Formal analysis and optimization of heterogeneous networks in industrial practice
- from networked systems to MpSoC

Marek Jersak, Symtavision

Rolf Ernst, TU Braunschweig

Overview

- **Part 1:**
  Formal analysis in industrial practice
  Marek Jersak, Symtavision

- **Part 2:**
  From networked systems to MpSoC
  Rolf Ernst, TU Braunschweig

Formal analysis and optimization of heterogeneous networks in industrial practice

Marek Jersak, Symtavision

Integration Challenges

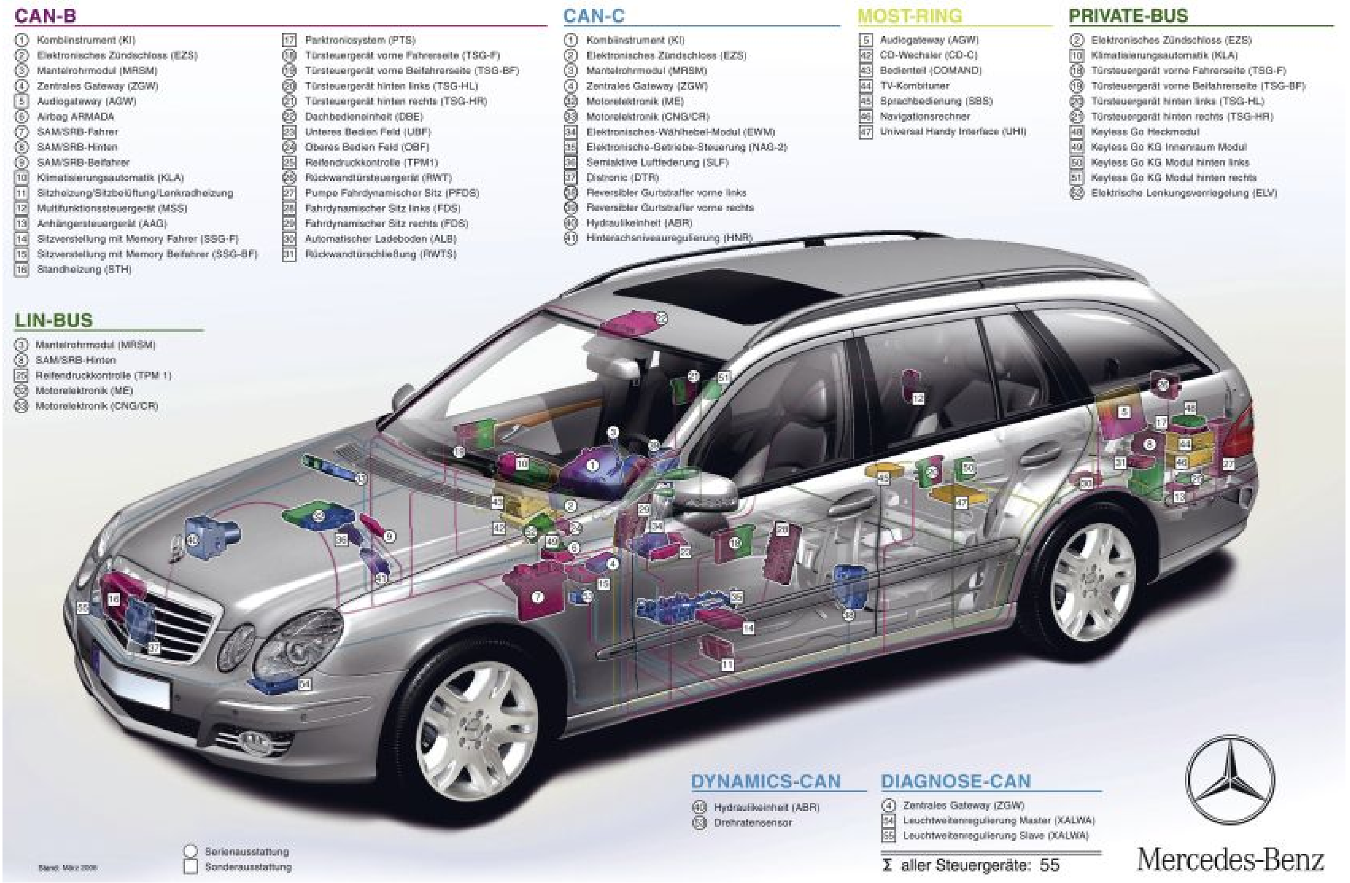
Complexity

- □ Hundreds of functions, many safety-critical
- □ 50+ ECUs
- □ Networked
- □ Many suppliers

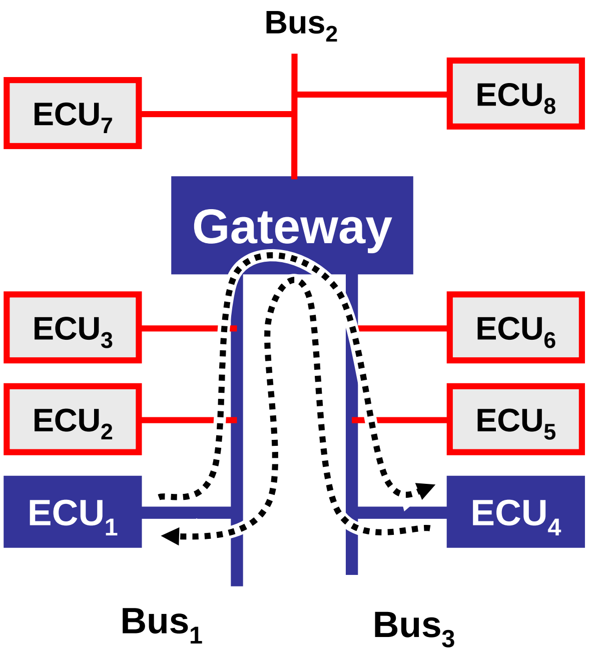
Integration challenges

- □ Reliability, quality, liability
- □ Meeting SOP target
- □ Development and production cost

Integration Challenges: 5 buses, 55 ECUs


CAN-B
1 Kombiinstrument (KI)
2 Elektronisches Zündschloss (EZS)
3 Mantelrohrmodul (MRSM)
4 Zentrales Gateway (ZGW)
5 Audiogateway (AGW)
6 Airbag ARMADA
7 SAM/SRB-Fahrer
8 SAM/SRB-Hinten
9 SAM/SRB-Beifahrer
10 Klimatisierungsautomatik (KLA)
11 Sitzheizung/Sitzbelüftung/Lenkradheizung
12 Multifunktionssteuergerät (MSS)
13 Anhängersteuergerät (AAG)
14 Sitzverstellung mit Memory Fahrer (SSG-F)
15 Sitzverstellung mit Memory Beifahrer (SSG-BF)
16 Standheizung (STH)
17 Parktronicsystem (PTS)
18 Türsteuergerät vorne Fahrerseite (TSG-F)
19 Türsteuergerät vorne Beifahrerseite (TSG-BF)
20 Türsteuergerät hinten links (TSG-HL)
21 Türsteuergerät hinten rechts (TSG-HR)
22 Dachbedieneinheit (DBE)
23 Unteres Bedien Feld (UBF)
24 Oberes Bedien Feld (OBF)
25 Reifendruckkontrolle (TPM1)
26 Rückwandtürsteuergerät (RWT)
27 Pumpe Fahrdynamischer Sitz (PFDS)
28 Fahrdynamischer Sitz links (FDS)
29 Fahrdynamischer Sitz rechts (FDS)
30 Automatischer Ladeboden (ALB)
31 Rückwandtürschließung (RWTS)
CAN-C
1 Kombiinstrument (KI)
2 Elektronisches Zündschloss (EZS)
3 Mantelrohrmodul (MRSM)
4 Zentrales Gateway (ZGW)
32 Motorelektronik (ME)
33 Motorelektronik (CNG/CR)
34 Elektronisches-Wählhebel-Modul (EWM)
35 Elektronische-Getriebe-Steuerung (NAG-2)
36 Semiaktive Luftfederung (SLF)
37 Distronic (DTR)
38 Reversibler Gurtstraffer vorne links
39 Reversibler Gurtstraffer vorne rechts
40 Hydraulikeinheit (ABR)
41 Hinterachsniveauregulierung (HNR)
MOST-RING
5 Audiogateway (AGW)
42 CD-Wechsler (CD-C)
43 Bedienteil (COMAND)
44 TV-Kombituner
45 Sprachbedienung (SBS)
46 Navigationsrechner
47 Universal Handy Interface (UHI)
PRIVATE-BUS
2 Elektronisches Zündschloss (EZS)
10 Klimatisierungsautomatik (KLA)
18 Türsteuergerät vorne Fahrerseite (TSG-F)
19 Türsteuergerät vorne Beifahrerseite (TSG-BF)
20 Türsteuergerät hinten links (TSG-HL)
21 Türsteuergerät hinten rechts (TSG-HR)
48 Keyless Go Heckmodul
49 Keyless Go KG Innenraum Modul
50 Keyless Go KG Modul hinten links
51 Keyless Go KG Modul hinten rechts
52 Elektrische Lenkungsverriegelung (ELV)
LIN-BUS
3 Mantelrohrmodul (MRSM)
8 SAM/SRB-Hinten
25 Reifendruckkontrolle (TPM 1)
32 Motorelektronik (ME)
33 Motorelektronik (CNG/CR)
DYNAMICS-CAN
40 Hydraulikeinheit (ABR)
53 Drehratensensor
DIAGNOSE-CAN
4 Zentrales Gateway (ZGW)
54 Leuchtweitenregulierung Master (XALWA)
55 Leuchtweitenregulierung Slave (XALWA)
Σ aller Steuergeräte: 55
○ Serienausstattung
□ Sonderausstattung
Stand: März 2006
Mercedes-Benz

Typical Automotive Architecture



Bus protocols (and arbitration)

- CAN – static-priority non-preemptive
- FlexRay – TDMA + SPNP
- Lin
- MOST
- Proprietary

ECU (electronic control unit)

