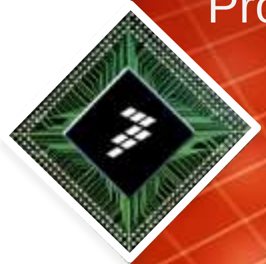
QorIQ Configuration Suite Tool Introduction

APF-ENT-T0579

Greg Hemstreet

Software Engineering Manager,
Processor Expert Team



April 2013

Promise

- **Before you leave today you will**
  - Understand why configuration tools will help you
  - Have a basic understanding of what will be available
  - Have undergone a basic walkthrough of the tools
  - Used actual configurations and modified them based on customer requests to configure:
    - RCW – pre-boot loader settings
    - DDR – memory controller settings
    - Device Trees – Linux® hardware device tree settings
    - Data Path graphs – configuring the DPAA
  - Know where to get more information

Agenda

- Explore why a configuration suite is important
- Describe how people get the tools
- Who is using QCS?
- Review each tool
- Pre-boot loader / RCW configuration
- DDR configuration
- Device Tree Editor
- Data Path graphs and configuring the DPAA
- Summary
- Where to get more information…
- Walkthrough Labs – backup slides
  - Lab1: Pre-boot loader / RCW Configuration
  - Lab2: DDR configuration
  - Lab3: Device Tree Editor
  - Lab4: Data Path graphs and configuring the DPAA

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 …

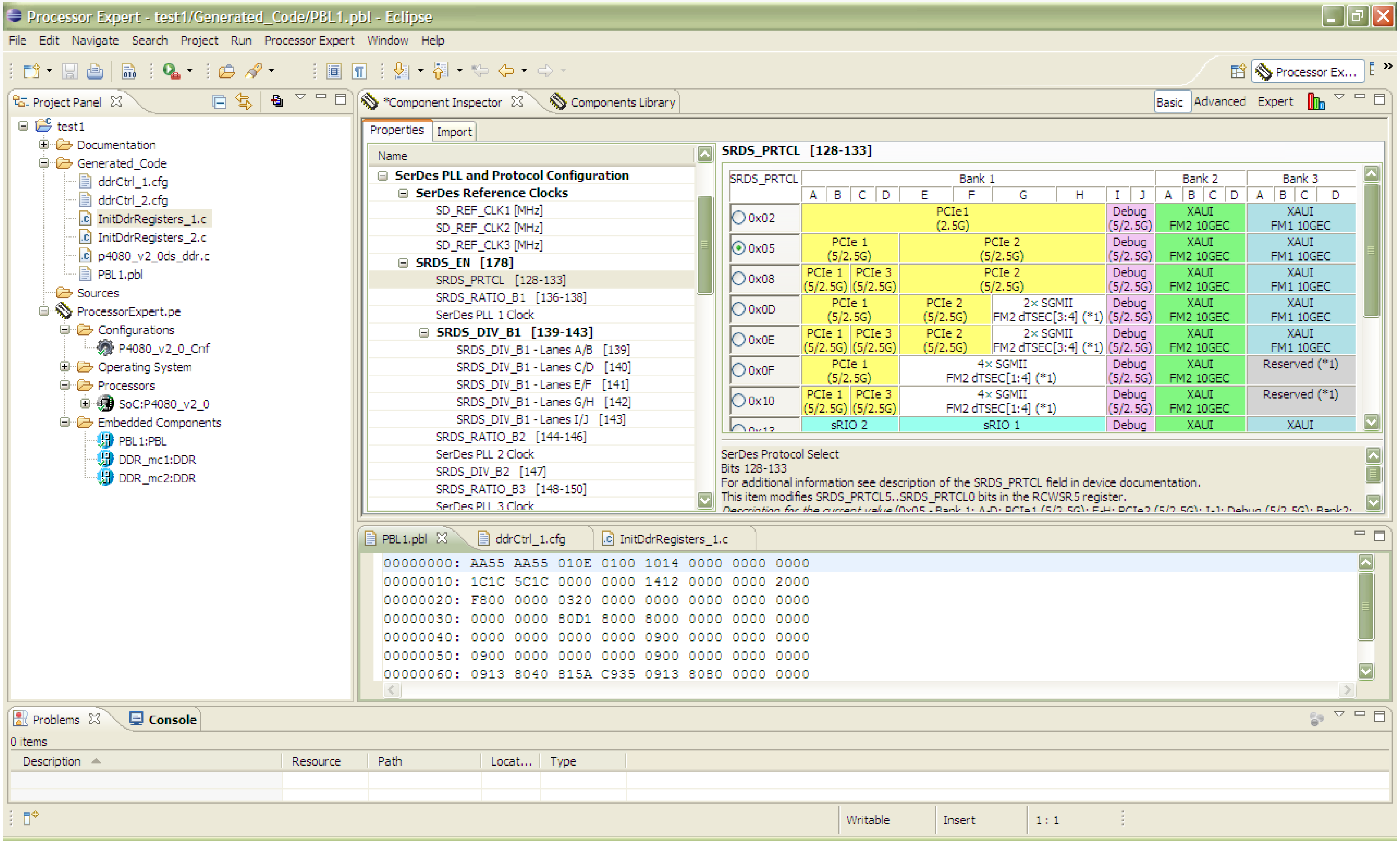
QorIQ Configuration Suite – Now Available!

- **QorIQ Configuration Suite v2.2 is NOW AVAILABLE!!!**
  - Supports all QorIQ and Qorivva devices
  - Works with Eclipse 3.5, Eclipse 3.6, Eclipse 3.7 development tools
    - Pure Java solution for maximum choice of host system support
    - Add-in to CodeWarrior Development Studio for PA, v10.1 or later
  - Available from www.freescale.com/QCS – FREE DOWNLOAD*
- **Includes the following four configuration tools all designed to collaborate on consistent configuration:**
  - PBL tool to define the Reset Control Word bit values and PBI data for the pre-boot
  - BOOTROM generator for those QorIQ without RCW functionality
  - DDR configuration supports setting the controller to a working state for any DDR
  - Data path graphical view helps to define data path configuration for the DPAA.
  - Hardware Device Tree editor supports references, synchronous GUI and XML editing, node validation based on specification bindings
  - Packaged as a separate product with installer and wizard functionality

* Must be a QorIQ customer or under QorIQ NDA for download permission

Actual URL is http://www.freescale.com/webapp/sps/site/prod_summary.jsp?code=PE_QORIQ_SUITE&tid=PEH

Processor Expert for QorIQ – Configuration Suite



Processor Expert - test1/Generated_Code/PBL1.pbl - Eclipse
File Edit Navigate Search Project Run Processor Expert Window Help
Processor Ex...
Project Panel
*Component Inspector
Components Library
Basic Advanced Expert
Properties Import

- test1
  - Documentation
  - Generated_Code
    - ddrCtrl_1.cfg
    - ddrCtrl_2.cfg
    - InitDdrRegisters_1.c
    - InitDdrRegisters_2.c
    - p4080_v2_0ds_ddr.c
    - PBL1.pbl
  - Sources
  - ProcessorExpert.pe
    - Configurations
      - P4080_v2_0_Cnf
    - Operating System
    - Processors
      - SoC:P4080_v2_0
    - Embedded Components
      - PBL1:PBL
      - DDR_mc1:DDR
      - DDR_mc2:DDR

Name

- **SerDes PLL and Protocol Configuration**
  - **SerDes Reference Clocks**
    - SD_REF_CLK1 [MHz]
    - SD_REF_CLK2 [MHz]
    - SD_REF_CLK3 [MHz]
  - **SRDS_EN [178]**
    - SRDS_PRTCL [128-133]
    - SRDS_RATIO_B1 [136-138]
    - SerDes PLL 1 Clock
    - **SRDS_DIV_B1 [139-143]**
      - SRDS_DIV_B1 - Lanes A/B [139]
      - SRDS_DIV_B1 - Lanes C/D [140]
      - SRDS_DIV_B1 - Lanes E/F [141]
      - SRDS_DIV_B1 - Lanes G/H [142]
      - SRDS_DIV_B1 - Lanes I/J [143]
    - SRDS_RATIO_B2 [144-146]
    - SerDes PLL 2 Clock
    - SRDS_DIV_B2 [147]
    - SRDS_RATIO_B3 [148-150]
    - SerDes PLL 3 Clock

SRDS_PRTCL [128-133]

SRDS_PRTCL	Bank 1										Bank 2				Bank 3			
	A	B	C	D	E	F	G	H	I	J	A	B	C	D	A	B	C	D
○ 0x02	PCIe1 (2.5G)								Debug (5/2.5G)		XAUI FM2 10GEC				XAUI FM1 10GEC			
● 0x05	PCIe 1 (5/2.5G)				PCIe 2 (5/2.5G)				Debug (5/2.5G)		XAUI FM2 10GEC				XAUI FM1 10GEC			
○ 0x08	PCIe 1 (5/2.5G)		PCIe 3 (5/2.5G)		PCIe 2 (5/2.5G)				Debug (5/2.5G)		XAUI FM2 10GEC				XAUI FM1 10GEC			
○ 0x0D	PCIe 1 (5/2.5G)				PCIe 2 (5/2.5G)		2× SGMII FM2 dTSEC[3:4] (*1)		Debug (5/2.5G)		XAUI FM2 10GEC				XAUI FM1 10GEC			
○ 0x0E	PCIe 1 (5/2.5G)		PCIe 3 (5/2.5G)		PCIe 2 (5/2.5G)		2× SGMII FM2 dTSEC[3:4] (*1)		Debug (5/2.5G)		XAUI FM2 10GEC				XAUI FM1 10GEC			
○ 0x0F	PCIe 1 (5/2.5G)				4× SGMII FM2 dTSEC[1:4] (*1)				Debug (5/2.5G)		XAUI FM2 10GEC				Reserved (*1)			
○ 0x10	PCIe 1 (5/2.5G)		PCIe 3 (5/2.5G)		4× SGMII FM2 dTSEC[1:4] (*1)				Debug (5/2.5G)		XAUI FM2 10GEC				Reserved (*1)			
○ 0x13	sRIO 2				sRIO 1				Debug		XAUI				XAUI			

SerDes Protocol Select
Bits 128-133
For additional information see description of the SRDS_PRTCL field in device documentation.
This item modifies SRDS_PRTCL5..SRDS_PRTCL0 bits in the RCWSR5 register.
Description for the current value (0x05 - Bank 1: A-D: PCIe1 (5/2.5G); E-H: PCIe2 (5/2.5G); I-J: Debug (5/2.5G); Bank2:

```
00000000: AA55 AA55 010E 0100 1014 0000 0000 0000
00000010: 1C1C 5C1C 0000 0000 1412 0000 0000 2000
00000020: F800 0000 0320 0000 0000 0000 0000 0000
00000030: 0000 0000 80D1 8000 8000 0000 0000 0000
00000040: 0000 0000 0000 0000 0900 0000 0000 0000
00000050: 0900 0000 0000 0000 0900 0000 0000 0000
00000060: 0913 8040 815A C935 0913 8080 0000 0000
```

0 items

Description	Resource	Path	Locat...	Type

What's Different About QorIQ Processor Expert?

- **MCUs use Processor Expert to generate source code that is code-size optimized and only includes the minimal functions and operations to support initialization and peripheral drivers**
  - Previously, Processor Expert was included in CodeWarrior only
  - Now, Processor Expert plug-ins can be installed into any Eclipse

- **Processors uses Processor Expert to generate configuration files used in the creation of a bootstrap typically to either Linux or another OS.**
  - Installs as an Eclipse update package (under 20MB)
  - Supports configuration complexity without altering OS / Application software

Installing Processor Expert for QorIQ

- **You need either CodeWarrior for PA 10.1 or later**

  OR, you download an Eclipse version for free

  OR, you use an existing Eclipse workbench you have installed (Wind River, QNX, GNU, etc.)

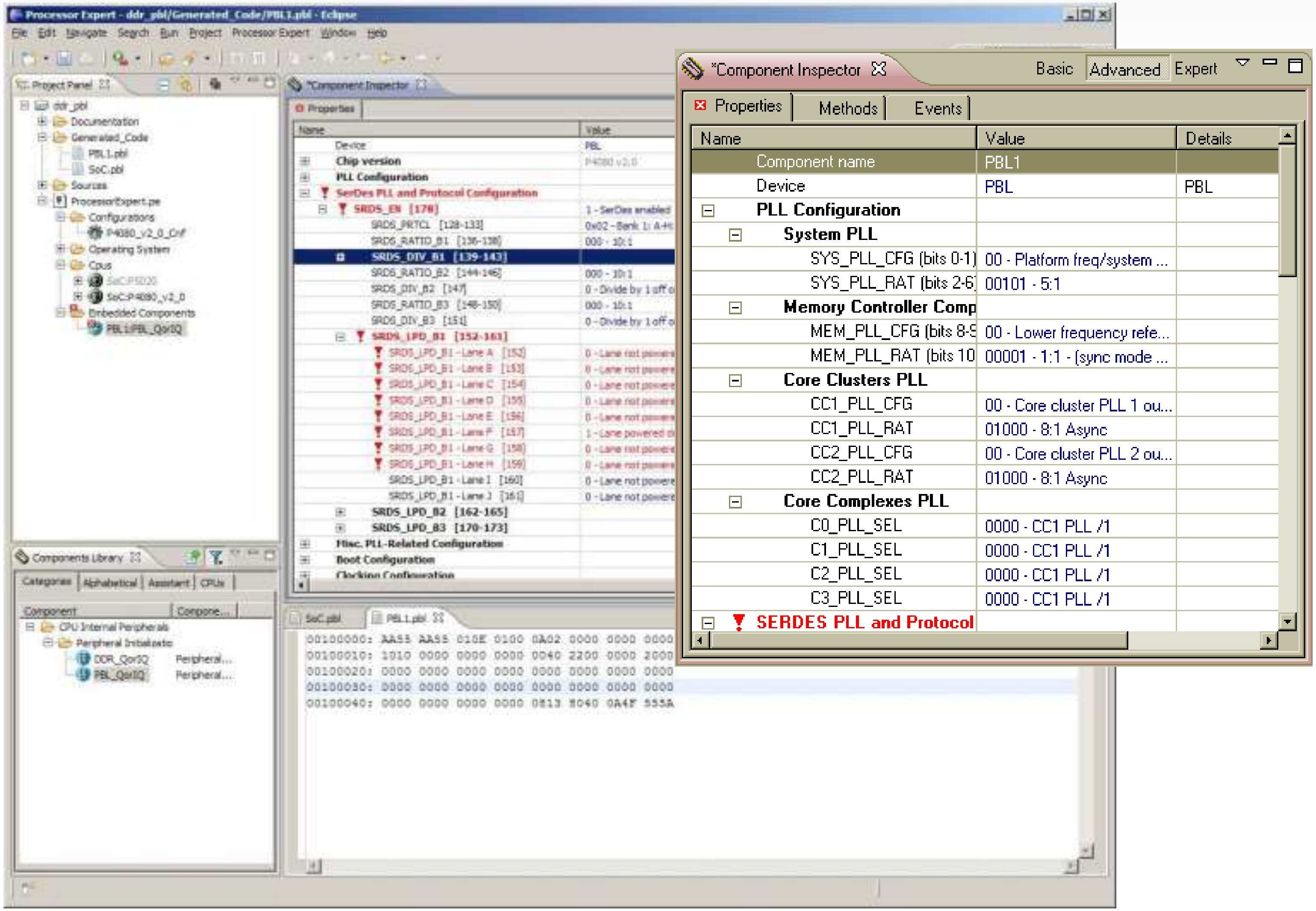
- **Processor Expert for QorIQ Configuration Suite installs using the Eclipse updater's "Add new software…" capability**

- **The Configuration Suite is 100% pure Java so it should run on any Eclipse 3.5.1 or later host environment (Windows, Linux, Solaris, Mac OS, 32-bit/64-bit, …)**

Pre-boot Loader

RCW Configuration Tool

Pre-boot Loader (RCW) Configuration



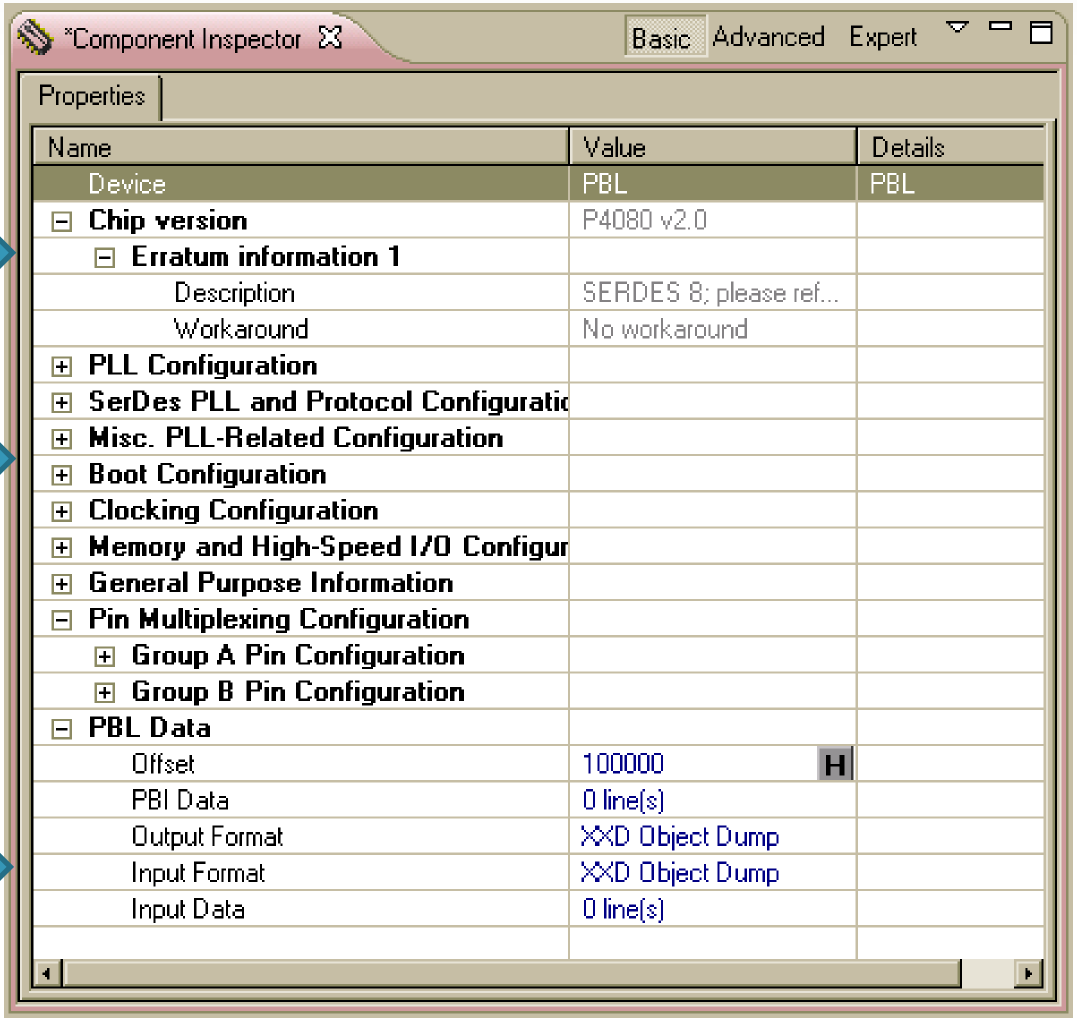
Name	Value
Device	PBL
Chip version	P4080 v2.0
PLL Configuration	
SerDes PLL and Protocol Configuration	
SRDS_EN [178]	1 - SerDes enabled
SRDS_PRTCL [128-133]	0x02 - Bank 1: A-H
SRDS_RATIO_B1 [136-138]	000 - 10:1
SRDS_DIV_B1 [139-143]	
SRDS_RATIO_B2 [144-146]	000 - 10:1
SRDS_DIV_B2 [147]	0 - Divide by 1 off o
SRDS_RATIO_B3 [148-150]	000 - 10:1
SRDS_DIV_B3 [151]	0 - Divide by 1 off o
SRDS_LPD_B1 [152-161]	
SRDS_LPD_B1 - Lane A [152]	0 - Lane not powere
SRDS_LPD_B1 - Lane B [153]	0 - Lane not powere
SRDS_LPD_B1 - Lane C [154]	0 - Lane not powere
SRDS_LPD_B1 - Lane D [155]	0 - Lane not powere
SRDS_LPD_B1 - Lane E [156]	0 - Lane not powere
SRDS_LPD_B1 - Lane F [157]	1 - Lane powered d
SRDS_LPD_B1 - Lane G [158]	0 - Lane not powere
SRDS_LPD_B1 - Lane H [159]	0 - Lane not powere
SRDS_LPD_B1 - Lane I [160]	0 - Lane not powere
SRDS_LPD_B1 - Lane J [161]	0 - Lane not powere
SRDS_LPD_B2 [162-165]	
SRDS_LPD_B3 [170-173]	
Misc. PLL-Related Configuration	
Boot Configuration	
Clocking Configuration	

```
00100000: AA55 AA55 010E 0100 0A02 0000 0000 0000
00100010: 1010 0000 0000 0000 0040 2200 0000 2000
00100020: 0000 0000 0000 0000 0000 0000 0000 0000
00100030: 0000 0000 0000 0000 0000 0000 0000 0000
00100040: 0000 0000 0000 0000 0813 8040 0A4F 555A
```

Name	Value	Details
Component name	PBL1	
Device	PBL	PBL
PLL Configuration		
System PLL		
SYS_PLL_CFG (bits 0-1)	00 - Platform freq/system ...	
SYS_PLL_RAT (bits 2-6)	00101 - 5:1	
Memory Controller Comp		
MEM_PLL_CFG (bits 8-9	00 - Lower frequency refe...	
MEM_PLL_RAT (bits 10	00001 - 1:1 - (sync mode ...	
Core Clusters PLL		
CC1_PLL_CFG	00 - Core cluster PLL 1 ou...	
CC1_PLL_RAT	01000 - 8:1 Async	
CC2_PLL_CFG	00 - Core cluster PLL 2 ou...	
CC2_PLL_RAT	01000 - 8:1 Async	
Core Complexes PLL		
C0_PLL_SEL	0000 - CC1 PLL /1	
C1_PLL_SEL	0000 - CC1 PLL /1	
C2_PLL_SEL	0000 - CC1 PLL /1	
C3_PLL_SEL	0000 - CC1 PLL /1	
SERDES PLL and Protocol		

Pre-boot Loader standard component interface

- Chip version and errata information
- Settings of RCW fields
- Input/Output format selection
- Possibility to add PBI data
- Possibility to import RCW settings



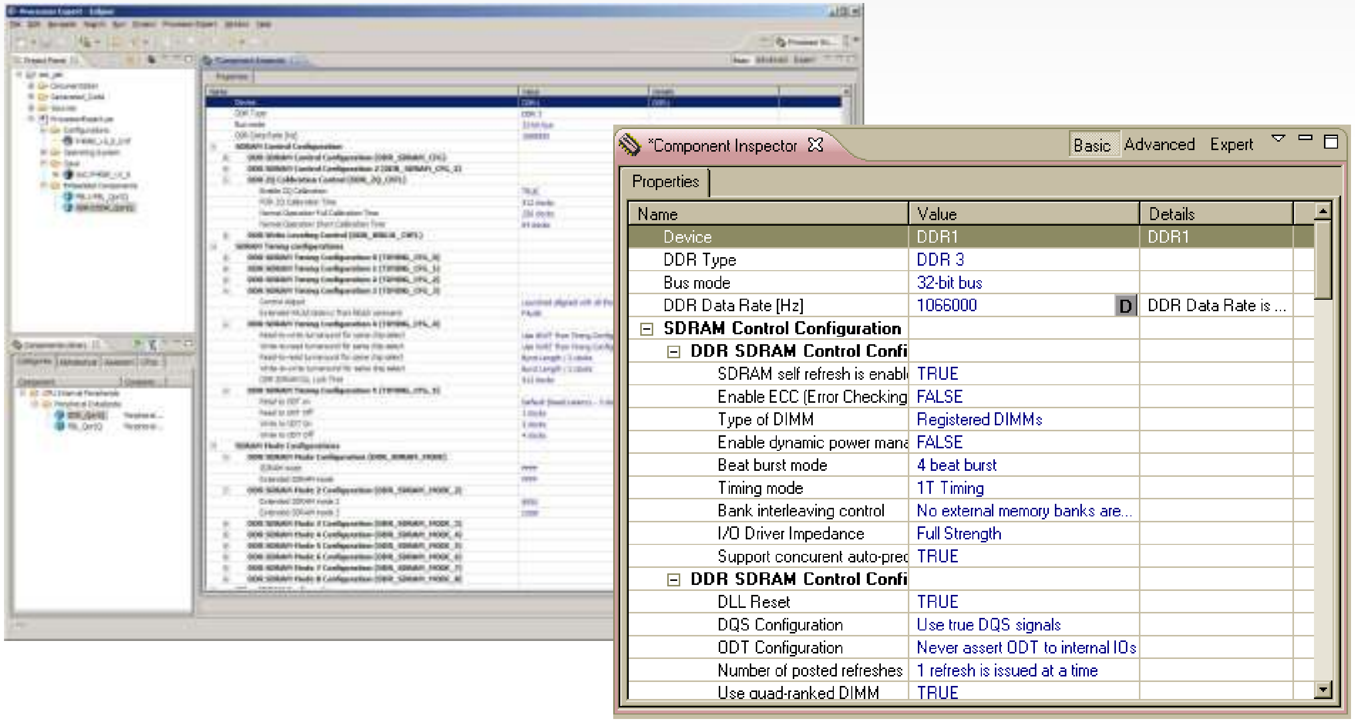
Name	Value	Details
Device	PBL	PBL
⊟ Chip version	P4080 v2.0	
⊟ Erratum information 1		
Description	SERDES 8; please ref...	
Workaround	No workaround	
⊞ PLL Configuration		
⊞ SerDes PLL and Protocol Configuratic		
⊞ Misc. PLL-Related Configuration		
⊞ Boot Configuration		
⊞ Clocking Configuration		
⊞ Memory and High-Speed I/O Configur		
⊞ General Purpose Information		
⊟ Pin Multiplexing Configuration		
⊞ Group A Pin Configuration		
⊞ Group B Pin Configuration		
⊟ PBL Data		
Offset	100000 H	
PBI Data	0 line(s)	
Output Format	XXD Object Dump	
Input Format	XXD Object Dump	
Input Data	0 line(s)	

freescale™ 飞思卡尔

DDR Configuration

DDR Configuration Tool

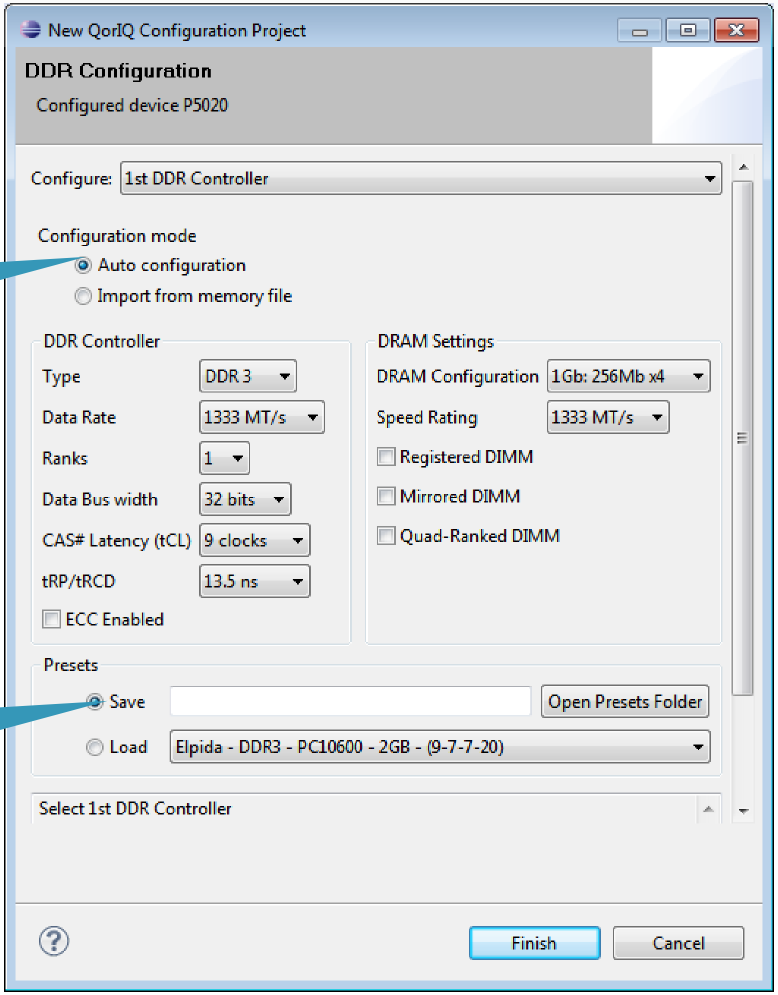
DDR Configuration



Name	Value	Details
Device	DDR1	DDR1
DDR Type	DDR 3	
Bus mode	32-bit bus	
DDR Data Rate [Hz]	1066000 D	DDR Data Rate is ...
⊟ **SDRAM Control Configuration**		
⊟ **DDR SDRAM Control Confi**		
SDRAM self refresh is enabl	TRUE	
Enable ECC (Error Checking	FALSE	
Type of DIMM	Registered DIMMs	
Enable dynamic power mana	FALSE	
Beat burst mode	4 beat burst	
Timing mode	1T Timing	
Bank interleaving control	No external memory banks are...	
I/O Driver Impedance	Full Strength	
Support concurent auto-prec	TRUE	
⊟ **DDR SDRAM Control Confi**		
DLL Reset	TRUE	
DQS Configuration	Use true DQS signals	
ODT Configuration	Never assert ODT to internal IOs	
Number of posted refreshes	1 refresh is issued at a time	
Use quad-ranked DIMM	TRUE	

- Eclipse-based GUI tool which performs configuration of DDR for QorIQ devices. The configurable parameters are consistent with the JEDEC standard and also with vendor-specific information
- GUI configuration is validated for consistency and meaningful errors/warnings are displayed
- Tool’s output is: C file containing memory registers values, CodeWarrior TCL initialization file, uBoot initialization file
- It is integrated into the configuration suite for QorIQ devices

DDR Wizard: Basic Configuration Mode – Custom Configuration



New QorIQ Configuration Project

DDR Configuration

Configured device P5020

Configure: 1st DDR Controller

Configuration mode

● Auto configuration

○ Import from memory file

DDR Controller

- Type: DDR 3
- Data Rate: 1333 MT/s
- Ranks: 1
- Data Bus width: 32 bits
- CAS# Latency (tCL): 9 clocks
- tRP/tRCD: 13.5 ns
- □ ECC Enabled

DRAM Settings

- DRAM Configuration: 1Gb: 256Mb x4
- Speed Rating: 1333 MT/s
- □ Registered DIMM
- □ Mirrored DIMM
- □ Quad-Ranked DIMM

Presets

● Save

○ Load: Elpida - DDR3 - PC10600 - 2GB - (9-7-7-20)

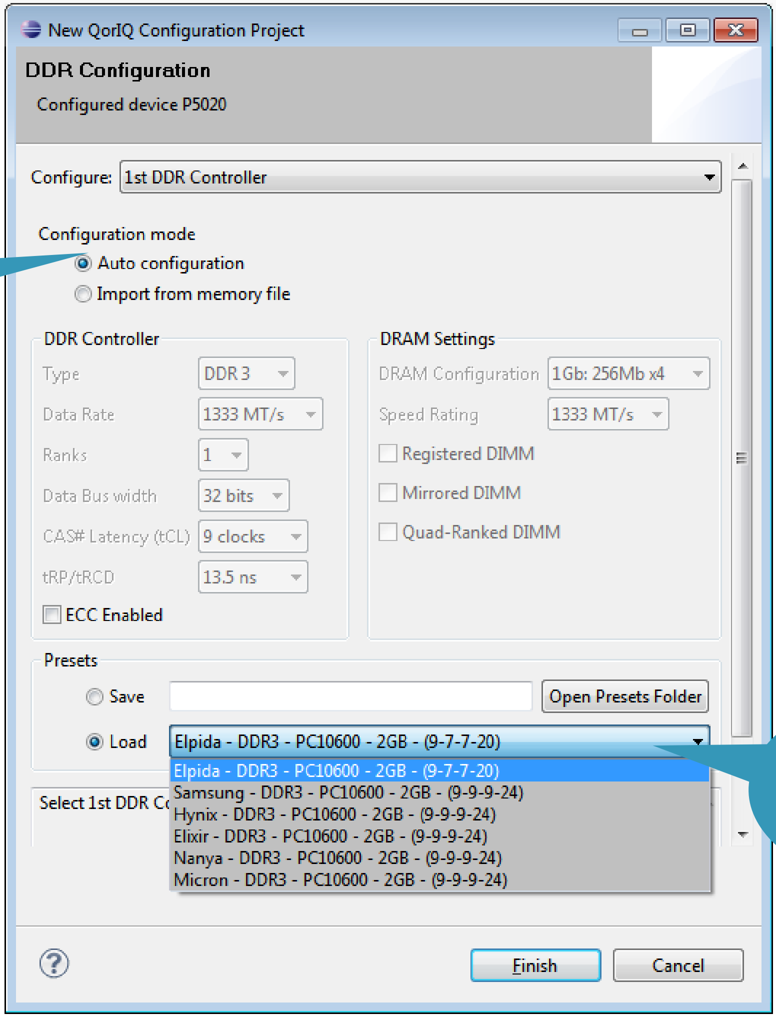
Select 1st DDR Controller

Auto configuration mode

Save configuration as a reusable preset

DDR Wizard: Basic Configuration Mode – Presets

Auto configuration mode



New QorIQ Configuration Project

DDR Configuration

Configured device P5020

Configure: 1st DDR Controller

Configuration mode

● Auto configuration

○ Import from memory file

DDR Controller

- Type: DDR 3
- Data Rate: 1333 MT/s
- Ranks: 1
- Data Bus width: 32 bits
- CAS# Latency (tCL): 9 clocks
- tRP/tRCD: 13.5 ns
- □ ECC Enabled

DRAM Settings

- DRAM Configuration: 1Gb: 256Mb x4
- Speed Rating: 1333 MT/s
- □ Registered DIMM
- □ Mirrored DIMM
- □ Quad-Ranked DIMM

Presets

○ Save

● Load: Elpida - DDR3 - PC10600 - 2GB - (9-7-7-20)

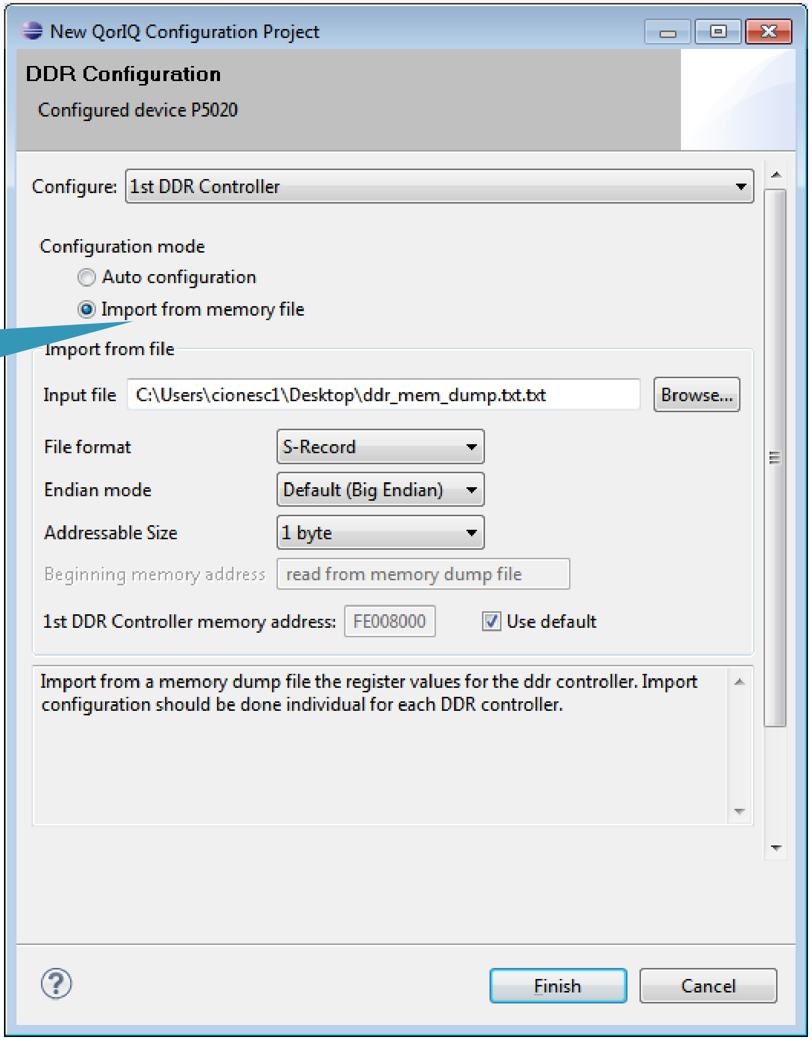
- Elpida - DDR3 - PC10600 - 2GB - (9-7-7-20)
- Samsung - DDR3 - PC10600 - 2GB - (9-9-9-24)
- Hynix - DDR3 - PC10600 - 2GB - (9-9-9-24)
- Elixir - DDR3 - PC10600 - 2GB - (9-9-9-24)
- Nanya - DDR3 - PC10600 - 2GB - (9-9-9-24)
- Micron - DDR3 - PC10600 - 2GB - (9-9-9-24)

Select 1st DDR C

Load a preset

DDR Wizard: Import Memory Dump Mode

Import from memory dump (various formats)



DDR Configuration

Configured device P5020

Configure: 1st DDR Controller

Configuration mode

○ Auto configuration

● Import from memory file

Import from file

Input file: C:\Users\cionesc1\Desktop\ddr_mem_dump.txt.txt

File format: S-Record

Endian mode: Default (Big Endian)

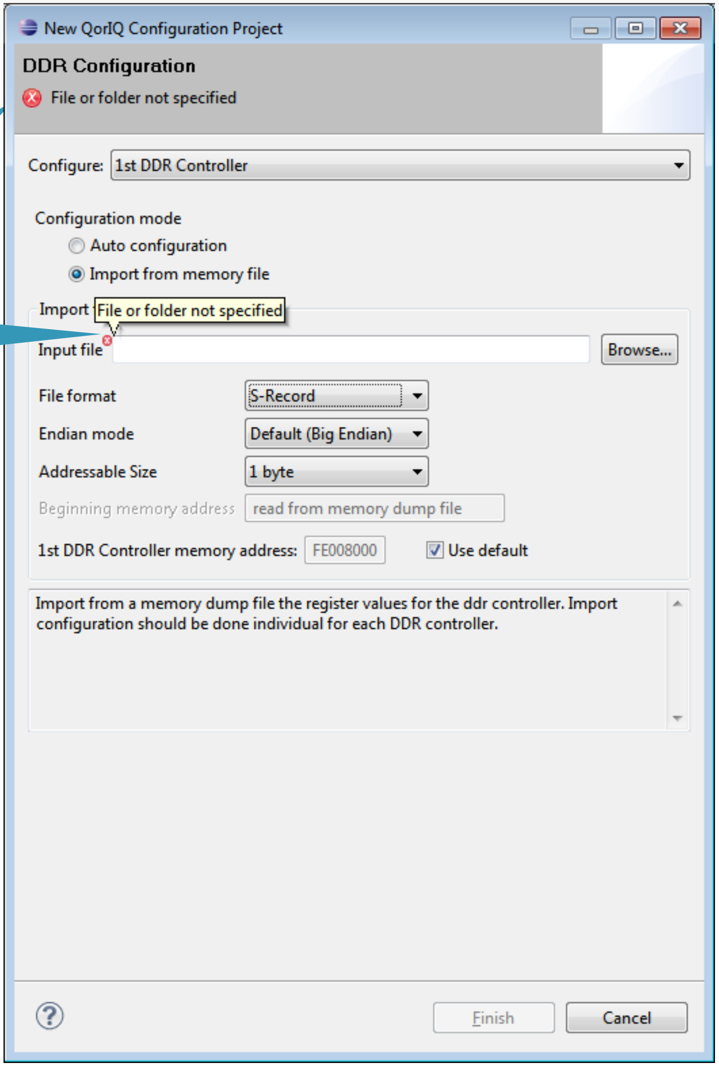
Addressable Size: 1 byte

Beginning memory address: read from memory dump file

1st DDR Controller memory address: FE008000 ☑ Use default

Import from a memory dump file the register values for the ddr controller. Import configuration should be done individual for each DDR controller.

DDR Wizard: Import Memory Dump Mode



New QorIQ Configuration Project

DDR Configuration

☒ File or folder not specified

Configure: 1st DDR Controller

Configuration mode

○ Auto configuration

● Import from memory file

Import: File or folder not specified

Input file:

File format: S-Record

Endian mode: Default (Big Endian)

Addressable Size: 1 byte

Beginning memory address: read from memory dump file

1st DDR Controller memory address: FE008000 ☑ Use default

Import from a memory dump file the register values for the ddr controller. Import configuration should be done individual for each DDR controller.

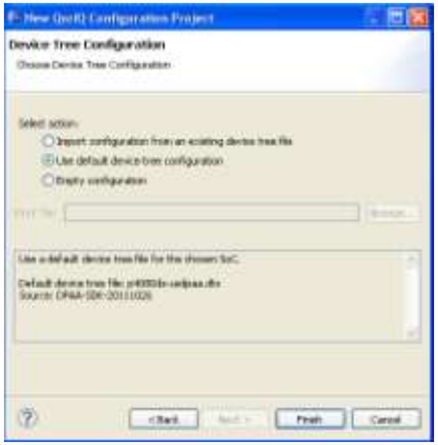
Error notifications

Device Tree Editor

Hardware Device Tree Tool

Hardware Device Tree Workflow

Create Project



New QorIQ Configuration Project

Device Tree Configuration

Choose Device Tree Configuration

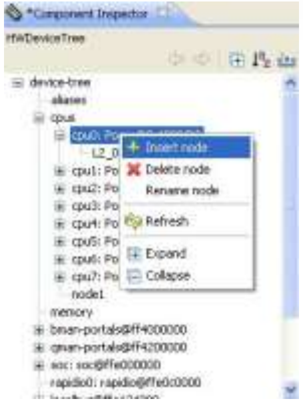
Select action:

○ Import configuration from an existing device tree file
● Use default device tree configuration
○ Empty configuration

Use a default device tree file for the chosen SoC.

Default device tree file: p4080ds-usdpaa.dts
Source: DPAA-SDK-20111026

Configure Component

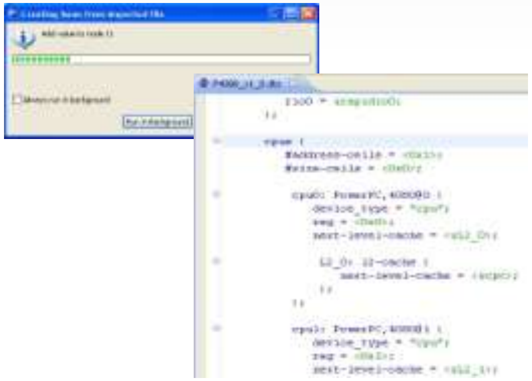


*Component Inspector

HWDeviceTree

```
device-tree
    aliases
    cpus
        cpu0: Po
            L2_0
        cpu1: Po
        cpu2: Po
        cpu3: Po
        cpu4: Po
        cpu5: Po
        cpu6: Po
        cpu7: Po
        node1
    memory
    bman-portals@ff4000000
    qman-portals@ff4200000
    soc: soc@ffe000000
    rapidio0: rapidio@ffe0c0000
```

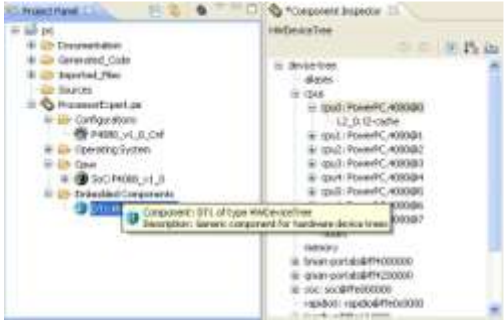
Generate Code



Creating beans from inspected file

```
            I2C0 = &ump_i2c0;
        };

        cpus {
            #address-cells = <0x1>;
            #size-cells = <0x0>;

            cpu0: PowerPC,4080@0 {
                device_type = "cpu";
                reg = <0x0>;
                next-level-cache = <&L2_0>;

                L2_0: l2-cache {
                    next-level-cache = <&cpc>;
                };
            };

            cpu1: PowerPC,4080@1 {
                device_type = "cpu";
                reg = <0x1>;
                next-level-cache = <&L2_1>;
```

Select Component

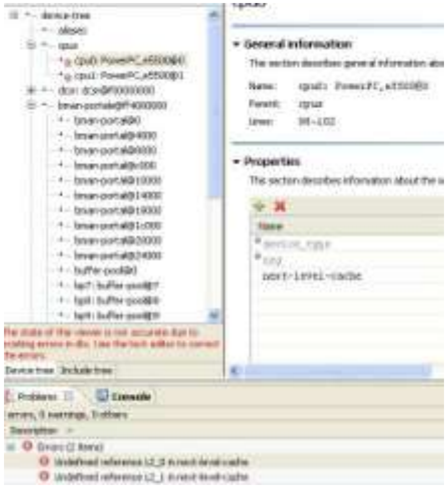


```
p4
    Documentation
    Generated_Code
    Imported_Files
    Sources
    ProcessorExpert.pe
        Configurations
            P4080_v1_0_Cnf
        Operating System
        Cpus
            SoC P4080_v1_0
        Embedded Components
            DT1
```

Component: DT1 of type HWDeviceTree
Description: Generic component for hardware device trees

```
device-tree
    aliases
    cpus
        cpu0: PowerPC,4080@0
            L2_0: l2-cache
        cpu1: PowerPC,4080@1
        cpu2: PowerPC,4080@2
        cpu3: PowerPC,4080@3
        cpu4: PowerPC,4080@4
        cpu5: PowerPC,4080@5
        cpu6: PowerPC,4080@6
        cpu7: PowerPC,4080@7
    memory
    bman-portals@ff4000000
    qman-portals@ff4200000
    soc: soc@ffe000000
    rapidio0: rapidio@ffe0c0000
```

Validate Component



```
device-tree
    aliases
    cpus
        cpu0: PowerPC,e5500@0
        cpu1: PowerPC,e5500@1
    dcsr: dcsr@f00000000
    bman-portals@ff4000000
        bman-portal@0
        bman-portal@4000
        bman-portal@8000
        bman-portal@c000
        bman-portal@10000
        bman-portal@14000
        bman-portal@18000
        bman-portal@1c000
        bman-portal@20000
        bman-portal@24000
        buffer-pool@0
        bp7: buffer-pool@7
        bp8: buffer-pool@8
        bp9: buffer-pool@9
```

The state of the viewer is not accurate due to existing errors in dts. Use the text editor to correct the errors.

