Speed Your Time to Market by Leveraging Freescale's Software, Tools and Professional Services for QorIQ Development

EUF-SNT-T1467

Gino Gatto | Senior Field Application Engineer

M A Y . 2 0 1 5

freescale™

External Use

Agenda

Overview

- Complexity Increasing
- Freescale Technology and Ecosystem

Freescale Commercial Capabilities

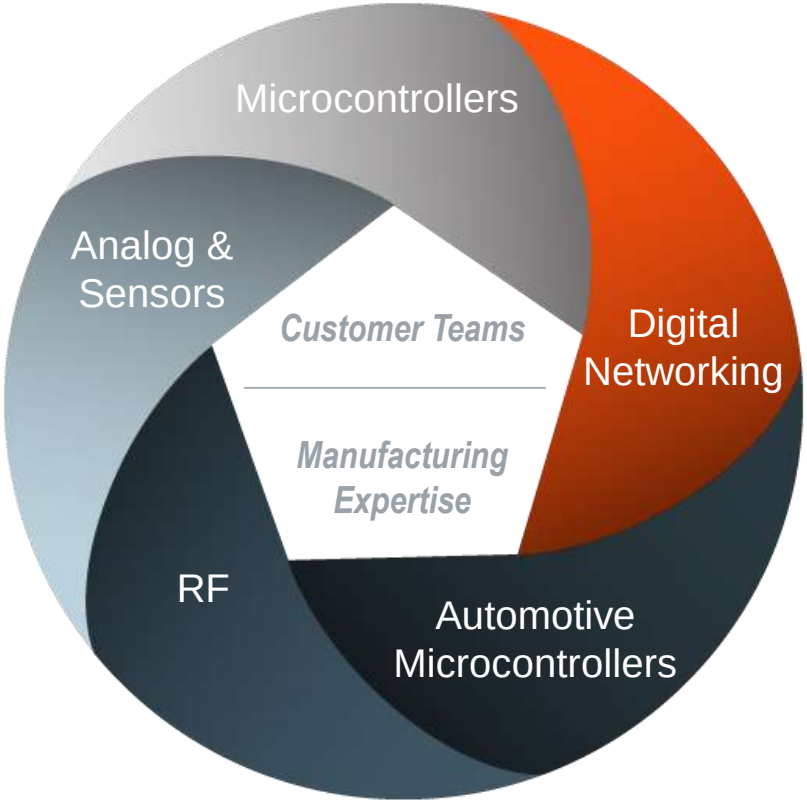
- Software
- Hardware
- Systems

Questions and Answers

We Are a Global Leader in
Embedded Processing Solutions

Automotive

Networking



Industrial

Consumer

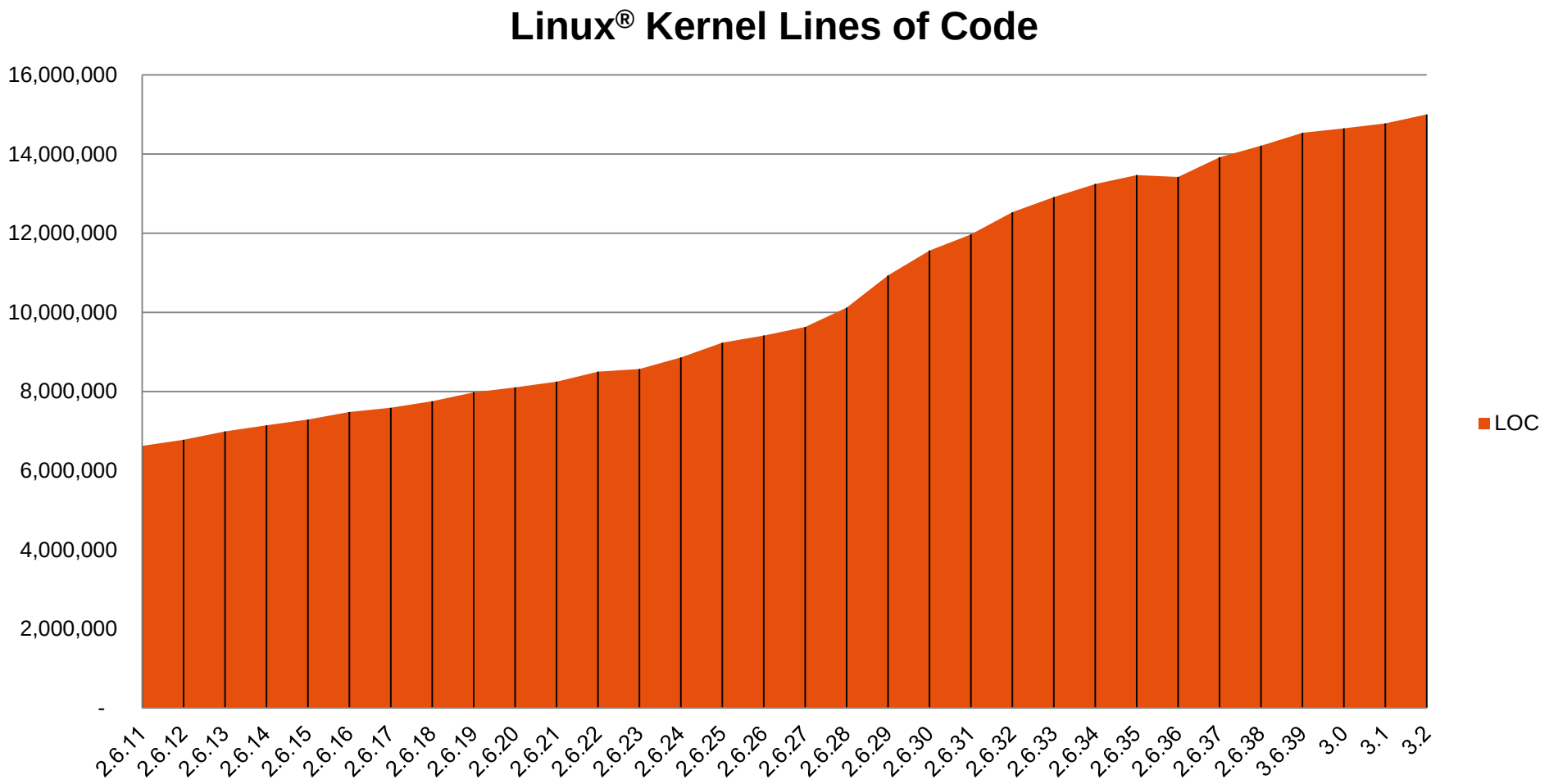
Five Core Product Groups

>50 Year Legacy
>6,400 Patent Families*

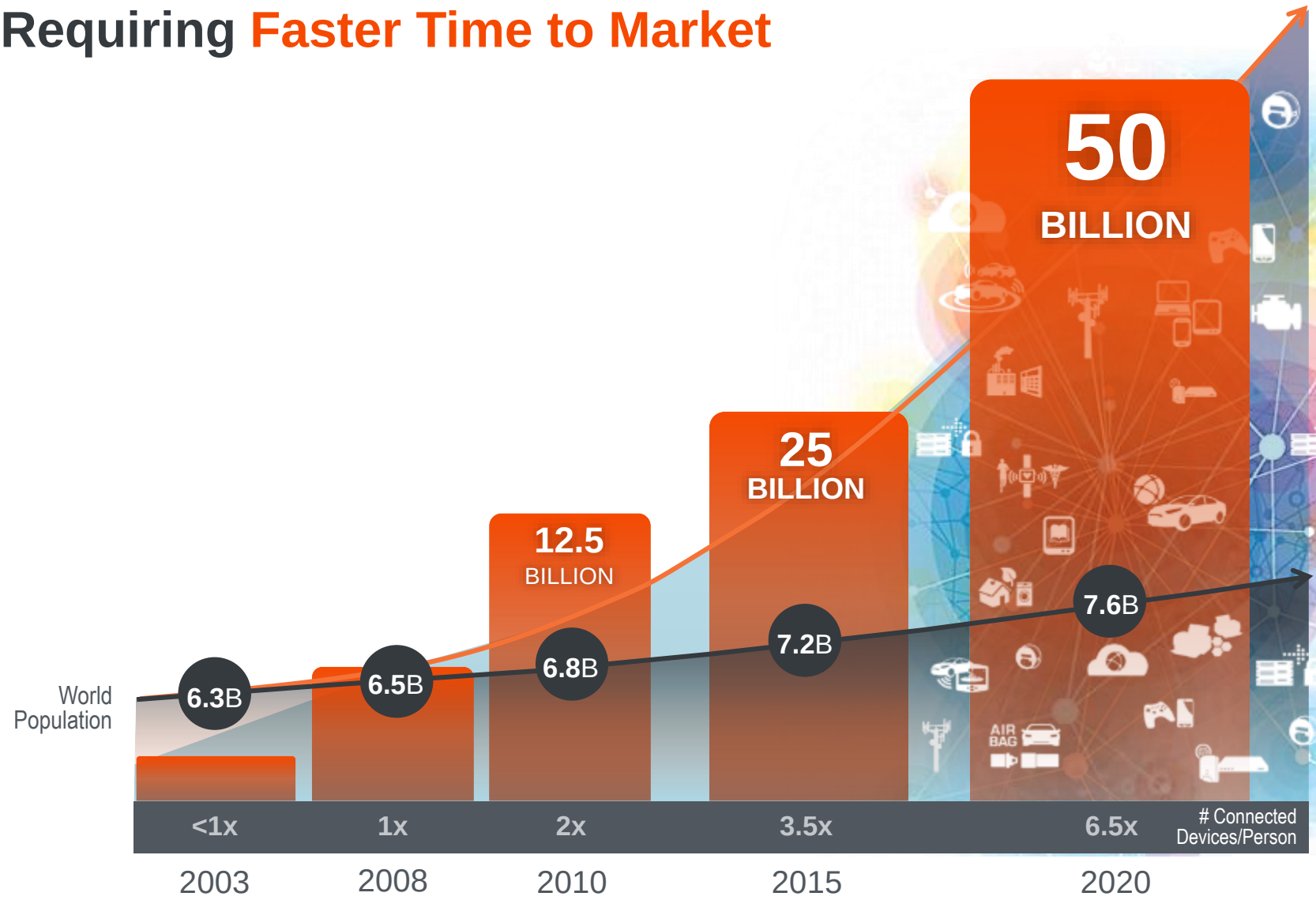
Four Primary Markets

* a set of patents taken in various countries to protect a single invention

Software Complexity Increasing



The Explosive Growth In Connected Devices – Requiring Faster Time to Market



* Sources: Ericsson, February 2011; Cisco Internet Business Solutions Group (IBSG), April 2011

Market Trends for Software in Embedded Devices

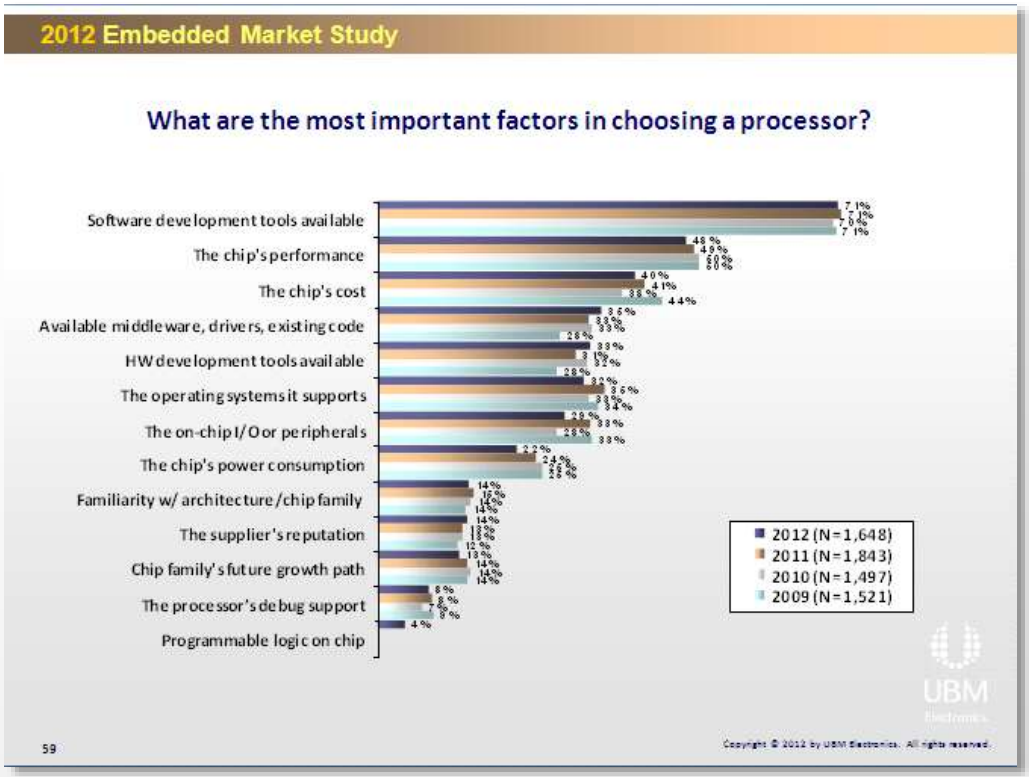
Processor complexity increasing 24+ cores, high speed I/O, more integration

Software costs account for >50% of development costs

Rise in customers moving to open source initiatives

Decrease in HW oriented engineers at customer-trend buy vs. make

Customers are focused on adding value through software!



Freescale Software and Systems Organization

HEADQUARTERS: **Oak Hill, Austin, Texas**

San Jose, **California**

Ottawa, **Ontario**

East Kilbride, Glasgow, **Scotland, UK**

Bucharest, **Romania**

Nice**, France**

Beijing, **China**
Shanghai, China
Chengdu, **China**
Shenzhen, **China**

Hyderabad, **India**
Noida, **India**
Bangalore, **India**

Herzelia, Israel

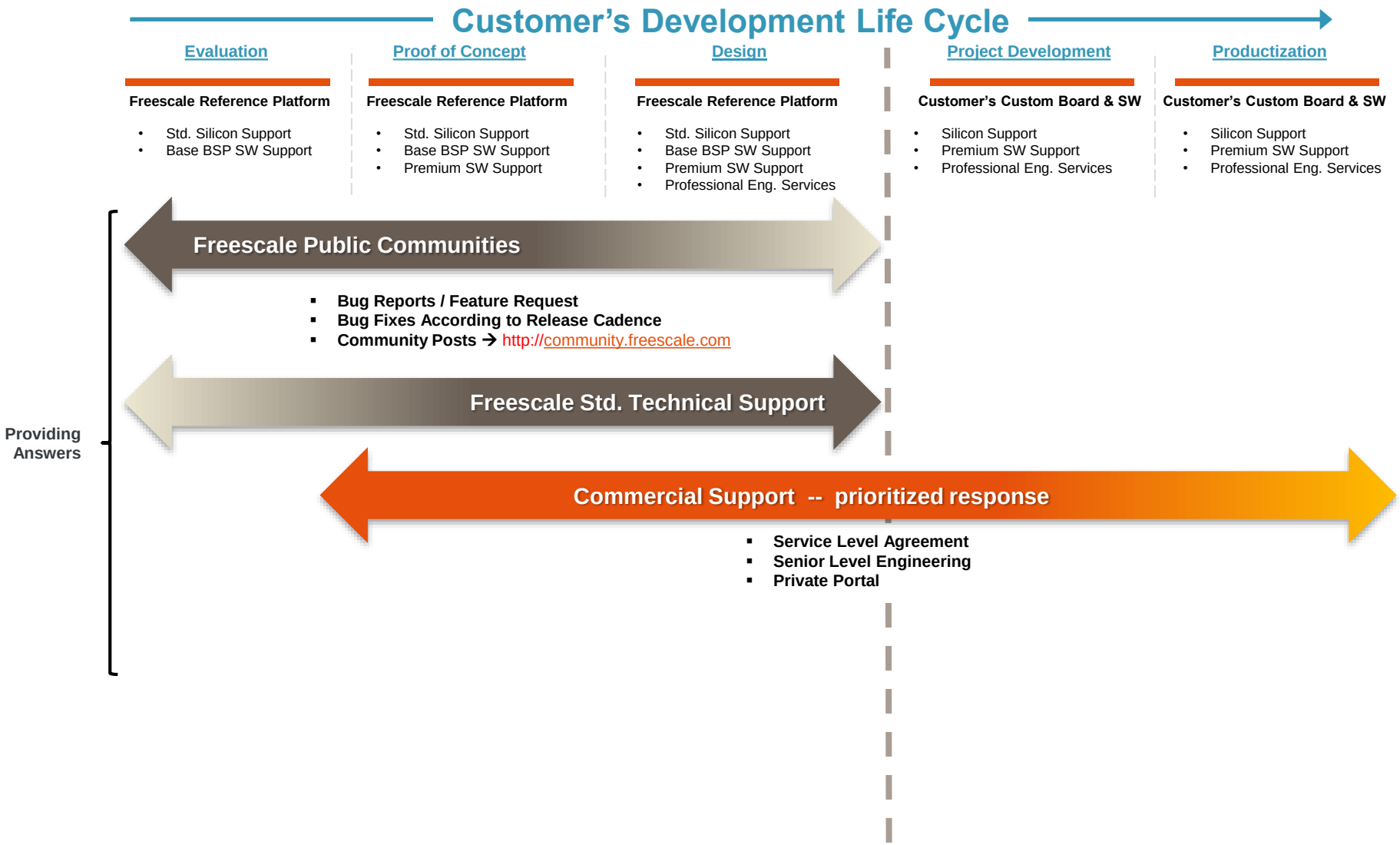
- **Freescale has 1,000+ software engineers, 700+ focused on Digital Networking**
- **Increasing investment on software through hiring and acquisition**
- **Run-Time Technologies, Multi-core, Tools, Key Applications**

Software Support / Professional Services Coverage

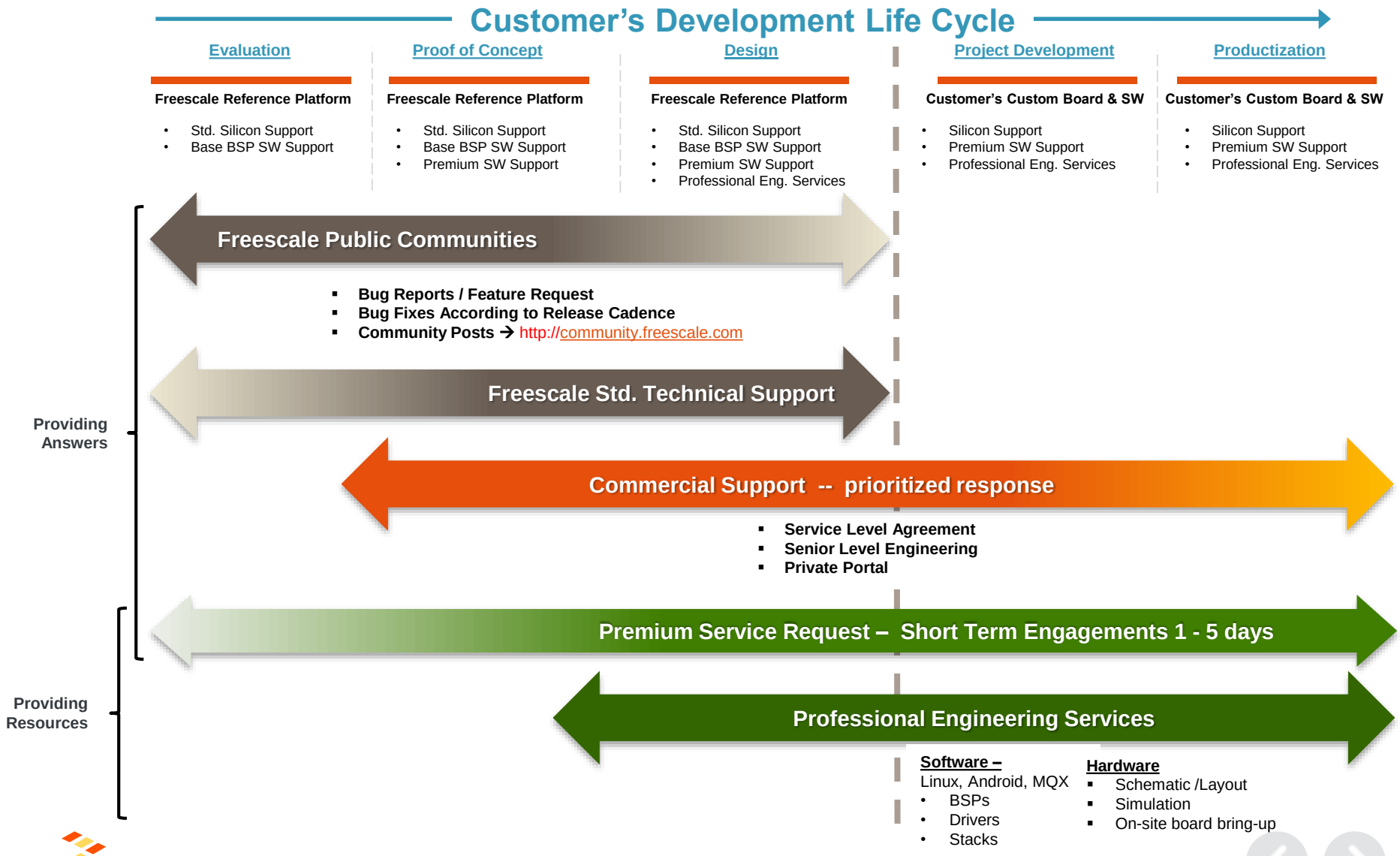
Customer's Development Life Cycle

Evaluation	Proof of Concept	Design	Project Development	Productization
Freescale Reference Platform	**Freescale Reference Platform**	**Freescale Reference Platform**	**Customer's Custom Board & SW**	**Customer's Custom Board & SW**
• Std. Silicon Support • Base BSP SW Support	• Std. Silicon Support • Base BSP SW Support • Premium SW Support	• Std. Silicon Support • Base BSP SW Support • Premium SW Support • Professional Eng. Services	• Silicon Support • Premium SW Support • Professional Eng. Services	• Silicon Support • Premium SW Support • Professional Eng. Services

Software Support / Professional Services Coverage



Software Support / Professional Services Coverage



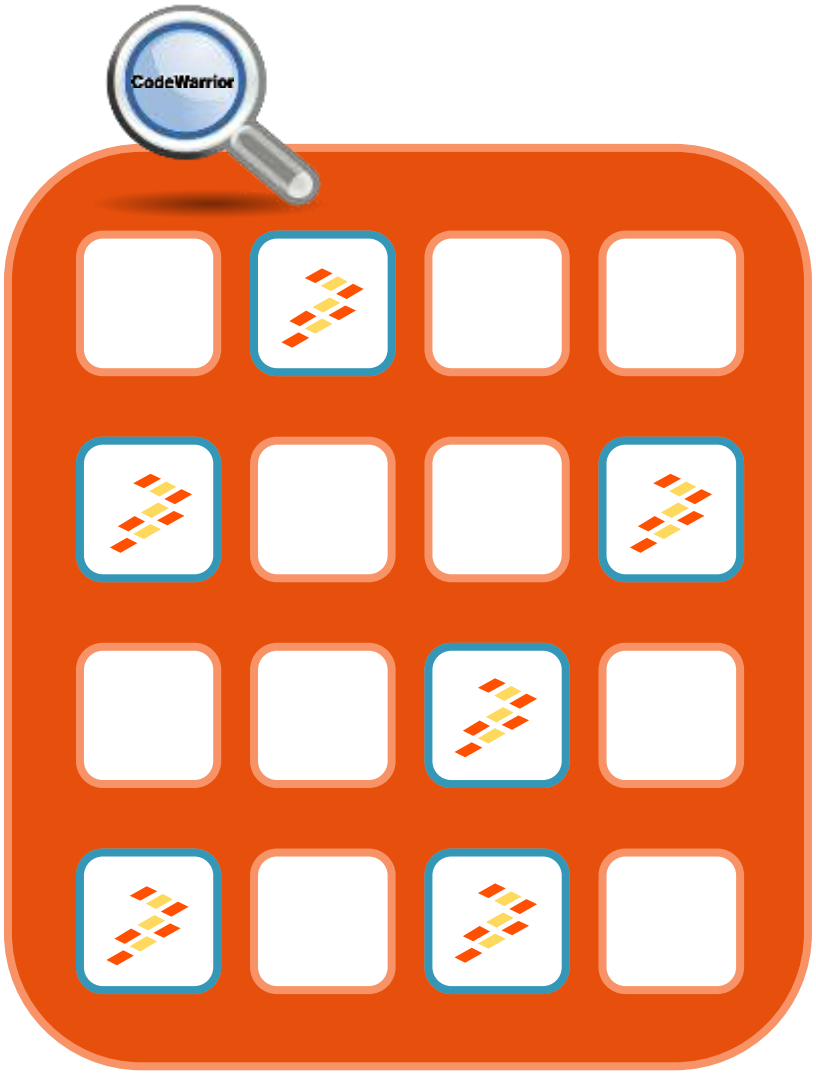
Networking Software and Services Group

Software Products and Custom Services

Development Tools	Runtime Products	Solutions Reference	Linux® Services	Integration Services
• CodeWarrior – IDE – Debug – Compiler – Trace • QorIQ Optimization Suite – Scenarios Tools – DDrV	• VortiQa Software Products – Application Identification Software (AIS) – Open Networking Switching Framework – Mobile Transport	• Storage Controller • SDN Switch • Wireless LAN • Data Concentrator • Smart Converged Gateway • Digital Signage	• Commercial Support • Frozen Branch • Application Specific Hardening • Feature Acceleration	• Systems Consulting • Design Services • Porting • Migration
CodeWarrior QorIQ	VortiQa			

