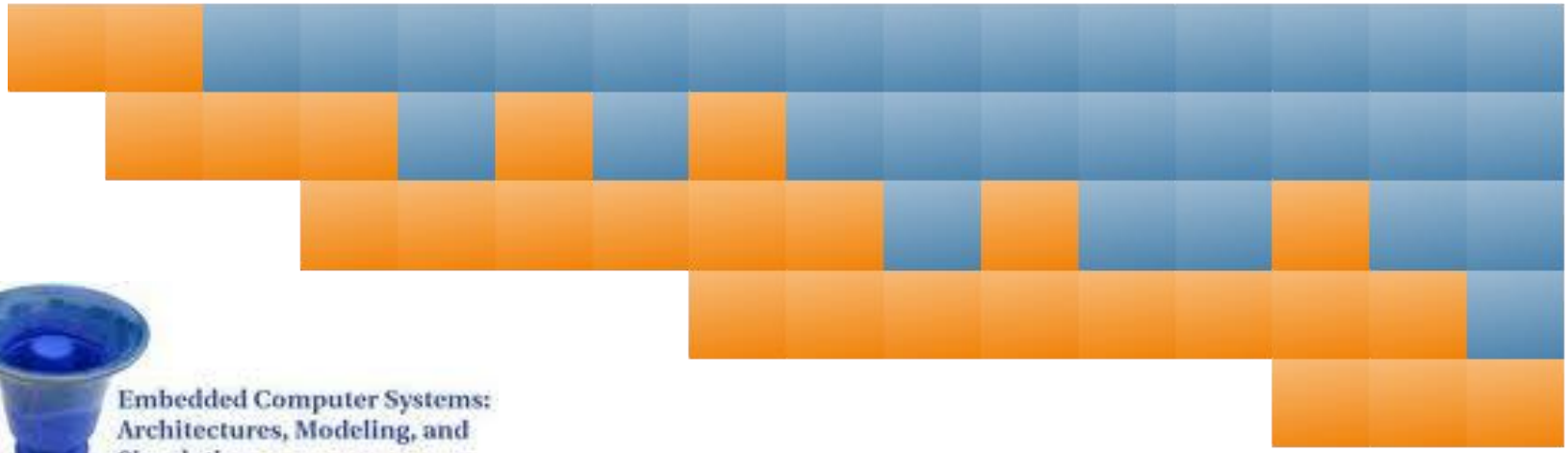
XIV International Conference on Embedded Computer and Systems: Architectures, MOdeling and Simulation
SAMOS XIV - 2014 July 14th - Samos Island (Greece)

Carlo Sau, Luigi Raffo
DIEE
Università degli Studi di Cagliari

Francesca Palumbo
POLCOMING
Università degli Studi di Sassari

Endri Bezati, Simone Casale-Brunet, Marco Mattavelli
SCI STI MM
École Politecnique Fédérale de Lausanne

Automated Design Flow for Coarse-Grained Reconfigurable Platforms: an RVC-CAL Multi-Standard Decoder Use-Case

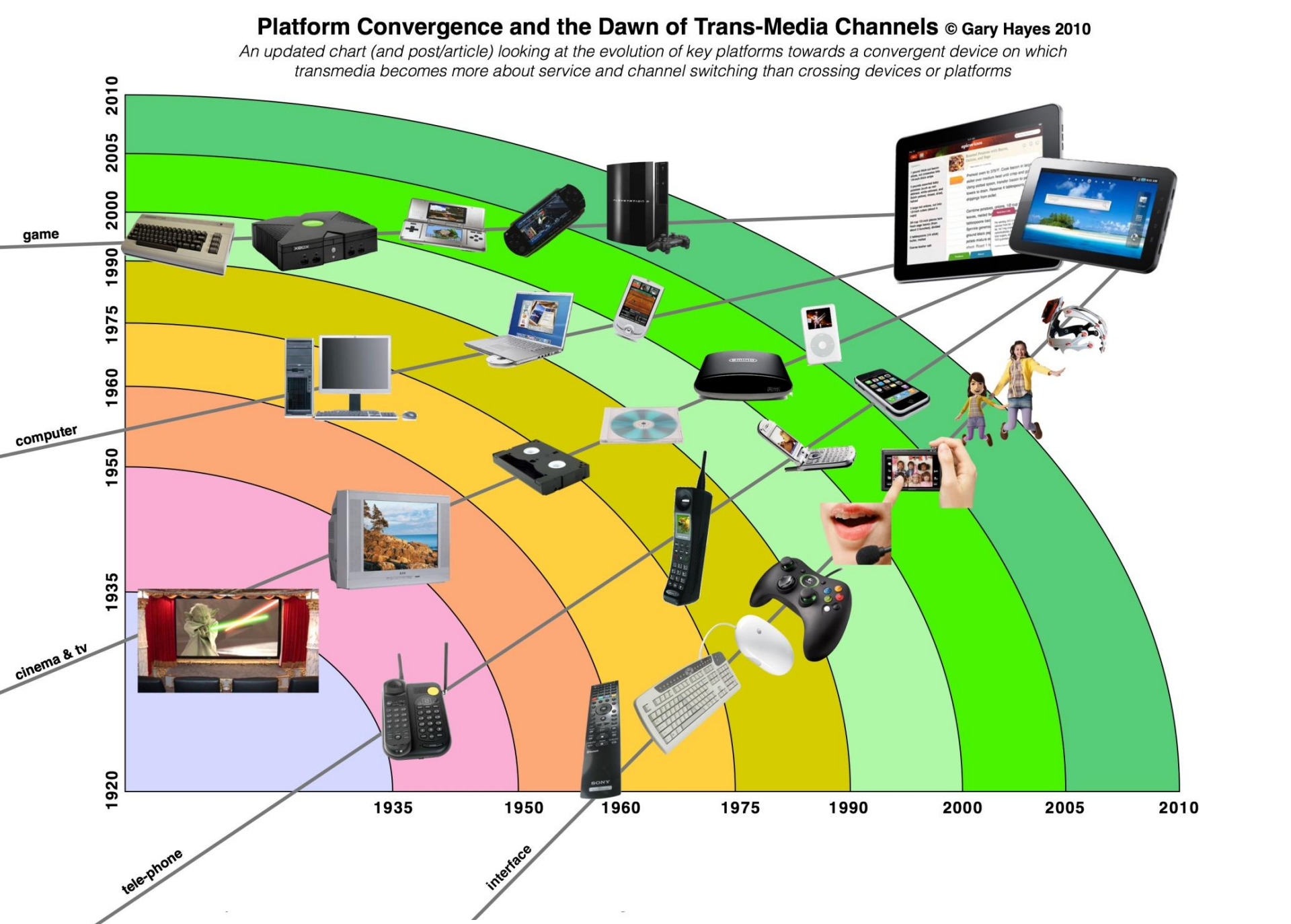


Outline

- Introduction
  - Problem Statement
  - Two Step Problem Solving
    - Step 1: Coarse-Grained Reconfigurability
    - Step 2: Dataflow Model of Computation and RVC-CAL
  - Generation of Multi-Dataflow Graphs
- Automated Design Flow
  - Composition: the Multi-Dataflow Composer
  - Optimization: TURNUS Co-Exploration Framework
  - RTL Generation: Xronos High Level Synthesis
  - Tools Integration
- Experimental Results
  - An MPEG-4 SP Decoder Use Case
  - Synthesis Results
  - Performance Results
- Conclusions

Outline

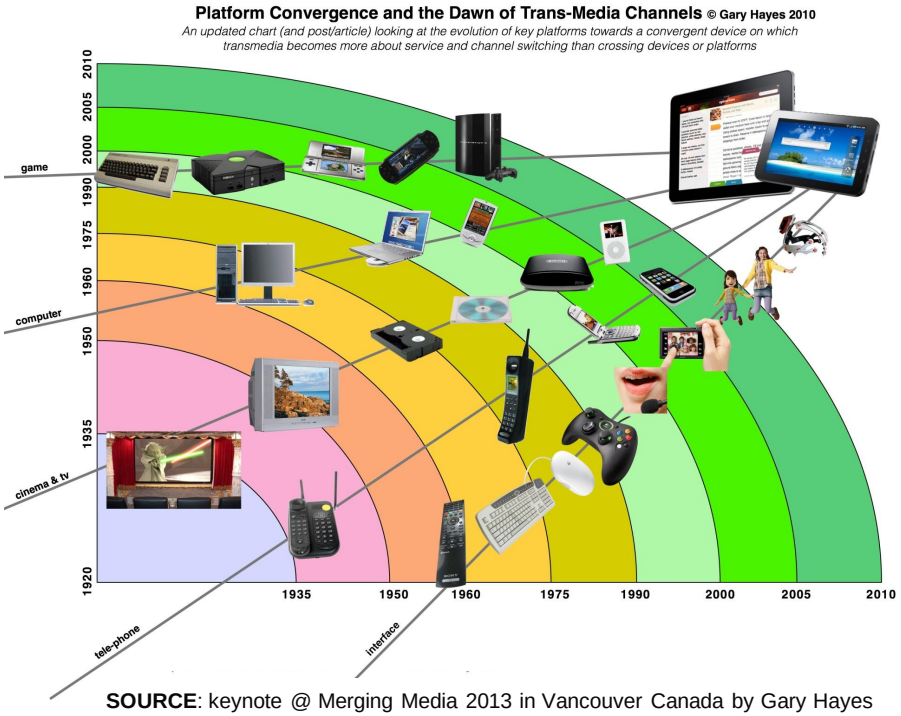
- Introduction
  - Problem Statement
  - Two Step Problem Solving
    - Step 1: Coarse-Grained Reconfigurability
    - Step 2: Dataflow Model of Computation and RVC-CAL
  - Generation of Multi-Dataflow Graphs
- Automated Design Flow
  - Composition: the Multi-Dataflow Composer
  - Optimization: TURNUS Co-Exploration Framework
  - RTL Generation: Xronos High Level Synthesis
  - Tools Integration
- Experimental Results
  - An MPEG-4 SP Decoder Use Case
  - Synthesis Results
  - Performance Results
- Conclusions



SOURCE: keynote @ Merging Media 2013 in Vancouver Canada by Gary Hayes

Problem Statement

Electronic devices are converging to platforms that are:

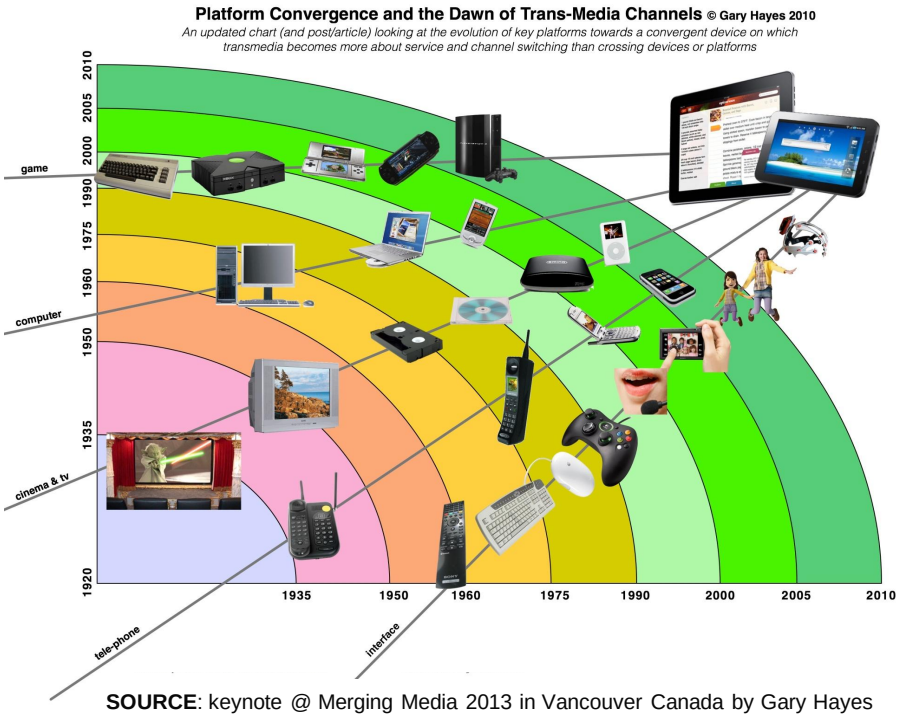


SOURCE: keynote @ Merging Media 2013 in Vancouver Canada by Gary Hayes

- **portable:** dimensions and battery life limits have to be taken into consideration.
- **multimedial:** different applications have to be executed, very often at the same time.
- **higly efficient:** real time response is needed in many cases.
- **easily evolvable:** technology and algorithms fast evolution have to be followed by the devices.

Problem Statement

Electronic devices are converging to platforms that are:



SOURCE: keynote @ Merging Media 2013 in Vancouver Canada by Gary Hayes

- **portable:** dimensions and battery life limits have to be taken into consideration.
- **multimedial:** different applications have to be executed, very often at the same time.
- **higly efficient:** real time response is needed in many cases.
- **easily evolvable:** technology and algorithms fast evolution have to be followed by the devices.

need of **new design flows** to address this complex scenario

Two Step Problem Solving

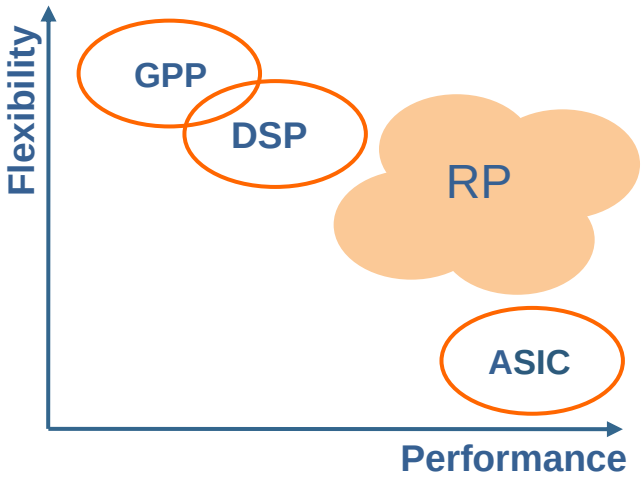
Step 1: Coarse-Grained Reconfigurability

- Systems are required to be **flexible** and **efficient**.

Two Step Problem Solving

Step 1: Coarse-Grained Reconfigurability

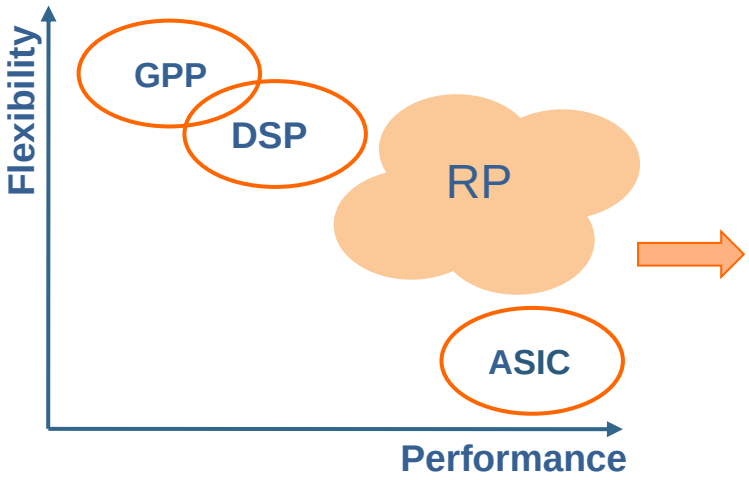
- Systems are required to be **flexible** and **efficient**.
- **Reconfigurable Paradigm** (RP) to hardware design: specialized computing platforms, capable of changing configuration to serve the targeted computations.



Two Step Problem Solving

Step 1: Coarse-Grained Reconfigurability

- Systems are required to be **flexible** and **efficient**.
- **Reconfigurable Paradigm** (RP) to hardware design: specialized computing platforms, capable of changing configuration to serve the targeted computations.
- The more the hardware is specialized, the more it is **difficult to program**.

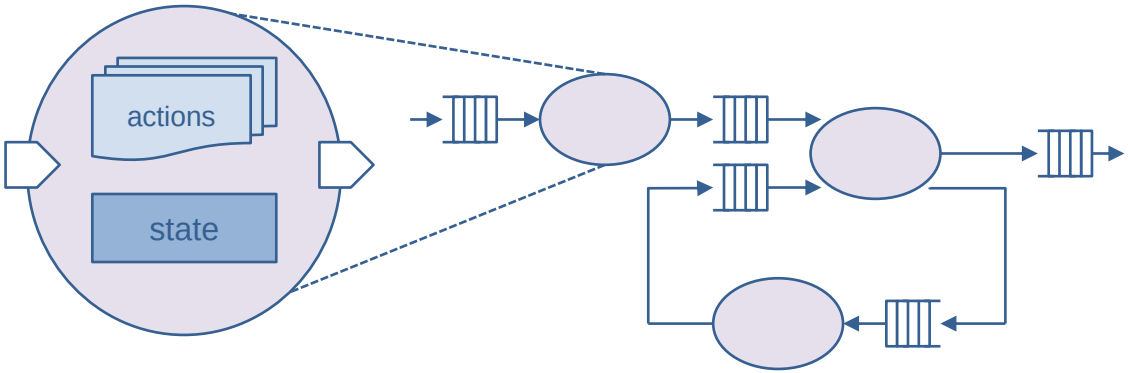


	FINE-GRAINED (FG)	COARSE-GRAINED (CG)
	Bit-level	Word-level
Flexibility	☺	😐
Reconf. Speed	😐	☺
Config. Storage	☹	😐

Two Step Problem Solving

Step 2: Dataflow Model of Computation and RVC-CAL (1)

A **dataflow program** is a directed graph of functional units (**actors**) exchanging sequences of data (**tokens**) through dedicated channels.

