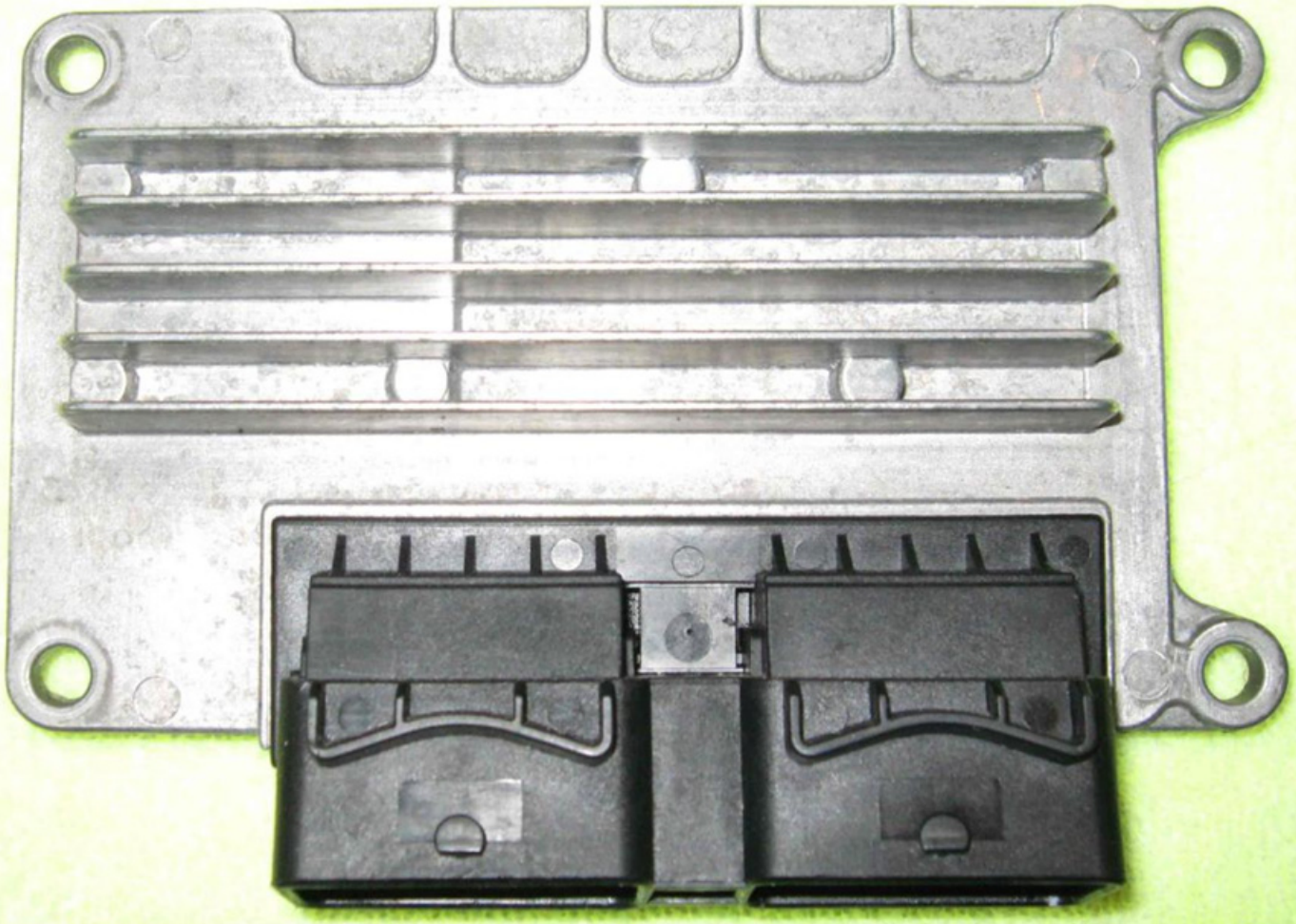
Embedded Health
The Step beyond Debugging

