PSoC® Creator™

Datapath Configuration Tool User Guide

Document # 001-96549 Rev *A

Cypress Semiconductor
198 Champion Court
San Jose, CA 95134-1709
http://www.cypress.com

© Cypress Semiconductor Corporation, 2017. The information contained herein is subject to change without notice. Cypress Semiconductor Corporation assumes no responsibility for the use of any circuitry other than circuitry embodied in a Cypress product. Nor does it convey or imply any license under patent or other rights. Cypress products are not warranted nor intended to be used for medical, life support, life-saving, critical control or safety applications, unless pursuant to an express written agreement with Cypress. Furthermore, Cypress does not authorize its products for use as critical components in life-support systems where a malfunction or failure may reasonably be expected to result in significant injury to the user. The inclusion of Cypress products in life-support systems application implies that the manufacturer assumes all risk of such use and in doing so indemnifies Cypress against all charges.

Any Source Code (software and/or firmware) is owned by Cypress Semiconductor Corporation (Cypress) and is protected by and subject to worldwide patent protection (United States and foreign), United States copyright laws and international treaty provisions. Cypress hereby grants to licensee a personal, non-exclusive, non-transferable license to copy, use, modify, create derivative works of, and compile the Cypress Source Code and derivative works for the sole purpose of creating custom software and or firmware in support of licensee product to be used only in conjunction with a Cypress integrated circuit as specified in the applicable agreement. Any reproduction, modification, translation, compilation, or representation of this Source Code except as specified above is prohibited without the express written permission of Cypress.

Disclaimer: CYPRESS MAKES NO WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, WITH REGARD TO THIS MATERIAL, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. Cypress reserves the right to make changes without further notice to the materials described herein. Cypress does not assume any liability arising out of the application or use of any product or circuit described herein. Cypress does not authorize its products for use as critical components in life-support systems where a malfunction or failure may reasonably be expected to result in significant injury to the user. The inclusion of Cypress' product in a life-support systems application implies that the manufacturer assumes all risk of such use and in doing so indemnifies Cypress against all charges.

Use may be limited by and subject to the applicable Cypress software license agreement.

Contents

Introduction **4**
General Functionality 4
Document History 4

Framework **5**
Interface 5
Menus 6

General Tasks **8**
Launching the Datapath Configuration Tool 8
Opening a Verilog File 8
Saving a File 9

Working with Bit Field Parameters **10**
Adding Parameter to Enumerated Bit Fields 10
Adding Parameter to Mask Bit Fields 11
Bit Field Dependencies 11

Working with Configurations **12**
Configuration naming 12
Editing Configuration 12
Copy, Paste Configuration 12
Resetting Configuration 13

Working with Datapath Instances **14**
Creating a New Datapath Instance 14
Deleting the Datapath Instance 14
Setting Initial Register Values 15

Support **16**

Introduction

The Datapath Configuration Tool is used to edit datapath instance configurations in a Verilog implementation of a PSoC component. This appendix provides instructions and information that will help you modify your Verilog files using the tool.

For information about implementing a component using Verilog, refer to the Component Author Guide.

General Functionality

Use the Datapath Configuration Tool to:

- read an existing Verilog file
- modify existing datapath configurations
- create new configurations
- delete existing configurations

You can then save changes back into the same Verilog file without affecting any of the other Verilog implementations. The tool works only with existing Verilog files; it cannot be used to create a new Verilog implementation.

After opening a Verilog file, it is parsed for existing datapath configurations. These could be defined as a parameter, localparam, or directly in the datapath as a named parameter. The tool displays all found configurations in the **Configuration** list drop down at the top of the application.

