AN5260

PBL Configuration using QCVS

Rev. 1 — 11/2020 Application Note

Contents

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

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, imported from an existing reset configuration word (RCW) memory dump (such memory dumps can be obtained using U-Boot), or created based on data read from target.

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
   - Read configuration from target's Reset Configuration Word Status Registers

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 want to start from the RCW read from target.

The figure below shows the **PBL configuration** page with the **Create default configuration** option selected.

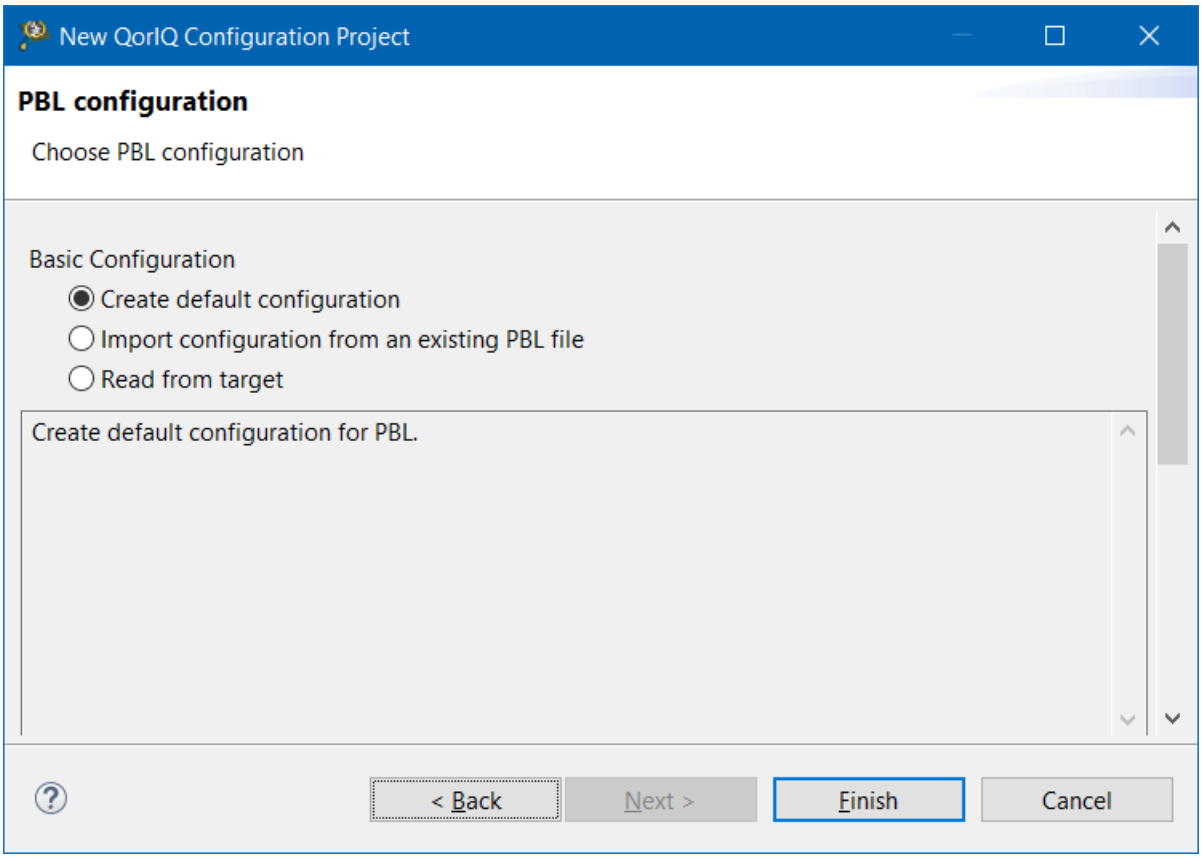

Figure 1. Creating a default configuration

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.

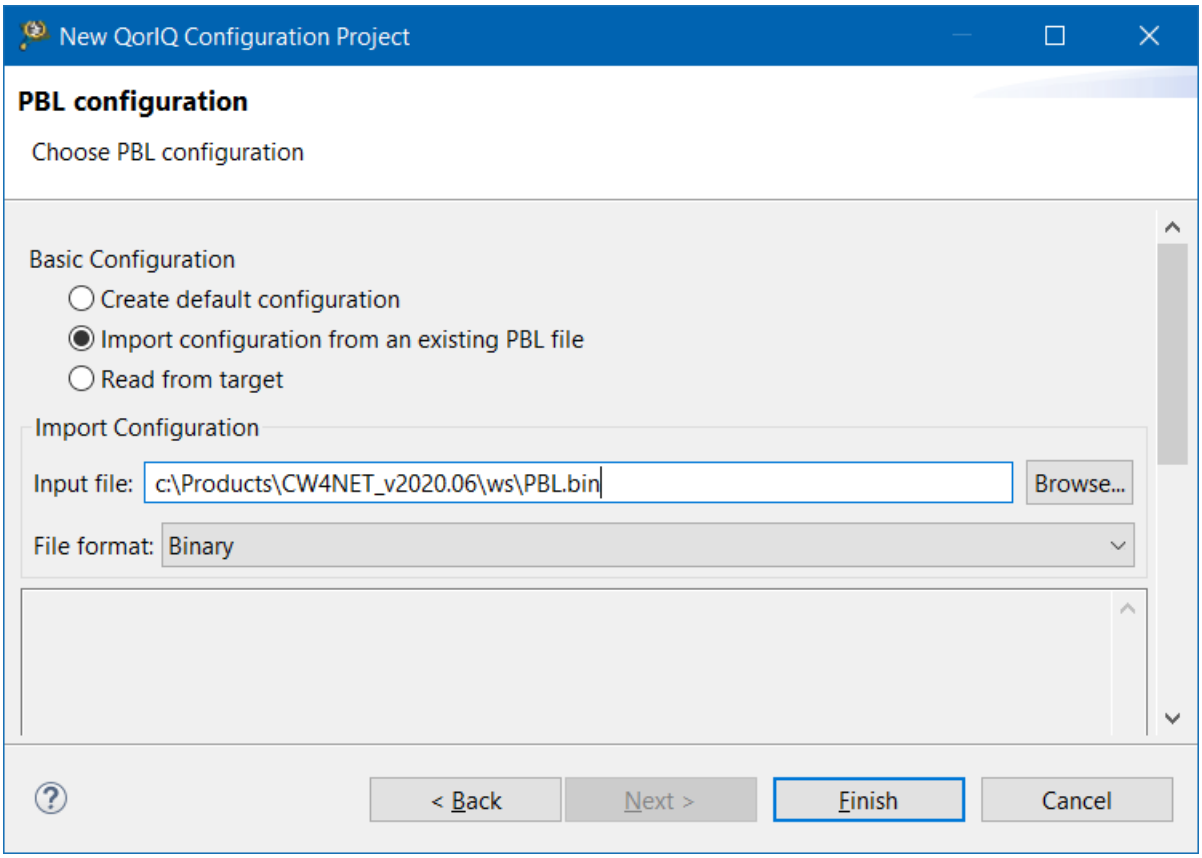

Figure 2. Importing a configuration from an existing PBL file

3.3 Read configuration from target's Reset Configuration Word Status Registers

The existing 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 **Read from target** option to create a QorIQ configuration project with the PBL tool. Target RCWs 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 **Read from target** option selected.

