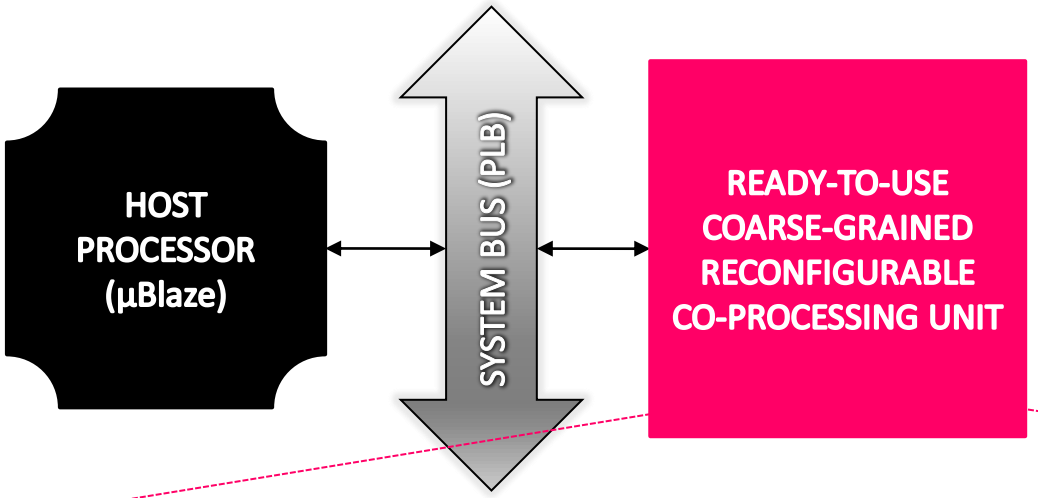
Results have been retrieved through the Xilinx Synthesis Technology tool targeting a Virtex 5 330 FPGA board. Only the co-processing unit without any coarse-grained reconfigurable datapath has been considered.

SLICES + 80%
FREQ - 8%

ACHIEVED RESULTS: USE CASE

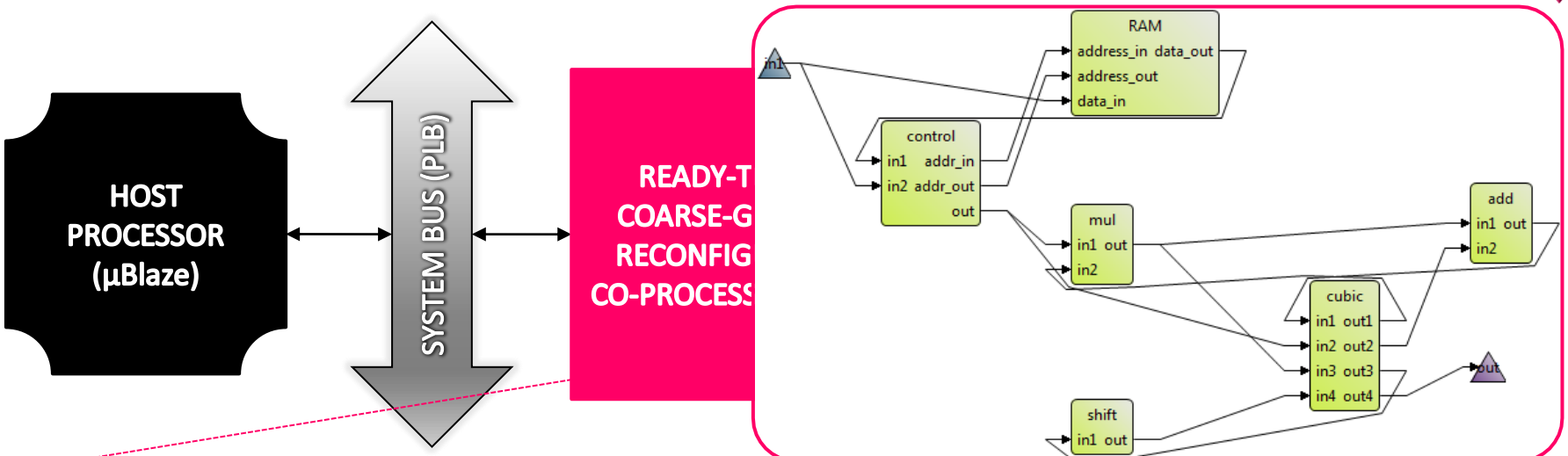


ACHIEVED RESULTS: USE CASE



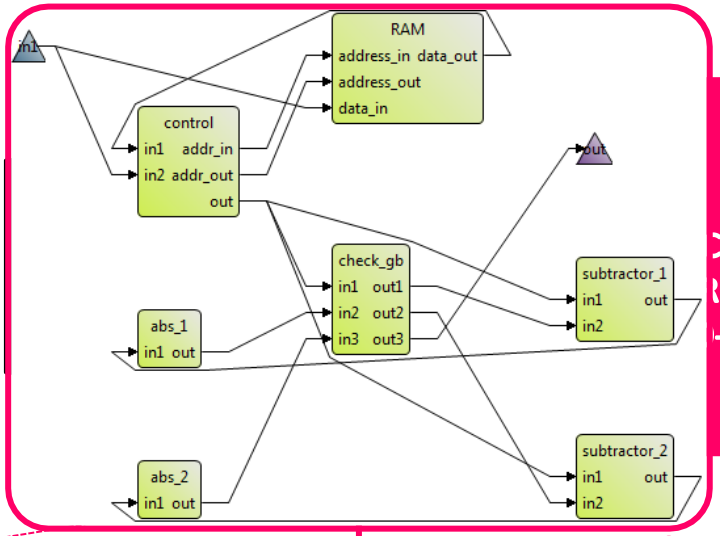
kernel (ID)	sorter (1)	max_min (2)	rgb2ycc (3)	ycc2rgb (4)	abs (5)	corr (6)	sbwlabel (7)	chgb (8)	cubic_conv (9)	median (10)	cubic (11)	filter (12)	clip_f (13)	inner (14)	mdiv (15)	nbit (16)	sign (17)	smear (18)	rgb2yuv (19)	yuv2rgb (20)
antialiasing	X	X	X	X	X	X														
zoom		X			X		X	X	X	X	X									
motion estimation																X		X		
deblocking deringing		X										X	X	X	X		X		X	X