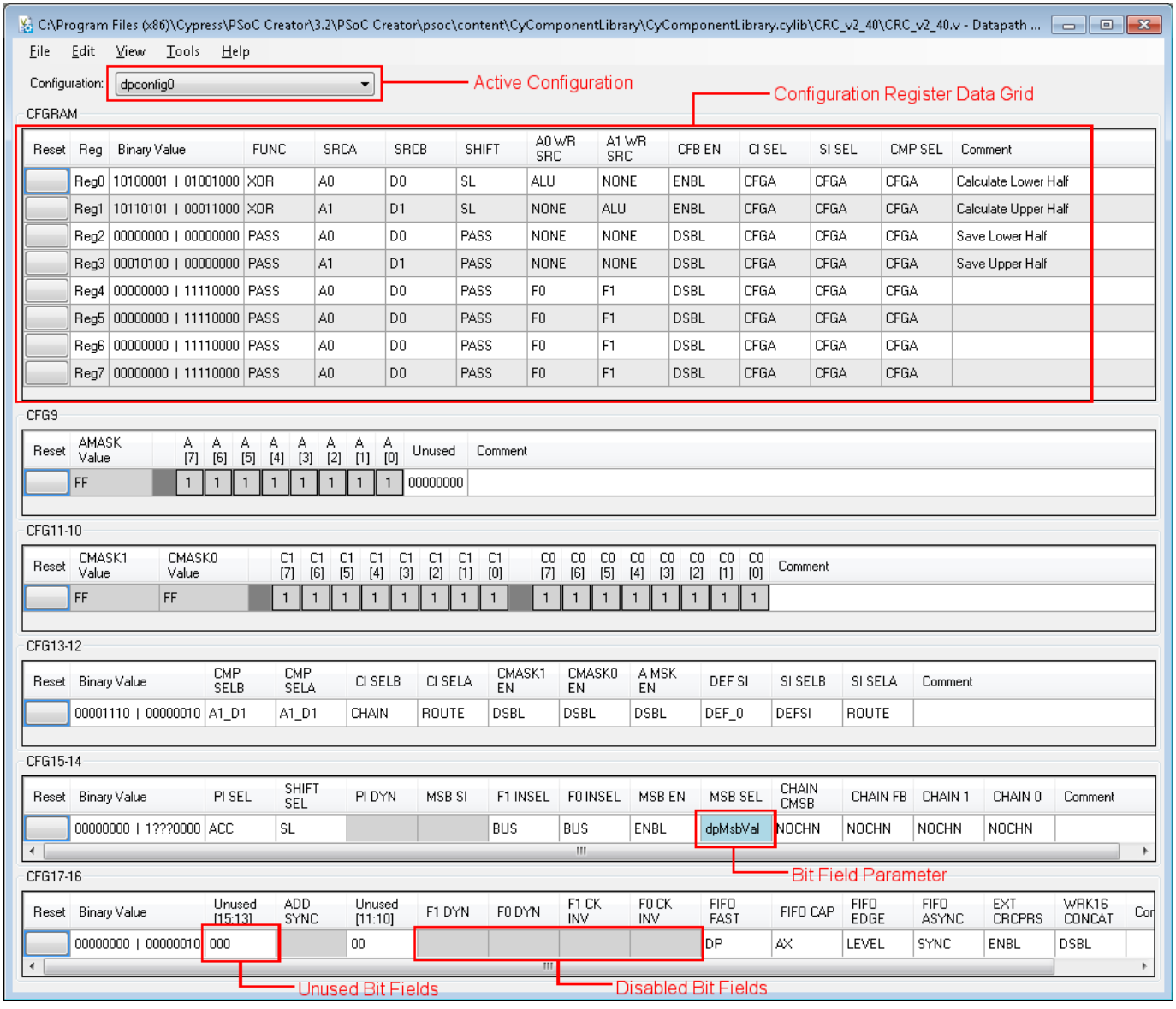
You can edit any bit field, edit comments, copy and paste data between configurations or within a single configuration, and perform whole datapath copy, paste, create or delete operations.

Some operations are applicable only for the datapath (such as deleting a datapath or editing the initial register values). The active datapath is the datapath to which the selected configuration belongs. If the configuration is defined as a parameter, no datapath is active and all datapath options are disabled.

