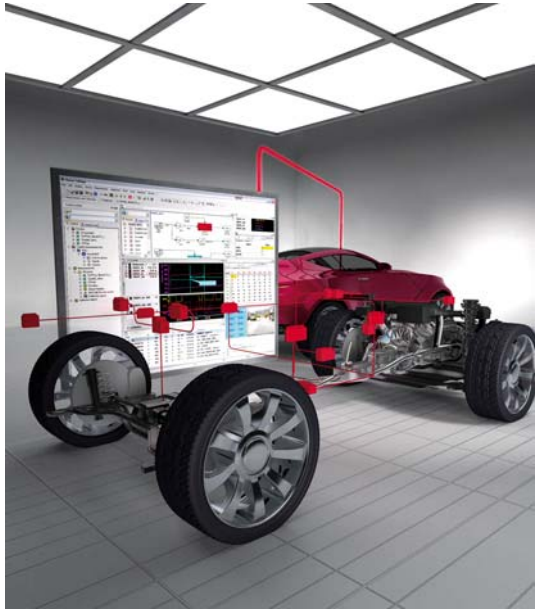


Current and Prospective High-speed Measurement Systems

Vector Congress 2010, Stuttgart

V0.01 2010-11-29

Agenda

> **Definition: Measurement and Calibration Hardware**

Customer Requirements

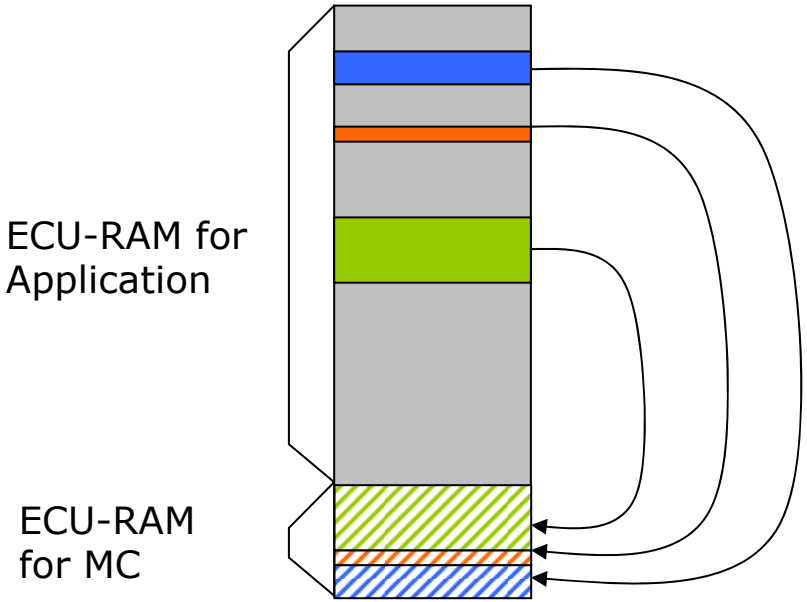
Vector Product Strategy

ECU Integration and Realization

Definition of Measurement & Calibration (MC)

- Calibration in terms of the MC use-cases means:
  Modification of ECU Flash parameters.
- Measurement in terms of the MC use-cases means:
  Measurement of signals located in the ECU-RAM or FLASH.
  Measurement is typically synchronized with ECU tasks.

- Standard use cases:
  XCPonCAN or XCPonFlexRay



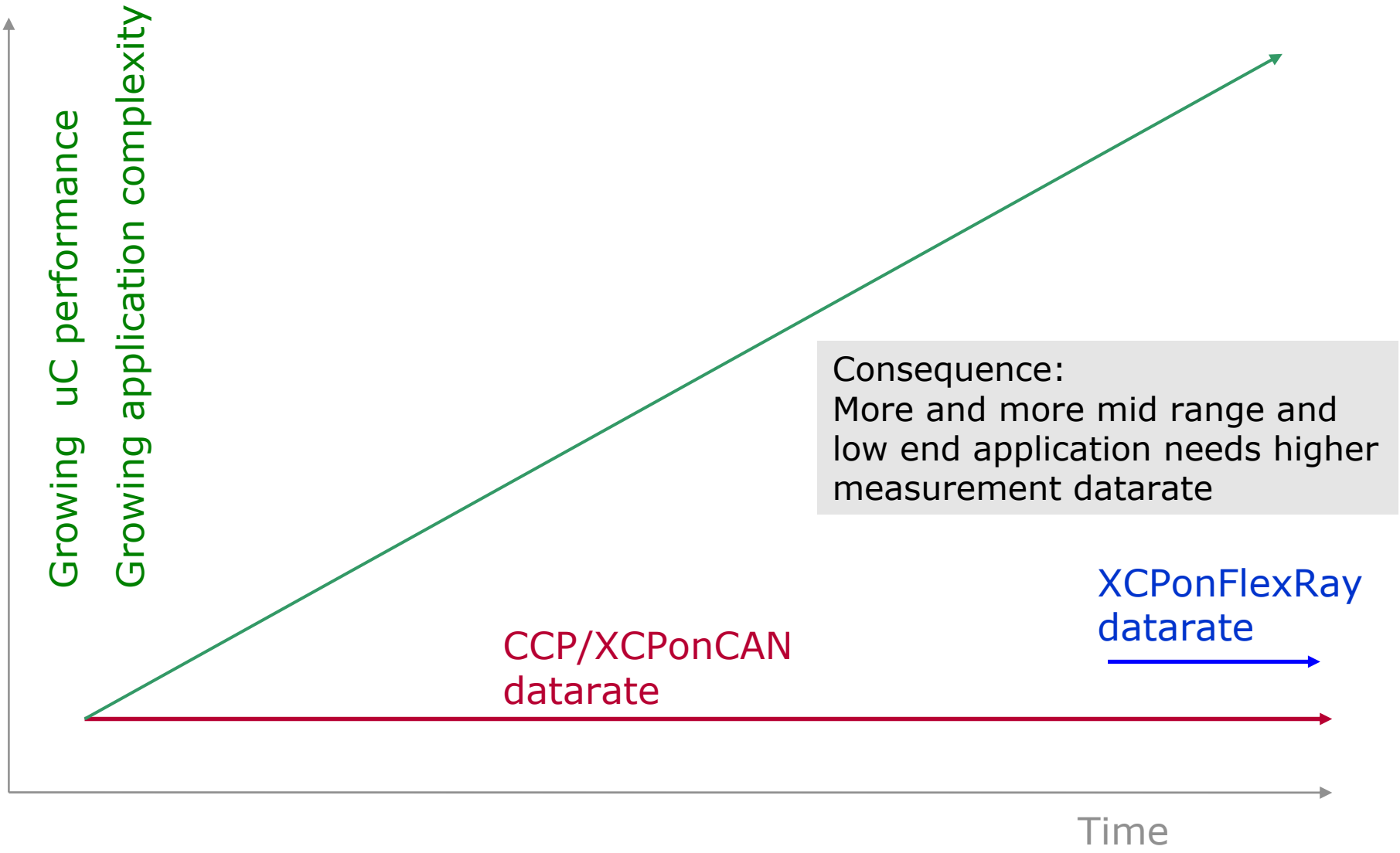
Standard concept needs ECU resources like:

- RAM (DAQ tables, data buffers)
- FLASH (CCP/XCP driver code)
- CPU time
- BUS bandwidth (CAN, FlexRay)

Send copied signal values via:

CAN	~50 kB/sec
Private CAN	~100 kB/sec
FlexRay	~100 kB/sec

µC Performance versus CAN/FR application datarate

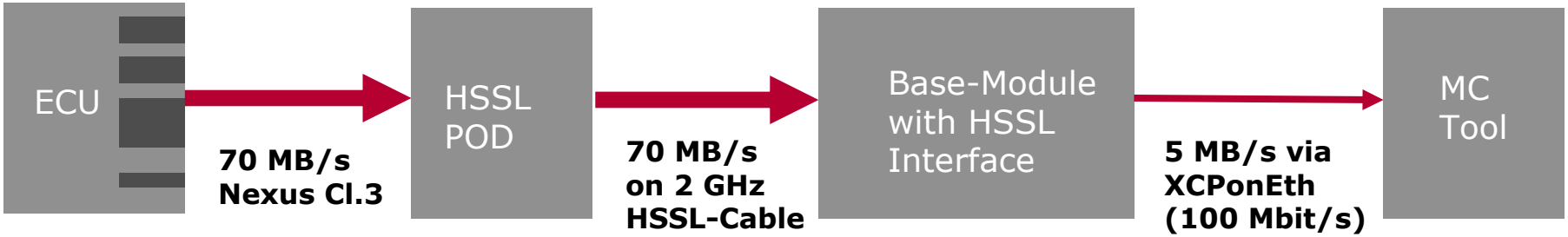


Customer requirements

Requirement	XCPonCAN	High Speed MC	Project Examples
High Data-rate	50 kByte/s	Up to 5000 kB/s New Requirement: 20MB/s	Chassis Driver Assistance
Short Event Cycles	min. 1 ms	15 µs	EV/HEV engine control 10-20 kHz Max. 80 MHz
Easy waterproof mounting	Normally no effort, because CAN is already available	Very easy with Serial POD, more complicated with HSSL POD	Engine compartment Powertrain/ Chassis
No CPU load for measurement	Not possible, but due to low data-rate not relevant	Possible, but µC dependant otherwise ~ 4 % CPU load per 1 MB/s	
Short Latency for Bypassing	Not possible	Depending on µC , short latency possible	
Cold-Start measurement	Possible with XCP. Not possible with CCP.	Possible	
Easy ECU-SW instrumentation	More complicated	Quite easy	
"Brain Dead" fast flash programming	Not possible, Debugger needed	Possible with high data-rate	
AUTOSAR Analyzing	Slow	Fast ECU access	

VX1000 Overview

Dataflow with HSSL POD



-4 x RAM Supervision windows

- Total: 512 kB

- 25 parallel ECU Pin's to connect

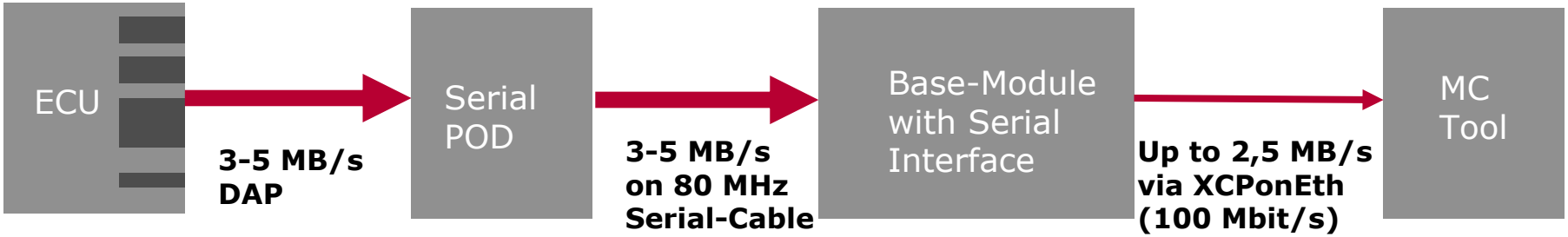
- Serial converter

Supported Microcontrollers:

- Freescale / STMicroelectronics
  MPC55xx/MPC56xx with NEXUS Class 3
- TI TMSx70 Family with RTP/DMM

