



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



External Use

Freescale, the Freescale logo, AllWin, C-S, CodeTEST, CodeWarrior, ColdFire, ColdFire+, C-Ware, the Energy Efficient Solutions logo, Kinetic, MagniV, mobileGT, PEG, PowerQUICC, Prosecc Expert, QorIQ, QorIQ Qonvergence, Qorivos, Ready Files, SafeAssure, the SafeAssure logo, StarCore, Synphony, Vortige, Vybrid and Xilinx are trademarks of Freescale Semiconductor, Inc., Reg. U.S. Pat. & Tm. Off. AirMat, BeeKit, BeeStack, CoreNet, Flexis, LayerStack, M3C, Platform in a Package, QUICC Engine, SMARTMO2, Tower, TurboLink and UMEMS are trademarks of Freescale Semiconductor, Inc. All other product or service names are the property of their respective owners. © 2015 Freescale Semiconductor, Inc.



Agenda

Overview

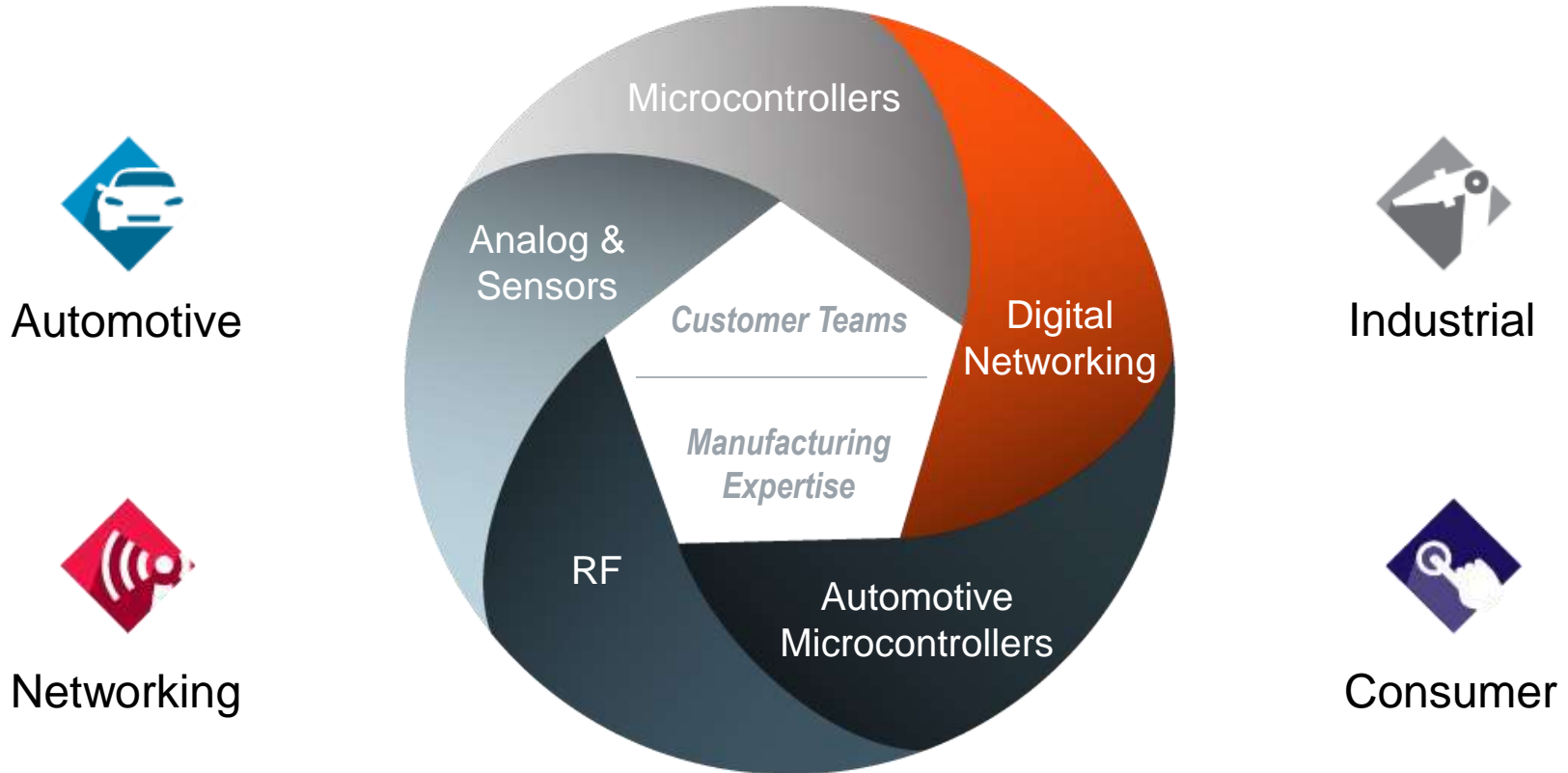
- Complexity Increasing
- Freescale Technology and Ecosystem

Freescale Commercial Capabilities

- Software
- Hardware
- Systems

Questions and Answers

We Are a Global Leader in Embedded Processing Solutions



Five Core Product Groups

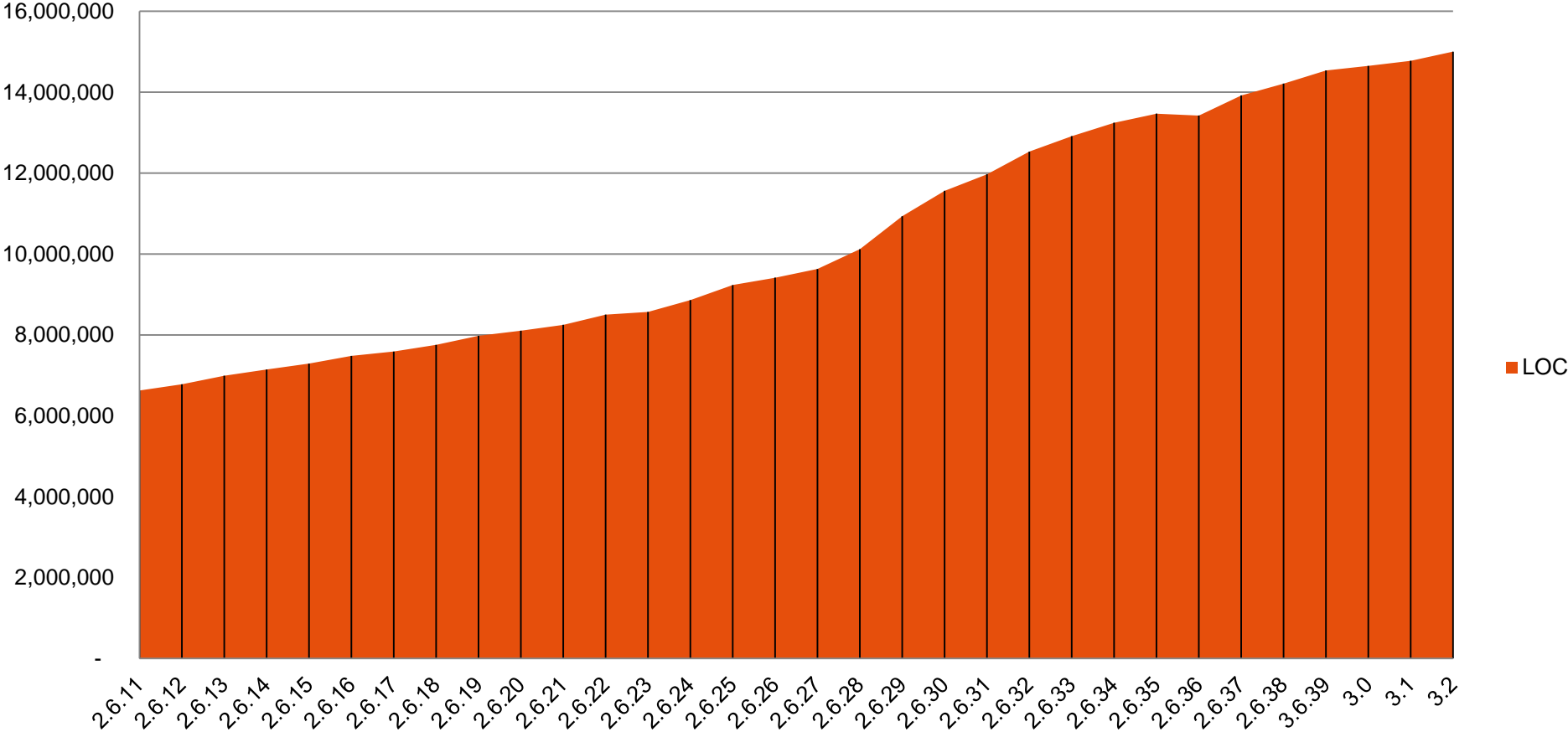
>50 Year Legacy
>6,400 Patent Families*

Four Primary Markets

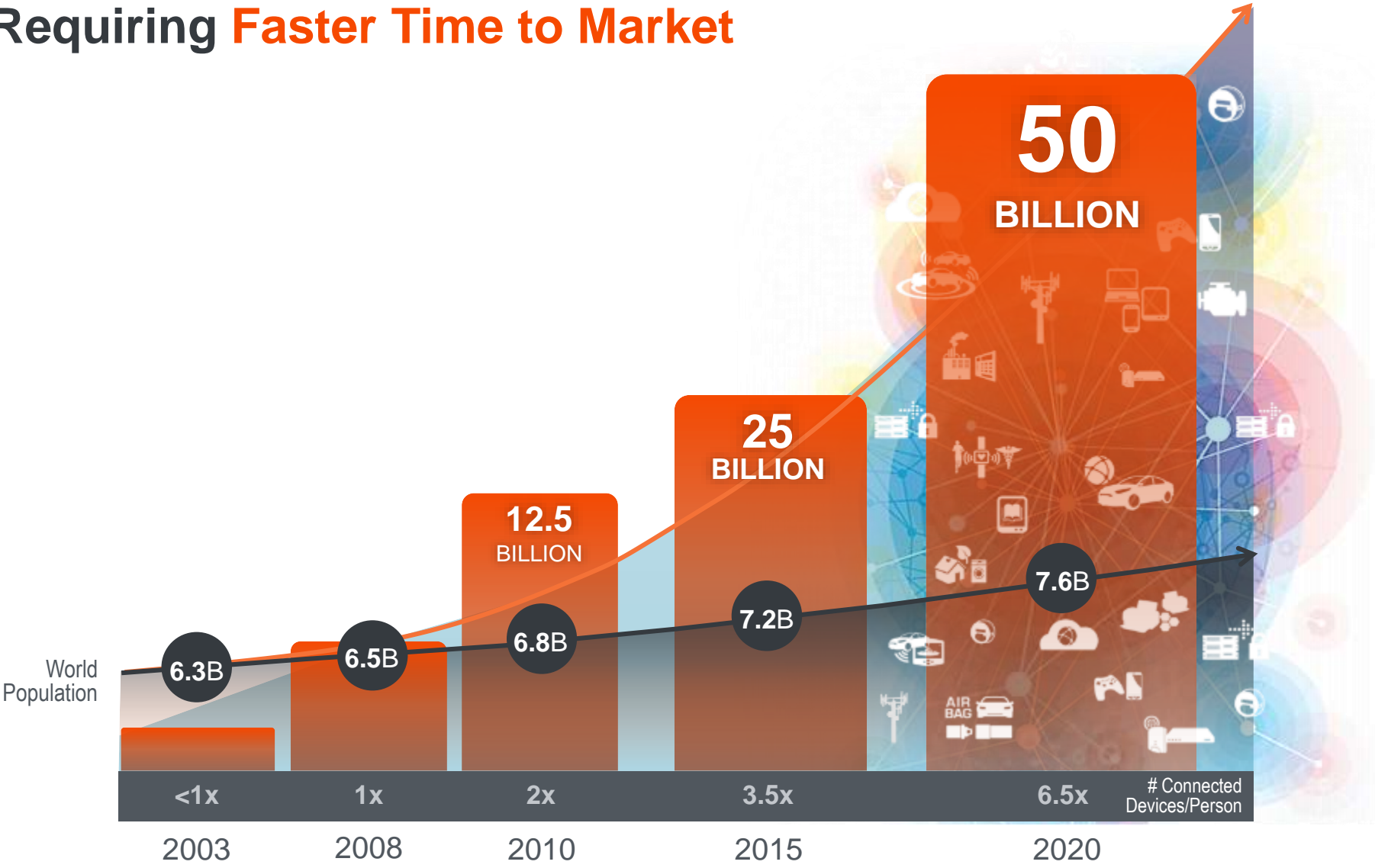


Software Complexity Increasing

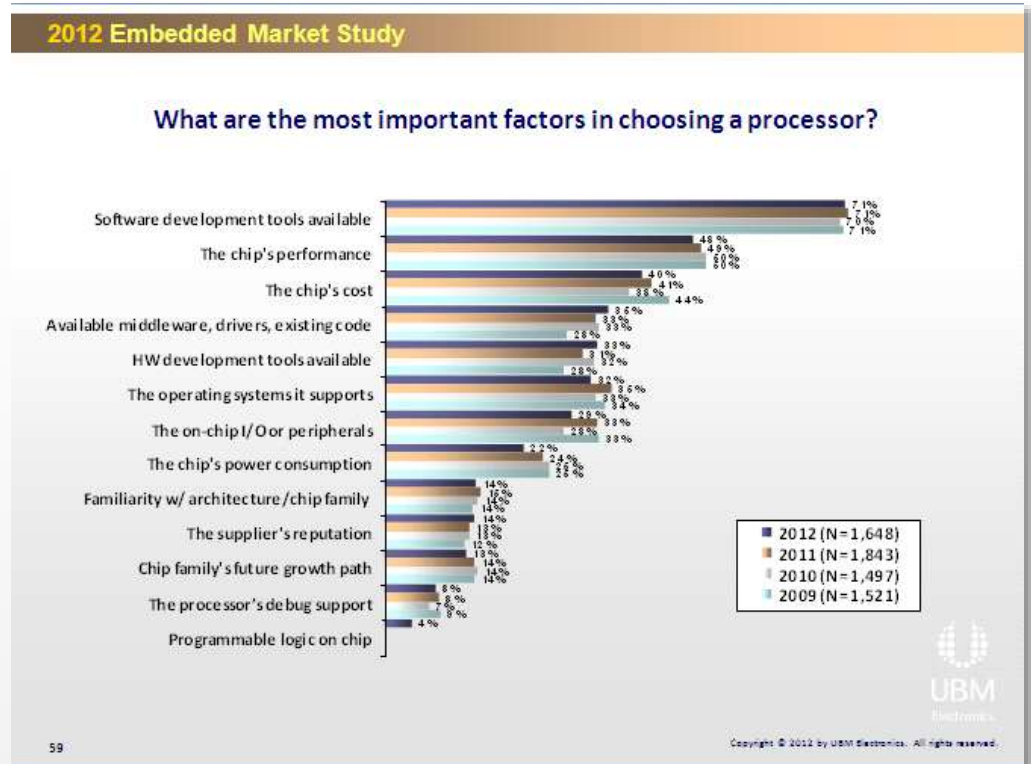
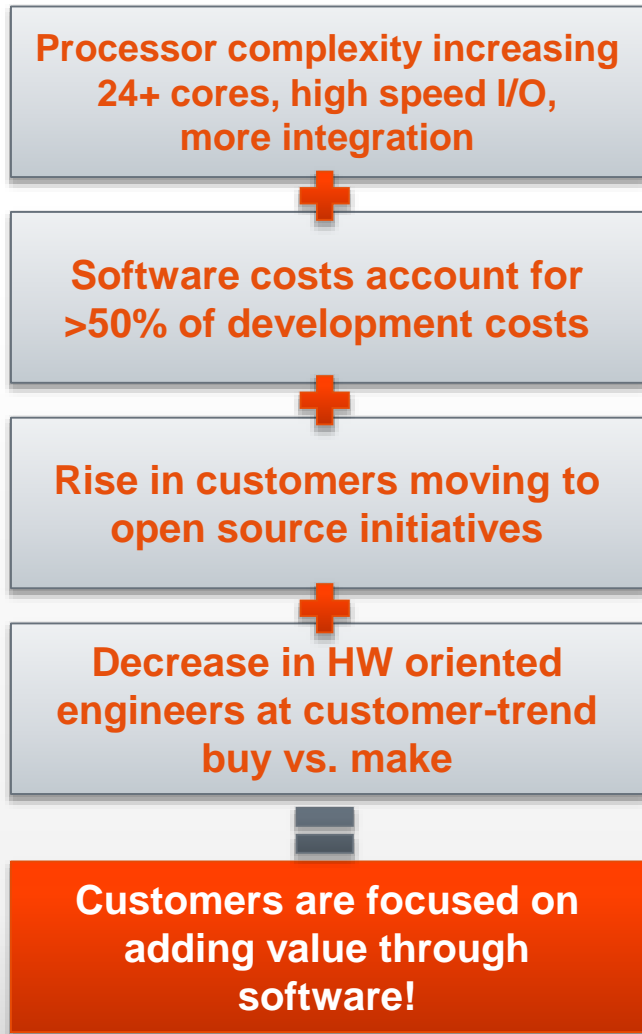
Linux® Kernel Lines of Code