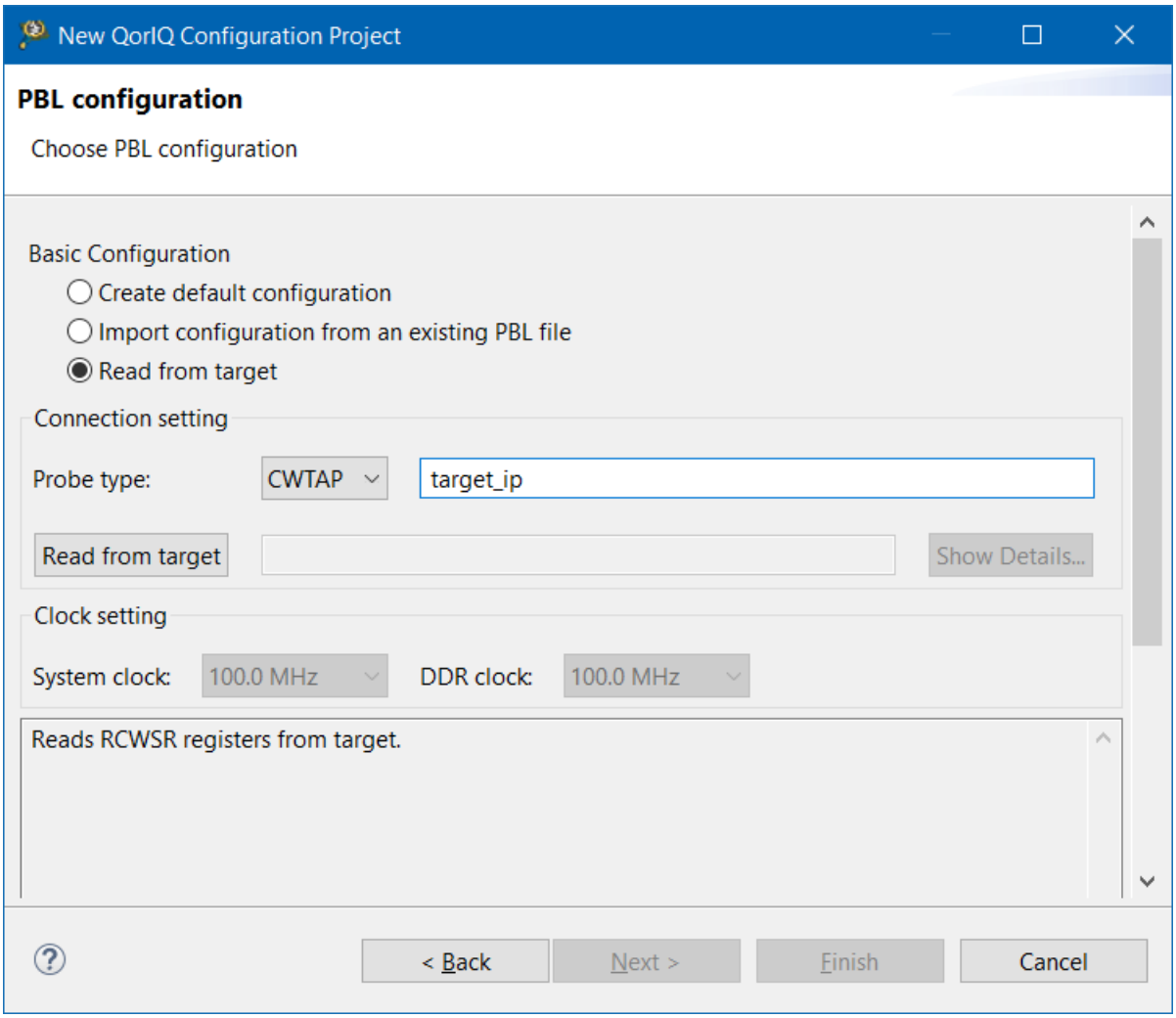

Figure 3. Reading configuration from target

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.

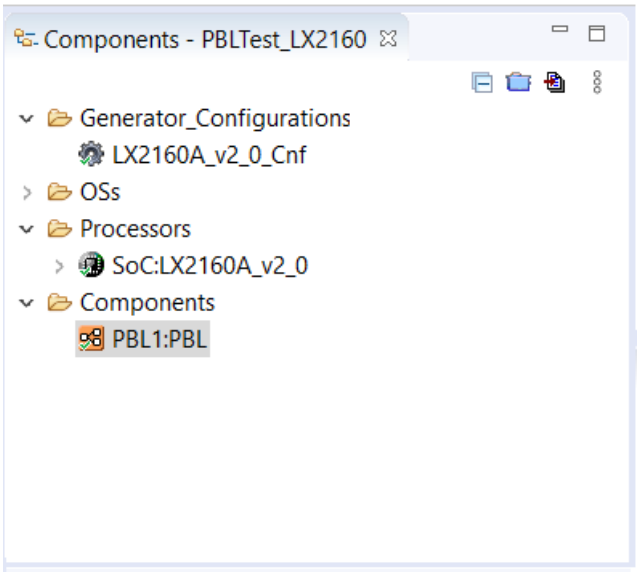

Figure 4. Components view

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.

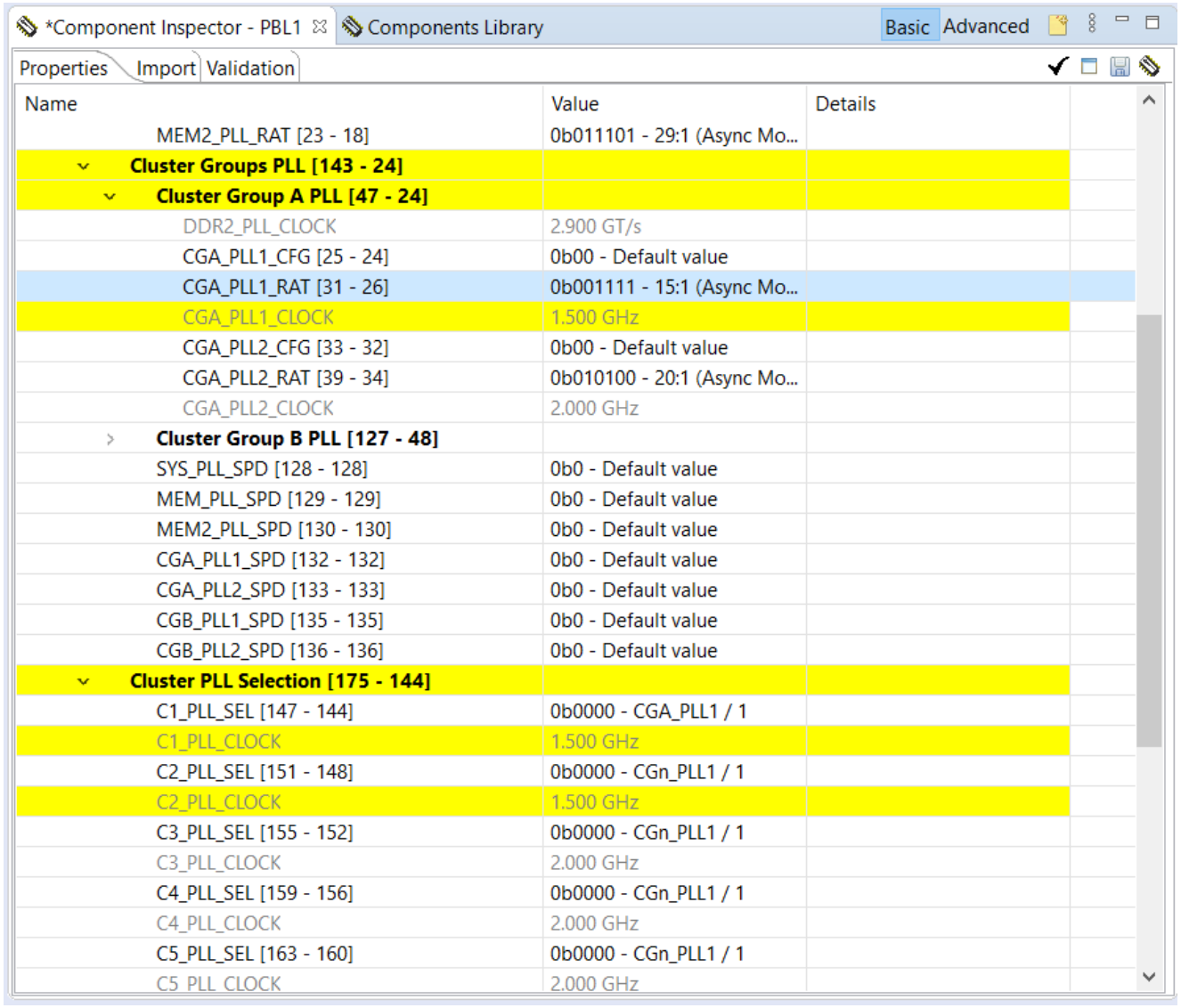

Figure 5. Component Inspector view

As you can see in the figure above, the properties of the PBL component are grouped under various categories and subcategories. The properties are sorted by the RCW position.

You can also notice that some of the component properties are grayed out. These are read-only properties; they cannot be changed. These properties are computed based on RCW fields.

Another useful feature of this presentation is that the most recently modified properties are displayed with yellow background. This way you can easily determine which other properties depend on the last modified properties.

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

- Change RCW bit field values
- Specify custom values
- Display and set reserved fields
- Add PBI commands to a PBL image
- Import a PBL configuration from a file

- Import a PBL configuration from target
- Generate a PBL image
- Automatic PBL validation and errata support
- PBL validation tool
- 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.

