



**LUND**  
UNIVERSITY



# Reconfigurable Cell Array as an Enabler for Future Processing

Chenxin Zhang<sup>1</sup>, Per Andersson<sup>2</sup>, and Viktor Öwall<sup>1</sup>

<sup>1</sup>Department of Electrical and Information Technology

<sup>2</sup>Department of Computer Science

Lund University, Sweden

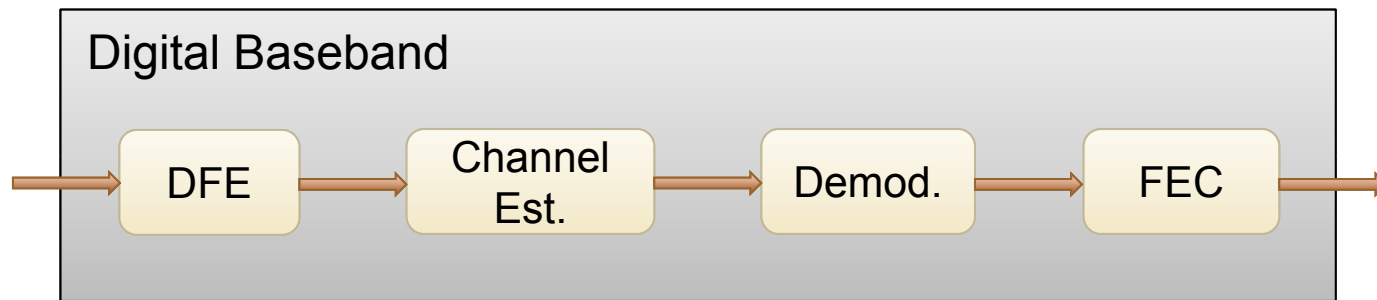
# Outline

- Motivation
- Coarse-grained reconfigurable cell array
  - Resource cell
  - System reconfiguration
- Software development
  - Application mapping
  - Design exploration
- Case study: multi-standard OFDM synchronization

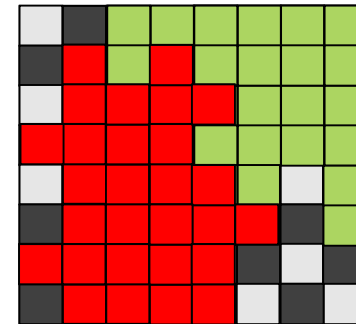


# Motivation

- Hardware sharing
  - Accelerators: poor hardware reusability

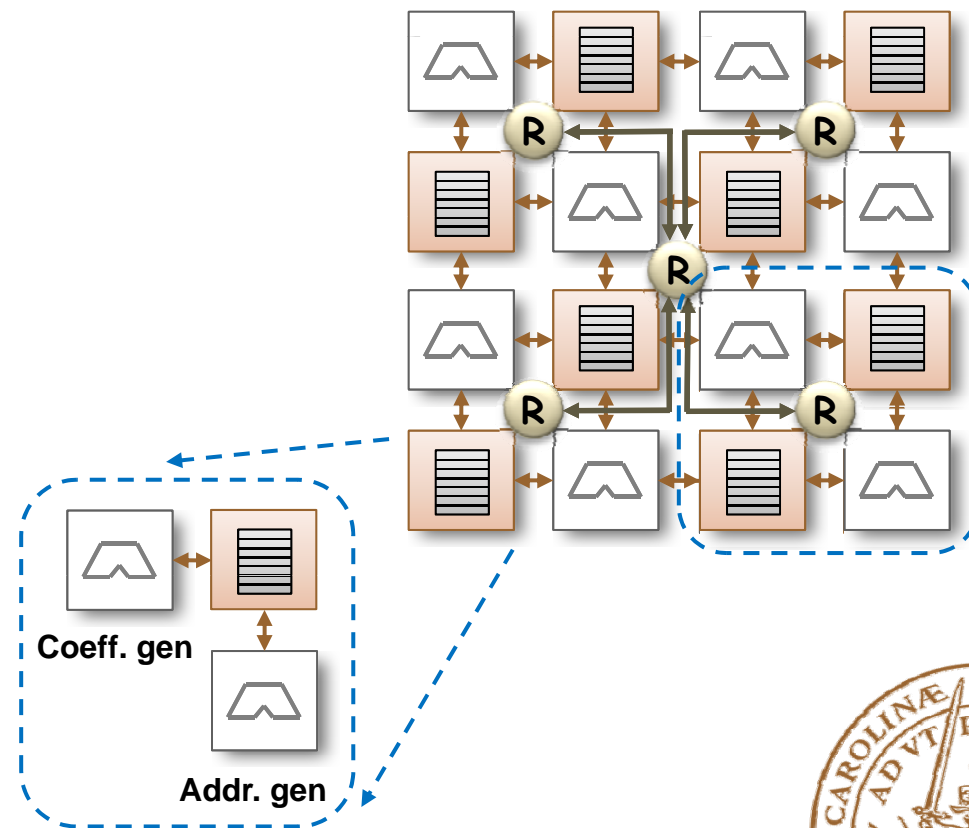


- Reconfigurable architecture
  - + Multi-task
  - + Multi-standard
  - + Multi-algorithm
  - Control overhead, e.g. area, power.



