

MBP 2017

Modeling

Notices

© Keysight Technologies Incorporated, 2002-2017

1400 Fountaingrove Pkwy., Santa Rosa, CA 95403-1738, United States

All rights reserved.

No part of this documentation may be reproduced in any form or by any means (including electronic storage and retrieval or translation into a foreign language) without prior agreement and written consent from Keysight Technologies, Inc. as governed by United States and international copyright laws.

Restricted Rights Legend

If software is for use in the performance of a U.S. Government prime contract or subcontract, Software is delivered and licensed as "Commercial computer software" as defined in DFAR 252.227-7014 (June 1995), or as a "commercial item" as defined in FAR 2.101(a) or as "Restricted computer software" as defined in FAR 52.227-19 (June 1987) or any equivalent agency regulation or contract clause.

Use, duplication or disclosure of Software is subject to Keysight Technologies' standard commercial license terms, and non-DOD Departments and Agencies of the U.S. Government will receive no greater than Restricted Rights as defined in FAR 52.227-19(c)(1-2) (June 1987). U.S. Government users will receive no greater than Limited Rights as defined in FAR 52.227-14 (June 1987) or DFAR 252.227-7015 (b)(2) (November 1995), as applicable in any technical data.

Portions of this software are licensed by third parties including open source terms and conditions. For detail information on third party licenses, see [Notice](#).

Contents

Task Tree	8
Checkpoints Customization	8
Checkpoint Customizer Overview	8
Major Parts	9
Concept of Task Tree	12
Task Tree example	13
Sample checkpoint	14
GUI and Main Modules of Task Tree	14
Main window with Task Tree launched	15
Upper Tool Bar	15
Lower Tool Bar	16
Content Panel	17
Real-time Configuration	18
Task Customization	18
Task Customizer Overview	18
Major Modules	19
Editing Utilities	20
Sample Task	23
Task Tree Overview	24
Managing Models	25
Binning Model	25
Binning Generation	25
Binning Window with point model loaded	26
Bin Select Window	27
Pseudo Bin Window	27
Bin boundary expansion	28
Parameter Window for Bin Model	28
Subcircuit bin model selection	29
Corner Model	29
Set skew parameters for FS and SF corner	32
Output corner models	33
Generate Point Model	34
Generate point models	34
Model File Manager	34
Model file manager	35
Tweaking Models	36
Binning Model Tweak	36
General	36
Binning model tweak	37
Model Tweaking	42
General	42
Global model tweaking	43

BSIM3v3 Modeling	46
Binning Method	46
An exemplar binning strategy that breaks devices width-length space into 12 subregions	47
Non-binnable I-V model parameters for BSIM3v3	47
BSIM3v3 DC Modeling	48
Do Measurements	48
Load Measurement Data	48
Check Measured Data	50
Set Process Parameter	50
Use the Task Tree to Extract Initial Model	51
Optimization	53
BSIM3v3 Overview	54
Device Requirements	55
An example of device geometry for DC parameter extraction	56
Extraction and Optimization of BSIM3v3 Model	56
Extraction Routine	56
Measurement Conditions	60
Diode Modeling	62
Diode C-V Modeling	62
Measuring C-V Data	62
AC Diode Model Extraction	63
Diode DC Modeling	65
Measuring I-V Data	65
Diode DC Model Extraction	66
Gummel-Poon BJT Modeling	70
AC Measurement of BJT	70
BJT Junction Capacitance Data Set	70
Measuring C-V Data	70
DC Measurement of BJT	72
Standard BJT Data Sets	72
Measuring I-V Data	74
Introduction to Gummel-Poon model	77
Equivalent Gummel-Poon DC Model	78
Gummel-Poon AC Model Extraction	78
Extract B-E capacitance	79
Extract B-C capacitance	79
Extract B-S capacitance	79
Gummel-Poon DC Model Extraction	79
Select Model Type	80
Load Measured Data	80
Extract Model Parameters	80
Further Extraction and Parameters Optimization	81
Export Model File	81
HiSIM HV Modeling	82
HiSIM HV Overview	82

Extraction of HiSIM HV Model	82
Do Measurements	82
Load Measurement Data	83
Use the Task Tree to Extract Initial Model	83
Optimization of HiSIM HV Model	84
Use the Optimization	84
Export Model File	85
Macro Modeling	86
Macro Model Structure	86
Node Assignment	86
Macro Models Optimization	87
Optimization window of exemplar macro model	87
The Equation Viewer for exemplar macro model	88
Macro Models Simulation	88
Set Model Type	88
Run Simulation with Internal Engine	89
Run Simulation with Calling Remote Hspice	89
MOSRA Modeling	91
Data Format	91
Pre-defined IMV pages for stress data	92
MOSRA simulation	92
Set Model Type	92
Run Simulation of MOSRA Model	93
PSP Modeling	95
PSP Overview	95
Extraction of PSP Model	96
Perform Measurements	96
Load Measurement Data	96
Set Process Parameter	97
Use the Task Tree to Extract Initial Model	98
Optimization of PSP Model	99
Use the Optimization	99
Use the PSP scaling tweaking	99
Export Model File	99
RF Modeling	100
Customization by Script	100
Customize Target	100
Plot Model Parameters versus Instance Parameter	104
De-embedding	106
Load data for de-embedding	106
Choose de-embedding method	107
Open De-embedding Method	107
Short De-embedding Method	107
Open Short De-embedding Method	108
Pad Open Short De-embedding Method [1]	108
Open Short Thru De-embedding Method [2]	110

Improved Open Short Thru De-embedding Method [3]	111
Cascaded Line Method	112
MBP Package and Solution	113
Standard Inductor Package Introduction	113
3-port Inductor	118
Parameter Conversion	119
RF Parameters Supported in MBP	119
Parameter Conversion in MBP	121
References	121
RF Data and Simulation	122
Data Format	122
S-parameter Simulation	123
Noise Simulation	124
External Simulator	125
Corner Modeling	126
Corner Modeling	126
Step 1: Open the corner tweaking window	126
Step 2: Prepare corner model	126
Step 3: Setup Corner Spec	127
Step 4: Tweak and Optimize	129

- Task Tree
- Managing Models
- Tweaking Models
- BSIM3v3 Modeling
- Diode Modeling
- Gummel-Poon BJT Modeling
- HiSIM HV Modeling
- Macro Modeling
- MOSRA Modeling
- PSP Modeling
- RF Modeling
- Corner Modeling

Task Tree

In MBP, a unique structure realizes both automated and user-customized modeling procedures: Task Tree. It is described how to utilize and configure it in this chapter. This section includes the following topics:




- Checkpoints Customization
- Concept of Task Tree
- GUI and Main Modules of Task Tree
- Task Customization
- Task Tree Overview

Checkpoints Customization

By using checkpoint, you can realize more flexible and robust model extraction procedures in Task Tree such as conditional loop, jump and etc.

Checkpoint Customizer Overview

To pop up CheckPoint Customizer window, do one of the following actions:

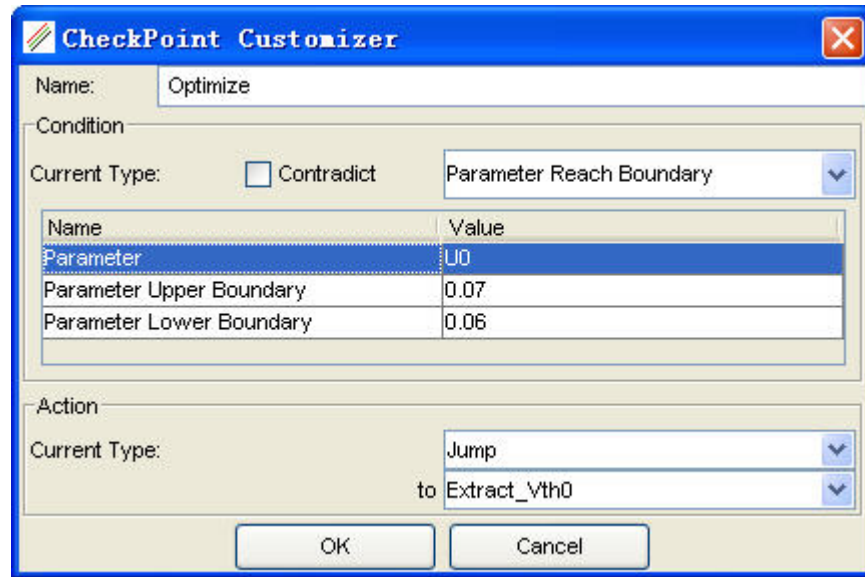
- Click a checkpoint, then click Edit icon  in the lower tool bar.
- Double-click target checkpoint.
- Click Add CheckPoint icon  or Append CheckPoint icon  in the lower tool bar.

As shown in [Figure: Main view of Checkpoint Customizer](#), the main window consists of the following parts:

- Checkpoint Name
- Condition panel
- Action panel

Set condition statements in the Condition panel and determines action to be taken in the Action panel.

Main view of Checkpoint Customizer



Major Parts

Checkpoint Name

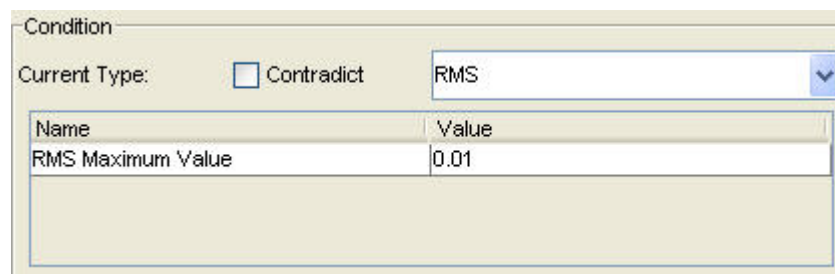
Although the name is not necessary for a checkpoint, You can assign one for easy understanding.

Condition panel

To execute checkpoint, MBP first performs a logic judgment on a specific expression for true or false. In the condition panel, MBP provides four kinds of expressions for evaluation, they are: RMS

To choose RMS as expression, assign a value as RMS Maximum Value, which is "0.01" in following figure. If the RMS value of the preceding task is larger than the RMS Maximum Value, it returns true; otherwise it returns false.

Expression of RMS



NOTE

The RMS value of the preceding task is calculated from the selected region or page (if region is not available).

Parameter Reach Boundary

This expression does not inherit any parameter from the preceding task. Specify one parameter and the corresponding upper/lower boundary. If the parameter value is not within upper and lower boundary, it returns true; otherwise it returns false. Take the settings in following figure as an example, if U0 value is larger than 0.06 or lower than 0.05, the evaluation results are true.

Expression of Parameter Reach Boundary

The screenshot shows a dialog box titled 'Condition'. It has a 'Current Type:' section with a 'Contradict' checkbox (unchecked) and a dropdown menu set to 'Parameter Reach Boundary'. Below this is a table with two columns: 'Name' and 'Value'.

Name	Value
Parameter	U0
Parameter Upper Boundary	0.06
Parameter Lower Boundary	0.05

Loop Number Reach

This expression is not merely decision statement, it also involves actions. MBP lists all tasks of the current Task Tree in the Loop through window, as shown in [Figure: Expression of Loop Number Reach](#). Sort out which tasks are to run in the loop. Only when cycle time reaches the Loop Number, loop stops and jumps to the action in this checkpoint.

To pick up the tasks for loop, hold Ctrl key and left click on the task in the Loop through window. The selected is highlighted in blue. These highlighted tasks could be either continuous or separated. Be careful not to fall into the dead loop in the Task Tree.

Expression of Loop Number Reach

The screenshot shows a dialog box titled 'Condition'. It has a 'Current Type:' section with a 'Contradict' checkbox (unchecked) and a dropdown menu set to 'Loop Number Reach'. Below this is a table with two columns: 'Name' and 'Value'.

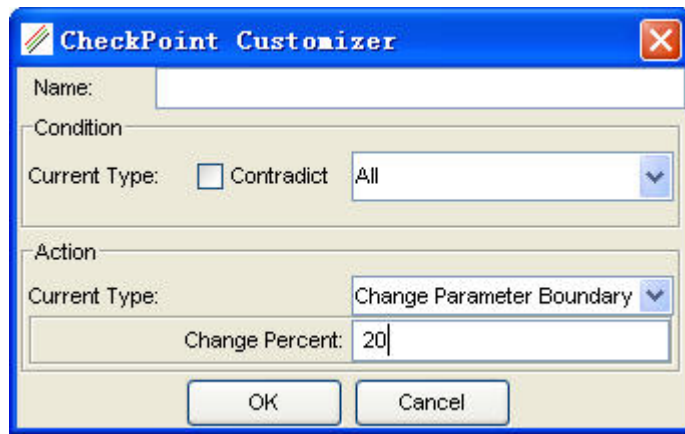
Name	Value
Loop Number	2

Below the table is a section labeled 'Loop through:' with a list of tasks: 'Extract_Vth0', 'Extract_K1_k2', and 'Extract_Lint_Rdsw'. The 'Extract_K1_k2' and 'Extract_Lint_Rdsw' tasks are highlighted in blue.

All

No judgment is done. The flow jumps to the action of current checkpoint directly at all events. The window is shown in following figure.

Expression of All



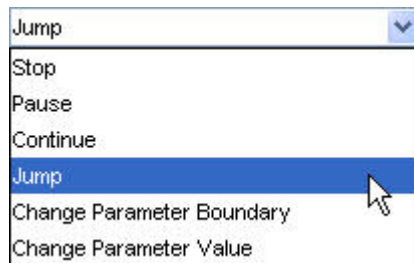
NOTE

Above description assumes the option "Contradict" is not checked. If this option is checked, evaluation returns the reverse value, that the original **true** turns **false**, and vice versa.

Action Panel

You can decide which action to be taken in the action panel. Six actions are currently available to choose, shown in following figure.

Pre-defined actions for the checkpoint



Stop

To stop running the current Task Tree. It is often used when fatal error occurs.

Pause

To pause the current Task Tree. It is often used to keep a close watch on the flow.

Continue

To continue the flow.

Jump

To redirect the desired task within the current Task Tree. See following figure.

Action of the Jump

Action	
Current Type:	Jump
to	Extract_Lint_Rdsw

Change Parameter Boundary

To reset parameter boundary. It is often used to get better fitting effects during parameter optimization. The parameters involved are from selected parameters of the preceding task. The boundary is enlarged by percentage, as shown in following figure.

Action of the Change Parameter Boundary

Action	
Current Type:	Change Parameter Boundary
Change Percent:	30.0

Change Parameter Value

To discard the result derived from the previous tasks and assign a value to the parameter arbitrarily. Such action is used to exclude un-physical parameters. See following figure.

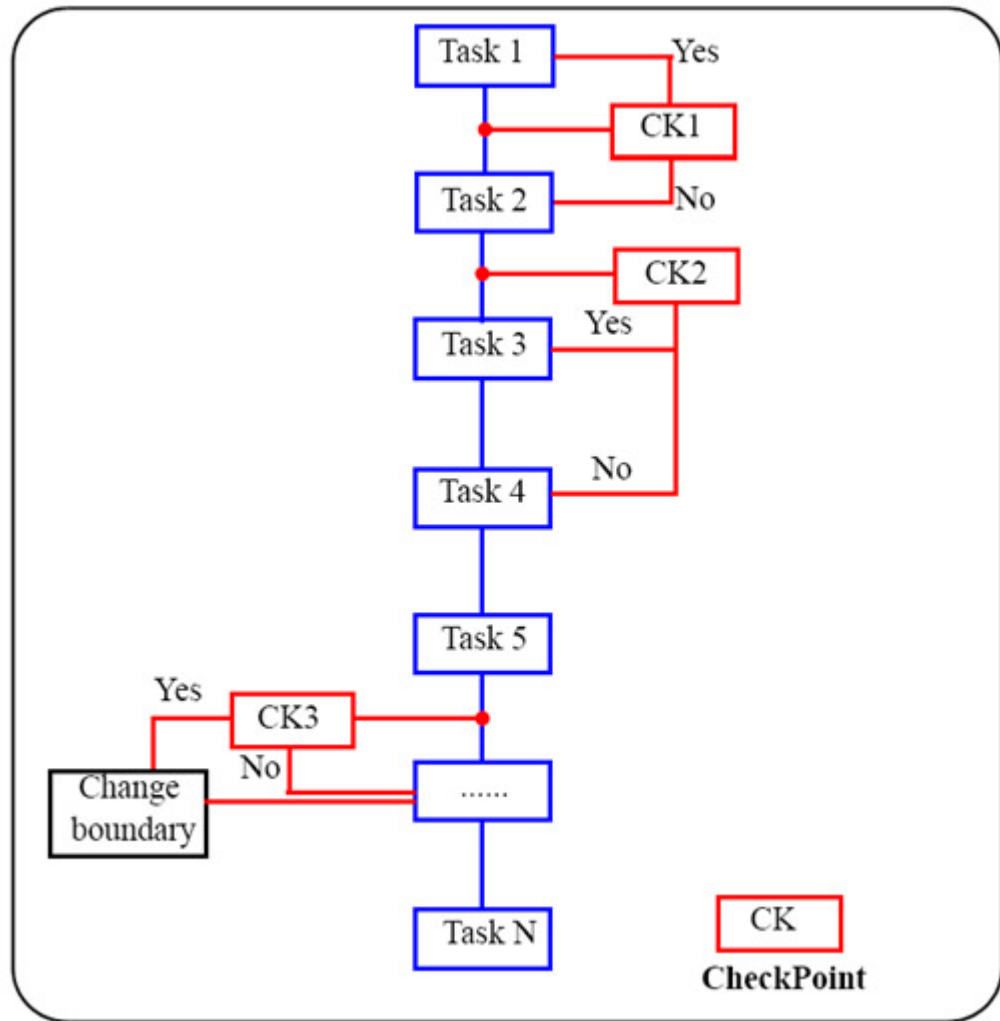
Action of the Change Parameter Value

Action	
Current Type:	Change Parameter Value
Parameter:	Cit
Value:	1e-4

Concept of Task Tree

Tasks and checkpoints constitute a Task Tree. A sample Task Tree structure is illustrated in following figure.

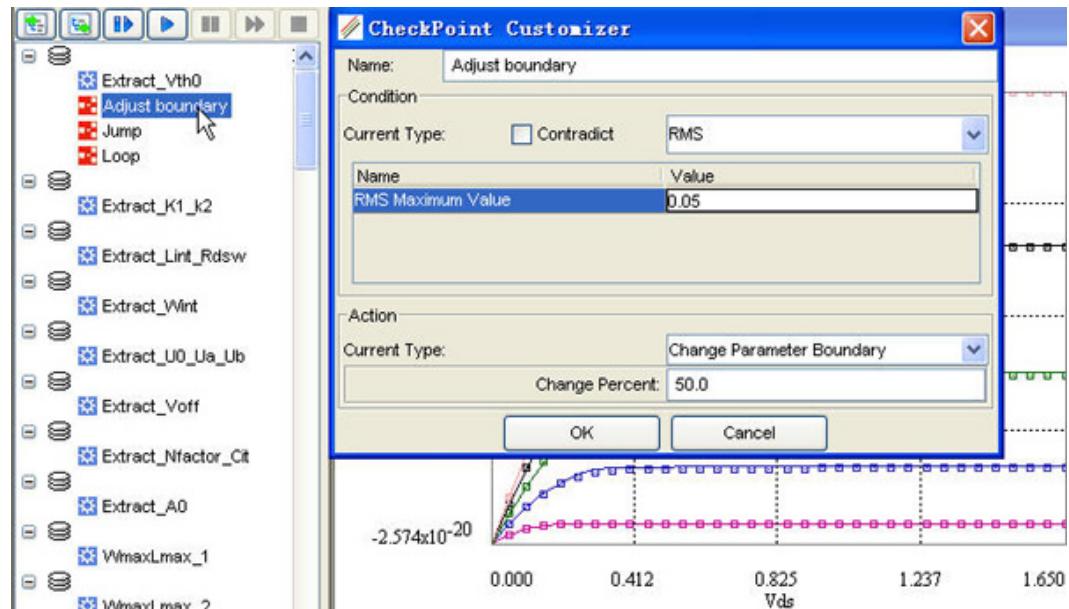
Task Tree example



Task is to perform specific actions. For example, extract.BSIM3v3 is a built-in Task Tree file in MBP and Extract_VTH0 is one of its tasks. When executing Extract_Vth0, program finds the device with largest geometry size in measurement data, locate to the Ids_Vgs_Vbs(Vds=0) page, and extract parameter VTH0 from it.


Checkpoint sets the condition of task execution. For example, we insert a checkpoint named Adjust boundary after the task Extract_Vth0. As shown in following figure, if the RMS value is larger than 0.05, i.e. 5%, after task Extract_Vth0 is done, the boundary of the parameter VTH0 is extended by 50%. For example, if the initial boundary is (1,10), after the change, the new boundary becomes (0.5,15). And, if the initial boundary is (-10,10), after the change, the new boundary becomes (-15,15).

Sample checkpoint



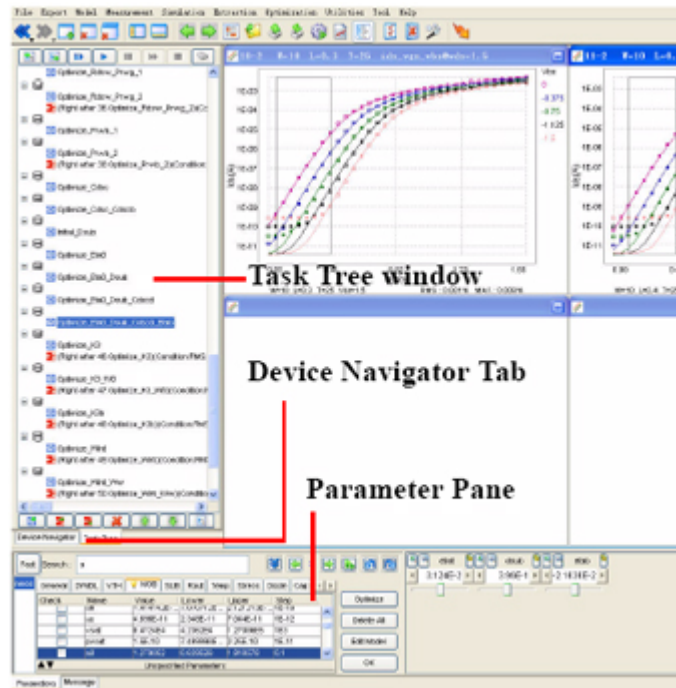
GUI and Main Modules of Task Tree

To enable Task Tree, use can either:

- Select Extraction > Task Tree from the main menu, or
- click the Task Tree icon  in the tool bar.

MBP uses tree view to display Task Tree. The device navigator and parameter panel appears with Task Tree, as shown in following figure.

Main window with Task Tree launched

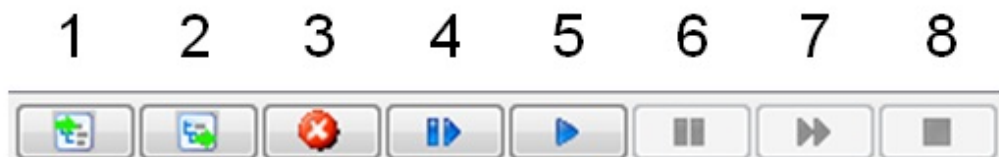


The main parts of Task Tree include upper tool bar, lower tool bar, and content panel. Following section describes the same.

Upper Tool Bar

The upper tool bar in task tree is shown in following figure.

Upper tool bar of Task Tree



Button 1 and 2 are for file management; 3 to 7 are for Task Tree running control.

- 1. Load Task Tree: To load the Task Tree file into MBP.
- 2. Save Task Tree: To save the Task Tree file as a new file or replace the existing one.
- 3. Clear Task Tree: To clear all the contents in the Task Tree.

- 4. Step: By default, a Task Tree is executed from beginning to end continuously. If Step is checked, the Task Tree eeds in steps. It facilitates to inspect every task and diagnose problem. Press Resume to go to the next action.
- 5. Start: To start running Task Tree, e.g. begin to execute extraction process.
- 6. Pause: To pause the running task.
- 7. Resume: To resume the Task Tree from the paused point.
- 8. Stop: Stop the whole process and return to the starting point.

NOTE

- It is easier to load built-in Task Tree to perform model extraction. MBP provides Task Tree for most of the supported models.
- Task Tree file name extension is consistent with the model type, e.g. **.BSIM3v3** and **.GP** are Task Tree files for BSIM3v3 model and Gummel-Poon BJT model respectively.
- **Pause** or **Stop** can take effect only after the current task is over.
- The Task Tree file is bound up with project.

Lower Tool Bar

The lower tool bar in task tree is shown in following figure.

Lower tool bar of Task Tree



- 1. Add Task: To add a new user-defined task. The created task gets added in front of the action list.
- 2. Add CheckPoint: To add a checkpoint right before the highlighted task.
- 3. Append CheckPoint: To add a checkpoint right after the highlighted task.
- 4. Remove: To remove the task or checkpoint from the Task Tree. To do this, highlight the one to remove and click Remove.
- 5. Up: This function is to adjust execution sequence of a Task Tree. Up is to move the highlighted task upward one step.
- 6. Down: This function is to adjust execution sequence of a Task Tree. Down is to move the highlighted task downward one step.
- 7. Edit: To edit highlighted task or checkpoint. Click Edit to pop up Task Customizer for the task or CheckPoint.

- 8. Task Tree Property: To provide a choice of recording each step of task tree result.
- 9. Customize Device Selection: To customize the device selection in graphic mode.

Content Panel

All tasks and checkpoints are listed and organized in the content panel in a hierarchical style. The highest level is Action. And Task, represented by blue icon, constitutes the basis of the action. Double-click a task or checkpoint to invoke corresponding configuration interface.

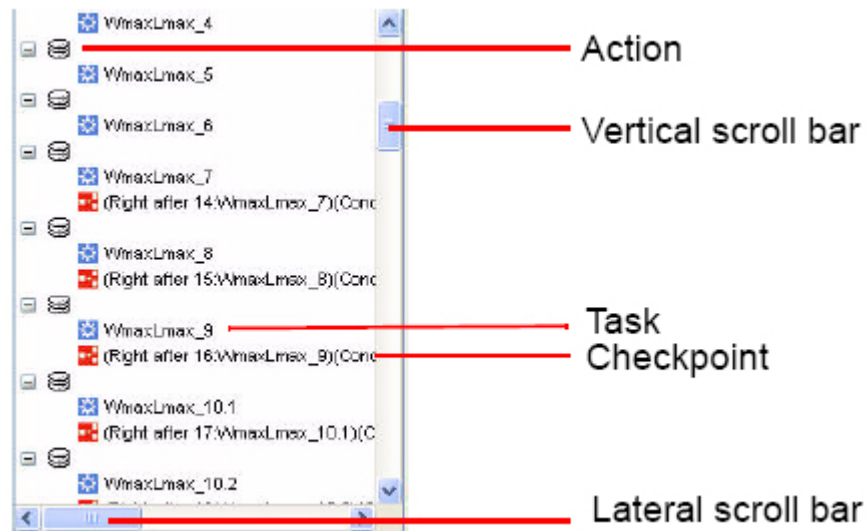
When running a Task Tree, it carries out in the sequence as arranged on the content panel (from top to bottom). You can simply adjust the sequence by highlighting target task and clicking Up or Down button in lower tool bar.

Unlike the task, if the name is not assigned, MBP shows the detail of a checkpoint in content panel. As shown in [Figure: Content pane of the Task Tree](#), the checkpoint right after task WmaxLmax_7 is displayed as:

```
(Right after 14: WmaxLmax_7)(Condition: RMS: [false, 0.01])
(Action: Change Parameter Boundary: 30.0)
```

In the first parenthesis, it indicates the inserting point; in the second parenthesis, it sets a condition using RMS; the third parenthesis comprises the action that MBP takes if the condition is met.

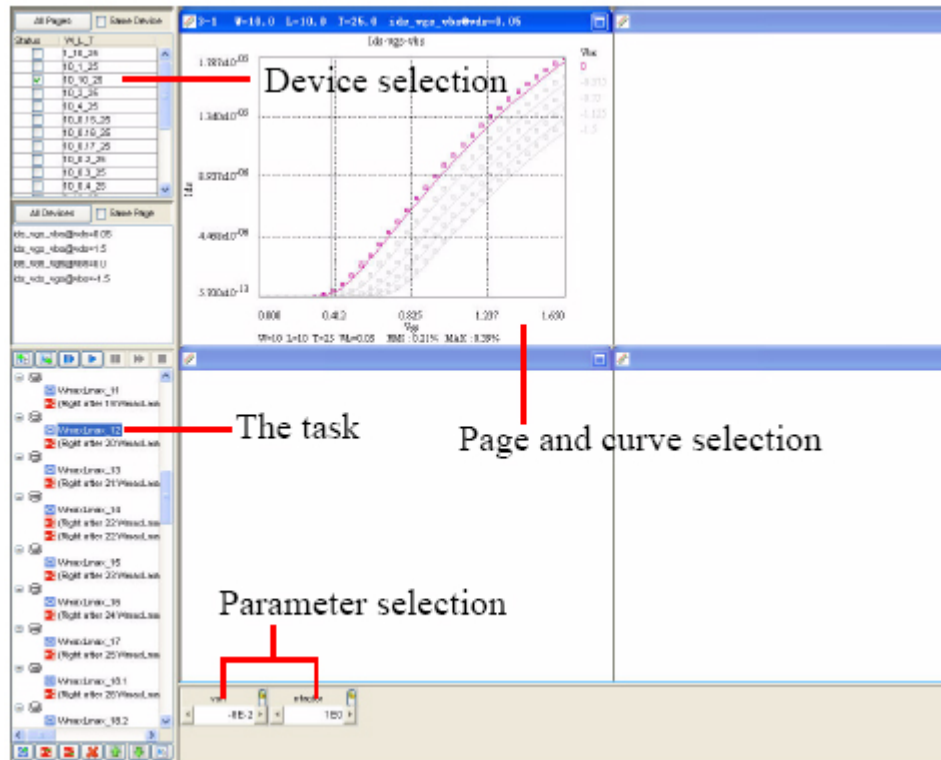
Content pane of the Task Tree



Real-time Configuration

Task Tree has a nice feature of showing real-time configuration of a task. Whenever a task is selected, MBP shows real-time device, page, and curve selection information together with involved parameters of this task on screen, as illustrated in following figure. It helps in understanding and configuring the task.

Real-time configuration of the task



NOTE


The checkpoint is tied to the specific task. Do not try to use **Up** or **Down** to move the checkpoint.

Task Customization

Use task and checkpoint to configure each action in the Task Tree. To build user-defined modeling flow by Task Tree, first configure separate task and checkpoint, and then organize them together. MBP provides Task Customizer to establish or modify the task.

Task Customizer Overview

To pop up Task Customizer window, do one of the following actions:

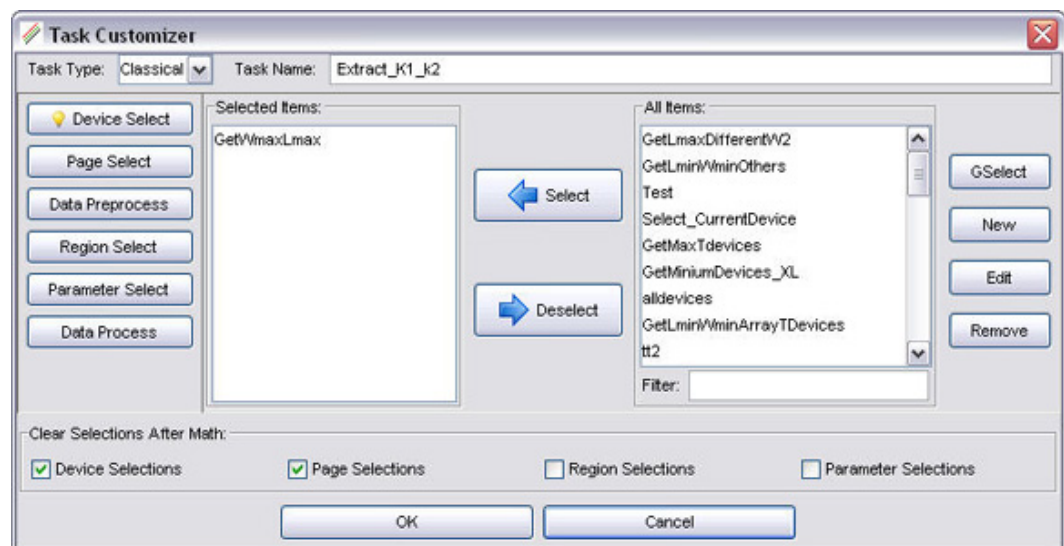
- Click a task, then click the Edit icon  in the lower tool bar.
- Double-click target task.

- Click the Add Task icon  in the lower tool bar.

The Task Customizer window (shown in Figure 6.8) includes the following major parts and editing utilities:

- Task Name.
- Select data points, including Select Device, Select Page, Qualify Data and Select Region.
- Select parameter.
- Data process & Extract.
- Extensions, i.e. options related to whether to keep data points and parameters selection after performs the task.
- The Gselect, Code, Edit, Remove button and drop-down box is to edit corresponding options.

Task Customizer window



Each part is explained in detail in the following sections.

Major Modules

Parts	Description
Task Name	To input the name of task. If an existing task is loaded, you can rename it in this box.
Device Select	To determine data points from which devices are processed. You can select either single device or multiple devices.
Page Select	

Parts	Description
	As described in Device Navigation , for a data file, a device may be characterized in different ways. We call each characteristic a page. For example, <code>Ids_Vgs_Vbs(Vds=0)</code> is one of the pages for MOS device. Unlike Device Select, you can only pick one page, or on same page for a group of devices in Page Select.
Data Pre-process	To prevent extraction or optimization from failure, you can choose and qualify the data here.
Region Select	Extraction and optimization always apply to region of the device characterization. This option is for you to pick out the desired data points from a page.
Parameter Select	To select parameters involved in the task. Its often used when performing optimization tasks. <div style="border: 1px solid black; padding: 5px; margin-top: 10px;"> <p>NOTE No necessary to specify the parameters for extraction tasks (such as "Extract_Vth0", "Extract_K1_k2" ...)</p> </div>
Data Process	This option determines how to process the selected data points. MBP integrates more than twenty functions for data processing, mainly covering extraction and optimization. You can access any of them from the drop-down box. In addition, you can implement your functions by Java scripting in MBP.

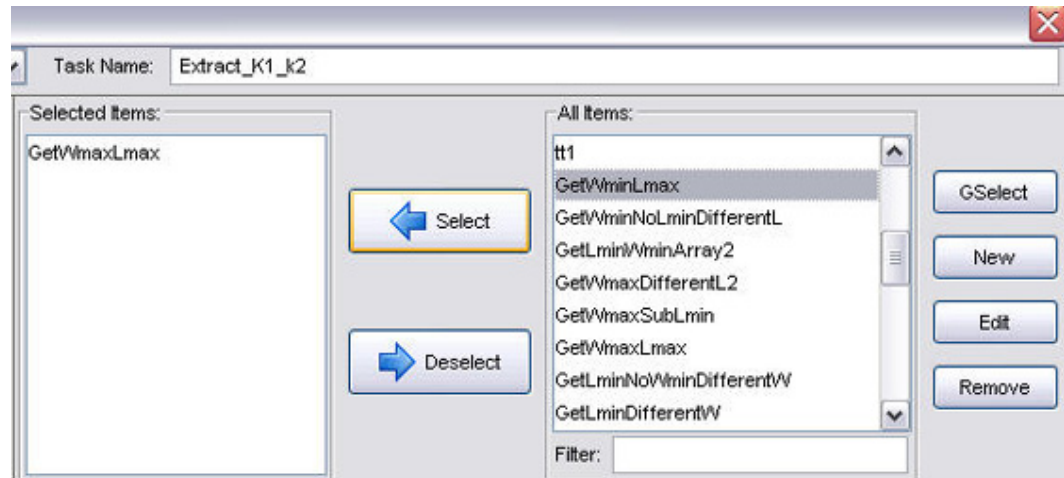
Editing Utilities

Editing utilities for the task include drop-down box, GSelect, New, Edit and Remove.

dropdown box

All available functions are listed in the right pane. In MBP, built-in functions meet general needs for data processing and selection. You can directly select the function by double-clicking one of the items from the list, or highlight one and press Select button. And once you define new functions in the Task Customizer, it gets added to list automatically, as shown in following figure.

