CHAIR FOR COMPILER CONSTRUCTION

cfaed CENTER FOR ADVANCING ELECTRONICS DRESDEN

Portable Libraries and Programming Environments

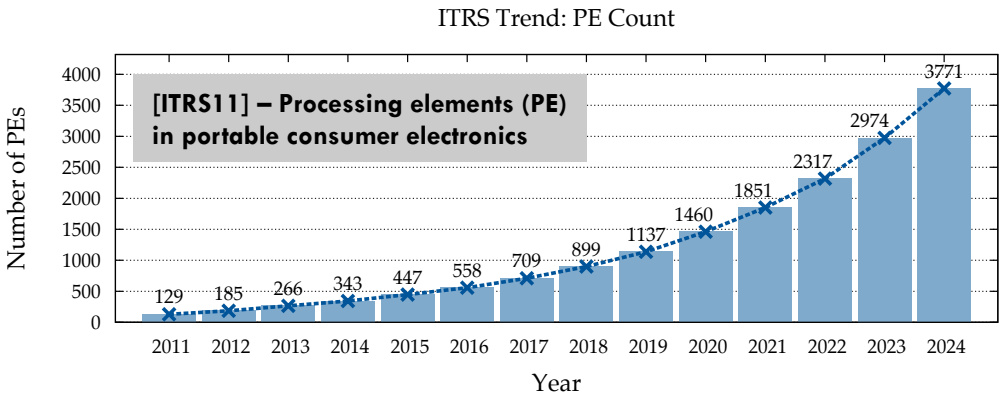
Jeronimo Castrillon
Chair for Compiler Construction
TU Dresden
jeronimo.castrillon@tu-dresden.de

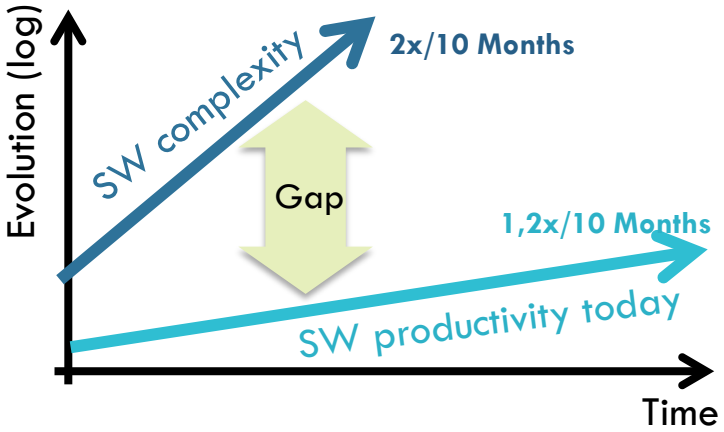
Thematic session: Towards Portable Libraries for Hybrid Systems
HiPEAC Computing Systems Week. Oslo, May 6th 2015

Acknowledgements

- German Cluster of Excellence: Center for Advancing Electronics Dresden (www.cfaed.tu-dresden.de)
- Silexica Software Solutions GmbH
- German Cluster of Excellence: Ultra High-Speed Mobile Information and Communication (UMIC), RWTH Aachen

Why new programming environments?



