

Moderne Software Debugging

Af Henrik Andersen
Nohau Danmark A/S

27 og 28 oktober 2010

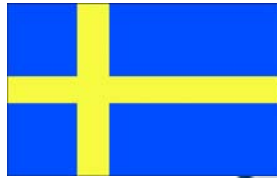
Lidt om Nohau



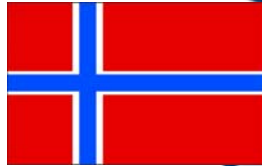
**Scandinavia's leading supplier of
professional tools for software
systems developers**

*"we make sure you succeed with
your application"*

Nohau in Scandinavia



- Since 1981
- Offices in Stockholm and Malmö
- Support center for all Scandinavia
- 25 people



- Since 2008
- Sales office for Norway
- 1 person

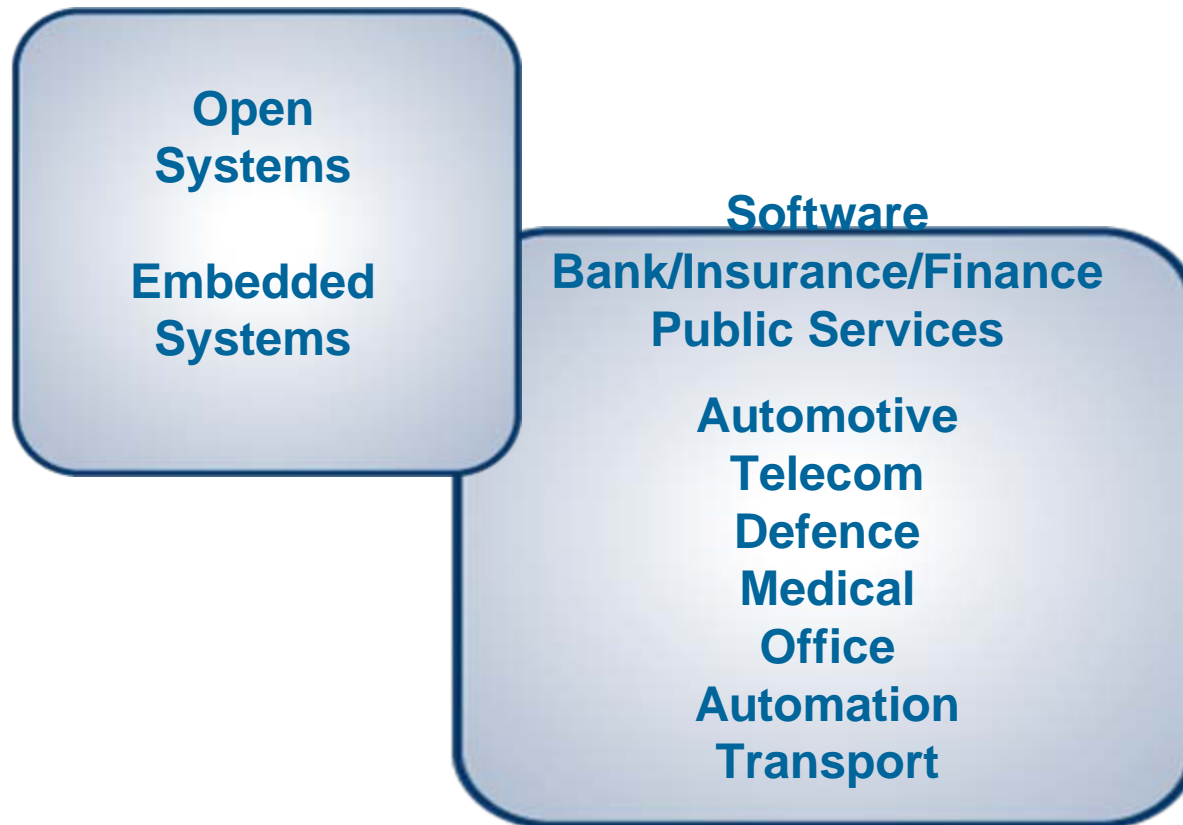


- Since 1991
- Sales office for Denmark
- 5 people



- Since 2004
- Sales office for Finland
- 2 people

Our markets



Some customers...