ACHIEVED RESULTS: USE CASE

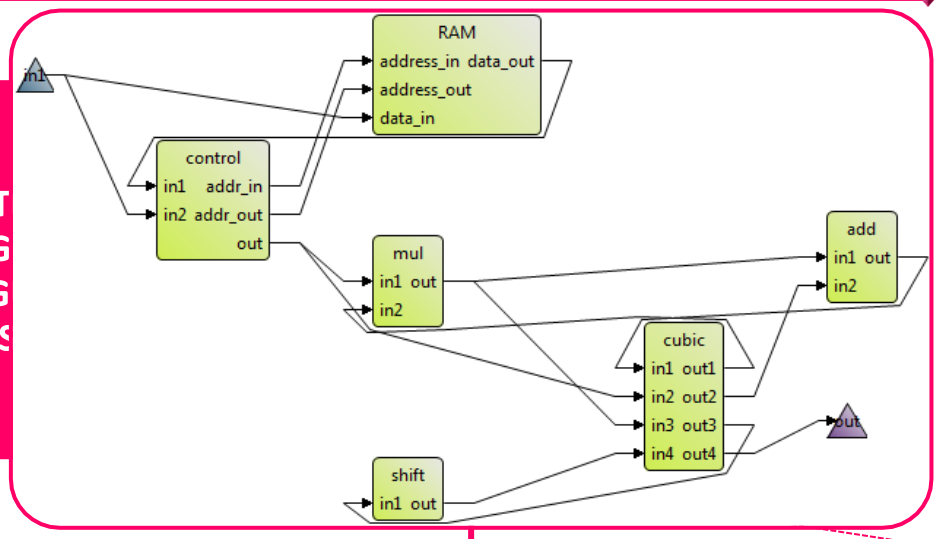


kernel (ID)	sorter (1)	max_min (2)	rgb2ycc (3)	ycc2rgb (4)	abs (5)	corr (6)	sbwlabel (7)	chgb (8)	cubic_conv (9)	median (10)	cubic (11)	filter (12)	clip_f (13)	inner (14)	mdiv (15)	nbit (16)	sign (17)	smear (18)	rgb2yuv (19)	yuv2rgb (20)
antialiasing	X	X	X	X	X	X														
zoom		X			X		X	X	X	X	X									
motion estimation																X		X		
deblocking deringing		X										X	X	X	X		X		X	X

