NXP Semiconductors
Application Note

Document Number: AN5260

PBL Configuration using QCVS

1. Introduction

This document describes the steps required to configure pre-boot loader (PBL) on NXP QorIQ platform using the PBL tool included in QorIQ Configuration and Validation Suite (QCVS).

This document explains:

- Purpose of the QCVS PBL tool
- How to configure PBL using the PBL tool
- PBL tool limitations

Contents

2. Preliminary background

The QCVS PBL tool provides you a graphical user interface (GUI) for editing a PBL binary in a decoded form. The PBL binary can be created from scratch or imported from an existing reset configuration word (RCW) memory dump. Such memory dumps can be obtained using U-Boot.

The RCW data contains reset configuration information that PBL loads from a memory device during power-on or hardware reset. All data read from the RCW source is written to the RCW status registers by PBL. If RCW selects pre-boot initialization (PBI), then the PBI commands are processed and routed to CCSR, DDR, and other memory spaces.

The PBL tool operates in the context of documented PBL configuration constraints and errata and prevents the user from violating them. The output of the PBL tool is a PBL binary that can be used to pre-program the platform.

3. Creating a QorIQ configuration project

Perform the following steps to create a QorIQ configuration project with the PBL tool:

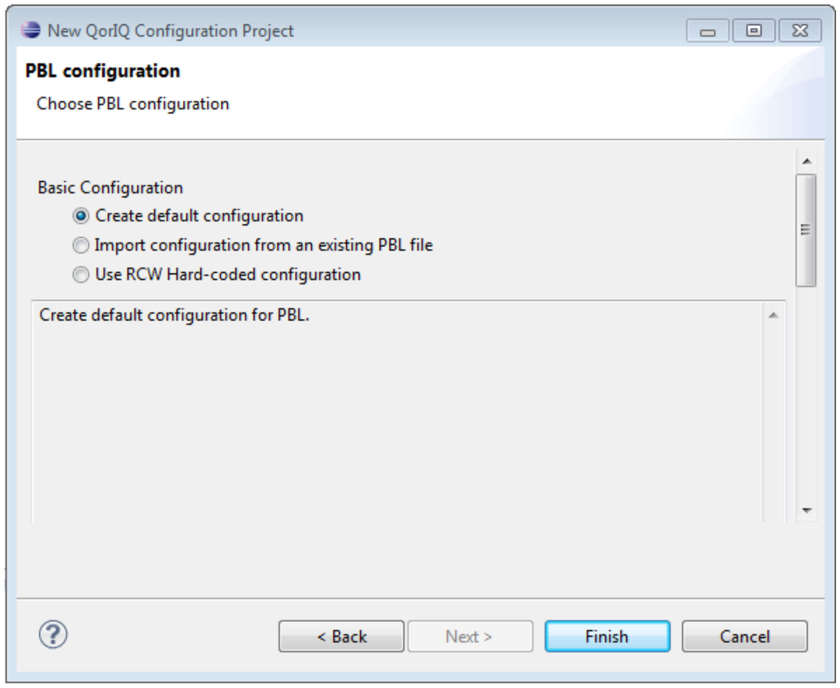
1. Open the QCVS Eclipse integrated development environment (IDE).
2. Choose **File > New > QorIQ Configuration Project** from the IDE menu bar. The **New QorIQ Configuration Project** wizard starts, displaying the **Create a QorIQ Configuration Project** page.
3. Specify the project name in the **Project name** text box, and click **Next**. The **Devices** page appears.
4. Choose a device and a device version, and click **Next**. The **Toolset selection** page appears.
5. Select the **PBL – Preboot Loader RCW configuration** checkbox, and click **Next**. The **PBL configuration** page appears, where you can choose an initial PBL configuration for your project using one of the following three options, specify other required settings, and complete project creation:

- Create default configuration
- Import configuration from an existing PBL file
- Use RCW Hard-coded configuration

3.1. Create default configuration

The default PBL configuration includes basic settings for RCW and no PBI commands. Use the **Create default configuration** option when neither you need to customize an existing RCW dump (see Import configuration from an existing PBL file) nor you have a hard-coded RCW configuration to work on (see Use RCW Hard-coded configuration).
The figure below shows the **PBL configuration** page with the **Create default configuration** option selected.

Figure 1. Creating a default configuration



PBL configuration

Choose PBL configuration

Basic Configuration

- [✓ Create default configuration]
- Import configuration from an existing PBL file
- Use RCW Hard-coded configuration

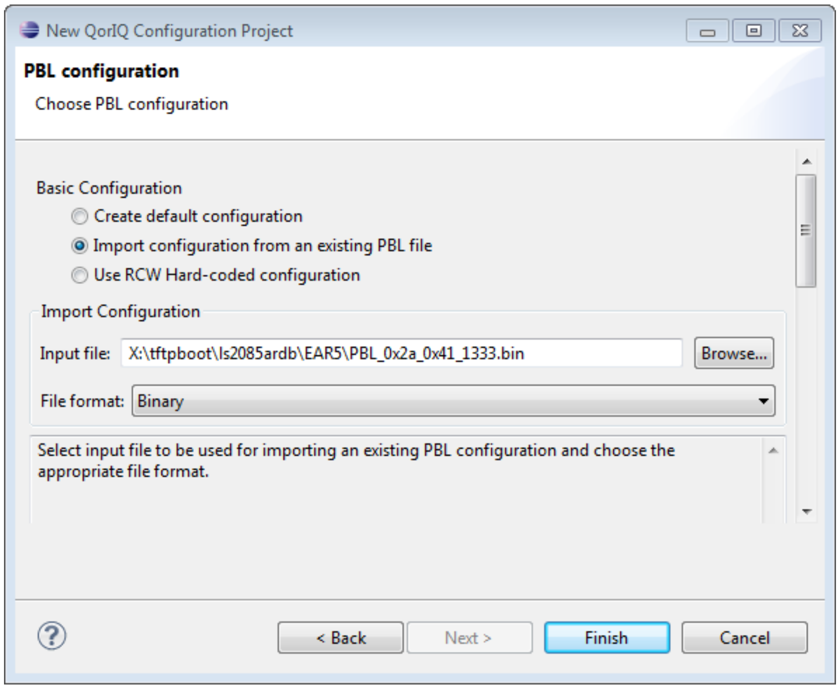
Create default configuration for PBL.

3.2. Import configuration from an existing PBL file

The **Import configuration from an existing PBL file** option allows you to import PBL from other projects/resources, such as SDK. This option is useful when you need to quickly investigate and/or customize an existing PBL.

The figure below shows the **PBL configuration** page with the **Import configuration from an existing PBL file** option selected.

Figure 2. Importing a configuration from an existing PBL file



Basic Configuration

- Create default configuration
- [✓ Import configuration from an existing PBL file]
- Use RCW Hard-coded configuration

Import Configuration

Input file: X:\tftpboot\ls2085ardb\EAR5\PBL_0x2a_0x41_1333.bin Browse...

File format: Binary

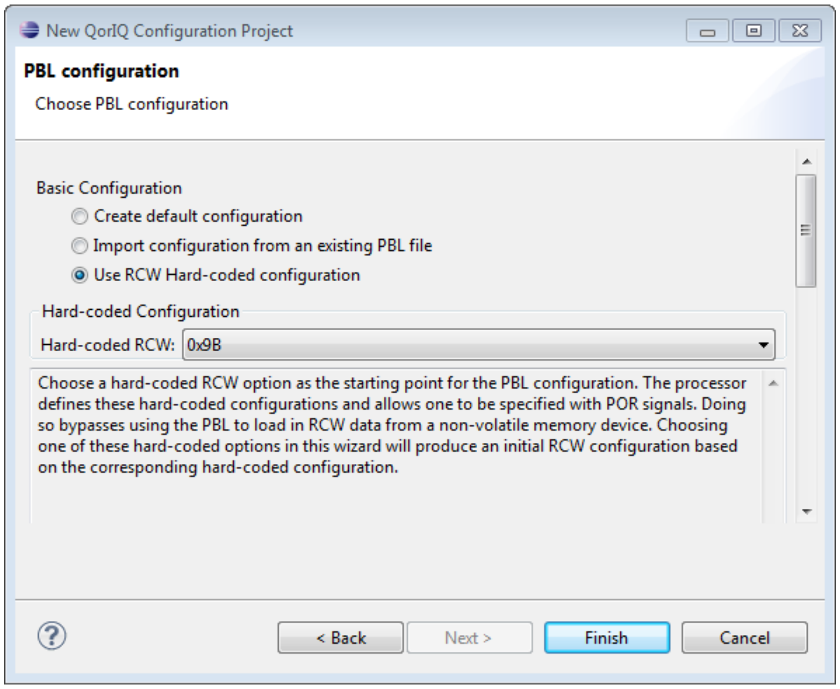
Select input file to be used for importing an existing PBL configuration and choose the appropriate file format.

3.3. Use RCW Hard-coded configuration

A hard-coded RCW configuration can be used as the starting point for the PBL configuration. If you are new to the QCVS PBL tool, then you should use the **Use RCW Hard-coded configuration** option to create a QorIQ configuration project with the PBL tool. Hard-coded RCWs have been tested with the QorIQ reference design boards (RDBs) and represent a good starting point to customize a PBL for a custom or reference design board.

The figure below shows the **PBL configuration** page with the **Use RCW Hard-coded configuration** option selected.

Figure 3. Using an RCW hard-coded configuration



PBL configuration

Choose PBL configuration

Basic Configuration

Create default configuration

Import configuration from an existing PBL file

[✓ Use RCW Hard-coded configuration]

Hard-coded Configuration

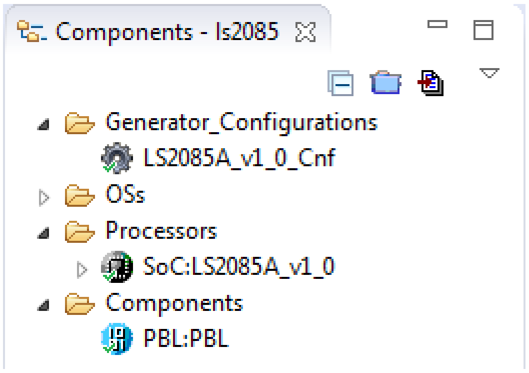
Hard-coded RCW: 0x9B

Choose a hard-coded RCW option as the starting point for the PBL configuration. The processor defines these hard-coded configurations and allows one to be specified with POR signals. Doing so bypasses using the PBL to load in RCW data from a non-volatile memory device. Choosing one of these hard-coded options in this wizard will produce an initial RCW configuration based on the corresponding hard-coded configuration.

4. Basic PBL operations

When you create a QorIQ configuration project with the PBL tool, a PBL component is created under the **Components** folder in the **Components** view, as shown in the figure below.