Automotive

Autoliv
SAAB Automobile
Scania
Volvo Cars
Volvo Truck

Defence & Aero

BAE
Hägglunds
Kongsberg
Patria
SAAB Avitronics
SAAB Bofors Dynamics
SAAB Space
Terma
Systematic

Medical

Radiometer
Gambro
GE Healthcare
Maquet
Siemens Elema

Telecom

Polycom
RTX Telecom
CISCO
Ericsson
Infinion
Motorola
Nokia
Telecom
Telia Sonera

Office / Consumer

Anoto
Axis
Bang & Olufsen
Hasselblad
ScanCoin

Administrative IT

Post Danmark
Posten Sverige
BRF
F-Secure
SDC
StreamServe
TDC

Industry

ABB
Alfa-Laval
Bombardier
Danfoss
ESAB
Grundfos
Kone
MAN Diesel
Skov
Siemens Flow
Siemens Windpower
Instruments
TetraPak
Mita Teknik

Some of our suppliers



**Ny
kurskatalog ute**

**Massor
av nyheter**

**We do training
for more than
2000 engineers
and
programmers
every year**

Agenda

- Overblik
 - Debugging teknologier
 - Simulatorer
 - Printf-debugging
 - Monitorer
 - In-circuit emulatorer
 - In-circuit debuggere
 - Statisk kodeanalyse

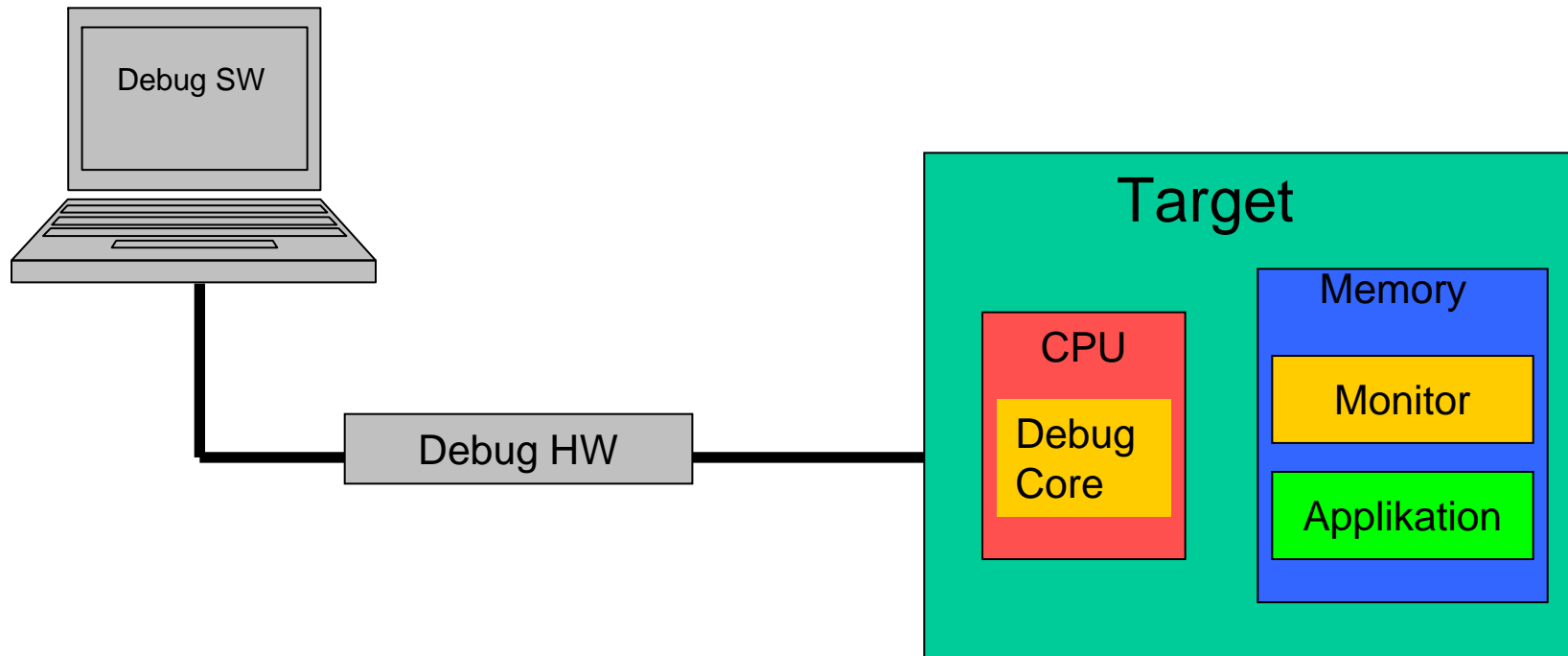
Agenda fortsat

- Avanceret debugging funktionalitet –DEMO
 - Overvågning af variabler i realtid
 - Data-breakpoints
 - Task specifikke breakpoints
 - Performance analyse
 - Realtids trace af programeksekering og data acces
 - Backstepping og genskabelse af konteksten (Context tracking)
 - Korrelation af programafvikling med effektforbrug og HW-logikanalyse
 - Memory allokeringanalyse
 - Detektering af "ikke funktionelle fejl" Statisk analyse
 - Check af kodningregler