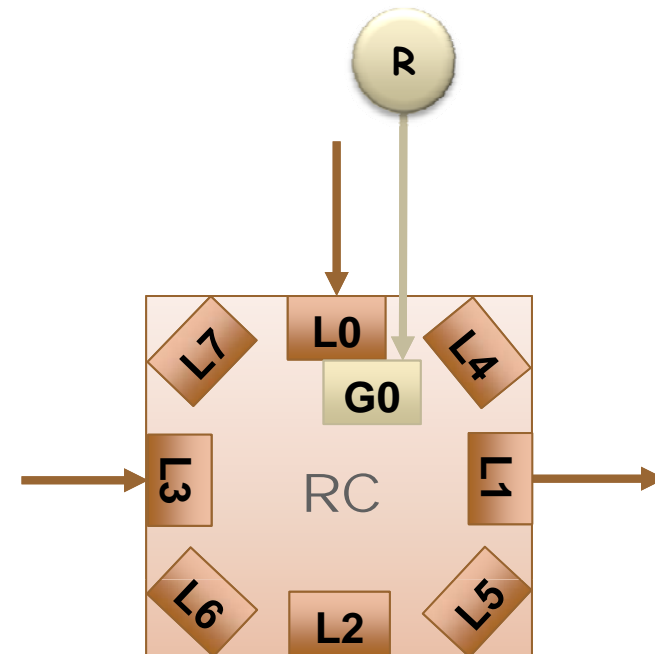
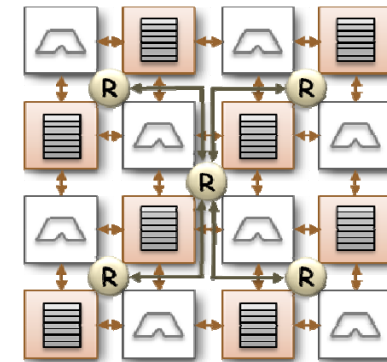
# System infrastructure

- An array of resource cells.
- Heterogeneous cell array:
  - Processing cell
  - Memory cell
  - Accelerator(e.g. no configuration)
- Hierarchical cell array.

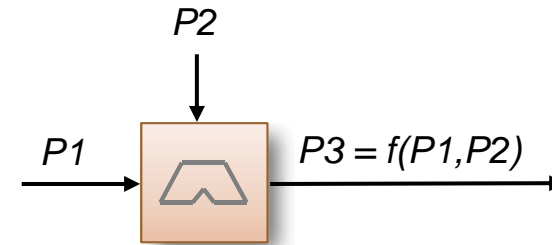


# Resource cell

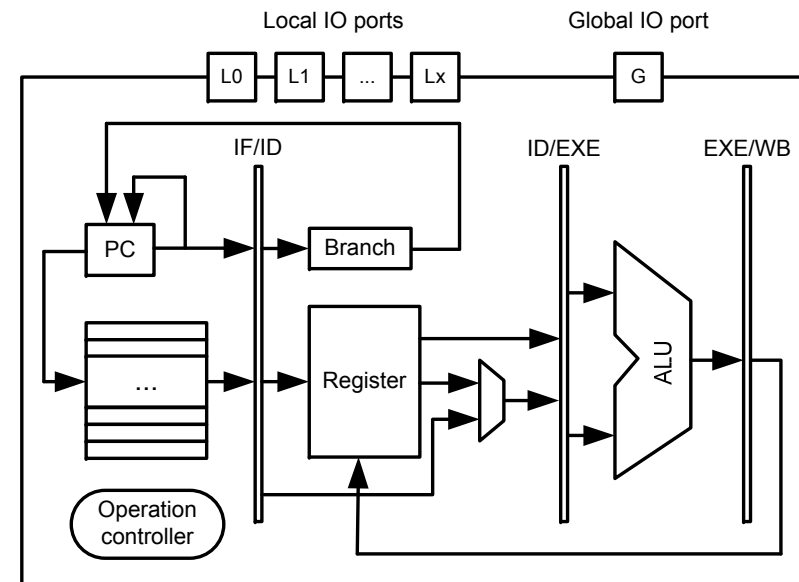
- Dedicated local interconnections:
  - High data throughput
- Hierarchical global routing network:
  - Flexible global data transmission
  - External data access
  - Global cell (re)configuration
- AMBA 4 AXI4-stream protocol
- Single-Cycle-Per-Hop latency
- Data driven synchronization
- GALS network data transmission



