Table 63: Kernel Module Project Launch Configuration - Modules tab settings (continued)

Option	Description
Prompt for symbolics path if not found	Prompts to locate the symbolics file if a mapping for it is not available in the settings A **Browse** dialog appears that allows you to browse for a module file containing symbolics. The debugger will add the specified symbolics to the modules' symbolics mapping.
Keep target suspended	Keeps the target suspended after the debugger loads the symbolics file for a module. This option is useful if you want to debug the module's initialization code. It allows you to set breakpoints in the module's initialization code before running it. **NOTE** This option is automatically enabled when activating the **Prompt for symbolics path if not found** option.

NOTE

Breakpoints are resolved each time a symbolics file is loaded and the debugger uses the modules unload events for symbolics disposal and breakpoints cleanup.

Chapter 8
JTAG Configuration Files

This chapter explains about JTAG configuration files that pass specific configuration settings to the debugger and support chaining of multiple devices.

A JTAG configuration file is a text file, specific to the CodeWarrior debugger, which describes a custom JTAG scan chain. You can specify the file in the remote system settings.

This chapter explains:

8.1 JTAG configuration file syntax

This section describes the syntax of a JTAG configuration file.

You can create a JTAG configuration file that specifies the type, chain order, and various settings for the devices you want to debug. To create the JTAG configuration file, list each device on a separate line, starting with the device that is directly connected to the transmit data out (TDO) signal (Pin 1) of the 16-pin COP/JTAG debug connector on the hardware target, and conclude with a blank line.

The listing below shows the complete syntax for a JTAG configuration file.

Figure 88: JTAG configuration file syntax

```
cfgfile:
     '\n'

     '#' 'any other characters until end of line'

     line

     cfgfile line

line:

     target

     target filter_list_or_params

target:

     target_name

     target_name = target_id

     'Generic' number number number

filter_list_or_params:

     filter_list_entity
```

```
     filter_list_or_params filter_list_entity

filter_list_entity:

     '(' number number ')'

     filter_name

     %
```

8.2 Using a JTAG configuration file to override RCW

You can use a JTAG configuration file to override reset configuration word (RCW) for a processor, such as LS1021A, revision 2.0.

In the following scenarios, the JTAG configuration files are used for overriding RCW:

- Programming RCW in a target board that does not have RCW programmed already
- New board bring-up
- Recovering a target board having a blank or corrupted flash: This feature is only available for LS1021A processor revision 2.0 and LS2085A. Board recovery scenarios involving a blank/corrupted flash require an external CodeWarrior TAP probe connection for overriding RCW; RCW override is not supported with a CMSIS-DAP connection.

NOTE

For more information on RCW, see the reference manual for your processor.

The CodeWarrior software includes example JTAG configuration files that can be used to override the RCW (see the listing below). The JTAG Configuration files are available at the following location:

```
<CWInstallDir>\CW_ARMv7\ARMv7\ARM_Support\Configuration_Files\jtag_chains
```

Figure 89: Sample JTAG configuration file for overriding RCW

```
# Example file to allow overriding a portion of the RCW
#
# Syntax:
#   LS102xA (0 RCW_source) (0x1000 RCW_option) (RCWn value) ...
#
#   where:
#   x = Processor version (LS1020/1/2A); default is LS1021A
#   RCW_source = The RCW source that you want to override (for example, 0x9b for hard-
coded mode)
#   RCW_option = 0 [RCW override disabled]
#                1 [RCW override enabled]
#
#   RCWn = 4096+n (n = 0 .. 15; index of RCW value)
#
#   value = 32-bit value
```

As specified in the listing above, the JTAG configuration files can be used to override a portion of the RCW for LS1021A, by specifying (index, value) pairs for some of the 16 x (32-bit words) of the RCW. For some targets

(for example, LS2085A), you do not need to specify `RCW_source`; therefore, you can remove `(0 RCW_source)` from the JTAG configuration file.

NOTE

You can use the pre-boot loader (PBL) tool to configure the various settings of the RCW and output the RCW in multiple formats, including CodeWarrior JTAG configuration files. For more information on the PBL tool, see QCVS PBL Tool User Guide.