General information

The section describes general information abo

Name: cpu0: PowerPC,e5500@0
Parent: cpus
Lines: 98-102

Properties

The section describes information about the s

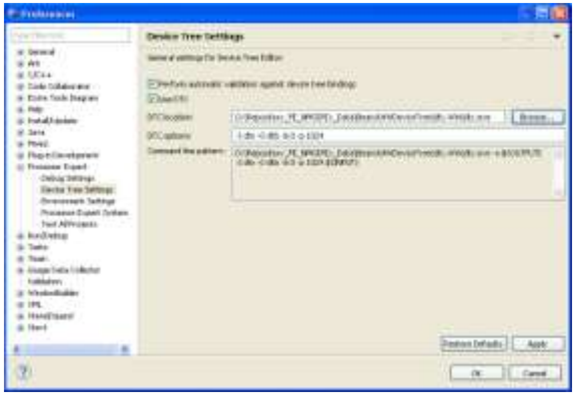
errors, 0 warnings, 0 others

Description

Errors (2 items)
Undefined reference L2_0 in next-level-cache
Undefined reference L2_1 in next-level-cache

Compile DTS

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



Preferences

Device Tree Settings

General settings for Device Tree Editor

☐ Perform automatic validation against device tree bindings
☑ Use DTC

DTC location: C:\Repository_PE_MASTER_Data\Beans\HWDeviceTree\dtc\Win\dtc.exe

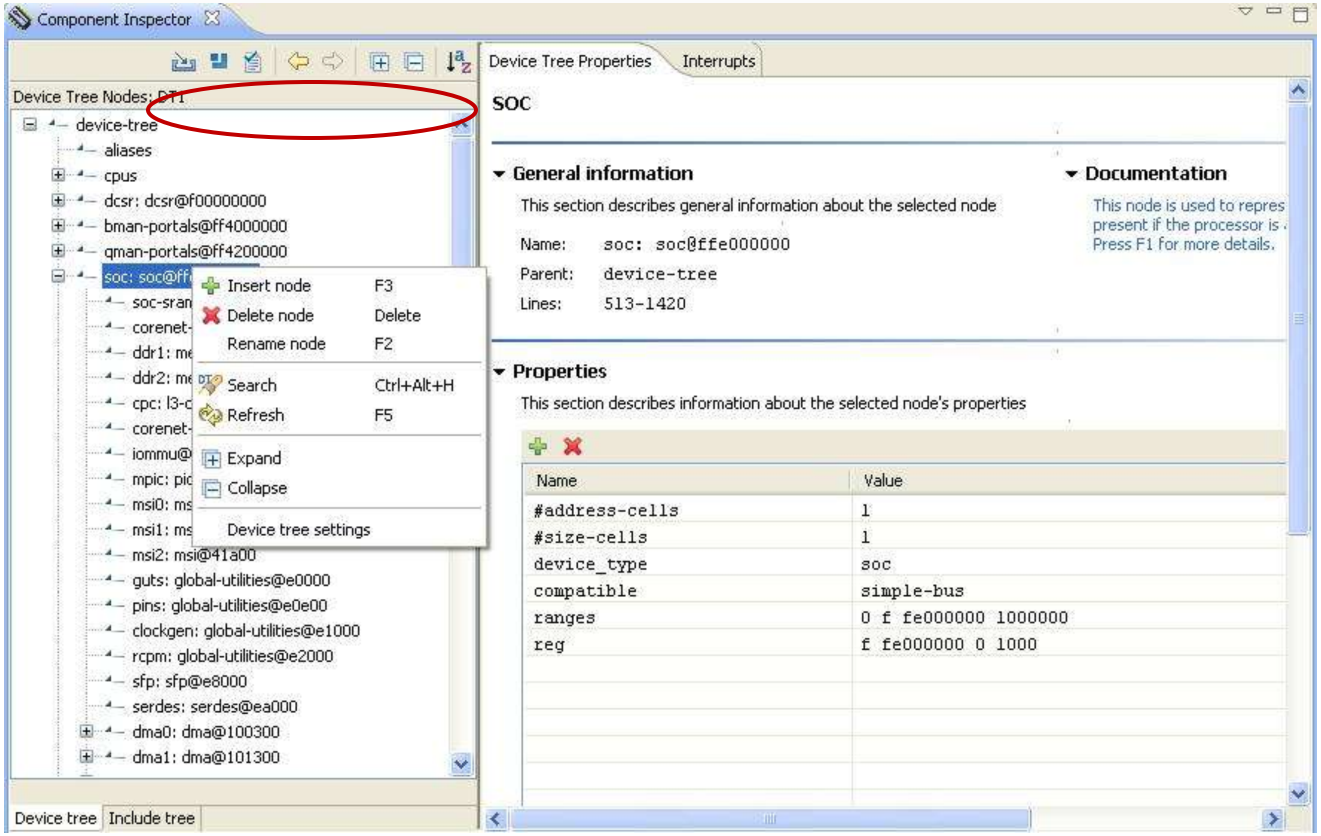
DTC options: -I dts -O dtb -b 0 -p 1024

Command line pattern: C:\Repository_PE_MASTER_Data\Beans\HWDeviceTree\dtc\Win\dtc.exe -o $(OUTPUT) -I dts -O dtb -b 0 -p 1024 $(INPUT)

freescale™
飞思卡尔

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



This section describes general information about the selected node

Name: soc: soc@ffe000000

Parent: device-tree

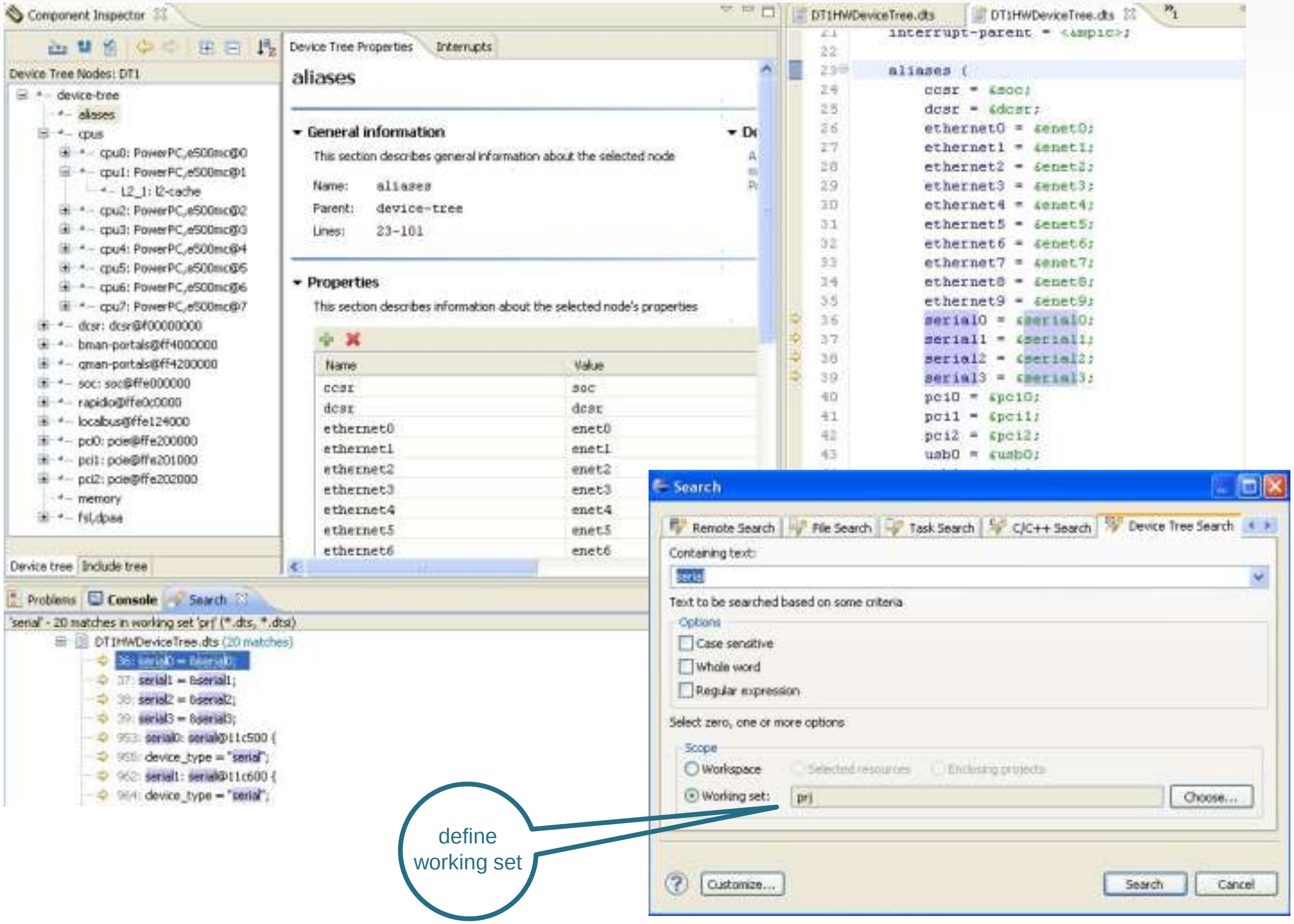
Lines: 513-1420

This node is used to repres present if the processor is Press F1 for more details.

This section describes information about the selected node's properties

Name	Value
#address-cells	1
#size-cells	1
device_type	soc
compatible	simple-bus
ranges	0 f fe000000 1000000
reg	f fe000000 0 1000

Search Capability



Device Tree Nodes: DT1

- device-tree
  - aliases
  - cpus
    - cpu0: PowerPC,e500mc@0
    - cpu1: PowerPC,e500mc@1
      - L2_1: l2-cache
    - cpu2: PowerPC,e500mc@2
    - cpu3: PowerPC,e500mc@3
    - cpu4: PowerPC,e500mc@4
    - cpu5: PowerPC,e500mc@5
    - cpu6: PowerPC,e500mc@6
    - cpu7: PowerPC,e500mc@7
  - dcsr: dcsr@f00000000
  - bman-portals@ff4000000
  - qman-portals@ff4200000
  - soc: soc@ffe000000
  - rapidio@ffe0c0000
  - localbus@ffe124000
  - pci0: pcie@ffe200000
  - pci1: pcie@ffe201000
  - pci2: pcie@ffe202000
  - memory
  - fsl,dpaa

aliases

General information

This section describes general information about the selected node

Name: aliases

Parent: device-tree

Lines: 23-101

Properties

This section describes information about the selected node's properties

Name	Value
ccsr	soc
dcsr	dcsr
ethernet0	enet0
ethernet1	enet1
ethernet2	enet2
ethernet3	enet3
ethernet4	enet4
ethernet5	enet5
ethernet6	enet6

```
    interrupt-parent = <&mpic>;

    aliases {
        ccsr = &soc;
        dcsr = &dcsr;
        ethernet0 = &enet0;
        ethernet1 = &enet1;
        ethernet2 = &enet2;
        ethernet3 = &enet3;
        ethernet4 = &enet4;
        ethernet5 = &enet5;
        ethernet6 = &enet6;
        ethernet7 = &enet7;
        ethernet8 = &enet8;
        ethernet9 = &enet9;
        serial0 = &serial0;
        serial1 = &serial1;
        serial2 = &serial2;
        serial3 = &serial3;
        pci0 = &pci0;
        pci1 = &pci1;
        pci2 = &pci2;
        usb0 = &usb0;
```

'serial' - 20 matches in working set 'prj' (*.dts, *.dtsi)

- DT1HWDeviceTree.dts (20 matches)
  - 36: serial0 = &serial0;
  - 37: serial1 = &serial1;
  - 38: serial2 = &serial2;
  - 39: serial3 = &serial3;
  - 953: serial0: serial@11c500 {
  - 958: device_type = "serial";
  - 962: serial1: serial@11c600 {
  - 964: device_type = "serial";

Search

Containing text: serial

Text to be searched based on some criteria

Options

- ☐ Case sensitive
- ☐ Whole word
- ☐ Regular expression

Select zero, one or more options

Scope

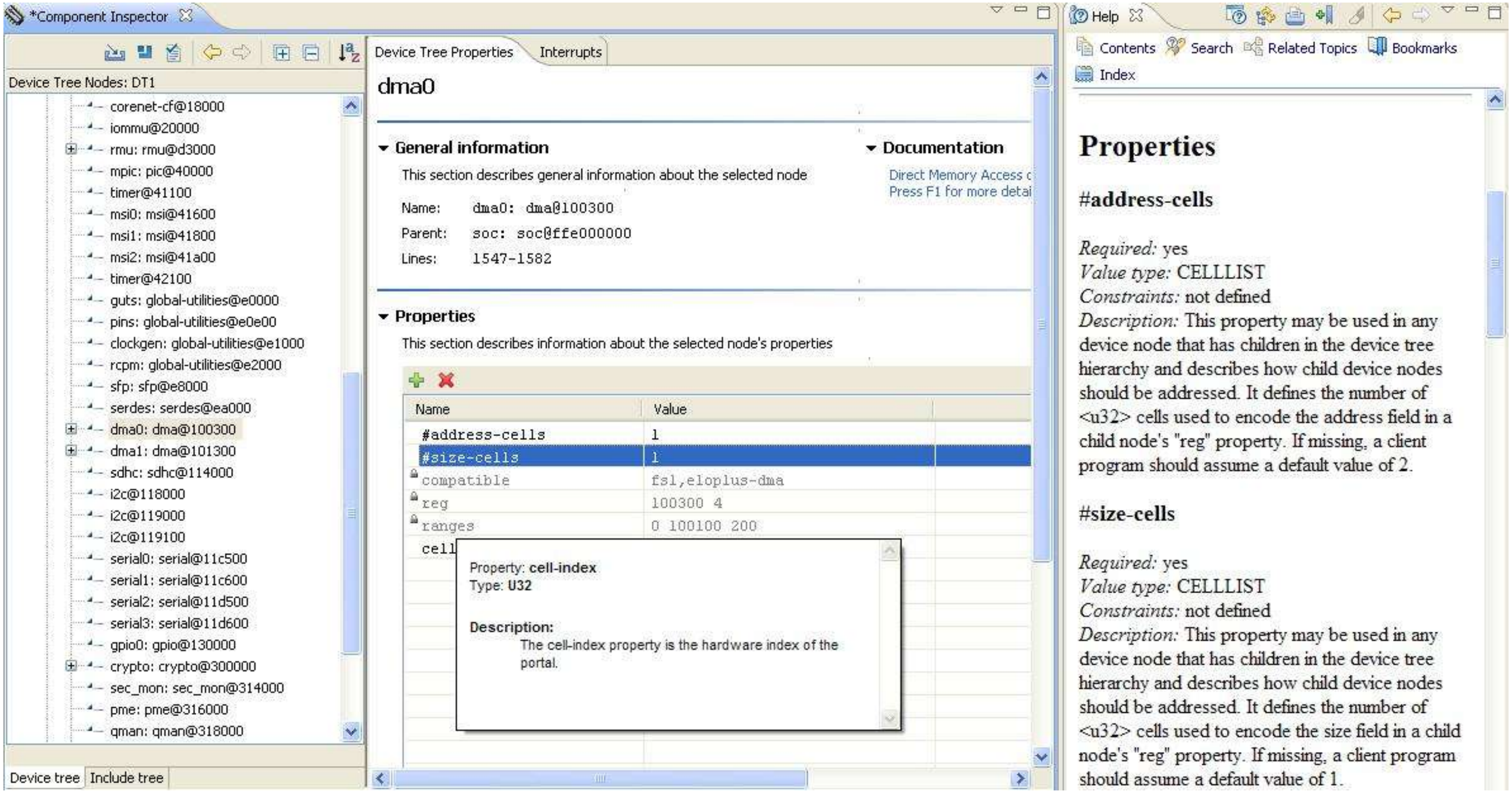
○ Workspace ○ Selected resources ○ Enclosing projects

● Working set: prj

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.



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

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 detai

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	

Property: cell-index
Type: U32

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

Properties

#address-cells

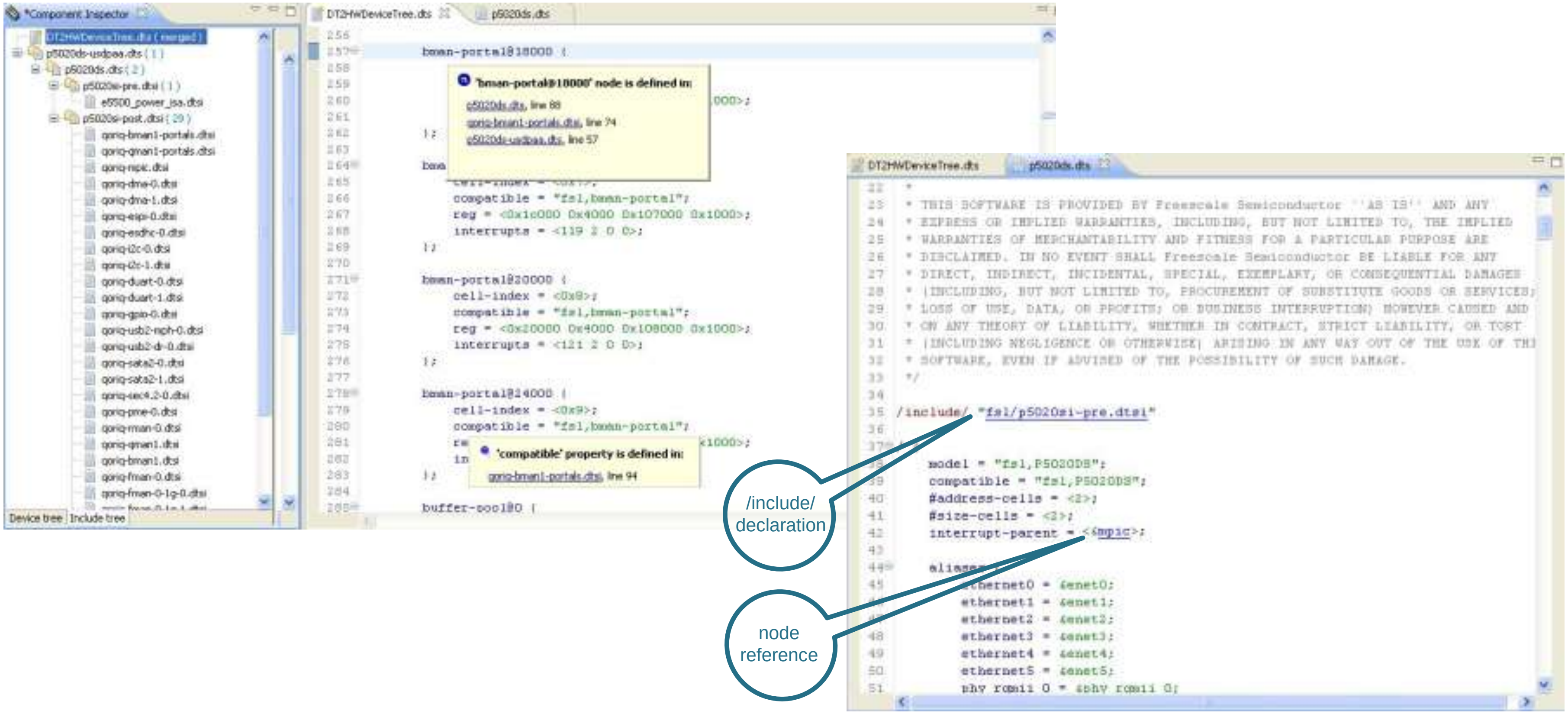
Required: yes
Value type: CELLLIST
Constraints: not defined
Description: This property may be used in any device node that has children in the device tree hierarchy and describes how child device nodes should be addressed. It defines the number of <u32> cells used to encode the address field in a child 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).



```
DT2HWDeviceTree.dts ( merged )
p5020ds-usdpaa.dts ( 1 )
  p5020ds.dts ( 2 )
    p5020si-pre.dtsi ( 1 )
      e5500_power_isa.dtsi
    p5020si-post.dtsi ( 29 )
      qoriq-bman1-portals.dtsi
      qoriq-qman1-portals.dtsi
      qoriq-mpic.dtsi
      qoriq-dma-0.dtsi
      qoriq-dma-1.dtsi
      qoriq-espi-0.dtsi
      qoriq-esdhc-0.dtsi
      qoriq-i2c-0.dtsi
      qoriq-i2c-1.dtsi
      qoriq-duart-0.dtsi
      qoriq-duart-1.dtsi
      qoriq-gpio-0.dtsi
      qoriq-usb2-mph-0.dtsi
      qoriq-usb2-dr-0.dtsi
      qoriq-sata2-0.dtsi
      qoriq-sata2-1.dtsi
      qoriq-sec4.2-0.dtsi
      qoriq-pme-0.dtsi
      qoriq-rman-0.dtsi
      qoriq-qman1.dtsi
      qoriq-bman1.dtsi
      qoriq-fman-0.dtsi
      qoriq-fman-0-1g-0.dtsi
```

```
	bman-portal@18000 {
	                                       000>;
	};

	bma
	    cell-index = <0x7>;
		compatible = "fsl,bman-portal";
		reg = <0x1c000 0x4000 0x107000 0x1000>;
		interrupts = <119 2 0 0>;
	};

	bman-portal@20000 {
		cell-index = <0x8>;
		compatible = "fsl,bman-portal";
		reg = <0x20000 0x4000 0x108000 0x1000>;
		interrupts = <121 2 0 0>;
	};

	bman-portal@24000 {
		cell-index = <0x9>;
		compatible = "fsl,bman-portal";
		re                                     x1000>;
		in
	};

	buffer-pool@0 {
```

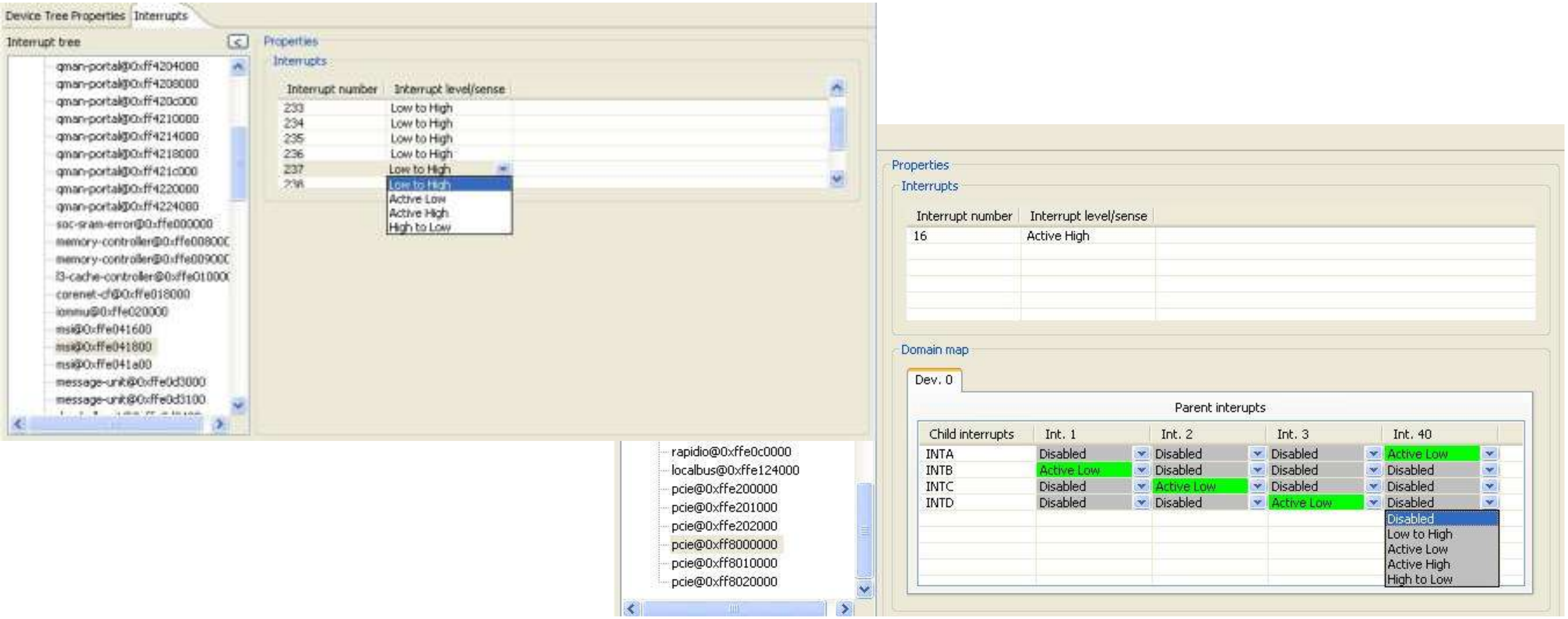
'bman-portal@18000' node is defined in:
p5020ds.dts, line 88
qoriq-bman1-portals.dtsi, line 74
p5020ds-usdpaa.dts, line 57

'compatible' property is defined in:
qoriq-bman1-portals.dtsi, line 94

```
 *
 * THIS SOFTWARE IS PROVIDED BY Freescale Semiconductor ``AS IS'' AND ANY
 * EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED
 * WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE
 * DISCLAIMED. IN NO EVENT SHALL Freescale Semiconductor BE LIABLE FOR ANY
 * DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES
 * (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES;
 * LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND
 * ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
 * (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THI
 * SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
 */

/include/ "fsl/p5020si-pre.dtsi"

/ {
	model = "fsl,P5020DS";
	compatible = "fsl,P5020DS";
	#address-cells = <2>;
	#size-cells = <2>;
	interrupt-parent = <&mpic>;

	aliases {
		ethernet0 = &enet0;
		ethernet1 = &enet1;
		ethernet2 = &enet2;
		ethernet3 = &enet3;
		ethernet4 = &enet4;
		ethernet5 = &enet5;
		phy_rgmii_0 = &phy_rgmii_0;
```

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.



qman-portal@0xff4204000
qman-portal@0xff4208000
qman-portal@0xff420c000
qman-portal@0xff4210000
qman-portal@0xff4214000
qman-portal@0xff4218000
qman-portal@0xff421c000
qman-portal@0xff4220000
qman-portal@0xff4224000
soc-sram-error@0xffe000000
memory-controller@0xffe00800C
memory-controller@0xffe00900C
l3-cache-controller@0xffe01000C
corenet-cf@0xffe018000
iommu@0xffe020000
msi@0xffe041600
msi@0xffe041800
msi@0xffe041a00
message-unit@0xffe0d3000
message-unit@0xffe0d3100

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	

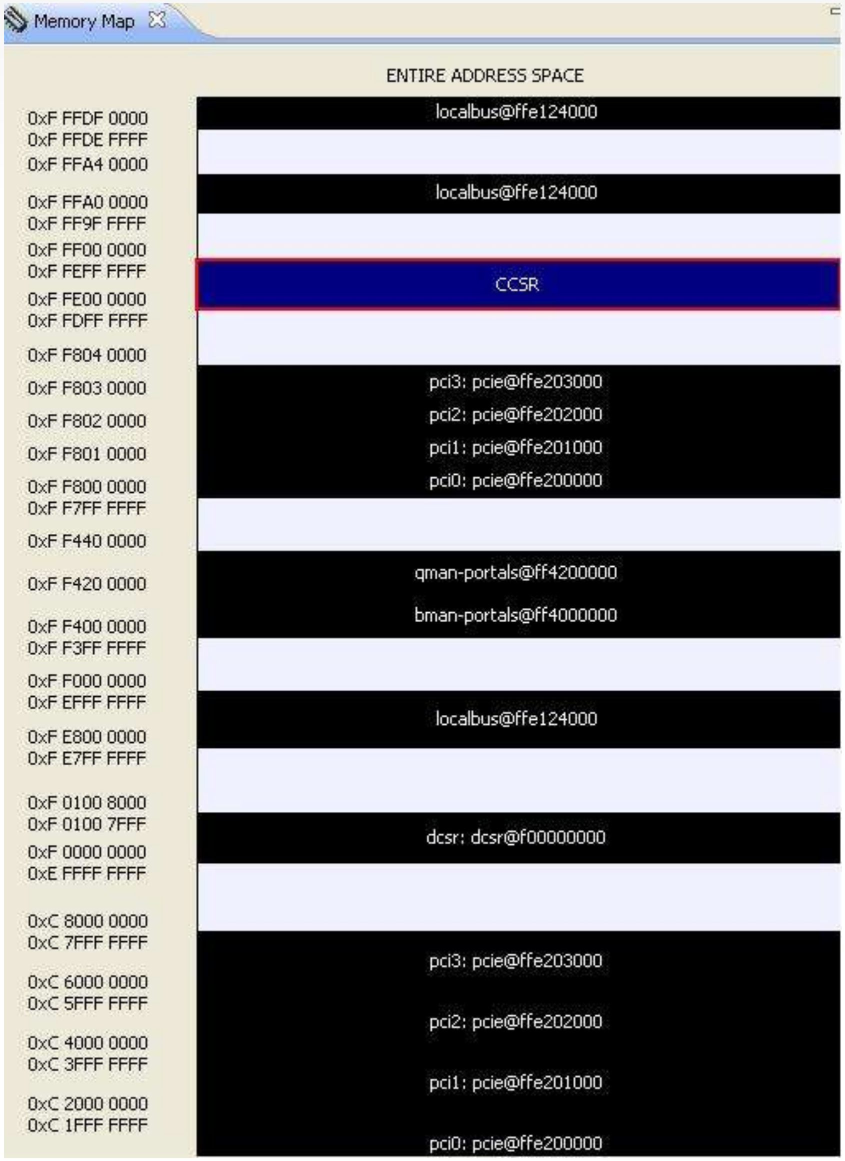
rapidio@0xffe0c0000
localbus@0xffe124000
pcie@0xffe200000
pcie@0xffe201000
pcie@0xffe202000
pcie@0xff8000000
pcie@0xff8010000
pcie@0xff8020000

Interrupt number	Interrupt level/sense
16	Active High

	Parent interrupts			
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

Memory Map view

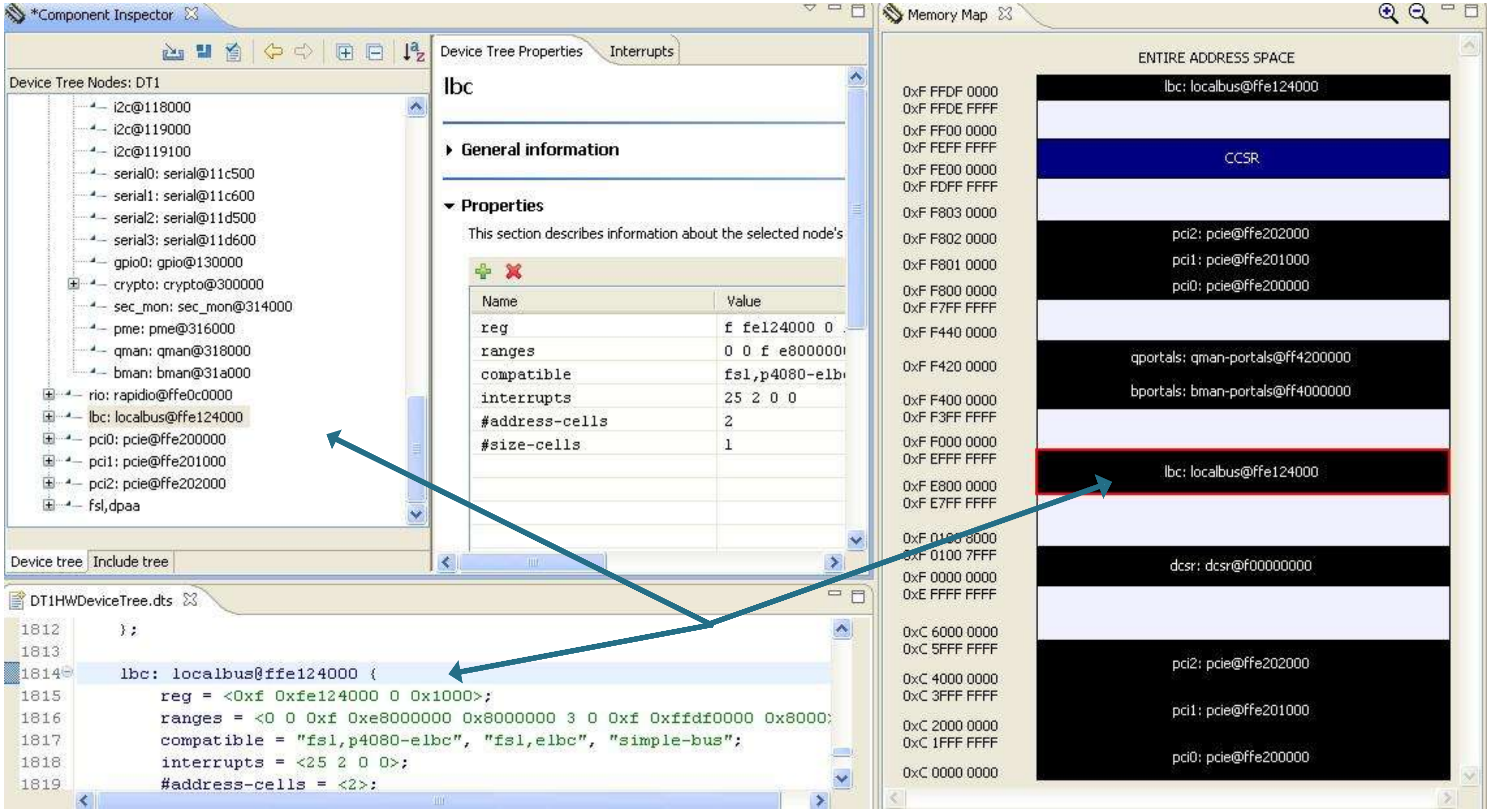
- Any hw device tree can be seen as a representation of different Local Access Windows (LAW).
- Each LAW maps to a specified target interface, such as DDR Controller, Localbus, PCI Express, etc.
- Each device tree node having *reg* and *ranges* properties defines a memory range inside/outside Configuration 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.



```
ENTIRE ADDRESS SPACE
0xF FFDF 0000    localbus@ffe124000
0xF FFDE FFFF
0xF FFA4 0000
0xF FFA0 0000    localbus@ffe124000
0xF FF9F FFFF
0xF FF00 0000
0xF FEFF FFFF    CCSR
0xF FE00 0000
0xF FDFF FFFF
0xF F804 0000
0xF F803 0000    pci3: pcie@ffe203000
0xF F802 0000    pci2: pcie@ffe202000
0xF F801 0000    pci1: pcie@ffe201000
0xF F800 0000    pci0: pcie@ffe200000
0xF F7FF FFFF
0xF F440 0000
0xF F420 0000    qman-portals@ff4200000
0xF F400 0000    bman-portals@ff4000000
0xF F3FF FFFF
0xF F000 0000
0xF EFFF FFFF    localbus@ffe124000
0xF E800 0000
0xF E7FF FFFF
0xF 0100 8000
0xF 0100 7FFF    dcsr: dcsr@f00000000
0xF 0000 0000
0xE FFFF FFFF
0xC 8000 0000
0xC 7FFF FFFF    pci3: pcie@ffe203000
0xC 6000 0000
0xC 5FFF FFFF    pci2: pcie@ffe202000
0xC 4000 0000
0xC 3FFF FFFF    pci1: pcie@ffe201000
0xC 2000 0000
0xC 1FFF FFFF    pci0: pcie@ffe200000
```

Device tree views synchronization

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



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
bman: bman@31a000
rio: rapidio@ffe0c0000
lbc: localbus@ffe124000
pci0: pcie@ffe200000
pci1: pcie@ffe201000
pci2: pcie@ffe202000
fsl,dpaa

lbc

General information

Properties

This section describes information about the selected node's

Name	Value
reg	f fe124000 0
ranges	0 0 f e800000
compatible	fsl,p4080-elb
interrupts	25 2 0 0
#address-cells	2
#size-cells	1

```
    };

    lbc: localbus@ffe124000 {
        reg = <0xf 0xfe124000 0 0x1000>;
        ranges = <0 0 0xf 0xe8000000 0x8000000 3 0 0xf 0xffdf0000 0x8000;
        compatible = "fsl,p4080-elbc", "fsl,elbc", "simple-bus";
        interrupts = <25 2 0 0>;
        #address-cells = <2>;
```

ENTIRE ADDRESS SPACE

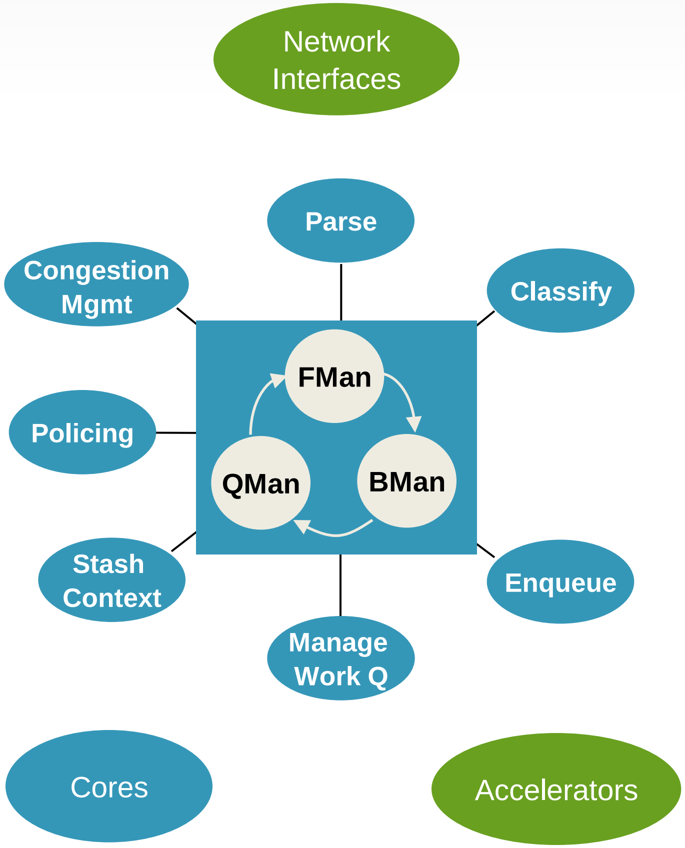
```
0xF FFDF 0000   lbc: localbus@ffe124000
0xF FFDE FFFF
0xF FF00 0000
0xF FEFF FFFF   CCSR
0xF FE00 0000
0xF FDFF FFFF
0xF F803 0000
0xF F802 0000   pci2: pcie@ffe202000
0xF F801 0000   pci1: pcie@ffe201000
0xF F800 0000   pci0: pcie@ffe200000
0xF F7FF FFFF
0xF F440 0000
0xF F420 0000   qportals: qman-portals@ff4200000
0xF F400 0000   bportals: bman-portals@ff4000000
0xF F3FF FFFF
0xF F000 0000
0xF EFFF FFFF
0xF E800 0000   lbc: localbus@ffe124000
0xF E7FF FFFF
0xF 0100 0000
0xF 0100 7FFF
0xF 0000 0000   dcsr: dcsr@f00000000
0xE FFFF FFFF
0xC 6000 0000
0xC 5FFF FFFF   pci2: pcie@ffe202000
0xC 4000 0000
0xC 3FFF FFFF   pci1: pcie@ffe201000
0xC 2000 0000
0xC 1FFF FFFF   pci0: pcie@ffe200000
0xC 0000 0000
```

DPAA Configuration

Data Path Graphing Tool

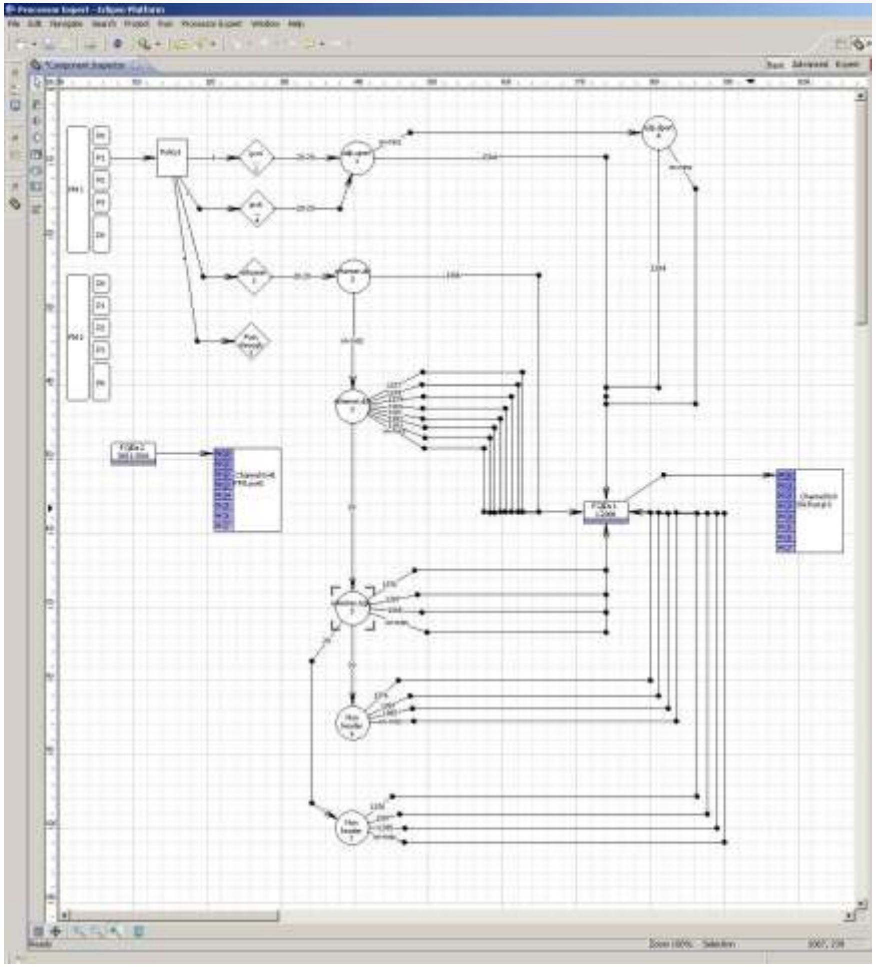
DPAA Overview

- **DPAA (Data Path Acceleration Architecture), provides the infrastructure to pass packets to/from cores, hardware accelerators and network interfaces**
- **The architecture contains several hardware components**
  - Frame Manager (FM)
  - Buffer Manager (BM)
  - Queue Manager (QM)
  - HW accelerators: Security (SEC), Pattern Matching Engine (PME)
- **Each hardware component is performing specific operations on the incoming/outgoing frames**
  - BM – Manages data storage buffer pools. Is a shared resource among cores, network interfaces, and HW accelerators
  - FM – supports in-line/off-line packet parsing and initial classification. It enables policing and flow and QoS – based packet distribution to the cores
  - QM – Manages the queuing of data between cores, network interfaces, and HW accelerators
  - SEC – provides cryptographic acceleration
  - PME – high performance hardware pattern matching functionality