# Processing cell

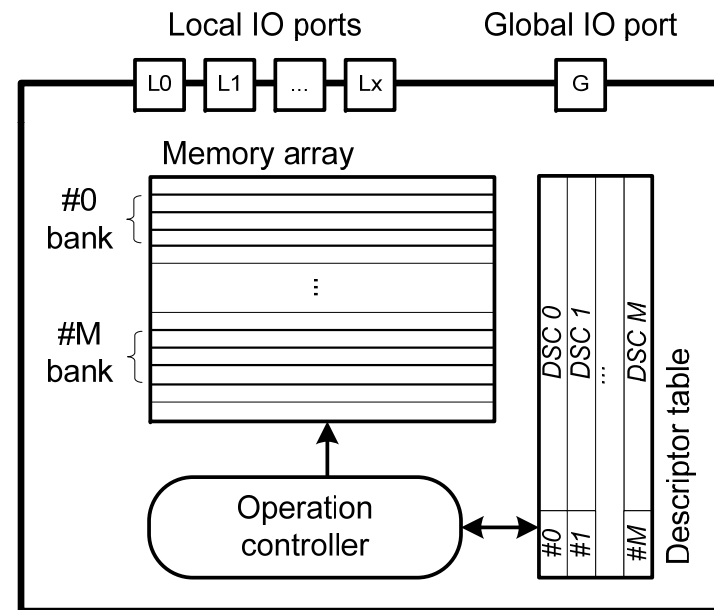


- Processing core
  - ALU, DSP, SIMD, VLIW, CORDIC...
  - Implicit load-store operations in all instructions.
  - Run-time control and conditional reconfiguration.
  - In-cell NoC supervision and reconfiguration.
- Processing shell
  - Network adapter

