cadence™

Pcell IDE – Quick Start Guide

Product Version ICADVM20.1
October 2020

© 2020 Cadence Design Systems, Inc. All rights reserved worldwide.

Printed in the United States of America.

Cadence Design Systems, Inc. (Cadence), 2655 Seely Ave., San Jose, CA 95134, USA.

Trademarks: Trademarks and service marks of Cadence Design Systems, Inc. contained in this document are attributed to Cadence with the appropriate symbol. For queries regarding Cadence's trademarks, contact the corporate legal department at the address shown above or call 800.862.4522.

All other trademarks are the property of their respective holders.

Restricted Permission: This publication is protected by copyright law and international treaties and contains trade secrets and proprietary information owned by Cadence. Unauthorized reproduction or distribution of this publication, or any portion of it, may result in civil and criminal penalties. Except as specified in this permission statement, this publication may not be copied, reproduced, modified, published, uploaded, posted, transmitted, or distributed in any way, without prior written permission from Cadence. Unless otherwise agreed to by Cadence in writing, this statement grants Cadence customers permission to print one (1) hard copy of this publication subject to the following conditions:

1. The publication may be used only in accordance with a written agreement between Cadence and its customer.
2. The publication may not be modified in any way.
3. Any authorized copy of the publication or portion thereof must include all original copyright, trademark, and other proprietary notices and this permission statement.
4. The information contained in this document cannot be used in the development of like products or software, whether for internal or external use, and shall not be used for the benefit of any other party, whether or not for consideration.

Disclaimer: Information in this publication is subject to change without notice and does not represent a commitment on the part of Cadence. Except as may be explicitly set forth in such agreement, Cadence does not make, and expressly disclaims, any representations or warranties as to the completeness, accuracy or usefulness of the information contained in this document. Cadence does not warrant that use of such information will not infringe any third party rights, nor does Cadence assume any liability for damages or costs of any kind that may result from use of such information.

Restricted Rights: Use, duplication, or disclosure by the Government is subject to restrictions as set forth in FAR52.227-14 and DFAR252.227-7013 et seq. or its successor.

Contents

Virtuoso Quick Start: Debugging Pcells 5
Pcell IDE — Debug Commands 6
Debugging the Super Master 7
Debugging an Instance 9
Debugging Hierarchical Pcells 10
Debugging an Abutment 11
Cross-highlight between Pcell IDE and SKILL IDE 12

© 2020 All Rights Reserved.

© 2020 All Rights Reserved.

Virtuoso Quick Start: Debugging Pcells

Purpose	This document provides you a quick overview on how SKILL Pcells use the following commands. ■ Debug Super Master ■ Debug Instances ■ Debug Abutment ■ Debug Hierarchy Pcell Mode For more information, see Debugging SKILL Pcells.
Prerequisites	■ The Pcell source file must be available. ■ The Pcell library must contain a `libInit.il` file that loads all the needed files. ■ Pcell IDE should be launched by selecting *Launch — Pcell IDE* from either of the VLS, VSE, and VSE-Symbol editor window.

© 2020 All Rights Reserved.

Pcell IDE — Debug Commands

This section lists various commands for debugging SKILL Pcells.

- Debugging the Super Master
- Debugging an Instance
- Debugging Hierarchical Pcells
- Debugging an Abutment
- Cross-highlight between Pcell IDE and SKILL IDE

© 2020 All Rights Reserved.

Debugging the Super Master

To debug the Pcell super master, you need to perform the following steps:

1. Select the Pcell source filename from the *Source File* drop-down list of the Pcell IDE assistant.
2. Debug the Pcell super master by using one of the methods provided in the table below:
   - **Direct Debug Pcells**: Click *Run* once it gets enabled on the Pcell IDE toolbar.
   - **Indirect Debug Pcells**: Type the function call either in CIW or SKILL IDE if the *Run* button is not enabled on the Pcell IDE toolbar after you load the Pcell source file.

Table 1-1 Debugging Methods

Debugging Methods	Steps
Direct Debug Pcells	**a.** From the Pcell IDE toolbar, click *Run* to start debugging. The debug flow stops at the initial breakpoint.
	b. From the SKILL IDE window, debug Pcell source file using SKILL IDE debug navigation commands.
	c. During the Pcell debugging process, you can see the graphic being updated simultaneously in the editor window.
Indirect Debug Pcells	**a.** Type the function call either in CIW or SKILL IDE if the *Run* button is not enabled on the Pcell IDE toolbar after you load the Pcell source file. This function contains the call to `pcDefinePCell` and enables the *Run* button on the Pcell IDE toolbar.
	b. The rest of the debugging steps are the same as the direct debugging Pcells method.