Sample list of the Select Device



GSelect

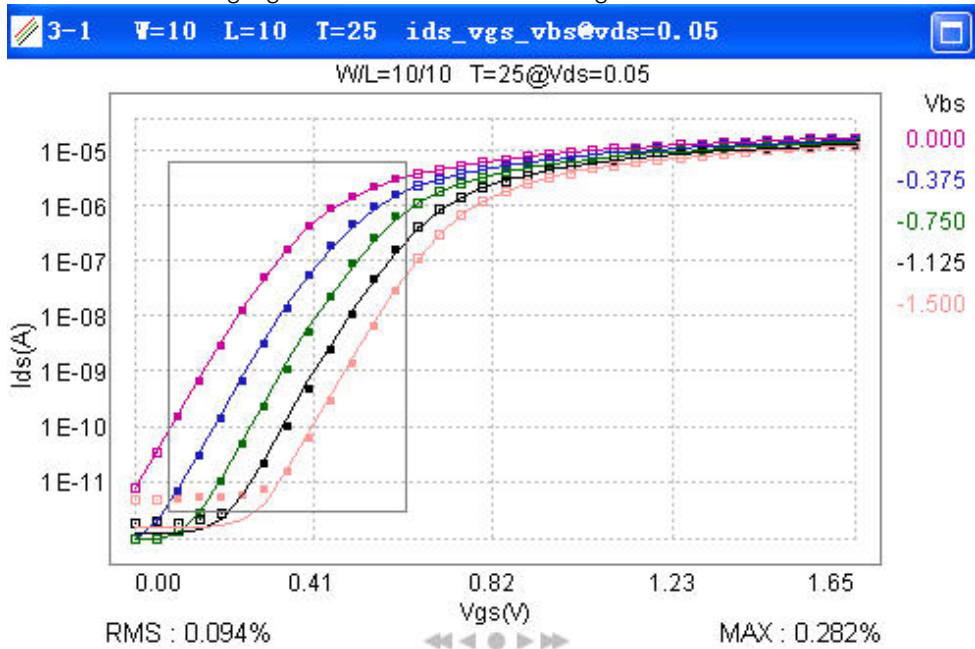
GSelect enables you to generate functions from graphical interface. We explain this by taking GSelect in Device Select as an example: click on the GSelect button and the name of GSelect button turns into "Finish". Check target devices listed in the Device Navigator. After making the selection, press Finish. Then type the name to save current device selection, as shown in following figure. The Java code is generated automatically and this option is appended to the drop-down box.

Save device selection



- To do GSelect of Parameter Select, bring up Optimization window.

- To do GSelect of Region Select, draw the region on target page directly, as shown in following figure. Gselect of Select Region



New

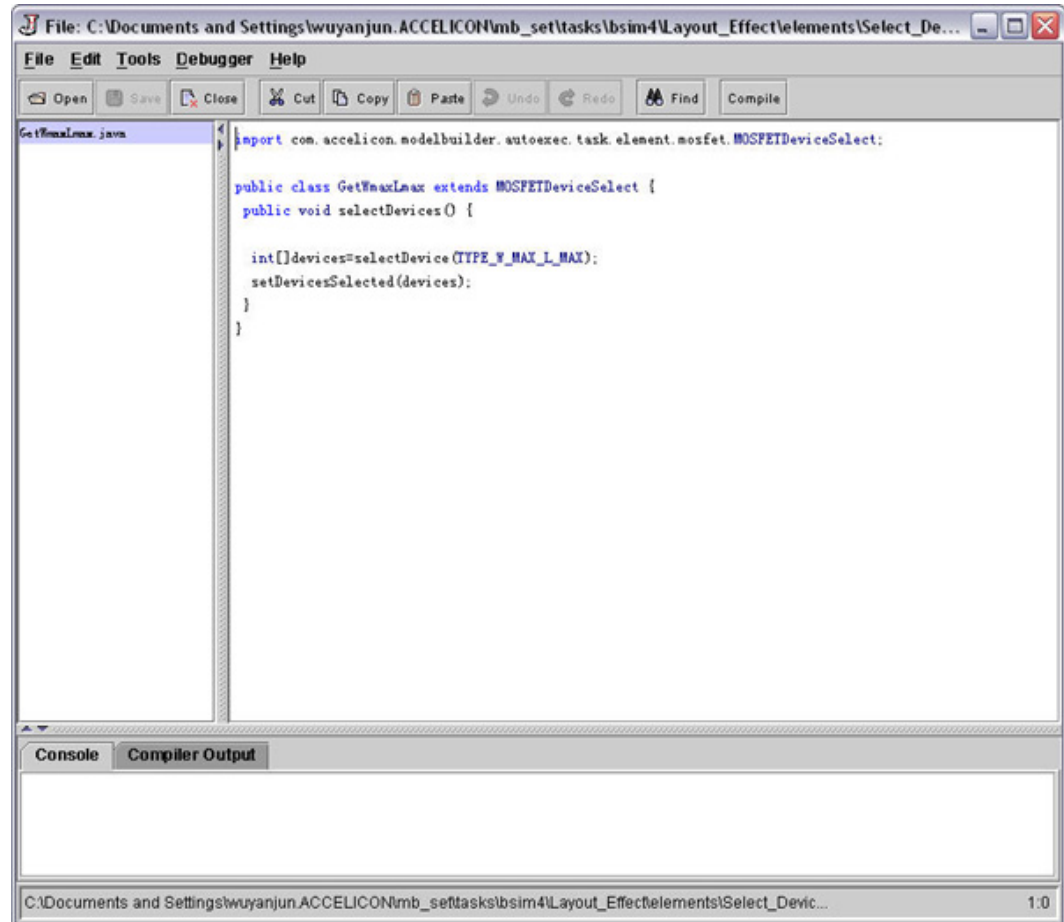
This button is used to write Java code for the function. Specify the name and class name for the function as shown in following figure. You can either define a new function or edit the existing one, depending on the Name and Class Name typed in.

Dialog box of starting Code



Click OK, Java programming interface appears as shown in following figure.

Java coding interface



NOTE

Check **Java Engine** to display the Java coding interface. Since MBP integrates compiler and debugger , it is not required to leave MBP for the coding.

Edit

To edit directly, access the code of selected function in drop-down box.

Remove

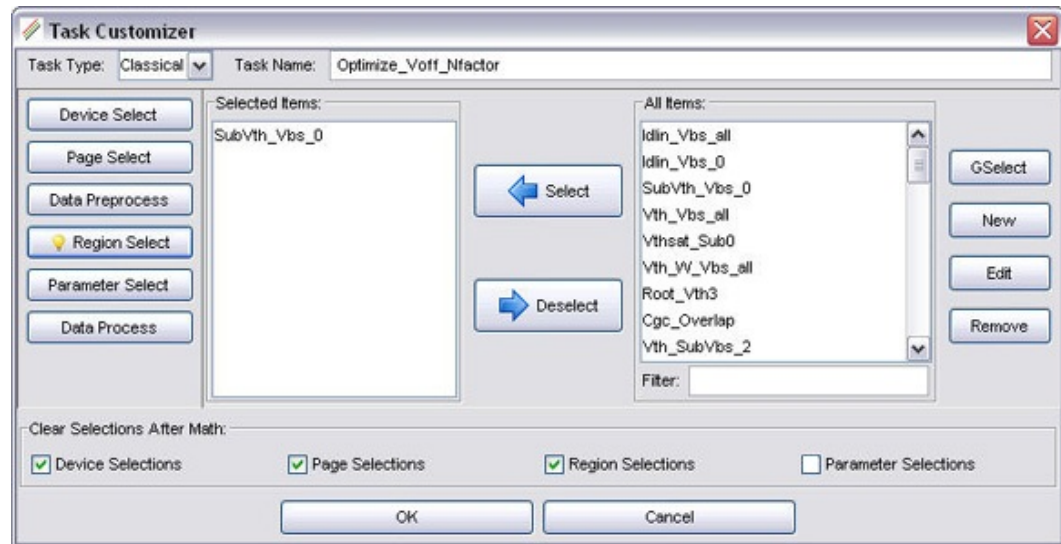
To remove the function from the list of drop-down box.

Sample Task

The following figure shows a sample task of optimizing parameters VOFF and NFACTOR. Root device (the one with maximum L and W in this case) and the page of Ids_Vgs_Vbs at Vds=0.05V are selected. Subthreshold region (SubVth_Vbs_0) acts as the target region; VOFF and NFACTOR (VOFF_NFACTOR) are parameters for fine-tuning. We choose built-in "Optimize" function to process data. Thus

during the task running, MBP's optimizer takes care of the whole optimization procedure. Since Parameter Selections is unchecked, parameters selected in this task are transferred to the next task. Other selections are cleared after completing this task.

Sample task for optimizing VOFF and NFACTOR



Task Tree Overview

Task Tree of MBP features flexibility, automation, powerful data processing, and efficiency.

- Flexibility: The parameter extraction and optimization procedures are all open for use. It enables modeling engineer to interact with whole modeling process. You can easily create, modify, and organize every task according to their own demands. The checkpoint provides a way to execute task conditionally thus helps you to further control model extraction procedures.
- Automation: MBP already integrates complete modeling solutions to implement automated modeling. In addition, you can implement their own model extraction tasks into task tree and execute the tasks automatically.
- Powerful data processing: More than 15 functions are integrated in the task tree to process the data used in model extraction and optimization.
- Efficiency: Both beginner and experienced can benefit from the specialized features of Task tree by building their own extraction and optimization flow. Moreover, the built-in extraction and optimization methodology is proved to be applicable to general device data.

Managing Models

- Binning Model
- Corner Model
- Generate Point Model
- Model File Manager

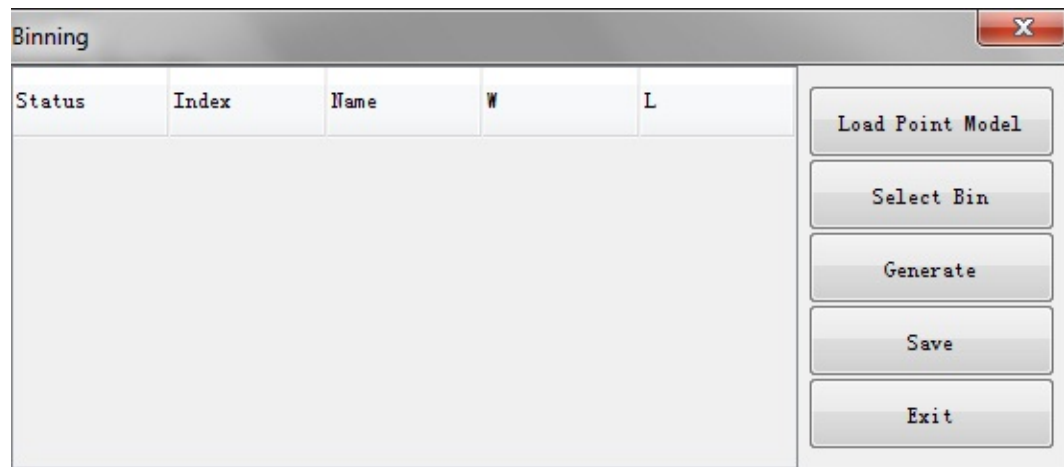
Binning Model

MBP supports binning model generation. Click Utilities > Binning, the binning generation window pops up as shown in following figure. Load point models first to generate the binning. Click Load Point Model. The Figure Binning Window with point model loaded shows a sample window with 8 point models loaded.

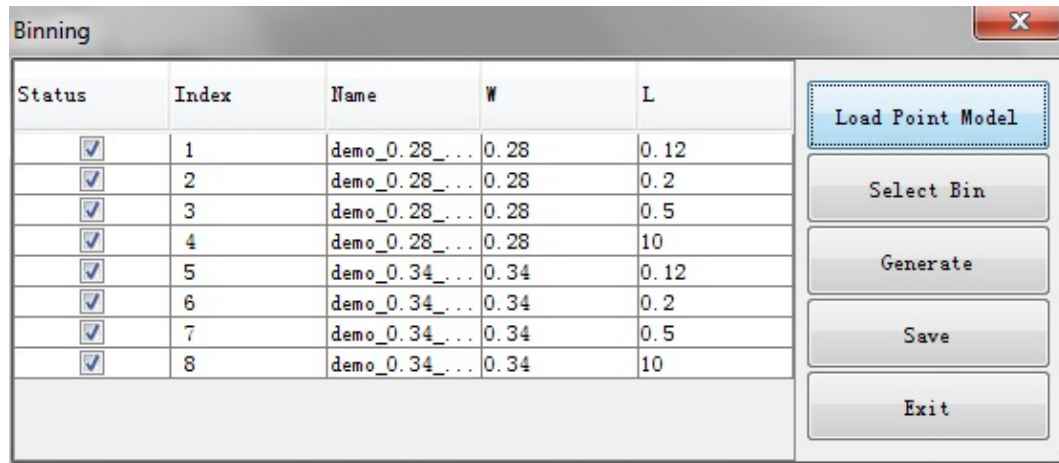
NOTE

Point model recognizable in MBP means $w_{max}=w_{min}$ and $l_{max}=l_{min}$ in model files. Or it does not proceed with binning process.

Binning Generation



Binning Window with point model loaded

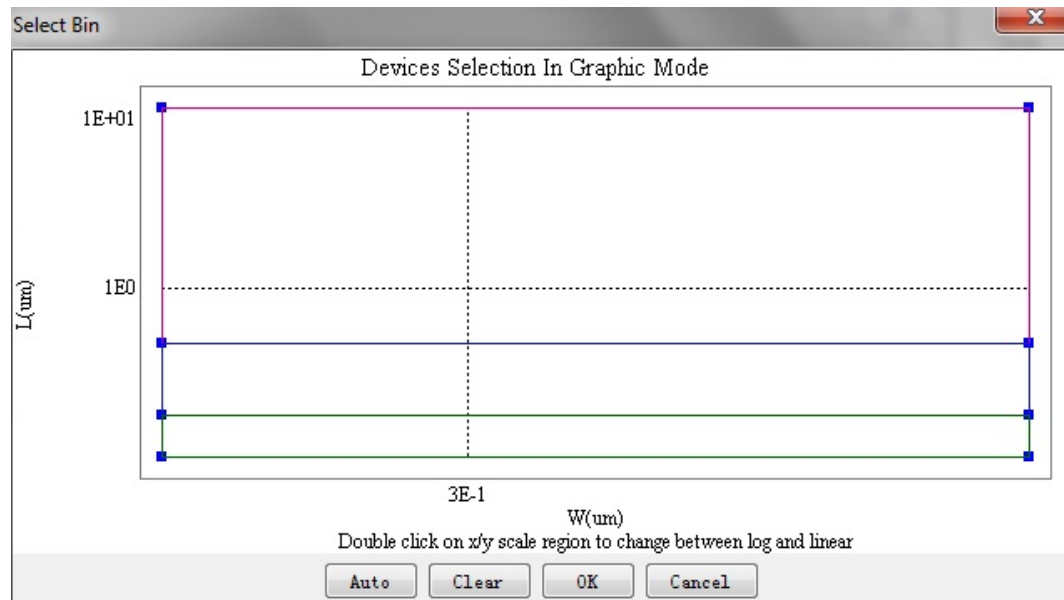


Binning region selection is done in a geometry plane. Click Select Bin and the graphical bin selection window pops up. Press and hold Ctrl, use mouse to select the bin region, a sample is shown in Figure: Bin Select Window , the 8 devices are divided to 3 bins. You can also press Auto button to let MBP auto-generate bins from every nearest 4 devices. To date, Auto function can only take effect when to every possible bin, none of 4 point models is missing. Overlapped bins are not allowed in modeling. However, in some cases, point models especially those with small sizes are too close to recognize in geometry plane. By dragging mouse over the geometry plane, you can zoom-in the covered region and draw binning region. Double-click returns the plane to the origin.

NOTE

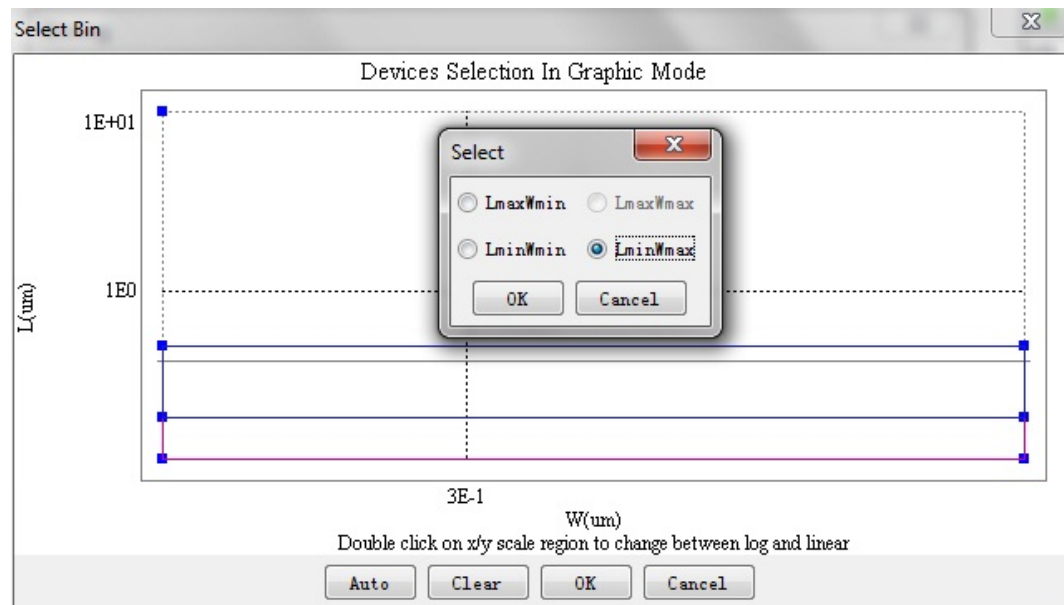
If selected binning region covers more than 4 point models in geometry plane, MBP always calculates binning parameters only from the four point models located in the corner.

Bin Select Window



If you do not have the point models that can perfectly occupy every corner of each bin, MBP provides options to create pseudo bin, you can select models of other devices for this bin as the pseudo model, as shown in Figure-Pseudo Bin Window. Click OK after the bin selection is done. After the bin selection is done, click Generate, put in the bin model name and click OK. To keep the generated binning model by pressing Save button. You can choose to expand bin boundary as shown in Figure: Bin boundary expansion.

Pseudo Bin Window

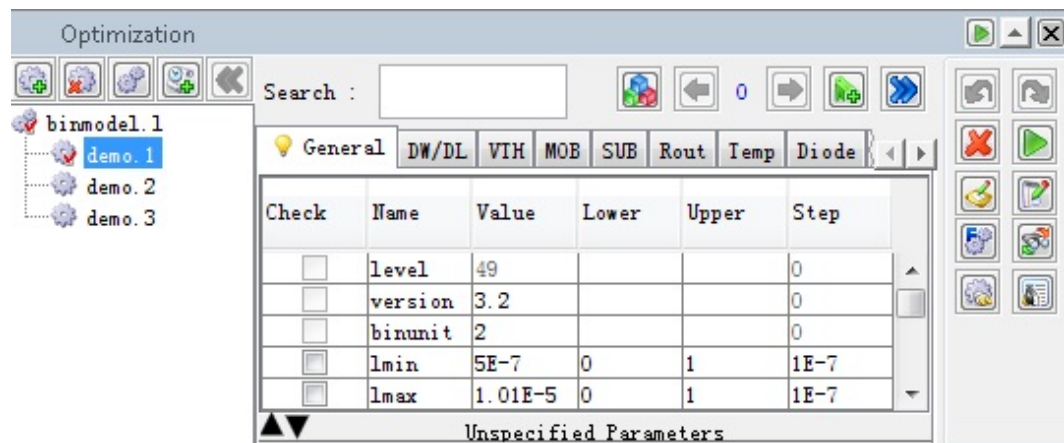


Bin boundary expansion



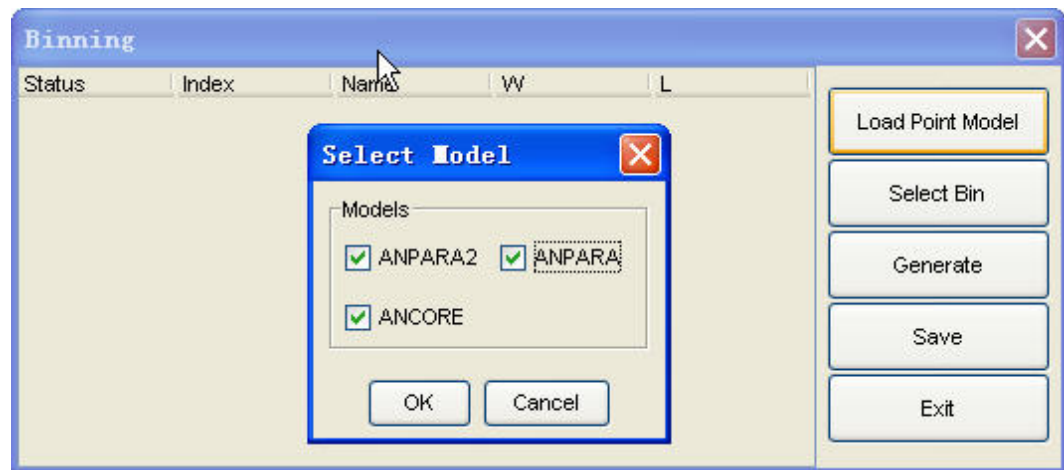
Then, you can load saved bin model and bring up the parameter window to show all the binning model parameters. The following figure for Bin Model shows an example binning model parameter window. In addition, you can choose to expand bin boundary for generated bin model in this step.

Parameter Window for Bin Model



MBP even support binning process to MOS models included in a subcircuit. After a group of point subcircuit models are loaded, MBP parses all MOS point models automatically. You can select elements to form the bin, as shown below.

Subcircuit bin model selection



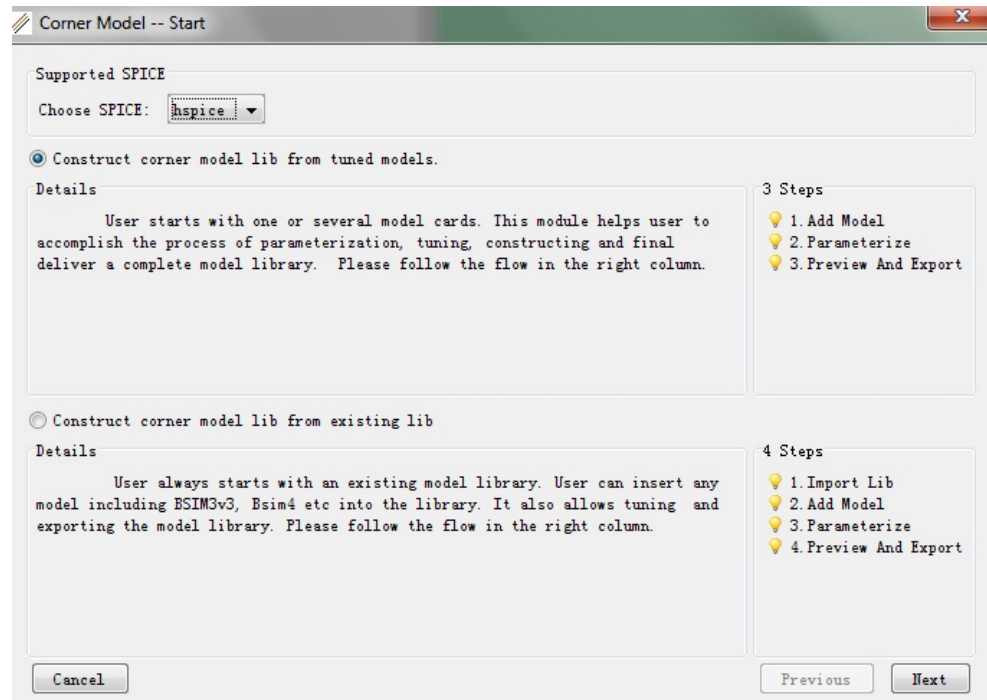
If more than one MOS elements in a subcircuit model are to bin simultaneously, MBP assume that they have the same size point models.

Corner Model

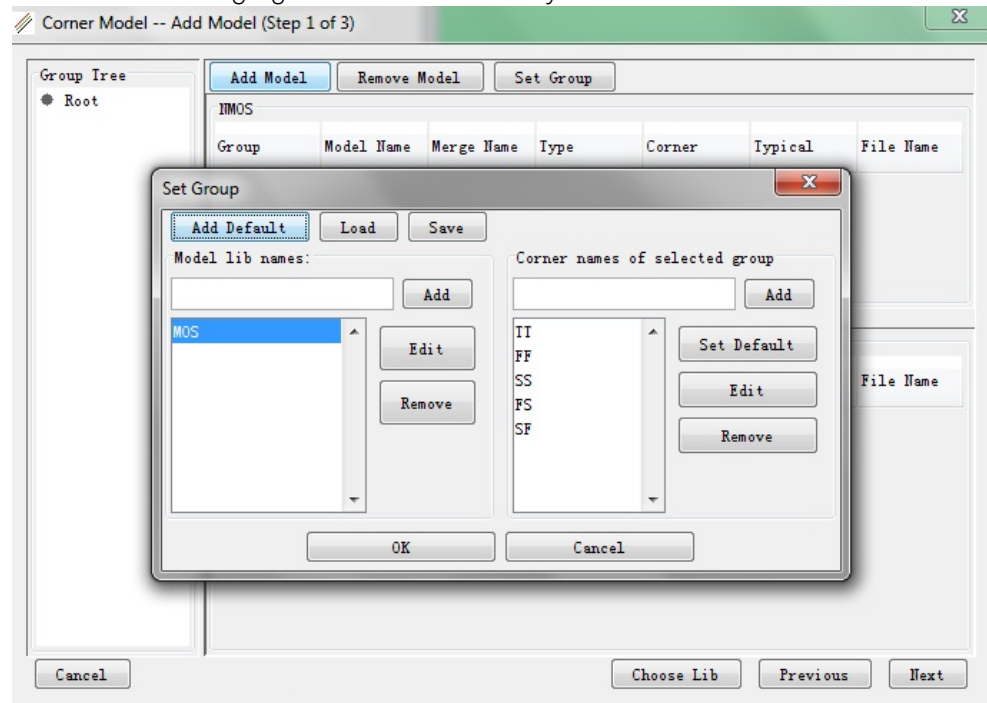
The Corner Model module is designed to facilitate modeling engineer to generate corner models. It finally offers a model library which contains all corners specified. We will use an example to explain the steps to generate corner model.

1. Click Utilities > Corner Model from the main menu to open the window shown in following figure. Choose the SPICE simulator first: hspice or spectre . There are two methods to construct the corner model library: from one or

several model cards, or from an existing model library by adding the model in it. We choose the former one as the example. Guidance for starting with the corner model



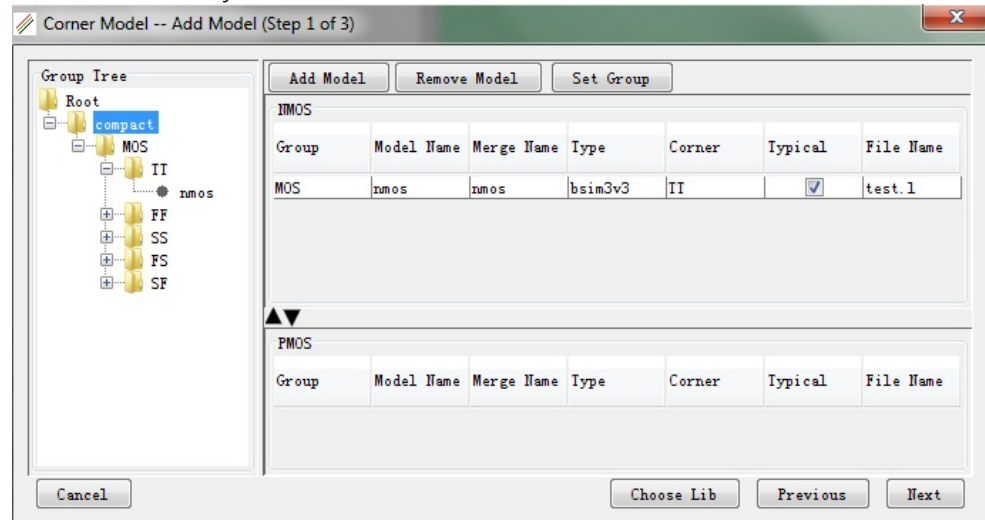
2. Add models. It allows to either load a single model card, or load one NMOS and PMOS model cards together. After loading model card, the window is shown in following figure. Set model library structure



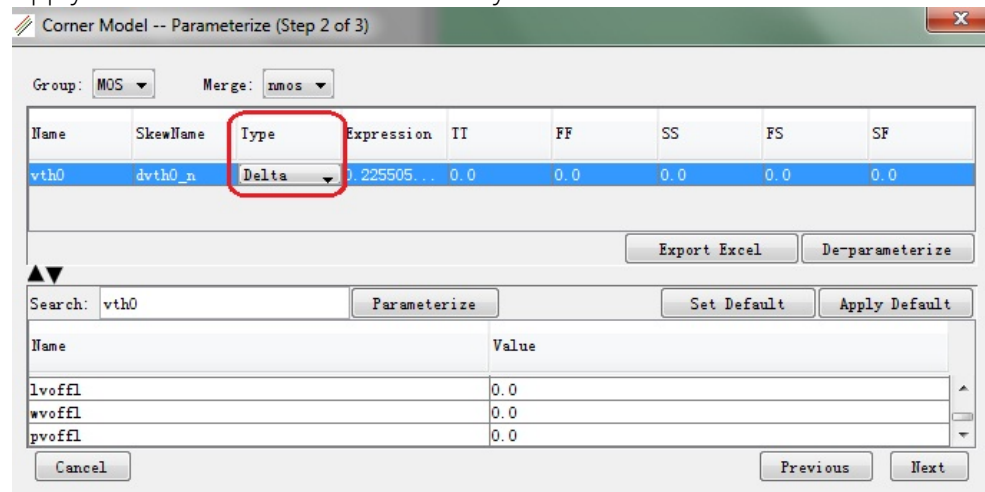
3. Specify the whole model library structure, as shown in [Figure: Set Model Library Structure](#) In the left column of the window, type in the name of the library for putting core model cards. In the right column, edit the corners. Here, you can also click Load Default button to get built-in mode library

template. After this step, the model library is initially established. Click OK button. MBP now analyzes this model library automatically and show its structure by tree view in the leftmost pane. To understand what exactly Group and Corner mean, refer to the tree view illuminated in the next step.

Set Model Library Structure



4. Go to the next step, a window pops up as shown in following figure. Select the parameter from the bottom pane for parametrization. MBP also offers a set of skew parameters by default, which can be invoked by Set Default or Apply Default button. Set Model Library Structure



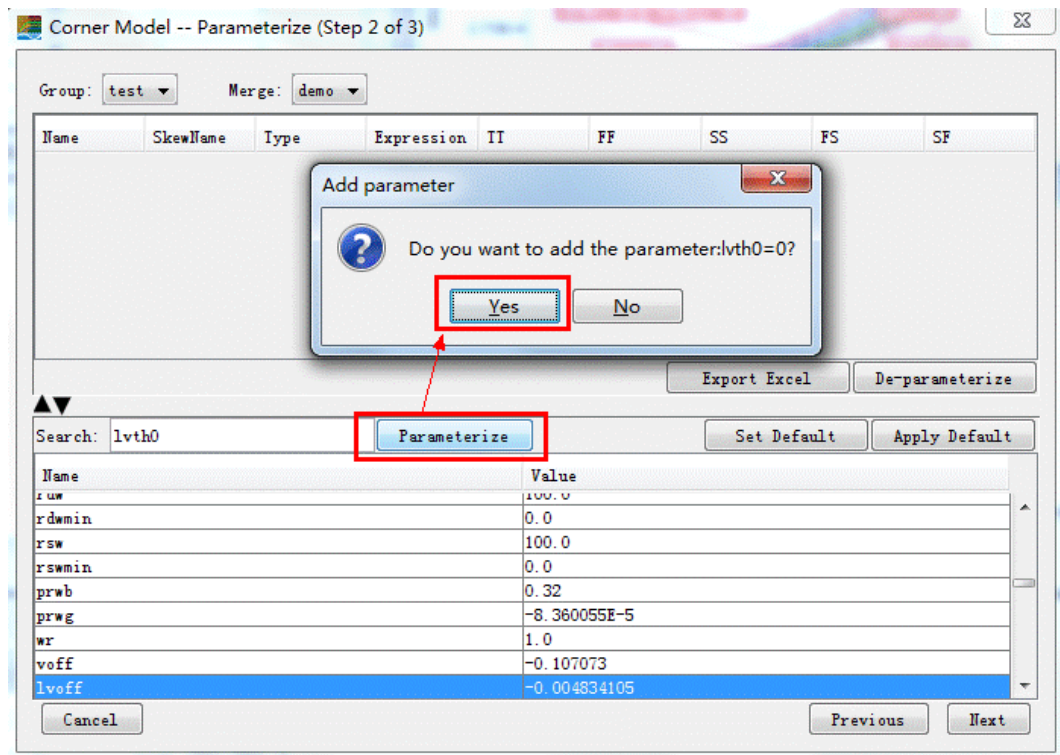
During parametrization, select Type of skewing parameters. Lets take vth0 as the example to explain the definition to "Type" of skew parameters ([Table: Definition to the TYPE of skew parameters](#)).

- Type Delta for dvth0_n means in the corner model vth0 gets expressed as $vth0 = original_vth0_value + dvth0_n$ and dvth0_n can be skewed.
- Type Absolute for dvth0_n means in the corner model vth0 gets expressed as $vth0 = dvth0_n$ and dvth0_n can be skewed.
- Type Ratio for dvth0_n means in the corner model vth0 gets expressed as $vth0 = original_vth0_value * dvth0_n$ and dvth0_n can be skewed.

- Type DRatio for dvth0_n means in the corner model vth0 gets expressed as $vth0 = original_vth0_value(1 + dvth0_n)$ and dvth0_n can be skewed. Table: Definition to the TYPE of skew parameters

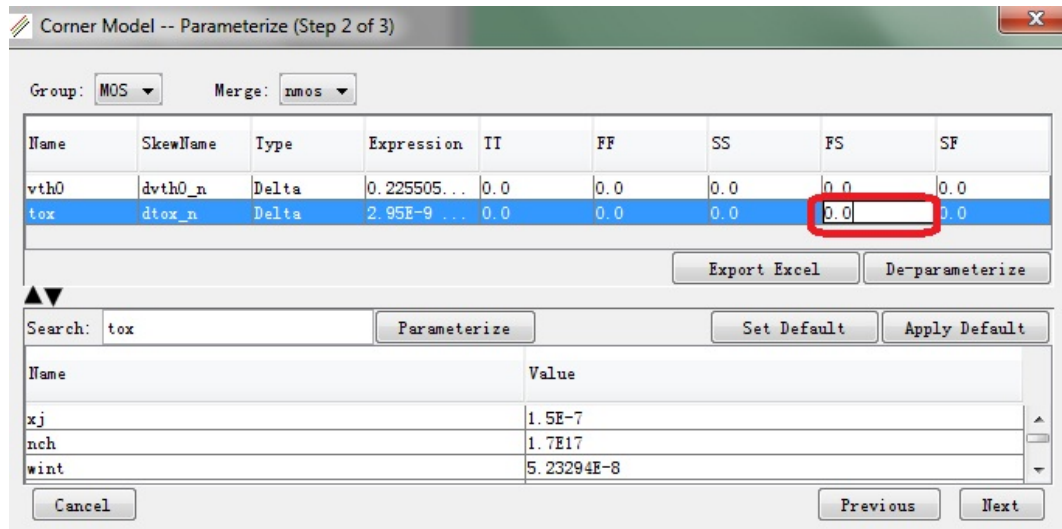
TYPE	Value Calculation
Delta	$S_{vth0} = 'vth0 + dvth0_n'$
Absolute	$S_{vth0} = 'dvth0_n'$
Ratio	$S_{vth0} = 'vth0 * dvth0_n'$
DRatio	$S_{vth0} = 'vth0 * (1 + dvth0_n)'$

- For these parameter which does not exists in all bins, MBP will add the parameter into the model and set it to "0"



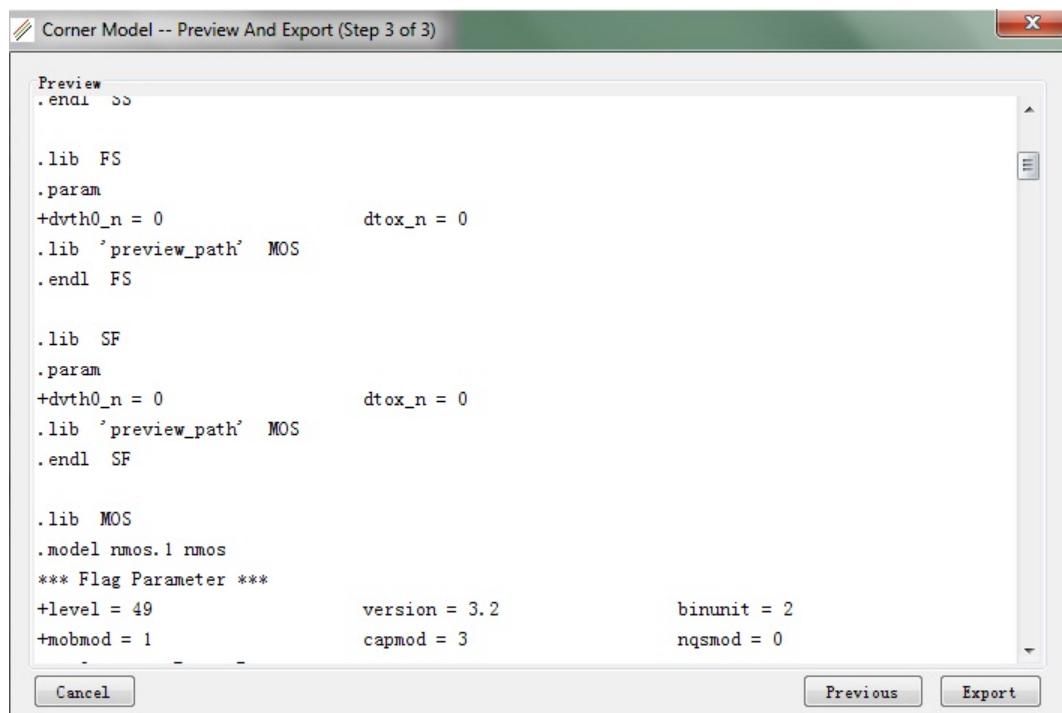
5. Set initial skew parameters. Type in values as the start of skew parameters, as shown in following figure.