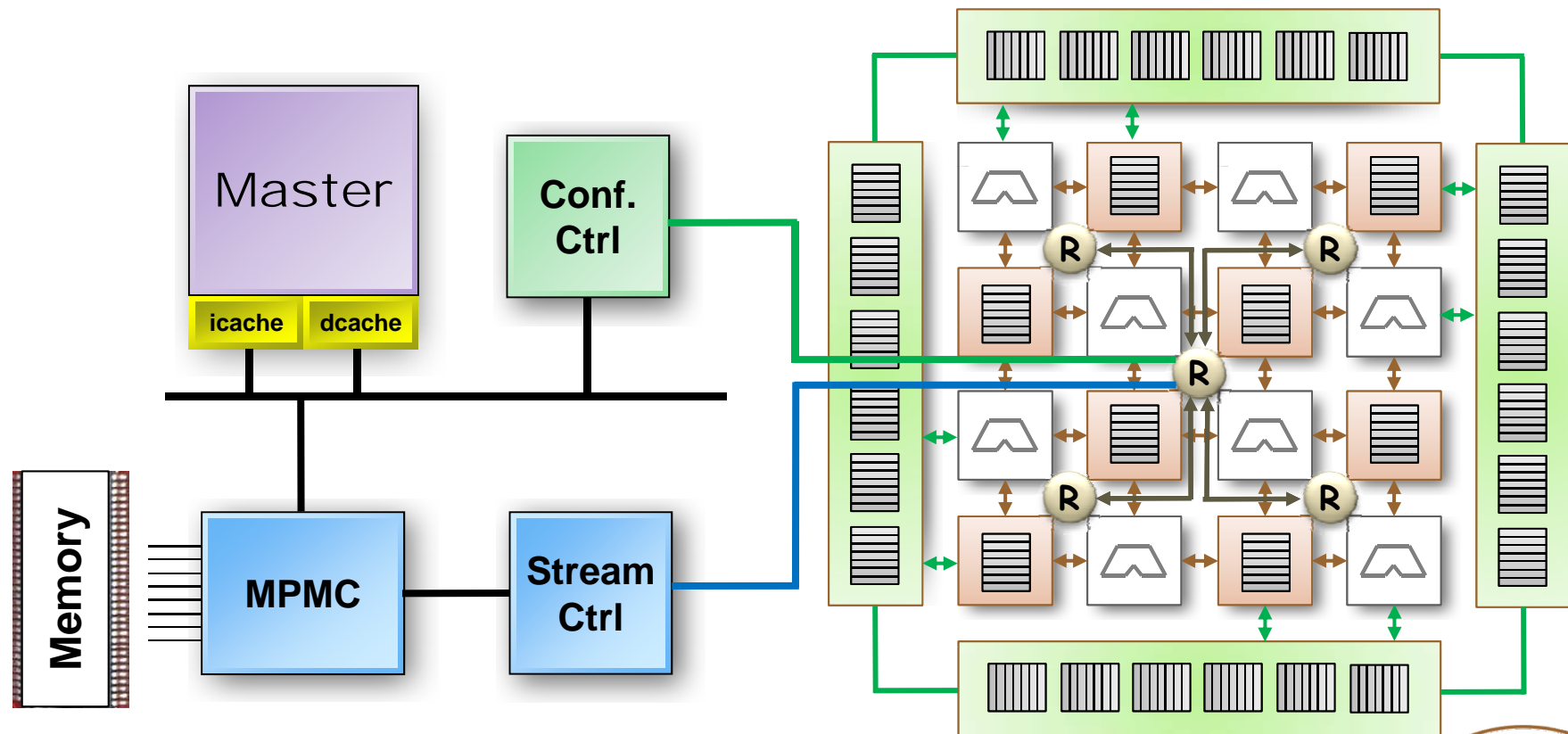


# Memory cell

- Cell structure:
  - Divided into banks
  - Configurations and operations handled locally
- Operation mode: FIFO & RAM
- Run-time memory cell concatenation in FIFO mode.
- Blocking/Non-blocking execution.
- Programmable descriptor execution.
- Micro-block memory access.

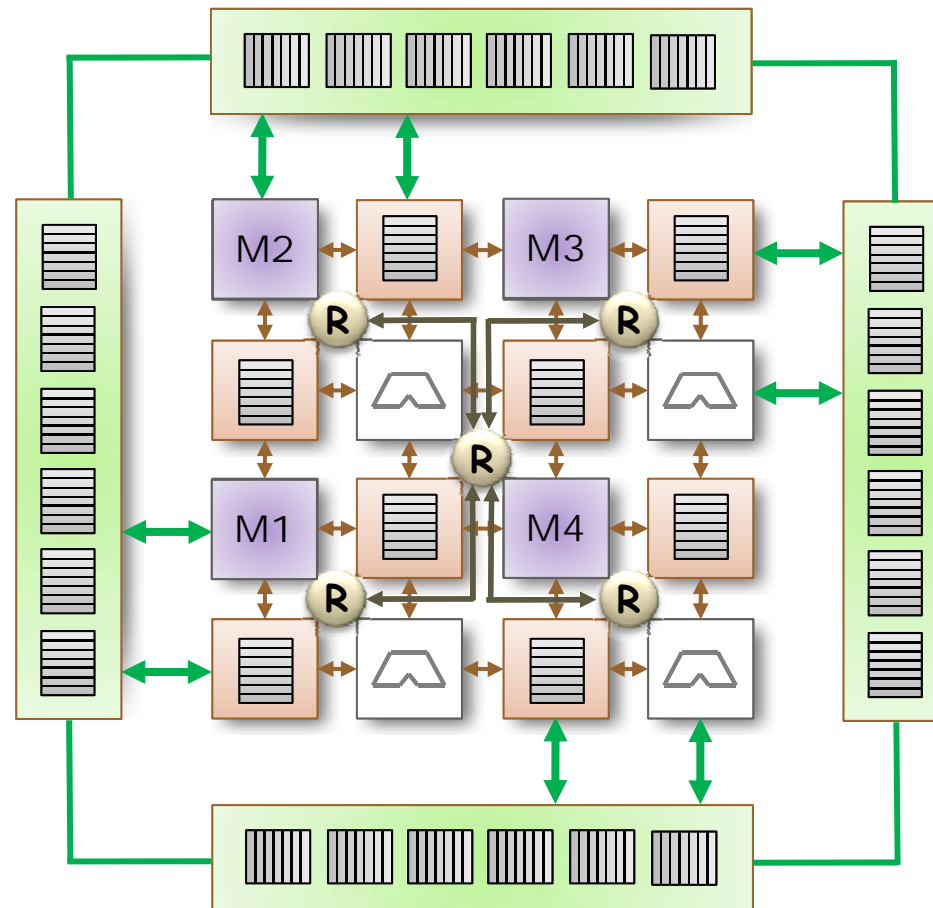


# Static & Dynamic configuration (I)



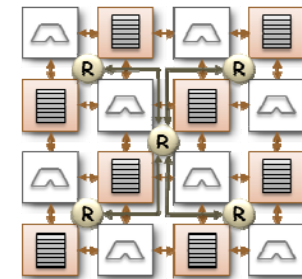
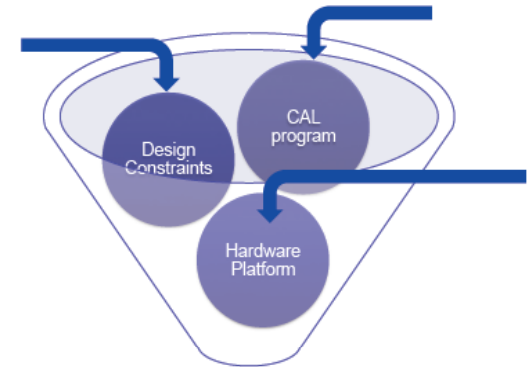