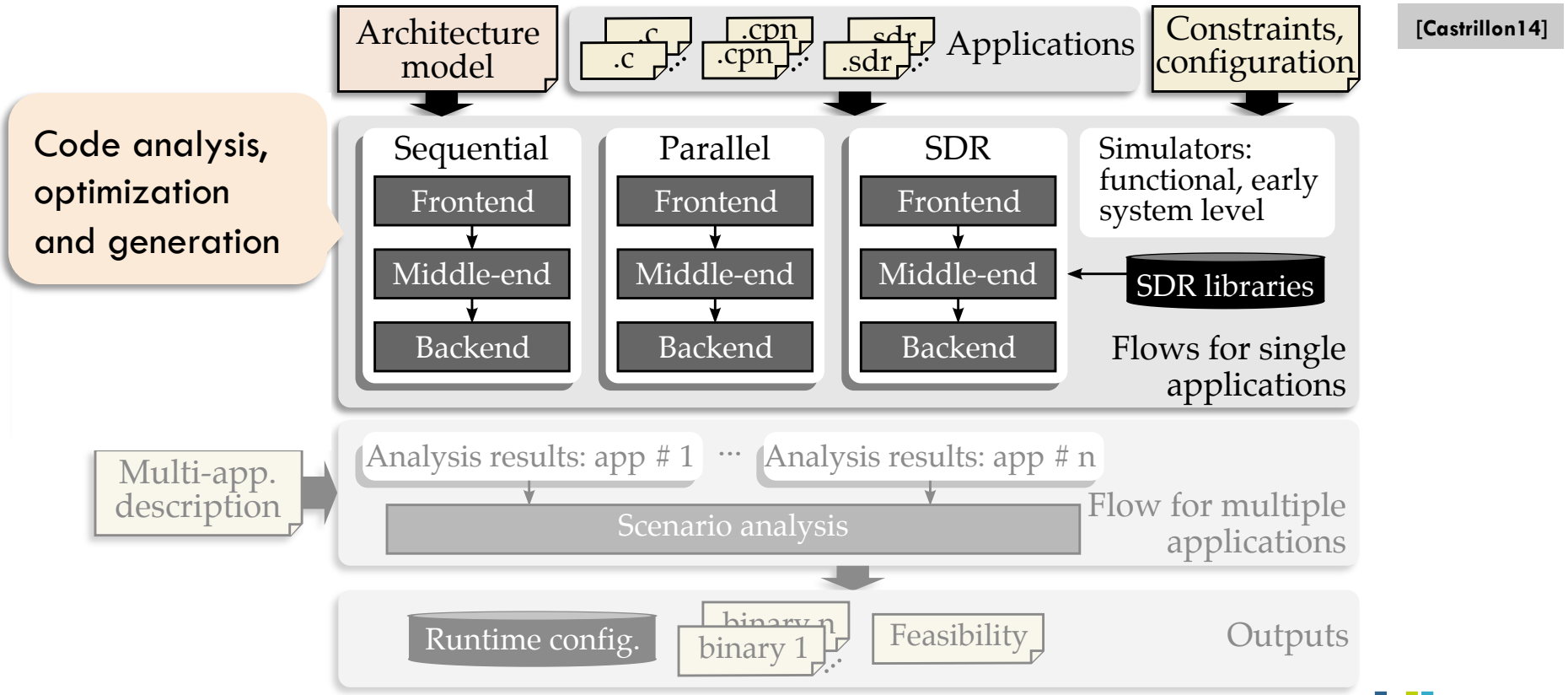


- Complex applications for complex architectures ➔ SW-productivity gap

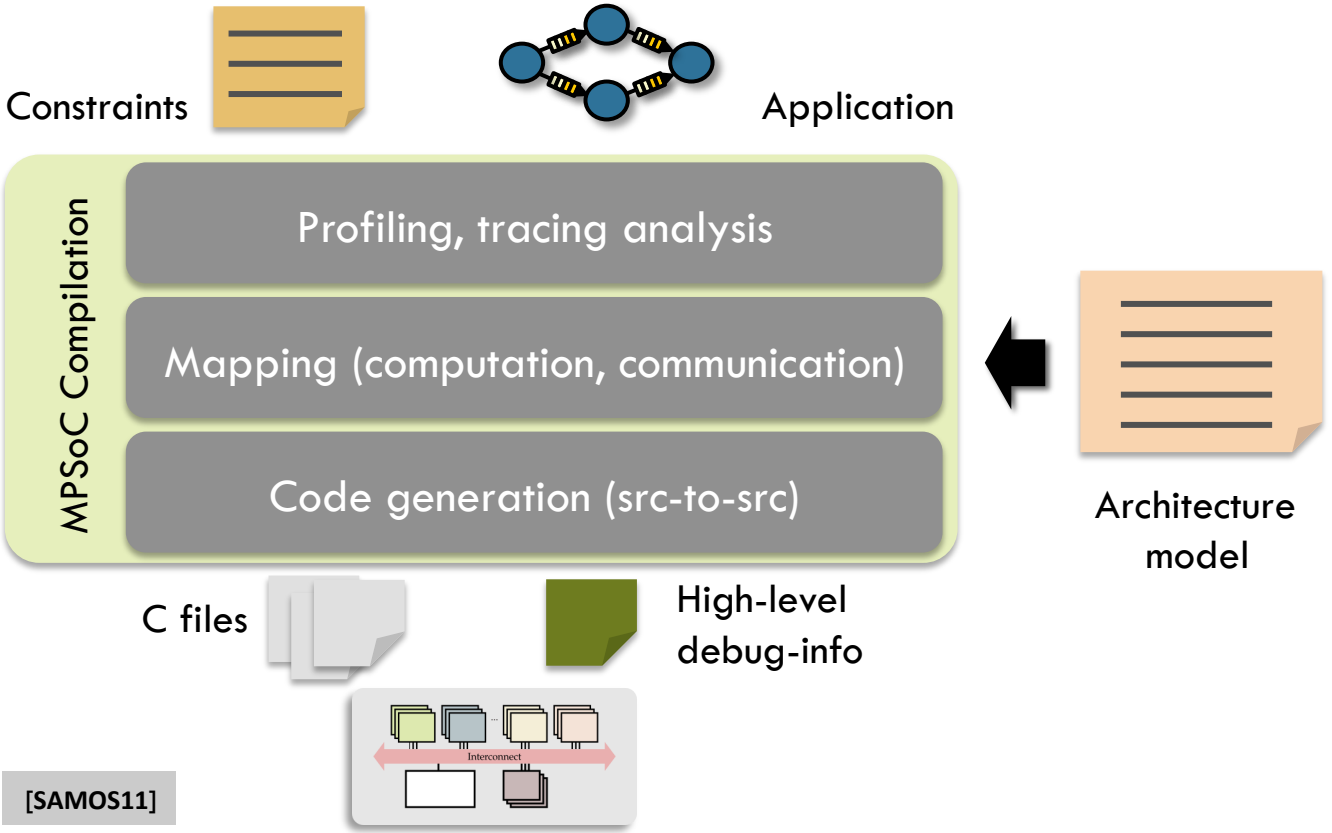
- Already difficult for heterogeneous multi-core (programmable) systems
- More difficult for systems with hardware accelerators and reconfigurable HW

Programming flows for heterogeneous MPSoCs

[Castrillon14]

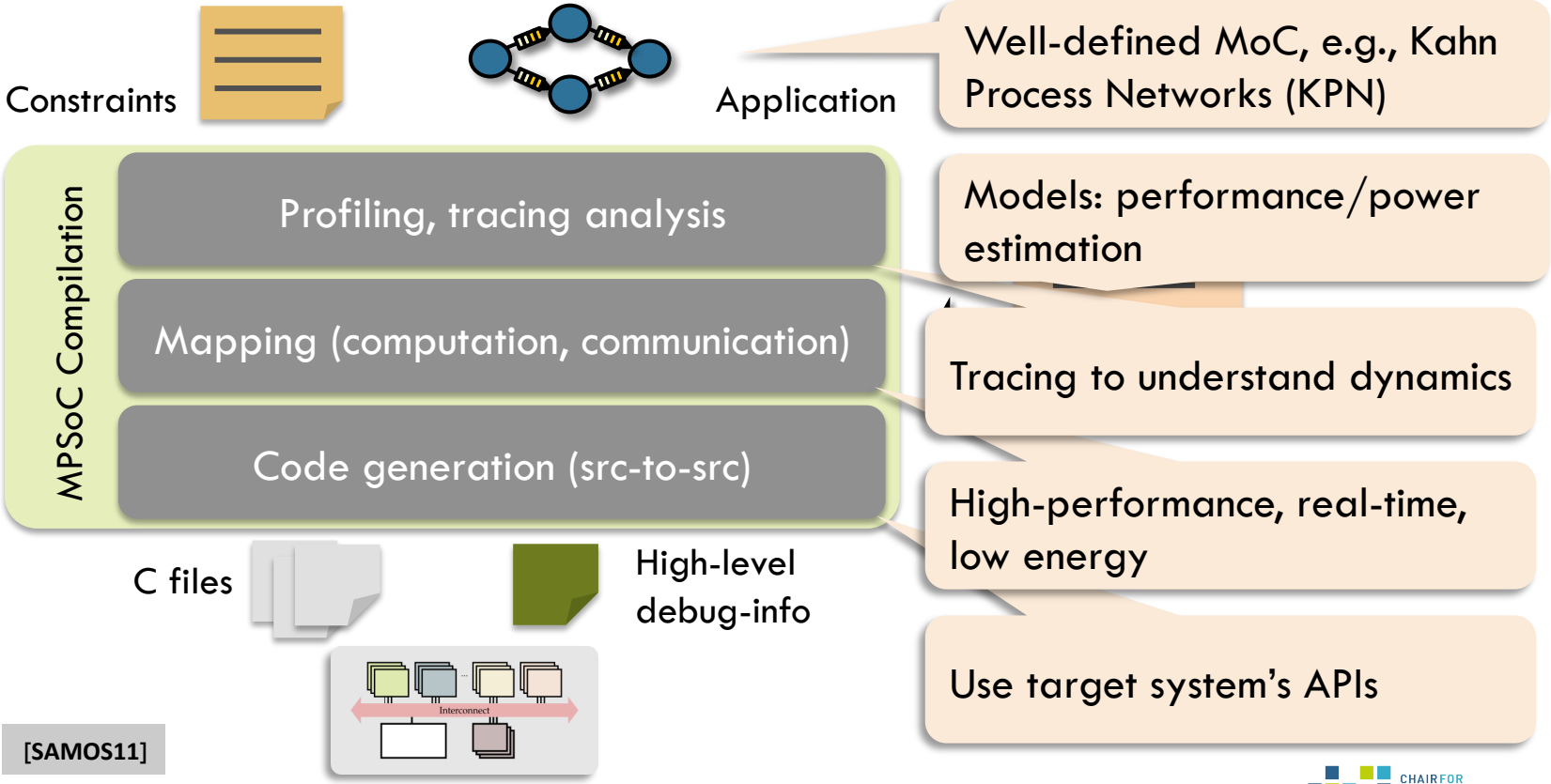


Parallel programming flow: Overview



[SAMOS11]