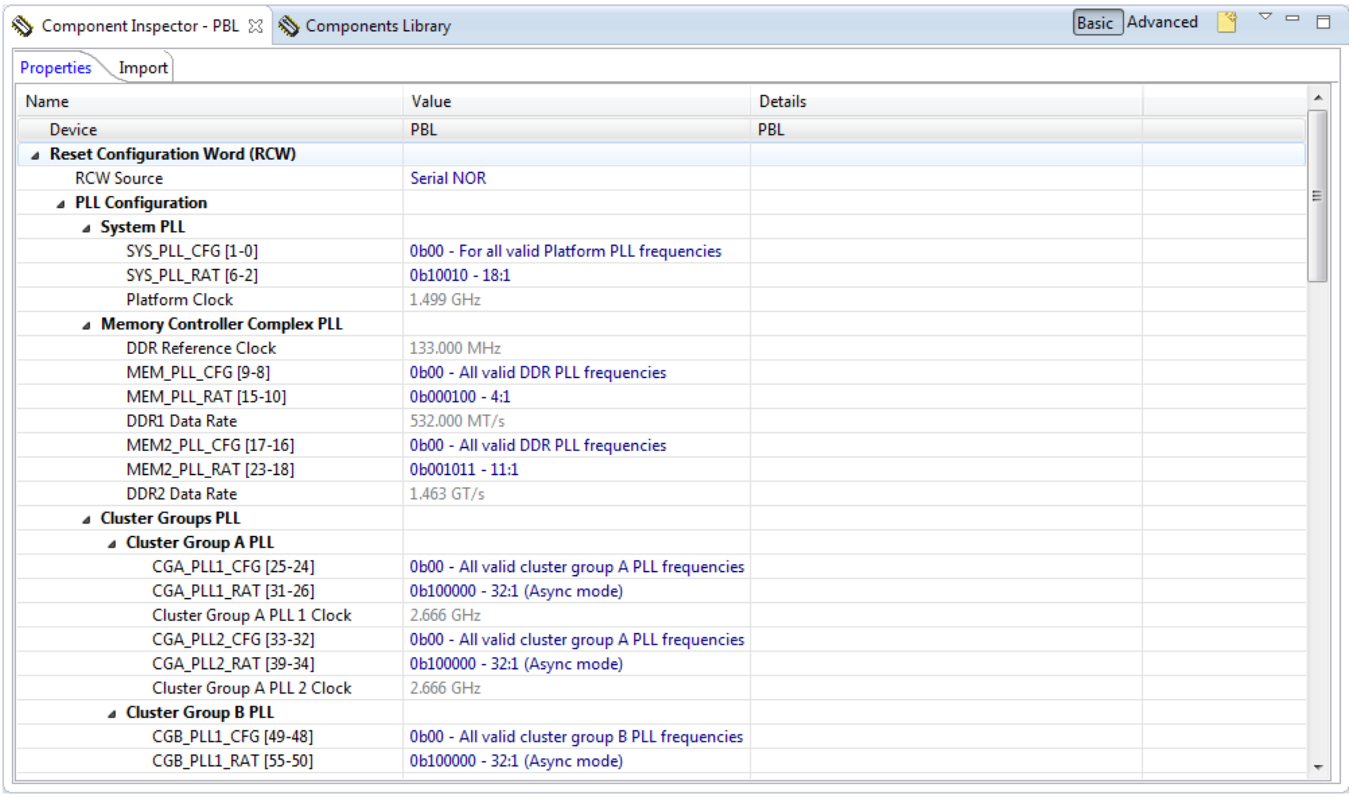
Figure 4. Components view



To view or edit the properties of the PBL component, select the PBL component in the **Components** view. The component properties are displayed on the **Properties** page of the **Component Inspector** view, as shown in the figure below.

NOTE If the **Component Inspector** view is not open already, then open it by right-clicking a component in the **Components** view and choosing **Inspector** from the shortcut menu.

Figure 5. Component Inspector view



Name	Value	Details
Device	PBL	PBL
Reset Configuration Word (RCW)		
RCW Source	Serial NOR	
PLL Configuration		
System PLL		
SYS_PLL_CFG [1-0]	0b00 - For all valid Platform PLL frequencies	
SYS_PLL_RAT [6-2]	0b10010 - 18:1	
Platform Clock	1.499 GHz	
Memory Controller Complex PLL		
DDR Reference Clock	133.000 MHz	
MEM_PLL_CFG [9-8]	0b00 - All valid DDR PLL frequencies	
MEM_PLL_RAT [15-10]	0b000100 - 4:1	
DDR1 Data Rate	532.000 MT/s	
MEM2_PLL_CFG [17-16]	0b00 - All valid DDR PLL frequencies	
MEM2_PLL_RAT [23-18]	0b001011 - 11:1	
DDR2 Data Rate	1.463 GT/s	
Cluster Groups PLL		
Cluster Group A PLL		
CGA_PLL1_CFG [25-24]	0b00 - All valid cluster group A PLL frequencies	
CGA_PLL1_RAT [31-26]	0b100000 - 32:1 (Async mode)	
Cluster Group A PLL 1 Clock	2.666 GHz	
CGA_PLL2_CFG [33-32]	0b00 - All valid cluster group A PLL frequencies	
CGA_PLL2_RAT [39-34]	0b100000 - 32:1 (Async mode)	
Cluster Group A PLL 2 Clock	2.666 GHz	
Cluster Group B PLL		
CGB_PLL1_CFG [49-48]	0b00 - All valid cluster group B PLL frequencies	
CGB_PLL1_RAT [55-50]	0b100000 - 32:1 (Async mode)	

As you can see in the figure above, the component properties are grouped under various groups and subgroups. All the settings and properties of the PBL component are sorted by the RCW position.

Following are some basic PBL operations you can perform in the **Component Inspector** view:

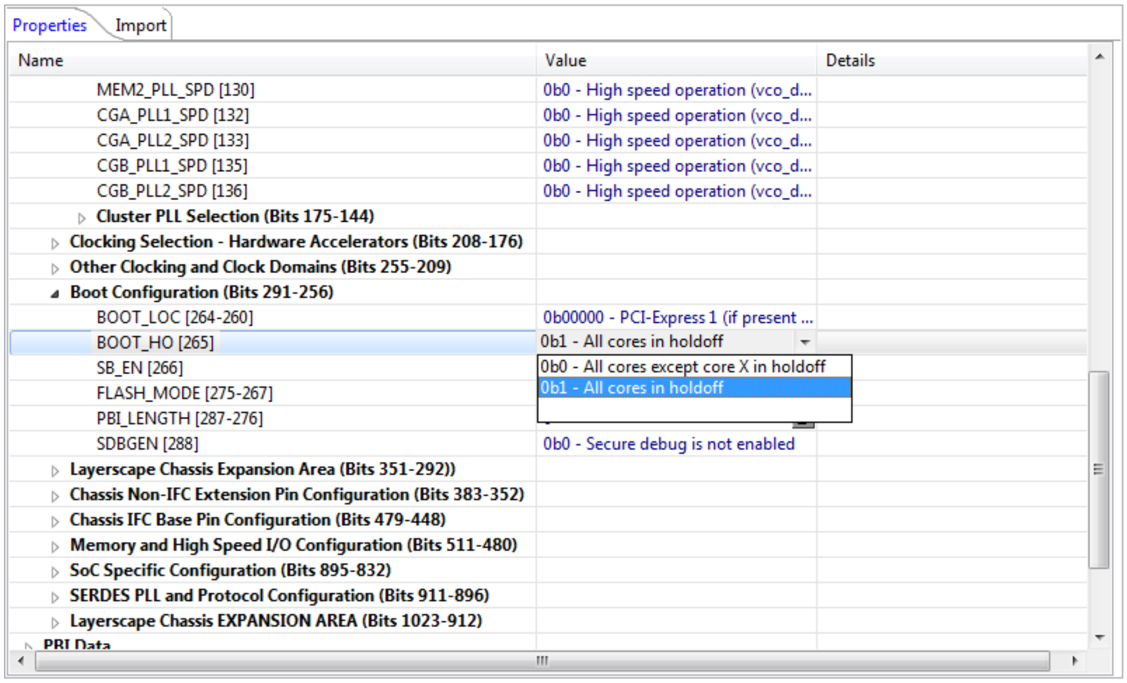
- Change RCW bit field values
- Add PBI commands to a PBL image
- Import a PBL configuration from a file
- Generate a PBL image
- Automatic PBL validation
- Synchronize PBL with other IP blocks
- View RCW status registers

4.1. Change RCW bit field values

For most of the RCW fields, you can change the value by typing in a new value or choosing another value from a menu. For other fields (for example, SerDes protocol options), a more advanced graphical user interface (GUI) is displayed to change the value.

The figure below shows an example of changing a field value by choosing another value from a menu.

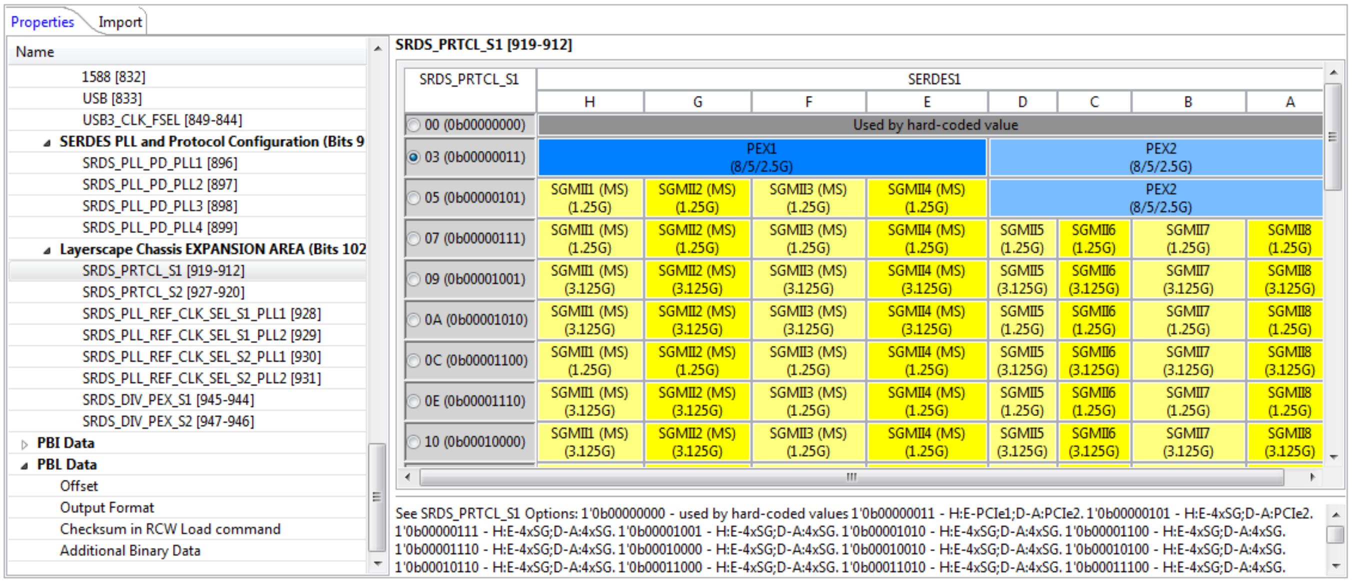
Figure 6. Changing a field value by choosing another value from a menu



Name	Value	Details
MEM2_PLL_SPD [130]	0b0 - High speed operation (vco_d...	
CGA_PLL1_SPD [132]	0b0 - High speed operation (vco_d...	
CGA_PLL2_SPD [133]	0b0 - High speed operation (vco_d...	
CGB_PLL1_SPD [135]	0b0 - High speed operation (vco_d...	
CGB_PLL2_SPD [136]	0b0 - High speed operation (vco_d...	
Cluster PLL Selection (Bits 175-144)		
Clocking Selection - Hardware Accelerators (Bits 208-176)		
Other Clocking and Clock Domains (Bits 255-209)		
Boot Configuration (Bits 291-256)		
BOOT_LOC [264-260]	0b00000 - PCI-Express 1 (if present ...	
BOOT_HO [265]	0b1 - All cores in holdoff	
SB_EN [266]	0b0 - All cores except core X in holdoff	
FLASH_MODE [275-267]	0b1 - All cores in holdoff	
PBI_LENGTH [287-276]		
SDBGEN [288]	0b0 - Secure debug is not enabled	
Layerscape Chassis Expansion Area (Bits 351-292))		
Chassis Non-IFC Extension Pin Configuration (Bits 383-352)		
Chassis IFC Base Pin Configuration (Bits 479-448)		
Memory and High Speed I/O Configuration (Bits 511-480)		
SoC Specific Configuration (Bits 895-832)		
SERDES PLL and Protocol Configuration (Bits 911-896)		
Layerscape Chassis EXPANSION AREA (Bits 1023-912)		
PBI Data		

The figure below shows an example of changing a field value by using an advanced GUI.