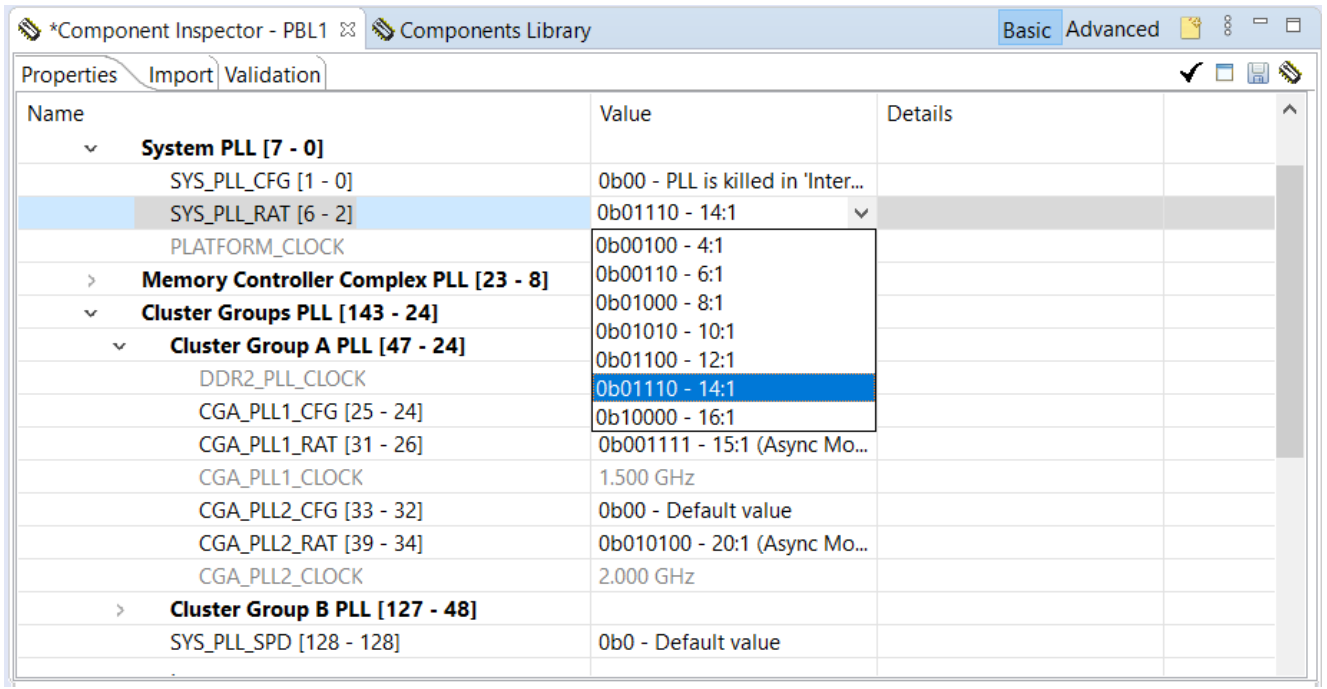

Figure 6. Changing a field value by choosing another value from a menu

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

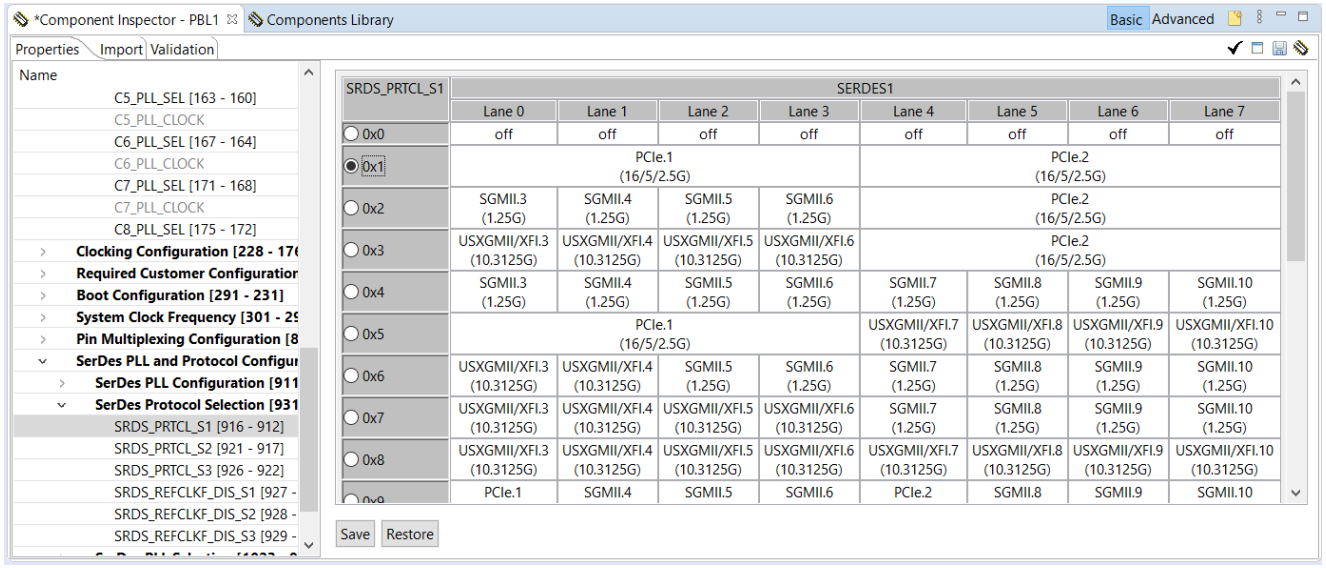

Figure 7. Changing a field value by using an advanced GUI

4.2 Specify custom values

You can observe that some fields on the **Properties** page do not show a list of possible values and some other fields show a list of values where few values are missing. This is due to the reason that by default, the PBL tool restricts field values to the ranges specified in the SoC reference manual. The PBL tool also validates entered values against known constraints and generates errors in the Component Inspector view and Problems view when constraints are violated.

Both these features can be turned off by enabling the **Skip error checking** option available on the **Enable/disable error checking** toolbar menu. This allows you to set any RCW field to any value, even if such a configuration is likely or certain to cause the SoC fail to come out of reset or function improperly.

To specify a custom value for an RCW field, follow these steps:

1. In the Component Inspector view, click **Enable/disable error checking** and choose **Skip error checking**, as shown in the figure below.

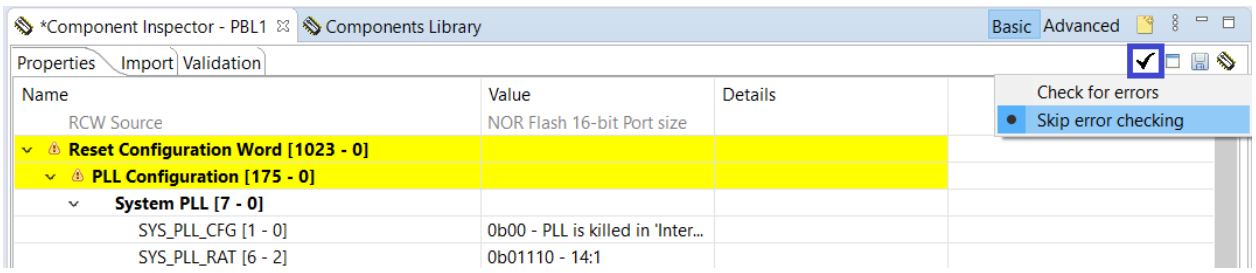

Figure 8. Disabling error checking

2. Select the property you want to specify a custom value for, add the new value in the **Value** column for the property (see figure below), and press **Enter**.

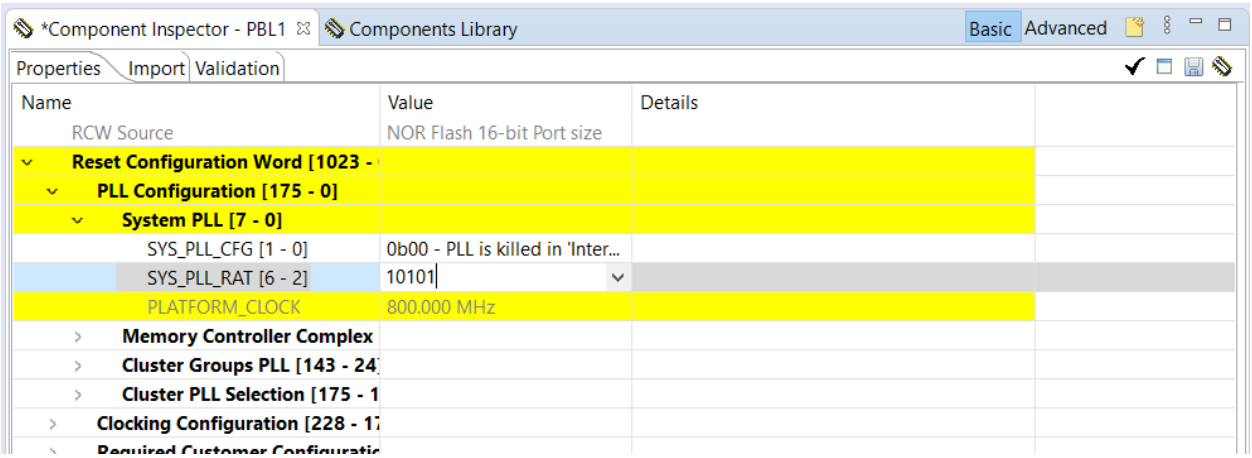

Figure 9. Adding custom value to a field

The value added by you is marked as a custom value for the field and it is available for selection in the list of values, as shown in the figure below.

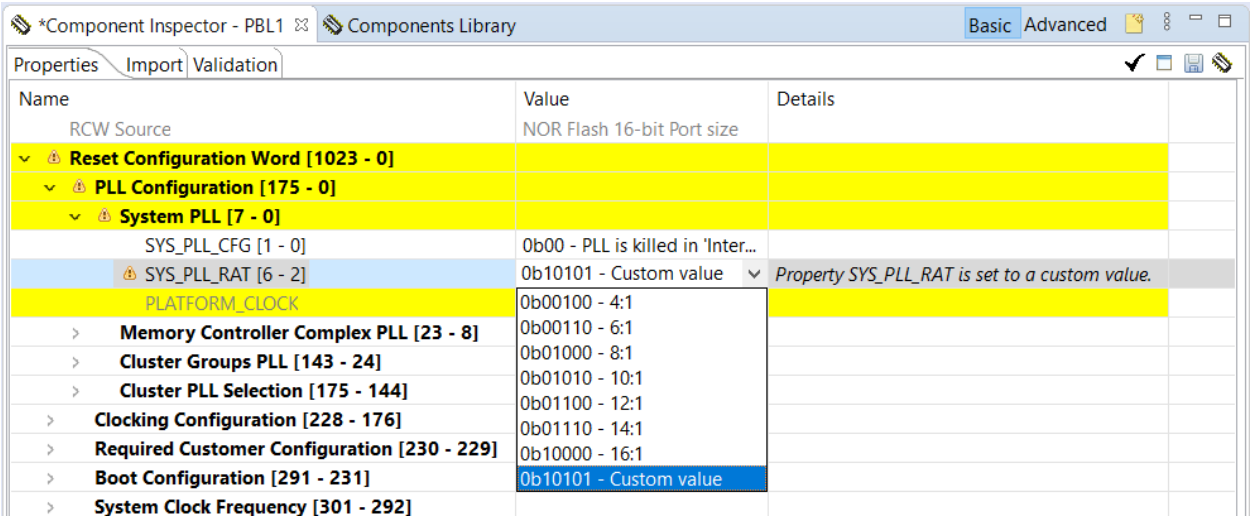

Figure 10. Selecting a custom value for a field

4.3 Display and set reserved fields

After enabling the **Skip error checking** option, you can edit the values for the reserved bit fields, apart from the non-reserved bit fields. However, before editing reserved bit field values, you need to make reserved bit fields visible by choosing **Display all fields** from the **View Mode** toolbar menu, as shown in the figure below.