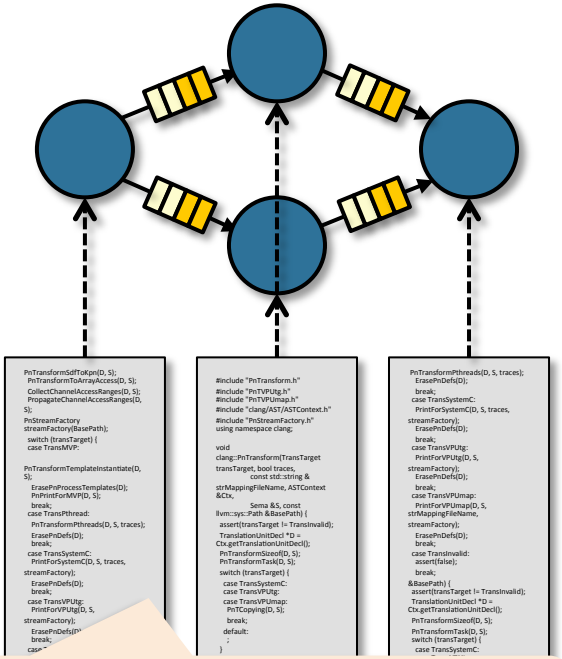
Parallel programming flow: Overview



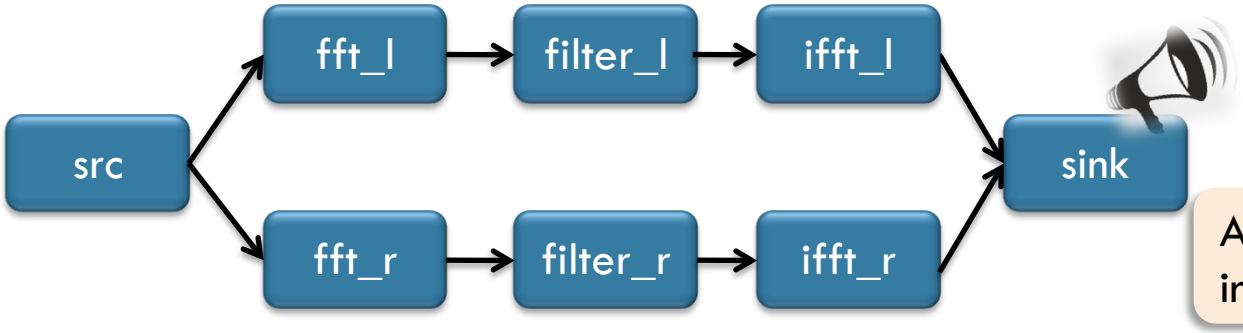
[SAMOS11]

Application: Dataflow and process networks

- Graph representation of applications
  - Process **communicate only** over FIFO buffers
  - Good model for streaming applications
  - Good match for signal processing & multi-media



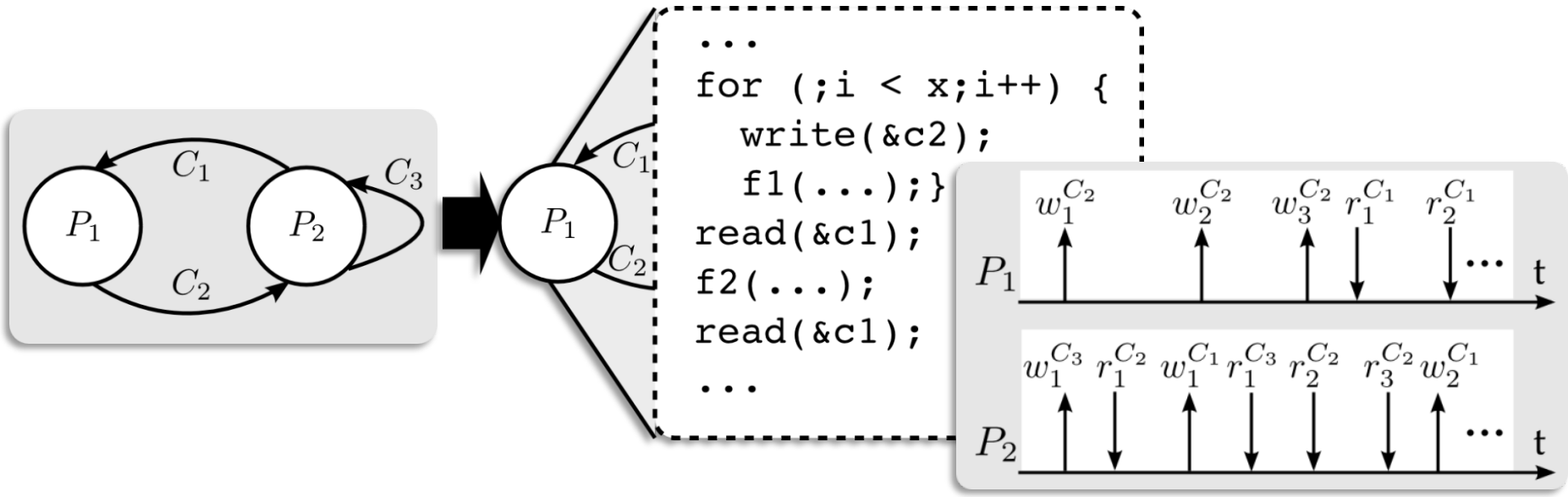
- Stereo digital audio filter



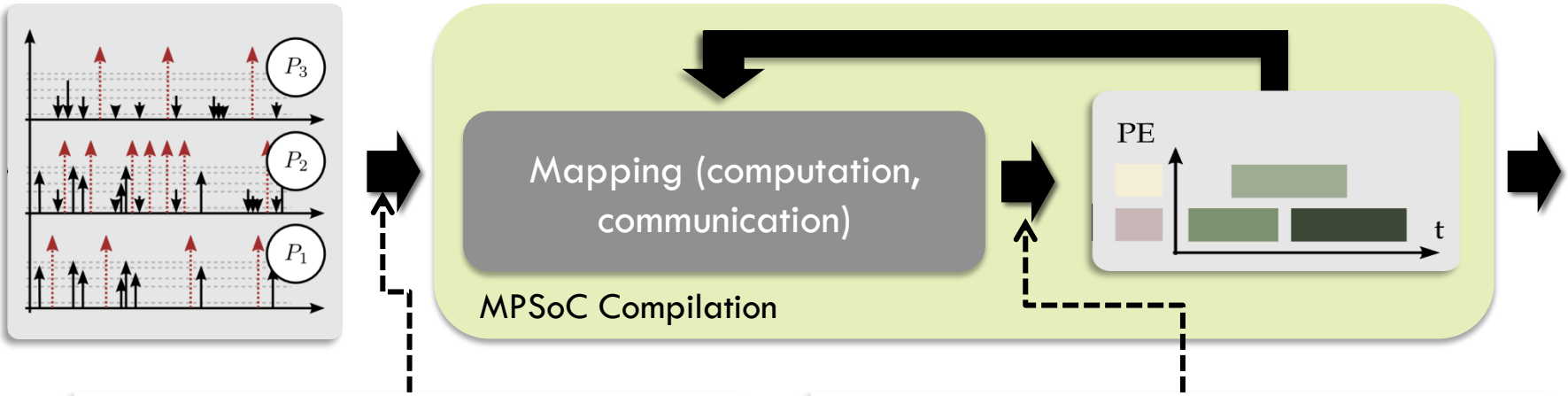
Allows control decision upon incoming data