VX1000 Overview

Dataflow with Serial POD via DAP Data-Trace



- 4 x RAM Supervision windows
- Total: 512 kB

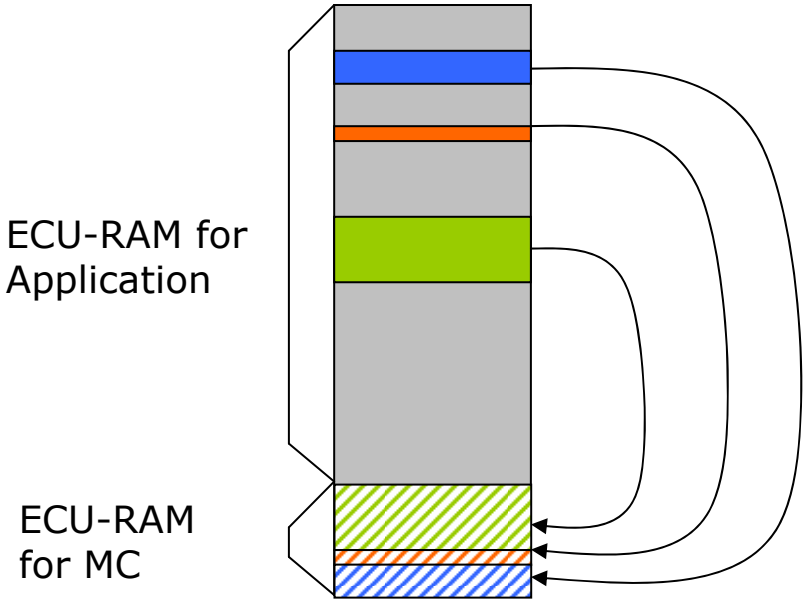
- 2 ECU Pin‘s (DAP) to connect

Supported Microcontrollers:

- Infineon TC17xx, TC13xx via DAP with Emulation Device (ED)

VX1000 Overview

Dataflow via JTAG or DAP „Copy“



Serial POD

Send copied signal values via:

DAP	0,6 -1	MByte/sec
JTAG	0,6 -0,8	MByte/sec

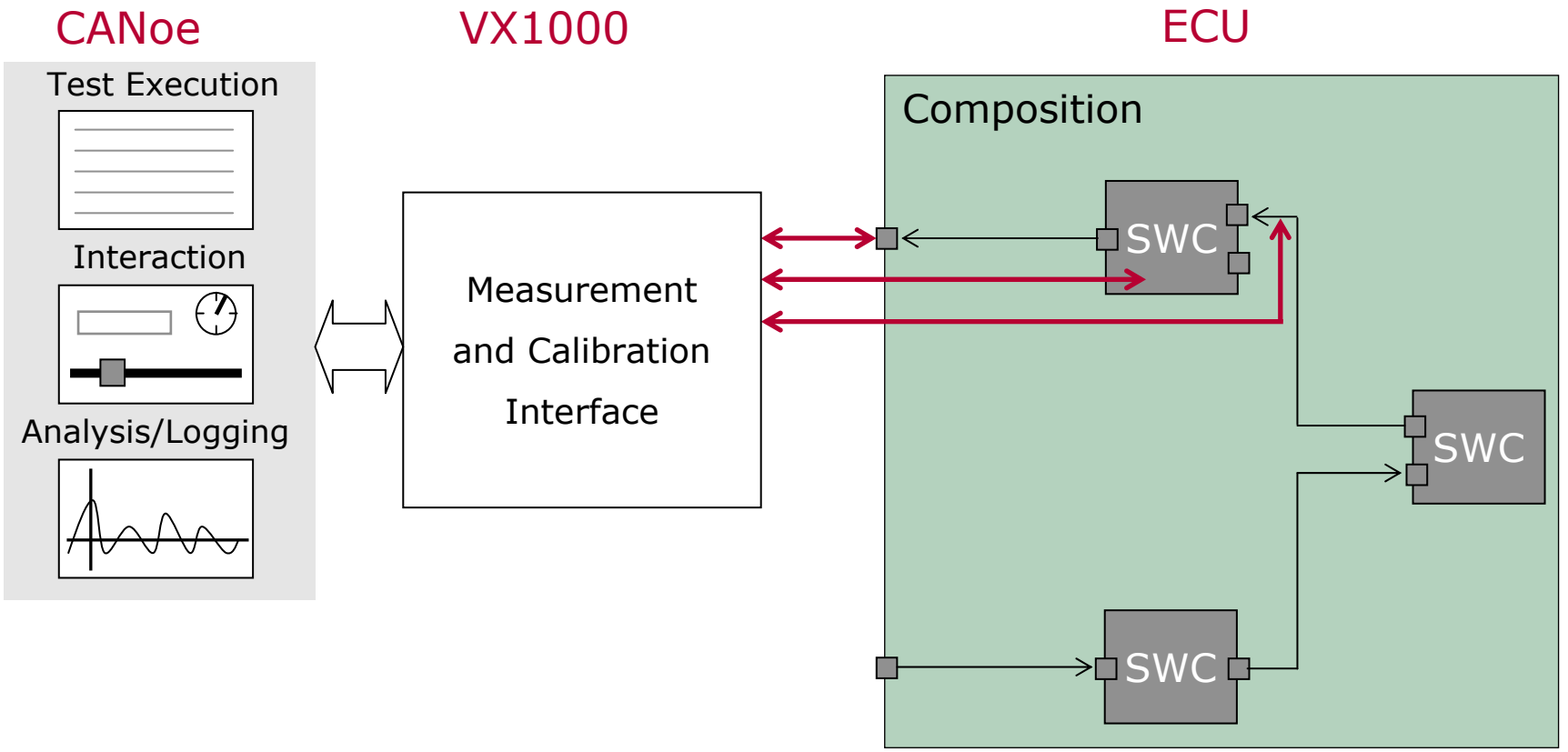
Supported Microcontrollers:

- Infineon TC17xx, TC13xx with DAP (PD or ED)
- Infineon XC2000 with DAP
- FreeScale / STMicroelectronics MPC55xx/MPC56xx with NEXUS Class 2+
- NEC V850 with NEXUS Class 2+