- OSEK RTOS (different flavours exist) – static priority preemptive

End-to-end timing is important

Many functional problems are in fact *timing* problems

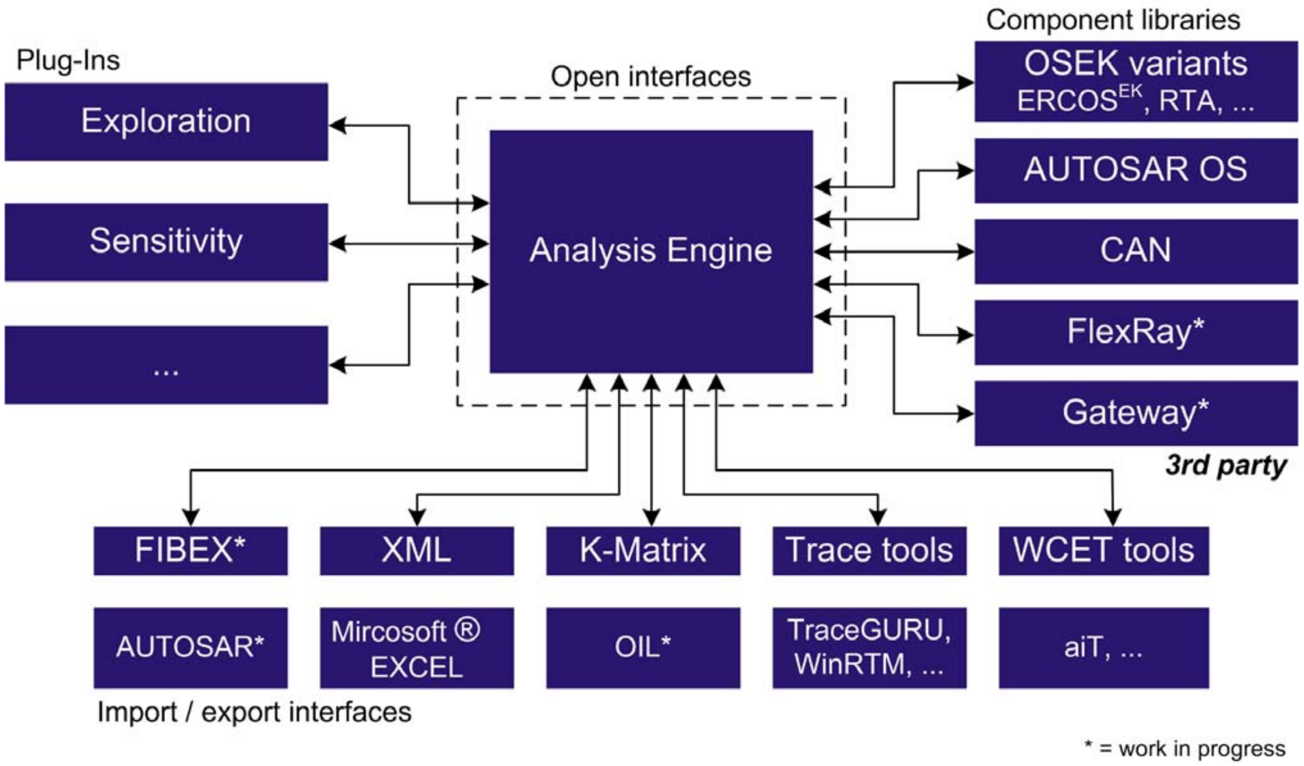
- □ ECUs (temporarily) overloaded
- □ tasks not always schedulable
- □ deadlines are missed

- □ network (temporarily) overloaded
- □ messages arrive "too late“ or with “too large” jitter
- □ messages are lost (buffer overflow)

- □ end-to-end deadlines of car function are missed
- □ stability of distributed control is compromised

➔ Carefully monitor performance and timing during design and integration

SymTA/S Tool Suite: Scheduling Analysis and Optimization



Component libraries

Plug-Ins

Open interfaces

Exploration

Sensitivity

...

Analysis Engine

OSEK variants ERCOS^{EK}, RTA, ...

AUTOSAR OS

CAN

FlexRay*

Gateway*

3rd party

FIBEX*

XML

K-Matrix

Trace tools

WCET tools

AUTOSAR*

Mircosoft ® EXCEL

OIL*

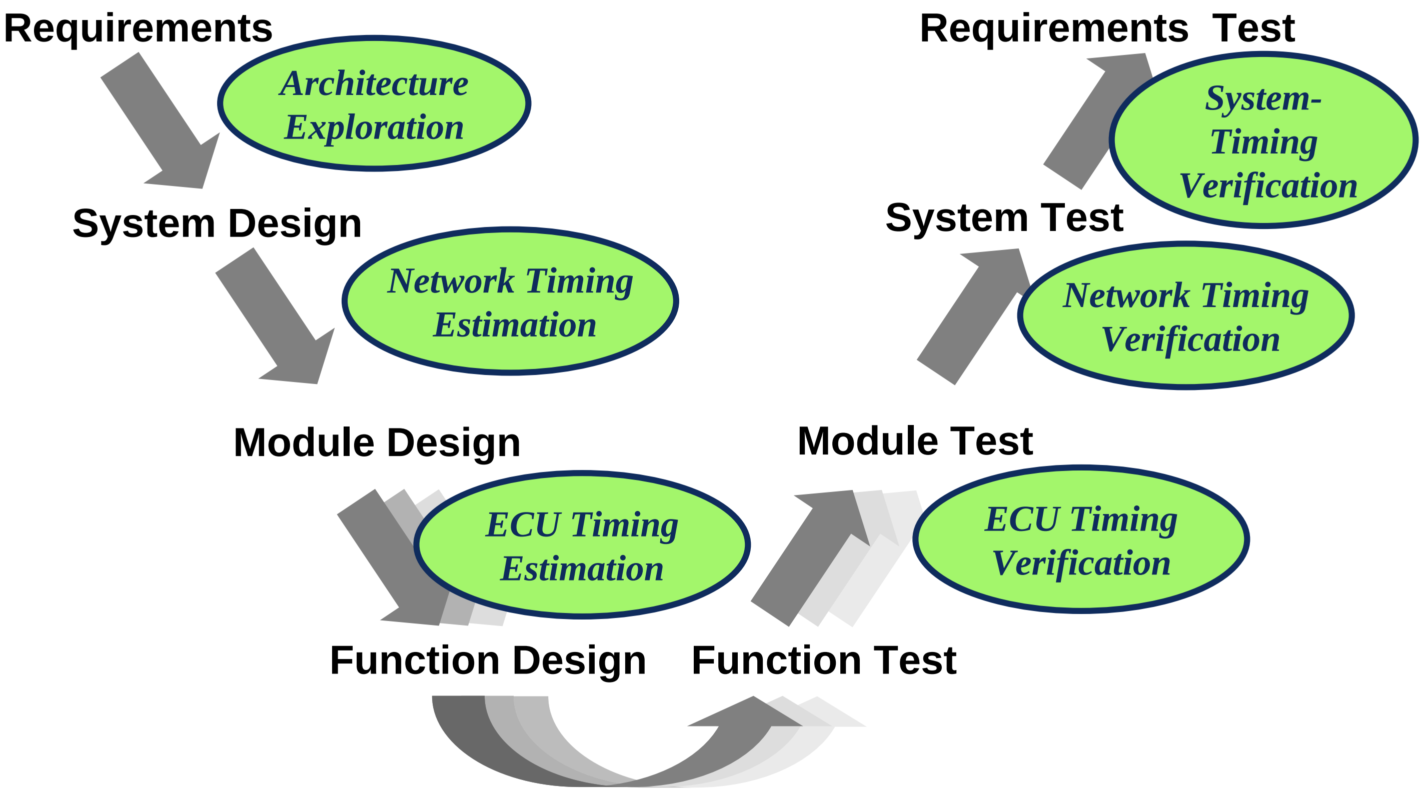
TraceGURU, WinRTM, ...

aiT, ...

Import / export interfaces

* = work in progress

Design-Phases



Requirements

Architecture Exploration

System Design

Network Timing Estimation

Module Design

ECU Timing Estimation

Function Design

Function Test

ECU Timing Verification

Module Test

Network Timing Verification

System Test

System-Timing Verification

Requirements Test

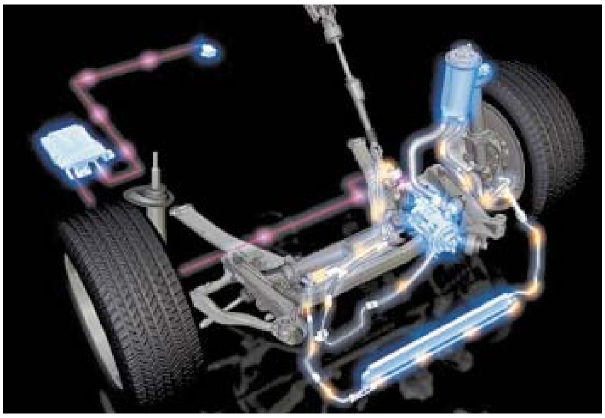
SW-Integration on ECUs

Verification of SIL-3 Project

Verify Critical ECU Timing

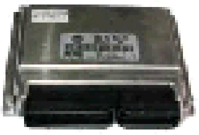
Active Front Steering

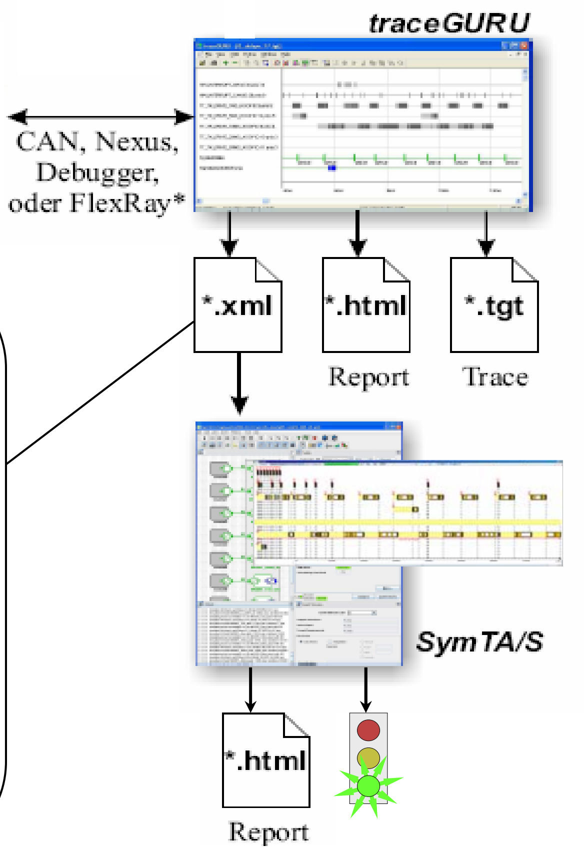
- ECU trace data import
- Timing analysis using SymTA/S including sensitivity analysis
- Result: reliable performance, cost savings (use of smaller CPU)