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



Market Trends for Software in Embedded Devices

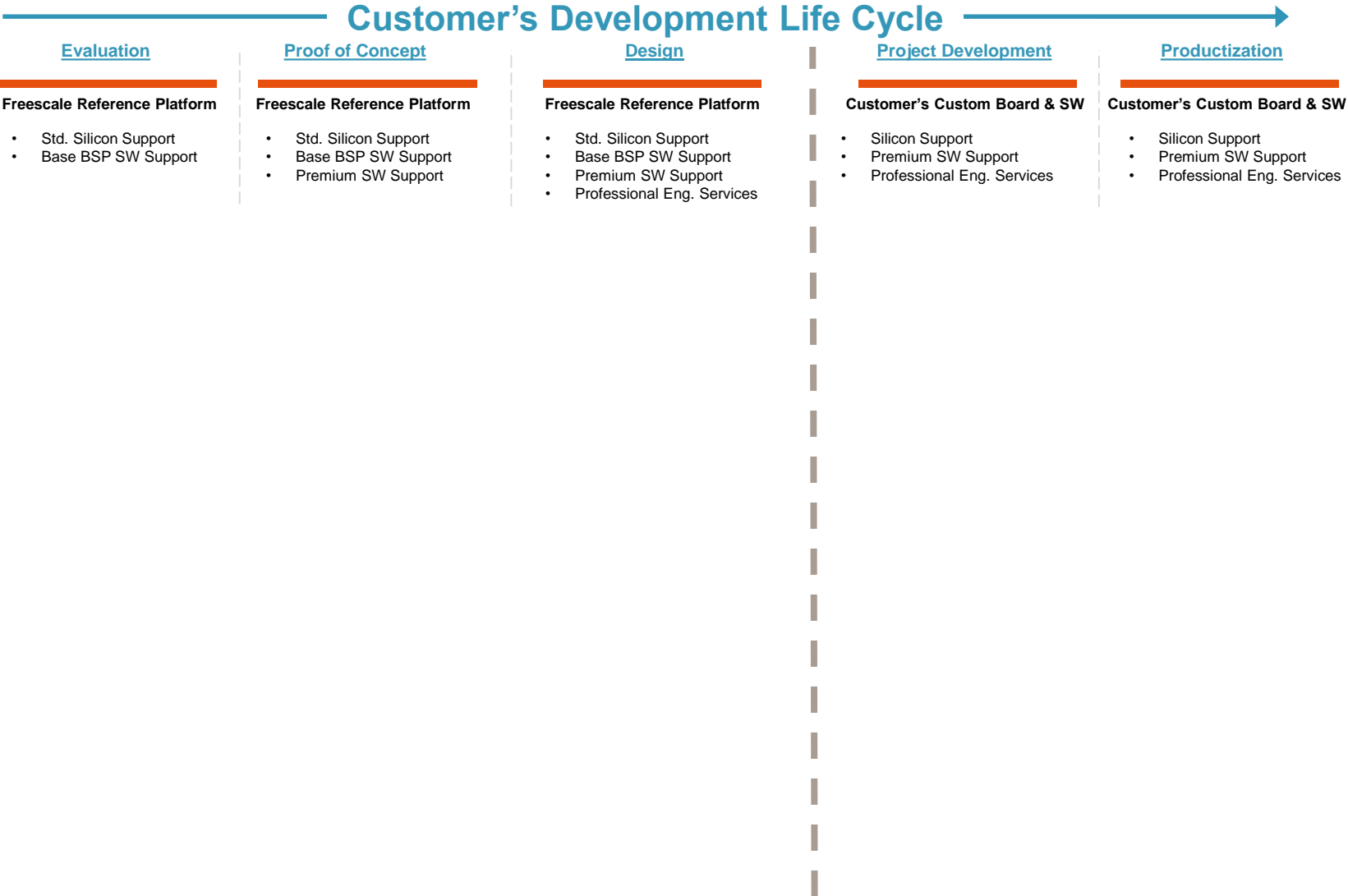


Freescale Software and Systems Organization

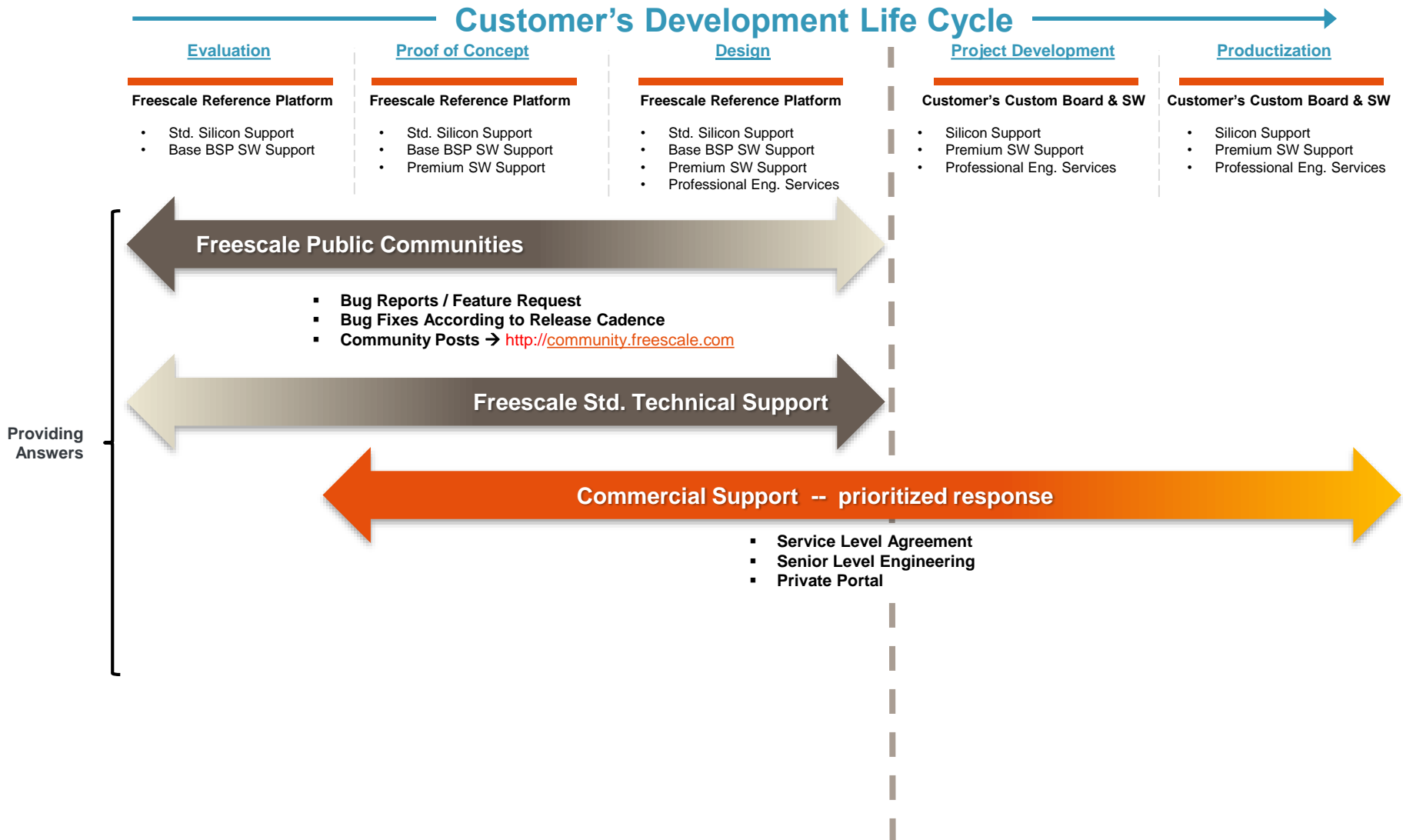


- Freescale has 1,000+ software engineers, 700+ focused on Digital Networking
- Increasing investment on software through hiring and acquisition
- Run-Time Technologies, Multi-core, Tools, Key Applications

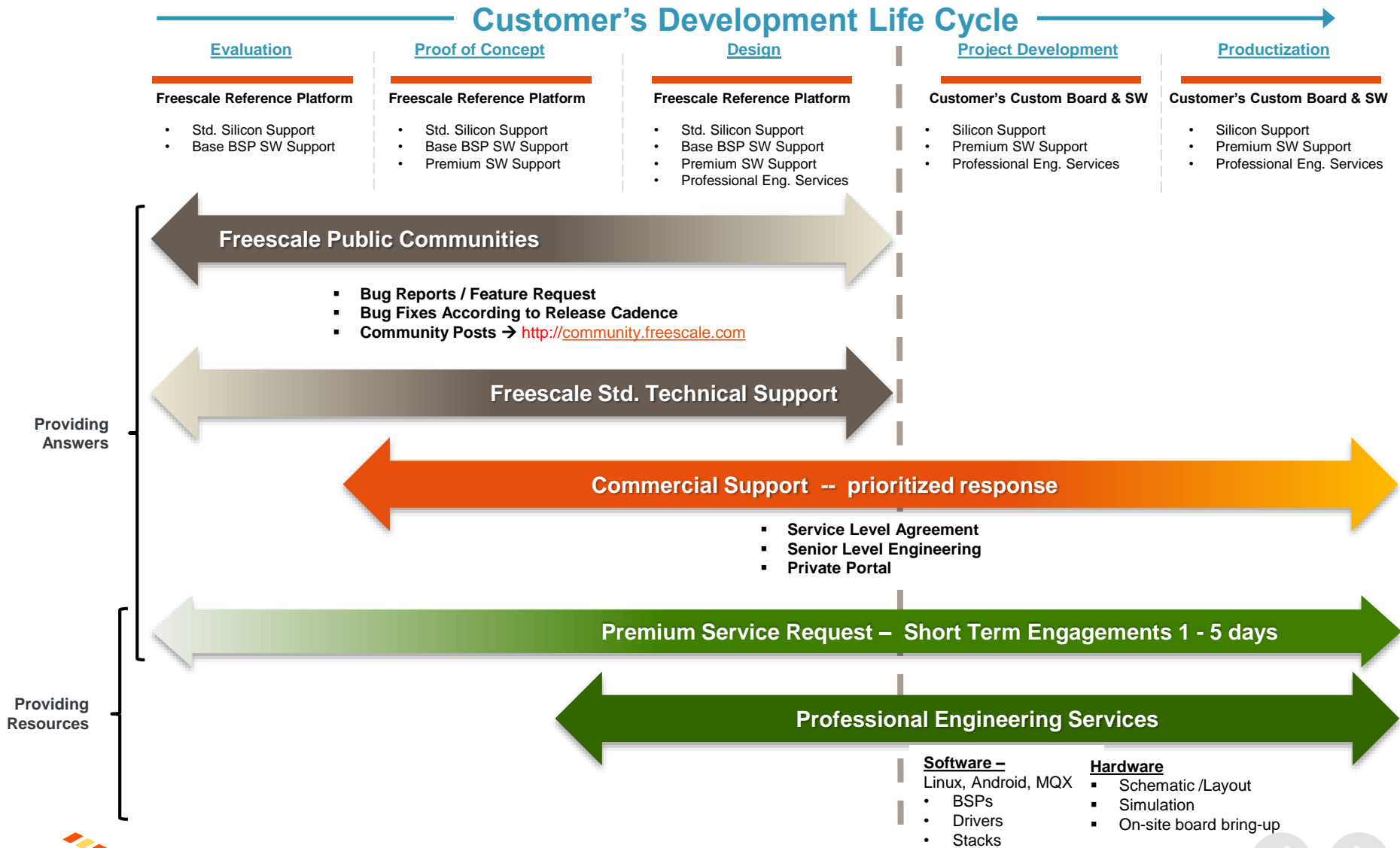
Software Support / Professional Services Coverage



Software Support / Professional Services Coverage






Software Support / Professional Services Coverage



Networking Software and Services Group

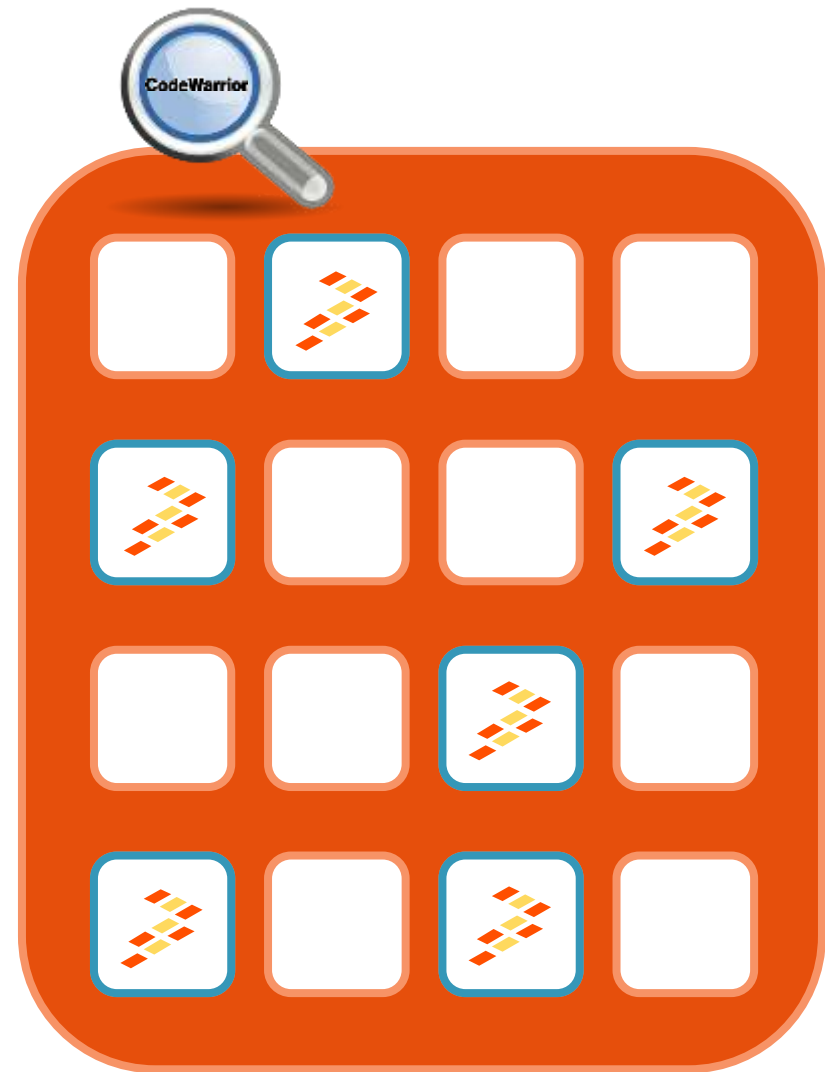
Software Products and Custom Services

Development Tools	Runtime Products	Solutions Reference	Linux® Services	Integration Services
<ul style="list-style-type: none"> • CodeWarrior <ul style="list-style-type: none"> - IDE - Debug - Compiler - Trace • QorIQ Optimization Suite <ul style="list-style-type: none"> - Scenarios Tools - DDrV 	<ul style="list-style-type: none"> • VortiQa Software Products <ul style="list-style-type: none"> - Application Identification Software (AIS) - Open Networking Switching Framework - Mobile Transport 	<ul style="list-style-type: none"> • Storage Controller • SDN Switch • Wireless LAN • Data Concentrator • Smart Converged Gateway • Digital Signage 	<ul style="list-style-type: none"> • Commercial Support • Frozen Branch • Application Specific Hardening • Feature Acceleration 	<ul style="list-style-type: none"> • Systems Consulting • Design Services • Porting • Migration
<p>CodeWarrior</p> <p>QorIQ</p>	<p>VortiQa</p>			