Figure 7. Changing a field value by using an advanced GUI



Properties | Import

Name

- 1588 [832]
- USB [833]
- USB3_CLK_FSEL [849-844]
- SERDES PLL and Protocol Configuration (Bits 9
  - SRDS_PLL_PD_PLL1 [896]
  - SRDS_PLL_PD_PLL2 [897]
  - SRDS_PLL_PD_PLL3 [898]
  - SRDS_PLL_PD_PLL4 [899]
- Layerscape Chassis EXPANSION AREA (Bits 102
  - SRDS_PRTCL_S1 [919-912]
  - SRDS_PRTCL_S2 [927-920]
  - SRDS_PLL_REF_CLK_SEL_S1_PLL1 [928]
  - SRDS_PLL_REF_CLK_SEL_S1_PLL2 [929]
  - SRDS_PLL_REF_CLK_SEL_S2_PLL1 [930]
  - SRDS_PLL_REF_CLK_SEL_S2_PLL2 [931]
  - SRDS_DIV_PEX_S1 [945-944]
  - SRDS_DIV_PEX_S2 [947-946]
- PBI Data
- PBL Data
  - Offset
  - Output Format
  - Checksum in RCW Load command
  - Additional Binary Data

SRDS_PRTCL_S1 [919-912]

SRDS_PRTCL_S1	SERDES1							
	H	G	F	E	D	C	B	A
○ 00 (0b00000000)	Used by hard-coded value							
● 03 (0b00000011)	PEX1 (8/5/2.5G)				PEX2 (8/5/2.5G)			
○ 05 (0b00000101)	SGMII1 (MS) (1.25G)	SGMII2 (MS) (1.25G)	SGMII3 (MS) (1.25G)	SGMII4 (MS) (1.25G)	PEX2 (8/5/2.5G)			
○ 07 (0b00000111)	SGMII1 (MS) (1.25G)	SGMII2 (MS) (1.25G)	SGMII3 (MS) (1.25G)	SGMII4 (MS) (1.25G)	SGMII5 (1.25G)	SGMII6 (1.25G)	SGMII7 (1.25G)	SGMII8 (1.25G)
○ 09 (0b00001001)	SGMII1 (MS) (3.125G)	SGMII2 (MS) (3.125G)	SGMII3 (MS) (3.125G)	SGMII4 (MS) (3.125G)	SGMII5 (3.125G)	SGMII6 (3.125G)	SGMII7 (3.125G)	SGMII8 (3.125G)
○ 0A (0b00001010)	SGMII1 (MS) (3.125G)	SGMII2 (MS) (3.125G)	SGMII3 (MS) (3.125G)	SGMII4 (MS) (3.125G)	SGMII5 (1.25G)	SGMII6 (1.25G)	SGMII7 (1.25G)	SGMII8 (1.25G)
○ 0C (0b00001100)	SGMII1 (MS) (1.25G)	SGMII2 (MS) (1.25G)	SGMII3 (MS) (1.25G)	SGMII4 (MS) (1.25G)	SGMII5 (3.125G)	SGMII6 (3.125G)	SGMII7 (3.125G)	SGMII8 (3.125G)
○ 0E (0b00001110)	SGMII1 (MS) (3.125G)	SGMII2 (MS) (3.125G)	SGMII3 (MS) (1.25G)	SGMII4 (MS) (1.25G)	SGMII5 (1.25G)	SGMII6 (1.25G)	SGMII7 (1.25G)	SGMII8 (1.25G)
○ 10 (0b00010000)	SGMII1 (MS) (3.125G)	SGMII2 (MS) (3.125G)	SGMII3 (MS) (1.25G)	SGMII4 (MS) (1.25G)	SGMII5 (3.125G)	SGMII6 (3.125G)	SGMII7 (3.125G)	SGMII8 (3.125G)

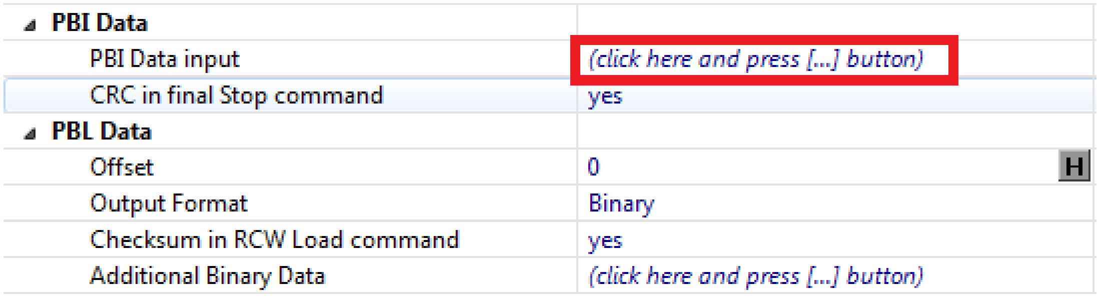
See SRDS_PRTCL_S1 Options: 1'0b00000000 - used by hard-coded values 1'0b00000011 - H:E-PCIe1;D-A:PCIe2. 1'0b00000101 - H:E-4xSG;D-A:PCIe2. 1'0b00000111 - H:E-4xSG;D-A:4xSG. 1'0b00001001 - H:E-4xSG;D-A:4xSG. 1'0b00001010 - H:E-4xSG;D-A:4xSG. 1'0b00001100 - H:E-4xSG;D-A:4xSG. 1'0b00001110 - H:E-4xSG;D-A:4xSG. 1'0b00010000 - H:E-4xSG;D-A:4xSG. 1'0b00010010 - H:E-4xSG;D-A:4xSG. 1'0b00010100 - H:E-4xSG;D-A:4xSG. 1'0b00010110 - H:E-4xSG;D-A:4xSG. 1'0b00011000 - H:E-4xSG;D-A:4xSG. 1'0b00011010 - H:E-4xSG;D-A:4xSG. 1'0b00011100 - H:E-4xSG;D-A:4xSG.

4.2. Add PBI commands to a PBL image

You can add the PBI commands to a PBL image by using the **PBI Data input** property under the **PBI Data** group on the **Properties** page of the **Component Inspector** view. Perform the following steps to add the PBI commands:

1. Select the **PBI Data input** property in the **Name** column and click the ellipsis (…) button in the **Value** column, as shown in the figure below.

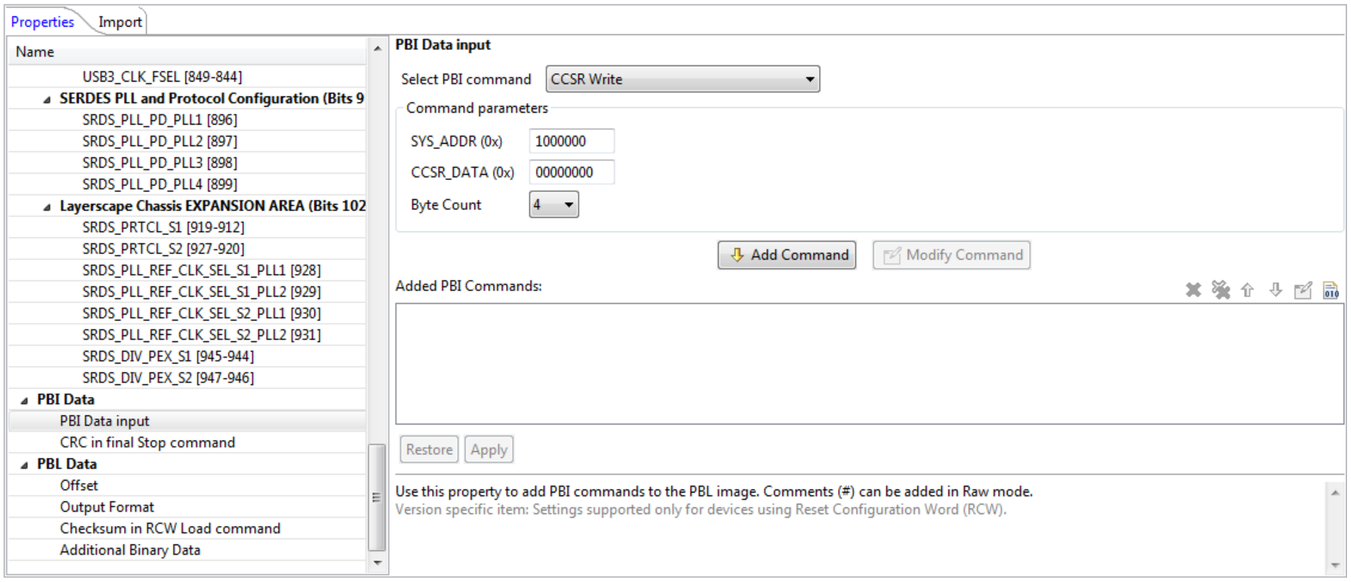
Figure 8. Setting PBI Data input property



PBI Data	
PBI Data input	(click here and press [...] button)
CRC in final Stop command	yes
PBL Data	
Offset	0
Output Format	Binary
Checksum in RCW Load command	yes
Additional Binary Data	(click here and press [...] button)

The **PBI Data input** editor opens, as shown in the figure below.

Figure 9. PBI Data input editor

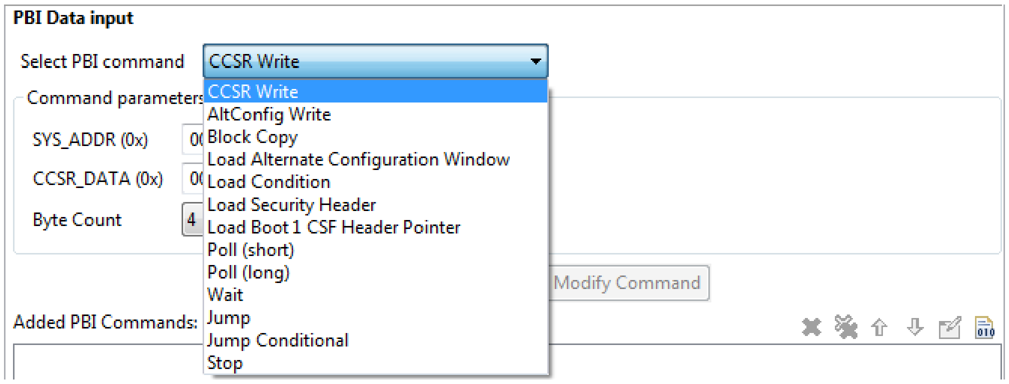


Select PBI command: CCSR Write
SYS_ADDR (0x): 1000000
CCSR_DATA (0x): 00000000
Byte Count: 4

Use this property to add PBI commands to the PBL image. Comments (#) can be added in Raw mode.
Version specific item: Settings supported only for devices using Reset Configuration Word (RCW).