Set skew parameters for FS and SF corner



6. Then MBP prints out the complete model library for your review, as shown in following figure. By this, it can be clearly seen how this model library is organized. You can go back to rebuild the model library until the satisfying result is obtained. Finally, export the model library directly.

Output corner models



NOTE

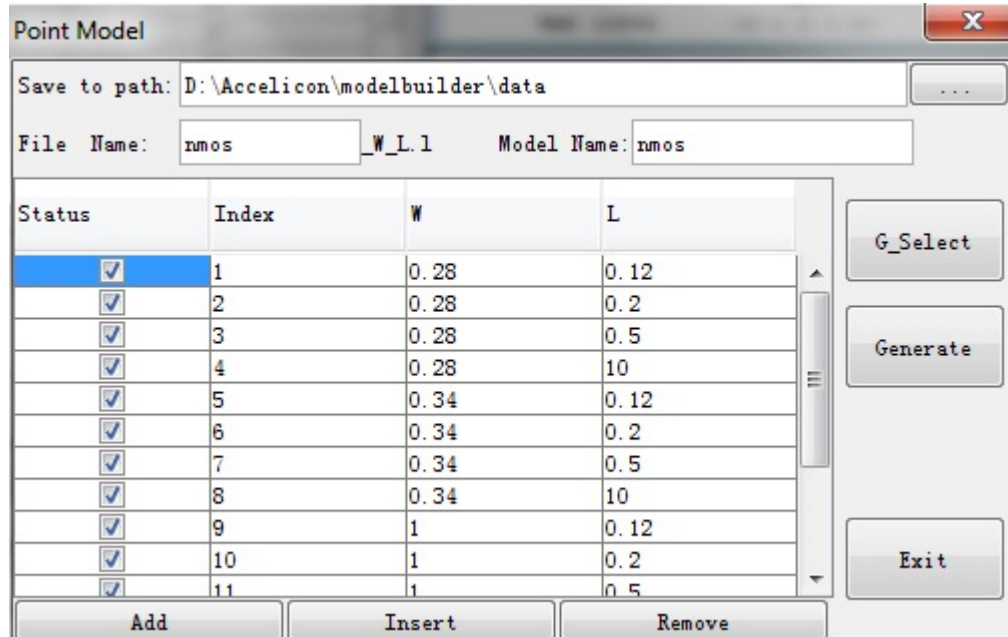
Above content describes how to build a corner model. As for how to tune a corner model to fit ones own target, refer to the document **Model Tweaking** in MBP Modeling Guide.*

Generate Point Model

For your convenience, MBP also offers a tool of generating point model from a binning model. The instance parameter, including W, L and T, for each point model comes from measurement data. Do one of the following to specify instance parameters:

- Open a project with full set of measurement data and a binning model already, then click Utilities > Point Model, select device size from the table or by G_Select.
- Open or create a project with a global or binning model only, then click Extraction > Point Model, use tool bar at the bottom of the window to edit measurement data, as shown in following figure. By pressing Generate, point models are saved to designated directory. The file names of point models are suffixed with W and L value, such as snmos_1_10.l in this case.

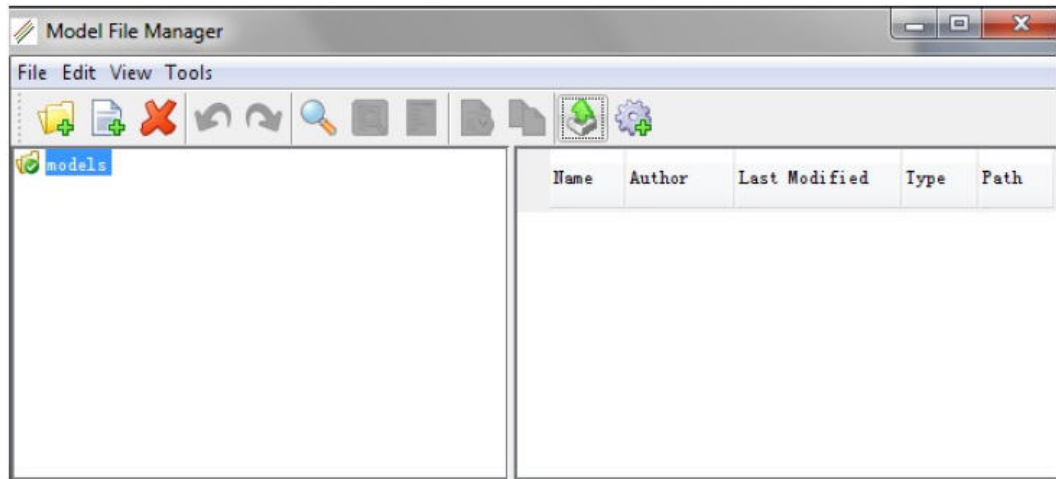
Generate point models



Model File Manager

Model Manger in MBP provides an efficient platform to manage model files. You can start up the window as shown in following figure by clicking Utilities > Model File Manager from the main menu.

Model file manager



Tweaking Models

- Binning Model Tweak
- Model Tweaking

Binning Model Tweak

General

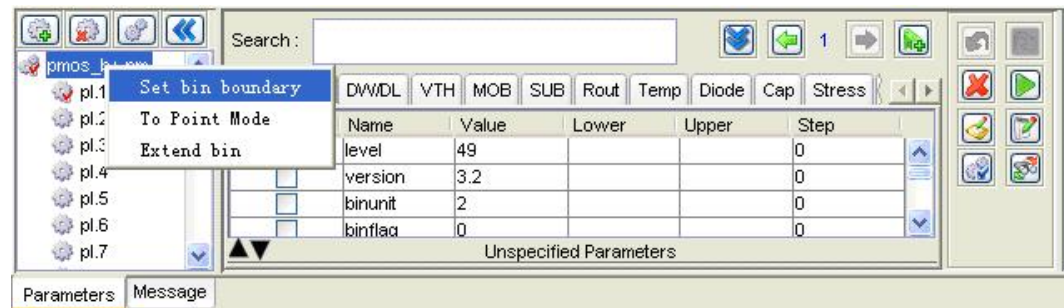
With process variation or the requirement of fitting a specific target, instead of tuning a model totally from the beginning, tweaking is used as a fast way of achieving corresponding model. For a binning model, modeling engineer should always keep continuity from bin-to-bin. MBP integrates binning tweak capability to meet such request.

MBP mainly supports the following three ways of tweaking a binning model:

- Change bin model directly, or say tune model parameters in each bin section.
- Change point model. That is breaking the binning model to point models. After tuning parameters of point models, redo the binning.
- Extend bin. It supports rescaling bin region and also inserting new binning region.

After loading any binning model, those options are available by right-clicking on the model node as shown in following figure.

Select Binning Solution



- Set bin boundary

For a binning model, one often extends the maximum channel width and length to an extent to verify the simulation of W_{max}/L_{max} device. For example, model is extracted from $W_{max}/L_{max}=10\mu m/10\mu m$ device. However, in final binning model, we get bin boundary of $W_{max}/L_{max}=10.1\mu m/10.1\mu m$. This option allows you to set maximum width and length and back.

- To point mode

Convert the loaded bin model to point models, and newly generated point models are right the original bin boundary points. After this operation, you can tune the bin boundary point models directly.

- Extend bin

Change to extend bin dialog. In this dialog, you can re-scale the bin region or generate a binning model which makes all the target devices including the new inserting points as bin boundary point devices.

Following are the methods to tweak a binning model:

1. Select a parameter (eg. Vth0) and then parametrize it (vth0='0.5+dvth0'), you can tune the parameter 'dvth0' directly;
2. Tune bin parameter directly.
3. Convert to point mode, then tweak the point model directly
4. Generate a new binning model which makes all the target devices as one of the bin boundary points. Then, the issue has been converted to tweak the new binning model.
5. Expand bin boundary.

Binning model tweak

Change binning model directly

Both 1) and 2) are the methods used to tweak a binning model directly. In this way, you can tweak a binning model just like tune a global model. The main advantage of this idea is convenient and easy to understand. The followings are the detail flow of tweaking a binning model directly.

Determine the new target device and target values

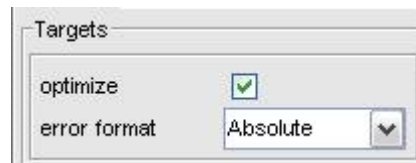
To define the new target devices and target values, activate the module of corner tweak. Through this module, MBP can display the target values of both the newly defined target devices and the original devices. After active the corner tweak window and define the target (has been introduced in the chapter of corner tweak), you can switch to the tweak interface as shown in following figure. In this interface, you can add new target device through click Add device button or select on Graph button. By double-click the detail information of instance parameter and _DES, you can change the value of device geometry and target.

- L/W/P parameter: This parameter can only be added to the current select bin. When the parameter is added, calculate current bin point values and then update the reference parameters of other bin. (E.g. lvth0)
- Remove: This section describes how to remove a parameter from the model.
 - Non-binable parameter: This parameter is removed from all bins. (E.g. tox)
 - Bin-Core parameter: Remove the parameter from the select bin. Then MBP calculates the current bin point values and then update the reference parameters of other bin. (E.g. vth0)
 - L/W/P parameter: Remove the parameter from the select bin. Then MBP calculates the current bin point values and then update the reference parameters of other bin. (E.g. lvth0)
- Parametrize: It can only be used on non-binable and bin-core parameters.
- Optimize: If the same bin parameters are selected from different bins, then optimizer does not work, else it works.

Optimize

Checking the optimize function as shown in following figure, you can do optimization.

Target Select



Save newly generated model

Check if the tweaking binning model can satisfy your requirement. If yes, you can save the newly generated model directly and if no, tweak this binning model furthermore.

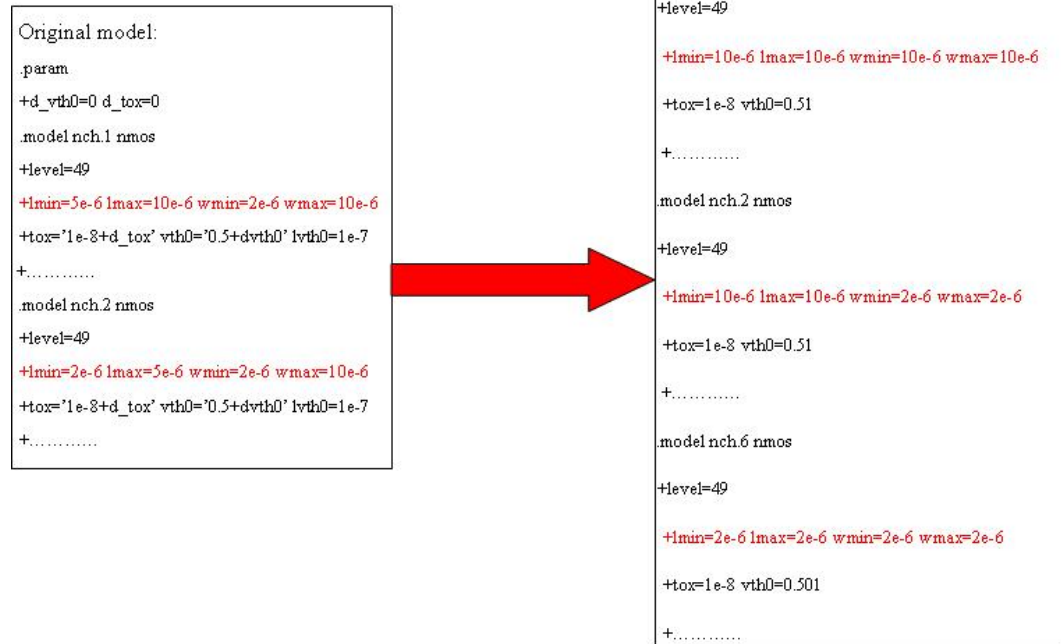
Change bin point models

3) is the method to tweak a binning model through tuning bin point models. It is essentially the way to tweak a binning model. The following introduces the detailed flow of tweaking a binning model through tuning the bin point models.

Change to point mode and determine the new target devices and target values

The second method mainly focuses on the bin-point models. Change to point mode first, once you break binning model to point models, MBP generates a set of point models with each bin boundary point is a point model as shown in following figure. Then, define new target devices and target value just as the way used in *change binning model directly.

Point Models



Determine which point model gets tweaked

Similar to the way used in change binning model directly, obtain the geometry of the target devices (the device selected to be optimized), then check which instance of the point model parameters are exactly equal to the target devices. These point models are the ones which gets tweaked.

Select tweaking parameters

According to the target value and the point models to be tweaked, select the parameters to tweak these point models. In the process of selecting tweaking parameters, you can do the followings:

- Add parameter: If a bin-core parameter is added to a point model, add this parameter to other points at the same time as well;
- Remove parameter:
 - If a bin-point model parameter is removed from a point model, all the reference points must delete this parameter at the same time as well;
- Select parameter:

- If one non-binable parameter is selected, then the same parameter in other point models cannot be selected again (E.g. tox). If it is a binable parameter, then we can select and change the same parameter of each point model (E.g. vth0).

Optimize

Checking the optimize function, you can do optimization.

Save newly generated model

Check if the tweaking point models can satisfy your requirement. If yes, you can save the newly generated model directly and if no, tweak these point models furthermore.

Extend bin

Both 4) and 5) are the methods used to tweak a binning model through extending bin. In this way, you can tweak a binning model by adding more point models or extend bin boundary. It mainly has two advantages: On one hand, you can expand the bin boundary of the current model to cover more devices (as the red point in fig. 5.); On the other hand, he can also generate a new binning model which makes all the target devices including the newly inserted devices are bin boundary point devices. Therefore, you can tune all the parameters of target devices to the expected value (as the blue point in [Figure: Extend Bin](#)). The followings are the detail flow of tweaking a binning model through extending bin.

Determine the new target devices and target values

Similar with that of the above 1 and 2 methods, the first step is to determine the new target devices and the target values through corner tweak module. Normally, if we use the function extending bin, the target device is often not the bin boundary point devices or is out of binning boundary.

Determine which bin/point model should be tweaked

This process is also similar with that of the above 1 and 2 method, you can tweak this binning model through tweaking the binning model directly or tweaking the point model.

Select tweaking parameters

Please see the details from the above 1 and 2 methods.

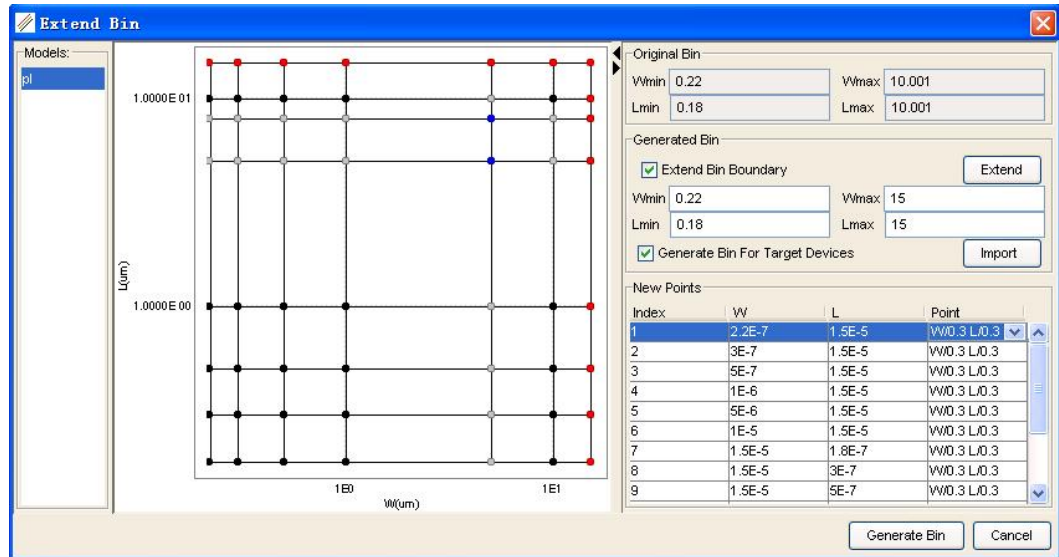
Optimize

Checking the optimize function, you can do optimization.

Save newly generated model

Check if the tweaking model can satisfy your requirement. If yes, you can save the newly generated model and if no, tweak these point models furthermore.

Extend Bin



NOTE

- In **Figure: Extend Bin** the black bin points are the original bin points, the blue ones are the newly inserted target devices which are not the bin boundary points, the red ones are the extended bin boundary generated points and the grey ones are the pseudo points.
- The use of the **extend** button in the figure is to extend the bin boundary to fit the new target, also, you can determine which point model can be used as the model for the extended points.
- The use of the **import** button in the fig is to provide you the tip that the blue points are originated from corner tweak. Also you can activate corner tweak module to re-edit the target information.

Model Tweaking

General

The model card is tied to the actual fabrication process used to manufacture the device. Even a well developed fabrication process results in variations in the devices produced. These variations are likely to affect the device characteristics and circuit behavior. To account for the variations in a semiconductor device fabrication process, based on an initial set of typical and corner device models, re-target the process dependent model parameters by fitting the process fluctuation.

The smart model tweaking module integrated with MBP enables easy model re-targeting, adjustment of global or binning models according to new specification. Both model cards and model libraries are well supported. We have an individual chapter on the binning tweak. This chapter mainly focuses on tweaking a global model.

Global model tweaking

Starting with a global model card, the main flow is as follows:

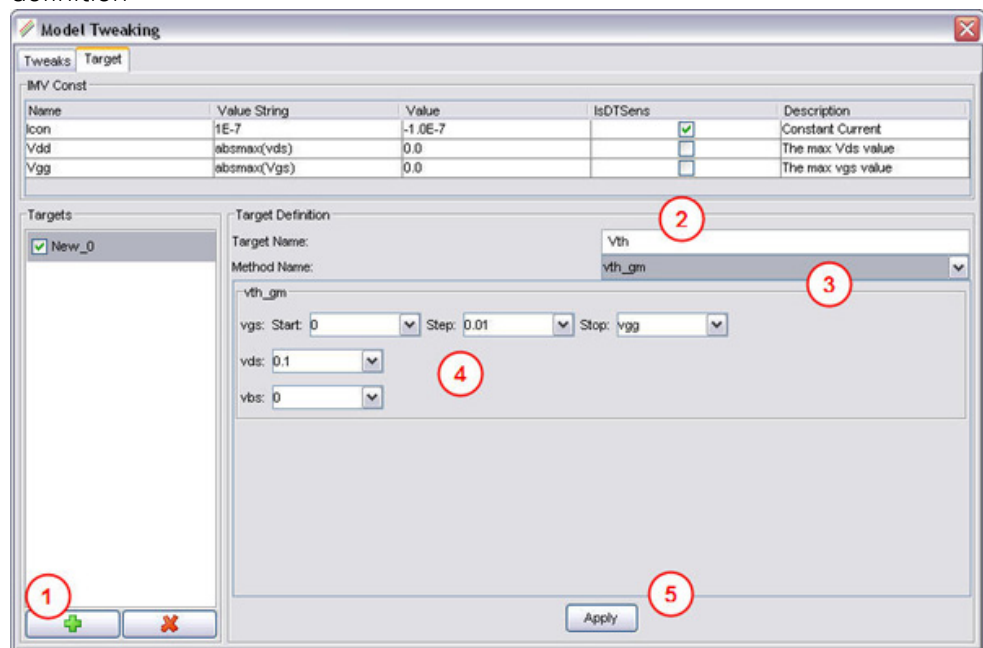
- Define the tweaking target
- Define target devices for tweaking
- Tweak the model to fit tweaking target of target device

Click Extraction > Model Tweaking to activate the key frame for model tweaking, within which you can interact with re targeting process.

Define the tweaking target

The process specification may include typical values and deviation values for specific electrical quantities, such as threshold voltage (V_{th}), drain saturation current (I_{dsat}), etc, what we call tweaking target . We cite typically included targets V_{th} and I_{dsat} as the example to append tweaking target in MBP.

1. Click cross sign to add a tweaking target
2. Assign the name to this target, V_{th} for example
3. Choose one built-in algorithm to achieve above target from this drop-down list. The V_{th_gm} means to get the V_{th} with max GM method.
4. Determine the input for performing the algorithm. For V_{th_gm} , specify bias and sweeping conditions, like constructing a virtual $I_{ds_Vgs_Vbs}$ curve at $V_{bs}=0$ and $V_{ds}=V_{dlin}$.
5. Click Apply button to confirm the configuration.
6. Repeat steps from 1-5 to add I_{dsat} as tweaking target. Tweaking target definition



To be stressed, IMV const represents constant values that you can refer in configuring bias conditions. It shares origin with IMV Pages, that defined in imv_const.ini under MBP HOME\jef\usr\imv\device\mosfet. If you do not want to refer any value from it, just type in the figure directly.

Here, ISDTSens is the flag to determine if this IMV constant is sensitive to the model type (nmos or pmos). For P-type devices, checking ISDTSens means the value string gets multiplied by '-1'. Refer to the table1 for details (we have set value string of icon as 1E-7). For NMOS devices, whether checking ISDTSens makes no difference.

Table 1 Sample Icon settings

Model type	Icon value (check IsDTSens)	Icon value (uncheck IsDTSens)
NMOS	1E-7	1E-7
PMOS	-1E-7	1E-7

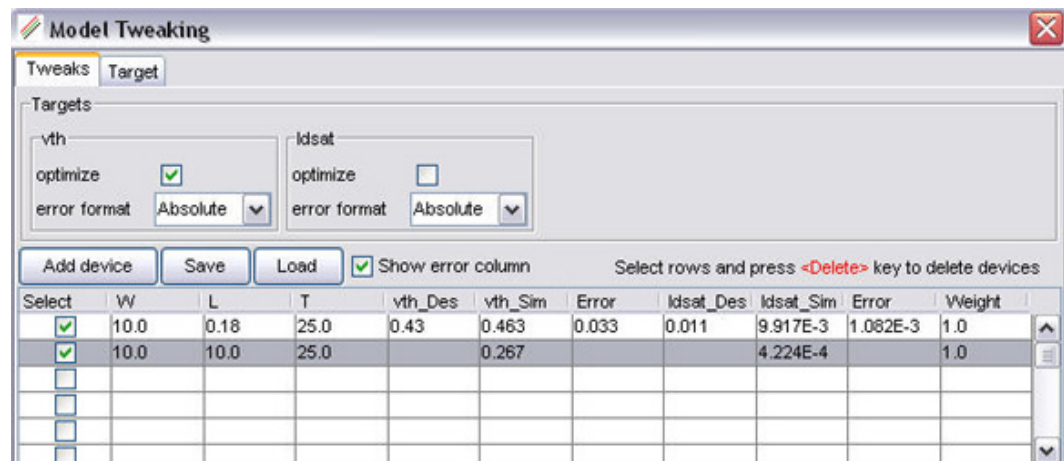
Define target devices

After defining tweaking targets, switch to tweaks pane as shown in Figure :Adding target devices. Press Add device button to make target device on the list. Then edit the device to requested size and condition, like changing W L and T in this case. Right-click on the table pops up instance parameter list. MBP allows you to save or load device information.

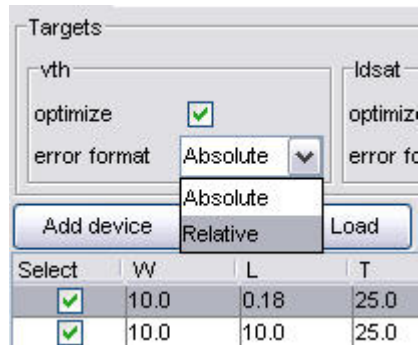
Following device information, MBP shows the design target (e.g. vth_Des and Idsat_Des, suffixed by _Des), simulated value (e.g. vth_Sim and Idsat_Sim), and their difference (e.g. Error). You can choose the error format from the drop-down list, as shown in Figure :Error format.

You are only allowed to edit design target. The simulated value and error response instantly once you tune model parameters

Adding target devices



Error format

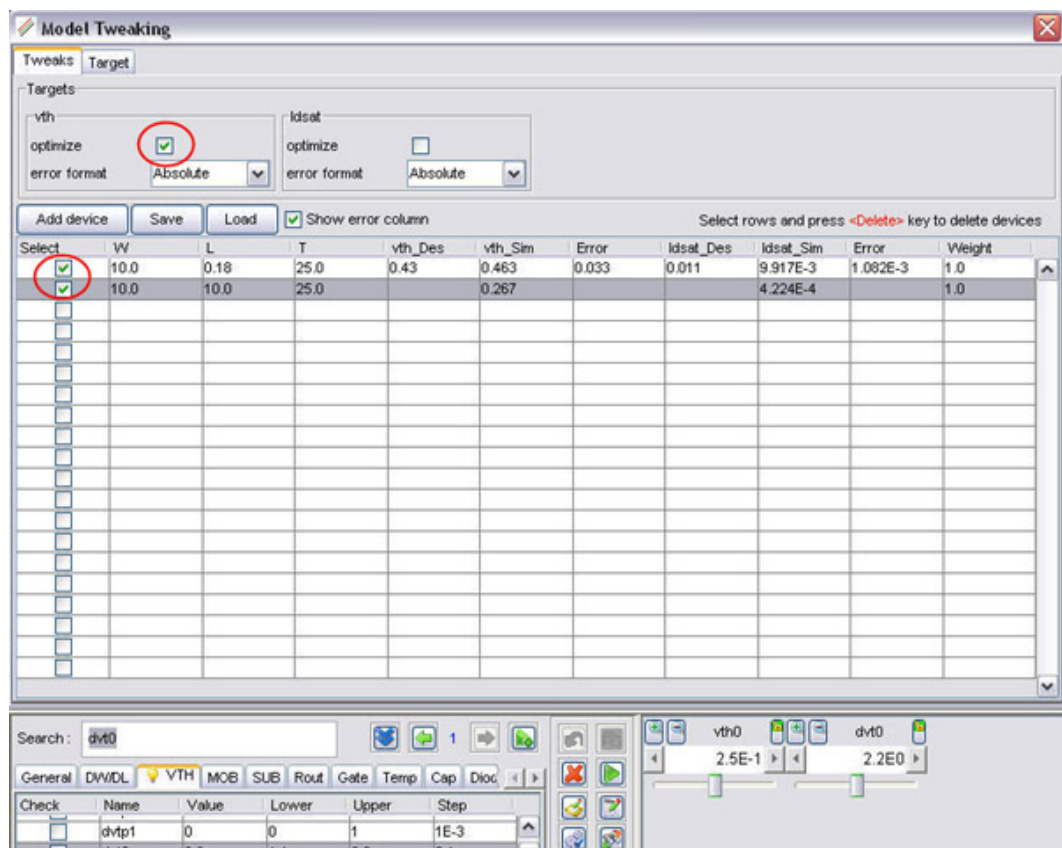


Tweak the model

Select proper parameters from the Optimization/Parameter window, then you can start model tweaking.

MBP provides both manual tune and semi-automated tune. For the latter, make sure that the design targets are set, parameters are selected, the optimized items for tweaking target are checked and the select items for target device are chosen. Then, press optimize button in the Optimization window, MBP invokes internal optimizer and get the solution in a few seconds.

Tweak the model



Once more than one target devices are set for semi-automated tune, you can differentiate the importance among devices by specifying the Weight item. Higher weight value with the device generally promises higher accuracy.

BSIM3v3 Modeling

This chapter is to give guidelines on how to extract BSIM3v3 DC model. We provide information on the following topics:

- [BSIM3v3 Overview](#)
- [Device Requirements](#)
- [Measurement Conditions](#)
- [Extraction and Optimization of BSIM3v3 Model](#)
- [BSIM3v3 DC Modeling](#)
- [Binning Method](#)

Binning Method

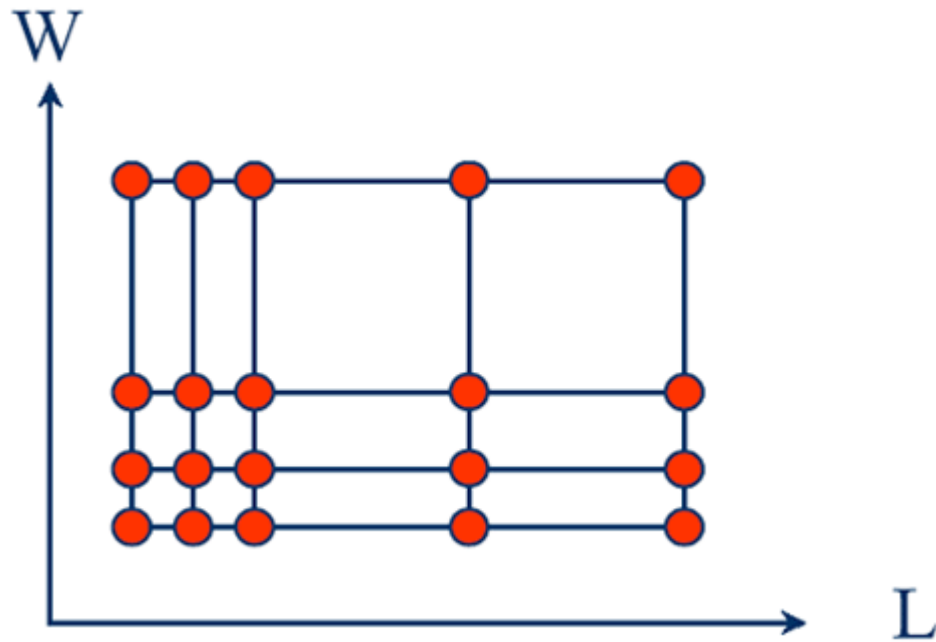
All above discussion is focused on global model. By implementing of single-piece equation with physical major parameters and all geometry parameters built in, global model can cover a wide range of geometry with reasonable accuracy. However, due to some fabrication-related issues and there are occasions the device physics is too complicated to understand, to extract a global model that is accurate over the entire geometry range is often difficult. To improve the accuracy for certain device sizes, bin model parameters are normally used. MBP also provide a complete flow for binning model extraction.

To bin is to calculate geometrical parameters to encompass all sizes of interests, as given below:

$$P = P0 + \frac{LP}{L_{eff}} + \frac{WP}{W_{eff}} + \frac{PP}{L_{eff} \times W_{eff}}$$

P stands for the binned model parameter, P0 is the zero-order term of the parameter, LP, WP and PP are its corresponding three binning parameters, account for the length dependence of P0, the width dependence of P0, and the cross term dependence of both the length and width respectively.

An exemplar binning strategy that breaks devices width-length space into 12 subregions



Not all the parameters are binnable, we list part of non-binnable parameters in Table 12.

Non-binnable I-V model parameters for BSIM3v3

Task Tree	Explanation
TOX	gate oxide thickness
TOXM	Nominal Tox at which parameter are extracted
WINT	Width offset fitting parameter without bias effect
WW	Coefficient of width dependence for width offset
WL	Coefficient of length dependence for width offset
WWL	Coefficient of length and width cross term for width offset
WLN	Power of length dependence of width offset
WWN	Power of width dependence of width offset

Task Tree	Explanation
LINT	Length offset fitting parameter without bias effect
LL	Coefficient of length dependence for length offset
LW	Coefficient of width dependence for length offset
LWL	Coefficient of length and width cross term for length offset
LLN	Power of length dependence of length offset
LWN	Power of width dependence of length offset

A general procedure to generate binning model is as follows:

1. Extract model parameters that cannot be binned, such as LINT, WINT.
2. Extract point models separately for each devices.
3. Create bin model subregions. Extract a complete set of model parameters for each bin. Use Utilities-Binning of MBP to do it.
4. Check the continuity of the parameter at all bin boundaries, since binning may result in discontinuity in device characteristics as the device geometry varies. Optimize P0, PL, PW, and PP for certain parameters if necessary.

NOTE

- As for how to use MBP to generate binning model, refer to **Utilities**.
- Although easier for extraction and more accurate for certain device sizes, binning parameters are un-physical parameters that describe un-physical scaling behavior. It should be only used for special cases.

BSIM3v3 DC Modeling

We present a compact flow in this section to demonstrate pmos global model extraction by using MBP.

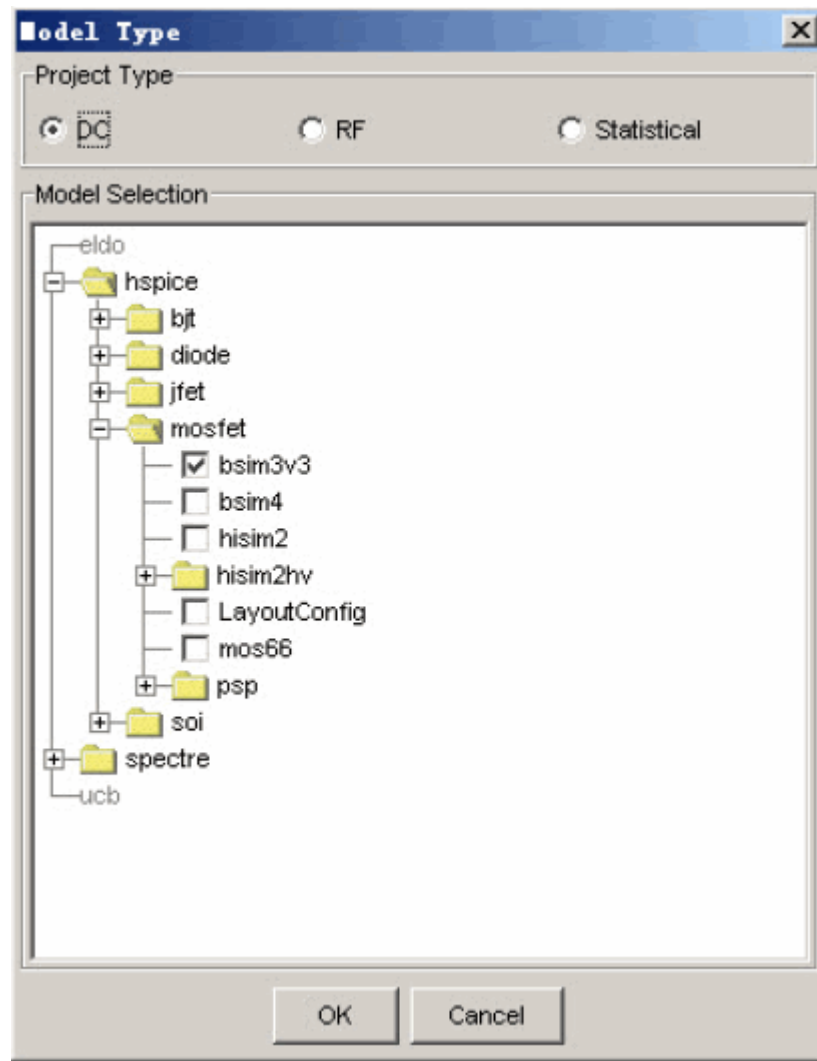
Do Measurements

Refer to **Measurement** for detailed information on how to carry out measurements.

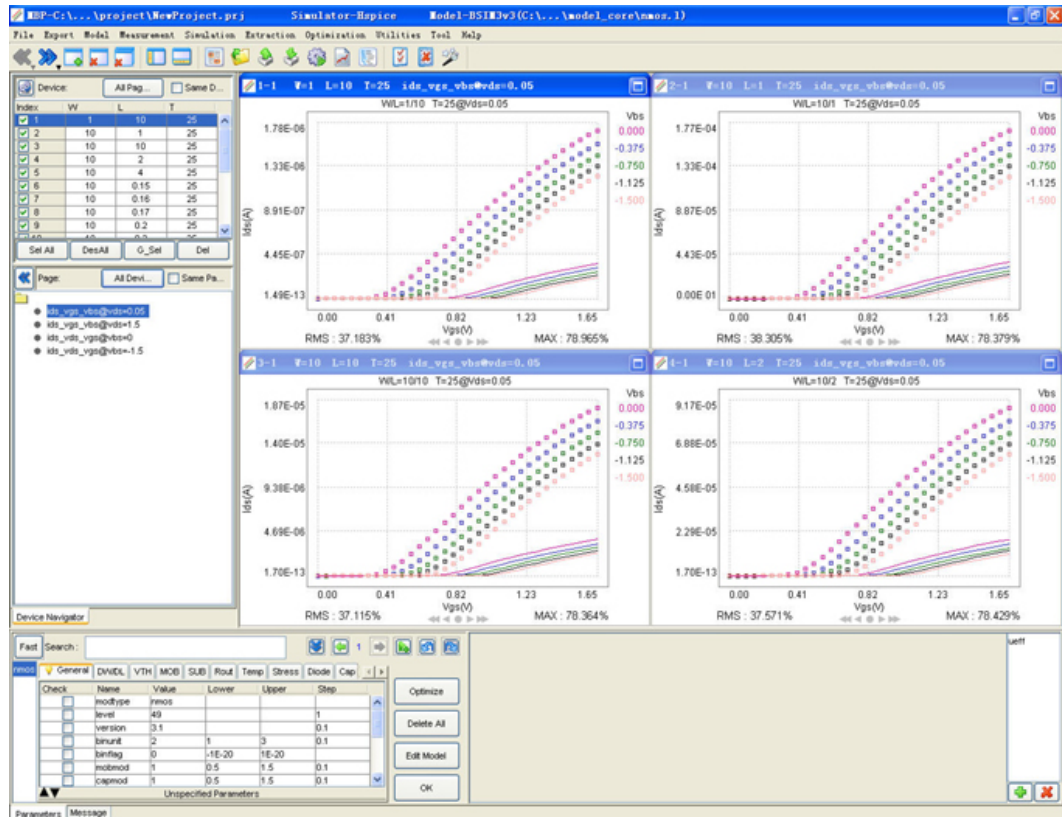
Load Measurement Data

Load all measured data into MBP. Acquaint with device geometry and bias condition through Device Navigator. To be noted, always set right model type before loading data.

