FZI

Forschungszentrum Informatik

an der Universität Karlsruhe

Microelectronic System Design (SIM)

Control-Flow Aware Communication and Conflict Analysis of Parallel Processes

Axel Siebenborn, Alexander Viehl, Oliver Bringmann

{siebenborn|viehl|bringmann}@fzi.de

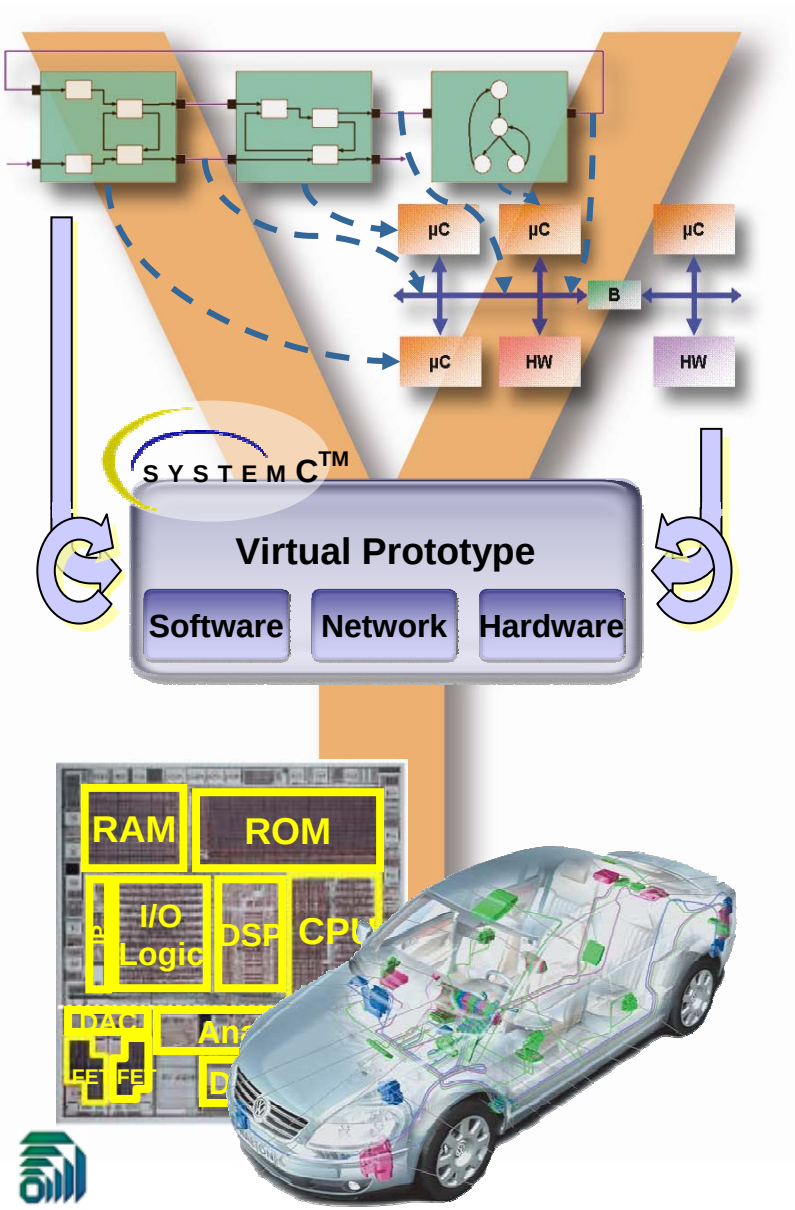
Wolfgang Rosenstiel

rosenstiel@informatik.uni-tuebingen.de

Outline

- System evaluation in platform based design
- Inclusion of temporal system environment
- Communication and performance analysis
- Conflict analysis and allocation of communication resources
- Case Study: Viterbi decoder