Dr. Albrecht Mayer
Sr. Principal Emulation Systems and Tooling

MAD Workshop, TUM Garching, 2013-11-15



bug

Google Search I'm Feeling Lucky

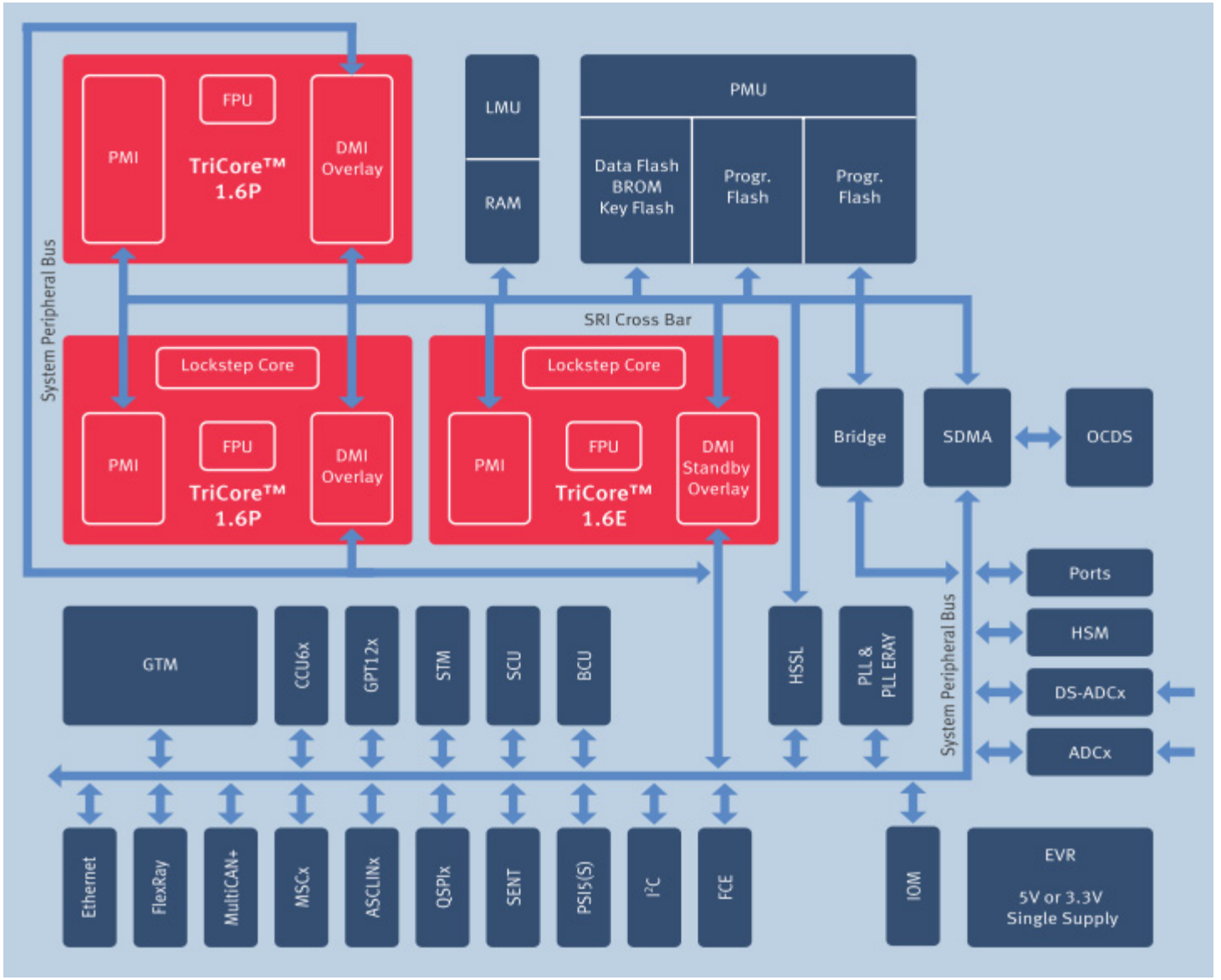
Embedded Health

- Automotive Warm-Up
- Debugging is not enough
- Symptoms and Diseases
- Learning from the Doctor
- Conclusions and Outlook