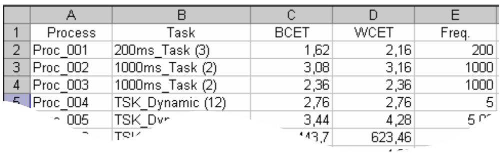
Source: BMW

Integration: Tracing + Scheduling Analyse



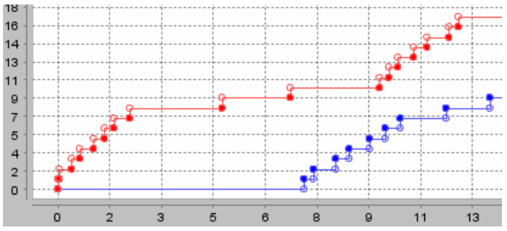


– Single function execution times



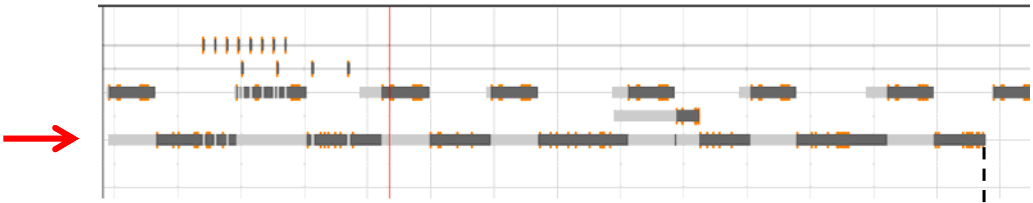
	A	B	C	D	E
1	Process	Task	BCET	WCET	Freq.
2	Proc_001	200ms_Task (3)	1,62	2,16	200
3	Proc_002	1000ms_Task (2)	3,08	3,16	1000
4	Proc_003	1000ms_Task (2)	2,36	2,36	1000
5	Proc_004	TSK_Dynamic (12)	2,76	2,76	5
	005	TSK_Dyn	3,44	4,28	5
		TSK	43,7	623,46	

– Interrupt Frequency

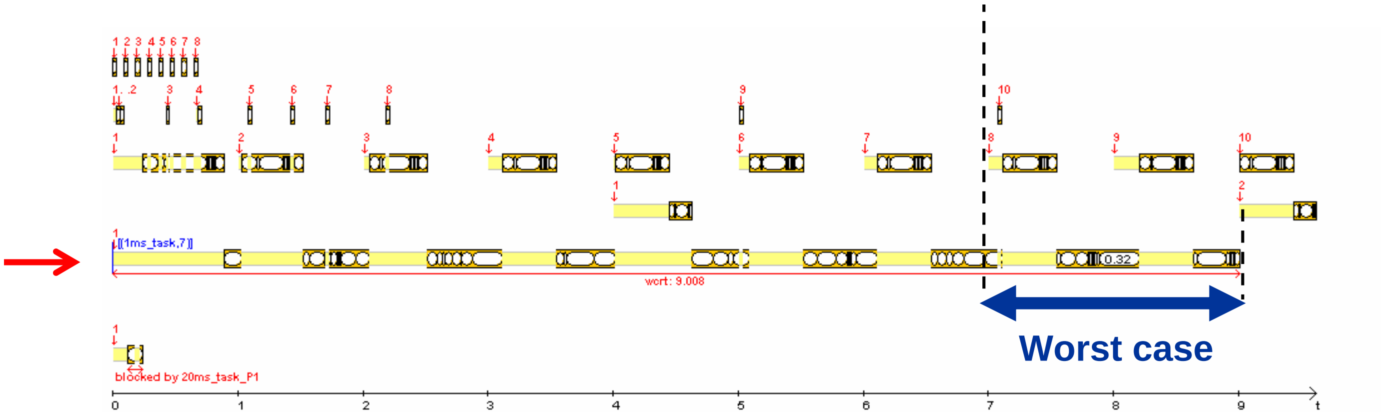


Measurement vs. SymTA/S Analysis

- Measured, 10ms task, Response Time 6,9ms
  - 4 CAN, 8 SPI Interrupts, 7 preemptions by 1ms task



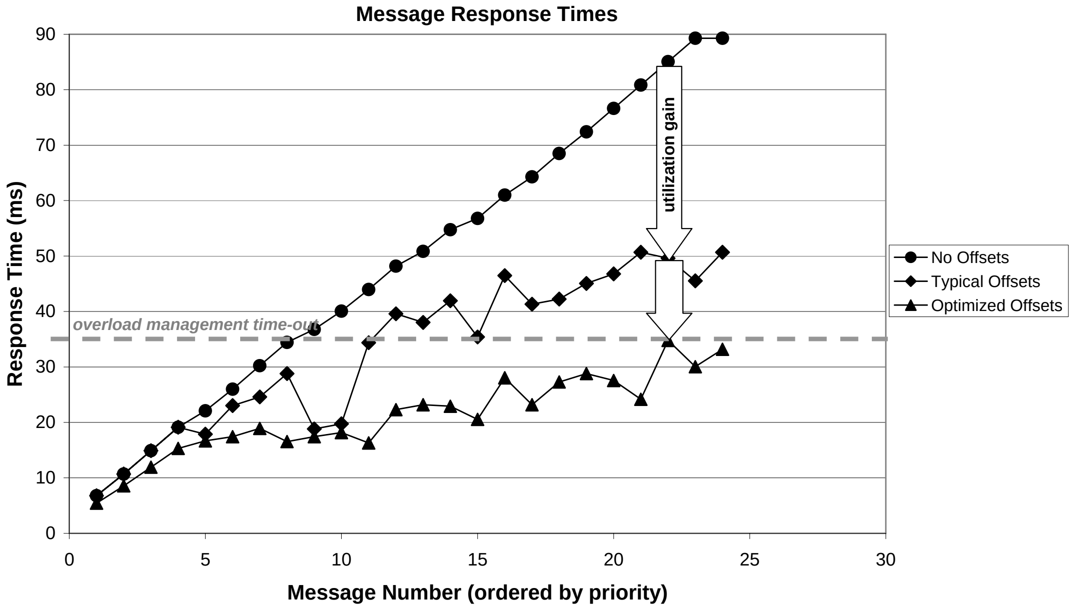
- SymTA/S Analysis, 10ms task, Response Time 9ms
  - 10 CAN, 8 SPI Interrupts, 9 preemptions by 1ms task, blocking



CAN Bus Optimization:
CAN IDs (== message priorities),
Message offsets (traffic shaping)

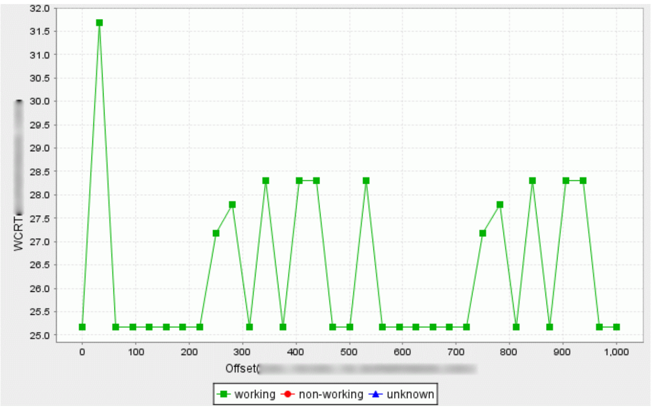
Reliable CAN Bus Extension

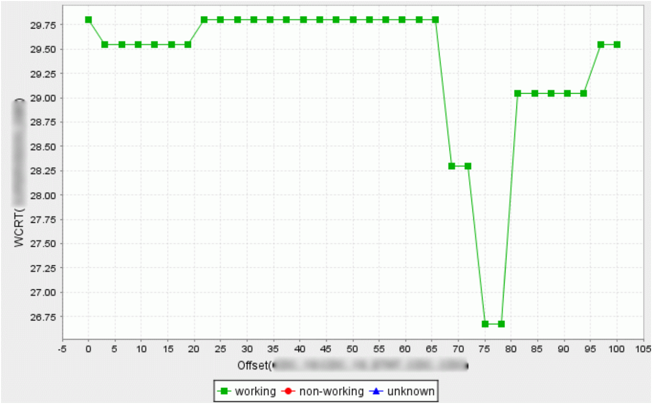
- Problem: CAN bus load high but needs to carry more frames
- SymTA/S: Offset /CAN Id optimization ➔ room for new frames
- Result: Increased utilization and reliable, safe extension

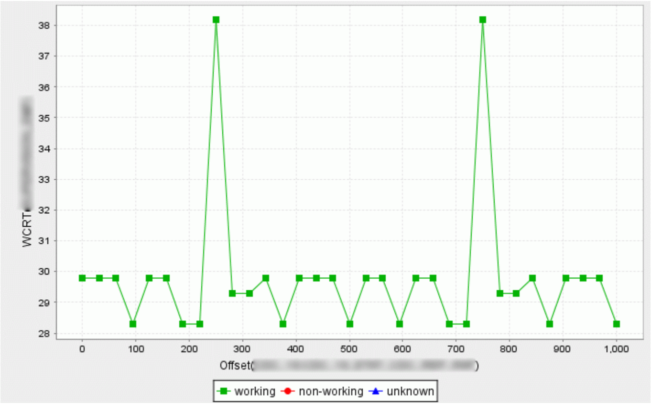


Detailed Sensitivity Analysis

$$\frac{\delta \text{ response time}}{\delta \text{ offset}}$$





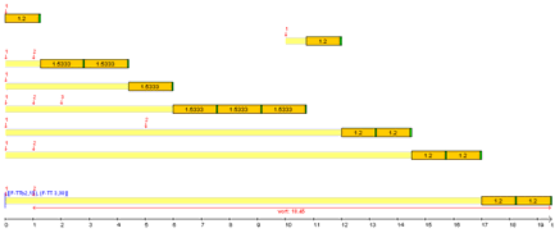


CAN bus load under

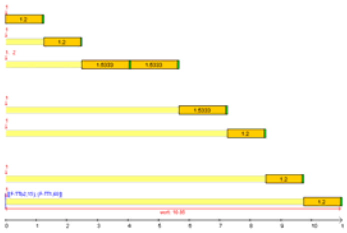
varying dynamic load situations

Analyzing Several Dynamic Load Profiles

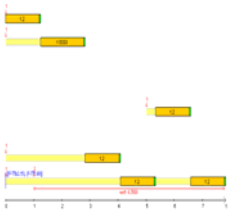
- □ all triggered signals considered
  (in direct and mixed frames)
  ➔ full dynamic CAN load