Introduction

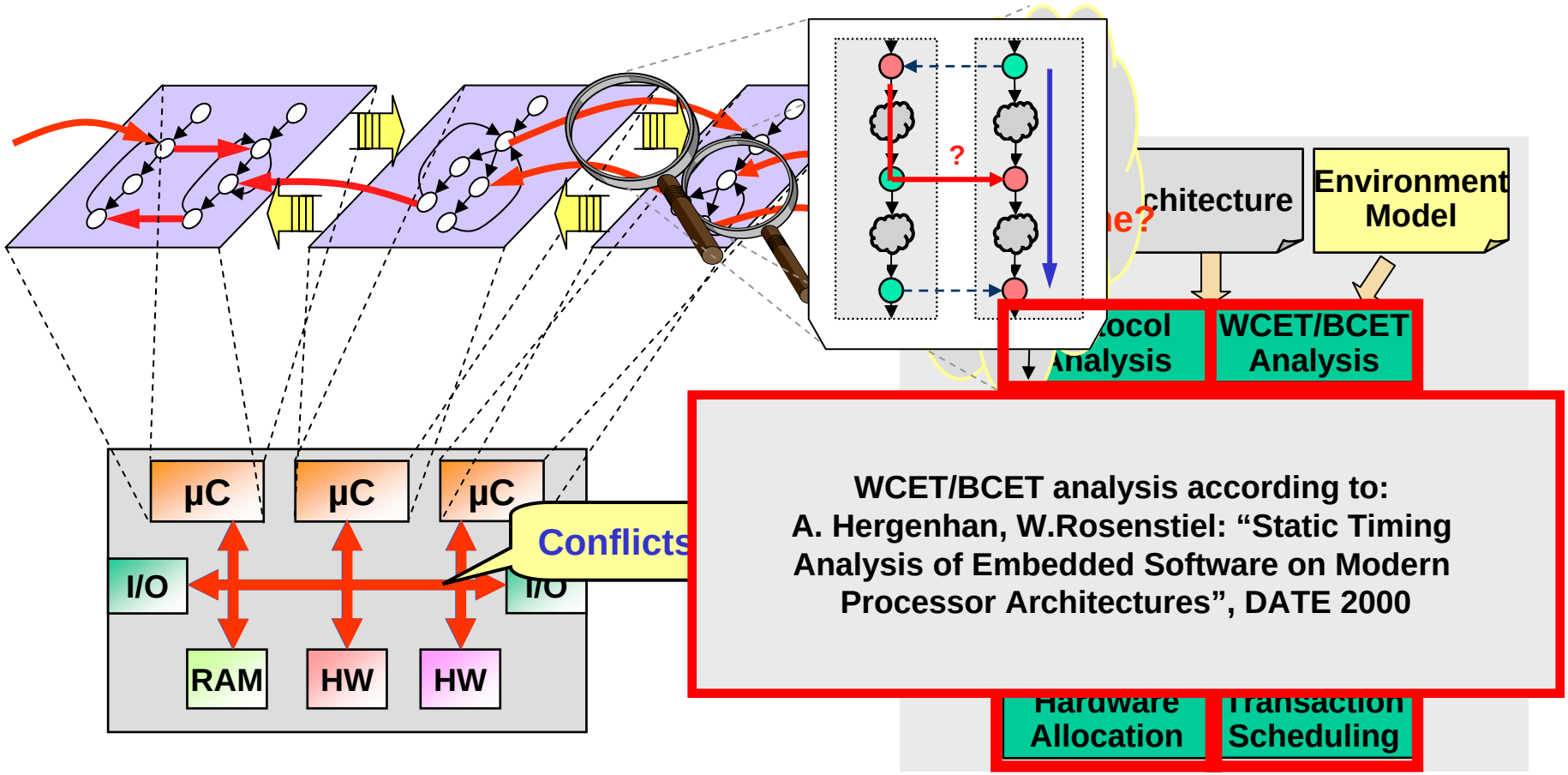


- Platform based design to handle complexity
- Early formalized specification
- Executable for functional validation
- Determination of NFP of a system design
- Early design space exploration with respect to functional and architectural specification
- Reduction of design cycles by including architectural specification at the beginning of the design flow
- Optimization of underlying architecture

Related Work – System Evaluation

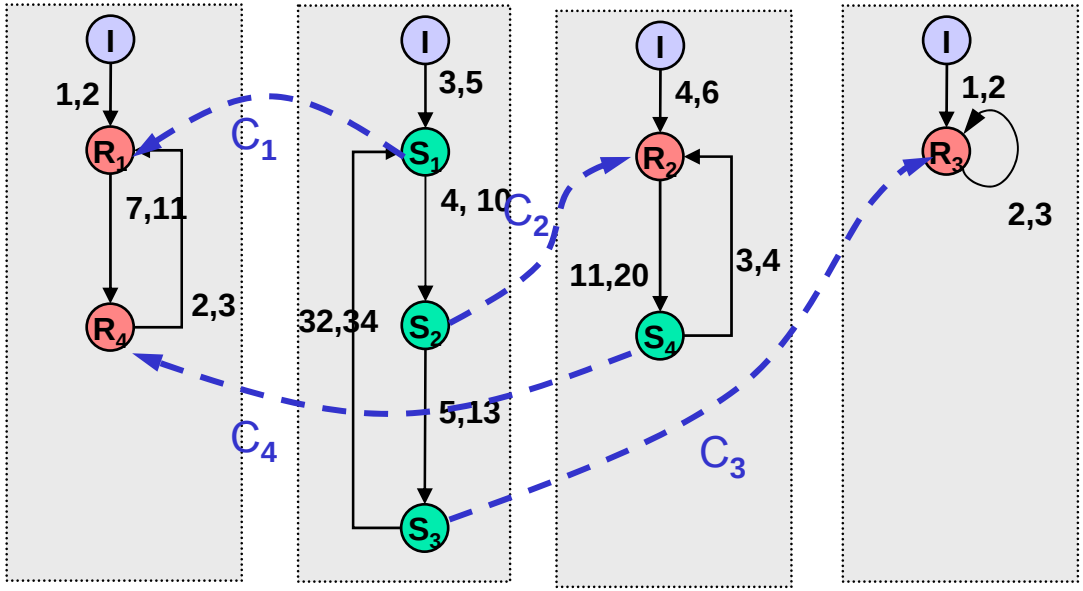
- Black-box approaches
  - Model abstraction
    - Abstract task model and task activation
    - Event streams and execution times of whole tasks
  - Propagation of event streams
  - Determination of fix-points
  - Approaches
    - Real-Time Calculus (Thiele, ETH Zurich)
    - Compositional event stream propagation: SymTA/S (Ernst, SymtaVision)
- White-box approaches
  - Inclusion of the control-flow of each process in system model
  - Global performance analysis considering control structures of all processes
  - Extraction of the control-flow from functional implementation or UML model
  - Environment modeling using event models or processes
  - Approaches
    - Analysis of communication dependencies: SysXplorer (Rosenstiel, FZI)
    - Control-flow based extraction of hierarchical event streams (Slomka, Oldenburg)
- Simulation-based approaches
  - No corner-case coverage
  - Resource and time efforts

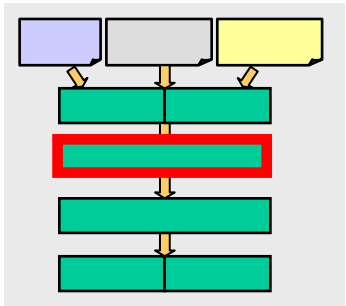
Analysis Flow



- Global performance analysis of HW/SW systems
- Automatic recognition of bottlenecks, timing violations and resource conflicts
- Optimization of the system architecture towards the application

Communication Analysis





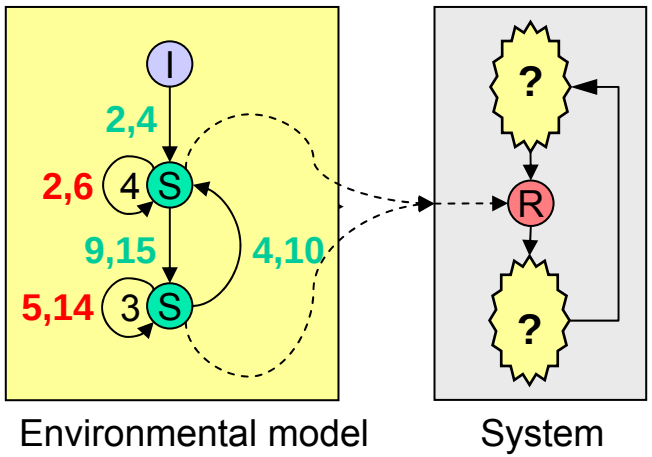
- Problem transformation to system of equations (using slack variables):

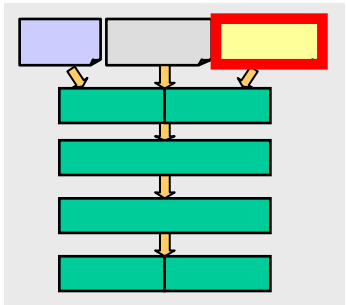
  Min time to wait: $\underline{x}_i$: min. path to (S), max. path to (R)

  Max time to wait $\overline{x}_i$: max. path to (S), min. path to (R)