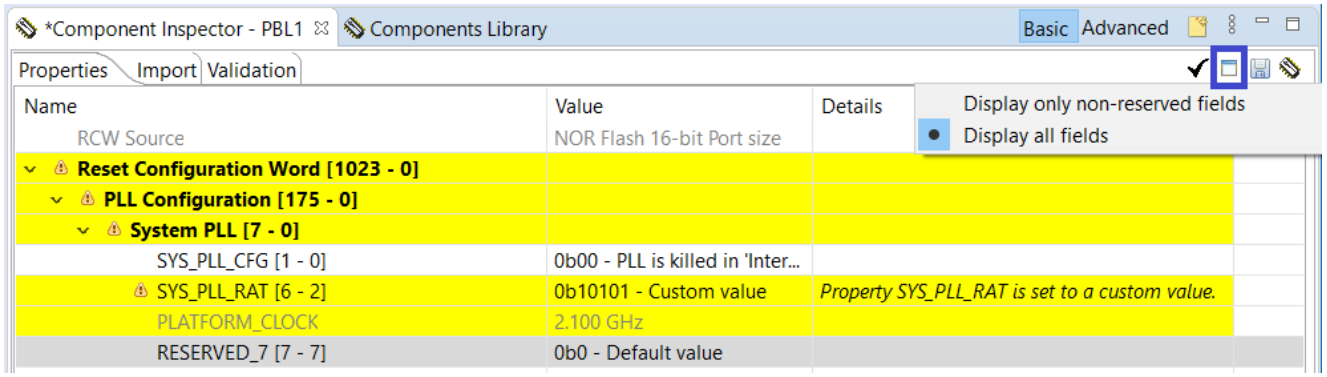

Figure 11. Displaying reserved fields

4.4 Add PBI commands to a PBL image

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

1. Expand the **PBI Data** property category on the **Properties** page of the **Component Inspector** view and click anywhere within any cell of the **Pbi Data** row, as shown in the figure below.

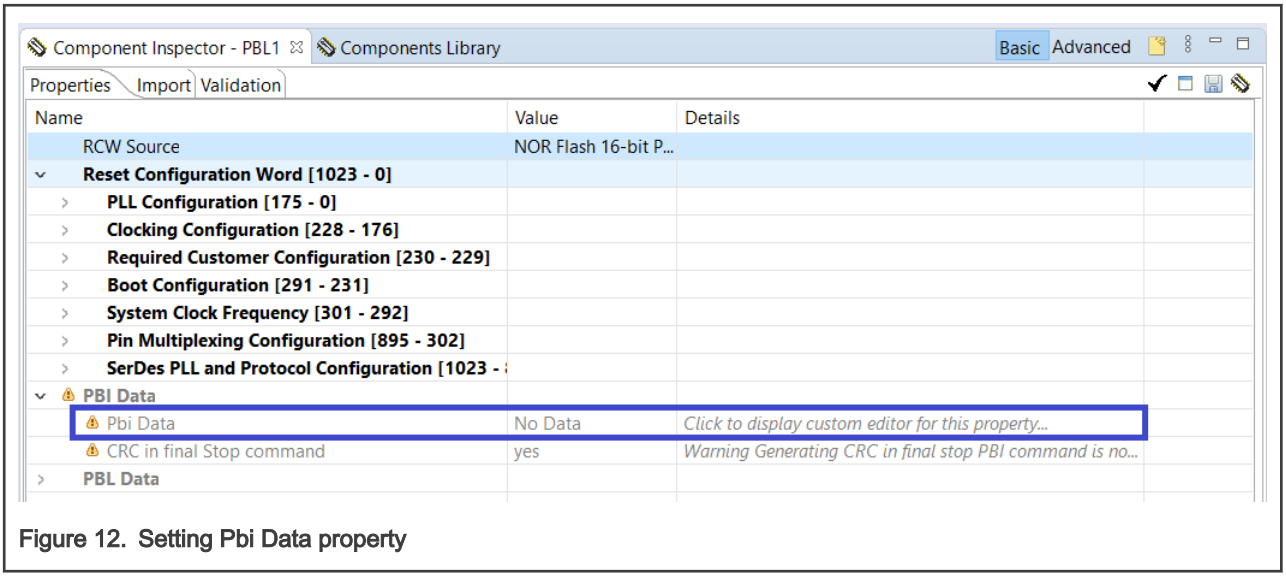

Figure 12. Setting Pbi Data property

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

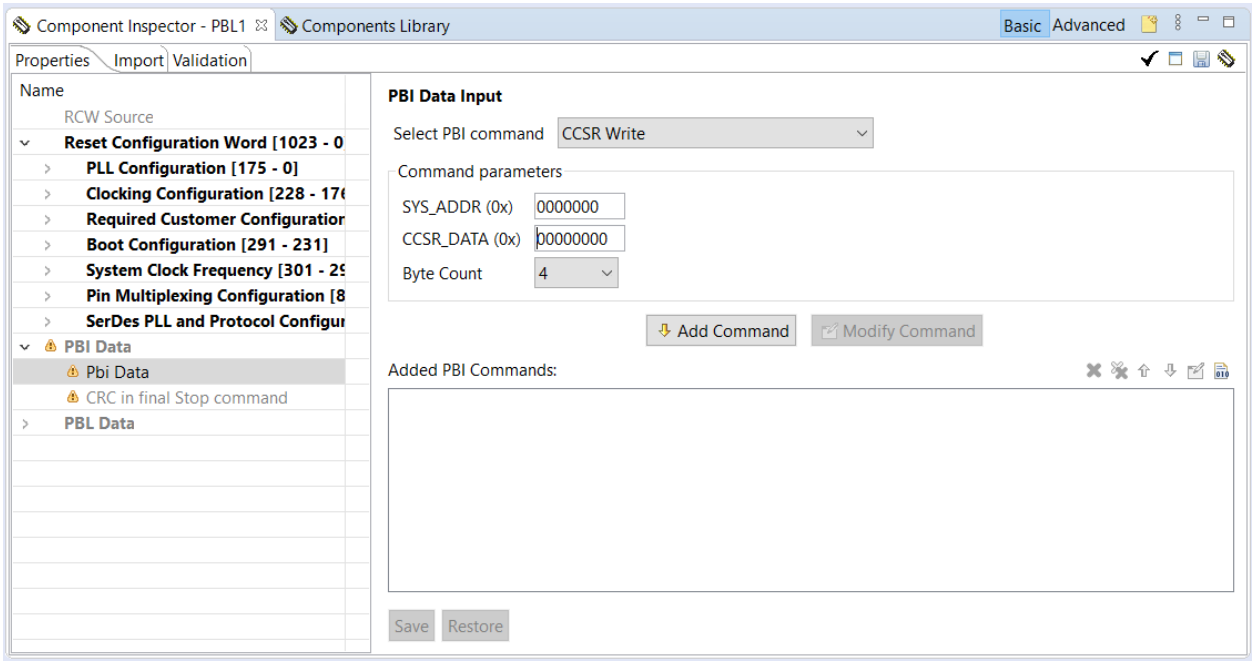

Figure 13. PBI Data Input editor

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

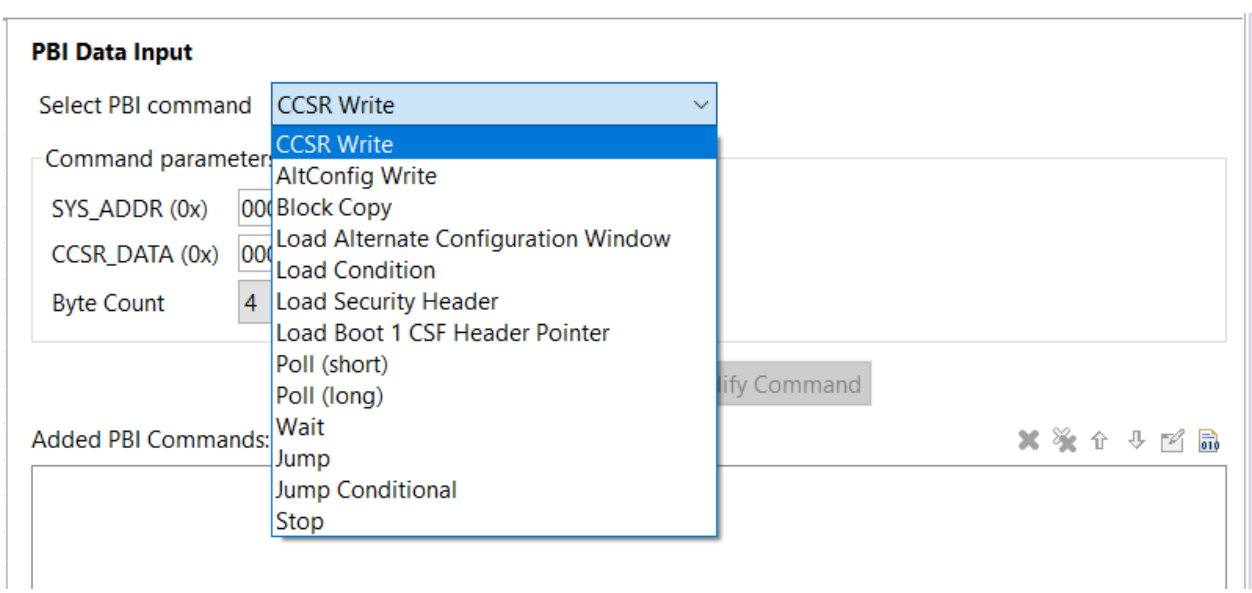

Figure 14. Choosing a PBI command

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.