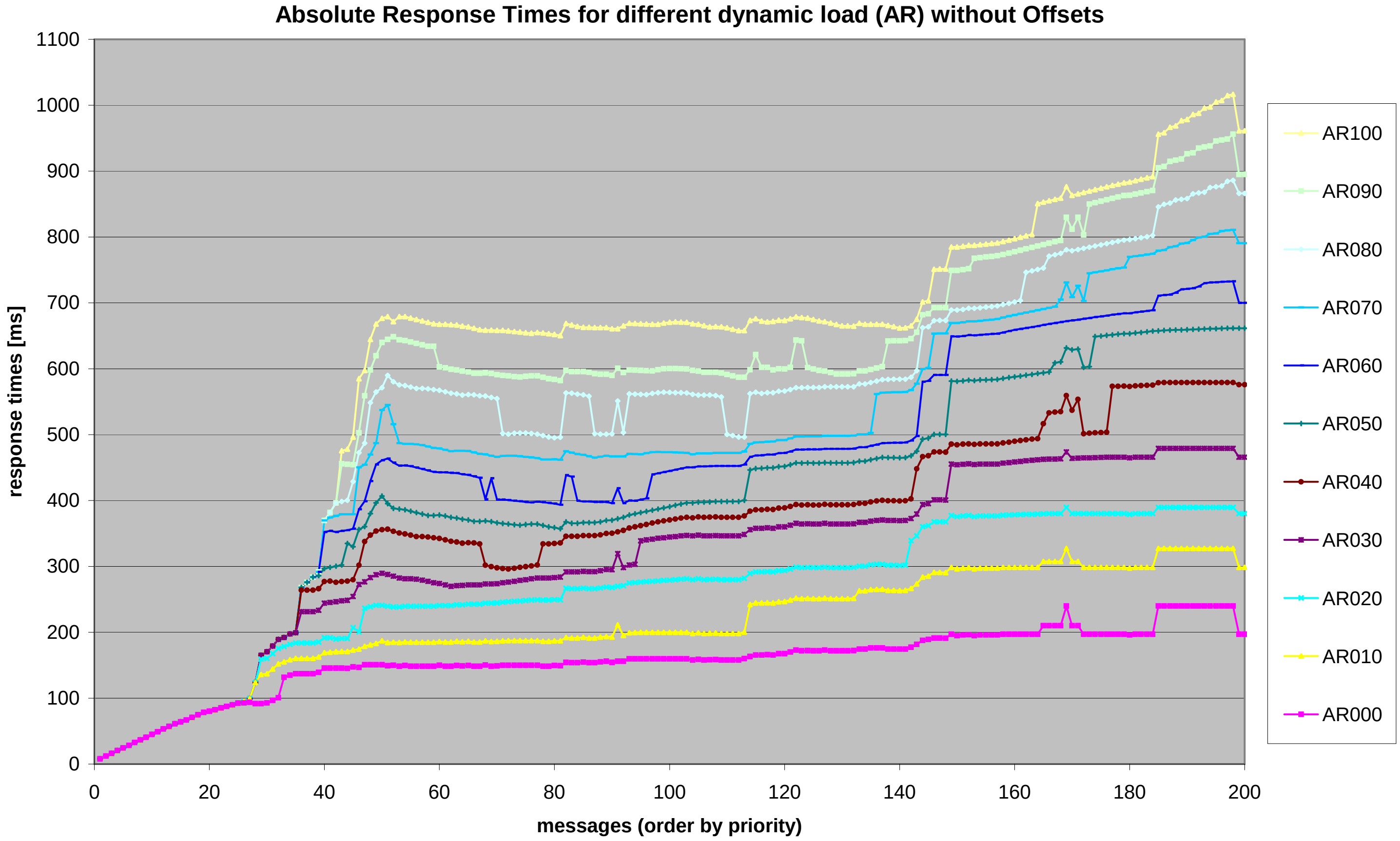
- □ subset of dynamic signals
  ➔ representative CAN load



- □ no dynamic / triggered signals
  ➔ only periodic CAN load

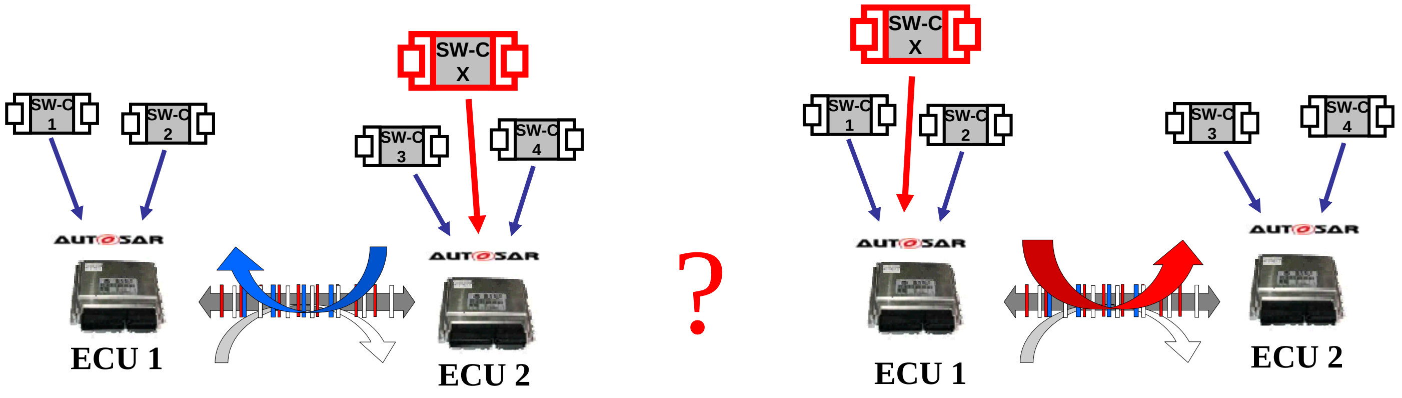


Comparing Dynamic Load Profiles



Architecture Use Cases (today and tomorrow)

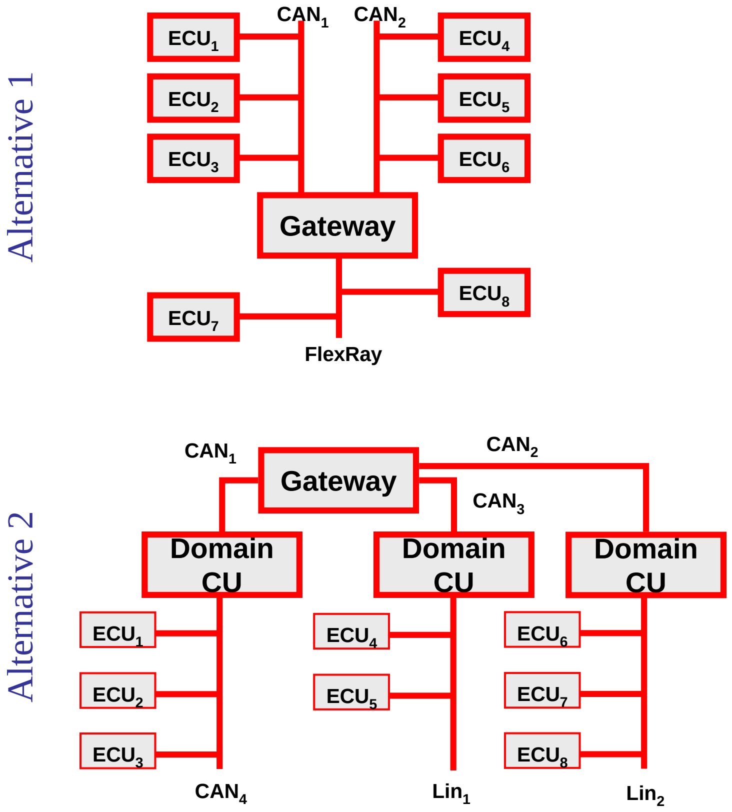
Software Component Integration (in AUTOSAR context)



- □ Evaluate alternatives („what if ?“)
  - □ Mapping
  - □ Scheduling
  - □ Communication

- □ Early
- □ Quickly
- □ Cost-efficient

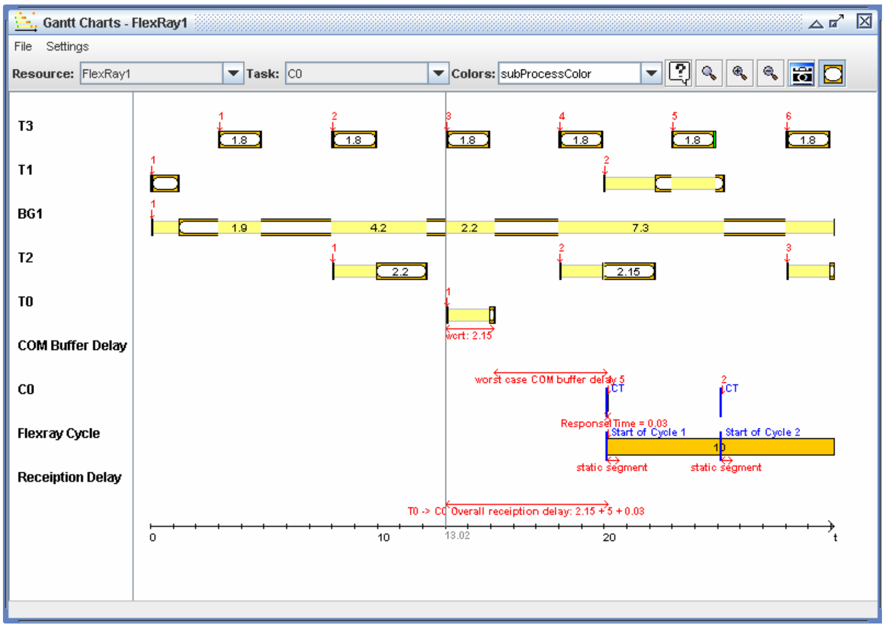
Gated Network Analysis and Optimization



Architecture questions

- □ Topology
- □ Dimensioning of ECUs and buses
- □ Gateway Design
- □ Function → ECU Mapping
- □ Signal → Communication Mapping