- Negative slack represents a violated synchronization condition
- Analysis according to previous publications
  - Siebenborn, Bringmann, Rosenstiel: Worst-Case Performance Analysis of Parallel, Communicating Software Processes, CODES 2002
  - Siebenborn, Bringmann, Rosenstiel: Communication Analysis for System on Chip Design, DATE 2004

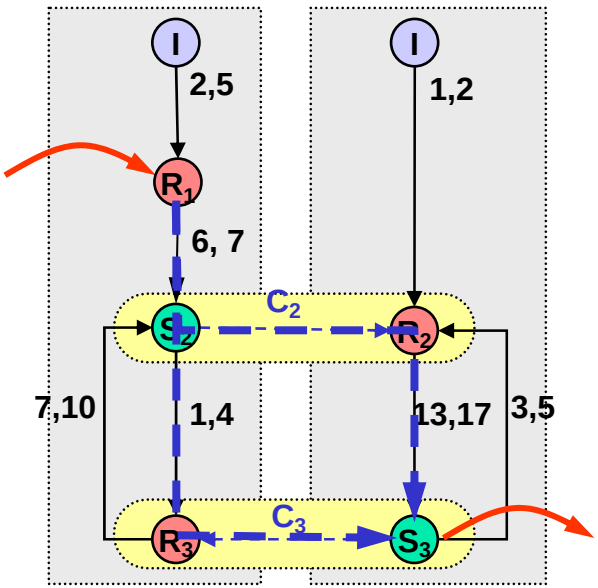
Modeling of the System Environment





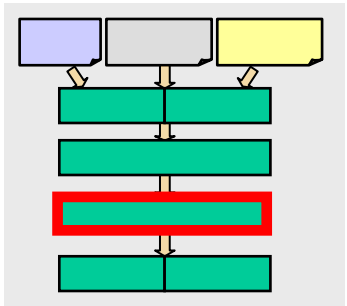
- Using a priori information
  - Application information ⇒ Control flow graph ⇒ Communication dependency graph
  - Environmental information ⇒ Event stream model ⇒ Environmental CDG
- Supporting common event models
  - Periodic events → $l_{min} = l_{max} = l_{period}$
  - Sporadic events → $l_{min} = min(l_{sporadic})$, $l_{max} = \infty$
  - Burst model → Loops
  - Jitter → Min/Max-Latencies ($l_{max} - l_{min}$)
- Inclusion of the environmental model in system evaluation

Performance Analysis: Latency



$$l_{max}(R_1 \rightarrow S_3) = l_{max}(R_1 \rightarrow S_2) + l_{max}(R_2 \rightarrow S_3)$$

$$= 7 + 17 = 24$$



$$l_{max}(R_1 \rightarrow S_3) = l_{max}(R_1 \rightarrow S_2) + l_{min}(S_2 \rightarrow R_3) + \bar{x}_3$$

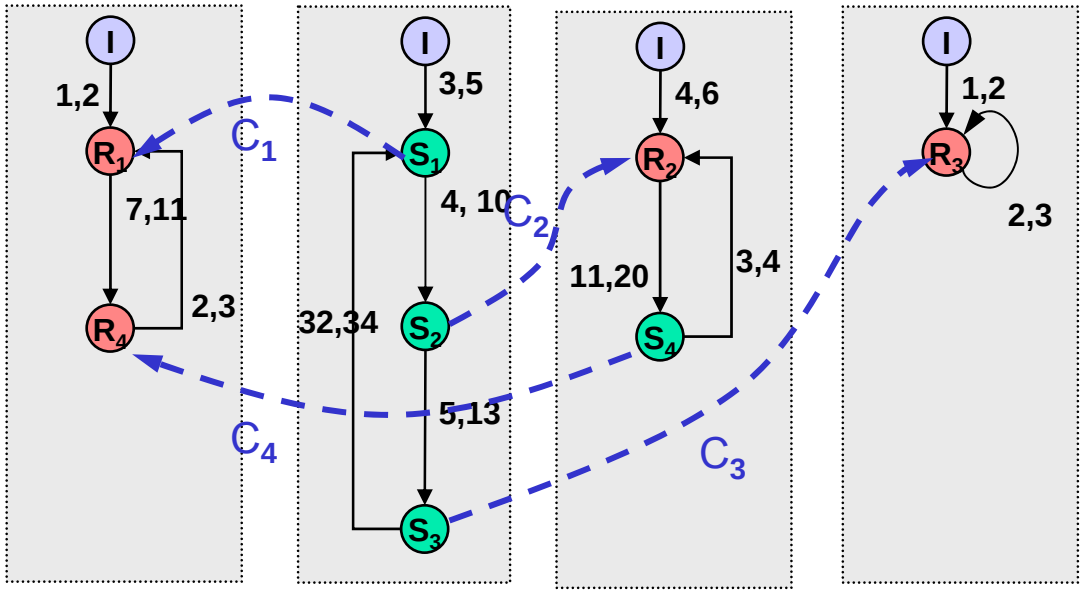
$$= 7 + 1 + 16 = 24$$

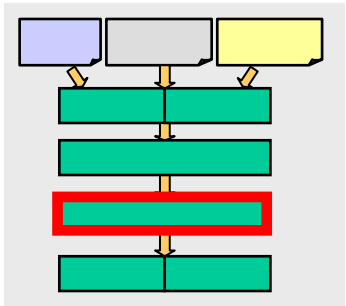
Synchronization points (SP) temporally relate parallel processes