QCS DPAA Component Overview

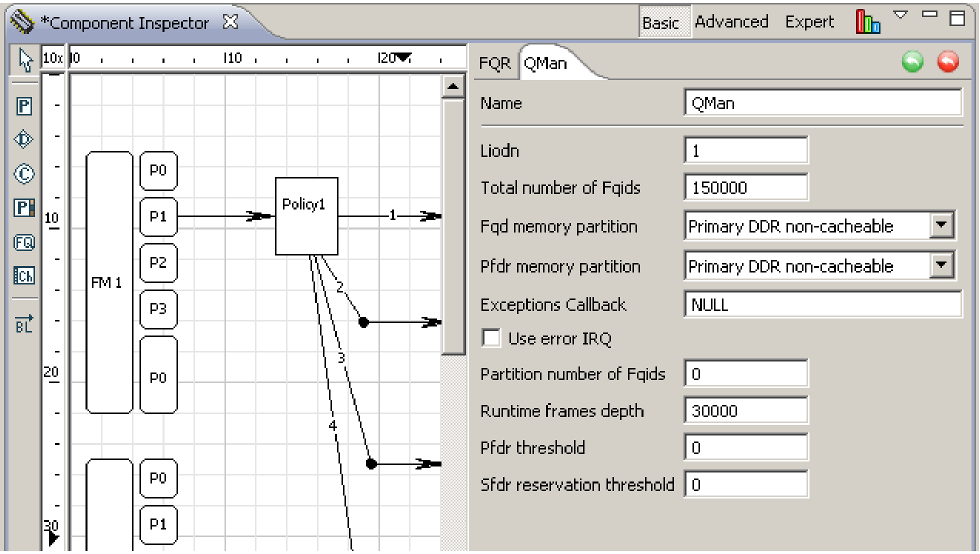
- Flowchart representation of DPAA component is a software solution intended to ease creation of complex DPAA configurations
- Have an intuitive graphical representation
- Easy to understand the overall architecture as well as individual DPAA components
- QCS integrated component designed to ease DPAA configuration for QorIQ devices
- Interactive and user friendly interface in order to provide the best user experience
- Allows customers to easily translate their own data flow into a valid driver configuration
- Designed to deal with complex DPAA user scenarios



```
<?xml version="1.0" encoding="utf-8"?>

<dpc xmlns:xi="http://www.w3.org/2001/XInclude">

<cfgdata>
    <config>
        <engine name="fm0">
            <port type="1G" number="0" policy="L2TPv3" portid="0x0"/>
            <port type="1G" number="3" policy="UDP" portid="0x0"/>
        </engine>
        <engine name="fm1"/>
    </config>
</cfgdata>

<netpcd>
    <distribution name="Pass-through">
        <queue count="16" base="0x85"/>
    </distribution>
    <distribution name="IPv4">
        <queue count="128" base="0x123"/>
        <protocols>
            <protocolref name="ipv4"/>
            <protocolref name="ipv6"/>
        </protocols>
    </distribution>
    <distribution name="MAC.dst">
        <queue count="128" base="0x980"/>
        <key>
            <fieldref name="ethernet.dst"/>
            <fieldref name="ethernet.src"/>
            <fieldref name="vlan.tci"/>
        </key>
        <action type="classification" name="MAC.dst"/>
    </distribution>
    <distribution name="UDP">
        <queue count="16" base="0x1450"/>
        <protocols>
            <protocolref name="udp"/>
        </protocols>
    </distribution>
    <classification name="UDP">
        <key>
            <fieldref name="ethernet.dst"/>
        </key>
        <entry>
            <data>0x0</data>
            <mask>0xf</mask>
            <queue base="0x1000"/>
        </entry>
        <entry>
            <data>0x0</data>
            <mask>0xf</mask>
            <queue base="0x1100"/>
        </entry>
        <action condition="on-miss" type="drop"/>
    </classification>
    <policy name="L2TPv3">
        <dist_order>
            <distributionref name="IPv4"/>
            <distributionref name="MAC.dst"/>
            <distributionref name="Pass-through"/>
        </dist_order>
    </policy>
    <policy name="UDP">
        <dist_order>
            <distributionref name="UDP"/>
        </dist_order>
    </policy>
</netpcd>

<bman name = "bman_master">
      <portals>
       <bmportal id = "0" irq = "false"/>
                    <bmportal id = "3" irq = "false"/>
      </portals>
      <liodn>20</liodn>
      <irq>false</irq>
</bman>

<bufferpool name = "bpool_1">
      <bpid>1</bpid>
      <buffers>1000</buffers>
                <size>400</size>
</bufferpool>

<bufferpool name = "bpool_2">
      <bpid>2</bpid>
      <buffers>1000</buffers>
                <size>400</size>
</bufferpool>

<qman name = "qman_master">
      <portals>
                    <qmportal id = "0" irq = "false" liodn = "1"/>
                    <qmportal id = "3" irq = "false" liodn = "1"/>
      </portals>
      <totalfqids>150000</totalfqids>
      <fqdmemorypartition> e_MEM_1ST_DDR_NON_CACHEABLE
</fqdmemorypartition>
      <pfdrmemorypartition> e_MEM_1ST_DDR_NON_CACHEABLE
</pfdrmemorypartition>
      <irq>false</irq>
                <fqBase>1</fqBase>
                <rtframesdepth>30000</rtframesdepth>
                <pfdrtreshold>0</pfdrtreshold>
                <sfdrtreshold>0</sfdrtreshold>
</qman>

<fmport name = "fm0port01" type = "1G" number = "1" engine = "fm0">
    <rx name = "fm0port01rx">
        <errFqid>1</errFqid>
        <dfltFqid>2</dfltFqid>
        <liodnOffset>0</liodnOffset>
        <bufferpools>
            <bpool name = "bpool_1"/>
            <bpool name = "bpool_2"/>
            <bpool name = "bpool_3"/>
        </bufferpools>
    </rx>
    <nonrx name = "fm0port01tx">
        <errFqid>3</errFqid>
        <dfltFqid>4</dfltFqid>
    </nonrx>
    <mac name = "fm0port01mac">
        <irq>false</irq>
        <addr>0x00049f000266</addr>
        <interface>e_ENET_IF_RGMII</interface>
        <speed>e_ENET_SPEED_1000</speed>
        <resetoninit>true</resetoninit>
        <loopback>true</loopback>
    </mac>
    <dpaport name = "dpaportFm0P01"/>
</fmport>

<fmport name = "fm0port03" type = "1G" number = "3" engine = "fm0">
    <rx name = "fm0port03rx">
        <errFqid>1</errFqid>
        <dfltFqid>2</dfltFqid>
        <liodnOffset>0</liodnOffset>
        <bufferpools>
            <bpool name = "bpool_1"/>
            <bpool name = "bpool_2"/>
            <bpool name = "bpool_3"/>
        </bufferpools>
    </rx>
    <nonrx name = "fm0port01tx">
        <errFqid>3</errFqid>
        <dfltFqid>4</dfltFqid>
    </nonrx>
    <mac name = "fm0port01mac">
        <irq>false</irq>
        <addr>0x00049f000266</addr>
        <interface>e_ENET_IF_RGMII</interface>
        <speed>e_ENET_SPEED_1000</speed>
        <resetoninit>true</resetoninit>
        <loopback>true</loopback>
    </mac>
    <dpaport name = "dpaportFm0P01"/>
</fmport>

<fqr name = "FQIDs_1-100">
    <dest>0x60</dest>
    <workqueue>0</workqueue>
    <force>true</force>
    <fqid>1</fqid>
    <count>100</count>
    <align>128</align>
    <txconfclbk>FQR_TxConfCB</txconfclbk>
</fqr>

<fqr name = "FQIDs_101-500">
    <dest>0x21</dest>
    <workqueue>2</workqueue>
    <force>true</force>
```

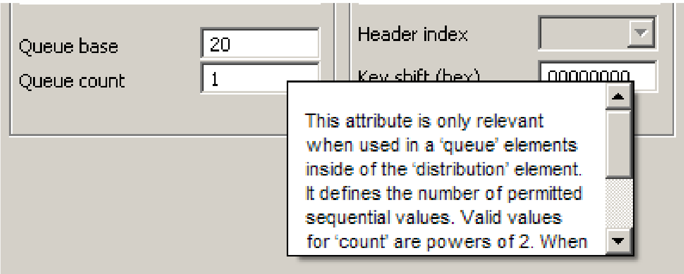
QCS DPAA Component Features (1)

- Default values capability
- Easy access to configuration settings for each DPAA element



Name	QMan
Liodn	1
Total number of Fqids	150000
Fqd memory partition	Primary DDR non-cacheable
Pfdr memory partition	Primary DDR non-cacheable
Exceptions Callback	NULL
□ Use error IRQ	
Partition number of Fqids	0
Runtime frames depth	30000
Pfdr threshold	0
Sfdr reservation threshold	0

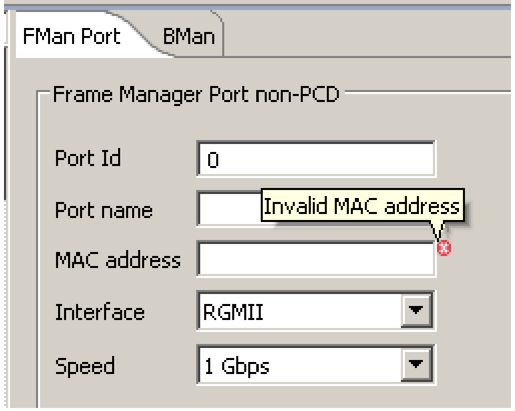
Instant display of relevant description for each configuration parameter



Queue base	20
Queue count	1
Header index	
Key shift (hex)	00000000

This attribute is only relevant when used in a 'queue' elements inside of the 'distribution' element. It defines the number of permitted sequential values. Valid values for 'count' are powers of 2. When

Automatic input validation, configuration constraints checking and instant display of relevant conflict messages



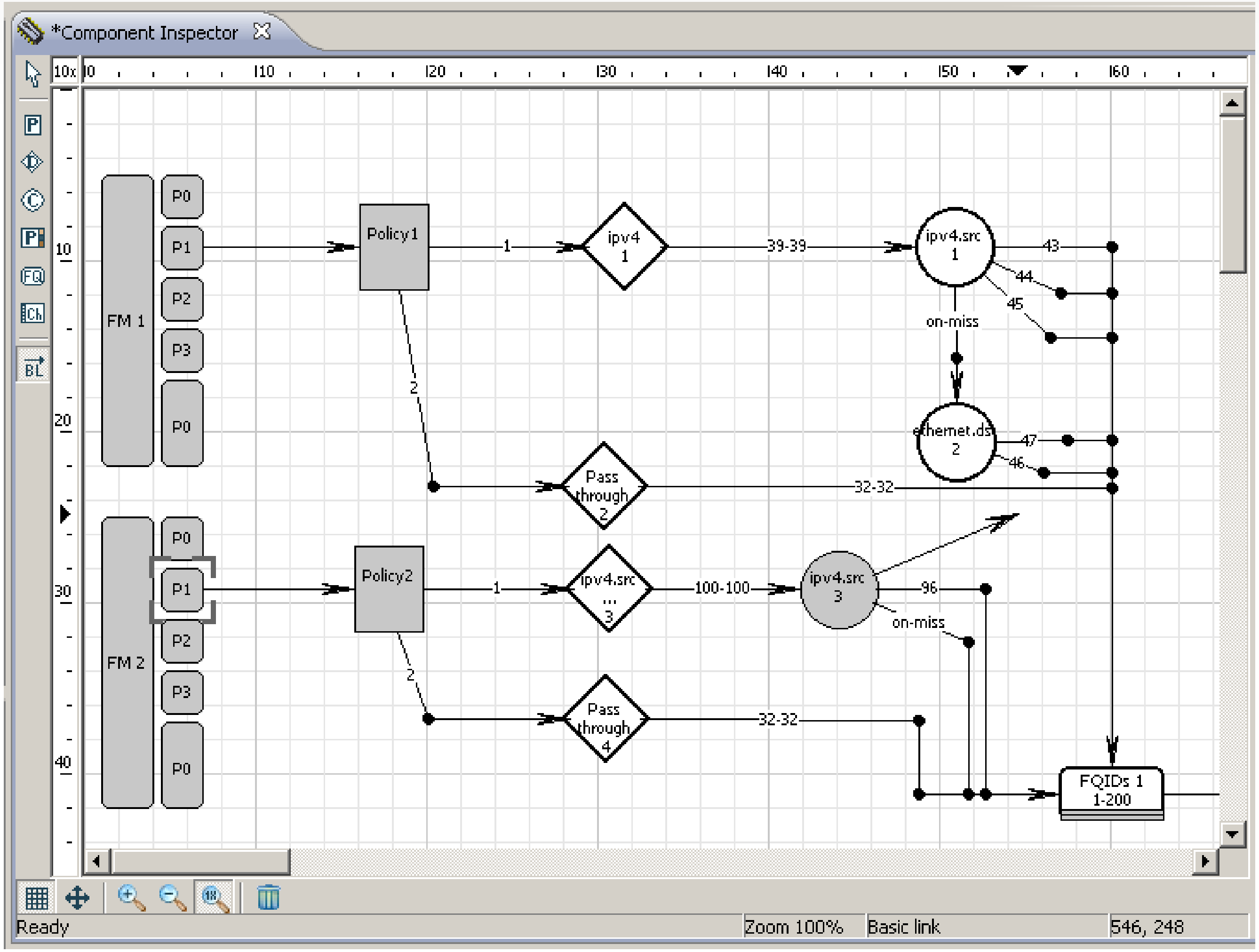
Frame Manager Port non-PCD

Port Id	0
Port name	
MAC address	
Interface	RGMII
Speed	1 Gbps

Invalid MAC address

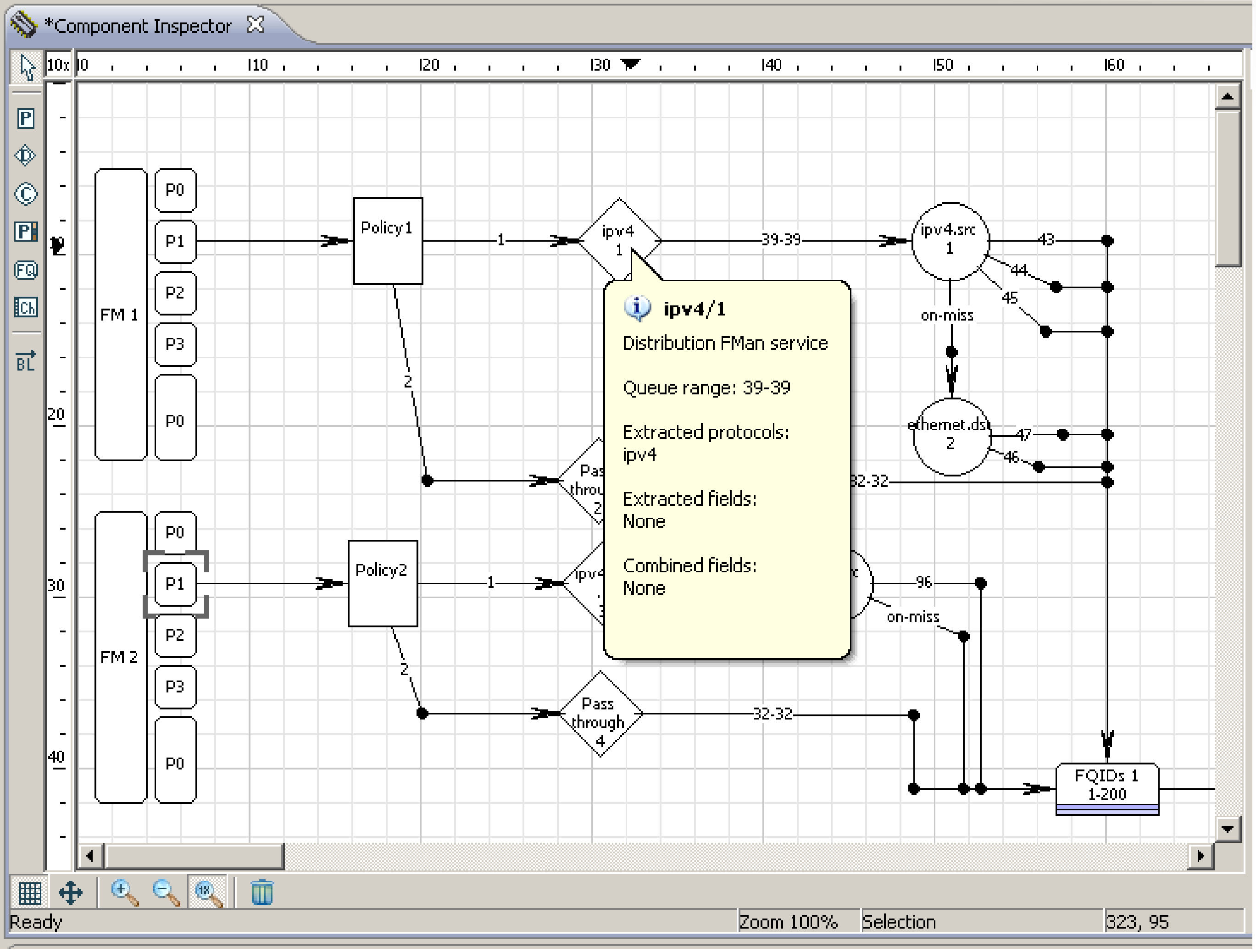
QCS DPAA Component Features (2)

- On-the-fly configuration validation by highlighting correct choices and graying out the invalid ones



QCS DPAA Component Features (3)

- Instant display of relevant configuration summary for each DPAA element



ipv4/1

Distribution FMan service

Queue range: 39-39

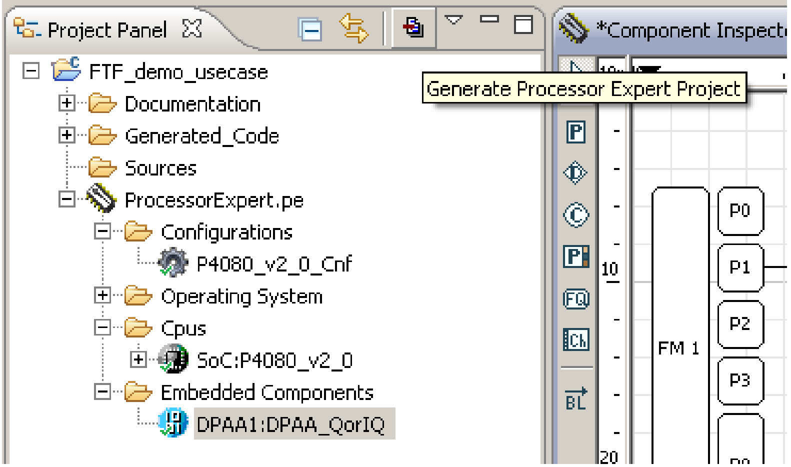
Extracted protocols:
ipv4

Extracted fields:
None

Combined fields:
None

QCS DPAA Component Features (4)

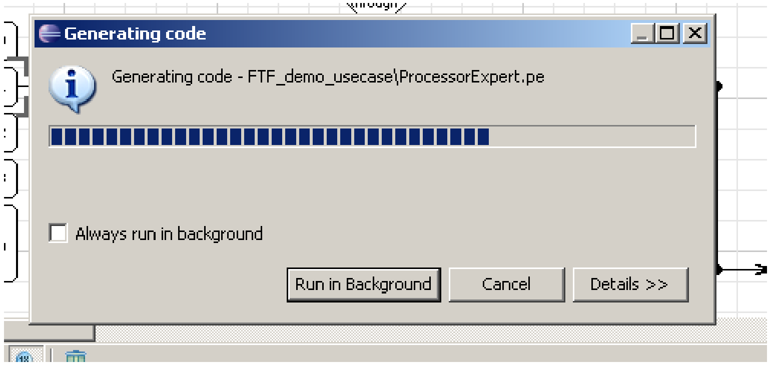
- Immediate code generation at user request in any stage of configuration



Project Panel

- FTF_demo_usecase
  - Documentation
  - Generated_Code
  - Sources
  - ProcessorExpert.pe
    - Configurations
      - P4080_v2_0_Cnf
    - Operating System
    - Cpus
      - SoC:P4080_v2_0
    - Embedded Components
      - DPAA1:DPAA_QorIQ

Generate Processor Expert Project

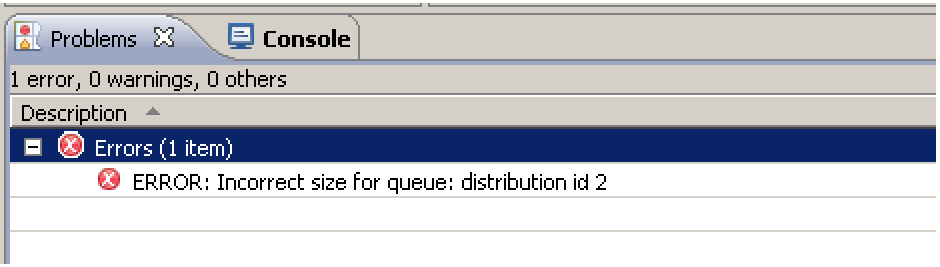


Generating code

Generating code - FTF_demo_usecase\ProcessorExpert.pe

☐ Always run in background

- Immediate notification for all errors occurred during the code generation process



1 error, 0 warnings, 0 others

Description
Errors (1 item)
ERROR: Incorrect size for queue: distribution id 2

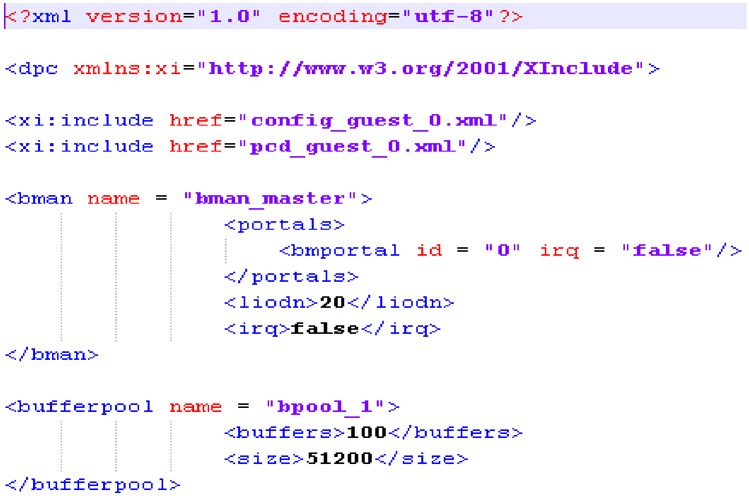
freescale™ 飞思卡尔

Import XMLs Feature

- **Imports DPAA configuration from Freescale extensions to NetPDL xml based files:**
  - DPAA objects
  - Connections between them
  - DPAA objects configuration
  - Automatic update of the objects and links after import is done
- **The xml files can be generated using the QCS solution or can be created by hand**

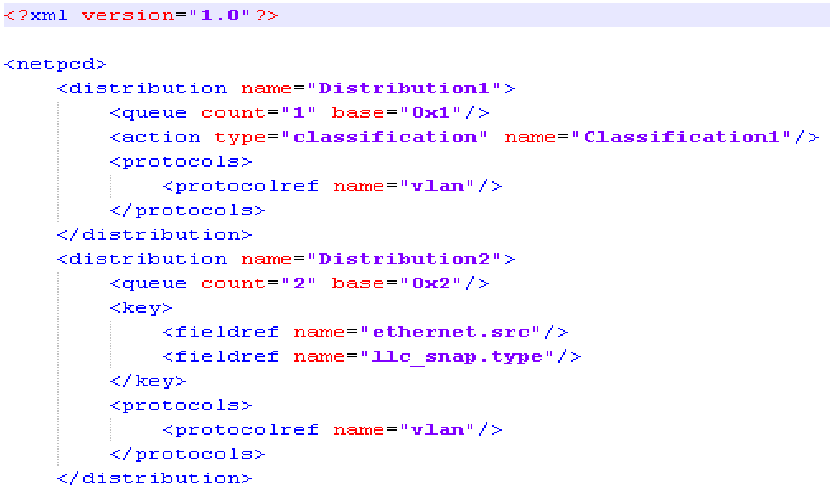
DPC Configuration

- this is the file that has to be imported



```
<?xml version="1.0" encoding="utf-8"?>

<dpc xmlns:xi="http://www.w3.org/2001/XInclude">

<xi:include href="config_guest_0.xml"/>
<xi:include href="pcd_guest_0.xml"/>

<bman name = "bman_master">
                <portals>
                    <bmportal id = "0" irq = "false"/>
                </portals>
                <liodn>20</liodn>
                <irq>false</irq>
</bman>

<bufferpool name = "bpool_1">
                <buffers>100</buffers>
                <size>51200</size>
</bufferpool>
```

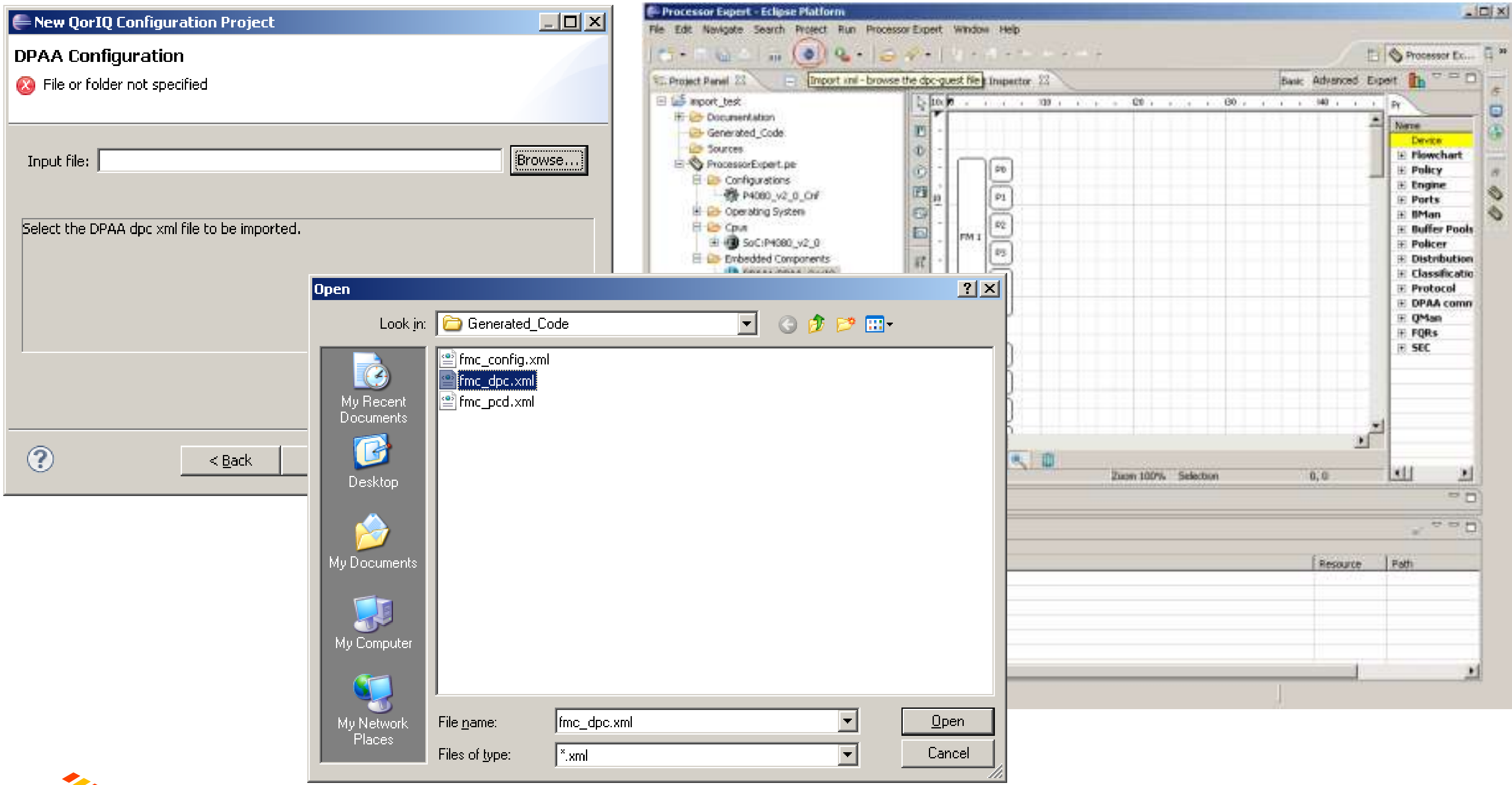
PCD Configuration



```
<?xml version="1.0"?>

<netpcd>
    <distribution name="Distribution1">
        <queue count="1" base="0x1"/>
        <action type="classification" name="Classification1"/>
        <protocols>
            <protocolref name="vlan"/>
        </protocols>
    </distribution>
    <distribution name="Distribution2">
        <queue count="2" base="0x2"/>
        <key>
            <fieldref name="ethernet.src"/>
            <fieldref name="llc_snap.type"/>
        </key>
        <protocols>
            <protocolref name="vlan"/>
        </protocols>
    </distribution>
```

Import XMLs Feature

- **The xmls can be imported at project creation time or later after the project is created**



New QorIQ Configuration Project

DPAA Configuration

File or folder not specified

Input file:

Select the DPAA dpc xml file to be imported.

Open

Look in: Generated_Code

fmc_config.xml
fmc_dpc.xml
fmc_pcd.xml

File name: fmc_dpc.xml

Files of type: *.xml

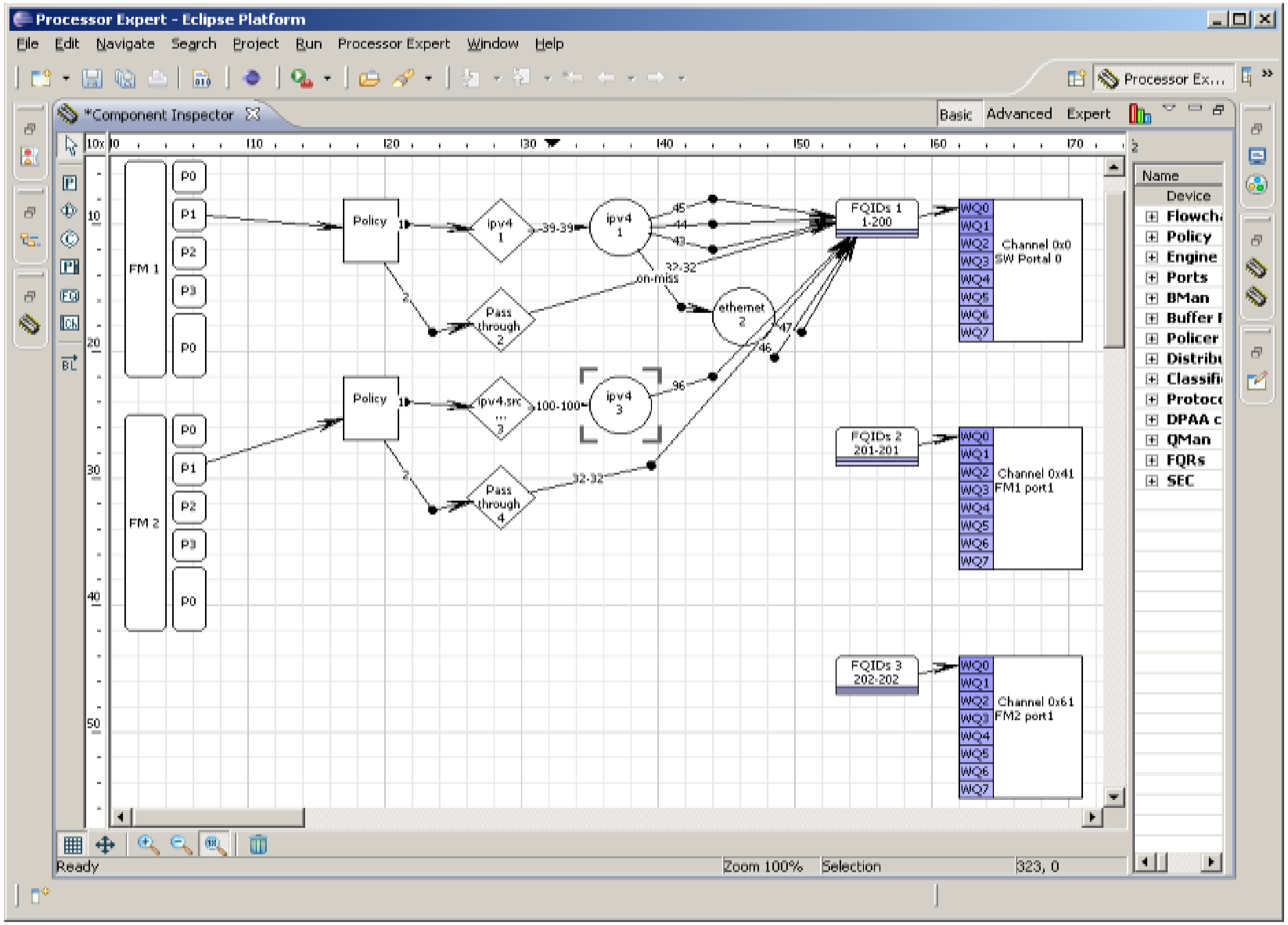
Processor Expert - Eclipse Platform

Import xml - browse the dpc-guest file

Import XMLs Feature – Demo

- Importing files previously generated by the QCS tool won’t produce the same output
- The objects’ coordinates are not saved in the xmls and are calculated using a “placement algorithm” at the import time
- Import demo:

  Using the xmls generated by the hands-on scenario presented in the upper slides



Summary

Summary

You should now

- Be familiar with the QorIQ Configuration Suite basics
  - v1 supports PBL & DDR configuration (available now)
  - v2 adds Device Tree and DPAA Graphing tools (preview in July)

- **Solution/Strategy**
  - **Extensible suite of tools to solve these problems**
  - 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…

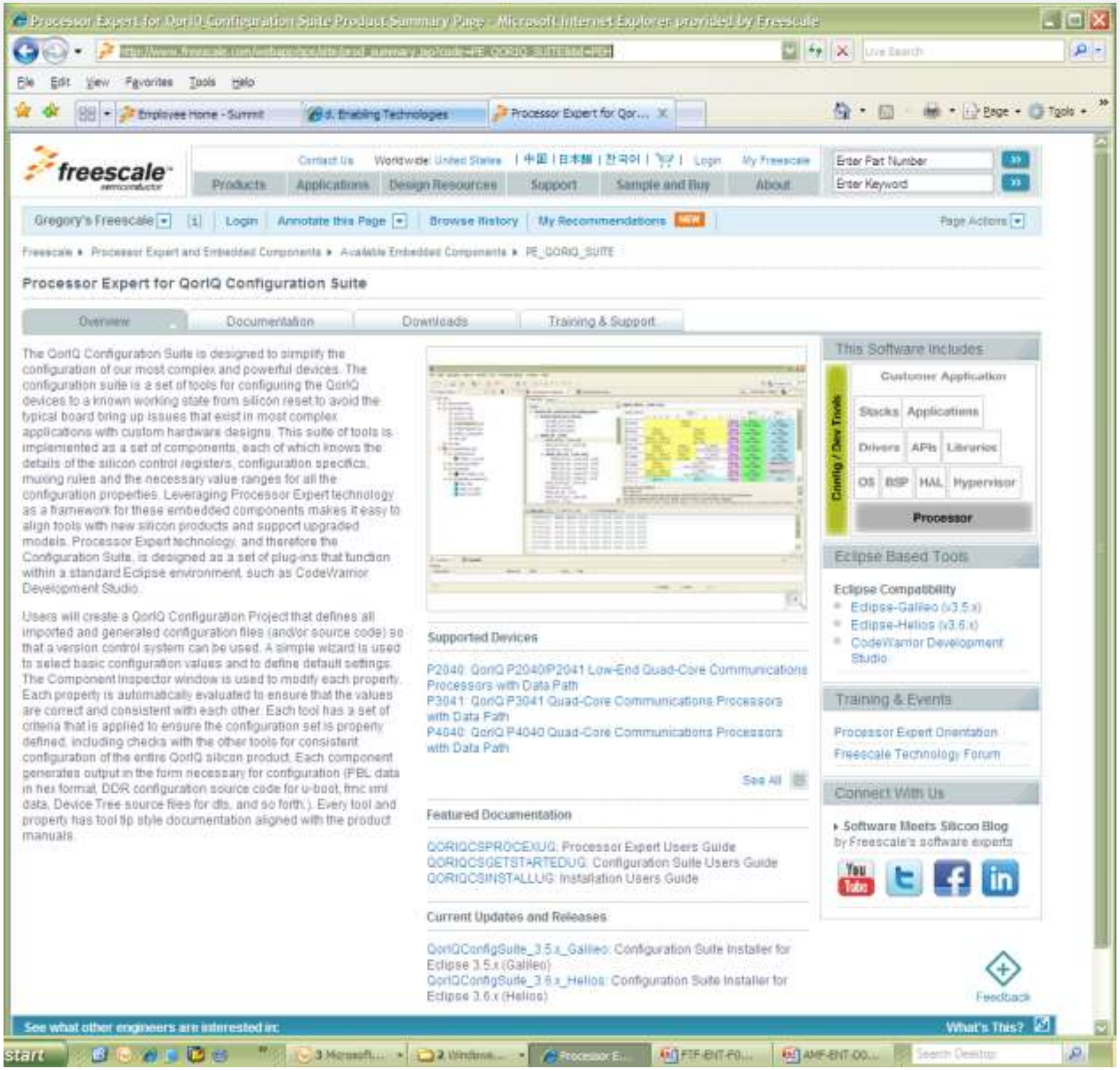
Processor Expert for QorIQ … For More Info

- Processor Expert for QorIQ Configuration Suite
  - http://www.freescale.com/webapp/sps/site/prod_summary.jsp?code=PE_QORIQ_SUITE&tid=PEH

- Freescale's Processor Expert landing page
  - http://www.freescale.com/webapp/sps/site/prod_summary.jsp?code=PROCESSOR-EXPERT&tid=PEH
  - http://www.processorexpert.com/

- Freescale Software & Tools website
  - http://www.freescale.com/webapp/sps/site/homepage.jsp?code=DEVELOPER_HOME

- Freescale Component Store – purchasing embedded software
  - http://www.freescale.com/webapp/sps/site/homepage.jsp?code=BEAN_STORE_MAIN&tid=SWnT


Processor Expert for QorIQ Configuration Suite Product Summary Page - Microsoft Internet Explorer provided by Freescale
File Edit View Favorites Tools Help
Employee Home - Summit
d. Enabling Technologies
Processor Expert for Qor...
Contact Us
Worldwide: United States
Login
My Freescale
Products
Applications
Design Resources
Support
Sample and Buy
About
Enter Part Number
Enter Keyword
Gregory's Freescale
Login
Annotate this Page
Browse History
My Recommendations
NEW
Page Actions
Freescale ▸ Processor Expert and Embedded Components ▸ Available Embedded Components ▸ PE_QORIQ_SUITE

Processor Expert for QorIQ Configuration Suite

The QorIQ Configuration Suite is designed to simplify the configuration of our most complex and powerful devices. The configuration suite is a set of tools for configuring the QorIQ devices to a known working state from silicon reset to avoid the typical board bring up issues that exist in most complex applications with custom hardware designs. This suite of tools is implemented as a set of components, each of which knows the details of the silicon control registers, configuration specifics, muxing rules and the necessary value ranges for all the configuration properties. Leveraging Processor Expert technology as a framework for these embedded components makes it easy to align tools with new silicon products and support upgraded models. Processor Expert technology, and therefore the Configuration Suite, is designed as a set of plug-ins that function within a standard Eclipse environment, such as CodeWarrior Development Studio.

Users will create a QorIQ Configuration Project that defines all imported and generated configuration files (and/or source code) so that a version control system can be used. A simple wizard is used to select basic configuration values and to define default settings. The Component Inspector window is used to modify each property. Each property is automatically evaluated to ensure that the values are correct and consistent with each other. Each tool has a set of criteria that is applied to ensure the configuration set is properly defined, including checks with the other tools for consistent configuration of the entire QorIQ silicon product. Each component generates output in the form necessary for configuration (PBL data in hex format, DDR configuration source code for u-boot, fmc xml data, Device Tree source files for dts, and so forth.). Every tool and property has tool tip style documentation aligned with the product manuals.

Supported Devices

P2040: QorIQ P2040/P2041 Low-End Quad-Core Communications Processors with Data Path
P3041: QorIQ P3041 Quad-Core Communications Processors with Data Path
P4040: QorIQ P4040 Quad-Core Communications Processors with Data Path

See All

Featured Documentation

QORIQCSPROCEXUG: Processor Expert Users Guide
QORIQCSGETSTARTEDUG: Configuration Suite Users Guide
QORIQCSINSTALLUG: Installation Users Guide

Current Updates and Releases

QorIQConfigSuite_3.5.x_Galileo: Configuration Suite Installer for Eclipse 3.5.x (Galileo)
QorIQConfigSuite_3.6.x_Helios: Configuration Suite Installer for Eclipse 3.6.x (Helios)

This Software Includes

Eclipse Based Tools

Eclipse Compatibility

- Eclipse-Galileo (v3.5.x)
- Eclipse-Helios (v3.6.x)
- CodeWarrior Development Studio

Training & Events

Processor Expert Orientation
Freescale Technology Forum

Connect With Us

▸ Software Meets Silicon Blog
by Freescale's software experts

Supporting the Most Sophisticated Customers…

FUTURE ELECTRONICS CO.LTD ADVA OPTICAL NETWORK AEONIUM SERVICES LTD ALCATEL-LUCENT ALCATEL-LUCENT AG ALVARION ARROW ELECTRONICS ASB AVNET ELECTRONICS BARCO N.V. CANOGA PERKINS CEAC INT'L CES CREATIVE ELECTRONIC CHANGWON UNIV. CISCO CISCO SYSTEMS CONTINOUS COMPUTING CRYPTO AG CYAN INC DIALOGIC, INC. DSPACE GMBH EDIXIA EMBEDDED SOLUTIONS EMCOSYS FUTURE ELECT INC GAMMA SP. Z O.O. GE INTELLEGENT PLATFORMS LTD GIGAMON SYSTEMS GUODIAN NANJING AUTOMATION HEIDENHAIN GMBH IEP GMBH INTERFACE CONCEPT IPWIRELESS ISKRATEL ELECTRONICS ITEC CONSULTING AS JUNIPER NETWORKS JUNIPER NETWORKS INDIA KDS KOLLMORGEN KONTRON MODULAR COMPUTERS KPIT CUMMINS INFOSYSTEMS LTD L-3 COMMUNICATIONS A LOCKHEED MARTIN AERONAUTICS CO. MAGNETI MARELLI MOTORSPORT MERCURY COMPUTER SYSTEMS MIRANDA TECHNOLOGIES MOTOROLA MOBILITY INC. MYSTICAL ROSE TECHNOLOGIES NARI NETWORKS NKB VS PURESILICON, INC. REDCOM LABORATORIES INC ROHDE & SCHWARZ ROHDESCHWARZ GMBHCO. KG ROSS VIDEO SASET CHENGDU SERVERGY SERVERGY INC. SIMENA SOUTHERN FEDERAL UNIVERSITY SUSQUEHANNA INTERNATIONAL GROUP LLP TECHNOLOGY OF INFINITY THERMO FISHER SCIENTIFIC WB ELECTRONICS S.A. WINDRIVER WT MICROELECTRONICS CO. ZODIAC AEROSPACE

QorIQ Configuration Suite

QorIQ Configuration Suite

Lab 1: Installing QCS

Installing QCS

- **Download QCS2.1Training.zip**

- **Skip if you already have QCS v2.1 installed**
  - Setup for labs:
    - Create a directory where you have read/write permissions
      - Eg C:/QCS21 … from now on, we'll call this directory <qcs>
    - If you already have QCS installed, use that directory as <qcs>

- **Copy the QCS2.1Training.zip "\labs" directory into <qcs>**

- **Follow the instructions in**

  <qcs>/labs/ QORIQCSINSTALLUG.pdf

Pre-boot Loader

Lab 2: Custom Hardware Configuration

Pre-boot Loader Hands-on

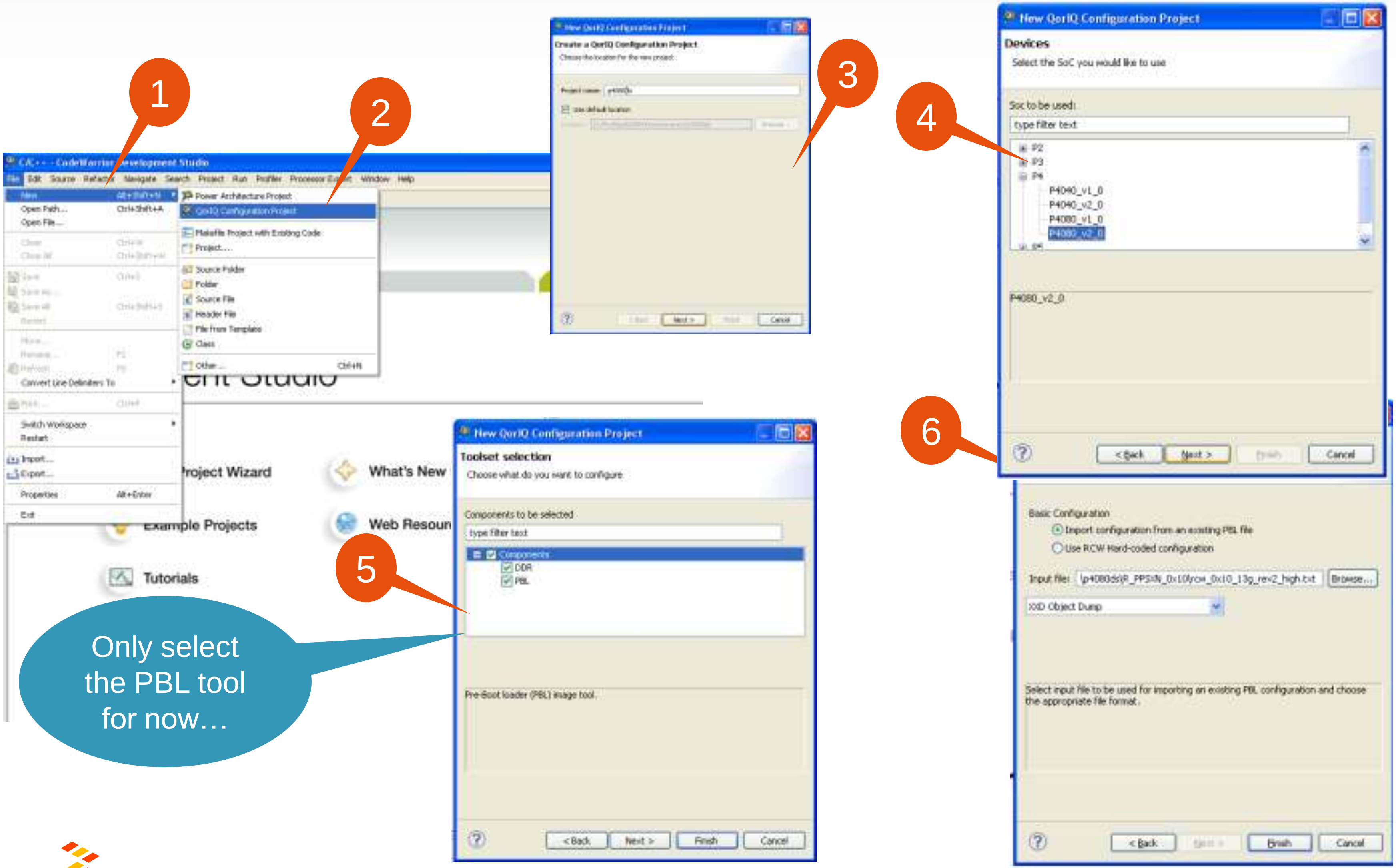
- **Step 1: Import and decode low speed config rcw_0x10_5g_rev2_low.bin.txt**
  - Platform clock : 600MHz
  - Core clock : 1.2GHz
  - FMAN1/2 clock : 450 MHz
  - DDR clock : 600MHz (1.2GHz)

- **Step 2: Use PBL tool to increase clock speed up to**
  - Platform clock : 800MHz
  - Core clock : 1.5GHz
  - FMAN1/2 clock : 600 MHz
  - DDR clock : 650MHz clock (1.3GHz)

- **Step 3: Use PBL tool to generate new RCW and compare outcomes with rcw_0x10_5g_rev2_high.bin.txt**

- **Step 4: Change Serdes Config to support 8 Gbe, compare result with rcw_0x16_all_rev2_high.bin**

Pre-Boot Loader Step 1: Create a New Project



New QorIQ Configuration Project

Create a QorIQ Configuration Project

Choose the location for the new project.