Analysis: Instrumentation, profiling and tracing

- **Process white model**: source code analysis and tracing to deal with control



Performance models: Sequential and parallel



Sequential performance estimation

Design

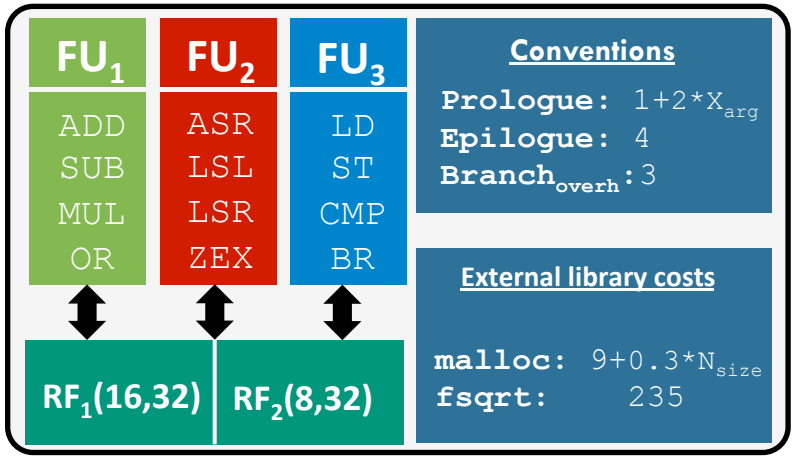
- Annotations & cost functions
- Abstract operation cost models
- Processor models/simulators
- Measurements

Parallel performance estimation

- Abstract cost models: OS, multi-tasking APIs, interconnect & memories
- System simulators/emulators
- Boards

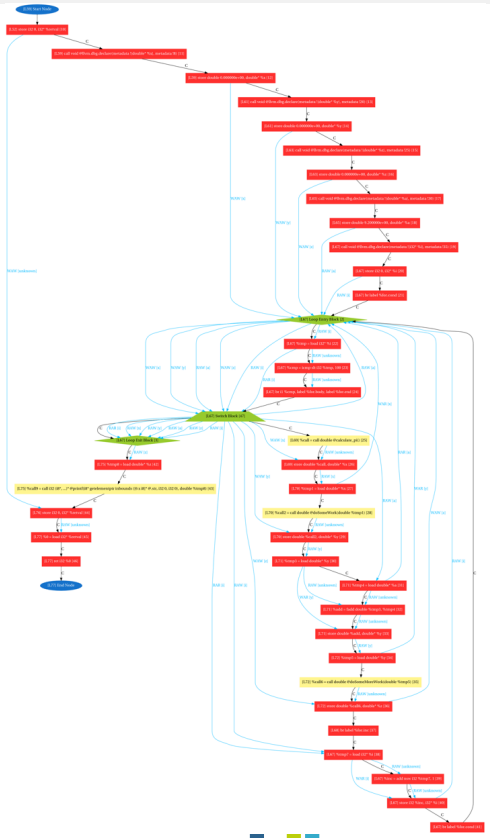
Performance estimation: Computation

- Cost tables
  - Equivalence: Low-level IR → Assembly instructions
  - Coarse estimation of instruction-level parallelism
- Processor + compiler models
  - Set of resources, operations, low-level APIs



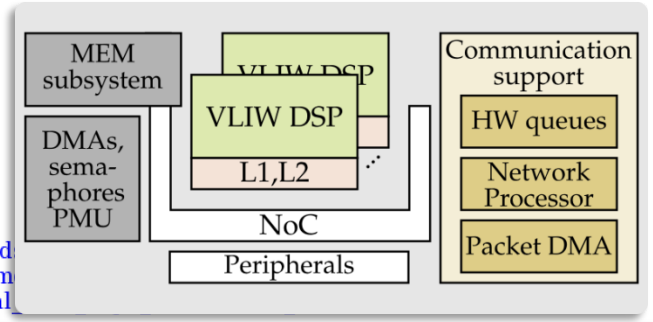
[SAMOS14]