⇒ SP used for propagation of latencies between processes

⇒ Different paths passing SPs deliver same result

Conflict Analysis: Communication Resources





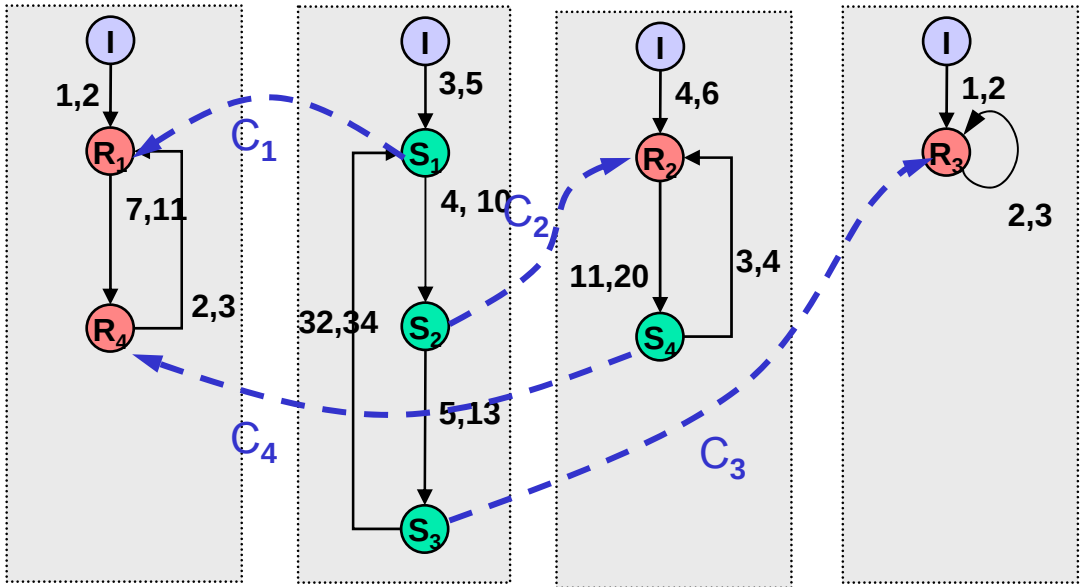
Which communication instances produce conflicts on shared resources?

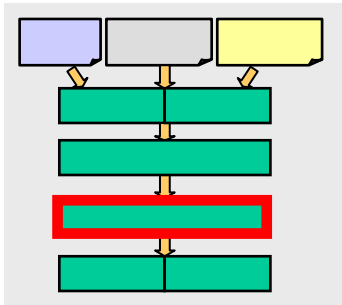
Determined slack variables can be used to calculate absolute time intervals in which communication instances takes place.

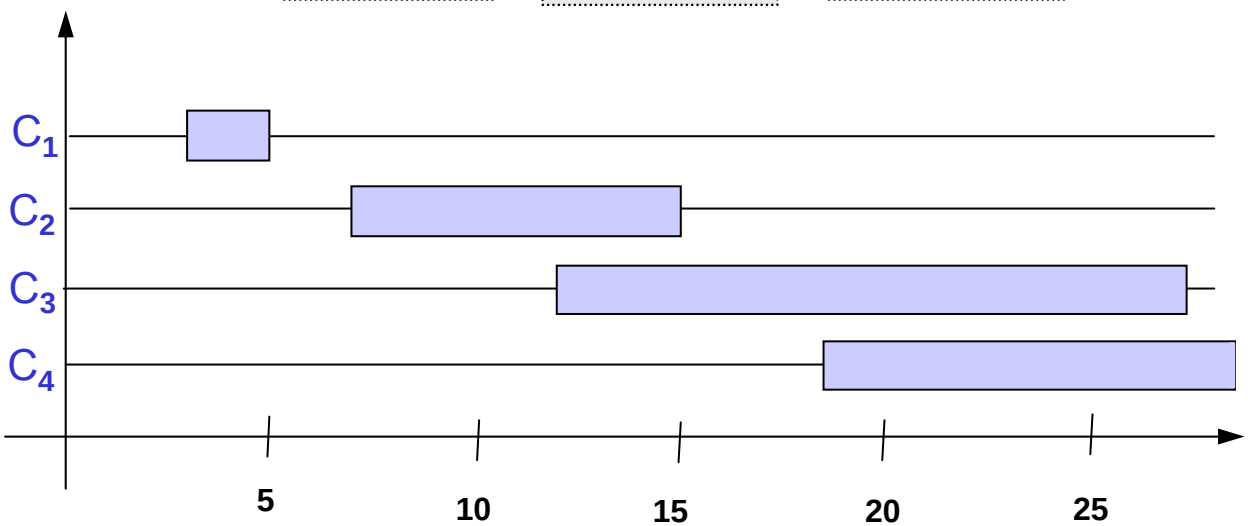
Overlapping time intervals even in cases where conflicts are not possible considering the sequential order of communications

⇨ Communication order is important

Conflict Analysis: Communication Resources

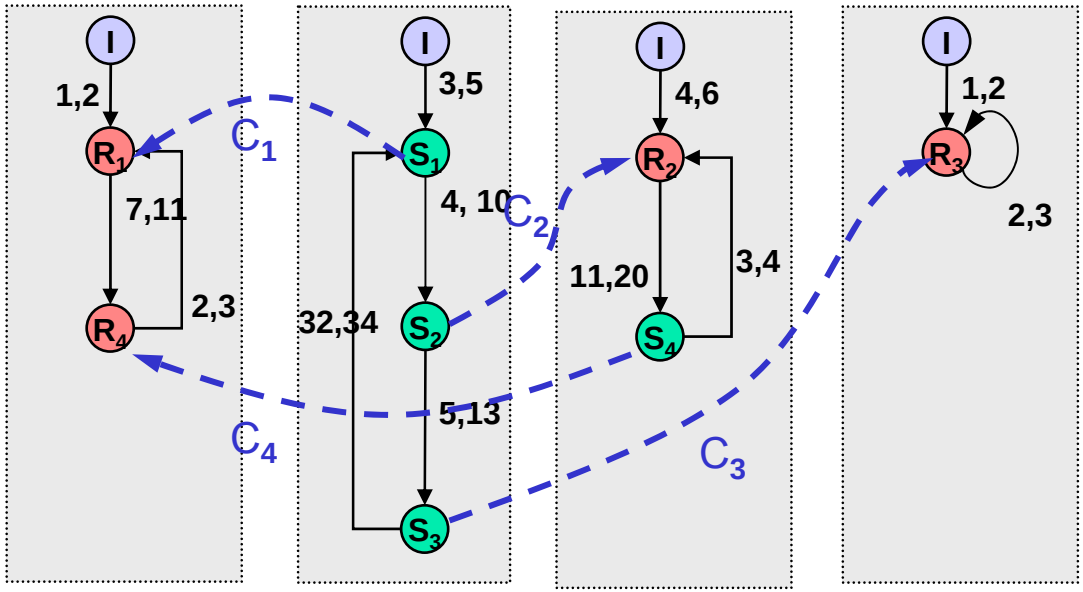


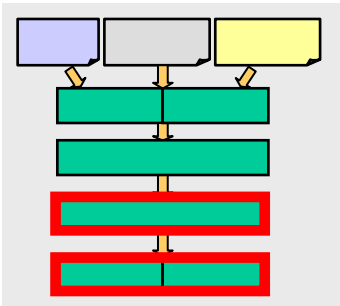




Intervals of the start times of the communications

Conflict Analysis: Communication Resources





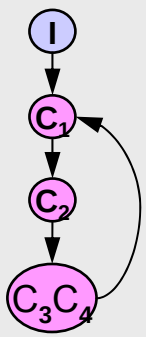
1. Communication Scheduling Graph

$[I_1, I_2, I_3, I_4]$

$[R_1, S_1, R_2, R_3]$

$[R_4, S_2, R_2, R_3]$

$[R_4, S_3, R_4, R_3]$



C_3 and C_4 potentially parallel!