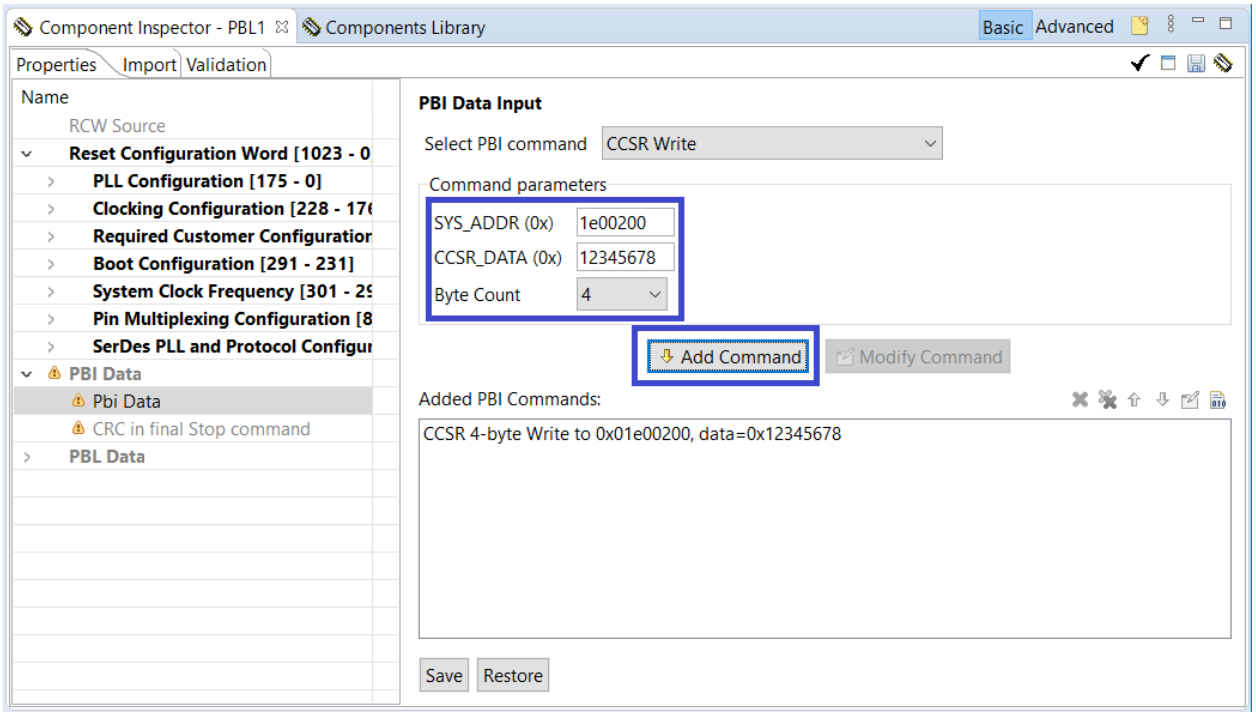

Figure 15. Editing a PBI command

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.

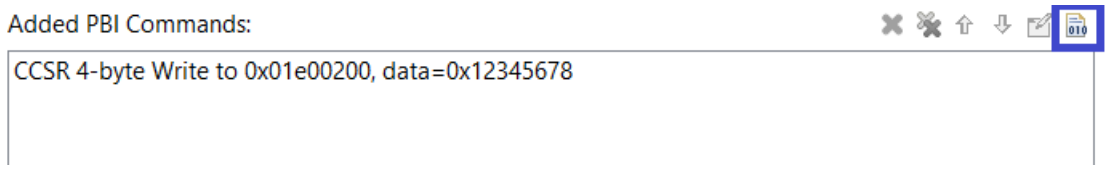

Figure 16. Disassembly view

Figure 17. Raw data view

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

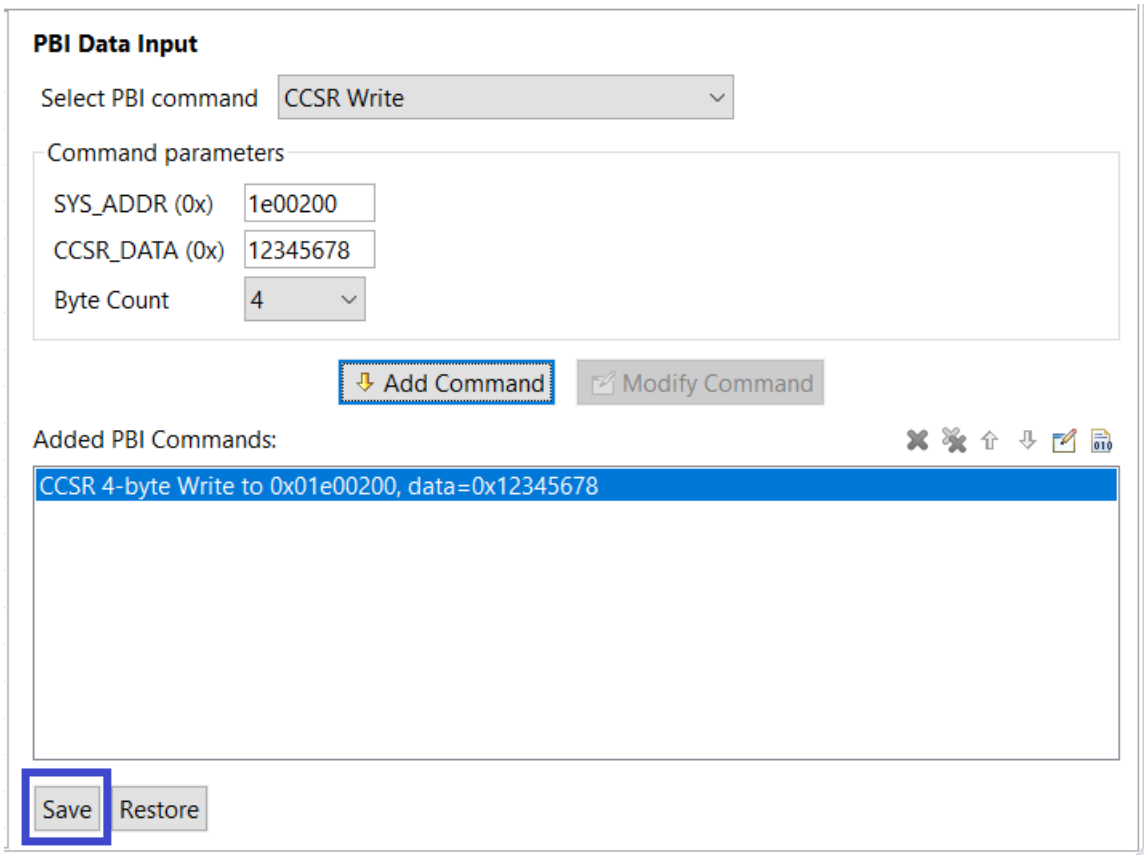

Figure 18. Adding PBI commands to a PBL image

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.

4.5 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 automatically detected 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.