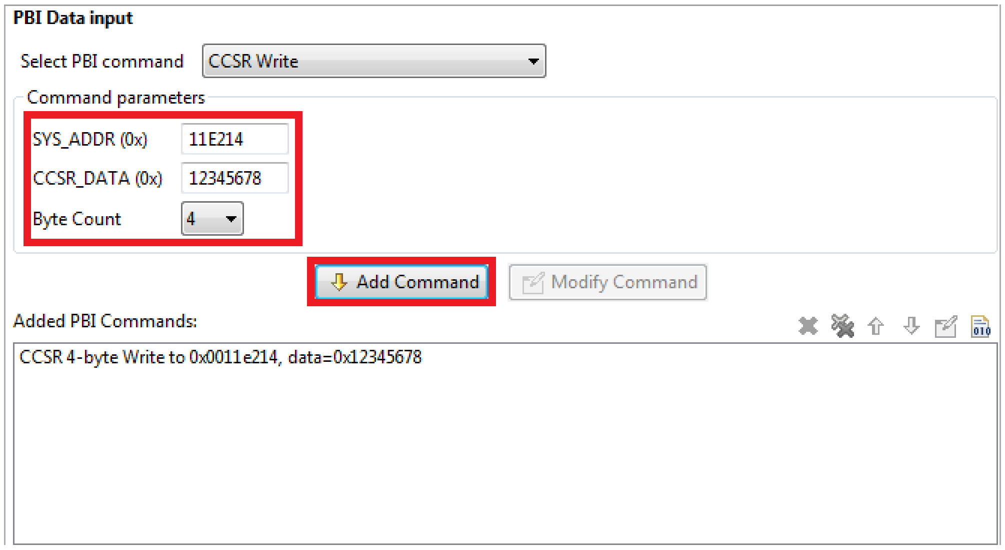
2. Choose the appropriate PBI command from the **Select PBI command** menu, as shown in the figure below.

Figure 10. Choosing a PBI command



3. Edit command parameters in the **Command parameters** group and click the **Add Command** button. The PBI command is added in the **Added PBI Commands** pane, as shown in the figure below.

Figure 11. Editing a PBI command



PBI Data input

Select PBI command: CCSR Write

Command parameters

SYS_ADDR (0x): 11E214

CCSR_DATA (0x): 12345678

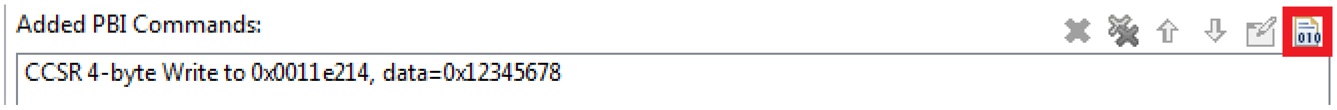
Byte Count: 4

Added PBI Commands:

CCSR 4-byte Write to 0x0011e214, data=0x12345678

You can view the PBI commands in two modes: disassembly view and raw data view. To switch between the two modes, click the rightmost button on the toolbar of the **Added PBI Commands** pane, as shown in the following figures.

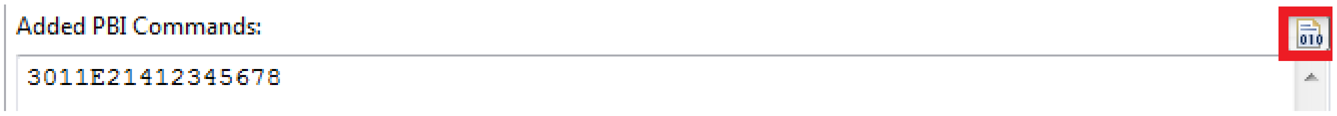
Figure 12. Disassembly view



Added PBI Commands:

CCSR 4-byte Write to 0x0011e214, data=0x12345678

Figure 13. Raw data view

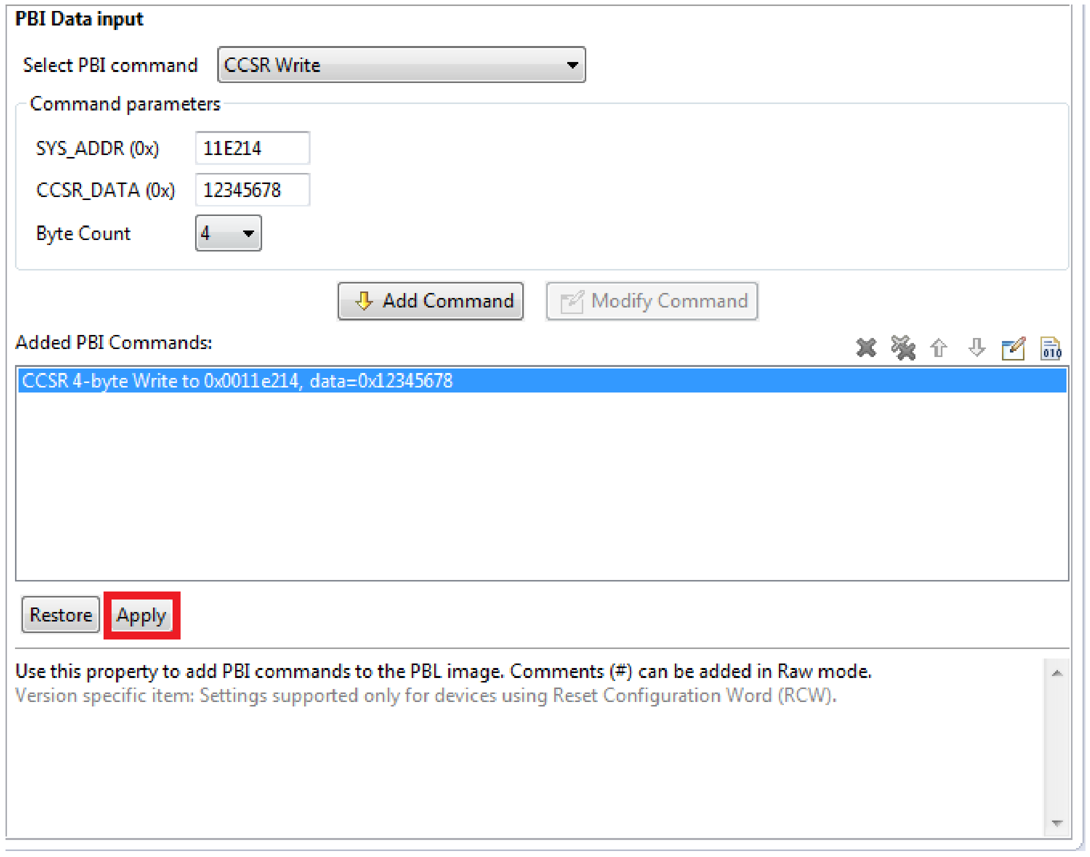


Added PBI Commands:

```
3011E21412345678
```

4. Click the **Apply** button to add the PBI commands to the PBL image.

Figure 14. Adding PBI commands to a PBL image



PBI Data input

Select PBI command: CCSR Write

Command parameters

SYS_ADDR (0x): 11E214

CCSR_DATA (0x): 12345678

Byte Count: 4

Added PBI Commands:

CCSR 4-byte Write to 0x0011e214, data=0x12345678

Use this property to add PBI commands to the PBL image. Comments (#) can be added in Raw mode.
Version specific item: Settings supported only for devices using Reset Configuration Word (RCW).

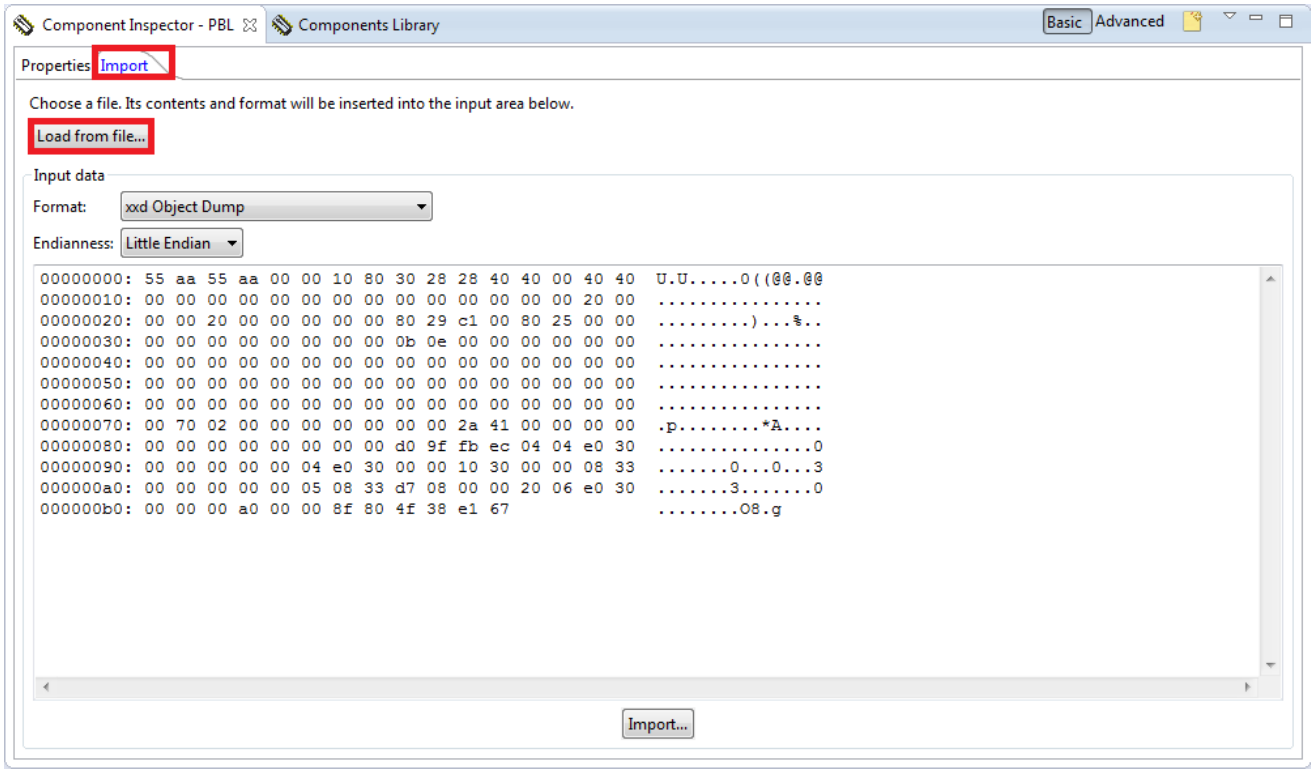
NOTE Depending on its parameters, a PBI command may need to be split into several PBI commands. The PBL tool does it automatically and informs the user about the split.

4.3. Import a PBL configuration from a file

To import a PBL configuration for an existing PBL component, perform these steps:

1. Click the **Import** tab in the **Component Inspector** view. The **Import** page appears.
2. Choose a PBL file by clicking the **Load from file** button. The file format of the chosen file is detected automatically and its content is displayed in the Rich Text Format in an editor available in the **Input data** group, as shown in the figure below.

Figure 15. Loading a PBL file



Choose a file. Its contents and format will be inserted into the input area below.