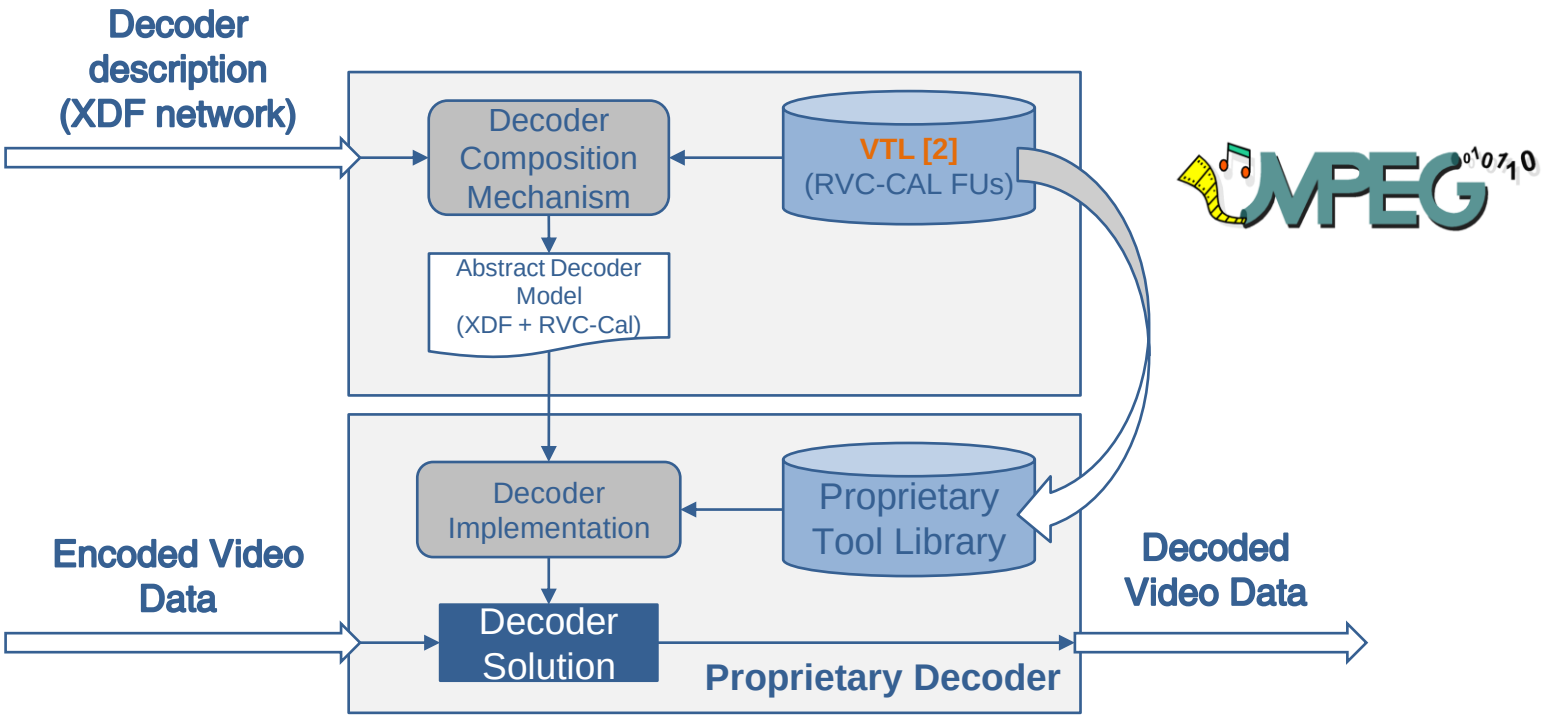


Actors encapsulate their state and communicate exclusively by sending and receiving tokens. Such an absence of race conditions, along with the intrinsic **modularity** of the dataflow graphs, make it possible to explicit the algorithmic **parallelism** of the programs.

Two Step Problem Solving

Step 2: Dataflow Model of Computation and RVC-CAL (2)

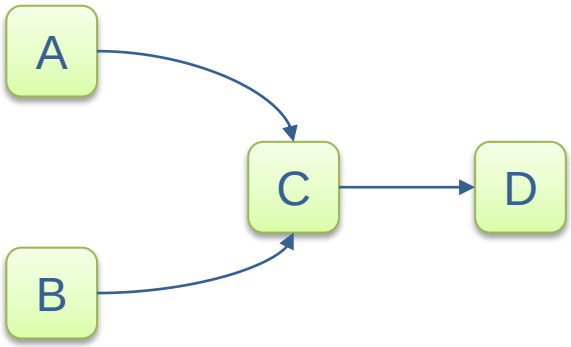
CAL[1] = Cal Actor Language, high-level programming language for describing dataflow actors.



[1] MPEG-B part 4 (2009), **[2]** MPEG-C part 4 (2010)

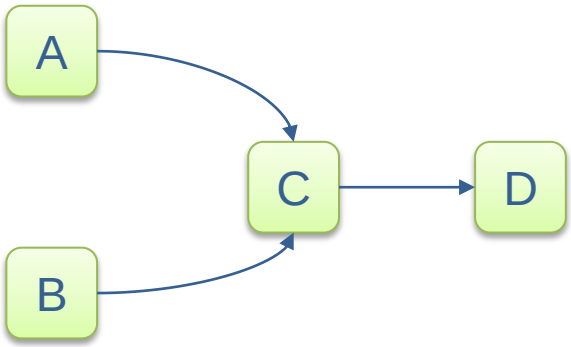
Generation of Multi-Dataflow Graphs (1)

RVC-CAL dataflow specifications



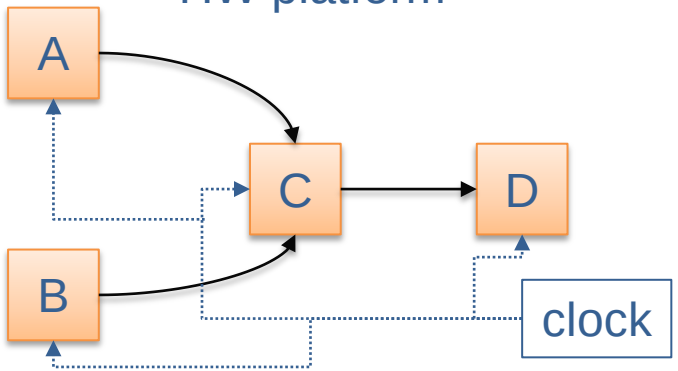
Generation of Multi-Dataflow Graphs (1)

RVC-CAL dataflow specifications

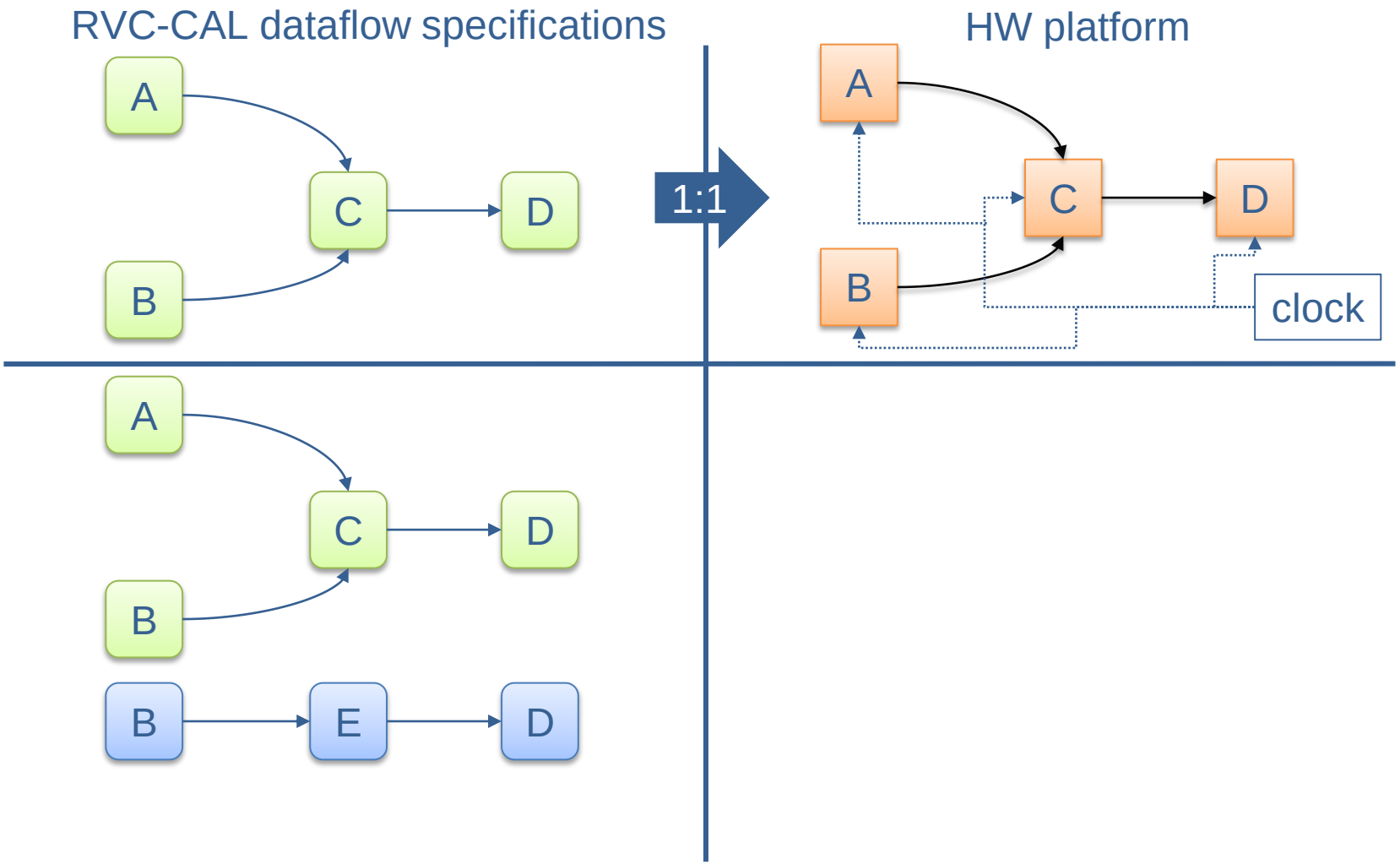


1:1

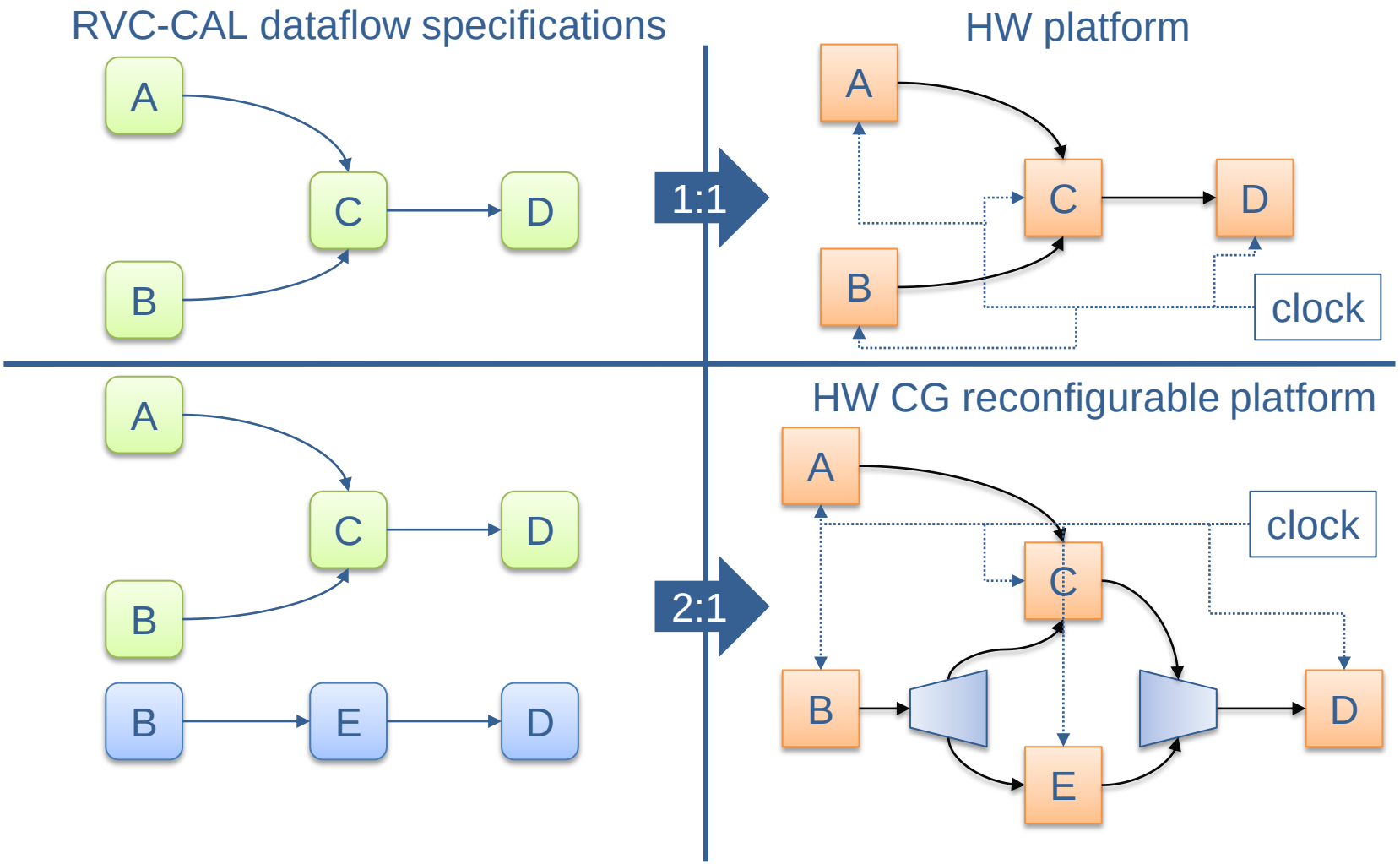
HW platform



Generation of Multi-Dataflow Graphs (1)



Generation of Multi-Dataflow Graphs (1)



Generation of Multi-Dataflow Graphs (2)

Challenges of the modern electronic devices design:

- low area and power (**portability**)
- flexibility (**multimediality**)
- performances (**high efficiency**)
- code reusability (**fast evolution**)

Generation of Multi-Dataflow Graphs (2)

Challenges of the modern electronic devices design:

- low area and power (**portability**) → resource sharing
- flexibility (**multimediality**) → CG reconfiguration
- performances (**high efficiency**) → dataflow parallelism highlighting
- code reusability (**fast evolution**) → dataflow modularity (VTL)

Generation of Multi-Dataflow Graphs (2)

Challenges of the modern electronic devices design:

- low area and power (**portability**) → resource sharing
- flexibility (**multimediality**) → CG reconfiguration
- performances (**high efficiency**) → dataflow parallelism highlighting
- code reusability (**fast evolution**) → dataflow modularity (VTL)

perfect matching, but…

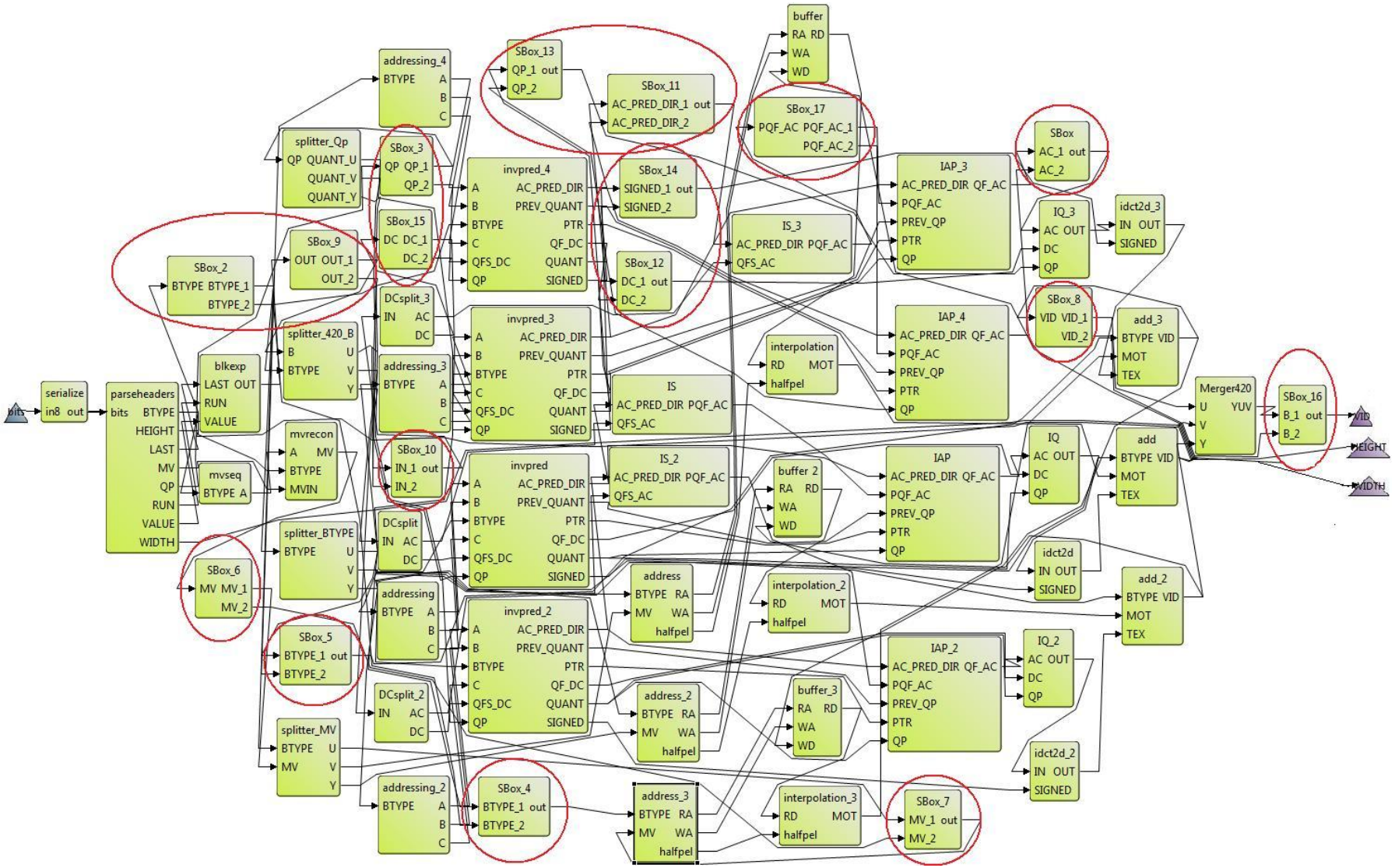
Generation of Multi-Dataflow Graphs (2)

Challenges of the modern electronic devices design:

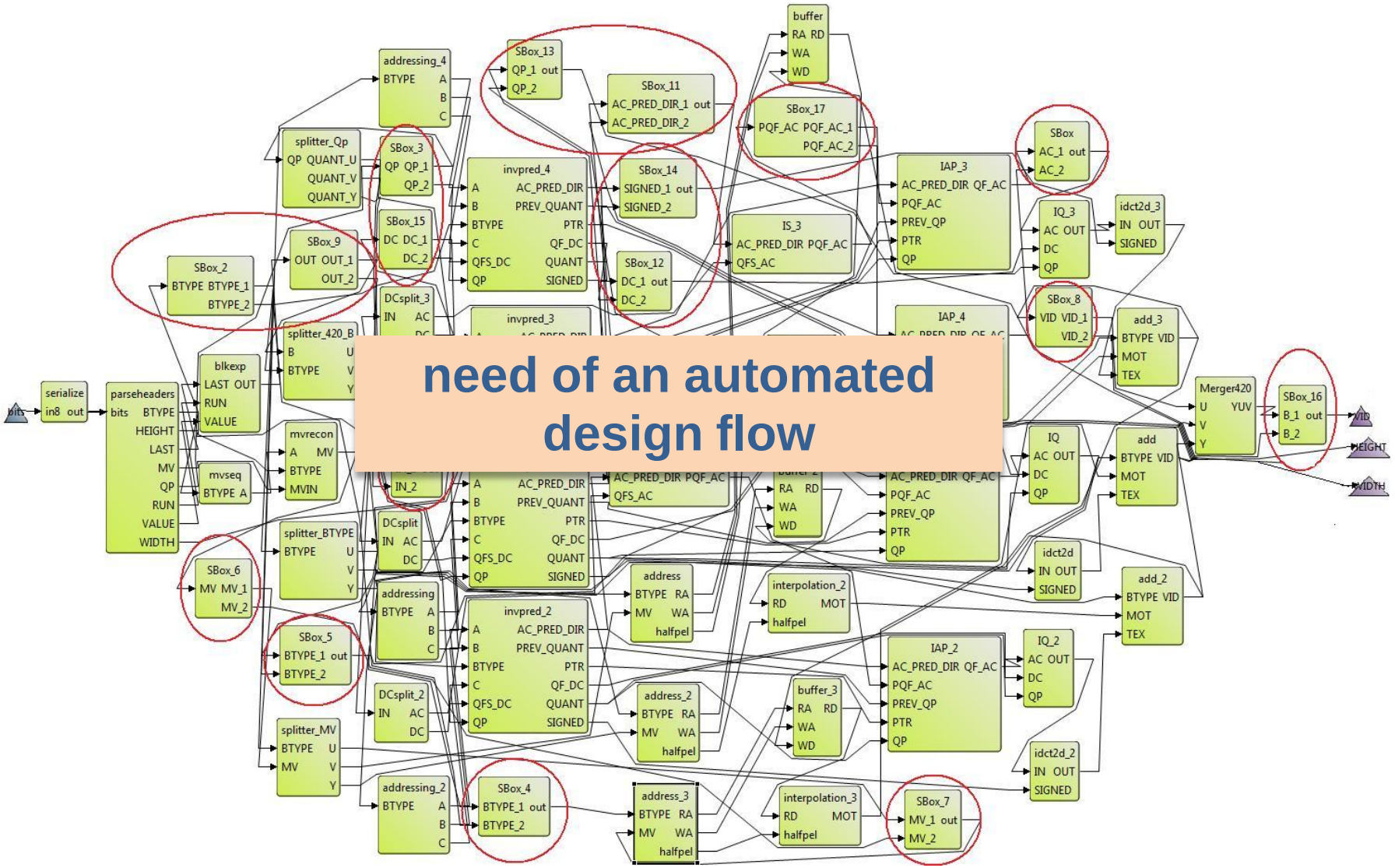
- low area and power (**portability**) → resource sharing
- flexibility (**multimediality**) → CG reconfiguration
- performances (**high efficiency**) → dataflow parallelism highlighting
- code reusability (**fast evolution**) → dataflow modularity (VTL)

perfect matching, but…what happens if the **design complexity** grows?

Generation of Multi-Dataflow Graphs (2)



Generation of Multi-Dataflow Graphs (2)



Outline

- Introduction
  - Problem Statement
  - Two Step Problem Solving
    - Step 1: Coarse-Grained Reconfigurability
    - Step 2: Dataflow Model of Computation and RVC-CAL
  - Generation of Multi-Dataflow Graphs
- Automated Design Flow
  - Composition: the Multi-Dataflow Composer
  - Optimization: TURNUS Co-Exploration Framework
  - RTL Generation: Xronos High Level Synthesis
  - Tools Integration
- Experimental Results
  - An MPEG-4 SP Decoder Use Case
  - Synthesis Results
  - Performance Results
- Conclusions

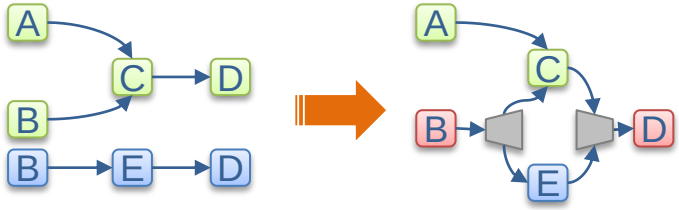
Automated Design Flow

Functionalities required by the automated design flow:

Automated Design Flow

Functionalities required by the automated design flow:

multi-dataflow network **composition**

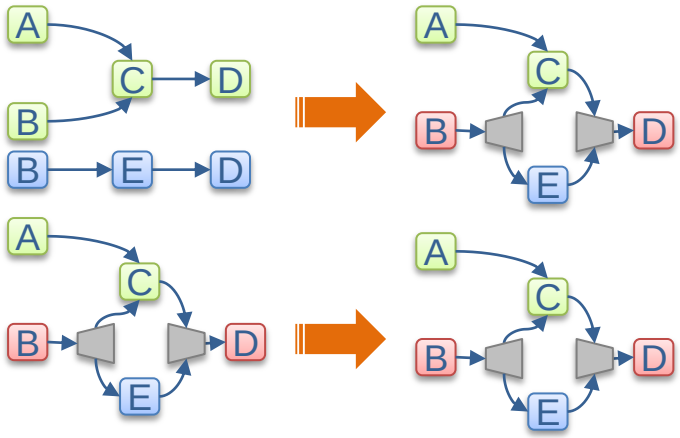


Automated Design Flow

Functionalities required by the automated design flow:

multi-dataflow network **composition**

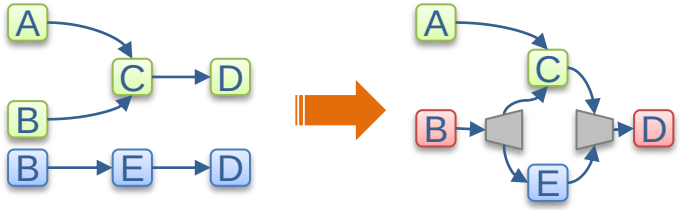
dataflow network buffers **optimization**



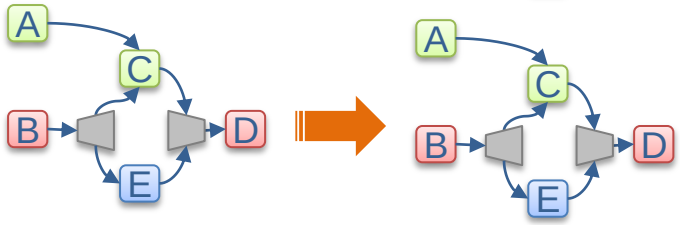
Automated Design Flow

Functionalities required by the automated design flow:

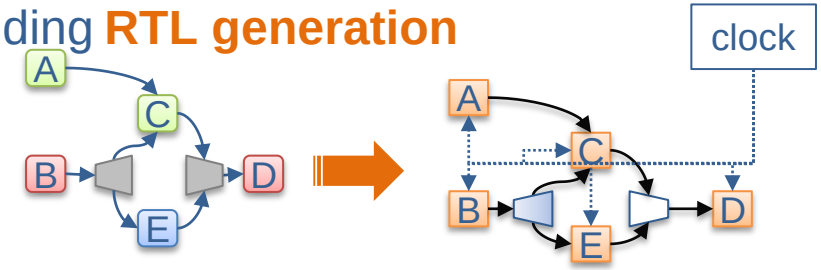
multi-dataflow network **composition**



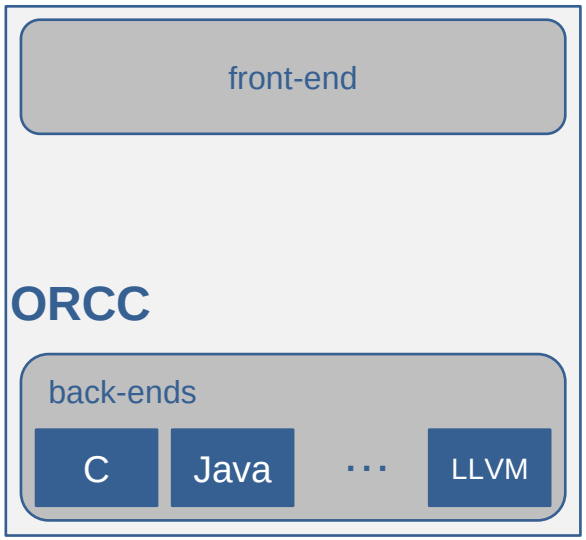
dataflow network buffers **optimization**



multi-dataflow corresponding **RTL generation**

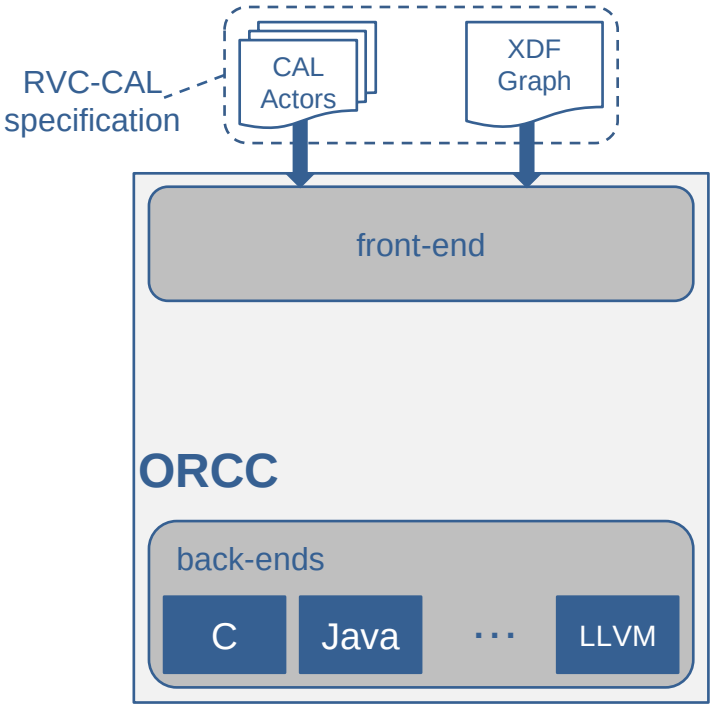


Composition: the Multi-Dataflow Composer (1)



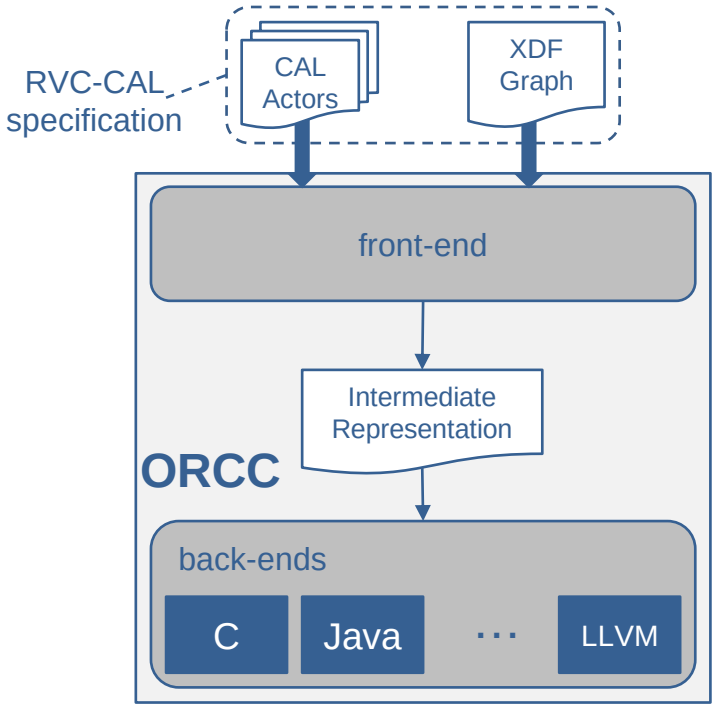
Open RVC-CAL Compiler (ORCC) is a framework able to generate, from an RVC-CAL specification, the corresponding source code for different target platforms (hardware, software or mixed).

Composition: the Multi-Dataflow Composer (1)



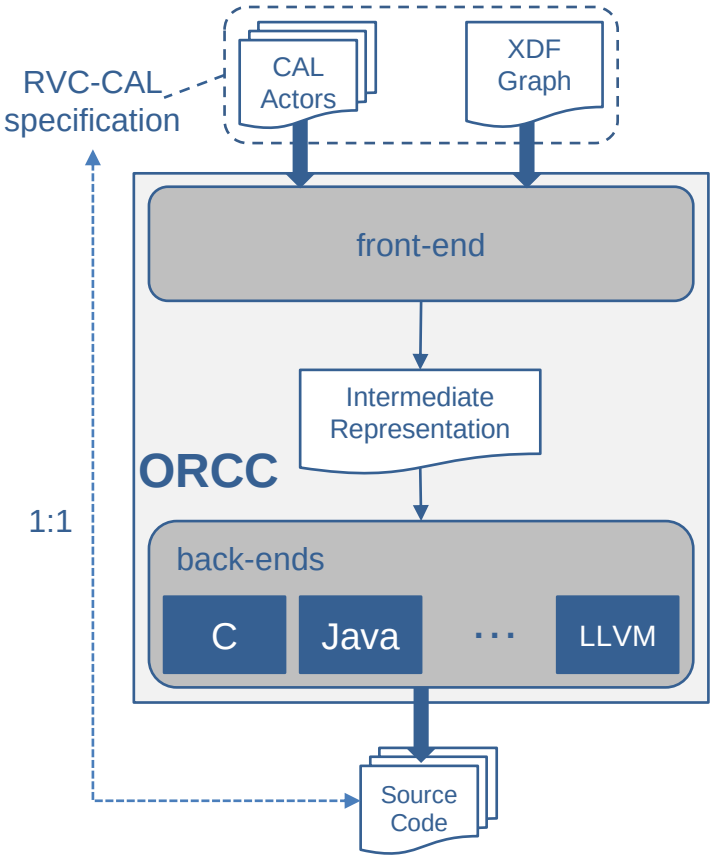
Open RVC-CAL Compiler (ORCC) is a framework able to generate, from an RVC-CAL specification, the corresponding source code for different target platforms (hardware, software or mixed).

Composition: the Multi-Dataflow Composer (1)



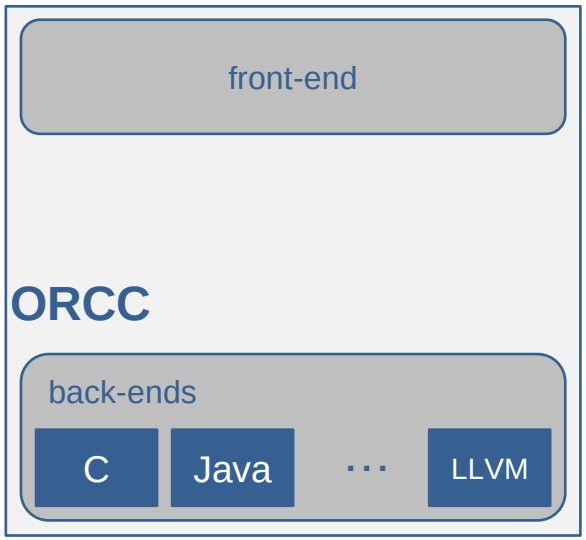
Open RVC-CAL Compiler (ORCC) is a framework able to generate, from an RVC-CAL specification, the corresponding source code for different target platforms (hardware, software or mixed).

Composition: the Multi-Dataflow Composer (1)

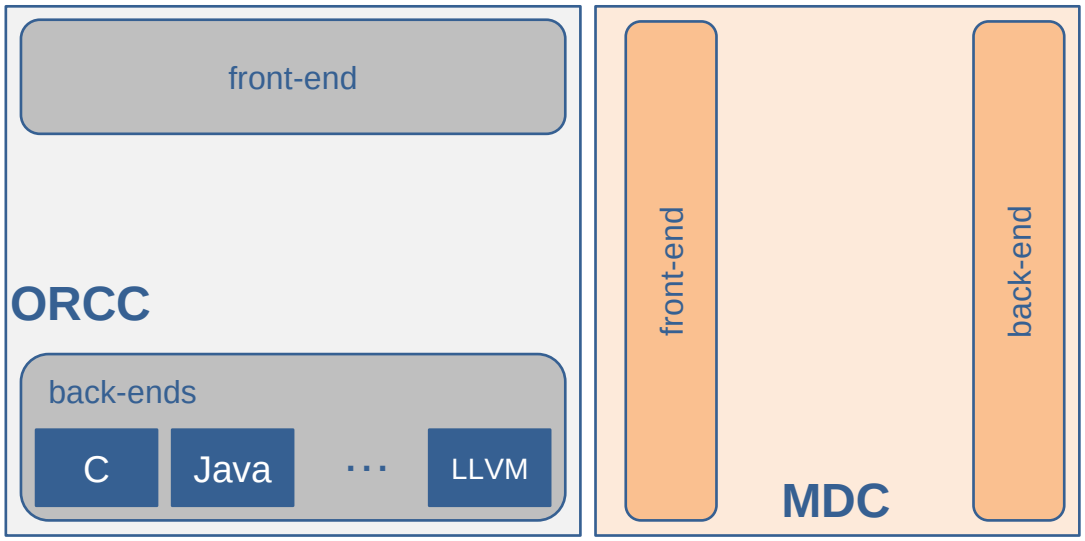


Open RVC-CAL Compiler (ORCC) is a framework able to generate, from an RVC-CAL specification, the corresponding source code for different target platforms (hardware, software or mixed).