# Static & Dynamic configuration (II)

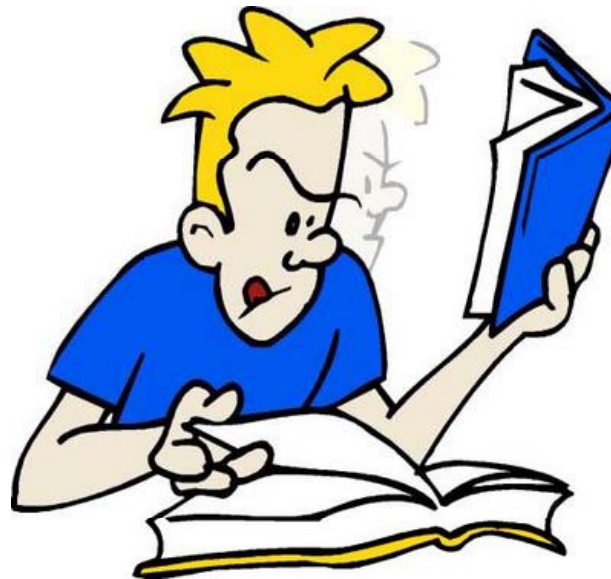
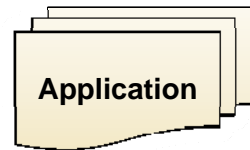


# Software development

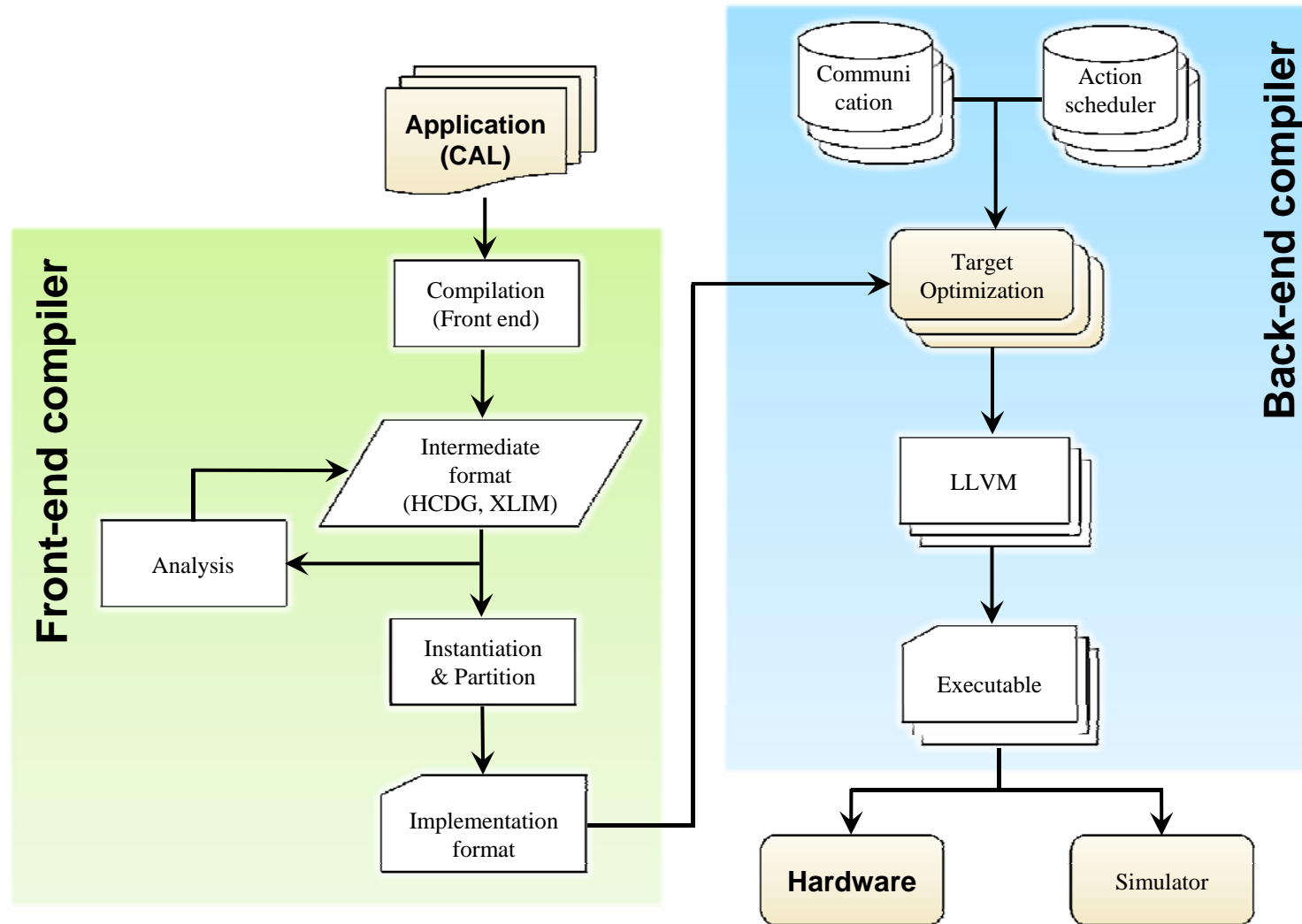
- Automated application mapping.
  - CAL dataflow language
  - Parallelism exploration
  - Within the HiPEC project together with CS dept.
- Design exploration with SCENIC framework.
  - SystemC-based virtual platform
  - Explore system behavior and evaluate performance