Courtesy: J. Esse

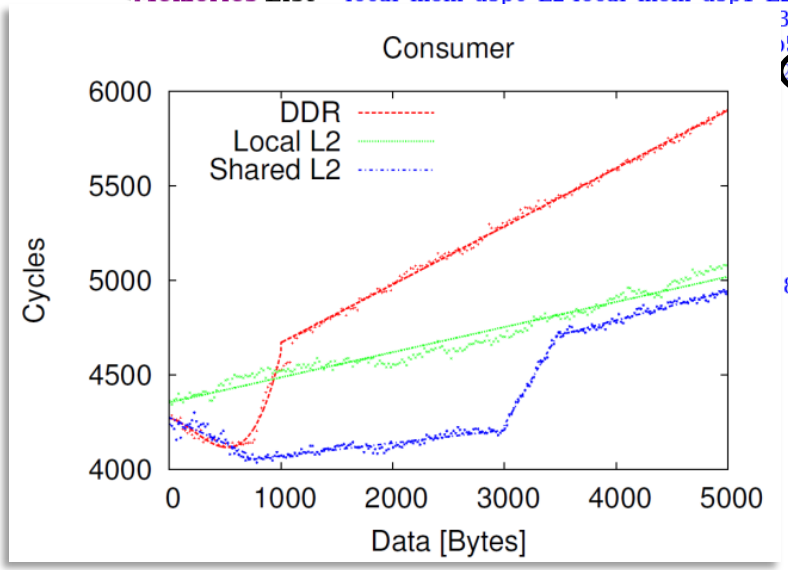


Performance estimation: Communication

- Characterization of different communication APIs
- Example: TI Keystone II



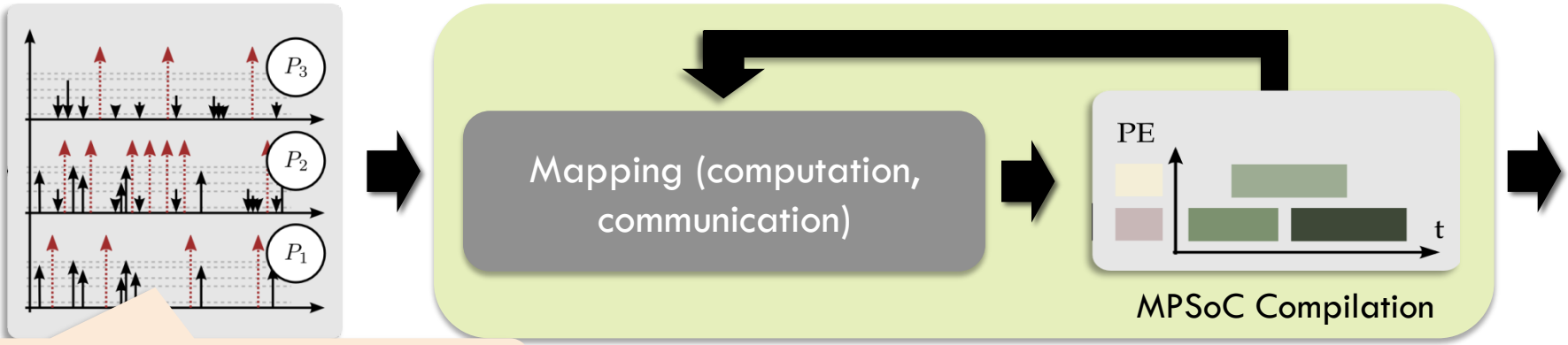

```
−<Platform>
  <Processors List="dsp0 dsp1 dsp2 dsp3 dsp4 dsp5 dsp6 dsp7"/>
  <Memories List="local_mem_dsp0_L2 local_mem_dsp1_L2 local_mem_dsp2_L2 local_mem_d
```




```
−<CommPrimitive>
 −<CPDMA Name="EDMA3_DDR">
    <Description>EDMA over DDR</Description>
   −<Costs>
      <Cost End="800" Function="11442.60163-0.15775*x"/>
      <Cost Start="801" Function="11204.94186+0.316143*x"/>
    </Costs>
    <DMAs List="local_mem_dsp0_DDR local_mem_dsp1_DDR local_
  </CPDMA>
</CommPrimitive>
```

[SOC13]

Application mapping: Trace-based

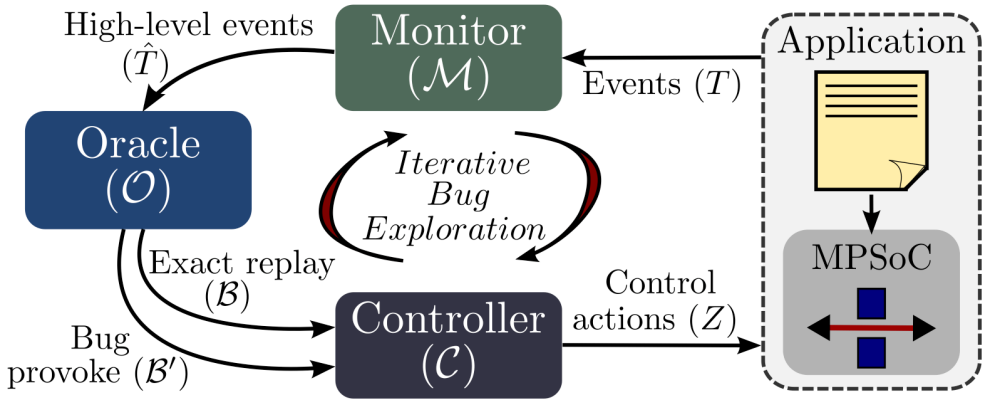


- Heuristics that work on a graph-representation of multiple-traces
  - For buffer sizing
  - For mapping and scheduling
  - Often iterative for real-time

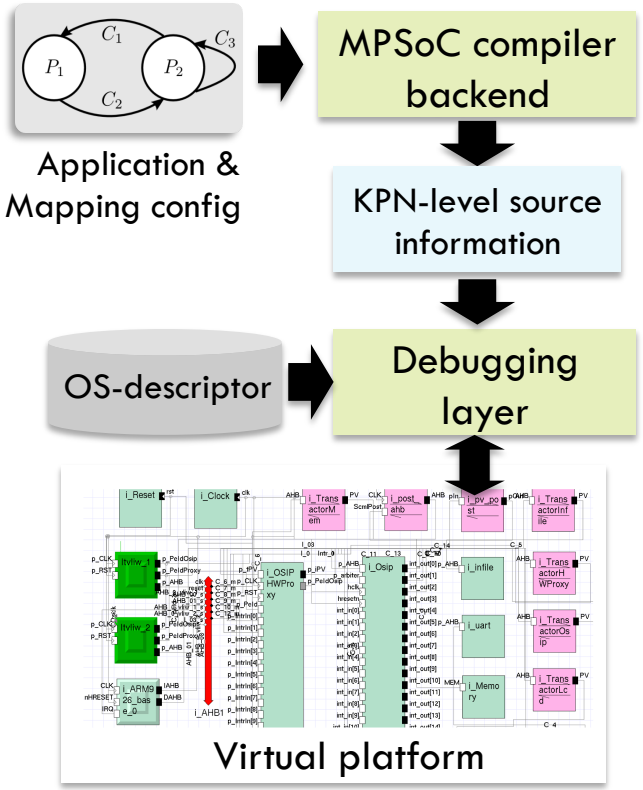
[DATE10, DAC12, IEEE-TII13]

Debugging: Layers and scripting for virtual platforms

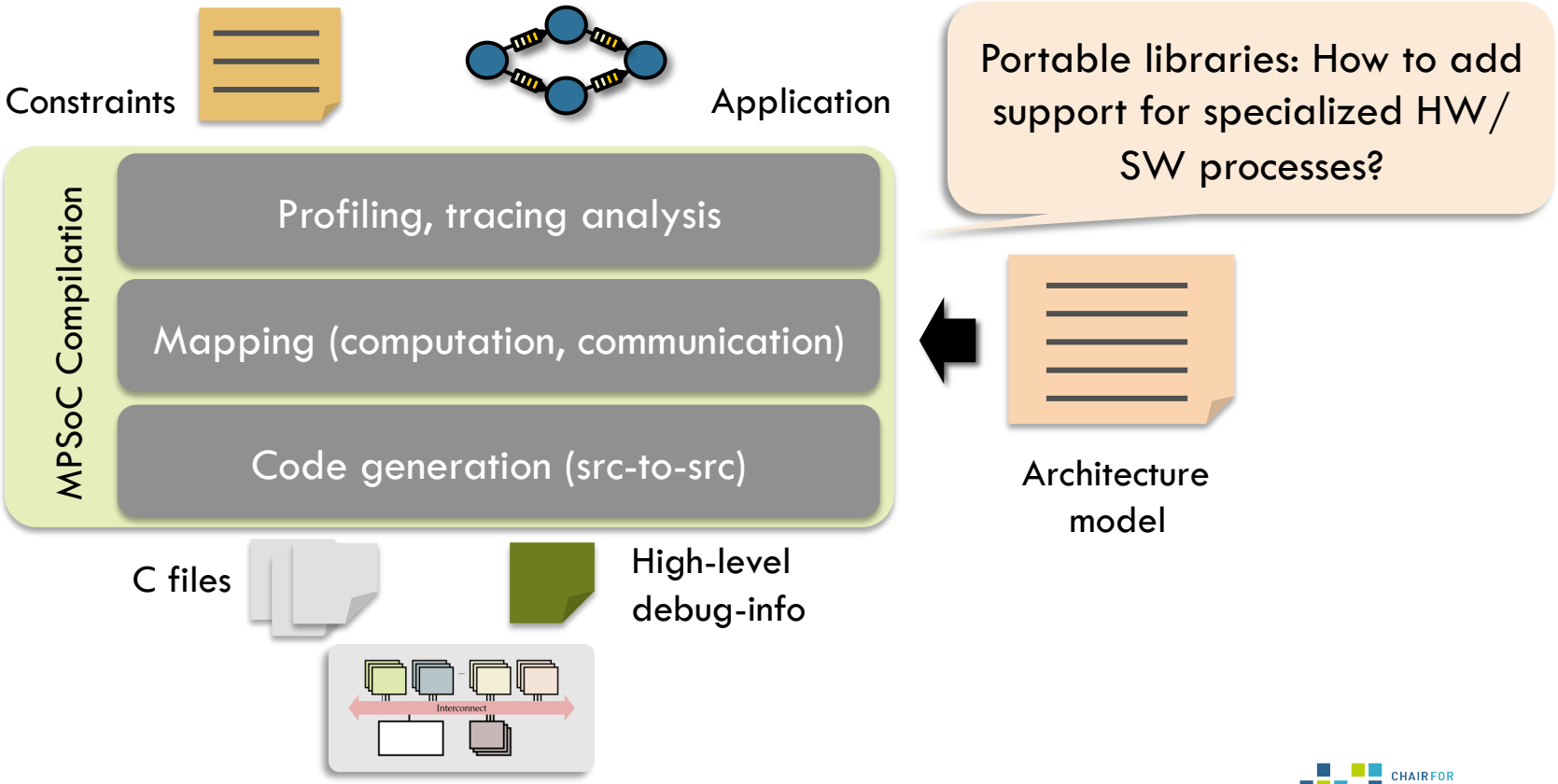
- Interactive debugging
- Deterministic replay for bug exploration