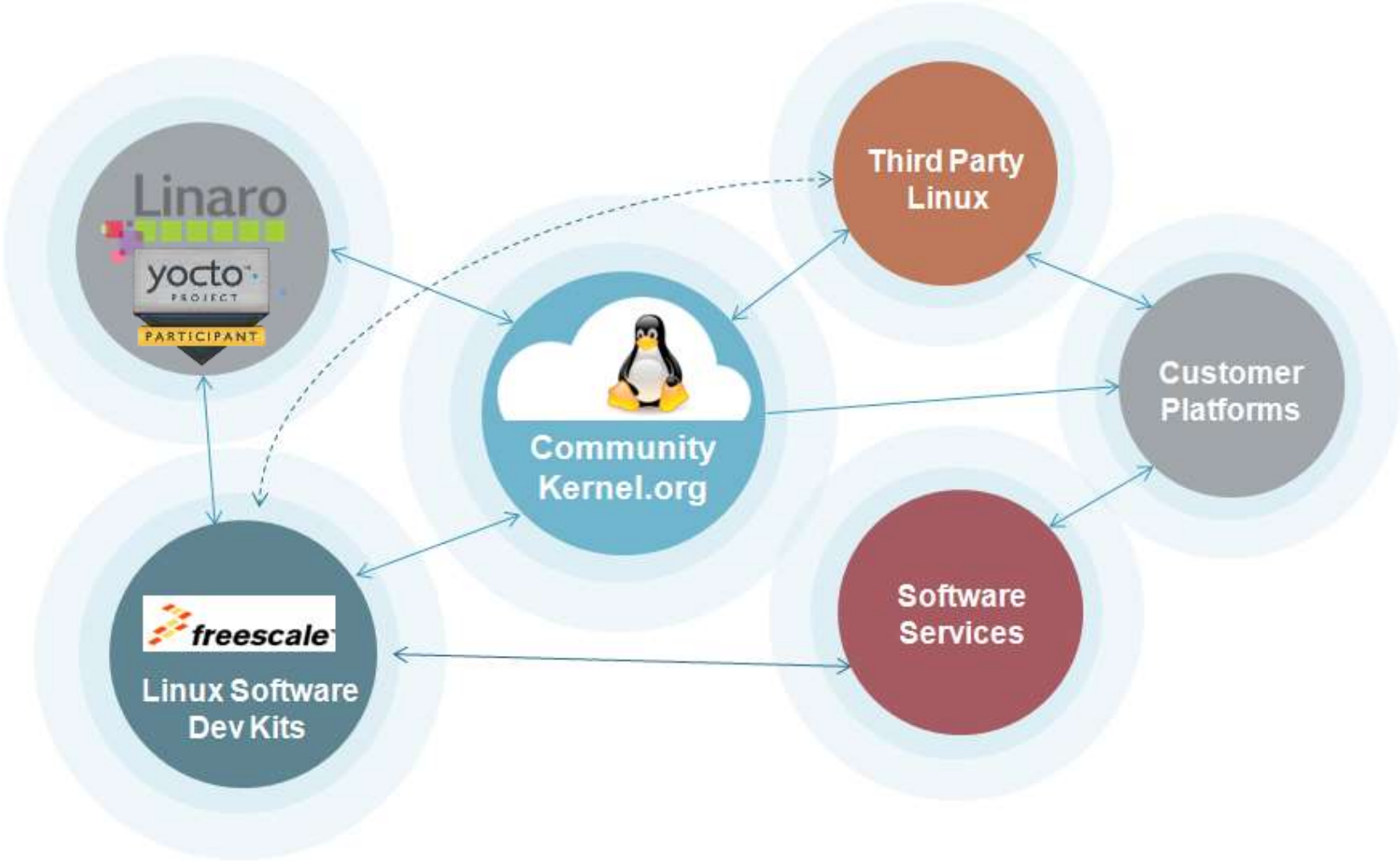
CodeWarrior for Networking

- Systems are complex - CodeWarrior makes debugging easy
  - Multicore or multiple cores
  - Heterogeneous or homogenous
  - Linux or Multi-OS Systems
  - ARM®, Power Architecture® and StarCore
- Use one CodeWarrior software



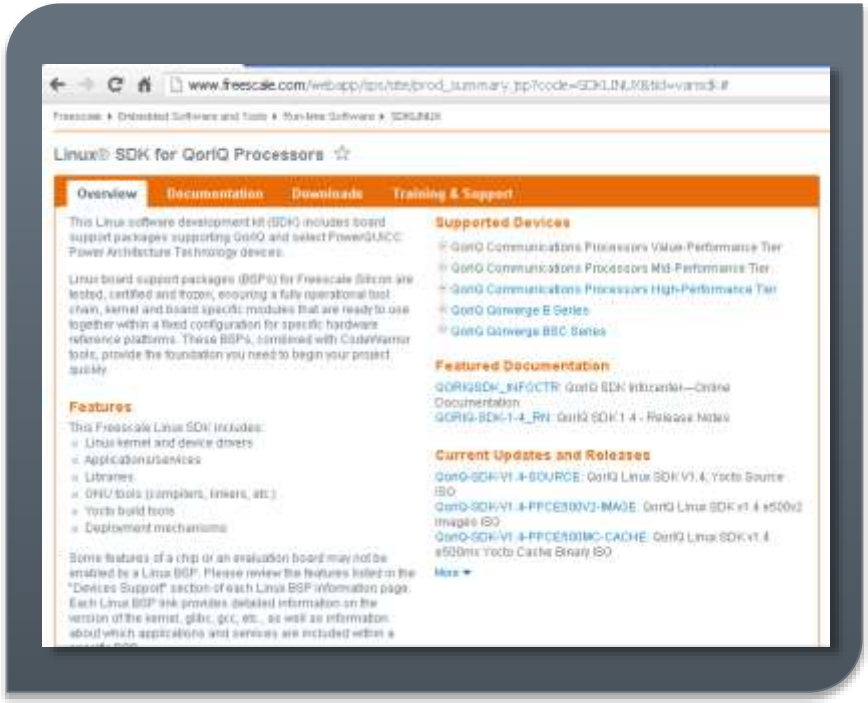
freescale.com/CodeWarrior

Freescale Linux Overview



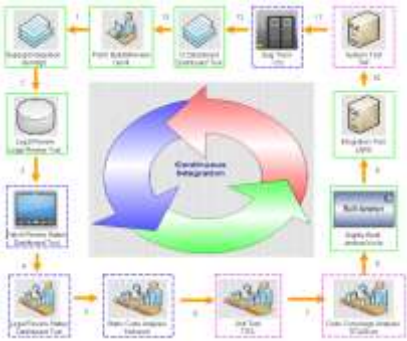
Freescale Networking Linux SDK

- Linux Software Development Kit (SDK) for Power Architecture
  - Optimized Linux software
    - Complete range of QorIQ and PowerQUICC platforms
    - Hardware accelerated
  - Rigorous testing
    - Multiple configurations, Host OSes
    - Performance tuned
  - Flexible AMP/SMP support
  - Yocto-based
  - Bi-annual update
  - No-cost download
- http://freescale.com/sdk



Freescale Linux SDK Highlights

- Freescale Linux Investment
  - Hundreds of man-years per year
  - Global Board Farms
  - Top 15 Company Contributor to `kernel.org`
  - Systems Designed, Tightly Integrated with Freescale SOC
  - Systems Validated - Tightly aligned with Freescale NPI
- Quality
  - ISO-9000 Quality Processes – Externally Audited
  - On-going Maintenance, regular kernel updates (LTSi)
  - Open Source Compatible – Upstreamed, Dedicated Team
    - Standards based – Yocto
- Ease of Use
  - Common Kernel Across support platforms
  - Combined P, T and Layerscape support in unified SDK



Segment Solutions Reference Platform Solutions

		
Enterprise AP-WLAN (802.11ac) (Single & Dual-core P10xx)	Security Appliance	IoT Smart Wireless Networked Smart Gateway P1, 8308, Layerscape LS1
		
DLNA VOD Video-on-Demand NVR – Network Video Recorder (Video Surveillance)-P1, P2041	Multi-service Gateway (Media & Voice Gateway NAS, NVR Video Server…)	NAS – Network Attached Storage (P1: 1Gb) SAN – Storage Area Network (P2 – P5: T4: 10Gb)
		
Industrial Gateway & Data Concentrator (Smart Grid) – P1	913x-WLAN Small Cell	T4/T2 1U Appliance (UTM, etc), L2/L3 Switch, L4-L7 Secured ADC

Data-Center SDN SAN Storage iNIC, vNIC T4/T2 PCIe Card

Business Models

Product	License Fee	Royalty	Paid Support	Engineering Services
Linux / Android Support & Services			✓	✓
VortiQa Run-Time Software	✓	✓	✓	✓
Embedded Applications			✓	✓
PEG	✓		✓	✓
Development Tools	✓		✓	

freescale VortiQa

freescale CodeWarrior

Freescale MQX Software

VortiQa Portfolio

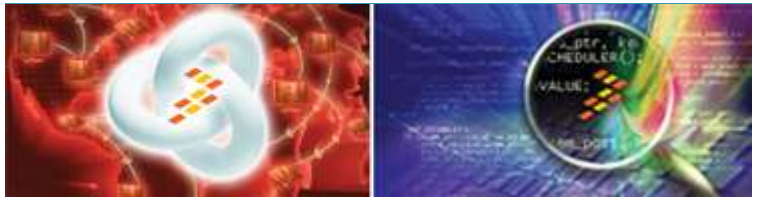
Module	Application
VortiQa—Application Identification Suite Deep Packet Inspection: Identify the type of traffic on the network	**VortiQa—Mobility Transport** Transport Functions: IPsec, QoS, PDCP and GTP

NEXT-GENERATION TECHNOLOGY

SDN—Open Network Switch	SDN—Open Network Director
VortiQa—Software Defined Networks (SDN)—Switch Data Plane Functions: layer 2, layer 3 and above; Firewall, NAT, QoS, DPI, etc. POC available; product in planning	**VortiQa—Software Defined Networks (SDN)—Controller** Controller Functions: PoC available; product in planning

Networking Software and Services Group

- ***Accelerate*** **Customer Time to Market**
  - Speed Adoption of Multicore
  - Dedicated expert staff with access to software and SoC teams



- ***Deliver*** **Commercial Software, Support, Services and Solutions**
  - Commercial Software: VortiQa, CodeWarrior, Processor Expert
  - Accelerate new technology adoption

- ***Simplify*** **Software Engagement with Freescale**
  - Consolidate Freescale software and solutions
  - Streamline business processes

- ***Create*** **Success!**
  - Partner with customers
  - Leverage *your* strengths, add *our* capabilities

Introduction to the **QorIQ Configuration Suite (QCS)**

QorIQ Software Enablement Strategy

- Provide development systems
  - Complete boards for evaluation of QorIQ devices
- Provide Runtime software for QorIQ products
  - Hypervisor, Linux BSPs, Reference Designs
- Provide bring-up tools and development systems
  - GNU tools, CodeWarrior debuggers, probes, boards
- Provide configuration tools to support your application of QorIQ on your custom board.
  - RCW, BootROM, Pin Mux
  - DDR Configuration
  - Device Tree Editing
- Provide optimization tools to support runtime visibility into these complex parts to help calibrate and debug your systems.
  - DDR validation tool to ensure DDR functionally configured for custom board
  - Serdes validation with internal and external loopback
  - Scenarios tool for collecting and visualizing runtime trace data
  - Packet tool for understanding the flow of packets within a QorIQ device

QorIQ Configuration Suite

The configuration suite supports the generation of valid configurations of a QorIQ / Qonverge part for a custom board design complimentary to boot loaders (aka uboot).

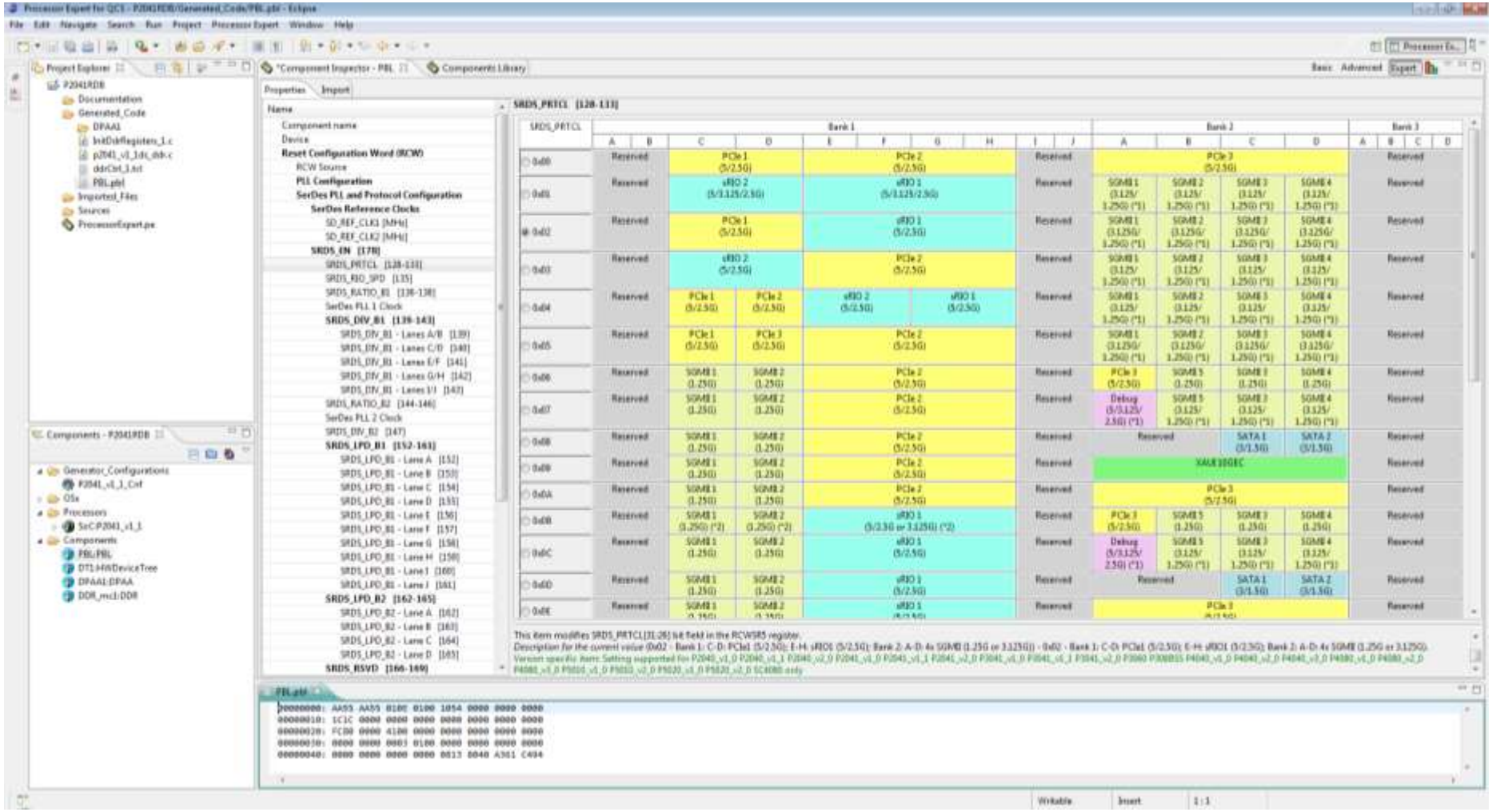
- **Pre-boot loader / RCW configuration**
  Defines the Reset Control Word configuration for pre-boot configuration
- **DDR Configuration Tool**
  Configures the DDR controllers, supports SPD, validates configuration on-chip
- **Boot ROM Tool**
  Supports configuration of pin strapping and Bootrom process in P1/P2 devices
- **Device Tree Editor**
  Supports visual editing of device trees

Why QorIQ Configuration Suite?

- Configuration of QorIQ processors is increasing in complexity
  - Even more complexity is around the corner
  - We support many, many configuration settings
- Reference manuals are huge and intimidating to new customers
- Configuration problems during board bring-up are HARD and COSTLY
- Learning command line tools requires more training, etc.

- Solution/Strategy to solve these problems:
  - Extensible suite of tools with a common user interface
  - Consolidate into a common tools framework (Processor Expert)
  - Provide new device support aligned with silicon roadmap
  - Add more configuration tools over time
  - Allow customers to add their own configuration tools to extend what we offer

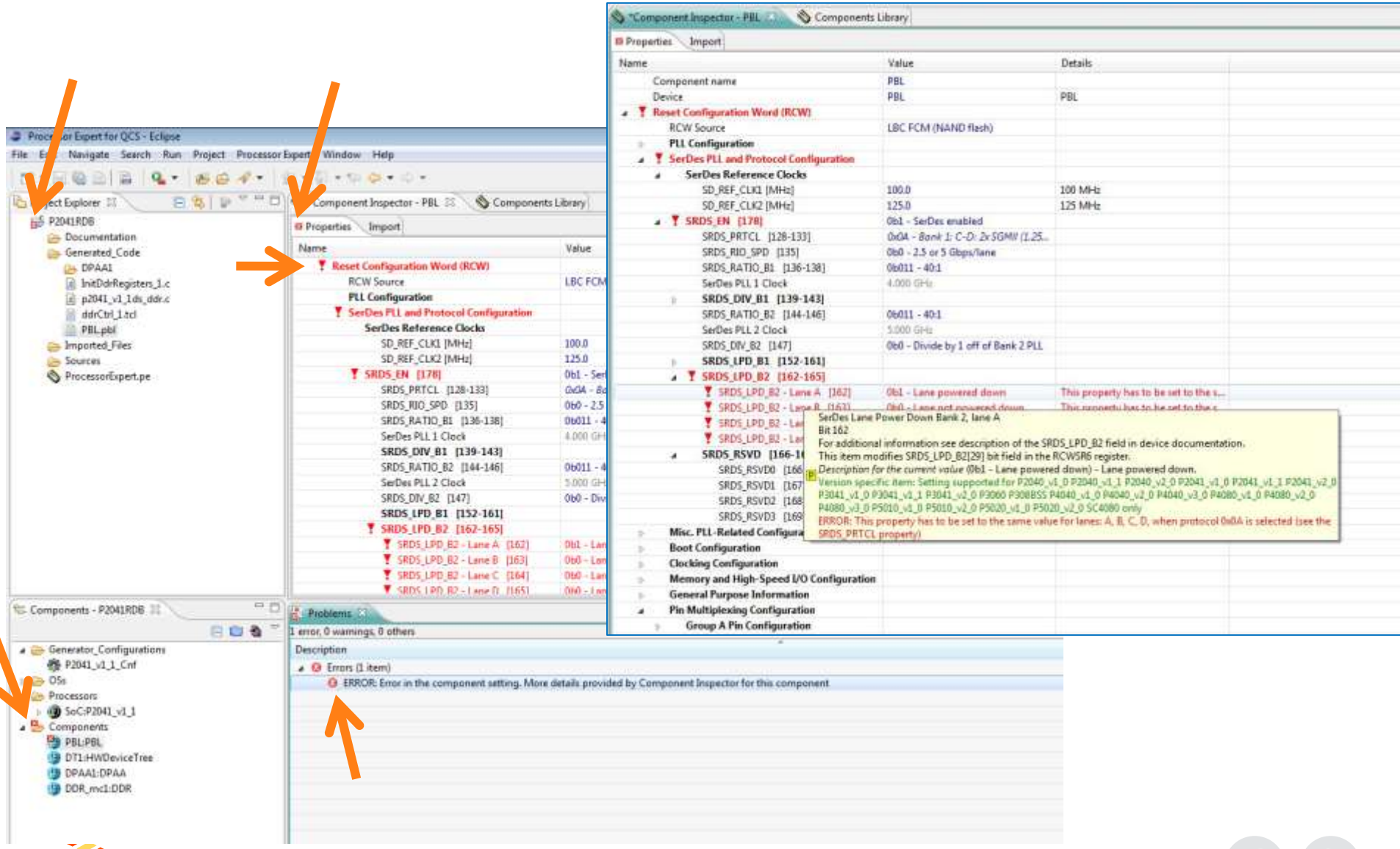
Processor Expert for QorIQ – Configuration Suite



Pre-Boot Loader

RCW Configuration Tool

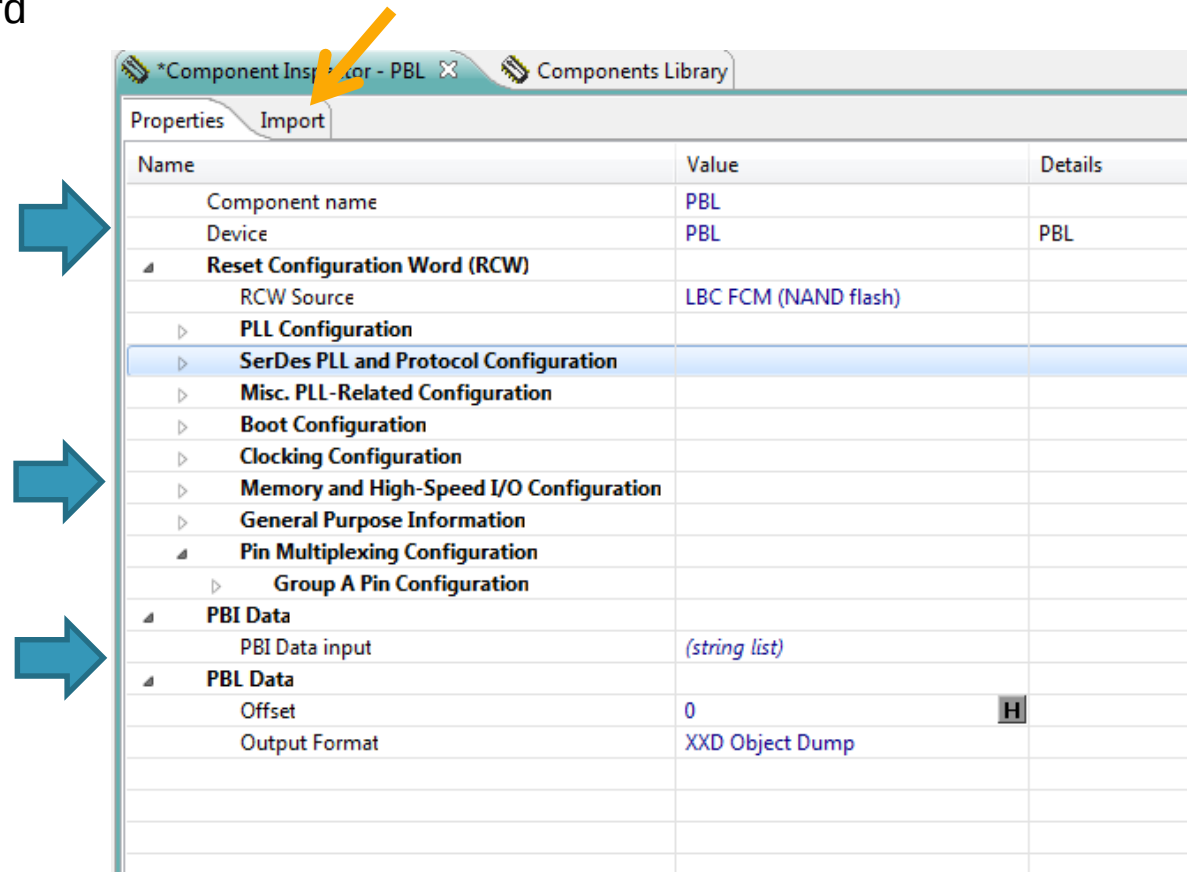
Pre-Boot Loader (RCW) Configuration



Pre-Boot Loader Standard Component Interface

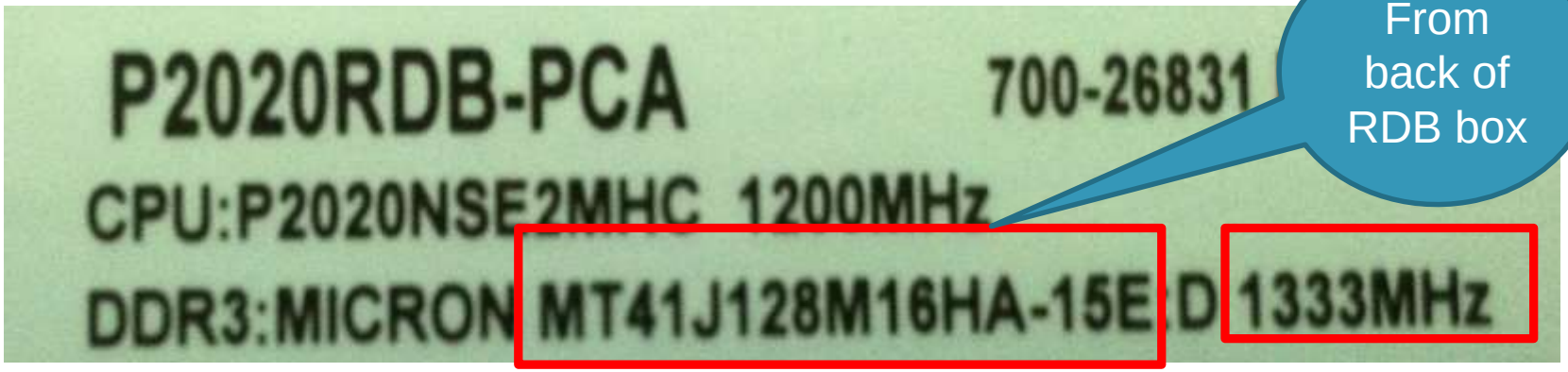
- Pre-Boot Loader (PBL) tool establishes all Reset Control Word settings
- PLL Configurations
- SerDes Configuration
- Pin Muxing Configuration
- Output format selection
- Possibility to add PBI data