End-to-end Timing Analysis is Key



Resource: FlexRay1 Task: C0 Colors: subProcessColor

Application Schedule

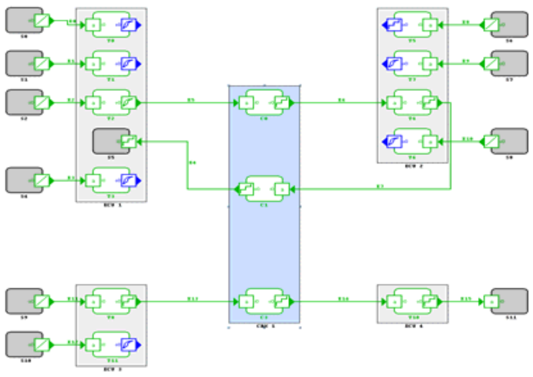
COM-Layer Schedule

Bus Schedule

- ☐ Verification and visualization of end-to-end timing
- ☐ Optimization (adding synthesis capabilities) (work in progress)

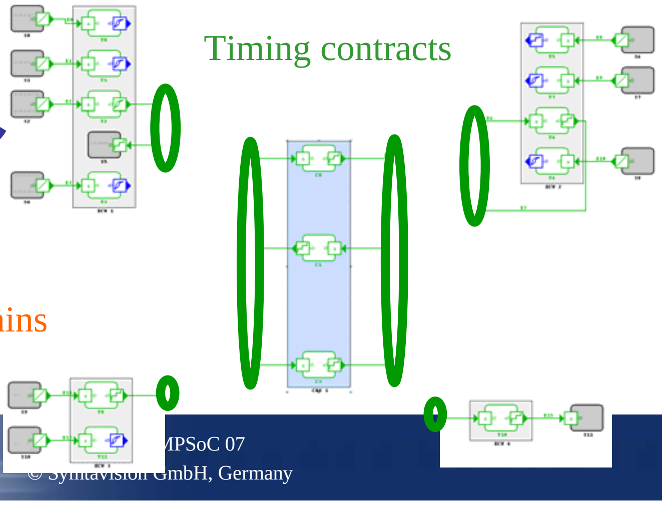
Black-box System Integration

Component Dependencies and Interfaces



- □ global analysis
- □ full knowledge about system behavior

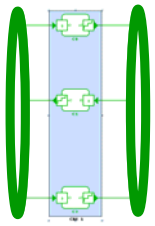
Restricted to components or domains



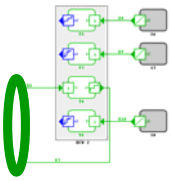
- □ local requirements
- □ local analysis
- □ black box integration
- □ IP protection

Between components or domains

Interfaces for the Supply Chain

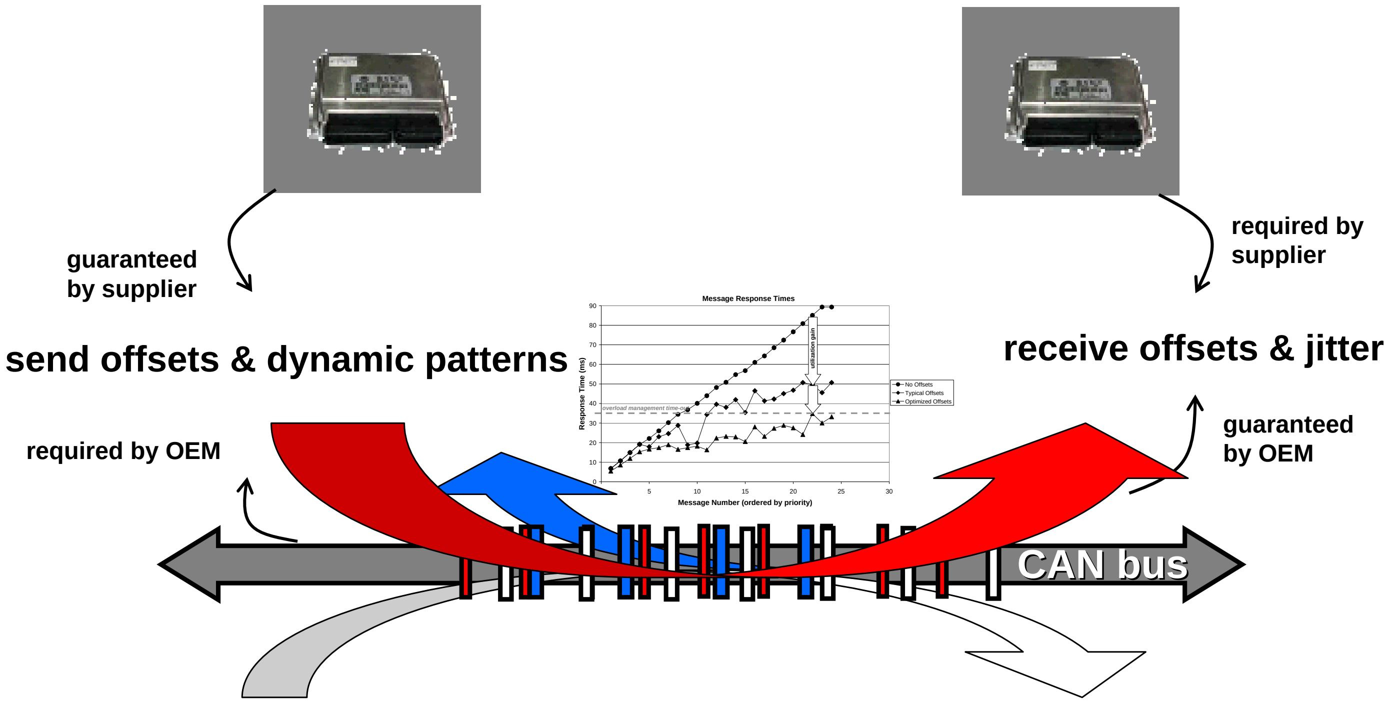


- □ Integrators: distribute the available time, bandwidth and flexibility among the suppliers
  - □ no details about subsystem needed



- □ Suppliers: work locally with available time, bandwidth and flexibility
  - □ optimally fulfill requirements
  - □ no details about „rest of system“ needed

Black-Box integration example (OEM view)



guaranteed by supplier

send offsets & dynamic patterns

required by OEM

required by supplier

receive offsets & jitter

guaranteed by OEM

Contact

Symtavision GmbH
Frankfurter Str. 3b
38122 Braunschweig
Germany

www.symtavision.com

info@symtavision.com
Tel +49 886 179-0

sales@symtavision.com
Tel +49 886 179-25
Fax +49 886 179-29

Formal analysis and optimization of heterogeneous networks in industrial practice