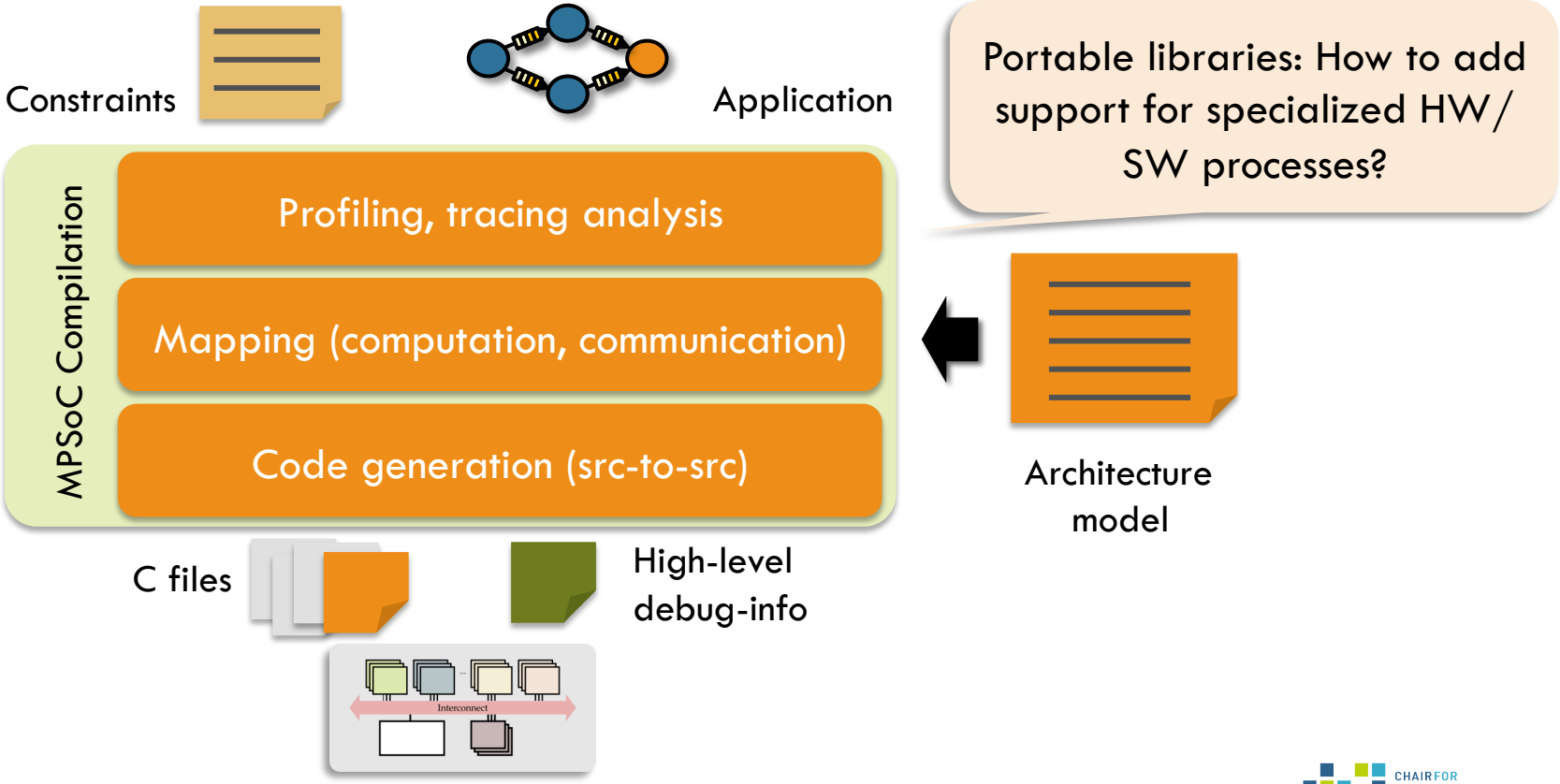
[LASCAS10, DATE14]



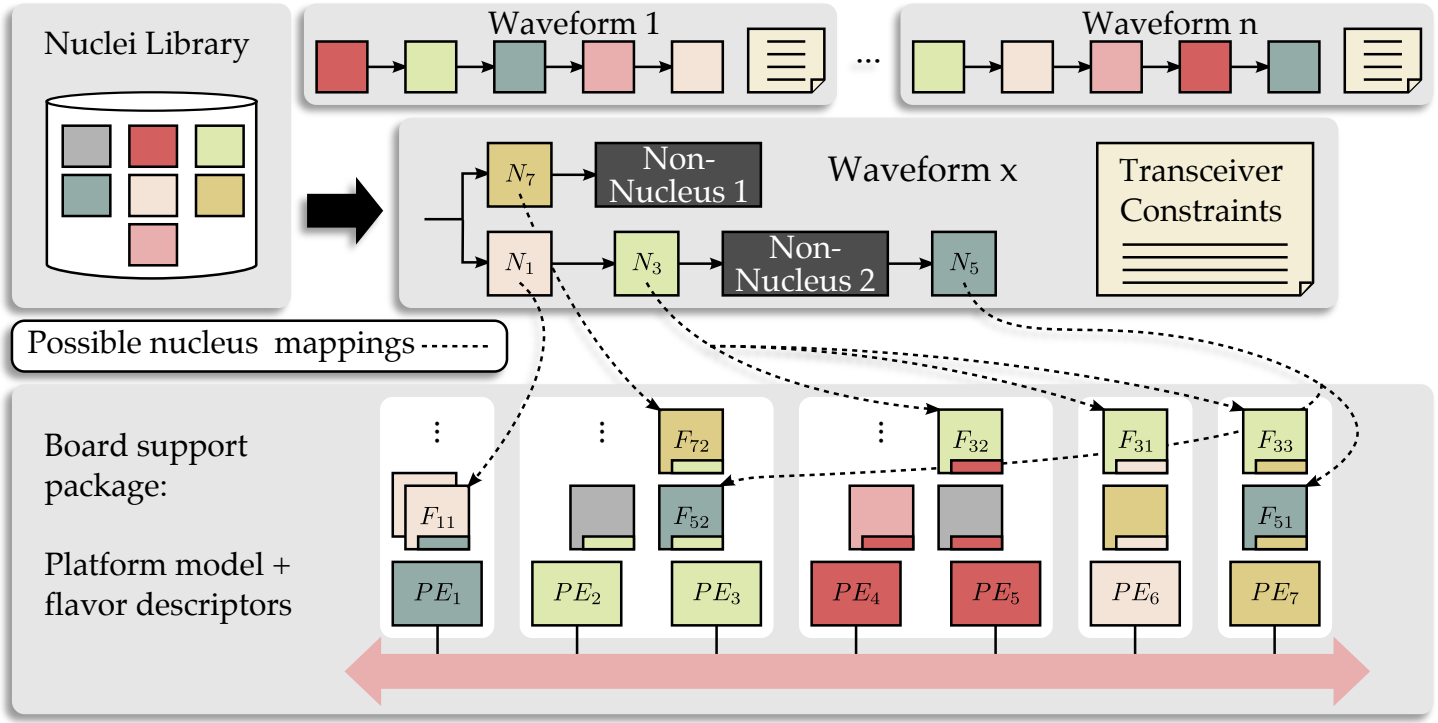
Parallel programming flow: Portable libraries