Select model type



Load measured data



Check Measured Data

To prevent bad data from impacting subsequent extraction and optimization, it's better to do measurement QA first. Select Extraction > Data Check. See whether these measured data follow the right physical trend. If not, pick out those unreasonable data, do measurements again, or delete it from the Device Navigator.

NOTE

MQA (Model Quality Assurance) is more efficient and powerful in Measurement QA, also developed by Keysight.

Set Process Parameter

Click Model > Set model options to input process parameters. Process parameters affect device characteristics markedly. And since they are referred in extraction, assure their accuracy.

Set process parameters

Use the Task Tree to Extract Initial Model

Select Extraction > Task Tree to launch Task Tree window. Load built-in extract. BSIM3v3 and start auto extraction. After a few minutes, an initial model file with improved fittings is displayed.

The Task Tree extract_complex. BSIM3v3 is applicable to extracting global model. It has certain requirements on measurement data, that is at least 8 devices with proper size distribution is needed. Make sure that measurement data meets requirements, or extraction procedure may even be halted.

The extract_complex. BSIM3v3 includes efforts and experience of our modeling experts. Different built-in Task Trees have different applications, as listed in Table 11. Proper checkpoints set to handle specific situations, you can easily achieve satisfying fittings with performing extraction.

Table 11 Built-in Task Tree for extracting BSIM3v3 model

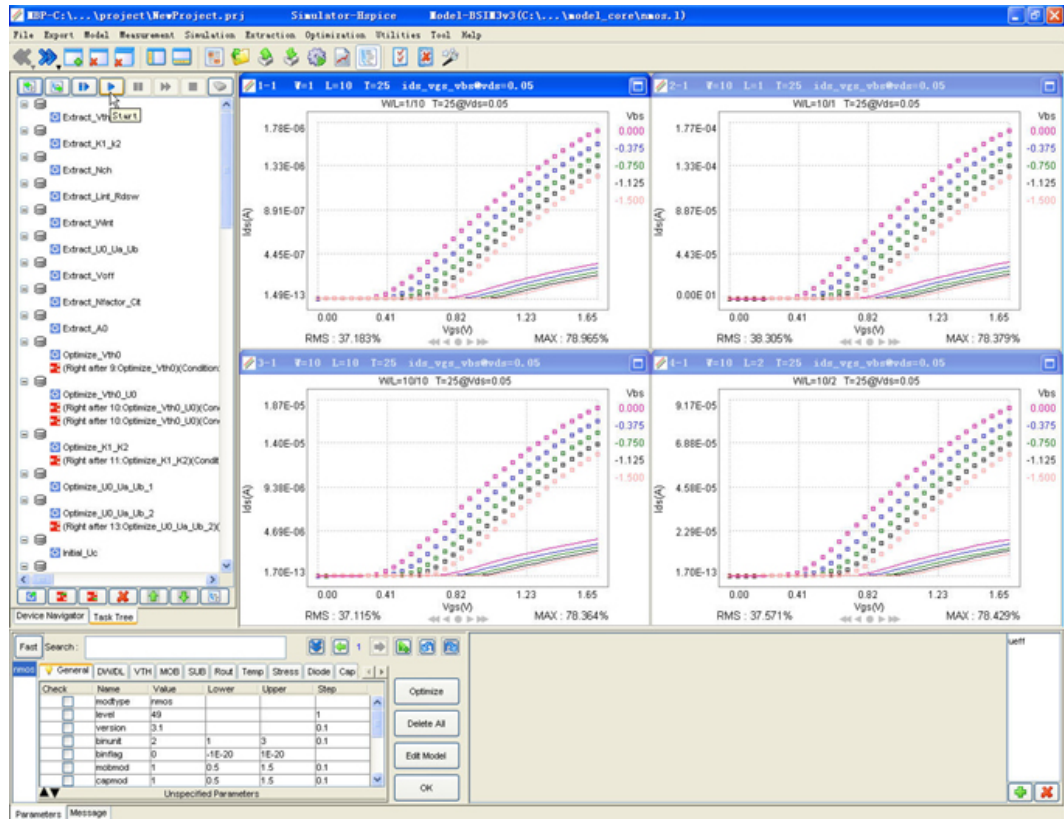
Task Tree	Explanation
autobin. BSIM3V3	or binning model generation. It starts with one point/global model and finishes with full set of point models, determined by geometry sizes of loaded devices.
autobin_temp. BSIM3V3	The supplement to autobin.BSIM4 tree for temperature coefficient extraction.
extract_CggCgc. BSIM3V3	For CV related parameters extraction.
extract_complex. BSIM3V3	For global model extraction. Generic DC parameters are extracted and optimized. Binning parameters are included in final extracted model.
extract_hv. BSIM3V3	Providing a starting model for our built-in HV modeling methodology.

Task Tree	Explanation
extract_nobin. BSIM3V3	For global model extraction. Generic DC parameters are extracted and optimized. Binning parameters are not allowed in this task tree.
LOD_STI. BSIM3V3	For STI effect related parameters extraction.
single device. BSIM3V3	For DC parameters of one device extraction. Since, only one device is available, the result may not be physical and reasonable.
temp.BSIM3V3	The supplement to extract_complex.BSIM4 tree for temperature coefficient extraction.

NOTE To perform extraction by extract.BSIM3v3, all devices are required to be measured on the same bias scheme. Or error may occur during automated extraction in MBP.

Refer to the [Task Tree](#) for running a Task Tree.

Extract the initial model by the Task Tree



Optimization

Subtle adjustment to model is always needed after auto-extraction. You can choose one of the following two ways for optimization:

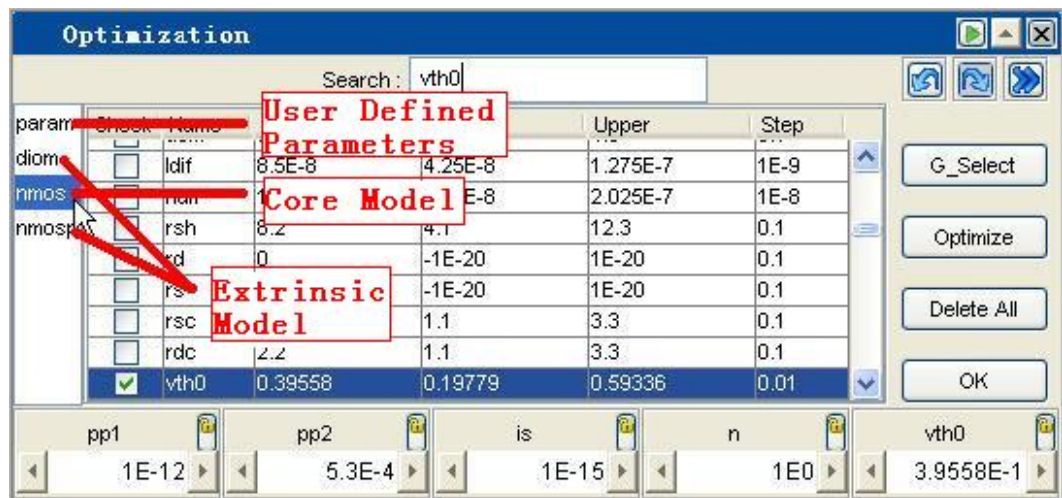
Use the Task Tree for optimization

Actually, reload extracted initial model (to let MBP reset parameter boundary) and run extract. BSIM3v3 again, it probably shows good fittings, since extract. BSIM3v3 also includes the complete flow of optimization. However, each data from different process has its own specialty. To ensure an efficient optimization process, you can establish a new Task Tree from extract.BSIM3v3.

Use the Optimization

Locate the page and region not fitted well, launch Optimization window, optimize parameters dominate selected region. to do it manually or by optimizer, as shown in following figure. By this method, one gets experienced in BSIM3v3 modeling. Refer to the extraction routine for understanding BSIM3v3 model parameters. Refer to [Optimization](#) optimization operation.

Model parameters optimization



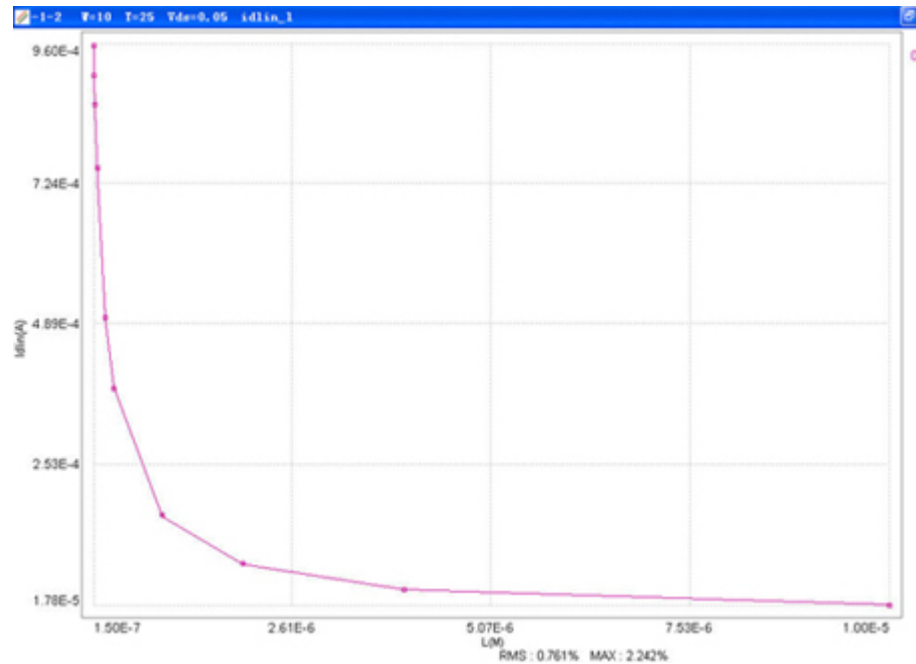
Optimization on IMV pages

MBP provides a useful feature by optimizing model parameters directly on IMV (intermediate variables) pages. An example of Idlin_L is shown in following figure. Apparently, IMV pages help you to know the entire trend and are of more physical meanings. This makes optimization more efficient and more accurate compared to separate IV pages. After MOS data loaded, MBP generates all IMV pages automatically. You can easily select the desired pages from IMV table.

NOTE

The IMV page is displayed on **active** window.

The IMV page of Idlin_L



Extract Temperature coefficient

Above extraction and optimization are for reference temperature. To model temperature behavior, load data taken under high, low, and room temperature together and extract temperature coefficient.

Export Model File

After the model is fine-tuned, you can export the model to a SPICE model file. This model file is compatible with HSPICE. It is recommended to save the model file for every update.

BSIM3v3 Overview

BSIM3 model is the industry-standard MOSFET model for deep-submicron digital and analog circuit designs down to 0.18 m technologies, developed by the BSIM Group at the University of California at Berkeley. It adopts a single-equation to describe device characteristics in various operating regions. This eliminates the discontinuity in the I-V and C-V characteristics. BSIM3 has evolved through three versions. BSIM3v1 and BSIM3v2 contain many mathematical problems so these are largely replaced by a third version, BSIM3v3. BSIM3v3 eventually grew into further variations with minor differences. It accounts for the following physical effect in MOSFET as:

- Short channel effects
- Narrow channel effects
- Non-uniform doping effects in both vertical and lateral directions
- Mobility reduction due to vertical field

- Bulk charge effect
- Velocity saturation
- Drain-induced barrier lowering (DIBL)
- Channel length modulation (CLM)
- Substrate current induced body effect (SCBE)
- Subthreshold conduction
- Source/drain parasitic resistances

More information on BSIM3 can be found at: <http://www-device.eecs.berkeley.edu/bsim3>. The version described in this section is BSIM3v3.2.

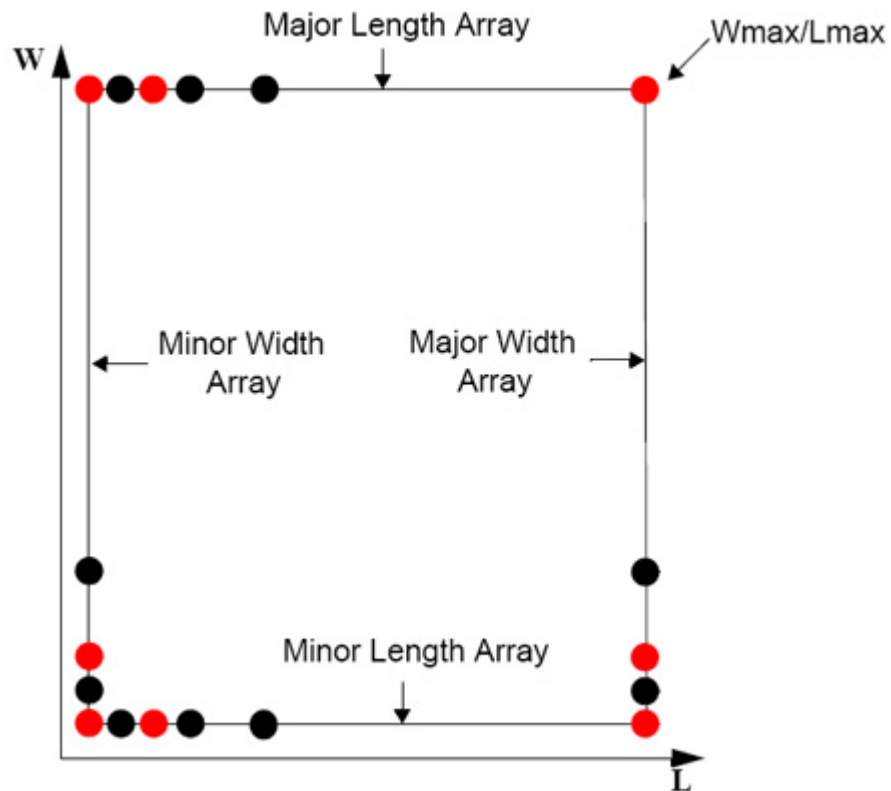
Device Requirements

BSIM3v3 is a scalable model that its model parameters apply to transistors of various lengths and widths. In other words, accurately modeled BSIM3v3 model correctly predicts device behavior in all operating regions for all the allowed device geometries called global model. For a proper extraction of global model, measurement data from a set of devices with different size is required. Devices

1. Wmax/Lmax: such as 10/10, used to extract to parameters which are independent of short and narrow channel effects and parasitic resistance.
2. Major Length Array: such as 10/10, 10/1, 10/0.7, 10/0.5, 10/0.3, 10/0.25. These devices are used to extract parameters that are related to the short channel effects.
3. Major Width Array: such as 10/5, 1/5, 0.5/5, 0.4/5, 0.3/5. These devices are used to extract parameters that are related to the narrow width effects.
4. Minor Length Array: such as 0.3/5, 0.3/1, 0.3/0.7, 0.3/0.5, 0.3/0.3, 0.3/0.25. These devices are used to adjust short channel effects related parameters.
5. Minor Width Array: such as 10/0.25, 1/0.25, 0.5/0.25, 0.4/0.25, 0.3/0.25. These devices are used to adjust narrow width effects related parameters.

For MBP, the minimum set of devices should comprise 8 devices for extraction, marked with red color in following figure. Four of them locate at the corner, including one device with long, wide channel (Wmax/Lmax); one device with short, narrow channel (the smallest device in designed process); one device with long, narrow channel and one device with short, wide channel. The other four devices locate on the major length array, major width array, minor length array, and minor width array respectively. They should embody short and narrow effects in I-V characteristics more or less. More devices with different sizes surely contribute to more accurate extraction.

An example of device geometry for DC parameter extraction



Extraction and Optimization of BSIM3v3 Model

Actually, modeling engineer could never take extraction and optimization apart. The optimization is often concomitant with the extraction. So we integrate them into a unified procedure. This section includes two parts:

- Suggested extraction and optimization procedure
- One example on BSIM3v3 model extraction and detailed explanation on referred modules.

Extraction Routine

To extract BSIM3v3 parameters correctly, we recommend you to follow the steps in tables below

Table 1 Proc.1: Input process parameter

Step	Device	Curves	Parameter
1			Measure CV to get TOX
2			Input process parameter such as TOX, XJ, NCH, Rsh, etc.

Table 2 Proc. 2: Basic parameters for Vth, subthreshold region and mobility

Step	Device	Curves	Extracted Parameter
3	Wmax/Lmax	Lin Gum. @Vbs=0	VTH0, U0, UA, UB, VOFF, CIT, NFACTOR, NCH, NSUB, VBX
4		Lin Gum. All Vbs	UC
5		Lin Gum. All Vbs	repeat 3-4
6		Lin Gum. All Vbs	K1, K2
7		Lin Gum. @Vbs=0	VOFF, CIT, NFACTOR, VTH0, U0, UA, UB
8		Lin Gum. All Vbs	UC

Table 3 Proc. 3: Narrow width related parameters

Step	Device	Curves	Extracted Parameter
9	Wmin/Lmax	Lin Gum. @Vbs=0	K3, W0, WINT
10	Wmin/Lmax Wmax/Lmax	Lin Gum. @Vbs=0	W0, WINT, K3, VTH0, U0
11	Major WA	Lin Gum. @Vbs=0	K3, VTH0, WINT, WW, U0
12		Lin Gum. All Vbs	K3B, K1, K2, K3 VTH0, U0, WINT, WW
13		Lin Gum. @Vbs=0	DWG
14		Lin Gum. All Vbs	VTH0, U0, UA, UB, UC, DWG, WINT, WW, K3, DWB, K3B

Table 4 Proc. 4: Short Channel related parameters

Step	Device	Curves	Extracted Parameter
15	Wmax/Lmin	Lin Gum. @Vbs=0	LINT

Step	Device	Curves	Extracted Parameter
16	Major LA	Lin Gum. All Vbs	DVT0, DVT1, DVT2, NLX, VTH0, K1, K2
17		Lin Gum. @Vbs=0	LINT, LL, U0
18		Lin Gum. All Vbs	LINT, LL, DVT0, DVT1, DVT2, NLX, VTH0, K1, K2, LINT, U0
19	Wmax /Lmax	Lin Gum. @Vbs=0	VOFF, CIT, NFACTOR
20	Major LA	Lin Gum. @Vbs=0	LVOFF, CDSC
21		Lin Gum. All Vbs	CDSCB, VTH0, K1, K2, DVT0, DVT1, DVT2, NLX, LINT, LL, U0, UC

Table 5 Proc. 5: Series Resistance related parameters

Step	Device	Curves	Extracted Parameter
22	Wmax /Lmin	Lin Gum. @Vbs=0	RDSW, PRWG
23	Major LA	Lin Gum. @Vbs=0	RDSW, PRWG, A1, A2, VTH0, K1, K2, DVT0, DVT1, NLX, LINT, LL, U0
24	Wmax /Lmin	Lin Gum. All Vbs	PRWB
25	Major LA	Lin Gum. All Vbs	PRWB, UC, DVT2
26	Minor LA Minor WA	Lin Gum. All Vbs	DVTOW, DVT1W, DVT2W, WL, LW
27		Lin Gum. @Vbs=0	PRDWS, PPRWG

Table 6 Proc. 6: Narrow width related parameters (II)

Step	Device	Curves	Extracted Parameter
28	Major WA	Lin Gum. @Vbs=0	VTH0, U0, UA, UB, DWG, WINT, WW, K3

Step	Device	Curves	Extracted Parameter
29		Lin Gum. All Vbs	K3B, K1, K2, UC, DWB
30	Major LA	Lin Gum. @Vbs=0	VTH0, U0, UA, UB, K1, K2, DVT0, DVT1, LINT, LL, NLX, RDSW, PRWG
31	Wmax /Lmax	Lin Gum. @Vbs=0	VOFF, CIT
32	Major LA	Lin Gum. @Vbs=0	LVOFF, CDSC
33		Lin Gum. All Vbs	CDSCB, UC, PRWB
34	Minor LA Minor WA	Lin Gum. All Vbs	DVT0W, DVT1W, DVT2W, WL, LW
35		Lin Gum. @Vbs=0	PRDSW

Table 7 Proc. 7: Parameters for subthreshold region (II)

Step	Device	Curves	Extracted Parameter
36	Major LA	Sat Gum. @Vbs=0	CDSCD, DSUB, ETA0
37		Sat Gum. All Vbs	CDSCD, DSUB, ETA0, ETAB

Table 8 Proc. 8: Parameters for I-V model

Step	Device	Curves	Extracted Parameter
8		Sat I-V @Vbs=0	VSAT, A0, LA0, AGS, LAGS
39		Sat I-V All Vbs	KETA, LKETA,
40	Major WA	Sat I-V @Vbs=0	B0

Table 9 Proc. 9: Parameters for Rout

Step	Device	Curves	Extracted Parameter
41	Major LA	Gds@Vbs=0	PVAG, DELTA, PCLM, PDIBLC1, PDIBLC2, DROUT

Step	Device	Curves	Extracted Parameter
42		Sat Gum. All Vbs	CDSCD, DSUB, ETA0, ETAB
43		Sat I-V @Vbs=0	A0, LA0, AGS, LAGS, VSAT
44		Sat I-V All Vbs	KETA, LKETA, B0
45		Gds@Vbs=0	PVAG, DELTA, PCLM, PDIBLC1, PDIBLC2, DROUT, VSAT, LPVAG, LDELTA, LPCLM, LPDIBLC2, RDSW, WR, PRWG
46		Gds All Vbs	PDIBLCB, PRWB

Table 10 Proc. 10: Further optimization

Step	Device	Curves	Extracted Parameter
47			repeat 42
48		Gds @Vbs=0 Sat I- V@Vbs=0	LPVAG, LDELTA, LPCLM, LPDIBLC2, PVAG, DELTA, PCLM, PDIBLC1, PDIBLC2, DROUT, VSAT, A0, LA0, AGS, LAGS
49			repeat 36-44
50			repeat 28-36
51			repeat 48
52			repeat 37-46
53			repeat 48

NOTE

This procedure is for reference. In some cases, no need or not possible to extract all the parameters, leave them as default value and skip corresponding steps. Also, you can reorganize it, depending on your experience and request.

Measurement Conditions

We recommend you to take the following data type for each device:

- Data Sets
 - Vdsmax=Vdd*1.1
 - Vbsmax=-Vdd*1.1
 - Vgsmax=Vdd*1.1

- Linear Characteristics
 $V_{gs}=0$ to V_{gsmax} by 0.1 (or 0.05 if V_{dd} in 1.2V)
 $V_{ds}=0.05$ (or 0.1)
 $V_{bs}=0$ to V_{bsmax} in 4 or 5 steps
- Saturation Characteristics
 $V_{gs}=0$ to V_{gsmax} by 0.1 (or 0.05 if V_{dd} in 1.2V)
 $V_{ds}=V_{dsmax}$
 $V_{bs}=0$ to V_{bsmax} in 4 or 5 steps
- I-V Characteristics
 $V_{ds}=0$ to V_{gsmax} by 0.1 (or 0.05 if V_{dd} in 1.2V)
 $V_{gs}=V_{th}$ to V_{gsmax} in 5 steps
 $V_{bs}=0, V_{bsmax}$
- Saturation I-V Characteristics
 $V_{ds}=V_{gs}=0$ to V_{gsmax} by 0.1 (or 0.05 if V_{dd} in 1.2V)
 $V_{bs}=0$ to V_{bsmax} V_{gsmax} in 4 or 5 steps
- Substrate Current
 $V_{ds}=0.5*(V_{dd}*1.1), 0.75*(V_{dd}*1.1), 1*(V_{dd}*1.1)$
 $V_{gs}=0$ to V_{gsmax} by 0.1 (or 0.05 if V_{dd} is 1.2V)
 $V_{bs}=0$

NOTE

These bias schemes are recommended but not necessary. You can take your own measurement setup, e.g. setting $V_{dsmax}=V_{dd}$ and taking more steps in I-V characteristics.

Diode Modeling

Diode is the most fundamental non-linear element of electronic circuits. This chapter describes how to model a diode. It provides information on the following topics:

- Diode DC Modeling
- Diode C-V Modeling

Diode C-V Modeling

The C-V behavior of diode is modeled using a space charge capacitance (dominant at reverse biasing) and a diffusion capacitance (dominant at forward bias). The later is usually measured using network analyzer, we provide guidelines on space charge capacitance modeling in this section.

Measuring C-V Data

Perform C-V measurement according to the following steps:

Start CV measurement

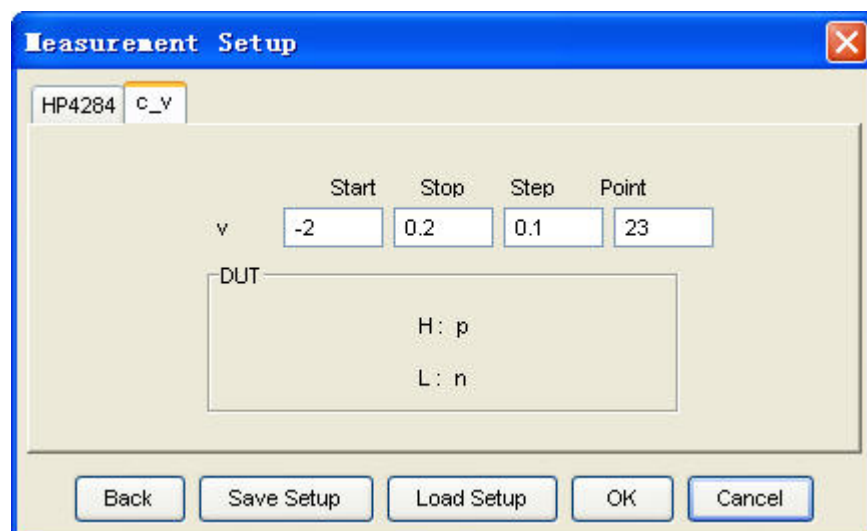
Go to Measurement > CV, the Measurement Monitor appears.

Specify the file name and directory for saving measured data

Set CV measurement

Click Setup, set Device type as D(diode), add devices for measurement. The bias condition is required to cover not only reverse region but also forward region, typically from 2 to 0.2V, as shown in following figure.

Set bias condition for diode CV measurement



Do zero

Do Zero before starting CV measurement to eliminate open parasitic capacitance.
Put probe heads a little above diode pads when doing zero.
Click Measure to execute CV measurement

Save measured data

Refer to Chapter 6 for detailed Measurement operation.

NOTE

As mentioned in Diode I-V Modeling, measure the two diode devices with different sizes for both area related parameters and side wall related CV parameters.

AC Diode Model Extraction

Follow the steps below to perform CV extraction:

Load measured C-V data

First, select model type as diode. Load C-V data of both area diode and finger diode together.

Extract C-V parameters

Define the space charge capacitance of the diode using:

$$C = CJ_{eff} \cdot \left(1 - \frac{V}{PB}\right)^{-MJ} + CJSW_{eff} \cdot \left(1 - \frac{V}{PHP}\right)^{-MJSW}$$

CJ, PB(VJ), and MJ are junction zero-bias depletion capacitance, junction built-in potential, and junction exponential factor respectively; CJSW, PHP, and MJSW are corresponding parameters for sidewall junction.

During extraction, for area diode, formula is reduced to:

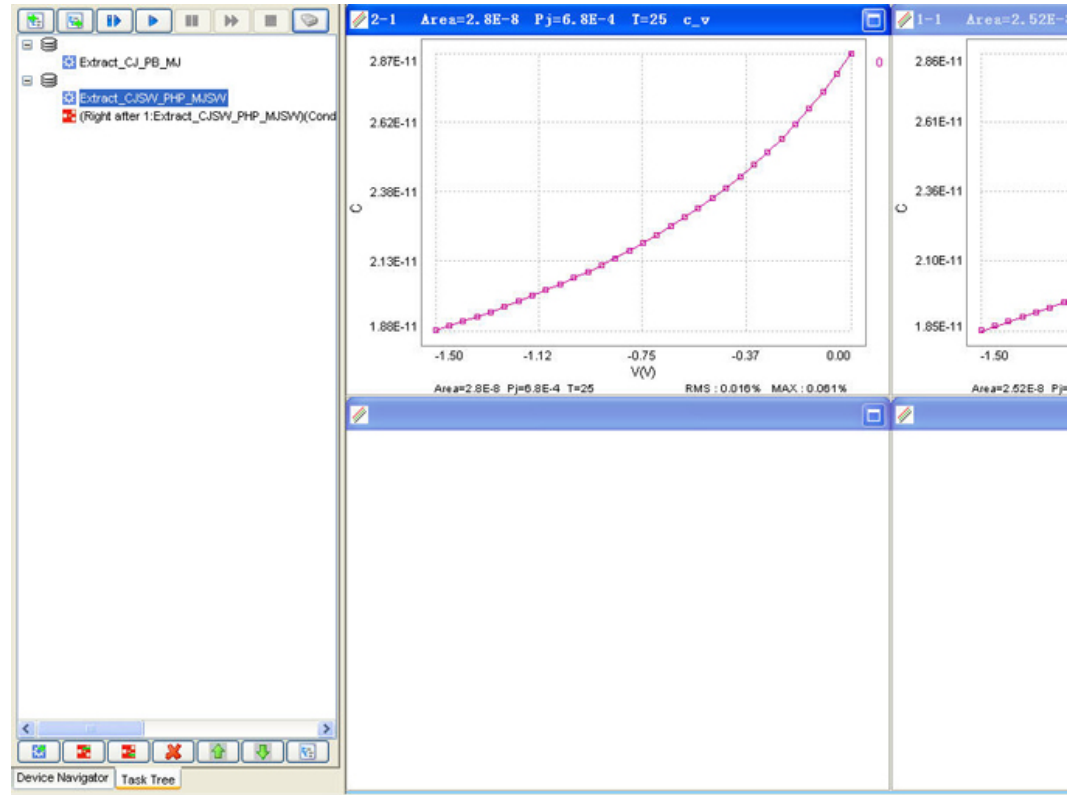
$$C \approx CJ_{eff} \cdot \left(1 - \frac{V}{PB}\right)^{-MJ} = CJ \cdot AREA \cdot M \cdot \left(1 - \frac{V}{PB}\right)^{-MJ}$$

For finger diode, formula is reduced to:

$$C \approx CJSW_{eff} \cdot \left(1 - \frac{V}{PHP}\right)^{-MJSW} = CJSW \cdot PJ \cdot M \cdot \left(1 - \frac{V}{PHP}\right)^{-MJSW}$$

Bring up the Task Tree window, load diodeCVExtractor.Diode as task tree file and execute extraction as shown below. All the above CV parameters are well extracted.

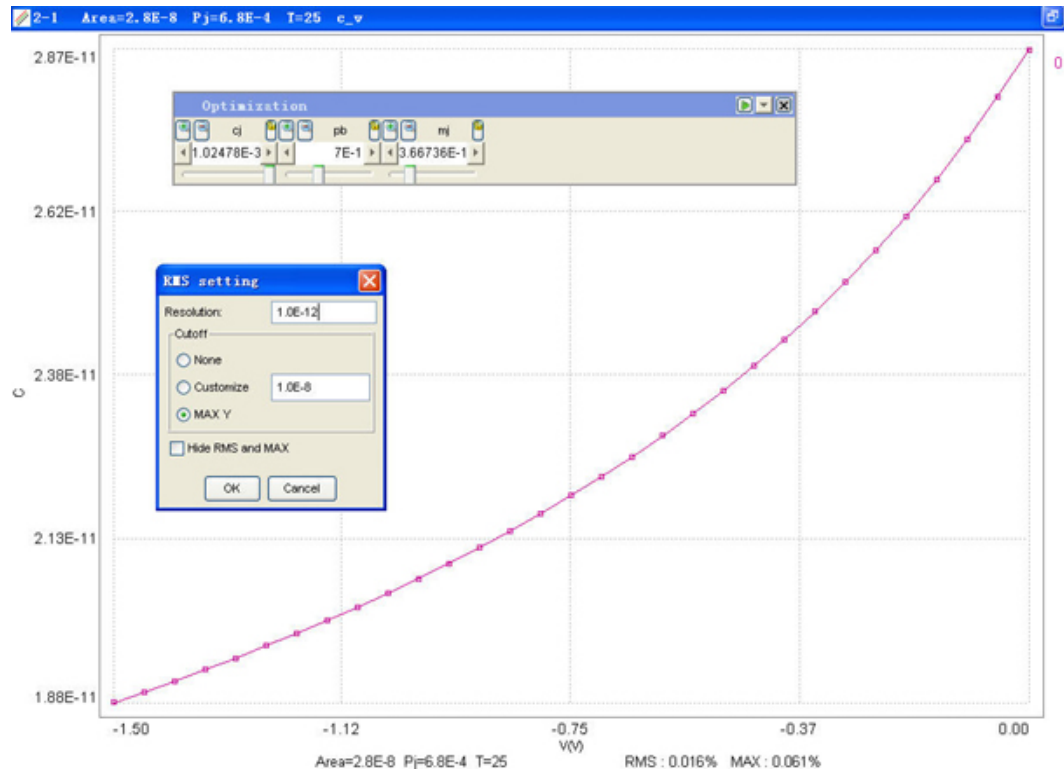
Diode C-V parameters auto extraction



Optimize C-V parameters

Our optimizer has good effects on the CV parameters optimization. Select CV parameters and data region (usually the whole region if no bad data exist) to do auto optimization, as shown in following figure. However, since junction capacitance is always small, note your RMS settings. Optimizer does not work with zero RMS, as we explained in [Equation Viewer](#).

CV optimization with our optimizer



Export model file

Save model file to the designated location.

NOTE

Equipped Task Tree diodeExtractor. Diode combines DC and AC extraction for diode devices.

Diode DC Modeling

Measuring I-V Data

The I-V measurements of the diode includes forward part and reverse part.

Forward I-V

Normally the forward bias is from 0 to 1.2V or to a voltage the current reaches expected value, such as 10 mA.

Reverse I-V

To model reverse characteristics, applying voltage till to the breakdown voltage (BV) or larger.

NOTE

- Unlike MOS and BJT measurements, bias conditions for diode measurement mainly depend on its operation region and ones experience.

- Ideally, at least two geometries for diode measurements: one area diode with a large area to perimeter ratio and one finger diode with a large perimeter to area ratio must be present. If only one diode is available, extract parameters related to area and set perimeter related parameters as default.

Diode DC Model Extraction

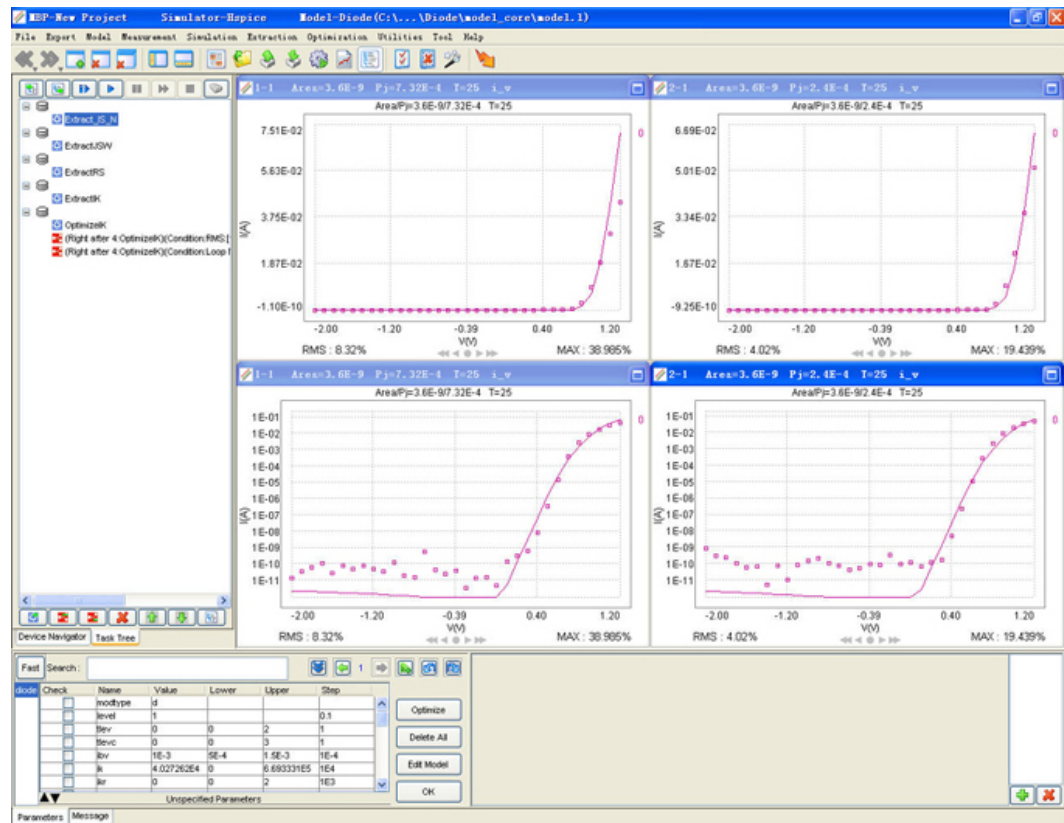
Do the following to perform DC extraction:

Load measured I-V data

In the first step, select model as diode. And then, load both area diode data and finger diode data, if available, into MBP. Extract DC parameters

Shipped diodeDCExtractor.Diode is for modeling diode DC behavior. Load diodeDCExtractor.Diode into MBP to extract DC parameters, as illustrated in following figure.