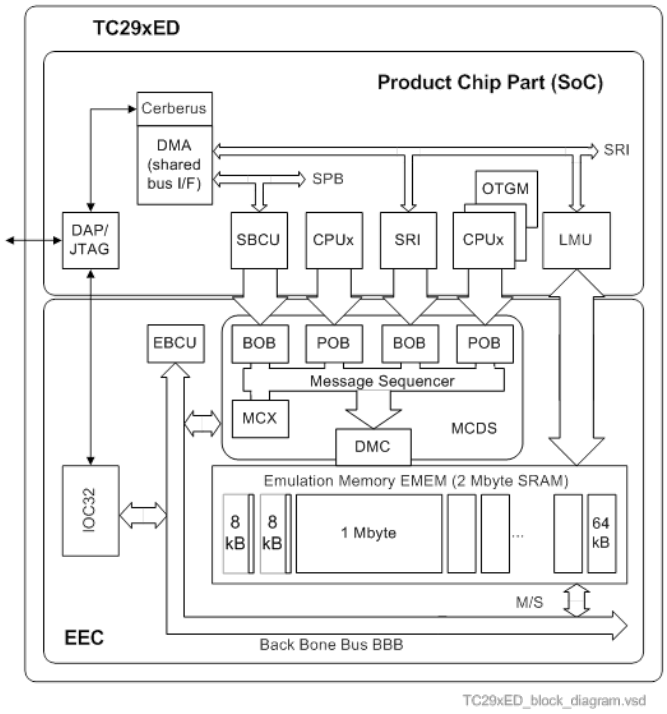
Embedded Health

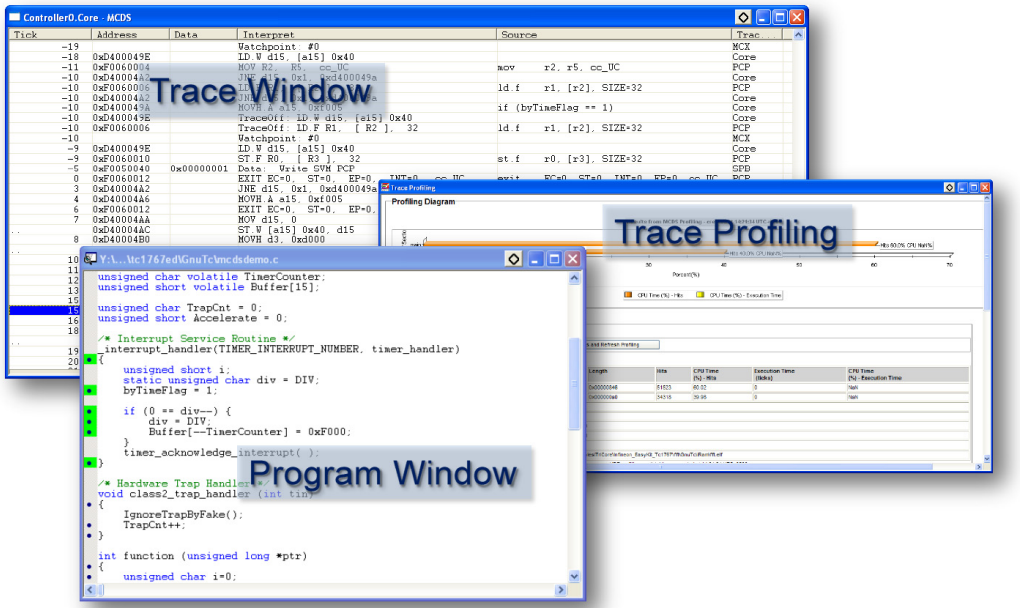
- Automotive Warm-Up
- Debugging is not enough
- Symptoms and Diseases
- Learning from the Doctor
- Conclusions and Outlook

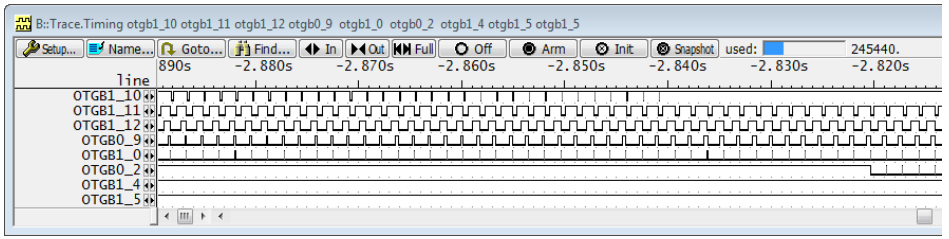
AURIX Automotive Microcontroller Family TC27x Block Diagram