ACHIEVED RESULTS: USE CASE

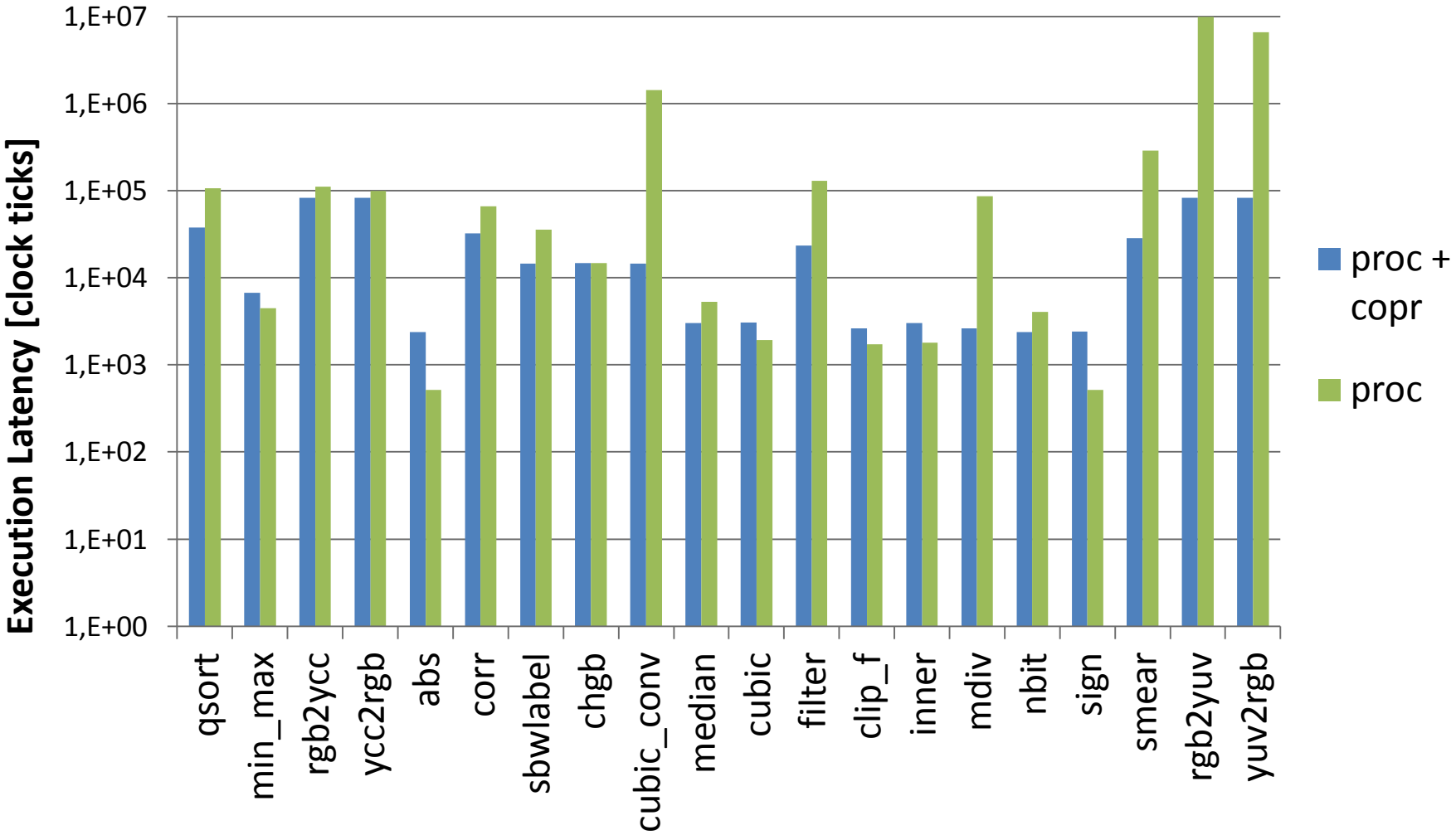


READY-T
COARSE-G
RECONFIG
-PROCESS



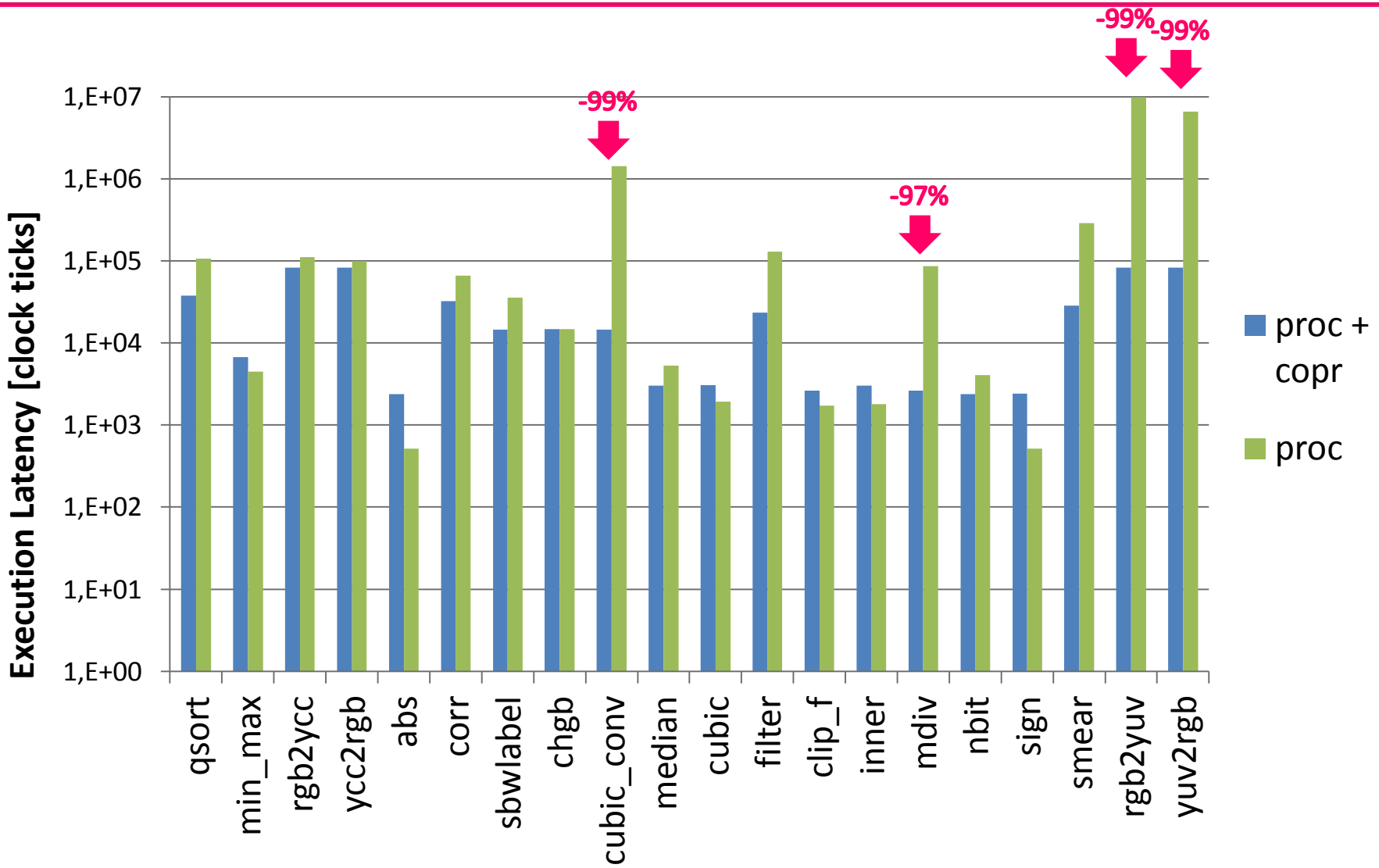
kernel (ID)	sorter (1)	max_min (2)	rgb2ycc (3)	ycc2rgb (4)	abs (5)	corr (6)	sbwlabel (7)	chgb (8)	cubic_conv (9)	median (10)	cubic (11)	filter (12)	clip_f (13)	inner (14)	mdiv (15)	nbit (16)	sign (17)	smear (18)	rgb2yuv (19)	yuv2rgb (20)
antialiasing	X	X	X	X	X	X														
zoom		X			X		X	X	X	X	X									
motion estimation																X		X		
deblocking deringing		X										X	X	X	X		X		X	X

ACHIEVED RESULTS: PERFOMANCES (1)