2. Check paths

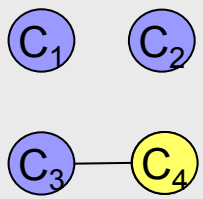
$R_2 \rightarrow S_4$ [11, 20]

$S_2 \rightarrow S_3$ [5, 13]

Overlaps of intervals

C_3 and C_4 can have conflict!

3. Coloring of the conflict graph



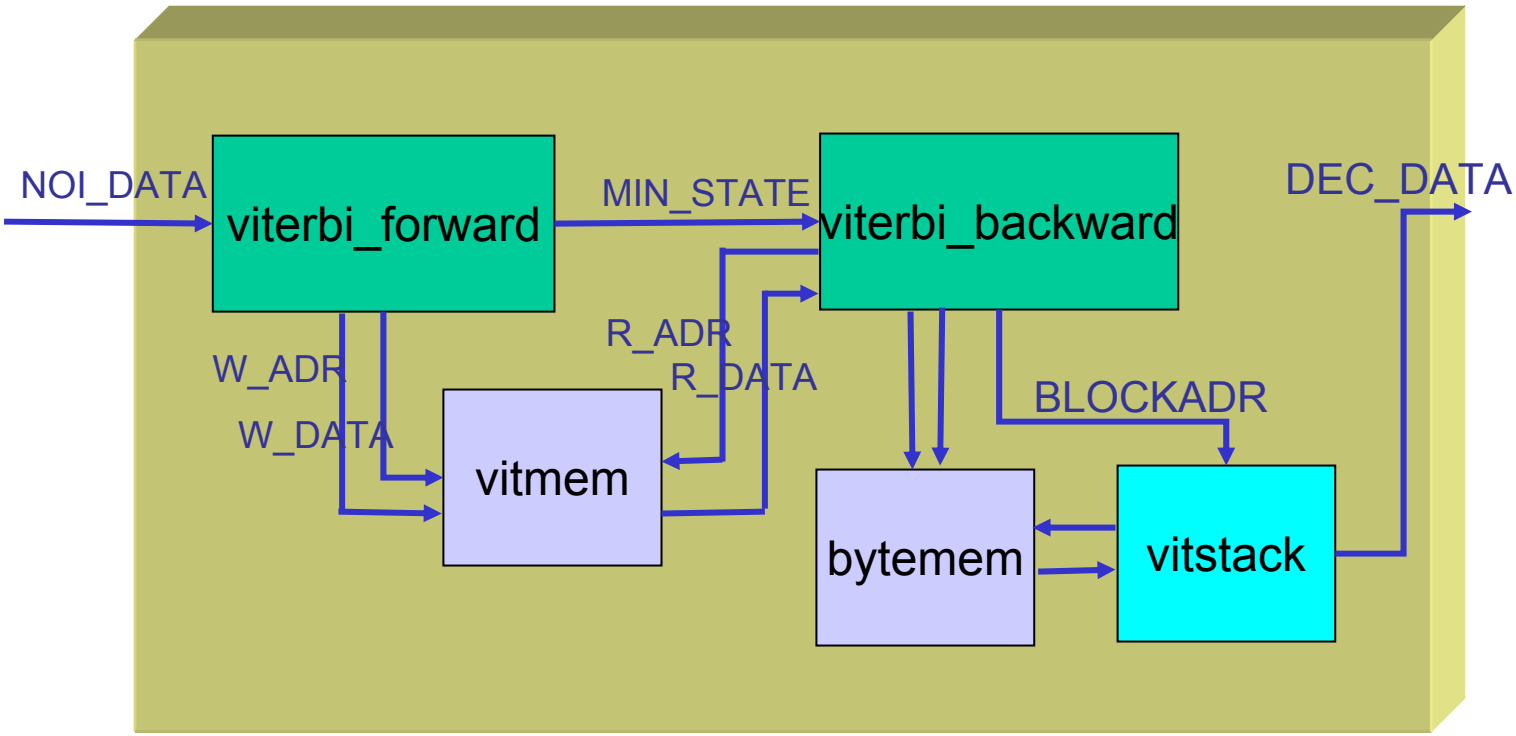
Res 1 C_1, C_2, C_3

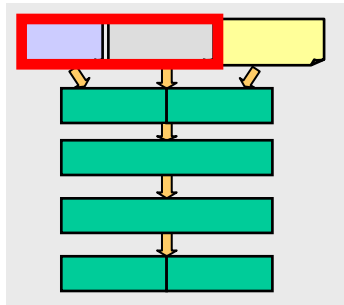
Res 2 C_4

Case Study: Viterbi-Decoder

- Maximal data rate at input (NOI_DATA)
- Latency between input and output data (WCRT)
- Maximal data rate at output (DEC_DATA)