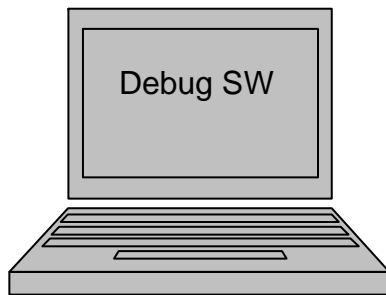
Debugging teknologier

- TEST: Validerering af softwaren op mod de stillede krav og verificering af den ønskede funktionalitet
- Debugging: Den aktivitet hvor vi finder årsagerne til de i testen fundne fejl.

Debugging teknologier



Debugging teknologier Simulatorer



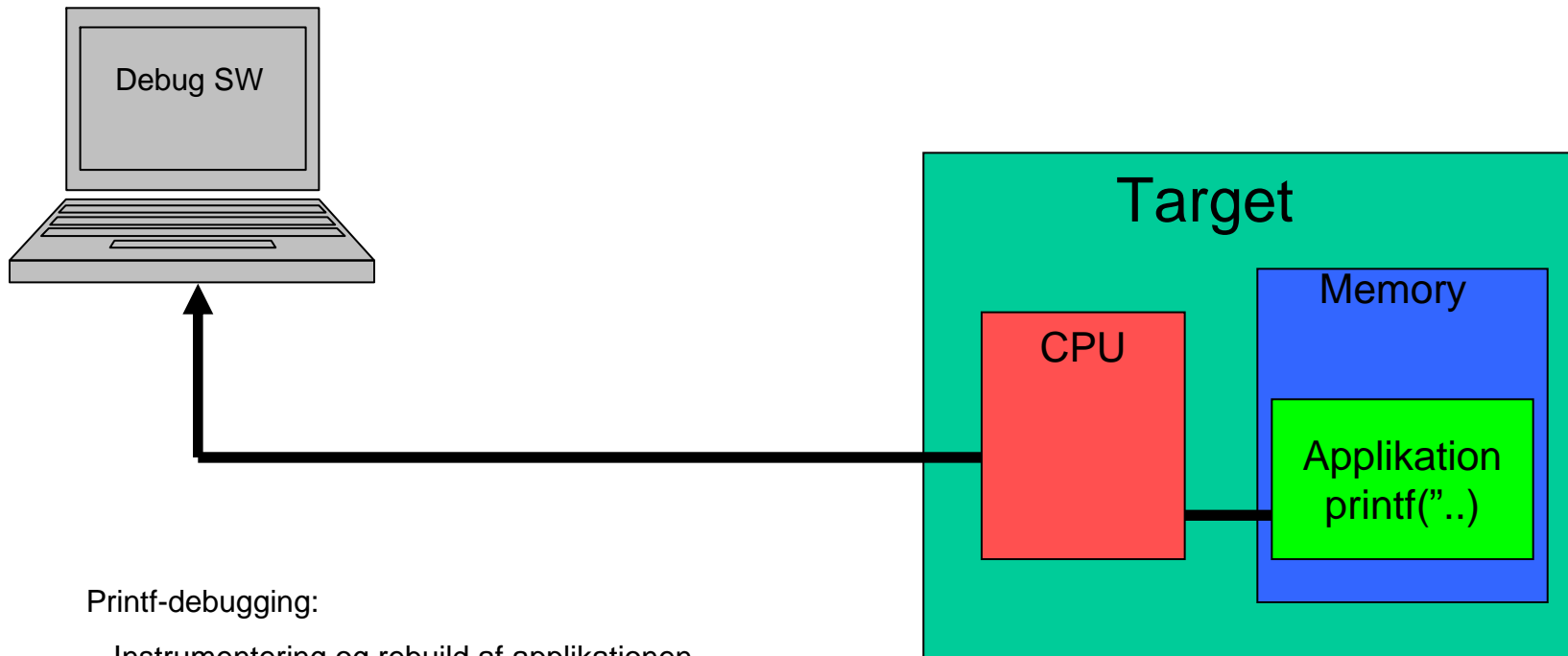
Simulator:

- Kræver intet Target (PC-applikation)
- Simulering af instruktionssættet
- Evt. simulering af periferi og interruptsystem