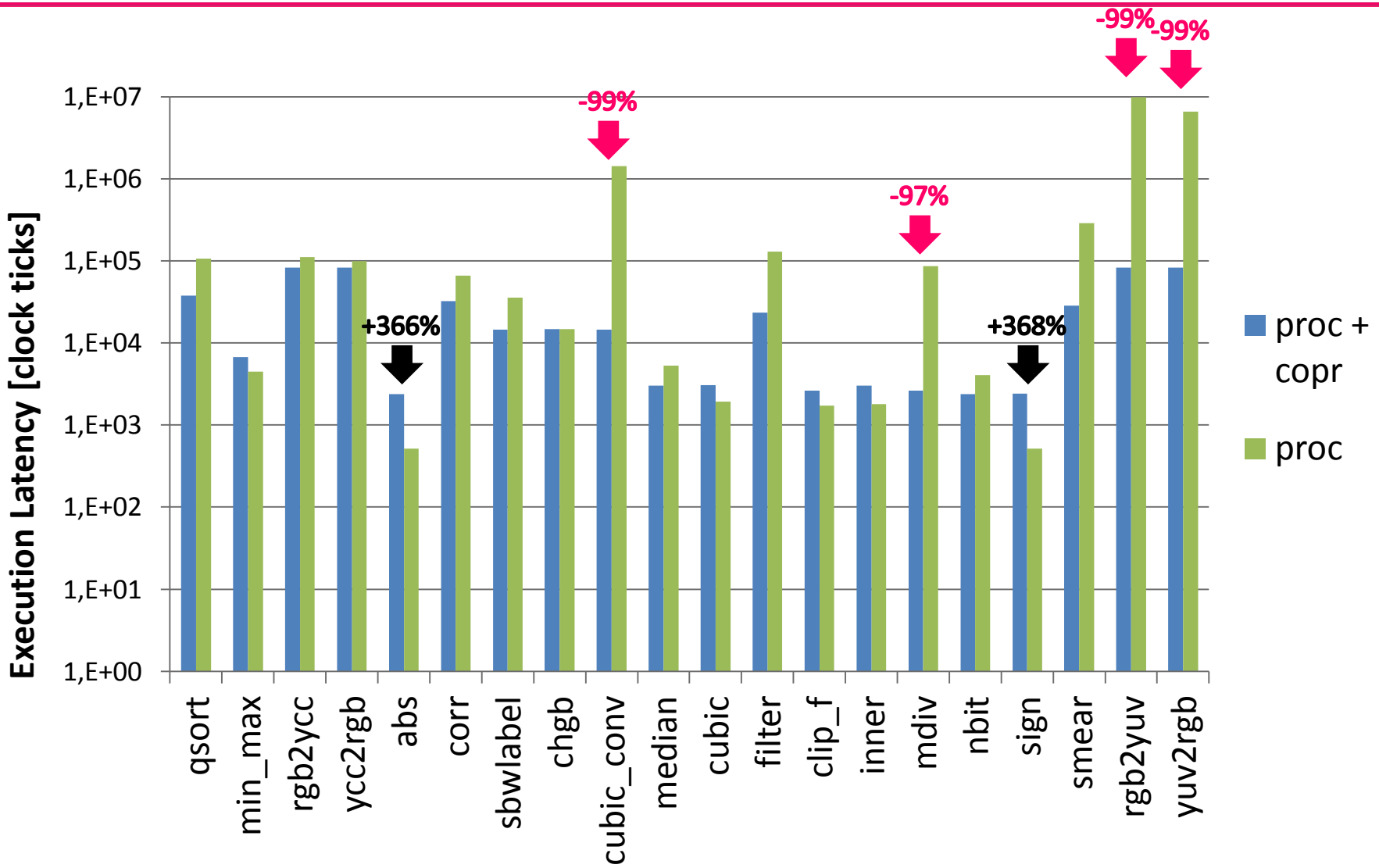
Results have been retrieved running the kernels in the targeted Virtex 5 330 FPGA board at an operating frequency of 125 MHz for the host processor and of 65 MHz (fixed by the CG reconfigurable datapath) for the co-processor.

ACHIEVED RESULTS: PERFOMANCES (1)



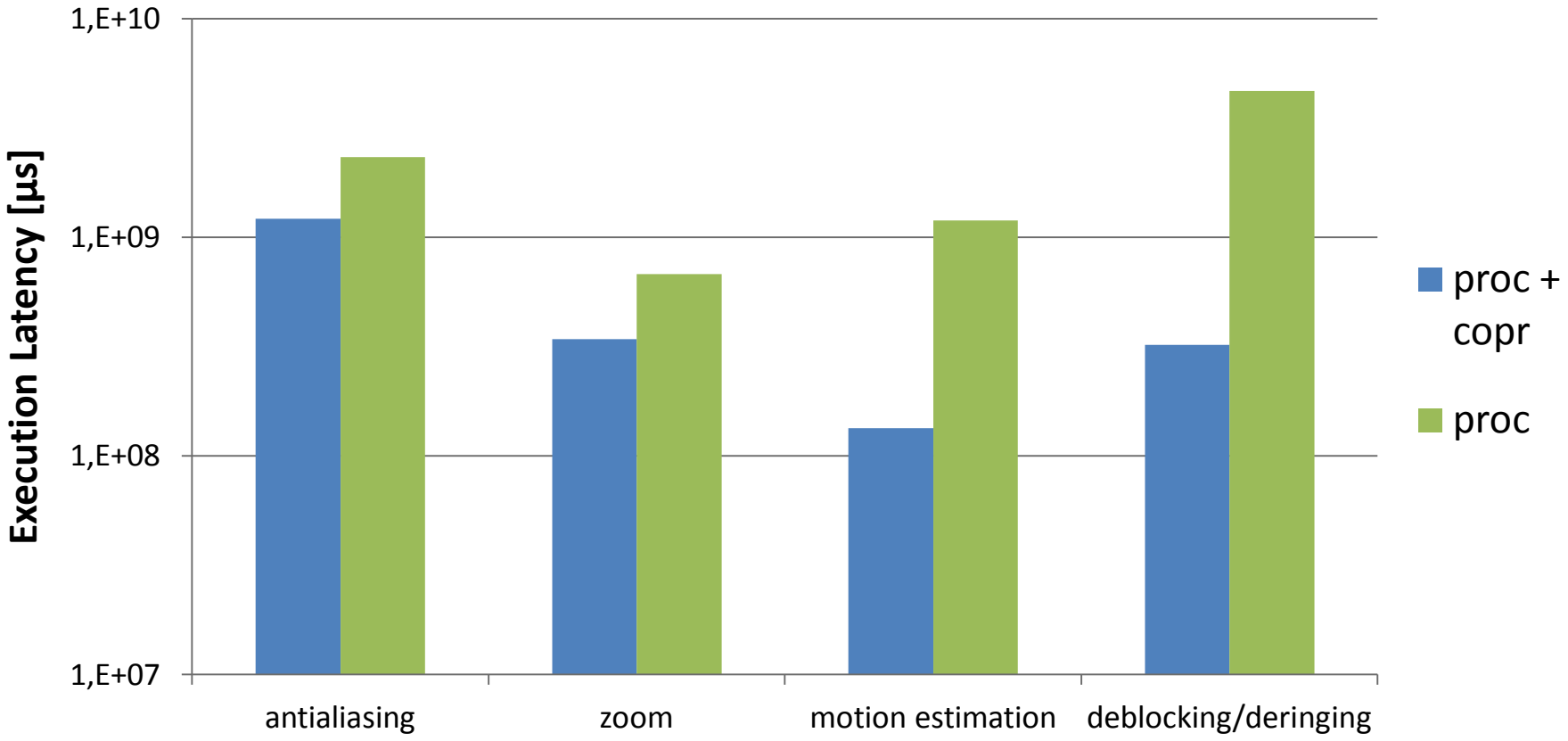
Results have been retrieved running the kernels in the targeted Virtex 5 330 FPGA board at an operating frequency of 125 MHz for the host processor and of 65 MHz (fixed by the CG reconfigurable datapath) for the co-processor.