Input data

Format: xxd Object Dump

Endianness: Little Endian

```
00000000: 55 aa 55 aa 00 00 10 80 30 28 28 40 40 00 40 40  U.U.....0((@@.@@
00000010: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 20 00  ................
00000020: 00 00 20 00 00 00 00 00 80 29 c1 00 80 25 00 00  .........)...%..
00000030: 00 00 00 00 00 00 00 00 0b 0e 00 00 00 00 00 00  ................
00000040: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00000050: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00000060: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00000070: 00 70 02 00 00 00 00 00 00 00 2a 41 00 00 00 00  .p........*A....
00000080: 00 00 00 00 00 00 00 00 d0 9f fb ec 04 04 e0 30  ...............0
00000090: 00 00 00 00 00 04 e0 30 00 00 10 30 00 00 08 33  .......0...0...3
000000a0: 00 00 00 00 00 05 08 33 d7 08 00 00 20 06 e0 30  .......3.......0
000000b0: 00 00 00 a0 00 00 8f 80 4f 38 e1 67              ........O8.g
```

3. Edit the PBL file in the editor, as needed.
4. Click the **Import** button to import the new PBL configuration.
5. Switch to the **Properties** tab to view or edit the imported PBL configuration.

Using the steps provided in this section, you can import files having the following file formats:

- XXD Object Dump
- S-record
- U-Boot Flash Dump
- Hex String
- U-Boot CCRS Startup Dump (RCW only)
- CW JTAG Config (RCW only)
- Hex String (RCW only)
- Test Table (RCW only)

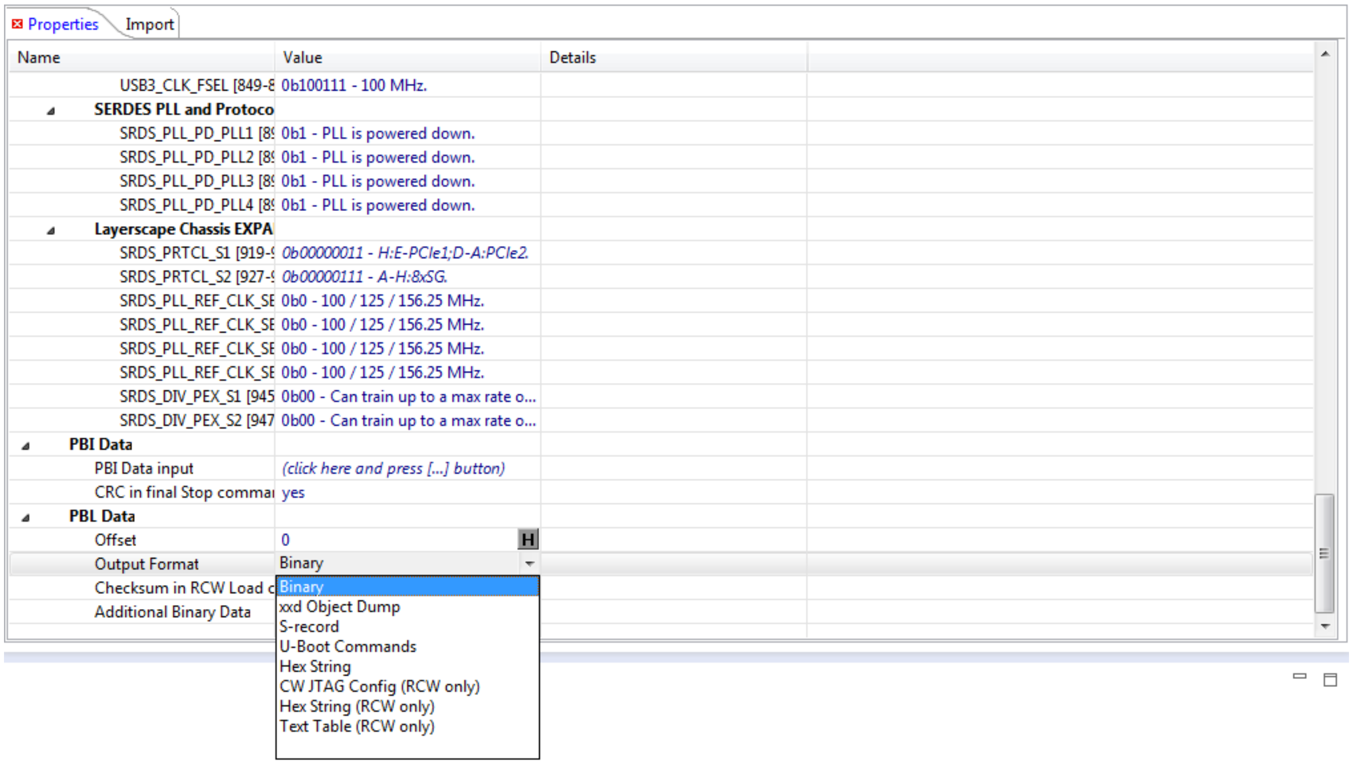
NOTE By default, the import operation tries to convert the memory dump into the XXD Object Dump format and displays it in the Rich Text Format.

4.4. Generate a PBL image

To generate a PBL image from the PBL configuration, perform these steps:

1. Select the **Output Format** property under the **PBL Data** group in the **Name** column on the **Properties** page of the **Component Inspector** view.
2. Click the corresponding cell in the **Value** column and choose a file format for the PBL image.

Figure 16. Choosing a file format for PBL file

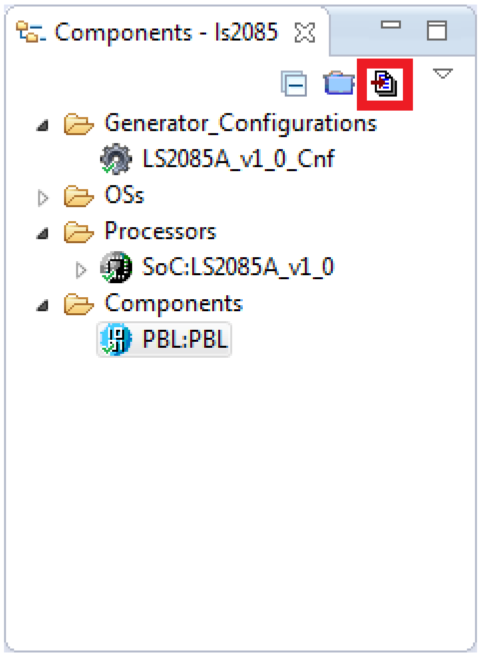


Properties | Import

Name	Value	Details
USB3_CLK_FSEL [849-8	0b100111 - 100 MHz.	
SERDES PLL and Protoco		
SRDS_PLL_PD_PLL1 [8!	0b1 - PLL is powered down.	
SRDS_PLL_PD_PLL2 [8!	0b1 - PLL is powered down.	
SRDS_PLL_PD_PLL3 [8!	0b1 - PLL is powered down.	
SRDS_PLL_PD_PLL4 [8!	0b1 - PLL is powered down.	
Layerscape Chassis EXPA		
SRDS_PRTCL_S1 [919-!	*0b00000011 - H:E-PCIe1;D-A:PCIe2.*	
SRDS_PRTCL_S2 [927-!	*0b00000111 - A-H:8xSG.*	
SRDS_PLL_REF_CLK_SE	0b0 - 100 / 125 / 156.25 MHz.	
SRDS_PLL_REF_CLK_SE	0b0 - 100 / 125 / 156.25 MHz.	
SRDS_PLL_REF_CLK_SE	0b0 - 100 / 125 / 156.25 MHz.	
SRDS_PLL_REF_CLK_SE	0b0 - 100 / 125 / 156.25 MHz.	
SRDS_DIV_PEX_S1 [945	0b00 - Can train up to a max rate o...	
SRDS_DIV_PEX_S2 [947	0b00 - Can train up to a max rate o...	
PBI Data		
PBI Data input	*(click here and press [...] button)*	
CRC in final Stop commai	yes	
PBL Data		
Offset	0	
Output Format	Binary	
Checksum in RCW Load c		
Additional Binary Data		

3. Click the **Generate Processor Expert Code** icon in the **Components** view to generate the PBL image.

Figure 17. Generating PBL image



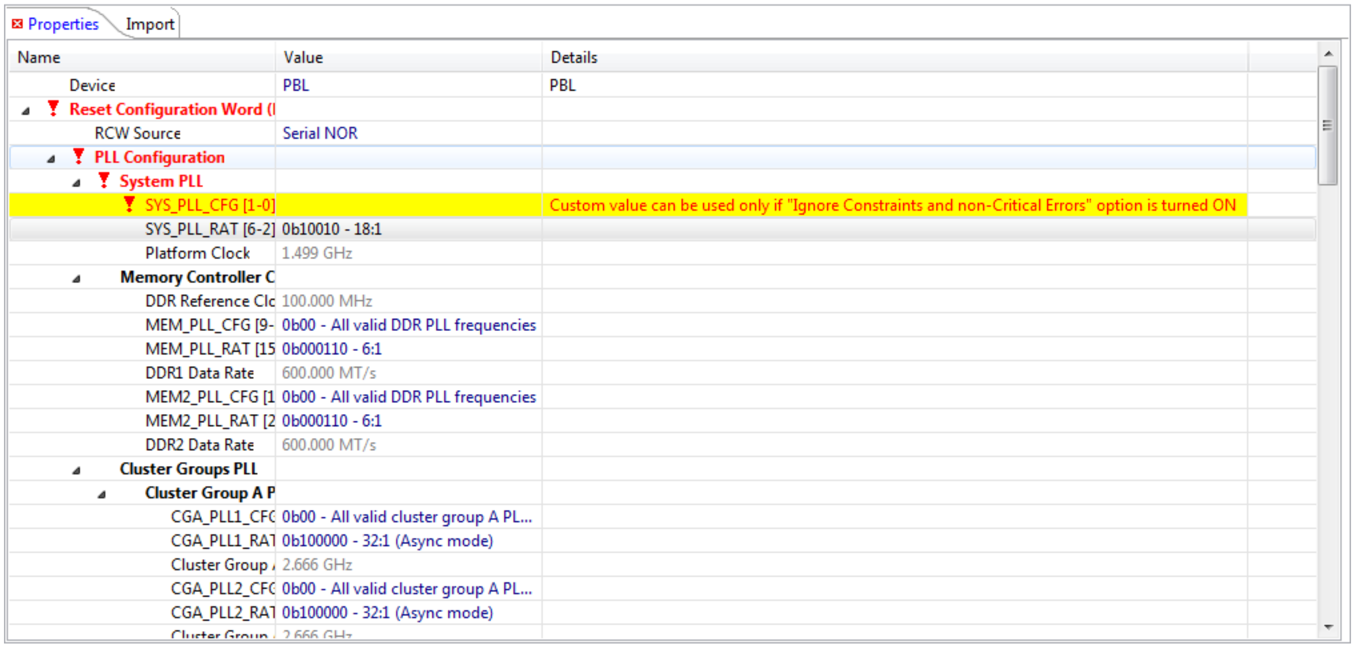
Using the steps provided in this section, you can generate PBL images with the following file formats:

- Binary
- XXD Object Dump
- S-record
- U-Boot Commands
- Hex String
- CW JTAG Config (RCW only)
- Hex String (RCW only)
- Text Table (RCW only)

4.5. Automatic PBL validation

Each time you change the PBL configuration, the PBL tool performs a check against known constraints and issues. If the configuration is found invalid, then the error or warning messages are displayed for the problematic RCW fields on the **Properties** page in the **Component Inspector** view, as shown in the figure below.