Document History

Document Title: PSoC® Creator™ Datapath Configuration Tool User Guide **Document Number: 001-96549**		
Revision	**Date**	**Description of Change**
**	5/7/15	New document.
*A	11/21/17	Minor updates.

Interface



The main form has the following elements:

- Configuration – Contains all configurations found in the active Verilog file. The current selection is called the “active configuration” or “selected configuration.”

- Configuration Register Data Grid – Contains the Reset register button to restore the whole register to its default configuration, a register binary value field, bit field configuration controls and comment field. The Binary value field is read-only. Bit field controls are typically a drop-down list with predefined values. CFG9, CFG10, CFG11 registers are mask registers and their values can be toggled by clicking on the individual bit buttons.
- Bit Field Parameters – Each bit field can be replaced by a parameter value instead of one of the constant value. When a bit-field is defined with a parameter that parameter name is displayed as the value and is displayed with the blue background color.
- Unused Bit Fields – Unused bit fields are read-only and are always initialized with a 0 value.
- Disabled Bit Fields – Bits that are not used in the active configuration. User can enable them using the context menu of these bits. Bits may be disabled with a particular revision of silicon. These bit fields are indicated as disabled with a gray background and no value in the field. They are always initialized with a 0 value.

Menus

File Menu

Menu Item	Shortcut	Description
Open	[Ctrl] + [o]	Displays a dialog to open an existing file (Filtered for .v Verilog files)
Close		Closes the active file
Recent Files		Provides access to previously opened files
Save	[Ctrl] + [s]	Saves the active document
Save As		Displays a dialog to save the active file to another file
Exit		Exit the PSoC Datapath Configuration Tool

Edit Menu

Menu Item	Shortcut	Description
Copy Datapath	[Ctrl] + [c]	Copies the selected datapath configuration
Paste Datapath	[Ctrl] + [v]	Pastes full datapath configuration from the Clipboard into the active datapath
Paste Dynamic		Pastes the full dynamic configuration information of the datapath configuration from the Clipboard into the active configuration (all 8 registers of the dynamic configuration are pasted)
Reset Datapath		Resets the current datapath configuration to the default bit field values
New Datapath		Add a new datapath instance
Delete Datapath		Delete a datapath instance

View Menu

Menu Item	Shortcut	Description
Initial Register Values		Displays a dialog to configure initial register values for the active datapath

Tools Menu

Menu Item	Shortcut	Description
Options > Splash screen on Startup		Check box to enable or diable the splash screen when the datapath configuration tool starts up.

Help Menu

Menu Item	Shortcut	Description
Documentation	[F1]	Open this documentation.
About		Opens the About dialog, which provides build information

General Tasks

This section covers general Datapath Configuration Tool tasks, including:

- Launching the Datapath Configuration Tool
- Opening a Verilog File
- Saving a File

Launching the Datapath Configuration Tool

To launch the tool, go to the **Start** menu and select **Cypress > PSoC Creator 1.0 > Component Development Kit > Datapath Configuration Tool**.