Project name: p4080ds

☐ Use default location

New QorIQ Configuration Project

Devices

Select the SoC you would like to use

Soc to be used:

type filter text

- P2
- P3
- P4
  - P4040_v1_0
  - P4040_v2_0
  - P4080_v1_0
  - P4080_v2_0
- P5

P4080_v2_0

New QorIQ Configuration Project

Toolset selection

Choose what do you want to configure

Components to be selected

type filter text

- ☑ Components
  - ☑ DDR
  - ☑ PBL

Pre-Boot loader (PBL) image tool.

Basic Configuration

● Import configuration from an existing PBL file

○ Use RCW Hard-coded configuration

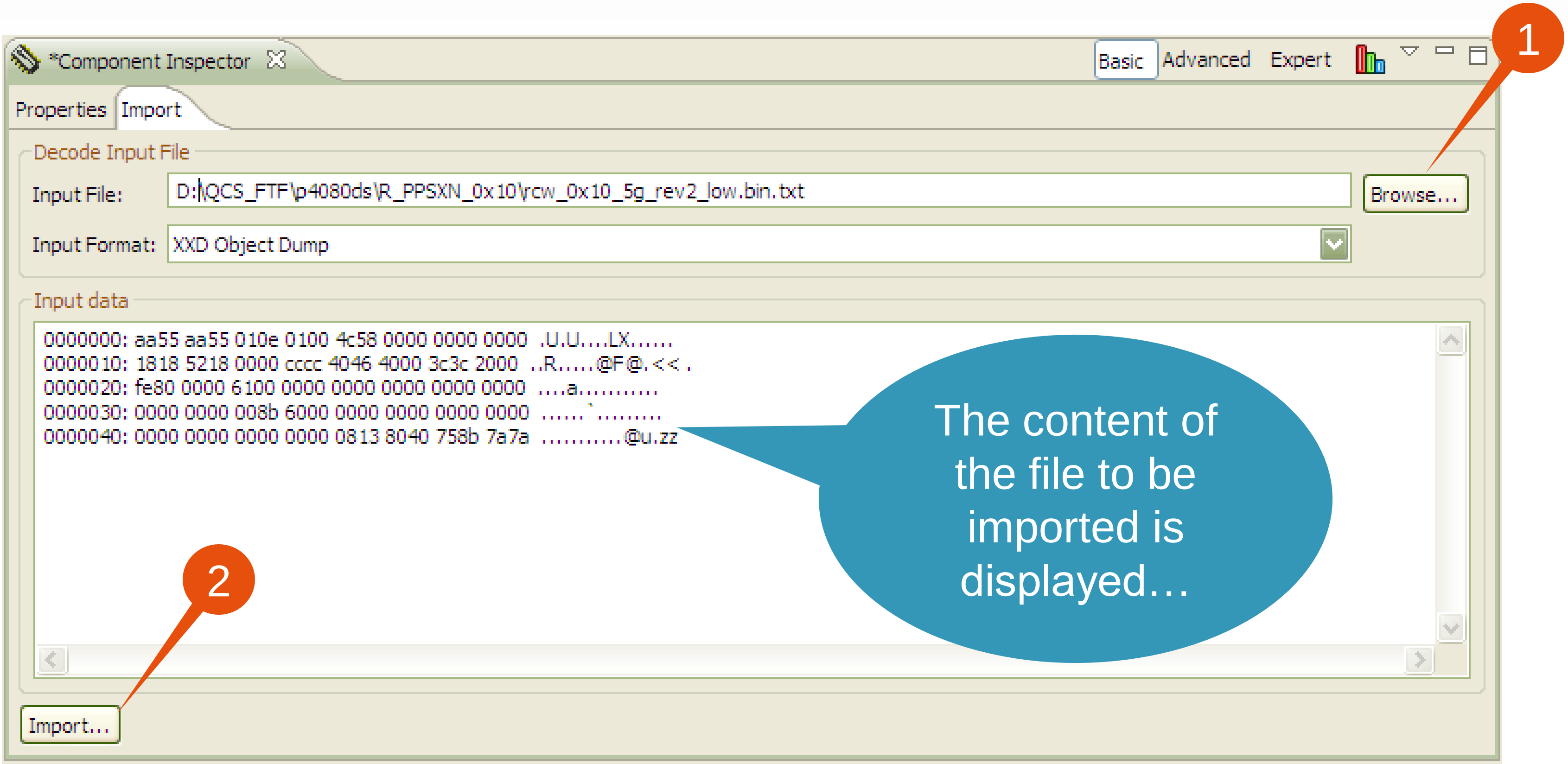
Input file: \p4080ds\R_PPSXN_0x10\rcw_0x10_13g_rev2_high.txt

XXD Object Dump

Select input file to be used for importing an existing PBL configuration and choose the appropriate file format.

Only select the PBL tool for now…

Pre-boot Loader Step 2a:
Import rcw_0x10_5g_rev2_low.bin.txt



Decode Input File

Input File: D:\QCS_FTF\p4080ds\R_PPSXN_0x10\rcw_0x10_5g_rev2_low.bin.txt

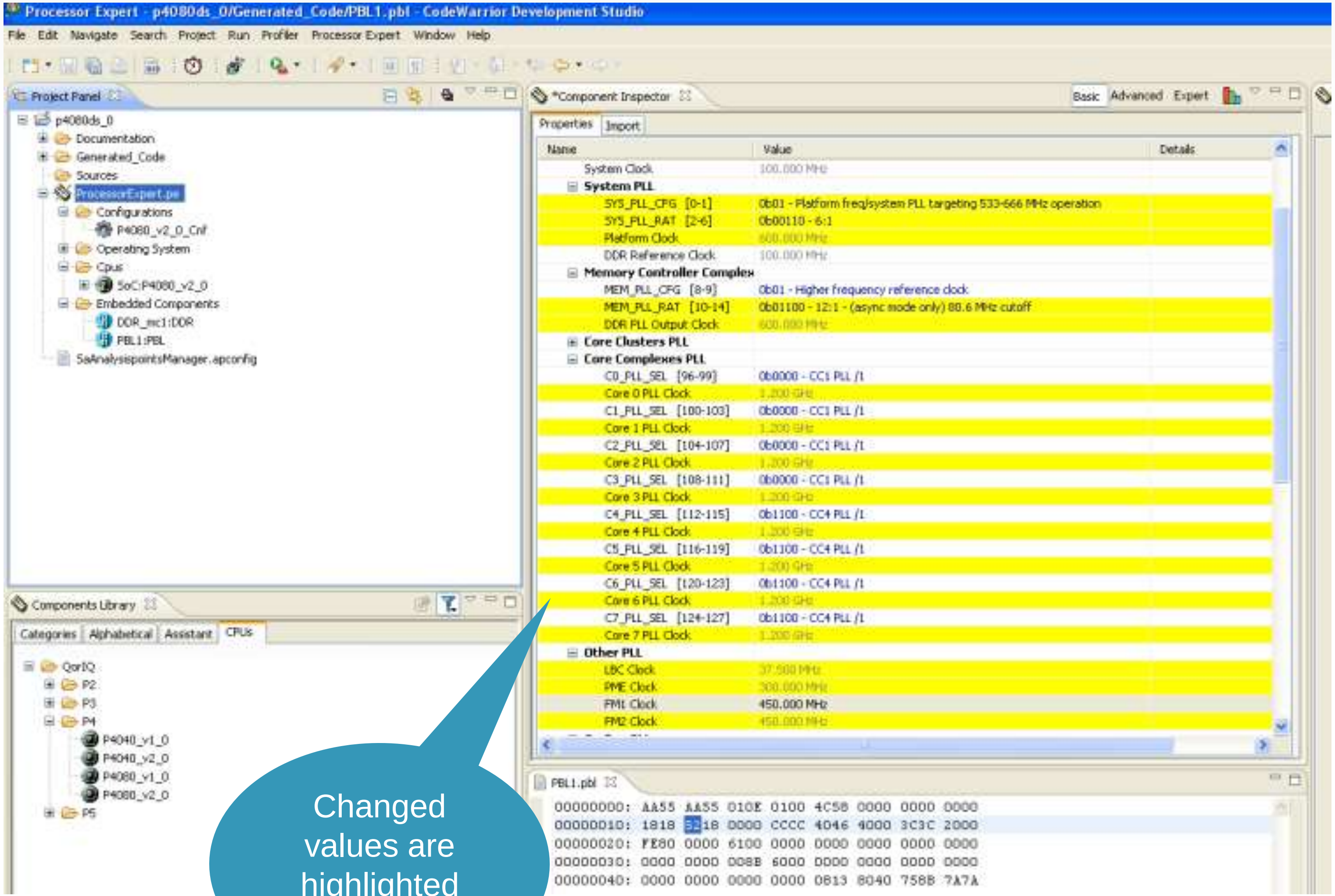
Input Format: XXD Object Dump

Input data

```
0000000: aa55 aa55 010e 0100 4c58 0000 0000 0000  .U.U....LX......
0000010: 1818 5218 0000 cccc 4046 4000 3c3c 2000  ..R.....@F@.<< .
0000020: fe80 0000 6100 0000 0000 0000 0000 0000  ....a...........
0000030: 0000 0000 008b 6000 0000 0000 0000 0000  ......`.........
0000040: 0000 0000 0000 0000 0813 8040 758b 7a7a  ...........@u.zz
```

The content of the file to be imported is displayed…

Pre-boot Loader Step 2b: See Differences Highlighted

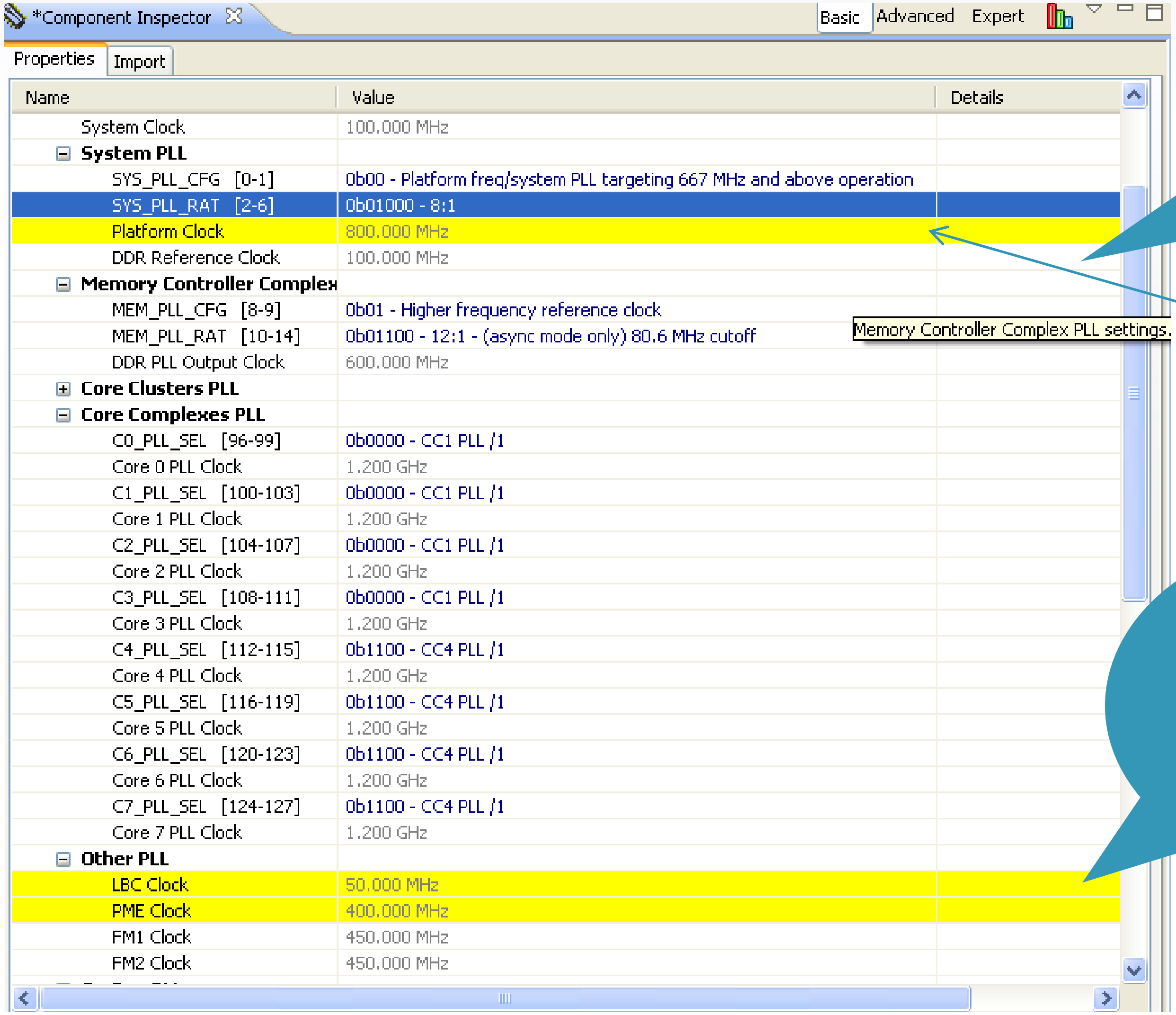


Name	Value	Details
System Clock	100.000 MHz	
System PLL		
SYS_PLL_CFG [0-1]	0b01 - Platform freq/system PLL targeting 533-666 MHz operation	
SYS_PLL_RAT [2-6]	0b00110 - 6:1	
Platform Clock	600.000 MHz	
DDR Reference Clock	100.000 MHz	
Memory Controller Complex		
MEM_PLL_CFG [8-9]	0b01 - Higher frequency reference clock	
MEM_PLL_RAT [10-14]	0b01100 - 12:1 - (async mode only) 80.6 MHz cutoff	
DDR PLL Output Clock	600.000 MHz	
Core Clusters PLL		
Core Complexes PLL		
C0_PLL_SEL [96-99]	0b0000 - CC1 PLL /1	
Core 0 PLL Clock	1.200 GHz	
C1_PLL_SEL [100-103]	0b0000 - CC1 PLL /1	
Core 1 PLL Clock	1.200 GHz	
C2_PLL_SEL [104-107]	0b0000 - CC1 PLL /1	
Core 2 PLL Clock	1.200 GHz	
C3_PLL_SEL [108-111]	0b0000 - CC1 PLL /1	
Core 3 PLL Clock	1.200 GHz	
C4_PLL_SEL [112-115]	0b1100 - CC4 PLL /1	
Core 4 PLL Clock	1.200 GHz	
C5_PLL_SEL [116-119]	0b1100 - CC4 PLL /1	
Core 5 PLL Clock	1.200 GHz	
C6_PLL_SEL [120-123]	0b1100 - CC4 PLL /1	
Core 6 PLL Clock	1.200 GHz	
C7_PLL_SEL [124-127]	0b1100 - CC4 PLL /1	
Core 7 PLL Clock	1.200 GHz	
Other PLL		
LBC Clock	37.500 MHz	
PME Clock	300.000 MHz	
FM1 Clock	450.000 MHz	
FM2 Clock	450.000 MHz	

```
00000000: AA55 AA55 010E 0100 4C58 0000 0000 0000
00000010: 1818 3218 0000 CCCC 4046 4000 3C3C 2000
00000020: FE80 0000 6100 0000 0000 0000 0000 0000
00000030: 0000 0000 008B 6000 0000 0000 0000 0000
00000040: 0000 0000 0000 0000 0813 8040 758B 7A7A
```

Changed values are highlighted after import.

Pre-boot Loader Step 3: Increase Platform Clock



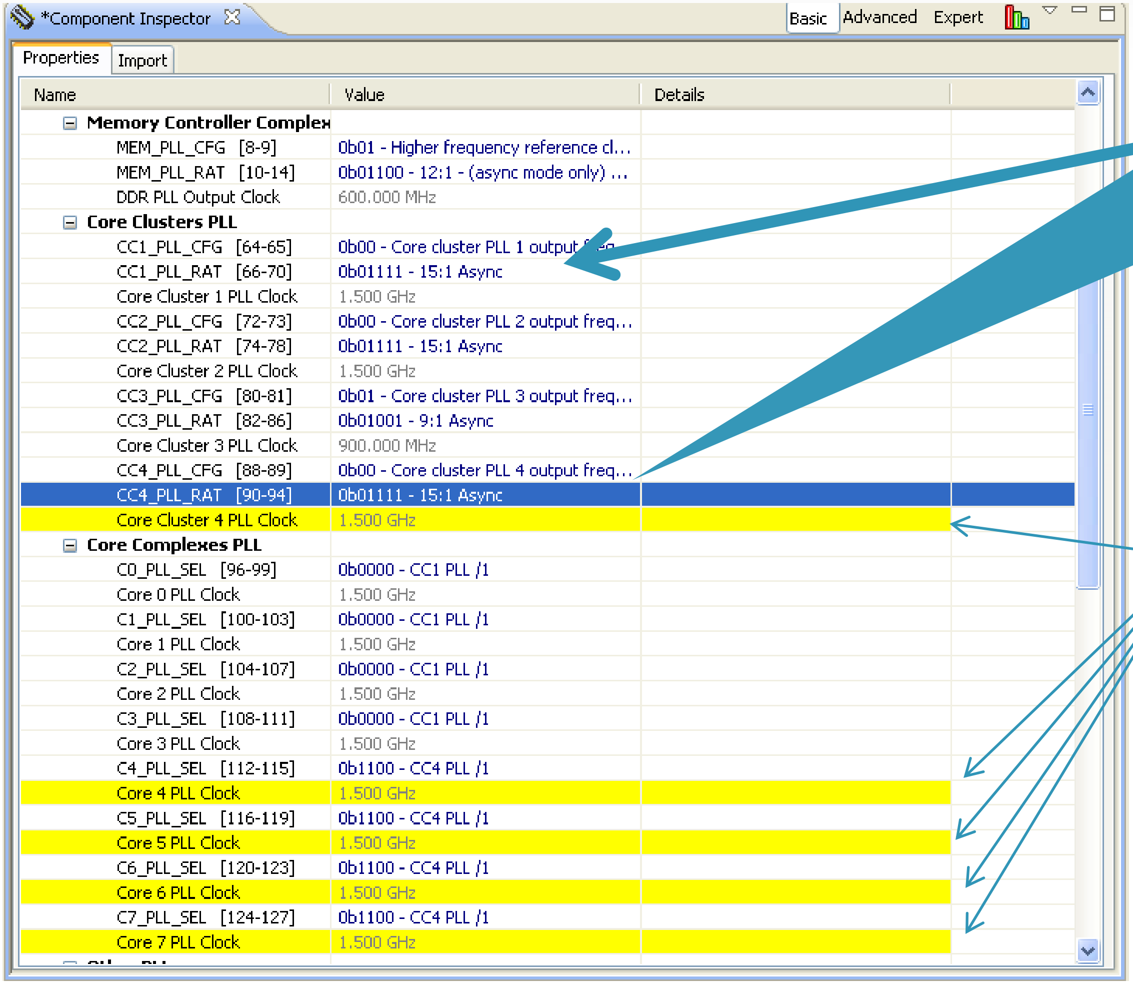
Name	Value	Details
System Clock	100.000 MHz	
⊟ **System PLL**		
SYS_PLL_CFG [0-1]	0b00 - Platform freq/system PLL targeting 667 MHz and above operation	
SYS_PLL_RAT [2-6]	0b01000 - 8:1	
Platform Clock	800.000 MHz	
DDR Reference Clock	100.000 MHz	
⊟ **Memory Controller Complex**		
MEM_PLL_CFG [8-9]	0b01 - Higher frequency reference clock	
MEM_PLL_RAT [10-14]	0b01100 - 12:1 - (async mode only) 80.6 MHz cutoff	Memory Controller Complex PLL settings.
DDR PLL Output Clock	600.000 MHz	
⊞ **Core Clusters PLL**		
⊟ **Core Complexes PLL**		
C0_PLL_SEL [96-99]	0b0000 - CC1 PLL /1	
Core 0 PLL Clock	1.200 GHz	
C1_PLL_SEL [100-103]	0b0000 - CC1 PLL /1	
Core 1 PLL Clock	1.200 GHz	
C2_PLL_SEL [104-107]	0b0000 - CC1 PLL /1	
Core 2 PLL Clock	1.200 GHz	
C3_PLL_SEL [108-111]	0b0000 - CC1 PLL /1	
Core 3 PLL Clock	1.200 GHz	
C4_PLL_SEL [112-115]	0b1100 - CC4 PLL /1	
Core 4 PLL Clock	1.200 GHz	
C5_PLL_SEL [116-119]	0b1100 - CC4 PLL /1	
Core 5 PLL Clock	1.200 GHz	
C6_PLL_SEL [120-123]	0b1100 - CC4 PLL /1	
Core 6 PLL Clock	1.200 GHz	
C7_PLL_SEL [124-127]	0b1100 - CC4 PLL /1	
Core 7 PLL Clock	1.200 GHz	
⊟ **Other PLL**		
LBC Clock	50.000 MHz	
PME Clock	400.000 MHz	
FM1 Clock	450.000 MHz	
FM2 Clock	450.000 MHz	

Changing the ratio to 8:1 makes platform clock 800 MHz

600 Mhz → 800 Mhz

And also changes the LBC and PME clocks

Pre-boot Loader: Change Clock Ratios



Name	Value	Details
⊟ **Memory Controller Complex**		
MEM_PLL_CFG [8-9]	0b01 - Higher frequency reference cl...	
MEM_PLL_RAT [10-14]	0b01100 - 12:1 - (async mode only) ...	
DDR PLL Output Clock	600.000 MHz	
⊟ **Core Clusters PLL**		
CC1_PLL_CFG [64-65]	0b00 - Core cluster PLL 1 output freq	
CC1_PLL_RAT [66-70]	0b01111 - 15:1 Async	
Core Cluster 1 PLL Clock	1.500 GHz	
CC2_PLL_CFG [72-73]	0b00 - Core cluster PLL 2 output freq...	
CC2_PLL_RAT [74-78]	0b01111 - 15:1 Async	
Core Cluster 2 PLL Clock	1.500 GHz	
CC3_PLL_CFG [80-81]	0b01 - Core cluster PLL 3 output freq...	
CC3_PLL_RAT [82-86]	0b01001 - 9:1 Async	
Core Cluster 3 PLL Clock	900.000 MHz	
CC4_PLL_CFG [88-89]	0b00 - Core cluster PLL 4 output freq...	
CC4_PLL_RAT [90-94]	0b01111 - 15:1 Async	
Core Cluster 4 PLL Clock	1.500 GHz	
⊟ **Core Complexes PLL**		
C0_PLL_SEL [96-99]	0b0000 - CC1 PLL /1	
Core 0 PLL Clock	1.500 GHz	
C1_PLL_SEL [100-103]	0b0000 - CC1 PLL /1	
Core 1 PLL Clock	1.500 GHz	
C2_PLL_SEL [104-107]	0b0000 - CC1 PLL /1	
Core 2 PLL Clock	1.500 GHz	
C3_PLL_SEL [108-111]	0b0000 - CC1 PLL /1	
Core 3 PLL Clock	1.500 GHz	
C4_PLL_SEL [112-115]	0b1100 - CC4 PLL /1	
Core 4 PLL Clock	1.500 GHz	
C5_PLL_SEL [116-119]	0b1100 - CC4 PLL /1	
Core 5 PLL Clock	1.500 GHz	
C6_PLL_SEL [120-123]	0b1100 - CC4 PLL /1	
Core 6 PLL Clock	1.500 GHz	
C7_PLL_SEL [124-127]	0b1100 - CC4 PLL /1	
Core 7 PLL Clock	1.500 GHz	

Changing the CC1/CC4 PLL_RAT for cores to 15:1 sets core clocks to 1.5 Ghz

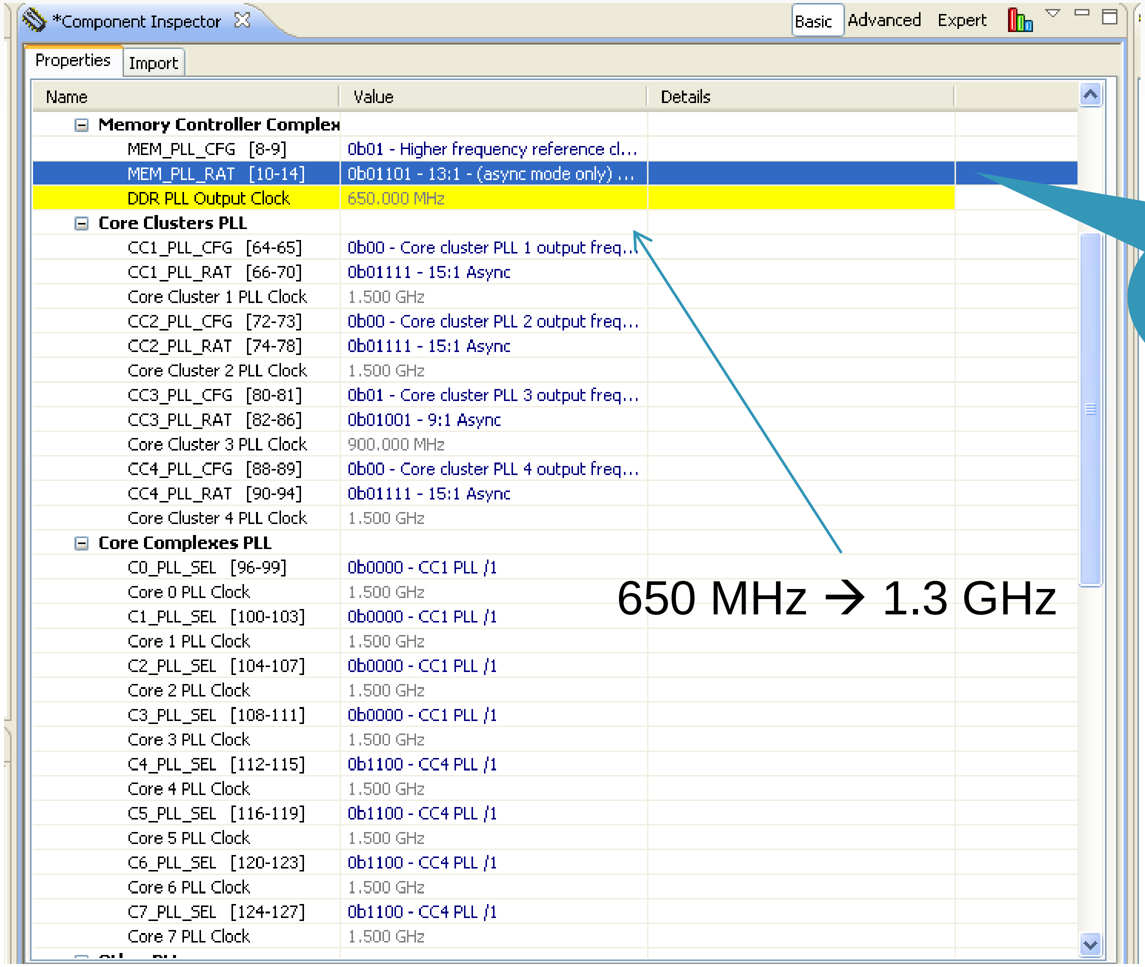
1.2 Ghz → 1.5 Ghz

CC1 clocks core 0-3
CC4 clocks core 4-7

Note: Please change CC2 also, even if unused, to reach exact high rcw

freescale™ 飞思卡尔

Pre-boot Loader: Increase DDR Output Clock

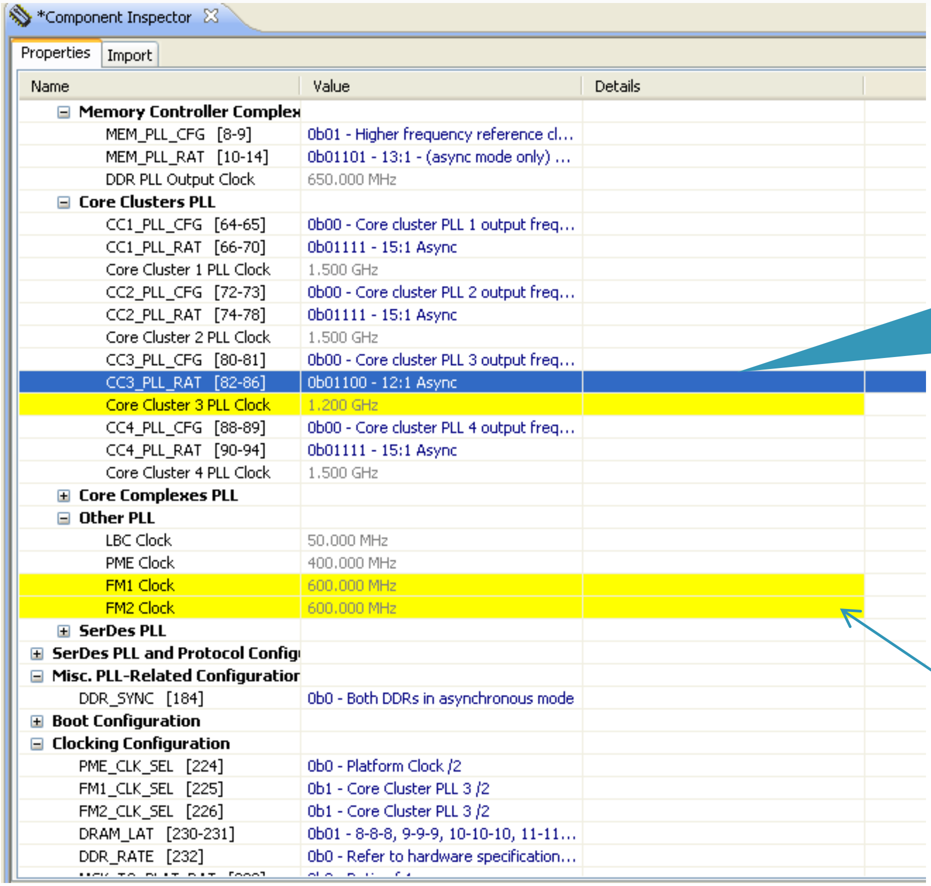


Name	Value	Details
⊟ **Memory Controller Complex**		
MEM_PLL_CFG [8-9]	0b01 - Higher frequency reference cl...	
MEM_PLL_RAT [10-14]	0b01101 - 13:1 - (async mode only) ...	
DDR PLL Output Clock	650.000 MHz	
⊟ **Core Clusters PLL**		
CC1_PLL_CFG [64-65]	0b00 - Core cluster PLL 1 output freq...	
CC1_PLL_RAT [66-70]	0b01111 - 15:1 Async	
Core Cluster 1 PLL Clock	1.500 GHz	
CC2_PLL_CFG [72-73]	0b00 - Core cluster PLL 2 output freq...	
CC2_PLL_RAT [74-78]	0b01111 - 15:1 Async	
Core Cluster 2 PLL Clock	1.500 GHz	
CC3_PLL_CFG [80-81]	0b01 - Core cluster PLL 3 output freq...	
CC3_PLL_RAT [82-86]	0b01001 - 9:1 Async	
Core Cluster 3 PLL Clock	900.000 MHz	
CC4_PLL_CFG [88-89]	0b00 - Core cluster PLL 4 output freq...	
CC4_PLL_RAT [90-94]	0b01111 - 15:1 Async	
Core Cluster 4 PLL Clock	1.500 GHz	
⊟ **Core Complexes PLL**		
C0_PLL_SEL [96-99]	0b0000 - CC1 PLL /1	
Core 0 PLL Clock	1.500 GHz	
C1_PLL_SEL [100-103]	0b0000 - CC1 PLL /1	
Core 1 PLL Clock	1.500 GHz	
C2_PLL_SEL [104-107]	0b0000 - CC1 PLL /1	
Core 2 PLL Clock	1.500 GHz	
C3_PLL_SEL [108-111]	0b0000 - CC1 PLL /1	
Core 3 PLL Clock	1.500 GHz	
C4_PLL_SEL [112-115]	0b1100 - CC4 PLL /1	
Core 4 PLL Clock	1.500 GHz	
C5_PLL_SEL [116-119]	0b1100 - CC4 PLL /1	
Core 5 PLL Clock	1.500 GHz	
C6_PLL_SEL [120-123]	0b1100 - CC4 PLL /1	
Core 6 PLL Clock	1.500 GHz	
C7_PLL_SEL [124-127]	0b1100 - CC4 PLL /1	
Core 7 PLL Clock	1.500 GHz	

650 MHz → 1.3 GHz

Set MEM_PLL_RAT to 13:1

Pre-boot Loader Step 6: Increase FMAN1/2 Speed

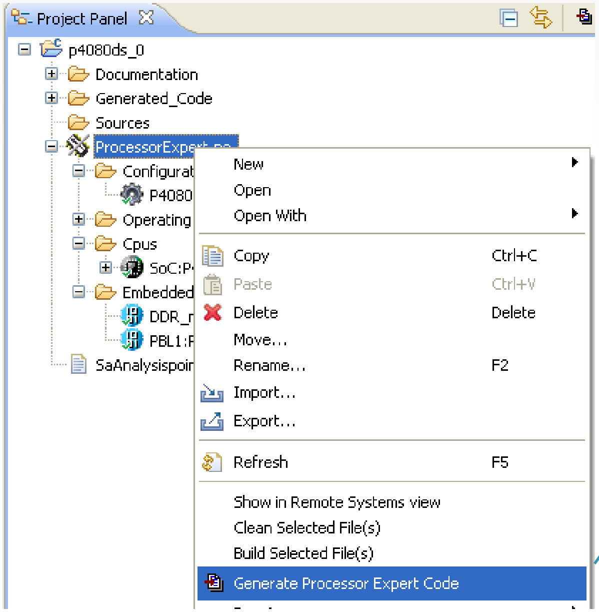


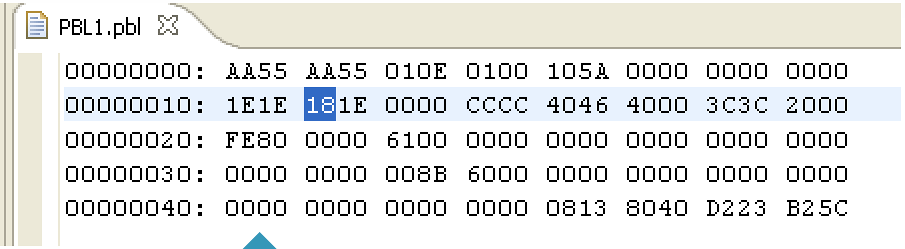
Name	Value	Details
⊟ Memory Controller Complex		
MEM_PLL_CFG [8-9]	0b01 - Higher frequency reference cl...	
MEM_PLL_RAT [10-14]	0b01101 - 13:1 - (async mode only) ...	
DDR PLL Output Clock	650.000 MHz	
⊟ Core Clusters PLL		
CC1_PLL_CFG [64-65]	0b00 - Core cluster PLL 1 output freq...	
CC1_PLL_RAT [66-70]	0b01111 - 15:1 Async	
Core Cluster 1 PLL Clock	1.500 GHz	
CC2_PLL_CFG [72-73]	0b00 - Core cluster PLL 2 output freq...	
CC2_PLL_RAT [74-78]	0b01111 - 15:1 Async	
Core Cluster 2 PLL Clock	1.500 GHz	
CC3_PLL_CFG [80-81]	0b00 - Core cluster PLL 3 output freq...	
CC3_PLL_RAT [82-86]	0b01100 - 12:1 Async	
Core Cluster 3 PLL Clock	1.200 GHz	
CC4_PLL_CFG [88-89]	0b00 - Core cluster PLL 4 output freq...	
CC4_PLL_RAT [90-94]	0b01111 - 15:1 Async	
Core Cluster 4 PLL Clock	1.500 GHz	
⊞ Core Complexes PLL		
⊟ Other PLL		
LBC Clock	50.000 MHz	
PME Clock	400.000 MHz	
FM1 Clock	600.000 MHz	
FM2 Clock	600.000 MHz	
⊞ SerDes PLL		
⊞ SerDes PLL and Protocol Config		
⊟ Misc. PLL-Related Configuration		
DDR_SYNC [184]	0b0 - Both DDRs in asynchronous mode	
⊞ Boot Configuration		
⊟ Clocking Configuration		
PME_CLK_SEL [224]	0b0 - Platform Clock /2	
FM1_CLK_SEL [225]	0b1 - Core Cluster PLL 3 /2	
FM2_CLK_SEL [226]	0b1 - Core Cluster PLL 3 /2	
DRAM_LAT [230-231]	0b01 - 8-8-8, 9-9-9, 10-10-10, 11-11...	
DDR_RATE [232]	0b0 - Refer to hardware specification...	