Figure 18. PBL validation

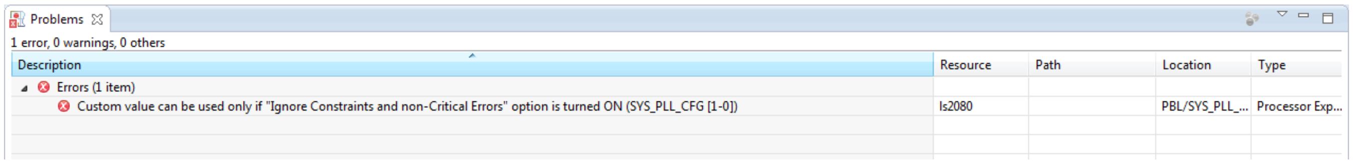


Properties | Import

Name	Value	Details
Device	PBL	PBL
Reset Configuration Word (I		
RCW Source	Serial NOR	
PLL Configuration		
System PLL		
SYS_PLL_CFG [1-0]		Custom value can be used only if "Ignore Constraints and non-Critical Errors" option is turned ON
SYS_PLL_RAT [6-2]	0b10010 - 18:1	
Platform Clock	1.499 GHz	
Memory Controller C		
DDR Reference Clc	100.000 MHz	
MEM_PLL_CFG [9-	0b00 - All valid DDR PLL frequencies	
MEM_PLL_RAT [15	0b000110 - 6:1	
DDR1 Data Rate	600.000 MT/s	
MEM2_PLL_CFG [1	0b00 - All valid DDR PLL frequencies	
MEM2_PLL_RAT [2	0b000110 - 6:1	
DDR2 Data Rate	600.000 MT/s	
Cluster Groups PLL		
Cluster Group A P		
CGA_PLL1_CFC	0b00 - All valid cluster group A PL...	
CGA_PLL1_RAT	0b100000 - 32:1 (Async mode)	
Cluster Group /	2.666 GHz	
CGA_PLL2_CFC	0b00 - All valid cluster group A PL...	
CGA_PLL2_RAT	0b100000 - 32:1 (Async mode)	
Cluster Group	2.666 GHz	

The error or warning messages are also displayed with additional details in the **Problems** view, as shown in the figure below.

Figure 19. Problems view



Problems

1 error, 0 warnings, 0 others

Description	Resource	Path	Location	Type
Errors (1 item)				
Custom value can be used only if "Ignore Constraints and non-Critical Errors" option is turned ON (SYS_PLL_CFG [1-0])	ls2080		PBL/SYS_PLL_...	Processor Exp...

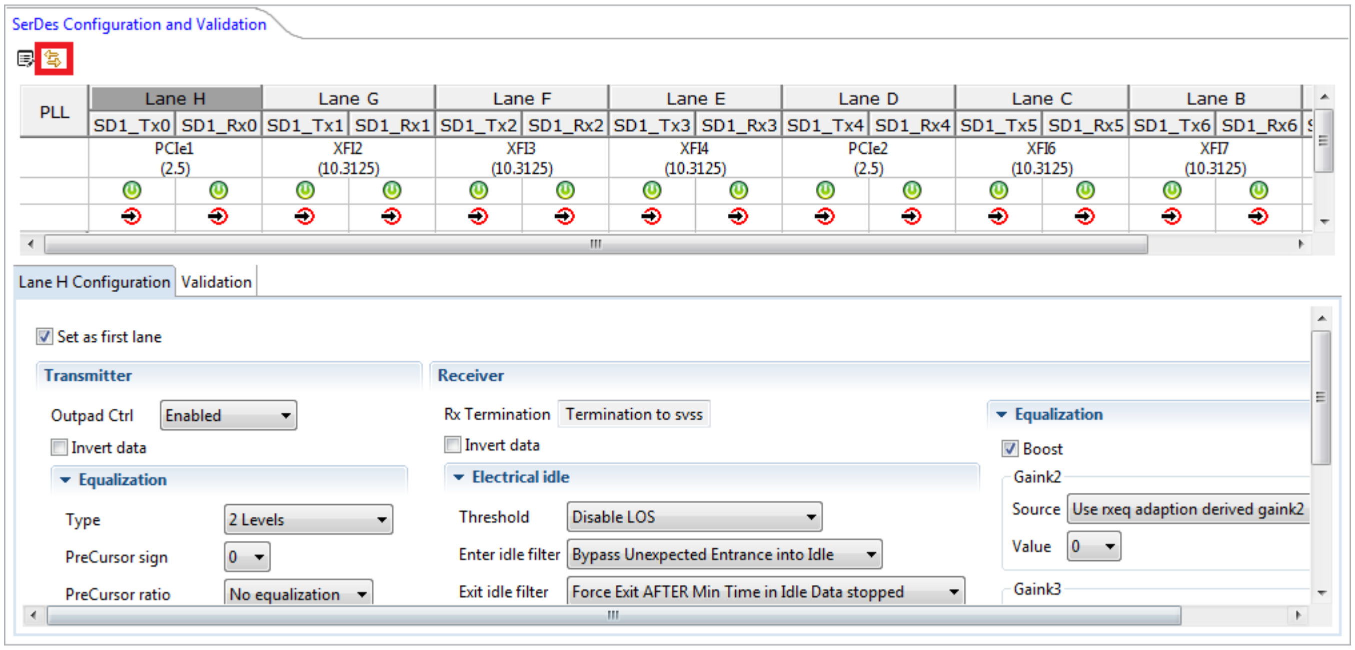
4.6. Synchronize PBL with other IP blocks

You can synchronize the PBL component with other intellectual property (IP) blocks, such as SerDes or DDR, if the corresponding component is available in the current project.

To synchronize the PBL component with a SerDes component, perform these steps:

1. Ensure that a SerDes block component is available in the current project.
2. Double-click a SerDes component grouped under the SerDes block component in the **Components** view. The properties of the SerDes component are displayed on the **SerDes Configuration and Validation** page in the **Component Inspector** view.
3. Click the **Apply the configuration to PBL component** button at the top-left corner (second button) of the **SerDes Configuration and Validation** page to synchronize the PBL component with the SerDes component, as shown in the figure below.

Figure 20. Synchronizing PBL and SerDes components



PLL	Lane H		Lane G		Lane F		Lane E		Lane D		Lane C		Lane B	
	SD1_Tx0	SD1_Rx0	SD1_Tx1	SD1_Rx1	SD1_Tx2	SD1_Rx2	SD1_Tx3	SD1_Rx3	SD1_Tx4	SD1_Rx4	SD1_Tx5	SD1_Rx5	SD1_Tx6	SD1_Rx6
	PCIe1 (2.5)		XFI2 (10.3125)		XFI3 (10.3125)		XFI4 (10.3125)		PCIe2 (2.5)		XFI6 (10.3125)		XFI7 (10.3125)	

☑ Set as first lane

Transmitter

Outpad Ctrl: Enabled
☐ Invert data

Equalization
Type: 2 Levels
PreCursor sign: 0
PreCursor ratio: No equalization

Receiver

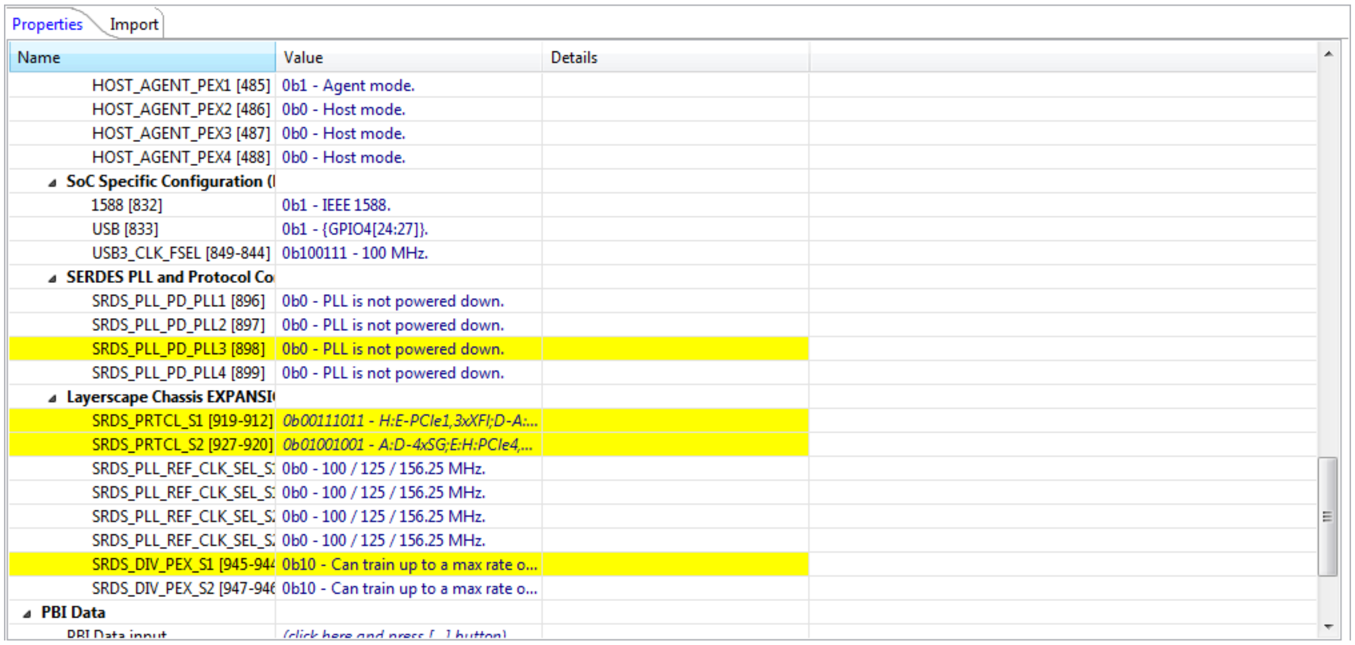
Rx Termination: Termination to svss
☐ Invert data

Electrical idle
Threshold: Disable LOS
Enter idle filter: Bypass Unexpected Entrance into Idle
Exit idle filter: Force Exit AFTER Min Time in Idle Data stopped

Equalization
☑ Boost
Gaink2
Source: Use rxeq adaption derived gaink2
Value: 0
Gaink3

4. Select the PBL component in the **Components** view and verify the SerDes fields on the **Properties** page of the **Component Inspector** view, as shown in the figure below.