Composition: the Multi-Dataflow Composer (1)

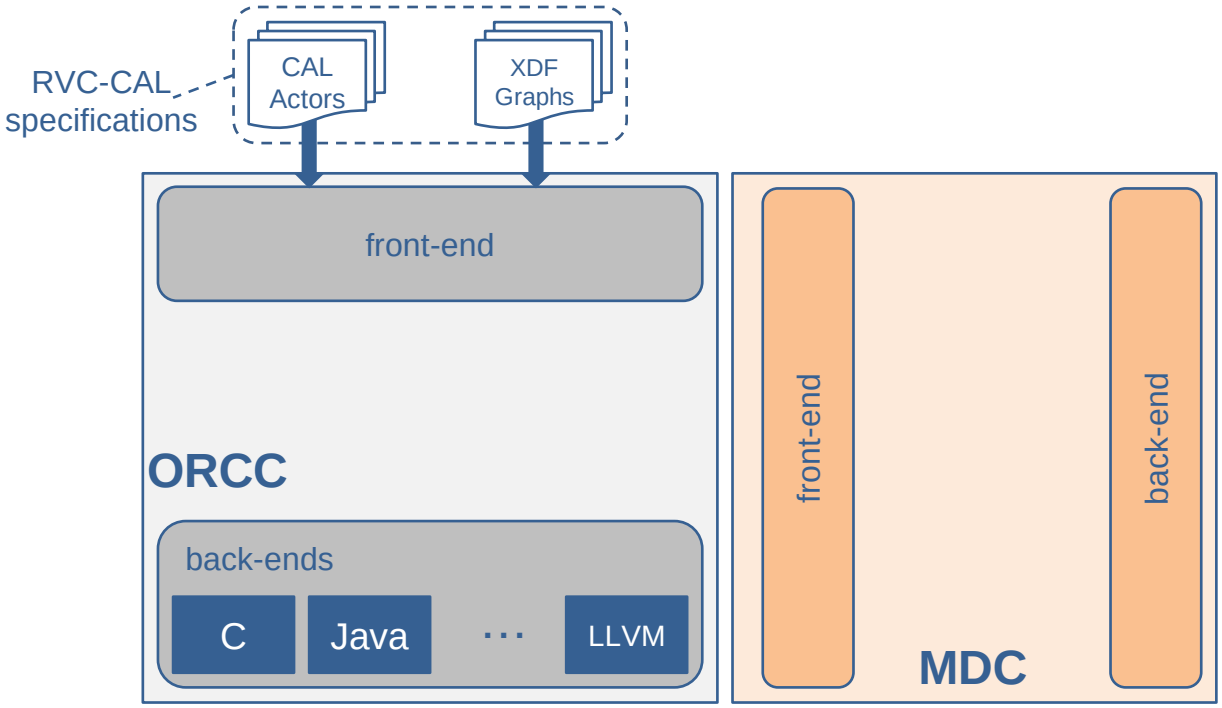


Composition: the Multi-Dataflow Composer (1)



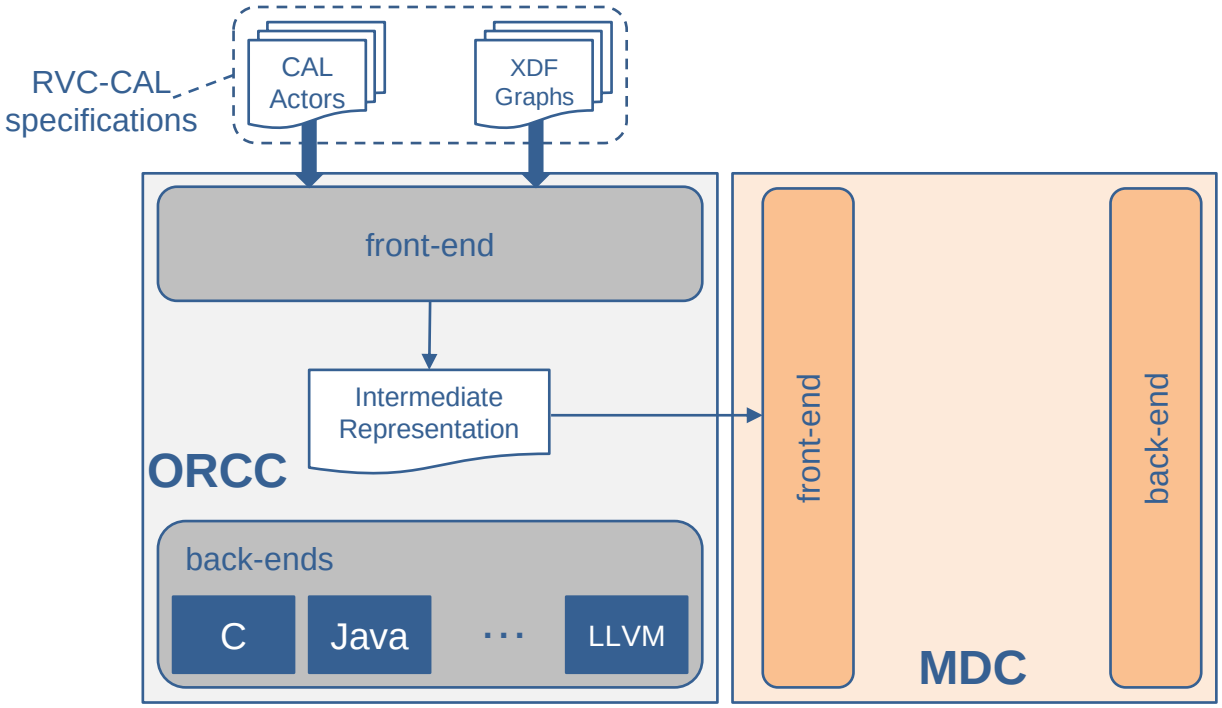
MDC exploits the ORCC front-end to generate, from a set of RVC-CAL specifications, the hardware description of a reconfigurable platform.

Composition: the Multi-Dataflow Composer (1)



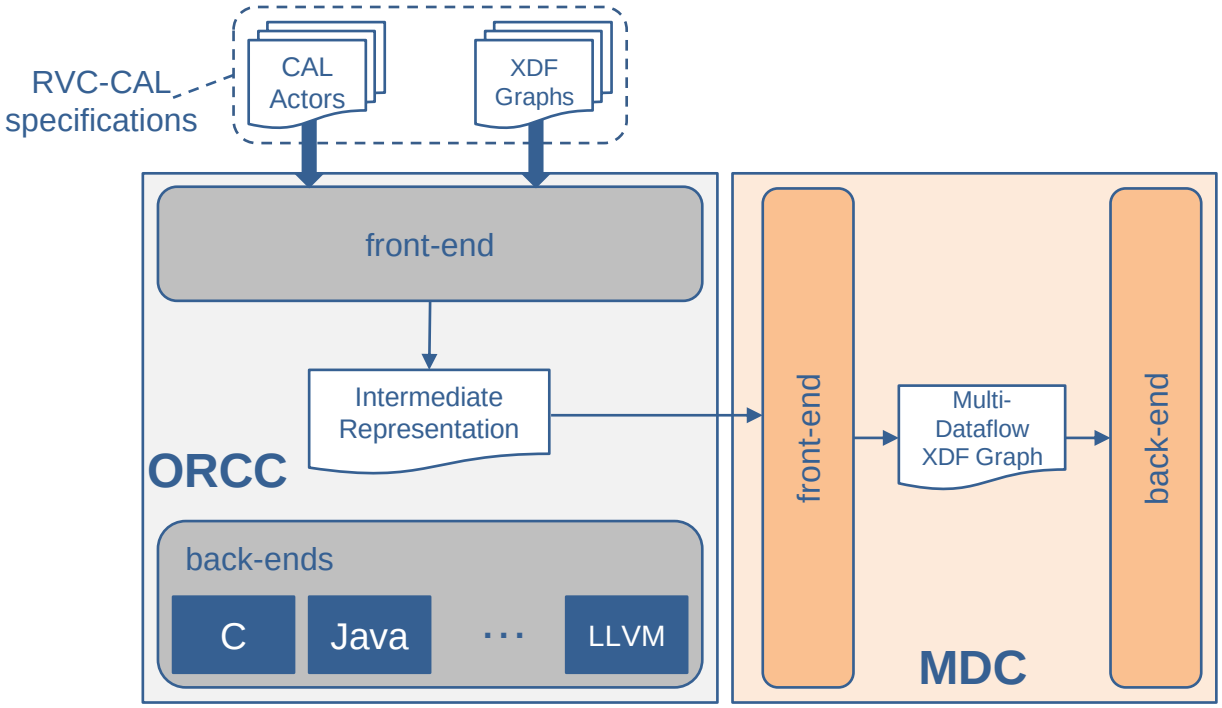
MDC exploits the ORCC front-end to generate, from a set of RVC-CAL specifications, the hardware description of a reconfigurable platform.

Composition: the Multi-Dataflow Composer (1)



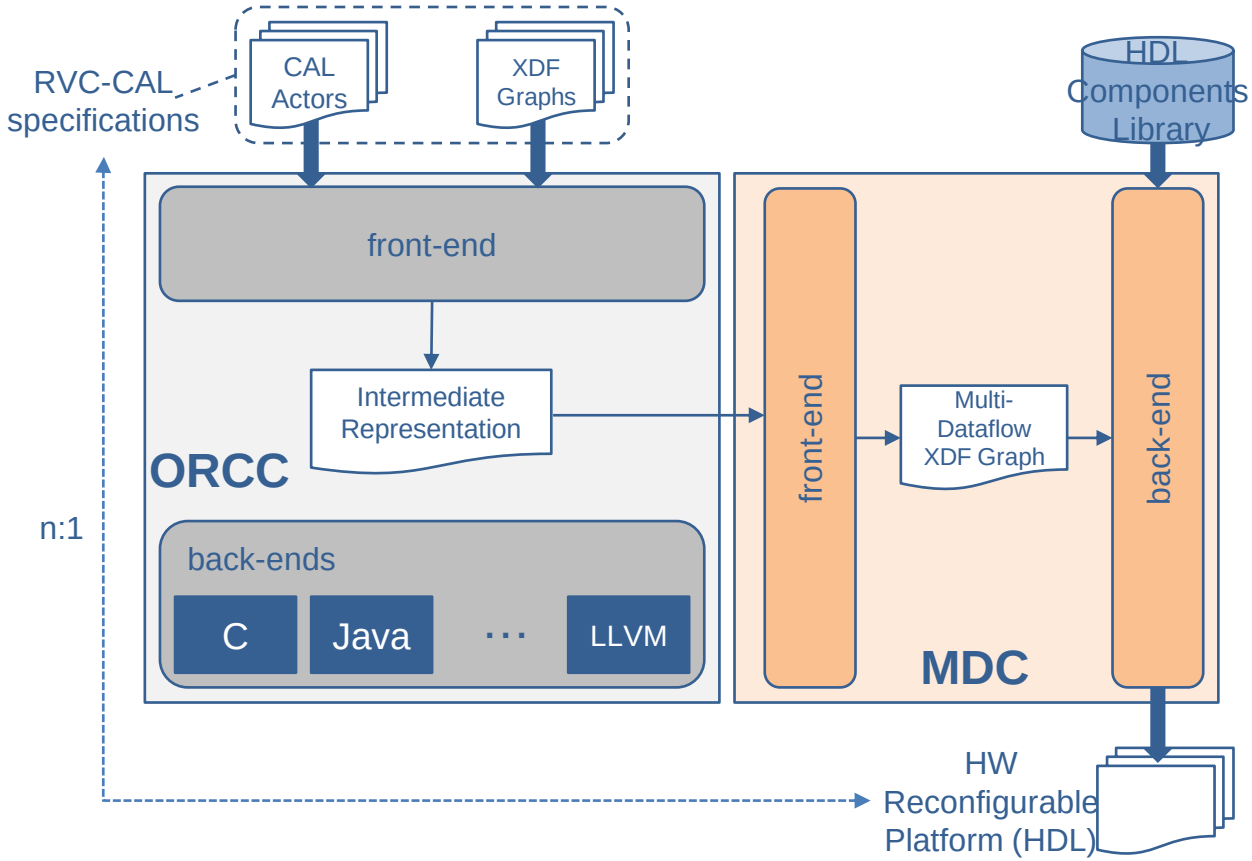
MDC exploits the ORCC front-end to generate, from a set of RVC-CAL specifications, the hardware description of a reconfigurable platform.

Composition: the Multi-Dataflow Composer (1)



MDC exploits the ORCC front-end to generate, from a set of RVC-CAL specifications, the hardware description of a reconfigurable platform.

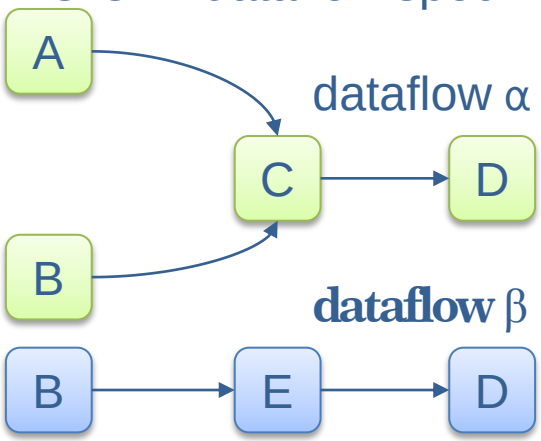
Composition: the Multi-Dataflow Composer (1)



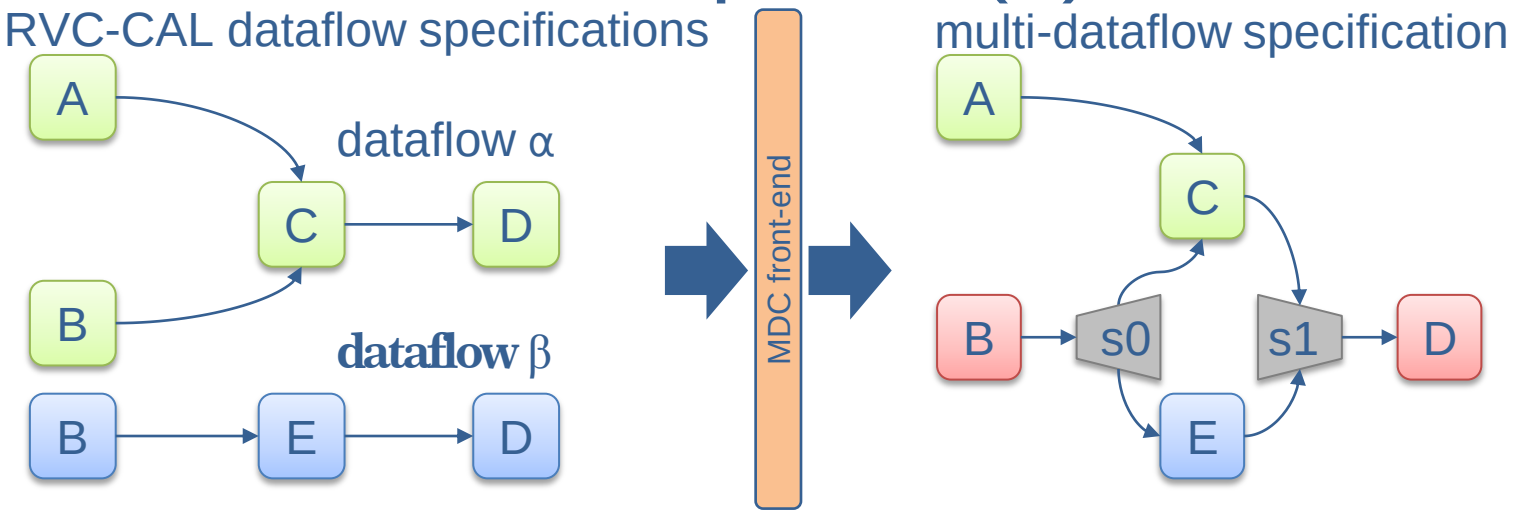
MDC exploits the ORCC front-end to generate, from a set of RVC-CAL specifications, the hardware description of a reconfigurable platform.

Composition: the Multi-Dataflow Composer (2)

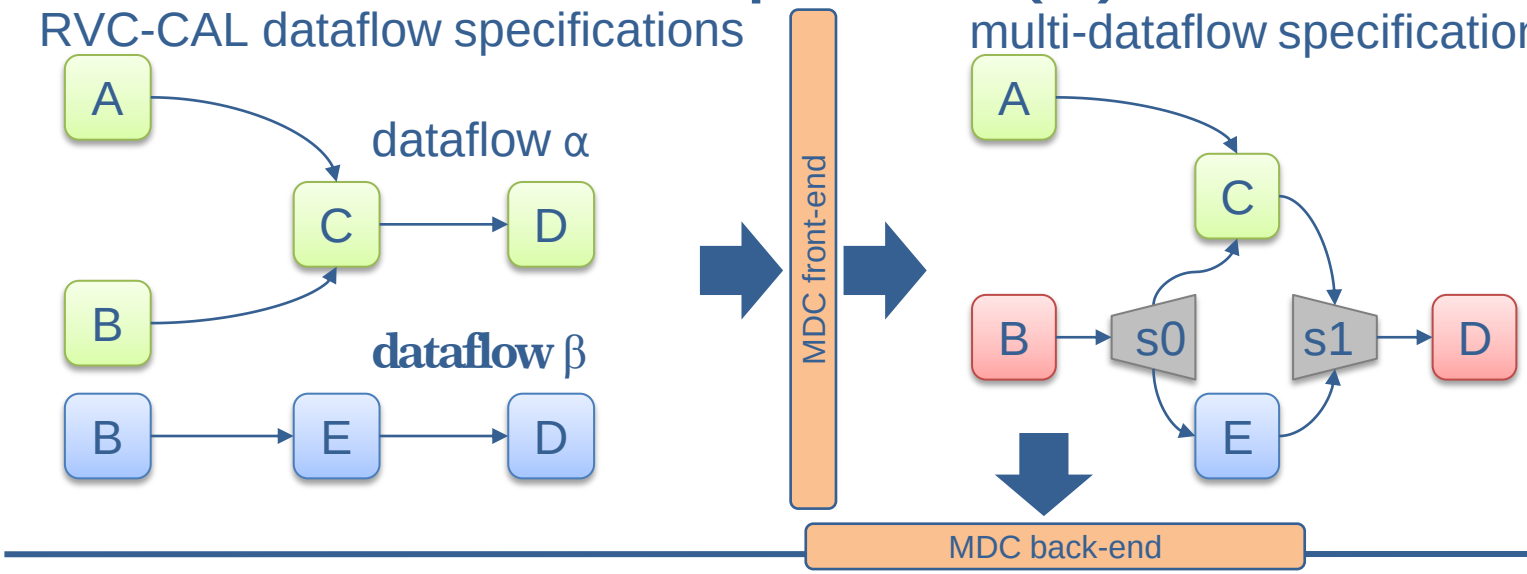
RVC-CAL dataflow specifications

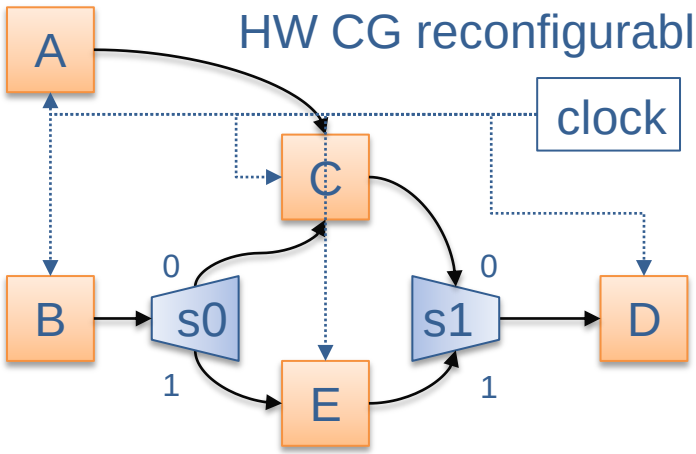


Composition: the Multi-Dataflow Composer (2)