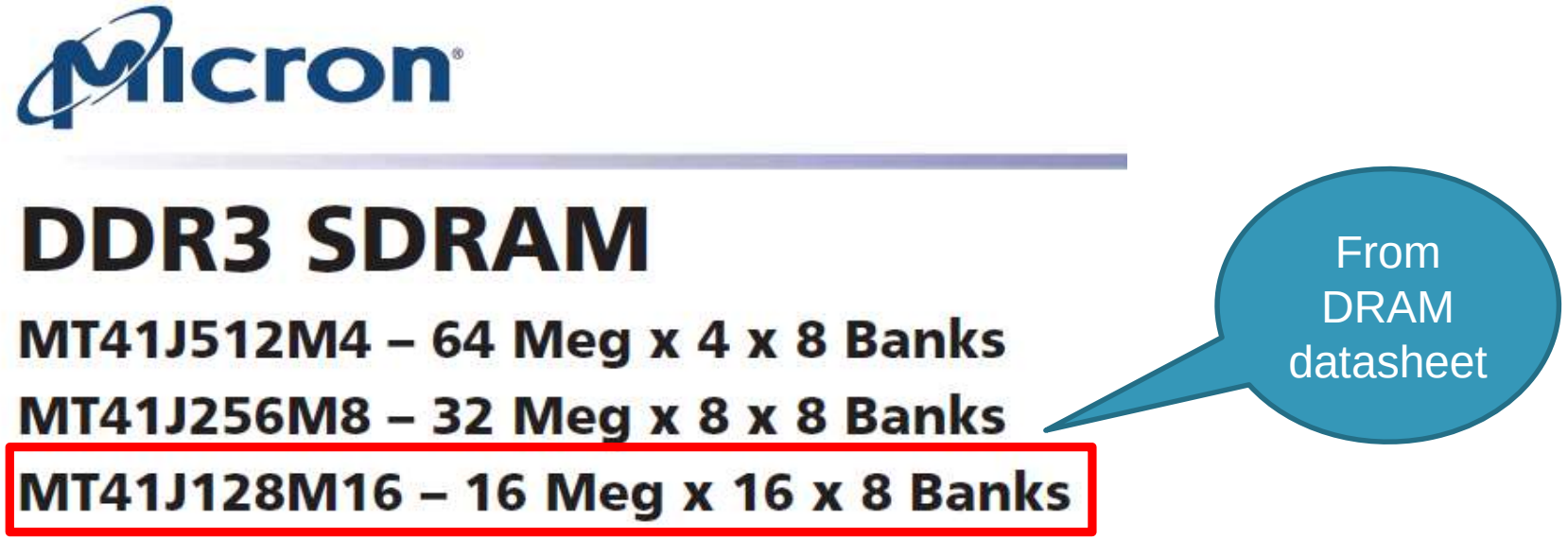
Possibility to import RCW settings



DDR Configuration

Get DRAM Information – P2020RDB-PCA





How about the Rest of the Timing Parameters?

Table 1: Key Timing Parameters

Speed Grade	Data Rate (MT/s)	Target ^{t}RCD-^{t}RP-CL	^{t}RCD (ns)	^{t}RP (ns)	CL (ns)
-093[1, 2, 3, 4]	2133	14-14-14	13.09	13.09	13.09
-107[1, 2, 3]	1866	13-13-13	13.91	13.91	13.91
-125[1, 2]	1600	11-11-11	13.75	13.75	13.75
-15E[1]	1333	9-9-9	13.5	13.5	13.5
-187E	1066	7-7-7	13.1	13.1	13.1

Features

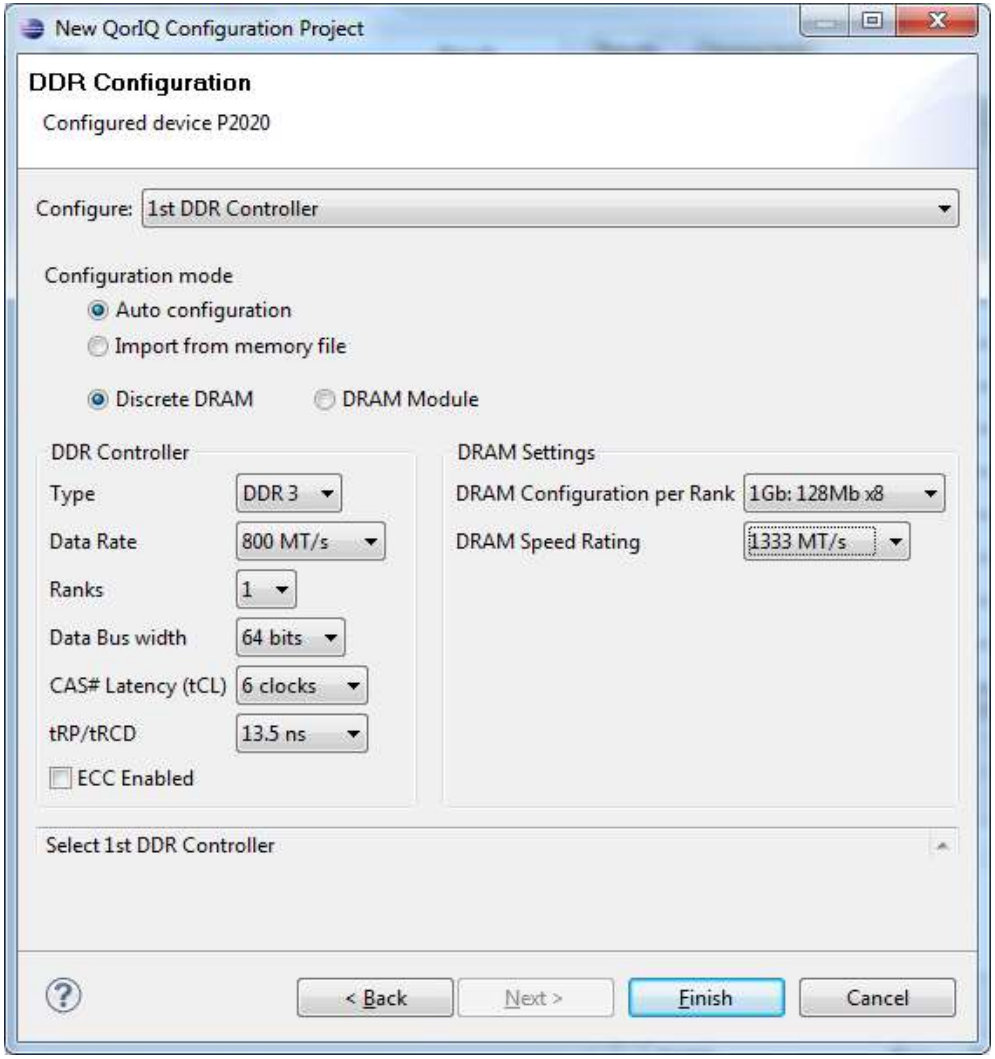
- $V_{DD} = V_{DDQ}$ = 1.5V ±0.075V
- 1.5V center-terminated push/pull I/O
- Differential bidirectional data strobe
- 8*n*-bit prefetch architecture
- Differential clock inputs (CK, CK#)
- 8 internal banks
- Nominal and dynamic on-die termination (ODT) for data, strobe, and mask signals
- Programmable CAS READ latency (CL)
- Posted CAS additive latency (AL)
- Programmable CAS WRITE latency (CWL) based on ^{t}CK
- Fixed burst length (BL) of 8 and burst chop (BC) of 4 (via the mode register set [MRS])
- Selectable BC4 or BL8 on-the-fly (OTF)
- Self refresh mode
- T_C of 0°C to 95°C
  - 64ms, 8192 cycle refresh at 0°C to 85°C
  - 32ms, 8192 cycle refresh at 85°C to 95°C
- Self refresh temperature (SRT)
- Write leveling
- Multipurpose register
- Output driver calibration

Options[1]	Marking
• Configuration	
– 512 Meg x 4	512M4
– 256 Meg x 8	256M8
– 128 Meg x 16	128M16
• FBGA package (Pb-free) – x4, x8	
– 78-ball (8mm x 10.5mm) Rev. H,M,J,K	DA
– 78-ball (9mm x 11.5mm) Rev. D	HX
• FBGA package (Pb-free) – x16	
– 96-ball (9mm x 14mm) Rev. D	HA
– 96-ball (8mm x 14mm) Rev. K	JT
• Timing – cycle time	
– 938ps @ CL = 14 (DDR3-2133)	-093
– 1.071ns @ CL = 13 (DDR3-1866)	-107
– 1.25ns @ CL = 11 (DDR3-1600)	-125
– 1.5ns @ CL = 9 (DDR3-1333)	-15E
– 1.87ns @ CL = 7 (DDR3-1066)	-187E
• Operating temperature	
– Commercial (0°C ≤ T_C ≤ +95°C)	None
– Industrial (–40°C ≤ T_C ≤ +95°C)	IT
• Revision	:D/:H/:J/:K/:M

Note: 1. Not all options listed can be combined to define an offered product. Use the part catalog search on http://www.micron.com for available offerings.

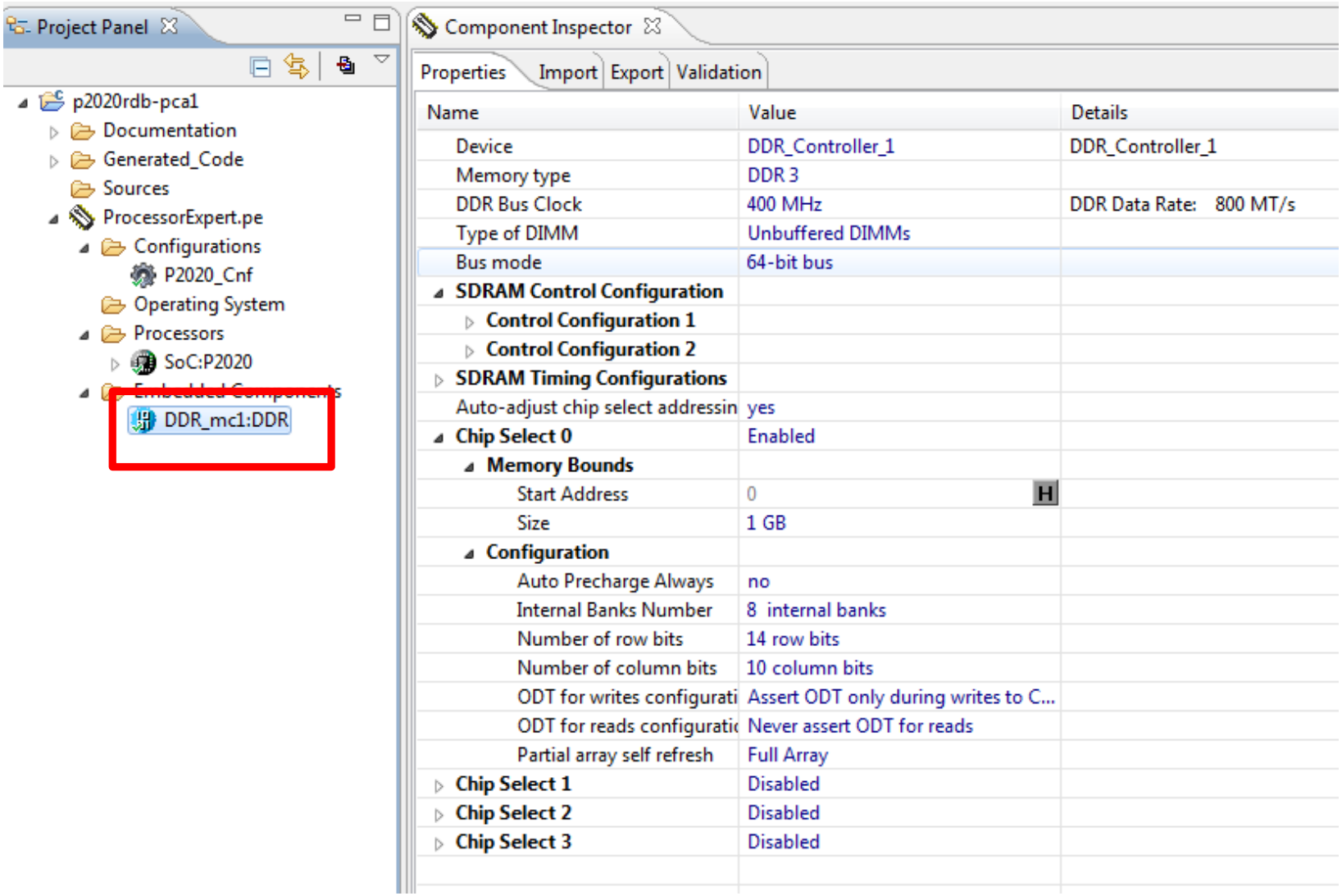
- Tool automatically computes tRCD, tRP, and CL!
  - User can change these values if required.

DDR Wizard Simplifies Configuration

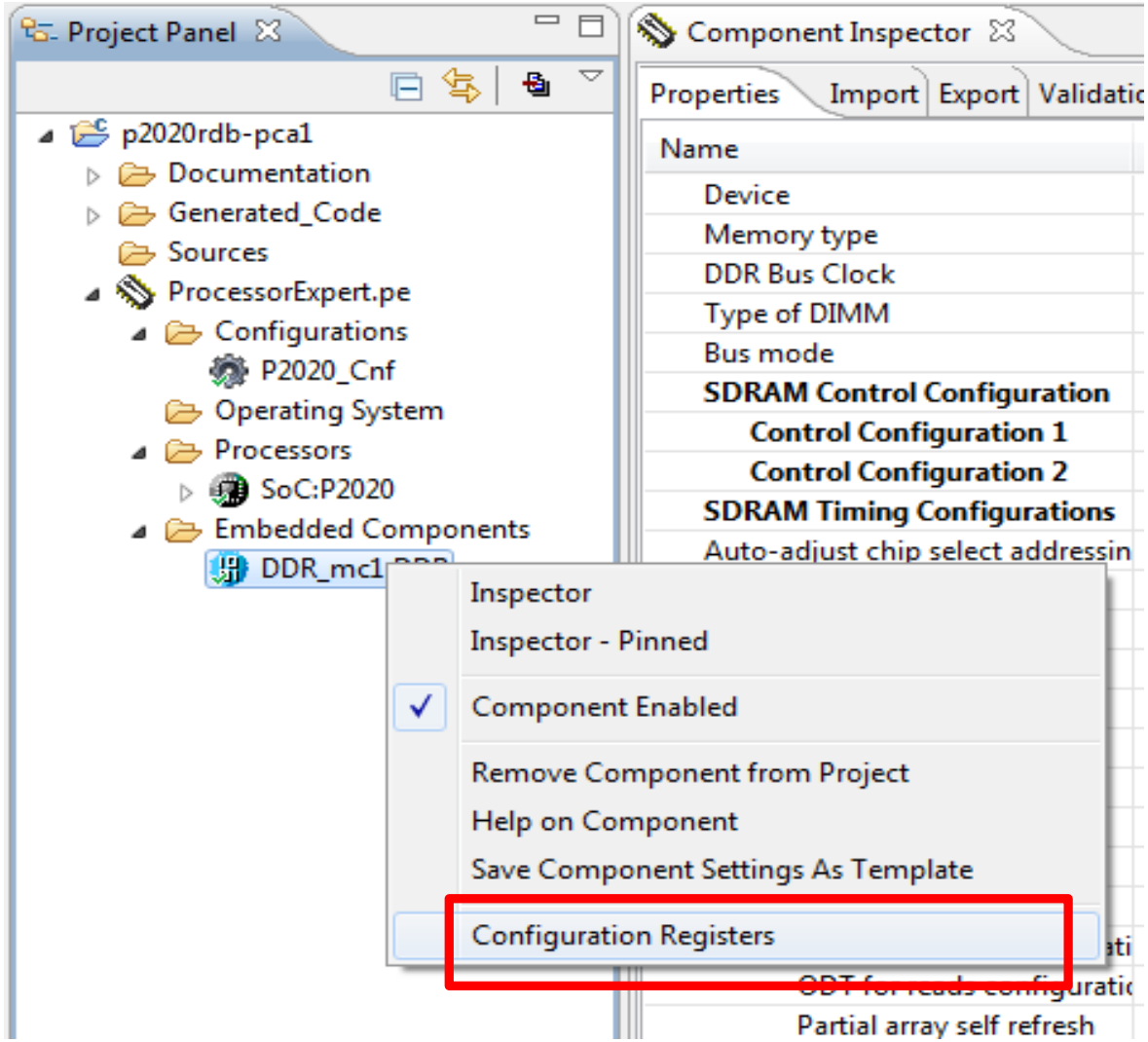


- From memory data sheet:
  - Maximum speed rating
  - Capacity

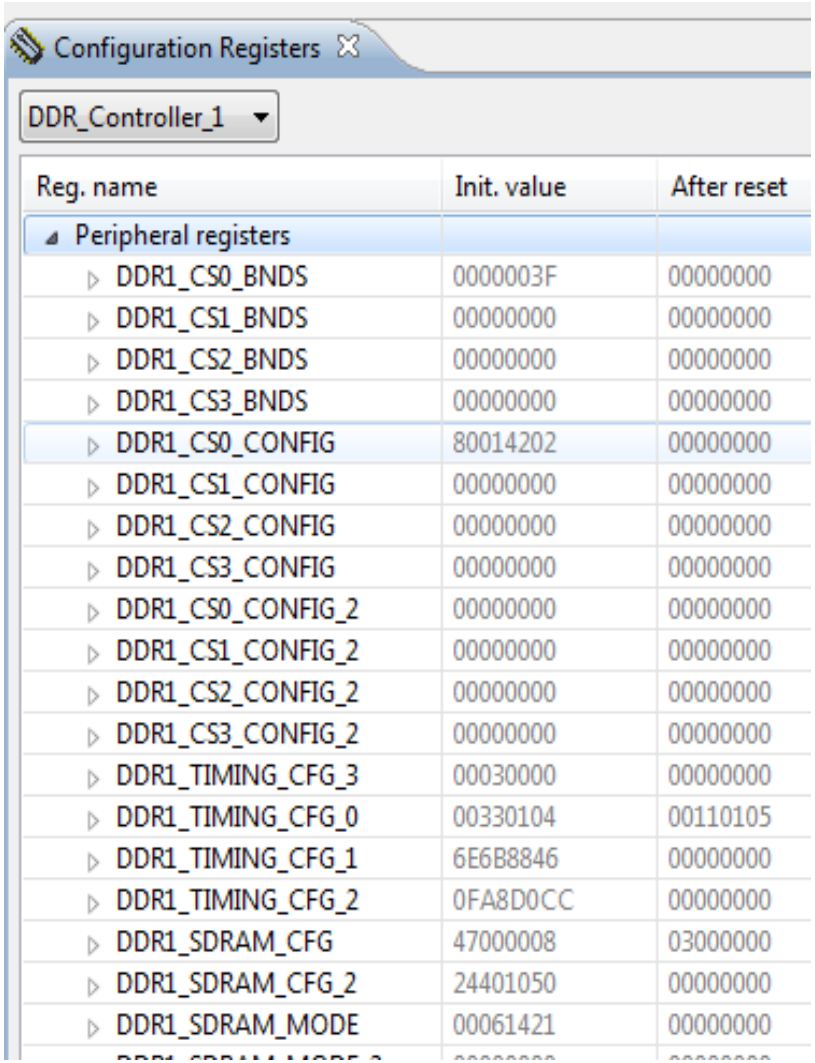
QCS Project Explorer



Review DDR Registers Values

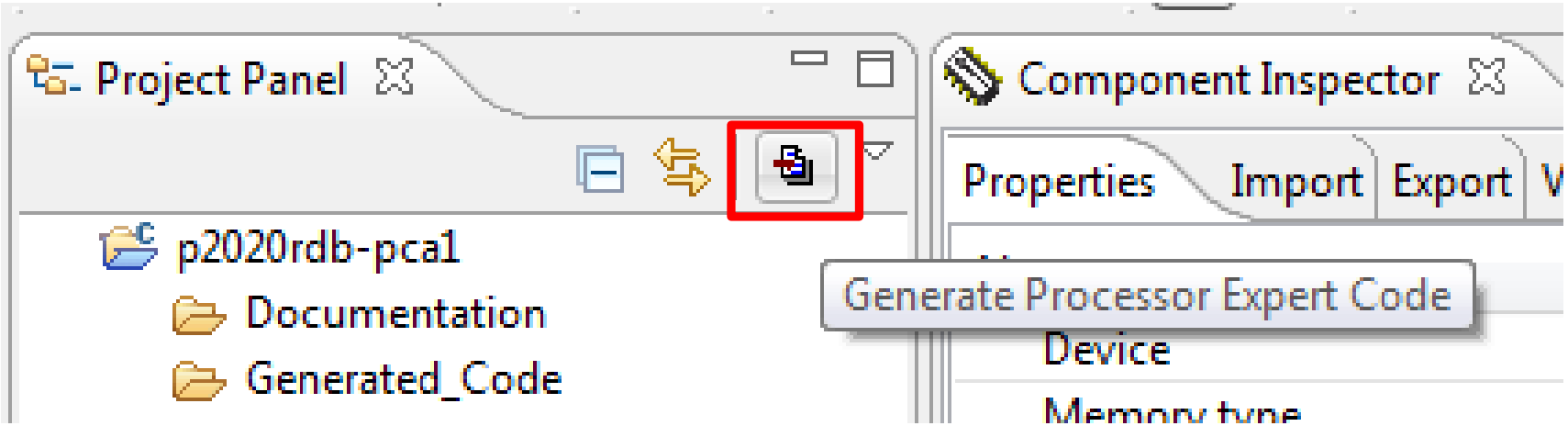


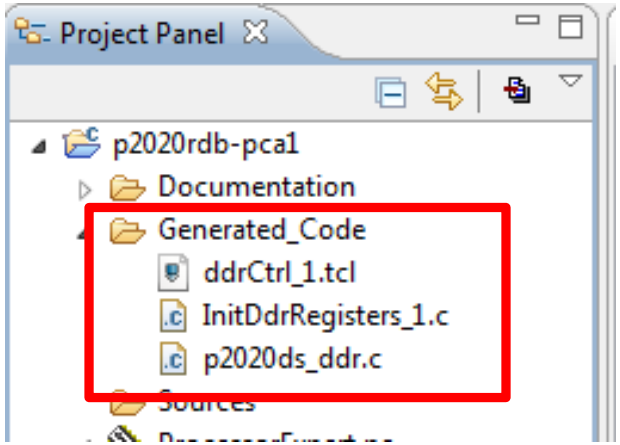
Review DDR Registers Values (continued)