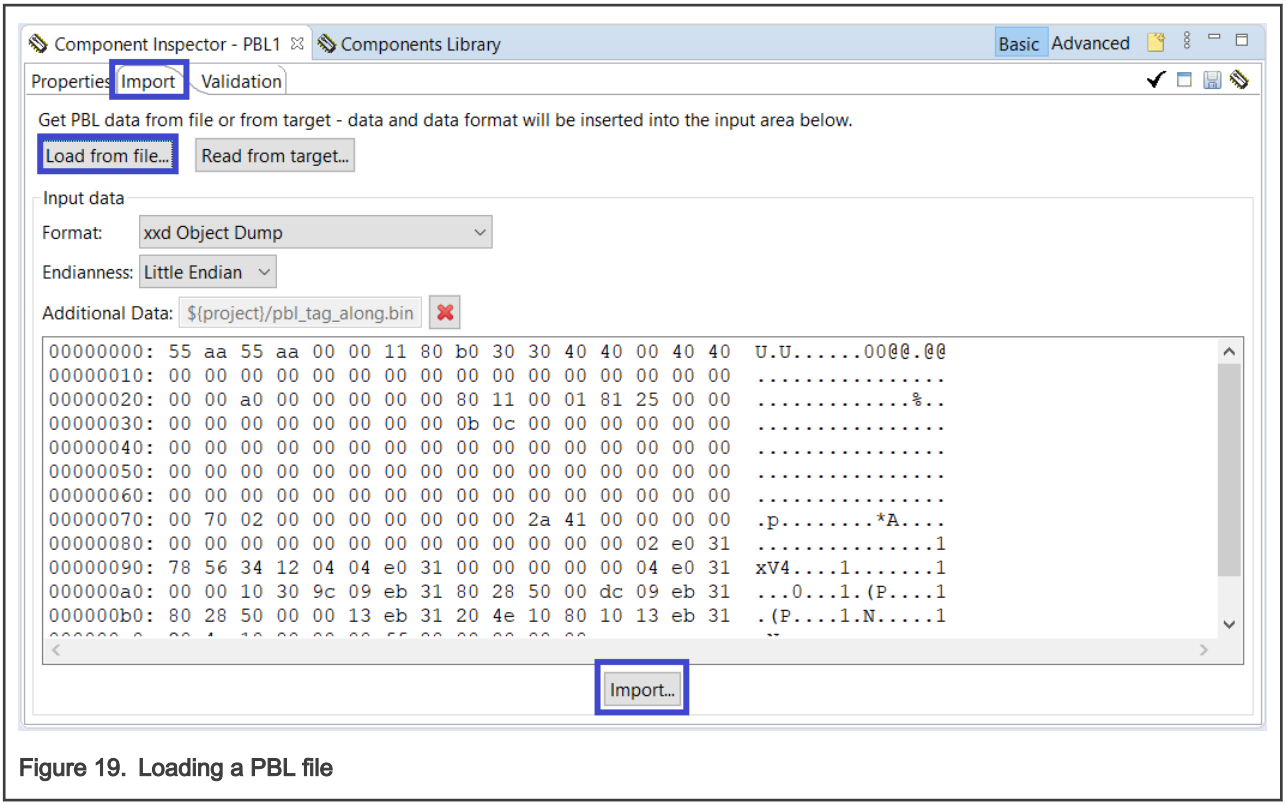

Figure 19. Loading a PBL file

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
- U-Boot Flash Dump
- Hex String
- U-Boot CCRS Startup Dump (RCW only)
- Hex String (RCW only)
- Text 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.

4.6 Import a PBL configuration from target

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. Click the **Read from target** button. The file format of the chosen file is automatically set to **Import From Target** and it starts reading the RCW information from the current target; when the data is read, it will be displayed in the **Input data** group, as shown in the figure below.

   Before choosing to read PBL information from target, ensure that you have activated the right target in the **Target connections** group of the **Connections View**.

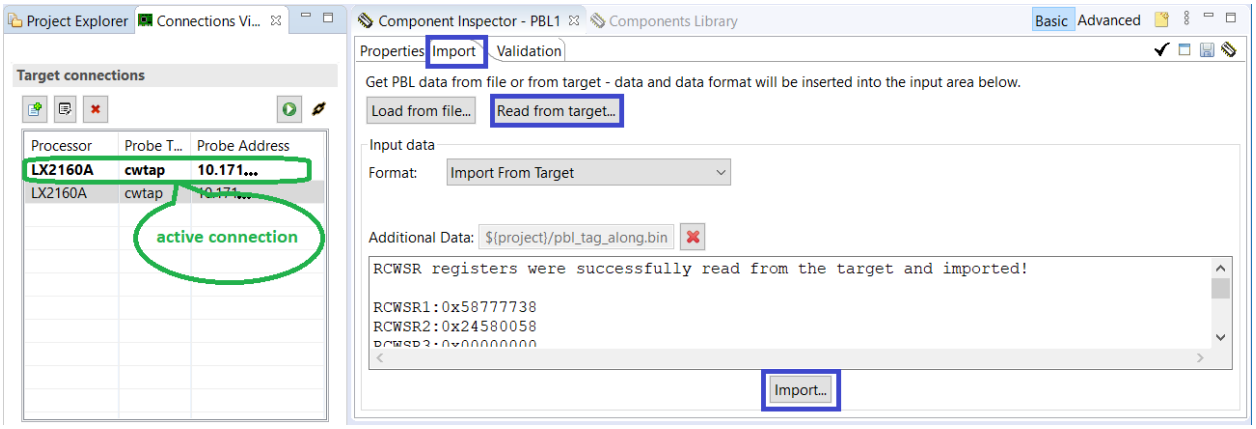

Figure 20. Read PBL configuration from target

3. Click the **Import** button to import the new PBL configuration.
4. Switch to the **Properties** tab to view or edit the imported PBL configuration.

4.7 Generate a PBL image

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

1. Select the **Output Format** toolbar button.
2. Choose a file format for the PBL image.

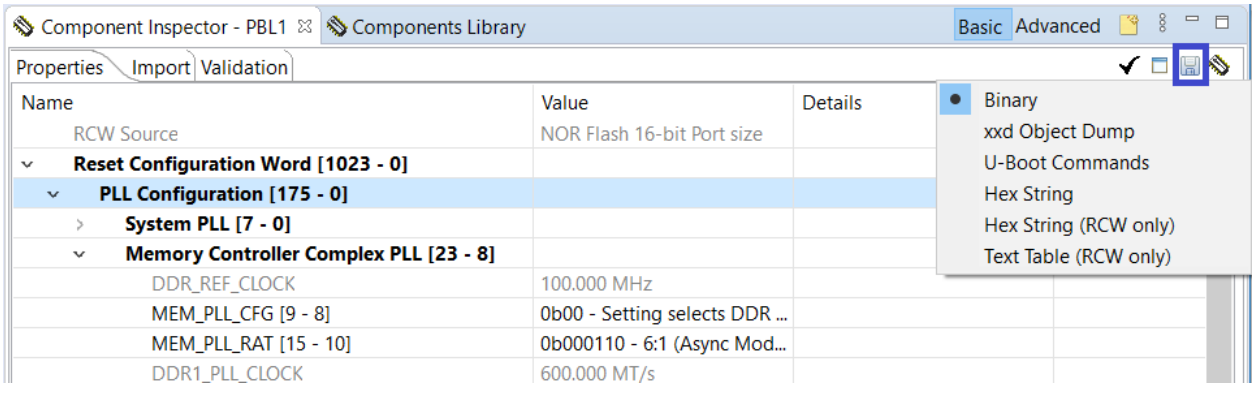

Figure 21. Choosing a file format for PBL file

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

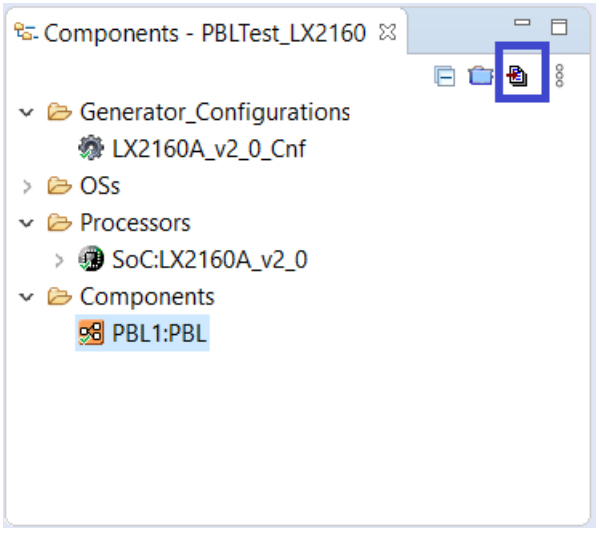

Figure 22. Generating PBL image

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

- Binary
- XXD Object Dump
- U-Boot Commands
- Hex String
- Hex String (RCW only)
- Text Table (RCW only)

4.8 Automatic PBL validation and errata support

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

You can use **Skip error checking** mode to suppress critical errors; they will be marked as warnings and you will be able to generate the configuration.

The figure below illustrates two situations: an error caused by a field set to a custom value and a warning caused by an errata check.

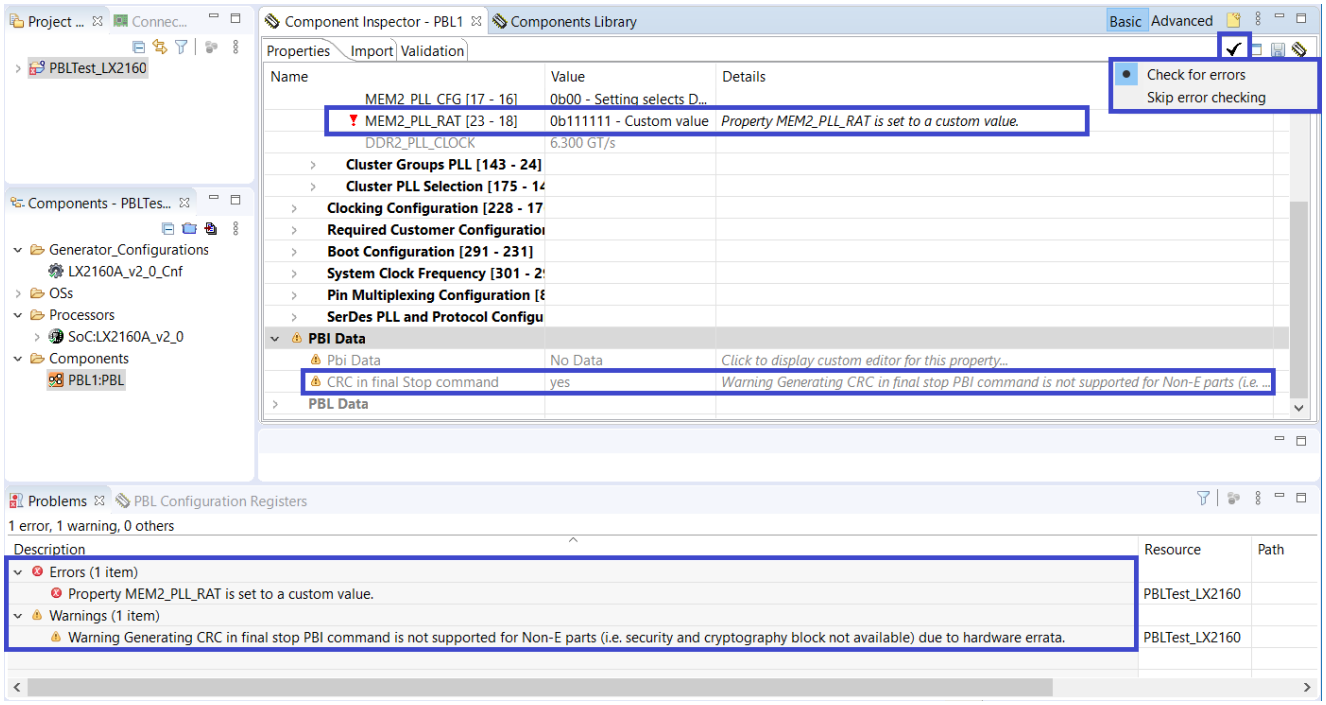

Figure 23. PBL validation

4.9 PBL validation tool

The PBL validation tool can be used for RCW validation, after a PBL configuration is created using the PBL configuration tool.