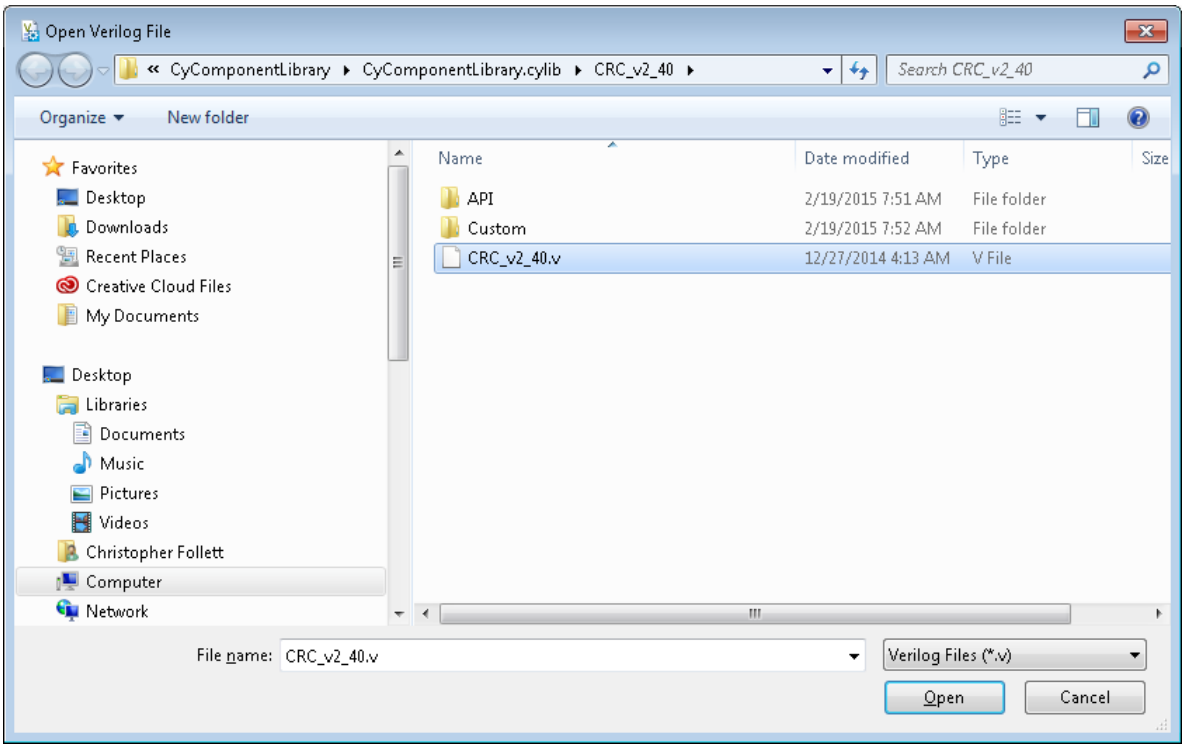
When the application launches, it will not contain any datapath configuration information.

To begin working with the tool you must open an existing Verilog file.

Opening a Verilog File

1. To open open a Verilog file, select **Open…** from the **File** menu.

   The Open Verilog Dialog displays.



Note: You can also open recent files without this dialog using **Recent Files** on the **File** menu or dragging a Verilog file directly onto the main form of the application.

2. Navigate to and select the file to update, and then click **Open**.

   The dialog closes and the tool gathers all appropriate information from the Verilog file and updates the rest of the window.

Saving a File

To save the active file, select **Save** from the **File** menu.

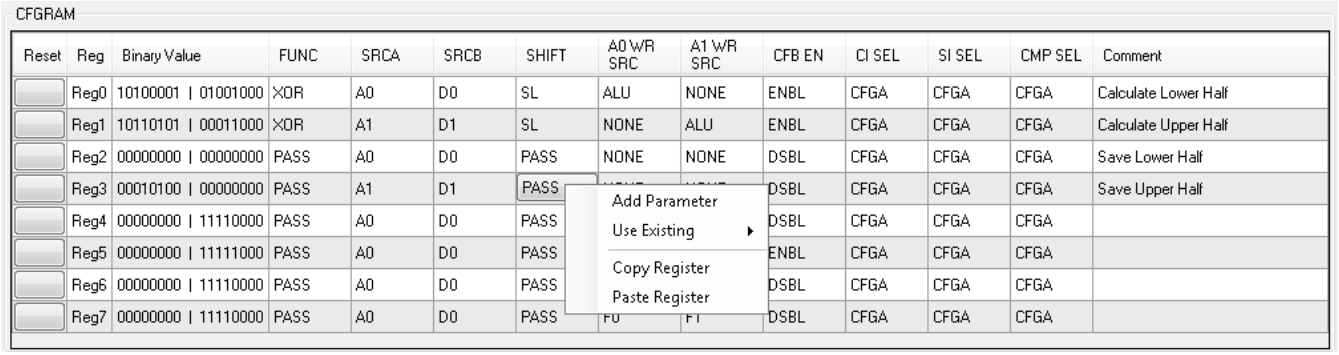
To save a file with another name and/or in a different location, select **Save As** from the **File** menu.