Diode DC parameters auto extraction

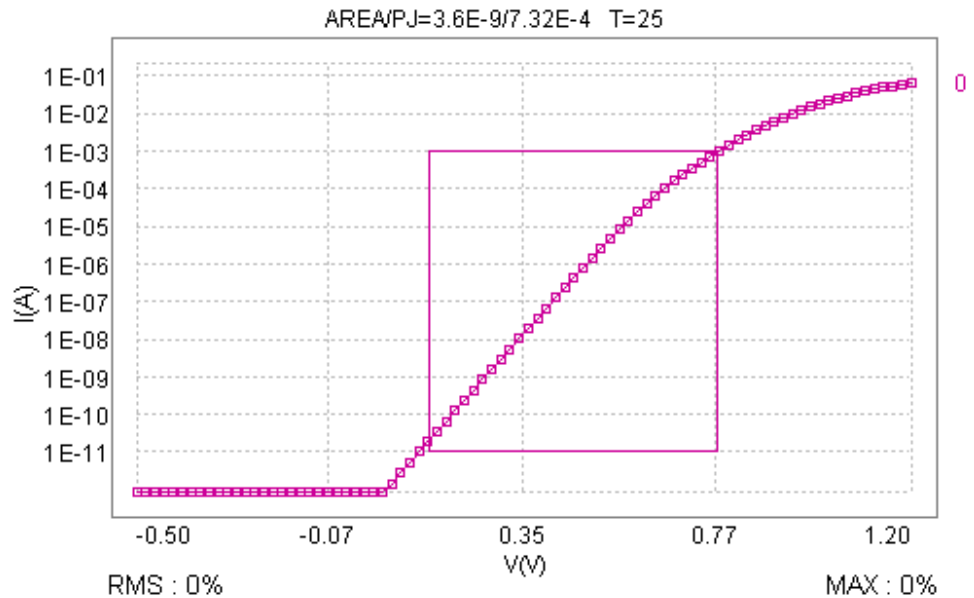


We illustrate extraction procedure of diodeDCExtractor.Diode as follows:

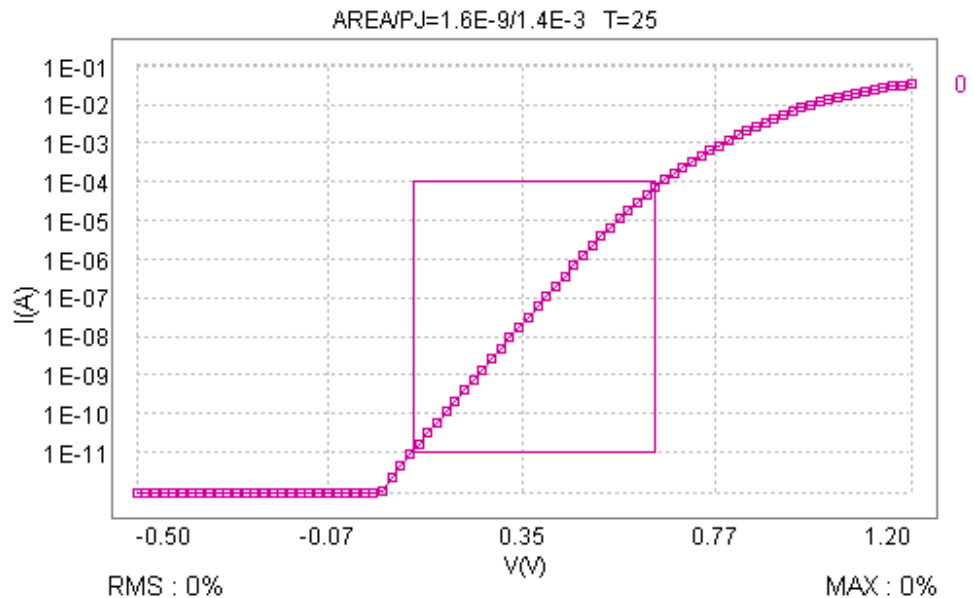
- Extract IS and N, from linear part of area diode I-V plot, see [Figure: Extract IS and N](#).
- Extract JSW, from linear part of finger diode I-V plot, see [Figure: Extract JSW](#).

- Extract RS, from high current part of I-V plot, where current deviates from ideal exponential curve due to series resistance in the diode and high injection effect as well, see [Figure: Extract RS](#).
- Extract IK, from transition part of I-V plot, where current begins to deviate from ideal exponential curve due to high injection effect and series resistance in the diode as well, see [Figure: Extract IK](#).

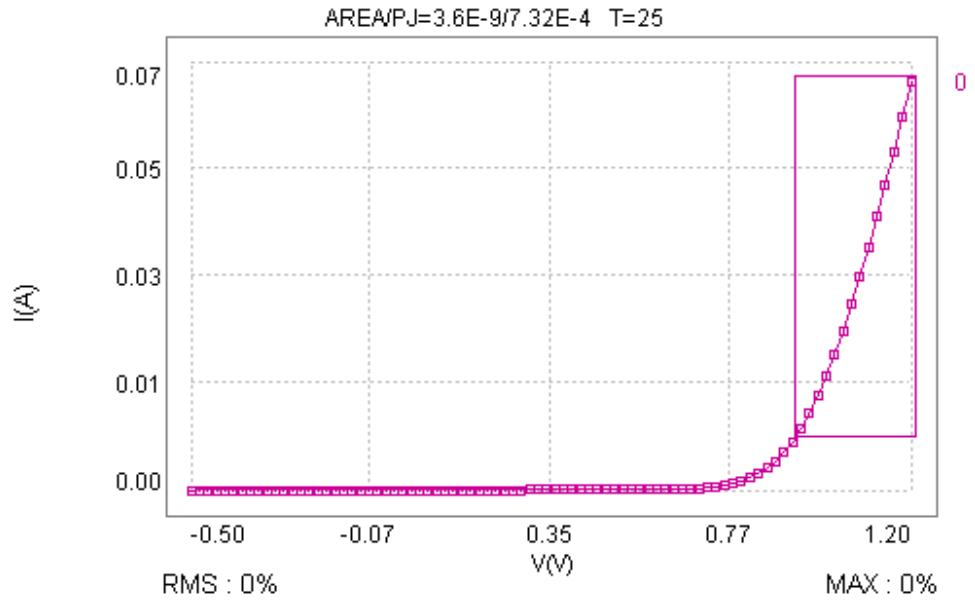
Extract IS and N



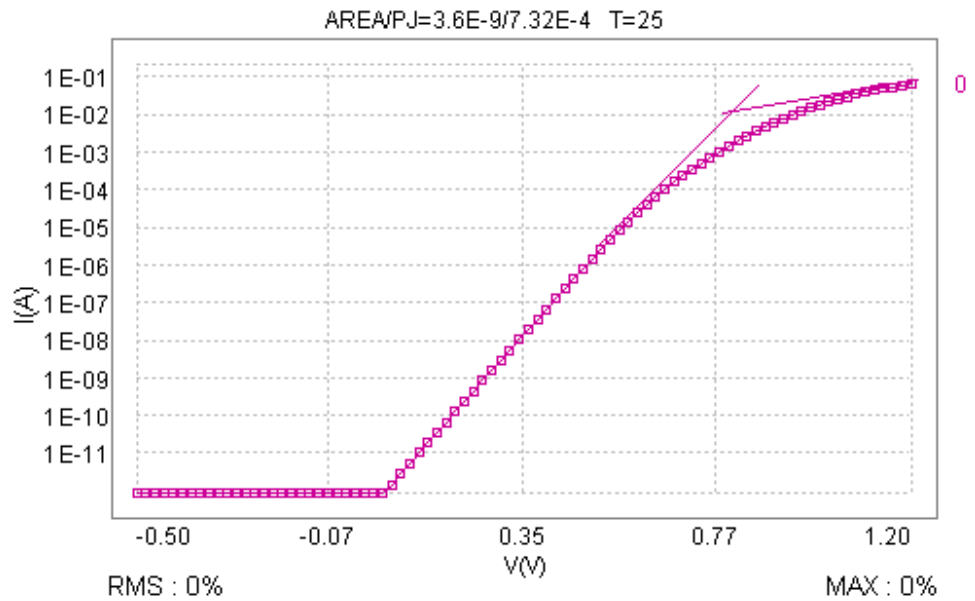
Extract JSW



Extract RS



Extract IK



Optimize DC parameters

Change the plot to the proper scale to view effect on parameters more clearly, such as X-Lin, Y-Log for optimizing IK. As shown in Figure-Diode I-V parameters optimization, open Optimization window, select parameters and target data points and then let the optimizer to do optimization, until we get the best fitting results.

Export model file Save the model file to the designated location.

Diode I-V parameters optimization

Optimization

Search: vth0

param	Check	Name	Value	Lower	Upper	Step
diom	<input type="checkbox"/>	ldif	8.5E-8	4.25E-8	1.275E-7	1E-9
nmos	<input type="checkbox"/>	nan	1E-8	2.025E-7	1E-8	1E-8
nmosp	<input type="checkbox"/>	rsh	8.2	4.1	12.3	0.1
	<input type="checkbox"/>	rd	0	-1E-20	1E-20	0.1
	<input type="checkbox"/>	rs		-1E-20	1E-20	0.1
	<input type="checkbox"/>	rsc	1.1	3.3	0.1	0.1
	<input type="checkbox"/>	rdc	1.1	3.3	0.1	0.1
	<input checked="" type="checkbox"/>	vth0	0.39558	0.19779	0.59336	0.01

pp1 1E-12 pp2 5.3E-4 is 1E-15 n 1E0 vth0 3.9558E-1

Buttons: G_Select, Optimize, Delete All, OK

Gummel-Poon BJT Modeling

This chapter describes DC modeling of bipolar transistors (BJT) in MBP including the following parts:

- AC Measurement of BJT
- DC Measurement of BJT
- Introduction to Gummel-Poon model
- Gummel-Poon AC Model Extraction
- Gummel-Poon DC Model Extraction

AC Measurement of BJT

BJT Junction Capacitance Data Set

In this section, we only discuss measuring space charge capacitors of BJT.

- C_{be} (capacitance of B-E junction) versus V_{be}
- C_{bc} (capacitance of B-C junction) versus V_{bc}
- C_{cs} (capacitance of C-S junction) versus V_{cs} if Vertical BJT
- C_{bs} (capacitance of B-S junction) versus V_{bs} if Lateral BJT

All these measurements follow the similar scheme, that is the junction is reverse biased, and recommended voltage range is 2V, 0.2V with step=0.1V.

Measuring C-V Data

Refer to the guidelines of measuring I-V data.

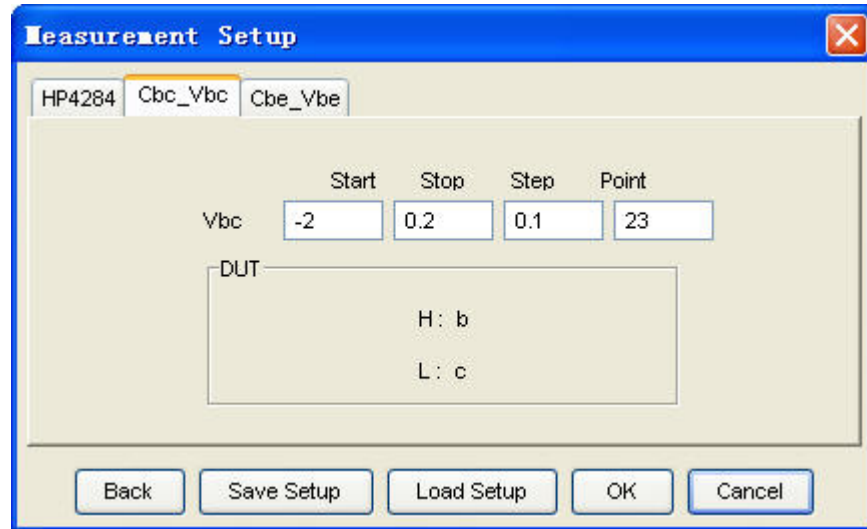
Start CV measurement

Go to Measurement > CV, Measurement Monitor appears.

Specify the file name and directory for saving CV data Set CV measurement

Differ from IV measurement, it is not required to configure node assignments. As seen in following figure, simply connect high port and low port to right node according to the instructions.

Instrument connection for BJT CV measurement

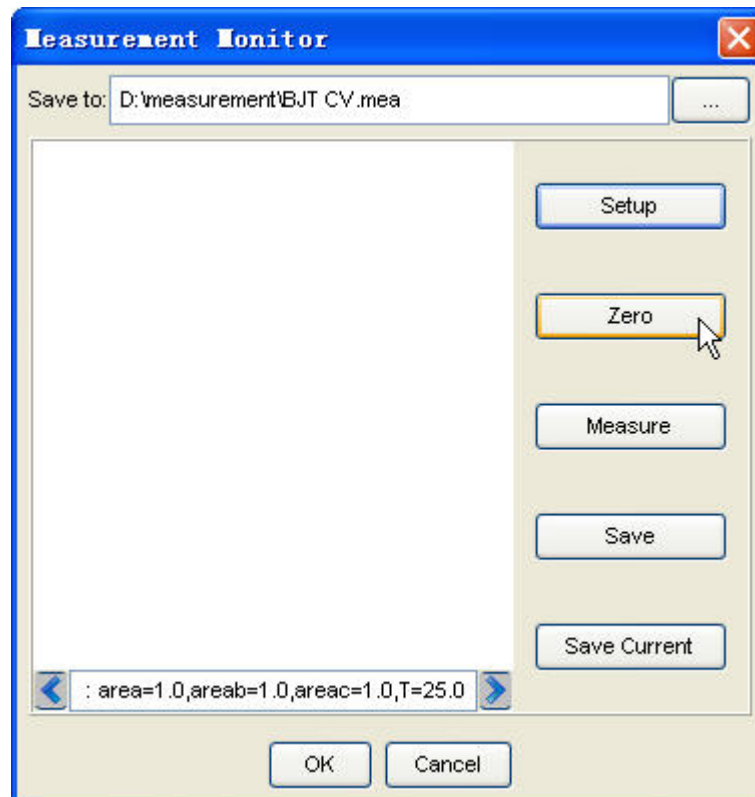


Do Measurements

To compensate parasitic capacitance of the measuring system, open calibration, i. e. zero, is recommended to be done before performing CV measurement, as shown in following figure. Then click Measure to execute CV measurement.

Strictly speaking, zero is recommended to be done before each measurement, not only before a series of measurements.

Zero calibration

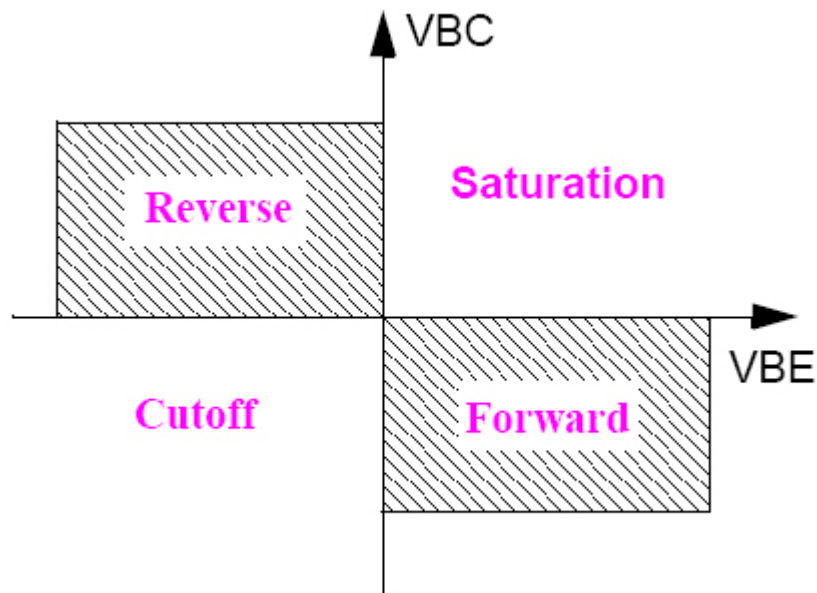


DC Measurement of BJT

Standard BJT Data Sets

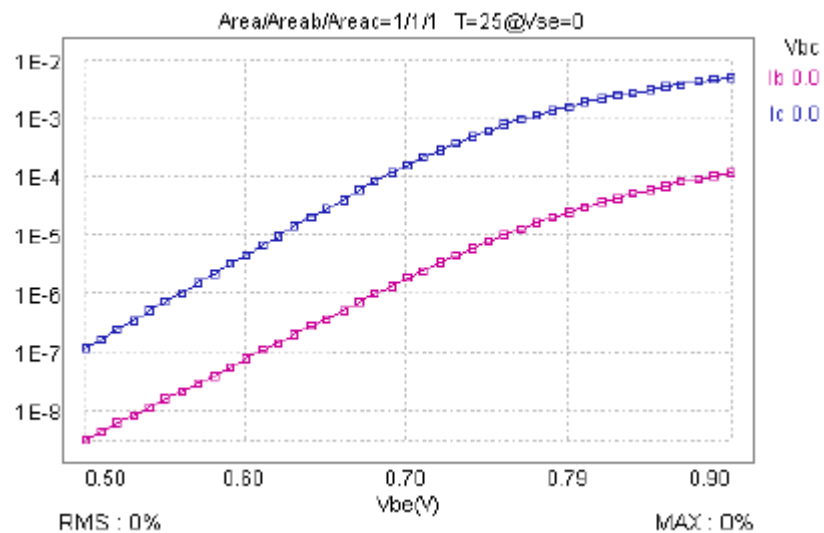
There are four operating modes of a BJT as shown in [Figure: Operating modes of a npn BJT](#). The bias conditions covers the forward and reverse region for DC extraction. Furthermore, the flyback method is proposed to measure R_E and R_C .

Operating modes of a npn BJT

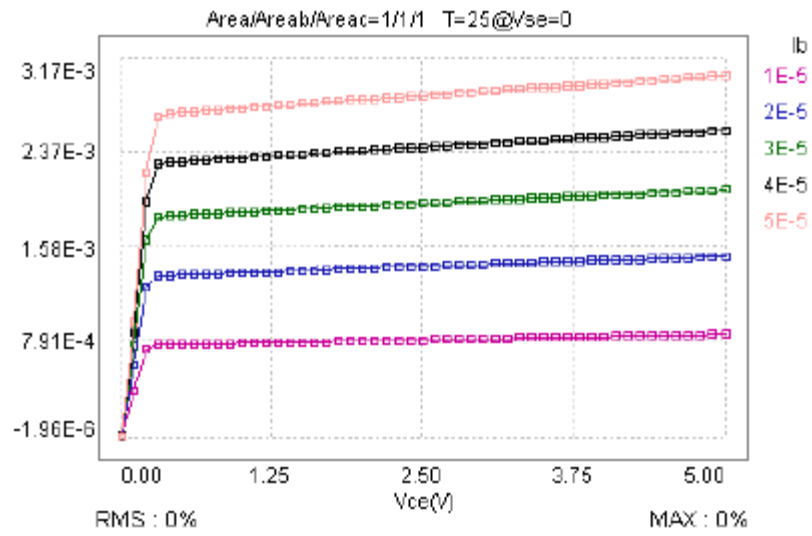


Forward measurement

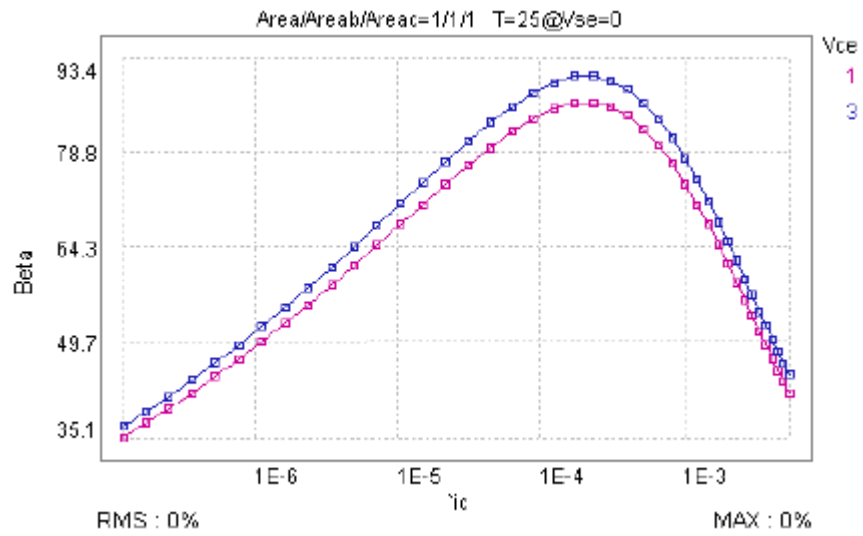
- Forward Gummel: Measure I_c and I_b , sweep V_{be} for $V_{bc}=0$, as shown below. An exemplar Forward Gummel



- Forward Output: Measure I_c , sweep V_{ce} for each discrete value of I_b , as shown below. An exemplar Forward Output



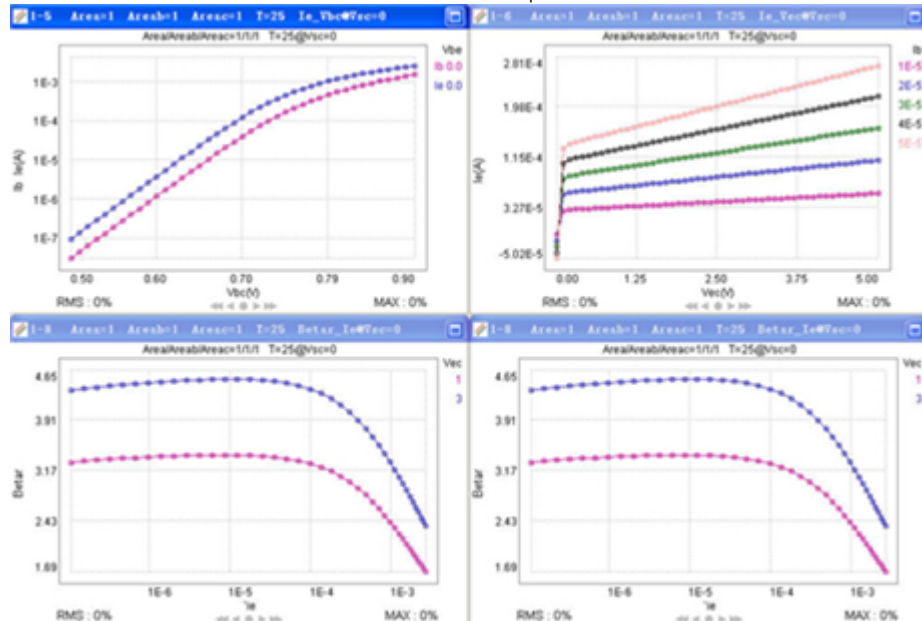
- Forward Beta: Measure Beta, sweep V_{be} for each discrete value of V_{ce} , as shown below. An exemplar Forward Beta



Reverse measurement

- Forward Beta: Measure reverse Beta, sweep V_{bc} for each discrete value of V_{ec}

To do reverse measurement, simply exchange emitter and collector node and take the similar bias scheme. We show a sample with all reverse characteristics as shown below. An example of reverse characteristics



Re flyback

- Set $I_c=0$
- Sweep I_b
- Measure V_{ce} versus I_b
- R_e can be extracted as the slope of V_{ce} vs I_b plot

Rc flyback

- Set $I_e=0$
- Sweep I_b
- Measure V_{ce} versus I_b
- R_c can be extracted as the slope of V_{ce} vs I_b plot

NOTE

The bias condition for BJT measurement is up to device characteristics and IC designers interest as well. Focus on those regions where BJT is operated.

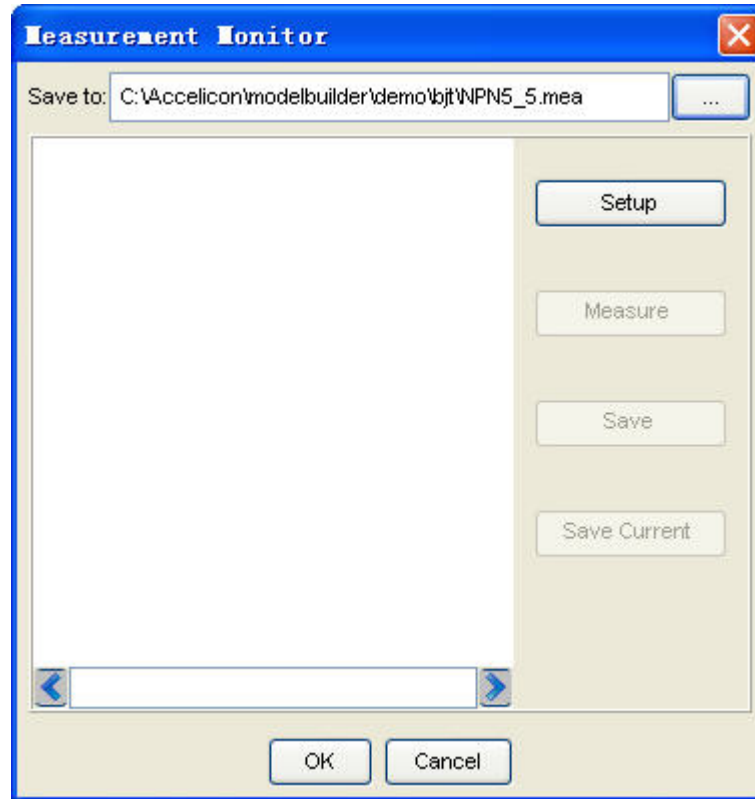
Measuring I-V Data

Follow the following steps to perform measurement:

Start IV measurement

Go to Measurement > IV, Measurement Monitor pops up. Specify the file name and directory for saving measured data.

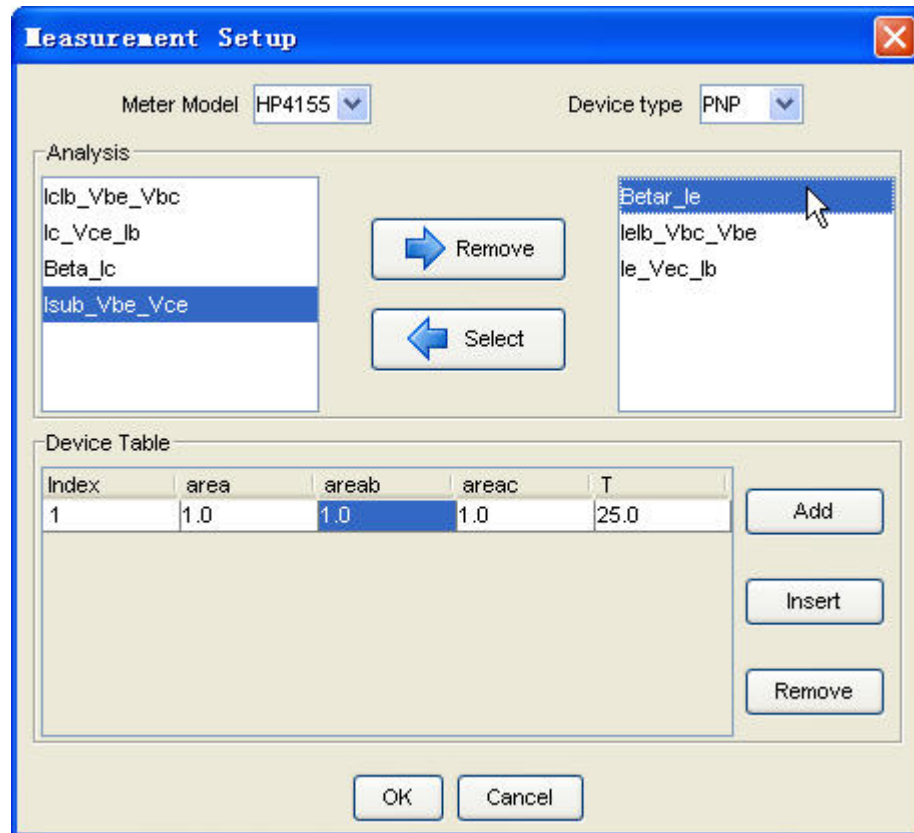
Measurement monitor window



Set DC measurement

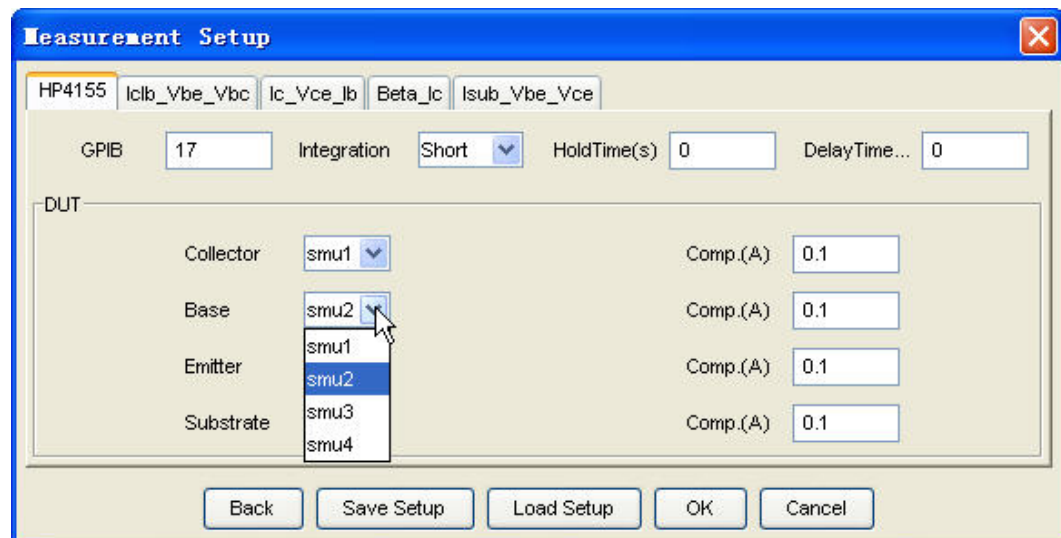
Click Setup, select the Device type (nnp or pnp) and Meter model respectively. Select the I-V curve to measure in the Analysis frame, then click Add or Insert to add device, as illustrated in following figure, click OK to confirm the setup. Measurement Setup window appears.

General Setup window

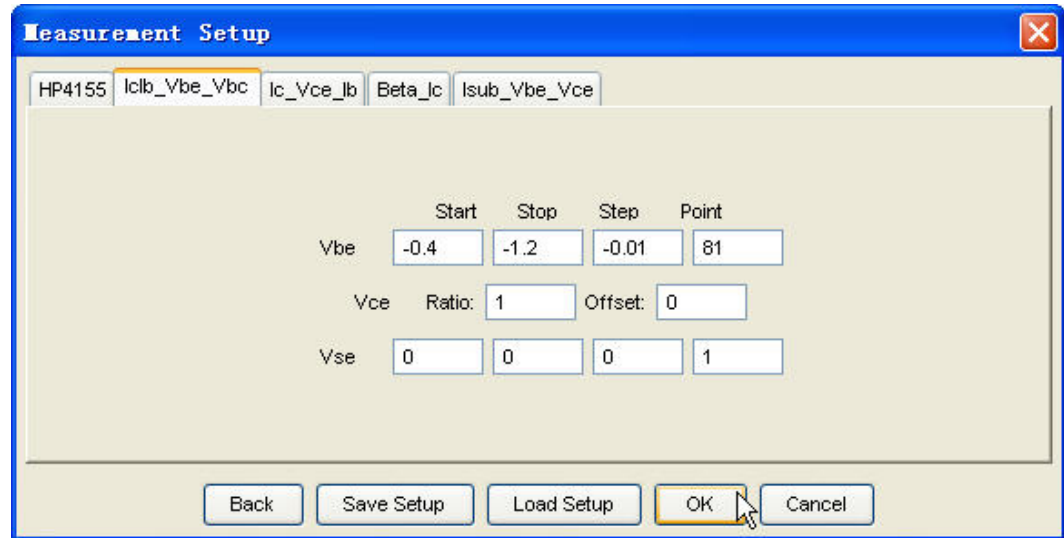


Set connection (Figure: Connection setup) and bias conditions (Figure: Bias condition setup), respectively.

Connection setup



Bias condition setup



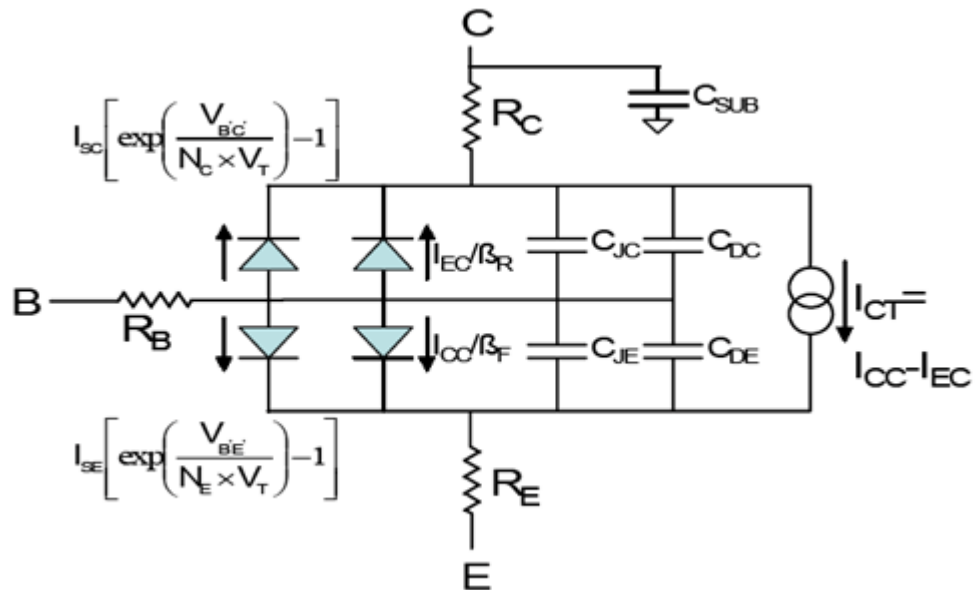
Do measurements

Click Measure button in Measurement Monitor to execute IV measurement. Refer [Measurement](#) for detailed Measurement setup and operation.

Introduction to Gummel-Poon model

Gummel and Poon developed the Gummel-Poon model. It has been the industry standard BJT model for more than 20 years. By introducing the integral charge control concept, this improved model adds several important features to the famous Ebers-Moll model. It inherently considers the base width modulation (Early effect) through using the normalized majority base charge. Further on, a current-dependent base resistance R_B was introduced. Later, the model was extended to include the Kirk effect. Charge storage effects are modeled in GP model with the help of three capacitors: Base Emitter junction capacitors ($C_{BE} = C_{JE} + C_{DE}$), Base Collector junction capacitance ($C_{BC} = C_{JC} + C_{DC}$) and substrate capacitance (C_{SUB}). The following Equivalent Gummel-Poon DC Model shows the equivalent circuit of the Gummel-Poon model.

Equivalent Gummel-Poon DC Model



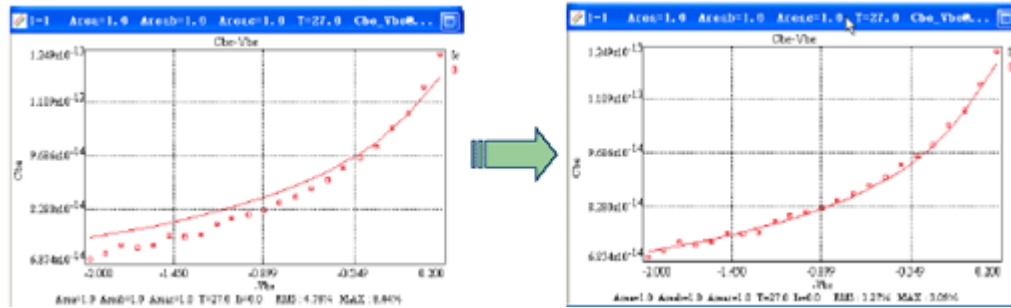
Gummel-Poon AC Model Extraction

GP CV parameters extraction in MBP proceeds like extracting junction capacitance parameters of separate three p-n junctions. For a BJT device, it has B-E junction, B-C junction, and C-S junction (vertical structure) or B-S (lateral structure). We can describe CV behavior for each junction with the same formula:

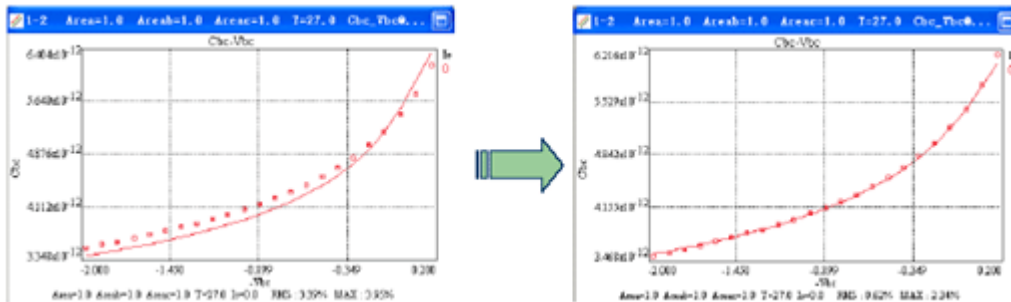
$$C_j = \frac{C_{j0}}{\left(1 - \frac{V_j}{V_{j0}}\right)^{M_j}}$$

C_{j0} , V_{j0} , and M_j are junction zero-bias depletion capacitance, junction built-in potential, and junction exponential factor respectively. So, you can take the same extraction strategy to B-E junction, B-C junction, and C-S/B-S junction respectively. In MBP, `extractorCV.GP` does all the extractions for you, as shown below.