The Component Inspector view displays a page, **Validation**, which represents the GUI of the PBL validation tool. To use the PBL validation tool, select the **Validation** tab of the **Component Inspector** view, as shown in the figure below.

If you click the **Write Reset Configuration Word** button, it will override the RCW on target and will perform a target reset. If the processor core is working correctly with the written RCW and it does not return any error, then the test will pass.

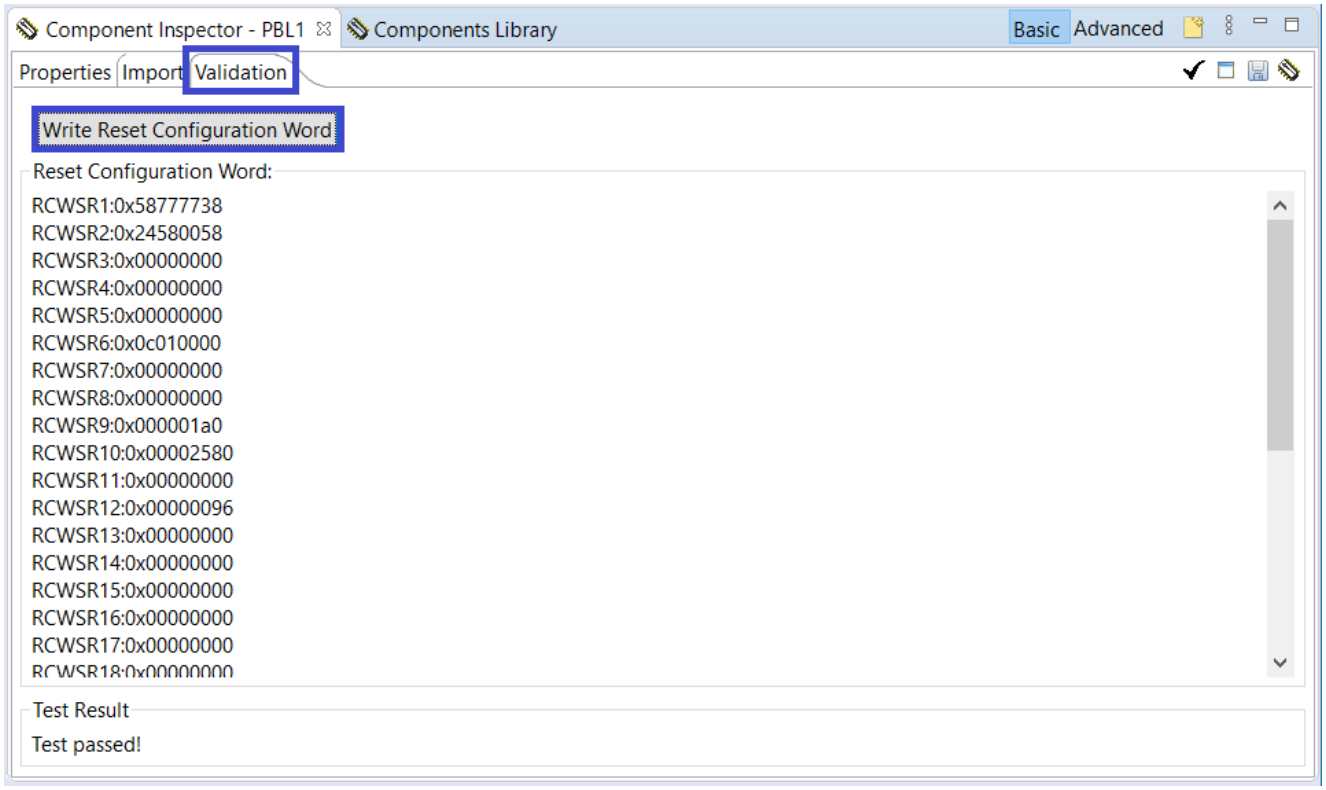

Figure 24. PBL validation tool

4.10 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.

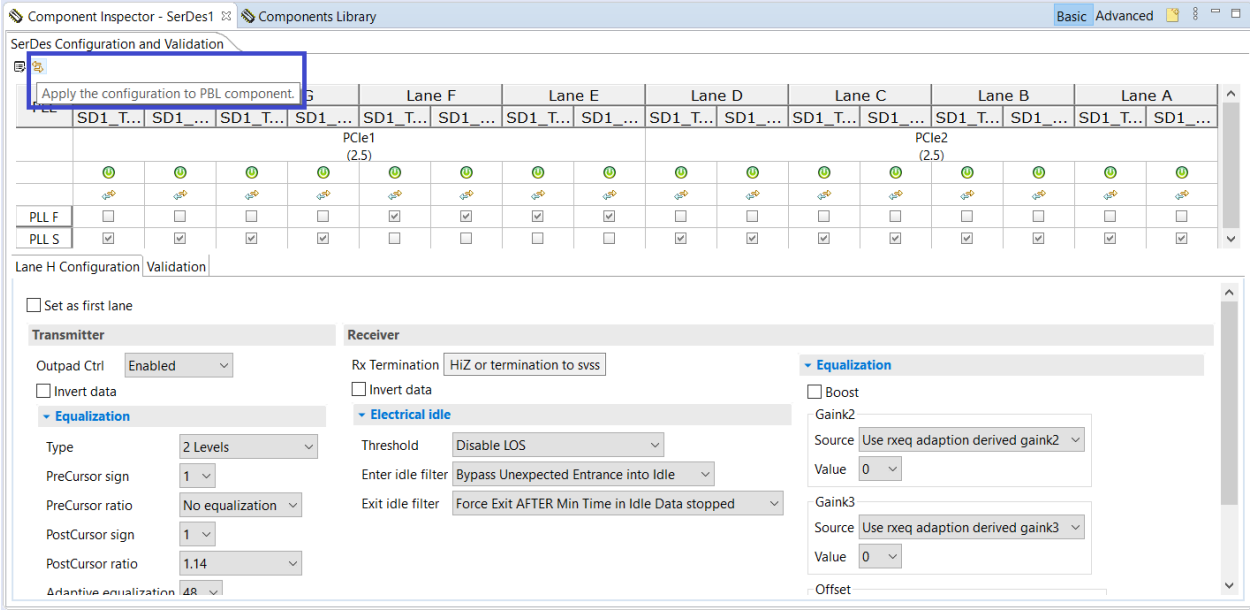

Figure 25. Synchronizing PBL and SerDes components

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.

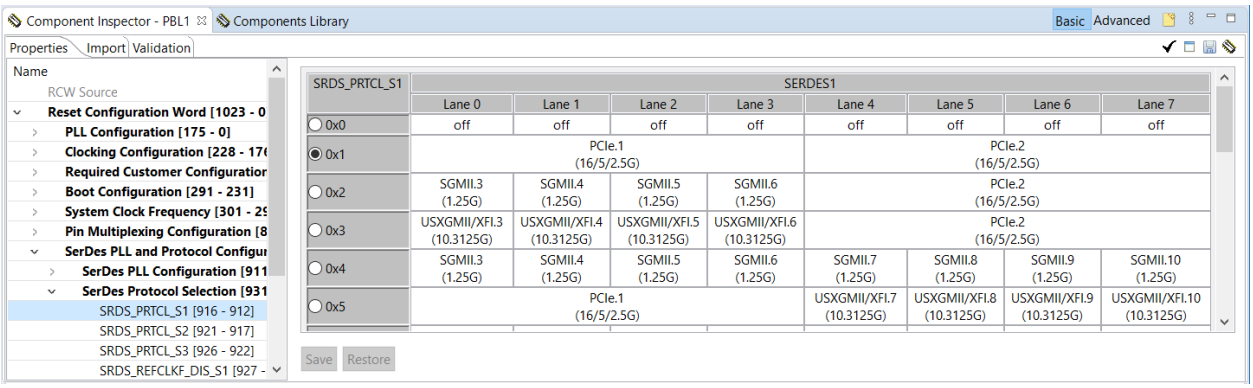

Figure 26. Verifying SerDes fields of PBL component

4.11 View RCW status registers

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

1. Choose **PBL Configuration Registers** from the toolbar. The **PBL Configuration Registers** view appears, displaying the details of the RCW status registers, as shown in the figure below.