Figure 21. Verifying SerDes fields of PBL component


Properties | Import

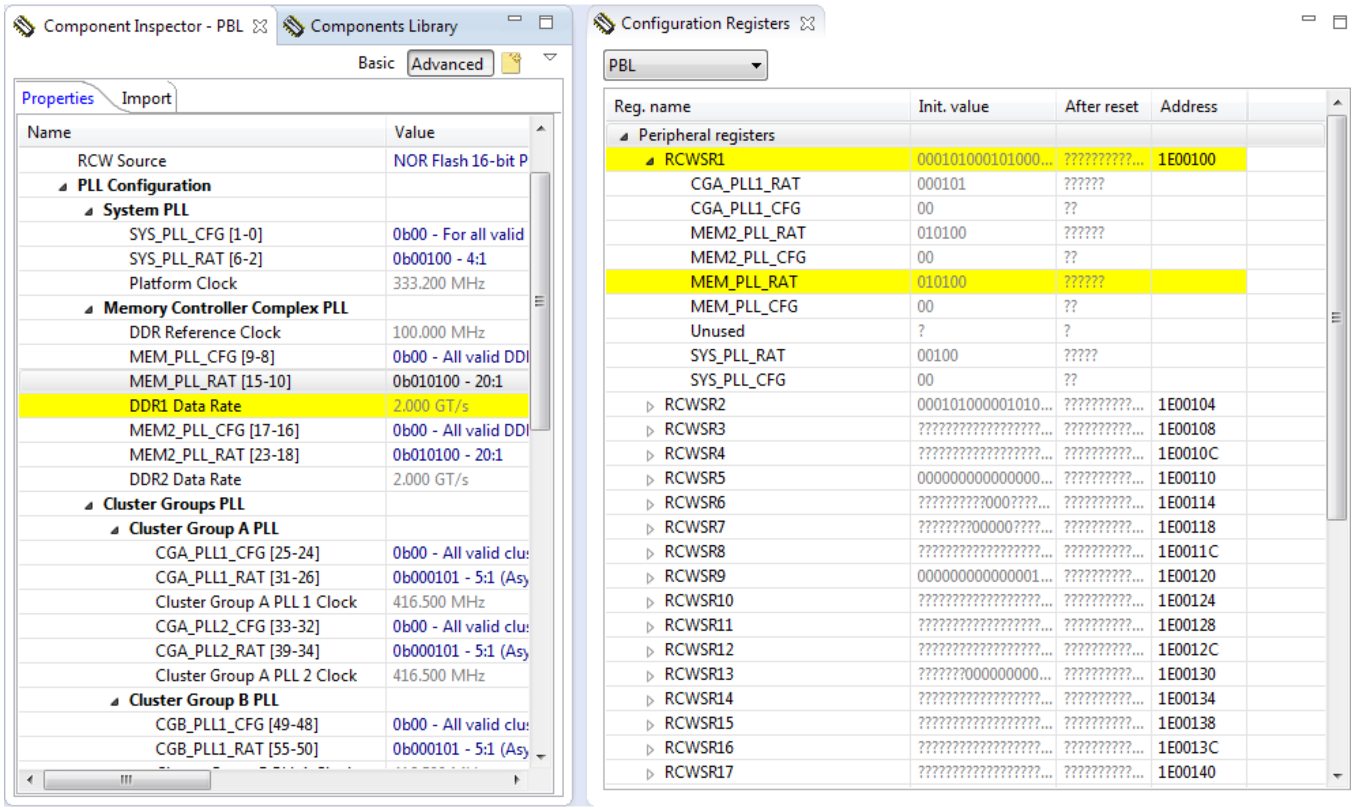
Name	Value	Details
HOST_AGENT_PEX1 [485]	0b1 - Agent mode.	
HOST_AGENT_PEX2 [486]	0b0 - Host mode.	
HOST_AGENT_PEX3 [487]	0b0 - Host mode.	
HOST_AGENT_PEX4 [488]	0b0 - Host mode.	
SoC Specific Configuration (I		
1588 [832]	0b1 - IEEE 1588.	
USB [833]	0b1 - {GPIO4[24:27]}.	
USB3_CLK_FSEL [849-844]	0b100111 - 100 MHz.	
SERDES PLL and Protocol Co		
SRDS_PLL_PD_PLL1 [896]	0b0 - PLL is not powered down.	
SRDS_PLL_PD_PLL2 [897]	0b0 - PLL is not powered down.	
SRDS_PLL_PD_PLL3 [898]	0b0 - PLL is not powered down.	
SRDS_PLL_PD_PLL4 [899]	0b0 - PLL is not powered down.	
Layerscape Chassis EXPANSI		
SRDS_PRTCL_S1 [919-912]	*0b00111011 - H:E-PCIe1,3xXFI;D-A:...*	
SRDS_PRTCL_S2 [927-920]	*0b01001001 - A:D-4xSG;E:H:PCIe4,...*	
SRDS_PLL_REF_CLK_SEL_S:	0b0 - 100 / 125 / 156.25 MHz.	
SRDS_PLL_REF_CLK_SEL_S:	0b0 - 100 / 125 / 156.25 MHz.	
SRDS_PLL_REF_CLK_SEL_S:	0b0 - 100 / 125 / 156.25 MHz.	
SRDS_PLL_REF_CLK_SEL_S:	0b0 - 100 / 125 / 156.25 MHz.	
SRDS_DIV_PEX_S1 [945-94	0b10 - Can train up to a max rate o...	
SRDS_DIV_PEX_S2 [947-94	0b10 - Can train up to a max rate o...	
PBI Data		
PBI Data input	*(click here and press [...] button)*	

4.7. View RCW status registers

To have an overview of the RCW status registers, perform these steps:

1. Choose **Window > Show View > Other** from the IDE menu bar. The **Show View** dialog appears.
2. Choose **Processor Expert > Configuration Registers** and click **OK**. The **Show View** dialog closes and the **Configuration Registers** view appears, displaying the details of the RCW status registers, as shown in the figure below.

Figure 22. Viewing RCW status registers



Name	Value
RCW Source	NOR Flash 16-bit P
PLL Configuration	
System PLL	
SYS_PLL_CFG [1-0]	0b00 - For all valid
SYS_PLL_RAT [6-2]	0b00100 - 4:1
Platform Clock	333.200 MHz
Memory Controller Complex PLL	
DDR Reference Clock	100.000 MHz
MEM_PLL_CFG [9-8]	0b00 - All valid DDI
MEM_PLL_RAT [15-10]	0b010100 - 20:1
DDR1 Data Rate	2.000 GT/s
MEM2_PLL_CFG [17-16]	0b00 - All valid DDI
MEM2_PLL_RAT [23-18]	0b010100 - 20:1
DDR2 Data Rate	2.000 GT/s
Cluster Groups PLL	
Cluster Group A PLL	
CGA_PLL1_CFG [25-24]	0b00 - All valid clu:
CGA_PLL1_RAT [31-26]	0b000101 - 5:1 (Asy
Cluster Group A PLL 1 Clock	416.500 MHz
CGA_PLL2_CFG [33-32]	0b00 - All valid clu:
CGA_PLL2_RAT [39-34]	0b000101 - 5:1 (Asy
Cluster Group A PLL 2 Clock	416.500 MHz
Cluster Group B PLL	
CGB_PLL1_CFG [49-48]	0b00 - All valid clu:
CGB_PLL1_RAT [55-50]	0b000101 - 5:1 (Asy

Reg. name	Init. value	After reset	Address
Peripheral registers			
RCWSR1	000101000101000...	?????????...	1E00100
CGA_PLL1_RAT	000101	??????	
CGA_PLL1_CFG	00	??	
MEM2_PLL_RAT	010100	??????	
MEM2_PLL_CFG	00	??	
MEM_PLL_RAT	010100	??????	
MEM_PLL_CFG	00	??	
Unused	?	?	
SYS_PLL_RAT	00100	?????	
SYS_PLL_CFG	00	??	
RCWSR2	000101000001010...	?????????...	1E00104
RCWSR3	?????????????????...	?????????...	1E00108
RCWSR4	?????????????????...	?????????...	1E0010C
RCWSR5	000000000000000...	?????????...	1E00110
RCWSR6	?????????000????...	?????????...	1E00114
RCWSR7	???????00000????...	?????????...	1E00118
RCWSR8	?????????????????...	?????????...	1E0011C
RCWSR9	000000000000001...	?????????...	1E00120
RCWSR10	?????????????????...	?????????...	1E00124
RCWSR11	?????????????????...	?????????...	1E00128
RCWSR12	?????????????????...	?????????...	1E0012C
RCWSR13	??????000000000...	?????????...	1E00130
RCWSR14	?????????????????...	?????????...	1E00134
RCWSR15	?????????????????...	?????????...	1E00138
RCWSR16	?????????????????...	?????????...	1E0013C
RCWSR17	?????????????????...	?????????...	1E00140

The **Configuration Registers** view reflects any changes made in the PBL configuration. Note that the don’t care bits are displayed using the “?” character.

5. Advanced PBL operations

This section is divided into the following subsections:

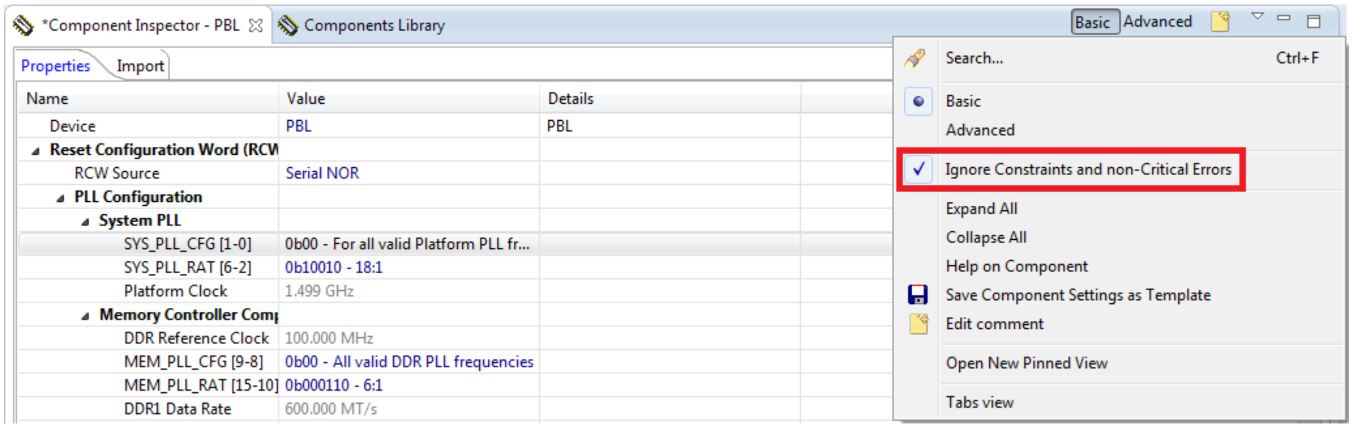
- Force RCW bit fields
- Add additional payload to a PBL image
- Errata support
- Endianness aspects

5.1. Force RCW bit fields

You can make an RCW bit field have a value that is not supported in the PBL tool, by default. Follow these steps to force an RCW bit field to have unsupported value:

1. Click the **View Menu** button (down arrow button) on the toolbar of the **Component Inspector** view and choose **Ignore Constraints and non-Critical Errors**, as shown in the figure below.

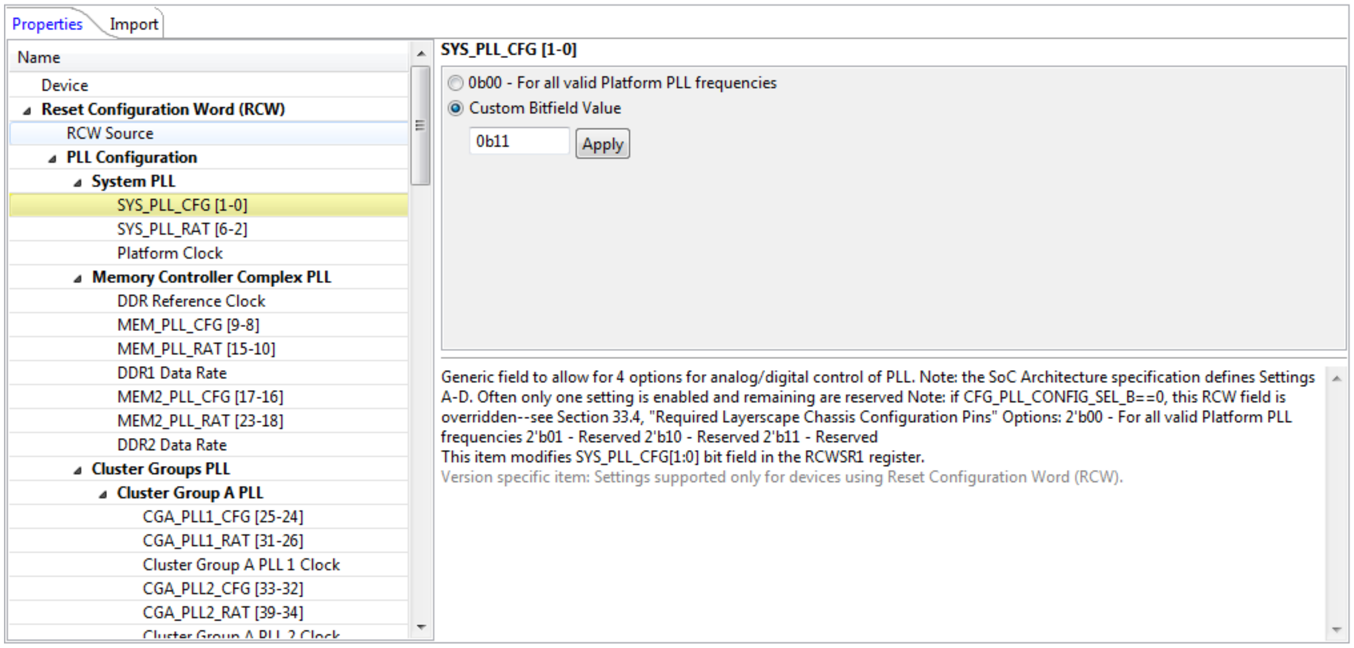
Figure 23. Ignoring constraints and non-critical errors