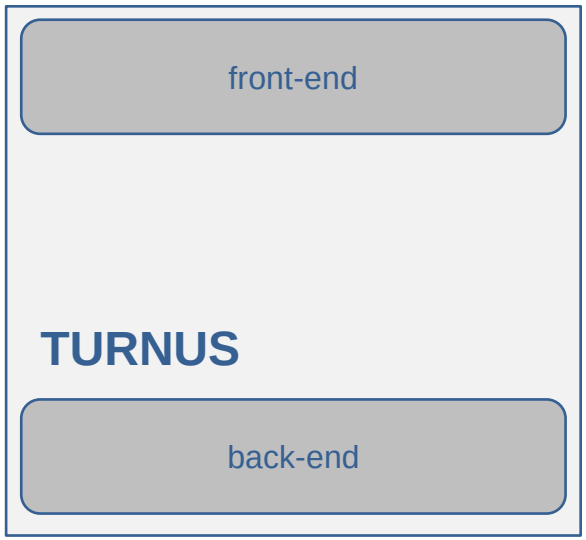
Composition: the Multi-Dataflow Composer (2)





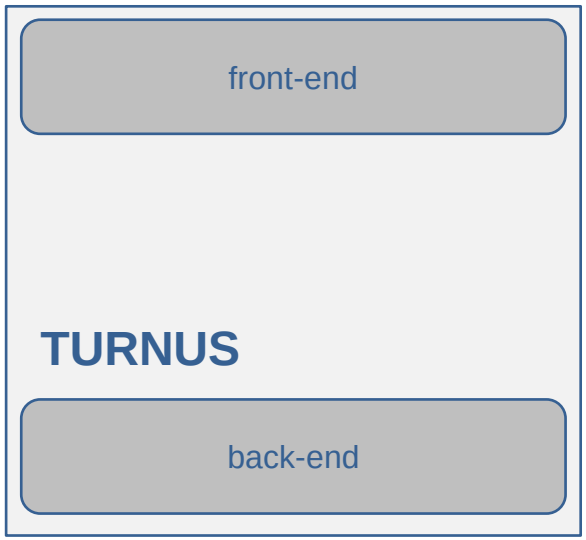
config LUT	α	β
s1	0	1
s2	0	1

Optimization: TURNUS Co-Exploration Framework



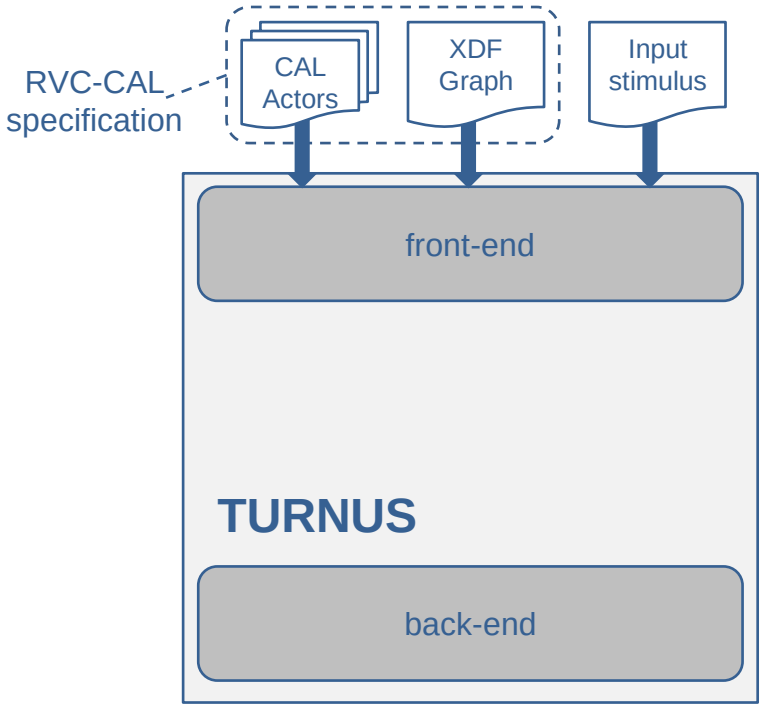
TURNUS is a design space exploration framework for heterogeneous parallel systems. It provides high-level modeling and simulation methods and tools
for system level performances estimation and optimization.

Optimization: TURNUS Co-Exploration Framework



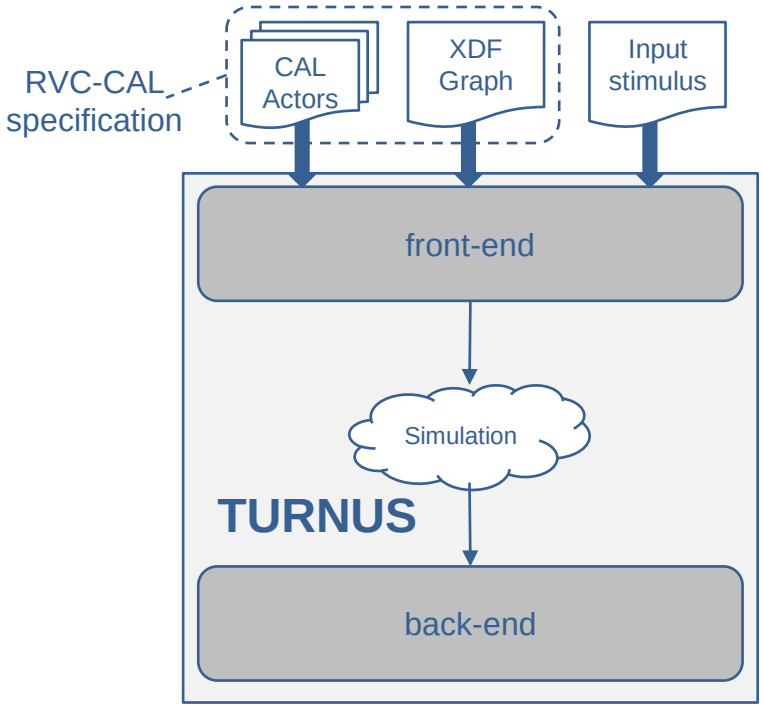
It provides the **execution causation traces** for a given dataflow specification in relation to a particular input stimulus. The causation traces are graphs describing the dependencies among the actions executed during the input stimulus processing.

Optimization: TURNUS Co-Exploration Framework



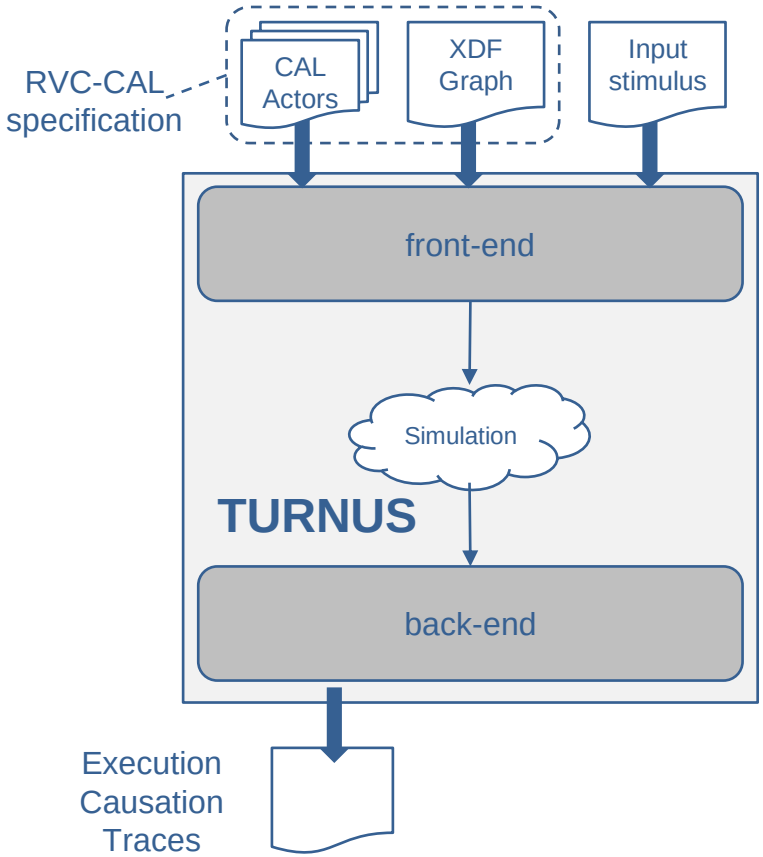
It provides the **execution causation traces** for a given dataflow specification in relation to a particular input stimulus. The causation traces are graphs describing the dependencies among the actions executed during the input stimulus processing.

Optimization: TURNUS Co-Exploration Framework



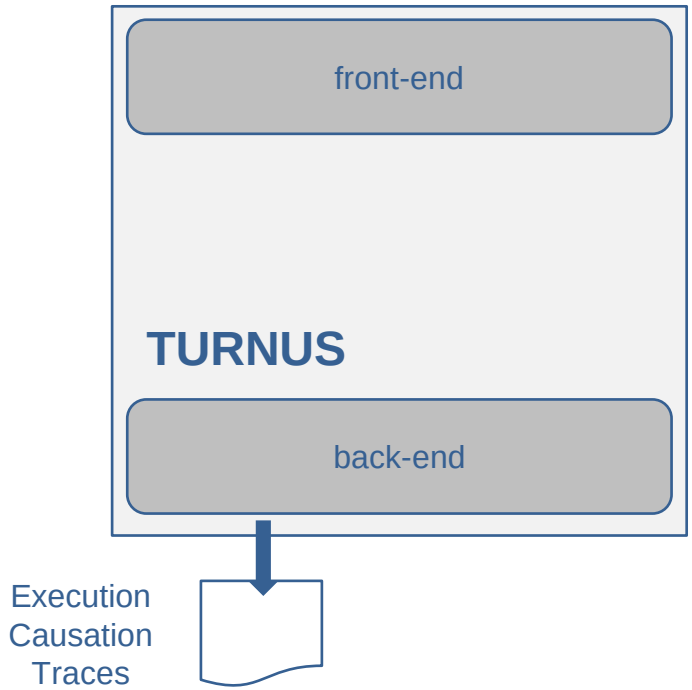
It provides the **execution causation traces** for a given dataflow specification in relation to a particular input stimulus. The causation traces are graphs describing the dependencies among the actions executed during the input stimulus processing.

Optimization: TURNUS Co-Exploration Framework



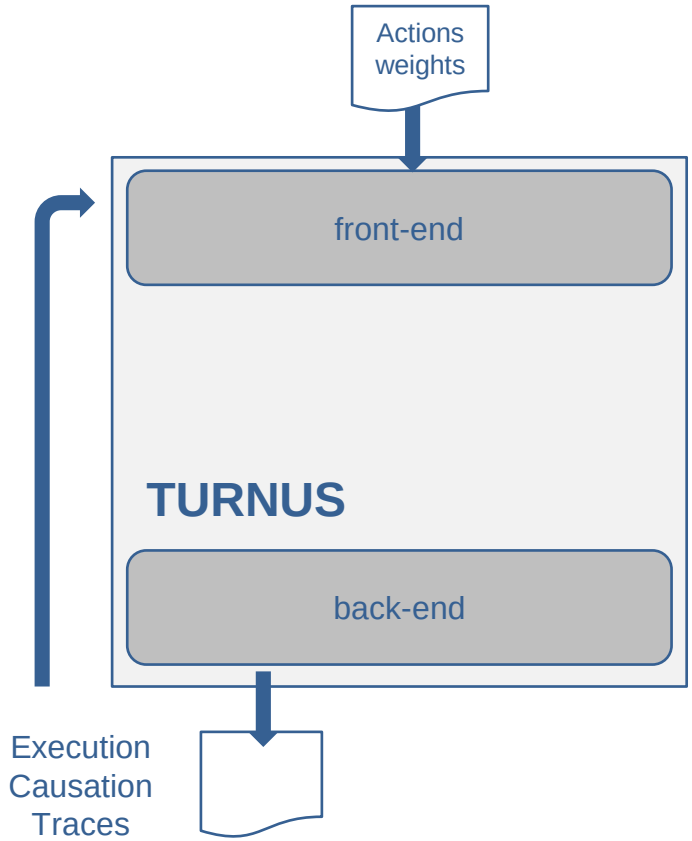
It provides the **execution causation traces** for a given dataflow specification in relation to a particular input stimulus. The causation traces are graphs describing the dependencies among the actions executed during the input stimulus processing.

Optimization: TURNUS Co-Exploration Framework



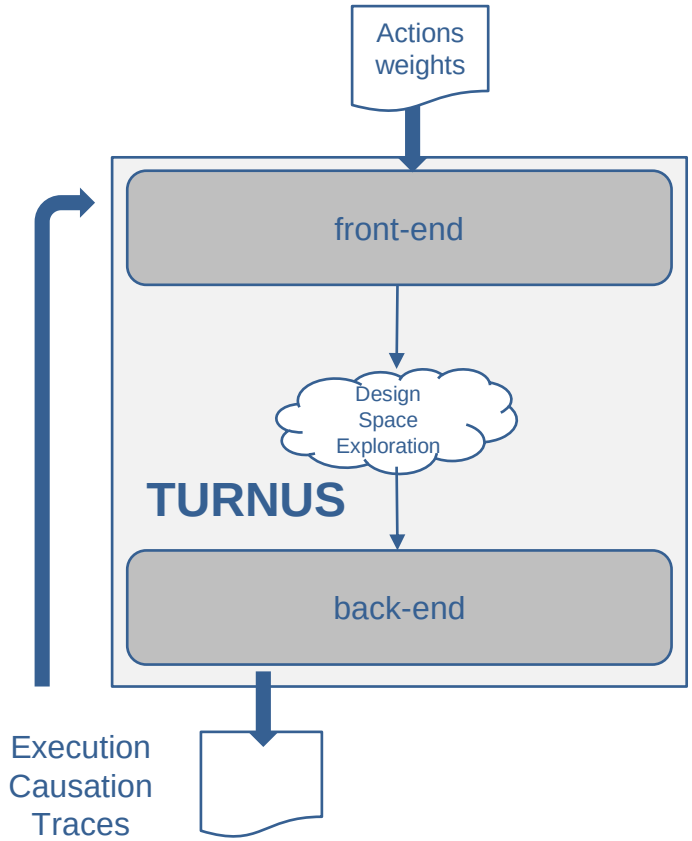
The **causation traces post-processing** analyses the causation traces along with given action weights (latency of the actions for a specific target platform) in order to optimize the size of the FIFO memories in the dataflow network through a Design Space Exploration.

Optimization: TURNUS Co-Exploration Framework



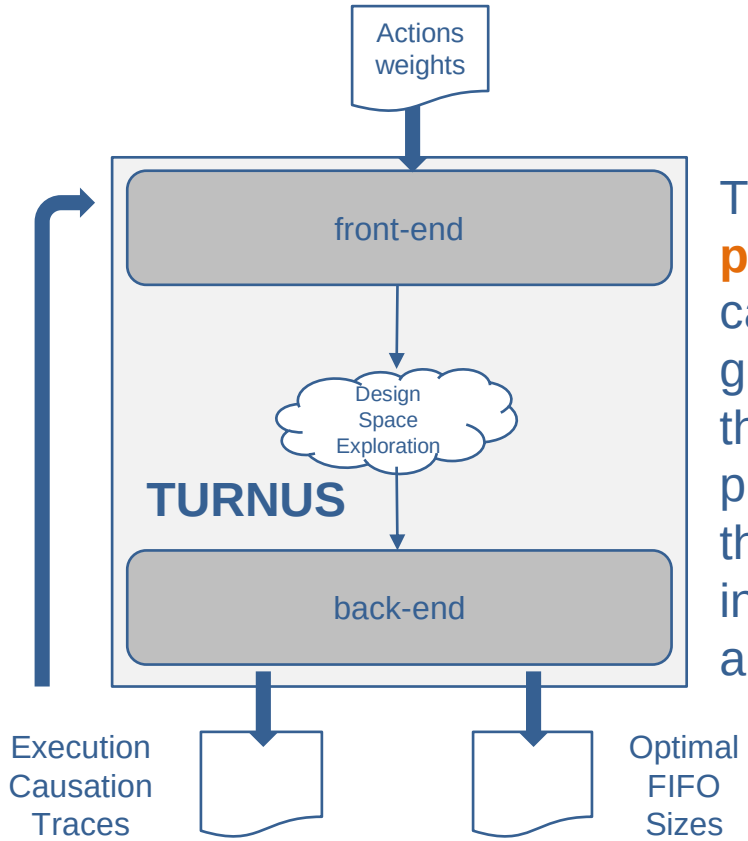
The **causation traces post-processing** analyses the causation traces along with given action weights (latency of the actions for a specific target platform) in order to optimize the size of the FIFO memories in the dataflow network through a Design Space Exploration.

Optimization: TURNUS Co-Exploration Framework



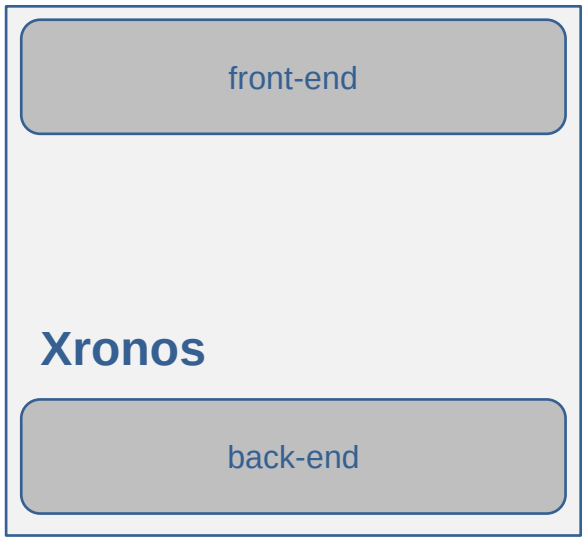
The **causation traces post-processing** analyses the causation traces along with given action weights (latency of the actions for a specific target platform) in order to optimize the size of the FIFO memories in the dataflow network through a Design Space Exploration.

Optimization: TURNUS Co-Exploration Framework



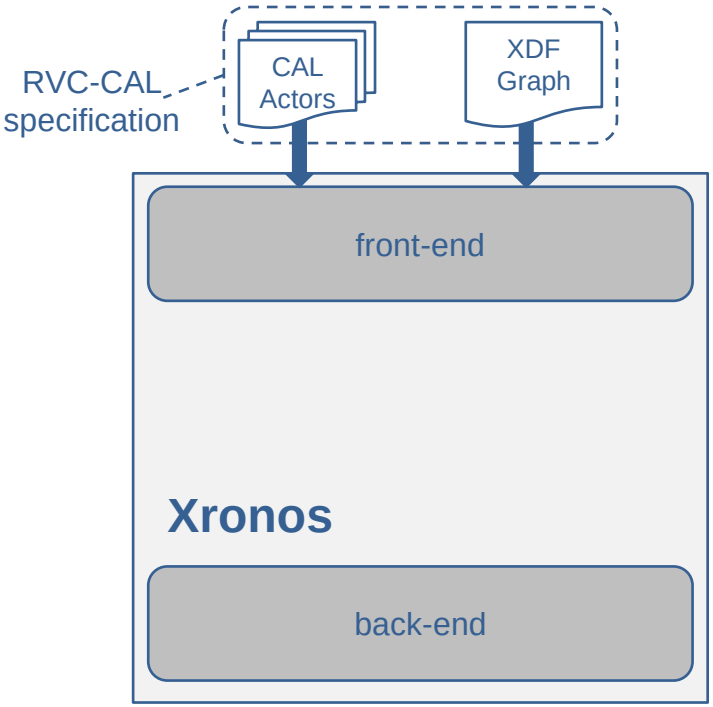
The **causation traces post-processing** analyses the causation traces along with given action weights (latency of the actions for a specific target platform) in order to optimize the size of the FIFO memories in the dataflow network through a Design Space Exploration.