Reg. name	Init. value	After reset
Peripheral registers		
DDR1_CS0_BNDS	0000003F	00000000
DDR1_CS1_BNDS	00000000	00000000
DDR1_CS2_BNDS	00000000	00000000
DDR1_CS3_BNDS	00000000	00000000
DDR1_CS0_CONFIG	80014202	00000000
DDR1_CS1_CONFIG	00000000	00000000
DDR1_CS2_CONFIG	00000000	00000000
DDR1_CS3_CONFIG	00000000	00000000
DDR1_CS0_CONFIG_2	00000000	00000000
DDR1_CS1_CONFIG_2	00000000	00000000
DDR1_CS2_CONFIG_2	00000000	00000000
DDR1_CS3_CONFIG_2	00000000	00000000
DDR1_TIMING_CFG_3	00030000	00000000
DDR1_TIMING_CFG_0	00330104	00110105
DDR1_TIMING_CFG_1	6E6B8846	00000000
DDR1_TIMING_CFG_2	0FA8D0CC	00000000
DDR1_SDRAM_CFG	47000008	03000000
DDR1_SDRAM_CFG_2	24401050	00000000
DDR1_SDRAM_MODE	00061421	00000000

Generate DDR Configuration





Generated Files – CW, uboot, ddrinit.c

```
# DDR Controller 1 Registers

  # DDR_SDRAM_CFG
  mem [0xFF702110] = 0x47000008

  # CS0_BNDS
  mem [0xFF702000] = 0x3F

  # CS0_CONFIG
  mem [0xFF702080] = 0x80014202

  # CS0_CONFIG_2
  mem [0xFF7020C0] = 0x00

  # TIMING_CFG_3
  mem [0xFF702100] = 0x00030000

  # TIMING_CFG_0
  mem [0xFF702104] = 0x00330104

  # TIMING_CFG_1
  mem [0xFF702108] = 0x6E6B8846

  # TIMING_CFG_2
  mem [0xFF70210C] = 0x0FA8D0CC

  # DDR_SDRAM_CFG_2
  mem [0xFF702114] = 0x24401050

  # DDR_SDRAM_MODE
  mem [0xFF702118] = 0x00061421
```

```
#define DDR_1_INIT_EXT_ADDR_ADDR     0xFF70214C
#define DDR_1_SDRAM_RCW_1_ADDR       0xFF702180
#define DDR_1_SDRAM_RCW_2_ADDR       0xFF702184
#define DDR_1_DATA_INIT_ADDR         0xFF702128
#define DDR_1_SDRAM_MD_CNTL_ADDR     0xFF702120
#define DDR_1_DDRCDR_1_ADDR          0xFF702B28
#define DDR_1_DDRCDR_2_ADDR          0xFF702B2C

#define SDRAM_CFG_MEM_EN_MASK        0x80000000
#define SDRAM_CFG2_D_INIT_MASK       0x00000010

/* DDR Controller configured registers' values */
#define DDR_1_CS0_BNDS_VAL           0x3F
#define DDR_1_CS1_BNDS_VAL           0x00
#define DDR_1_CS2_BNDS_VAL           0x00
#define DDR_1_CS3_BNDS_VAL           0x00
#define DDR_1_CS0_CONFIG_VAL         0x80014202
#define DDR_1_CS1_CONFIG_VAL         0x00
```

```
#define PEX_CONFIG_DDR1_INIT_EXT_ADDR     0x00000000
#define PEX_CONFIG_DDR1_TIMING_4          0x00220001
#define PEX_CONFIG_DDR1_TIMING_5          0x02401400
#define PEX_CONFIG_DDR1_ZQ_CNTL           0x89080600
#define PEX_CONFIG_DDR1_WRLVL_CNTL        0x8655F614
#define PEX_CONFIG_DDR1_RCW_1             0x00000000
#define PEX_CONFIG_DDR1_RCW_2             0x00000000


/* DDR Controller 1 configuration global structures */
fsl_ddr_cfg_regs_t ddr_cfg_regs_0 = {
  .cs[0].bnds = PEX_CONFIG_DDR1_CS0_BNDS,
  .cs[1].bnds = PEX_CONFIG_DDR1_CS1_BNDS,
  .cs[2].bnds = PEX_CONFIG_DDR1_CS2_BNDS,
  .cs[3].bnds = PEX_CONFIG_DDR1_CS3_BNDS,
  .cs[0].config = PEX_CONFIG_DDR1_CS0_CONFIG,
  .cs[1].config = PEX_CONFIG_DDR1_CS1_CONFIG,
  .cs[2].config = PEX_CONFIG_DDR1_CS2_CONFIG,
```

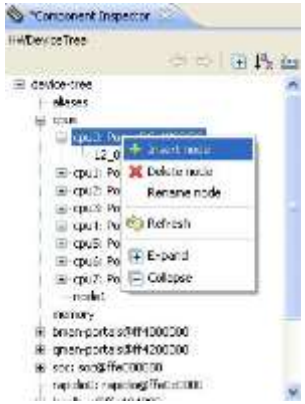
Device Tree Editor

Supports Hardware Device Trees (*.dts)

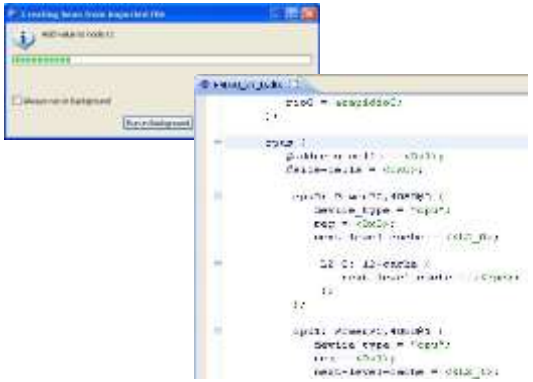
Hardware Device Tree Workflow

Create Project

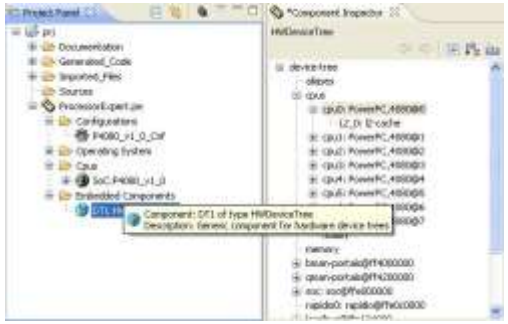
Configure Component



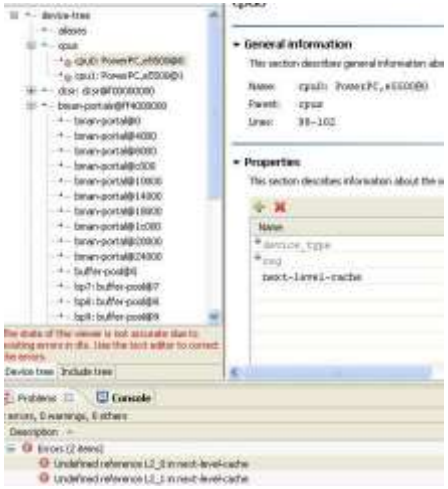
Generate Code



Select Component

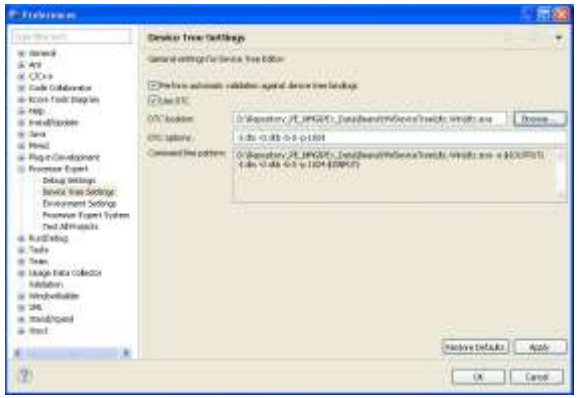


Validate Component



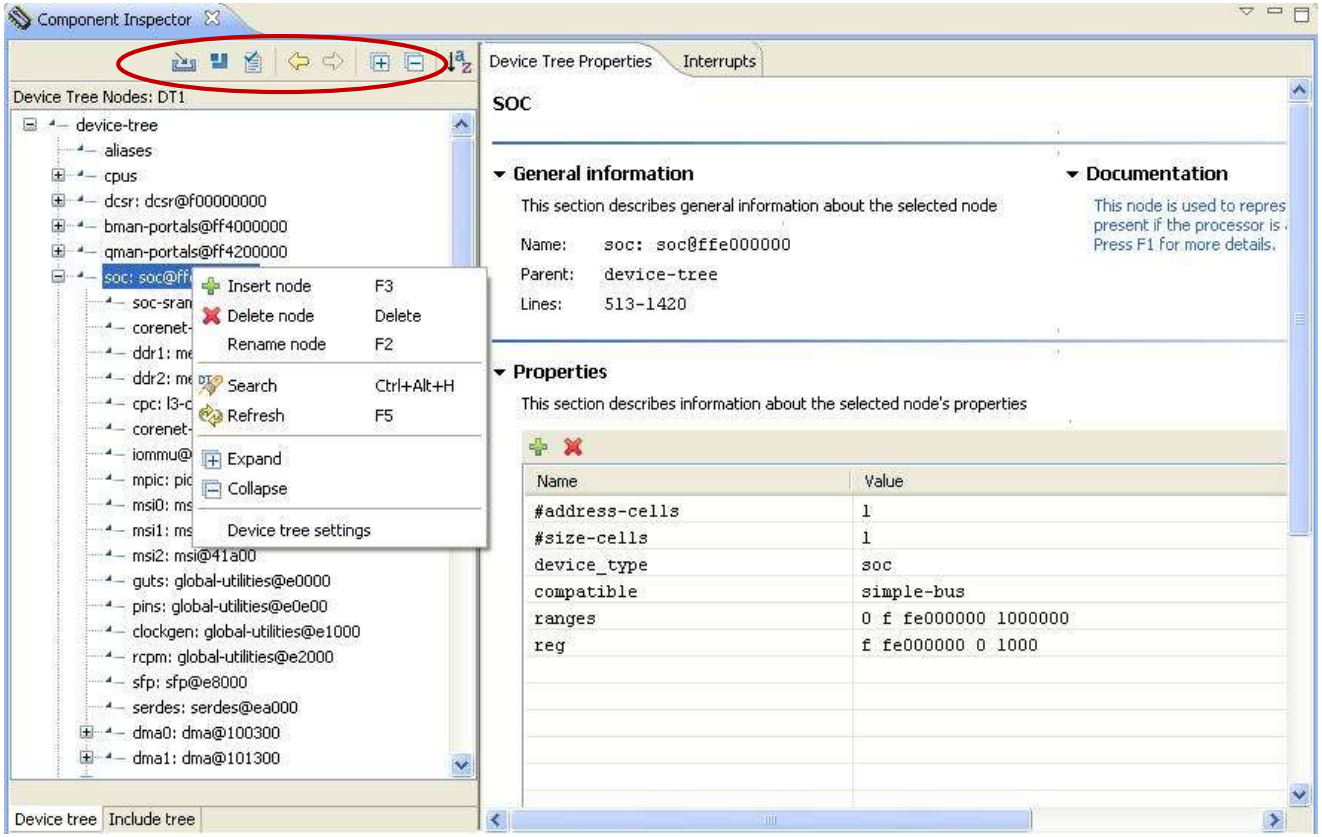
Compile DTS

```
#!/bin/bash
dtc -f -b 0 -p 0x8000 -R 8 -I dts -O dtb $1.dts
```

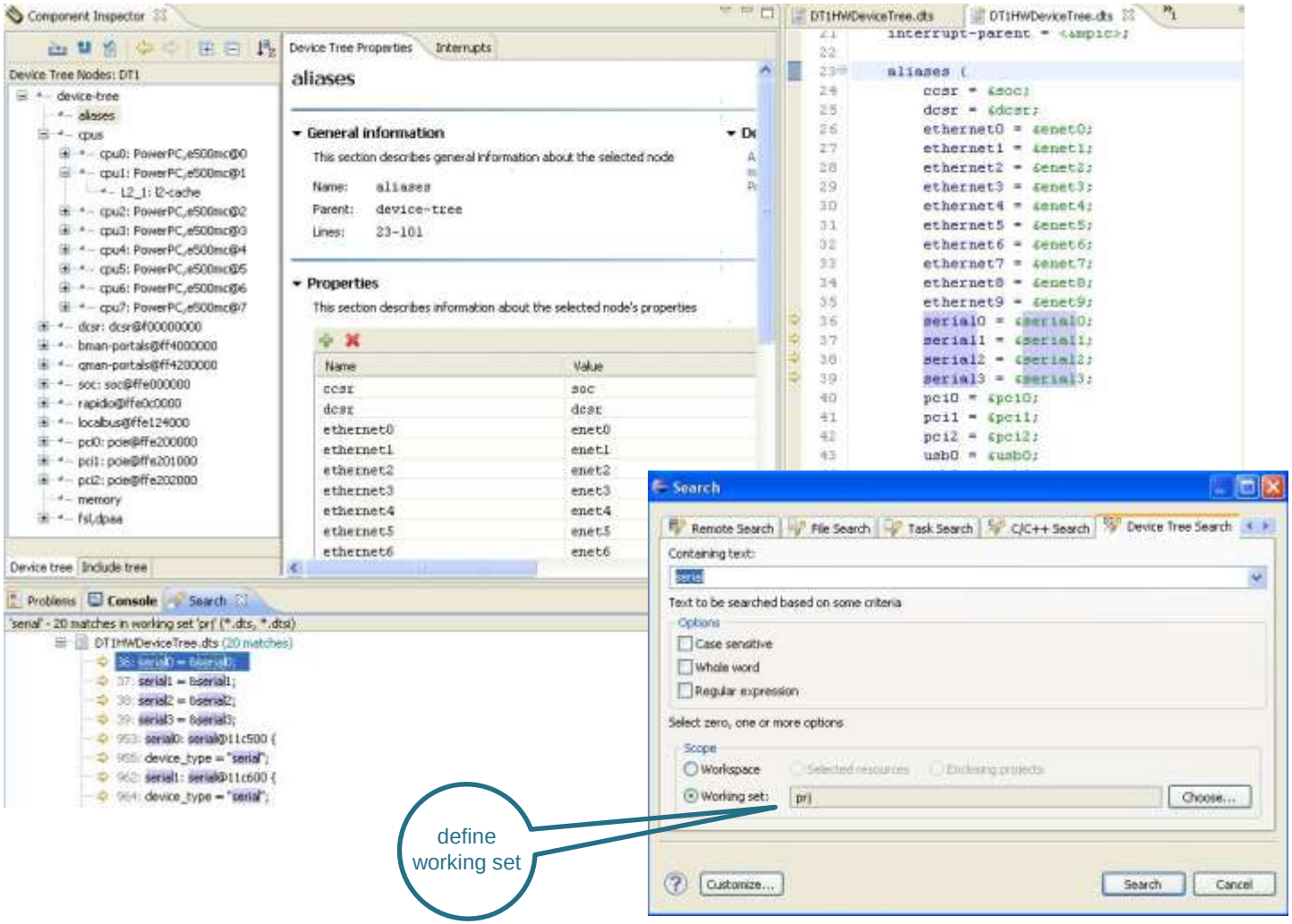


Explorer Tree View

- Operations on nodes
  - Go back / forward
  - Expand/collapse
  - Ascending/descending sort
  - Insert node
  - Delete node
  - Rename node
- Other operations
  - Import device tree
  - Include device tree
  - Validate device tree
  - Search in device tree

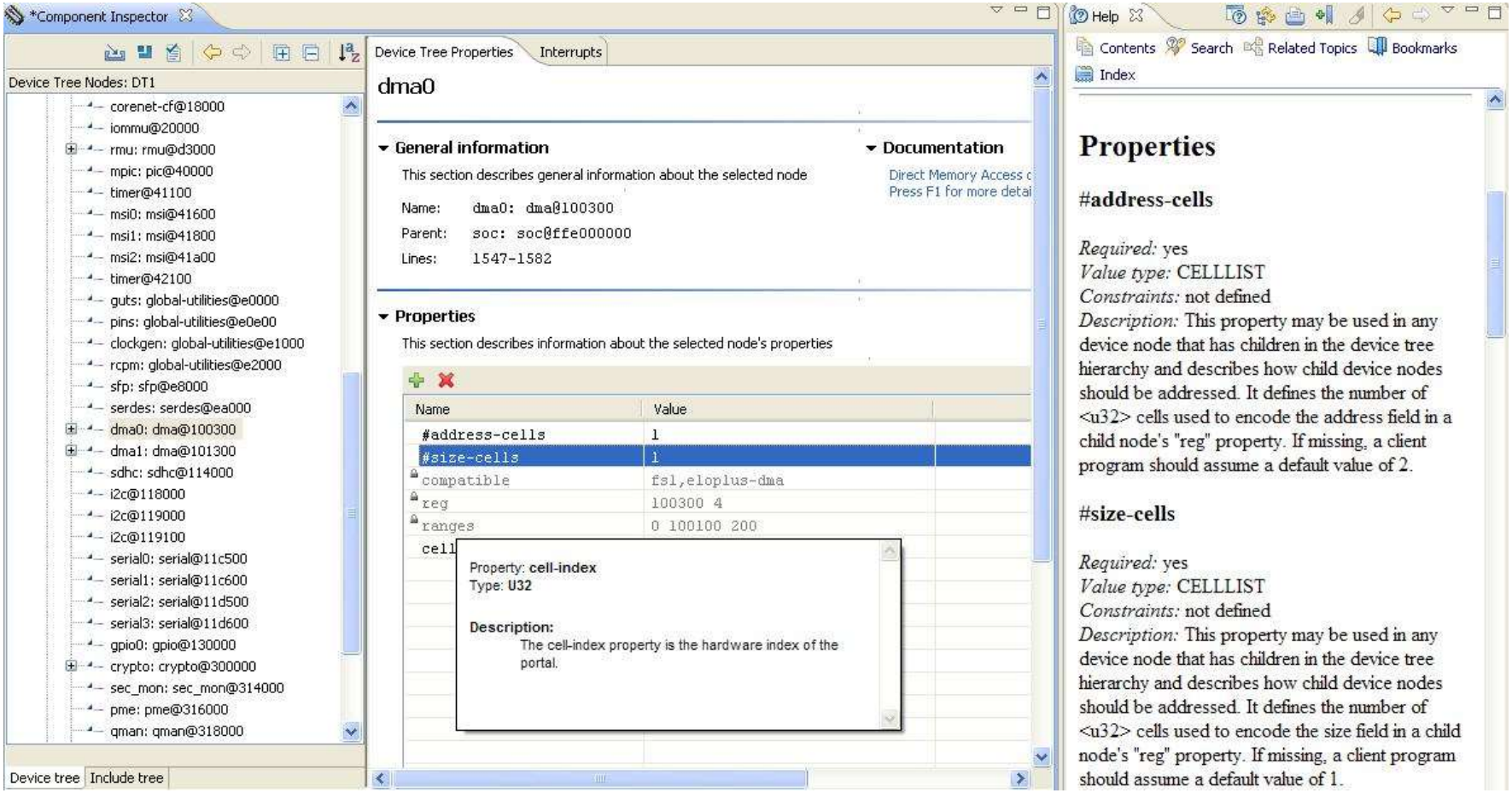


Search Capability



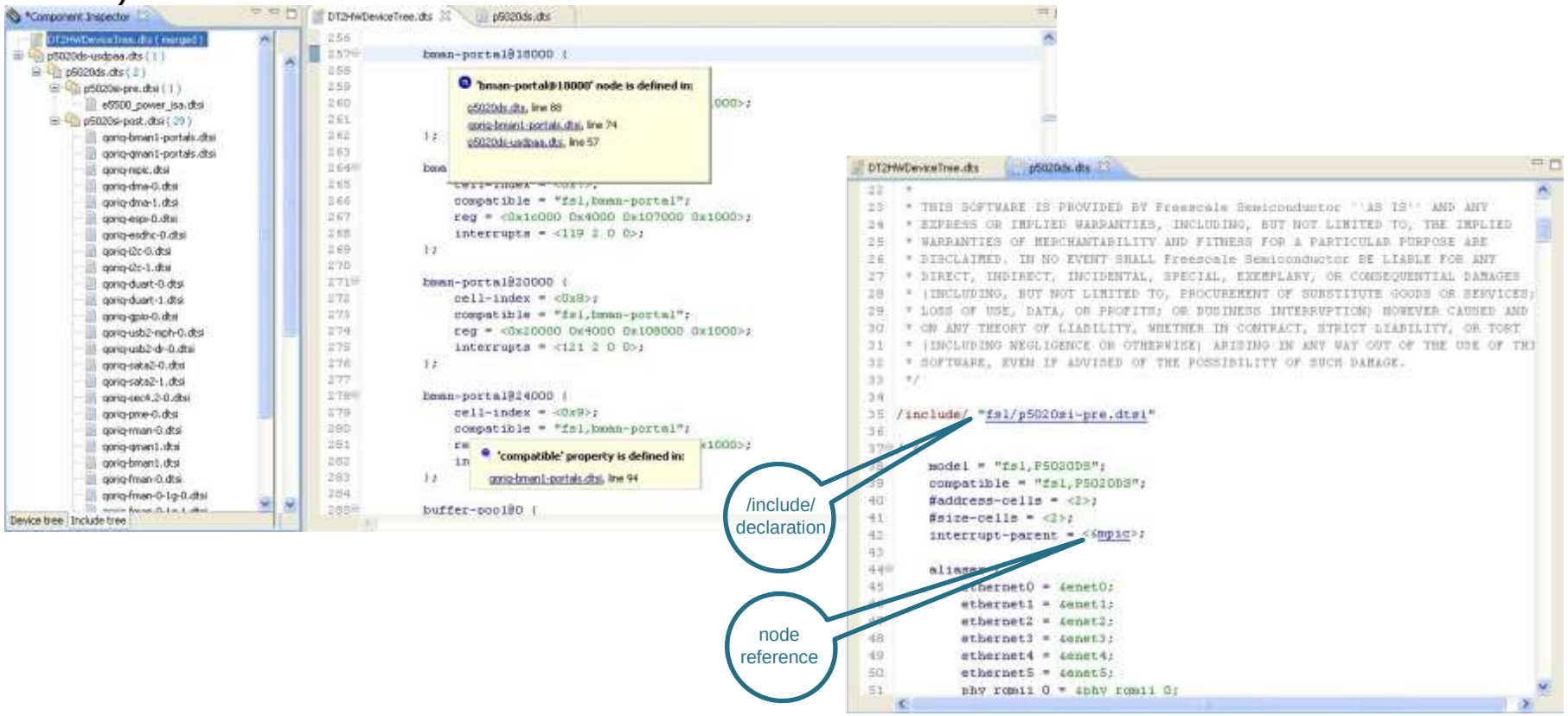
Device Tree Bindings

- Each node has a “binding” representing its schema. It describes what properties are optional or required and what each means.



