

# **Fault Simulation Task Assistant**

**Product Version ICADVM20.1**

**October 2020**

© 2020 Cadence Design Systems, Inc. All rights reserved.  
Printed in the United States of America.

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

Open SystemC, Open SystemC Initiative, OSCI, SystemC, and SystemC Initiative are trademarks or registered trademarks of Open SystemC Initiative, Inc. in the United States and other countries and are used with permission.

**Trademarks:** Trademarks and service marks of Cadence Design Systems, Inc. (Cadence) 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

---

1	4
Fault Simulation Task Assistant	4
How to inject faults in a design?	4
How to preview the fault list for simulation?	5
How does fault sampling affect the fault list?	5
How to implement automatic fault dropping during simulation run?	6
Is it possible to merge results from multiple fault simulations?	6

---

# Fault Simulation Task Assistant

---

Fault simulations in ADE Assembler provide the capability to run defect-oriented tests on analog IC designs. These tests allow you to evaluate the ability to eliminate a die with manufacturing defects and resulting test escapes that cause field failures. It can also be used to optimize wafer test, reducing the number of tests required to achieve the target defect coverage by eliminating over-testing and potentially reducing the number of tests.

The following topics describe the important tasks you can perform for fault simulation:

- [How to inject faults in a design?](#)
- [How to preview the fault list for simulation?](#)
- [How does fault sampling affect the fault list?](#)
- [How to implement automatic fault dropping during simulation run?](#)
- [Is it possible to merge results from multiple fault simulations?](#)

**Related Content:** [Running Fault Simulations](#) 

## How to inject faults in a design?


The Fault Setup Assistant provides the following methods to specify the faults to be injected:

- Using Fault Rules
- Creating Individual Faults
- Using Fault Files
- Importing Fault Rules from Other Cellviews

**Related Topic:** [Injecting Faults](#) 

## How to preview the fault list for simulation?

Before running a fault simulation, you can preview the fault list :

1. Select a fault group in the *Fault Group to Run* drop-down list on top of the Fault Setup assistant.
2. Click *Preview all faults for fault group selected to run* () on the Faults Setup assistant.

ADE Assembler runs Spectre info analysis to identify the fault list. If available, it also applies the sampling options to arrive at the final list of fault samples in the *Faults* table. It then displays the list in the Fault Group Preview window.

**Related Topic:** [Previewing Faults](#) 

## How does fault sampling affect the fault list?

When the fault universe consists of a large number of faults, it is practically impossible to run simulations for all the faults. Sampling helps in selecting a subset of fewer faults that can represent the whole fault universe. This helps in estimating the overall coverage with very less simulations.

In ADE Assembler, you can use the following sampling methods supported by Spectre:

- random
- randomuniform
- randomweighted
- weightsorted

**Related Help Topic**

[Sampling Options](#) 

## How to implement automatic fault dropping during simulation run?

You can create a run plan using the fault setup to run multiple simulations, one for each test, in a sequence where each subsequent run in the plan uses a reduced list of faults that were not detected in the previous run. This way of dropping faults that are detected by the previous run helps in focusing on fault coverage and also saves time because the simulation is not run for the faults that are already detected.

**Related Topic:** [Automatic Fault Dropping Using Run Plan](#) 

## Is it possible to merge results from multiple fault simulations?

You can merge the results of multiple fault simulation histories to view the consolidated results in one history.

This feature saves the time required to rerun simulations in the following scenarios:

- After running a fault simulation, you modify the fault rules or the test setup, or make any other change in the active setup, and run simulation. You can merge the results saved before and after making these changes and view a consolidated report.
- You want to check the fault detection rate by dropping faults in subsequent runs, and then want to merge results of all runs. For more details, refer to Checking Fault Detection Rate.

**Related Topic:** [Merging Results from Multiple Fault Simulations](#) 