Working with Bit Field Parameters

In addition to a fixed set of predefined constant fields, bit fields support a parameter value.

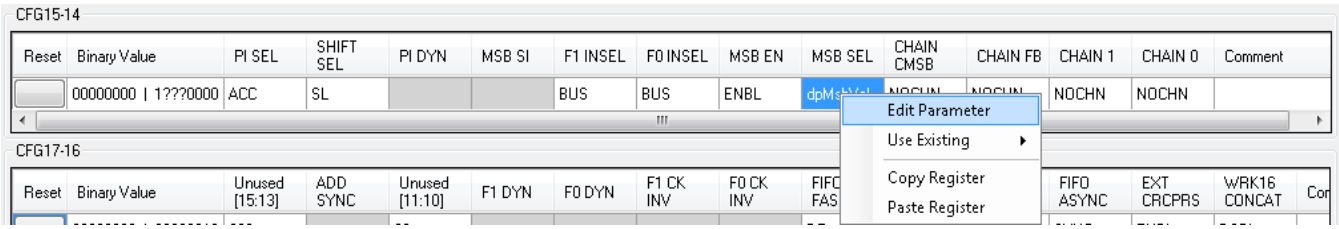
Adding Parameter to Enumerated Bit Fields

Enumerated bit fields contain an **Add Parameter** context menu item.



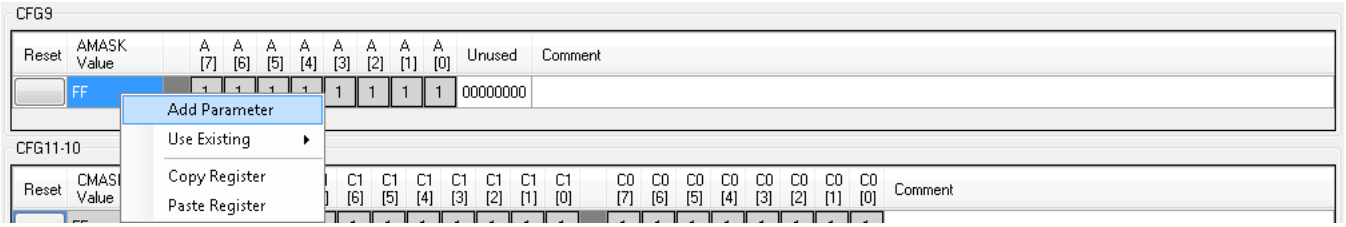
This command allows you to provide any parameter name as an additional choice for the bit field. After adding a parameter, the drop down menu of the bit field will include all the predefined constants, as well as any added parameter value as selection choices. When the parameter value is selected, it will be displayed with blue color indicating that it is not one of the predefined values.

To change a parameter name, right click on the bit field and select **Edit Parameter** from the context menu.