Parallel programming flow: Portable libraries

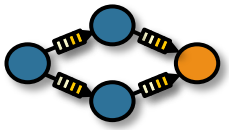


Solution approach: Nucleus project (for SW-defined Radio)



[SDR10, ALOG11,Castrillon14]

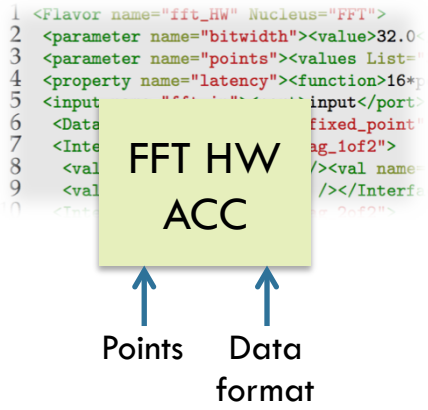
Nucleus programming flow: Inputs



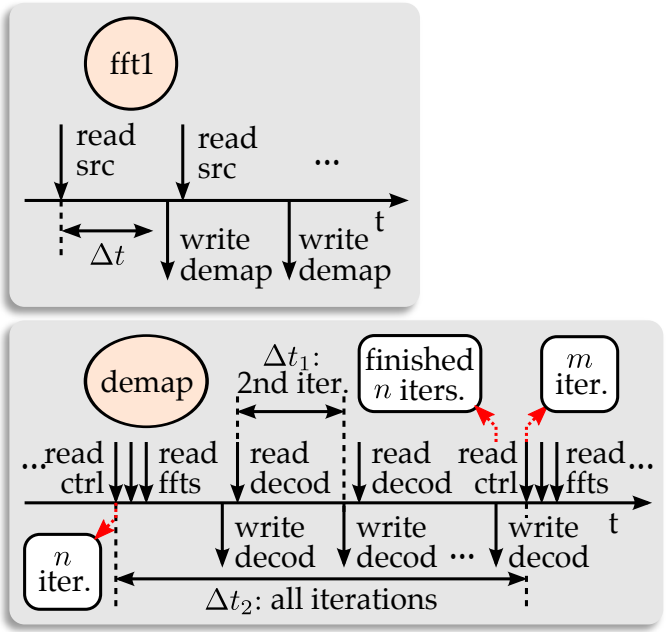
- Extended application specification
  - Selected processes are algorithmic kernels with **algorithmic parameters**
- Extended platform model
  - SW/HW accelerated kernels and their **implementation parameters**

Nucleus programming flow: Mapping

- Extended mapping
  - Find matching HW/SW support: Algorithmic to implementation parameters
  - Integrate into trace-based framework