CodeWarrior for Networking

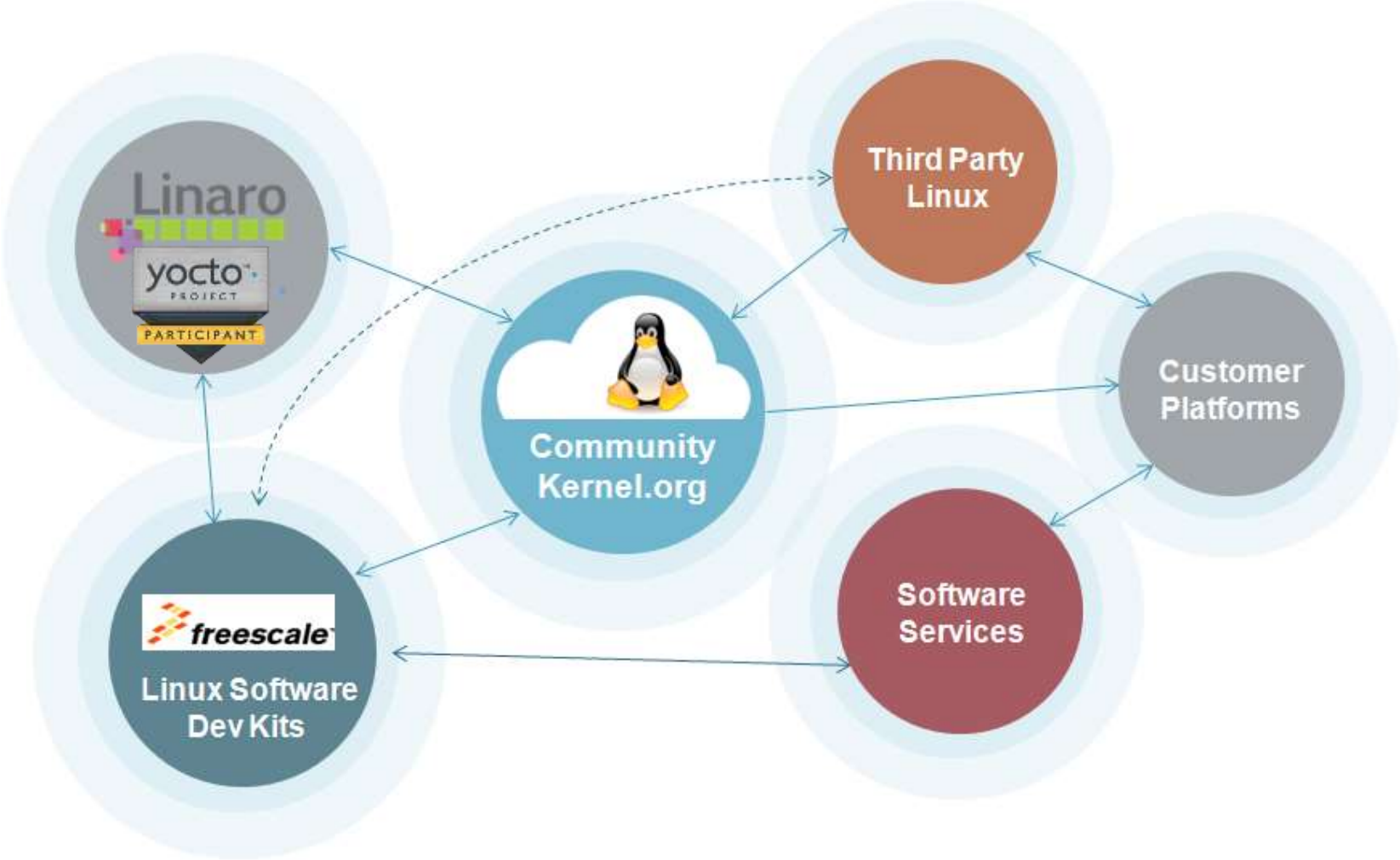
- Systems are complex - CodeWarrior makes debugging easy
 - Multicore or multiple cores
 - Heterogeneous or homogenous
 - Linux or Multi-OS Systems
 - ARM[®], Power Architecture[®] and StarCore
- Use one CodeWarrior software

freescale.com/CodeWarrior





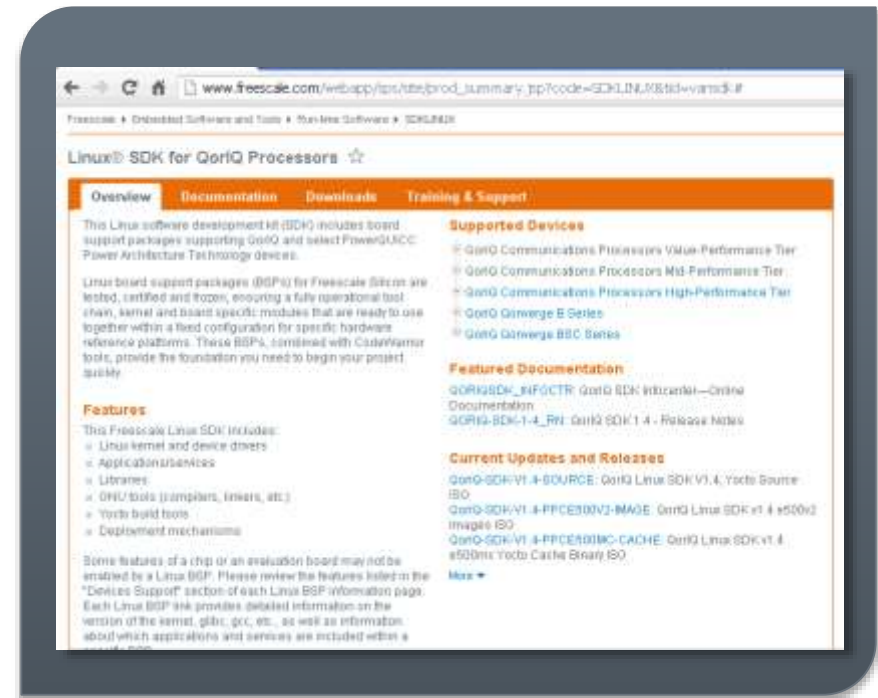
Freescal Linux Overview





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>





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	✓		✓	



VortiQa Portfolio

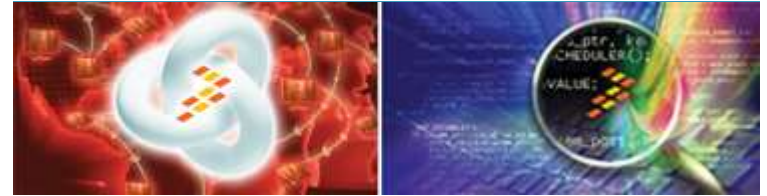
Module	Application
<p>VortiQa—Application Identification Suite</p> <p>Deep Packet Inspection: Identify the type of traffic on the network</p>	<p>VortiQa—Mobility Transport</p> <p>Transport Functions: IPsec, QoS, PDCP and GTP</p>
NEXT-GENERATION TECHNOLOGY	
<p>SDN—Open Network Switch</p>	<p>SDN—Open Network Director</p>
<p>VortiQa—Software Defined Networks (SDN)—Switch</p> <p>Data Plane Functions: layer 2, layer 3 and above; Firewall, NAT, QoS, DPI, etc. POC available; product in planning</p>	<p>VortiQa—Software Defined Networks (SDN)—Controller</p> <p>Controller Functions: PoC available; product in planning</p>



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

Pre-Boot Loader

RCW Configuration Tool



Pre-Boot Loader (RCW) Configuration

The screenshot displays the Processor Expert for QCS - Eclipse environment. The Project Explorer on the left shows the project structure for P2041RDB, with the PBL component selected. The Component Inspector window in the center shows the configuration for the PBL component, including the Reset Configuration Word (RCW) and SerDes PLL and Protocol Configuration. The Problems window at the bottom shows an error message: "ERROR: Error in the component setting. More details provided by Component Inspector for this component".

Name	Value	Details
Component name	PBL	
Device	PBL	PBL
Reset Configuration Word (RCW)		
RCW Source	LBC FCM (NAND flash)	
PLL Configuration		
SerDes PLL and Protocol Configuration		
SerDes Reference Clocks		
SD_REF_CLK1 [MHz]	100.0	100 MHz
SD_REF_CLK2 [MHz]	125.0	125 MHz
SRDS_EN [178]	0b1 - SerDes enabled	
SRDS_PRTCL [128-133]	0x0A - Bank 1: C-D; 2x SGMV [1,25...	
SRDS_RATIO_SPD [135]	0b0 - 2.5 or 5 Gbps/lane	
SRDS_RATIO_B1 [136-138]	0b011 - 40:1	
SerDes PLL 1 Clock	4.000 GHz	
SRDS_DIV_B1 [139-143]	0b011 - 40:1	
SRDS_RATIO_B2 [144-146]	0b011 - 40:1	
SerDes PLL 2 Clock	5.000 GHz	
SRDS_DIV_B2 [147]	0b0 - Divide by 1 off of Bank 2 PLL	
SRDS_LPD_B1 [152-161]		
SRDS_LPD_B2 - Lane A [162]	0b1 - Lane powered down	This property has to be set to the s...
SRDS_LPD_B2 - Lane B [163]	0b0 - Lane not powered down	This property has to be set to the c...
SRDS_LPD_B2 - Lane C [164]	0b0 - Lane not powered down	This property has to be set to the c...
SRDS_LPD_B2 - Lane D [165]	0b0 - Lane not powered down	This property has to be set to the c...
SRDS_RSVD0 [166-168]		
SRDS_RSVD1 [167]		
SRDS_RSVD2 [168]		
SRDS_RSVD3 [169]		

Problems Window:

1 error, 0 warnings, 0 others

Description

Errors (1 item)

ERROR: Error in the component setting. More details provided by Component Inspector for this component



Pre-Boot Loader Standard Component Interface

- Pre-Boot Loader (PBL) tool establishes all Reset Control Word settings

- PLL Configurations

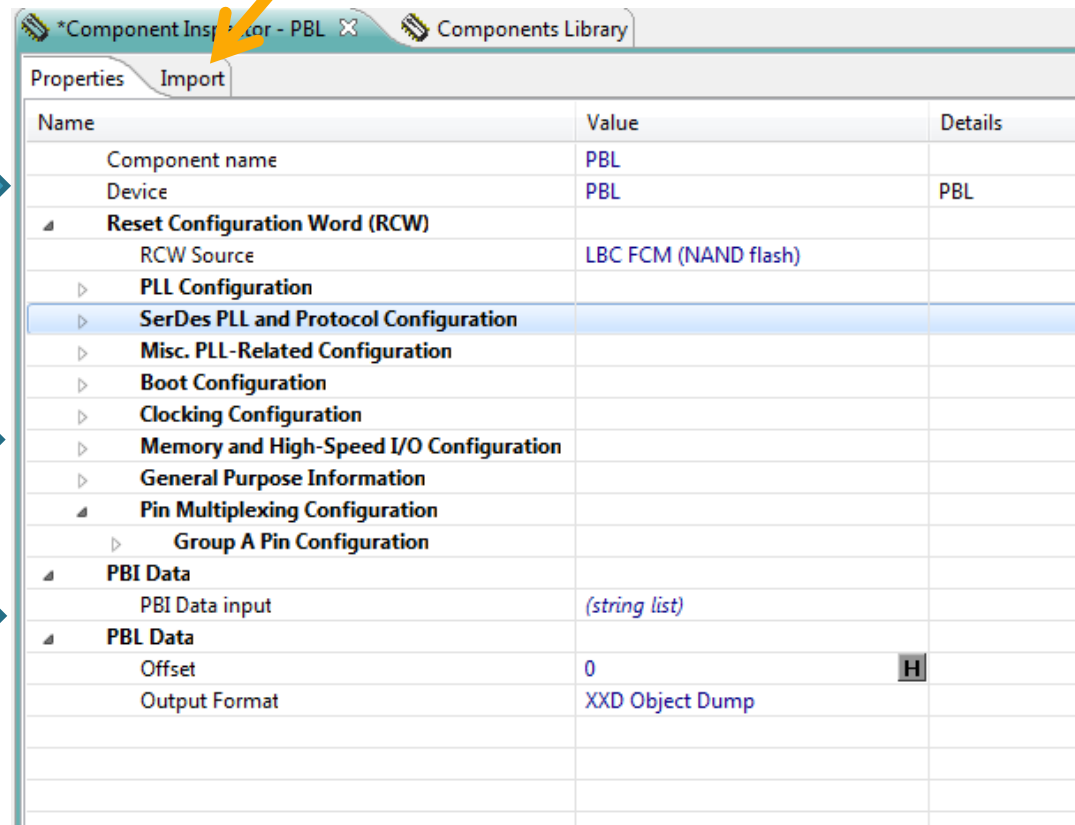
- SerDes Configuration

- Pin Muxing Configuration

- Output format selection

- Possibility to add PBI data

Possibility to import RCW settings



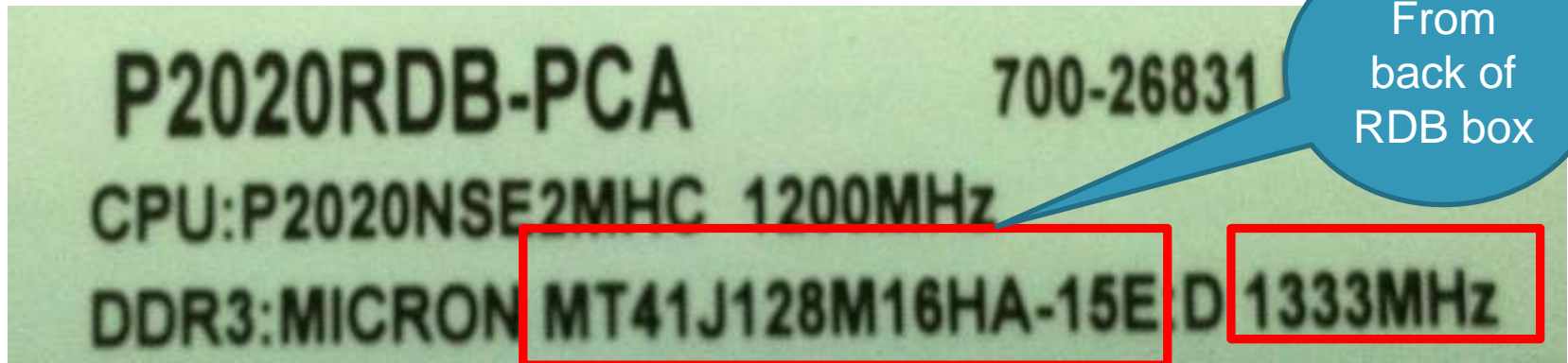
The screenshot shows the 'Component Inspector - PBL' window with the 'Import' tab selected. An orange arrow points to the 'Import' tab. The table below lists various configuration categories and their values.

Name	Value	Details
Component name	PBL	
Device	PBL	PBL
Reset Configuration Word (RCW)		
RCW Source	LBC FCM (NAND flash)	
▶ PLL Configuration		
▶ SerDes PLL and Protocol Configuration		
▶ Misc. PLL-Related Configuration		
▶ Boot Configuration		
▶ Clocking Configuration		
▶ Memory and High-Speed I/O Configuration		
▶ General Purpose Information		
▶ Pin Multiplexing Configuration		
▶ Group A Pin Configuration		
▶ PBI Data		
PBI Data input	(string list)	
▶ PBL Data		
Offset	0	H
Output Format	XXD Object Dump	

DDR Configuration



Get DRAM Information – P2020RDB-PCA



DDR3 SDRAM

MT41J512M4 – 64 Meg x 4 x 8 Banks

MT41J256M8 – 32 Meg x 8 x 8 Banks

MT41J128M16 – 16 Meg x 16 x 8 Banks

From DRAM datasheet

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

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

Features

- V_{DD} = V_{DDQ} = 1.5V ±0.075V
- 1.5V center-terminated push/pull I/O
- Differential bidirectional data strobe
- 8n-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 ^tCK
- 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

Options¹

- Configuration
 - 512 Meg x 4
 - 256 Meg x 8
 - 128 Meg x 16
- FBGA package (Pb-free) - x4, x8
 - 78-ball (8mm x 10.5mm) Rev. H,M,I,K
 - 78-ball (9mm x 11.5mm) Rev. D
- FBGA package (Pb-free) - x16
 - 96-ball (9mm x 14mm) Rev. D
 - 96-ball (8mm x 14mm) Rev. K
- Timing - cycle time
 - 938ps @ CL = 14 (DDR3-2133)
 - 1.071ns @ CL = 13 (DDR3-1866)
 - 1.25ns @ CL = 11 (DDR3-1600)
 - 1.5ns @ CL = 9 (DDR3-1333)
 - 1.87ns @ CL = 7 (DDR3-1066)
- Operating temperature
 - Commercial (0°C ≤ T_C ≤ +95°C)
 - Industrial (-40°C ≤ T_C ≤ +95°C)
- Revision

Marking

512M4
256M8
128M16
DA
HX
HA
JT
-093
-107
-125
-15E
-187E
None
IT
: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.