Adding Parameter to Mask Bit Fields

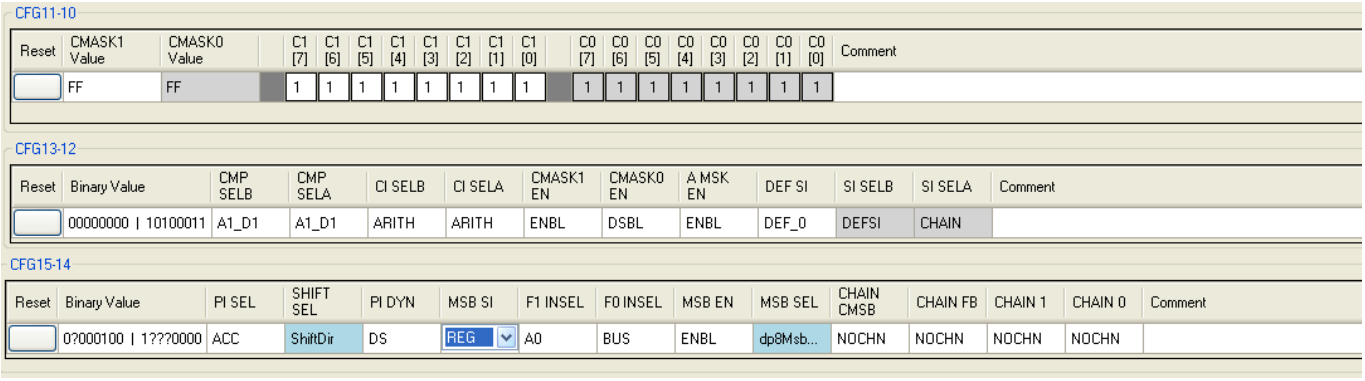
To add a parameter to a mask bit field, select **Add Parameter** from the context menu in the **Value** column, and type the name of parameter. After adding a parameter, separate bits become read-only.



To remove a parameter, select **Remove Parameter** from the context menu in the **Value** column.

Bit Field Dependencies

Datapath configuration has a few bits which have an influence on other bits. Depending on their value, other bit fields may become unused. To make it visual for user, some bit fields became disabled depending on the value of other fields.



On the screenshot above bits CMASK0, SI SELB, and SI SELA are disabled. The reason is that CMASK0 EN bit is set to DSBL and MSB SI bit is set to REG.

Working with Configurations

There are two methodologies that can be used to configure the elements of a datapath instance.

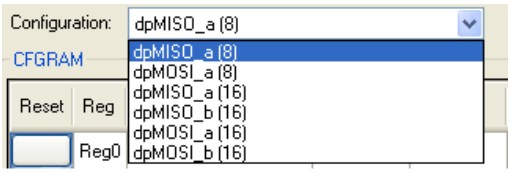
The first methodology directly places the configuration of the datapath in the datapath instance. For multi-byte datapath instances, a separate set of values is used for each byte of the overall instance. This is a common methodology used when a particular configuration is only used once in a Verilog file.

The second methodology assigns the configuration for a datapath to a parameter. That parameter is then used by datapath instances. This methodology is convenient when the same configuration is used by multiple instances to eliminate duplication and simplify maintenance of the component. This methodology is used by the Counter library component. Both methodologies can be used within the same Verilog file.

Both methodologies are supported by PSoC Datapath Configuration Tool. All configurations are added to the **Configuration** drop down list.

Configuration naming

For the datapath instance case the label is the datapath instance name. For the dp8, dp16, dp24 and dp32 datapath instances the configuration labels are suffixed with "_a", "_b", "_c" or "_d" depending on which byte of the instance the configuration is for. Also the datapath size (8, 16, 24, 32) is added at the end of the configuration name in the drop down. For the parameter instances case the label is the parameter name.



Editing Configuration

To make the configuration active, select it in the **Configuration** drop down list. Bit fields of each register can be changed by choosing one of the predefined values in the cell drop down list or adding a parameter. All changes made to the active configuration are stored automatically to object storage in the application, so the user can switch between different configurations without losing data. The data is not saved to the verilog file until the main menu operation "Save" or "Save As" is completed.

Copy, Paste Configuration

To copy the active configuration data to the Clipboard, select **Copy** from the **Edit** menu.

To paste the configuration data from the Clipboard, select **Paste** from the **Edit** menu. This command will replace all bit fields and comments of the active configuration with values from the Clipboard.

To paste only the dynamic registers data, select **Paste Dynamic** from the **Edit** menu.

Note: If you want to paste only the dynamic registers data, to copy it initially you should use the **Copy** command as well.

To copy a separate register, select **Copy Register** from the context menu of any bit field that belongs to that register.