Debug software:

- Læse/skrive datamemory
- Disassemblere program memory
- "Run Control"
- High Level symbolsk debugging

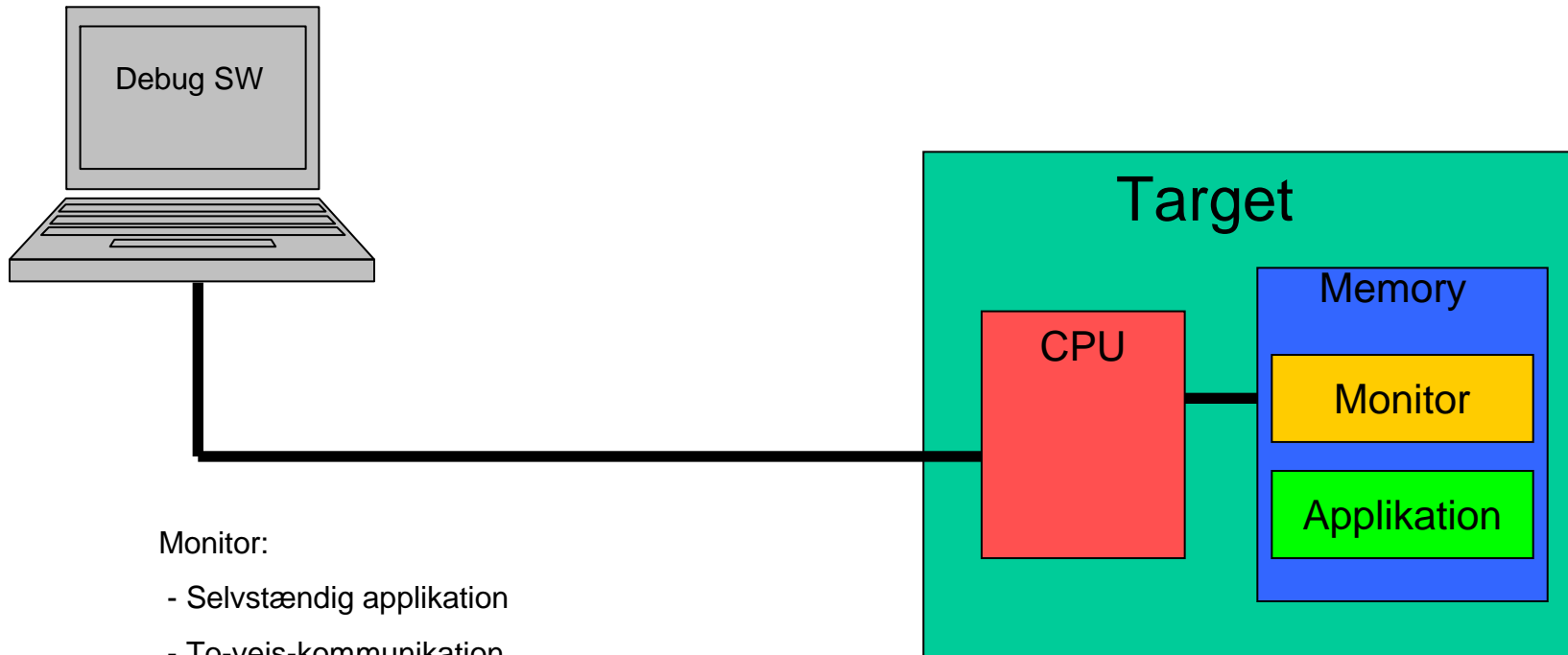
Debugging teknologier Printf



Printf-debugging:

- Instrumentering og rebuild af applikationen
- énjevs kommunikation fra applikation
- Logning af selekteret data i realtid
- "hjemmelavet" Debug-SW
- Kræver Com-port eller lignende m/ driver