© 2020 All Rights Reserved.

3. After the evaluation is successfully completed, the Pcell default parameters are displayed in the Pcell IDE assistant. However, these parameters are not editable, as shown below.

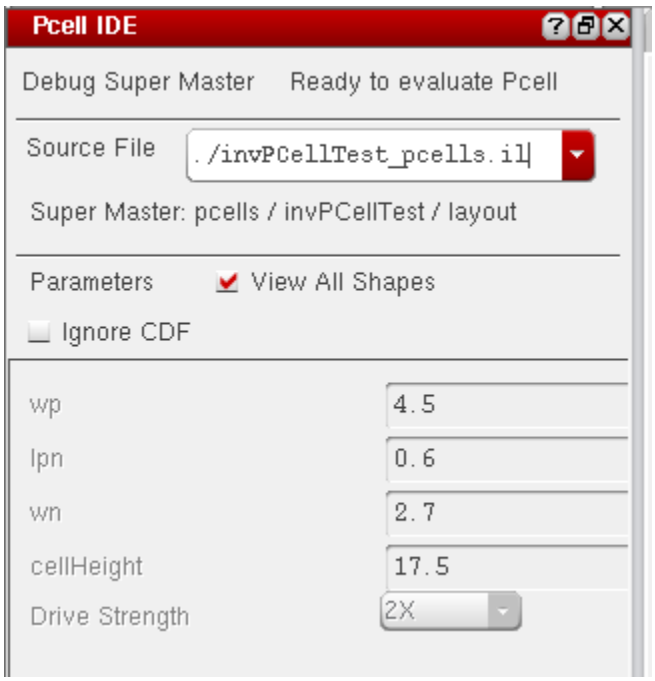

Figure 1-1 The Debug Super Master

© 2020
All Rights Reserved.

Debugging an Instance

To debug a Pcell instance, you need to perform the following steps:

1. Select a Pcell instance at any level of the hierarchy design.
2. Select the Pcell source file path from the *Source File* drop-down list.

   Note: If the Pcell source file does not match with the selected Pcell instance, the *Run* button on the Pcell IDE toolbar is not enabled and the “SuperMaster not loaded" message is displayed. To start debugging the Pcell instance, you need to either select the corresponding Pcell instance, or select the correct Pcell source file to enable the *Run* button on the Pcell IDE toolbar.

   There might be instances where the *Run* button is not enabled on the Pcell IDE toolbar after you load the Pcell source file. In such cases, you need to type the function call either in CIW or SKILL IDE to enable the *Run* button on the Pcell IDE toolbar.

   For more information, refer to Table 1-1 on page 7.
3. Pcell parameters are displayed in the Pcell IDE assistant. You can modify the value of any parameter. In addition, you can copy another Pcell instance parameter using the *Copy* button.
4. On the SKILL IDE window, click *Step* to step through the Pcell source code. The graphic is incrementally updated and the Pcell instance is displayed in the editor window, as shown below.

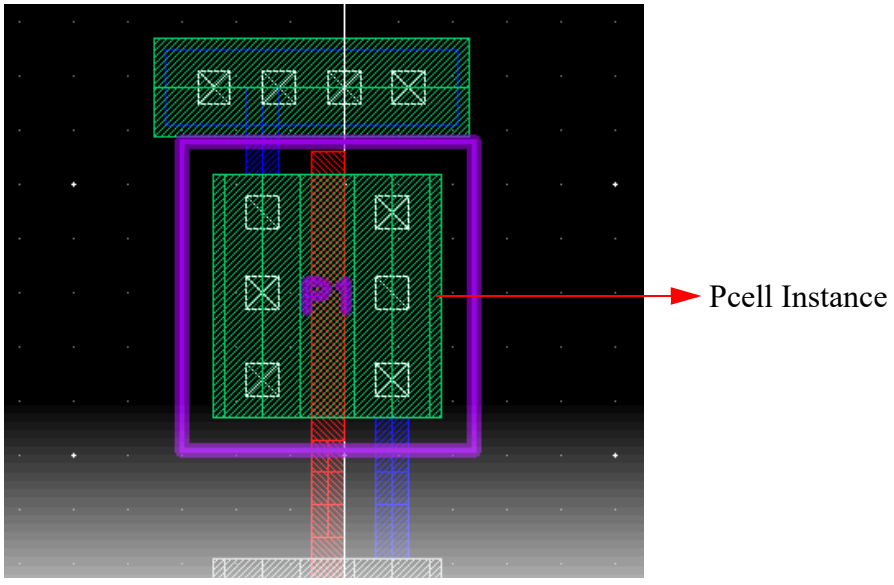

Figure 1-2 Pcell Instance in the Editor Window

© 2020
All Rights Reserved.