DDR Wizard Simplifies Configuration

New QorIQ Configuration Project

DDR Configuration

Configured device P2020

Configure: 1st DDR Controller

Configuration mode

- Auto configuration
- Import from memory file

Discrete DRAM DRAM Module

DDR Controller

Type: DDR 3

Data Rate: 800 MT/s

Ranks: 1

Data Bus width: 64 bits

CAS# Latency (tCL): 6 clocks

tRP/tRCD: 13.5 ns

ECC Enabled

DRAM Settings

DRAM Configuration per Rank: 1Gb: 128Mb x8

DRAM Speed Rating: 1333 MT/s

Select 1st DDR Controller

? < Back Next > Finish Cancel

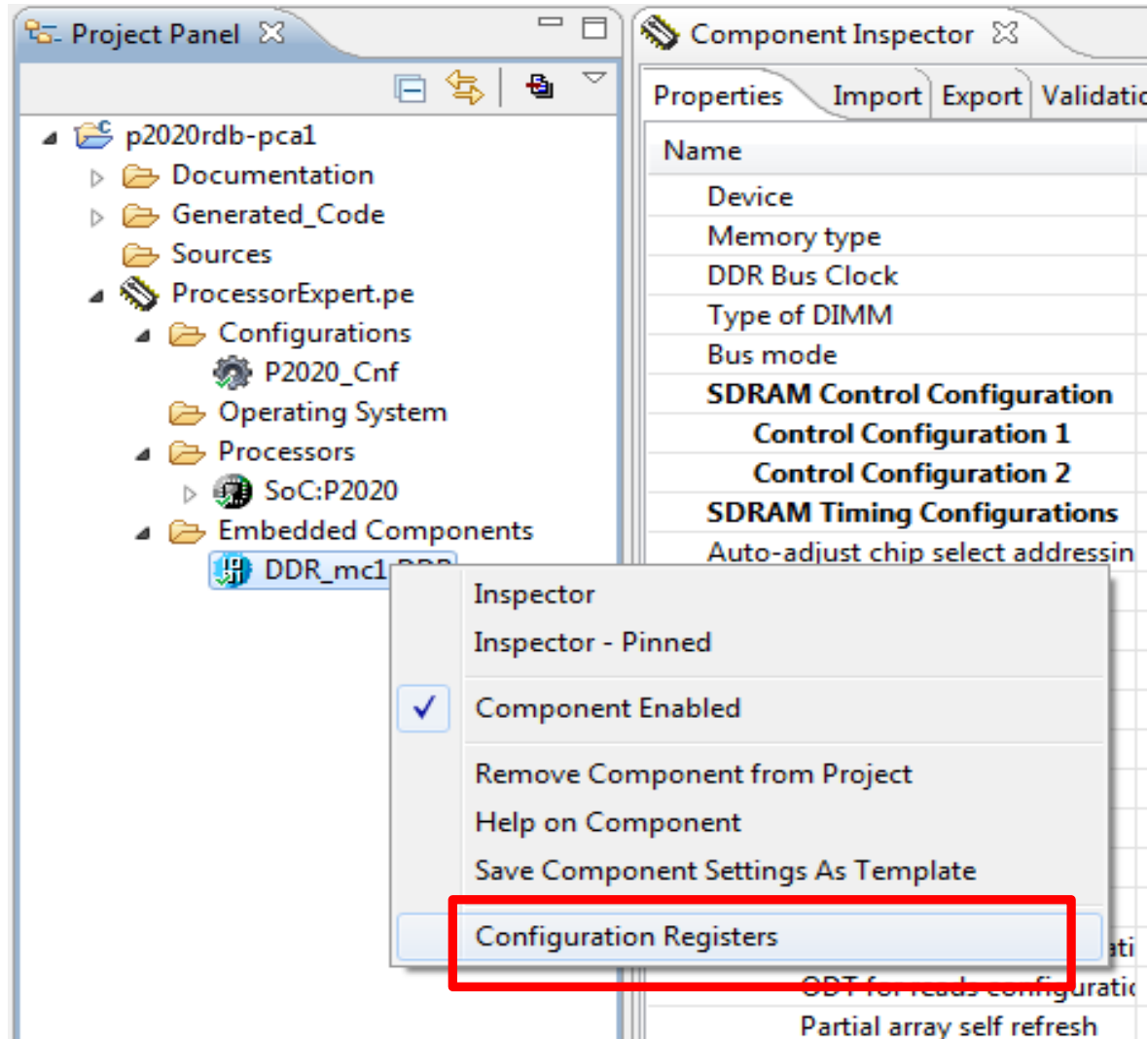
- From memory data sheet:
 - Maximum speed rating
 - Capacity

QCS Project Explorer

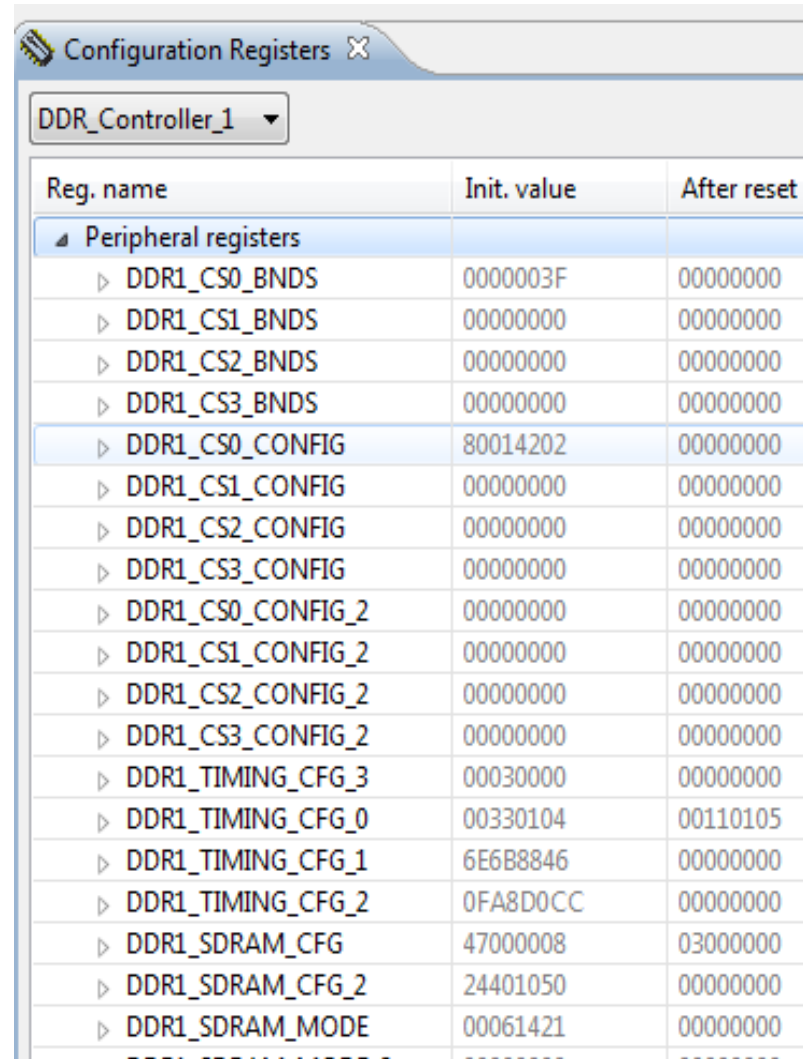
The screenshot displays the QCS Project Explorer interface. On the left, the Project Panel shows a tree view of the project structure. The 'Embedded Components' folder is expanded, and the 'DDR_mc1:DDR' component is highlighted with a red rectangle. On the right, the Component Inspector shows the properties of the selected component. The 'Properties' tab is active, displaying a table of properties.

Name	Value	Details
Device	DDR_Controller_1	DDR_Controller_1
Memory type	DDR 3	
DDR Bus Clock	400 MHz	DDR Data Rate: 800 MT/s
Type of DIMM	Unbuffered DIMMs	
Bus mode	64-bit bus	
SDRAM Control Configuration		
Control Configuration 1		
Control Configuration 2		
SDRAM Timing Configurations		
Auto-adjust chip select addressin	yes	
Chip Select 0	Enabled	
Memory Bounds		
Start Address	0	H
Size	1 GB	
Configuration		
Auto Precharge Always	no	
Internal Banks Number	8 internal banks	
Number of row bits	14 row bits	
Number of column bits	10 column bits	
ODT for writes configurati	Assert ODT only during writes to C...	
ODT for reads configurati	Never assert ODT for reads	
Partial array self refresh	Full Array	
Chip Select 1	Disabled	
Chip Select 2	Disabled	
Chip Select 3	Disabled	

Review DDR Registers Values



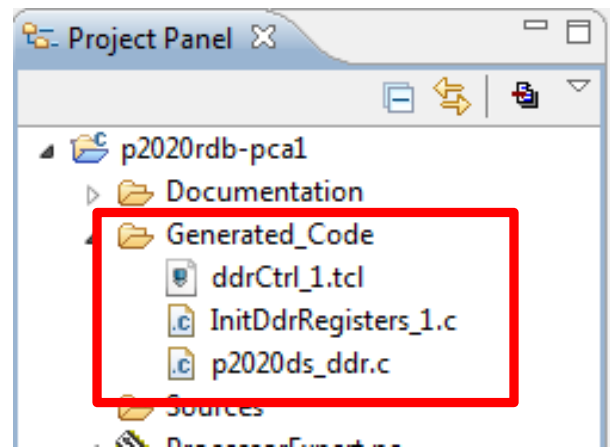
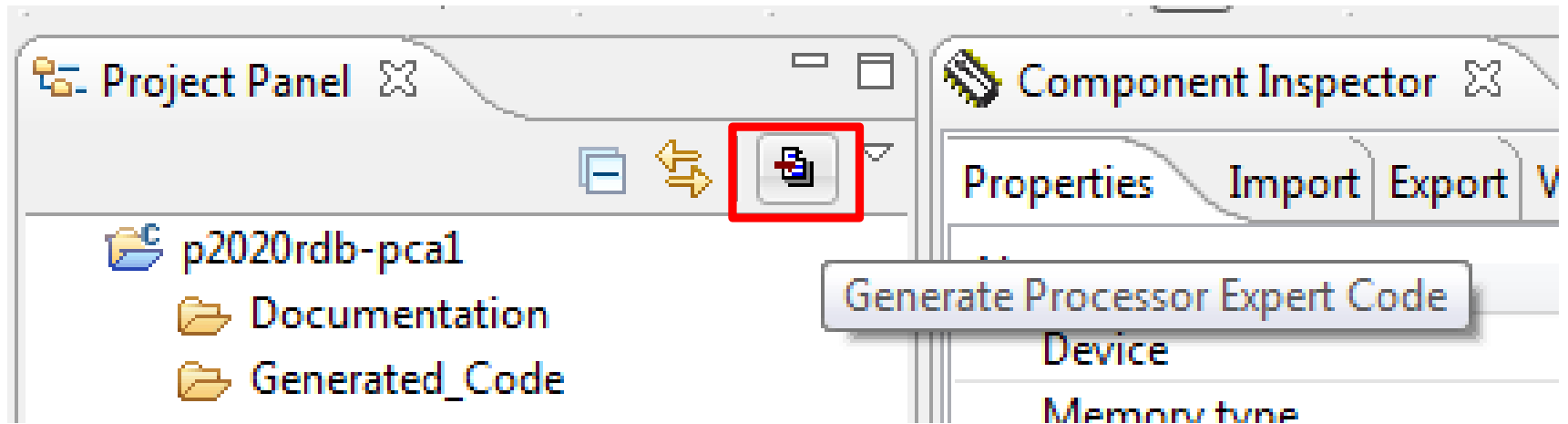
Review DDR Registers Values (continued)



The screenshot shows a software window titled "Configuration Registers" with a close button. Below the title bar is a dropdown menu set to "DDR_Controller_1". The main content is a table with three columns: "Reg. name", "Init. value", and "After reset". The table lists various peripheral registers for the DDR controller, including address bounds, configuration registers, timing registers, and SDRAM mode registers. The register "DDR1_CS0_CONFIG" is highlighted in blue.

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,
```



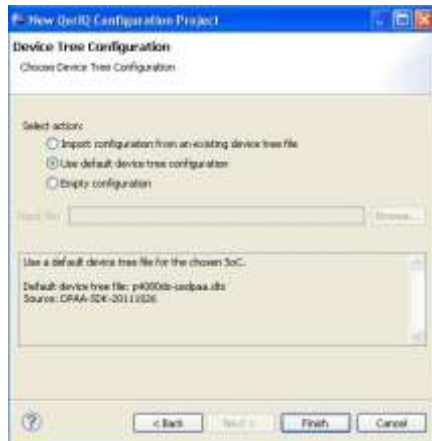
Device Tree Editor

Supports Hardware Device Trees (*.dts)

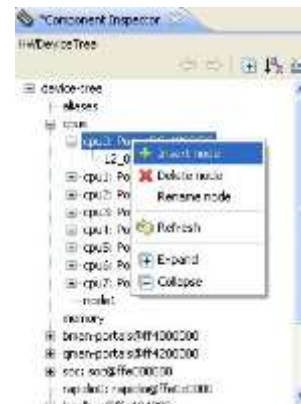


Hardware Device Tree Workflow

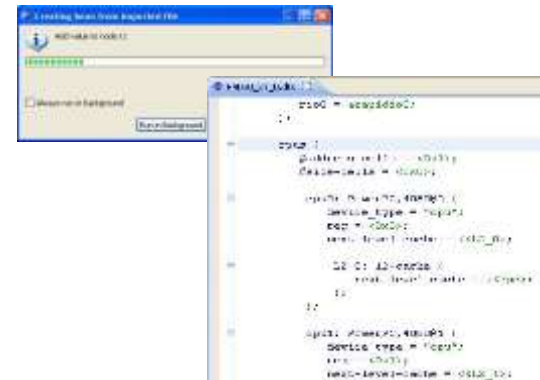
Create Project



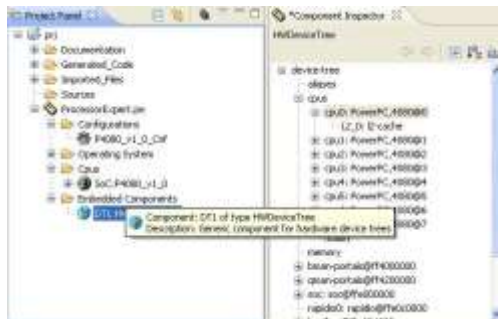
Configure Component



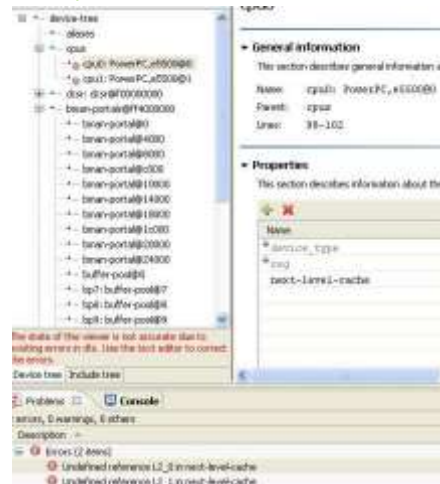
Generate Code



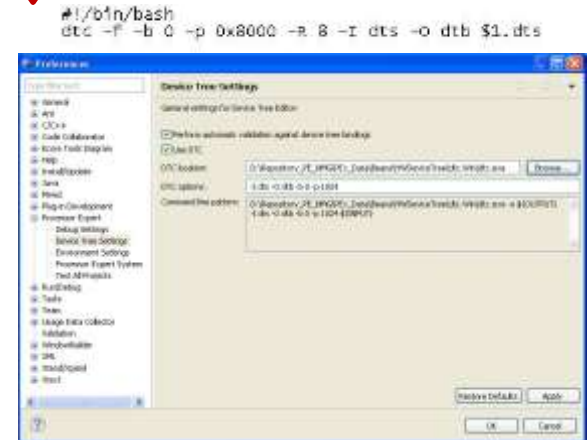
Select Component



Validate Component

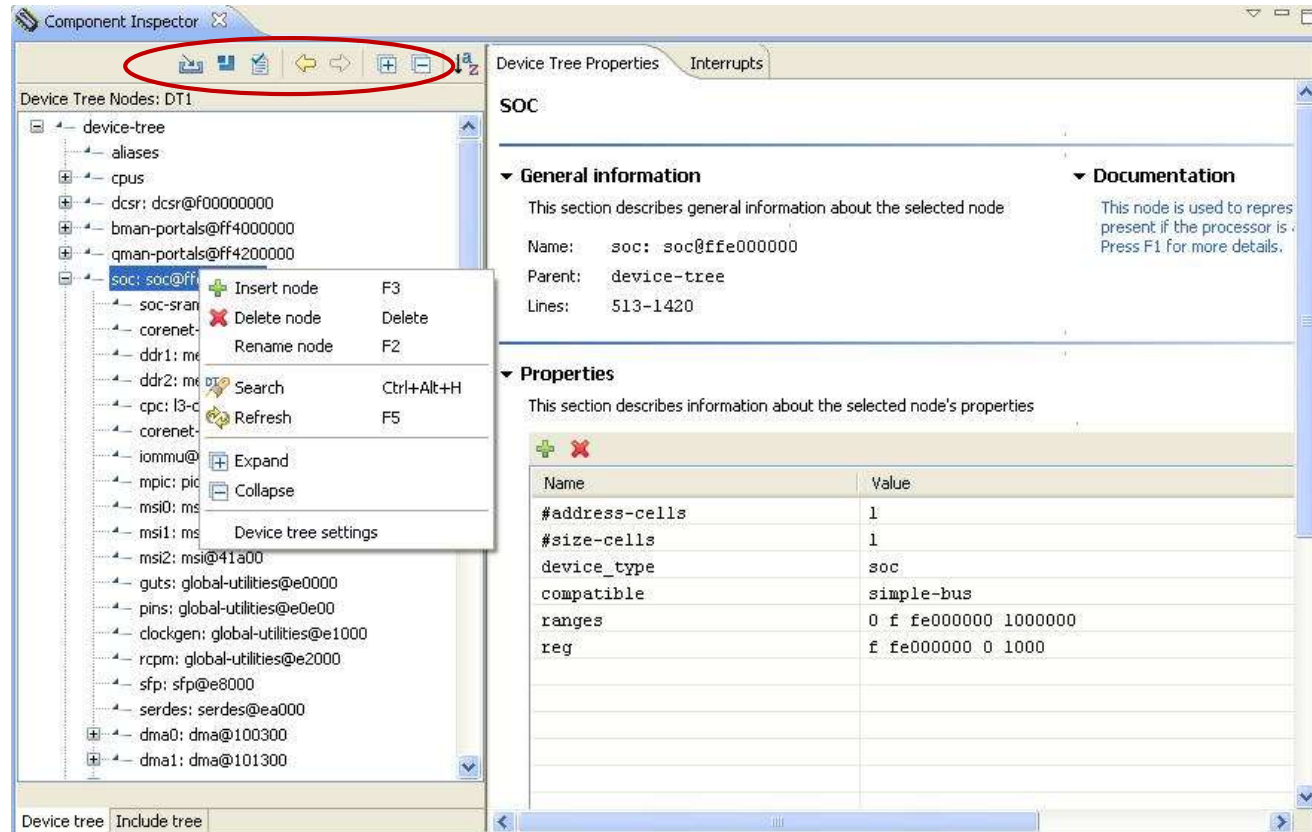


Compile DTS



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

The screenshot displays the Component Inspector interface with the following components:

- Device Tree Nodes:** A tree view showing nodes like `aliases`, `cpus`, `dcscr`, `bman-ports`, `qman-ports`, `soc`, `rapidio`, `localbus`, `pc0`, `pc1`, `pc2`, `memory`, and `fsldpaa`.
- Device Tree Properties:** A panel for the selected `aliases` node, showing general information (Name: `aliases`, Parent: `device-tree`, Lines: 23-101) and a table of properties.
- Properties Table:**

Name	Value
ccsr	soc
dcscr	dcscr
ethernet0	enet0
ethernet1	enet1
ethernet2	enet2
ethernet3	enet3
ethernet4	enet4
ethernet5	enet5
ethernet6	enet6
- Search Dialog:** A modal window titled "Search" with the "Device Tree Search" tab selected. It shows a search for "serial" with 20 matches. The "Working set" is defined as "prj".
- Search Results:** A list of 20 matches in the console, including:
 - 23: serial0 = <serial0>
 - 27: serial1 = <serial1>
 - 30: serial2 = <serial2>
 - 33: serial3 = <serial3>
 - 953: serial0: serial@11c500 {
 - 958: device_type = "serial"
 - 962: serial1: serial@11c600 {
 - 964: device_type = "serial"

define working set



Device Tree Bindings

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

The screenshot shows the Component Inspector interface. On the left, a tree view lists device tree nodes under 'DT1'. The 'dma0' node is selected. The main pane displays the 'dma0' node's properties and documentation. A table lists properties like '#address-cells', '#size-cells', 'compatible', 'reg', and 'ranges'. A tooltip for 'cell-index' is visible. On the right, a 'Properties' section explains the '#address-cells' and '#size-cells' properties.

Device Tree Nodes: DT1

- corenet-cf@18000
- iommu@20000
- rmu: rmu@d3000
- mpic: pic@40000
- timer@41100
- msi0: msi@41600
- msi1: msi@41800
- msi2: msi@41a00
- timer@42100
- guts: global-utilities@e0000
- pins: global-utilities@e0e00
- clockgen: global-utilities@e1000
- rcpm: global-utilities@e2000
- sfp: sfp@e8000
- serdes: serdes@ea000
- dma0: dma@100300**
- dma1: dma@101300
- sdhc: sdhc@114000
- i2c@118000
- i2c@119000
- i2c@119100
- serial0: serial@11c500
- serial1: serial@11c600
- serial2: serial@11d500
- serial3: serial@11d600
- gpio0: gpio@130000
- crypto: crypto@300000
- sec_mon: sec_mon@314000
- pme: pme@316000
- qman: qman@318000

Device Tree Properties

dma0

General information

This section describes general information about the selected node

Name: dma0: dma@100300
Parent: soc: soc@ffe000000
Lines: 1547-1582

Documentation

Direct Memory Access c
Press F1 for more detail

Properties

This section describes information about the selected node's properties

Name	Value
#address-cells	1
#size-cells	1
compatible	fsl,eloplus-dma
reg	100300 4
ranges	0 100100 200

cell-index

Property: cell-index
Type: U32

Description:
The cell-index property is the hardware index of the portal.

Properties

#address-cells

Required: yes
Value type: CELLLIST
Constraints: not defined
Description: This property may be used in any device node that has children in the device tree hierarchy and describes how child device nodes should be addressed. It defines the number of <u32> cells used to encode the address field in a child node's "reg" property. If missing, a client program should assume a default value of 2.

#size-cells

Required: yes
Value type: CELLLIST
Constraints: not defined
Description: This property may be used in any device node that has children in the device tree hierarchy and describes how child device nodes should be addressed. It defines the number of <u32> cells used to encode the size field in a child node's "reg" property. If missing, a client program should assume a default value of 1.



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)

The screenshot displays the Component Inspector interface with three main panes. The left pane shows a tree view of device tree files, including `DT2HWDeviceTree.dts` and `p5020ds.dts`. The middle pane shows the content of `DT2HWDeviceTree.dts`, with a tooltip indicating that the `'bman-portal@10000'` node is defined in `p5020ds.dts`, line 88. The right pane shows the content of `p5020ds.dts`, with a tooltip indicating that the `'compatible'` property is defined in `qorq-bman1-portals.dts`, line 94. Two callouts point to the `/include/` declaration and the `node reference` in the `p5020ds.dts` file.