Debugging teknologier Monitorer

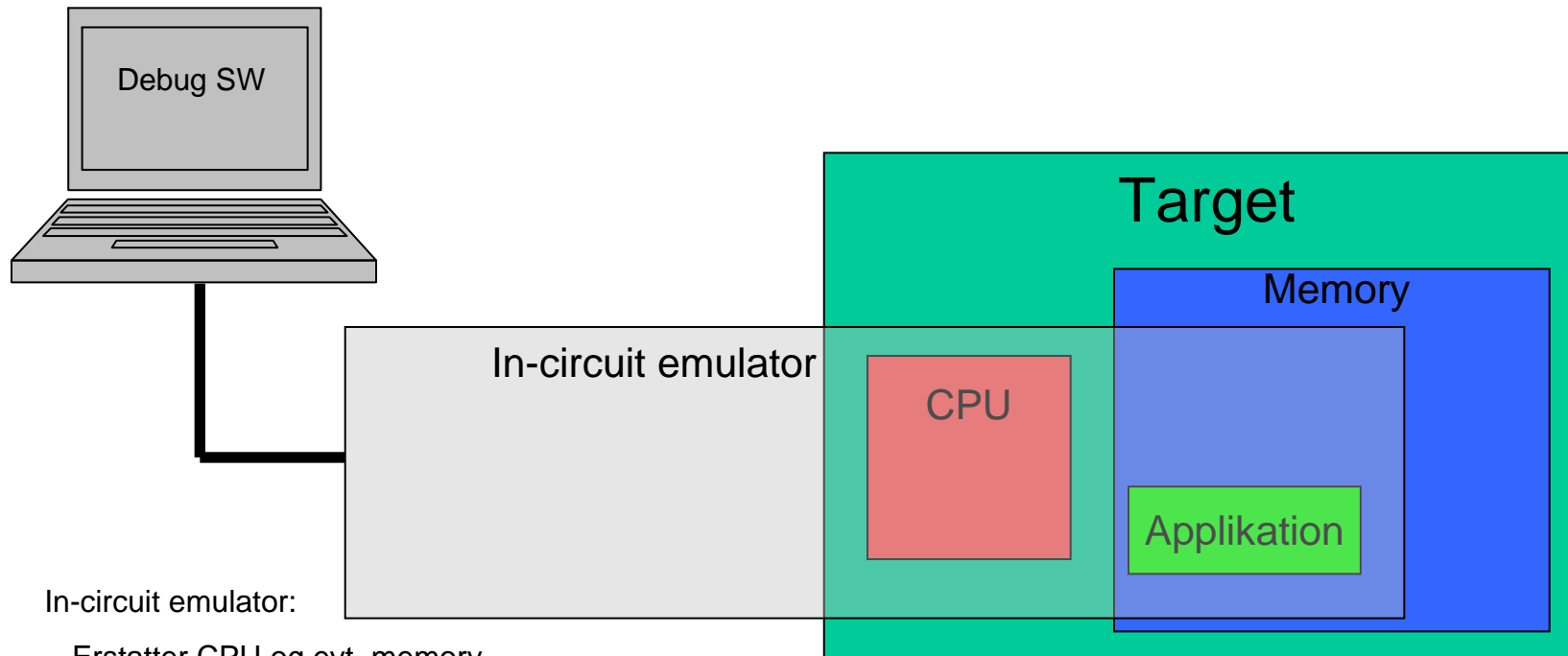


Monitor:

- Selvstændig applikation
- To-vejs-kommunikation
- HLL-debugging med Debug SW mulig
- ingen realtids memory access
- Breakpoints kræver applikation i RAM
- Manuel break kræver NMI
- Kræver Com-port eller lignende