RTL Generation: Xronos High Level Synthesis (1)



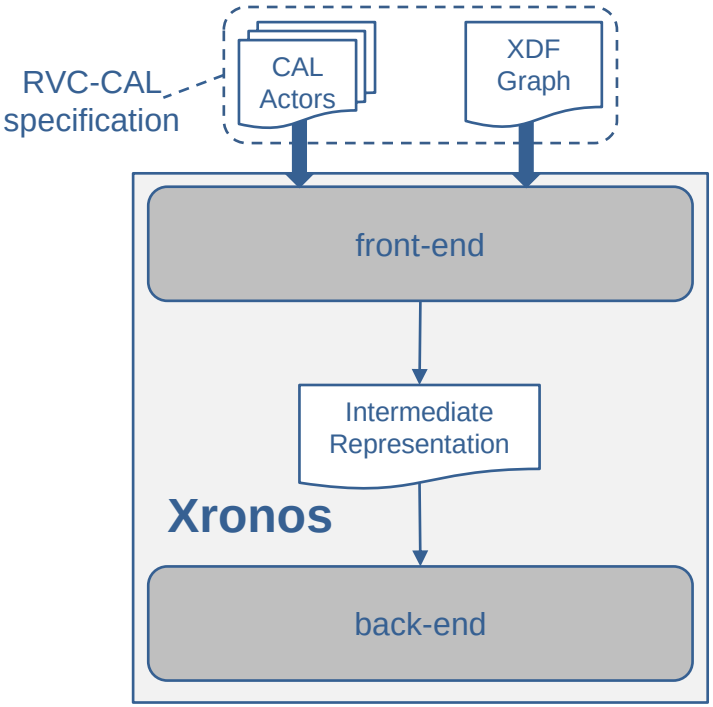
Xronos is a framework for the generation of RTL descriptions from dataflow or sequential applications.
It supports three basic data types (int, uint and bool), flow control statements (branches, loops, parallel blocks) and procedural abstractions (sequences of statements) called *tasks.*

RTL Generation: Xronos High Level Synthesis (1)



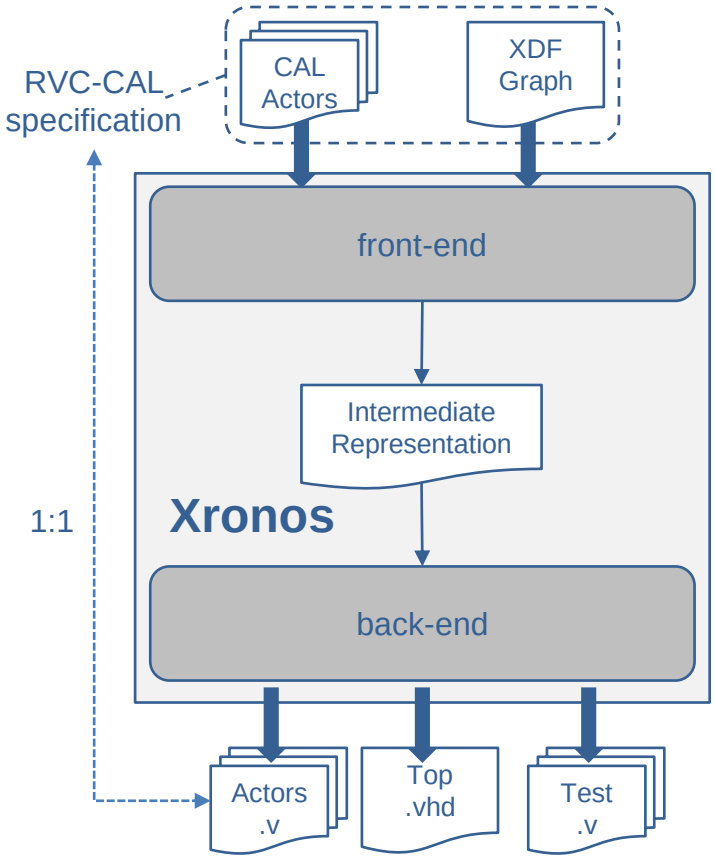
Xronos is a framework for the generation of RTL descriptions from dataflow or sequential applications.
It supports three basic data types (int, uint and bool), flow control statements (branches, loops, parallel blocks) and procedural abstractions (sequences of statements) called *tasks.*

RTL Generation: Xronos High Level Synthesis (1)



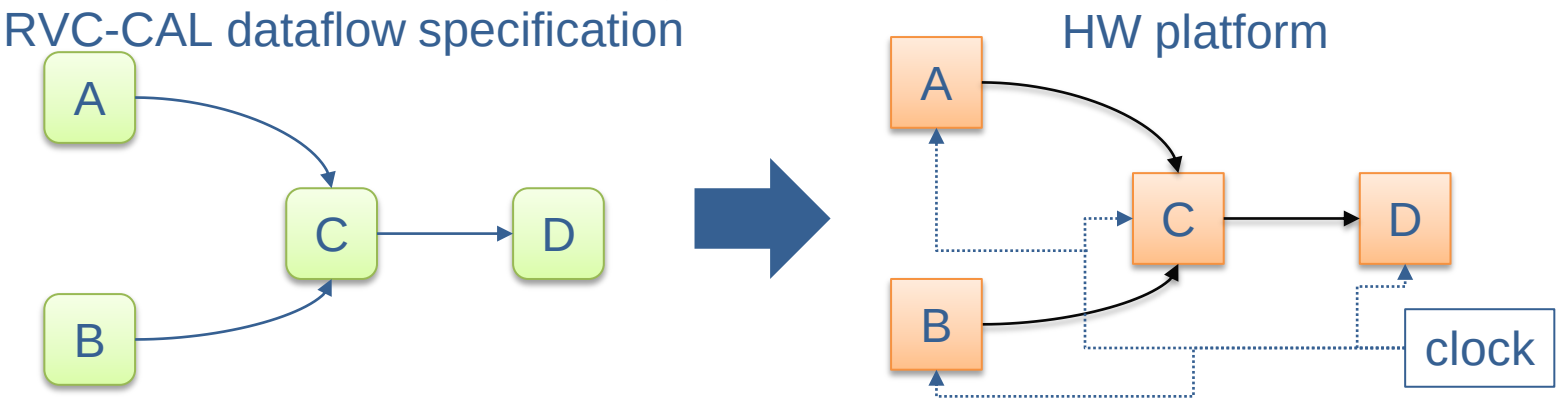
Xronos is a framework for the generation of RTL descriptions from dataflow or sequential applications.
It supports three basic data types (int, uint and bool), flow control statements (branches, loops, parallel blocks) and procedural abstractions (sequences of statements) called *tasks.*

RTL Generation: Xronos High Level Synthesis (1)

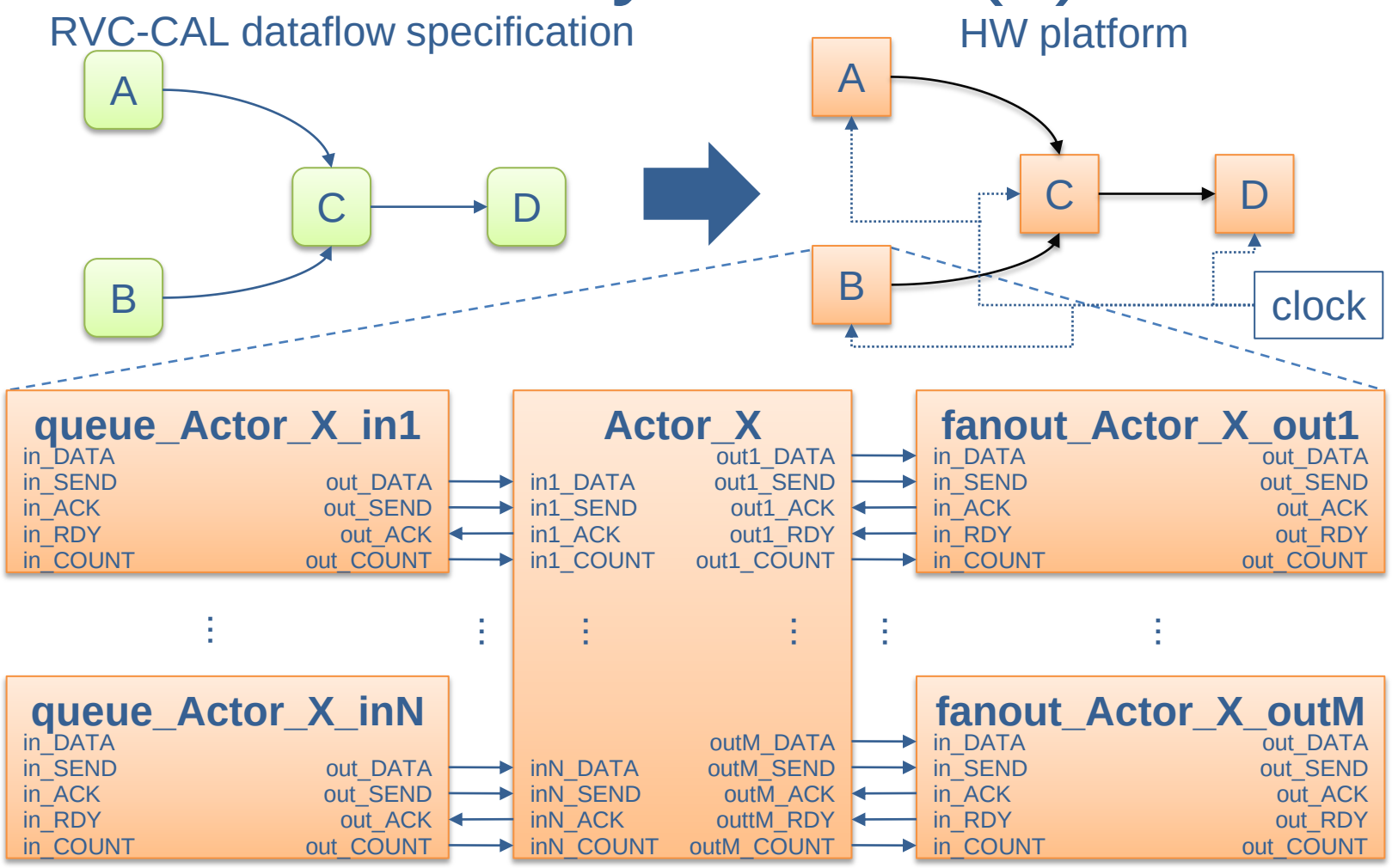


Xronos is a framework for the generation of RTL descriptions from dataflow or sequential applications.
It supports three basic data types (int, uint and bool), flow control statements (branches, loops, parallel blocks) and procedural abstractions (sequences of statements) called *tasks.*

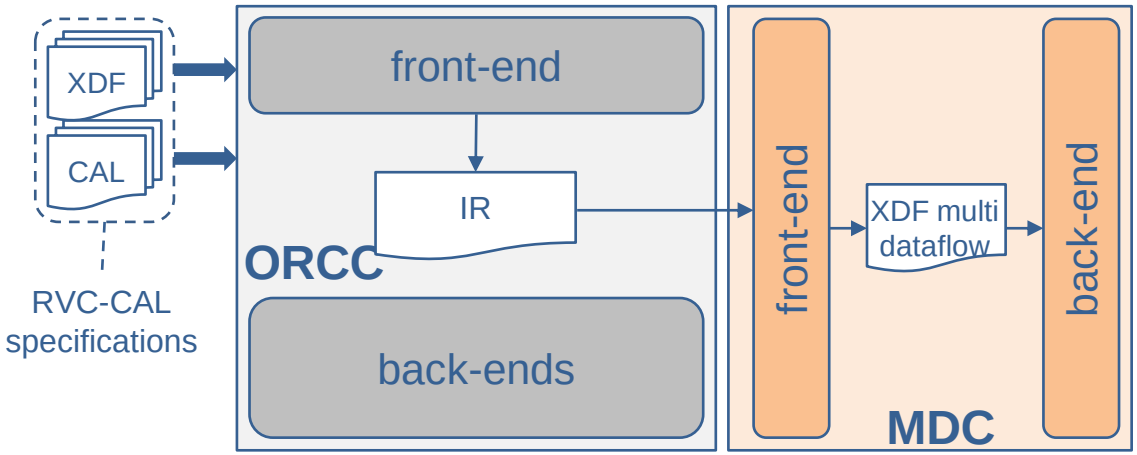
RTL Generation: Xronos High Level Synthesis (2)



RTL Generation: Xronos High Level Synthesis (2)

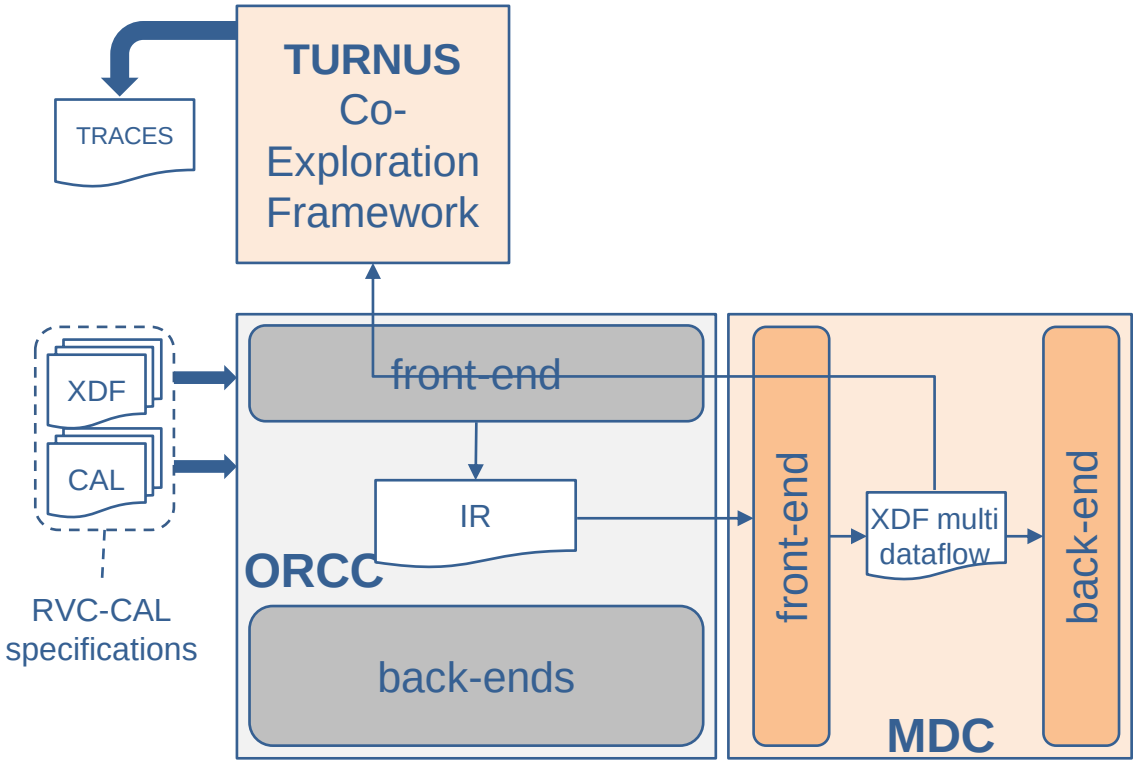


Tools Integration



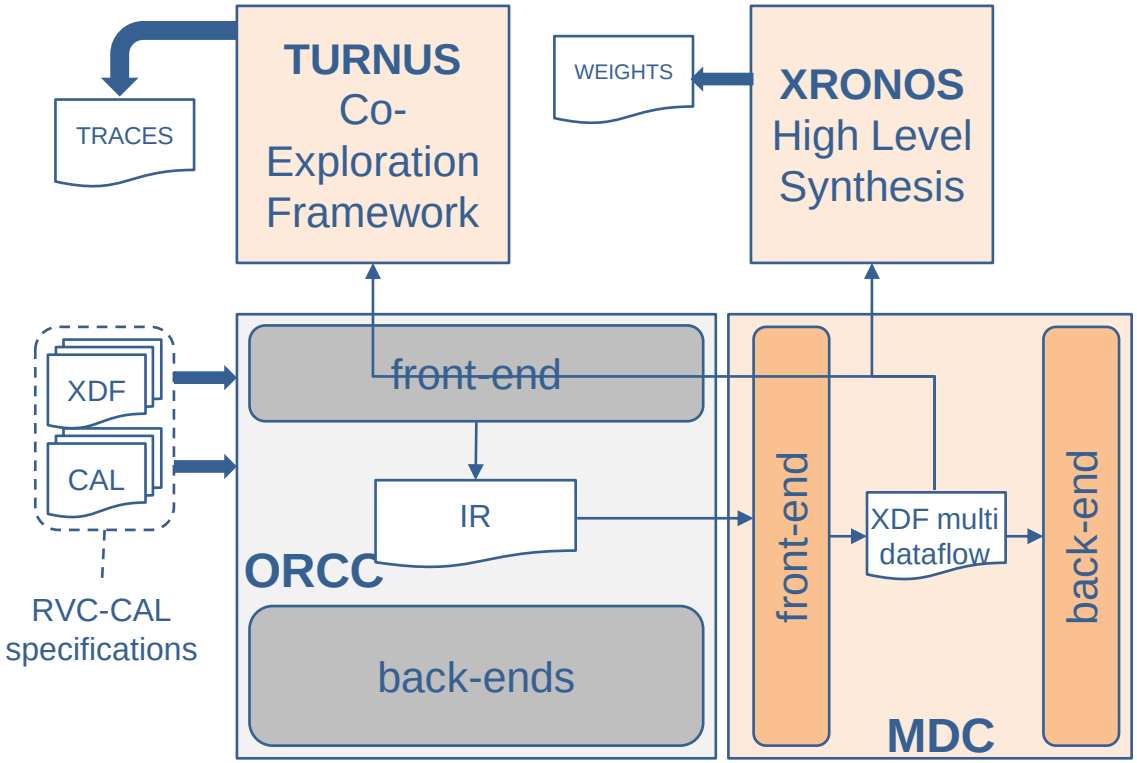
START: MDC compeses the multi-dataflow network