Debugging with DAP2 + ED: Trace at "JTAG tooling" Price Point

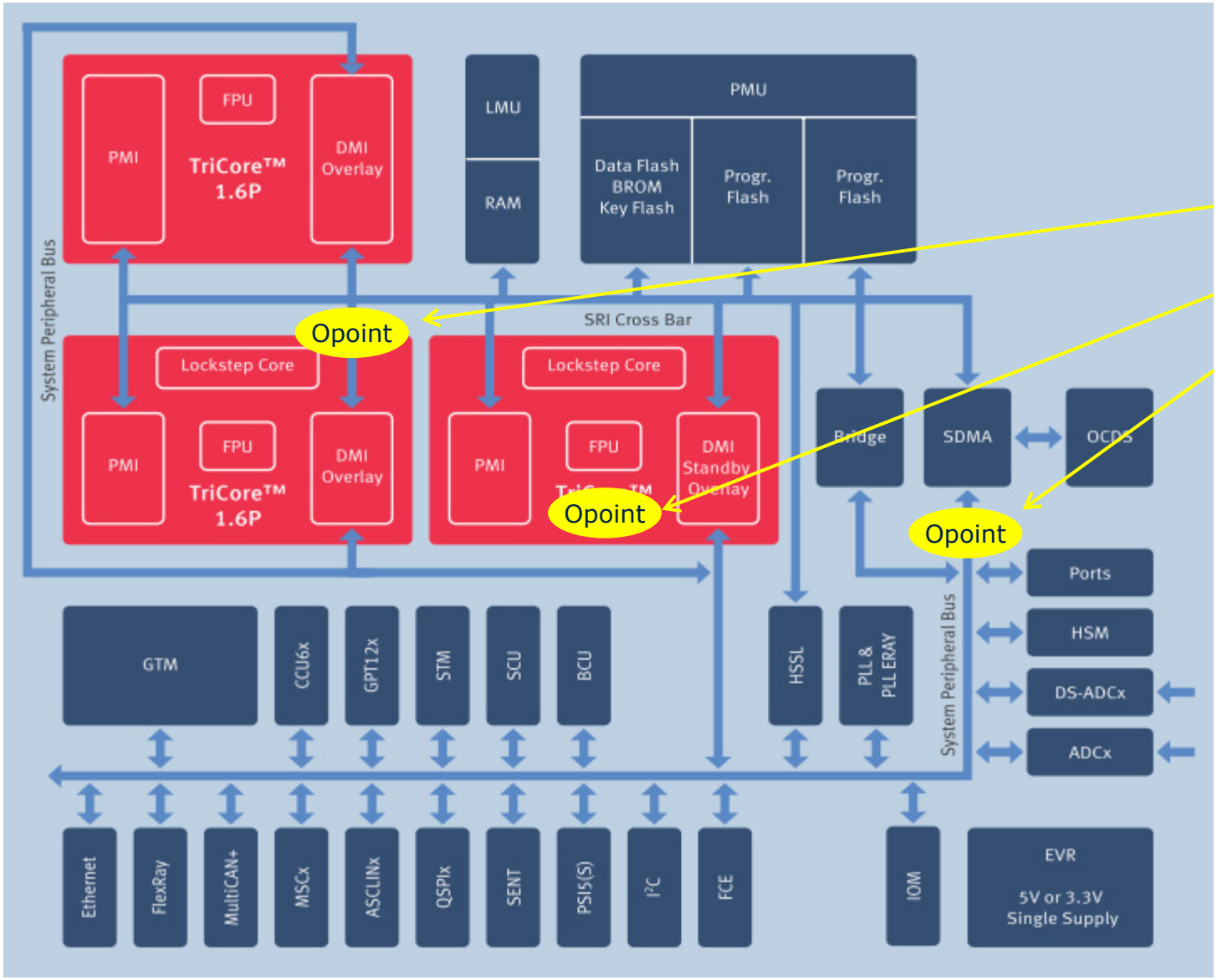






- Just ED (same package) and DAP connector on board
- New: logic analyzer functionality for peripherals and pins
- New: continuous Compact Function Trace (CFT) via DAP2
- **Trace tooling on every engineer's desk**

TC27x Block Diagram



Observation Points

SRI_CPU1

CPU0

SPB

used for example on next slide

MCDS Trace Example: Unaligned 64 bit write CPU0 to DSPR CPU1

TimeR	Ticks	Opoint	Origin	Data	Operation	Address
0	0	CPU0	CPU0	89AB	W 16	60000000
3	3	SRI_CPU1	CPU0.DMI	89AB	W 16 SV	60000000
6	3	CPU0	CPU0	123456780EF0CD00	W 64	60000002
7	1	CPU0	CPU0	00000001	W 32	D0009BFC
9	2	SRI_CPU1	CPU0.DMI	CD00	W 16 SV	60000002
13	4	CPU0	CPU0	80208020	W 32	F003D318
13	0	SRI_CPU1	CPU0.DMI	56780EF0	W 32 SV	60000004
17	4	CPU0	CPU0	1234567800000010	W 64	60000002
17	0	SRI_CPU1	CPU0.DMI	1234	W 16 SV	60000008
22	5	SPB	CPU0.DMI	80208020	W 32 SV	F003D318
26	4	SRI_CPU1	CPU0.DMI	0010	W 16 SV	60000002

- **Parallel trace at several observation points**
- **Very accurate time information with ticks**

Warm-up comments

State of the art:

- Can generate huge amounts of trace data
- Moore’s law + multicore means ever more trace data
- Can look at very fine grained details

The problem:

- Debugging done by humans
- Size of brains is not growing

Multicore, MCDS, ED + DAP: Trace tooling for every engineer everywhere

Embedded Health

- Automotive Warm-Up
- Debugging is not enough
- Symptoms and Diseases
- Learning from the Doctor
- Conclusions and Outlook

Complex Systems

There is no healthy human.

… only not enough examinated ones.

Why debugging falls short

- A bug has a symptom (or is hidden)
- A symptom means something is wrong

- Can you always distinguish between right or wrong?
- Could you make your system bug free if you try hard?

- Not sick/injured?
- Healthy?

Example: Can you always distinguish between right or wrong?

- Automotive pedestrian recognition system with camera:

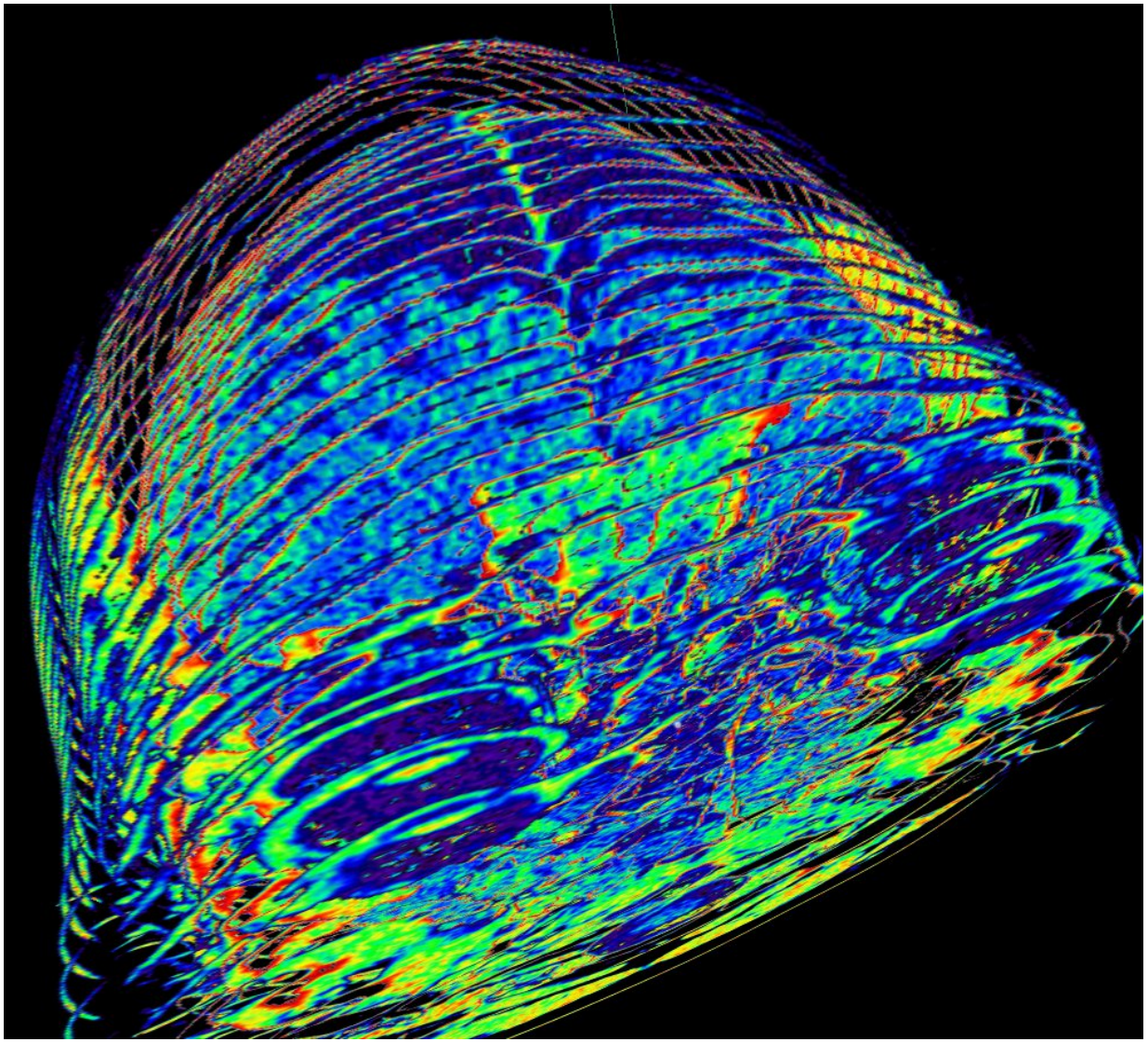
Potential failures:

- A pedestrian is not recognized
  - Fortunately the driver breaks
  - False negative
- E.g. a plastic bag in the wind is recognized as pedestrian
  - Car automatically breaks and car behind crashes
  - False positive

Embedded Health

- Automotive Warm-Up
- Debugging is not enough
- **Symptoms and Diseases**
- Learning from the Doctor
- Conclusions and Outlook

Do you see the bug?



Diseases can have no visible symptoms

- "Healthy" human has cancer
- When will he or she go to the doctor?

When there is a symptom

– e.g. a very costly field return

So earlier would be better

– e.g. in development phase

What would help:

- Examination is free and takes no time
- Examination is scheduled/done automatically

Nothing has happened (yet)

How to find bugs without a symptom!

- Debug is only done when something has happened
- How close are parts of your system to the cliff edge?
- Could you see all those parts if you would look?
- Do you know where to look?

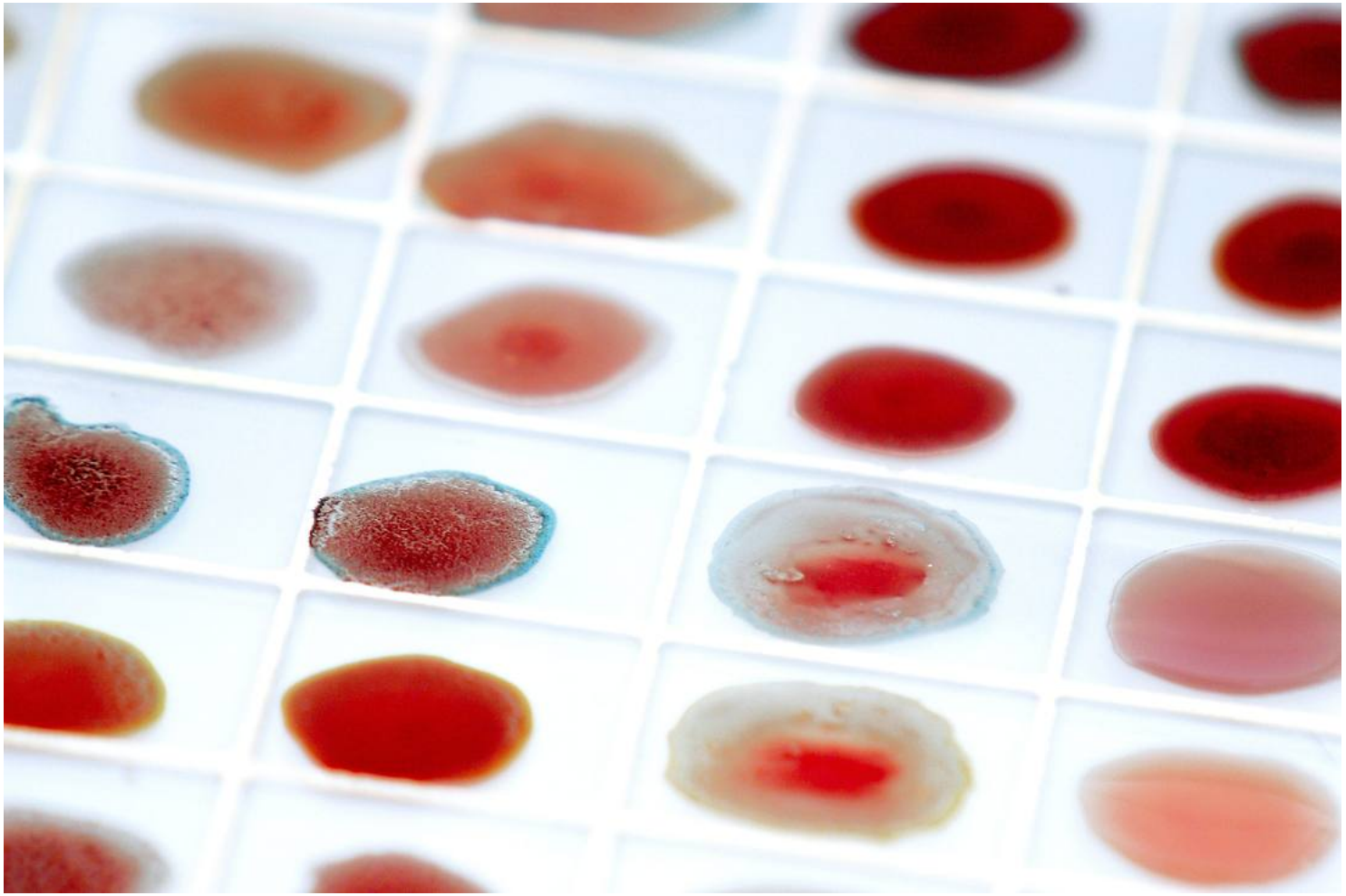
→ Need automatic detection of critical spots