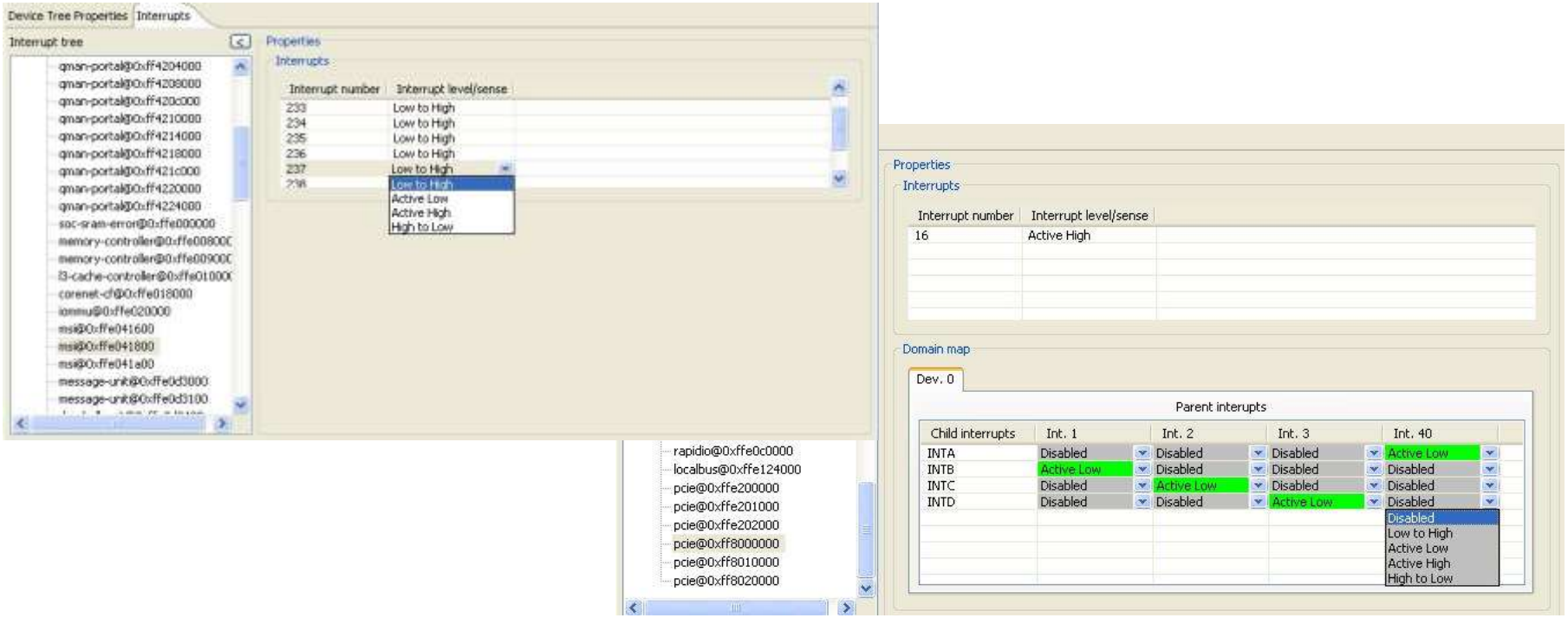
Device Trees Inclusion

- The Include tree allows easy navigation among device tree fragments (dts, dtsi)
- Hovering support for properties and nodes: a tool-tip appears displaying their initial locations
- Hyperlink detection for /include/ declarations and device tree references (Ctrl + left click)



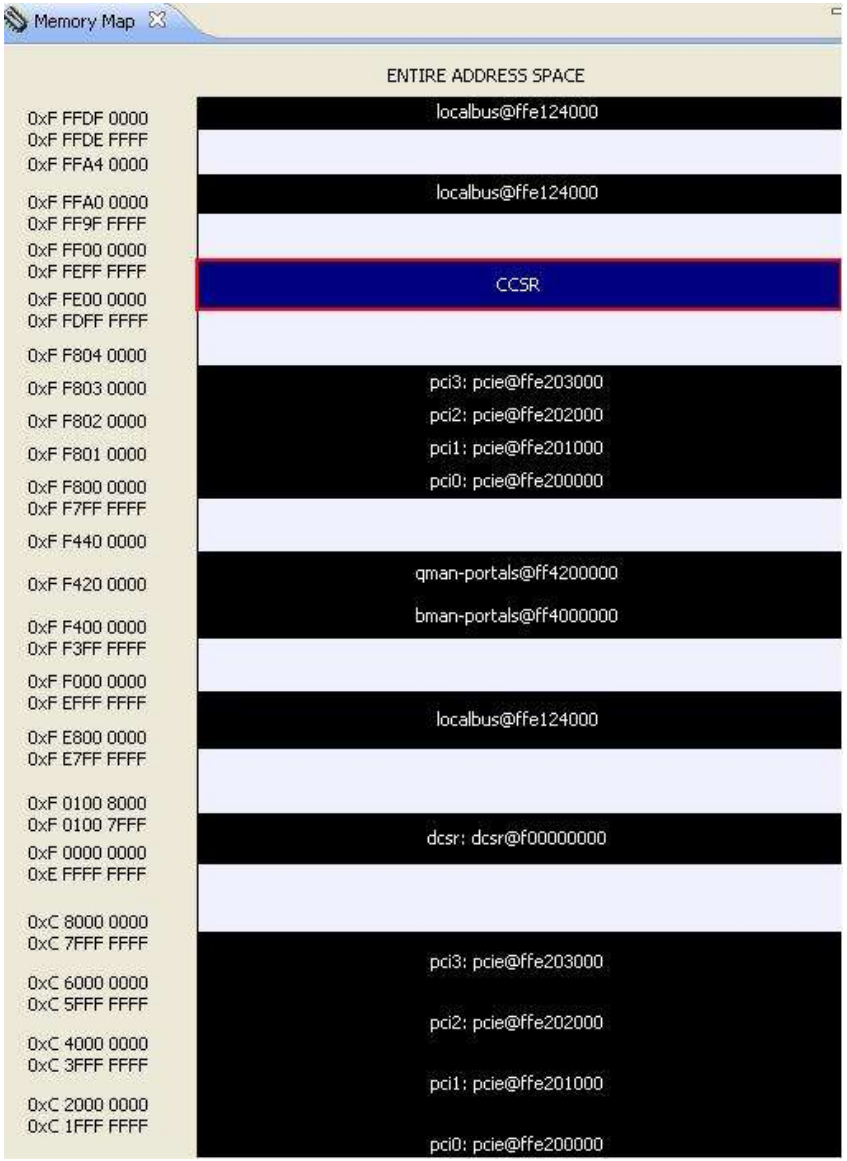
Interrupts Tree

- The Interrupts tree represents the hierarchy and routing of interrupts in the platform hardware
- The left side displays the actual representation of the Interrupt Tree starting from the root interrupt controller
- The right side displays the interrupts sources for the selected device tree node



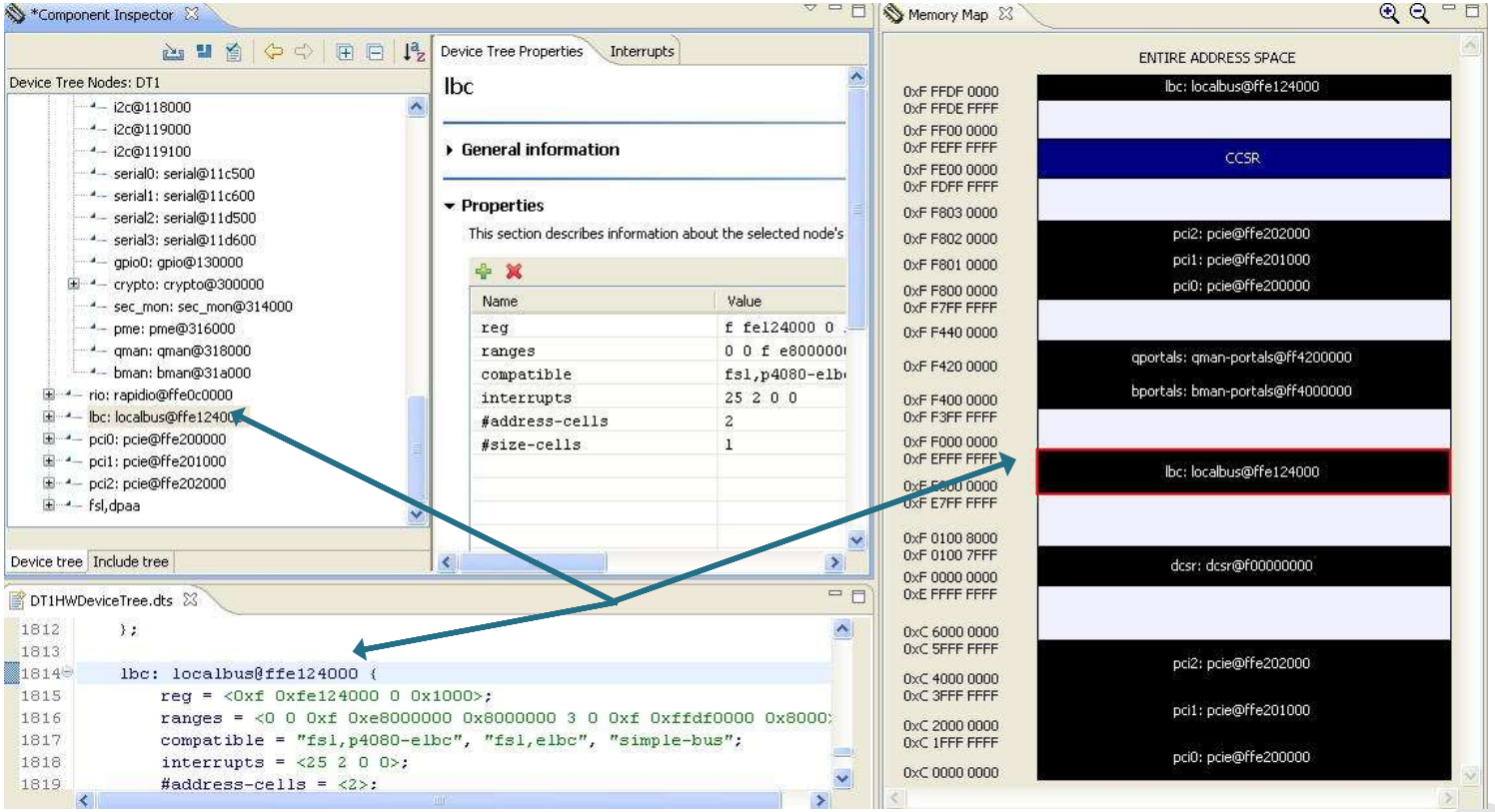
Memory Map view

- Any hw device tree can be seen as a representation of different Local Access Windows (LAW)
- Each LAW maps to a specified target interface, such as DDR Controller, Localbus, PCI Express, etc.
- Each device tree node having reg and ranges properties defines a memory range inside/outside Configuration Control and Status Register (CCSR) space area
- The Memory Map view pops-up automatically when a device tree component is selected inside Component Inspector view



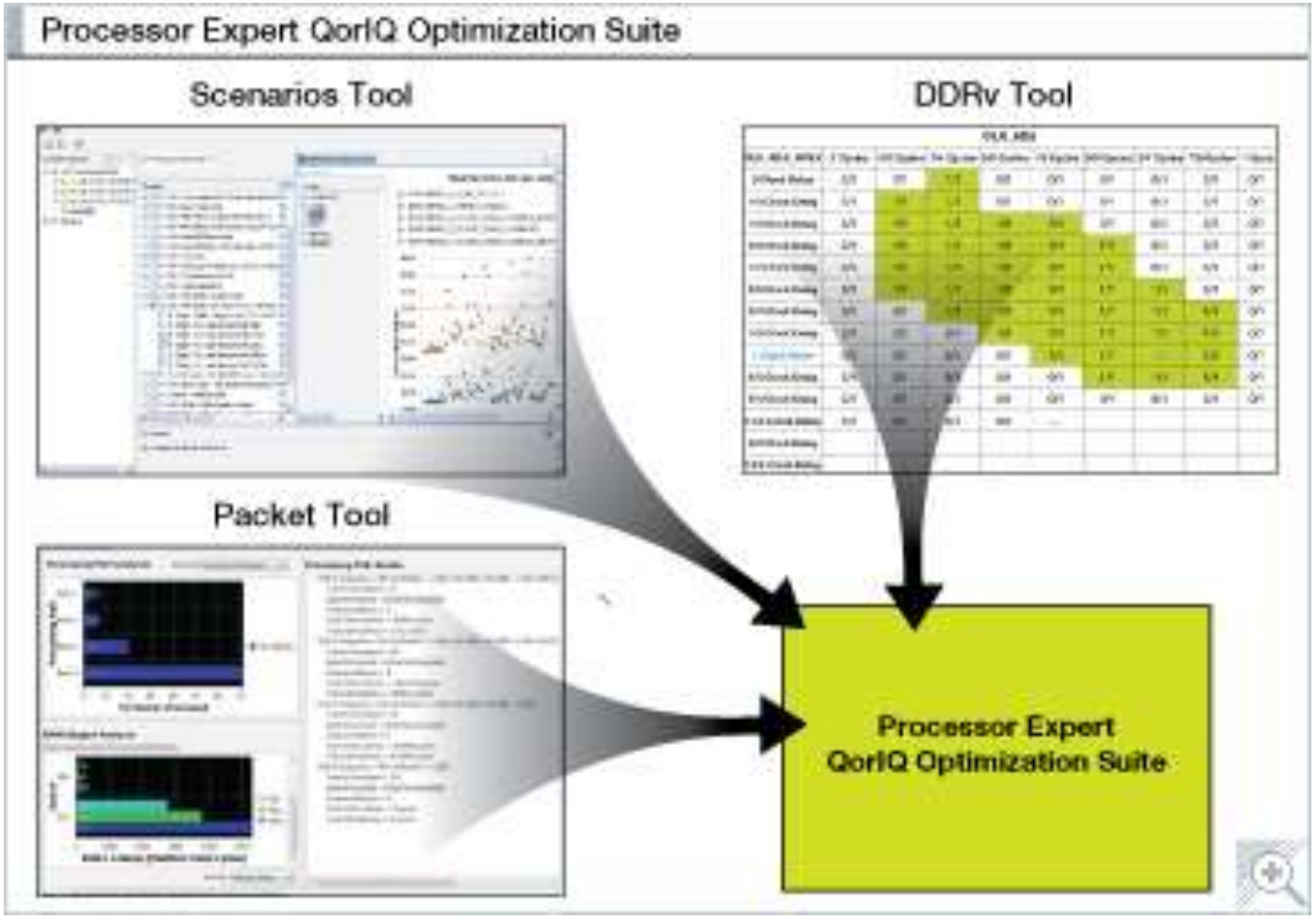
Device Tree Views Synchronization

- Device tree views
- GUI <=> text editor symmetry
- Memory map view => GUI editor symmetry
- Modifications are reflected in all editors



QorIQ Optimization Suite

QorIQ Optimization Suite



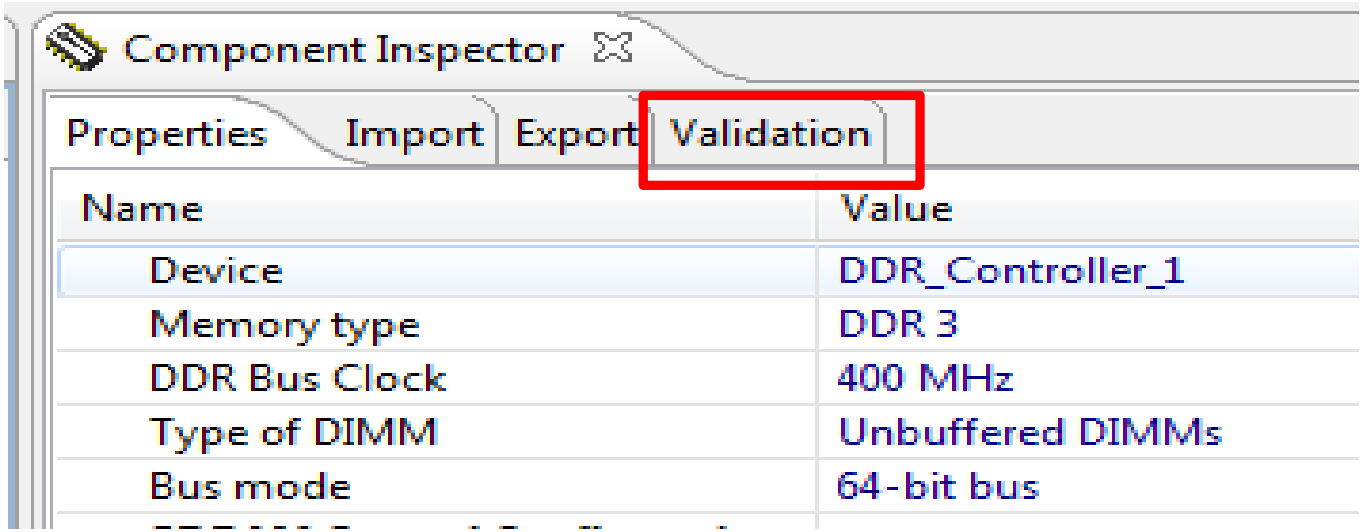
What is the PEx Optimization Suite?

- A new generation of products aimed at allowing customers to solve systems and application performance problems in the QorIQ and Layerscape family of devices
  - Users can analyze their applications unencumbered by the complexity of the debug IP
  - Provides a simple, clear and concise way of configuring the QorIQ debug IP to solve performance problems.
  - Continues the usage of the proven Scenarios Concept providing customers with ‘recipes’ to analyze common and complex performance problems
  - Transfer Freescale’s knowledge to customers – Scenarios
  - Supports bare metal and Linux applications. Special focus on Linux User space applications

DDR Validation Tool

… Extends QorIQ Configuration Suite

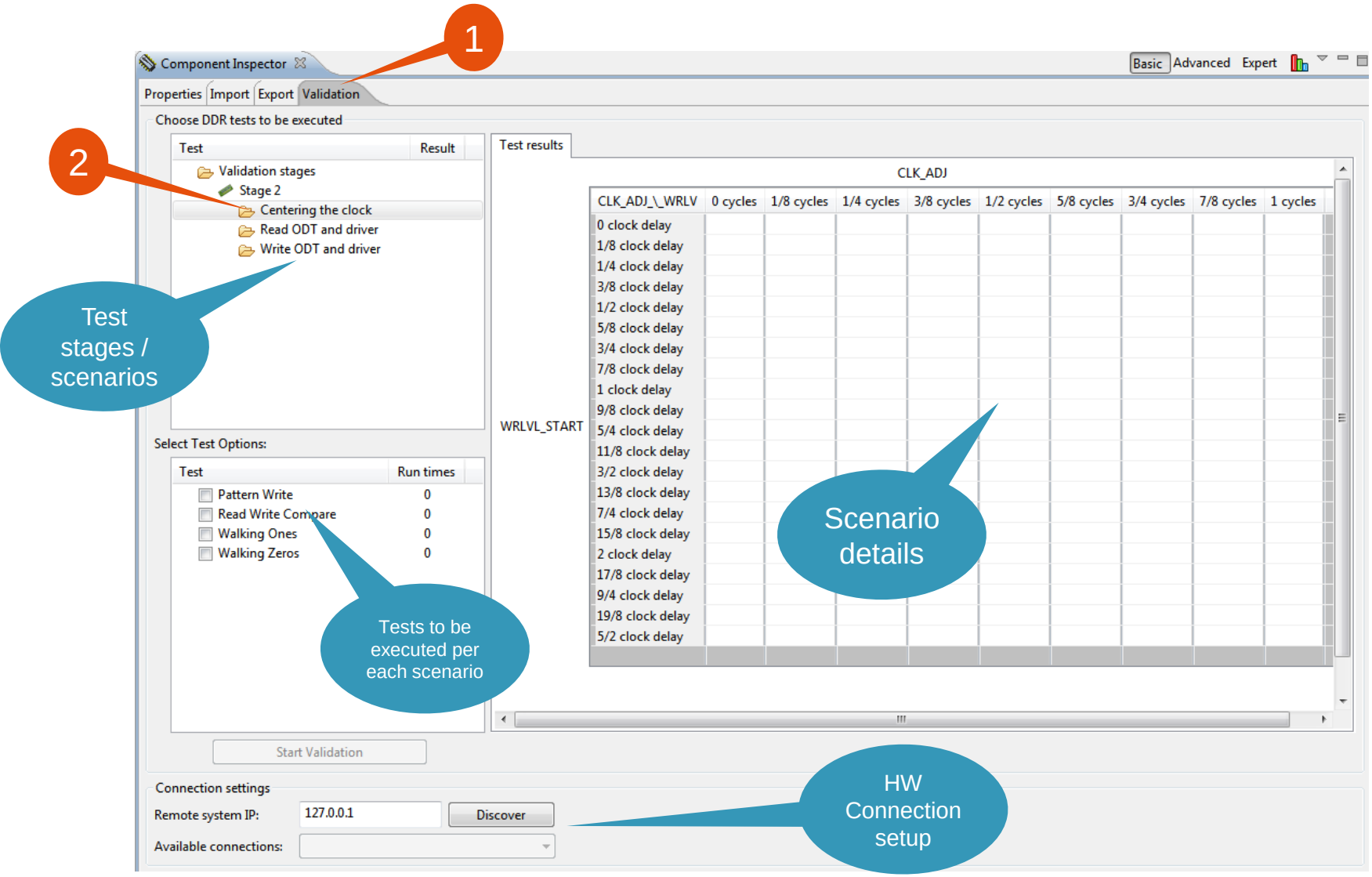
DDR Validation is a Licensed Product Leveraging QCS