- from networked systems to MpSoC

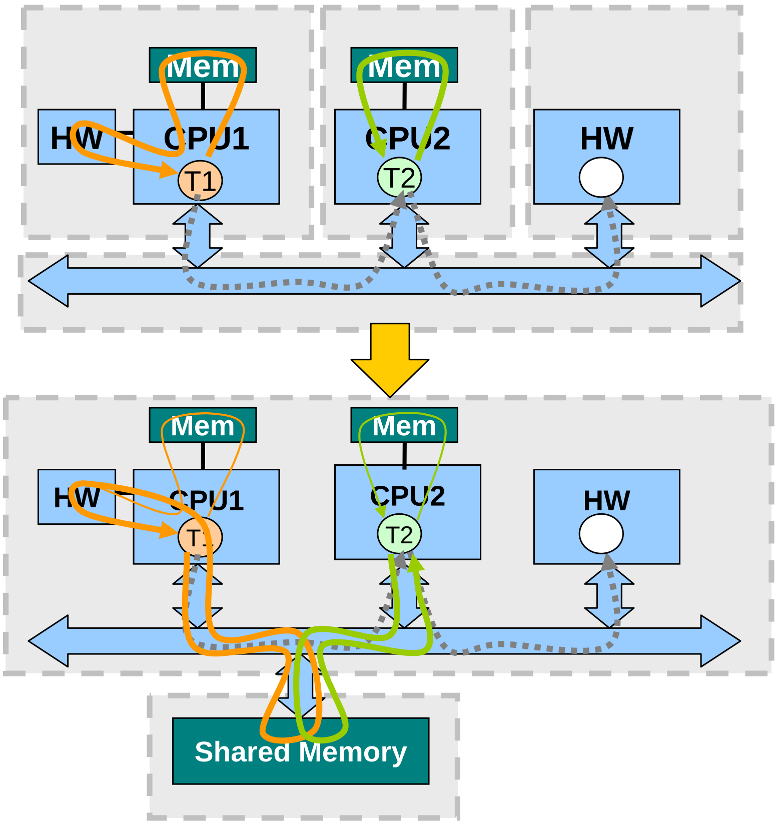
Rolf Ernst, TU Braunschweig

Lessons learned

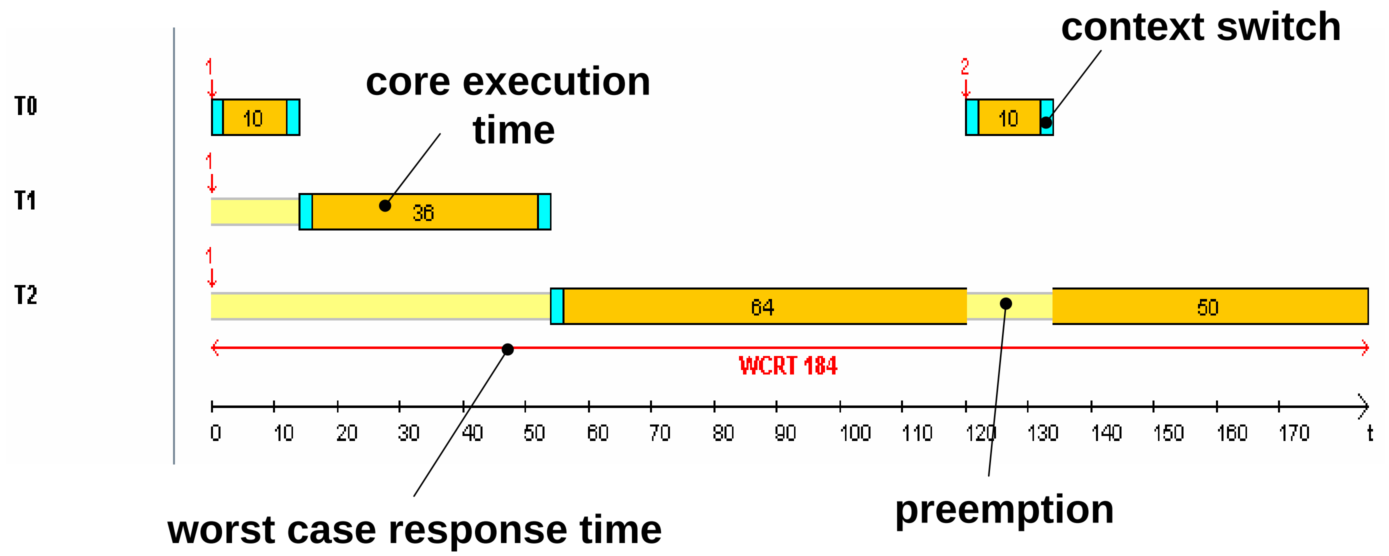
- **formal performance analysis and optimization have proven to be practical in industrial design**
  - **in different phases of a design**
  - **even in very complex architectures and design processes such in automotive electronics with many players**
  - **greatly improving predictability and lowering design risk**
- **main practical challenges are**
  - **acquisition of basic model data**
    - **obtained by a variety of techniques from estimation to tracing to WCET analysis**
  - **design process enhancement**
    - **tool coupling, method coupling**
    - **feasible even in conservative industries**
  - **libraries**
    - **simplified in case of standardization**

But – MpSoC needs different model

- distributed embedded systems (automotive)
  - local computation and memory resources
  - network mainly used for process communication
- MpSoC
  - process communication and global memory accesses are superimposed
  - more complex traffic with feedback to process execution

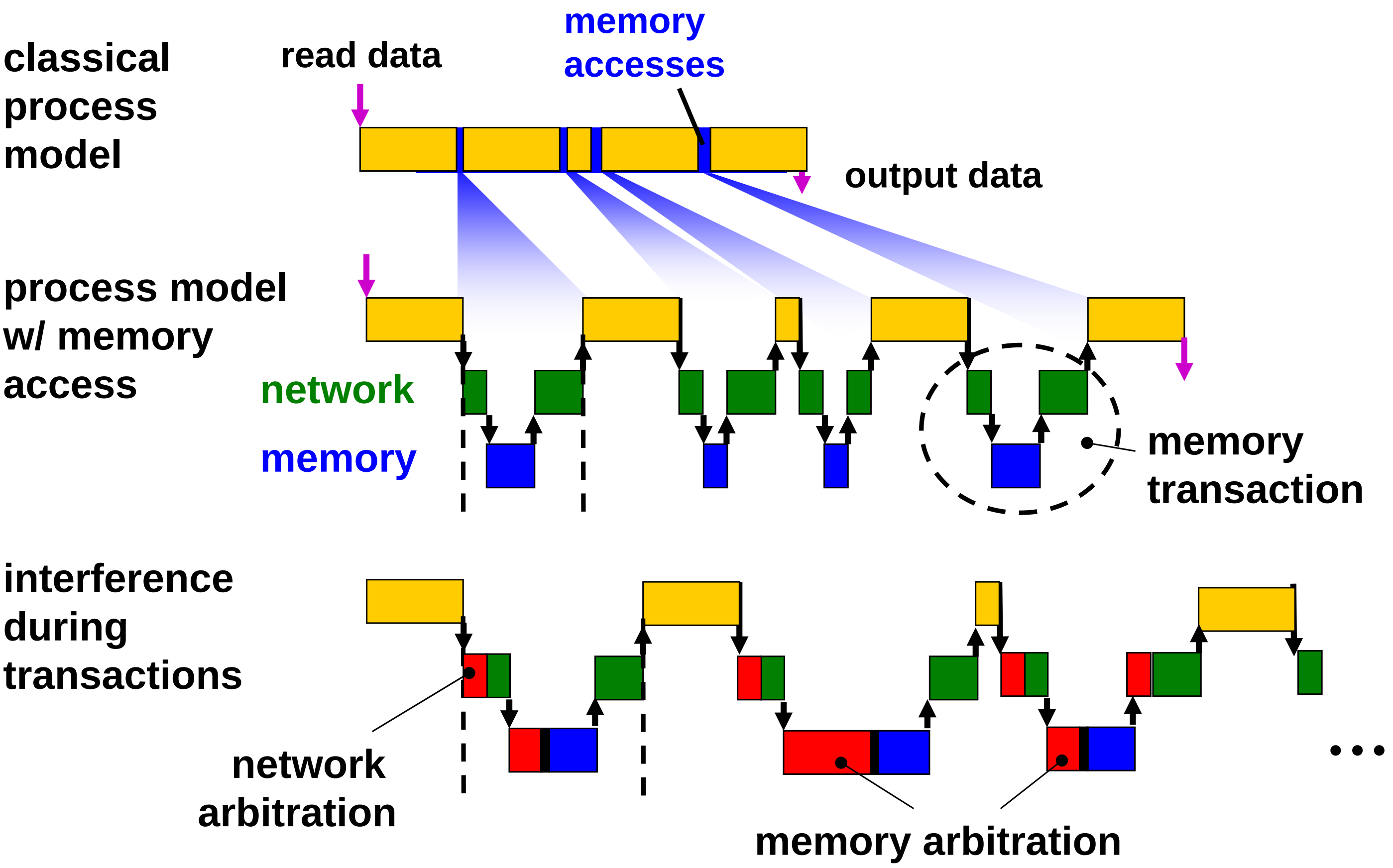


Process timing analysis



- traditional: task „execution“ times are combined to calculate response times
- need to include memory accesses for MPSoCs!

MpSoC process execution



classical process model

read data

memory accesses

output data

process model w/ memory access

network

memory

memory transaction

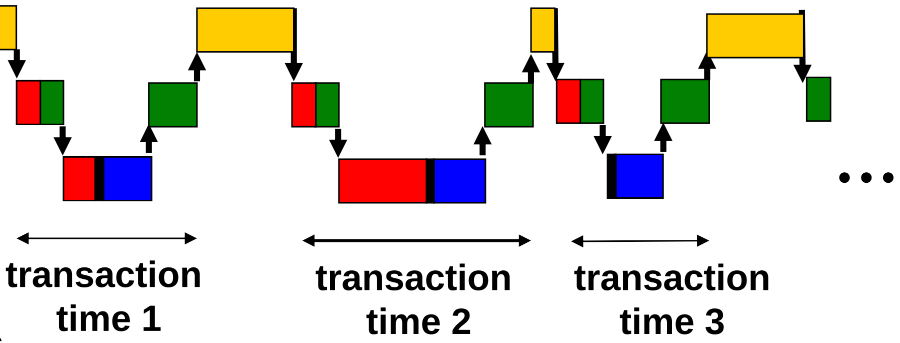
interference during transactions

network arbitration

memory arbitration

MpSoC process performance analysis

interference has highly dynamic behavior!



transaction time 1

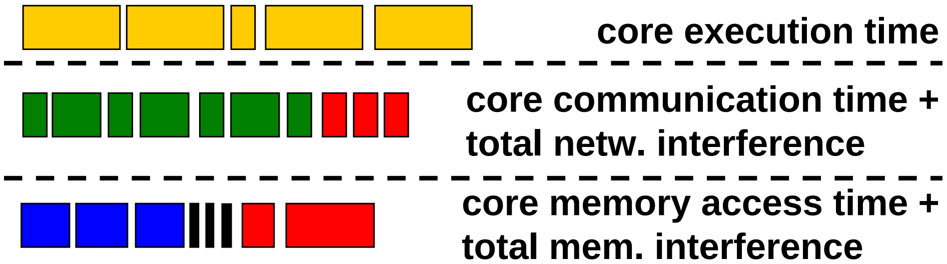
transaction time 2

transaction time 3

assuming worst case transaction is too conservative

Solution:

- derive upper total interference bound using formal analysis
- superimpose and continue with classical analysis