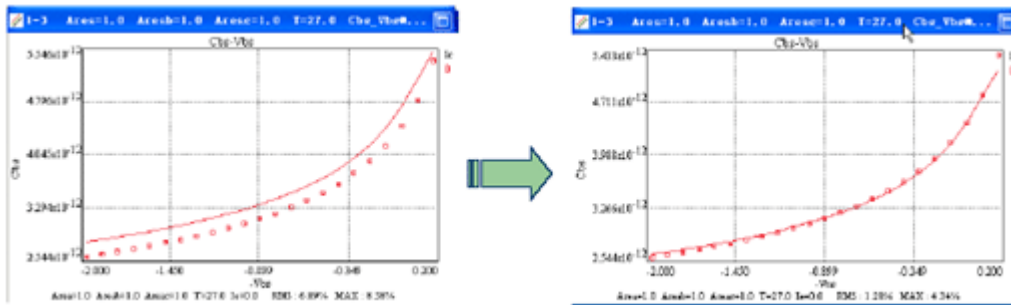
Extract B-E capacitance



Extract B-C capacitance



Extract B-S capacitance



In practice there are always parasitic capacitances, e.g. package or pad capacitances, with such measurement. If they are not calculated out of the measured data, in some cases, the extracted CV parameter may have rather unreasonable values, especially to small capacitance.

Some CV parameters such as transit time parameters are required to be characterized using S-parameter. MBP is capable of modeling S-parameter soon. Equipped Task Tree extractor.GP combines DC and CV extraction for BJT devices.

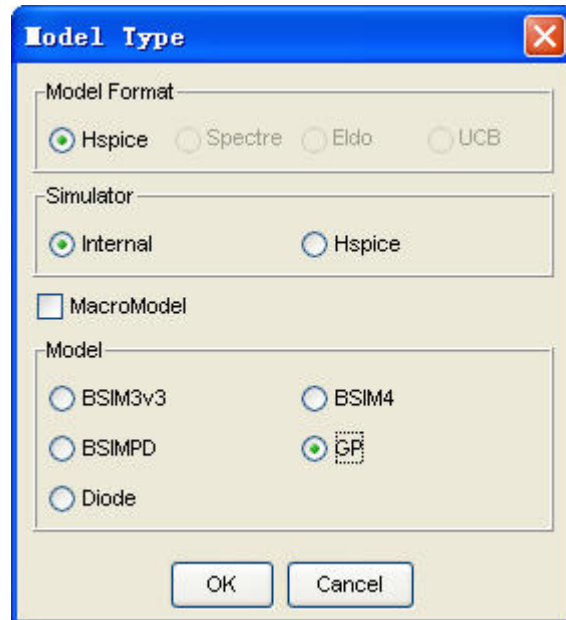
Gummel-Poon DC Model Extraction

Follow the following steps to perform extraction:

Select Model Type

Be sure of choosing model type before you do any extraction. Click Model > Select Model, select GP as model type as shown below.

Select GP as model type



Load Measured Data

Load measured BJT data to the Model Builder. By default, the Gummel plot displays in X-Lin, Y-Log scale, and Beta_{Ic}(I_e) in X-Log, Y-Lin scale.

MBP automatically sets a default model that matches measurement data. For example, once PNP data is loaded, then MBP uses the model file of model_pnp.ini (default pnp G-P model under ~etc\Hspice\GP\model_core) as an initial model for the project.

Extract Model Parameters

Select Extraction > Task Tree to pop up the Task Tree window. Load extractorDC.GP to run extraction, in the sequence of extracting VAF, IS, NF, IKF, RE, ISE, NE, BF, XTI, and XTB. The following is the extraction strategy for extractorDC.GP:

- Extract VAF from the Forward Output plot
- Extract IS, NF from linear part of Forward Gummel plot
- Extract IKF from transition part of Forward Gummel plot
- Extract RE from high current part of Forward Gummel plot
- Extract ISE, NE, BF from low current part of Forward Gummel plot
- Extract XTI from multiple (under different temperatures) Forward Gummel plots
- Extract XTB from multiple (under different temperatures) Beta_{Ic} plots

And as we have emphasized, MBP allows you to configure their own extraction strategy.

Further Extraction and Parameters Optimization

Since test scheme for BJT is flexible, MBP only provides auto extraction procedure for forward characteristics. Bring up optimization window, select the rest of the parameters for extraction, adjust their value to make simulated data best fit measured data. See [Optimization](#) for more details.

NOTE

The modeling of reverse characteristics is similar to the modeling of forward characteristics. And if reverse parameters are extracted, it probably affects the forward modeling. Iterate optimization for several times, you can get a good fit.

Export Model File

Follow similar steps used in MOS model extraction.

HiSIM HV Modeling

This chapter is to give a general guideline on how to extract HiSIM_HV model. We provide information on the following topics:

- [HiSIM HV Overview](#)
- [Extraction of HiSIM HV Model](#)
- [Optimization of HiSIM HV Model](#)

HiSIM HV Overview

HV devices are widely used nowadays, and HiSIM_HV model has already been defined as standard by CMC. HiSIM_HV is valid for modeling both asymmetrical laterally diffused structure (LDMOS) and symmetrical structure type, and has been developed as an extension of the HiSIM model for conventional MOSFETs. HiSIM (Hiroshima-university STARC IGFET Model) is the first completesurface-potential-based MOSFET model for circuit simulation based on the drift-diffusion theory. The basic modeling method of HiSIM_HV is taken over from the HiSIM2 model for advanced MOSFETs, and additional equations for capturing the drift-region effects are included. The high-voltage effects covered by HiSIM_HV model are listed below:

- Quasi-saturation effect
- Self-heating effect
- Independent bias-dependent S/D resistance
- Asymmetric S/D structures
- Gm fall-off in the saturation region
- Bias dependent drift resistance
- Impact Ionization effect

Extraction of HiSIM HV Model

This section introduces the suggested extraction and optimization procedure of HiSIM_HV model in MBP. This routine is just for reference and because of IP protection issue, we only provide the rough operation flow and explanations in the modeling guide. To get more detailed information, please contact mbp_pdl-eesof@keysight.com.

In the following section, the extraction related operations are demonstrated.

Do Measurements

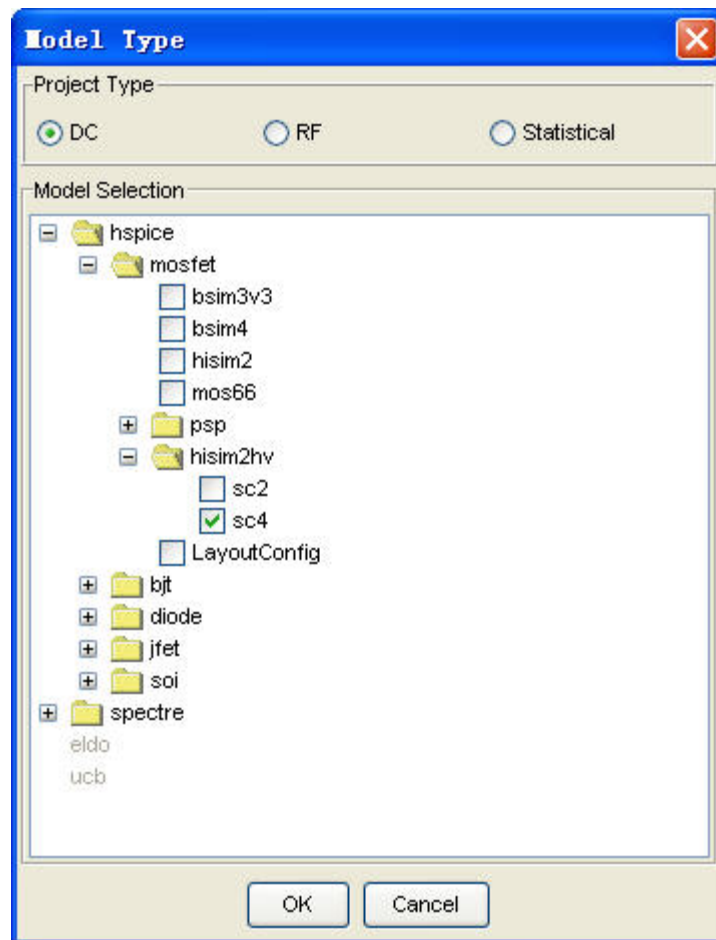
Refer to [Measurement](#) for detailed information on how to carry out measurements.

Load Measurement Data

Load all measured data into MBP. Acquaint with device geometry and bias condition through Device Navigator. To be noted, always set right model type before loading data.

sc4 stands for the latest HiSIM_HV version.

Select model type



Use the Task Tree to Extract Initial Model

Select Extraction > Task Tree to launch Task Tree window. Load built-in task tree and start auto extraction. MBP now provides different task tree for sc2 and sc4 which cover the most important extraction steps for HiSIM_HV model.

To be noticed, because of the great different performance of HV devices, the default auto extraction flow is only for reference. We recommend you to involve into the extraction flow or build his own task tree.

The default task tree list is shown below:

Task Tree	Explanation
extract_DC_RT.HISIM2HV	For generic DC extraction in room temperature.
extract_sc2.HISIM2HV	For extraction with sc2.

The first task in the tree extract_DC_RT.HISIM2HV is to confirm the device characteristic. If the HV device has strong self-heating effect or Ionization Current, tune the model manually at certain step. Click yes in the confirm window, see Confirm window as shown in following figure, to let the task tree paused at the pre-defined step.

Confirm window



The second task is to confirm whether the device is symmetry or asymmetry. Select Yes to confirm the symmetrical structure. (see following figure)

Confirm window 2



NOTE

The default trees could not cover all the different HV devices, however the task tree is flexible. You can pause the flow and fine tune the model manually or to build ones own tree if better results are expected. Please mail to mbp_pdl-eesof@keysight.com if any questions or requirements.

Optimization of HiSIM HV Model

Subtle adjustment to model is always needed after auto extraction.

Use the Optimization

Locate the page and region not fitted well, launch Optimization window, optimize parameters dominate selected region. To do it manually or by optimizer, you must be experienced in HiSIM_HV modeling. You can refer to our Parameter Guide for the extraction tips provided by Keysight.

NOTE

Refer to [Optimization](#) for optimization operation, and Application Note Parameter Guide for more details of the Comments.

Export Model File

After the model is fine-tuned, you can export the model to a SPICE model file. This model file is compatible with HSPICE. Indeed, we recommend you to save the model file for every update.

Macro Modeling

This section describes how to generate, simulate, and optimize an MBP recognizable macro model.

- Macro Models Construction
- Macro Models Simulation
- Macro Models Optimization

Macro Model Structure

Macro model techniques are often applied for high voltage, RF, and other specific applications. A macro model always includes an intrinsic compact model and an extrinsic part composed of parasitic components to form an equivalent circuit over a specified range. There should be only one intrinsic (core) model. Extrinsic elements could be also compact models or behavior models like dependent current /voltage source. MBP differentiate macro models by intrinsic models.

Node Assignment

To construct a macro model that MBP can recognize, follow the node assignment rules below:

Intrinsic model	Node assignment
BSIM3	D, G, S, B
BSIM4	D, G, S, B
BSIMPD	D, G, S, B
Gummel-Poon BJT Modeling Guide	C, B, E
Diode Modeling Guide	P, N

Table 1 Node assignment rules

For example, core model is of BSIM3v3 type, macro model is defined as:

```

macro model name      node assignment
.subckt symnmos D G S B W=1e-6 L=1e-6 count=1
Rd D DI 10
Rs S SI 10
mcore DI G SI B nmos W=W L=L M=count
.....
.ends
    
```

instance parameters

NOTE

- Actually, you can edit analysis.ini to assign nodes to macro models freely.
- If the nodes are assigned to a macro model without following MBP rules and no proper modification to analysis.ini, MBP cannot run simulation or operate model correctly.

Macro Models Optimization

After MBP reads in a macro model, it groups parameters according to elements in optimization windows. As shown in following figure, you can left-click the model name or param (para) in the left column to access corresponding model parameters or user-defined parameters.

Optimization window of exemplar macro model

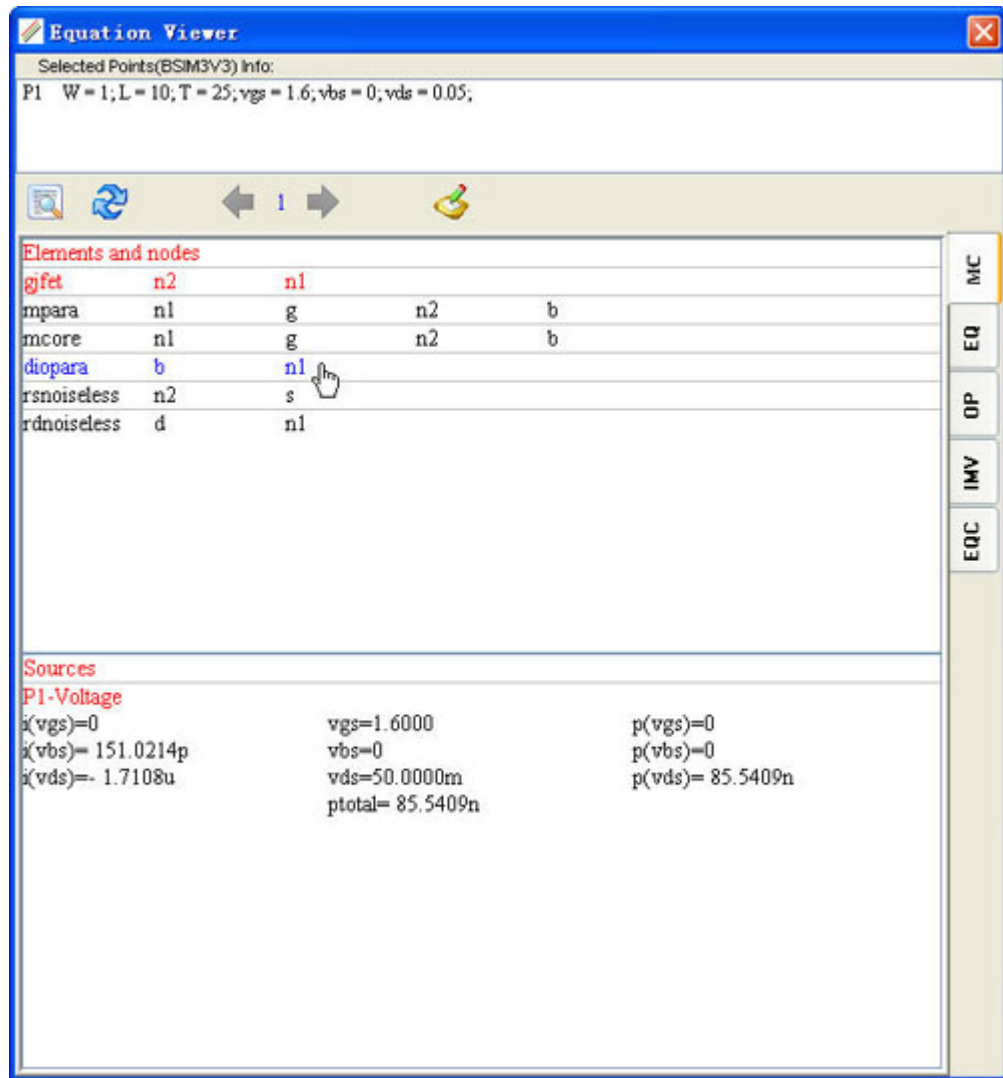
The screenshot shows the 'Optimization' window with a search filter 'vth0'. The table lists parameters with columns for 'Check', 'Name', 'Lower', 'Upper', and 'Step'. Red boxes and arrows categorize the parameters:

- User Defined Parameters:** Includes 'diom' and 'rimos'.
- Core Model:** Includes 'rimosp'.
- Extrinsic Model:** Includes 'rd', 'rs', 'rsc', and 'rdc'.
- Selected Parameter:** 'vth0' is checked and highlighted in blue.

At the bottom, there are sliders for parameters: pp1 (1E-12), pp2 (5.3E-4), is (1E-15), n (1E0), and vth0 (3.9558E-1).

The rest optimization procedure performs similarly to optimizing compact models described in [Optimization](#). You can select parameters and data points to invoke the optimizer or do optimization manually. Moreover, another great feature of MBP is that you can use the Equation Viewer to analysis macro models, as shown in following figure for exemplar macro model. Click any element to link to corresponding equations, operating points or other functions. .

The Equation Viewer for exemplar macro model



NOTE

- Our optimizer works with macro models.
- The MBP optimizer works with remote simulator.

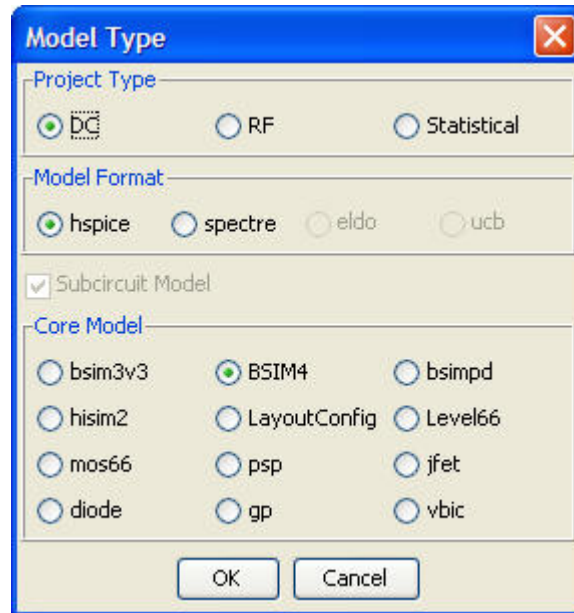
Macro Models Simulation

You can simulate macro models either with our internal engine or call remote HSPICE.

Set Model Type

Since, MBP can recognize macro model automatically, you are required to check the right core model type first as shown below.

Set model type for macro models



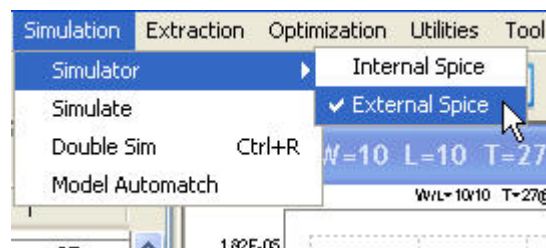
Run Simulation with Internal Engine

With its internal engine, you can simulate most of macro models with good efficiency and speed. All supported compact models in MBP, including BSIM3v3, BSIM4, BSIMPD, GP, and Diode, can act as core models. As for extrinsic elements, besides compact models, it provides fully support to behavior elements such as controlled sources.

Run Simulation with Calling Remote Hspice

MBP also provides you with an option of calling remote HSPICE for verification. Check External Spice in Simulation > Simulator (Figure: Check External Spice for remote call) for invoke external simulator. Moreover, to call remote HSPICE successfully, specify spice.ini properly. Refer to [MBP Files](#) for configuring file `spice.ini`. MBP saves the latest netlist (`.sp`) and simulation results (`*.lis`, `*.ac0`) to `~\Keysight\MBP_201x_xx\hspice\`. You can debug with these files if meeting any issues in remote call.

Check External Spice for remote call



NOTE

- Calling remote simulator leads to substantial increase in the actual simulation time.
- Calling remote HSPICE is also applicable to compact models.

MOSRA Modeling

MBP offers environment for MOSRA device reliability analysis, capable of simulating HCI and NBTI aging effect and embedded in HSPICE. This chapter describes how to MOSRA implementation in MBP. It provides information on the following topics:

- [Data Format](#)
- [MOSRA simulation](#)

Data Format

The data format for MOSRA analysis is similar to general measurement data in MBP (see [MBP Files](#) for details). A sample MOSRA data is shown as follows.

```
condition {corner = tt,date = oct_20_02,instrument=
  (hp4145, probe_station), mode=forward, datatype=
  mosra, version=1.0, type=nmos }
  Page (name=ids_vgs_vbs, x=vgs, p=vbs, y=ids) { vds=0.05
  , w=10.0, L=0.13, T=25.0 } stress (time=0.0, vds=2.0,
  vgs=1.0, vbs=0.0)
  curve { 0.0 }
  0.0      5.000E-14
  0.05     5.002E-14
  0.1      5.010E-14
  0.15     5.055E-14
  0.2      5.315E-14
  .....
```

The keyword datatype is required to be specified as mosra. The *version=1.0* corresponds to MOSRA level. And stress item describes bias conditions and also the timing node during aging test.

MBP also supports another kind of MOSRA data that allows to take aging span as the variable.

```
condition{corner = tt,date = oct_20_02,instrument=
  (hp4145, probe_station),mode=forward}
Datatype {s_target}
Version{2.1}
type {nmos}
Delimiter{,}
Instance{L, W, T}
Strss_Condtion{S_vgs=1, S_vds=2, S_vbs=0, S_time}
Input{Vgg=2, Vdd=2, Vbb=-1,Vdlin=0.05}
Data{ w, l, t, S_vgs, S_vds,S_vbs, S_time, vth_lin,
vthsat, Idlin, Idsat, Ioff, gm}
```

```

10, 2, 125, 1, 2, 0, 0, 0.728628, 0.697769, 7.348435E-5
, 8.848617E-4, 2.029162E-12, 4.2957E-4}}
10, 2, 125, 1, 2, 0, 1e5, 0.728985, 0.698127, 7.
343337E-5, 8.8407E-4, 2.028872E-12, 4.2903E-4}}
.....

```

The Datatype is s_target as the differential from general DP data (see [DP data format](#) for reference).

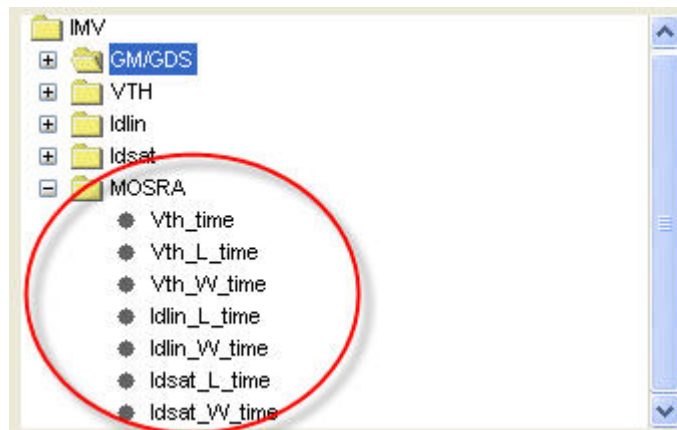
NOTE

All variables in the MOSRA data must start with s. For example, name the Vgs during aging test as S_vgs for the data.

The whole data, including instance parameters, bias conditions, timing nodes, and physical quantities is stored in the following data session.

In MBP, we pre-defined several IMV pages to help you to catch the trend of stress data, as shown below.

Pre-defined IMV pages for stress data



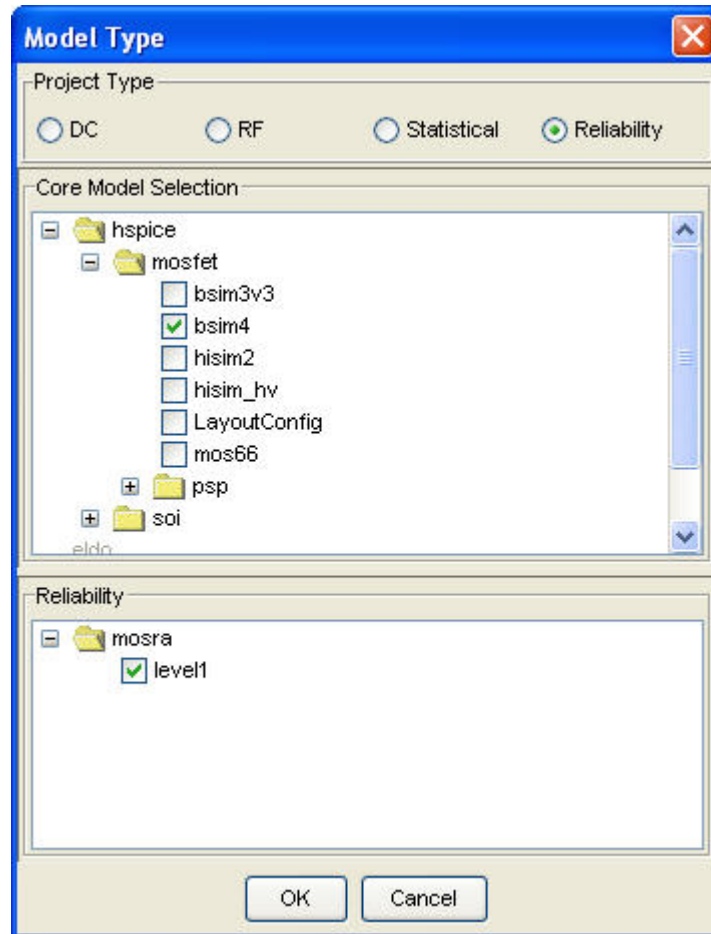
MOSRA simulation

MBP invokes external HSPICE for simulating MOSRA model. Make sure that remote HSPICE has been set up correctly.

Set Model Type

Choose Model > Select Model from main menu and select Reliability. Then, determine the core model in the upper section and MOSRA in the lower section, as shown below.

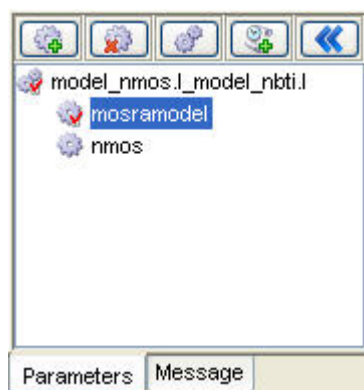
Selecting MOSRA analysis



Run Simulation of MOSRA Model

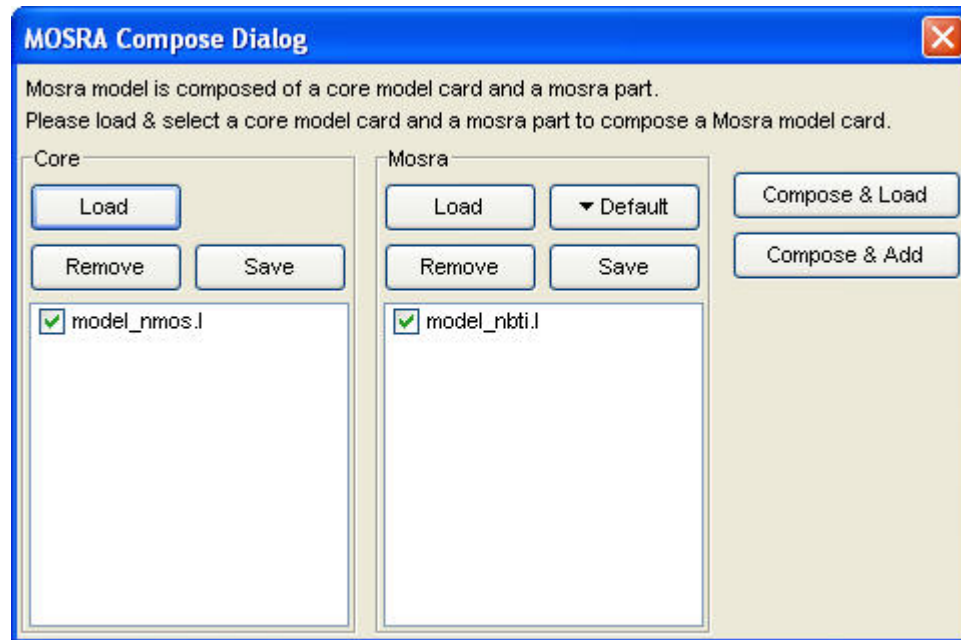
After setup of the model type, MBP merges defaulted core model and defaulted MOSRA model as per selection. For example, as seen from [Figure: Selecting MOSRA analysis](#), the upper one is defaulted MOSRA level 1 model and the lower one is defaulted BSIM4 core model. Double-clicking to access any of them.

Model Viewer for MOSRA



If clicking Load Model in such case, it pops up MOSRA compose dialog window as shown following figure. Hence, you can deal with core model and MOSRA model separately. The Compose & Load is to replace current MOSRA model. The Compose & Add is to append a MOSRA model.

Model Composer for MOSRA



PSP Modeling

This section provides general guideline on how to extract PSP models.

- [PSP Overview](#)
- [Extraction of PSP Model](#)
- [Optimization of PSP Model](#)

PSP Overview

The PSP model is a compact MOSFET model intended for digital, analog, and RF-design, which is jointly developed by NXP Semiconductors Research (formerly part of Philips) and Arizona State University (formerly at The Pennsylvania State University). The roots of PSP lie in both MOS Model 11 (developed by Philips) and SP (developed by Penn State University). PSP is a surface-potential based MOS Model, containing all relevant physical effects to model present-day and upcoming deep-submicron bulk CMOS technologies. The source/drain junction model, c.q. the JUNCAP2 model, is fully integrated in PSP. PSP accounts for the following physical effect:

- Non-uniform lateral doping
- Non-uniform vertical doping
- Field-dependent mobility
- Velocity saturation
- Conductance effects
- Series-resistance
- Short-channel effects
- Narrow-width effects
- Gate poly-depletion
- Quantum-mechanical corrections
- Overlap capacitances (ϕ -based)
- Impact ionization current
- Gate leakage current
- Gate-induced drain/source leakage (GIDL, GISL)
- Junction diode IV and CV (forward and reverse)
- Diode reverse breakdown
- Noise (1/f, thermal, induced gate and shot noise)
- Non-quasi-static effects
- Gate and bulk resistances
- STI stress effect

The PSP model has a hierarchical structure which means that there is a strict separation of the geometry scaling in the global model and the model equations in the local model. Consequently, PSP can be used at either one of two levels.

- Global parameter set: To model a geometry range with instance parameters.
- Local parameter set: To model one specific geometry.

For more information about Parameter Extraction, refer [PSP documentation](#) page.

Extraction of PSP Model

This section introduces the suggested extraction and optimization procedure of PSP model in MBP. This routine is just for reference and because of IP protection issue, we only provide the rough operation flow and explanations in the modeling guide.

We recommend a rough extraction routine via following steps:

1. Establish temperature range, model specs, and determine process parameters.
2. CV Model Extraction.
3. DC Model Extraction.
4. Gate Current and GIDL model Extraction.
5. Temperature Coefficients Extraction.
6. Check Error criteria, if not satisfied, do the iterations from Step 2.

In the following section, the related operations are demonstrated.

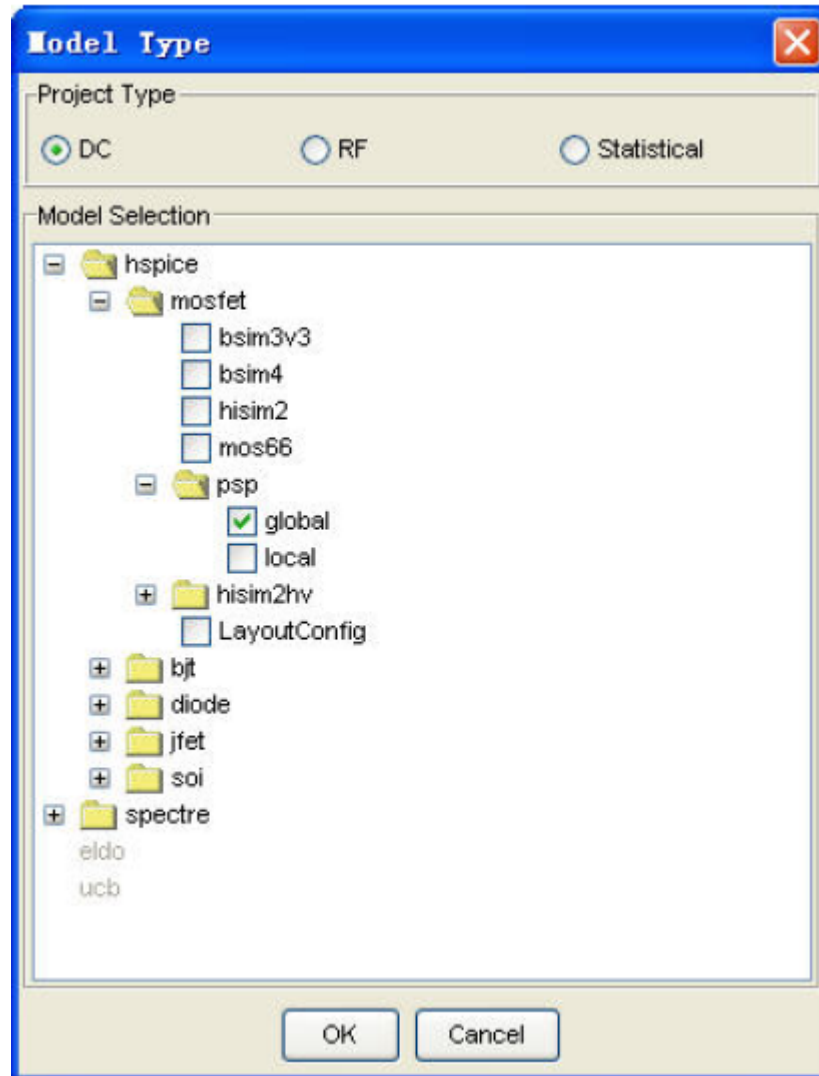
Perform Measurements

See [Measurement](#) for detailed information on how to perform measurements.

Load Measurement Data

Load all measured data into MBP. Acquaint with device geometry and bias condition through Device Navigator. To be noted, always set right model type before loading data.

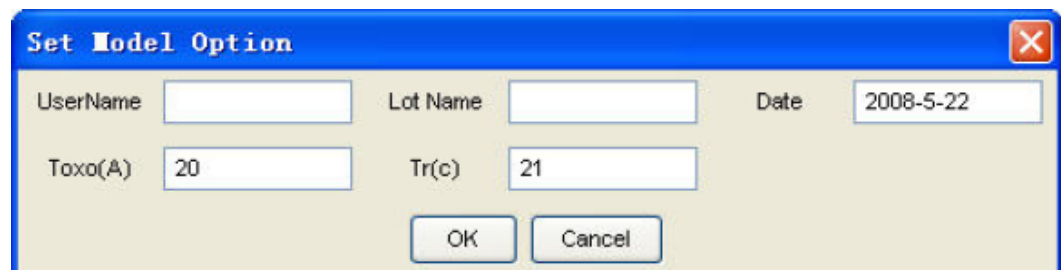
Select model type



Set Process Parameter

Click Model > Set model options to input process parameters. Process parameters affect device characteristics markedly. And since they are referred in extraction, assure their accuracy.

Set process parameters



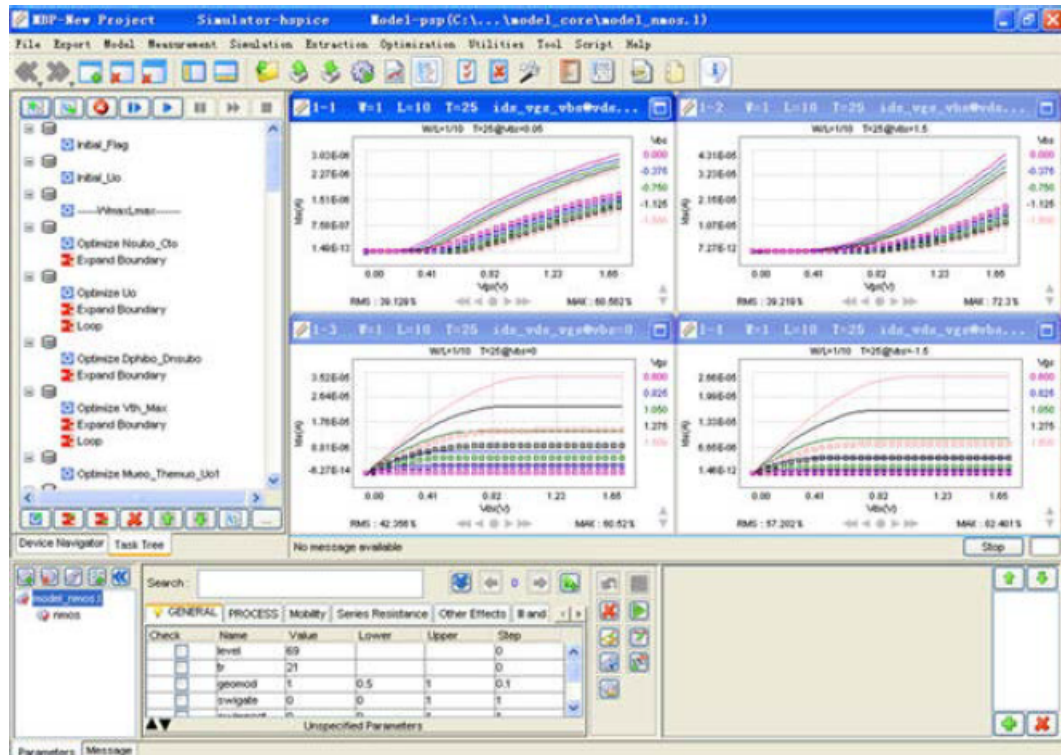
Use the Task Tree to Extract Initial Model

Select Extraction > Task Tree to launch Task Tree window. Load built-in task tree and start auto extraction. After a few minutes, you get an initial model file with improved fittings. MBP now provides several different task tree to cover the different requirement or steps in the extraction flow with PSP model. Divide the extraction flow in to several different tree to let you involve in the auto extraction process more freely and fine turn the model after each step, which could improve the extraction result at the end. Besides, you can add the check points to control the process in the middle of the task tree or customize their own extraction flow. For Task Tree related introduction, see [Task Tree](#) section.