Set CC3_PLL_RAT to 12:1 to set Frame Manager clocks

450 MHz → 600 MHz

Pre-boot Loader Step 7 – Generate Upgraded RCW

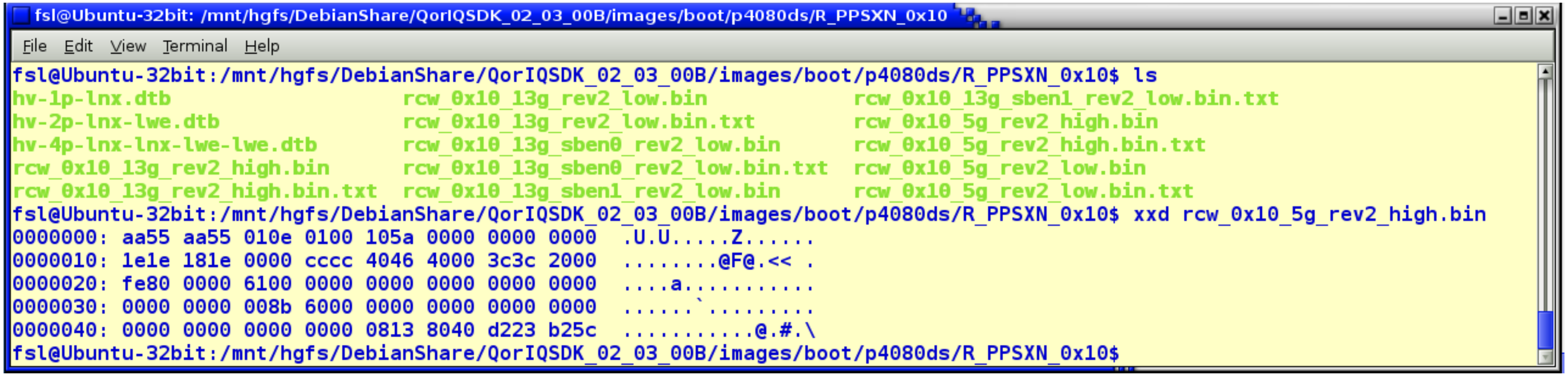




PBL1.pbl

```
00000000: AA55 AA55 010E 0100 105A 0000 0000 0000
00000010: 1E1E 181E 0000 CCCC 4046 4000 3C3C 2000
00000020: FE80 0000 6100 0000 0000 0000 0000 0000
00000030: 0000 0000 008B 6000 0000 0000 0000 0000
00000040: 0000 0000 0000 0000 0813 8040 D223 B25C
```

Generating the PBL provides the same RCW as the high-speed RCW seen in the SDK

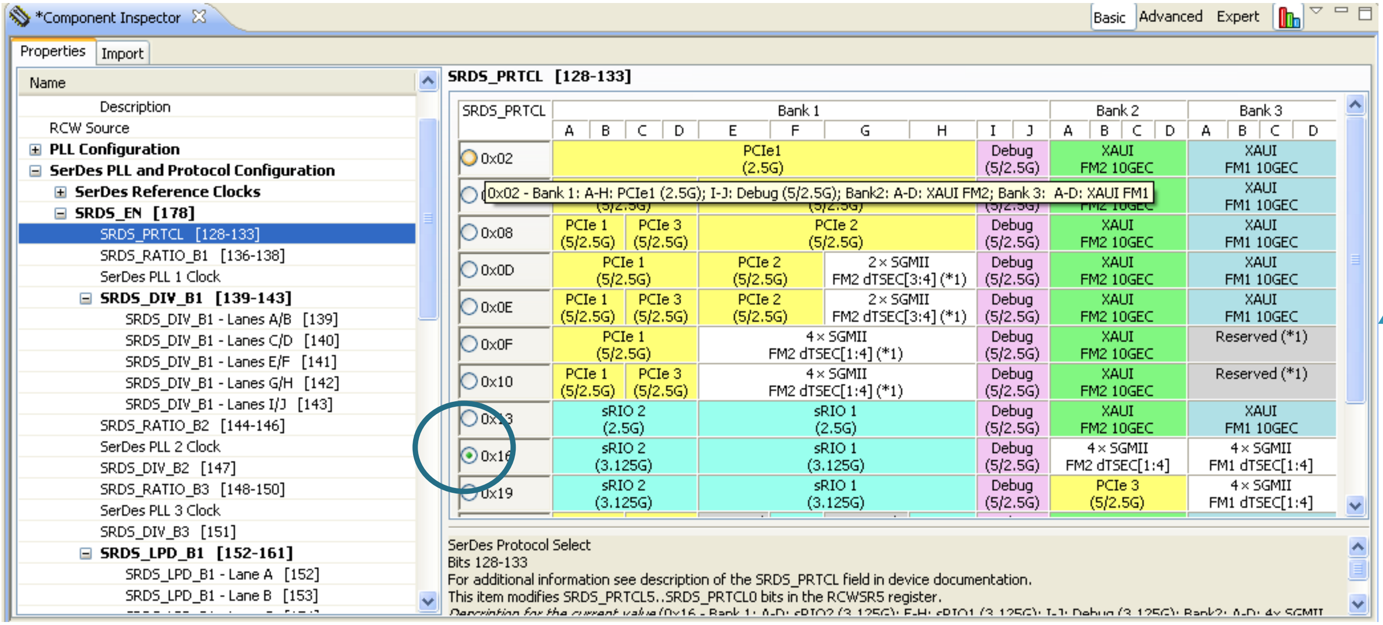


fsl@Ubuntu-32bit: /mnt/hgfs/DebianShare/QorIQSDK_02_03_00B/images/boot/p4080ds/R_PPSXN_0x10

```
fsl@Ubuntu-32bit:/mnt/hgfs/DebianShare/QorIQSDK_02_03_00B/images/boot/p4080ds/R_PPSXN_0x10$ ls
hv-1p-lnx.dtb                rcw_0x10_13g_rev2_low.bin          rcw_0x10_13g_sben1_rev2_low.bin.txt
hv-2p-lnx-lwe.dtb            rcw_0x10_13g_rev2_low.bin.txt      rcw_0x10_5g_rev2_high.bin
hv-4p-lnx-lnx-lwe-lwe.dtb    rcw_0x10_13g_sben0_rev2_low.bin    rcw_0x10_5g_rev2_high.bin.txt
rcw_0x10_13g_rev2_high.bin   rcw_0x10_13g_sben0_rev2_low.bin.txt rcw_0x10_5g_rev2_low.bin
rcw_0x10_13g_rev2_high.bin.txt rcw_0x10_13g_sben1_rev2_low.bin  rcw_0x10_5g_rev2_low.bin.txt
fsl@Ubuntu-32bit:/mnt/hgfs/DebianShare/QorIQSDK_02_03_00B/images/boot/p4080ds/R_PPSXN_0x10$ xxd rcw_0x10_5g_rev2_high.bin
0000000: aa55 aa55 010e 0100 105a 0000 0000 0000  .U.U.....Z......
0000010: 1e1e 181e 0000 cccc 4046 4000 3c3c 2000  ........@F@.<< .
0000020: fe80 0000 6100 0000 0000 0000 0000 0000  ....a...........
0000030: 0000 0000 008b 6000 0000 0000 0000 0000  ......`.........
0000040: 0000 0000 0000 0000 0813 8040 d223 b25c  ...........@.#.\
fsl@Ubuntu-32bit:/mnt/hgfs/DebianShare/QorIQSDK_02_03_00B/images/boot/p4080ds/R_PPSXN_0x10$
```

Pre-boot Loader Step 8: Change Serdes for 2 x 4 SGMII

- Starting from previous rcw_0x10_5g_rev2_high.bin.txt config, let’s adapt Serdes protocol configuration to allow 2x4 SGMII



Description
RCW Source
⊞ PLL Configuration
⊟ SerDes PLL and Protocol Configuration
⊞ SerDes Reference Clocks
⊟ SRDS_EN [178]
SRDS_PRTCL [128-133]
SRDS_RATIO_B1 [136-138]
SerDes PLL 1 Clock
⊟ SRDS_DIV_B1 [139-143]
SRDS_DIV_B1 - Lanes A/B [139]
SRDS_DIV_B1 - Lanes C/D [140]
SRDS_DIV_B1 - Lanes E/F [141]
SRDS_DIV_B1 - Lanes G/H [142]
SRDS_DIV_B1 - Lanes I/J [143]
SRDS_RATIO_B2 [144-146]
SerDes PLL 2 Clock
SRDS_DIV_B2 [147]
SRDS_RATIO_B3 [148-150]
SerDes PLL 3 Clock
SRDS_DIV_B3 [151]
⊟ SRDS_LPD_B1 [152-161]
SRDS_LPD_B1 - Lane A [152]
SRDS_LPD_B1 - Lane B [153]

SRDS_PRTCL [128-133]

SRDS_PRTCL	Bank 1										Bank 2				Bank 3			
	A	B	C	D	E	F	G	H	I	J	A	B	C	D	A	B	C	D
○ 0x02	PCIe1 (2.5G)								Debug (5/2.5G)		XAUI FM2 10GEC				XAUI FM1 10GEC			
○	(5/2.5G)				(5/2.5G)				(5/2.5G)		FM2 10GEC				XAUI FM1 10GEC			
○ 0x08	PCIe 1 (5/2.5G)		PCIe 3 (5/2.5G)		PCIe 2 (5/2.5G)				Debug (5/2.5G)		XAUI FM2 10GEC				XAUI FM1 10GEC			
○ 0x0D	PCIe 1 (5/2.5G)				PCIe 2 (5/2.5G)		2 × SGMII FM2 dTSEC[3:4] (*1)		Debug (5/2.5G)		XAUI FM2 10GEC				XAUI FM1 10GEC			
○ 0x0E	PCIe 1 (5/2.5G)		PCIe 3 (5/2.5G)		PCIe 2 (5/2.5G)		2 × SGMII FM2 dTSEC[3:4] (*1)		Debug (5/2.5G)		XAUI FM2 10GEC				XAUI FM1 10GEC			
○ 0x0F	PCIe 1 (5/2.5G)				4 × SGMII FM2 dTSEC[1:4] (*1)				Debug (5/2.5G)		XAUI FM2 10GEC				Reserved (*1)			
○ 0x10	PCIe 1 (5/2.5G)		PCIe 3 (5/2.5G)		4 × SGMII FM2 dTSEC[1:4] (*1)				Debug (5/2.5G)		XAUI FM2 10GEC				Reserved (*1)			
○ 0x13	sRIO 2 (2.5G)				sRIO 1 (2.5G)				Debug (5/2.5G)		XAUI FM2 10GEC				XAUI FM1 10GEC			
● 0x16	sRIO 2 (3.125G)				sRIO 1 (3.125G)				Debug (5/2.5G)		4 × SGMII FM2 dTSEC[1:4]				4 × SGMII FM1 dTSEC[1:4]			
○ 0x19	sRIO 2 (3.125G)				sRIO 1 (3.125G)				Debug (5/2.5G)		PCIe 3 (5/2.5G)				4 × SGMII FM1 dTSEC[1:4]			

0x02 - Bank 1: A-H: PCIe1 (2.5G); I-J: Debug (5/2.5G); Bank2: A-D: XAUI FM2; Bank 3: A-D: XAUI FM1

SerDes Protocol Select
Bits 128-133
For additional information see description of the SRDS_PRTCL field in device documentation.
This item modifies SRDS_PRTCL5..SRDS_PRTCL0 bits in the RCWSR5 register.
Description for the current value (0x16 - Bank 1: A-D: sRIO2 (3.125G); E-H: sRIO1 (3.125G); I-J: Debug (3.125G); Bank2: A-D: 4x SGMII

SRDS_PRTCL
0x10 to 0x16

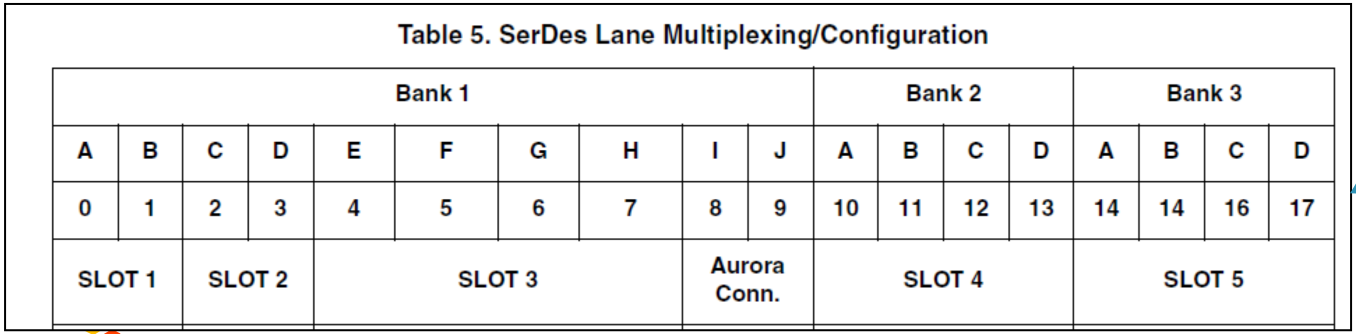
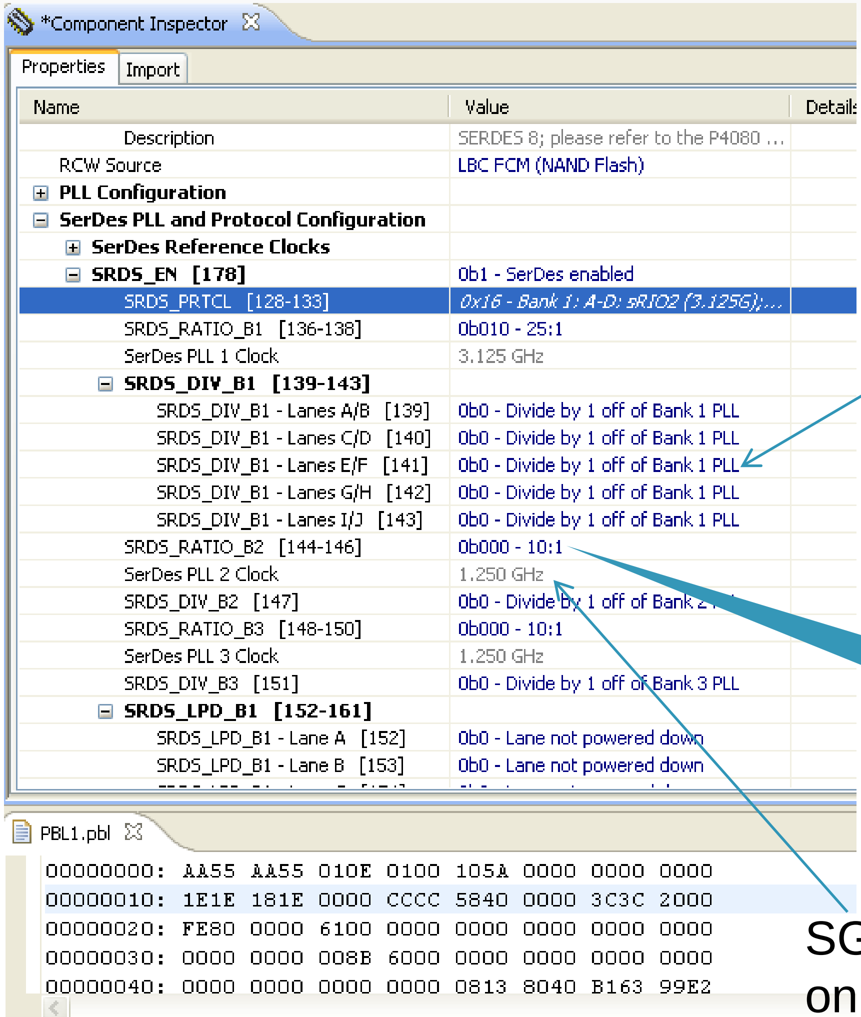


Table 5. SerDes Lane Multiplexing/Configuration

Bank 1										Bank 2				Bank 3			
A	B	C	D	E	F	G	H	I	J	A	B	C	D	A	B	C	D
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	14	16	17
SLOT 1		SLOT 2		SLOT 3				Aurora Conn.		SLOT 4				SLOT 5			

P4080DS slots
as per
user manual

Pre-boot Loader Step 9: Adapt Serdes Clocks



Name	Value	Details
Description	SERDES 8; please refer to the P4080 ...	
RCW Source	LBC FCM (NAND Flash)	
⊞ PLL Configuration		
⊟ SerDes PLL and Protocol Configuration		
⊞ SerDes Reference Clocks		
⊟ SRDS_EN [178]	0b1 - SerDes enabled	
SRDS_PRTCL [128-133]	*0x16 - Bank 1: A-D: sRIO2 (3.125G);...*	
SRDS_RATIO_B1 [136-138]	0b010 - 25:1	
SerDes PLL 1 Clock	3.125 GHz	
⊟ SRDS_DIV_B1 [139-143]		
SRDS_DIV_B1 - Lanes A/B [139]	0b0 - Divide by 1 off of Bank 1 PLL	
SRDS_DIV_B1 - Lanes C/D [140]	0b0 - Divide by 1 off of Bank 1 PLL	
SRDS_DIV_B1 - Lanes E/F [141]	0b0 - Divide by 1 off of Bank 1 PLL	
SRDS_DIV_B1 - Lanes G/H [142]	0b0 - Divide by 1 off of Bank 1 PLL	
SRDS_DIV_B1 - Lanes I/J [143]	0b0 - Divide by 1 off of Bank 1 PLL	
SRDS_RATIO_B2 [144-146]	0b000 - 10:1	
SerDes PLL 2 Clock	1.250 GHz	
SRDS_DIV_B2 [147]	0b0 - Divide by 1 off of Bank 2	
SRDS_RATIO_B3 [148-150]	0b000 - 10:1	
SerDes PLL 3 Clock	1.250 GHz	
SRDS_DIV_B3 [151]	0b0 - Divide by 1 off of Bank 3 PLL	
⊟ SRDS_LPD_B1 [152-161]		
SRDS_LPD_B1 - Lane A [152]	0b0 - Lane not powered down	
SRDS_LPD_B1 - Lane B [153]	0b0 - Lane not powered down	

PBL1.pbl

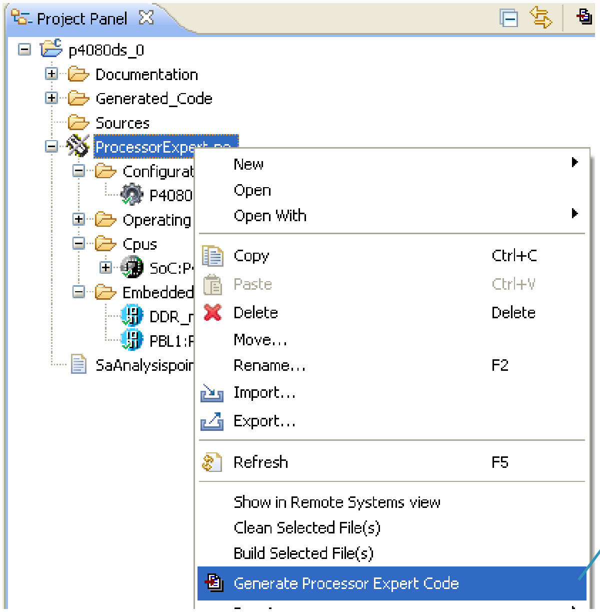
```
00000000: AA55 AA55 010E 0100 105A 0000 0000 0000
00000010: 1E1E 181E 0000 CCCC 5840 0000 3C3C 2000
00000020: FE80 0000 6100 0000 0000 0000 0000 0000
00000030: 0000 0000 008B 6000 0000 0000 0000 0000
00000040: 0000 0000 0000 0000 0813 8040 B163 99E2
```

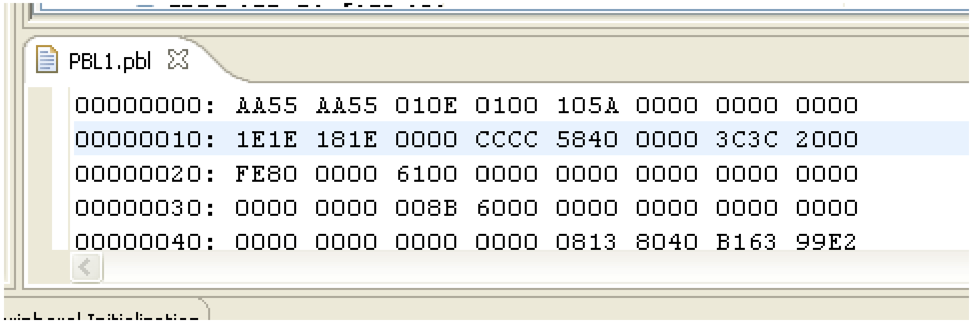
SRIO requires
3.125 Ghz on Bank1 lane
E,F,G,H.
Set to Divide by 1.

Set
SRDS_RATIO_B2
to 10:1

SGMII requires **1.250 GHz**
on Bank2 and Bank3

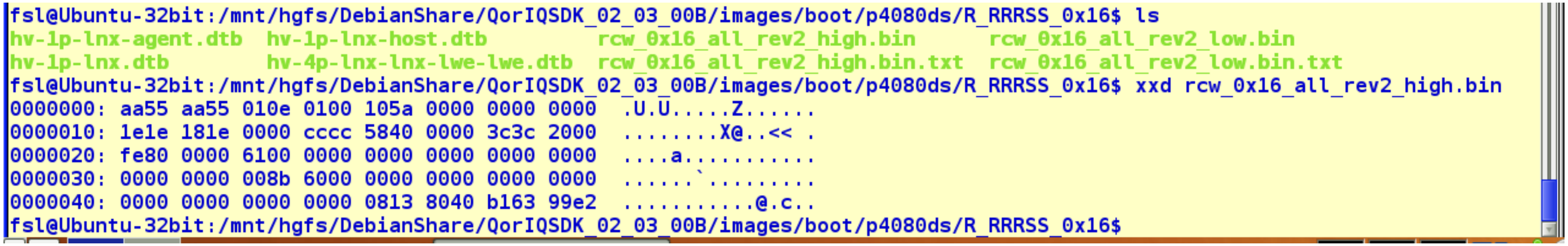
PBL Step 9: Generate Pre-boot Loader and Compare with the RCW Provided in the SDK




PBL1.pbl

```
00000000: AA55 AA55 010E 0100 105A 0000 0000 0000
00000010: 1E1E 181E 0000 CCCC 5840 0000 3C3C 2000
00000020: FE80 0000 6100 0000 0000 0000 0000 0000
00000030: 0000 0000 008B 6000 0000 0000 0000 0000
00000040: 0000 0000 0000 0000 0813 8040 B163 99E2
```

Generating the PBL now provides us the same RCW than the RCW seen in the SDK2.3



```
fsl@Ubuntu-32bit:/mnt/hgfs/DebianShare/QorIQSDK_02_03_00B/images/boot/p4080ds/R_RRRSS_0x16$ ls
hv-1p-lnx-agent.dtb  hv-1p-lnx-host.dtb         rcw_0x16_all_rev2_high.bin      rcw_0x16_all_rev2_low.bin
hv-1p-lnx.dtb        hv-4p-lnx-lnx-lwe-lwe.dtb  rcw_0x16_all_rev2_high.bin.txt  rcw_0x16_all_rev2_low.bin.txt
fsl@Ubuntu-32bit:/mnt/hgfs/DebianShare/QorIQSDK_02_03_00B/images/boot/p4080ds/R_RRRSS_0x16$ xxd rcw_0x16_all_rev2_high.bin
0000000: aa55 aa55 010e 0100 105a 0000 0000 0000  .U.U.....Z......
0000010: 1e1e 181e 0000 cccc 5840 0000 3c3c 2000  ........X@..<< .
0000020: fe80 0000 6100 0000 0000 0000 0000 0000  ....a...........
0000030: 0000 0000 008b 6000 0000 0000 0000 0000  ......`.........
0000040: 0000 0000 0000 0000 0813 8040 b163 99e2  ...........@.c..
fsl@Ubuntu-32bit:/mnt/hgfs/DebianShare/QorIQSDK_02_03_00B/images/boot/p4080ds/R_RRRSS_0x16$
```

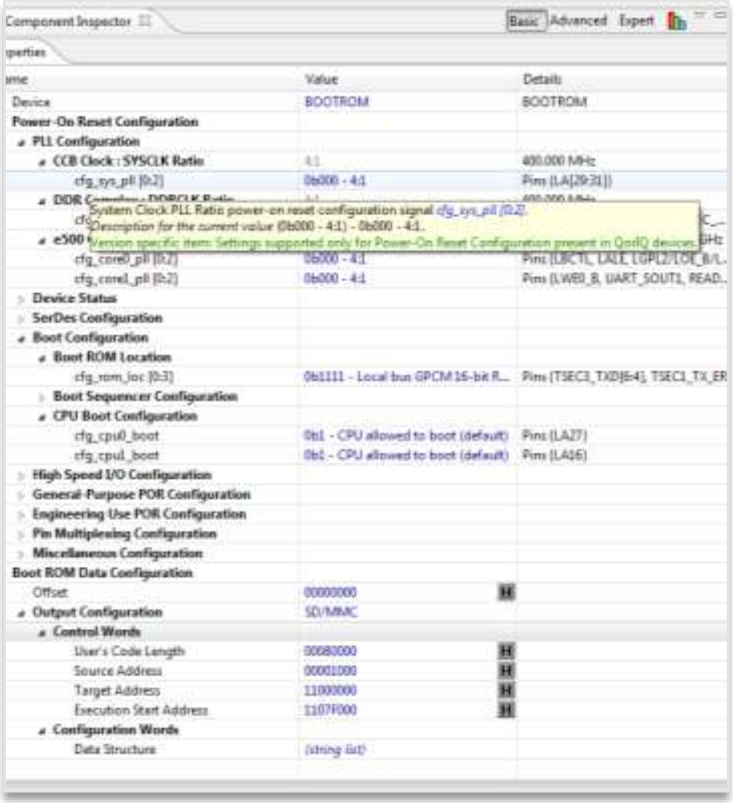
BOOTROM Configuration

Lab 3: Custom Hardware Configuration

BOOTROM Configuration tool

Overview

- Helps in Power-on Reset (POR) device configuration by definition of values for POR configuration signals and generates overview report including POR value required for each POR configuration pin (device specific)
- Helps in building a configuration file to be used in a boot image creation process (*boot_format*) for various memory interfaces

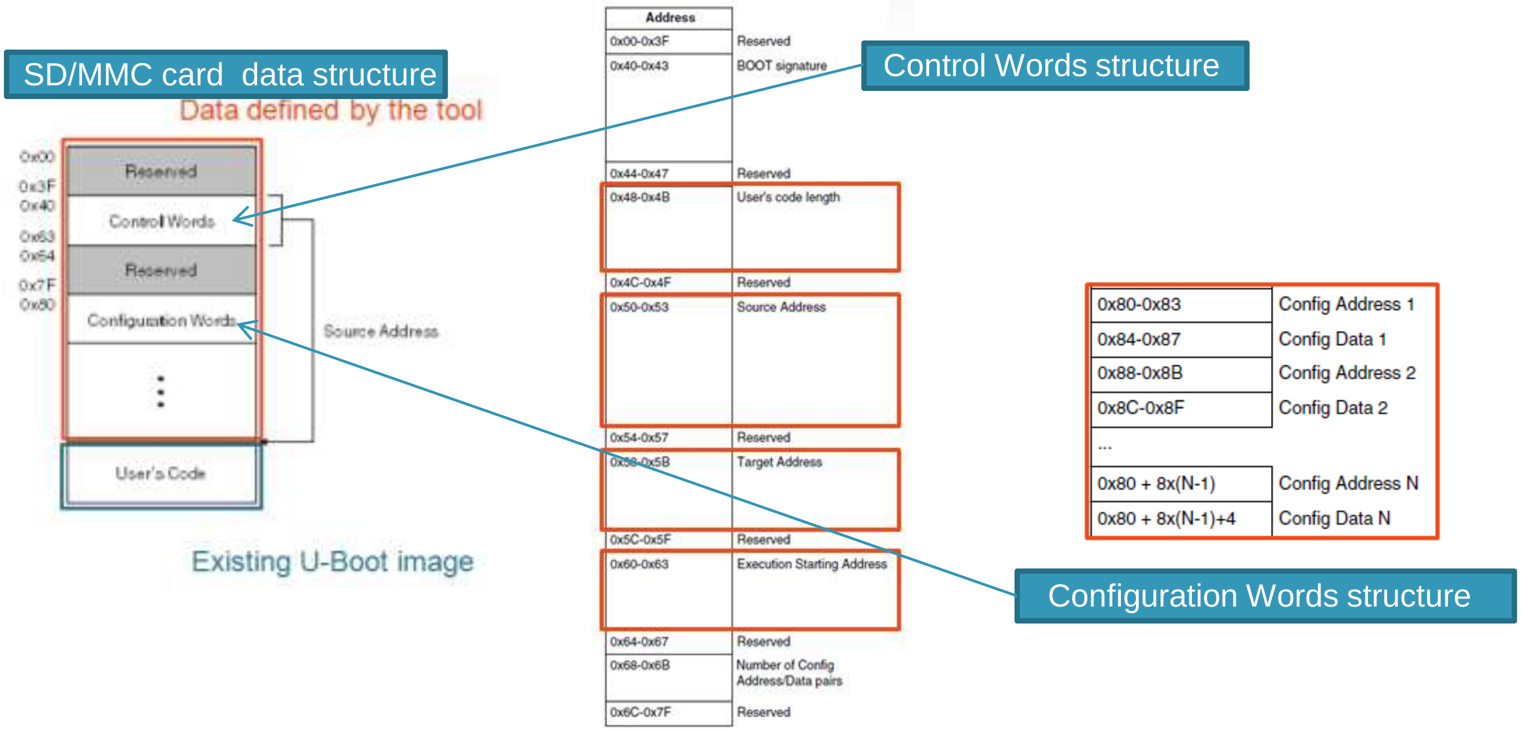


me	Value		Details
Device	BOOTROM		BOOTROM
Power-On Reset Configuration			
PLL Configuration			
CCB Clock : SYSCLK Ratio	4:1		400.000 MHz
cfg_sys_pll [0:2]	0b000 - 4:1		Pins [LA[29:31]]
DDR C	4:1		400.000 MHz
cfg			C_
e500			GHz
cfg_core0_pll [0:2]	0b000 - 4:1		Pins [LBCTL, LALE, LGPL2/LOE_B/L...
cfg_core1_pll [0:2]	0b000 - 4:1		Pins [LWE0_B, UART_SOUT1, READ...
Device Status			
SerDes Configuration			
Boot Configuration			
Boot ROM Location			
cfg_rom_loc [0:3]	0b1111 - Local bus GPCM 16-bit R...		Pins [TSEC3_TXD[6:4], TSEC1_TX_ER
Boot Sequencer Configuration			
CPU Boot Configuration			
cfg_cpu0_boot	0b1 - CPU allowed to boot (default)		Pins [LA27]
cfg_cpu1_boot	0b1 - CPU allowed to boot (default)		Pins [LA16]
High Speed I/O Configuration			
General-Purpose POR Configuration			
Engineering Use POR Configuration			
Pin Multiplexing Configuration			
Miscellaneous Configuration			
Boot ROM Data Configuration			
Offset	00000000	H	
Output Configuration	SD/MMC		
Control Words			
User's Code Length	00080000	H	
Source Address	00001000	H	
Target Address	11000000	H	
Execution Start Address	1107F000	H	
Configuration Words			
Data Structure	(string list)		

System Clock PLL Ratio power-on reset configuration signal *cfg_sys_pll [0:2]*.
Description for the current value (0b000 - 4:1) - 0b000 - 4:1.
Version specific item: Settings supported only for Power-On Reset Configuration present in QorIQ devices.

BOOTROM Configuration tool

- *Power-on Reset (POR) configuration signal value –* binary value 0b0 represents a signal pulled down to GND and a value 0b1 represents a signal pulled up to Vdd, regardless of the sense of the functional signal name on the signal
- *Configuration file –* data structure including control and configuration words, two parts that needs to be put together with user's code (typically u-boot image) to create booting image for a device



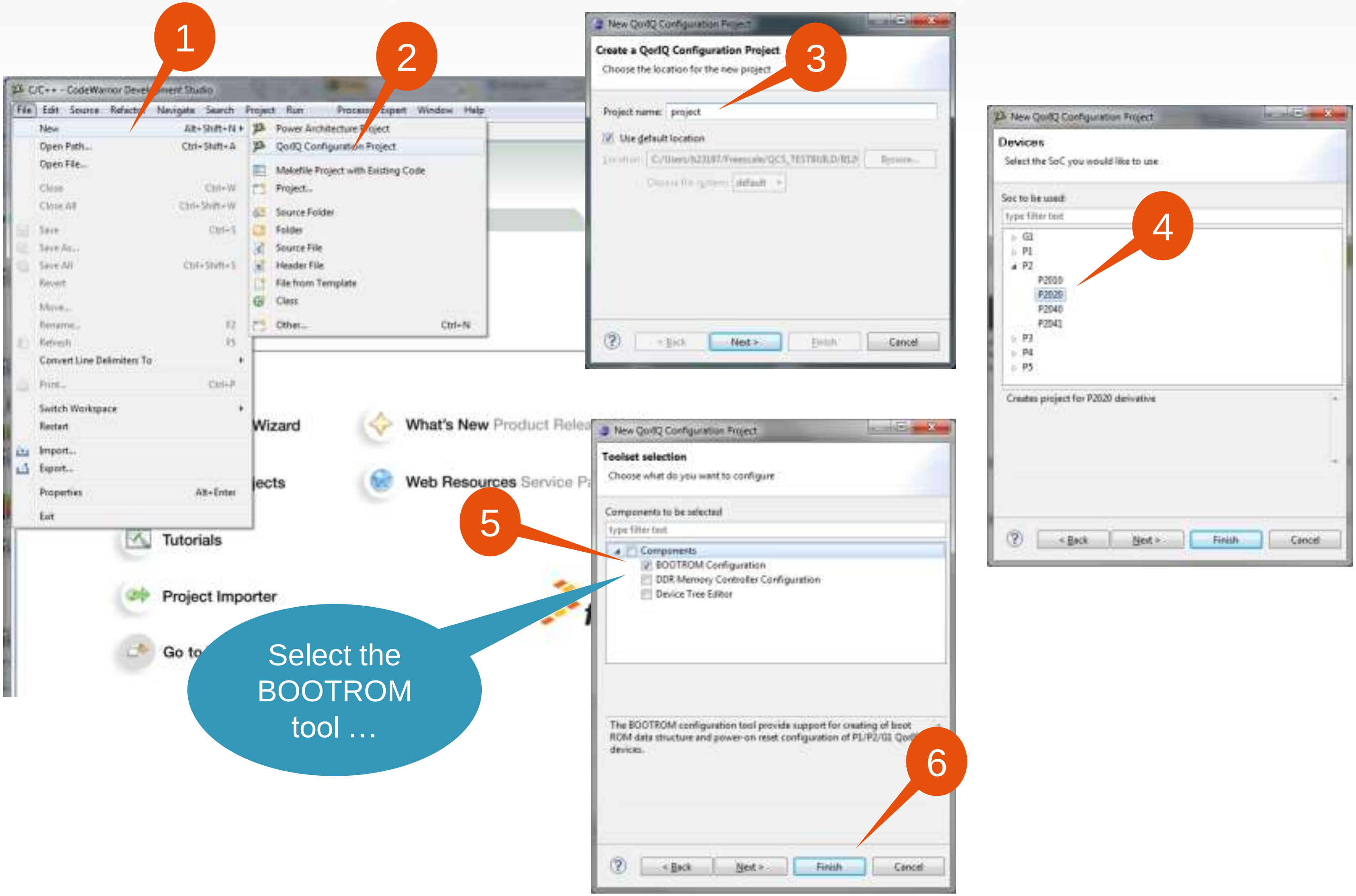
Address	
0x00-0x3F	Reserved
0x40-0x43	BOOT signature
0x44-0x47	Reserved
0x48-0x4B	User's code length
0x4C-0x4F	Reserved
0x50-0x53	Source Address
0x54-0x57	Reserved
0x58-0x5B	Target Address
0x5C-0x5F	Reserved
0x60-0x63	Execution Starting Address
0x64-0x67	Reserved
0x68-0x6B	Number of Config Address/Data pairs
0x6C-0x7F	Reserved

0x80-0x83	Config Address 1
0x84-0x87	Config Data 1
0x88-0x8B	Config Address 2
0x8C-0x8F	Config Data 2
...	
0x80 + 8x(N-1)	Config Address N
0x80 + 8x(N-1)+4	Config Data N

BOOTROM Hands-on

- Step 1: Create configuration project for the P2020 device
- Step 2: Use BOOTROM tool to review & change settings
  - Details:
    - Change CCB and core clocking – 600 / 900 MHz
    - I/O port configuration – required interfaces: PCIe, SRIO and eTSEC
    - boot location – set to boot from SD/MMC card
- Step 3: Observe Power-on Reset overview details
    - Generated overview report – txt/HTML
- Step 4: Use BOOTROM tool to prepare configuration data file for boot image processing
    - Generated configuration file (.dat) – support of booting from on-chip ROM (e.g. eSDHC or eSPI), base data file for boot image processing using external booting utility application
- Step 5: Usage of configuration data file with external booting utility application (*using boot_format - this application is a part of BSP release*)

BOOTROM Step 1: Create a New Project



Select the BOOTROM tool …

Create a QorIQ Configuration Project

Choose the location for the new project

Project name: project

☑ Use default location

Location: C:/Users/b23187/Freescale/QCS_TESTBUILD/B12 Browse...

Choose file system: default

Devices

Select the SoC you would like to use

Soc to be used:

type filter text

- ▷ G1
- ▷ P1
- ◢ P2
  - P2010
  - P2020
  - P2040
  - P2041
- ▷ P3
- ▷ P4
- ▷ P5

Creates project for P2020 derivative

Toolset selection

Choose what do you want to configure

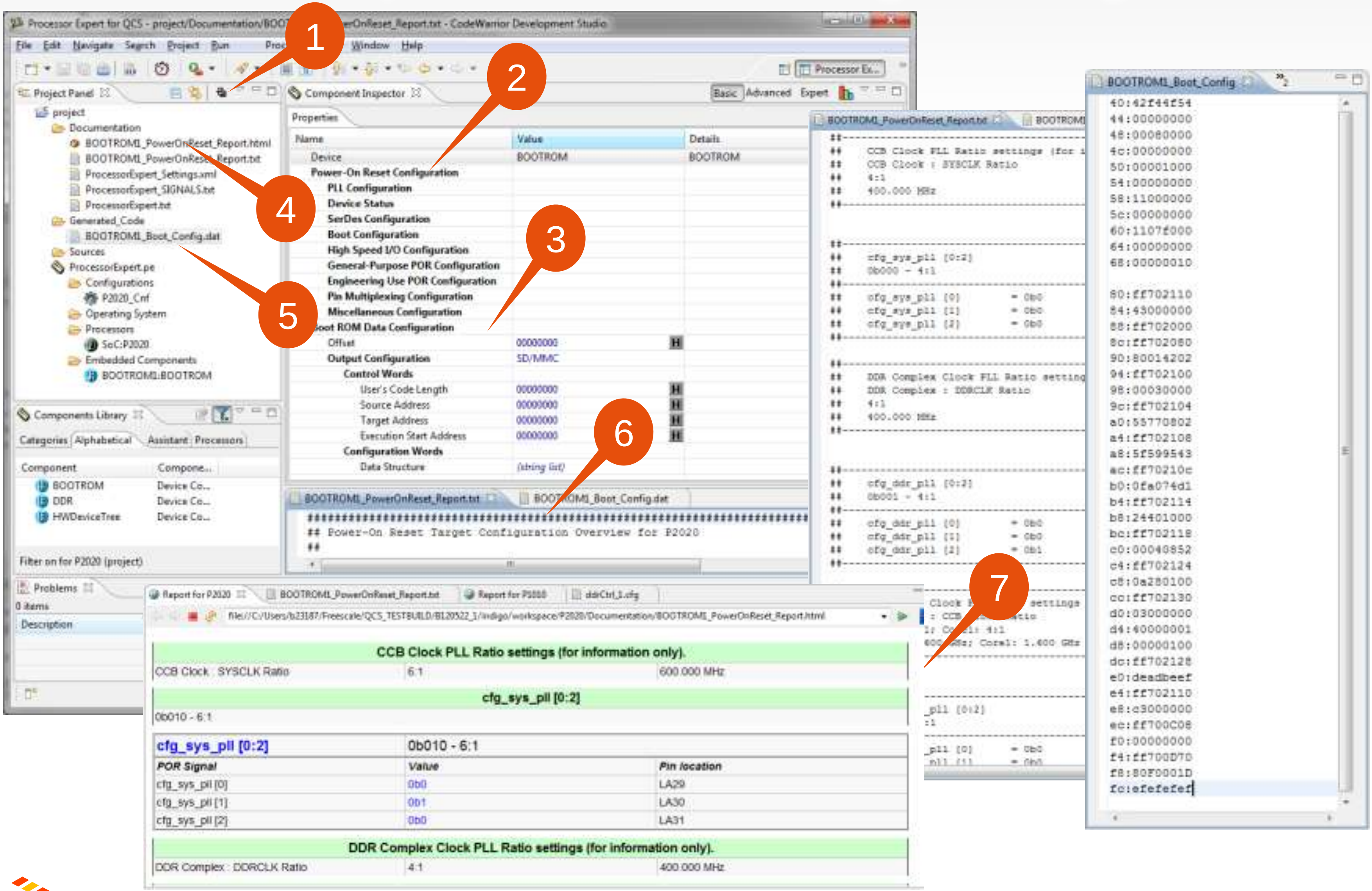
Components to be selected

type filter text

- ◢ □ Components
  - ☑ BOOTROM Configuration
  - □ DDR Memory Controller Configuration
  - □ Device Tree Editor

The BOOTROM configuration tool provide support for creating of boot ROM data structure and power-on reset configuration of P1/P2/G1 QorIQ devices.

BOOTROM Step 2: Review & change settings …



Name	Value	Details
Device	BOOTROM	BOOTROM
Power-On Reset Configuration		
PLL Configuration		
Device Status		
SerDes Configuration		
Boot Configuration		
High Speed I/O Configuration		
General-Purpose POR Configuration		
Engineering Use POR Configuration		
Pin Multiplexing Configuration		
Miscellaneous Configuration		
Boot ROM Data Configuration		
Offset	00000000 H	
Output Configuration	SD/MMC	
Control Words		
User's Code Length	00000000 H	
Source Address	00000000 H	
Target Address	00000000 H	
Execution Start Address	00000000 H	
Configuration Words		
Data Structure	(string list)	

```
##########################################################################
## Power-On Reset Target Configuration Overview for P2020
##
```

```
##------------------------------------------------------
##    CCB Clock PLL Ratio settings (for i
##    CCB Clock : SYSCLK Ratio
##    4:1
##    400.000 MHz
##------------------------------------------------------


##------------------------------------------------------
##    cfg_sys_pll [0:2]
##    0b000 - 4:1
##------------------------------------------------------
##    cfg_sys_pll [0]          = 0b0
##    cfg_sys_pll [1]          = 0b0
##    cfg_sys_pll [2]          = 0b0
##------------------------------------------------------

##------------------------------------------------------
##    DDR Complex Clock PLL Ratio setting
##    DDR Complex : DDRCLK Ratio
##    4:1
##    400.000 MHz
##------------------------------------------------------


##------------------------------------------------------
##    cfg_ddr_pll [0:2]
##    0b001 - 4:1
##------------------------------------------------------
##    cfg_ddr_pll [0]          = 0b0
##    cfg_ddr_pll [1]          = 0b0
##    cfg_ddr_pll [2]          = 0b1
##------------------------------------------------------
```

```
------------------------------------------
Clock P           settings
: CCB             atio
1: Co     l: 4:1
600    Hz; Core1: 1.600 GHz
------------------------------------------

------------------------------------------
_pll [0:2]
:1
------------------------------------------
_pll [0]          = 0b0
_pll [1]          = 0b0
```

file:///C:/Users/b23187/Freescale/QCS_TESTBUILD/B120522_1/indigo/workspace/P2020/Documentation/BOOTROM1_PowerOnReset_Report.html

CCB Clock PLL Ratio settings (for information only).		
CCB Clock : SYSCLK Ratio	6:1	600.000 MHz

cfg_sys_pll [0:2]
0b010 - 6:1

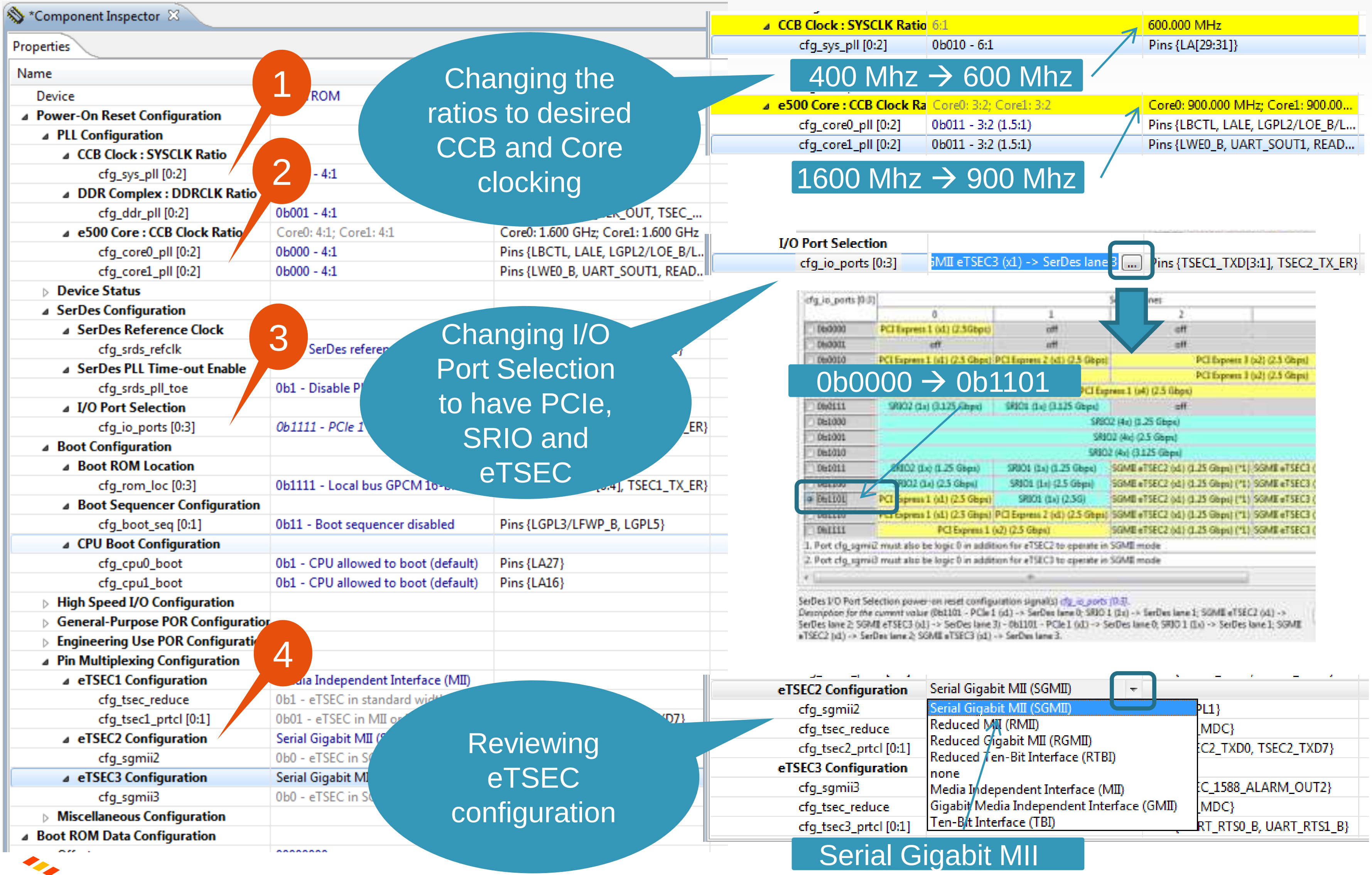
cfg_sys_pll [0:2]	0b010 - 6:1	
POR Signal	*Value*	*Pin location*
cfg_sys_pll [0]	0b0	LA29
cfg_sys_pll [1]	0b1	LA30
cfg_sys_pll [2]	0b0	LA31

DDR Complex Clock PLL Ratio settings (for information only).		
DDR Complex : DDRCLK Ratio	4:1	400.000 MHz

```
40:42f44f54
44:00000000
48:00080000
4c:00000000
50:00001000
54:00000000
58:11000000
5c:00000000
60:1107f000
64:00000000
68:00000010

80:ff702110
84:43000000
88:ff702000
8c:ff702080
90:80014202
94:ff702100
98:00030000
9c:ff702104
a0:55770802
a4:ff702108
a8:5f599543
ac:ff70210c
b0:0fa074d1
b4:ff702114
b8:24401000
bc:ff702118
c0:00040852
c4:ff702124
c8:0a280100
cc:ff702130
d0:03000000
d4:40000001
d8:00000100
dc:ff702128
e0:deadbeef
e4:ff702110
e8:c3000000
ec:ff700C08
f0:00000000
f4:ff700D70
f8:80F0001D
fc:efefefef
```

freescale™
飞思卡尔

BOOTROM Step 2: Review & change settings …



Name		
Device	ROM	
Power-On Reset Configuration		
PLL Configuration		
CCB Clock : SYSCLK Ratio		
cfg_sys_pll [0:2]	- 4:1	
DDR Complex : DDRCLK Ratio		
cfg_ddr_pll [0:2]	0b001 - 4:1	K_OUT, TSEC_...
e500 Core : CCB Clock Ratio	Core0: 4:1; Core1: 4:1	Core0: 1.600 GHz; Core1: 1.600 GHz
cfg_core0_pll [0:2]	0b000 - 4:1	Pins {LBCTL, LALE, LGPL2/LOE_B/L..
cfg_core1_pll [0:2]	0b000 - 4:1	Pins {LWE0_B, UART_SOUT1, READ..
Device Status		
SerDes Configuration		
SerDes Reference Clock		
cfg_srds_refclk	SerDes referen	}
SerDes PLL Time-out Enable		
cfg_srds_pll_toe	0b1 - Disable P	
I/O Port Selection		
cfg_io_ports [0:3]	*0b1111 - PCIe 1*	_ER}
Boot Configuration		
Boot ROM Location		
cfg_rom_loc [0:3]	0b1111 - Local bus GPCM 16-b	4], TSEC1_TX_ER}
Boot Sequencer Configuration		
cfg_boot_seq [0:1]	0b11 - Boot sequencer disabled	Pins {LGPL3/LFWP_B, LGPL5}
CPU Boot Configuration		
cfg_cpu0_boot	0b1 - CPU allowed to boot (default)	Pins {LA27}
cfg_cpu1_boot	0b1 - CPU allowed to boot (default)	Pins {LA16}
High Speed I/O Configuration		
General-Purpose POR Configuratio		
Engineering Use POR Configurati		
Pin Multiplexing Configuration		
eTSEC1 Configuration	ia Independent Interface (MII)	
cfg_tsec_reduce	0b1 - eTSEC in standard wid	
cfg_tsec1_prtcl [0:1]	0b01 - eTSEC in MII or	D7}
eTSEC2 Configuration	Serial Gigabit MII (S	
cfg_sgmii2	0b0 - eTSEC in S	
eTSEC3 Configuration	Serial Gigabit MI	
cfg_sgmii3	0b0 - eTSEC in S	
Miscellaneous Configuration		
Boot ROM Data Configuration		

Changing the ratios to desired CCB and Core clocking

CCB Clock : SYSCLK Ratio	6:1	600.000 MHz
cfg_sys_pll [0:2]	0b010 - 6:1	Pins {LA[29:31]}

400 Mhz → 600 Mhz

e500 Core : CCB Clock Ra	Core0: 3:2; Core1: 3:2	Core0: 900.000 MHz; Core1: 900.00...
cfg_core0_pll [0:2]	0b011 - 3:2 (1.5:1)	Pins {LBCTL, LALE, LGPL2/LOE_B/L...
cfg_core1_pll [0:2]	0b011 - 3:2 (1.5:1)	Pins {LWE0_B, UART_SOUT1, READ...