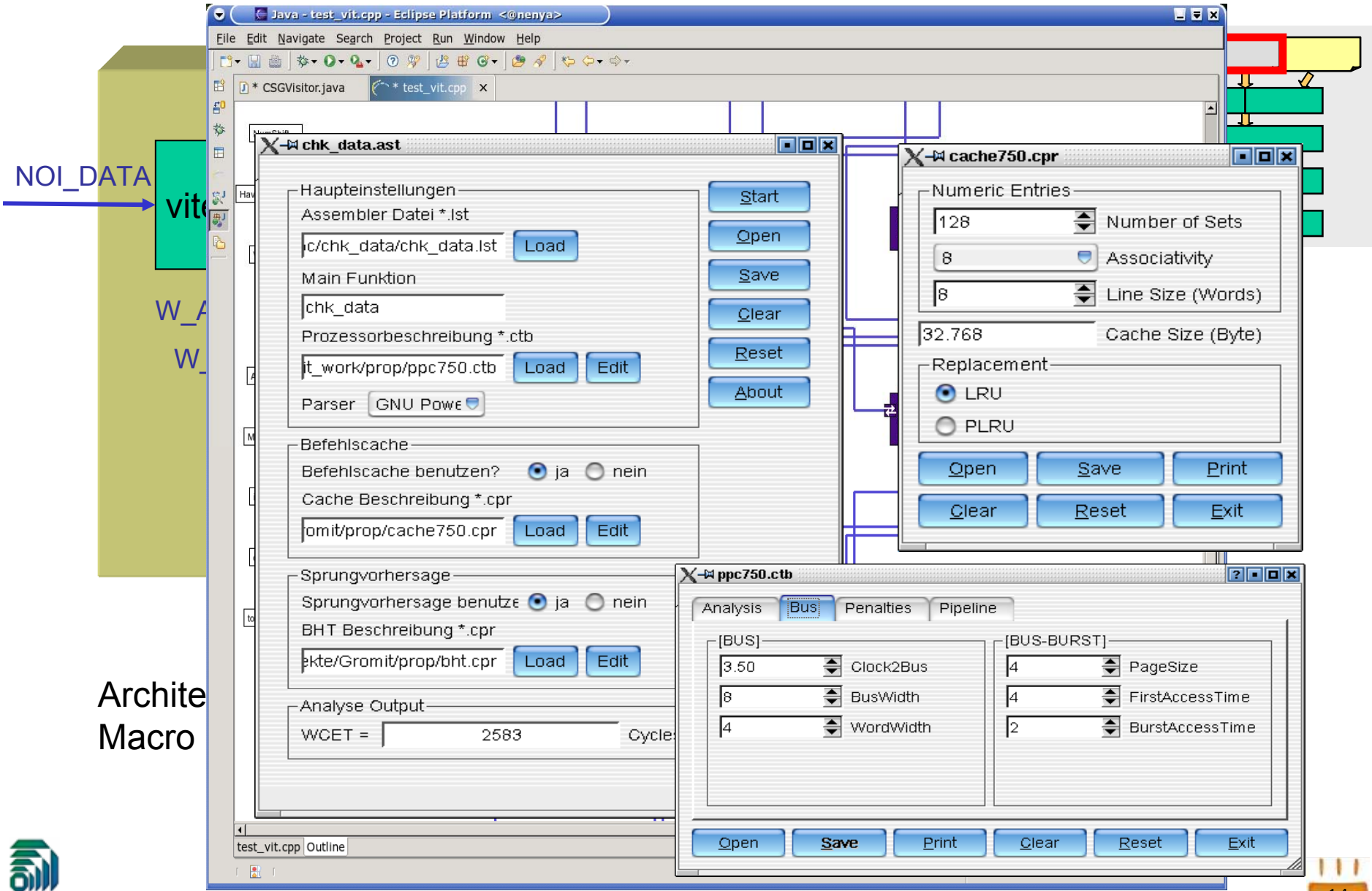
Viterbi-Decoder: Structure



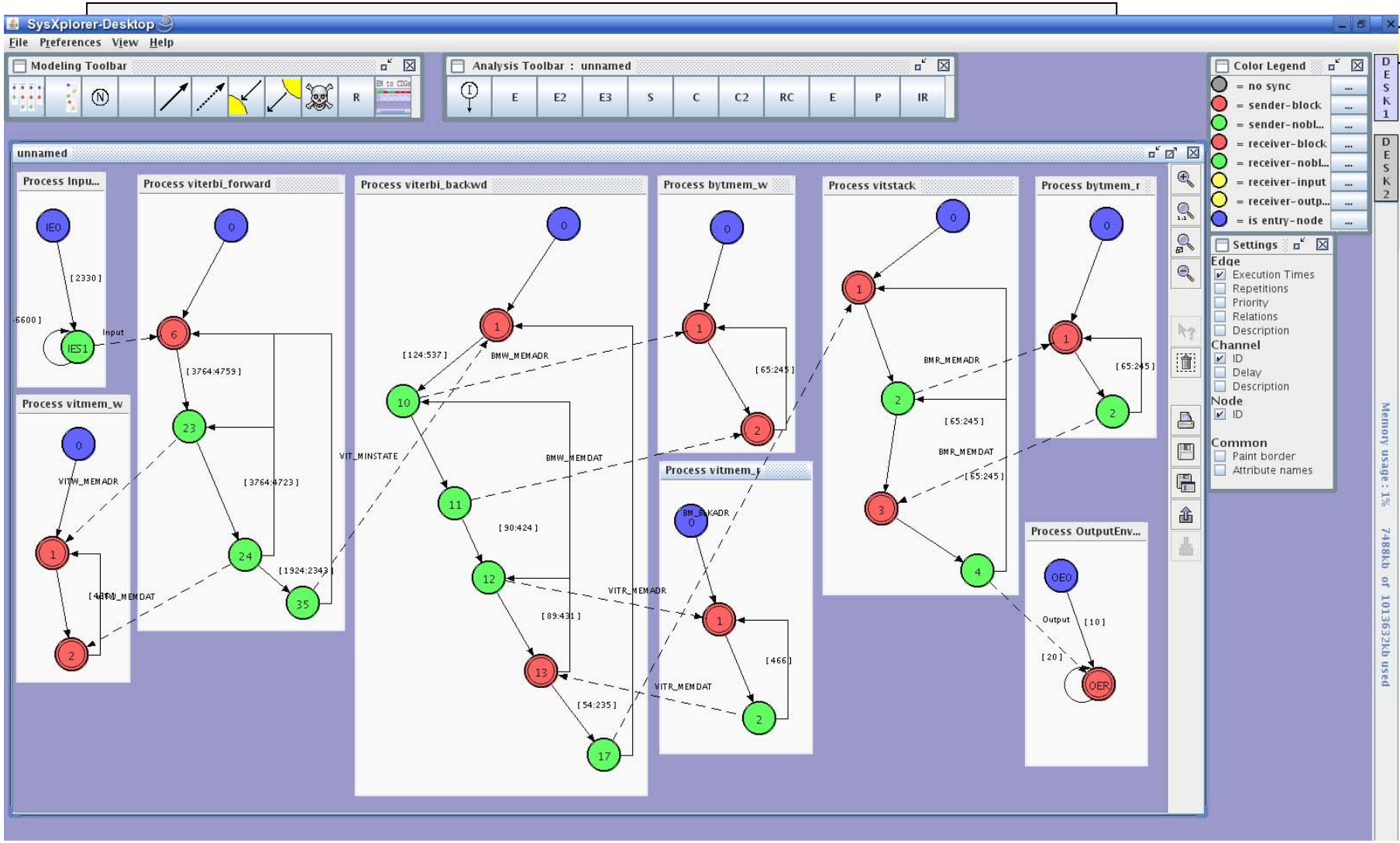


- Latency of the blocks
- Synchronization, deadlock, data loss
- Requirements for blocks