Debugging Hierarchical Pcells

To debug Pcells at any of the displayed hierarchical levels, you need to perform the following steps:

1. Turn on the Debug Hierarchy Pcell mode by clicking the *Debug Hier Mode* button from the Pcell IDE toolbar, any time while debugging Pcells.
2. When the `dbCreateParamInst` SKILL statement is met, click *Step* or *Continue* from the SKILL IDE window. The Load Pcell Source File form is displayed in the Pcell IDE assistant for you to select the Pcell source file.

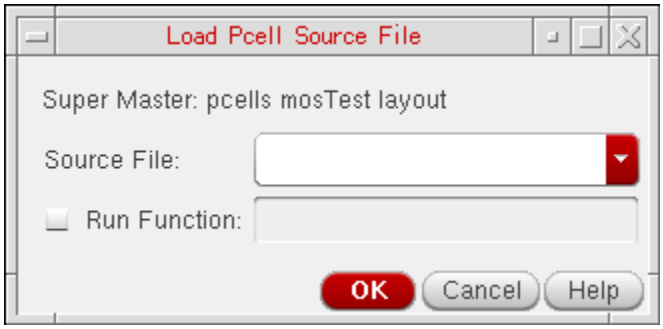

Figure 1-3 Load Pcell Source File Window

Note: To skip debugging the next level, click the *Next* button from the Debug toolbar of the SKILL IDE window.

3. The Load Pcell Source File form shows the debugged super master's library name, cell name, and view name.
4. You need to select the Pcell source filename in *Source File* drop-down list.

   Note: If it is an indirectly generated Pcell, click *Run* and enter the function call.
5. Click *OK* to debug the next-level Pcell.
6. Click *Cancel* to skip the next-level Pcell debugging.

© 2020
All Rights Reserved.

Debugging an Abutment

To debug a Pcell abutment, you need to perform the following steps:

1. Click the *Debug Abutment* button from the Pcell IDE toolbar.
2. Select the abutment action and set the desired abutment events to be debugged.
3. In the Pcell IDE assistant, fill in the abutment instance and pin information.

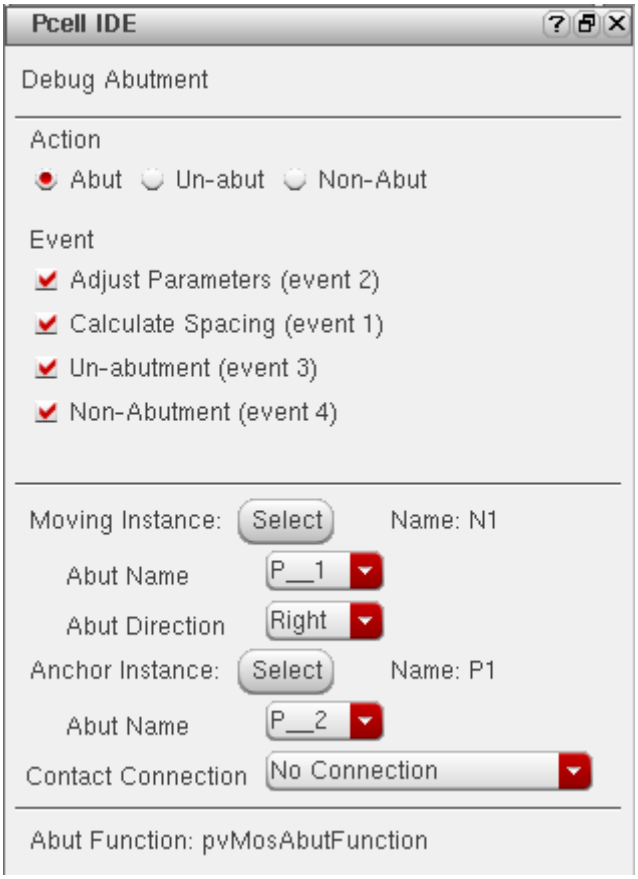

Figure 1-4 Debug an Abutment

4. Click the *Run* button to start the abutment simulation.

Note: To debug the Pcell evaluation during debug abutment, you can turn on the Debug Hierarchy Pcell mode by clicking the *Debug Hier Mode* button when any one of these functions — `dbReplaceProp`, `dbReplacePropList`, or `dbReplaceInstParamList` is called.