1600 Mhz → 900 Mhz

Changing I/O Port Selection to have PCIe, SRIO and eTSEC

I/O Port Selection		
cfg_io_ports [0:3]	GMII eTSEC3 (x1) -> SerDes lane 3	Pins {TSEC1_TXD[3:1], TSEC2_TX_ER}

cfg_io_ports [0:3]	SerDes lanes		
	0	1	2
○ 0b0000	PCI Express 1 (x1) (2.5Gbps)	off	off
○ 0b0001	off	off	off
○ 0b0010	PCI Express 1 (x1) (2.5 Gbps)	PCI Express 2 (x1) (2.5 Gbps)	PCI Express 3 (x2) (2.5 Gbps)
			PCI Express 3 (x2) (2.5 Gbps)
	PCI Express 1 (x4) (2.5 Gbps)		
○ 0b0111	SRIO2 (1x) (3.125 Gbps)	SRIO1 (1x) (3.125 Gbps)	off
○ 0b1000	SRIO2 (4x) (1.25 Gbps)		
○ 0b1001	SRIO2 (4x) (2.5 Gbps)		
○ 0b1010	SRIO2 (4x) (3.125 Gbps)		
○ 0b1011	SRIO2 (1x) (1.25 Gbps)	SRIO1 (1x) (1.25 Gbps)	SGMII eTSEC2 (x1) (1.25 Gbps) (*1) | SGMII eTSEC3 (
○	SRIO2 (1x) (2.5 Gbps)	SRIO1 (1x) (2.5 Gbps)	SGMII eTSEC2 (x1) (1.25 Gbps) (*1) | SGMII eTSEC3 (
● 0b1101	PCI Express 1 (x1) (2.5 Gbps)	SRIO1 (1x) (2.5G)	SGMII eTSEC2 (x1) (1.25 Gbps) (*1) | SGMII eTSEC3 (
○	PCI Express 1 (x1) (2.5 Gbps)	PCI Express 2 (x1) (2.5 Gbps)	SGMII eTSEC2 (x1) (1.25 Gbps) (*1) | SGMII eTSEC3 (
○ 0b1111	PCI Express 1 (x2) (2.5 Gbps)		SGMII eTSEC2 (x1) (1.25 Gbps) (*1) | SGMII eTSEC3 (

1. Port cfg_sgmii2 must also be logic 0 in addition for eTSEC2 to operate in SGMII mode
2. Port cfg_sgmii3 must also be logic 0 in addition for eTSEC3 to operate in SGMII mode

SerDes I/O Port Selection power-on reset configuration signal(s) cfg_io_ports [0:3].
Description for the current value (0b1101 - PCIe 1 (x1) -> SerDes lane 0; SRIO 1 (1x) -> SerDes lane 1; SGMII eTSEC2 (x1) -> SerDes lane 2; SGMII eTSEC3 (x1) -> SerDes lane 3) - 0b1101 - PCIe 1 (x1) -> SerDes lane 0; SRIO 1 (1x) -> SerDes lane 1; SGMII eTSEC2 (x1) -> SerDes lane 2; SGMII eTSEC3 (x1) -> SerDes lane 3.

0b0000 → 0b1101

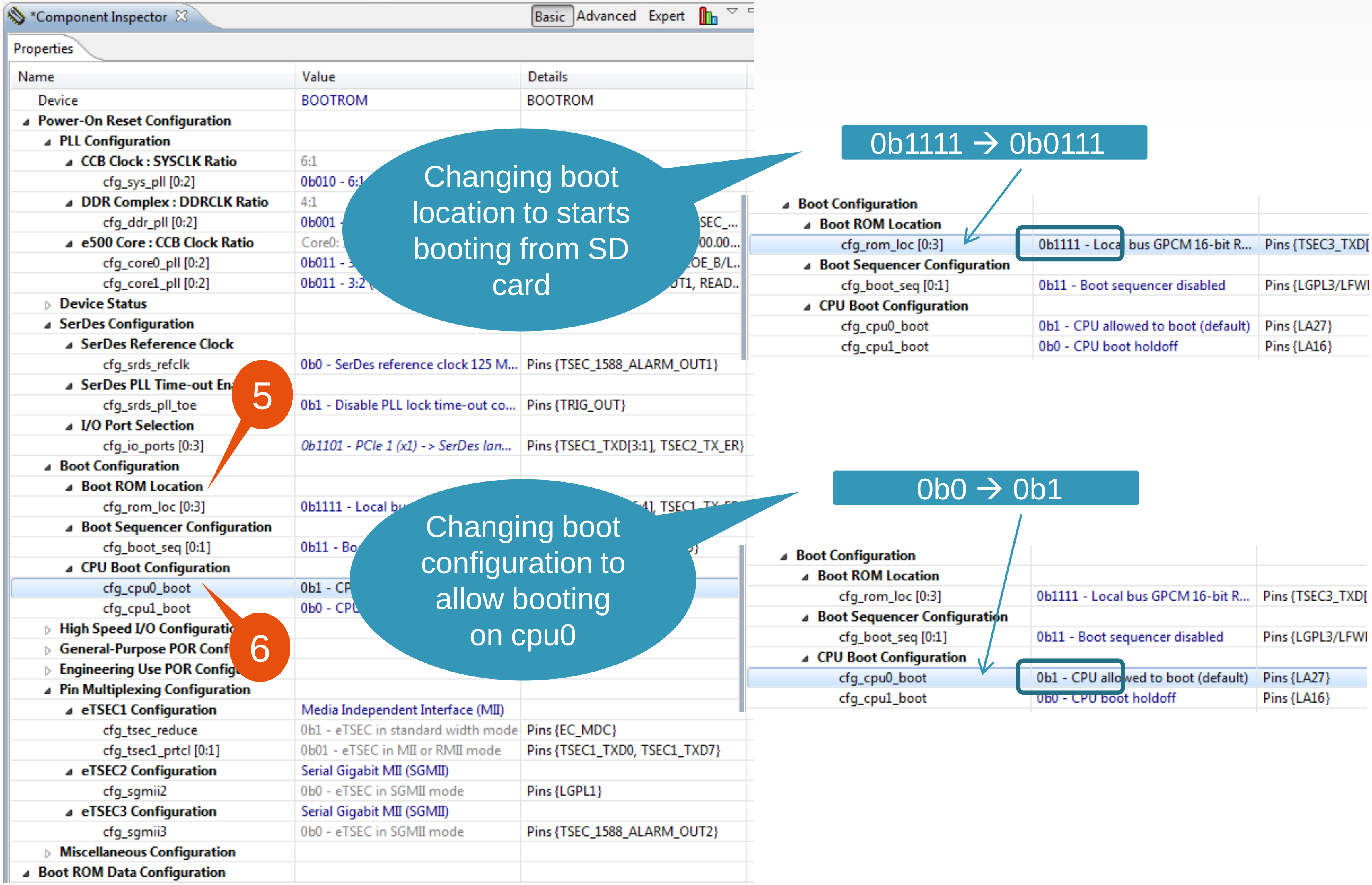
Reviewing eTSEC configuration

eTSEC2 Configuration	Serial Gigabit MII (SGMII)	
cfg_sgmii2		PL1}
cfg_tsec_reduce		MDC}
cfg_tsec2_prtcl [0:1]		C2_TXD0, TSEC2_TXD7}
eTSEC3 Configuration		
cfg_sgmii3		C_1588_ALARM_OUT2}
cfg_tsec_reduce		MDC}
cfg_tsec3_prtcl [0:1]		RT_RTS0_B, UART_RTS1_B}

- Serial Gigabit MII (SGMII)
- Reduced MII (RMII)
- Reduced Gigabit MII (RGMII)
- Reduced Ten-Bit Interface (RTBI)
- none
- Media Independent Interface (MII)
- Gigabit Media Independent Interface (GMII)
- Ten-Bit Interface (TBI)

Serial Gigabit MII

BOOTROM Step 2: Configuration for SD booting



Name	Value	Details
Device	BOOTROM	BOOTROM
Power-On Reset Configuration		
PLL Configuration		
CCB Clock : SYSCLK Ratio	6:1	
cfg_sys_pll [0:2]	0b010 - 6:1	
DDR Complex : DDRCLK Ratio	4:1	
cfg_ddr_pll [0:2]	0b001 -	SEC_...
e500 Core : CCB Clock Ratio	Core0:	00.00...
cfg_core0_pll [0:2]	0b011 - 3	OE_B/L...
cfg_core1_pll [0:2]	0b011 - 3:2	T1, READ...
Device Status		
SerDes Configuration		
SerDes Reference Clock		
cfg_srds_refclk	0b0 - SerDes reference clock 125 M...	Pins {TSEC_1588_ALARM_OUT1}
SerDes PLL Time-out Ena		
cfg_srds_pll_toe	0b1 - Disable PLL lock time-out co...	Pins {TRIG_OUT}
I/O Port Selection		
cfg_io_ports [0:3]	*0b1101 - PCIe 1 (x1) -> SerDes lan...*	Pins {TSEC1_TXD[3:1], TSEC2_TX_ER}
Boot Configuration		
Boot ROM Location		
cfg_rom_loc [0:3]	0b1111 - Local bu	4], TSEC1_TX_ER
Boot Sequencer Configuration		
cfg_boot_seq [0:1]	0b11 - Bo	}
CPU Boot Configuration		
cfg_cpu0_boot	0b1 - CP	
cfg_cpu1_boot	0b0 - CPU	
High Speed I/O Configuratic		
General-Purpose POR Conf		
Engineering Use POR Config		
Pin Multiplexing Configuration		
eTSEC1 Configuration	Media Independent Interface (MII)	
cfg_tsec_reduce	0b1 - eTSEC in standard width mode	Pins {EC_MDC}
cfg_tsec1_prtcl [0:1]	0b01 - eTSEC in MII or RMII mode	Pins {TSEC1_TXD0, TSEC1_TXD7}
eTSEC2 Configuration	Serial Gigabit MII (SGMII)	
cfg_sgmii2	0b0 - eTSEC in SGMII mode	Pins {LGPL1}
eTSEC3 Configuration	Serial Gigabit MII (SGMII)	
cfg_sgmii3	0b0 - eTSEC in SGMII mode	Pins {TSEC_1588_ALARM_OUT2}
Miscellaneous Configuration		
Boot ROM Data Configuration		

5

Changing boot location to starts booting from SD card

0b1111 → 0b0111

Boot Configuration		
Boot ROM Location		
cfg_rom_loc [0:3]	0b1111 - Loca bus GPCM 16-bit R...	Pins {TSEC3_TXD[
Boot Sequencer Configuration		
cfg_boot_seq [0:1]	0b11 - Boot sequencer disabled	Pins {LGPL3/LFWI
CPU Boot Configuration		
cfg_cpu0_boot	0b1 - CPU allowed to boot (default)	Pins {LA27}
cfg_cpu1_boot	0b0 - CPU boot holdoff	Pins {LA16}

6

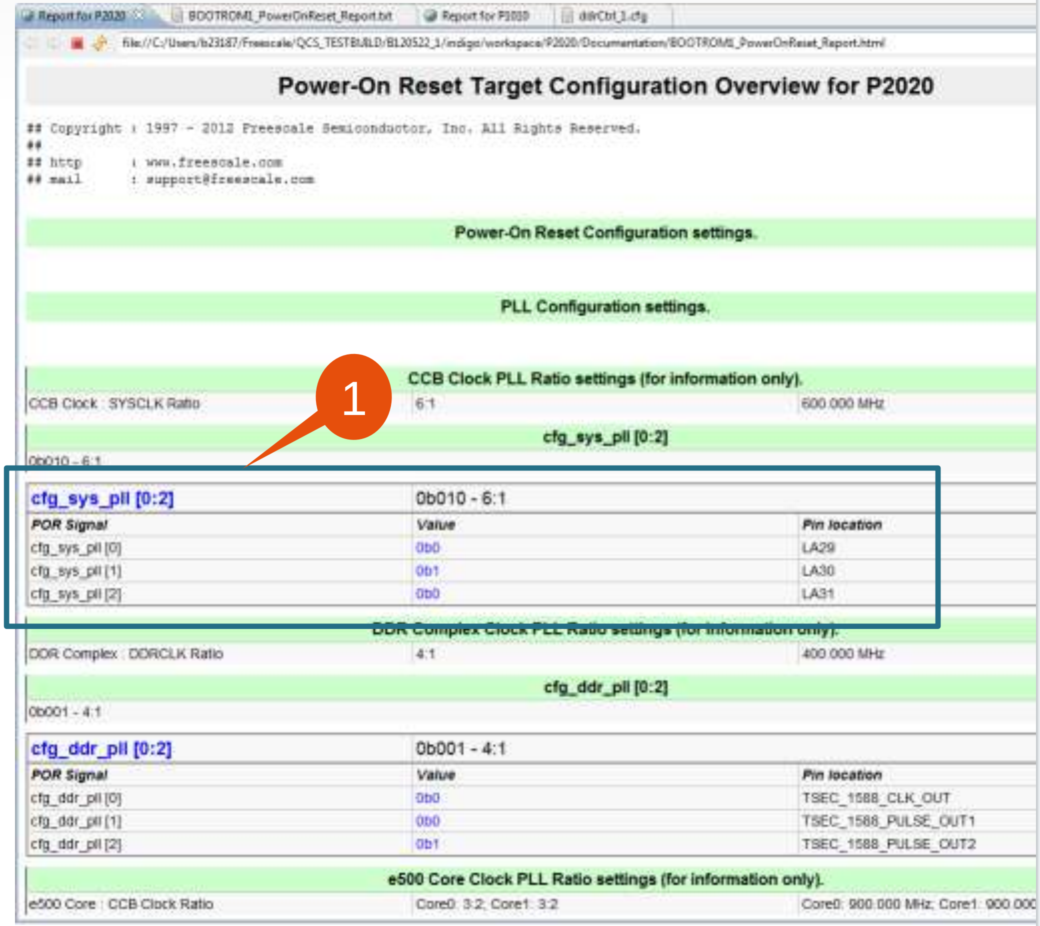
Changing boot configuration to allow booting on cpu0

0b0 → 0b1

Boot Configuration		
Boot ROM Location		
cfg_rom_loc [0:3]	0b1111 - Local bus GPCM 16-bit R...	Pins {TSEC3_TXD[
Boot Sequencer Configuration		
cfg_boot_seq [0:1]	0b11 - Boot sequencer disabled	Pins {LGPL3/LFWI
CPU Boot Configuration		
cfg_cpu0_boot	0b1 - CPU allowed to boot (default)	Pins {LA27}
cfg_cpu1_boot	0b0 - CPU boot holdoff	Pins {LA16}

freescale™
飞思卡尔

BOOTROM Step 3: Observe Power-on reset overview details …


Report for P2020
BOOTROM1_PowerOnReset_Report.txt
Report for P1010
ddrCtrl_1.cfg
file:///C:/Users/b23187/Freescale/QCS_TESTBUILD/B120522_1/indigo/workspace/P2020/Documentation/BOOTROM1_PowerOnReset_Report.html
1

Power-On Reset Target Configuration Overview for P2020

```
## Copyright : 1997 - 2012 Freescale Semiconductor, Inc. All Rights Reserved.
##
## http      : www.freescale.com
## mail      : support@freescale.com
```

Power-On Reset Configuration settings.

PLL Configuration settings.

CCB Clock PLL Ratio settings (for information only).

CCB Clock : SYSCLK Ratio	6:1	600.000 MHz

cfg_sys_pll [0:2]

0b010 - 6:1

cfg_sys_pll [0:2]	0b010 - 6:1	
POR Signal	***Value***	***Pin location***
cfg_sys_pll [0]	0b0	LA29
cfg_sys_pll [1]	0b1	LA30
cfg_sys_pll [2]	0b0	LA31

DDR Complex Clock PLL Ratio settings (for information only).

DDR Complex : DDRCLK Ratio	4:1	400.000 MHz

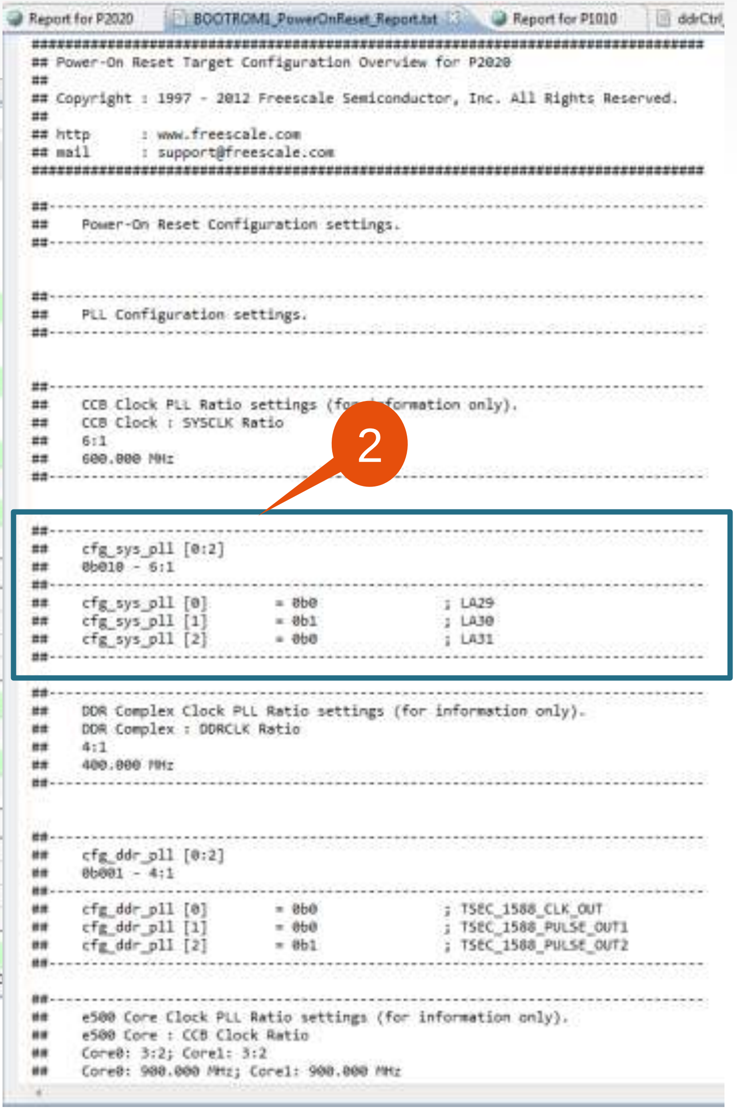
cfg_ddr_pll [0:2]

0b001 - 4:1

cfg_ddr_pll [0:2]	0b001 - 4:1	
POR Signal	***Value***	***Pin location***
cfg_ddr_pll [0]	0b0	TSEC_1588_CLK_OUT
cfg_ddr_pll [1]	0b0	TSEC_1588_PULSE_OUT1
cfg_ddr_pll [2]	0b1	TSEC_1588_PULSE_OUT2

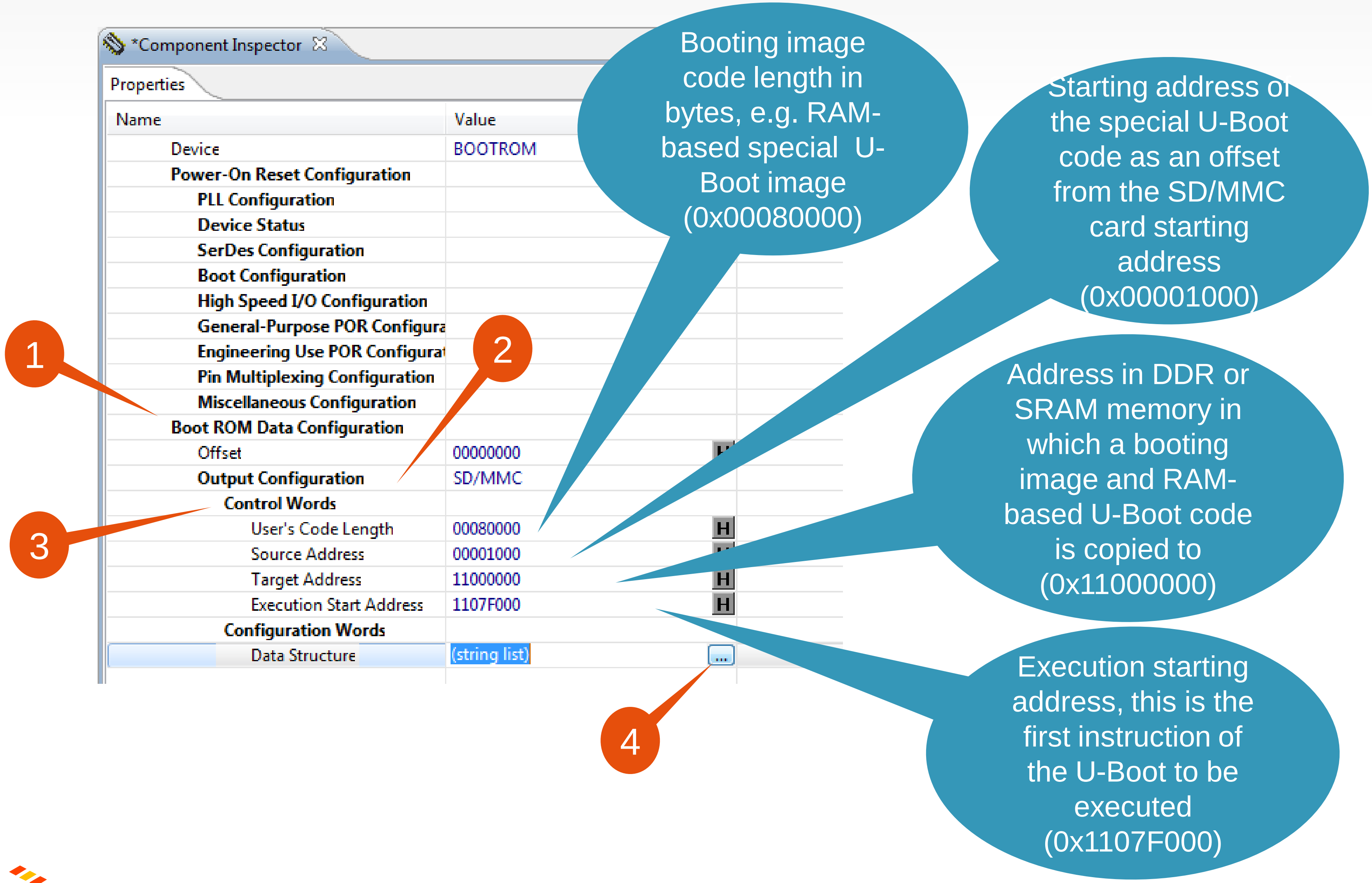
e500 Core Clock PLL Ratio settings (for information only).

e500 Core : CCB Clock Ratio	Core0: 3:2; Core1: 3:2	Core0: 900.000 MHz; Core1: 900.000



```
##################################################################################
## Power-On Reset Target Configuration Overview for P2020
##
## Copyright : 1997 - 2012 Freescale Semiconductor, Inc. All Rights Reserved.
##
## http      : www.freescale.com
## mail      : support@freescale.com
##################################################################################


##--------------------------------------------------------------------------------
##    Power-On Reset Configuration settings.
##--------------------------------------------------------------------------------



##--------------------------------------------------------------------------------
##    PLL Configuration settings.
##--------------------------------------------------------------------------------



##--------------------------------------------------------------------------------
##    CCB Clock PLL Ratio settings (for information only).
##    CCB Clock : SYSCLK Ratio
##    6:1
##    600.000 MHz
##--------------------------------------------------------------------------------


##--------------------------------------------------------------------------------
##    cfg_sys_pll [0:2]
##    0b010 - 6:1
##--------------------------------------------------------------------------------
##    cfg_sys_pll [0]       = 0b0                ; LA29
##    cfg_sys_pll [1]       = 0b1                ; LA30
##    cfg_sys_pll [2]       = 0b0                ; LA31
##--------------------------------------------------------------------------------

##--------------------------------------------------------------------------------
##    DDR Complex Clock PLL Ratio settings (for information only).
##    DDR Complex : DDRCLK Ratio
##    4:1
##    400.000 MHz
##--------------------------------------------------------------------------------



##--------------------------------------------------------------------------------
##    cfg_ddr_pll [0:2]
##    0b001 - 4:1
##--------------------------------------------------------------------------------
##    cfg_ddr_pll [0]       = 0b0                ; TSEC_1588_CLK_OUT
##    cfg_ddr_pll [1]       = 0b0                ; TSEC_1588_PULSE_OUT1
##    cfg_ddr_pll [2]       = 0b1                ; TSEC_1588_PULSE_OUT2
##--------------------------------------------------------------------------------

##--------------------------------------------------------------------------------
##    e500 Core Clock PLL Ratio settings (for information only).
##    e500 Core : CCB Clock Ratio
##    Core0: 3:2; Core1: 3:2
##    Core0: 900.000 MHz; Core1: 900.000 MHz
```

BOOTROM Step 4: Prepare configuration data file



Name	Value
Device	BOOTROM
Power-On Reset Configuration	
PLL Configuration	
Device Status	
SerDes Configuration	
Boot Configuration	
High Speed I/O Configuration	
General-Purpose POR Configura	
Engineering Use POR Configurat	
Pin Multiplexing Configuration	
Miscellaneous Configuration	
Boot ROM Data Configuration	
Offset	00000000
Output Configuration	SD/MMC
Control Words	
User's Code Length	00080000
Source Address	00001000
Target Address	11000000
Execution Start Address	1107F000
Configuration Words	
Data Structure	(string list)

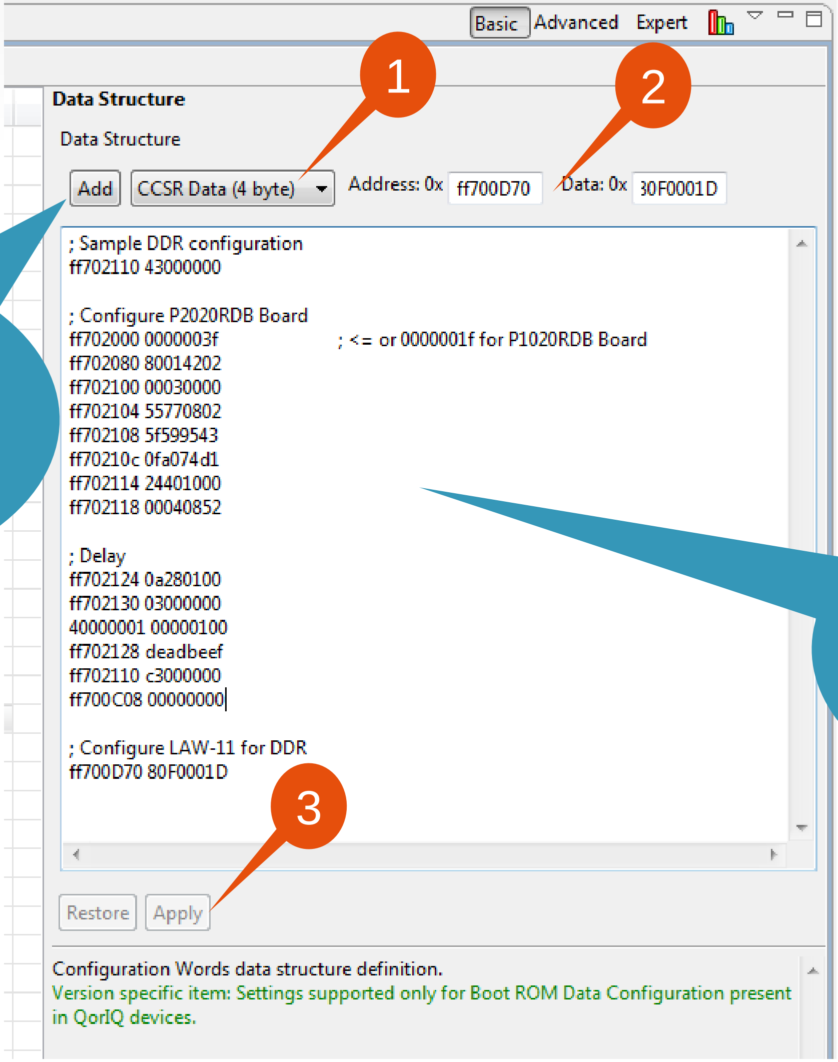
Booting image code length in bytes, e.g. RAM-based special U-Boot image (0x00080000)

Starting address of the special U-Boot code as an offset from the SD/MMC card starting address (0x00001000)

Address in DDR or SRAM memory in which a booting image and RAM-based U-Boot code is copied to (0x11000000)

Execution starting address, this is the first instruction of the U-Boot to be executed (0x1107F000)

BOOTROM Step 4: Prepare configuration data file



Data Structure

Data Structure

Add | CCSR Data (4 byte) | Address: 0x ff700D70 | Data: 0x 30F0001D

```
; Sample DDR configuration
ff702110 43000000

; Configure P2020RDB Board
ff702000 0000003f             ; <= or 0000001f for P1020RDB Board
ff702080 80014202
ff702100 00030000
ff702104 55770802
ff702108 5f599543
ff70210c 0fa074d1
ff702114 24401000
ff702118 00040852

; Delay
ff702124 0a280100
ff702130 03000000
40000001 00000100
ff702128 deadbeef
ff702110 c3000000
ff700C08 00000000

; Configure LAW-11 for DDR
ff700D70 80F0001D
```

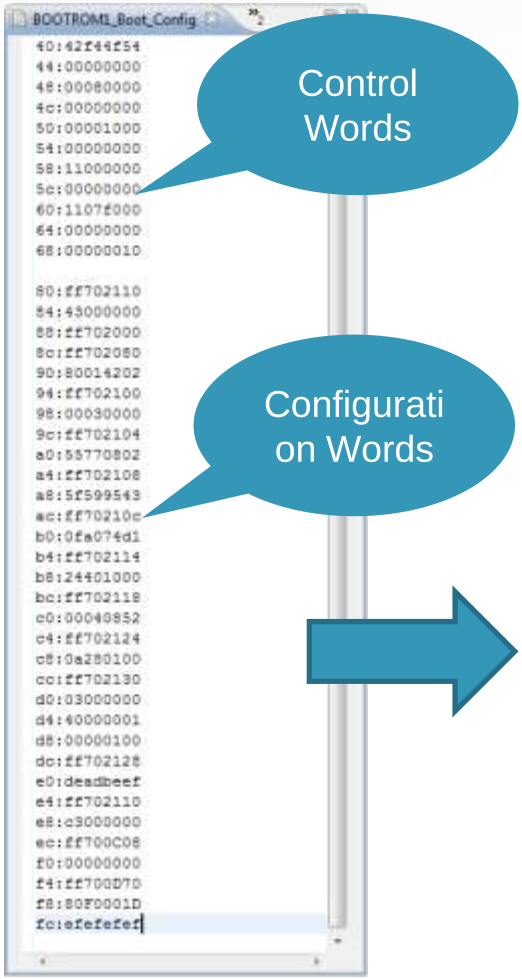
Restore | Apply

Configuration Words data structure definition.
Version specific item: Settings supported only for Boot ROM Data Configuration present in QorIQ devices.

Adding configuration pairs (address + data) to be included into configuration data file

Sample DDR configuration data structure

BOOTROM Step 5: Usage of configuration data file with external booting utility application



```
40:42f44f54
44:00000000
48:00080000
4c:00000000
50:00001000
54:00000000
58:11000000
5c:00000000
60:1107f000
64:00000000
68:00000010

80:ff702110
84:43000000
88:ff702000
8c:ff702080
90:80014202
94:ff702100
98:00030000
9c:ff702104
a0:55770802
a4:ff702108
a8:5f599543
ac:ff70210c
b0:0fa074d1
b4:ff702114
b8:24401000
bc:ff702118
c0:00040852
c4:ff702124
c8:0a280100
cc:ff702130
d0:03000000
d4:40000001
d8:00000100
dc:ff702128
e0:deadbeef
e4:ff702110
e8:c3000000
ec:ff700C08
f0:00000000
f4:ff700D70
f8:80F0001D
fc:efefefef
```

Use a booting utility *boot_format* from a BSP package for P2020RDB/P102RDB

Once you have installed BSP and let configured ltib to build root file system *rootfs.tar.gz.uboot* for Linux boot. Boot the board using this root file system and *boot_format* utility can be located under:

[root@P1020RDB /]# *cd /boot_format/*
[root@P1020RDB */boot_format]# ls -l*

```
-rwxr-xr-x 1 root root 10400 Apr 7 2010 boot_format
-rw-r--r-- 1 root root   530 Apr 7 2010 config_sram.dat
```

The utility shows how to use it when typing ***./boot_format***

```
Usage: ./boot_format config_file image -sd dev [-o out_config] |
-spi  spiimage

config_file : includes boot signature and config words
image       : the U-Boot image for booting from eSDHC/eSPI
dev         : SDCard's device node(e.g. /dev/sdb, /dev/mmcblk0)
spiimage    : boot image for SPI mode
out_config  : modified config file for SD mode
```

BOOTROM tool generated config file can be used together with U-Boot image and put on an SD/MMC card using command line, e.g. for /dev/mmcblk0:

```
./boot_format BOOTROM1_Boot_Config.dat u-boot.bin -sd /dev/mmcblk0
```

freescale™

DDR Configuration

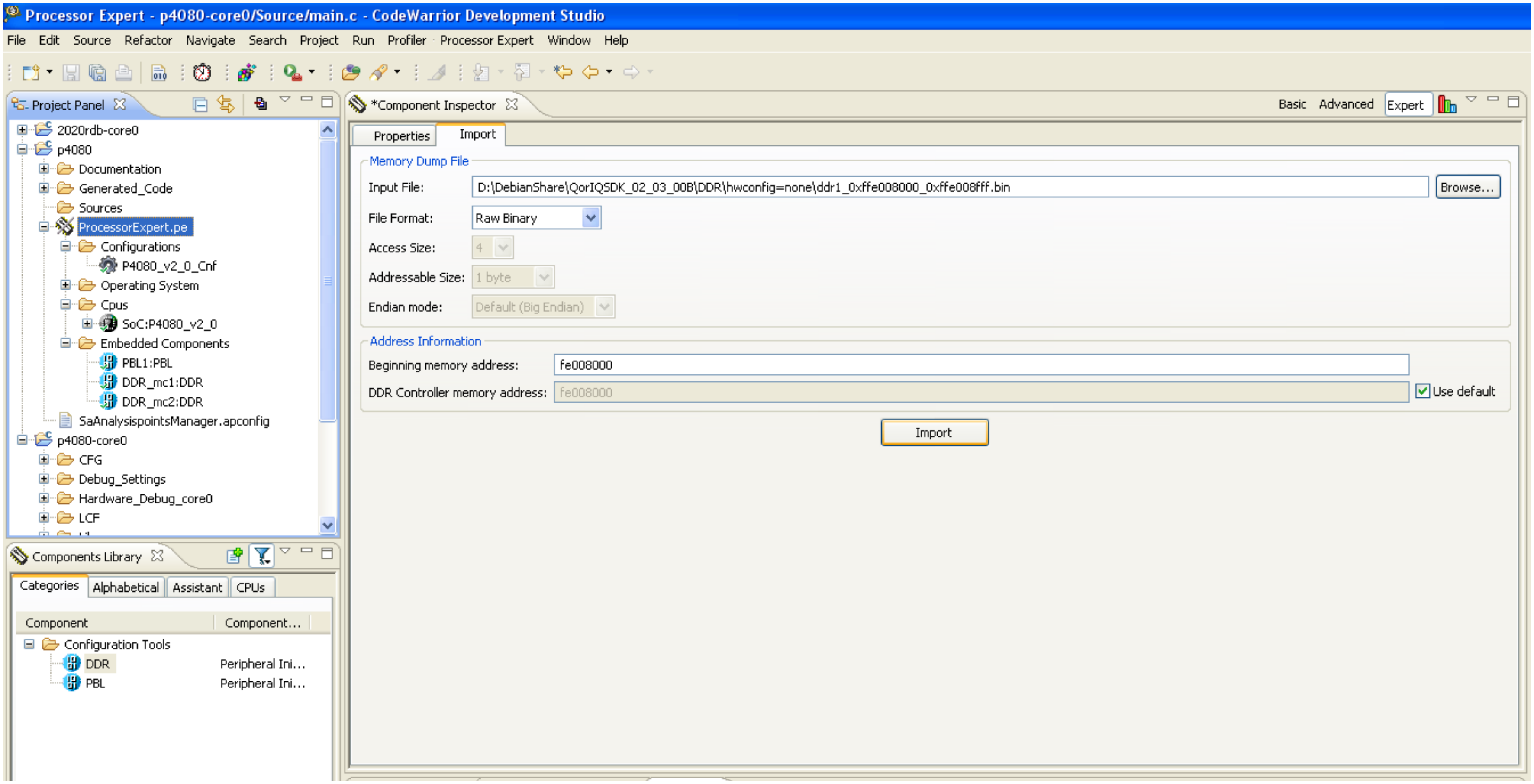
Lab 4: Changing the DDR Configuration

DDR Step 1: Dump u-boot DDR Registers

- Deploy SDK2.3 u-boot
- No interleaving "fsl_ddr:ctlr_intlv=null“
- Use CW to connect and dump DDR1 and DDR2 registers (Memory browser export function)

```
U-Boot 2010.12-00001-g612800e (Jan 28 2011 - 22:20:46)

CPU0:  P4080E, Version: 2.0, (0x82080020)
Core:  E500MC, Version: 2.0, (0x80230020)
Clock Configuration:
       CPU0:1499.985 MHz, CPU1:1499.985 MHz, CPU2:1499.985 MHz,
CPU3:1499.985 MHz,
       CPU4:1499.985 MHz, CPU5:1499.985 MHz, CPU6:1499.985 MHz,
CPU7:1499.985 MHz,
       CCB:799.992 MHz,
       DDR:649.994 MHz (1299.987 MT/s data rate) (Asynchronous),
LBC:99.999 MHz
       FMAN1: 599.994 MHz
       FMAN2: 599.994 MHz
       PME:   399.996 MHz
L1:    D-cache 32 kB enabled
       I-cache 32 kB enabled
Board: P4080DS, Sys ID: 0x17, Sys Ver: 0x01, FPGA Ver: 0x0c, vBank: 4
36-bit Addressing
Reset Configuration Word (RCW):
       00000000: 105a0000 00000000 1e1e181e 0000cccc
       00000010: 40464000 3c3c2000 fe800000 61000000
       00000020: 00000000 00000000 00000000 008b6000
       00000030: 00000000 00000000 00000000 00000000
SERDES Reference Clocks: Bank1=125MHz Bank2=125MHz Bank3=125MHz
I2C:   ready
DRAM:  Initializing....using SPD
Detected UDIMM(s)
Detected UDIMM(s)
2 GiB left unmapped
    DDR: 4 GiB (DDR3, 64-bit, CL=9, ECC on)
Testing 0x00000000 - 0x7fffffff
Testing 0x80000000 - 0xffffffff
Remap DDR 2 GiB left unmapped

Hit any key to stop autoboot:  0
=> md 0xfe008000
```

DDR Step 2: Import DDR1 and DDR2 Registers Dump Under QCS



Input File: D:\DebianShare\QorIQSDK_02_03_00B\DDR\hwconfig=none\ddr1_0xffe008000_0xffe008fff.bin

File Format: Raw Binary

Access Size: 4

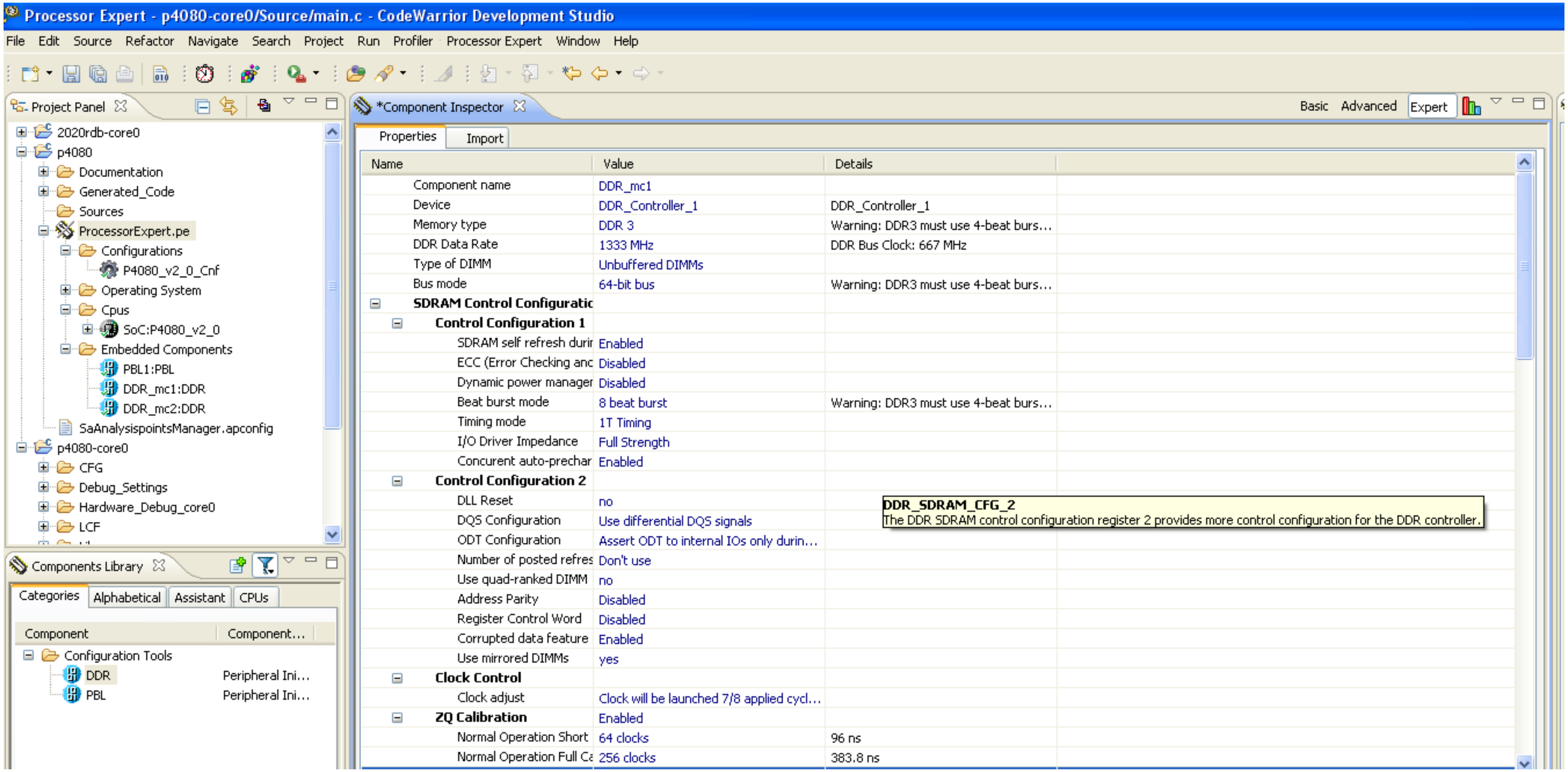
Addressable Size: 1 byte

Endian mode: Default (Big Endian)

Beginning memory address: fe008000

DDR Controller memory address: fe008000 ☑ Use default

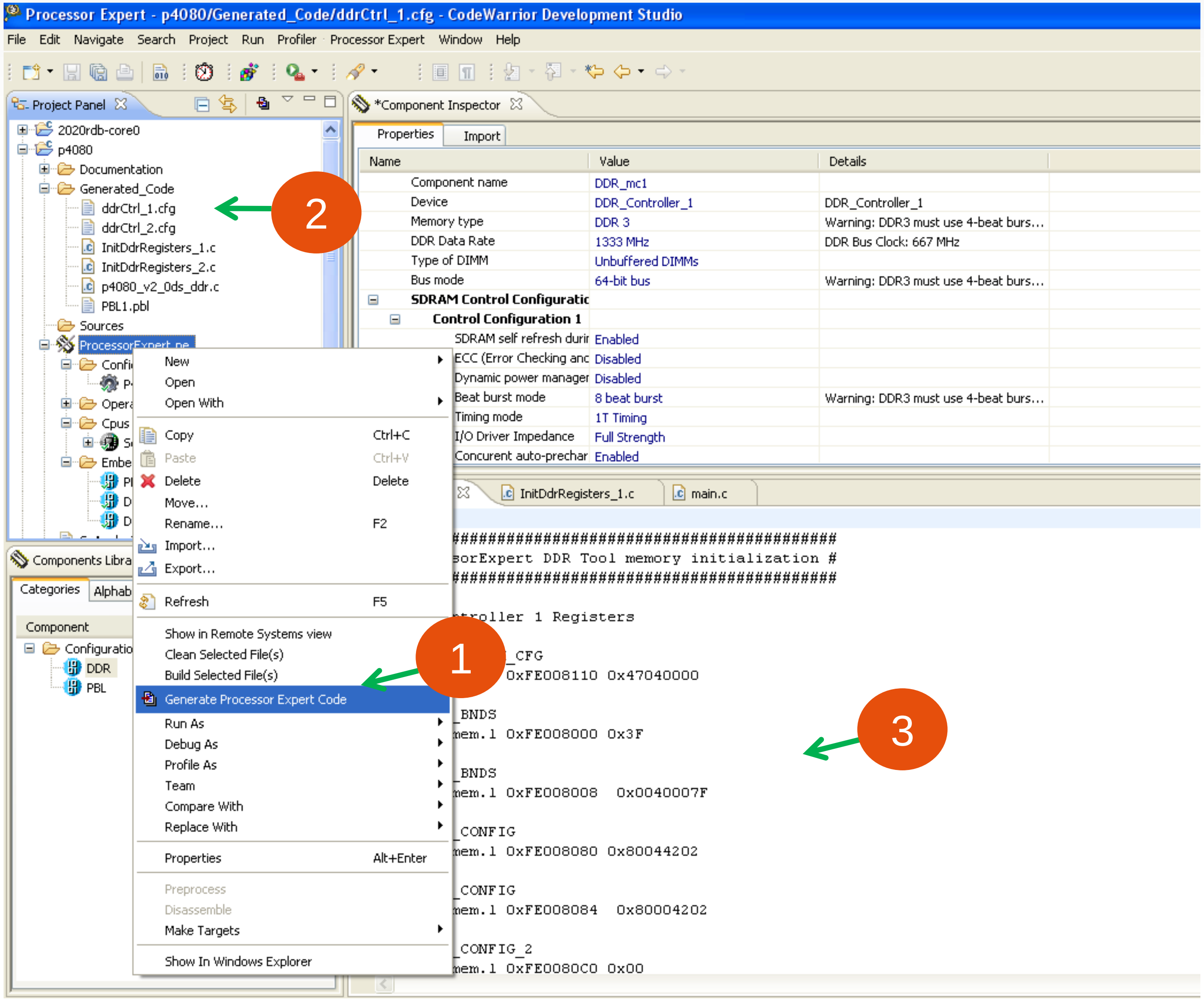
DDR Step 3: Review Decoded Configurations



Name	Value	Details
Component name	DDR_mc1	
Device	DDR_Controller_1	DDR_Controller_1
Memory type	DDR 3	Warning: DDR3 must use 4-beat burs...
DDR Data Rate	1333 MHz	DDR Bus Clock: 667 MHz
Type of DIMM	Unbuffered DIMMs	
Bus mode	64-bit bus	Warning: DDR3 must use 4-beat burs...
⊟ SDRAM Control Configuratic		
⊟ Control Configuration 1		
SDRAM self refresh durir	Enabled	
ECC (Error Checking anc	Disabled	
Dynamic power manager	Disabled	
Beat burst mode	8 beat burst	Warning: DDR3 must use 4-beat burs...
Timing mode	1T Timing	
I/O Driver Impedance	Full Strength	
Concurent auto-prechar	Enabled	
⊟ Control Configuration 2		
DLL Reset	no	
DQS Configuration	Use differential DQS signals	
ODT Configuration	Assert ODT to internal IOs only durin...	
Number of posted refres	Don't use	
Use quad-ranked DIMM	no	
Address Parity	Disabled	
Register Control Word	Disabled	
Corrupted data feature	Enabled	
Use mirrored DIMMs	yes	
⊟ Clock Control		
Clock adjust	Clock will be launched 7/8 applied cycl...	
⊟ ZQ Calibration	Enabled	
Normal Operation Short	64 clocks	96 ns
Normal Operation Full Ca	256 clocks	383.8 ns

DDR_SDRAM_CFG_2
The DDR SDRAM control configuration register 2 provides more control configuration for the DDR controller.