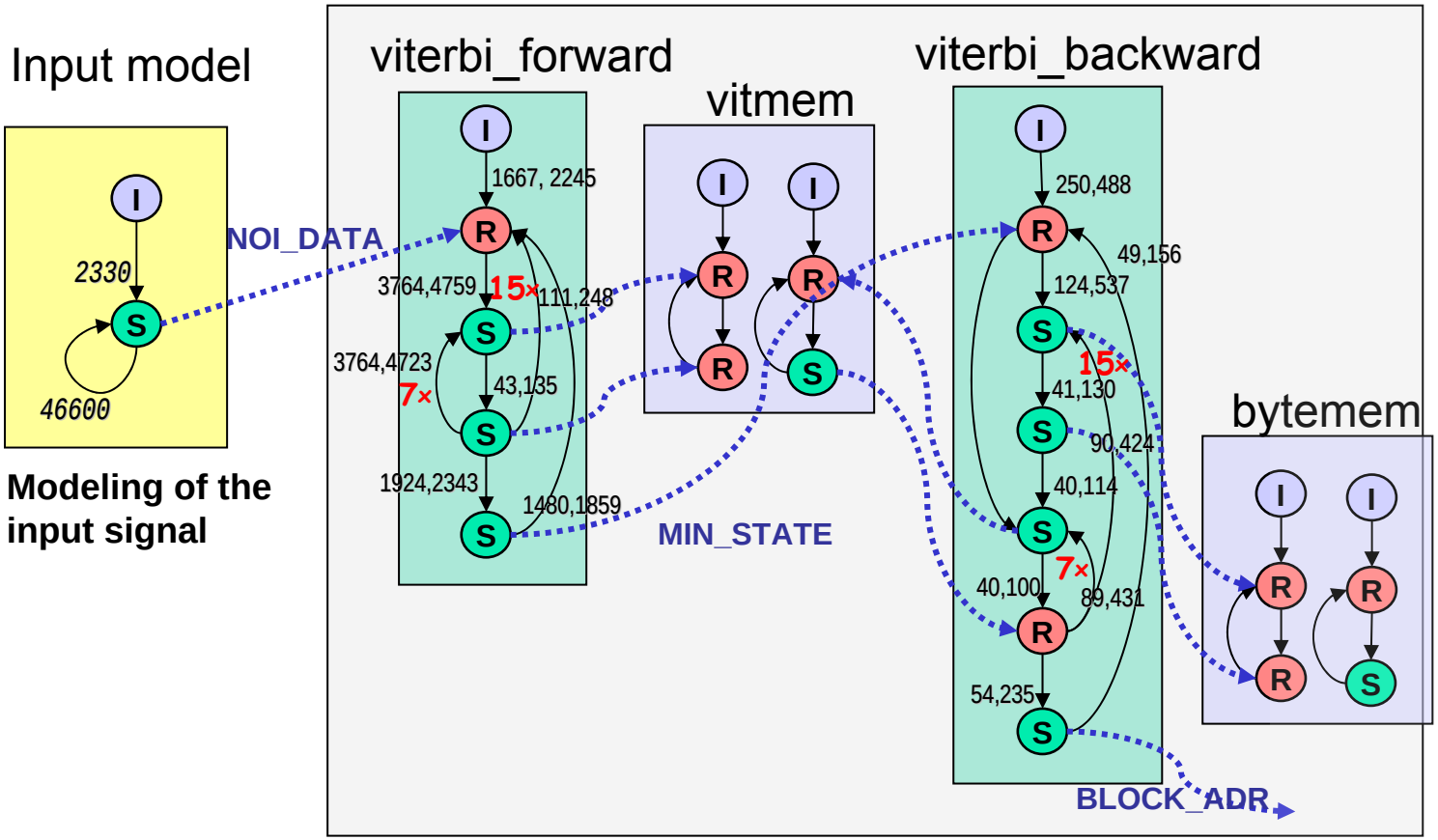
Viterbi-Decoder: Platform Mapping



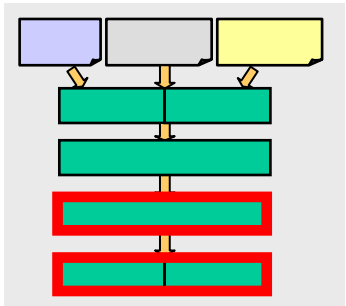
Viterbi-Decoder: CDG



Viterbi Decoder: Analysis Examples



Modeling of the input signal

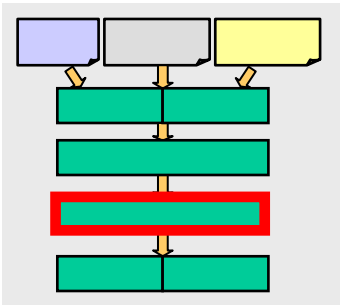


Synchronization points and latency:

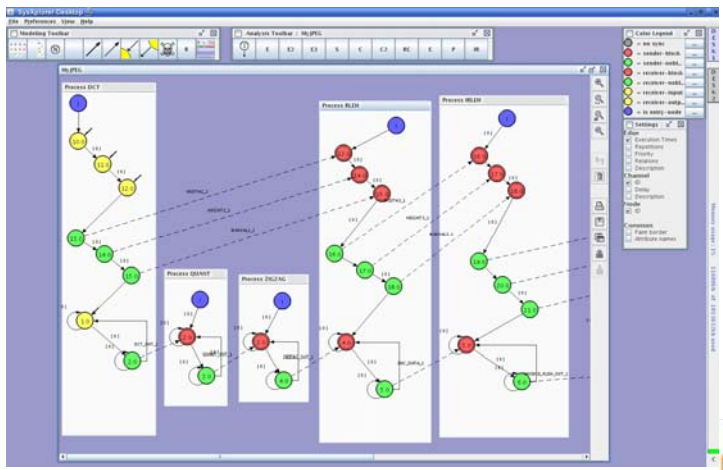
- NOI_DATA (C_2): $C_2 \rightarrow C_2$: $\Rightarrow \underline{x}_{2.1}$ = 3498 wait states
- MIN_STATE (C_5): $C_5 \rightarrow C_5$: $\Rightarrow \underline{x}_{5.1}$ = 557769 wait states $\triangleq$ 2393.85 µs (233MHz)
- WCRT: NOI_DATA (C_2) $\rightarrow$ BLOCK_ADR (C_{11}): $L_{path,max}(C_2 \rightarrow C_{11})$ = 862331 $\triangleq$ 3701 µs