The build-in task tree list for PSP global model is shown below:

Task Tree	Explanation
extract_withCV.PSP	For both DC and CV extraction.
PSP_QuickDC.PSP	For the extraction of a comparatively coarse model in a short time
GIDL and Gate Current.PSP	For Gate/GIDL current related parameters extraction.
LOD_STI.PSP	For STI effect related parameters extraction.
temp.PSP	For temperature coefficient extraction.

Extract the initial model by the Task Tree



Optimization of PSP Model

Subtle adjustment to model is always required after auto-extraction.

Use the Optimization

Locate the page and region not fitted well, launch Optimization window, optimize parameters dominate selected region. To do it manually or by optimize. For this method, your require experience in the PSP modeling. Refer to our Parameter Guide for the extraction tips that Keysight provides. See [Optimization](#) for optimization operation, and Application Note > Parameter Guide, for more details of the Comments.

Use the PSP scaling tweaking

MBP provides a new feature named Scaling tweaking especially for the extraction of Global parameters from Local parameters. For more details, see Application Note > PSP scaling tweaking for more details of the Comments.

Export Model File

After the model is fine-tuned, you can export the model to a SPICE model file. This model file is compatible with HSPICE. Indeed, we recommend you to save the model file for every update.

RF Modeling

- RF Data and Simulation
- Parameter Conversion
- De-embedding
- Customization by Script
- MBP Package and Solution
- References

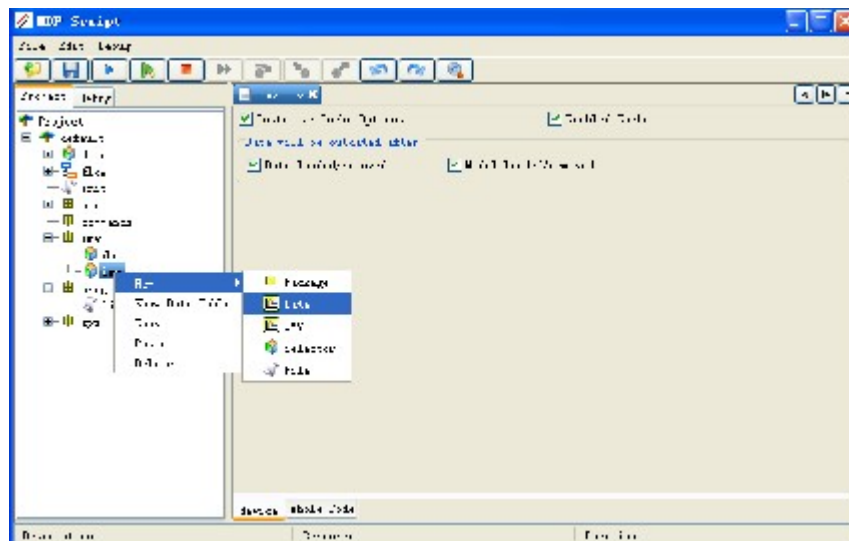
Customization by Script

Customize Target

MBP defines many useful targets for RF Application in default package, such as Leff, Reff, Q for inductor model and ft, fmax for MOSFET model. Additionally, MBP supports user-defined targets. For example, take the implementation of Leff, Reff, Q.

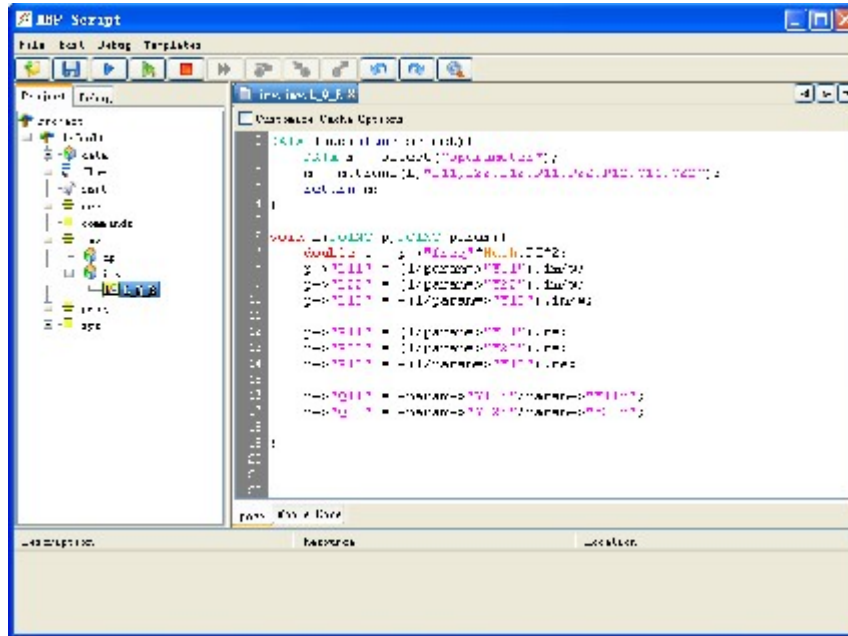
Select Script > Script Project from the main menu. In the left Project panel, select default > imv > imv. Right-click and choose New > Data in the popup menu, as shown in following figure.

Create New Data



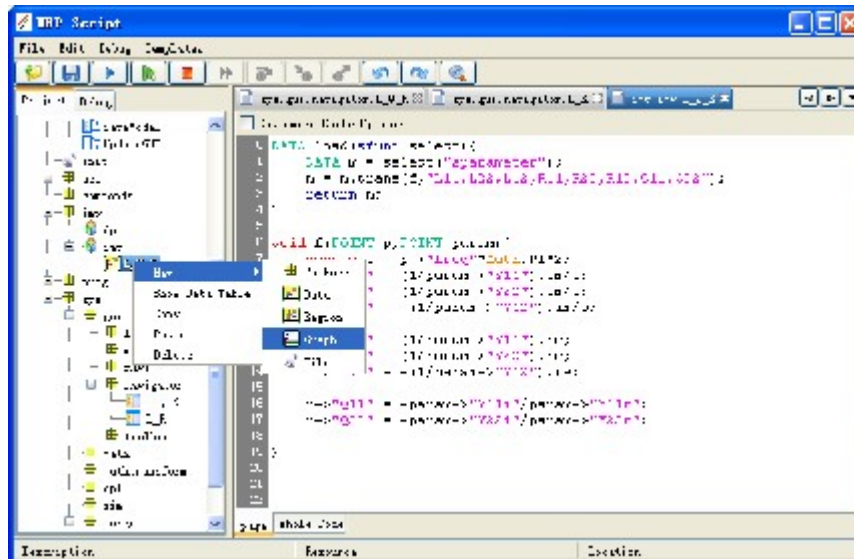
Define the target in the right panel with MBP Script, as shown in following figure.

Target Definition



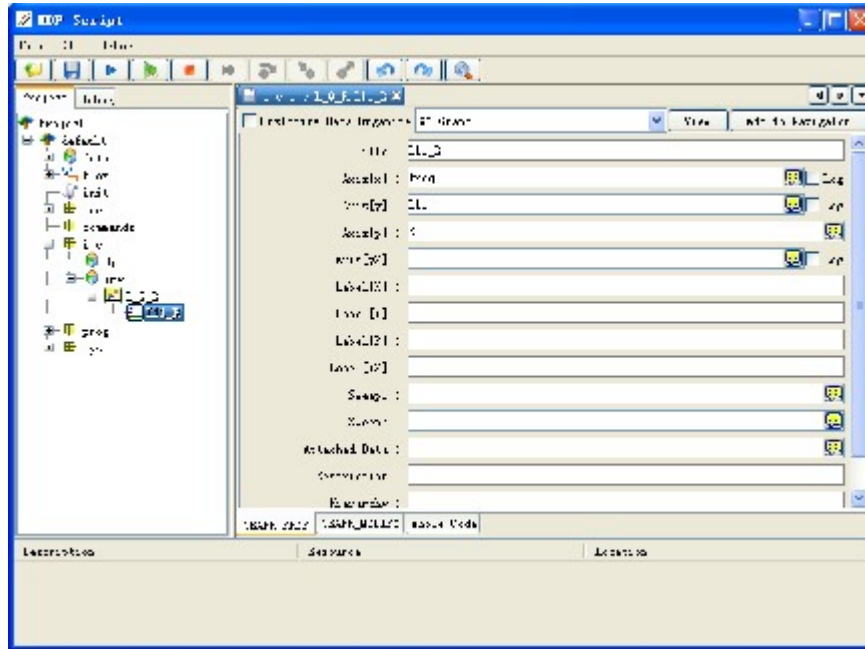
Right-click and select New > Graph from the popup menu, as shown in following figure.

Create New Graph



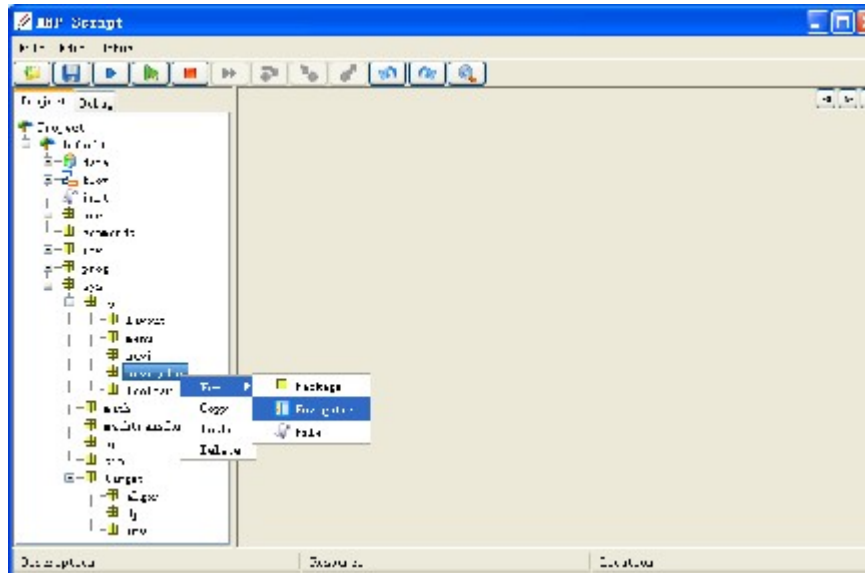
Edit the property of graph. Input the information such as Title, Axis[x], Axis[y] in the GRAPH_PROP tab, as shown in following figure.

Edit Graph Property



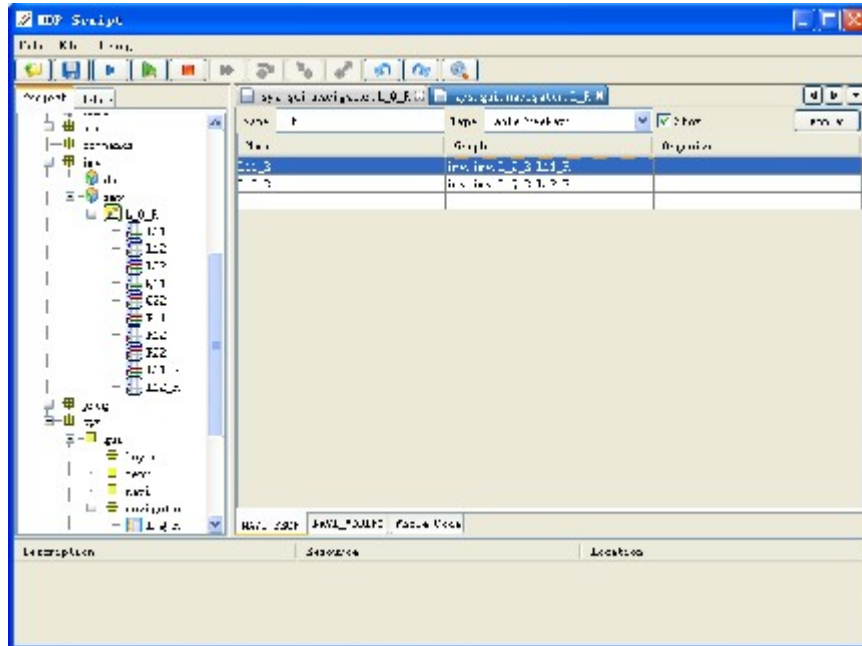
Select default > sys > gui > navigator in the left Project panel. Right-click and choose New > Navigator from the popup menu, as shown in following figure.

Add New Navigator



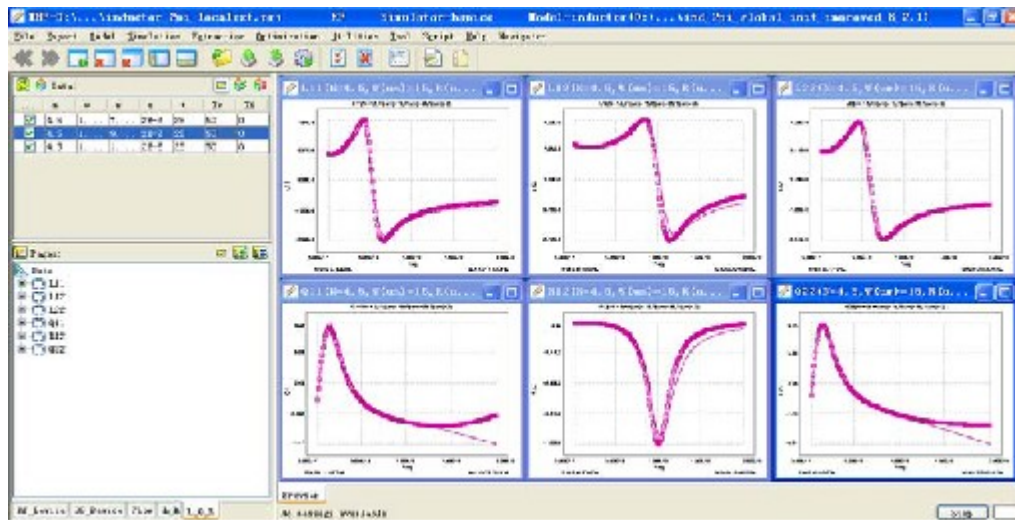
Select the navigator. Right-click and choose New > Graph from the popup menu, as shown in following figure.

Add Graph to the Navigator

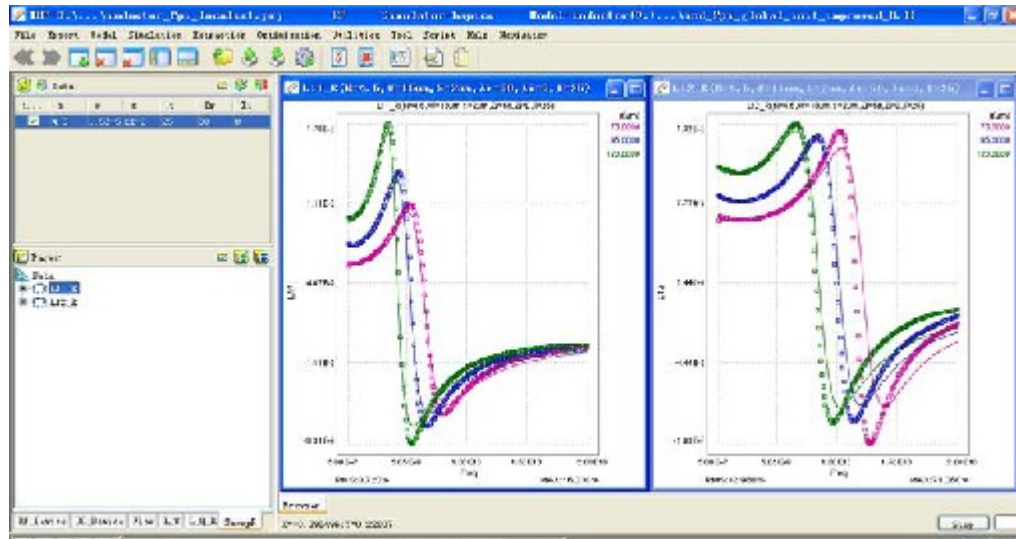


Click the run button. The new navigator is shown in the main GUI of MBP. Two example results are shown in [Figure:New navigator L_Q_R in GUI](#) and [Figure:New navigator SweepR in GUI](#), respectively.

New navigator L_Q_R in GUI



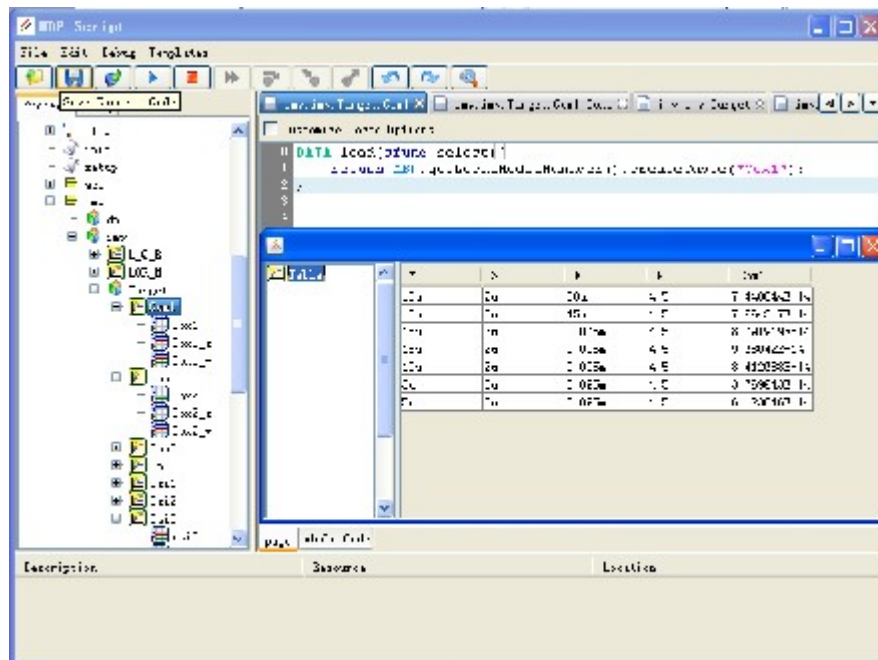
New navigator SweepR in GUI



Plot Model Parameters versus Instance Parameter

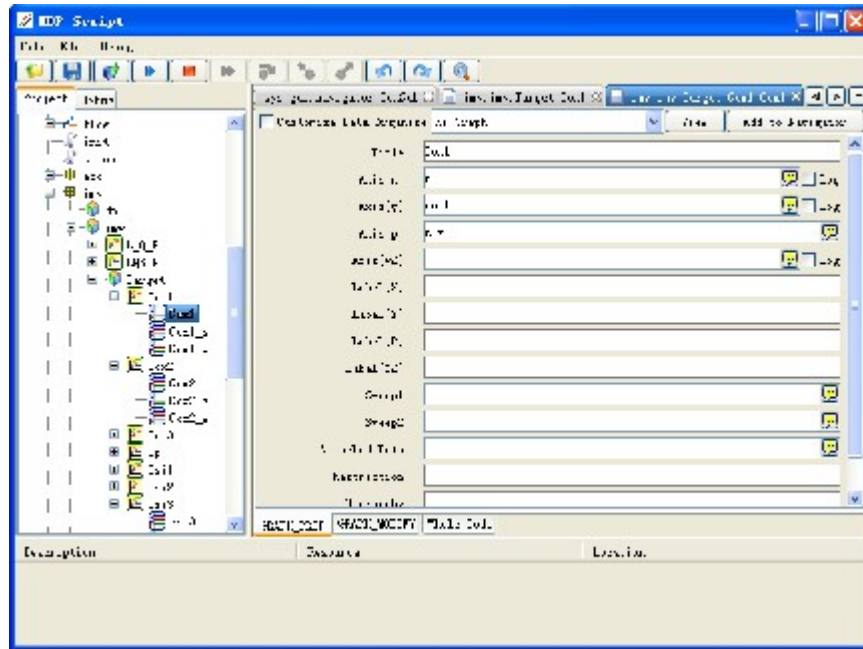
Choose Script > Script Project from the main menu to pop up the MBP Script interface. Define the data (default > imv > imv > Target > Cox1), as shown in following figure. This example can be loaded by choosing Model > Select Model > RF > standard > scalable from the main menu.

Show Data Table



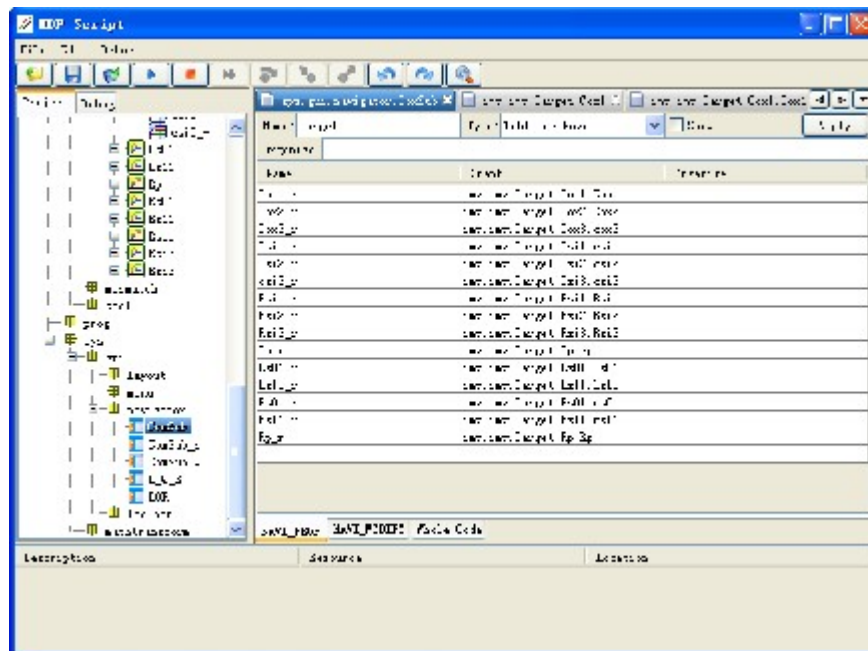
Define the plot (default > imv > imv > Target > Cox1 > Cox1), as shown in following figure. Select the targets for X, Y, and P axes.

Define Plot



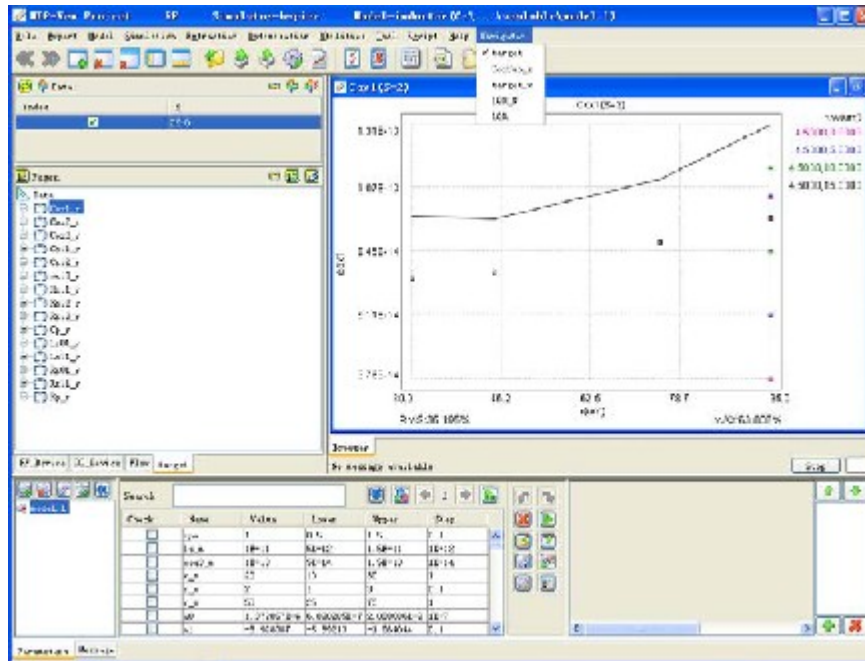
Add the plot to the navigator (default > sys > gui > navigator), shown in following figure.

Add Plot to Navigator



Load the initial scalable model first and load point models from Model > Local Model Manager. Select the navigator from the Navigator menu, as shown in following figure

Select Navigator



De-embedding

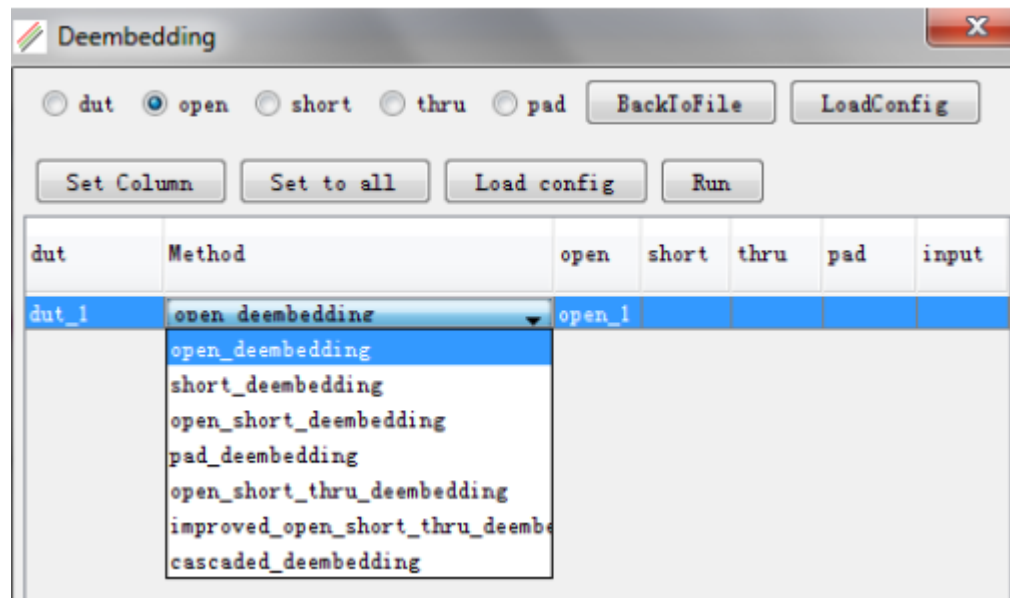
The De-embedding window can be enabled by choosing Utilities > Deembedding from the main menu. After loading the measurement data for different test structures (such as dut, open, and so on), click the Deembedding button, as shown in following figure.

Load data for de-embedding



Choose the de-embedding method from the drop-down menu, as shown in following figure. Finally, click the Run button and save the de-embedding result.

Choose de-embedding method



Currently, there are seven de-embedding methodologies available in MBP. The subsequent subsections introduce them one by one.

Open De-embedding Method

Step 1. $S_{dut} \rightarrow Y_{dut}; S_{open} \rightarrow Y_{open}$

Step 2. $Y_{de-emb} = Y_{dut} - Y_{open}$

Step 3. $Y_{de-emb} \rightarrow S_{de-emb}$

Short De-embedding Method

Step 1. $S_{dut} \rightarrow Z_{dut}; S_{short} \rightarrow Z_{short}$

Step 2. $Z_{de-emb} = Z_{dut} - Z_{short}$

Step 3. $Z_{de-emb} \rightarrow S_{de-emb}$

Open Short De-embedding Method

$$\text{Step 1. } S_{dut} \rightarrow Y_{dut}; S_{open} \rightarrow Y_{open}; S_{short} \rightarrow Y_{short}$$

$$\text{Step 2. } Y_1 = Y_{dut} - Y_{open}; Y_2 = Y_{short} - Y_{open}$$

$$\text{Step 3. } Y_1 \rightarrow Z_1; Y_2 \rightarrow Z_2; Z_{de-emb} = Z_1 - Z_2$$

$$\text{Step 4. } Z_{de-emb} \rightarrow S_{de-emb}$$

Pad Open Short De-embedding Method [1]

$$\text{Step 1. } S_{dut} \rightarrow Y_{dut}; S_{open} \rightarrow Y_{open}; S_{short} \rightarrow Y_{short}; S_{pad} \rightarrow Y_{pad}$$

$$\text{Step 2. } Y_1 = Y_{dut} - Y_{pad}; Y_2 = Y_{short} - Y_{pad}; Y_3 = Y_{open} - Y_{pad}$$

$$\text{Step 3. } Y_1 \rightarrow Z_1; Y_2 \rightarrow Z_2; Y_3 \rightarrow Z_3; Z_4 = Z_1 - Z_2; Z_5 = Z_3 - Z_2$$

$$\text{Step 4. } Z_4 \rightarrow Y_4; Z_5 \rightarrow Y_5; Y_{de-emb} = Y_4 - Y_5$$

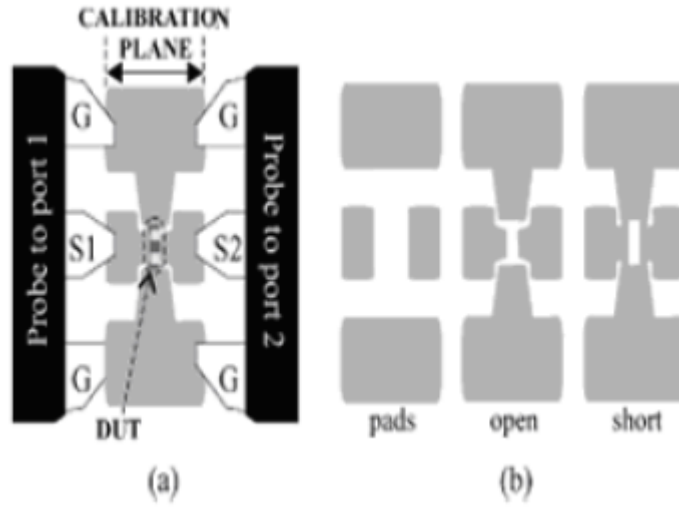
$$\text{Step 5. } Y_{de-emb} \rightarrow S_{de-emb}$$

The layout of test and dummy structure is shown in [Figure: Layout of Test and Dummy Structure](#), and the equivalent circuit is shown in [Figure: Equivalent Circuit](#).

Layout of Test and Dummy Structure

(a) Layout of the measured test structure consisting of a MOSFET surrounded by pads in GSG configuration

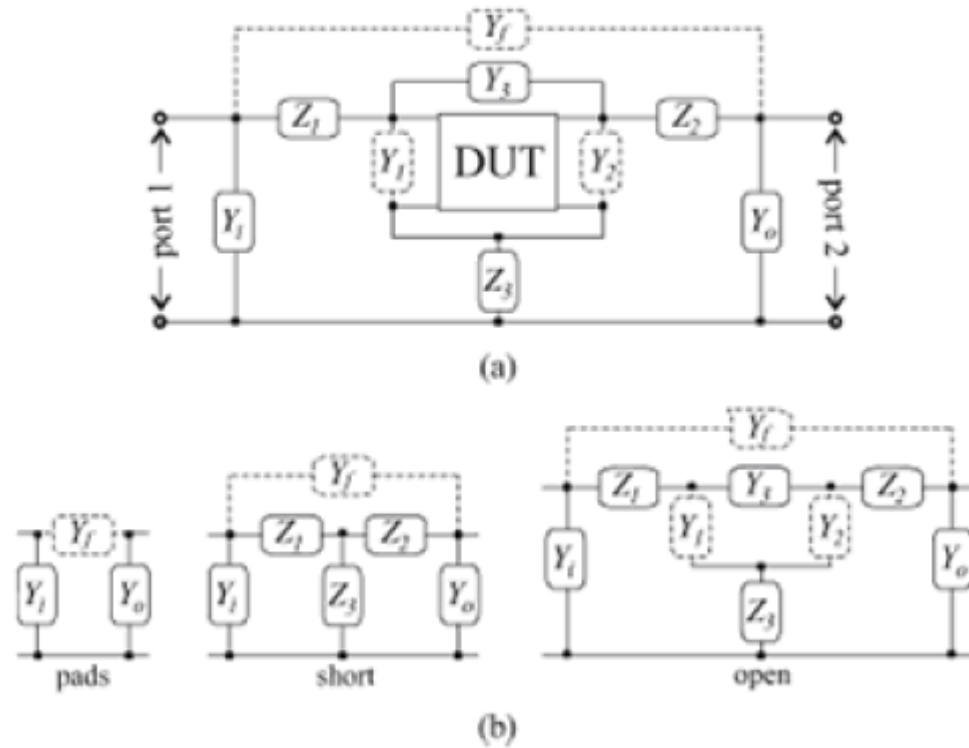
(b) Layouts of the dummy structures used to determine the pad parasitic effects



Equivalent Circuit

(a) General model for complete test-structure

(a) Assumed equivalent circuits for the pads, short, and open structures



Open Short Thru De-embedding Method [2]

$$\text{Step 1.} \quad G_1 = Y_{open}[1][1] + Y_{open}[1][2]; \quad G_2 = Y_{open}[2][2] + Y_{open}[1][2]; \quad G_3 = -Y_{open}[1][2]$$

$$Y_{dut}^1[1][1] = Y_{dut}[1][1] - G_1; \quad Y_{dut}^1[2][2] = Y_{dut}[2][2] - G_2$$

$$\text{Step 2.} \quad Z_1 = \frac{1}{2} \left(\frac{1}{Y_{thru}[1][2]} + \frac{1}{Y_{short}[1][1]} - \frac{1}{Y_{short}[2][2]} \right);$$

$$Z_2 = \frac{1}{2} \left(\frac{1}{Y_{thru}[1][2]} - \frac{1}{Y_{short}[1][1]} + \frac{1}{Y_{short}[2][2]} \right);$$

$$Z_3 = \frac{1}{2} \left(-\frac{1}{Y_{thru}[1][2]} + \frac{1}{Y_{short}[1][1]} + \frac{1}{Y_{short}[2][2]} \right)$$

$$Y_{dut}^1 \rightarrow Z_{dut}^1; \quad Z_{dut}^s[1][1] = Z_{dut}^1[1][1] - Z_1 - Z_3; \quad Z_{dut}^s[1][2] = Z_{dut}^1[1][2] - Z_3;$$

$$Z_{dut}^s[2][1] = Z_{dut}^1[2][1] - Z_3; \quad Z_{dut}^s[2][2] = Z_{dut}^1[2][2] - Z_2 - Z_3$$

$$\text{Step 3.} \quad Z_{dut}^s \rightarrow Y_{dut}^s; \quad Y_{dut}^q[1][1] = Y_{dut}^s[1][1] - G_3; \quad Y_{dut}^q[1][2] = Y_{dut}^s[1][2] + G_3;$$

$$Y_{dut}^q[2][1] = Y_{dut}^s[2][1] + G_3; \quad Y_{dut}^q[2][2] = Y_{dut}^s[2][2] - G_3$$

$$Y_{dut}^q \rightarrow S_{dut}^s$$

Improved Open Short Thru De-embedding Method [3]

$$\text{Step 1. } G_1 = Y_{open}[1][1] + Y_{open}[1][2]; \quad G_2 = Y_{open}[2][2] + Y_{open}[1][2];$$

$$G_3 = \frac{1}{2} \left(\frac{-1}{Y_{open}[1][1]} + \frac{1}{Y_{thru}[1][2]} \right)^{-1}$$

$$Z_1 = \frac{1}{2} \left(\frac{-1}{Y_{thru}[1][2]} + \frac{1}{Y_{short}[1][1] - G_1} - \frac{1}{Y_{short}[2][2] - G_2} \right)$$

$$Z_2 = \frac{1}{2} \left(\frac{-1}{Y_{thru}[1][2]} - \frac{1}{Y_{short}[1][1] - G_1} + \frac{1}{Y_{short}[2][2] - G_2} \right)$$

$$Z_3 = \frac{1}{2} \left(\frac{1}{Y_{thru}[1][2]} + \frac{1}{Y_{short}[1][1] - G_1} + \frac{1}{Y_{short}[2][2] - G_2} \right)$$

$$Y_A = Y_{meas} - \begin{bmatrix} G_1 & 0 \\ 0 & G_2 \end{bmatrix}$$

$$\text{Step 2. } Z_B = Z_A - \begin{bmatrix} Z_1 + Z_3 & Z_3 \\ Z_3 & Z_2 + Z_3 \end{bmatrix}$$

$$\text{Step 3. } Y_{DUT} = Y_B - \begin{bmatrix} G_3 & -G_3 \\ -G_3 & G_3 \end{bmatrix}$$

Cascaded Line Method

$$\text{Step 1. } S_{dut} \rightarrow Y_{dut}; S_{open} \rightarrow Y_{open}; S_{thru} \rightarrow Y_{thru}$$

$$\text{Step 2. } Y_{dutopen} = Y_{dut} - Y_{open}; Y_{thruopen} = Y_{thru} - Y_{open}$$

$$\text{Step 3. } Y_{dutopen} \rightarrow A_{dutopen}, \text{ where } A_{dutopen} = A_{s1} A_{ind} A_{s2}$$

$$\text{Step 4. } Z = Z0 \left[\frac{(1 + S_{11,thru,open})^2 - S_{21,thru,open}^2}{(1 - S_{11,thru,open})^2 - S_{21,thru,open}^2} \right]^{\frac{1}{2}}$$

$$D = \frac{1 - S_{11,thru,open}^2 + S_{21,thru,open}^2}{[(1 + S_{11,thru,open})^2 - S_{21,thru,open}^2]^{\frac{1}{2}} [(1 - S_{11,thru,open})^2 - S_{21,thru,open}^2]^{\frac{1}{2}}}$$

$$\lambda = \frac{\ln\left(\frac{D+1}{D-1}\right)}{2l}$$

$$A_{s1,2} = \begin{bmatrix} \cosh(\gamma l_{1,2}) & Z \sinh(\gamma l_{1,2}) \\ \frac{\sinh(\gamma l_{1,2})}{Z} & \cosh(\gamma l_{1,2}) \end{bmatrix}$$

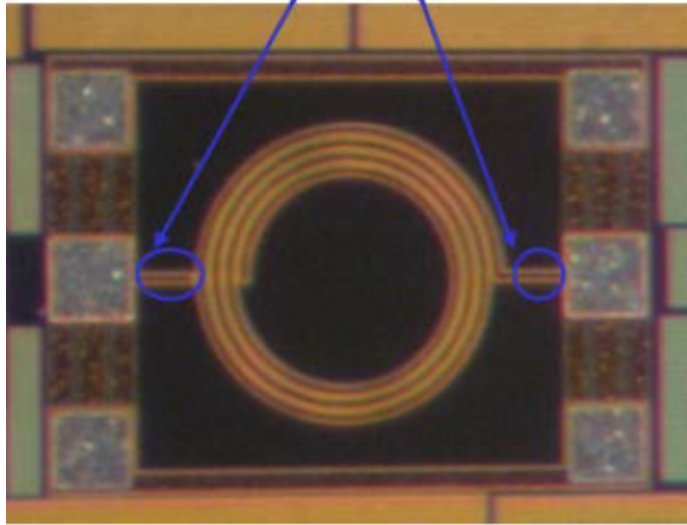
$$\text{Step 5. } A_{Ind} = A_{s1}^{-1} A_{dut,open} A_{s2}^{-1}$$

$$A_{ind} \rightarrow S_{ind}$$

Some parameters should be set to reasonable values before de-embedding: $L1$, $L2$ are the length of extending lines (as shown in following figure) that should be de-embedded from the initial data. L is the total length of the transmission line in the thru de-embedding structure. $Z0$ is the characteristic impedance of test system (Its default value is 50).