core execution time

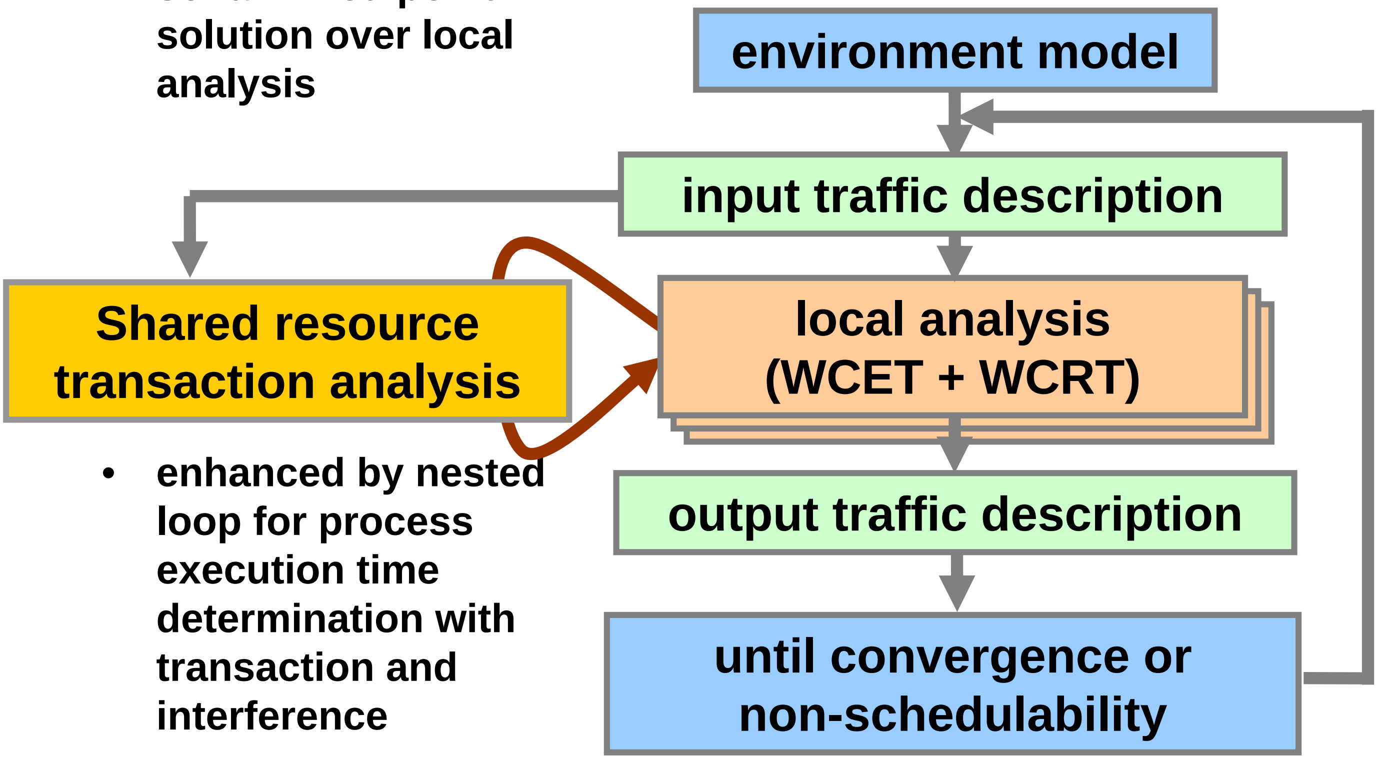
core communication time + total netw. interference

core memory access time + total mem. interference

Enhanced SymTA/S analysis engine

- so far: fixed point solution over local analysis

- enhanced by nested loop for process execution time determination with transaction and interference



environment model

input traffic description

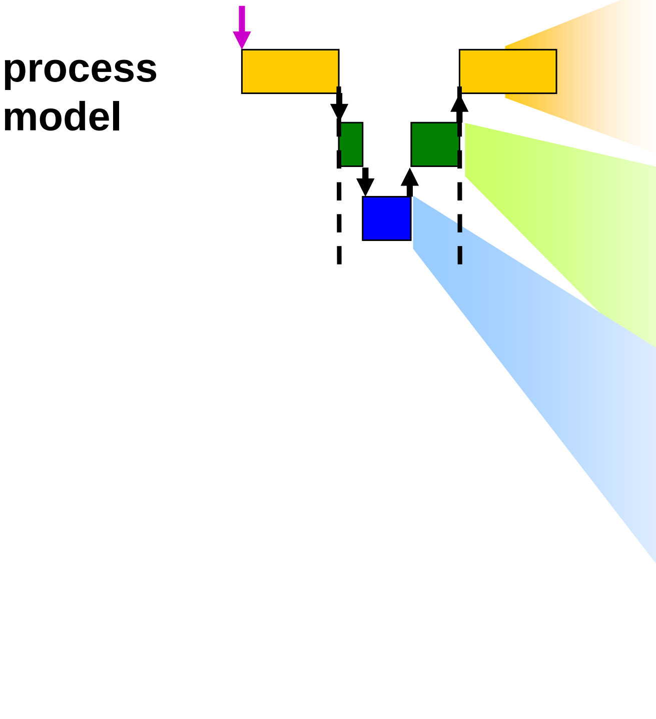
Shared resource transaction analysis

local analysis (WCET + WCRT)

output traffic description

until convergence or non-schedulability

Data needed for enhanced analysis

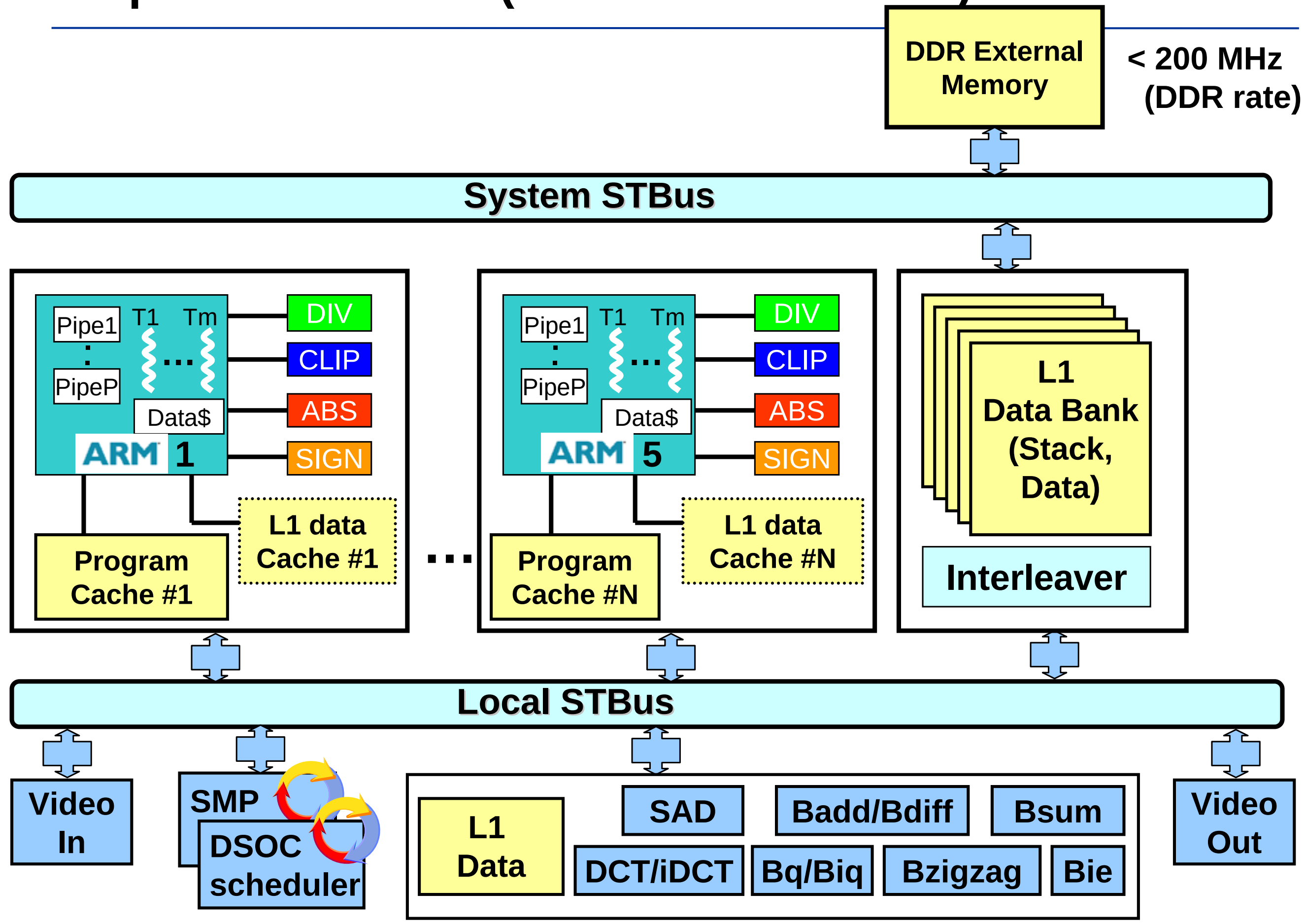


- Task behavior on CPU
  - process execution time
  - number and target of transactions
- Bus / Network
  - latency and arbitration policy
- Shared Memory
  - timing
  - arbitration and scheduling policy
  - similar for coprocessors

How to acquire these data?

- task behavior on CPU
  - execution time
    - simulation-traces
    - WCET analysis where applicable/available
  - transactions
    - simulation traces
    - cache traces
    - task communication analysis where applicable
- bus / network
  - scheduling model
  - formal system level analysis using SymTA/S
- memory
  - word-level timing (from datasheet)
  - arbitration model

Example architecture (STMicroelectronics)



DDR External Memory

< 200 MHz (DDR rate)

System STBus

Pipe1 : PipeP T1 … Tm Data$ ARM 1 DIV CLIP ABS SIGN

Program Cache #1

L1 data Cache #1

…

Pipe1 : PipeP T1 … Tm Data$ ARM 5 DIV CLIP ABS SIGN

Program Cache #N

L1 data Cache #N

L1 Data Bank (Stack, Data)