8.3 Using JTAG configuration file to specify multiple linked devices on a JTAG chain

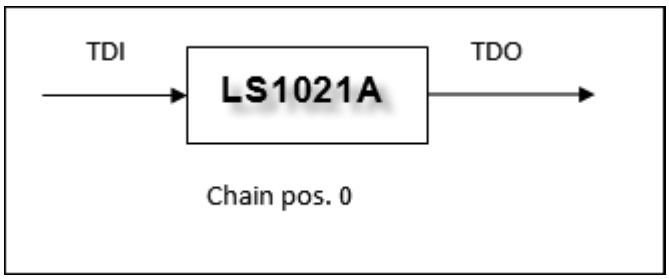
This section explains how to connect multiple processors through a single JTAG chain and how to describe such a JTAG chain in a JTAG configuration file.

The listing and figure below show a sample JTAG initialization file with a single core.

Figure 90: Sample JTAG initialization file for LS1021A processor

```
# A single device in the chain
LS1021A
```

Figure 91: A single device in a JTAG chain



The listing below show a sample JTAG initialization file with two devices in a JTAG chain.

Figure 92: Sample JTAG initialization file for LS1020A and LS1021A processors

```
# Two devices in a JTAG chain
LS1020A

LS1021A
```

NOTE

The devices are enumerated in the direction starting from TDO output to TDI input.

The listing below show two devices connected in a JTAG chain.

Figure 93: Sample JTAG initialization file for LS1020A and LS1021A processors

```
# Two devices in a JTAG chain
LS1020A (0x80000000 1)

LS1021A (2 1) (210005 0x90404000)
```

The listing below show two devices connected in a JTAG chain with a filter applied for the second device.

Figure 94: Sample JTAG initialization file for two devices with filter for second device

```
# Two devices in a JTAG chain
8306 (1 1) (2 0x44050006) (3 0x00600000)

8309 log
```

In the above example, the entry for the 8306 also includes the Hard Reset Control Word (HRCW) data that will overwrite the HRCW fetched by the 8306 upon power up or Hard Reset. The Hard Reset Control Word parameters are optional.

The CodeWarrior debugger not only supports NXP devices but also supports non-NXP devices in a JTAG scan chain. Each non-NXP device used in a scan chain is declared as "Generic" and it takes the following three parameters:

- JTAG Instruction Length
- Bypass Command
- Bypass Length

The values for these three parameters are available in the device's data sheet or can be obtained from the manufacturer of the device.

The listing below show an NXP device, 8560, connected with a non-NXP device, PLA, in a JTAG scan chain. From the PLA's data sheet, the JTAG Instruction Length = 5, the Bypass Command = 1, and the Bypass Length = 0x1F.

Figure 95: Sample JTAG initialization file including non-NXP devices

```
8560
Generic 5 1 0x1F
```

8.4 Setting up a remote system to use a JTAG configuration file

This section explains how to configure a remote system to use a JTAG configuration file.

To connect to a JTAG chain, specify these settings in the launch configurations:

1. Create a JTAG initialization file that describes the items on the JTAG chain. For more information on how to create a JTAG initialization file, see JTAG configuration file syntax on page 171 and Using JTAG configuration file to specify multiple linked devices on a JTAG chain on page 173.
2. Open the CodeWarrior project you want to debug.
3. Select **Run > Debug Configurations.**

   The **Debug Configurations** dialog appears with a list of debug configurations that apply to the current application.
4. Expand the **CodeWarrior** tree control.
5. From the expanded list, select the debug configuration for which you want to modify the debugger settings.

   The **Debug** view shows the settings for the selected configuration.
6. Select a remote system from the **Connection** drop-down list.
7. Select a core from the **Target** list.

8. In the **Connection** group, click **Edit.**

   The **Properties for** *<project>* window appears.

9. Click **Edit** next to the **Target** list.

   The **Properties for** *<remote system>* window appears.

10. Click **Edit** next to the **Target type** drop-down list.

    The **Target Types** dialog appears.

11. Click **Import.**

12. The **Import Target Type** dialog appears.

13. Select the JTAG initialization file that describes the items on the JTAG chain from this location:

```
<CWInstallDir>\CW_ARMv7\ARMv7\ARM_Support\Configuration_Files\jtag_chains
```

14. Click **OK.**

    The items on the JTAG chain described in the file appear in the **Target Types** dialog.

15. Click **OK.**

    The selected JTAG configuration file appears on the **Advanced** tab.

16. Click **OK.**

17. Click the **Debugger** tab.

    The **Debugger** page appears.

18. Ensure that the **Stop on startup at** checkbox is selected and `main` is specified in the **User specified** text box.

19. Click **Apply** to save the changes.

    You have successfully configured a debug configuration.