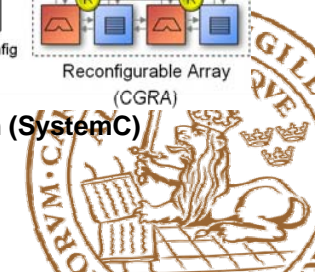
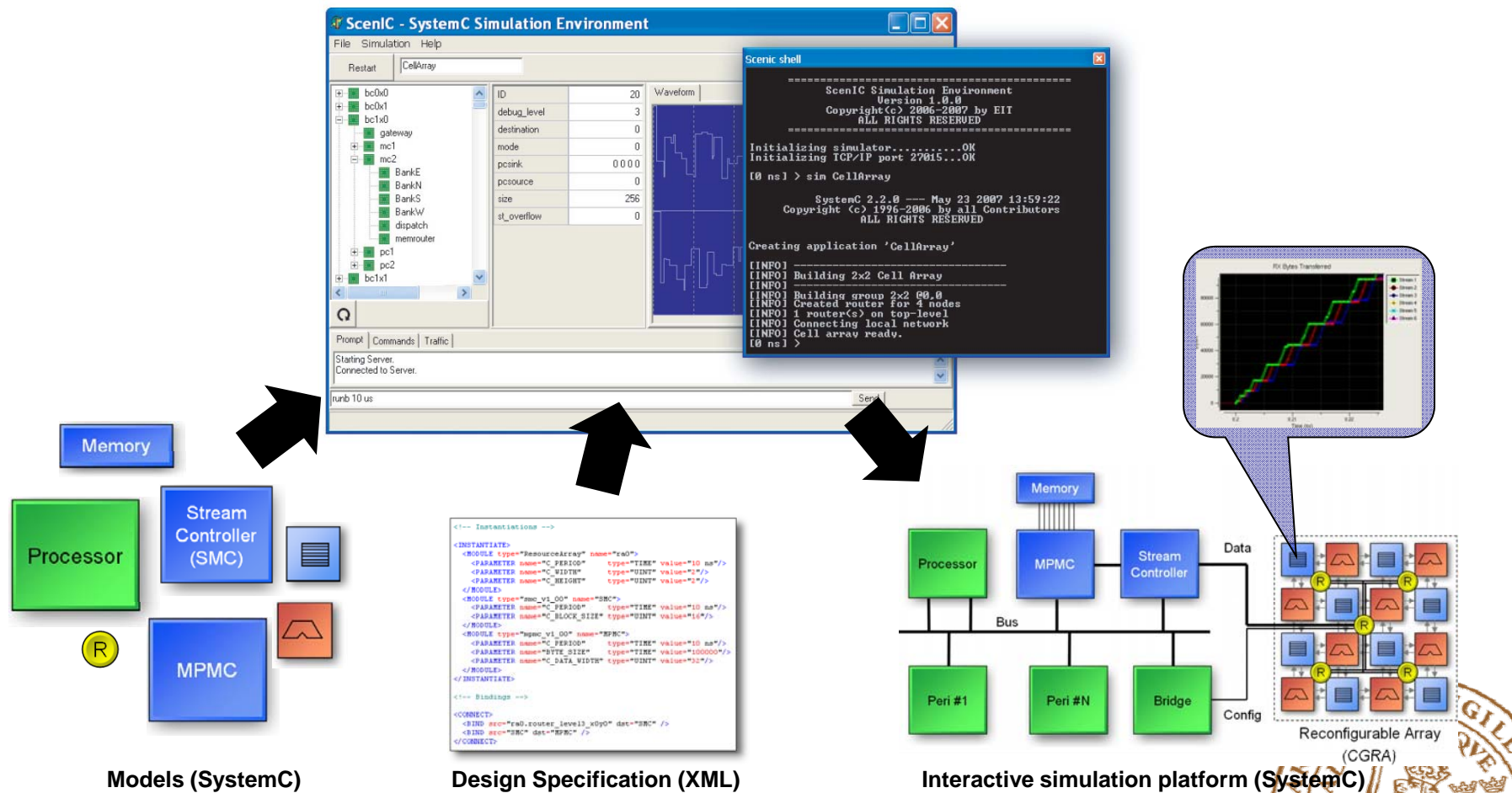
# Application mapping