```
256
257 bman-portal@10000 {
258
259
260
261
262
263
264
265
266 compatible = "fsl,bman-portal";
267 reg = <0x1c000 0x4000 0x107000 0x1000>;
268 interrupts = <119 2 0 0>;
269
270
271
272 bman-portal@30000 {
273 cell-index = <0x8>;
274 compatible = "fsl,bman-portal";
275 reg = <0x20000 0x4000 0x108000 0x1000>;
276 interrupts = <121 2 0 0>;
277
278
279
280 bman-portal@4000 {
281 cell-index = <0x9>;
282 compatible = "fsl,bman-portal";
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000
```

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

The screenshot shows the 'Device Tree Properties' tool with the 'Interrupts' tab selected. The left pane shows a tree of device nodes, including various 'qman-portal' and 'message-unit' nodes. The right pane shows the 'Properties' for the selected node, with a table of interrupt levels and a 'Domain map' section.

Interrupts Properties Table:

Interrupt number	Interrupt level/sense
233	Low to High
234	Low to High
235	Low to High
236	Low to High
237	Low to High
238	Low to High
	Active Low
	Active High
	High to Low

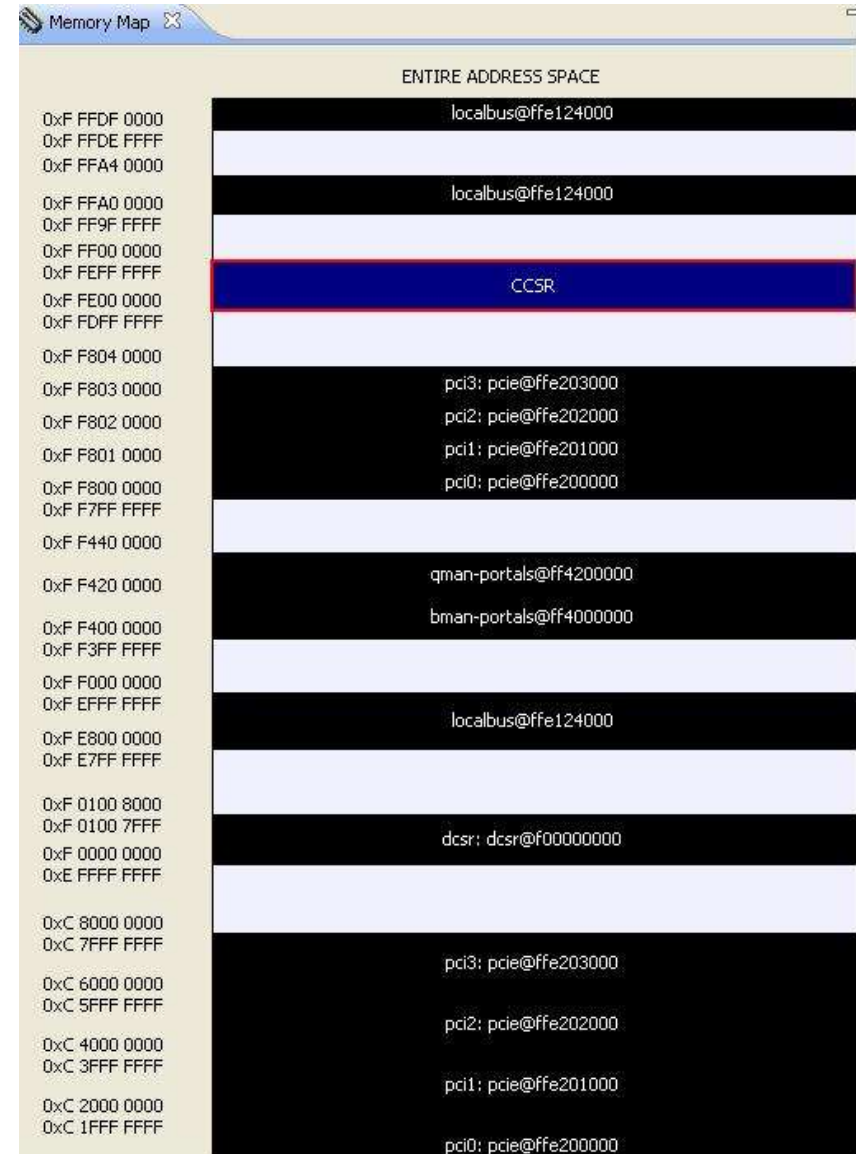
Domain map Table (Dev. 0):

Child interrupts	Int. 1	Int. 2	Int. 3	Int. 40
INTA	Disabled	Disabled	Disabled	Active Low
INTB	Active Low	Disabled	Disabled	Disabled
INTC	Disabled	Active Low	Disabled	Disabled
INTD	Disabled	Disabled	Active Low	Disabled
				Disabled
				Low to High
				Active Low
				Active High
				High to Low



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

The screenshot displays the Component Inspector GUI with three main panels illustrating synchronization:

- Device Tree Nodes:** A tree view showing nodes like `lbc: localbus@ffe12400`. A blue arrow points from this node to the Properties panel.
- Device Tree Properties:** Shows the properties for the selected `lbc` node, including a table:

Name	Value
reg	f fe124000 0
ranges	0 0 f e8000000
compatible	fsl,p4080-elbc
interrupts	25 2 0 0
#address-cells	2
#size-cells	1

A blue arrow points from this table to the Memory Map panel.
- Memory Map:** Shows the `ENTIRE ADDRESS SPACE` with various memory regions. A red box highlights the `lbc: localbus@ffe12400` region, with a blue arrow pointing back to the Device Tree Nodes panel.

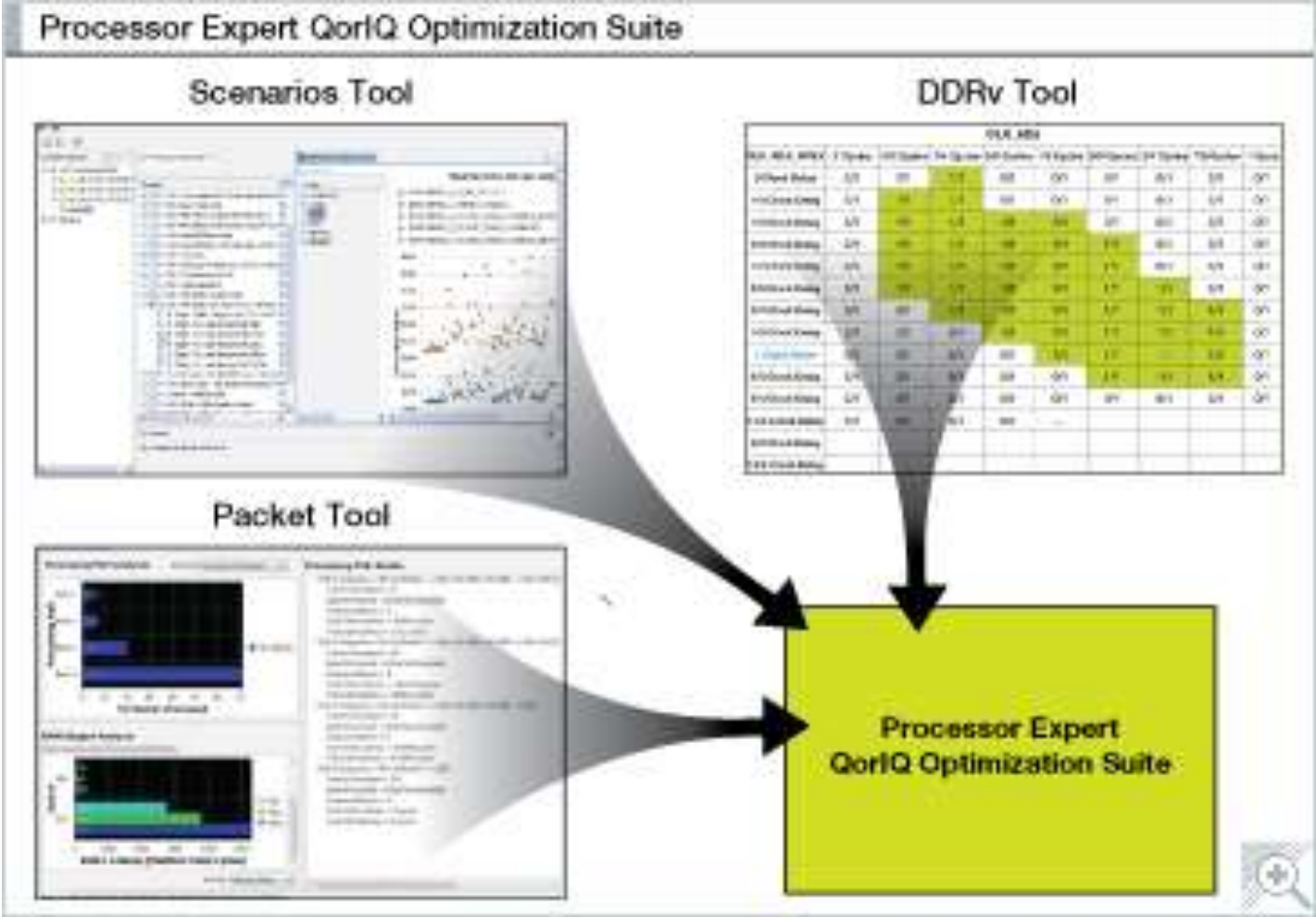
At the bottom, the `DT1HWDeviceTree.dts` text editor shows the corresponding device tree source code for the `lbc` node, with a blue arrow pointing from the Properties panel to it.



QorIQ Optimization Suite



QorIQ Optimization Suite



What is the PEx Optimization Suite?

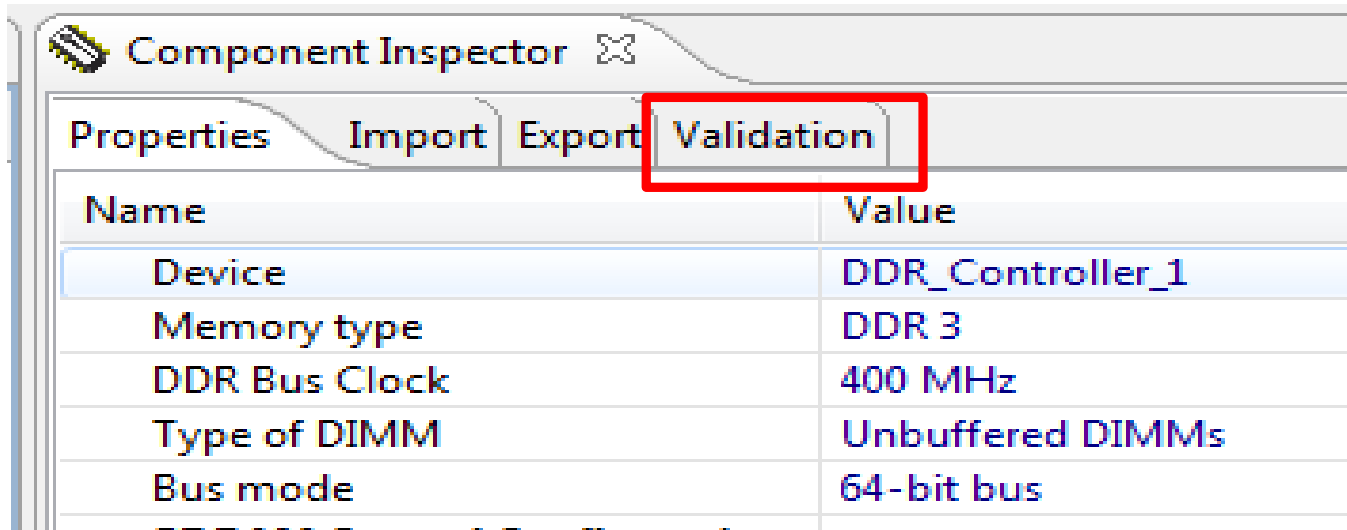
- A new generation of products aimed at allowing customers to solve systems and application performance problems in the QorIQ and Layerscape family of devices
 - Users can analyze their applications unencumbered by the complexity of the debug IP
 - Provides a simple, clear and concise way of configuring 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

The screenshot shows the DDR Validation Panel interface with the following components and callouts:

- 1**: Points to the **Validation** tab in the top navigation bar.
- 2**: Points to the **Test** list in the "Choose DDR tests to be executed" section.
- Test stages / scenarios**: A blue callout bubble pointing to the "Validation stages" folder and its sub-items: "Stage 2", "Centering the clock", "Read ODT and driver", and "Write ODT and driver".
- Tests to be executed per each scenario**: A blue callout bubble pointing to the "Select Test Options" section, which contains a table:

Test	Run times
<input type="checkbox"/> Pattern Write	0
<input type="checkbox"/> Read Write Compare	0
<input type="checkbox"/> Walking Ones	0
<input type="checkbox"/> Walking Zeros	0
- Scenario details**: A blue callout bubble pointing to the "Test results" table.
- HW Connection setup**: A blue callout bubble pointing to the "Connection settings" section at the bottom, which includes a "Remote system IP" field (127.0.0.1) and a "Discover" button.

The "Test results" table is titled "CLK_ADJ" and has columns for "CLK_ADJ_WRLV", "0 cycles", "1/8 cycles", "1/4 cycles", "3/8 cycles", "1/2 cycles", "5/8 cycles", "3/4 cycles", "7/8 cycles", and "1 cycles". The rows list various clock delays from "0 clock delay" to "5/2 clock delay".

Configure DDR Tests To Be Run

Check the scenarios to be tested (TBD)

Double click on test to see its content

Choose which test to be executed and how many times

1

2

3

Component Inspector

Properties Import Export Validation

Choose DDR tests to be executed

Test	Result
<input checked="" type="checkbox"/> Validation stages	
<input checked="" type="checkbox"/> Stage 1	
<input checked="" type="checkbox"/> Centering the clock	
<input checked="" type="checkbox"/> Read ODT and driver	
<input checked="" type="checkbox"/> Write ODT and driver	
<input type="checkbox"/> Basic Testing	
<input type="checkbox"/> DDR Test	

Select Test Options:

Test	Run times
<input type="checkbox"/> Pattern Write	1
<input checked="" type="checkbox"/> Read Write Compare	3
<input type="checkbox"/> Walking Ones	1
<input type="checkbox"/> Walking Zeros	1

Start Validation

Observe DDR Validation Test Results

Component Inspector

Properties Import Export Validation

Choose DDR tests to be executed

Test Result

- Validation stages
 - Stage 1
 - Centering the clock
 - Read ODT and driver
 - Write ODT and driver
 - Smoke Tests

Select Test Options:

Test	Run times
<input type="checkbox"/> Pattern Write	1
<input checked="" type="checkbox"/> Read Write Compare	3
<input type="checkbox"/> Walking Ones	1
<input type="checkbox"/> Walking Zeros	1

Cancel

Connection settings

Remote system IP: 127.0.0.1 Disconnect

Available connections: System: P5020; ETAP id:10.82.138.191

Test results

CLK_ADJ

CLK_ADJ_WRLV	0 cycles	1/8 cycles	1/4 cycles	3/8 cycles	1/2 cycles	5/8 cycles	3/4 cycles	7/8 cycles	1 cycles
0 clock delay	0/3	0/3	0/3	0/3	0/3	0/3	0/3	x	x
1/8 clock delay	0/3	0/3	0/3	0/3	0/3	0/3	0/3	0/3	0/3
1/4 clock delay	0/3	0/3	3/3	3/3	0/3	0/3	0/3	0/3	0/3
3/8 clock delay	0/3	0/3	3/3	3/3	2/3	0/3	0/3	0/3	0/3
1/2 clock delay	0/3	0/3	3/3	3/3	3/3	3/3	0/3	0/3	0/3
5/8 clock delay	0/3	0/3	3/3	3/3	3/3	3/3	0/3	0/3	0/3
3/4 clock delay	0/3	0/3	3/3	3/3	3/3	3/3	3/3	3/3	0/3
7/8 clock delay	0/3	0/3	2/3	3/3	3/3	3/3	3/3	3/3	0/3
1 clock delay	0/3	0/3	0/3	0/3	3/3	3/3	3/3	3/3	0/3
9/8 clock delay	0/3	0/3	0/3	0/3	0/3	3/3	3/3	3/3	0/3
5/4 clock delay	0/3	0/3	0/3	0/3	0/3	0/3	3/3	3/3	0/3
11/8 clock delay	0/3	0/3	0/3	0/3	0/3	0/3	0/3	3/3	0/3
3/2 clock delay	0/3	0/3	0/3	0/3	0/3	0/3	0/3	0/3	0/3
13/8 clock delay	0/3	0/3	0/3	0/3	0/3	0/3	0/3	0/3	0/3
7/4 clock delay									
15/8 clock delay									
2 clock delay									
17/8 clock delay									
9/4 clock delay									
19/8 clock delay									
5/2 clock delay									

WRLVL_START

Test results per DDR configuration

Optimal DDR Configuration (Read ODT)

Optimal DDR configuration is bolded

1

2

3

4

Tests to be executed can be changed between executions

Proceed to next validation step

Component Inspector

Basic Advanced Expe

Properties Import Export Validation

Choose DDR tests to be executed

Test	Result
<input checked="" type="checkbox"/> Validation stages	
<input checked="" type="checkbox"/> Stage 1	
<input type="checkbox"/> Centering the clock	
<input checked="" type="checkbox"/> Read ODT and driver	✓
<input checked="" type="checkbox"/> Write ODT and driver	
<input type="checkbox"/> Basic Testing	
<input type="checkbox"/> DDR Test	

Test Options:

Test	Run times
<input type="checkbox"/> Pattern Write	1
<input checked="" type="checkbox"/> Read Write Compare	1
<input type="checkbox"/> Walking Ones	1
<input type="checkbox"/> Walking Zeros	1

Test results

DRAM driver strength

Controller_DRAM	40 ohm - half drive strength	34 ohm - full driver strength
43 ohm	1/1	1/1
50 ohm	1/1	1/1
55 ohm	1/1	1/1
60 ohm	1/1	1/1
75 ohm	1/1	1/1
120 ohm	1/1	1/1
150 ohm	1/1	1/1

controller ODT

Next step

P2020RDB-PCA DDRV Optimized Test Results

Test results

CLK_ADJ_WRLV	CLK_ADJ								
	0 cycles	1/8 cycles	1/4 cycles	3/8 cycles	1/2 cycles	5/8 cycles	3/4 cycles	7/8 cycles	1 cycl
0 clock delay	0/1	0/1	1/1	0/1	0/1	0/1	0/1	0/1	0/1
1/8 clock delay	0/1	1/1	1/1	0/1	0/1	0/1	0/1	0/1	0/1
1/4 clock delay	0/1	1/1	1/1	1/1	1/1	0/1	0/1	0/1	0/1
3/8 clock delay	0/1	1/1	1/1	1/1	1/1	1/1	0/1	0/1	0/1
1/2 clock delay	0/1	1/1	1/1	1/1	1/1	1/1	1/1	0/1	0/1
5/8 clock delay	0/1	1/1	1/1	1/1	1/1	1/1	1/1	0/1	0/1
3/4 clock delay	0/1	0/1	1/1	1/1	1/1	1/1	1/1	1/1	0/1
7/8 clock delay	0/1	0/1	0/1	1/1	1/1	1/1	1/1	1/1	0/1
1 clock delay	0/1	0/1	0/1	0/1	1/1	1/1	1/1	1/1	0/1
9/8 clock delay	0/1	0/1	0/1	0/1	0/1	1/1	1/1	1/1	0/1
5/4 clock delay	0/1	0/1	0/1	0/1	0/1	0/1	0/1	0/1	0/1
11/8 clock delay	0/1			0/1	...				
3/2 clock delay									
13/8 clock delay									

WRLVL_START

Optimal settings

DDR_Controller_1

Reg. name	Init. value
DDR1_SDRAM_CFG	47000008
DDR1_SDRAM_CFG_2	24401050
DDR1_SDRAM_MODE	00061421
DDR1_SDRAM_MODE_2	00000000
DDR1_SDRAM_MD_CNTL	00000000
DDR1_SDRAM_INTERVAL	0C30030C
DDR1_DATA_INIT	00000000
DDR1_SDRAM_CLK_CNTL	02000000
DDR1_INIT_ADDR	00000000
DDR1_INIT_EXT_ADDRESS	00000000
DDR1_TIMING_CFG_4	00220001
DDR1_TIMING_CFG_5	02401400
DDR1_ZQ_CNTL	89080600
DDR1_WRLVL_CNTL	8655F605
DDR1_SR_CNTR	00000000
DDR1_SDRAM_RCW_1	00000000
DDR1_SDRAM_RCW_2	00000000
DDR1_WRLVL_CNTL_2	00000000
DDR1_WRLVL_CNTL_3	00000000
DDR1_SDRAM_MODE_3	00000000
DDR1_SDRAM_MODE_4	00000000
DDR1_SDRAM_MODE_5	00000000
DDR1_SDRAM_MODE_6	00000000
DDR1_SDRAM_MODE_7	00000000
DDR1_SDRAM_MODE_8	00000000
DDR1_DDRDSR_1	00000000
DDR1_DDRDSR_2	00000000

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.
```