DDR Step 4: Generate DDR Config File



Name	Value	Details
Component name	DDR_mc1	
Device	DDR_Controller_1	DDR_Controller_1
Memory type	DDR 3	Warning: DDR3 must use 4-beat burs...
DDR Data Rate	1333 MHz	DDR Bus Clock: 667 MHz
Type of DIMM	Unbuffered DIMMs	
Bus mode	64-bit bus	Warning: DDR3 must use 4-beat burs...
SDRAM Control Configuratic		
Control Configuration 1		
SDRAM self refresh durir	Enabled	
ECC (Error Checking anc	Disabled	
Dynamic power manager	Disabled	
Beat burst mode	8 beat burst	Warning: DDR3 must use 4-beat burs...
Timing mode	1T Timing	
I/O Driver Impedance	Full Strength	
Concurent auto-prechar	Enabled	

```
##########################################
sorExpert DDR Tool memory initialization #
##########################################

troller 1 Registers

_CFG
 0xFE008110 0x47040000

BNDS
mem.l 0xFE008000 0x3F

BNDS
mem.l 0xFE008008  0x0040007F

CONFIG
mem.l 0xFE008080 0x80044202

CONFIG
mem.l 0xFE008084  0x80004202

CONFIG_2
mem.l 0xFE0080C0 0x00
```

freescale™
飞思卡尔

DDR Step 5: Adapt CW Config File

- **Open the CW config file you want to adapt**

  D:\Program Files\Freescale\CW PA v10.1\PA\PA_Support\Initialization_Files\QorIQ_P4\P4080DS_init_core0.tcl

- **Replace DDR1 config section with the one from:**

  D:\Profiles\b08844\workspace\p4080\Generated_Code\ddrCtrl_1.tcl

- **Use the new config file with your stationary project**

freescale™
飞思卡尔

Device Tree Editor

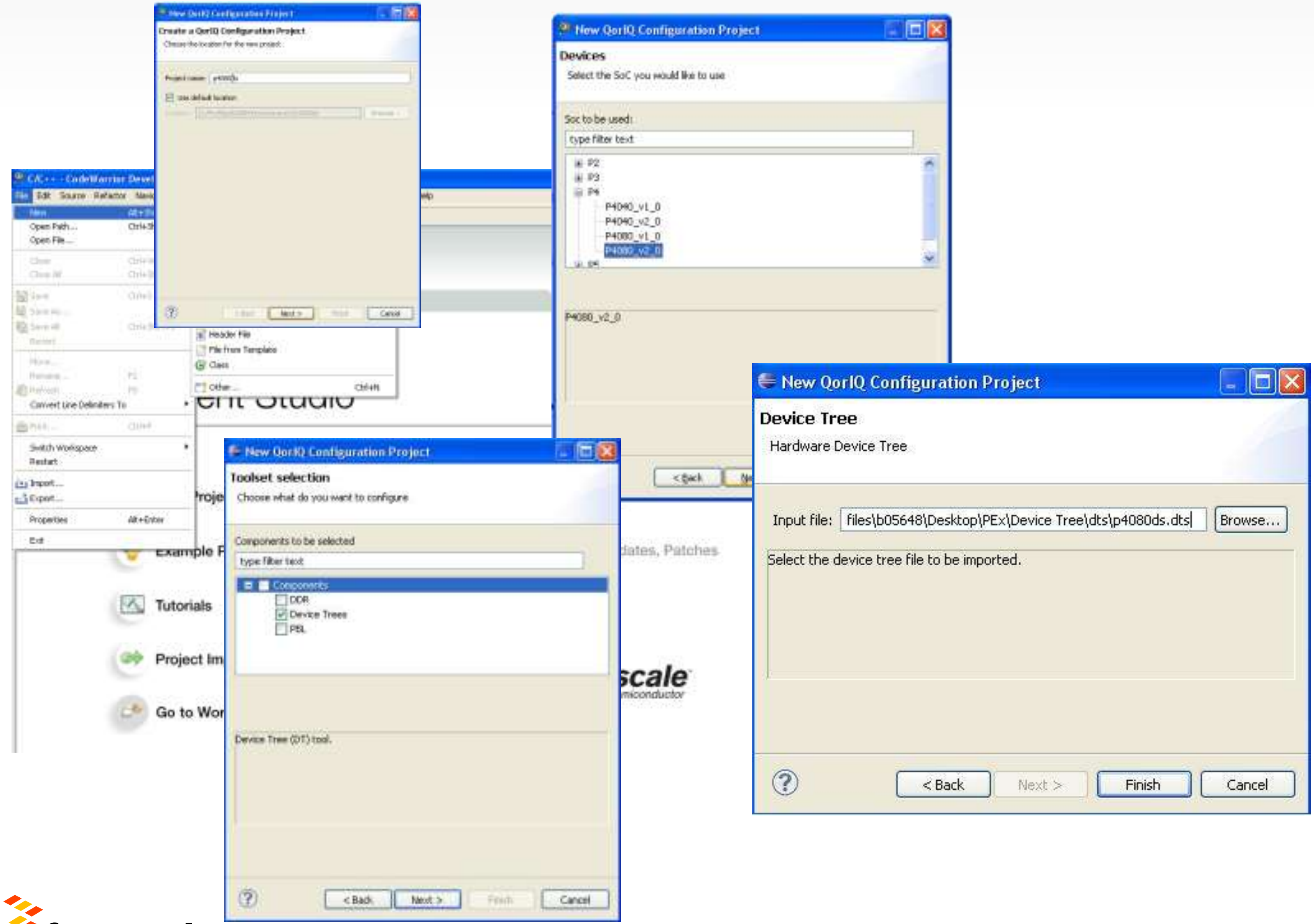
Lab 5: Changing the Hardware Device Tree

Device Tree Hands-on

What we will do:

- Define hardware device tree for P4040 starting from P4080 device tree
  - Import the P4080 (which has 8 cores)
  - Configure to P4040 (which has 4 cores)
  - Note P4080 and P4040 are same SOC otherwise
- Walk through the next slides using QCS Hardware Device Tree editor to solve this scenario

Device Tree Step 1: Create New Project



New QorIQ Configuration Project

Create a QorIQ Configuration Project

Choose the location for the new project

Project name: p4080ds

☐ Use default location

New QorIQ Configuration Project

Devices

Select the SoC you would like to use

Soc to be used:

type filter text

- P2
- P3
- P4
  - P4040_v1_0
  - P4040_v2_0
  - P4080_v1_0
  - P4080_v2_0
- P5

P4080_v2_0

New QorIQ Configuration Project

Toolset selection

Choose what do you want to configure

Components to be selected

type filter text

- ☐ Components
  - ☐ DDR
  - ☑ Device Trees
  - ☐ PBL

Device Tree (DT) tool.

New QorIQ Configuration Project

Device Tree

Hardware Device Tree

Input file: files\b05648\Desktop\PEx\Device Tree\dts\p4080ds.dts

Select the device tree file to be imported.

Device Tree Step 1: Create New Project (cont.)

1. Steps:
   - File -> New -> QorIQ Configuration Project
   - enter Project name -> Next
   - select SoC (p4080_v2_0) -> Next
   - select «Device Trees» component -> Next
   - browse to an existing p4080ds.dts file -> Finish

2. Wait while the device tree component is updating with the imported data

3. Expand *ProcessorExpert.pe* -> *Embedded Components* -> click on *DT1: HWDeviceTree* component

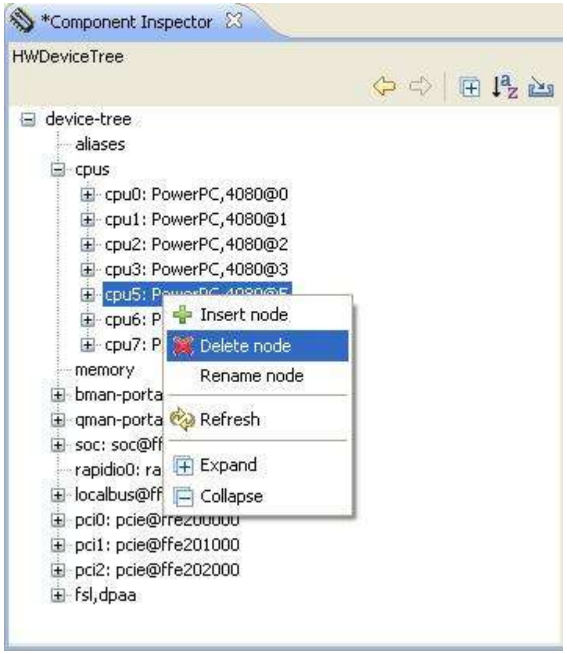
4. The imported dts file is added under *Imported_Files* folder

5. Open generated device tree file *Generated_Code/P4080_v2_0.dts*. At this moment the imported dts and the generated one are identical

Device Tree Step 2: Remove Unnecessary Nodes

6. Search for CPU nodes in the tree view; you should see 8 cores with 3 properties each
7. Delete cpu4-cpu7, you have two options:

- *Using graphical editor*

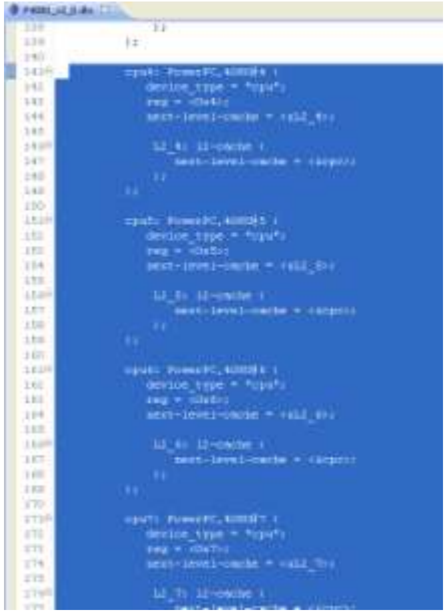


- device-tree
  - aliases
  - cpus
    - cpu0: PowerPC,4080@0
    - cpu1: PowerPC,4080@1
    - cpu2: PowerPC,4080@2
    - cpu3: PowerPC,4080@3
    - cpu5: PowerPC,4080@5
    - cpu6: P
    - cpu7: P
  - memory
  - bman-porta
  - qman-porta
  - soc: soc@ff
  - rapidio0: ra
  - localbus@ff
  - pci0: pcie@ffe200000
  - pci1: pcie@ffe201000
  - pci2: pcie@ffe202000
  - fsl,dpaa

Delete one node at a time.
After each deletion, dts file is automatically generated.

OR

- *Using text editor*

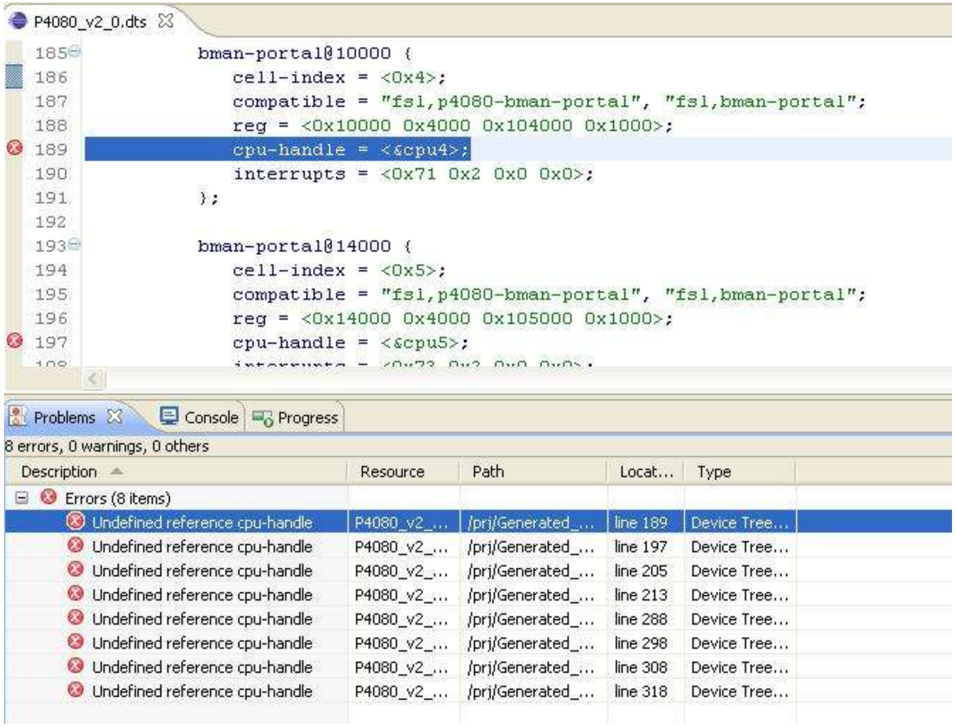


```
        };
    };

    cpu4: PowerPC,4080@4 {
        device_type = "cpu";
        reg = <0x4>;
        next-level-cache = <&L2_4>;

         L2_4: l2-cache {
            next-level-cache = <&cpc>;
         };
    };

    cpu5: PowerPC,4080@5 {
        device_type = "cpu";
        reg = <0x5>;
        next-level-cache = <&L2_5>;

         L2_5: l2-cache {
            next-level-cache = <&cpc>;
         };
    };

    cpu6: PowerPC,4080@6 {
        device_type = "cpu";
        reg = <0x6>;
        next-level-cache = <&L2_6>;

         L2_6: l2-cache {
            next-level-cache = <&cpc>;
         };
    };

    cpu7: PowerPC,4080@7 {
        device_type = "cpu";
        reg = <0x7>;
        next-level-cache = <&L2_7>;

         L2_7: l2-cache {
            next-level-cache = <&cpc>;
```

Select all 4 nodes and press delete.
To reflect the modifications in the graphical editor too, save the file (Ctrl+S).

Device Tree Step 3: Solve Validation Errors

8. Look in the Properties view - There are 8 errors
9. Click on each error and go to the corresponding line in the device tree file
10. There are undefined references in some nodes pointing to the removed cpus:
    bman-portal@10000
    bman-portal@14000
    bman-portal@18000
    bman-portal@1c000
    qportal4
    qportal5
    qportal6
    qportal7
11. Remove the above nodes in a similar manner
12. Save your changes



P4080_v2_0.dts

```
185         bman-portal@10000 {
186             cell-index = <0x4>;
187             compatible = "fsl,p4080-bman-portal", "fsl,bman-portal";
188             reg = <0x10000 0x4000 0x104000 0x1000>;
189             cpu-handle = <&cpu4>;
190             interrupts = <0x71 0x2 0x0 0x0>;
191         };
192
193         bman-portal@14000 {
194             cell-index = <0x5>;
195             compatible = "fsl,p4080-bman-portal", "fsl,bman-portal";
196             reg = <0x14000 0x4000 0x105000 0x1000>;
197             cpu-handle = <&cpu5>;
198             interrupts = <0x73 0x2 0x0 0x0>;
```

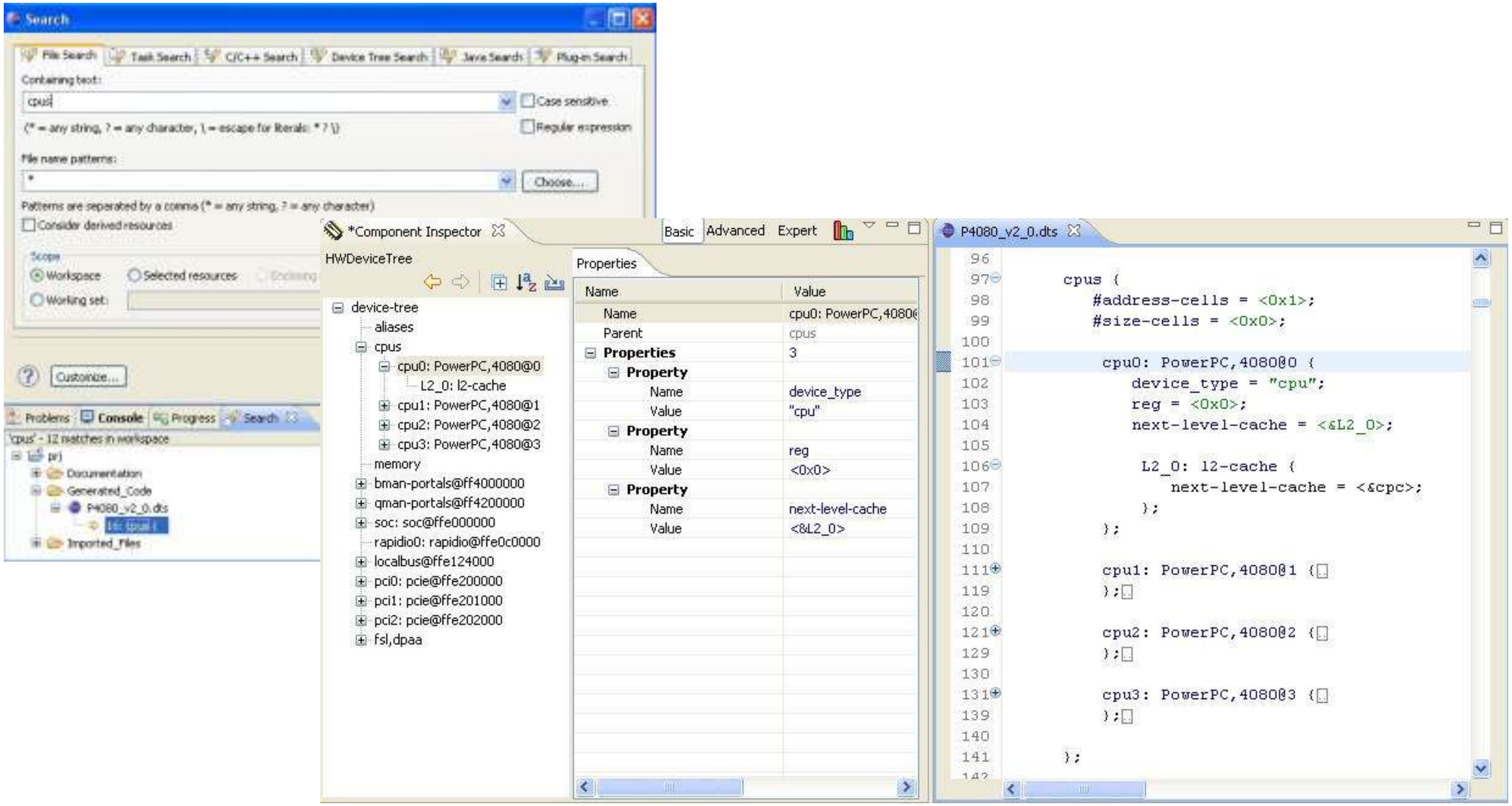
8 errors, 0 warnings, 0 others

Description	Resource	Path	Locat...	Type
Errors (8 items)				
Undefined reference cpu-handle	P4080_v2_...	/prj/Generated_...	line 189	Device Tree...
Undefined reference cpu-handle	P4080_v2_...	/prj/Generated_...	line 197	Device Tree...
Undefined reference cpu-handle	P4080_v2_...	/prj/Generated_...	line 205	Device Tree...
Undefined reference cpu-handle	P4080_v2_...	/prj/Generated_...	line 213	Device Tree...
Undefined reference cpu-handle	P4080_v2_...	/prj/Generated_...	line 288	Device Tree...
Undefined reference cpu-handle	P4080_v2_...	/prj/Generated_...	line 298	Device Tree...
Undefined reference cpu-handle	P4080_v2_...	/prj/Generated_...	line 308	Device Tree...
Undefined reference cpu-handle	P4080_v2_...	/prj/Generated_...	line 318	Device Tree...

Device Tree Step 4: P4040 Device Tree

13. Go to *Search* menu -> select *Device Tree Search* tab -> enter *cpus* text -> press *Search*
14. In *Search* view select the found matches associated with the generated file

- New device tree has 4 cores and looks as follows:



Search

Containing text: cpus

☐ Case sensitive

☐ Regular expression

(* = any string, ? = any character, \ = escape for literals: * ? \)

File name patterns: *

Patterns are separated by a comma (* = any string, ? = any character)

☐ Consider derived resources

Scope: ● Workspace ○ Selected resources ○ Enclosing ○ Working set:

'cpus' - 12 matches in workspace

```
pri
  Documentation
  Generated_Code
    P4080_v2_0.dts
      16: cpus {
  Imported_Files
```

*Component Inspector

HWDeviceTree

```
device-tree
  aliases
  cpus
    cpu0: PowerPC,4080@0
      L2_0: l2-cache
    cpu1: PowerPC,4080@1
    cpu2: PowerPC,4080@2
    cpu3: PowerPC,4080@3
  memory
  bman-portals@ff4000000
  qman-portals@ff4200000
  soc: soc@ffe000000
  rapidio0: rapidio@ffe0c0000
  localbus@ffe124000
  pci0: pcie@ffe200000
  pci1: pcie@ffe201000
  pci2: pcie@ffe202000
  fsl,dpaa
```

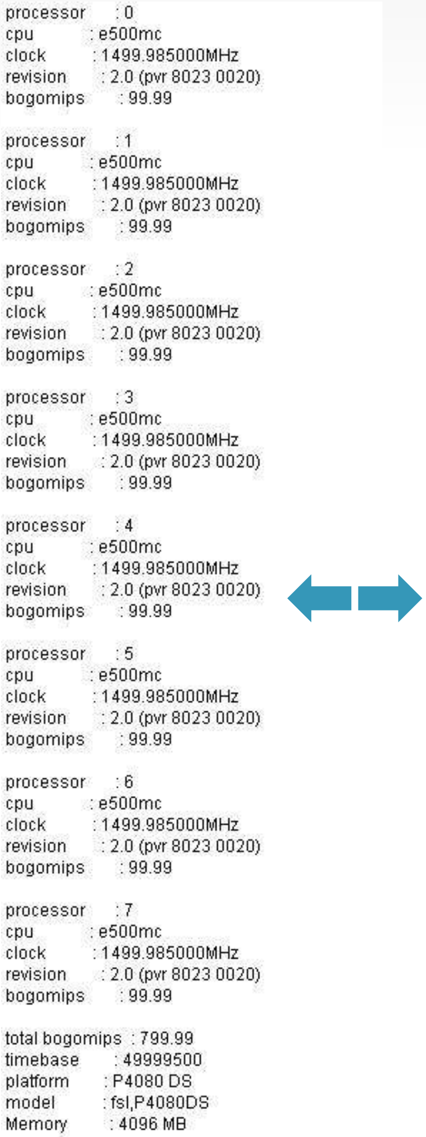
Properties

Name	Value
Name	cpu0: PowerPC,4080(
Parent	cpus
⊟ Properties	3
⊟ Property	
Name	device_type
Value	"cpu"
⊟ Property	
Name	reg
Value	<0x0>
⊟ Property	
Name	next-level-cache
Value	<&L2_0>

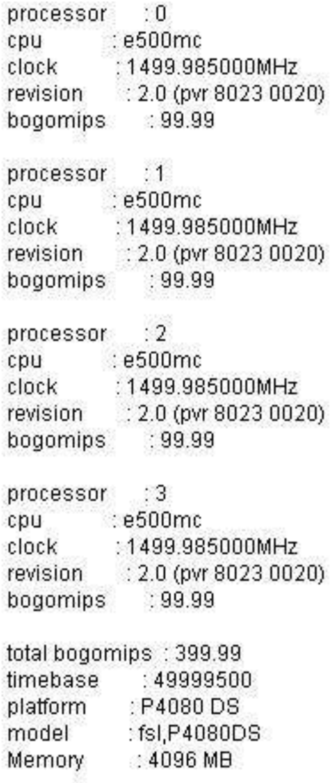
P4080_v2_0.dts

```

    cpus {
       #address-cells = <0x1>;
       #size-cells = <0x0>;

        cpu0: PowerPC,4080@0 {
           device_type = "cpu";
           reg = <0x0>;
           next-level-cache = <&L2_0>;

            L2_0: l2-cache {
               next-level-cache = <&cpc>;
            };
        };

        cpu1: PowerPC,4080@1 {□
        };□

        cpu2: PowerPC,4080@2 {□
        };□

        cpu3: PowerPC,4080@3 {□
        };□

    };
```

Device Tree Step 5: Apply and Test Changes



```
processor     : 0
cpu           : e500mc
clock         : 1499.985000MHz
revision      : 2.0 (pvr 8023 0020)
bogomips      : 99.99

processor     : 1
cpu           : e500mc
clock         : 1499.985000MHz
revision      : 2.0 (pvr 8023 0020)
bogomips      : 99.99

processor     : 2
cpu           : e500mc
clock         : 1499.985000MHz
revision      : 2.0 (pvr 8023 0020)
bogomips      : 99.99

processor     : 3
cpu           : e500mc
clock         : 1499.985000MHz
revision      : 2.0 (pvr 8023 0020)
bogomips      : 99.99

processor     : 4
cpu           : e500mc
clock         : 1499.985000MHz
revision      : 2.0 (pvr 8023 0020)
bogomips      : 99.99

processor     : 5
cpu           : e500mc
clock         : 1499.985000MHz
revision      : 2.0 (pvr 8023 0020)
bogomips      : 99.99

processor     : 6
cpu           : e500mc
clock         : 1499.985000MHz
revision      : 2.0 (pvr 8023 0020)
bogomips      : 99.99

processor     : 7
cpu           : e500mc
clock         : 1499.985000MHz
revision      : 2.0 (pvr 8023 0020)
bogomips      : 99.99

total bogomips  : 799.99
timebase        : 49999500
platform        : P4080 DS
model           : fsl,P4080DS
Memory          : 4096 MB
```



```
processor     : 0
cpu           : e500mc
clock         : 1499.985000MHz
revision      : 2.0 (pvr 8023 0020)
bogomips      : 99.99

processor     : 1
cpu           : e500mc
clock         : 1499.985000MHz
revision      : 2.0 (pvr 8023 0020)
bogomips      : 99.99

processor     : 2
cpu           : e500mc
clock         : 1499.985000MHz
revision      : 2.0 (pvr 8023 0020)
bogomips      : 99.99

processor     : 3
cpu           : e500mc
clock         : 1499.985000MHz
revision      : 2.0 (pvr 8023 0020)
bogomips      : 99.99

total bogomips  : 399.99
timebase        : 49999500
platform        : P4080 DS
model           : fsl,P4080DS
Memory          : 4096 MB
```

- On a Linux® machine, create the device tree binaries before and after changes
- Boot the Linux kernel on a p4080DS board
- Check the number of CPUs that Linux kernel sees before and after changes (use /proc/cpuinfo command)
- You should obtain the results from left side, only 4 cores are in use with the new device tree

Conclusions

- **The sequence of steps for modifying hardware device trees has been presented**

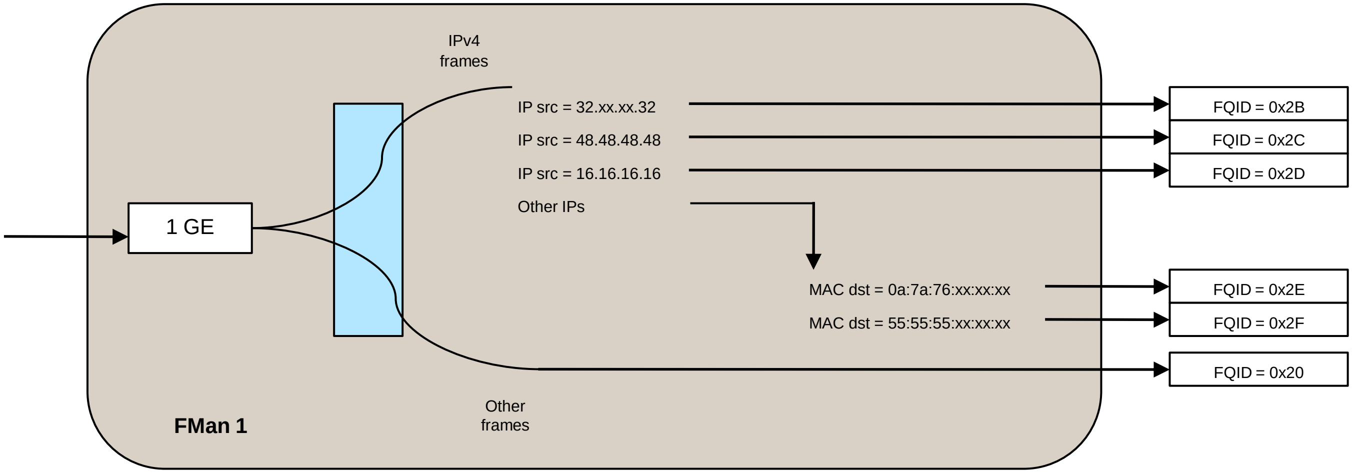
- **The benefits of using Hardware Device Tree tool are:**
  - First device tree editor including two modes for editing, GUI and text
  - Easy to understand device trees structure due to the visual representation
  - Supports device tree bindings and validation
  - Allows users to add their own device trees
  - Provides features for all the main aspects of hardware device trees
  - It is an editor and a validation tool for creating valid and well-formed device trees
  - Works on Linux and Windows hosts

DPAA Configuration

Lab 6: Using Data Path Graphs to configure the DPAA

DPAA Hands-on – Problem Statement

- **Receive 1GE traffic on first FMAN**
- **Split incoming traffic in IP frames and others**
- **Frames with specified IP source are directed in specified FQIDs and then in SW Portal 0**

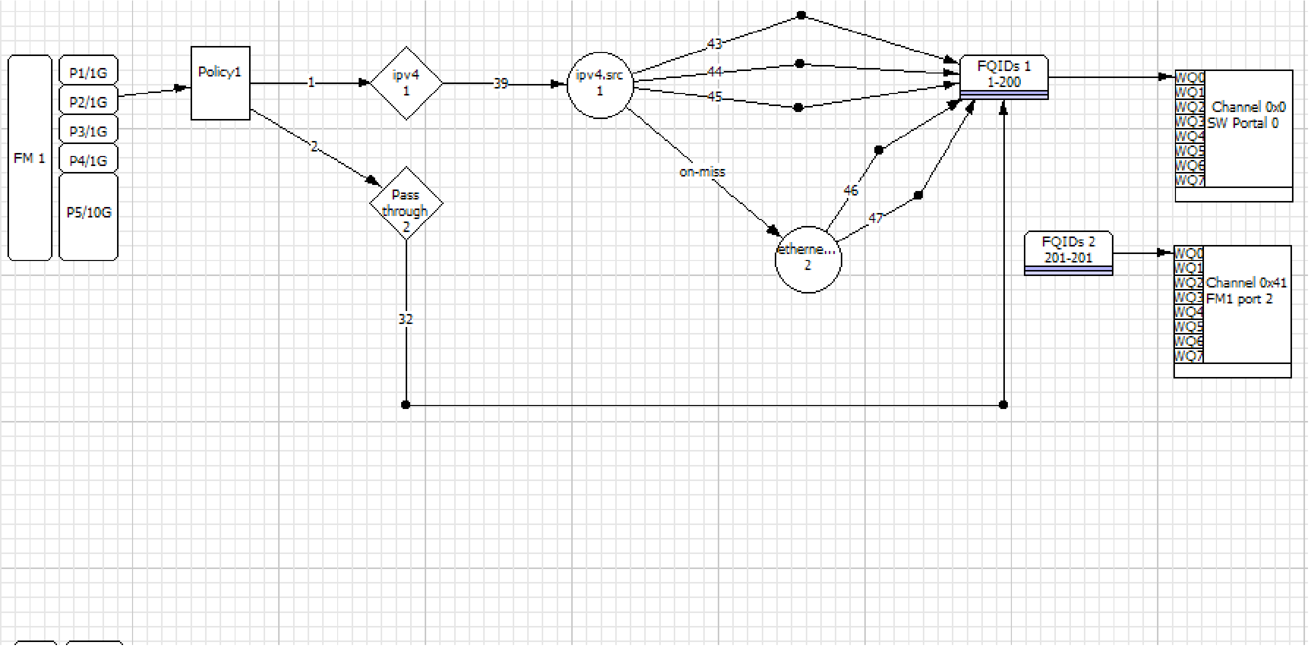


DPAA Hands-on – QCS Solution

- **Build PCD configuration according to hands-on requirements**
- **Additional 1 FQ ranges and 1 FMan port channel to be used for transmission**

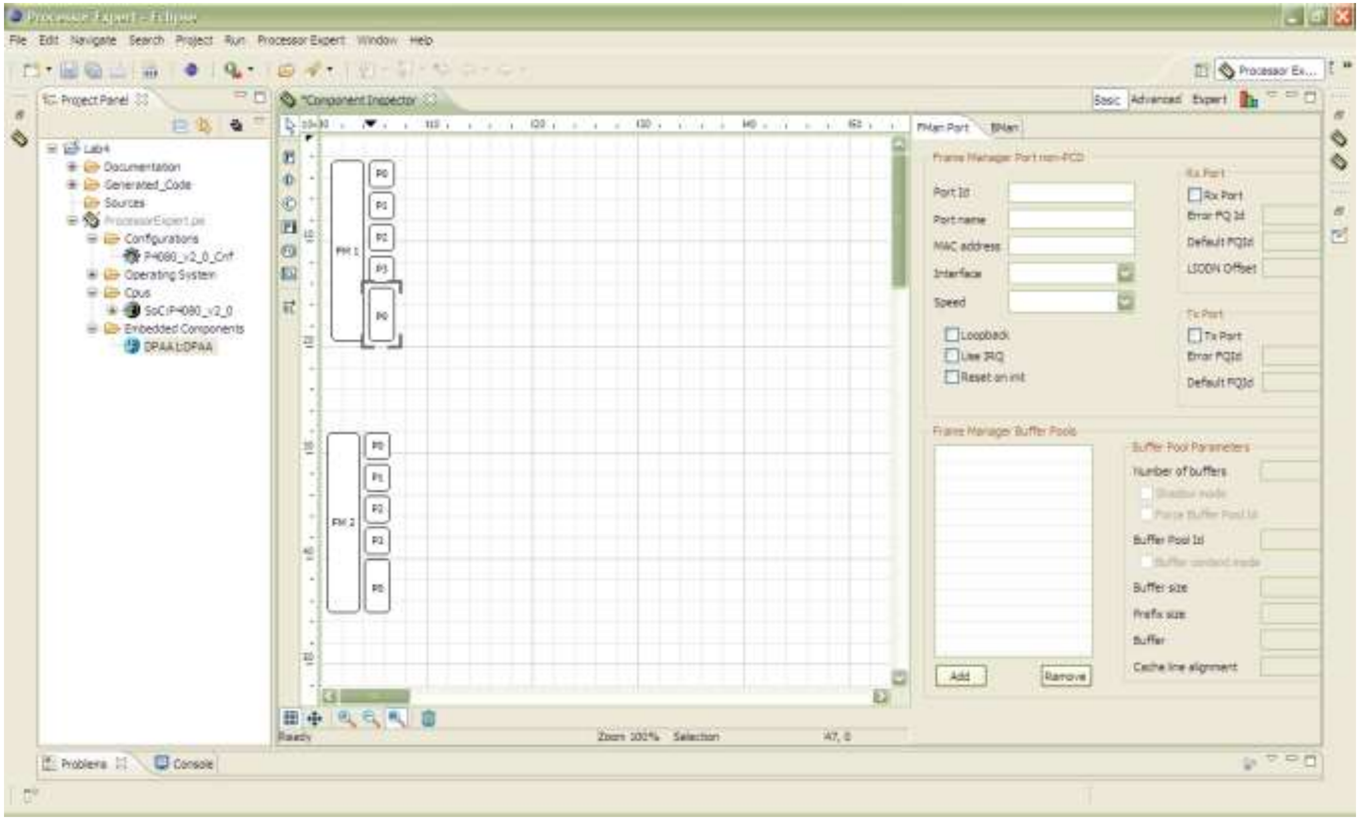
FMan1 PCD flow:

- Traffic received by FM1 port1 is split in IP frames and other frames by: Policy1, ipv4 distribution
- IP frames classified in one of the 3 defined ranges of IP source are directed into FQIDs 43-45
- All other IP frames are classified by MAC destination and are directed into FQIDs 46 & 47
- All other non IP frames are directed in FQID 32

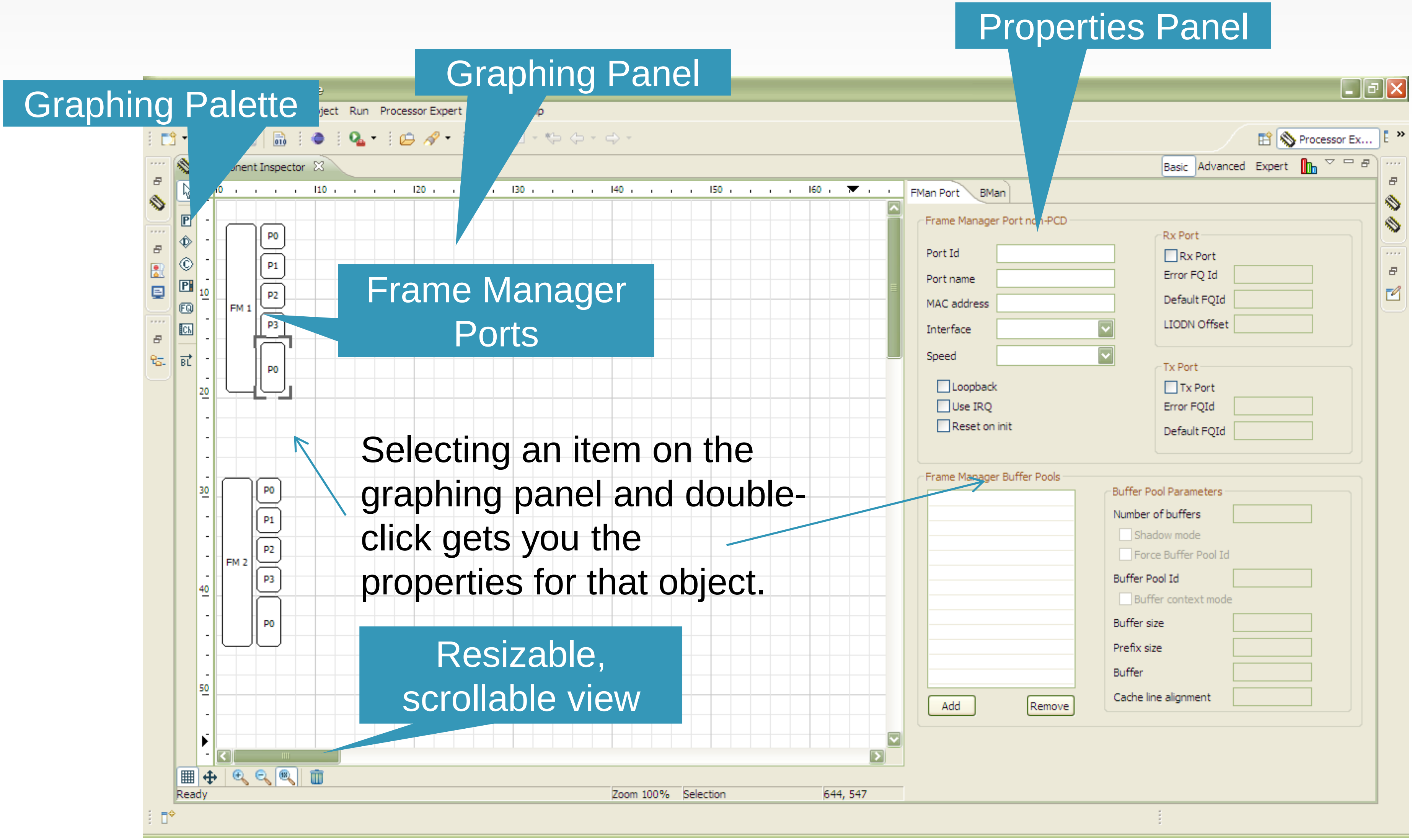


Lab 4: DPAA Hands-on

1. Create a new QCS project called Lab4
    1. Choose P4080 rev2.0
    2. Choose a DPAA component and select empty component
2. Maximize the DPAA Component Inspector



Explore the Graphing Interface



Graphing Palette

Graphing Panel

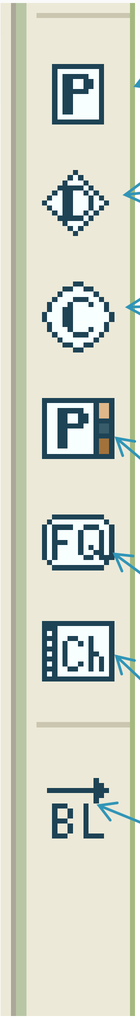
Properties Panel

Frame Manager Ports

Selecting an item on the graphing panel and double-click gets you the properties for that object.