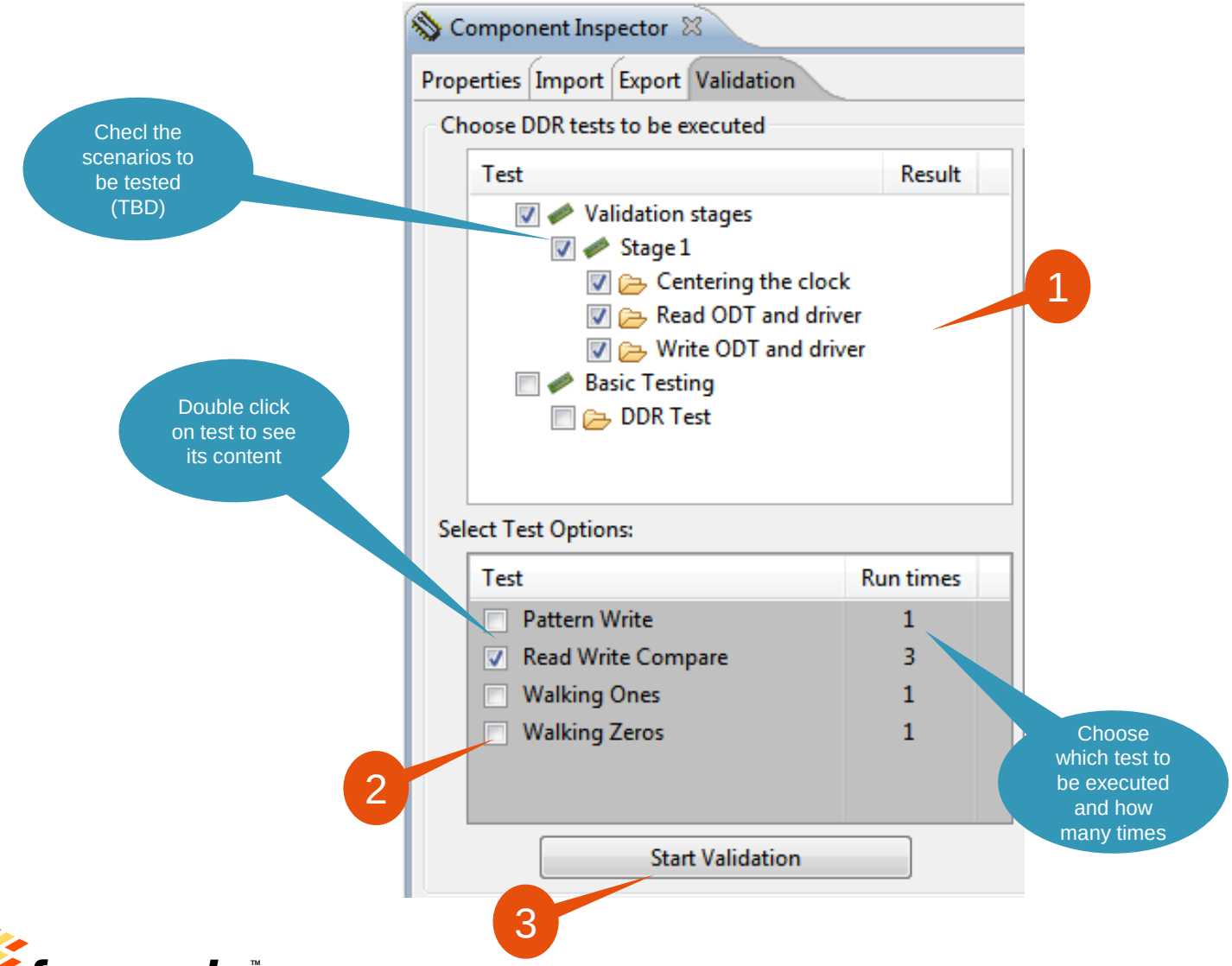
Pricing $995

License file:
<QCS Install directory>/eclipse/Optimization/license.dat

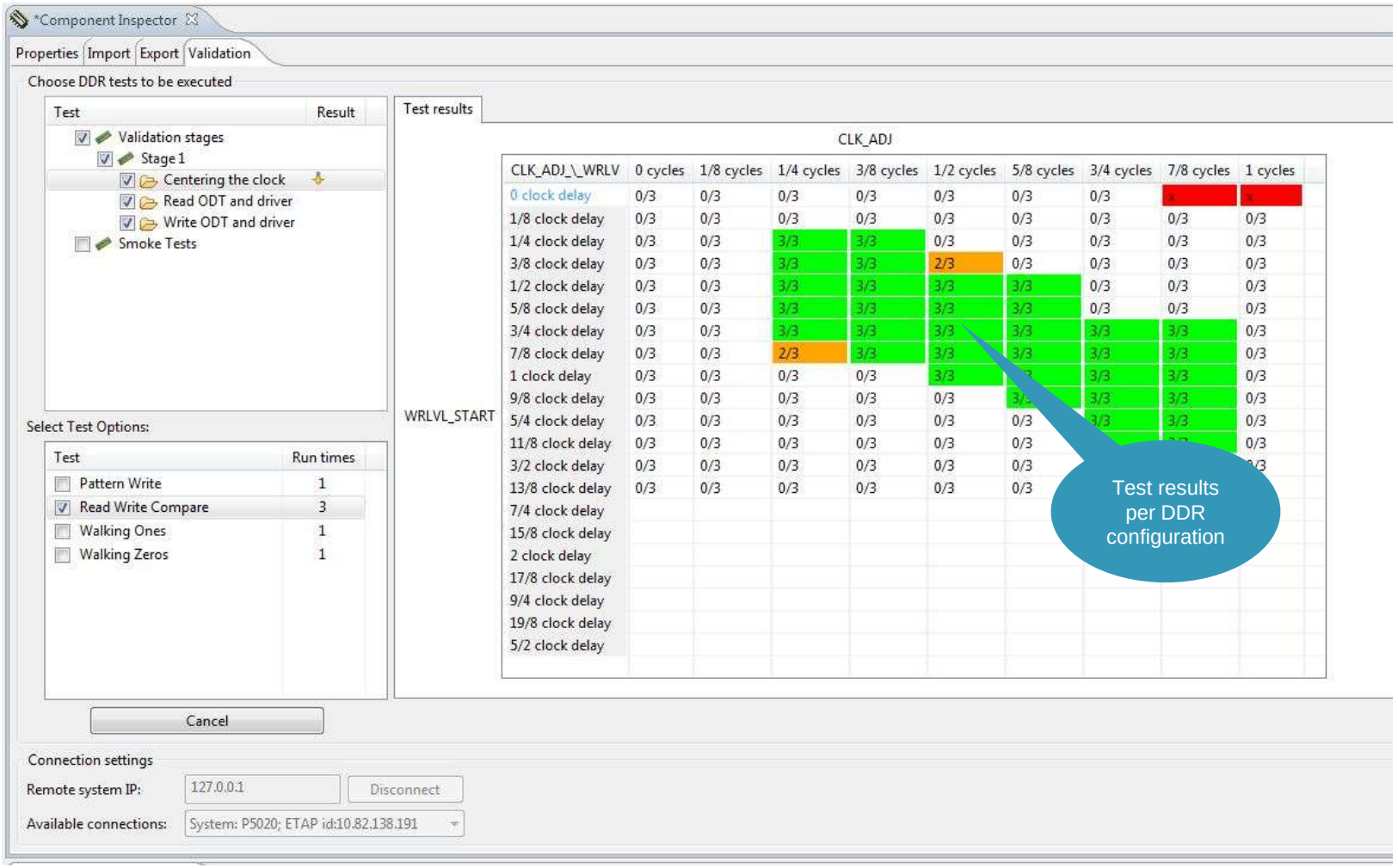
DDR Validation Panel



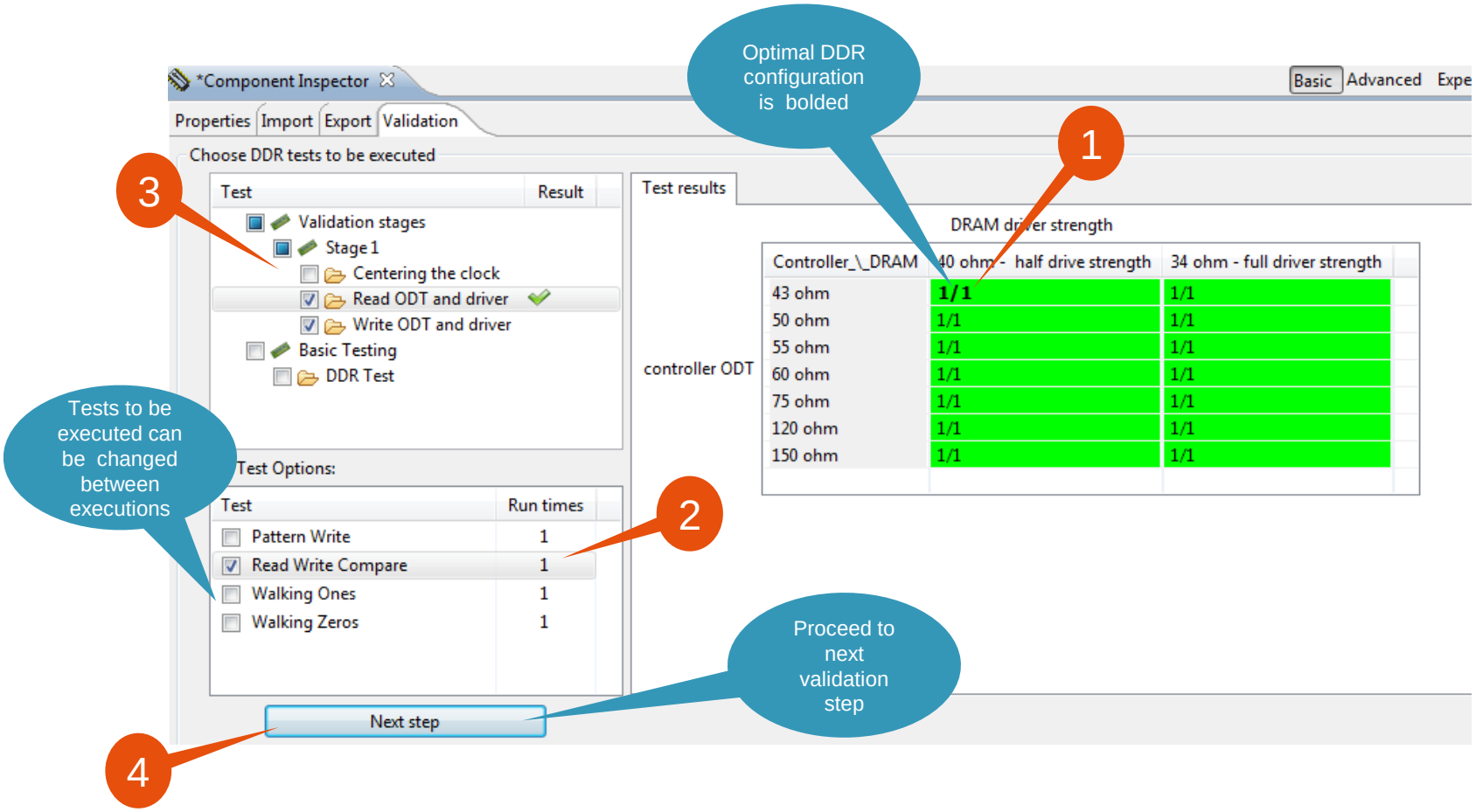
Configure DDR Tests To Be Run



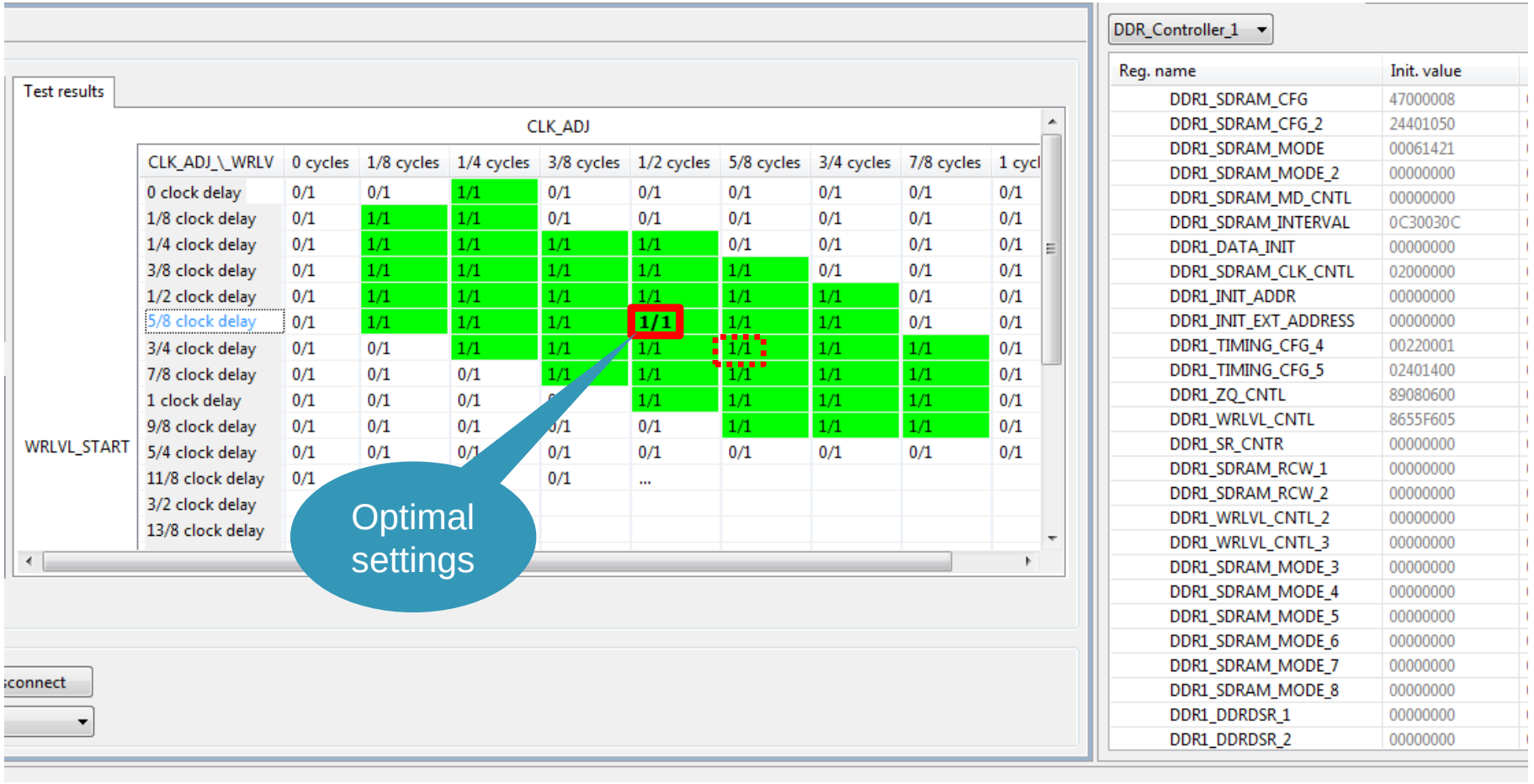
Observe DDR Validation Test Results



Optimal DDR Configuration (Read ODT)



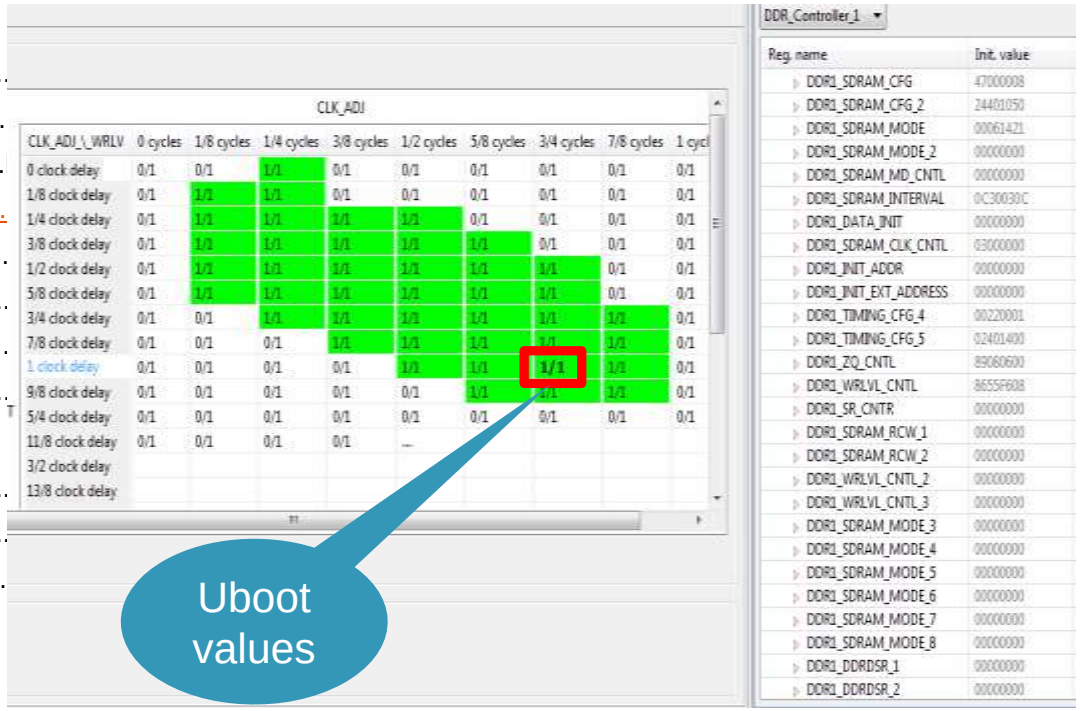
P2020RDB-PCA DDRV Optimized Test Results



P2020RDB-PCA: Compare optimal DDR configuration with uboot chosen values

Read DDR configuration from uboot

```
=> md ffe02000
ffe02000: 0000003f 00000000 00000000 00000000    ...?....
ffe02080: 80014202 00000000 00000000 00000000    ..B.....
ffe02100: 00030000 00110104 6f6b8846 0fa8c8cc    ........ok.
ffe02110: c7000008 24401040 00441421 00000000    ....$@..
ffe02120: 00000000 0c300100 deadbeef 00000000    .....0..
ffe02130: 03000000 00000000 00000000 00000000    ........
ffe02160: 00220001 02401400 00000000 00000000    ."...@..
ffe02170: 89080600 8675f608 00000000 00000000    .....u..
=> md ffe02b00
ffe02b00: 00000000 00000000 00000000 00000000    ........
ffe02b10: 00000000 00000000 00000000 00000000    ........
ffe02b20: 5dc07777 77000000 00000000 00000000    ].www...
```



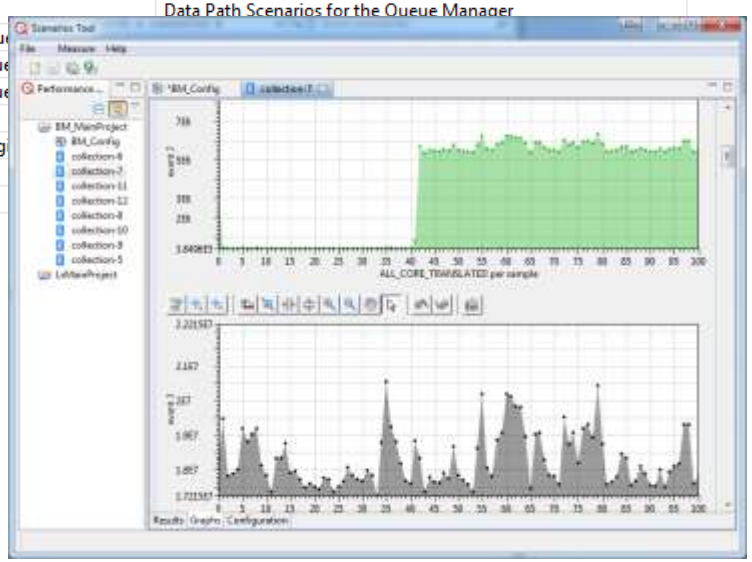
QorIQ Scenario Tool

. . . A Tool in QorIQ Optimization Suite

Optimization Suite – Scenarios Tool

- **Customer Benefit:**
  - System Optimization – Cores and SoC
  - Complexity Abstraction and ease of use
  - Streamlined to solve several performance issues
  - Deliver Freescale expertise to users
  - Probe-less, field-based usage
- **Target areas:**
  - Select QorIQ devices (P3 – T4/B4) and future Layerscape devices
  - Linux Systems (focus), but also supports bare metal
  - Performance Analysis including visualization

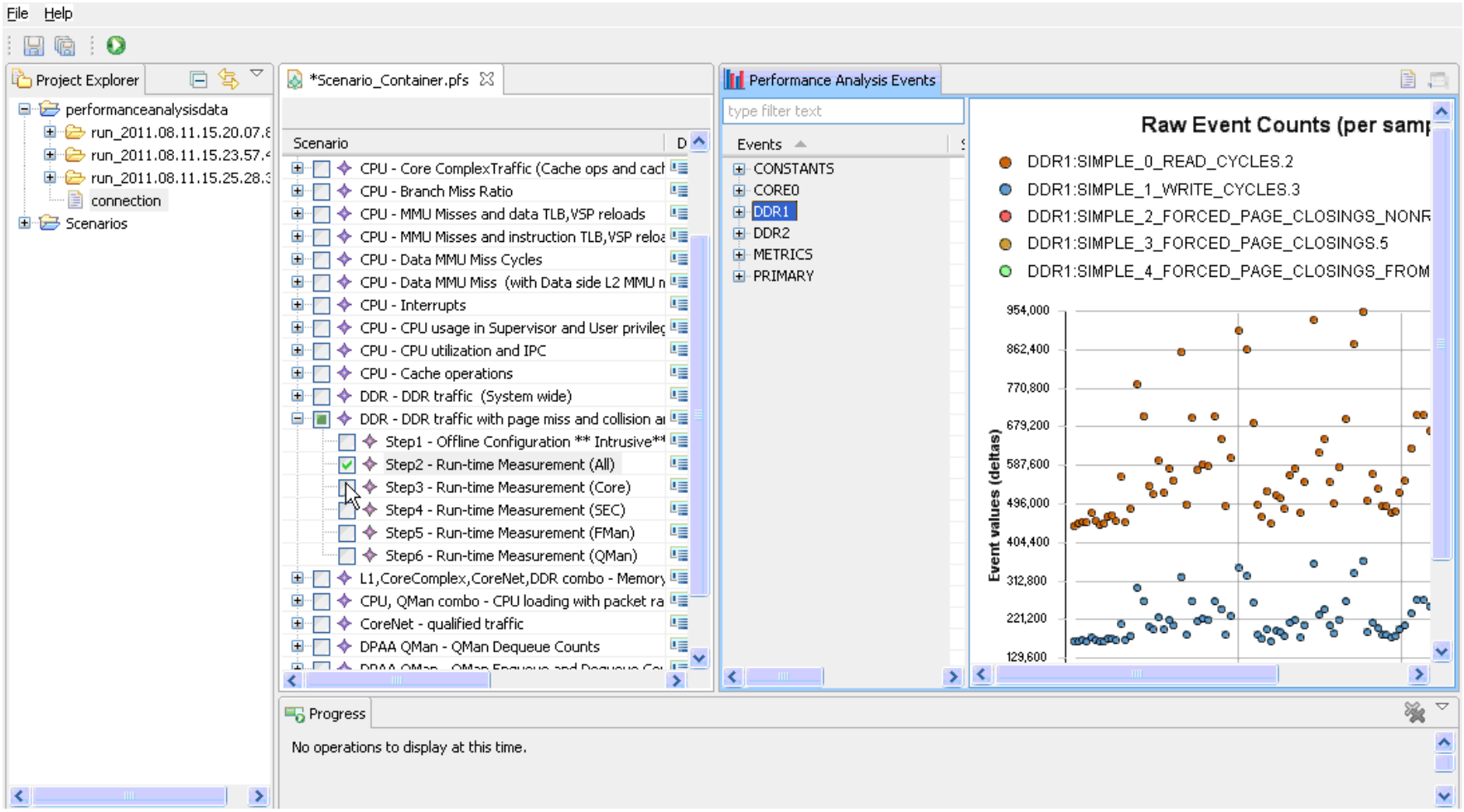
CPU	Core Related Scenarios
Utilization	
Cache	
MMU	
Core Complex	
Load Store Unit	
DDR	DDR Related Scenarios
DDR Traffic	Measures whole system
DDR Traffic by Whole System o	DDR Traffic Scenarios that measure whole system or individual ...
CoreNet	CoreNet Coherency Fabric Scenarios
CoreNet Traffic (All)	CoreNet Traffic (All)
CoreNet Traffic (PCIe)	CoreNet Traffic (PCIe)
CoreNet Traffic (BMan Portal)	CoreNet Traffic (BMan Portal)
CoreNet Traffic (QMan Portal)	CoreNet Traffic (QMan Portal)
CoreNet Traffic (CCSR)	CoreNet Traffic (CCSR)
CoreNet Traffic (DMA)	CoreNet Traffic (DMA)
DPAA QMan	Data Path Scenarios for the Queue Manager
QMan Deque	
QMan Enque	
QMan Enque	
DPAA SEC	
Security Engi	
OCeaN	
DMA	



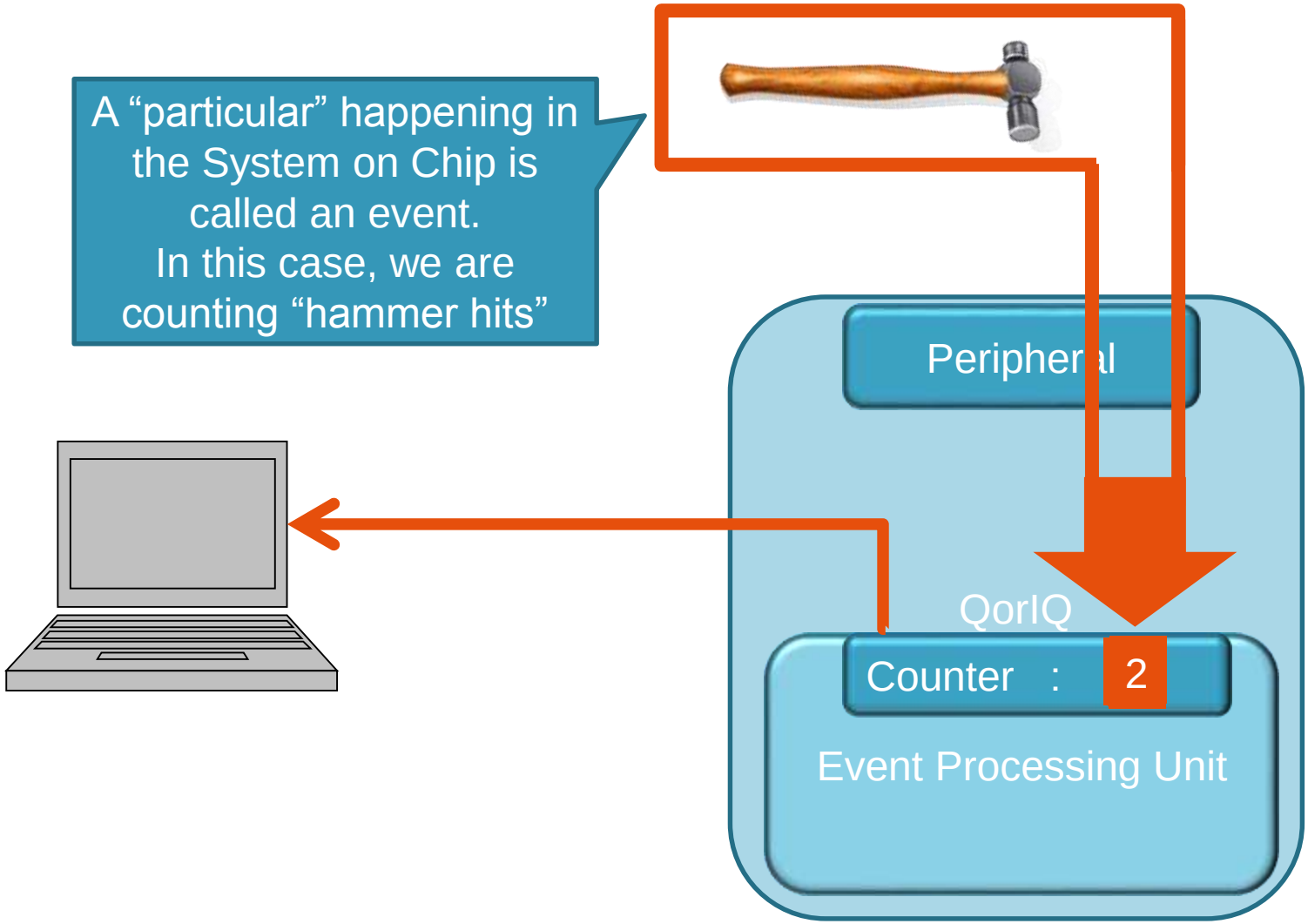
Optimized workflow for efficiently narrowing down performance issues anywhere on the system

What Is Scenarios Tool?