The screenshot shows the DDR Controller configuration tool. On the left, a table titled 'CLK_ADJ' displays timing parameters for various clock delays. The columns represent different cycle counts: 0 cycles, 1/8 cycles, 1/4 cycles, 3/8 cycles, 1/2 cycles, 5/8 cycles, 3/4 cycles, 7/8 cycles, and 1 cycle. The rows represent clock delays from 0 to 13/8. The values in the table are mostly 0/1, with some green highlights. A red box highlights the value '1/1' in the row for '1 clock delay' under the '1/1 cycles' column. A blue callout bubble with the text 'Uboot values' points to this red box.

CLK_ADJ\WRLV	0 cycles	1/8 cycles	1/4 cycles	3/8 cycles	1/2 cycles	5/8 cycles	3/4 cycles	7/8 cycles	1 cycle
0 clock delay	0/1	0/1	1/1	0/1	0/1	0/1	0/1	0/1	0/1
1/8 clock delay	0/1	1/1	1/1	0/1	0/1	0/1	0/1	0/1	0/1
1/4 clock delay	0/1	1/1	1/1	1/1	1/1	0/1	0/1	0/1	0/1
3/8 clock delay	0/1	1/1	1/1	1/1	1/1	1/1	0/1	0/1	0/1
1/2 clock delay	0/1	1/1	1/1	1/1	1/1	1/1	1/1	0/1	0/1
5/8 clock delay	0/1	1/1	1/1	1/1	1/1	1/1	1/1	0/1	0/1
3/4 clock delay	0/1	0/1	1/1	1/1	1/1	1/1	1/1	1/1	0/1
7/8 clock delay	0/1	0/1	0/1	1/1	1/1	1/1	1/1	1/1	0/1
1 clock delay	0/1	0/1	0/1	0/1	1/1	1/1	1/1	1/1	0/1
9/8 clock delay	0/1	0/1	0/1	0/1	0/1	1/1	1/1	1/1	0/1
5/4 clock delay	0/1	0/1	0/1	0/1	0/1	0/1	0/1	0/1	0/1
11/8 clock delay	0/1	0/1	0/1	0/1
3/2 clock delay
13/8 clock delay

Register List (DDR_Controller_1):

- DDR1_SDRAM_CFG: 47000008
- DDR1_SDRAM_CFG_2: 24401050
- DDR1_SDRAM_MODE: 00061421
- DDR1_SDRAM_MODE_2: 00000000
- DDR1_SDRAM_MD_CNTL: 00000000
- DDR1_SDRAM_INTERVAL: 0C30030C
- DDR1_DATA_INIT: 00000000
- DDR1_SDRAM_CLK_CNTL: 03000000
- DDR1_INIT_ADDR: 00000000
- DDR1_INIT_EXT_ADDRESS: 00000000
- DDR1_TIMING_CFG_4: 00220001
- DDR1_TIMING_CFG_5: 02401400
- DDR1_ZQ_CNTL: 89080600
- DDR1_WRLVL_CNTL: 8655f608
- DDR1_SR_CNTR: 00000000
- DDR1_SDRAM_RCW_1: 00000000
- DDR1_SDRAM_RCW_2: 00000000
- DDR1_WRLVL_CNTL_2: 00000000
- DDR1_WRLVL_CNTL_3: 00000000
- DDR1_SDRAM_MODE_3: 00000000
- DDR1_SDRAM_MODE_4: 00000000
- DDR1_SDRAM_MODE_5: 00000000
- DDR1_SDRAM_MODE_6: 00000000
- DDR1_SDRAM_MODE_7: 00000000
- DDR1_SDRAM_MODE_8: 00000000
- DDR1_DDRDSR_1: 00000000
- DDR1_DDRDSR_2: 00000000

QorIQ Scenario Tool

. . . A Tool in QorIQ Optimization Suite



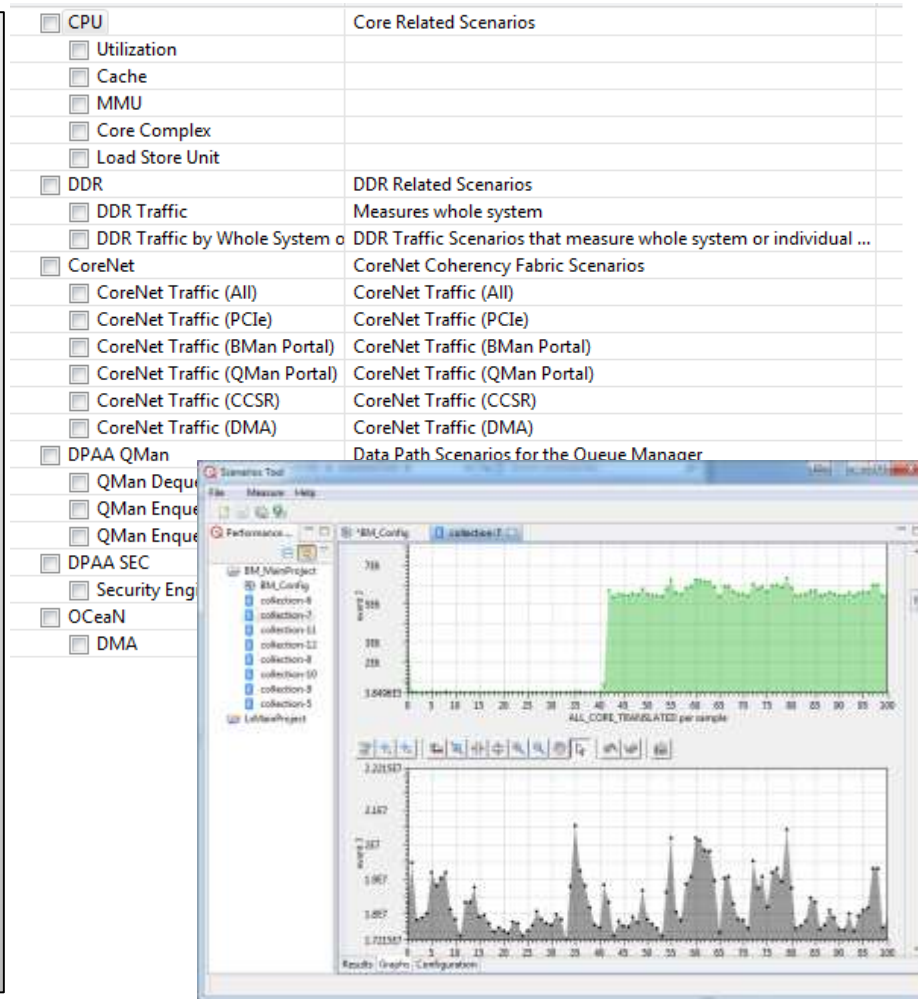
Optimization Suite – Scenarios Tool

- **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

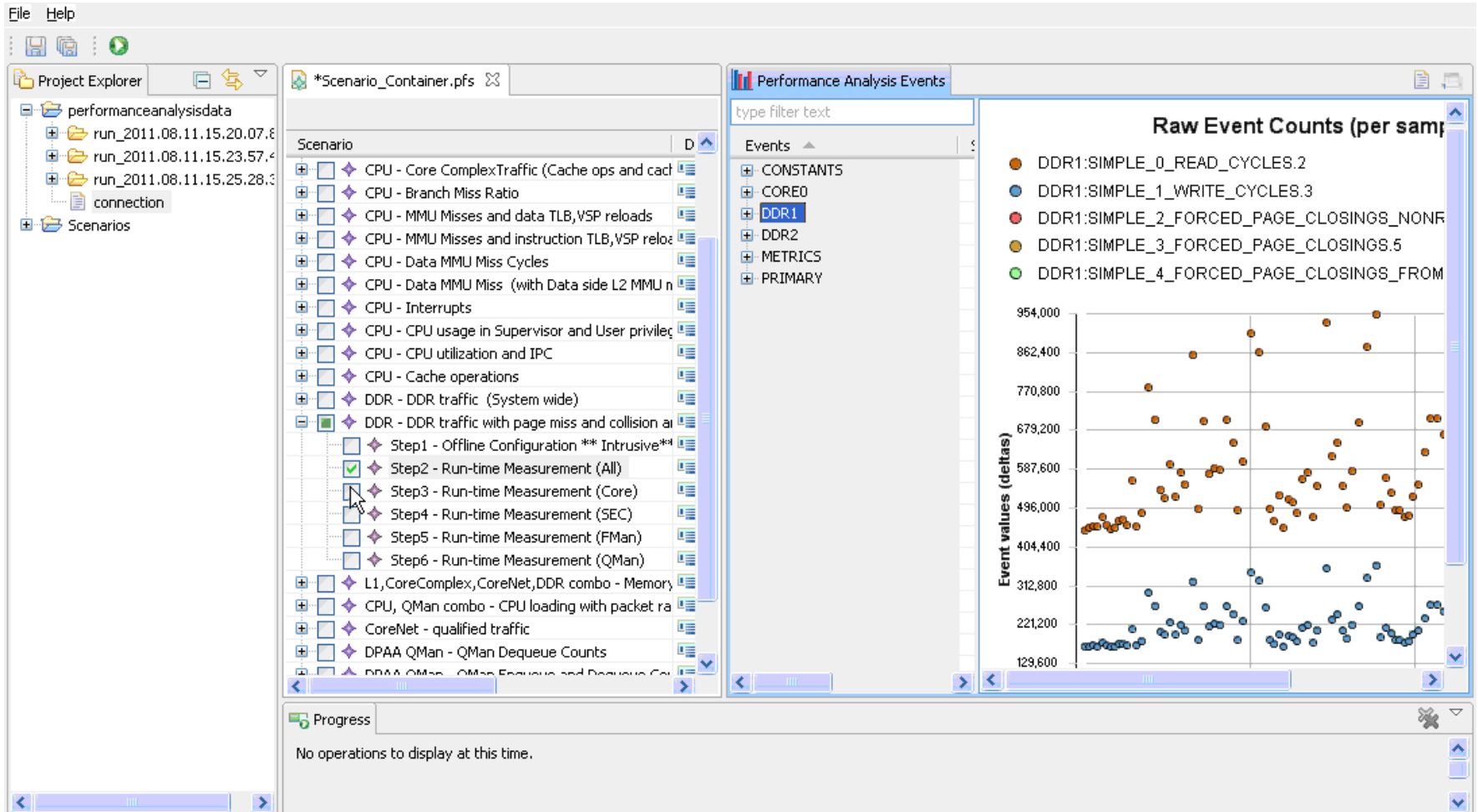


Optimized workflow for efficiently narrowing down performance issues anywhere on the system