ACHIEVED RESULTS: PERFOMANCES (1)



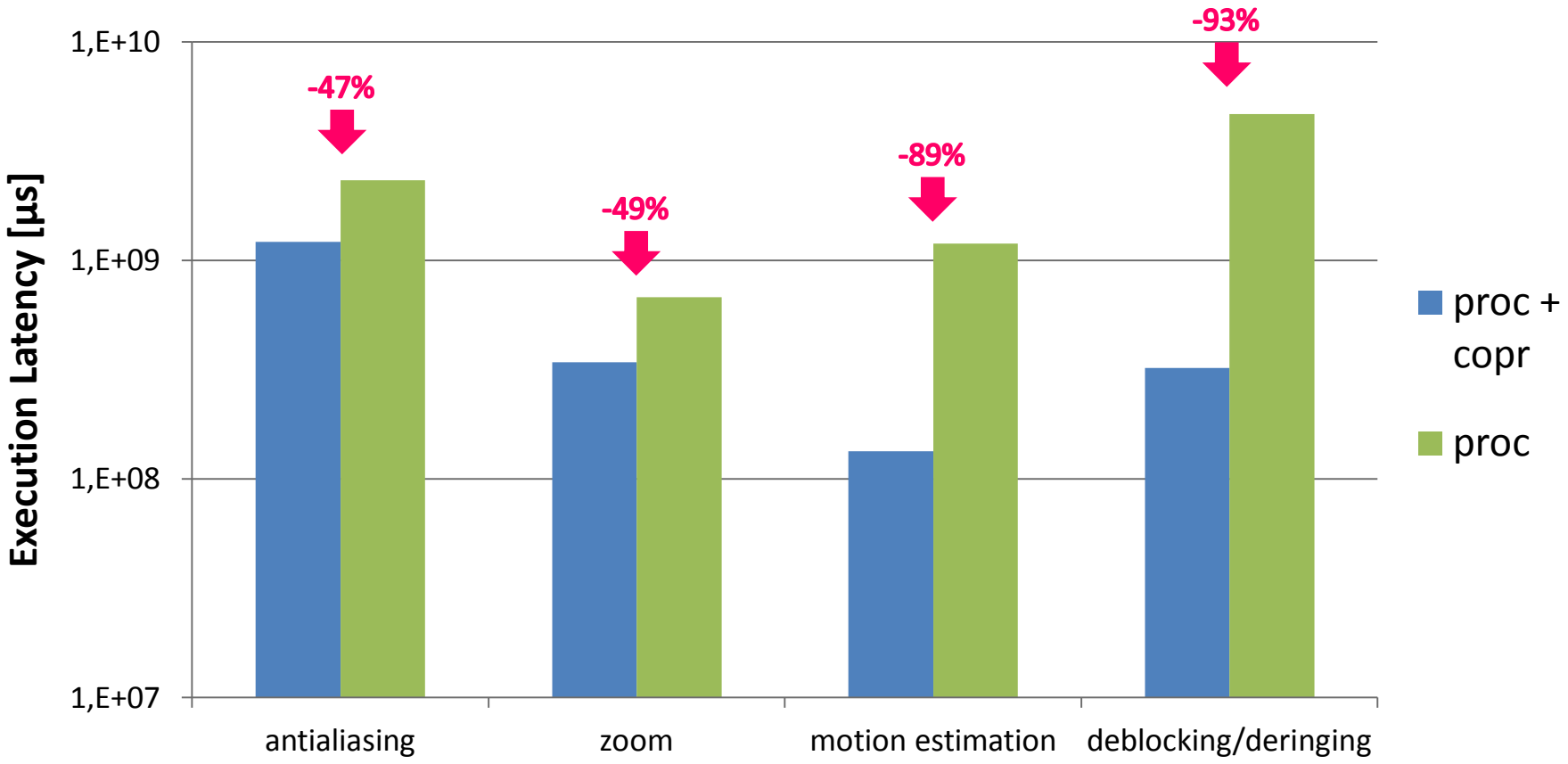
Results have been retrieved running the kernels in the targeted Virtex 5 330 FPGA board at an operating frequency of 125 MHz for the host processor and of 65 MHz (fixed by the CG reconfigurable datapath) for the co-processor.