Autosar Testing with VX1000 & CANoe

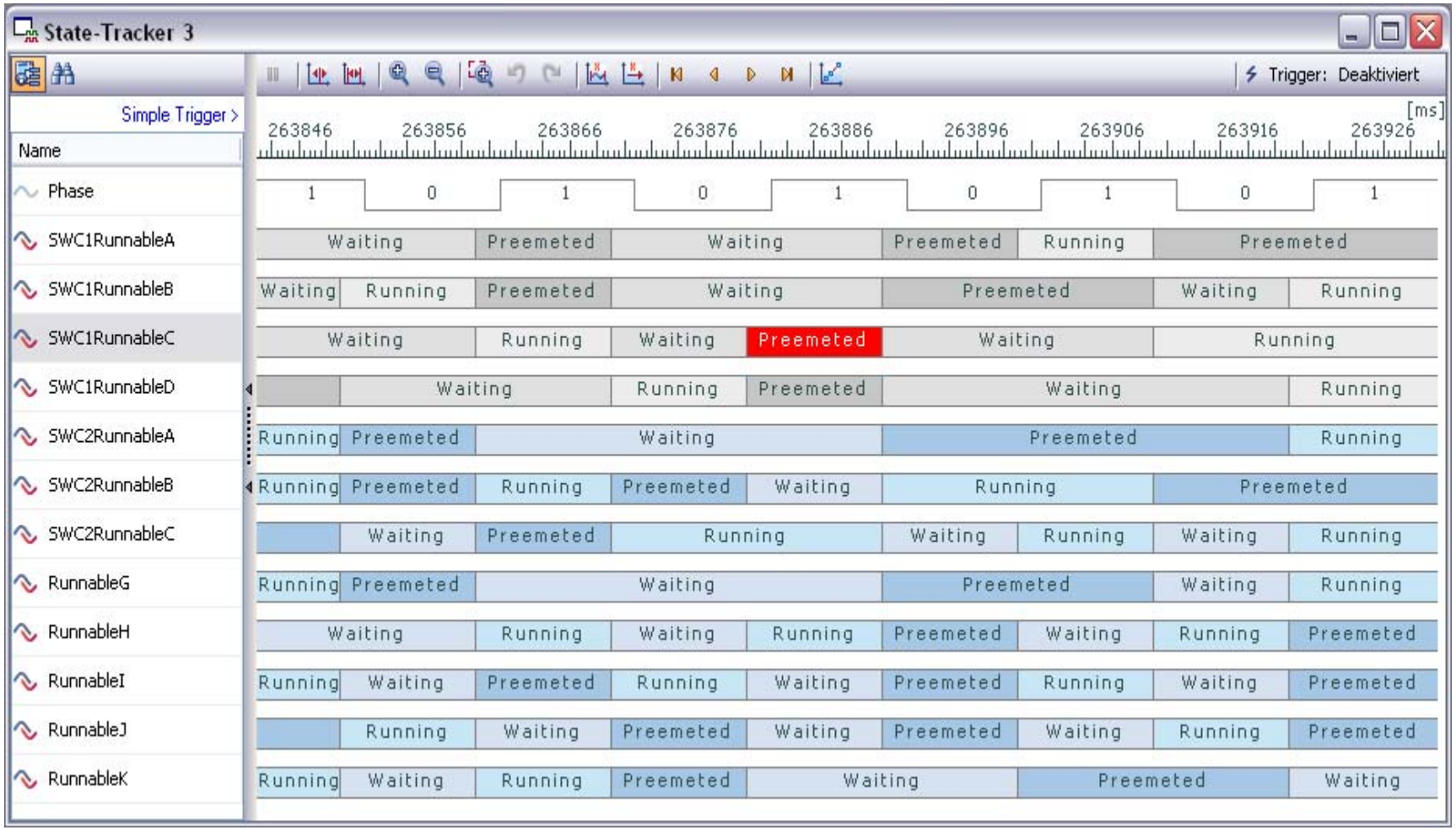
Access to SWC and BSW on real ECU

- Access to **S**oft**W**are**C**omponent internal data and trigger
- Symbolic description of internal symbols
- Used for stimulation, analysis and testing



Autosar Testing with VX1000 & CANoe

Example: CANoe State Tracker Window (planned)

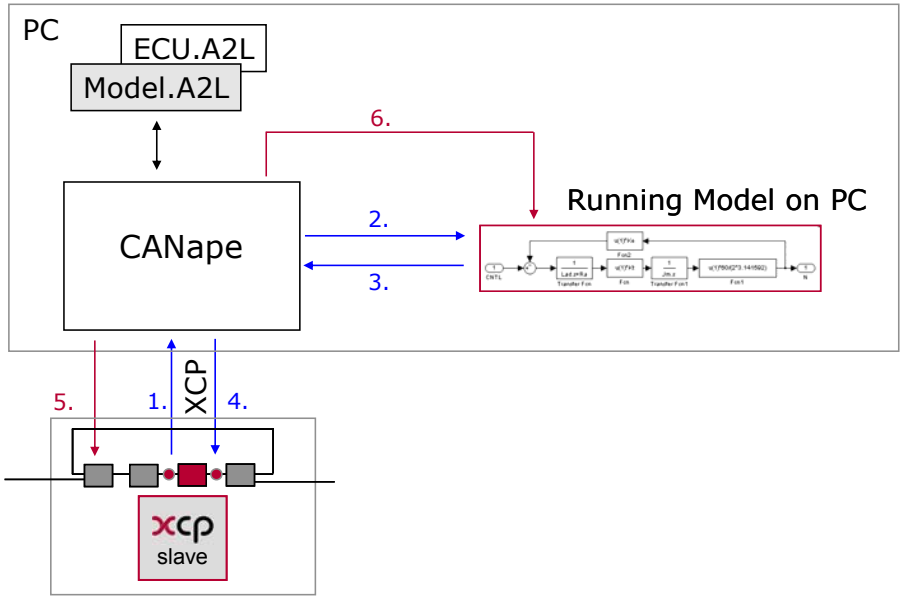


Bypassing with VN8900 & VX1000

VN8900 for Realtime Bypass-Execution

Bypassing in terms of the MC use-cases means:

- Read out ECU parameter
- Calculate new value outside ECU
- Write back new values according ECU task



Performance Overview

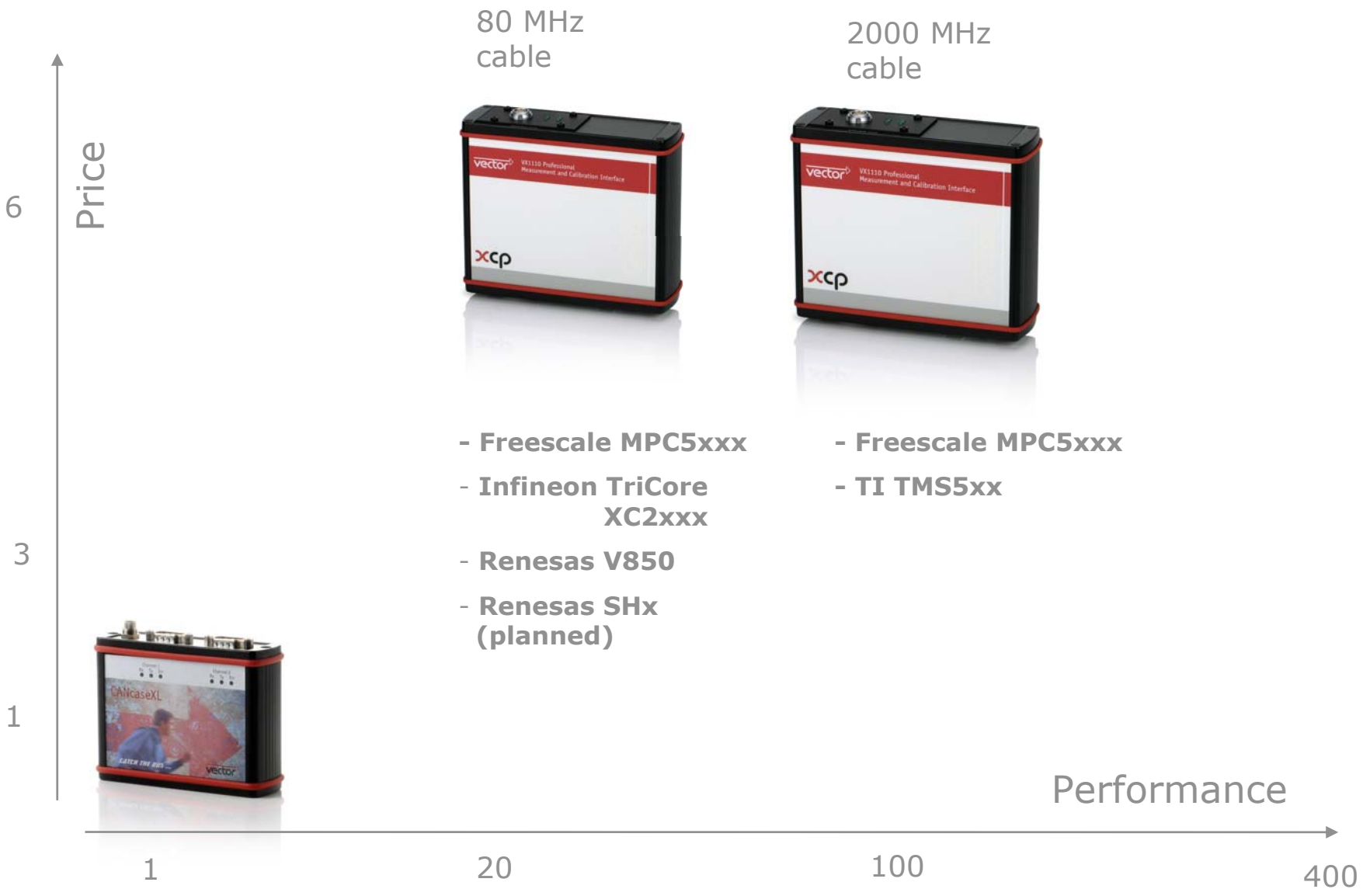
ECU-Interface	ECU SW needed	ECU RAM	DAQ data rate	Min. Event Rate in µs	Latency e.g. for Bypass	DAQ ECU CPU load	ECU Pins	Max. Event	Max. Signal
JTAG/DAP Infineon via DAP TC1xxxPD/ED XC2000 Freescale MPC 5xxx Nexus Cl.2+ Renesas/NEC #: V850 Nexus Cl. 2+	Table driven DAQ Transfer Routines	~ 4 Byte per signal	200-800 kB/s	<100	Good	~ 4% per MB/s	DAP: 2 + GND/ Reset JTAG: 4 + 2 Rst + GND	31	100.000
Infineon DAP Data Trace Infineon via DAP TC1xxxED	Very little	256 kB from ED	1500-2500 kB/s	<15	No possible	None	DAP: 2 + GND/ Reset	256	100.000
Data Trace Freescale MPC *5xxx Nexus Cl.3* *Texas Instruments* *TMS570 RTP/DMM*	Very little	None	5000 kB/s	<15	Very good	<1%	JTAG+ 15 AUX-Ports Min 24	256	100.000