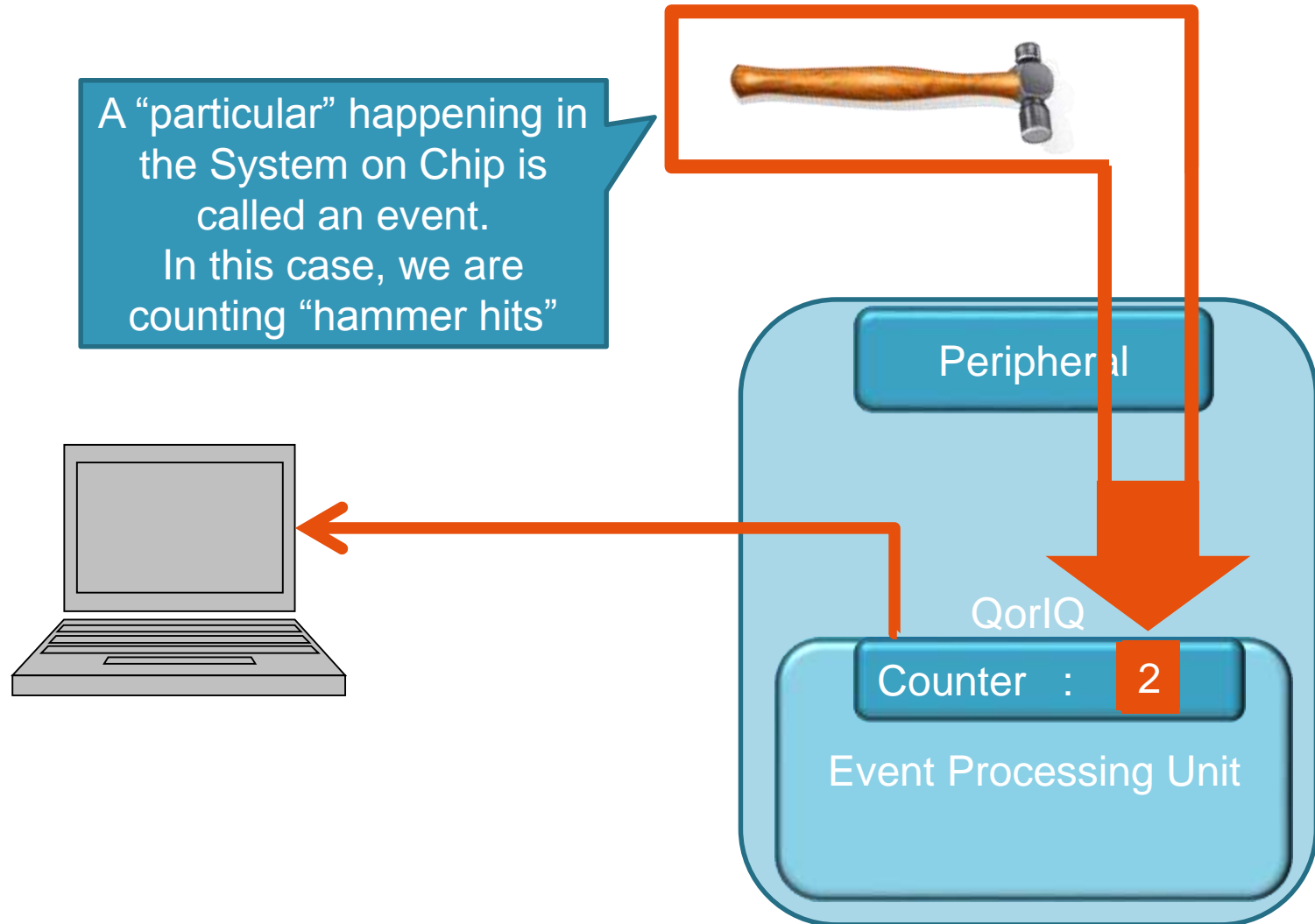


What Is Scenarios Tool?

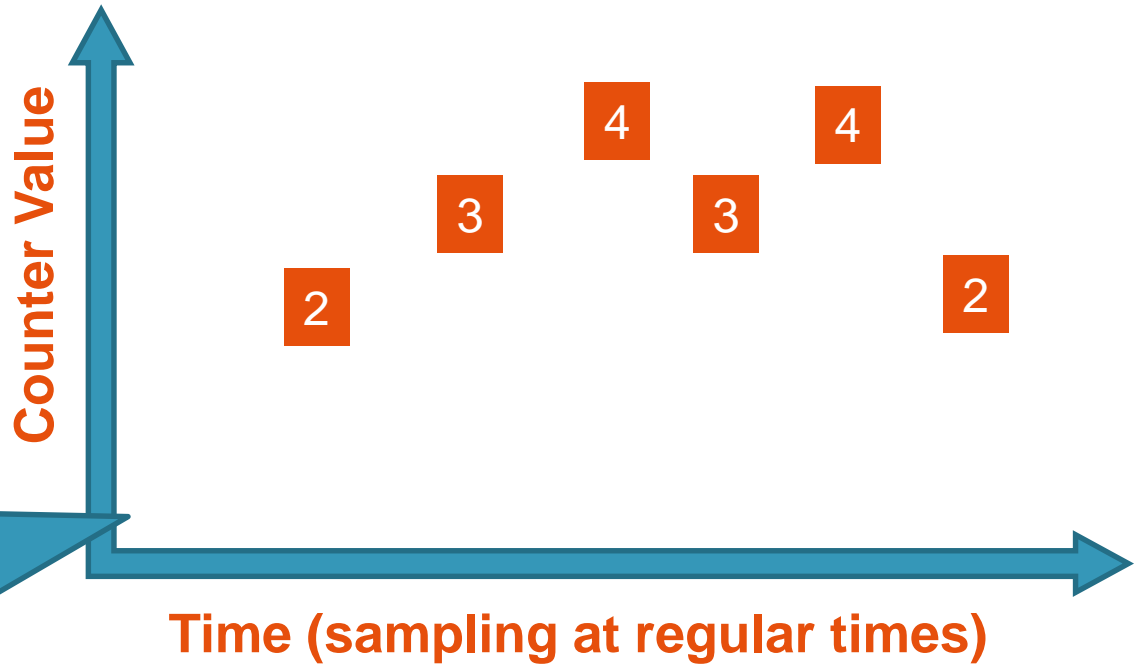
- Visually identify the system level problem areas in seconds.



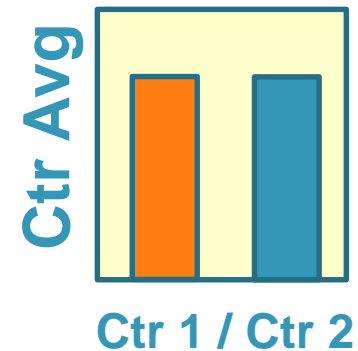
How "Counters" Work



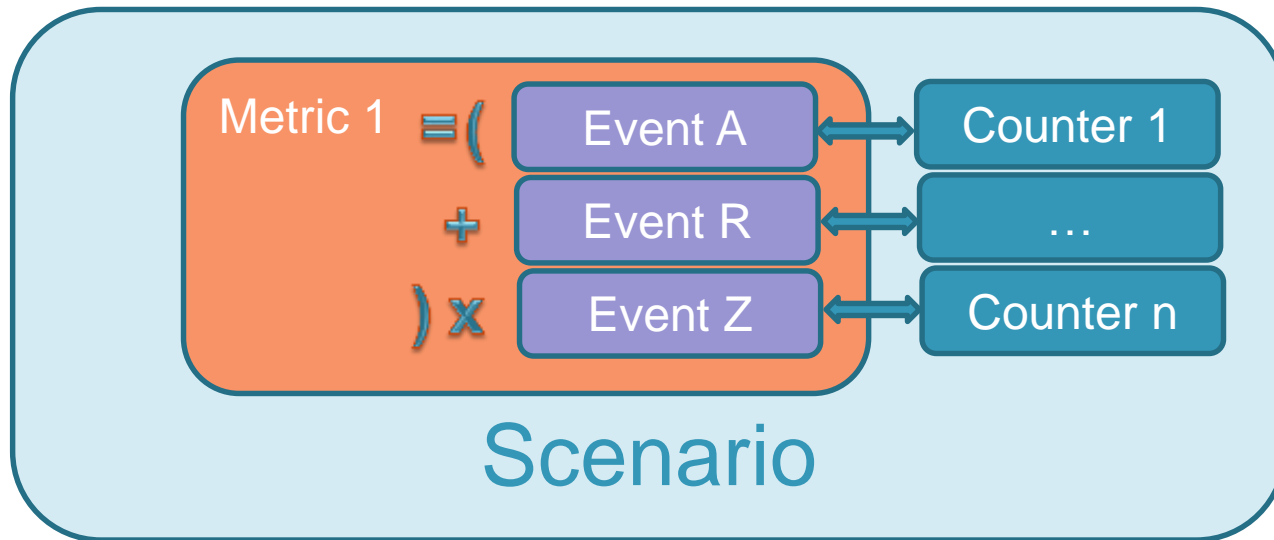
Making Sense of Counter Data



These counters could be counting:
Cache Misses
DDR Access
Or MORE!!!



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

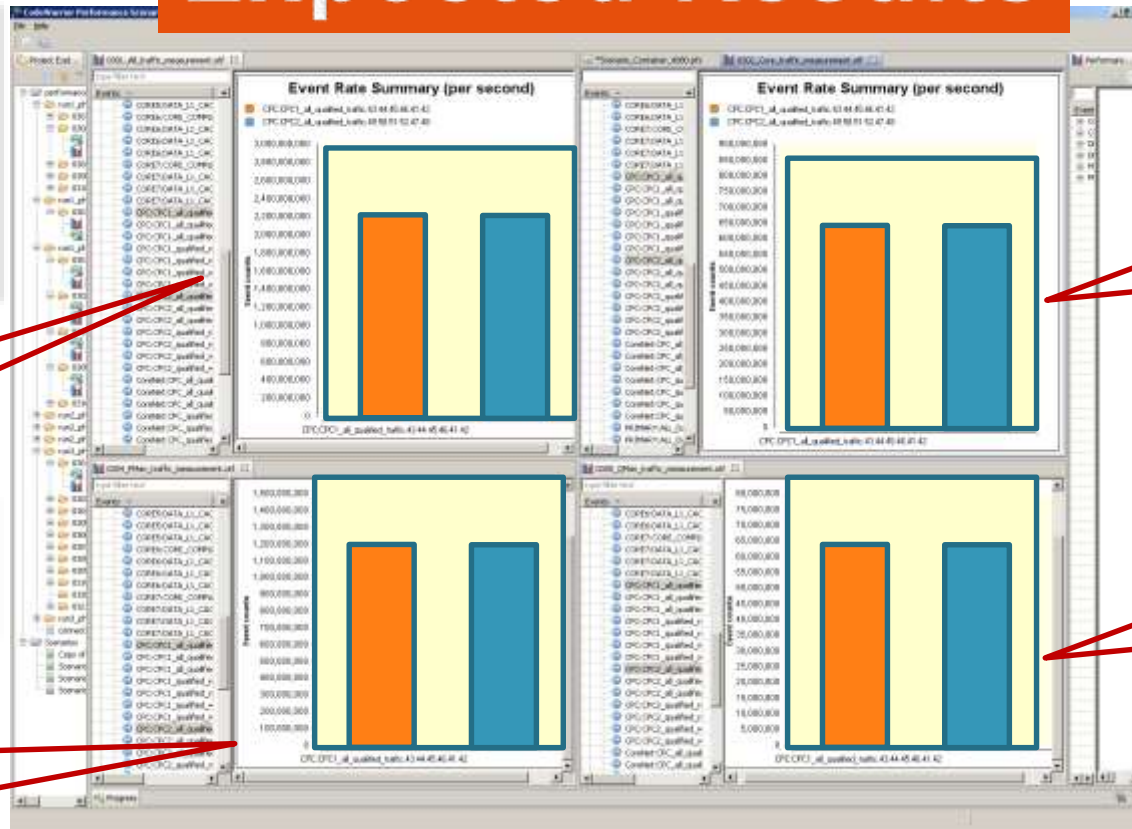
Total traffic to CPC1 and CPC2

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

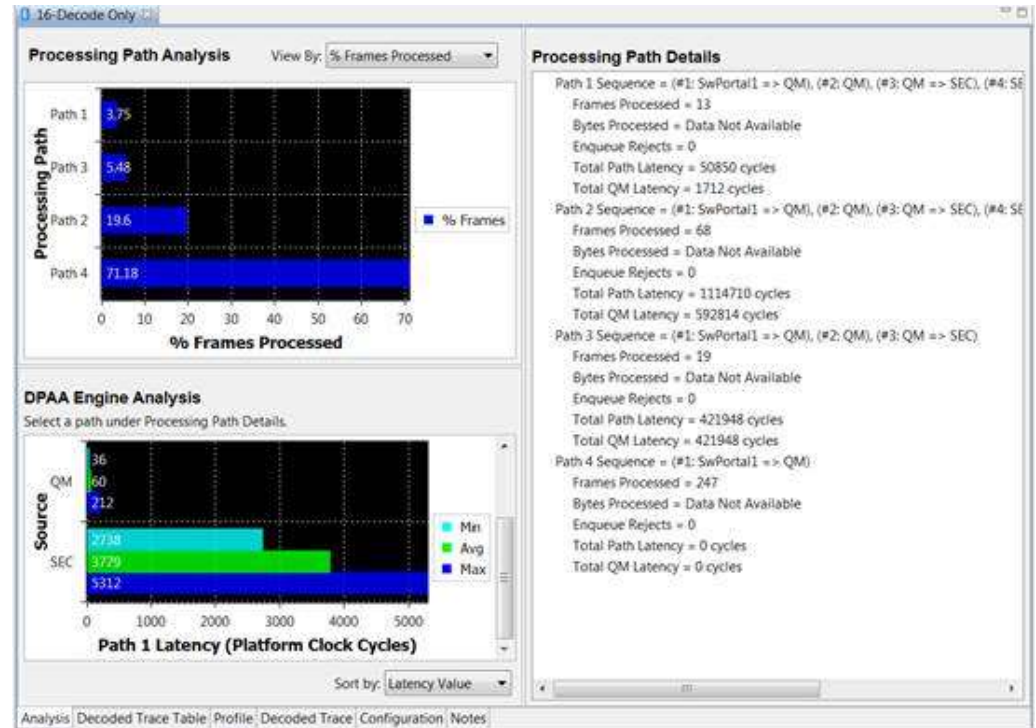
- Complexity abstraction and ease of use
- Enables key use cases:
 - Packet-Oriented System Level Performance Analysis
 - SoC Data Plane Configuration Debug
 - Packet Processing Latency Analysis
 - Packet Processing Critical Resource Monitoring