Interleaver

Local STBus

Video In

SMP

DSOC scheduler

L1 Data

SAD Badd/Bdiff Bsum

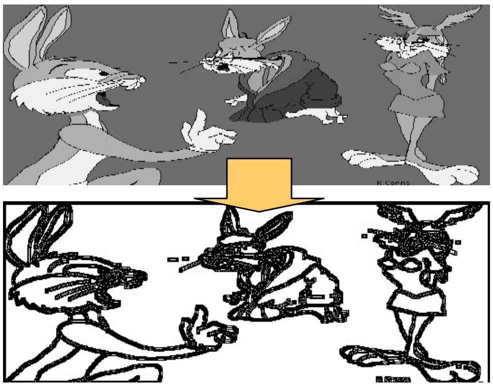
DCT/iDCT Bq/Biq Bzigzag Bie

Video Out

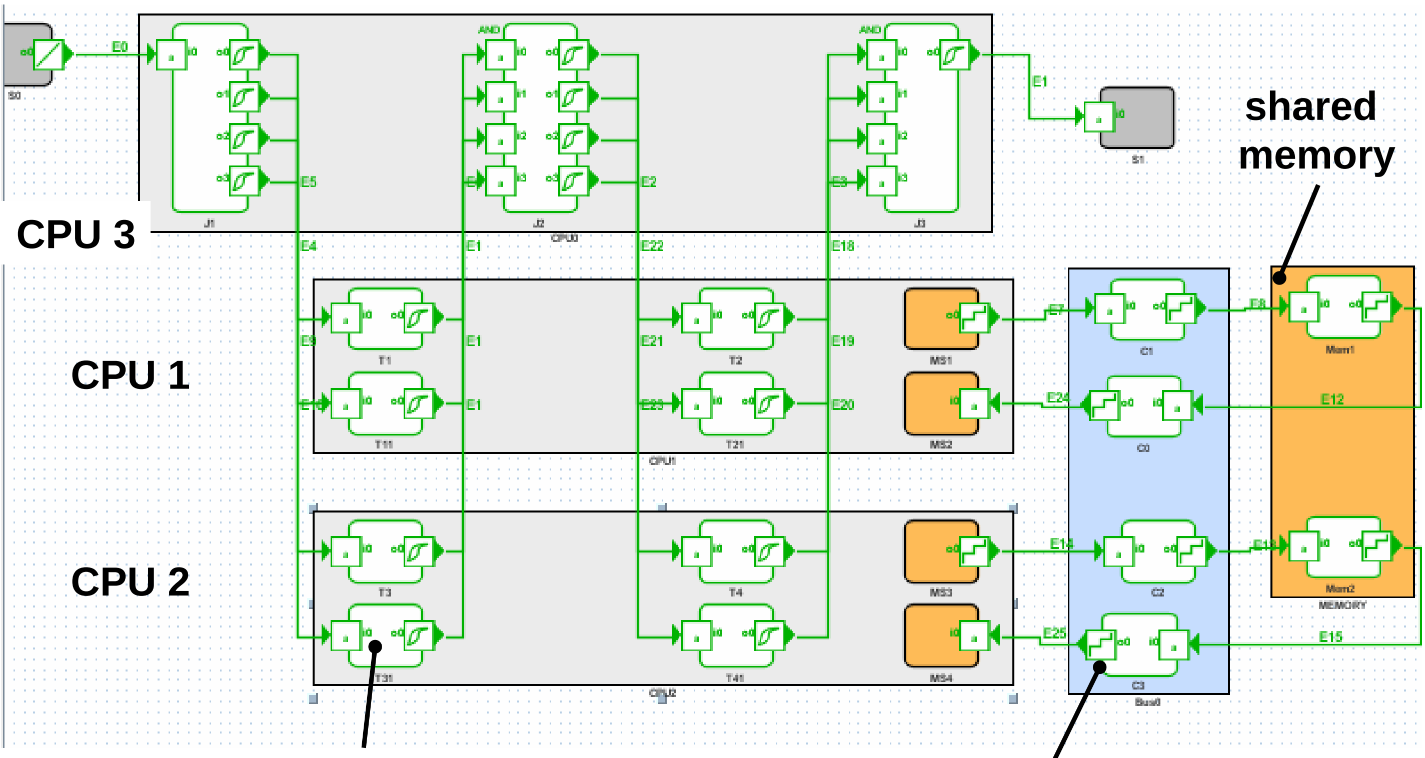
Courtesy: Pierre Paulin

Application: MPEG 4 Contour Detection

- contour detection algorithm from École Polytechnique de Montreal (Gabriela Nicolesu)
- 2 – 4 processor architecture
- 2 threads per processor
  - round-robin scheduling
- StepNP simulator available



Application model



CPU 3

CPU 1

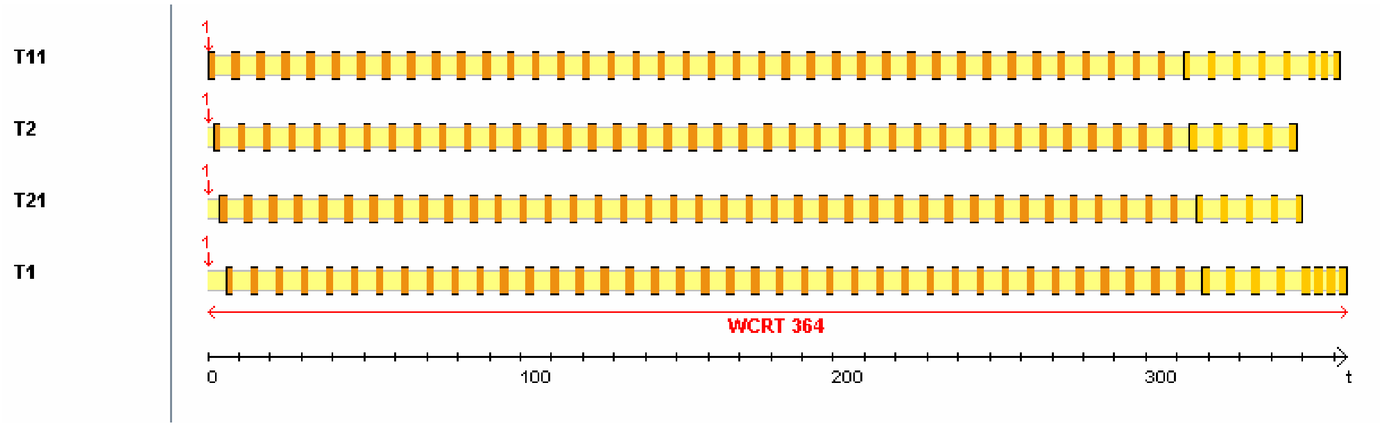
CPU 2

shared memory

task

bus w/ communication tasks

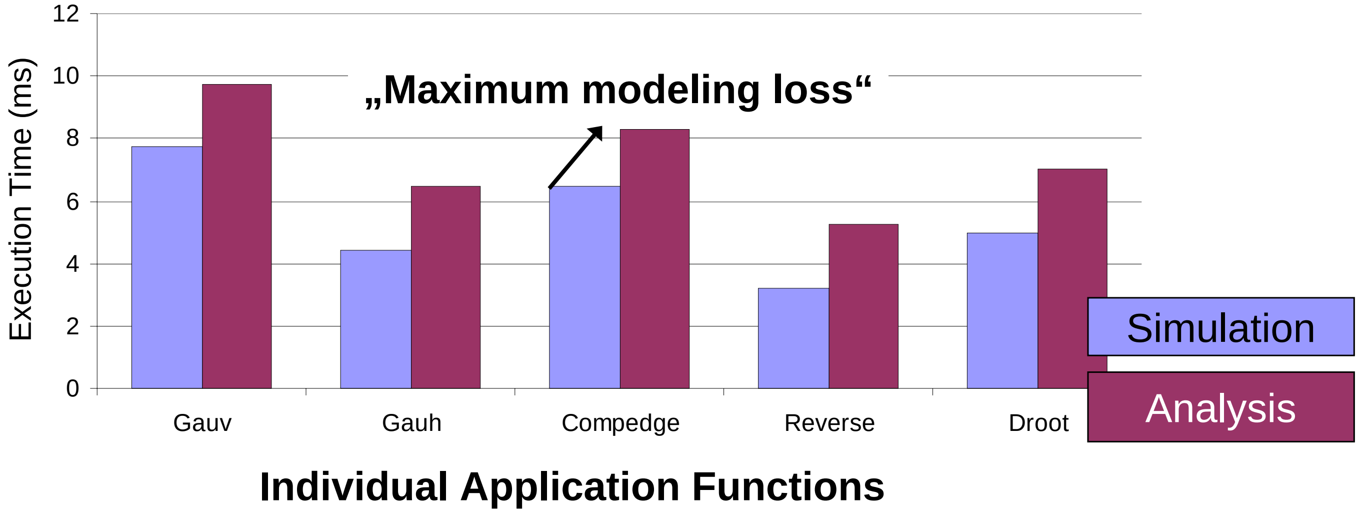
Execution Timing - Example



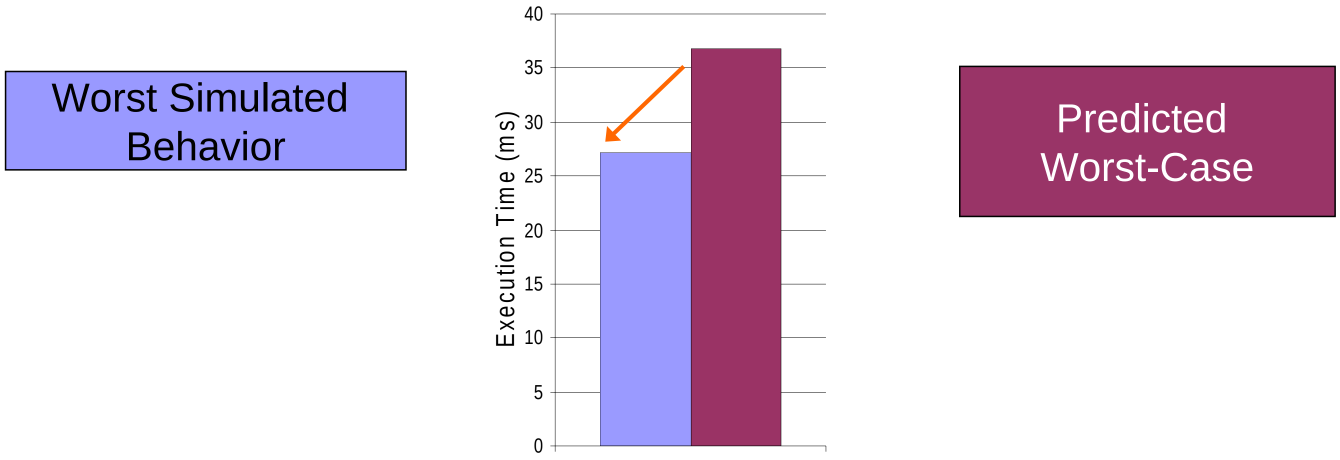
- system worst case task response time for task T1 based on single task simulation data
- shows little worst case interference, bus is sufficient in this simple example

Results of STMicroelectronics Example

- simulation
  - system simulation completely performed with StepNP simulator
- analysis
  - single task simulation
  - SymTA/S analysis based on single task simulation data


„Maximum modeling loss“

System Level Prediction



Concurrent Behavior

- Very fast prediction of worst case behavior considering
  - Bus/Network Congestion (if any)
  - Memory Congestion
  - coprocessors ...
- Possible investigation of processor sharing, degree of parallelism / pipelining, …

Conclusion

- new approaches to formal performance analysis have found their way into industrial practice
- worst case design successfully used for predictable and robust systems integration – supported by tools
- more complex behavior of MpSoC due to conflicting task communication and memory access
- new technique for MpSoC presented and demonstrated with a practical experiment

Acknowledgement - Literature

- more info can be found
  - www.ida.ing.tu-bs.de
- the experiments and part of the slide figures have been provided by Simon Schliecker and Mircea Negrean