Embedded Health

- Automotive Warm-Up
- Debugging is not enough
- Symptoms and Diseases
- **Learning from the Doctor**
- Conclusions and Outlook

Complex System Debug

Blood Test



Standard KPI Measurement (“Blood Test”)

Basic Idea

- Different, standardized KPIs:
  - General (IPC, cache hits/misses, bus conflicts, etc.)
  - Specific (e.g. function execution times)
- Green/yellow/red classification
- Information can be e.g. in trend for different builds

Advantages

- Can be highly automated
- On module and system level
- Automated meta analysis (e.g. SW module characteristics are different for different systems)

Standardize KPIs

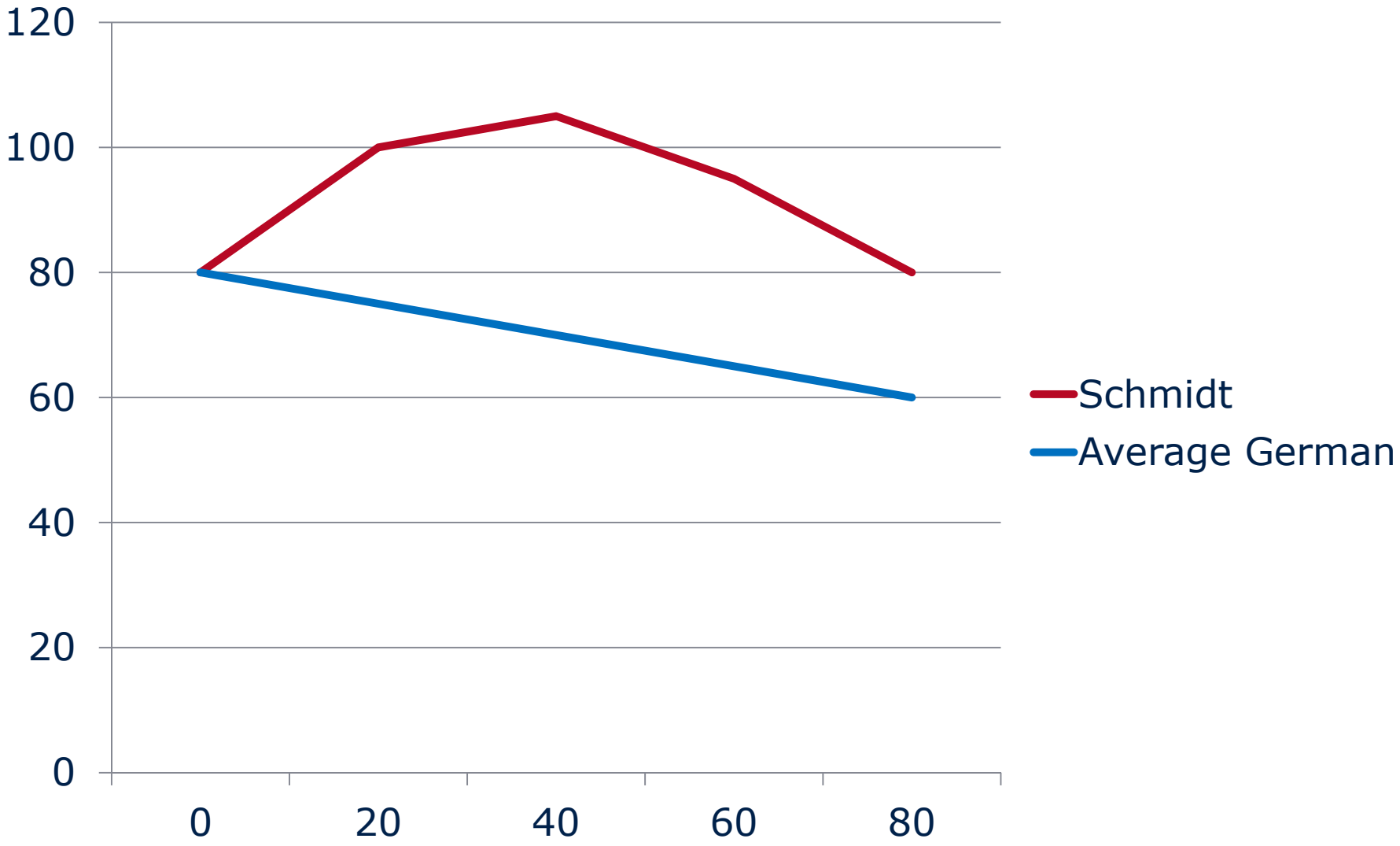
- What is measured?
- How is it measured?
- How is it represented?