# Application mapping from CAL



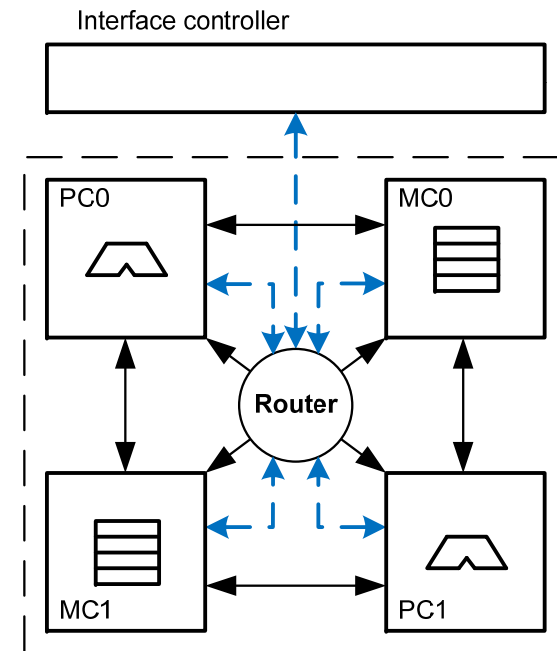
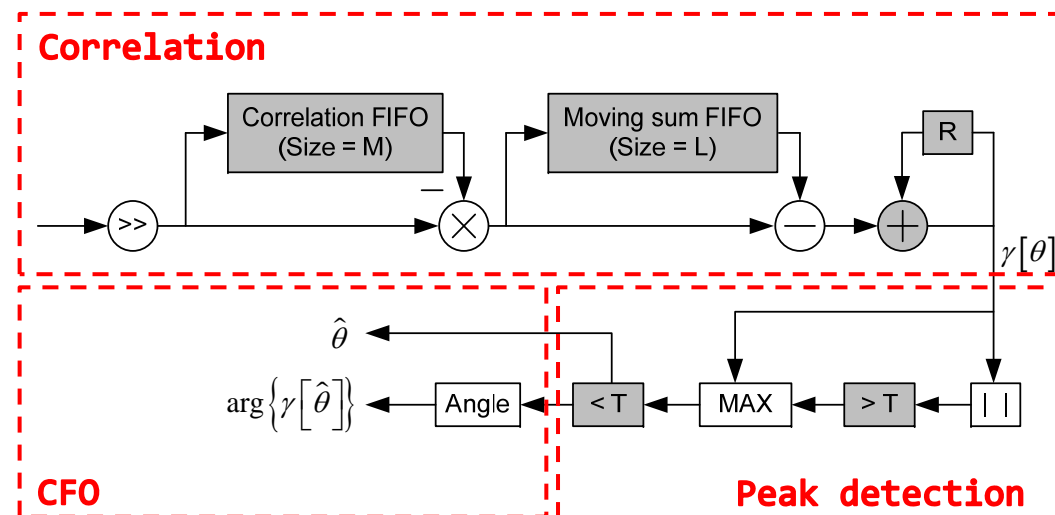
# SCENIC – Design Exploration



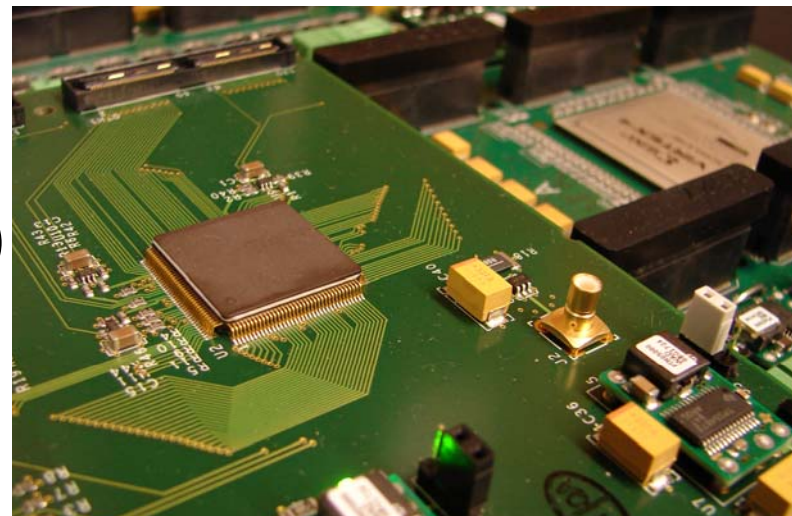
# Case study: multi-standard OFDM synchronization