To paste a separate register, select **Paste Register** from the context menu of any bit field that belongs to that register.

Resetting Configuration

To reset the current datapath configuration to the default bit field values, use **Reset Datapath** on the **Edit** menu.

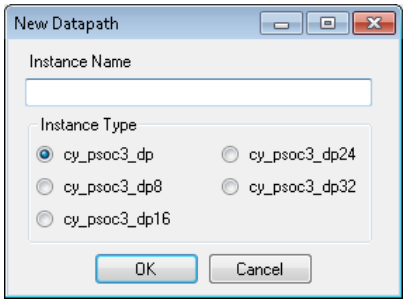
To reset a particular register of the datapath configuration to its default values, press the button of the **Reset** column of data grid in that row on the main form.

The default value for all of the bit fields, except the mask fields of CFG9-11, is **0**. For the mask bit fields of CFG9-11 the default is **1**. The default comment value for dynamic registers is **Idle** and for all other registers is an empty string.

Working with Datapath Instances

Creating a New Datapath Instance

To create a new datapath, select **New Datapath** from the **Edit** menu. The dialog **New Datapath** will appear.



Type the name of the new datapath in the **Instance Name** text box, select datapath type, and press **OK**.

A new datapath will be inserted at the end of the current verilog file before the **endmodule**.

New datapath configurations would be initialized with all of the default bit-field values, all input signals would be connected to 1'b0 and all output signals would be unconnected.

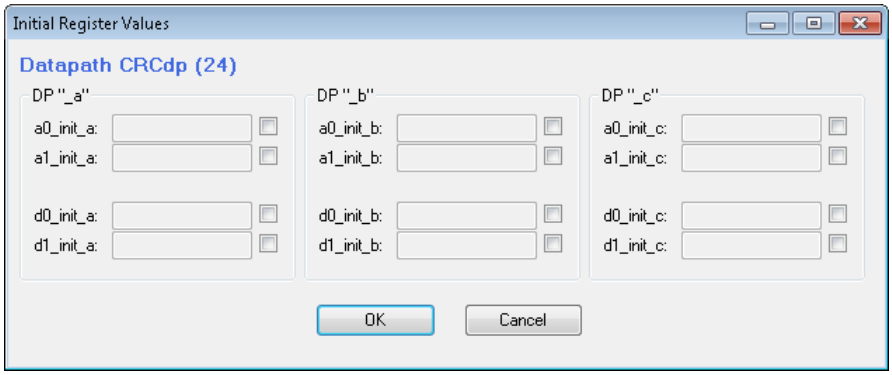
Deleting the Datapath Instance

To delete the active datapath select **Delete Datapath** from the **Edit** menu.

If there are no active datapaths, this option is disabled in the menu. If the datapath has multiple configurations (dp16, dp24, or dp32), all of them will be deleted.

Setting Initial Register Values

To set the initial register values for the active datapath, use **Initial Register Values** on the **View** menu. The dialog **Initial Register Values** will appear.



Check the boxes near the entries that will be defined and enter either an 8-bit hexadecimal value (8'h?) or a parameter value in the edit field. Then press OK to submit changes. Initial values are passed to a datapath as named parameters, so any values not defined will not be passed to the datapath and will be set internally to their default values.

If there are no active datapaths, this option is disabled in the menu.

Support

Visit http://www.cypress.com/support for online technical support. The resources include:

- Training Seminars
- Discussion Forums
- Application Notes
- Developer Community
- Knowledge Base
- Technical Support.

You can also view and participate in discussion threads about a wide variety of PSoC device topics at http://www.cypress.com/support/.

Before using Cypress support services, confirm the version of PSoC Creator installed on your system. Select **Help > About** to determine the tool version.

Upgrades and version enhancements for PSoC Creator are provided by Cypress free of charge. The upgrades are available through the Cypress Update Manager. The Cypress Update Manager searches for updates to all Cypress products.