Tools Integration



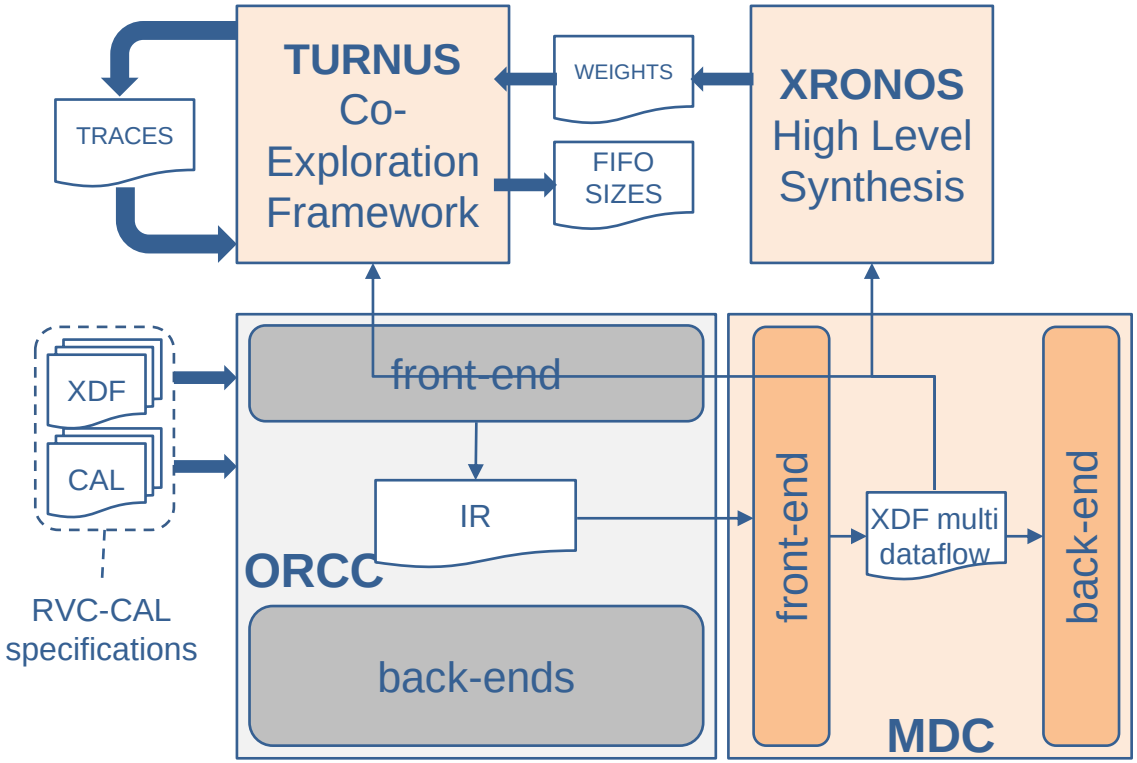
STEP 1: TURNUS generates the execution causation traces

Tools Integration



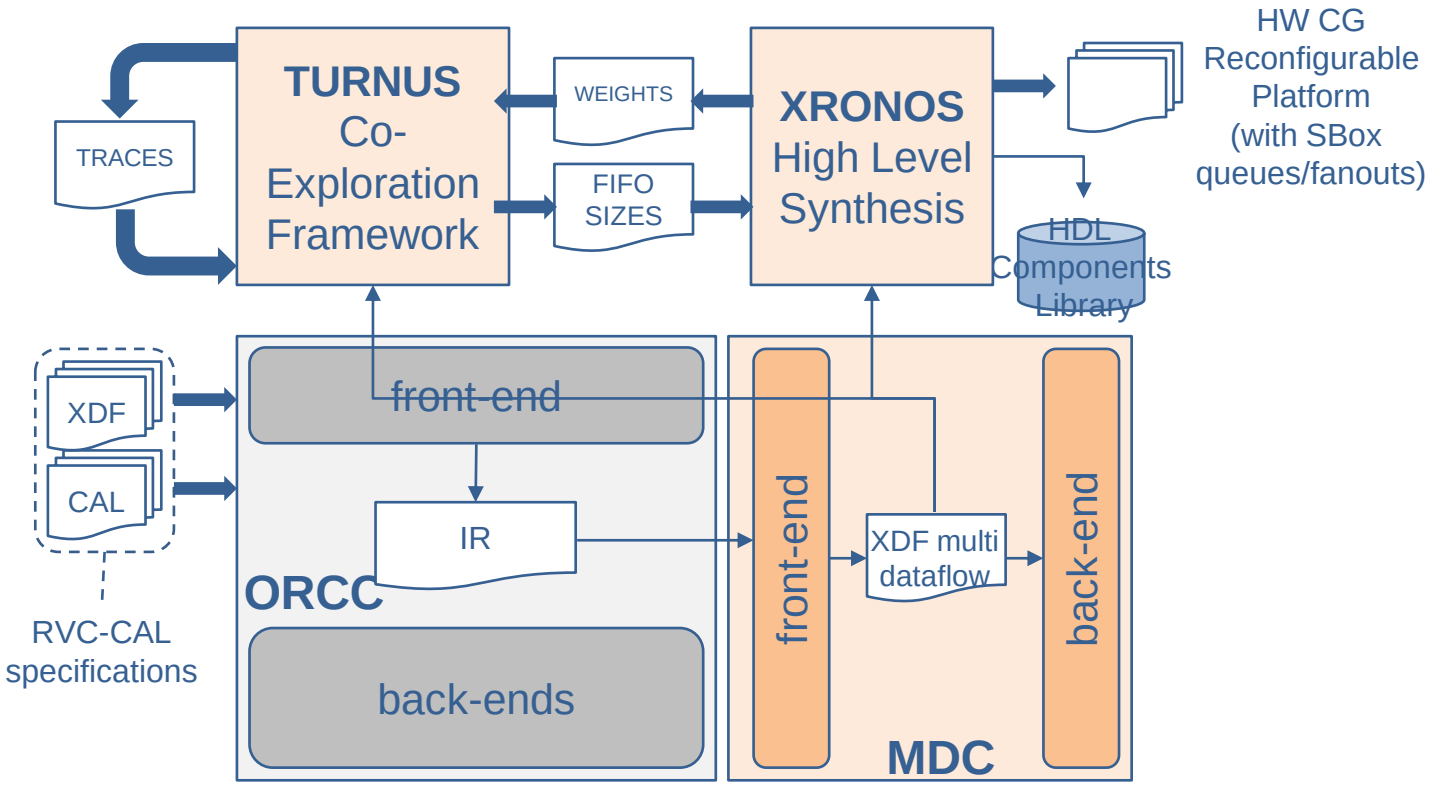
STEP 2: Xronos generates the actions weights

Tools Integration



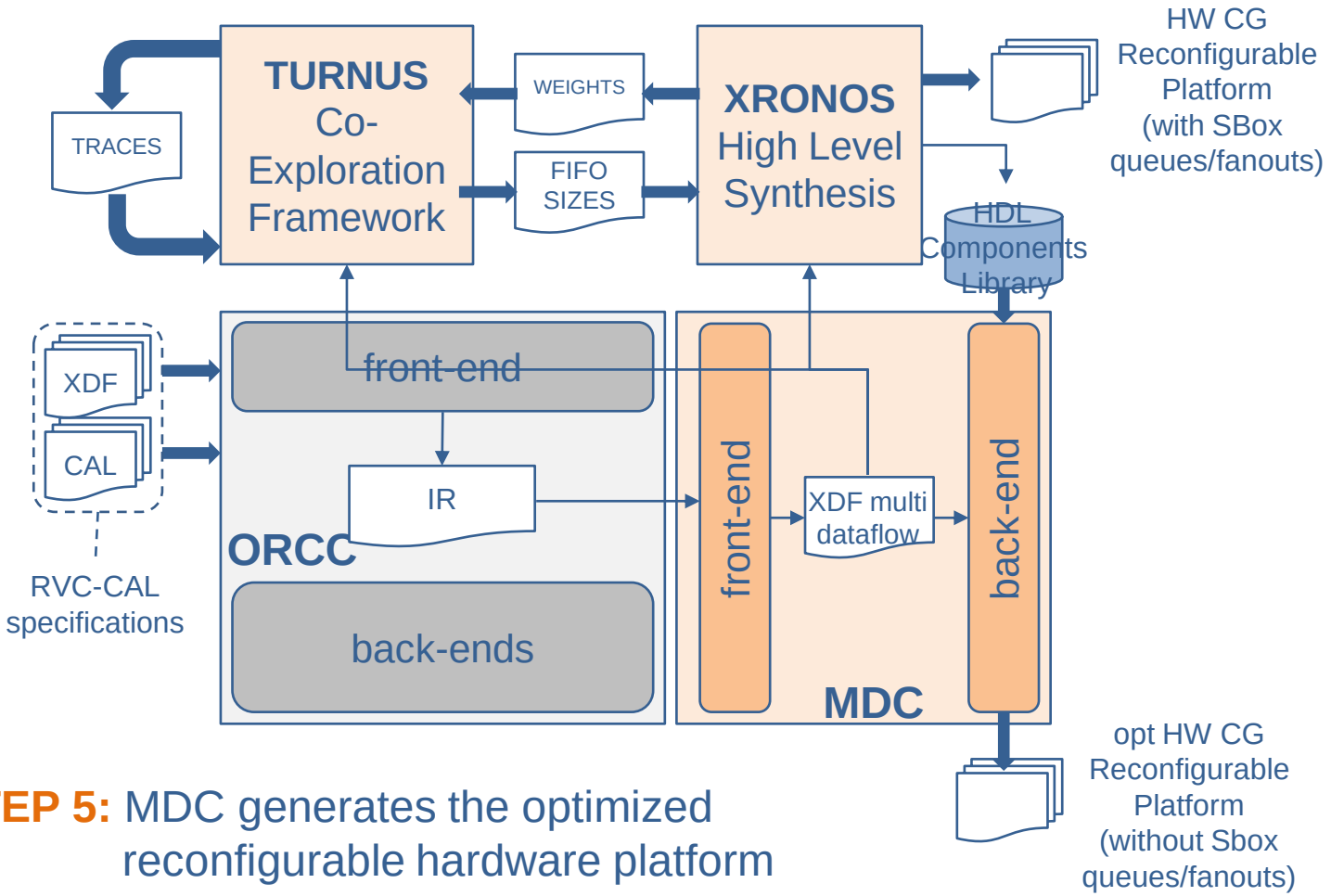
STEP 3: TURNUS generates the FIFO optimal sizes

Tools Integration



STEP 4: Xronos generates the RVC compliant reconfigurable hardware platform

Tools Integration

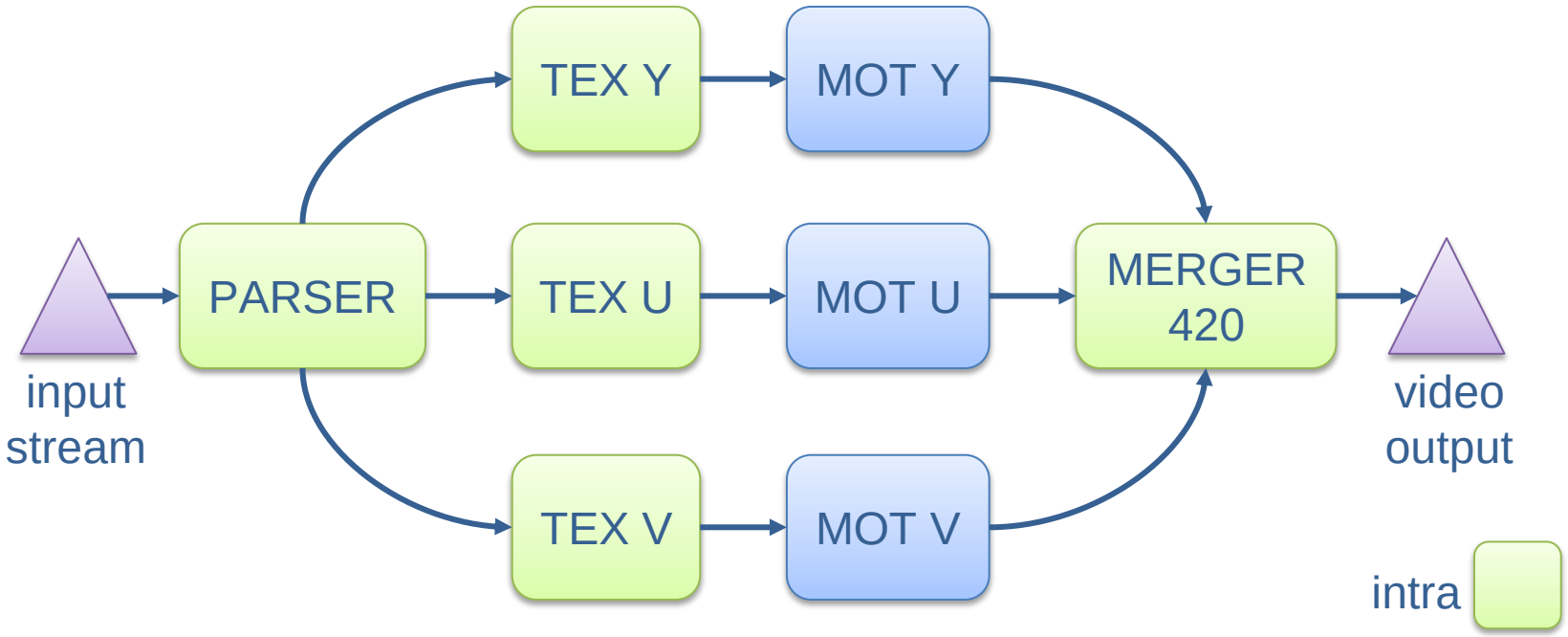


STEP 5: MDC generates the optimized reconfigurable hardware platform

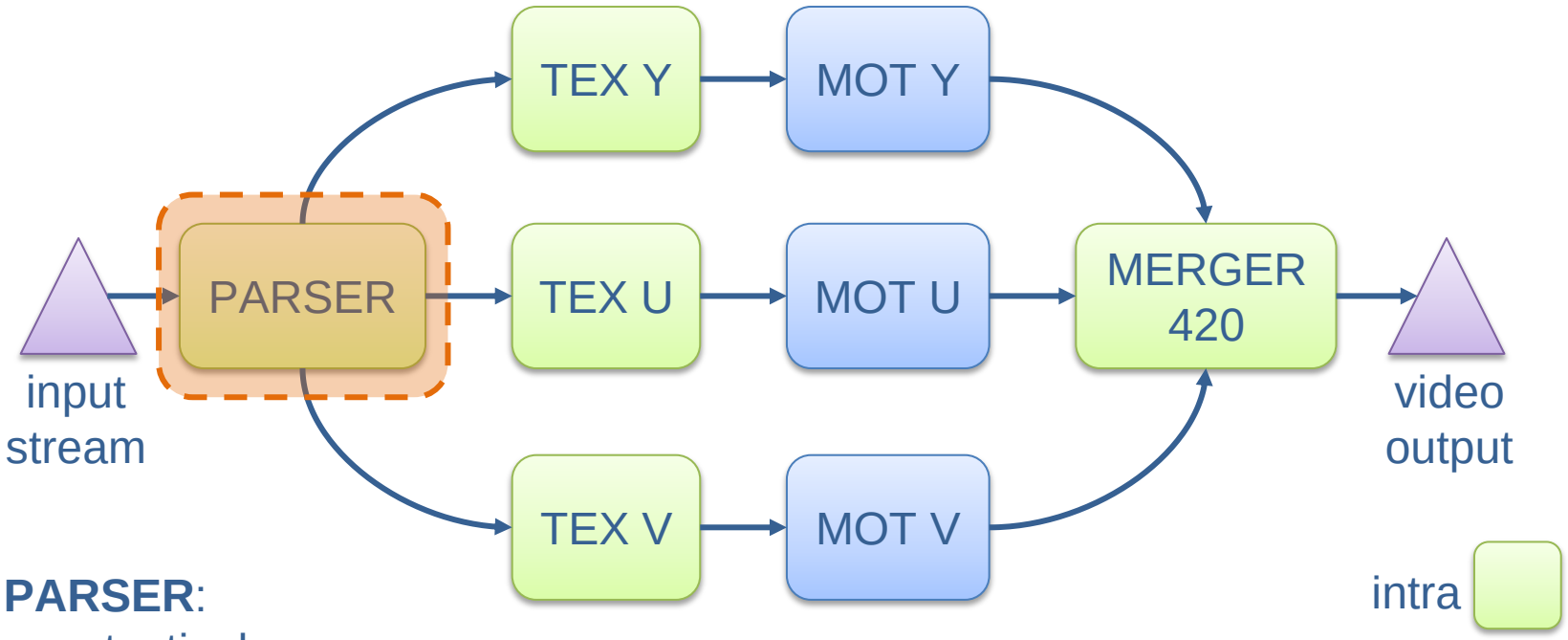
Outline

- Introduction
  - Problem Statement
  - Two Step Problem Solving
    - Step 1: Coarse-Grained Reconfigurability
    - Step 2: Dataflow Model of Computation and RVC-CAL
  - Generation of Multi-Dataflow Graphs
- Automated Design Flow
  - Composition: the Multi-Dataflow Composer
  - Optimization: TURNUS Co-Exploration Framework
  - RTL Generation: Xronos High Level Synthesis
  - Tools Integration
- Experimental Results
  - An MPEG-4 SP Decoder Use Case
  - Synthesis Results
  - Performance Results
- Conclusions

An MPEG-4 SP Decoder Use Case (1)