m/driver

Debugging teknologier In-circuit emulatorer

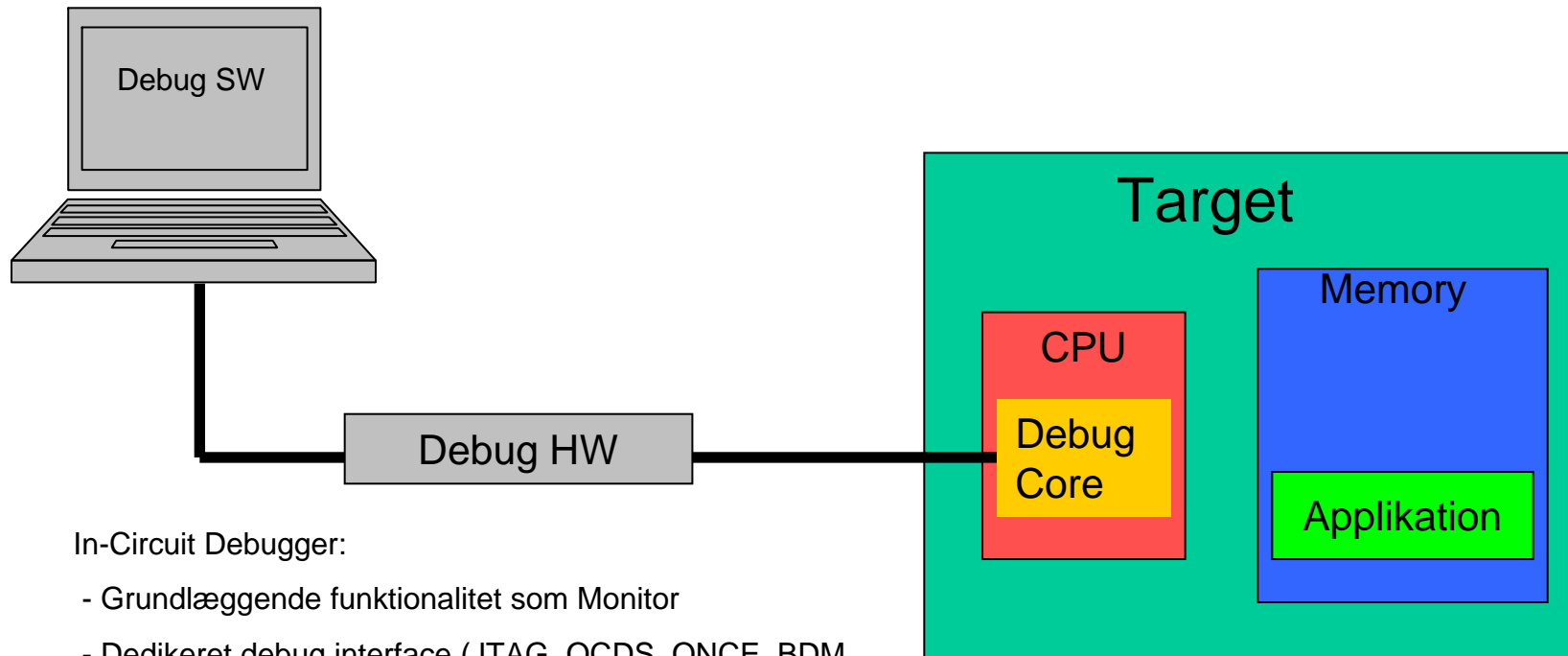


In-circuit emulator:

- Erstatte CPU og evt. memory
- To-vejs-kommunikation
- HLL-debugging med Debug SW mulig
- Realtids memory access mulig
- Breakpoints implementeret i HW
- Realtids trace mulig
- Baseret på speciel Bond-out controller
- Avancerede HW-triggere
- non-intrusive debugging
- Dyr
- Adaptering et problem

Debugging teknologier

In-circuit debugger (JTAG etc.)



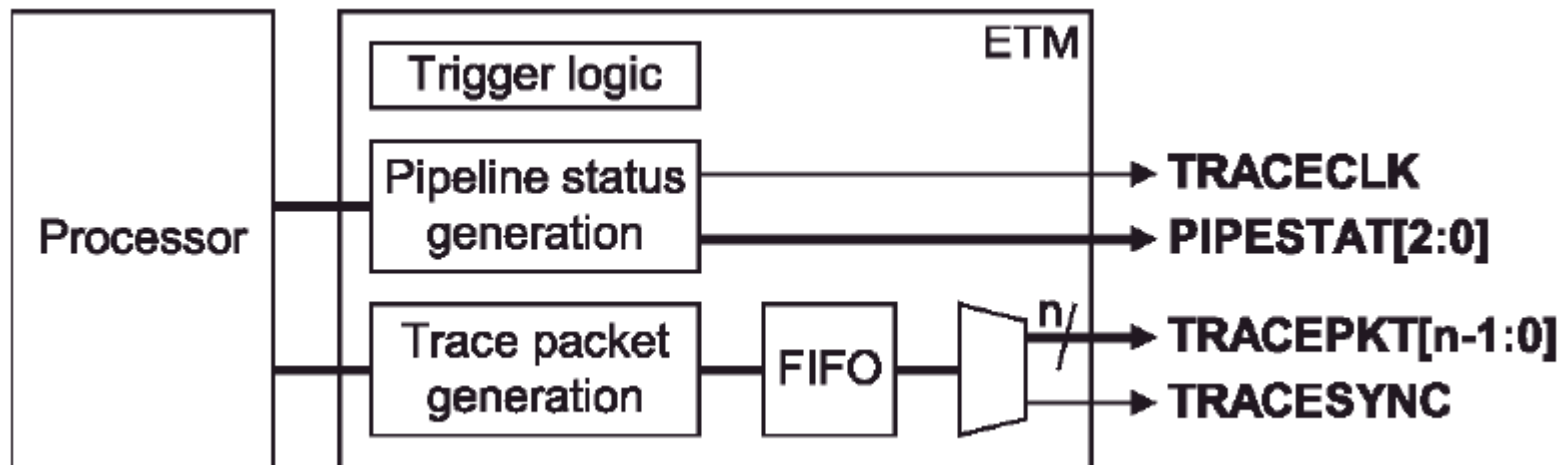
In-Circuit Debugger:

- Grundlæggende funktionalitet som Monitor
- Dedikeret debug interface (JTAG, OCDS, ONCE, BDM osv.)
- Samme funktionalitet som ICE med avanceret debug core
(ETM, Coresight, Nexus....)