Viterbi Decoder: Architectural Exploration

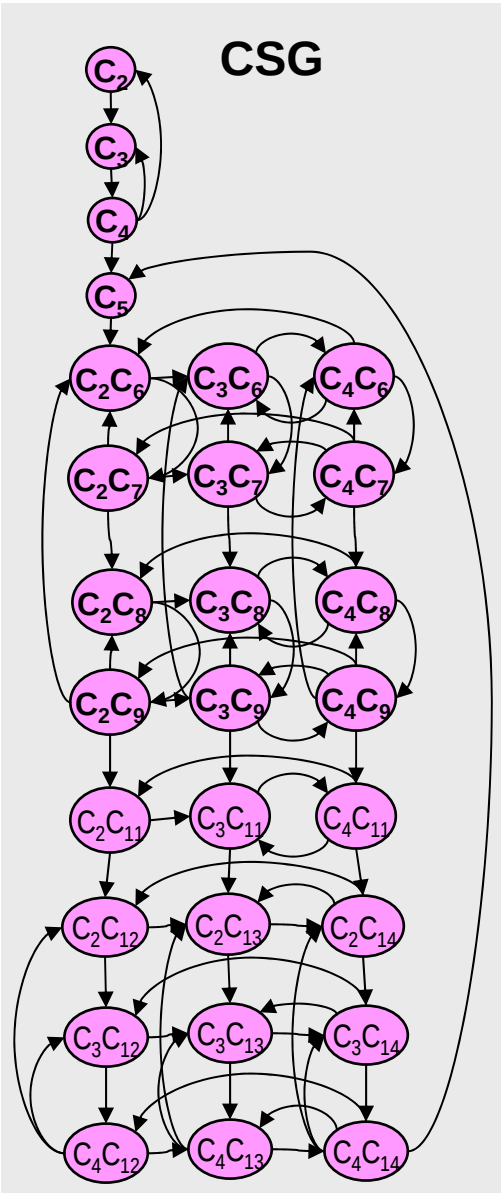
Conf. #	viterbi_forward	viterbi_backward	$l_{in,per}$	$\underline{x}_5$	$\underline{x}_2$
1	233 MHz I, D-Cache	233 MHz I,D-cache	200 µs	2906 µs	6 µs
2	233 MHz I,D-cache	100 MHz no caches	200 µs	1495 µs	6 µs
3	100 MHz I,D-cache	100MHz I,D-cache	500 µs	6294 µs	49 µs
4	233 MHz D-cache	100MHz I, D-cache	1500 µs	2283 µs	46 µs

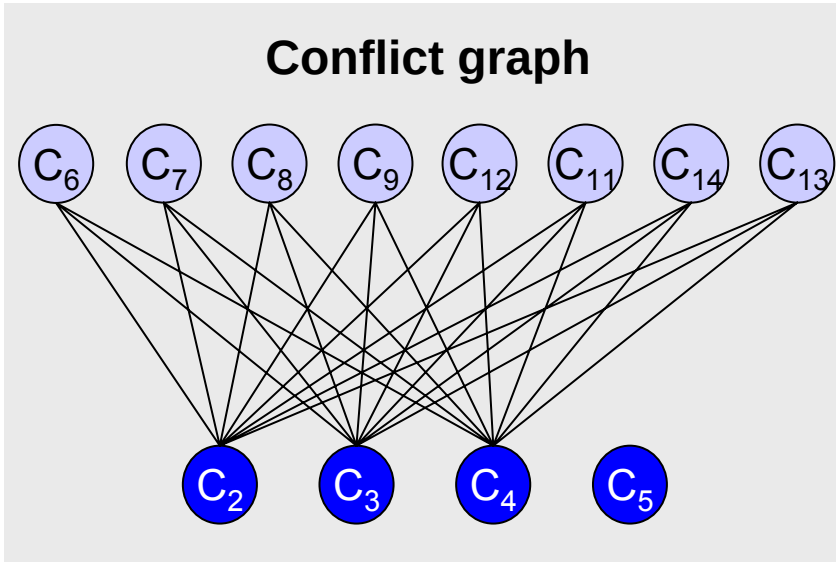


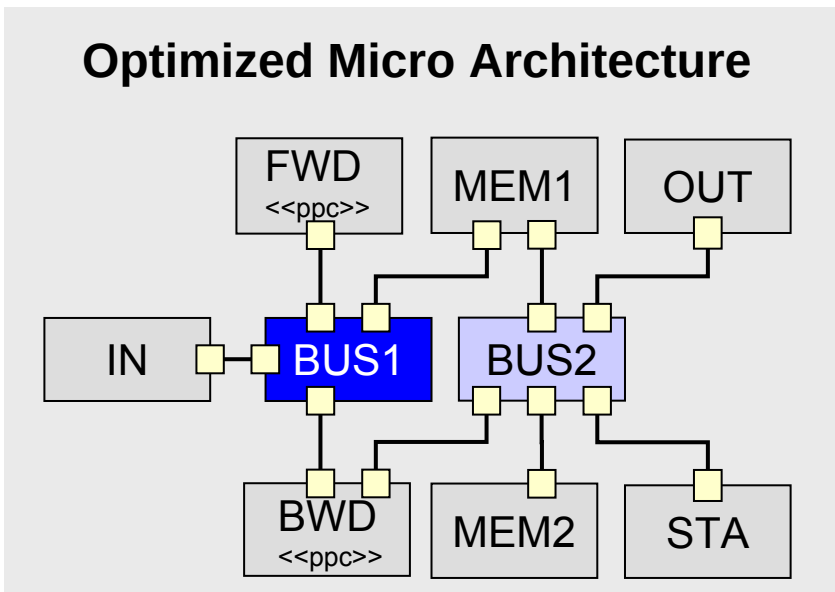
- Implementation of analysis methods in SysXplorer tool
  - Environmental modeling
  - Automated analysis
  - Design space exploration

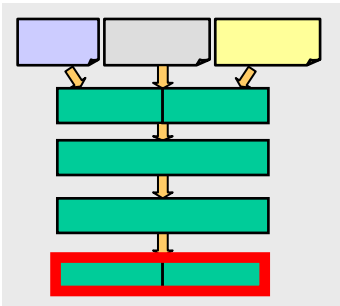


Viterbi-Decoder: Allocation and Binding









Conclusion

- Methods for communication and runtime analysis
  - Considering the control flow of the processes during analysis
  - Synchronization points set processes into temporal relation
  - Determination of the minimal and maximal wait time in communications
- Recognition of possible resource conflicts
- Inclusion of temporal system environment by common event models
- Example: Viterbi Decoder
  - Maximal I/O data rates
  - Latency between input and output data (WCRT)
  - Latency of single blocks
  - Synchronization, deadlock, data loss
  - Resource conflicts
  - Allocation and binding of resources

Questions