© 2020
All Rights Reserved.

Cross-highlight between Pcell IDE and SKILL IDE

You can cross-highlight between the graphic objects and the source code.

To do this, you can do either of the following:

- If you select an object from the editor window, the corresponding SKILL statement is highlighted in the SKILL IDE window.
- On the other hand, if you select a SKILL statement from the SKILL IDE window, the corresponding object(s) is highlighted in the editor window.

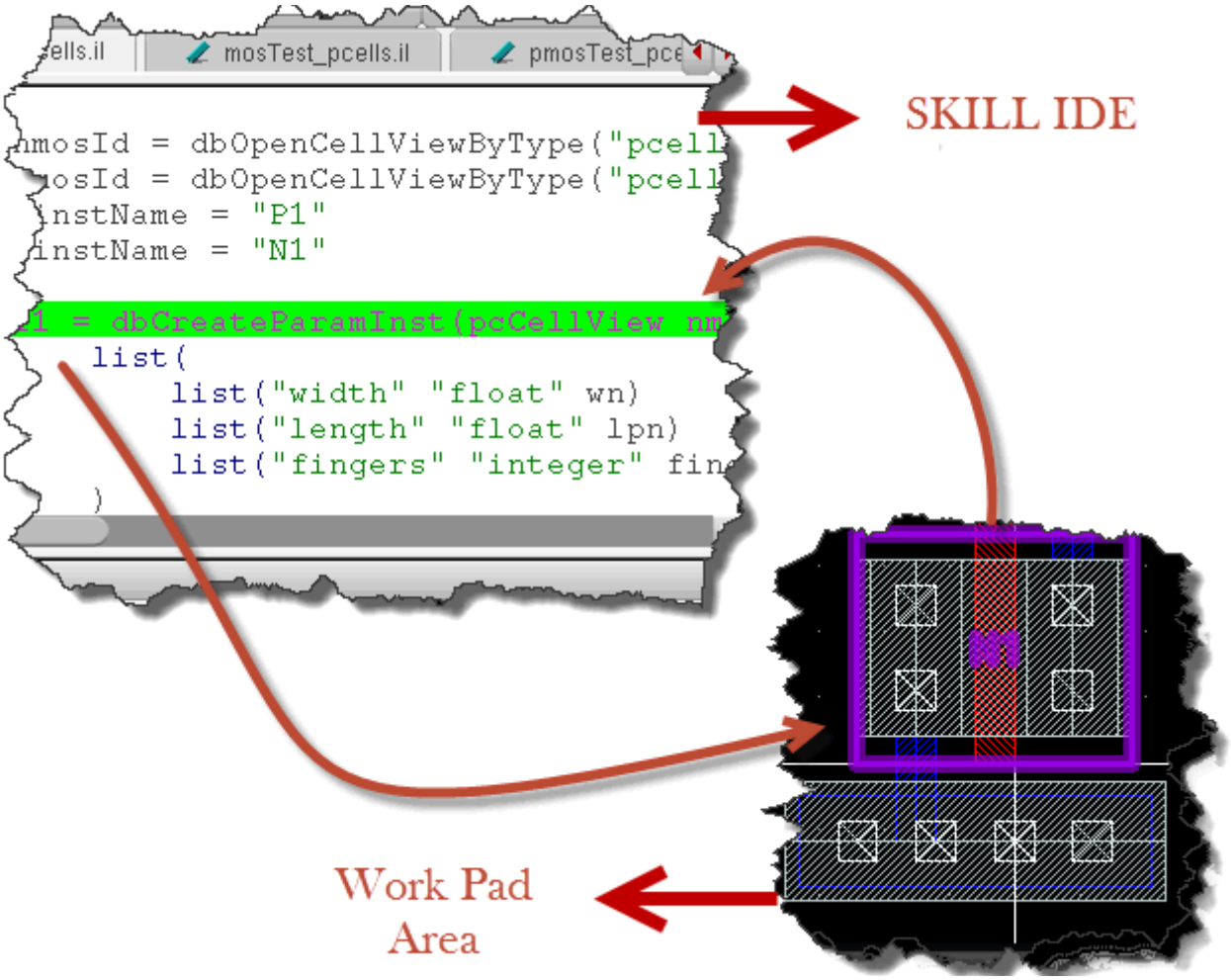

Figure 1-5 Cross-highlight Pcell IDE and SKILL IDE

© 2020
All Rights Reserved.