POD selection for Freescale MPC55xx and MPC56xx

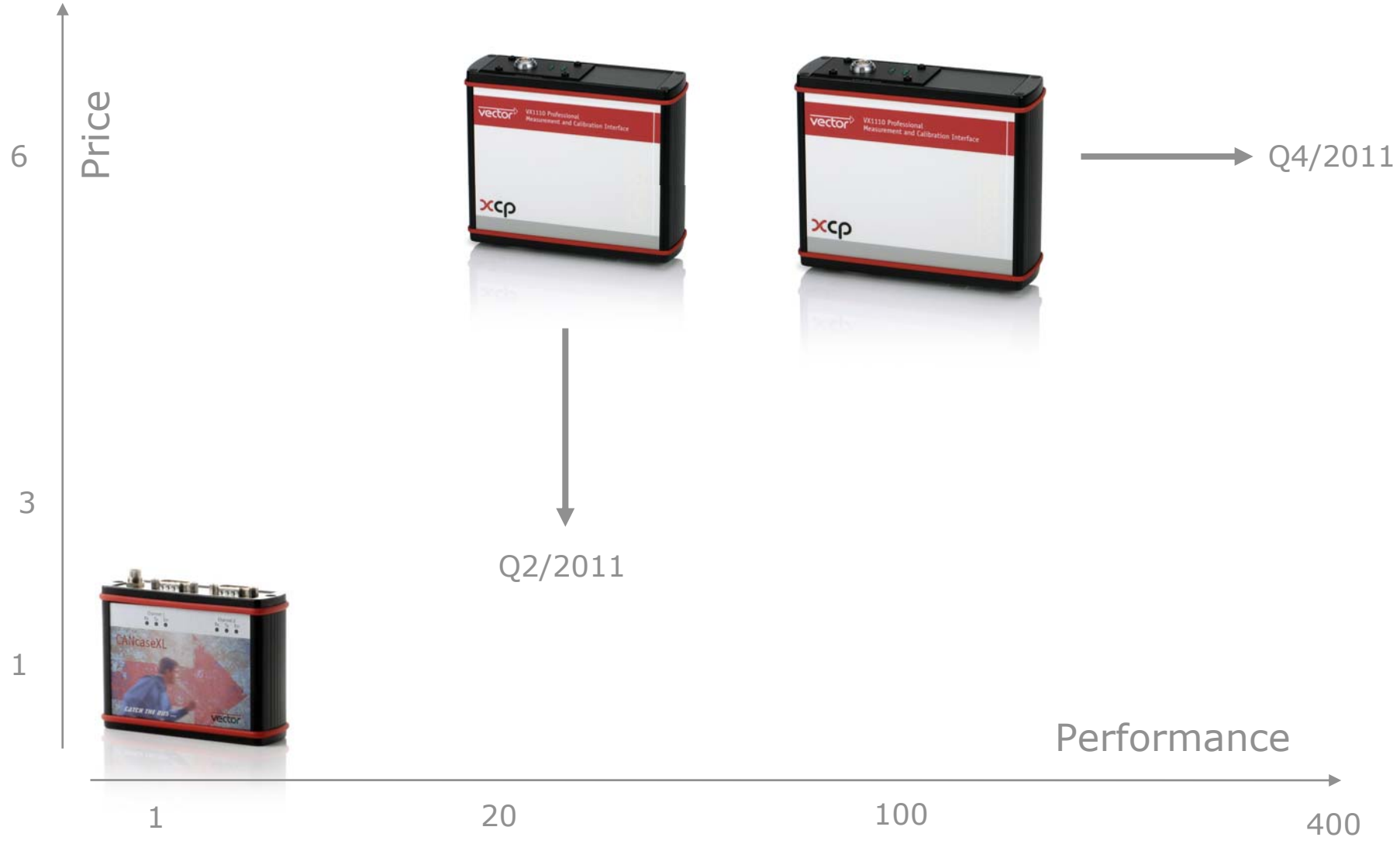
MPC55xx	Nexus Class	Serial POD	HSSL POD
MPC5510	2+	√	
MPC5534	3+	√	√
MPC5553	3+	√	√
MPC5554	3+	√	√
MPC5561	3+	√	√
MPC5565	3+	√	√
MPC5566	3+	√	√
MPC5567	3+	√	√
MPC5534	3+	√	√

MPC56xx	Nexus Class	Serial POD	HSSL POD
MPC560xB/C -208 BGA	2+	√	
MPC560xB/C -all other	1+		
MPC560xP	2+	√	
MPC560xS	2+	√	
MPC563xM	3+	√	√
MPC564xA	3+	√	√
MPC564xL	3+	√	√
MPC5668	3+	√	√
MPC56xF	3+	√	√