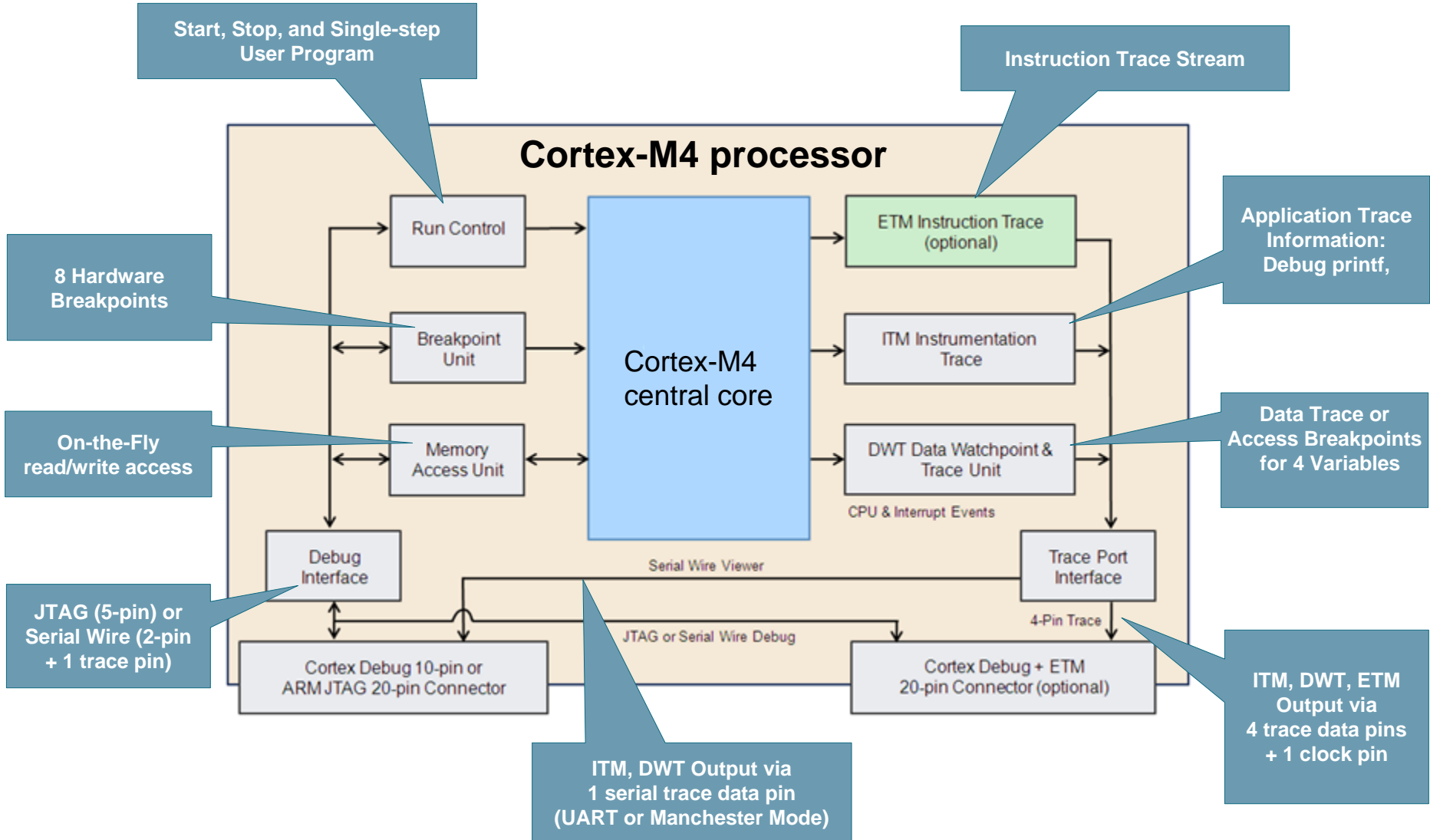
Debugging teknologier

In-circuit debugger (JTAG etc.)