- Balance cost and benefit of indicators

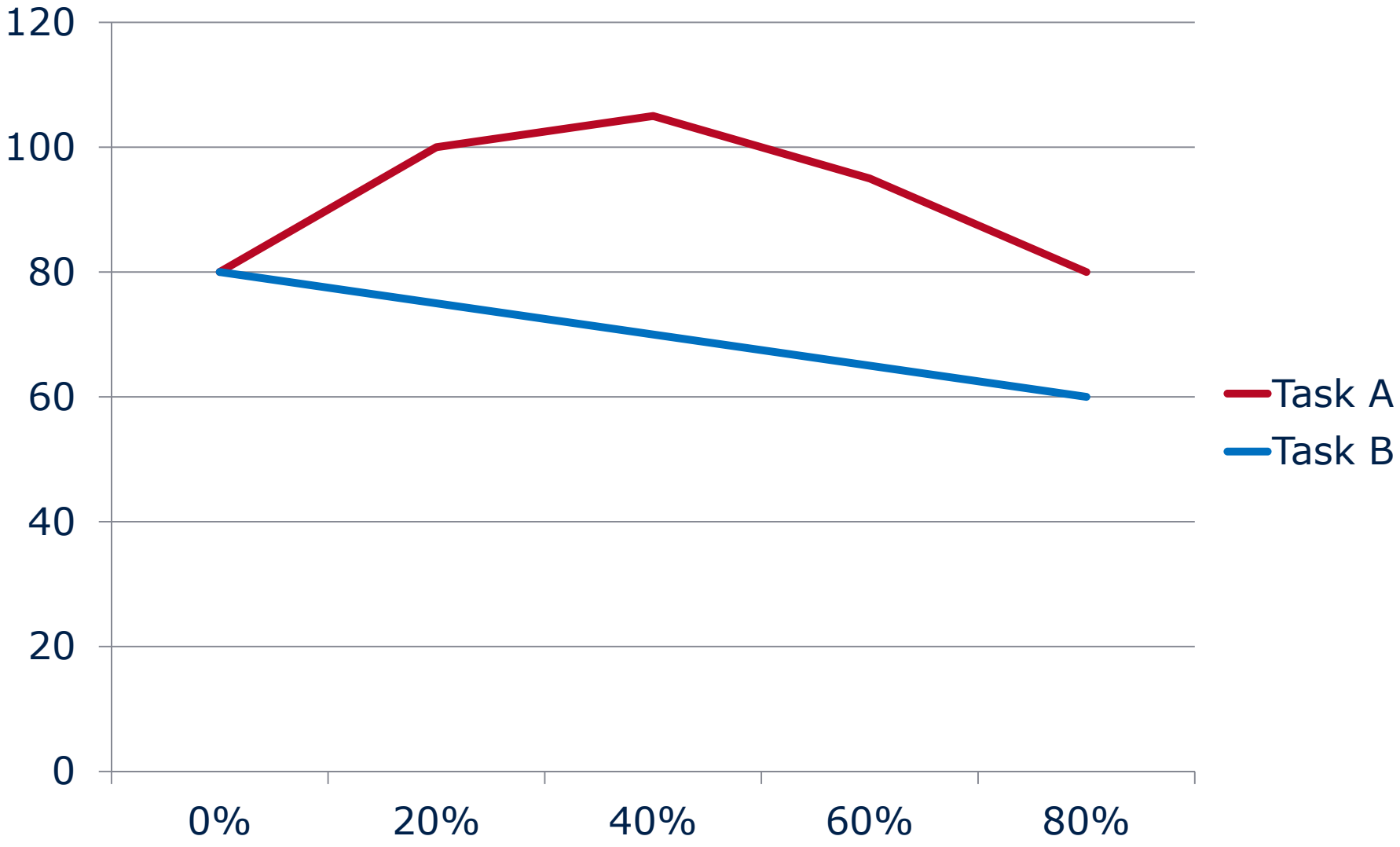
- Requires collaboration of
  - System vendors
  - Chip/IP vendors
  - Tool vendors
  - OS vendors