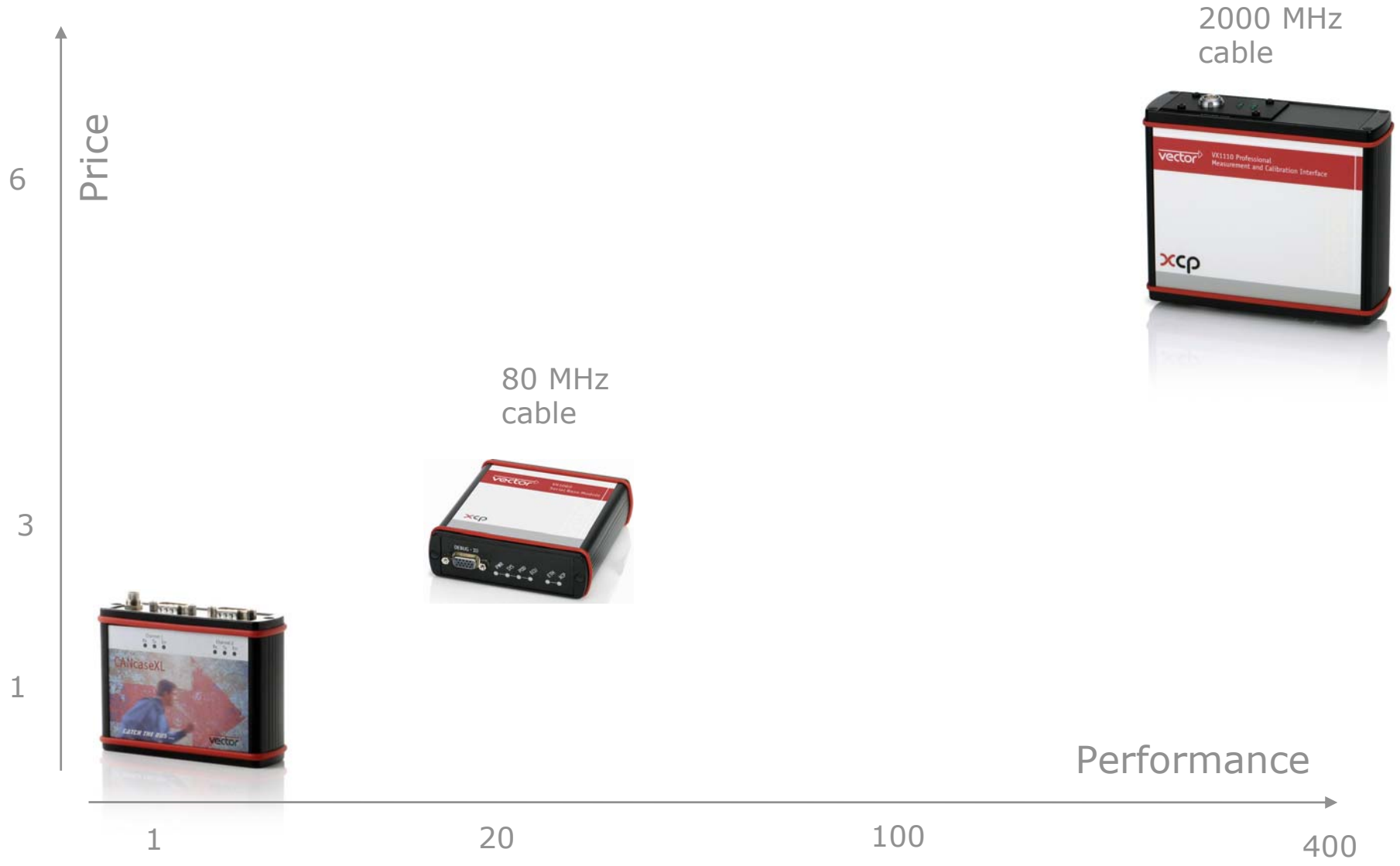
VX1000 price / performance ratio: 2010



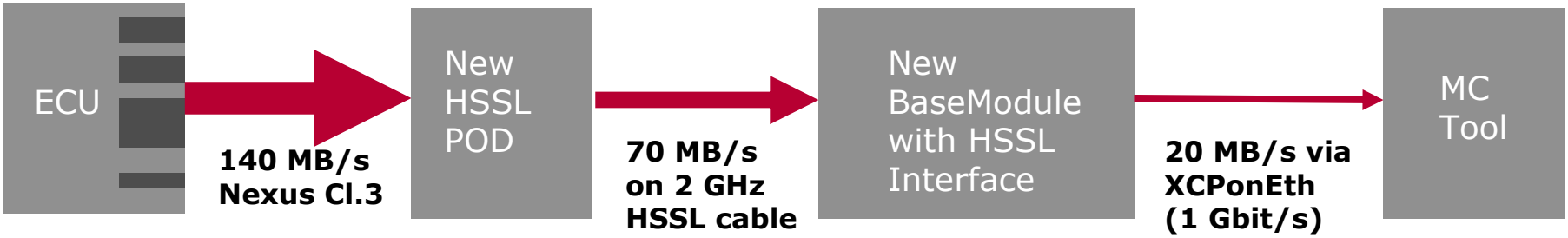
VX1000 price / performance ratio: 2011



VX1000 price / performance ratio: 2011



Outlook: VX1000 Dataflow with Next Generation HSSL POD



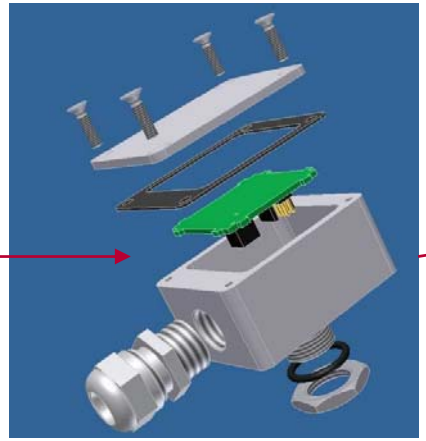
- 4 x RAM Supervision windows
- Total: 512 kB

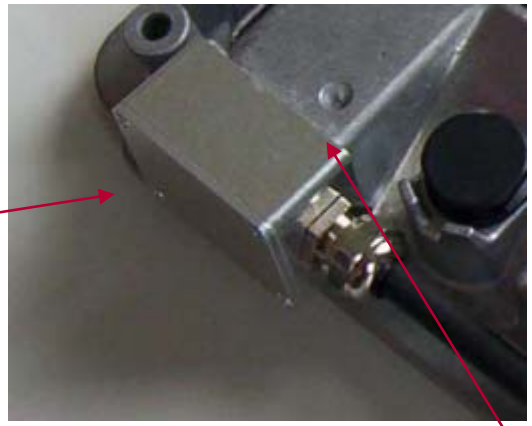
-25 parallel ECU Pin‘s to connect
- Serial converter