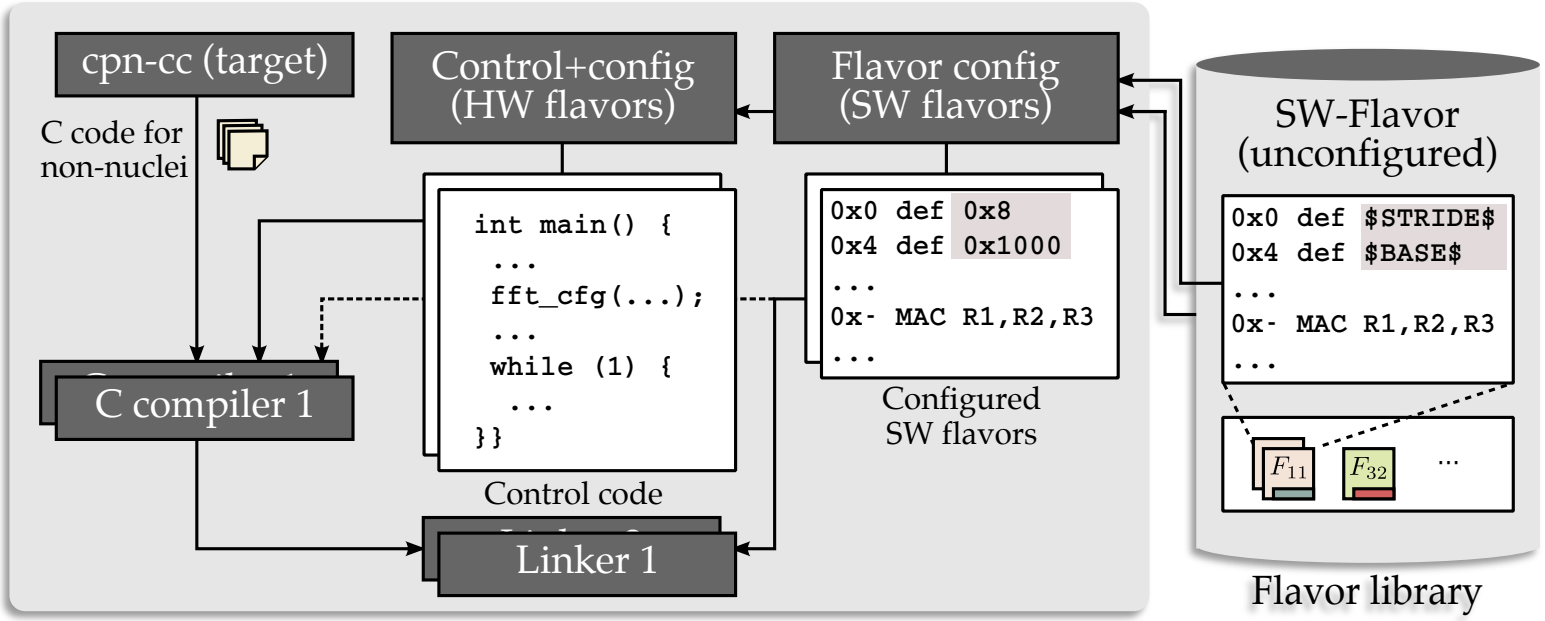


Configuration, characterization



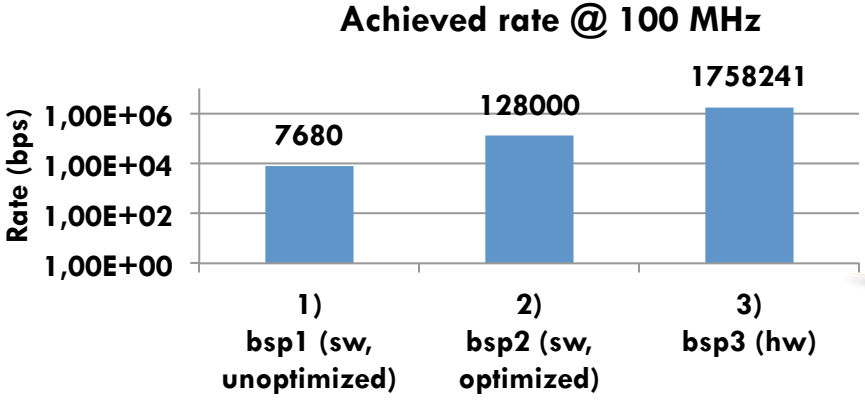
Nucleus programming flow: Code generation

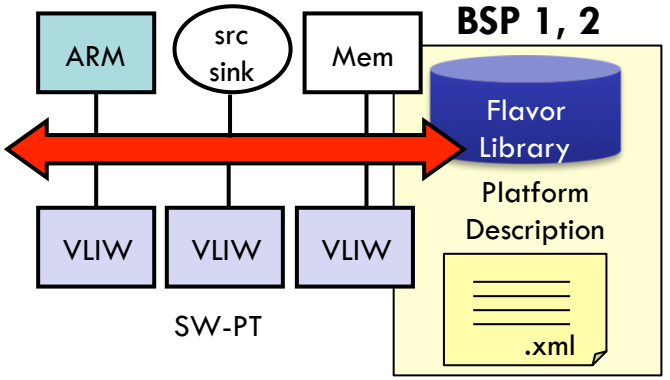
- Bare-metal implementation

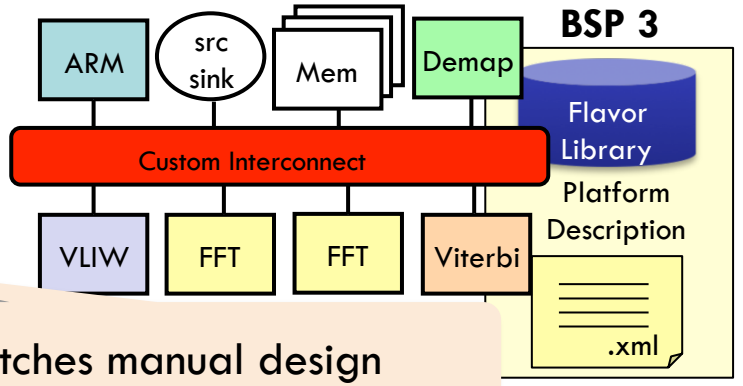


Case study (brief): Portable performance

- Application: MIMO OFDM receiver
- Hardware
  - Platform 1: Baseline software
  - Platform 2: Optimized software
  - Platform 3: Optimized SW + HW







Matches manual design

Discussion towards library portability

- Programming flow extension to deal with HW/SW acceleration
- Desired library characterization (standards?)
  - For mapping
    - Times, data-types, rates (and more complex behaviors – time diagrams?)
  - For debugging
    - Debugging interfaces and true integration (so far: bare metal)
    - Interfacing patterns to be proven by trace analysis
  - For code generation (so far: bare metal)
    - OS integration, interaction with resource manager
- Not addressed: Reconfiguration & synthesis

References

[**ITRS11**] International Technology Roadmap for Semiconductors (ITRS), "System Drivers," 2011. [Online]. Available: http://www.itrs.net/

[**Castrillon14**] J. Castrillon and R. Leupers, Programming Heterogeneous MPSoCs: Tool Flows to Close the Software Productivity Gap. Springer, 2014

[**SAMOS11**] J. Castrillon, W. Sheng and R. Leupers, "Trends in embedded software synthesis" *Embedded Computer Systems (SAMOS), 2011 International Conference on,* 2011, 347-354

[**SAMOS14**] J.F. Eusse, C. Williams, L.G. Murillo, R. Leupers and G. Ascheid, G., "Pre-architectural performance estimation for ASIP design based on abstract processor models," *Embedded Computer Systems: Architectures, Modeling, and Simulation (SAMOS XIV), 2014* pp.133-140, 2014

[**DATE10**] J. Castrillon, R. Velasquez, A. Stulova, W. Sheng, J. Ceng, R. Leupers, G. Ascheid, H. Meyr, "Trace-based KPN composability analysis for mapping simultaneous applications to MPSoC platforms", Proceedings of the Conference on Design, Automation and Test in Europe, pp. 753-758, 2010

[**DAC12**] J. Castrillon, A. Tretter, R. Leupers, and G. Ascheid, "Communication-aware mapping of KPN applications onto heterogeneous MPSoCs," in Proceedings of the Design Automation Conference, 2012

[**IEEE-TII13**] J. Castrillon, R. Leupers, and G. Ascheid, "MAPS: Mapping concurrent dataflow applications to heterogeneous MPSoCs," IEEE Transactions on Industrial Informatics, vol. 9, no. 1, pp. 527–545, 2013

[**SOC13**] M. Odendahl, J. Castrillon, V. Volevach, R. Leupers and G. Ascheid, "Split-cost communication model for improved MPSoC application mapping", In International Symposium on System on Chip (SoC) pp. 1-8, 2013

[**SDR10**] J. Castrillon, S. Schürmans, A. Stulova, W. Sheng, T. Kempf, R. Leupers, G. Ascheid, and H. Meyr, "Component-based waveform development: The nucleus tool flow for efficient and portable SDR," Wireless Innovation Conference and Product Exposition (SDR), 2010

[**ALOG11**] J. Castrillon, S. Schürmans, A. Stulova, W. Sheng, T. Kempf, R. Leupers, G. Ascheid, and H. Meyr, "Component-based waveform development: The nucleus tool flow for efficient and portable software defined radio", Analog Integrated Circuits and Signal Processing, vol. 69, no. 2–3, pp. 173–190, 2011

[**LASCAS14**] J. Castrillon, A. Shah, L. G. Murillo, R. Leupers, G. Ascheid, "Backend for virtual platforms with hardware scheduler in the MAPS framework", Latin American Symposium on Circuits and Systems (LASCAS), pp. 1-4, 2011

[**DATE14**] L. G. Murillo, S. Wawroschek, J. Castrillon, R. Leupers, G. Ascheid, "Automatic detection of concurrency bugs through event ordering constraints", Design, Automation and Test in Europe Conference and Exhibition (DATE'14), pp. 1-6, 2014