ETMv1.x for ARM7/ARM9:



Debugging teknologier CoreSight™



Debugging teknologier

Statisk kodeanalyse

- En test validere om koden gør, hvad der forventes
- Statisk kodeanalyse sikrer, at koden ikke gør noget, der ikke forventes

Debugging teknologier

Statisk kodeanalyse

Critical

- Array out of bounds violations
- Buffer overflows
- Memory leaks
- NULL pointer dereference
- Freeing non-heap memory
- Freeing unallocated memory
- Un-validated inputs
- Un-initialized variables
- Freeing NULL pointer

Serious

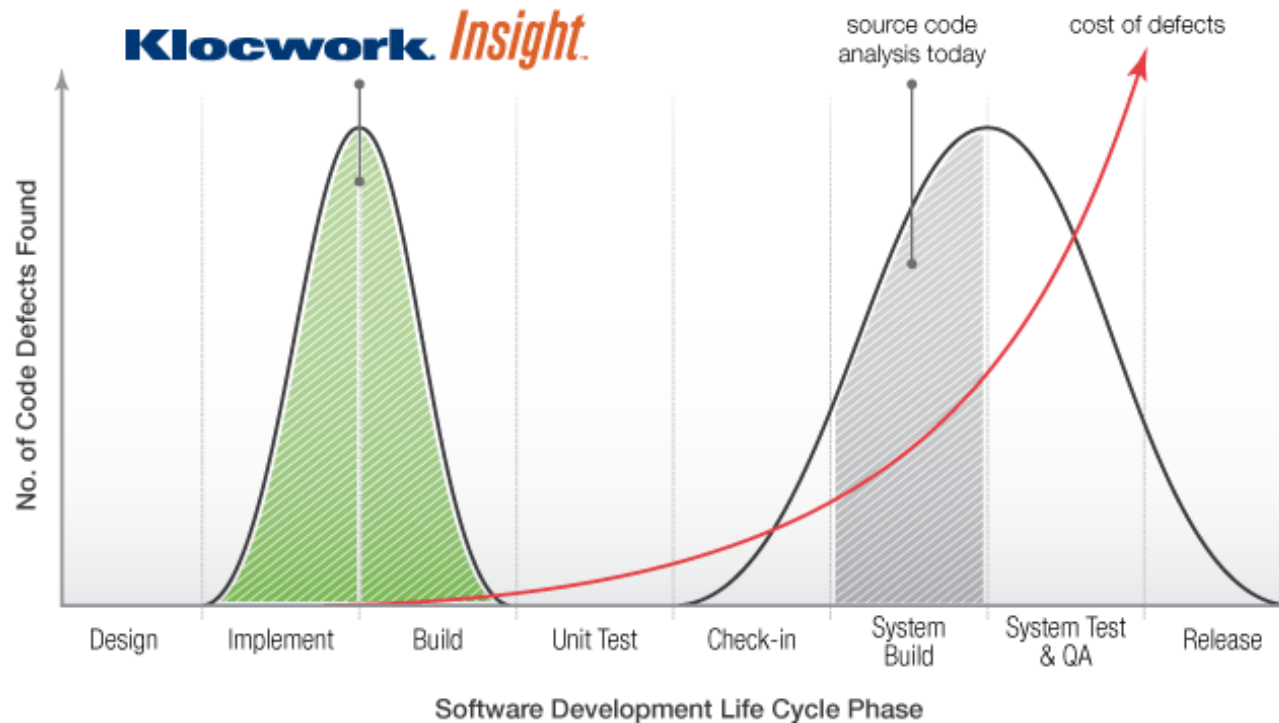
- Assignment in condition
- Suspiciously placed semi-colon
- Label unused
- Statement has no effect
- Unreachable code
- Ignored return values
- Invalid pointer arithmetic
- Non-void function returns void value
- Void function returns void value
- Race conditions
- Use of invalid iterator

Maintainability

- Multiple declarations
- Unused software entities
- Global object used locally
- Value is never used after assignment
- Use of unintended copy
- Only declaration found for object
- Use of dangerous and/or insecure functions
- Usage of object without declaration

Debugging teknologier

Statisk kodeanalyse



Debugging teknologier

Demonstration af avancerede eksempler:

- Overvågning af variabler i realtid
- Data-breakpoints
- Task specifikke breakpoints
- Performance analyse
- Realtids trace af programeksekvierung og data acces
- Backstepping og genskabelse af konteksten (Context tracking)
- Korrelation af programafvikling med effektforbrug og HW-logikanalyse
- Memory allokeringsanalyse
- Detektering af "ikke funktionelle fejl"
- Check af kodningregler
-