Resizable, scrollable view

The Graphing Palette



Policy – establishes the order of distribution

Distribution – sorts incoming frames

Classification – defines the classification and software portal

Policer – allows enforcement

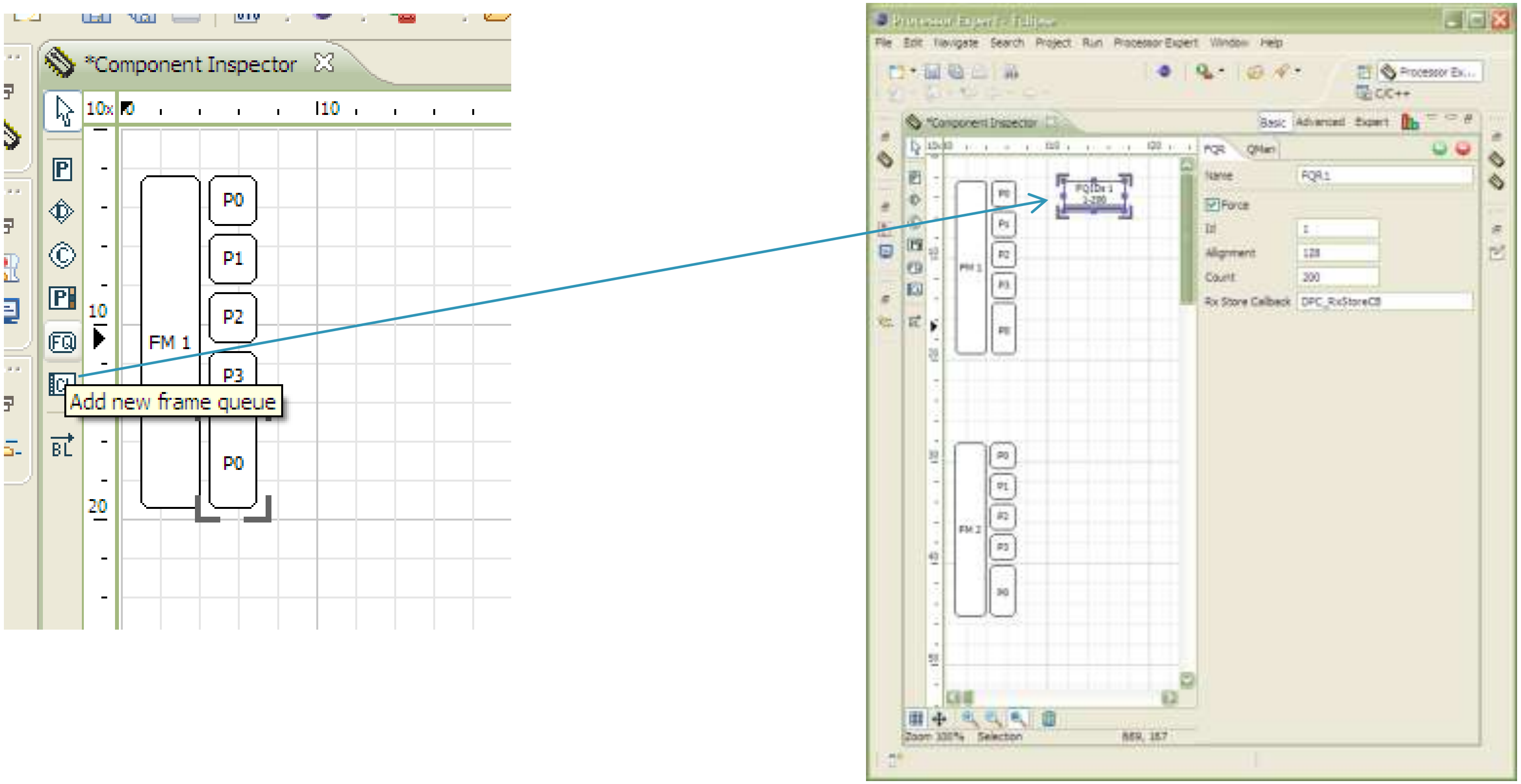
Frame Queue(s)

Software Portals & Channels

Link between objects

Add a Set of Frame Queues

1. QMan configuration:
   1. Add a FQIDs range (FQR1) for enqueued frames and configure: *FQId=1 and count=200*
   2. Switch to QMan tab and make the following settings: *Total FQIDs=150000, Fqd/Pfdr mem partition=Primary DDR non-cacheable*



Name: FQR1

☑ Force

Id: 1

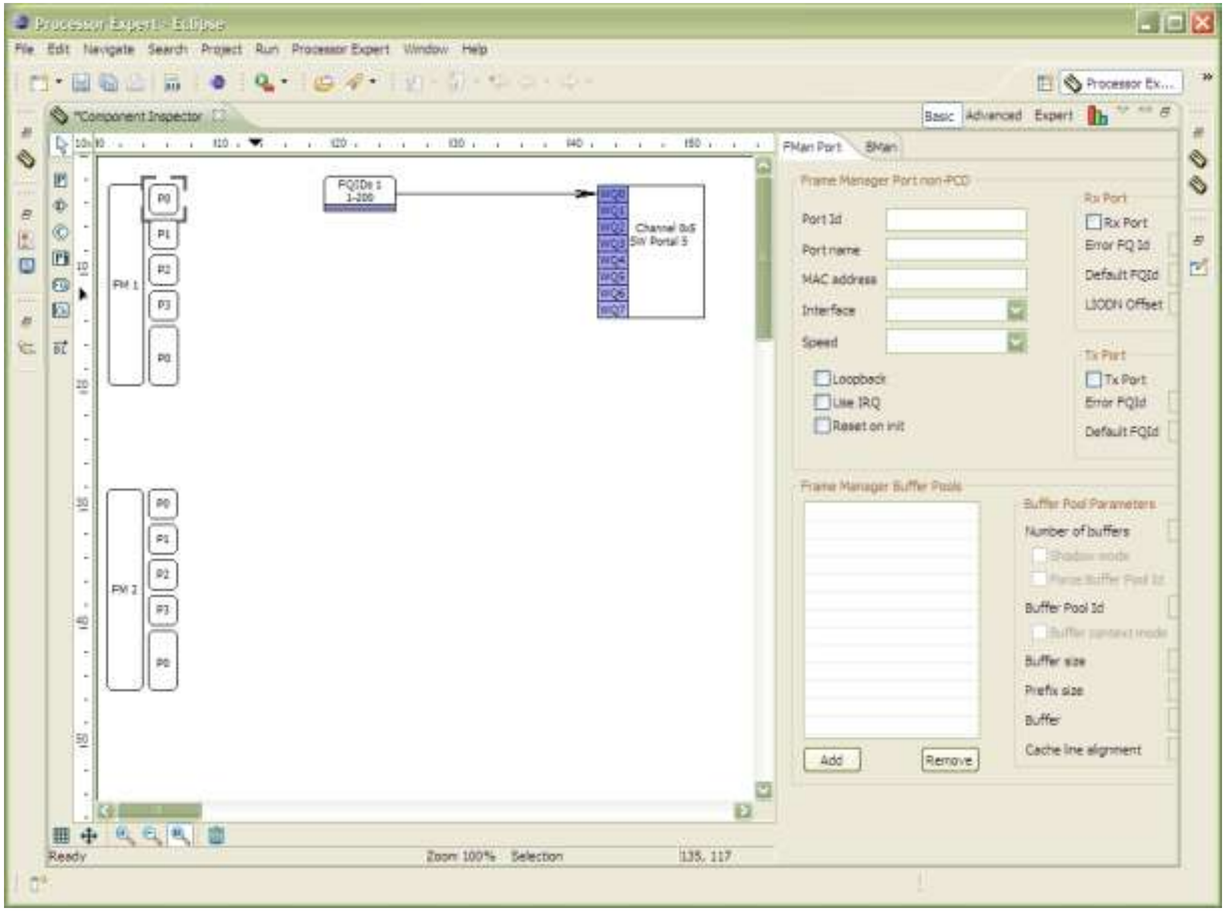
Alignment: 128

Count: 200

Rx Store Callback: DPC_RxStoreCB

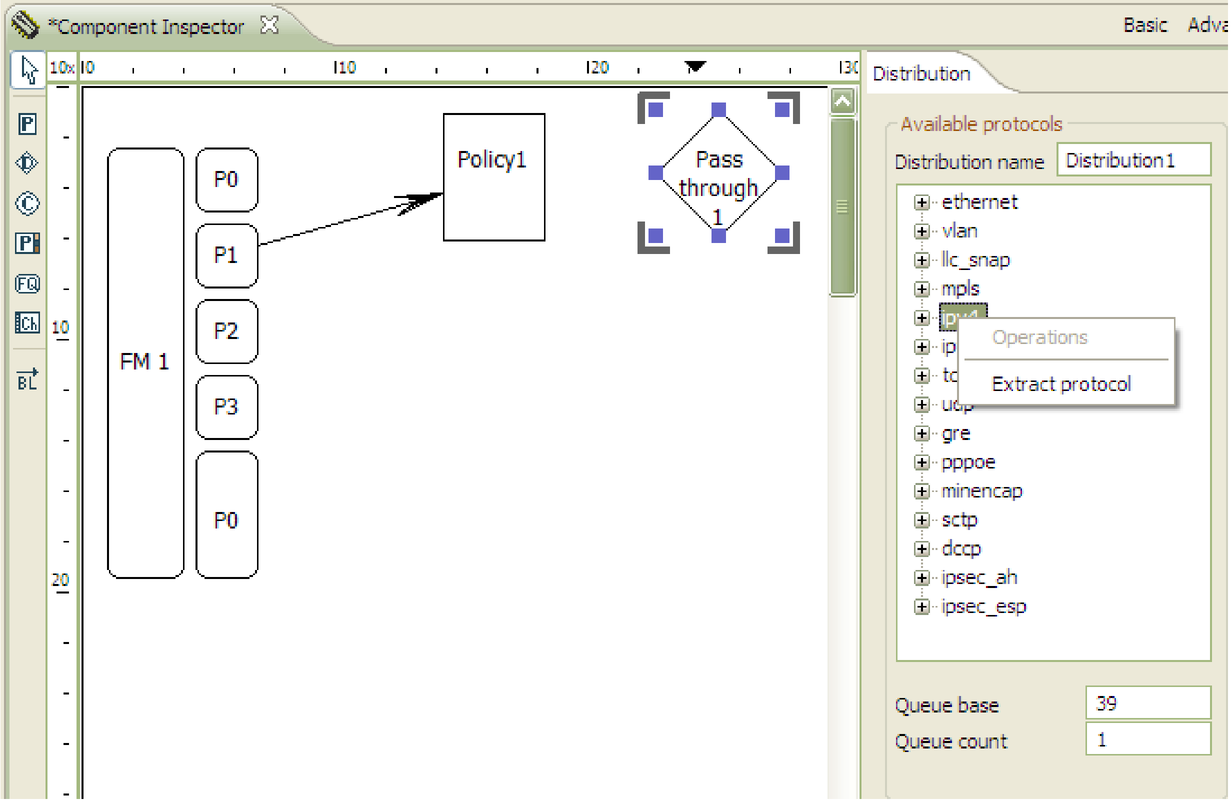
Configure a SW Portal Configuration:

1. Add SW Portal0 channel
2. Link FQIDs1 to WQ0 of SW Portal0



Define the Parse Classify Distribute Configuration

- Add a Policy1 to split IP frames traffic and then a link from port FM1 P2 to receive incoming frames
- Add a distribution for IP frames and extract IPv4 protocol then configure: *QBase = 39*



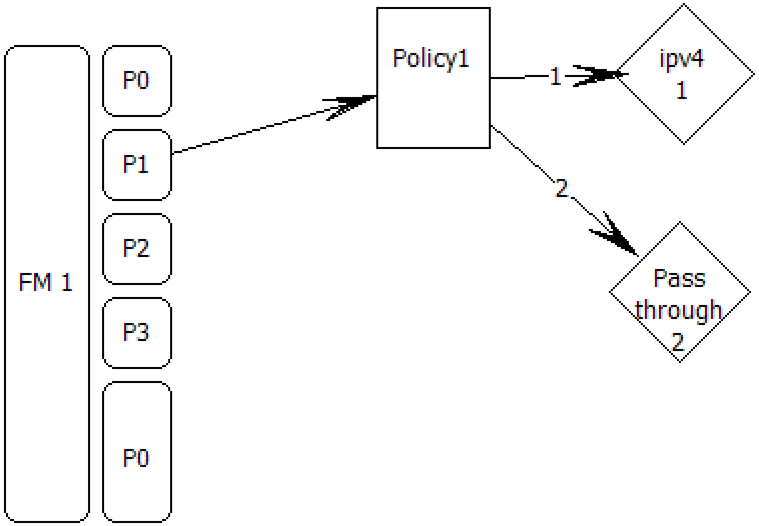
Distribution name: Distribution1

Queue base: 39

Queue count: 1

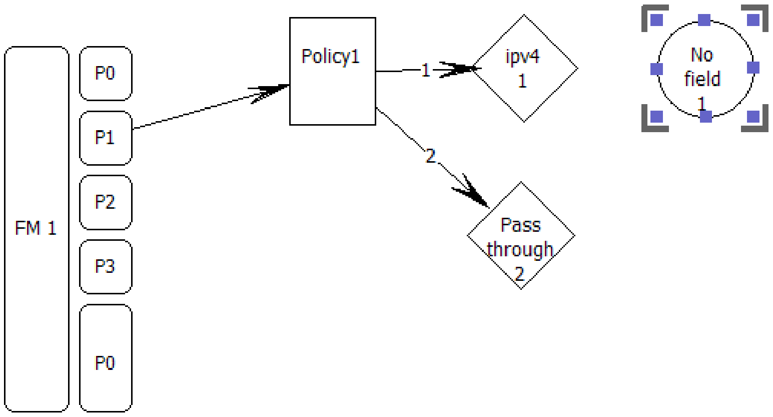
Link the Distributions to the Policy

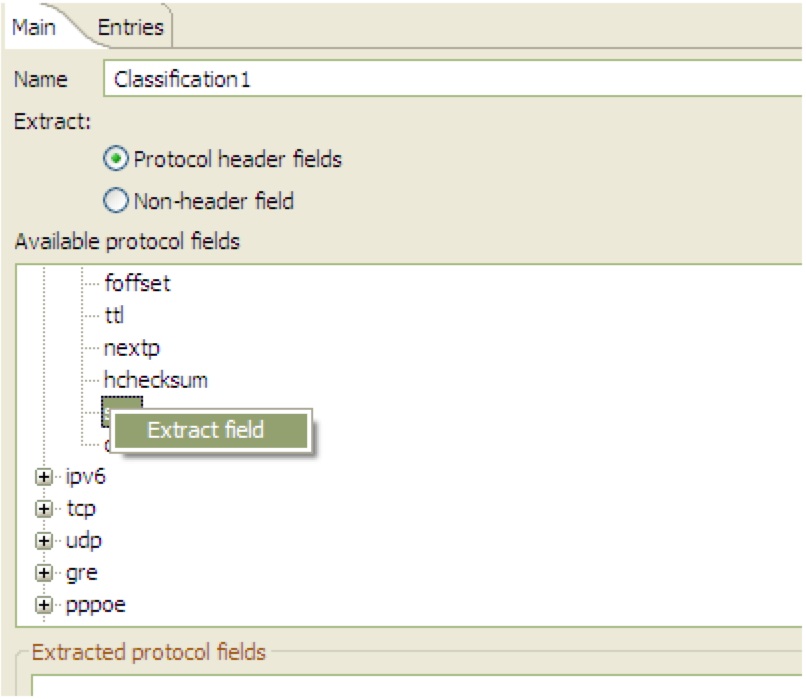
1. Add a Pass through distribution for other non IP frames and configure: *QBase = 32*
2. Link Policy1 to IPv4 distribution and then to Pass through in this order



Add a Classification

- Add a classification for IP source and extract ipv4.src and remove ethernet.type





Name: Classification1

Extract:

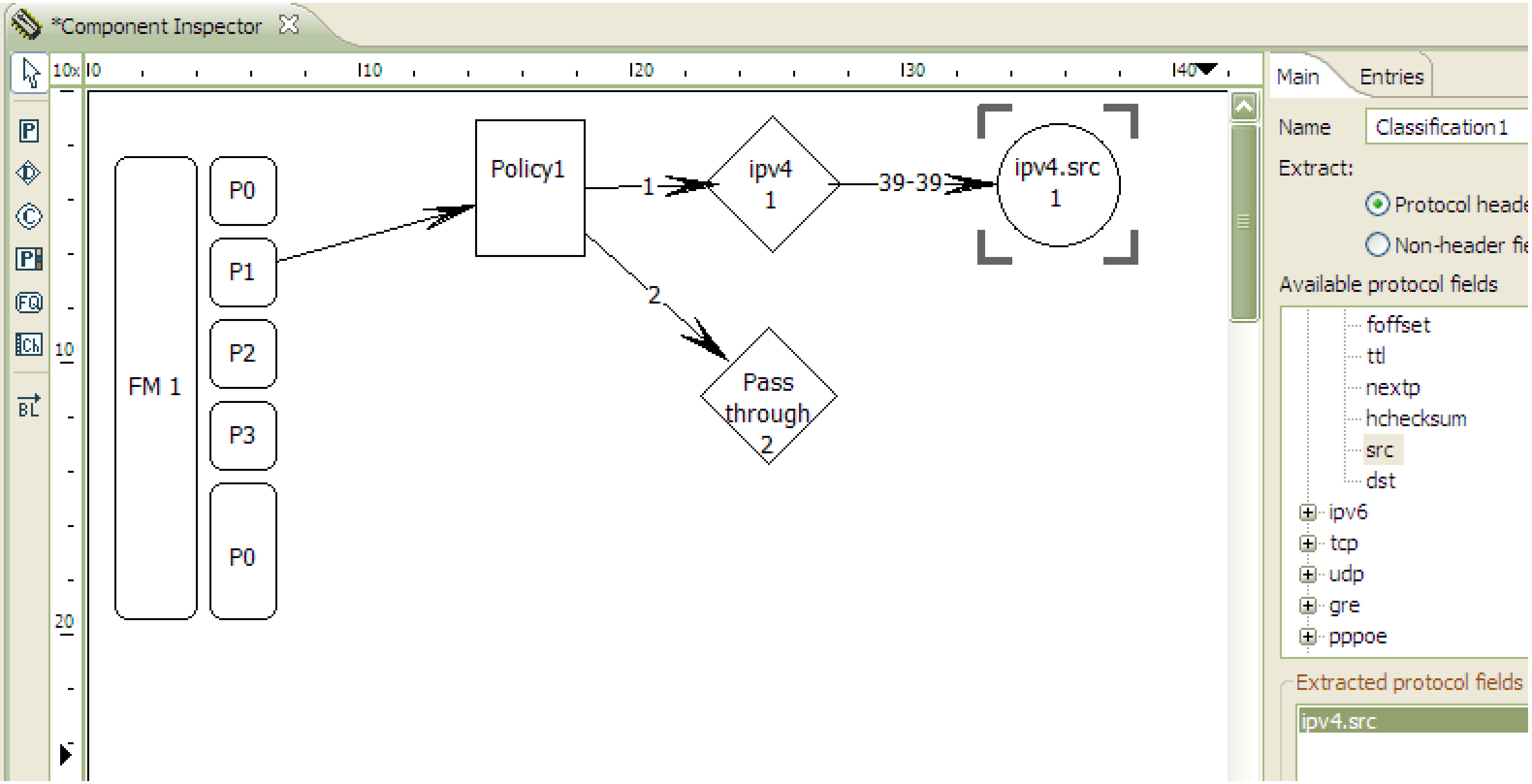
● Protocol header fields

○ Non-header field

Available protocol fields

Extracted protocol fields

Link IPv4 Distribution to ipv4.src Classification



Main | Entries

Name: Classification1

Extract:

● Protocol heade

○ Non-header fie

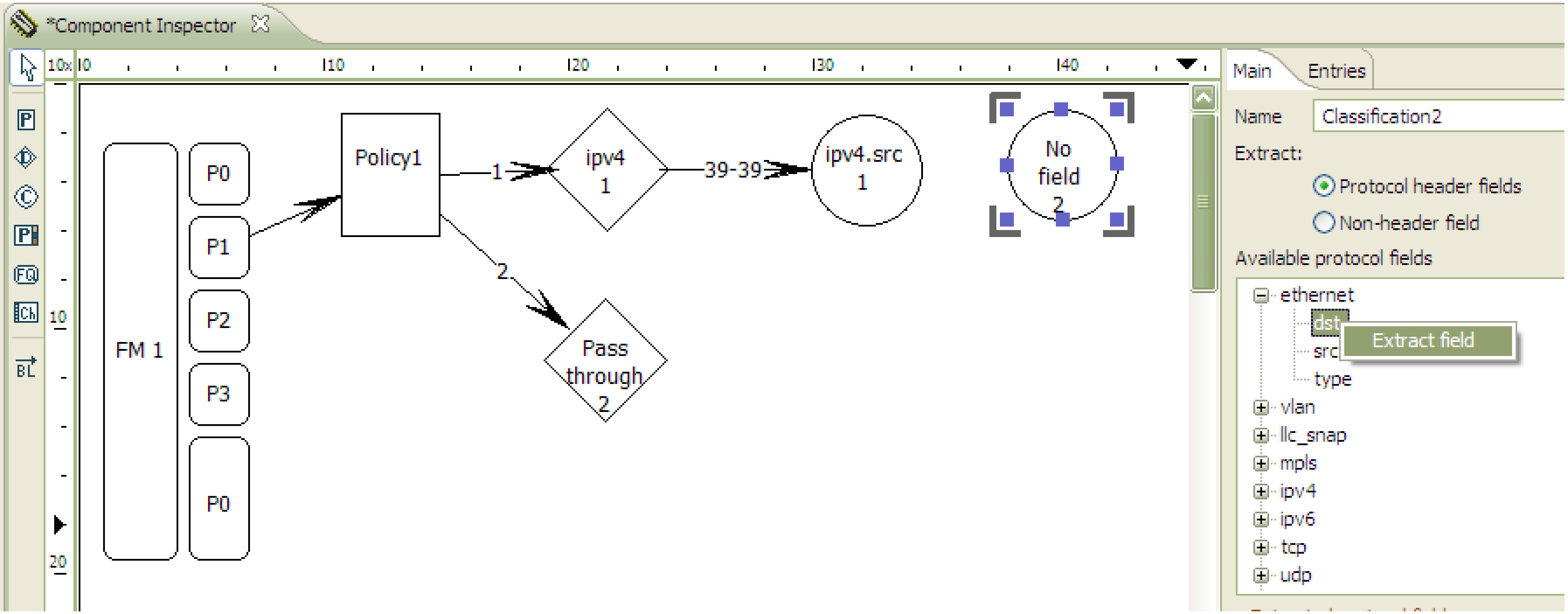
Available protocol fields

- foffset
- ttl
- nextp
- hchecksum
- src
- dst
- ⊞ ipv6
- ⊞ tcp
- ⊞ udp
- ⊞ gre
- ⊞ pppoe

Extracted protocol fields

ipv4.src

Add a Classification and Extract ethernet.dst and remove ethernet.type



Name: Classification2

Extract:

● Protocol header fields

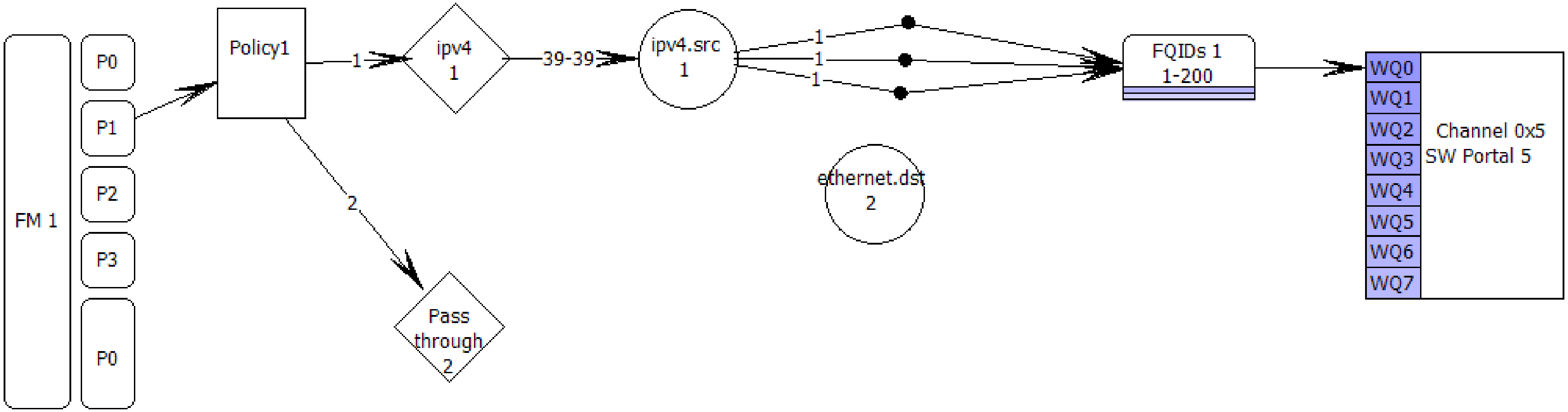
○ Non-header field

Available protocol fields

- ⊟ ethernet
  - dst
  - src
  - type
- ⊞ vlan
- ⊞ llc_snap
- ⊞ mpls
- ⊞ ipv4
- ⊞ ipv6
- ⊞ tcp
- ⊞ udp

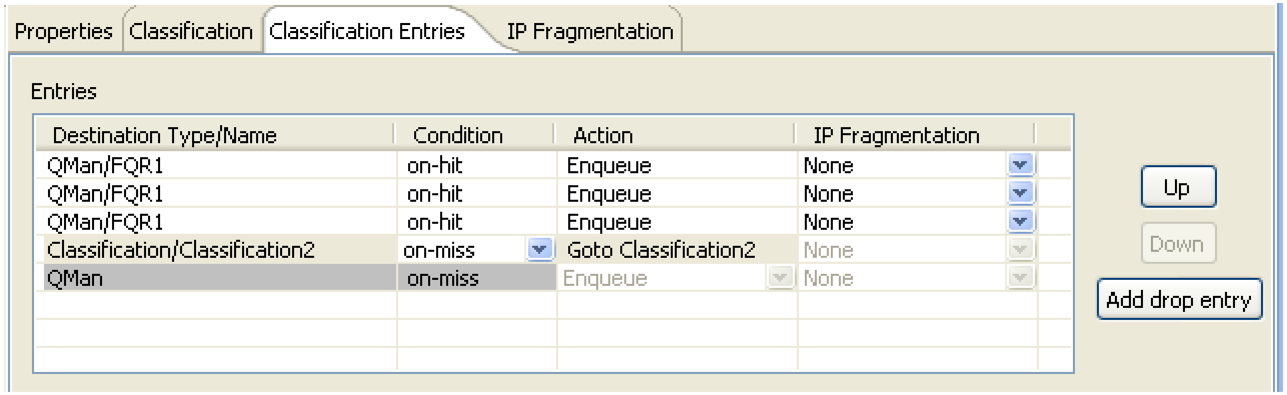
Establish 3 Classification Paths

1. Link ipv4.src classification to ethernet.dst classification
2. Draw 3 links from ipv4.src classification to FQIDs1



Configure ipv4.src Classification Entries

1. Select the ipv4.src classification
2. Go to the Entries tab in Property Panel
3. Set ethernet.dst Classification2: *on-miss entry*



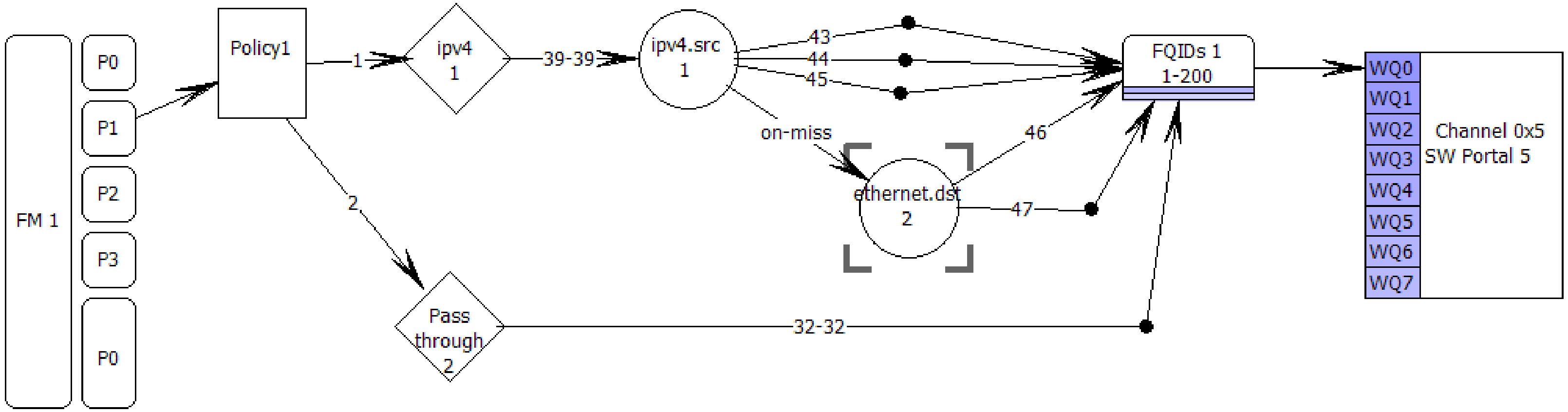
Entries

Destination Type/Name	Condition	Action	IP Fragmentation
QMan/FQR1	on-hit	Enqueue	None
QMan/FQR1	on-hit	Enqueue	None
QMan/FQR1	on-hit	Enqueue	None
Classification/Classification2	on-miss	Goto Classification2	None
QMan	on-miss	Enqueue	None

4. To FQIDs1:
   1. *Data=0x20EEEE20 / Mask=FF0000FF / Queue base=0000002B*
   2. *Data=0x30303030 / Mask=FFFFFFFF / Queue base=0000002C*
   3. *Data=0x10101010 / Mask=FFFFFFFF / Queue base=0000002D*

Configure the ethernet.dst Entries

1. Link Pass through distribution to FQIDs1
2. Draw 2 links from ethernet.dst classification to FQIDs1
3. Configure ethernet.dst classification Entries to FQIDs1:
   1. *Data= 0a7a76000000 / Mask=FFFFFF000000 / Queue base=0000002E*
   2. *Data= 555555000000 / Mask= FFFFFF000000 / Queue base= 0000002F*

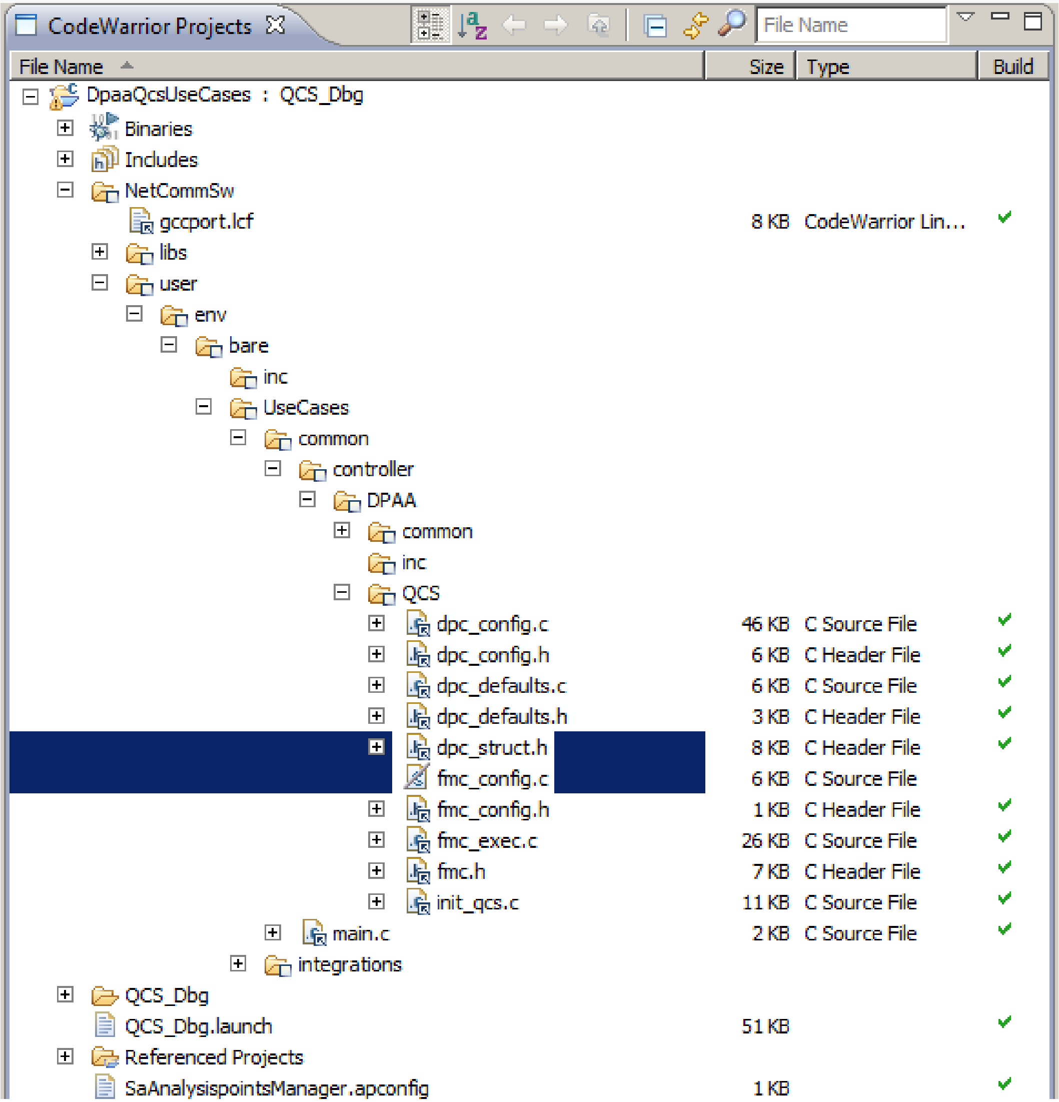


Configure the FMan Port

6. FMan 1 Port 1 configuration:
    1. *Port name= fm0port01 / MAC address=00:04:9f:00:02:66 / Interface=RGMII / Speed=1Gbps*
    2. Enable *Loopback* and *Reset on Init*
    3. Enable Rx port: *Error FQId=20 / Default FQId=20*
    4. Enable Tx port: *Error FQId=40 / Default FQId=0*
    5. Add and use BufferPool0: *Number of buffers=100 / Buffer pool Id=0 / Buffer Size=51200*
7. FMan 1 Tx configuration:
    1. Channel for FM1 Port 1:
        1. Add a FQIDs range (FQR2) used to transmit frames on FM1 and configure: *FQId=201 and count=1*
        2. Add FM1 port1 channel
        3. Link FQIDs2 to FM1 port1 channel WQ0

DPAA Hands-on Generated Code Usage

- HW configuration
  - P4080 v2.0
  - Loopback on FM ports used
- SW configuration
  - CodeWarrior PA 10.1.2
  - NetCommSw v4.5
- Usecase running
  - Import DpaaQcsUseCases, NetCommSw and UserEnv for P4080
  - Replace generated code files:
  - *\user\env\bare\UseCases\common\controller\DPAA\QCS*
    - *fmc_config.c*
    - *dpc_struct.h*
  - Clean & build project
  - Run on target
  - Receive output



```
File Name                                              Size  Type                 Build
⊟ DpaaQcsUseCases : QCS_Dbg
    ⊞ Binaries
    ⊞ Includes
    ⊟ NetCommSw
          gccport.lcf                                  8 KB  CodeWarrior Lin...   ✓
        ⊞ libs
        ⊟ user
            ⊟ env
                ⊟ bare
                      inc
                    ⊟ UseCases
                        ⊟ common
                            ⊟ controller
                                ⊟ DPAA
                                    ⊞ common
                                      inc
                                    ⊟ QCS
                                        ⊞ dpc_config.c      46 KB  C Source File   ✓
                                        ⊞ dpc_config.h       6 KB  C Header File   ✓
                                        ⊞ dpc_defaults.c     6 KB  C Source File   ✓
                                        ⊞ dpc_defaults.h     3 KB  C Header File   ✓
                                        ⊞ dpc_struct.h       8 KB  C Header File   ✓
                                          fmc_config.c       6 KB  C Source File
                                        ⊞ fmc_config.h       1 KB  C Header File   ✓
                                        ⊞ fmc_exec.c        26 KB  C Source File   ✓
                                        ⊞ fmc.h              7 KB  C Header File   ✓
                                        ⊞ init_qcs.c        11 KB  C Source File   ✓
                            ⊞ main.c                         2 KB  C Source File   ✓
                        ⊞ integrations
    ⊞ QCS_Dbg
      QCS_Dbg.launch                                  51 KB                        ✓
    ⊞ Referenced Projects
      SaAnalysispointsManager.apconfig                 1 KB                        ✓
```

DPAA Hands-on Usecase Output Report

Use a serial terminal to receive usecase output

Input frames

- Frames transmitted by usecase on FMAN port2: (5 frames on **FM0 Port**)
  - IPV4
  - IPV4/UDP
  - VLAN
  - ARP
  - IPV4/TCP

Output report

- Frames received are enqueued according to PCD:

FM0 Port1:

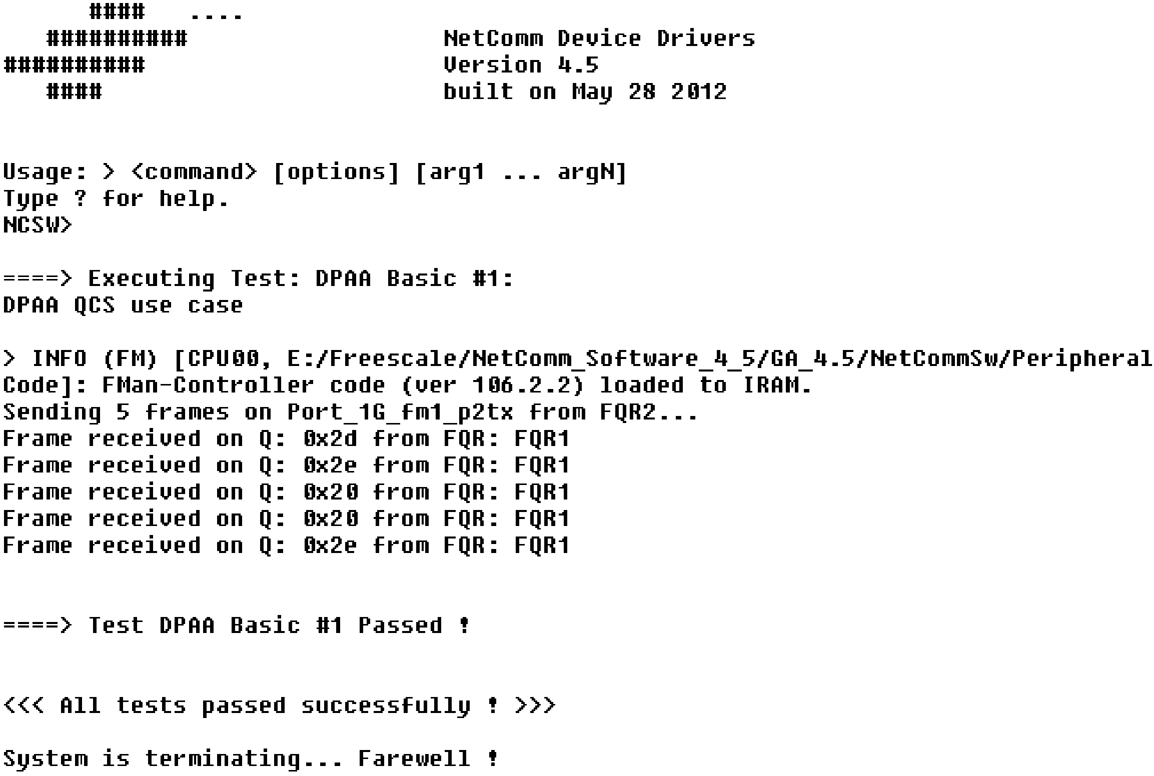
FQID=45(0x2D)

FQID=46(0x2E)

FQID=32(0x20)

FQID=32(0x20)

FQID=46(0x2E)



```
      ####   ....
   ##########               NetComm Device Drivers
##########                  Version 4.5
   ####                     built on May 28 2012


Usage: > <command> [options] [arg1 ... argN]
Type ? for help.
NCSW>

====> Executing Test: DPAA Basic #1:
DPAA QCS use case

> INFO (FM) [CPU00, E:/Freescale/NetComm_Software_4_5/GA_4.5/NetCommSw/Peripheral
Code]: FMan-Controller code (ver 106.2.2) loaded to IRAM.
Sending 5 frames on Port_1G_fm1_p2tx from FQR2...
Frame received on Q: 0x2d from FQR: FQR1
Frame received on Q: 0x2e from FQR: FQR1
Frame received on Q: 0x20 from FQR: FQR1
Frame received on Q: 0x20 from FQR: FQR1
Frame received on Q: 0x2e from FQR: FQR1


====> Test DPAA Basic #1 Passed !


<<< All tests passed successfully ! >>>

System is terminating... Farewell !
```

DPAA Hands-on Output Analysis

Output analysis

- FMan0:

  Frame1: IPV4 Frame - IP src = 16.16.16.16 : IP frame -> IP miss -> MAC match -> Enqueued: FQID=45(0x2D)

  Frame2: IPV4/UDP Frame - MAC dst = 0a:7a:76:5b:67:e9 : IP frame -> IP miss -> MAC match -> Enqueued: FQID=46(0x2E)

  Frame3: VLAN/non-IPV4 Frame: non-IP frame -> Enqueued: FQID=32(0x20)

  Frame4: ARP/non-IPV4 Frame: non-IP frame -> Enqueued: FQID=32(0x20)

  Frame2: IPV4/TCP Frame - MAC dst = 0a:7a:76:5b:67:e9 : IP frame -> IP miss -> MAC match -> Enqueued: FQID=46(0x2E)

DPAA Hands-on Conclusions

- Outcome generated code for DPAA Hands-on:
  - In order to accomplish DPAA hands-on requirements the following initialization code must be written:
    - 220 lines of XML code
    - 750 lines of C code
- Benefits of using QCS DPAA:
  - By using QCS DPAA tool the same result can be accomplished as follows:
    - Requested configuration accomplished in approximately ten minutes
    - Configuration done by using a few mouse clicks and visual parameters settings
    - Provides an easy-to-understand overview of the entire DPAA hands-on architecture
    - DPAA tool helps accomplish your desired configuration
      - by highlighting valid choices and prevent you making invalid selections
      - by performing automatic constraints checking
      - by providing instant access to configuration settings
      - by displaying relevant summary of current configuration
      - by immediate code generation at request in any stage of your work

freescale™

飞思卡尔