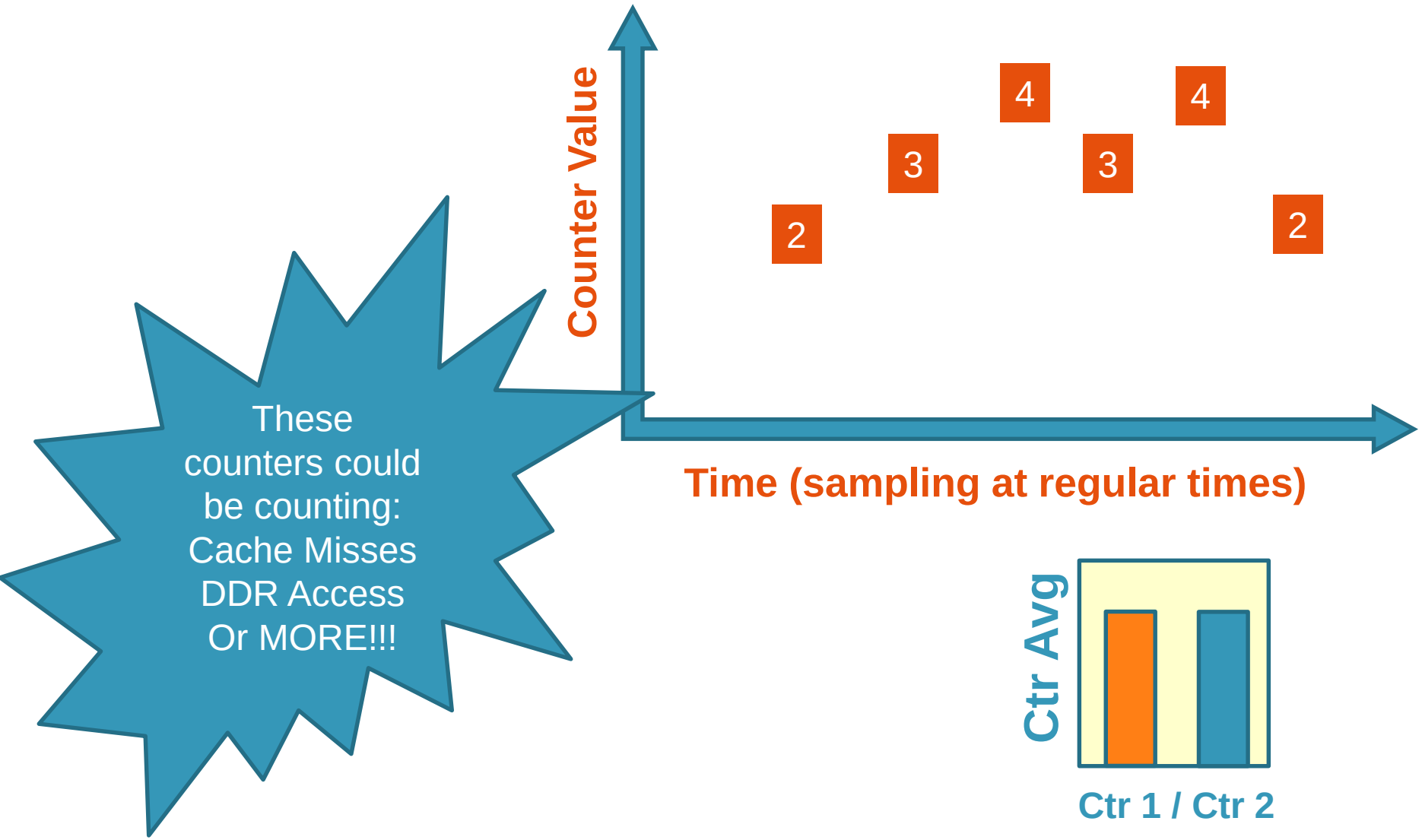
- **Visually identify the system level problem areas in seconds.**



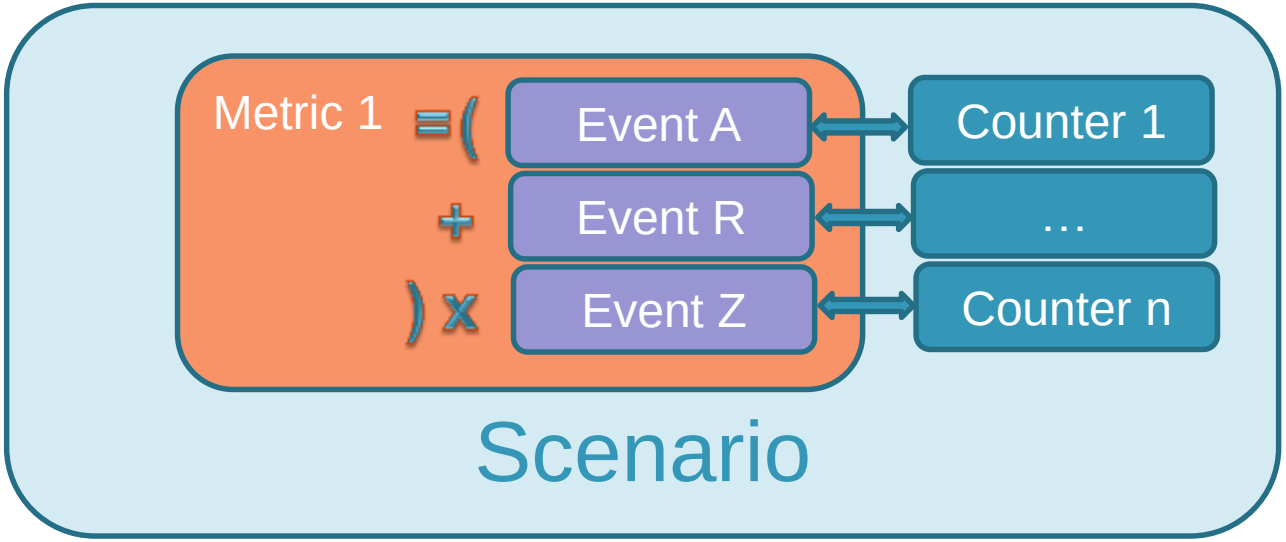
How "Counters" Work



Making Sense of Counter Data



What Is a Scenario?



A Scenario is a CONTAINER of:

- Required counters
- Key events for a measurement
- Counter<->event connections
- Metrics
  - a metric is a math equation of captured event-counts

A GOOD scenario would count and combine (using metrics) all of the information required to measure something useful, such as "cache misses" or "buffer overflows"

Here Scenarios Tool Measures Cache Accesses

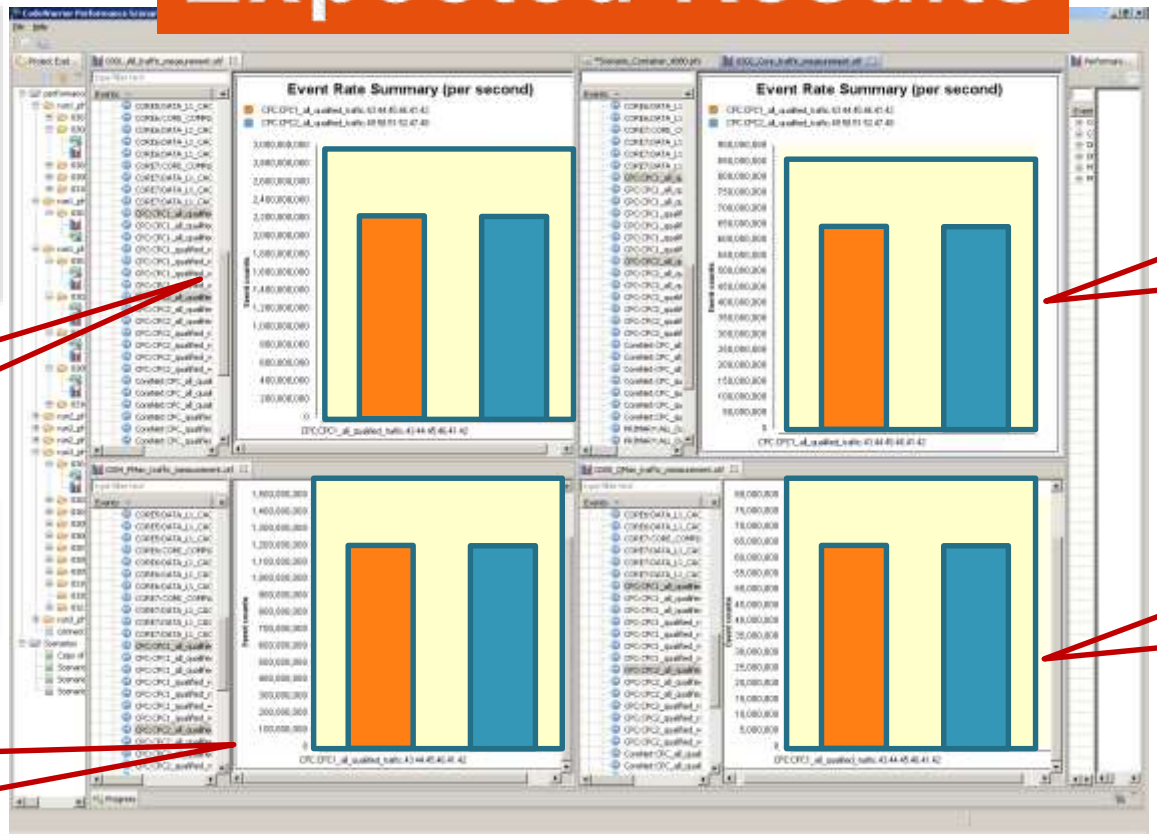
Expected Results

Selected scenario captures overall traffic as well as individual subsystem traffic including traffic from cores, Fman and Qman

Total traffic to CPC1 and CPC2

Unbalance between platform caches

Fman CPC2 Access



CPU Accesses

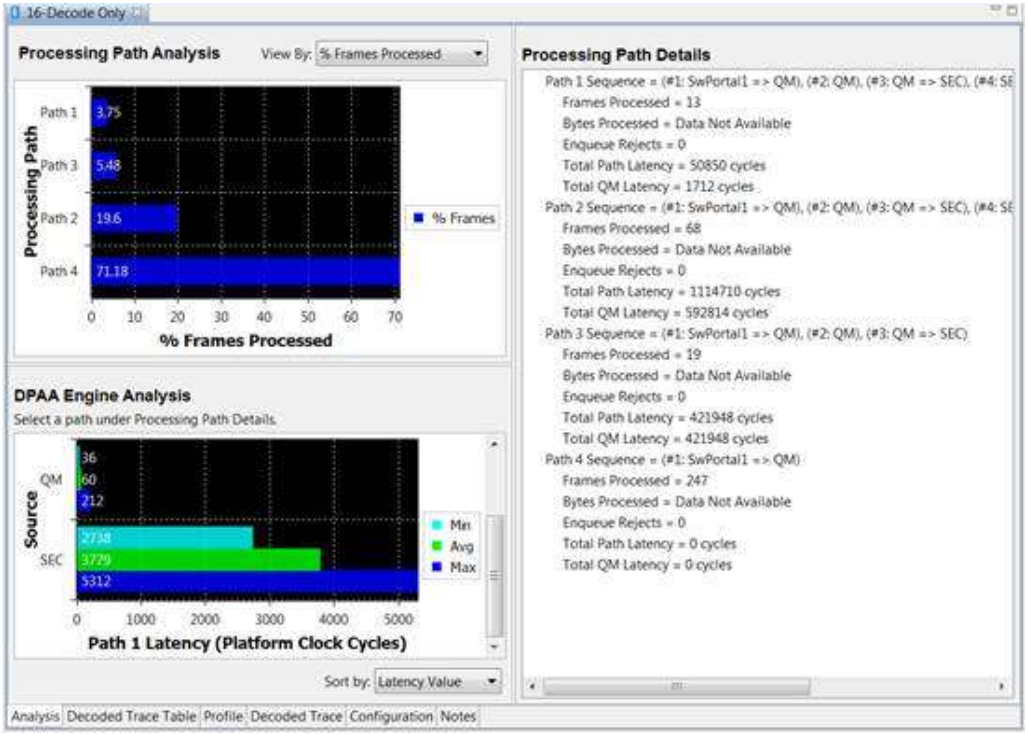
Qman CPC2 Access

- Imbalance can be seen by analyzing these subsystems:
  - Majority of CPC1 accesses are made by the CPU (top right)
  - Majority of CPC2 access are from Fman and Qman
  - Remainder of the traffic is due to PCI
- LACK OF BALANCED ACCESSES suggests DDR interleaving is not configured properly
- Fixing the configuration provides a performance boost

QorIQ Packet Analysis Tool

Packet Analysis Tool

- **Customer Benefits**
  - Complexity abstraction and ease of use
  - Enables key use cases:
    - Packet-Oriented System Level Performance Analysis
    - SoC Data Plane Configuration Debug
    - Packet Processing Latency Analysis
    - Packet Processing Critical Resource Monitoring
- **Target areas:**
  - SoC debug/analysis feature enablement
  - Linux Systems
  - Analysis data interpretation and visualization
- **Users**
  - External customers
  - Freescale internal developers



Main Use Cases

Packet Tracing	Shows which parts of the system processes the frames. For example, use this to verify that the frame flow is what you expect.
Lost Packet Analysis	Understand why the frames become “lost” in the system. For example, use this to check how the FM PCD changes affect where frames are sent.
Latency Analysis	Precisely measure the time spent processing frames at various points in the system.
Packet Sequence Analysis	See how an entire sequence of frames was processed. For example, use this to measure the performance of the SEC.
QM Performance Analysis	Use QM profile data to measure the performance of the system, at “data-flow” level.

QorIQ DataPath Trace

- Visibility into FM and QM activities via Nexus Trace
- **Trace data**
  - Can be collected from a running system
    - Without interrupting it and
    - Without affecting its performance
  - Can be collected to on-board trace buffer
  - Is timestamped so it can be used to precisely measure the timings
- **FM trace**
  - Optionally output by each FM engine: BMI, KeyGen, Parser, etc.
  - Timestamped internally by the FM clock
  - The trace data contains: FD, FM port number, NIA, etc.
- **QM trace**
  - Optionally output by each QM enqueue and dequeue point
  - The trace data contains: FQID, channel, frame address, frame length, enqueue/dequeue flag, portal type and number, etc.
- **Traced frames**
  - Only the tagged frames are traced. “Tagged” = FD[DD] bits set
  - Rx flow – the frames tagged by the FM, as configured by the Packet Analysis Tool
  - Other flows – the frames tagged by the instrumented software running on the cores

What Tools Are New

Why SerDes Configuration and Validation?

- Increased complexity of SerDes module configuration due to:
  - Growth of SerDes IP revisions
  - Growth of SerDes Lane Assignment and Multiplexing
- Configuration problems during board bring-up encountered by customers
- Useful built-in testing capabilities not exposed to customers
- More and more signal integrity problems coming from customers that are harder to debug

Solution to Solve these Problems

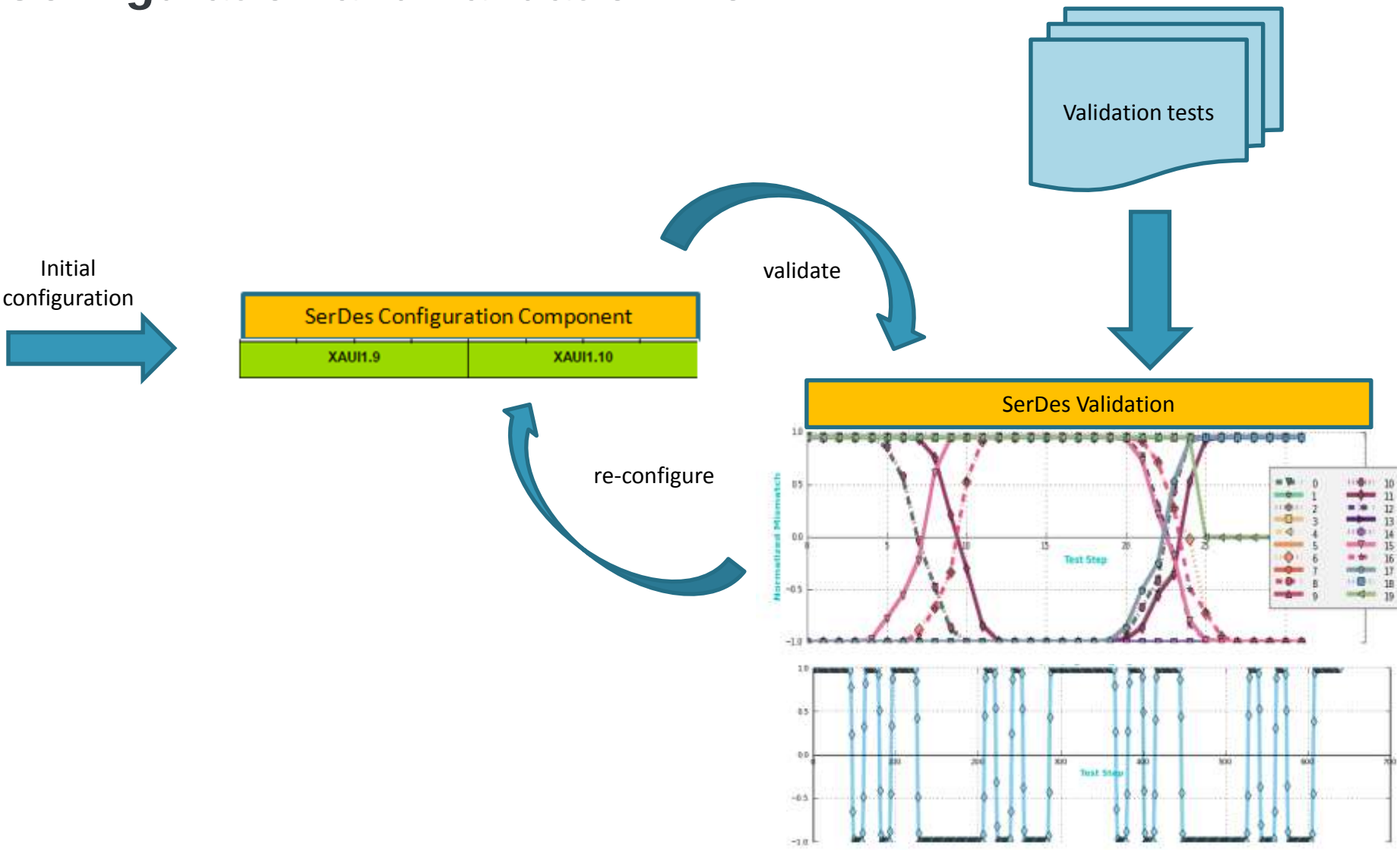
- A configuration and validation tool to help:
  - Board bring up and configuration
  - Debug and verify different configuration
  - Silicon validation
- User-friendly interface for SerDes configuration and validation capabilities
- Software to monitor performance of the lanes and report eye quality by rolling out jitter scope features to FAE and customers

Objectives

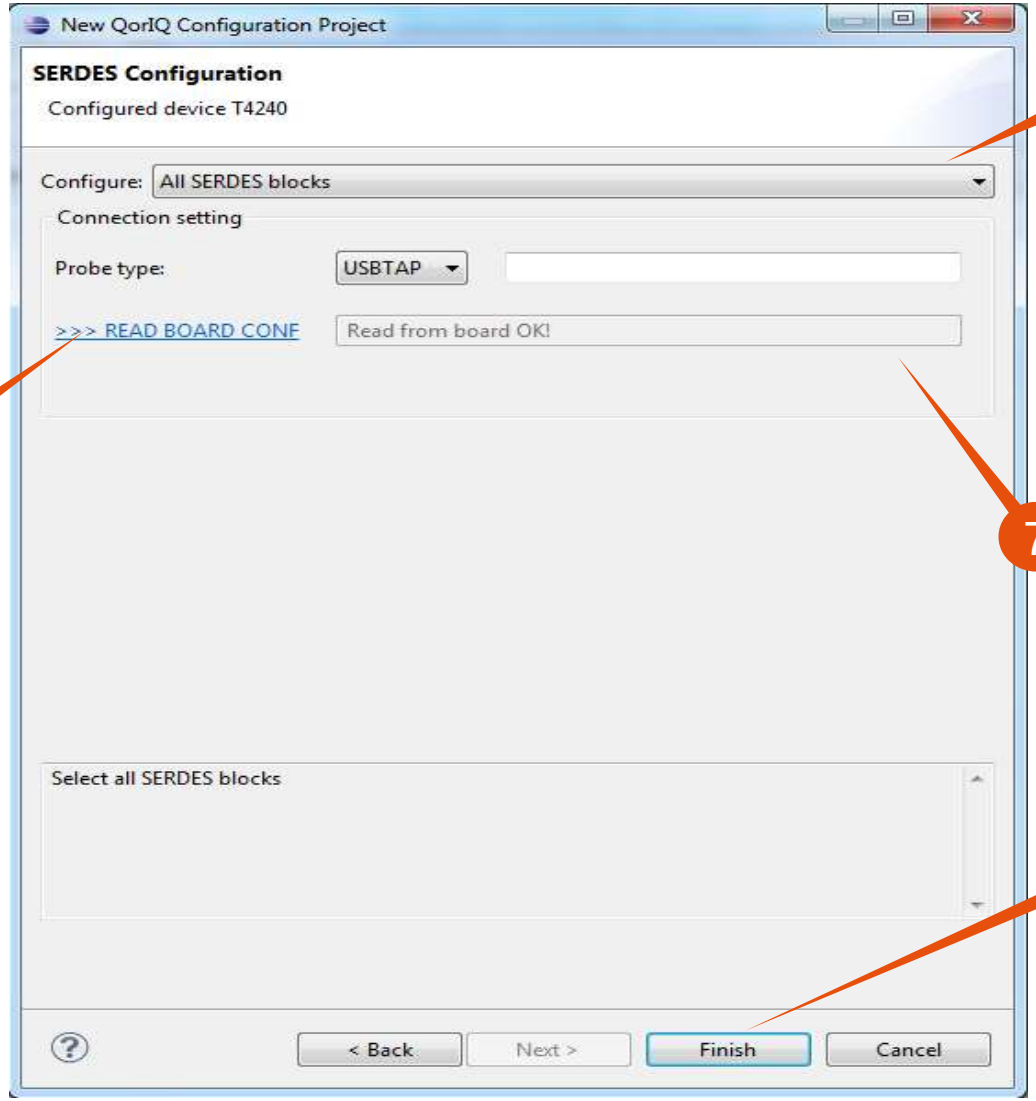
- SerDes configuration
  - Program SerDes blocks: PLLs, Lanes
  - Cross validation with RCW configuration for SerDes Lanes Mux in PBL