ACHIEVED RESULTS: PERFOMANCES (2)



Results have been retrieved running the applications in the targeted Virtex 5 330 FPGA board at an operating frequency of 125 MHz for the host processor and of 65 MHz (fixed by the CG reconfigurable datapath) for the co-processor.

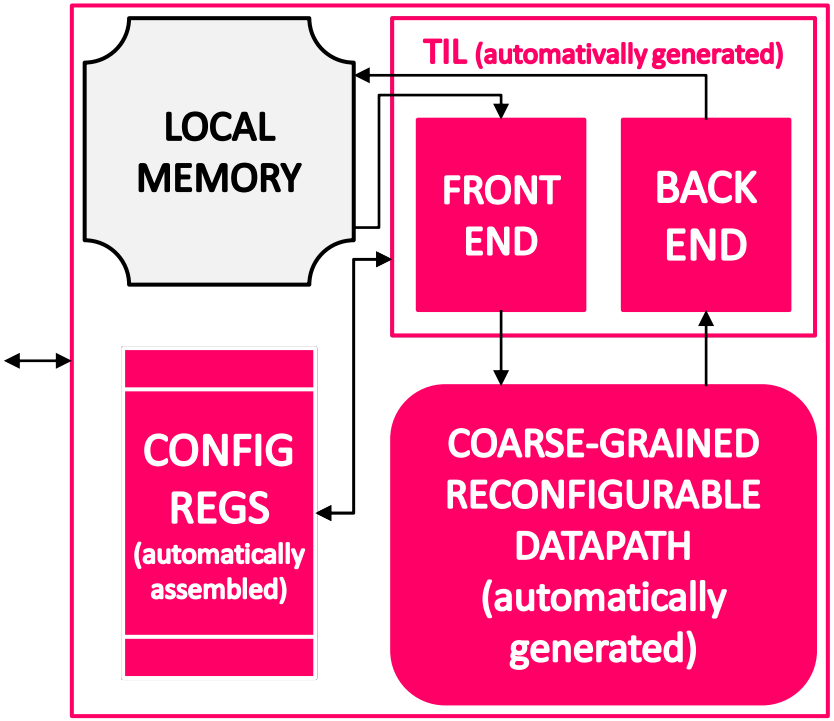
ACHIEVED RESULTS: PERFOMANCES (2)



Results have been retrieved running the applications in the targeted Virtex 5 330 FPGA board at an operating frequency of 125 MHz for the host processor and of 65 MHz (fixed by the CG reconfigurable datapath) for the co-processor.

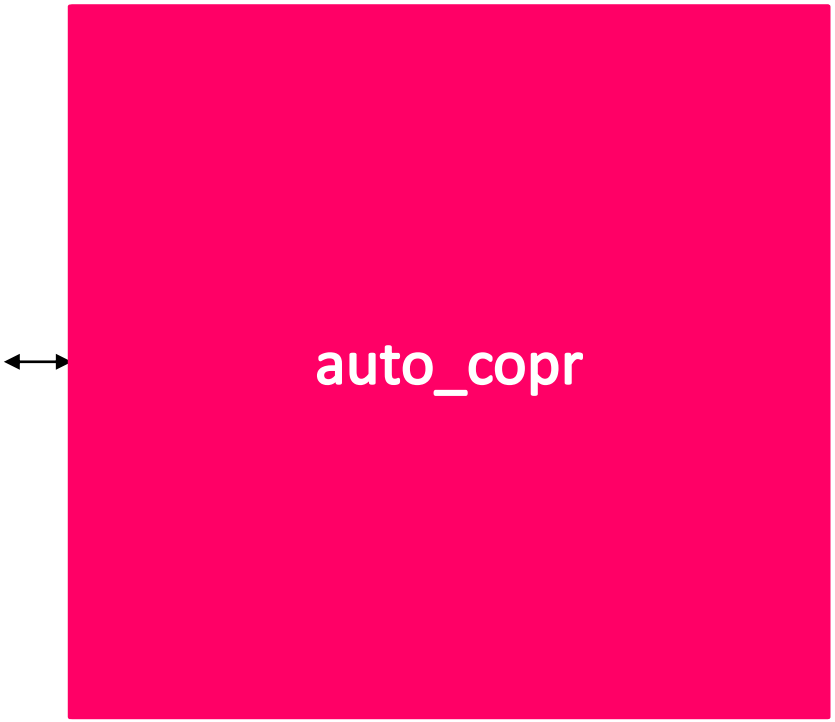
ACHIEVED RESULTS: COMPARISON (1)

READY-TO-USE COARSE-GRAINED RECONFIGURABLECO-PROCESSING UNIT



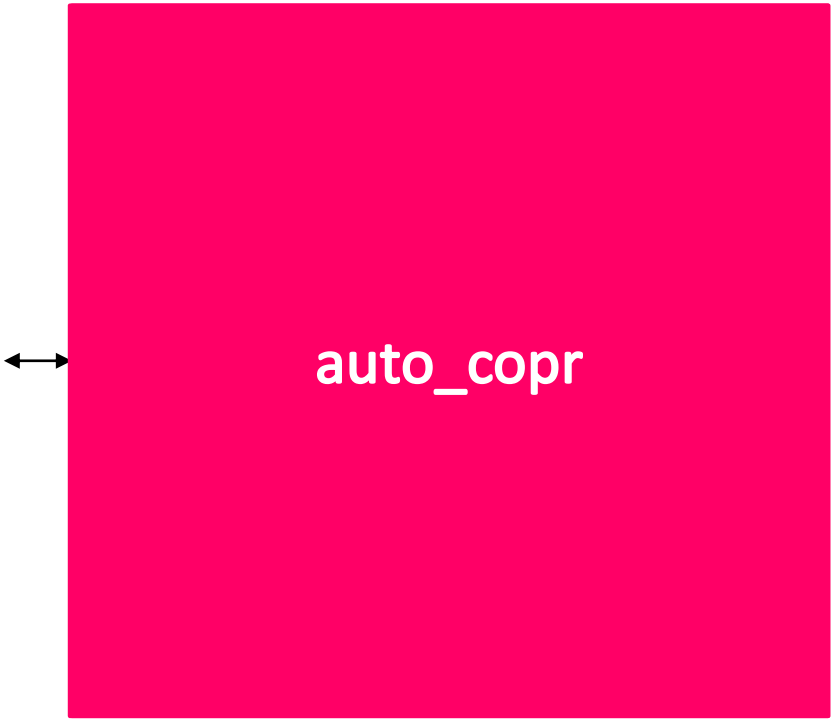
ACHIEVED RESULTS: COMPARISON (1)