- Multiple wireless radio standards
- Concurrent data stream processing
- Coarse Time Synchronization
- Carrier Frequency Offset (CFO) estimation



# Implementation results (I)



- 65 nm low-power regular VT CMOS:
  - Area: 0.479 mm<sup>2</sup>
  - Clock frequency: 534 MHz
- Adaptive word length scheduling.
- Adoption of different algorithms, e.g. Novel sign-bit OFDM acquisition.

Concurrency	Standard	Quantization accuracy	Memory utilization
Single-Stream	802.11n	4 bits	08.48%
	LTE	4 bits	65.18%
	DVB-H 2K	4 bits	85.71%
	DVB-H 4K	2 or Sign bit	85.71%
Dual-Stream	DVB-H 8K	Sign bit	85.71%
	802.11n & 802.11n	4 or 2 bits	16.96%
	802.11n & LTE	2 bits	45.09%
	802.11n & DVB-H 2K	2 bits	65.63%
	LTE & LTE	2 bits	73.21%
LTE & DVB-H 2K	2 bits	93.75%	





# Implementation results (II)

- Compare with ASIC solution, ~4 times more area cost.
- Case study does not explore the potential usage.
  - Simple algorithms
  - No task-level hardware sharing
- Currently looking at channel estimation & MIMO detection for LTE-A.
  - Task-level hardware sharing
  - Cooperation with the DARE project





# Configuration generator

The screenshot displays the Configuration Generator software interface, which is used for configuring hardware components. It consists of several windows:

- Processor Cell 0 Configuration:** A window for configuring the processor cell, including options for PC control, operation, target, condition, and control options.
- MULTI-BASE OVM Data Generator:** A window showing a grid of configuration targets for Processor Cells (PC 0, PC 1, PC 2) and Memory Cells (MC 0, MC 1, MC 2, MC 4).
- Memory Cell 0 Descriptor Configuration:** A window for configuring the memory cell descriptor, including options for DSC configuration, DSC execution, and DSC state reset.
- Configuration Memory Address Setup:** A window showing a table of memory addresses for various components.
- Configuration Generator:** A central window showing a list of configuration items and a configuration database.

Object name	Start address [Dec]	Start address [Hex]	End address [Dec]	End address [Hex]
PC0 PGM & Ctrl_Reg	0	0	1023	3FF
PC1 PGM & Ctrl_Reg	1024	400	2047	7FF
MCD0 Data Memory	2048	800	3071	BFF
MCI Data Memory	3072	C00	3839	EFF
MCD Descriptor	3840	F00	3875	F23
MCI Descriptor	3876	F24	3911	F47
MC2 Descriptor	3912	F48	3947	F6B
Reconfig_Reg[0]	3948	F6C	3948	F6C
Reconfig_Reg[1]	3949	F6D	3949	F6D
Reconfig_Reg[2]	3950	F6E	3950	F6E
Reconfig_Reg[3]	3951	F6F	3951	F6F
Reconfig_Reg[4]	3952	F70	3952	F70
Reconfig_Reg[5]	3953	F71	3953	F71
Reconfig_Reg[6]	3954	F72	3954	F72
Reconfig_Reg[7]	3955	F73	3955	F73
Reconfig_Reg[8]	3956	F74	3956	F74
Reconfig_Reg[9]	3957	F75	3957	F75
Reconfig_Reg[10]	3958	F76	3958	F76
Reconfig_Reg[11]	3959	F77	3959	F77
Reconfig_Reg[12]	3960	F78	3960	F78
Reconfig_Reg[13]	3961	F79	3961	F79



# Conclusion

- Reconfigurable cell array enables hardware sharing at different levels, i.e., task-, function-, and algorithm-level.
- *Coarse-grained reconfigurable cell array* comprises distributed processing and memory cells, and a *hierarchical* NoC structure.
- In-cell *dynamic reconfiguration* enables fast context switching.
- Application mapping from CAL and design exploration with SCENIC tool.
- The system flexibility is illustrated by performing OFDM synchronization for multiple standards.





# Thanks & Questions

---