Name	Value	Details
Device	PBL	PBL
Reset Configuration Word (RCW		
RCW Source	Serial NOR	
PLL Configuration		
System PLL		
SYS_PLL_CFG [1-0]	0b00 - For all valid Platform PLL fr...	
SYS_PLL_RAT [6-2]	0b10010 - 18:1	
Platform Clock	1.499 GHz	
Memory Controller Comp		
DDR Reference Clock	100.000 MHz	
MEM_PLL_CFG [9-8]	0b00 - All valid DDR PLL frequencies	
MEM_PLL_RAT [15-10]	0b000110 - 6:1	
DDR1 Data Rate	600.000 MT/s	

2. Select an RCW bit field on the **Properties** page.
3. Select the **Custom Bitfield Value** option and specify a custom value for the RCW bit field, as shown in the figure below.

Figure 24. Specifying a custom RCW bit field value



○ 0b00 - For all valid Platform PLL frequencies
● Custom Bitfield Value
0b11

Generic field to allow for 4 options for analog/digital control of PLL. Note: the SoC Architecture specification defines Settings A-D. Often only one setting is enabled and remaining are reserved Note: if CFG_PLL_CONFIG_SEL_B==0, this RCW field is overridden--see Section 33.4, "Required Layerscape Chassis Configuration Pins" Options: 2'b00 - For all valid Platform PLL frequencies 2'b01 - Reserved 2'b10 - Reserved 2'b11 - Reserved
This item modifies SYS_PLL_CFG[1:0] bit field in the RCWSR1 register.
Version specific item: Settings supported only for devices using Reset Configuration Word (RCW).

4. Click **Apply**. The new value is added to the PBL configuration.

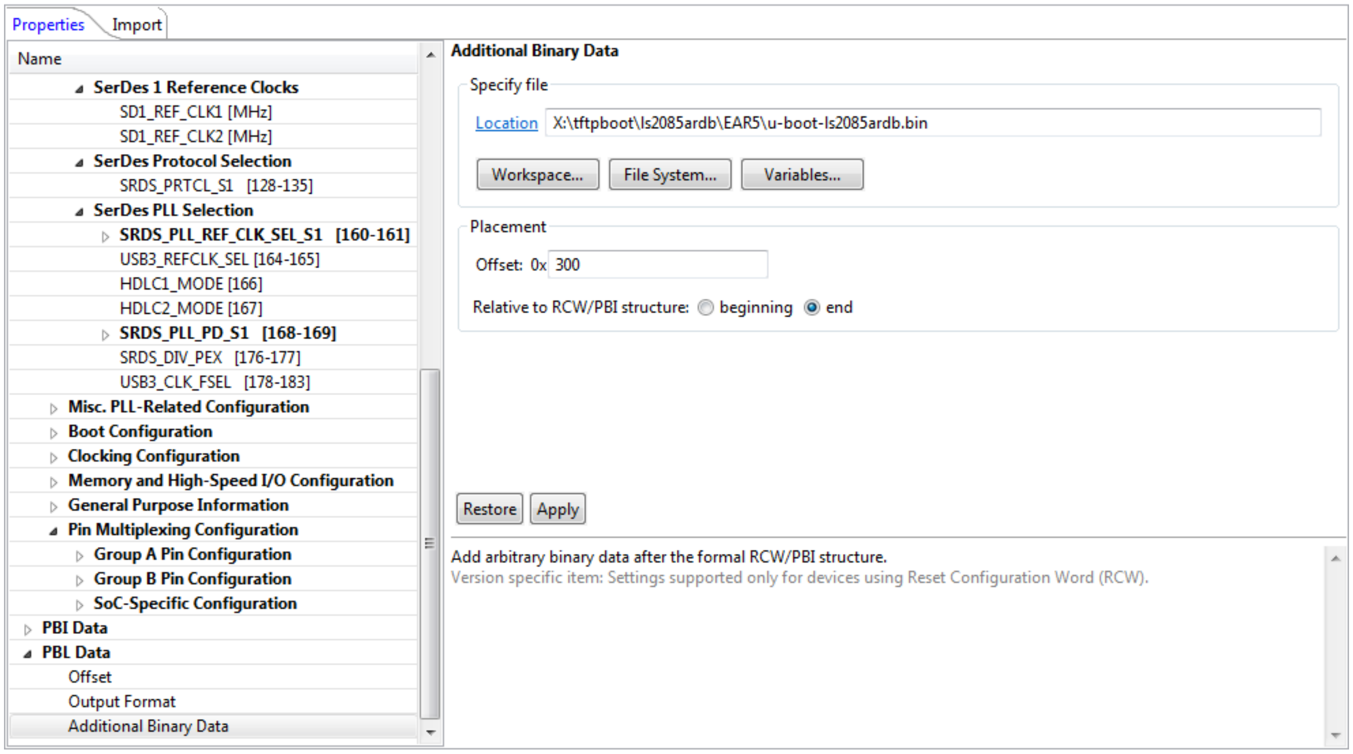
5.2. Add additional payload to a PBL image

You can add to a PBL image additional binary payload, such as U-Boot. This is useful to create boot images for the SPI/SD/NAND flash when PBL can be edited without decoupling it from the U-Boot binary. At code generation, the binary payload is automatically re-attached to the modified PBL.

To add additional binary data to the PBL image, perform these steps:

1. Select the **Additional Binary Data** property under the **PBL Data** group in the **Name** column on the **Properties** page of the **Component Inspector** view.
2. Click the ellipsis (…) button in the **Value** column. The **Additional Binary Data** editor opens.
3. Specify a binary file to be added to the PBL image, in the **Location** field of the **Specify file** group.
4. Specify the offset and placement for the binary payload in the **Placement** group, as shown in the figure below.

Figure 25. Additional Binary Data editor



Location X:\tftpboot\ls2085ardb\EAR5\u-boot-ls2085ardb.bin

Offset: 0x 300

Relative to RCW/PBI structure: beginning [✓ end]

Add arbitrary binary data after the formal RCW/PBI structure.
Version specific item: Settings supported only for devices using Reset Configuration Word (RCW).

5. Click **Apply** to apply the changes.

5.3. Errata support

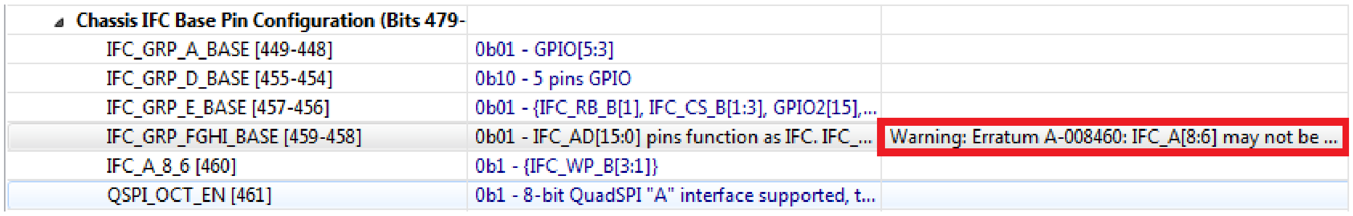
The PBL tool handles errata in the following two ways:

- If the PBL tool can implement the workaround for the erratum impacting the current PBL configuration, then the tool automatically implements the erratum workaround, without notifying the user. Examples of such errata include:
  - An RCW bit field option may not be available anymore
  - A chosen RCW bit field option may translate into a different value (as specified by the errata document of the chosen system-on-chip (SoC)) in the generated PBL binary

- If the PBL tool cannot implement the workaround for the erratum impacting the current PBL configuration, then the tool displays a warning message with the details of the erratum. An example is an erratum that depends on the conditions that cannot be determined by only assessing the PBL configuration (such as, operating temperature, dual inline package (DIP) switch configuration). For such an erratum, the user needs to manually implement the erratum workaround.

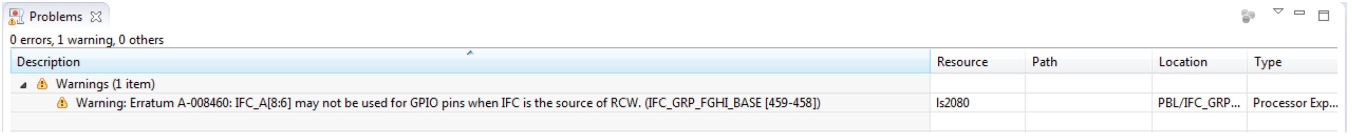
The warning message is displayed for the problematic field on the **Properties** page in the **Component Inspector** view and also in the **Problems** view, as shown in the following figures.

Figure 26. Warning displayed in Component Inspector view



◢ Chassis IFC Base Pin Configuration (Bits 479-		
IFC_GRP_A_BASE [449-448]	0b01 - GPIO[5:3]	
IFC_GRP_D_BASE [455-454]	0b10 - 5 pins GPIO	
IFC_GRP_E_BASE [457-456]	0b01 - {IFC_RB_B[1], IFC_CS_B[1:3], GPIO2[15],...	
IFC_GRP_FGHI_BASE [459-458]	0b01 - IFC_AD[15:0] pins function as IFC. IFC_...	Warning: Erratum A-008460: IFC_A[8:6] may not be ...
IFC_A_8_6 [460]	0b1 - {IFC_WP_B[3:1]}	
QSPI_OCT_EN [461]	0b1 - 8-bit QuadSPI "A" interface supported, t...	

Figure 27. Warning displayed in Problems view



Problems

0 errors, 1 warning, 0 others

Description	Resource	Path	Location	Type
◢ Warnings (1 item)				
Warning: Erratum A-008460: IFC_A[8:6] may not be used for GPIO pins when IFC is the source of RCW. (IFC_GRP_FGHI_BASE [459-458])	ls2080		PBL/IFC_GRP...	Processor Exp...

5.4. Endianness aspects

When you import data in the XXD Object Dump format, you can specify the endianness of the data. However, specifying the endianness is only useful when the data is organized into multibyte words. The endianness option is automatically set to the endianness of the chosen SoC, for example, little endian for the ARMv8-based SoCs.

6. PBL tool limitations

The PBL tool has some known limitations related to:

- Reserved bits
- PBI commands

6.1. Reserved bits

When generating a PBL image, the PBL tool uses the default values of the bits that are marked as *Reserved* in the SoC reference manual. Due to this limitation, some mismatches may occur between the imported PBL image and generated PBL image.

6.2. PBI commands

If the current PBL configuration has PBI commands defined and you import a new PBL image that does not have PBI commands, then the PBI commands are not preserved and you need to manually add them for the new PBL configuration.

Document Number: AN5260

23 February 2016