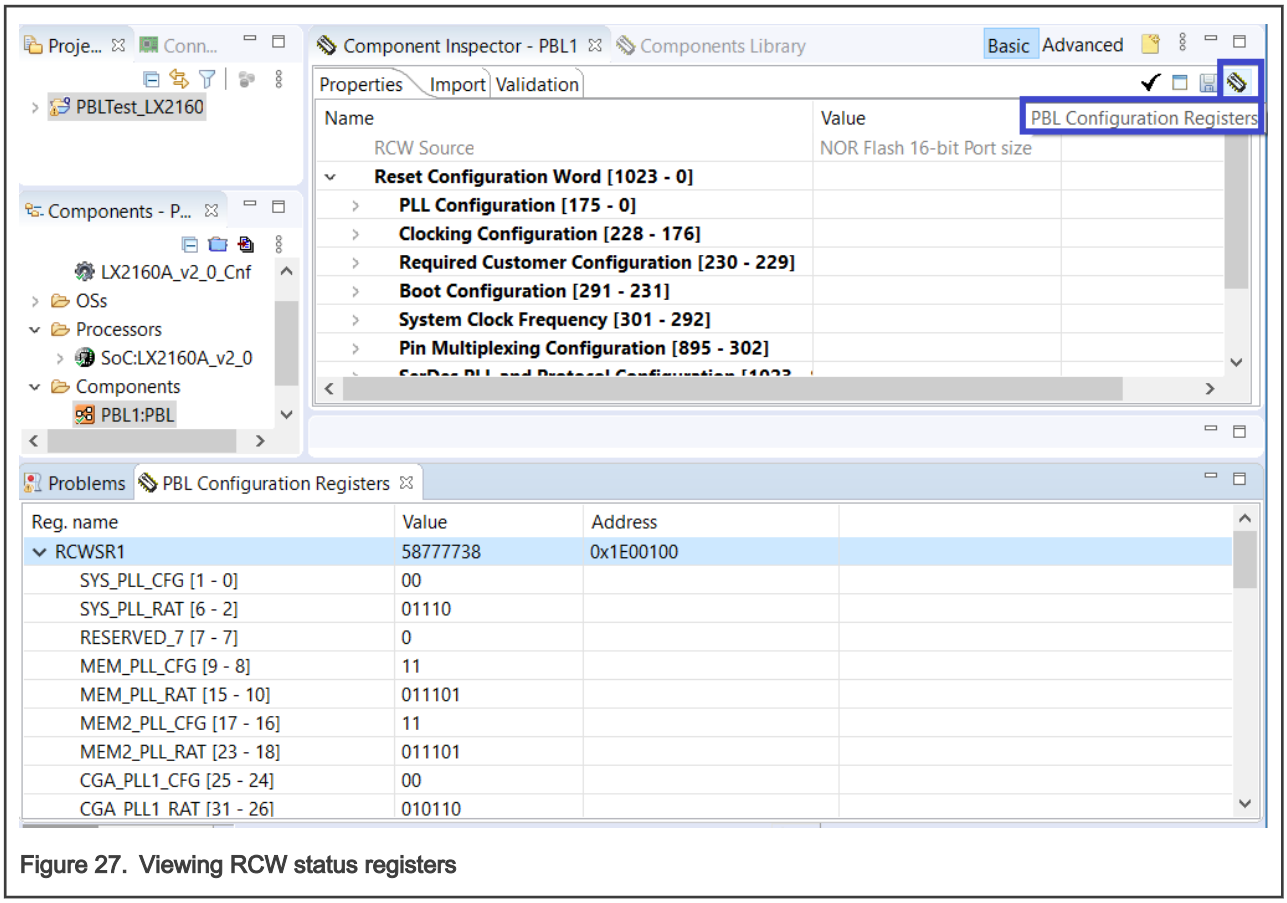

Figure 27. Viewing RCW status registers

The **PBL Configuration Registers** view reflects any changes made in the PBL configuration. Also, from this view you can edit register bit field values or specify a certain value for the register; if you change the register value, then its bit fields and the corresponding properties will automatically be updated.

5 Advanced PBL operations

This section is divided into the following subsections:

5.1 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. Expand the **PBL Data** property category on the **Properties** page of the **Component Inspector** view and click the **Additional Binary Data** property. The **Additional Binary Data** editor opens, as shown in the figure below.

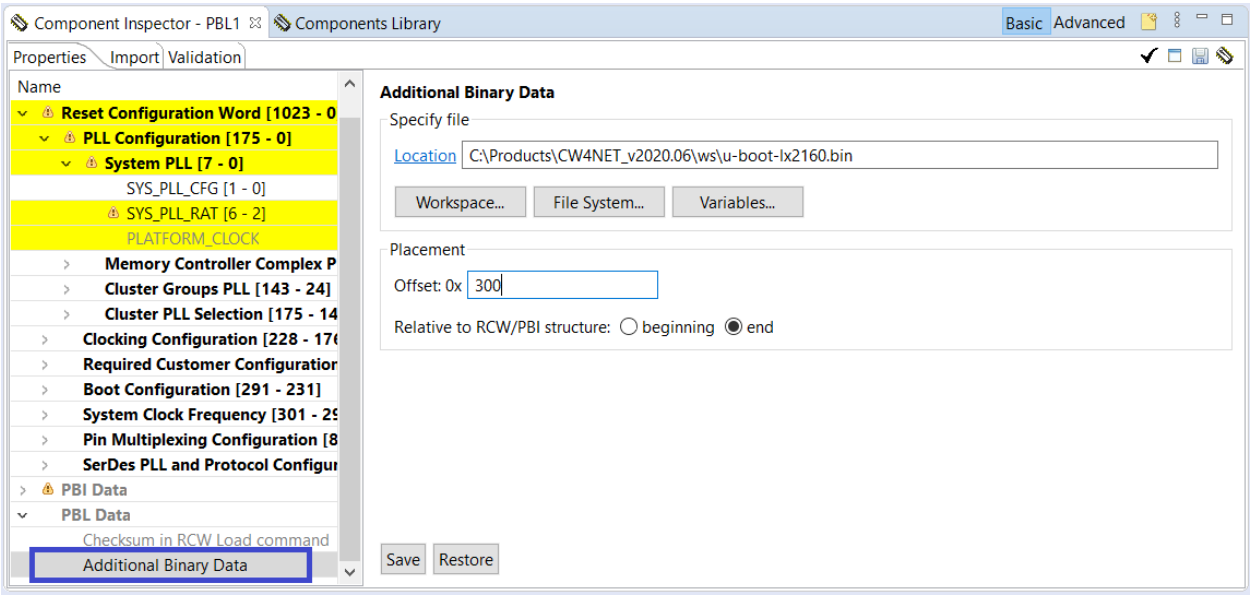

Figure 28. Additional Binary Data editor

2. Specify a binary file to be added to the PBL image, in the **Location** field of the **Specify file** group.
3. Specify the offset and placement for the binary payload in the **Placement** group.
4. Click **Save** to apply your changes.

5.2 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:

- PBI commands

6.1 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.

A Revision history

The table below summarizes revisions to this document.

Table 1. Revision history

Revision	Date	Topic cross-reference	Change description
Rev. 1	11/2020		• Updated entire document as per latest QCVS PBL tool

Table continues on the next page...

Table 1. Revision history (continued)

Revision	Date	Topic cross-reference	Change description
			• Updated images throughout the document to show a latest supported device
Rev. 0	02/2016		Initial public release

How To Reach Us

Home Page:

nxp.com

Web Support:

nxp.com/support

Information in this document is provided solely to enable system and software implementers to use NXP products. There are no express or implied copyright licenses granted hereunder to design or fabricate any integrated circuits based on the information in this document. NXP reserves the right to make changes without further notice to any products herein.

NXP makes no warranty, representation, or guarantee regarding the suitability of its products for any particular purpose, nor does NXP assume any liability arising out of the application or use of any product or circuit, and specifically disclaims any and all liability, including without limitation consequential or incidental damages. “Typical” parameters that may be provided in NXP data sheets and/or specifications can and do vary in different applications, and actual performance may vary over time. All operating parameters, including “typicals,” must be validated for each customer application by customer's technical experts. NXP does not convey any license under its patent rights nor the rights of others. NXP sells products pursuant to standard terms and conditions of sale, which can be found at the following address: nxp.com/SalesTermsandConditions.

Security — Customer understands that all NXP products may be subject to unidentified or documented vulnerabilities. Customer is responsible for the design and operation of its applications and products throughout their lifecycles to reduce the effect of these vulnerabilities on customer’s applications and products. Customer’s responsibility also extends to other open and/or proprietary technologies supported by NXP products for use in customer’s applications. NXP accepts no liability for any vulnerability. Customer should regularly check security updates from NXP and follow up appropriately. Customer shall select products with security features that best meet rules, regulations, and standards of the intended application and make the ultimate design decisions regarding its products and is solely responsible for compliance with all legal, regulatory, and security related requirements concerning its products, regardless of any information or support that may be provided by NXP. NXP has a Product Security Incident Response Team (PSIRT) (reachable at PSIRT@nxp.com) that manages the investigation, reporting, and solution release to security vulnerabilities of NXP products.

For more information, please visit: http://www.nxp.com
For sales office addresses, please send an email to: salesaddresses@nxp.com

Date of release: 11/2020
Document identifier: AN5260

arm

BUILT ON Power