• 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

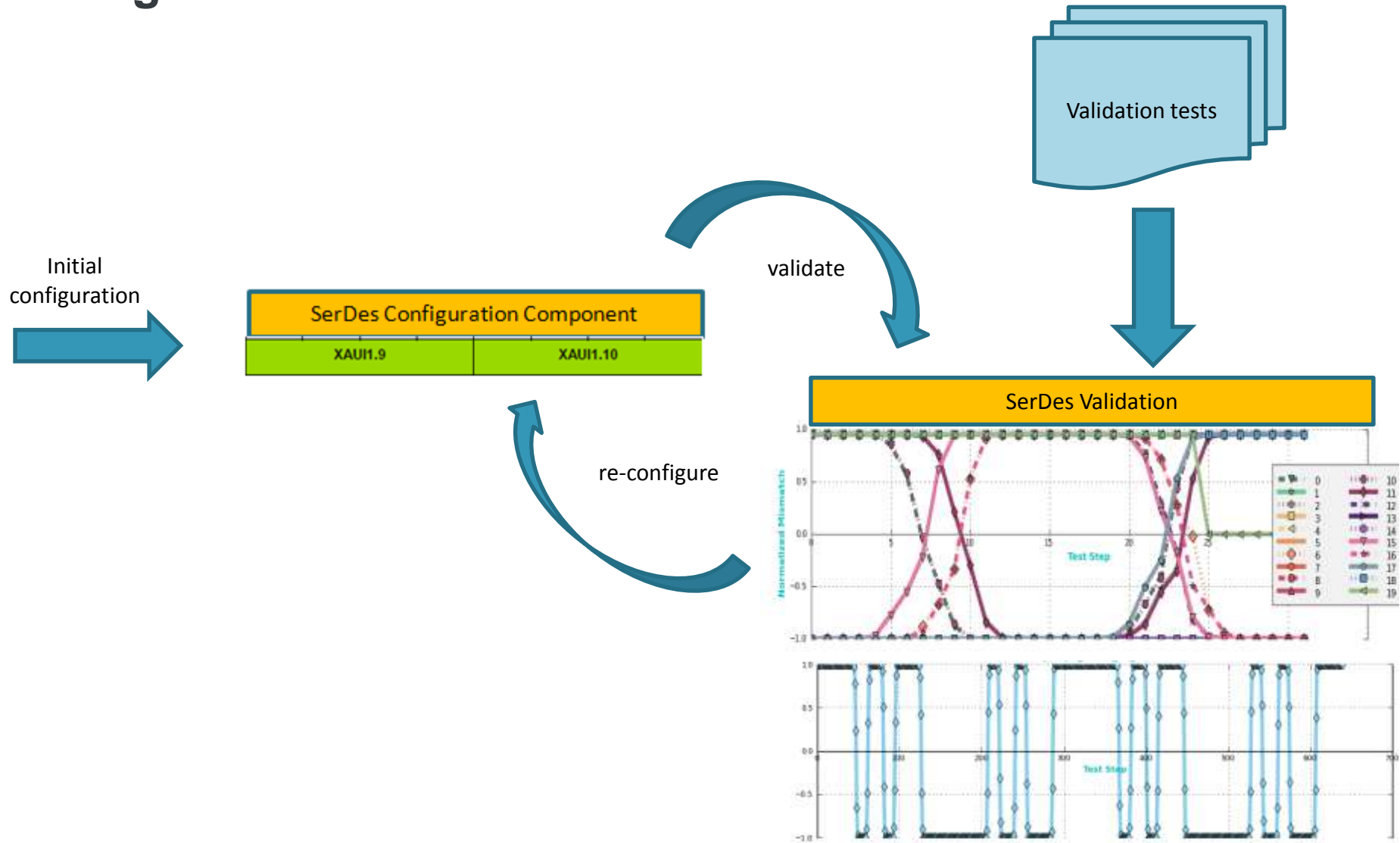
Solution to Solve these Problems

- 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

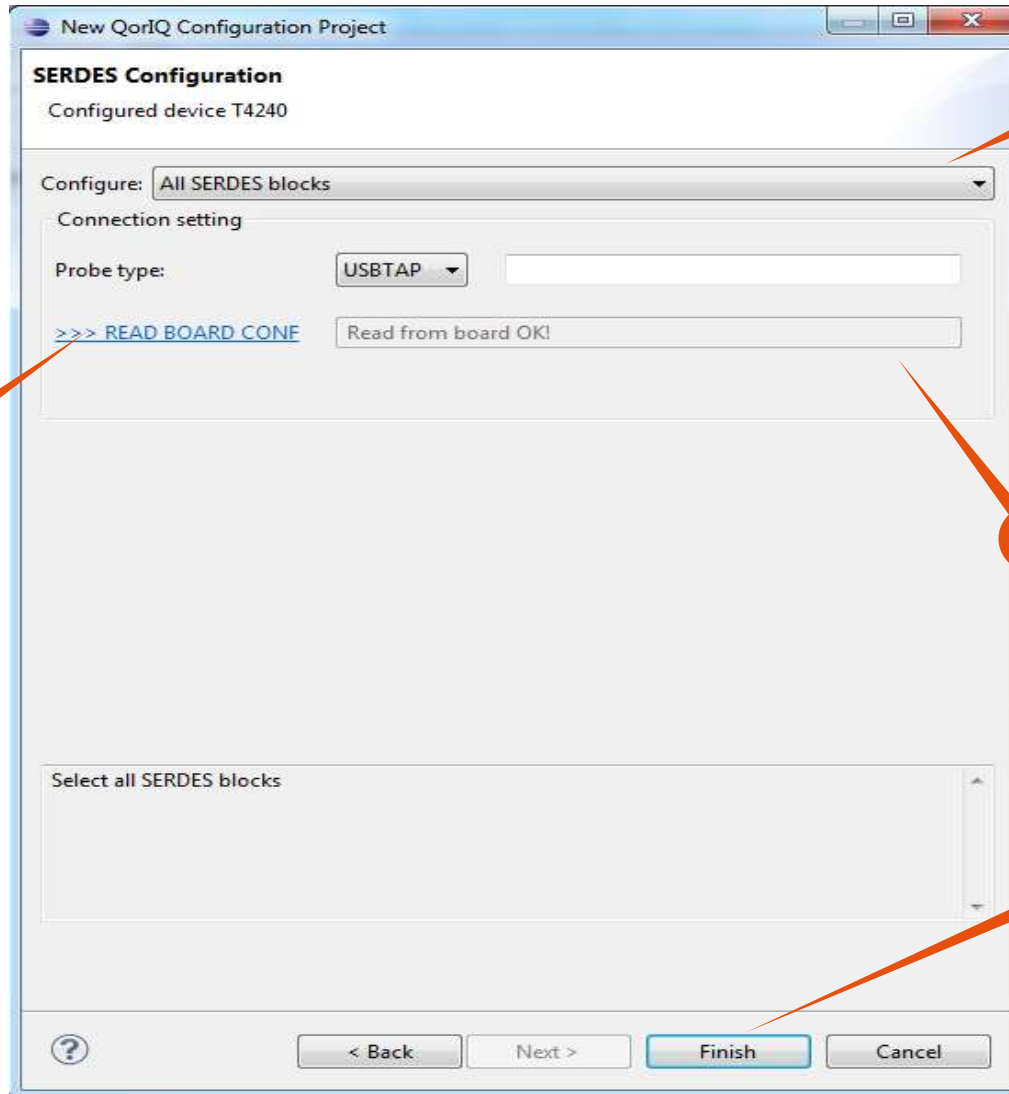
Objectives

- 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

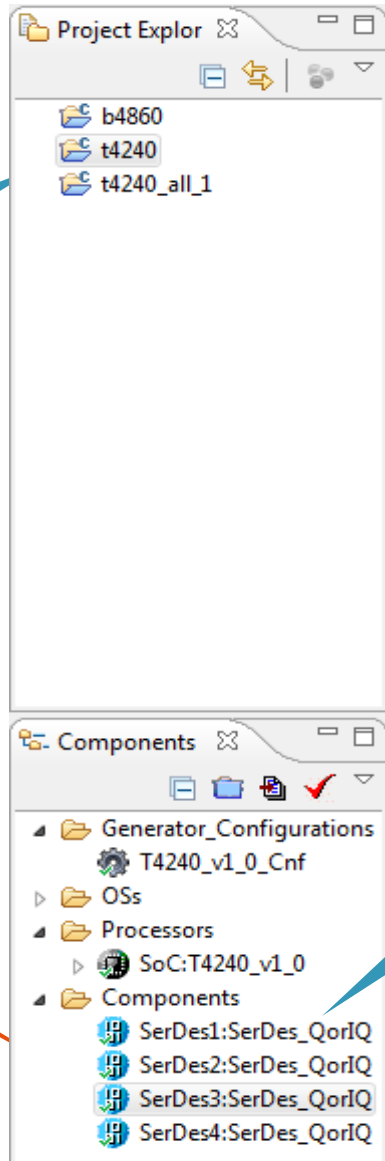
7

Status of reading action

8

Press finish to create SerDes component

Select SerDes Configuration and Validation Panel



Project was created

Select SerDes module to view

1

Components for each SerDes block were created

SerDes Configuration and Validation Panel

Components Library | Component Inspector - SerDes3

Basic | Advanced

SerDes Configuration and Validation

PLL	Lane 0		Lane 1		Lane 2		Lane 3		Lane 4		Lane 5		Lane 6		Lane 7	
	Tx	Rx	Tx	Rx	Tx	Rx	Tx	Rx	Tx	Rx	Tx	Rx	Tx	Rx	Tx	Rx
	PCIe 2.5	PCIe 2.5	PCIe 2.5	PCIe 2.5	PCIe 2.5	PCIe 2.5	PCIe 2.5	PCIe 2.5	SRIO 2.5	SRIO 2.5	SRIO 2.5	SRIO 2.5	SRIO 2.5	SRIO 2.5	SRIO 2.5	SRIO 2.5
PLL 1	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
PLL 2	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Lane 0 Configuration | Validation

Protocol: PCIe

Set as first lane

Transmitter

Output Ctrl: Enabled

Invert data

Equalization

Type: 2 Levels

PreCursor sign: Negative

PreCursor ratio: No equalization

PostCursor sign: Positive

PostCursor ratio: No Equalization

Adaptive equalization: 48

Amplitude reduction: 1.0

Receiver

Set as first lane

Rx Termination: HiZ or termination to xcorevss

Invert data

Electrical idle

Threshold: Vlow = 65mV Vhigh = 175mV

Enter idle filter: ~1 microsec Majority Filter

Exit idle filter: 80 UI Glitch Free Filter Data stopped

Equalization

Boost

Gain2

Source: Use rxeq adaption derived gaink2

Value: 0

Gain3

Source: Use rxeq adaption derived gaink3

Value: 0

Offset

Source: Use rxeq adaption derived eq_offset

Value: No imposed offset



SerDes Configuration and Validation Panel (continued)

Select Lane 1 configuration and validation view

2

PLL	Lane 0		Lane 1		Lane 2		Lane 3		Lane 4	
	Tx	Rx	Tx	Rx	Tx	Rx	Tx	Rx	Tx	Rx
	PCIe 2.5	PCIe 2.5	PCIe 2.5	PCIe 2.5	PCIe 2.5	PCIe 2.5	PCIe 2.5	PCIe 2.5	SRIO 2.5	SRIO 2.5
PLL 1	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
PLL 2	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

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)

The screenshot shows the 'Lane 1 Configuration' panel with the following settings:

- Protocol:** PCIe
- Transmitter:**
 - Output Ctrl: Enabled
 - Invert data:
 - Equalization Type: 2 Levels
 - PreCursor sign: Negative
 - PreCursor ratio: No equalization
 - PostCursor sign: Positive
 - PostCursor ratio: No Equalization
 - Adaptive equalization: 48
 - Amplitude reduction: 1.0
- Receiver:**
 - Set as first lane:
 - Fx Termination: HiZ or termination to xcorevss
 - Invert data:
 - Electrical idle:
 - Threshold: Vlow = 65mV Vhigh = 175mV
 - Enter idle filter: ~1 microsec Majority Filter
 - Exit idle filter: 80 UI Glitch Free Filter Data stopped
- Equalization:**
 - Boost:
 - Gain2:
 - Source: Use rxeq adaption derived gain2
 - Value: 0
 - Gain3:
 - Source: Use rxeq adaption derived gain3
 - Value: 0
 - Offset:
 - Source: Use rxeq adaption derived eq_offset
 - Value: No imposed offset

6

Select desired lane parameters

SerDes Configuration and Validation Panel (continued)

SerDes Configuration and Validation

PLL	Lane 0		Lane 1		Lane 2		Lane 3		Lane 4		
	Tx	Rx	Tx	Rx	Tx	Rx	Tx	Rx	Tx	Rx	
	PCIe 2.5	PCIe 2.5	PCIe 2.5	PCIe 2.5	PCIe 2.5	PCIe 2.5	PCIe 2.5	PCIe 2.5	PCIe 2.5	SRIO 2.5	SRIO 2.5
PLL 1	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
PLL 2	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Lane 0 Configuration Validation

Test	Result
<input checked="" type="checkbox"/> BIST	
<input type="checkbox"/> Jitter_scope	
<input type="checkbox"/> TX Pattern Generation	

Parameters

Pattern: MFTP Loopback: DIGITAL

Count Window: CW_1_05E_07bits

Results

Insert Error: 0 CDR Lock

Number of errors: -1 BIST Pattern Sync

Test result:

Apply configuration and start validation

7

List of tests

BIST testing parameters

SerDes Configuration and Validation Panel (continued)

Lane 0 Configuration Validation

Test	Result
<input checked="" type="checkbox"/> BIST	<input checked="" type="checkbox"/>
<input type="checkbox"/> Jitter_scope	
<input type="checkbox"/> TX Pattern Genera...	

Parameters

Pattern: MFTP Loopback: DIGITAL

Count Window: CW_1_05E_07bits

Results

Insert Error: 0 CDR Lock:

Number of errors: 0 BIST Pattern Sync:

Test result:

BIST TEST PASSED!

BIST test results

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 mmaped 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.

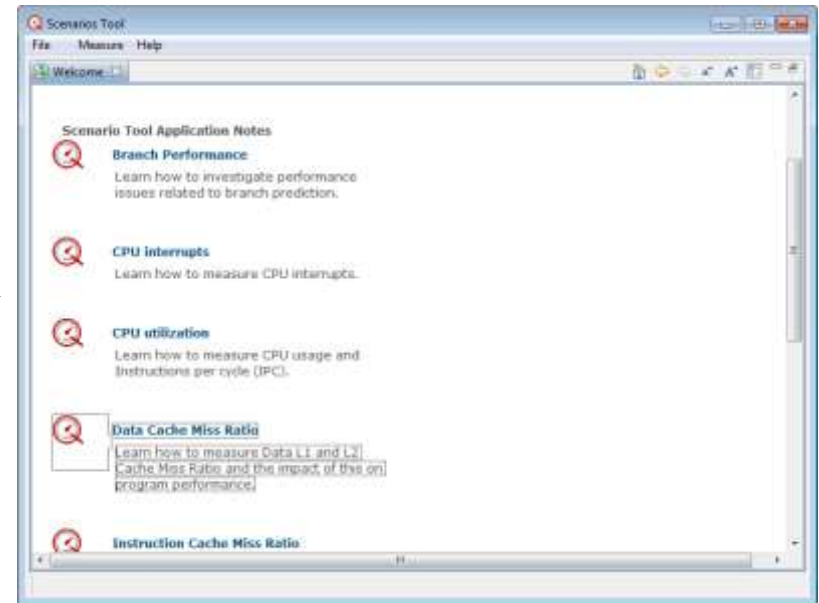
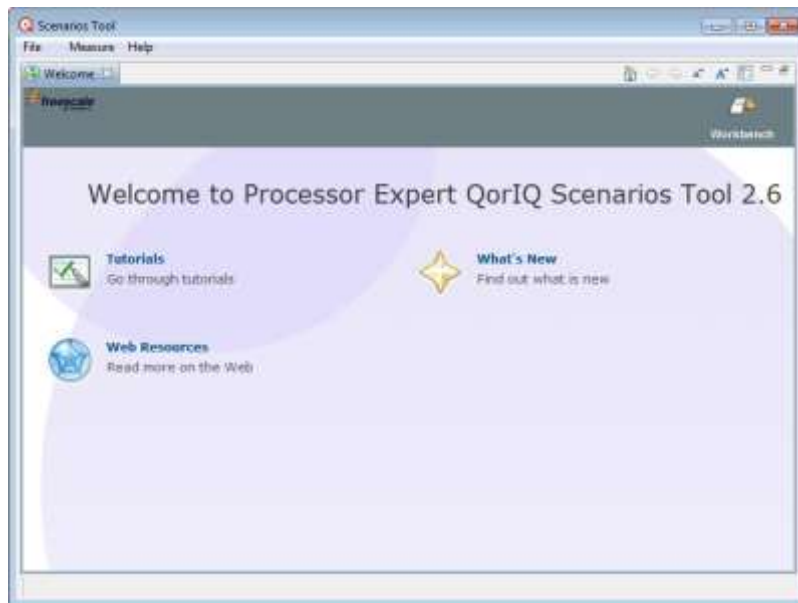
'sched': tracing/measuring of scheduler actions and latencies.

'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