READY-TO-USE COARSE-GRAINED RECONFIGURABLECO-PROCESSING UNIT

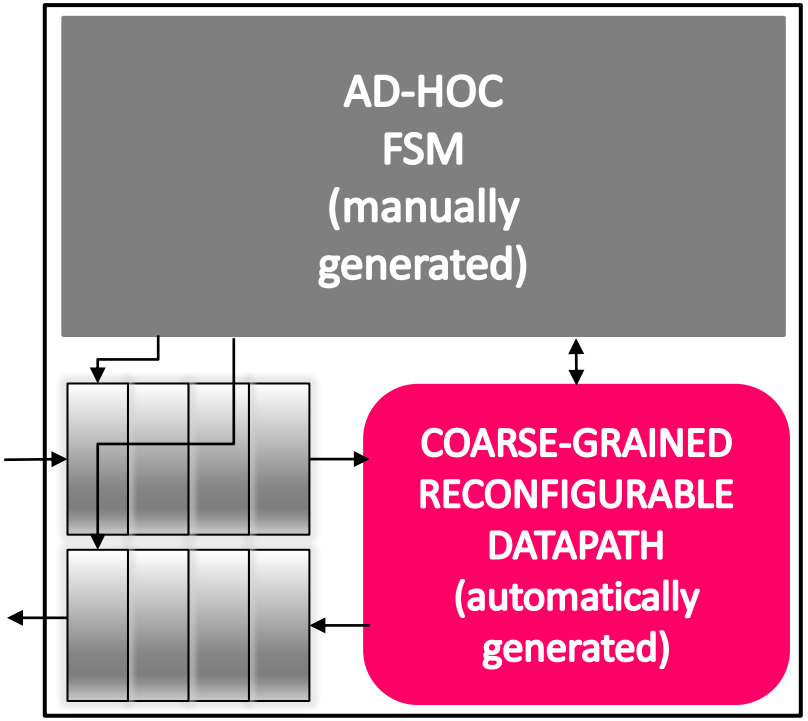


ACHIEVED RESULTS: COMPARISON (1)

READY-TO-USE COARSE-GRAINED RECONFIGURABLECO-PROCESSING UNIT



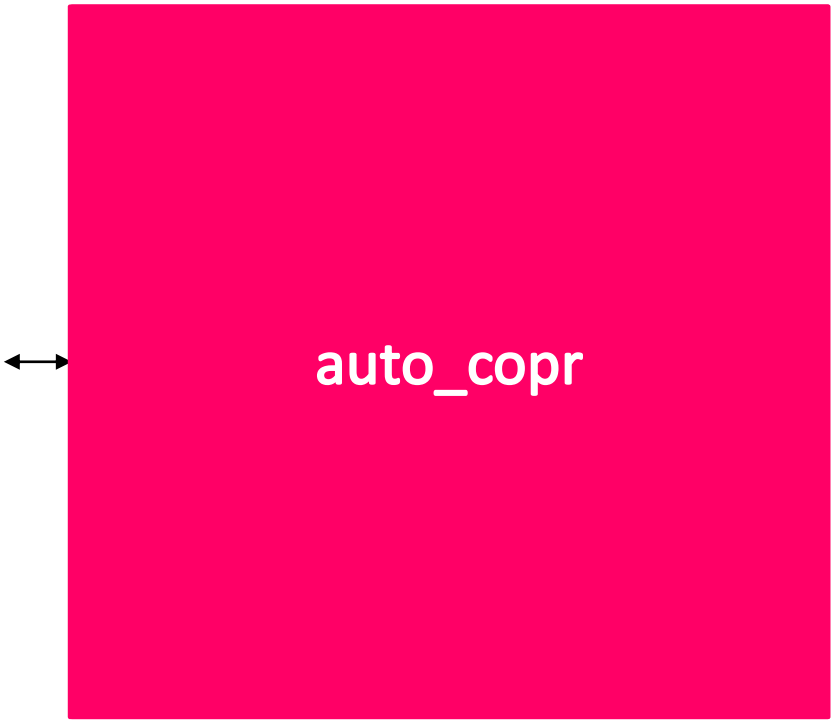
AD-HOC BIG EFFORT DEVELOPMENT REQUIRING CO-PROCESSOR [1]



[1] F. Palumbo, N. Carta, D. Pani, P. Meloni and L. Raffo, *The multi-dataflow composer tool: generation of on-the-fly reconfigurable platforms*, Journal of Real-Time Image Processing, vol. 9, no. 1, pp 233-249, 2012.

ACHIEVED RESULTS: COMPARISON (1)

READY-TO-USE COARSE-GRAINED RECONFIGURABLECO-PROCESSING UNIT



AD-HOC BIG EFFORT DEVELOPMENT REQUIRING CO-PROCESSOR [1]

[1] F. Palumbo, N. Carta, D. Pani, P. Meloni and L. Raffo, *The multi-dataflow composer tool: generation of on-the-fly reconfigurable platforms*, Journal of Real-Time Image Processing, vol. 9, no. 1, pp 233-249, 2012.