His lifespan could be more than 100 years … without smoking

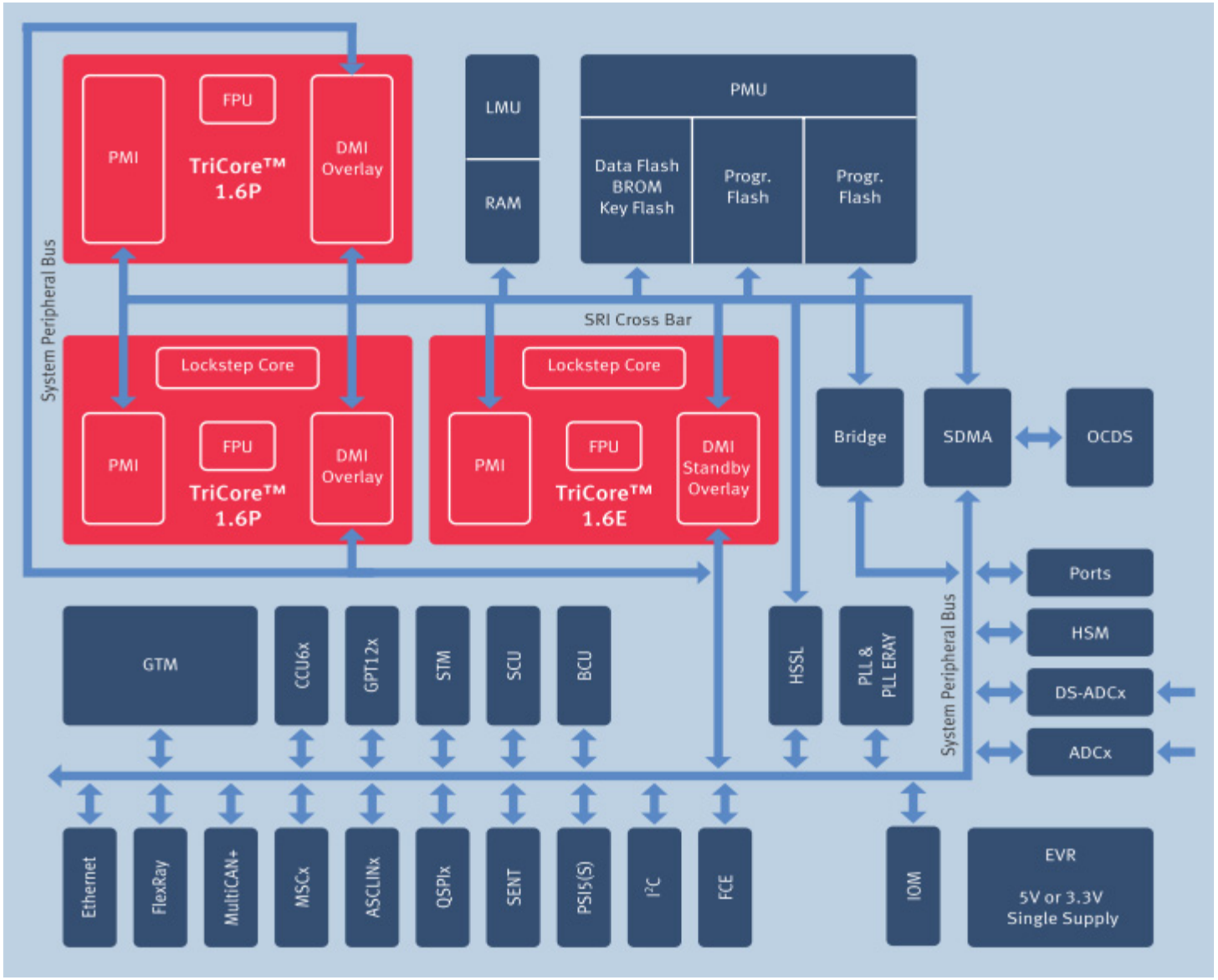
Sensitivity analysis
Live span over number of cigarettes/day



Sensitivity analysis
Remaining time to task deadline over bus load



AURIX Automotive Microcontroller Family TC27x Block Diagram



What's the point?

Standardized measurement:

- Automatically collect data
- Automatic, standardized stimulation
- Automatically detect anomalies

Fitness/Health Check

Measure KPIs of a system or SW component with stress like

- Artificial bus load
- Artificial interrupt load
- Accesses to shared resources by other masters

Sensitivity analysis

- E.g. execution time dependency on foreign accesses to shared RAM

Meta analysis

- Compare expected KPI values with real values

All this can be done automatically!

Embedded Health

- Automotive Warm-Up
- Debugging is not enough
- Symptoms and Diseases
- Learning from the Doctor
- Conclusions and Outlook

Conclusions

- Complex systems have bugs and diseases
- Some of them will have no symptoms
- Standardized, automated examinations needed
- Automatic highlighting of anomalies
- No hassle and low cost solutions
  … enable broad and early usage

Buugle

sporadic bug

Searched 34695509 LOC – not found

Further Reading

- Albrecht Mayer, “What Debug and Medicine have in Common”, http://chipdesignmag.com/display.php?articleId=5004

ENERGY EFFICIENCY
MOBILITY
SECURITY

Innovative semiconductor solutions for energy efficiency, mobility and security.