-Prefilering

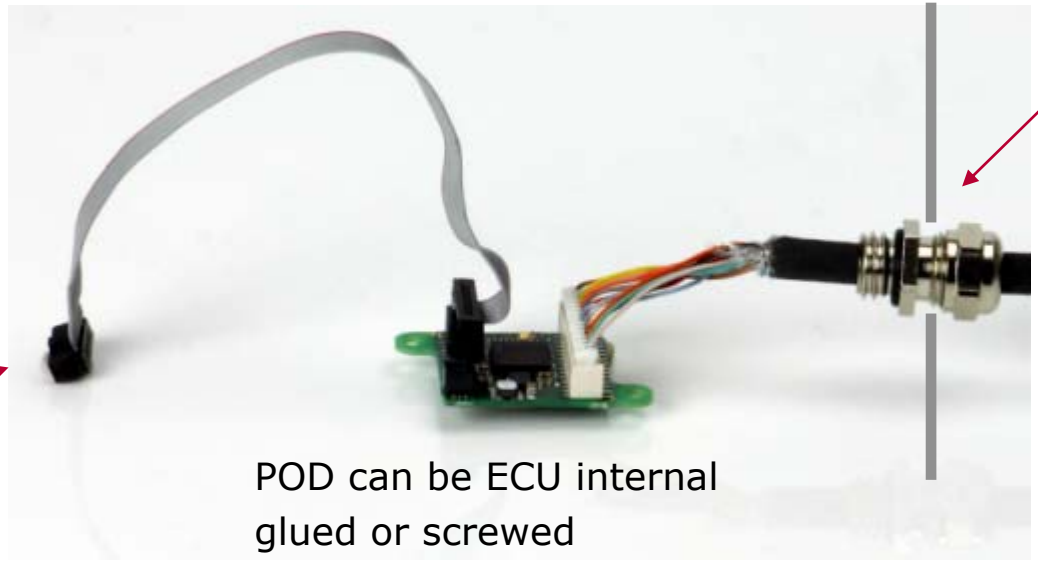
Mounting Examples of Serial POD

POD with waterproof Housing





POD ECU internal mounted



ECU housing break-hrough (10 mm)

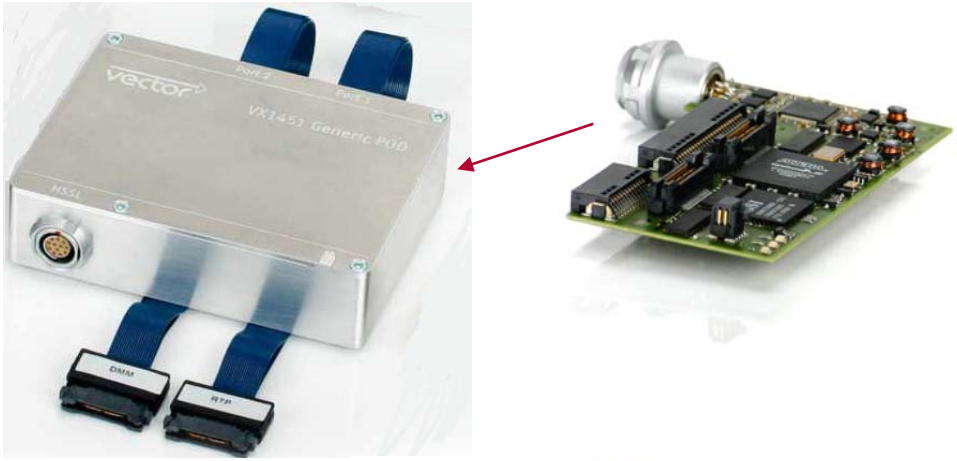
ECU JTAG/DAP connector

POD can be ECU internal glued or screwed

Mounting Examples of HSSL POD

Standard POD housing

(not waterproof)

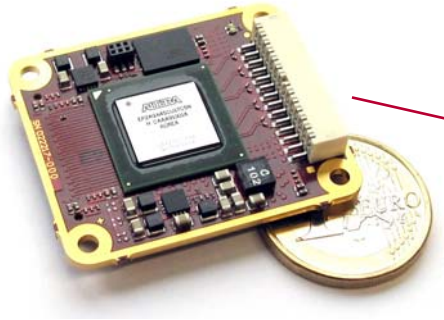


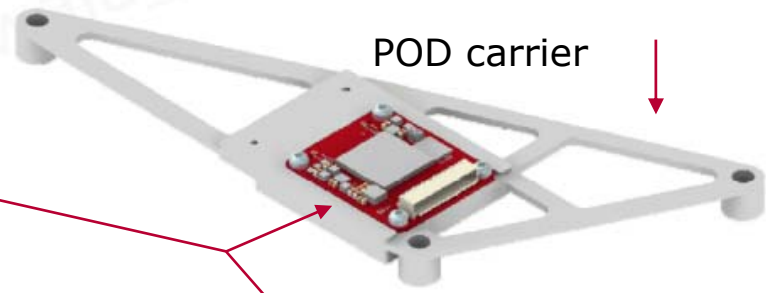
Customized POD in ECU

cover-plate (waterproof)

Customized aluminum

POD carrier





Flex-PCB

Nexus/RTP ECU connector

VX1000 ECU SW Instrumentation

```
//-------------------------------------------------------------------------
#include "VX1240.h"
//-------------------------------------------------------------------------
// Global data of the VX1000 system
VX1000_DATA;
//-------------------------------------------------------------------------
void main( void ) {
  // Initialize the VX1000 system
  // Must be called before any other VX1000_xxx function is called
  VX1000_INIT();

    // Measurement
    VX1000_EVENT(0); // Trigger XCP DAQ event number 0


void task1 {
  ...
  // Measurement
  VX1000_EVENT(7); // Trigger XCP DAQ event number 7
```

e.g:
100 µs
Task

e.g:
10 ms
Task

Conclusion

- VX1000 system is a cost efficient and future proof solution for Measurement, Calibration, Bypassing and Flash-programming usecases.

- Project examples with successful VX1000 usage:
  - Chassis (ABS/ESP, Suspension Control)
  - Steering (EPS)
  - Driver Assistance
  - Powertrain (EV/HEV)
  - Body (Safety/Airbag)

Thank you for your attention.

For detailed information about Vector
and our products please have a look at:
www.vector-informatik.com

Author: Alfred.Kless@vector-informatik.de
Vector Informatik GmbH
Ingersheimer Str. 24
70499 Stuttgart