ACHIEVED RESULTS: COMPARISON (2)

	resource (% on available)				
	Slice Regs (207360)	Slice LUTs (207360)	BUFGs (32)	BRAMs (288)	frequency [MHz]
custom_copr	352 (0.2)	1041 (0.5)	1 (3.1)	8 (2.8)	155.1
auto_copr	163 (0.1)	372 (0.2)	1 (3.1)	4 (1.4)	226.7
auto vs. custom	-53.7	-68.6	0.0	-50.0	+46.2

Results have been retrieved through the Xilinx Synthesis Technology tool targeting a Virtex 5 330 FPGA board. Only the co-processing unit without any coarse-grained reconfigurable datapath has been considered.

ACHIEVED RESULTS: COMPARISON (2)

RESOURCE -50%
FREQ +45%

	resource (% on available)				
	Slice Regs (207360)	Slice LUTs (207360)	BUFGs (32)	BRAMs (288)	frequency [MHz]
custom_copr	352 (0.2)	1041 (0.5)	1 (3.1)	8 (2.8)	155.1
auto_copr	163 (0.1)	372 (0.2)	1 (3.1)	4 (1.4)	226.7
auto vs. custom	-53.7	-68.6	0.0	-50.0	+46.2

Results have been retrieved through the Xilinx Synthesis Technology tool targeting a Virtex 5 330 FPGA board. Only the co-processing unit without any coarse-grained reconfigurable datapath has been considered.

ACHIEVED RESULTS: COMPARISON (2)

RESOURCE -50%
FREQ +45%

	resource (% on available)				
	Slice Regs (207360)	Slice LUTs (207360)	BUFGs (32)	BRAMs (288)	frequency [MHz]
custom_copr	352 (0.2)	1041 (0.5)	1 (3.1)	8 (2.8)	155.1
auto_copr	163 (0.1)	372 (0.2)	1 (3.1)	4 (1.4)	226.7
auto vs. custom	-53.7	-68.6	0.0	-50.0	+46.2

Results have been retrieved through the Xilinx Synthesis Technology tool targeting a Virtex 5 330 FPGA board. Only the co-processing unit without any coarse-grained reconfigurable datapath has been considered.

	custom_copr			auto_copr		
packet size	1	4	16	1	4	16
loading [# cycles]	3	6	18	3	9	33
loading [µs] @maxf	0.19	0.39	1.16	0.13	0.40	1.46
storing [# cycles]	1	-	-	1	-	-
storing [µs] @maxf	0.06	-	-	0.04	-	-

ACHIEVED RESULTS: COMPARISON (2)

RESOURCE -50%
FREQ +45%

	resource (% on available)				
	Slice Regs (207360)	Slice LUTs (207360)	BUFGs (32)	BRAMs (288)	frequency [MHz]
custom_copr	352 (0.2)	1041 (0.5)	1 (3.1)	8 (2.8)	155.1
auto_copr	163 (0.1)	372 (0.2)	1 (3.1)	4 (1.4)	226.7
auto vs. custom	-53.7	-68.6	0.0	-50.0	+46.2

Results have been retrieved through the Xilinx Synthesis Technology tool targeting a Virtex 5 330 FPGA board. Only the co-processing unit without any coarse-grained reconfigurable datapath has been considered.

	custom_copr			auto_copr		
packet size	1	4	16	1	4	16
loading [# cycles]	3	6	18	3	9	33
loading [µs] @maxf	0.19	0.39	1.16	0.13	0.40	1.46
storing [# cycles]	1	-	-	1	-	-
storing [µs] @maxf	0.06	-	-	0.04	-	-

ACHIEVED RESULTS: COMPARISON (2)

RESOURCE -50% FREQ +45%

	resource (% on available)				
	Slice Regs (207360)	Slice LUTs (207360)	BUFGs (32)	BRAMs (288)	frequency [MHz]
custom_copr	352 (0.2)	1041 (0.5)	1 (3.1)	8 (2.8)	155.1
auto_copr	163 (0.1)	372 (0.2)	1 (3.1)	4 (1.4)	226.7
auto vs. custom	-53.7	-68.6	0.0	-50.0	+46.2

Results have been retrieved through the Xilinx Synthesis Technology tool targeting a Virtex 5 330 FPGA board. Only the co-processing unit without any coarse-grained reconfigurable datapath has been considered.

LOADING +21%

	custom_copr			auto_copr		
packet size	1	4	16	1	4	16
loading [# cycles]	3	6	18	3	9	33
loading [µs] @maxf	0.19	0.39	1.16	0.13	0.40	1.46
storing [# cycles]	1	-	-	1	-	-
storing [µs] @maxf	0.06	-	-	0.04	-	-

OUTLINE: FINAL REMARKS

- Introduction:
  - Problem statement
  - Background
  - Goals
- Co-processing units generation:
  - Approach and baseline Multi-Dataflow Composer
  - Template Interface Layer: hardware and automatic composition
- Performance assessment
  - Use-case scenario
  - Results
- Final remarks and future directions

FINAL REMARKS

- Automatic generation tools are needed to cut down time to market of CG reconfigurable co-processors
- MDC has been developed within the RVC domain to:
  - Implement coarse-grained reconfigurable datapaths but lacks of an interface able to exploit the datapath as co-processing unit in a complete system
- In this work we have:
  - defined an adaptable interface for the MDC generated datapaths
  - integrated the generation and adaptation of this interface in the MDC framework

FINAL REMARKS

- Automatic generation tools are needed to cut down time to market of CG reconfigurable co-processors
- MDC has been developed within the RVC domain to:
  - Implement coarse-grained reconfigurable datapaths but lacks of an interface able to exploit the datapath as co-processing unit in a complete system
- In this work we have:
  - defined an adaptable interface for the MDC generated datapaths
  - integrated the generation and adaptation of this interface in the MDC framework
- Future developments:
  - Optimize loading
  - Support to other kinds of communication schemes
  - Automatic APIs library

Conference on Design & Architectures for Signal & Image Processing

October 8-10, 2014

Madrid, Spain

dasip

ecsi

THANK YOU

Automatic Generation of Dataflow-Based Reconfigurable Co-processing Units

Francesca Palumbo
University of Sassari
PolComIng Dept. – Information Eng. Unit