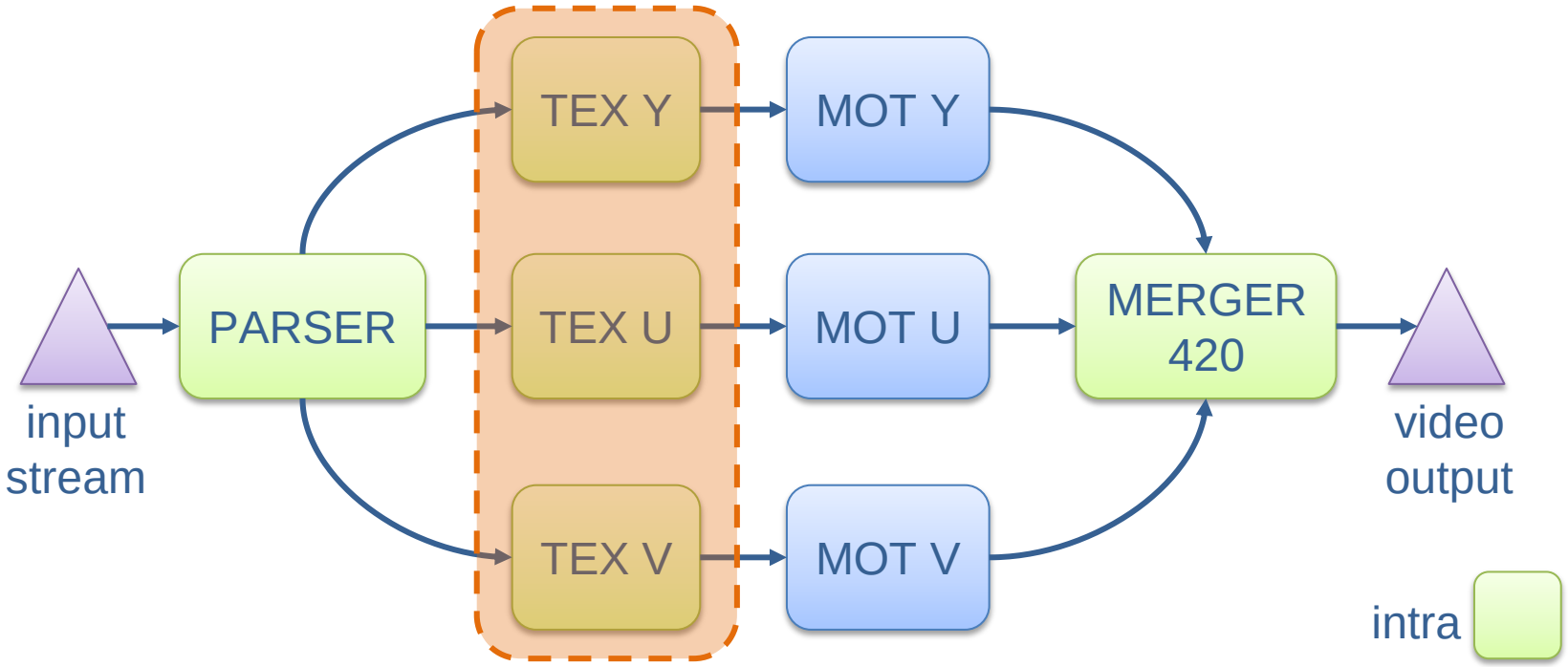


An MPEG-4 SP Decoder Use Case (1)



PARSER:
- syntactical parser
- variable length coding

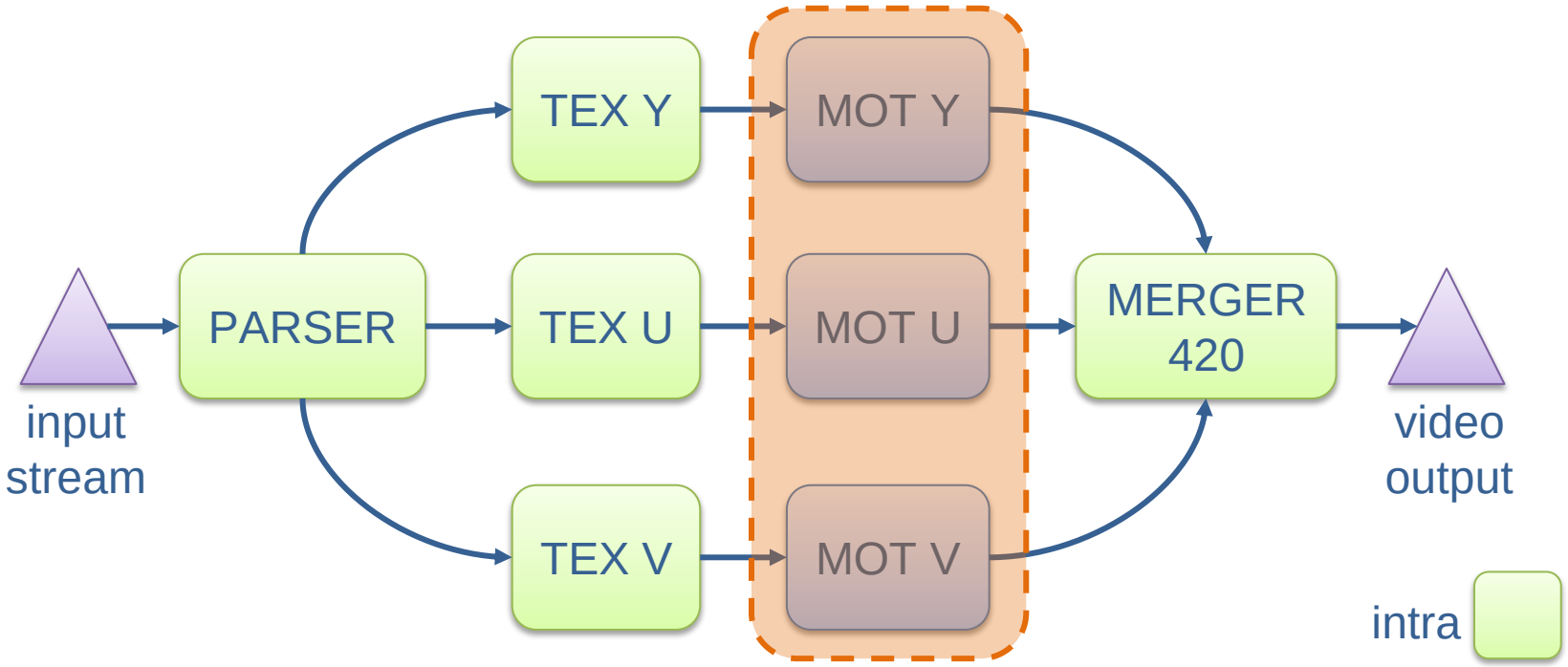
An MPEG-4 SP Decoder Use Case (1)



TEX (residual decoding):

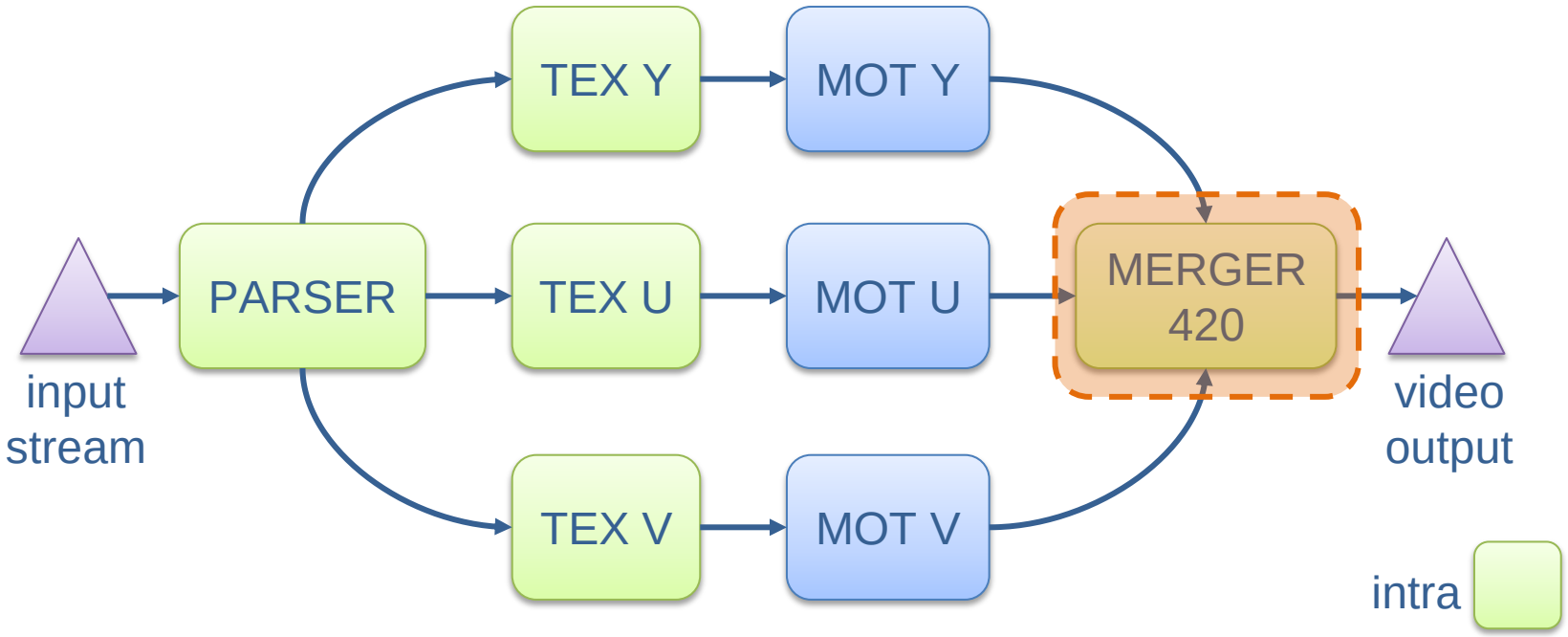
- AC-DC prediction
- inverse scan
- inverse quantization
- IDCT

An MPEG-4 SP Decoder Use Case (1)



MOT (motion compensation):
- framebuffer
- interpolation
- residual error addition

An MPEG-4 SP Decoder Use Case (1)



MERGER 420:
• 420 decoded components merging

An MPEG-4 SP Decoder Use Case (2)

Different designs composition in terms of functionality and role of the involved actors.

design	number of actors				
	ordinary*	SBox	not shared**	shared**	overall
intra	32	0	32	0	32
full	38	0	38	0	38
parallel	70	0	70	0	70
reconf	45	45	20	25	90
opt_reconf	45	45	20	25	90

* computational actors (not SBoxes).
** referred only to the computational actors (not to SBoxes).

An MPEG-4 SP Decoder Use Case (2)

Different designs composition in terms of functionality and role of the involved actors.

design	number of actors				
	ordinary*	SBox	not shared**	shared**	overall
intra	32	0	32	0	32
full	38	0	38	0	38
parallel	70	0	70	0	70
reconf	45	45	20	25	90
opt_reconf	45	45	20	25	90

* computational actors (not SBoxes).
** referred only to the computational actors (not to SBoxes).

An MPEG-4 SP Decoder Use Case (2)

Different designs composition in terms of functionality and role of the involved actors.

design	number of actors				
	ordinary*	SBox	not shared**	shared**	overall
intra	32	0	32	0	32
full	38	0	38	0	38
parallel	70	0	70	0	70
reconf	45	45	20	25	90
opt_reconf	45	45	20	25	90

* computational actors (not SBoxes).
** referred only to the computational actors (not to SBoxes).

An MPEG-4 SP Decoder Use Case (2)

Different designs composition in terms of functionality and role of the involved actors.

design	number of actors				
	ordinary*	SBox	not shared**	shared**	overall
intra	32	0	32	0	32
full	38	0	38	0	38
parallel	70	0	70	0	70
reconf	45	➡45	20	➡25	90
opt_reconf	45	➡45	20	➡25	90

* computational actors (not SBoxes).
** referred only to the computational actors (not to SBoxes).

An MPEG-4 SP Decoder Use Case (2)

Different designs composition in terms of functionality and role of the involved actors.

design	number of actors				
	ordinary*	SBox	not shared**	shared**	overall
intra	32	0	32	0	32
full	38	0	38	0	38
parallel	70	0	70	0	70
reconf	45	45	20	25	90
opt_reconf	45	45	20	25	90

* computational actors (not SBoxes).
** referred only to the computational actors (not to SBoxes).

Synthesis Results (1)

Results retrieved from the Xilinx Synthesis Technology tool targeting a Xilinx Virtex 5 330 FPGA board.

resource	design				
	parallel	reconf	Δ%	opt_reconf	Δ%
Slices	22495	18447	-19	16383	-27
Slice Regs	27567	23034	-16	23674	-14
Slice LUTs	67445	52836	-22	48545	-28
LUT-FF pairs	71011	55946	-21	52518	-26
BRAMs	148	154	+4	112	-24
DSPs	36	18	-50	18	-50
Max freq [MHz]	25,43	25,22	-1	25,38	0

Δ% percentage increment/reduction between the parallel and the reconfigurable designs.

Synthesis Results (1)

Results retrieved from the Xilinx Synthesis Technology tool targeting a Xilinx Virtex 5 330 FPGA board.

resource	design				
	parallel	reconf	Δ%	opt_reconf	Δ%
Slices	22495	18447	-19	16383	-27
Slice Regs	27567	23034	-16	23674	-14
Slice LUTs	67445	52836	-22	48545	-28
LUT-FF pairs	71011	55946	-21	52518	-26
BRAMs	148	154	+4	112	-24
DSPs	36	18	-50	18	-50
Max freq [MHz]	25,43	25,22	-1	25,38	0

Δ% percentage increment/reduction between the parallel and the reconfigurable designs.

Synthesis Results (1)

Results retrieved from the Xilinx Synthesis Technology tool targeting a Xilinx Virtex 5 330 FPGA board.

resource	design				
	parallel	reconf	Δ%	opt_reconf	Δ%
Slices	22495	18447	-19	16383	-27
Slice Regs	27567	23034	-16	23674	-14
Slice LUTs	67445	52836	-22	48545	-28
LUT-FF pairs	71011	55946	-21	52518	-26
BRAMs	148	154	+4	112	-24
DSPs	36	18	-50	18	-50
Max freq [MHz]	25,43	25,22	-1	25,38	0

Δ% percentage increment/reduction between the parallel and the reconfigurable designs.

Synthesis Results (1)

Results retrieved from the Xilinx Synthesis Technology tool targeting a Xilinx Virtex 5 330 FPGA board.

resource	design				
	parallel	reconf	Δ%	opt_reconf	Δ%
Slices	22495	18447	-19	16383	-27
Slice Regs	27567	23034	-16	23674	-14
Slice LUTs	67445	52836	-22	48545	-28
LUT-FF pairs	71011	55946	-21	52518	-26
BRAMs	148	154	+4	112	-24
DSPs	36	18	-50	18	-50
Max freq [MHz]	25,43	25,22	-1	25,38	0

Δ% percentage increment/reduction between the parallel and the reconfigurable designs.

Synthesis Results (2)

Results retrieved from the XPower Analyzer tool targeting a Xilinx Virtex 5 330 FPGA board for a QCIF video sequence decoding.

resource power	design				
	parallel	reconf	Δ%	opt_reconf	Δ%
Clock	0,382	0,347	-9	0,323	-15
Logic	0,045	0,025	-44	0,024	-47
Signals	0,050	0,031	-38	0,031	-38
BRAMs	0,090	0,090	0	0,059	-34
TOT	0,567	0,493	-13	0,437	-23

Δ% percentage increment/reduction between the parallel and the reconfigurable designs.

Synthesis Results (2)

Results retrieved from the XPower Analyzer tool targeting a Xilinx Virtex 5 330 FPGA board for a QCIF video sequence decoding.

resource power	design				
	parallel	reconf	Δ%	opt_reconf	Δ%
Clock	0,382	0,347	-9	0,323	-15
Logic	0,045	0,025	-44	0,024	-47
Signals	0,050	0,031	-38	0,031	-38
BRAMs	0,090	0,090	0	0,059	-34
TOT	0,567	0,493	-13	0,437	-23

Δ% percentage increment/reduction between the parallel and the reconfigurable designs.

Synthesis Results (2)

Results retrieved from the XPower Analyzer tool targeting a Xilinx Virtex 5 330 FPGA board for a QCIF video sequence decoding.

resource power	design				
	parallel	reconf	Δ%	opt_reconf	Δ%
Clock	0,382	0,347	-9	0,323	-15
Logic	0,045	0,025	-44	0,024	-47
Signals	0,050	0,031	-38	0,031	-38
BRAMs	0,090	0,090	0	0,059	-34
TOT	0,567	0,493	-13	0,437	-23

Δ% percentage increment/reduction between the parallel and the reconfigurable designs.

Performance Results

Results are given in frames per second (fps) and are obtained as the average of fps for the intra and the full decoder configurations.

video sequence resolution	design				
	parallel	reconf	Δ%	opt_reconf	Δ%
QCIF	110	105	-5	105	-5
CIF	28	26	-6	26	-6

Δ% percentage increment/reduction between the parallel and the reconfigurable designs.

Performance Results

Results are given in frames per second (fps) and are obtained as the average of fps for the intra and the full decoder configurations.

video sequence resolution	design				
	parallel	reconf	Δ%	opt_reconf	Δ%
QCIF	110	105	-5	105	-5
CIF	28	26	-6	26	-6

Δ% percentage increment/reduction between the parallel and the reconfigurable designs.

Outline

- Introduction
  - Problem Statement
  - Two Step Problem Solving
    - Step 1: Coarse-Grained Reconfigurability
    - Step 2: Dataflow Model of Computation and RVC-CAL
  - Generation of Multi-Dataflow Graphs
- Automated Design Flow
  - Composition: the Multi-Dataflow Composer
  - Optimization: TURNUS Co-Exploration Framework
  - RTL Generation: Xronos High Level Synthesis
  - Tools Integration
- Experimental Results
  - An MPEG-4 SP Decoder Use Case
  - Synthesis Results
  - Performance Results
- Conclusions

Conclusions

• Challenge of modern electronic systems development: coupling **portability**, **flexibility** and **efficiency** with the **minimum design effort**.

Conclusions

- Challenge of modern electronic systems development: coupling **portability**, **flexibility** and **efficiency** with the **minimum design effort**.
- A solution is possible by exploiting the **dataflow** programming paradigm along with the **coarse-grained reconfigurability**.

Conclusions

- Challenge of modern electronic systems development: coupling **portability**, **flexibility** and **efficiency** with the **minimum design effort**.
- A solution is possible by exploiting the **dataflow** programming paradigm along with the **coarse-grained reconfigurability**.
- An **automated design flow** has been assembled, by the integration of different RVC-CAL tools (MDC, TURNUS, Xronos).

Conclusions

- Challenge of modern electronic systems development: coupling **portability**, **flexibility** and **efficiency** with the **minimum design effort**.
- A solution is possible by exploiting the **dataflow** programming paradigm along with the **coarse-grained reconfigurability**.
- An **automated design flow** has been assembled, by the integration of different RVC-CAL tools (MDC, TURNUS, Xronos).
- The proposed design flow has been validated on a real use case involving two configurations of the MPEG-4 Simple Profile decoder.
  - Results highlighted the effectiveness of the approach by achieving more than 25% of saving in terms of **resource utilization** and more than 20% of saving in terms **power consumption**.
  - The generated reconfigurable designs present a very **small performance penalty** (5-6%) with respect to the original decoder designs.

Acknowledgements

The research leading to these results has received funding from:


• the Region of Sardinia L.R.7/2007 under grant agreement CRP-18324 [RPCT Project].


• the Region of Sardinia, Young Researchers Grant, POR Sardegna FSE 2007-2013, L.R.7/2007 “Promotion of the scientific research and technological innovation in Sardinia”

XIV International Conference on Embedded Computer and Systems: Architectures, MOdeling and Simulation
SAMOS XIV - 2014 July 14th - Samos Island (Greece)

Carlo Sau
DIEE
Università degli Studi di Cagliari
carlo.sau@diee.unica.it

Automated Design Flow for Coarse-Grained Reconfigurable Platforms: an RVC-CAL Multi-Standard Decoder Use-Case

QUESTIONS