- SerDes validation
  - Execute different testing capabilities
  - Capture jitter scope data samples per each SerDes lane via the SerDes control registers
  - Display as graphs the **eye diagram** and **recovered data stream**

Configuration and Validation Flow



SerDes Configuration Wizard Page



5

Select SerDes options:
- SerDes block
- Probe type (usb, gtap, etap)

6

Press read from board button

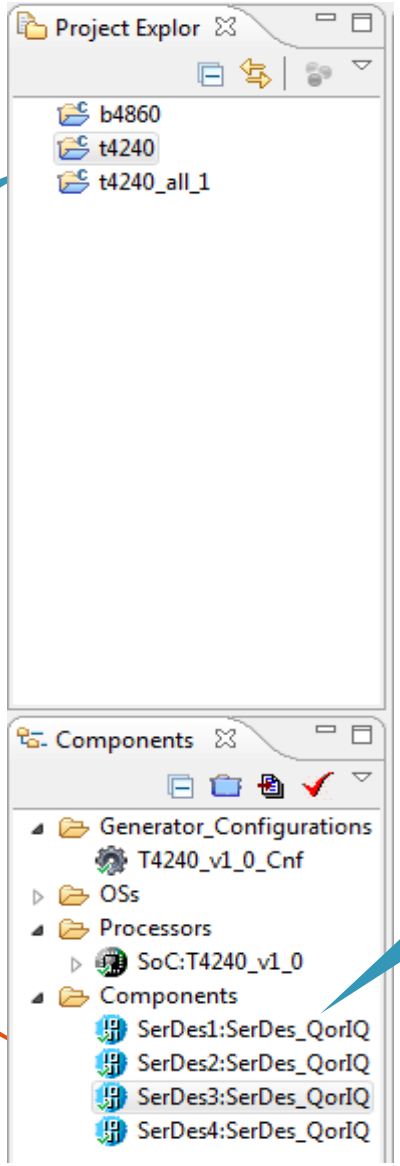
7

Status of reading action

8

Press finish to create SerDes component

Select SerDes Configuration and Validation Panel



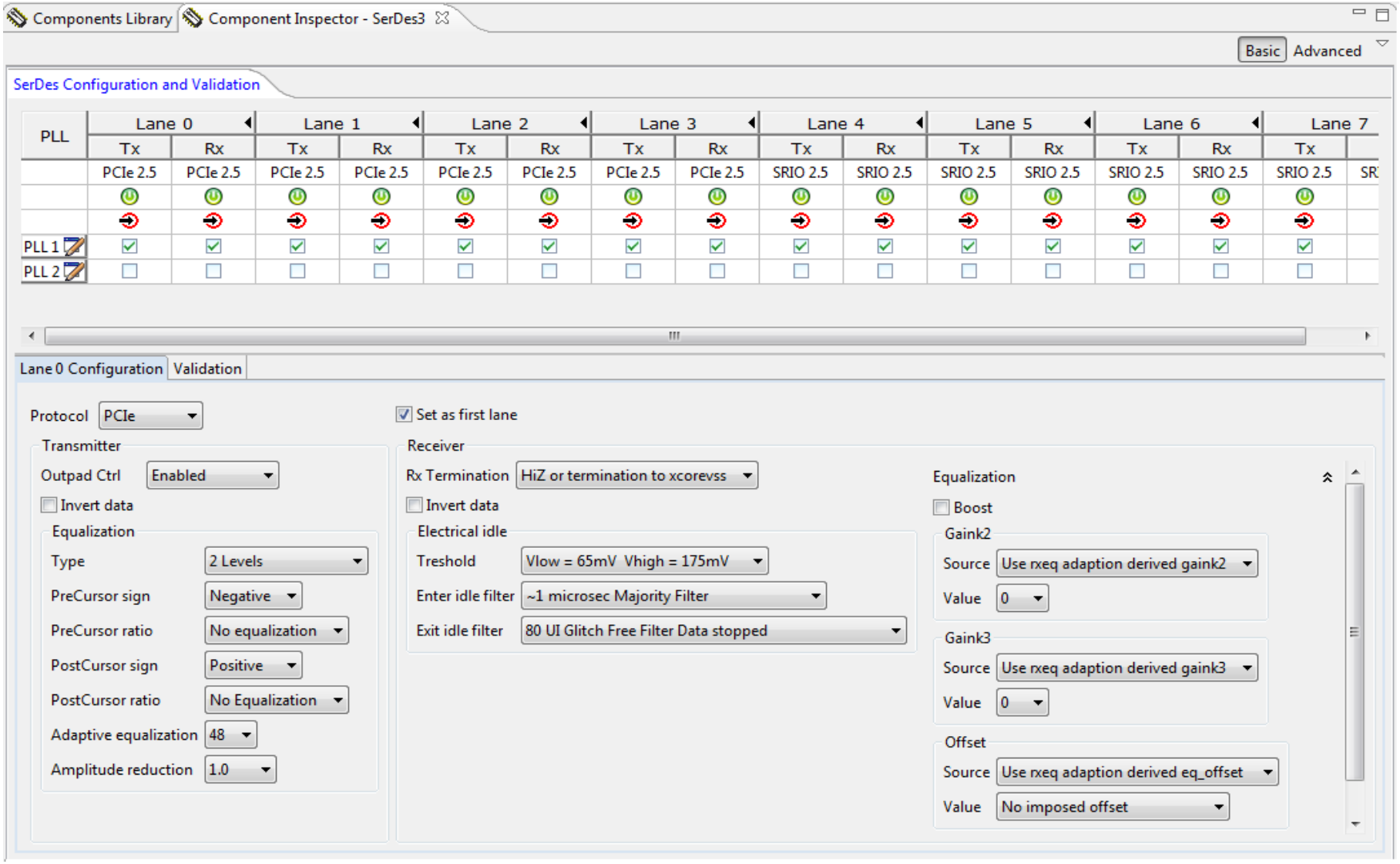
Project was created

Components for each SerDes block were created

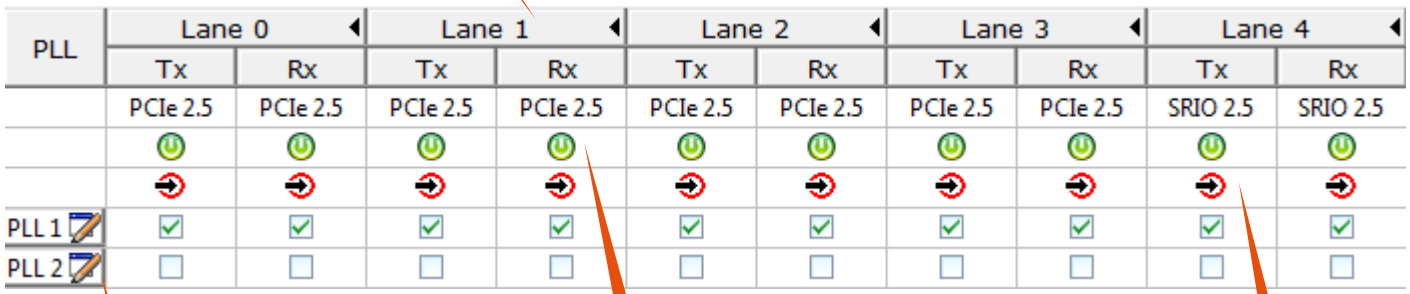
Select SerDes module to view

1

SerDes Configuration and Validation Panel



SerDes Configuration and Validation Panel (continued)



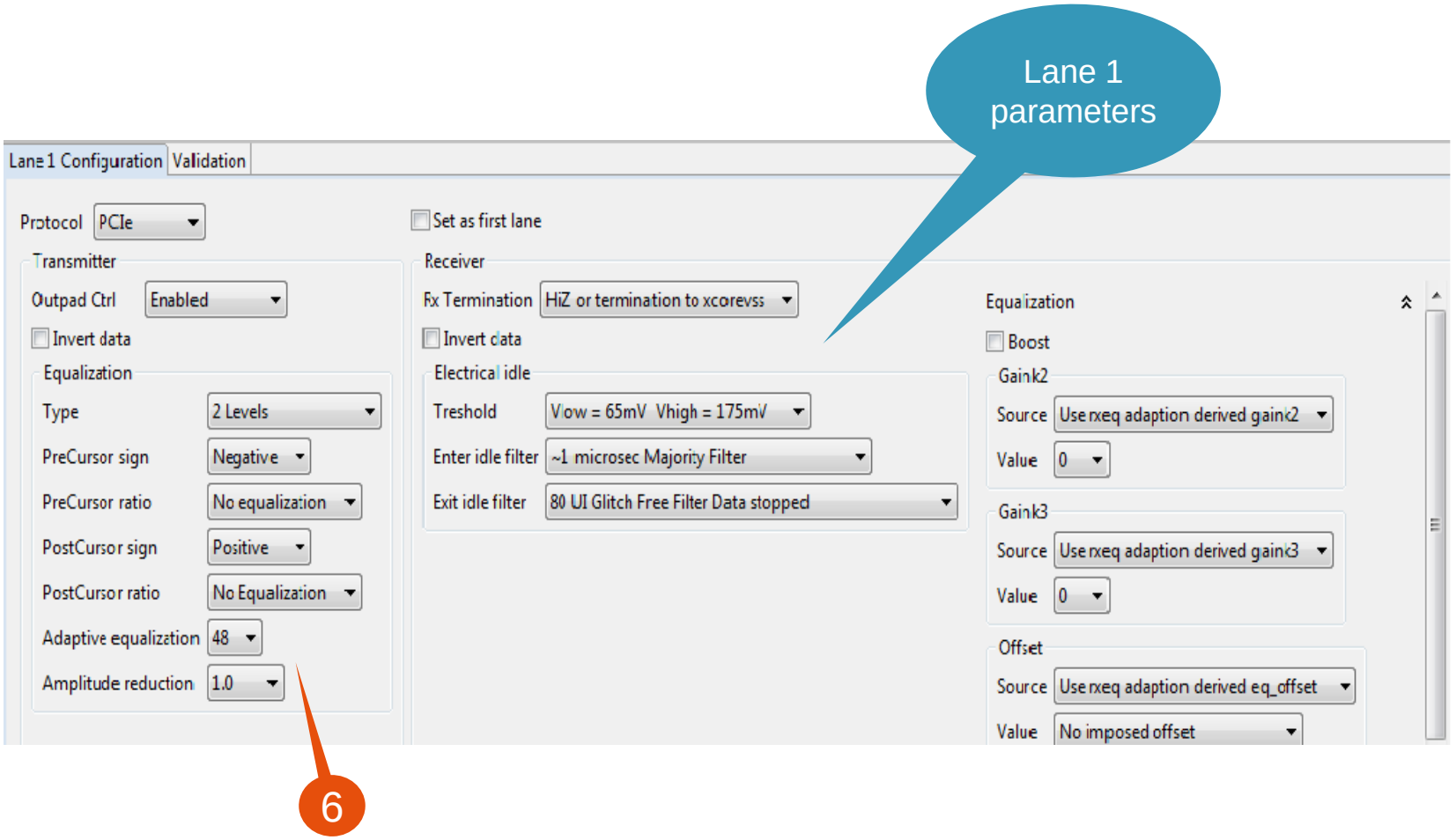
2 Select Lane 1 configuration and validation view

3 Select PLL2 configuration view

4 Power down/up rx for lane 1

5 Reset tx for lane 4

SerDes Configuration and Validation Panel (continued)

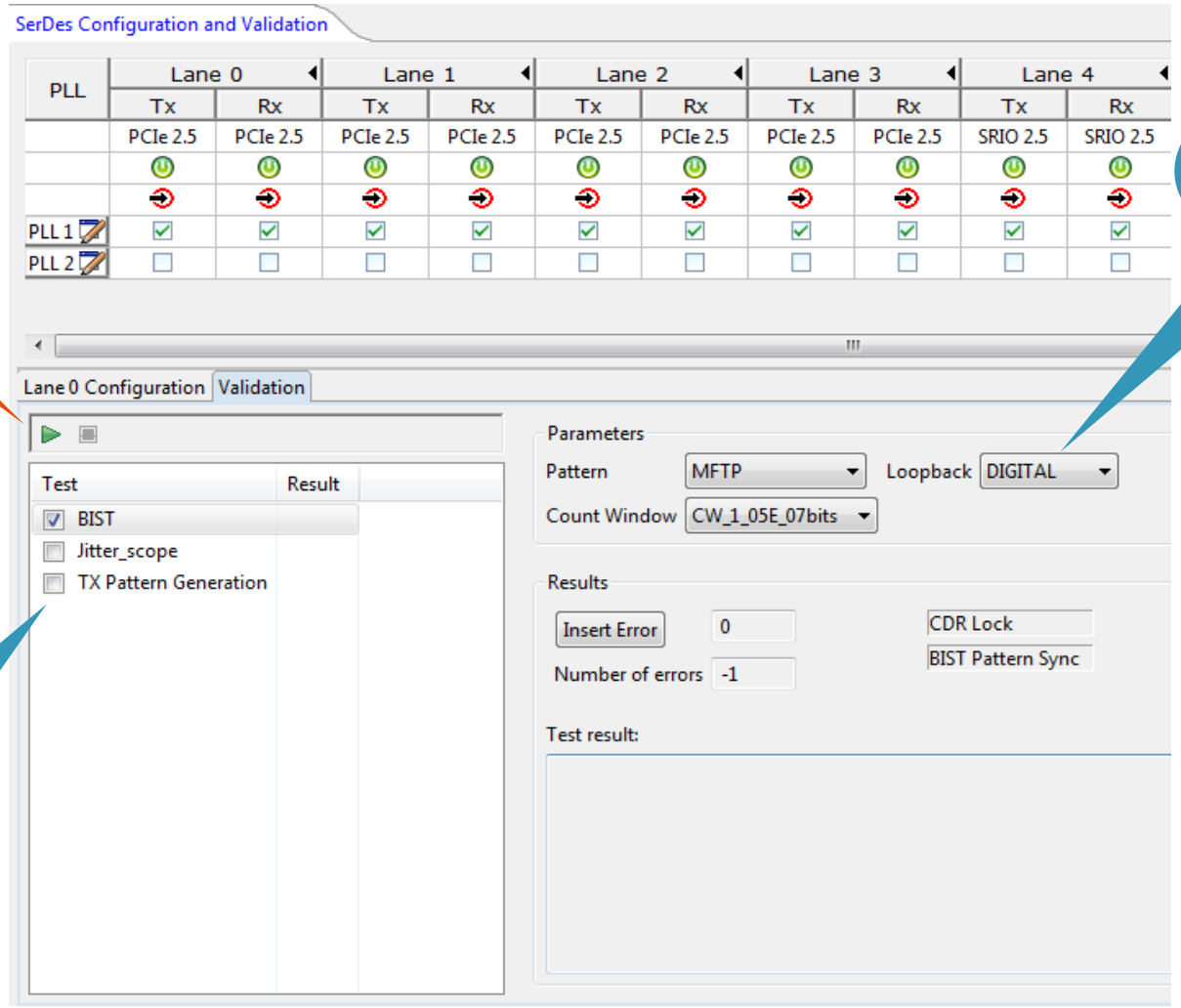


Select desired lane parameters

SerDes Configuration and Validation Panel (continued)

Apply configuration and start validation

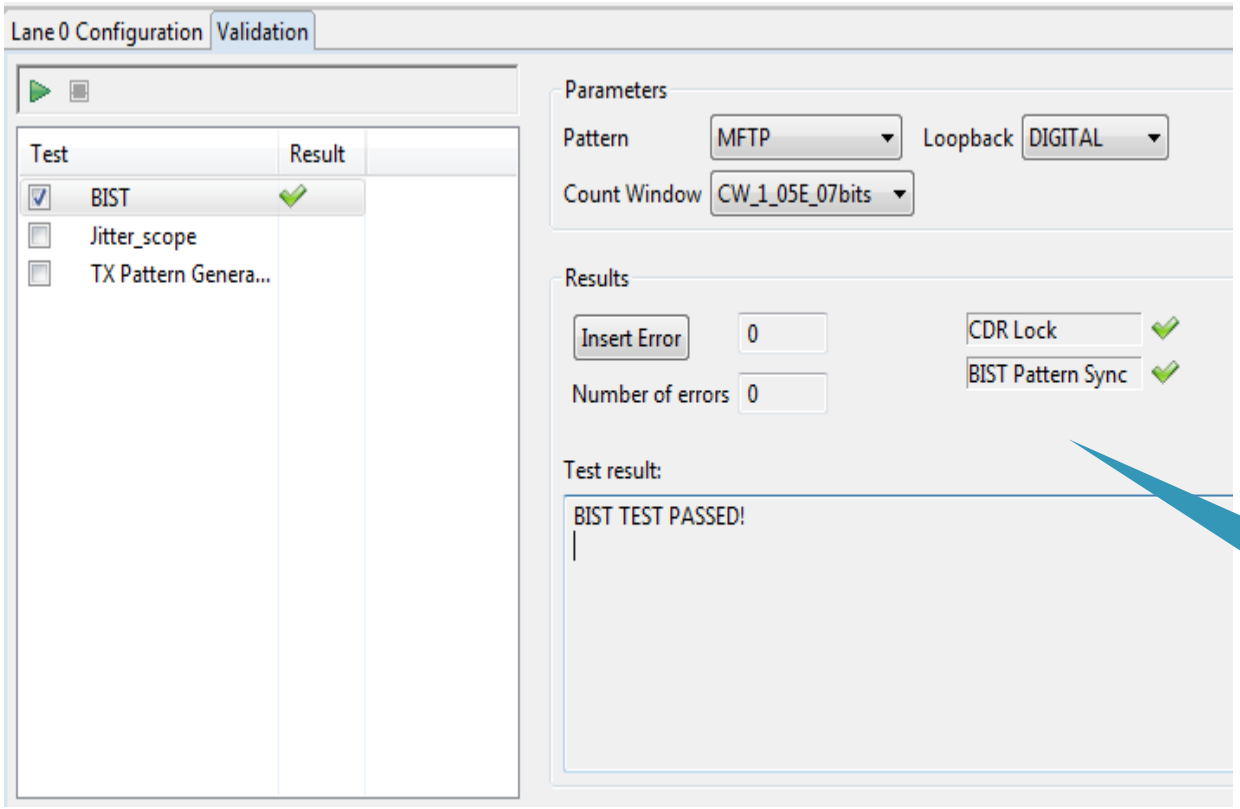
7



BIST testing parameters

List of tests

SerDes Configuration and Validation Panel (continued)



Use **Dynamic Analysis Tools** on Linux®

Intro to Perf (1)

- Perf is a performance analysis tool that is based on the perf_events interface made available in Linux Kernels Version 2.6 and higher
- Perf is a user space utility that is part of the kernel repository. Typically you'd obtain Perf with your Linux kernel
- The interface between a Perf utility and the kernel consists of one syscall and is done via a file descriptor and a mmapped memory region (maps file into memory)

Intro to Perf (2)

- Perf

  `https://perf.wiki.kernel.org/index.php/Main_Page`
- The Perf command on a command line interface:

  `usage: perf [--version] [--help] COMMAND [ARGS]`
- Perf is used with several commands:

  `'stat': obtain event counts.`

  `'top': see live event count.`

  `'record': record events for later reporting.`

  `'report': break down events by process, function, etc.`

  `'annotate': annotate assembly or source code with event counts.`

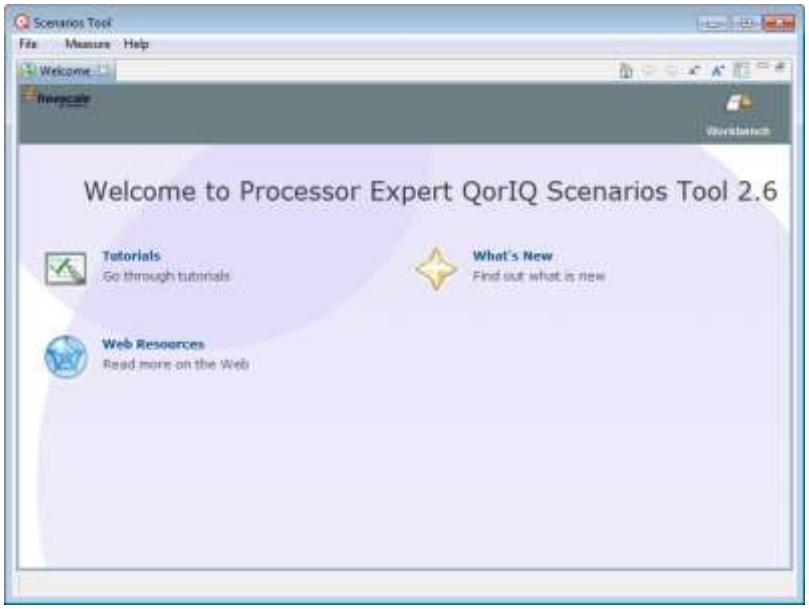
  `'sched': tracing/measuring of scheduler actions and latencies.`

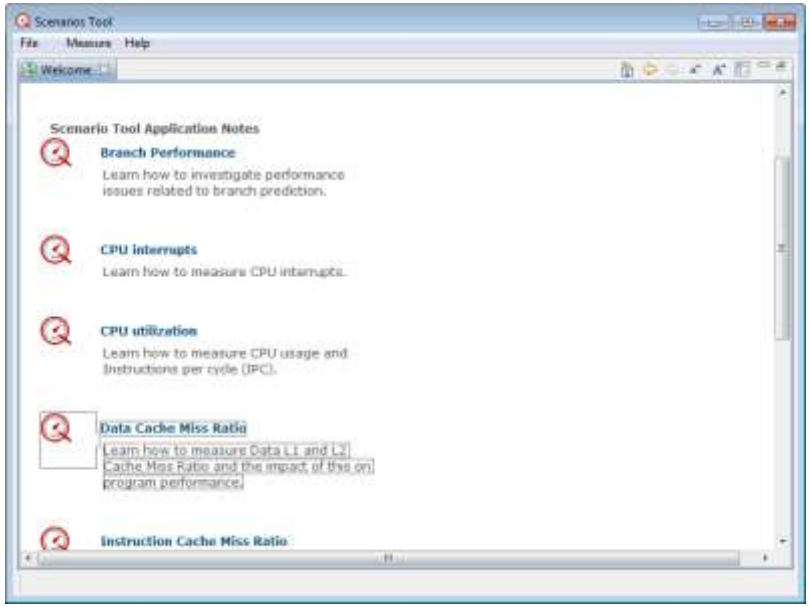
  `'list': list available events.`

Intro to Perf (3)

- Freescale's QorIQ Performance Analysis tools provide a user interface that hides much of the complexity. It provides App Notes and User Manuals too. Search for "PE_QORIQ_SCENT" on www.freescale.com:

http://www.freescale.com/webapp/sps/site/prod_summary.jsp?code=PE_QORIQ_SCENT





Intro to Valgrind Memcheck (1)

- Valgrind is an instrumentation framework for building dynamic analysis tools
- Valgrind is a collection of tools for dynamic analysis including these and more:
  - Memcheck detects memory management problems
  - Cachegrind – a cache profiler
  - Massif – a heap profiler
  - Helgrind – thread debugger which finds data races in multithreaded programs
- This session will focus on Valgrind Memcheck

Intro to Valgrind Memcheck (2)

- The Valgrind project is located at:
  - http://valgrind.org/
  - You can download a version from there or from the SDK for your silicon product
- The Valgrind Memcheck command on a command-line interface. Memcheck is the default tool:

```
usage: valgrind [--version] [--help] [--tool=memcheck] foo
  [foo's args]
```

- Memcheck can detect:
  - Use of uninitialised memory
  - Reading/writing memory after it has been freed
  - Reading/writing off the end of malloc'ed blocks
  - Reading/writing inappropriate areas on the stack
  - Memory leaks -- where pointers to malloc'ed blocks are lost forever
  - Mismatched use of malloc/new/new [] vs. free/delete/delete []
  - Overlapping src and dst pointers in memcpy() and related functions
  - Some misuses of the POSIX pthreads API

Benefits of Using Open Source Tools

- Open source tools are tools where the source code is published and available to view, use, modify, and redistribute. The tool is maintained by a collaborative community
- These are some of the benefits of using open source tools:
  - Free of charge
  - Source code is available to view, use, modify, and redistribute
  - Technical development by involvement with a community of experts
- These are some of the costs of using open source tools:
  - When something goes wrong, you can’t call the vendor for support
  - You have to consider licensing terms when distributing software
  - You should contribute changes to the community

www.Freescale.com