Extending lines de-embedding

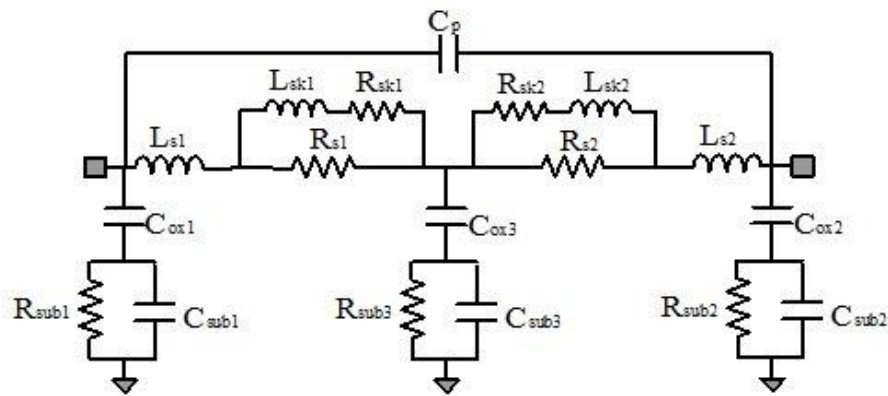
Extending lines



MBP Package and Solution

Standard Inductor Package Introduction

MBP provides inductor package for generic usage of standard inductor modeling. The default inductor model has 2-pi structure. Its equivalent circuit is shown in following figure.



Equivalent circuit of the inductor with 2-pi structure

The definitions of L_{eff} , R_{eff} , and Q are shown:


$$L_{11} = \frac{\text{imag}\left(\frac{1}{Y_{11}}\right)}{2 \times \text{PI} \times \text{Freq}}; \quad L_{22} = \frac{\text{imag}\left(\frac{1}{Y_{22}}\right)}{2 \times \text{PI} \times \text{Freq}}; \quad L_{12} = \frac{-\text{imag}\left(\frac{1}{Y_{12}}\right)}{2 \times \text{PI} \times \text{Freq}}$$

$$R_{11} = \frac{\text{real}\left(\frac{1}{Y_{11}}\right)}{2 \times \text{PI} \times \text{Freq}}; \quad R_{22} = \frac{\text{real}\left(\frac{1}{Y_{22}}\right)}{2 \times \text{PI} \times \text{Freq}}; \quad R_{12} = \frac{-\text{real}\left(\frac{1}{Y_{12}}\right)}{2 \times \text{PI} \times \text{Freq}}$$

$$Q_{11} = \frac{\text{imag}(Y_{11})}{\text{real}(Y_{11})}; \quad Q_{22} = \frac{\text{imag}(Y_{22})}{\text{real}(Y_{22})};$$

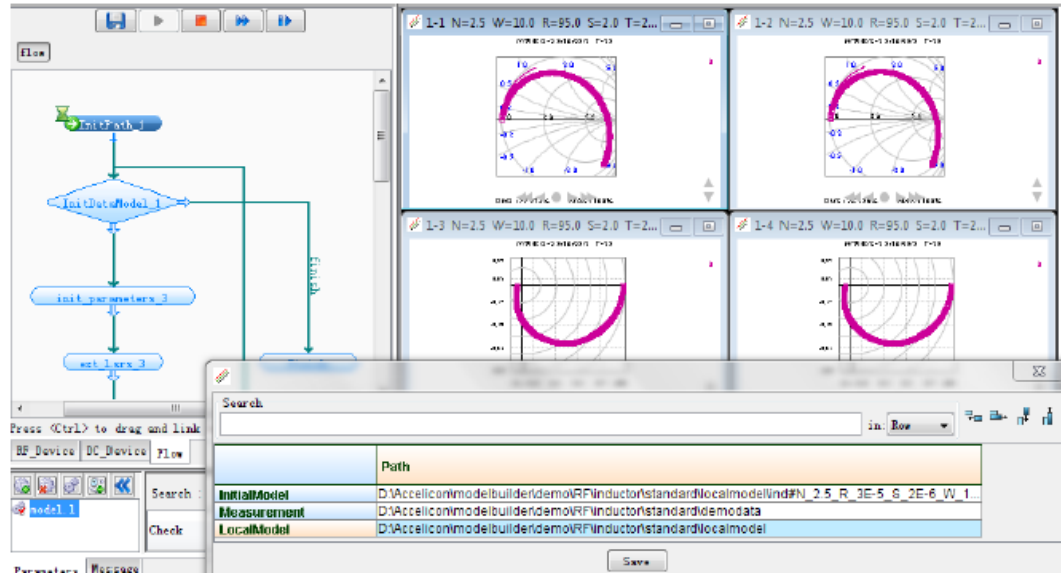
For standard inductor model extraction, there are two projects in MBP inductor package: local project and scalable project.

Local Project

In the local project, each device is extracted by a point model. You can open it by choosing Mode > Select Model > RF > inductor > standard > local from the main menu. The flow is displayed. Click the Run icon  to run the flow from InitPath_1. A window pops up and you should input the following information, as shown in [Figure: Set Initialized Condition](#).

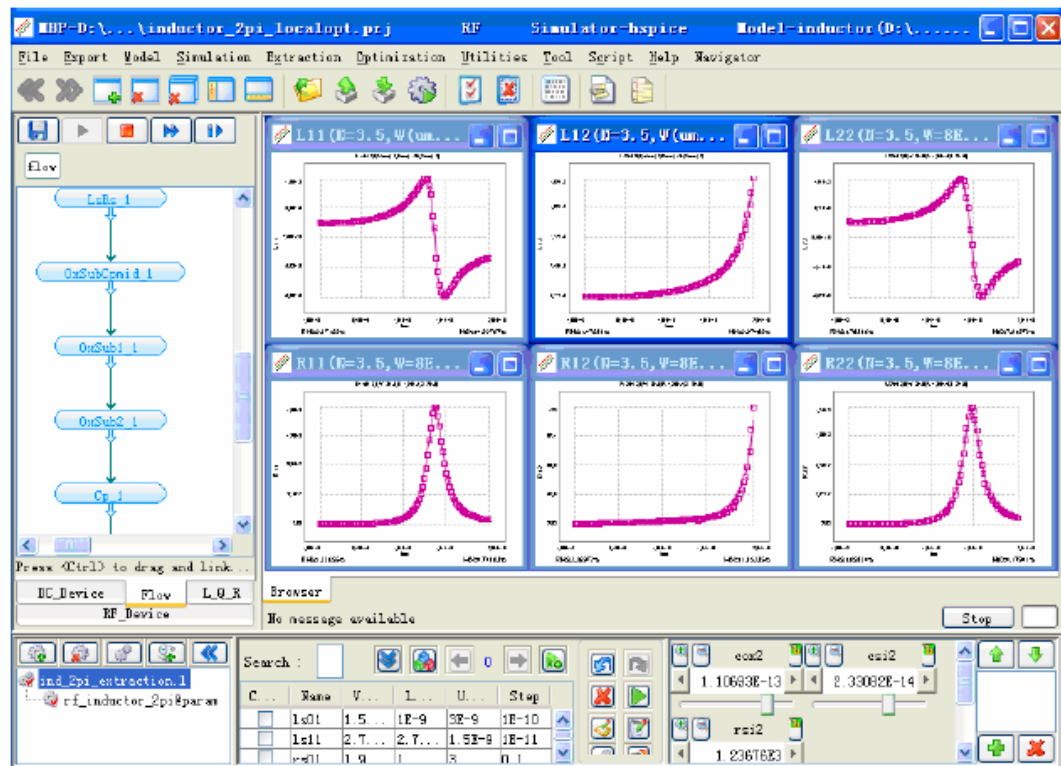
1. InitialModel: to set initial model name.
2. Measurement: to set measurement data folder.
3. LocalModel: to set folder for all the generated point models.

Set Initialized Condition



After entering the information, MBP runs the flow automatically. It loads one device every time, then generates the corresponding point model. The fitting results are shown in following figure.

Generation of point models

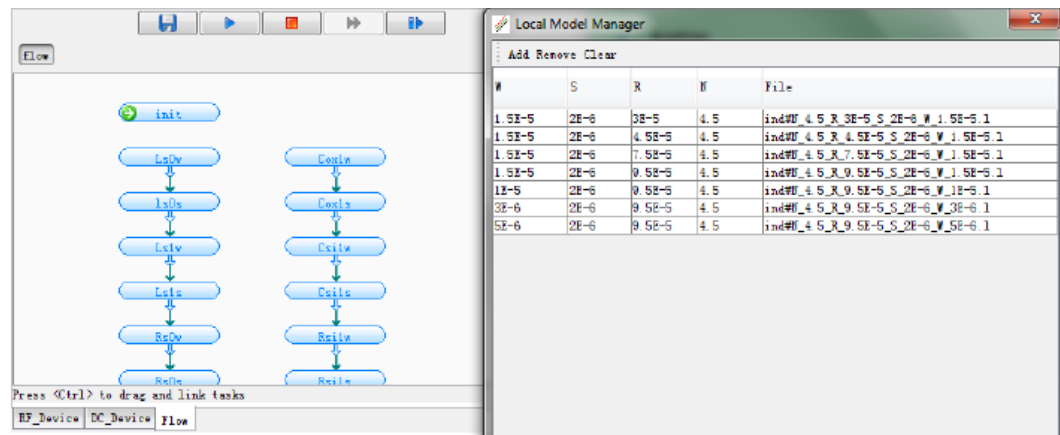


All the point models are saved automatically in the target folder. If the point model is not good, it can be due to the boundary of parameters or the initial model. You can extend the boundary or select the point model of similar size as the initial model, and run the flow again.

Scalable Project

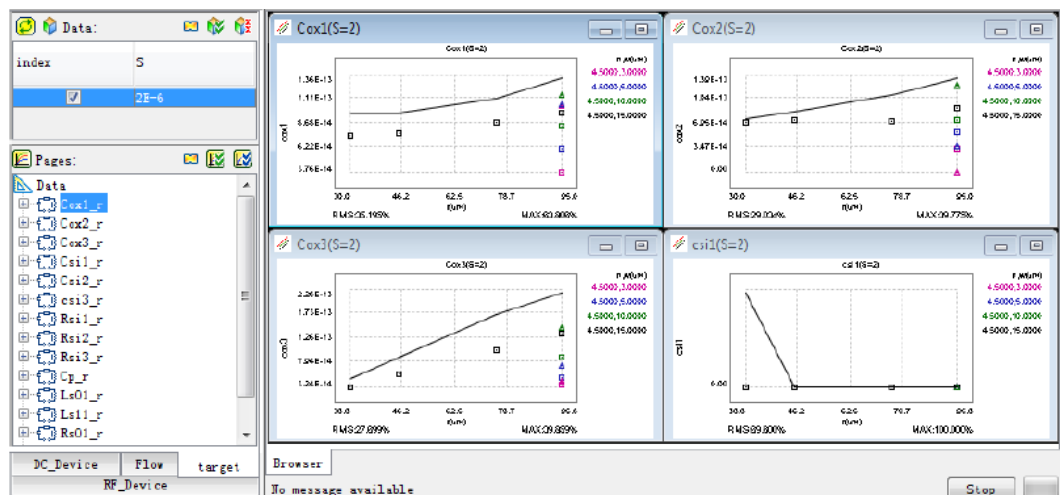
All the generated point models are optimized with different instance parameters. You can open the scalable project by choosing Model > Select Model > RF > inductor > standard > scalable from the main menu. Select Model > Local Model Manager > Add and load the point models with the same turn number. In this example, the point models with $n=4.5$ are loaded, as shown in following figure.

Load point models



Click Navigator > target from the main menu. Check the target versus instance parameters, as shown in following figure.

Target vs. instance parameters

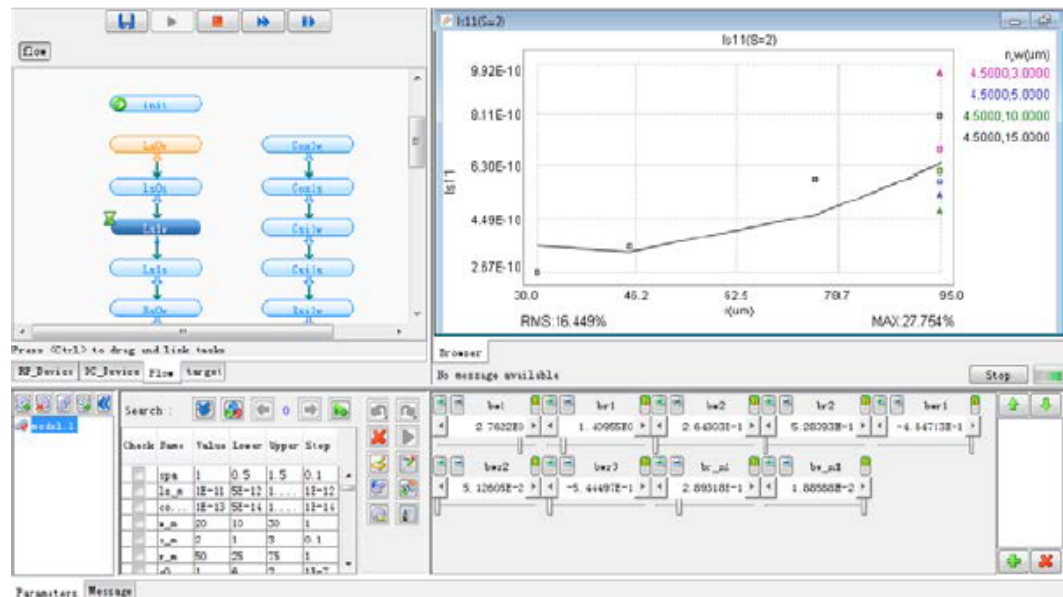


Run the extraction flow or single task to fit devices in all geometries. In the first step (init) of this flow, a table of all the scalable parameters pops up, as shown in [Figure: Scalable Parameter Table](#). Close this window and click the step Ls0w to continue. The flow runs, as shown in [Figure: Run Extraction Flow](#). Modification of parameters boundary and initial values can improve the fitting result.

Scalable Parameter Table

	params
ls0w	a0,a1,a2,a3,a4,a5,a6,aw...
ls0s	as1,as2,awrs1,aw_n2,ar...
rs0w	e0,e1,ew1,ew2,er1,er2,e...
rs0s	es1,es2,ewrs1,ew_n2,er...
ls1w	bw1,br1,bw2,br2,bwr1,b...
ls1s	bs1,bs2,brsw1,bw_n2,br...
rs1w	dw1,dr1,dw2,dr2,dwr1,d...
rs1s	ds1,ds2,drsw1,dw_n2,dr...
cox3w	gr1,gw1,grw1,gr2,gr_n1,...

Run Extraction Flow

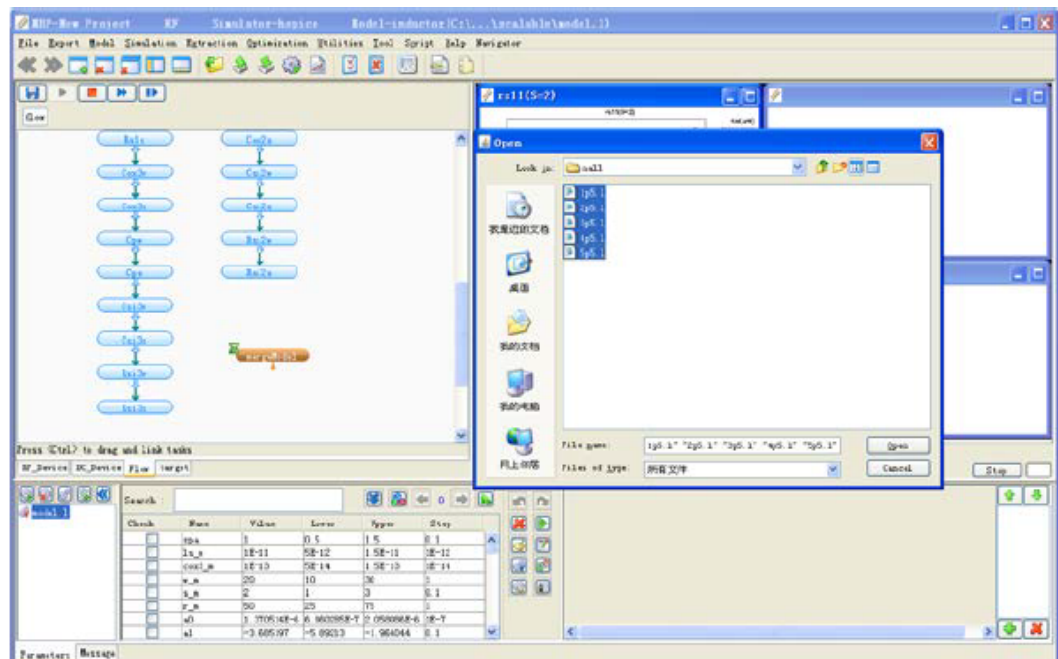


Save the scalable model when all the tasks are completed. There are some requirements for the name of the scalable model. If the point models' turn number is 4.5, the scalable model should be named as 4p5.l.

Clear the point models in the local model manager and other load point models with different turn number, for example, the point models with $n=3.5$. Repeat the steps and save the generated scalable model as 3p5.l.

Finally, run the task named mergeModel to merge the scalable models. Select all the scalable models with different turn numbers, as shown in following figure. Save the final model.

Merge the model




NOTE

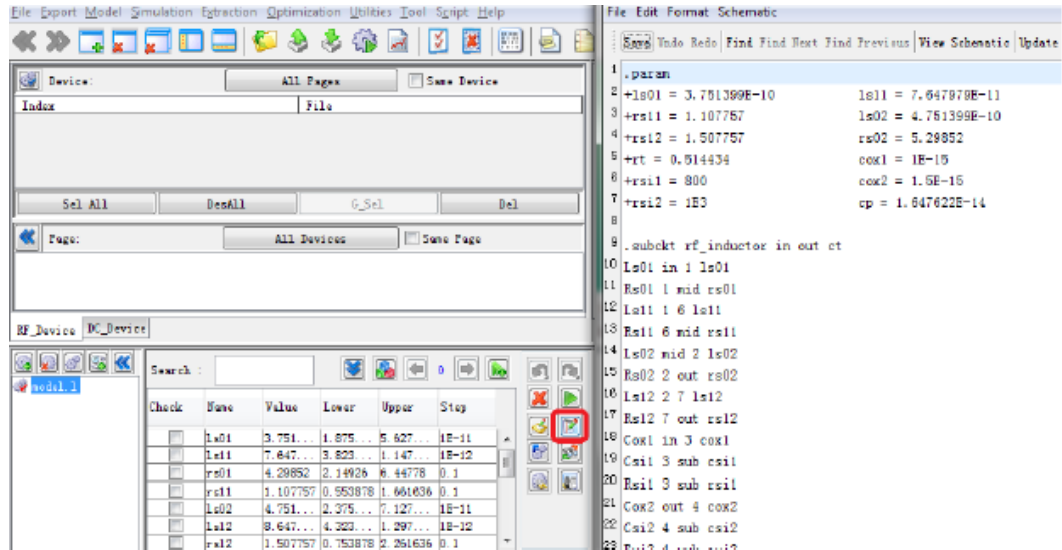
- Pay attention to the boundary of parameters. They are varied for different data.
- Deselect the **bad** point model that deviates from the line.
- If the result is not acceptable, the scalable function should be modified for better result.
- If the number of point models with turn number is high, it is given preference over other point models with turn number. For example, there are eight-point models whose turn number is 4.5 and seven-point models whose turn number is 3.5. In this case, select point models whose turn number is 4.5 prior to 3.5.

3-port Inductor

Choose Model > Select Model > RF > inductor > 3port from the main menu to open the default project. Check the default 3-port inductor model by clicking the Edit

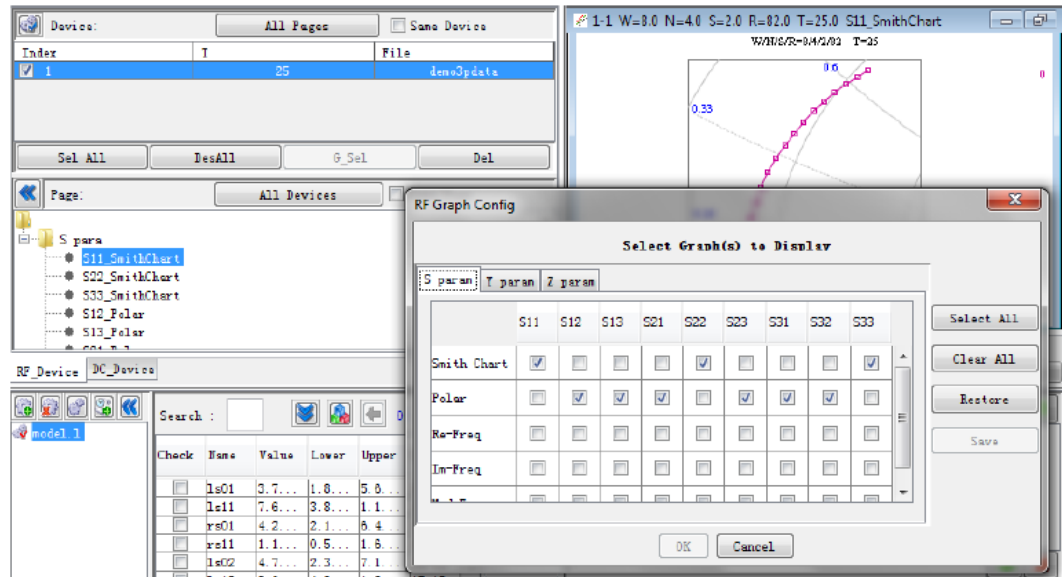
Model icon , as shown in following figure. You can load the user-defined model freely.

Default 3-port inductor model



Choose File > Data > Load to load the measurement data. The demo data is located under modelbuilder\demo\RF\inductor\3port. The RF data conversion window can be enabled by choosing Tool > GUI Options > RF Graph Config from the main menu, as shown in following figure.

RF Graph Config



Parameter Conversion

RF Parameters Supported in MBP

MBP supports different kinds of parameters such as S, Y, Z, T, and H. The parameter conversion equations are listed in following table.

Table: RF Parameter Conversion

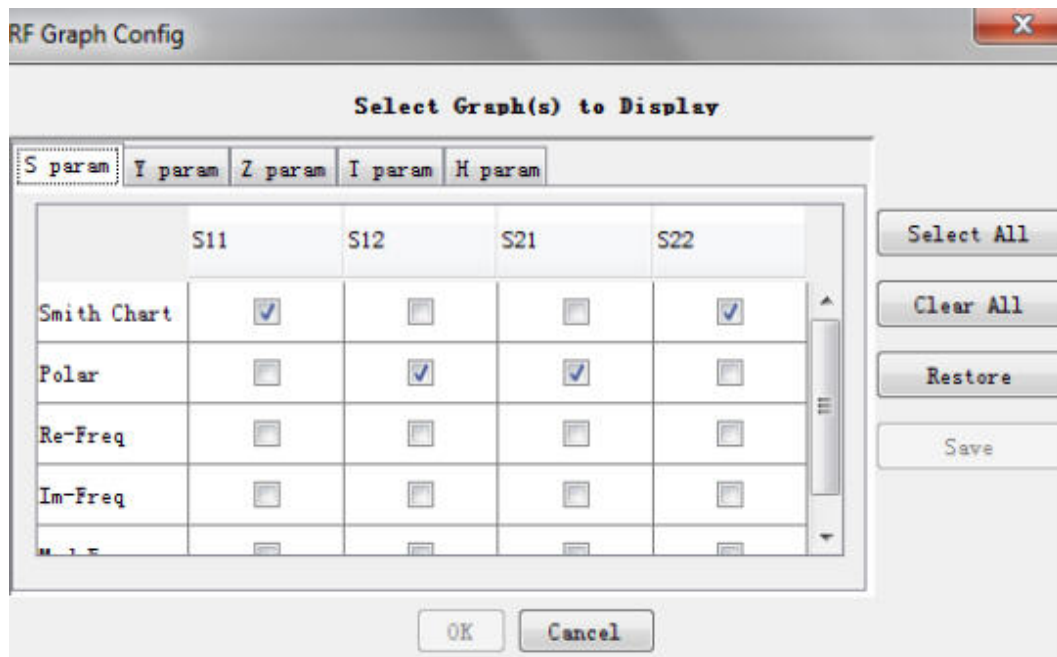
T		Y	
S11	$\frac{A + B/Z_0 - CZ_0 - D}{A + B/Z_0 + CZ_0 + D}$	$\frac{(Y_0 - Y_{11})(Y_0 + Y_{22}) + Y}{(Y_0 + Y_{11})(Y_0 + Y_{22}) - Y}$	
S12	$\frac{2(AD - BC)}{A + B/Z_0 + CZ_0 + D}$	$\frac{-2Y_{12}Y_0}{(Y_0 + Y_{11})(Y_0 + Y_{22}) - Y}$	
S21	$\frac{2}{A + B/Z_0 + CZ_0 + D}$	$\frac{-2Y_{21}Y_0}{(Y_0 + Y_{11})(Y_0 + Y_{22}) - Y}$	
S22	$\frac{D + B/Z_0 - CZ_0 - A}{A + B/Z_0 + CZ_0 + D}$	$\frac{(Y_0 + Y_{11})(Y_0 - Y_{22}) + Y}{(Y_0 + Y_{11})(Y_0 + Y_{22}) - Y}$	
Z		H	
S11	$\frac{(Z_{11} - Z_0)(Z_{22} + Z_0) - Z_{12}Z_{21}}{(Z_{11} + Z_0)(Z_{22} + Z_0) - Z_{12}Z_{21}}$	$\frac{(H_{11} - Z_0)(H_{22} + 1/Z)}{(H_{11} + Z_0)(H_{22} + 1/Z)}$	
S12	$\frac{2Z_{12}Z_0}{(Z_{11} + Z_0)(Z_{22} + Z_0) - Z_{12}Z_{21}}$	$\frac{2H_{12}}{(H_{11} + Z_0)(H_{22} + 1/Z)}$	
S21	$\frac{2Z_{21}Z_0}{(Z_{11} + Z_0)(Z_{22} + Z_0) - Z_{12}Z_{21}}$	$\frac{2H_{21}}{(H_{11} + Z_0)(H_{22} + 1/Z)}$	

	T	Y
S22	$\frac{(Z_{11} + Z_0)(Z_{22} - Z_0) - Z_{12}Z_{21}}{(Z_{11} + Z_0)(Z_{22} + Z_0) - Z_{12}Z_{21}}$	$\frac{(H_{11} - Z_0)(1/Z_0 - H_{22})}{(H_{11} + Z_0)(H_{22} + 1/Z_0)}$

Parameter Conversion in MBP

In MBP, you can convert S-parameter to other type's parameter easily. Select Tool > GUI options > RF Graph Config from the main menu (as shown in following figure) and then choose the parameter/plot type to be displayed.

RF Parameter Conversion



References

1. Reydezel Torres-Torres, "Analytical Model and Parameter Extraction to Account for the Pad Parasitics in RF-CMOS" IEEE TRANSACTIONS ON ELECTRON DEVICES, VOL. 52, NO. 7, JULY 2005.
2. Hanjin Cho and Dorothea E. Burk, "A Three-Step Method for the De-Embedding of High-Frequency S-Parameter Measurements" IEEE TRANSACTIONS ON ELECTRON DEVICES. VOL. 38. NO. 6, JUNE 1991.
3. Ewout P. Vandamme, Dominique M. M.-P. Schreurs, and Cees van Dinter, "Improved Three-Step De-Embedding Method to Accurately Account for the Influence of Pad Parasitics in Silicon On-Wafer RF Test-Structures" IEEE TRANSACTIONS ON ELECTRON DEVICES, VOL. 48, NO. 4, APRIL 2001.

RF Data and Simulation

MBP provides complete RF modeling platform for various devices such as BJTs, MOSFETs, inductors, resistors, and capacitors. This section provides a brief introduction to the data structure.

Data Format

An example of RF data file supported by MBP is shown below.

[General]

```
Datatype:rf // Fixed for RF data.
Version:2.0 // Can be 1.0 or 2.0. It is related to the
rfoutput option in the Measurement Condition section.
Operator: Date:
Lot: Wafer: Die:-1,0
Tile:label 0
Device-type:mosfet // The device type can be mosfet, bjt,
capacitor, and so on.
Type:nmos // The polarity of the device. For example, nmos
or pmos for mosfet, npn or pnp for bjt....
Nodes:
Instances:w=32,l=0.25,nf=8,ad=2e-11,as=5e-11,pd=5e-5,ps=5e-5
Temperature:27
Process:
Ports:g,d // Node info. Match with MBP setup files, etc. Z0:
50
```

[Measurement Condition]

```
DataName:Sparameter // Need to be consistent with the
definition in the file modelbuilder\etc\analysis.xml
Measurement-condition:Sparameter
Biasing-type:Sparameter
Biases:vgs,Vbs,Vds
rfoutputs:S11_r,S12_r,S21_r,S22_r // The rfoutputs should
be like S for version 1.0 and SXX_r, SXX_m for v2.0.
```

[DC Data]

[RF Data]

```
{Vds=0,vgs=0,Vbs=0,x=freq}
1.0E8, 0.999983, -0.005338426, 1.184265E-5, 0.002201805,
1.184264E-5, 0.002201805,
0.999982, -0.005427228
4.98E8, 0.999585, -0.0265799,2.935445E-4, 0.0109595,
2.935442E-4, 0.0109595,
0.99956, -0.0270204
8.96E8, 0.998659, -0.0477995, 9.494757E-4,0.0196952,
9.494748E-4, 0.0196952,
```

```

0.998578, -0.0485849
.....
{Vds=0,vgs=0.25,Vbs=0,x=freq}
1.0E8, 0.999984, -0.005277605, 1.177794E-5, 0.002201806,
1.177793E-5, 0.002201806,
0.999982, -0.005427228
4.98E8, 0.999594, -0.0262772, 2.919413E-4, 0.0109595,
2.919411E-4, 0.0109595,
0.99956, -0.0270204
.....

```

There are four sections in an RF data file:

- [General]: Describes general information such as device type, instance, temperature, and so on.
- [Measurement Condition]: The example describes the measurement condition of an S-parameter data file.
- [DC Data]: It is optional in the RF data file. The example does not contain DC data.
- [RF Data]: The RF data is displayed in the following sequence: *frequency*, *S11r*, *S11i*, *S12r*, *S12i*, *S21r*, *S21i*, *S22r*, *S22i* in every line.

NOTE

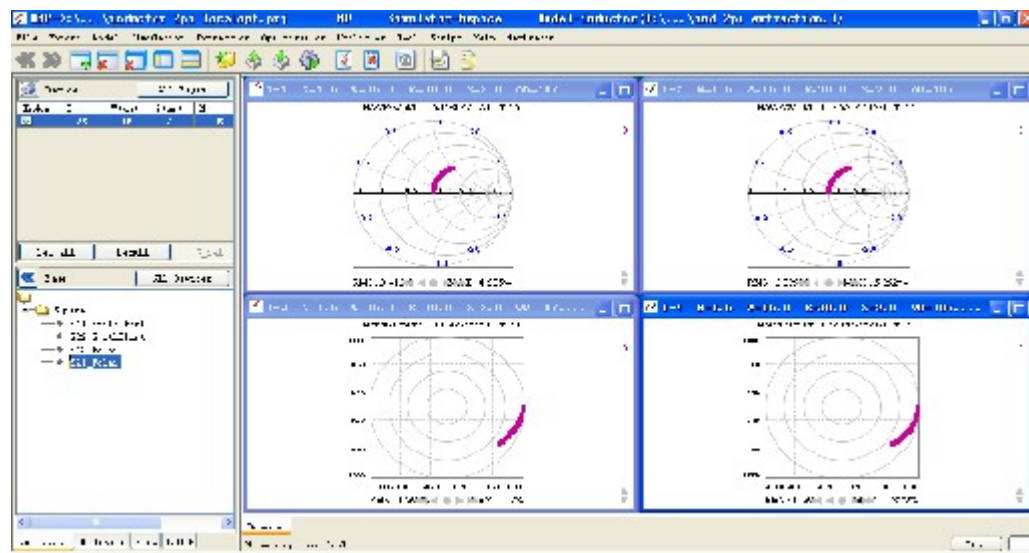
Currently, MBP supports S-parameter or Y-parameter as the input data. If the raw data contains other kinds of parameters such as H or Z, convert them into S-parameter or Y-parameter first.

S-parameter Simulation

You can check the S-parameter simulation result with different plot styles, including Smith Chart, Polar Chart, or simple X-Y graph.

One example of inductor is shown in following figure.

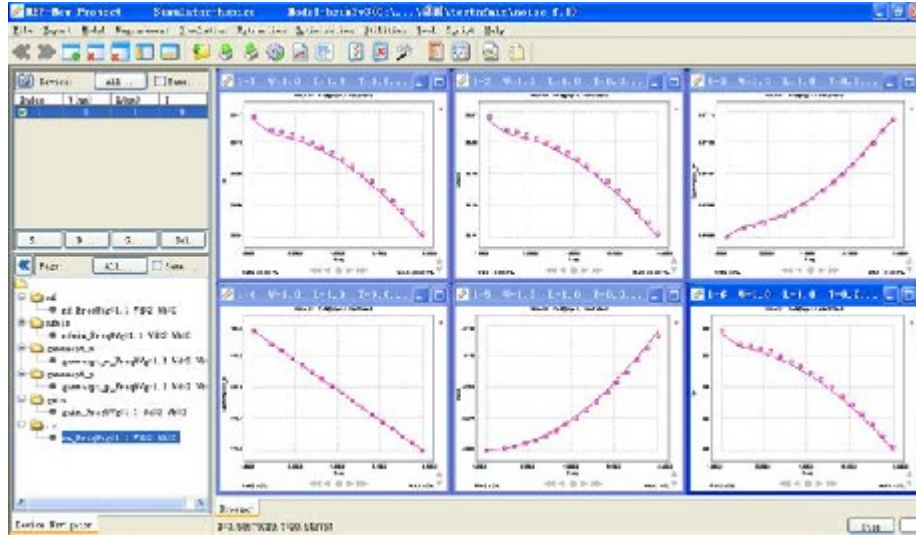
S-parameter simulation



Noise Simulation

MBP supports noise targets in RF package, including $NF(dB)$, $NFmin(dB)$, $gamma_opt(magnitude)$, $gamma_opt(phase)$, and $g_as(dB)$. One noise simulation result is shown in following figure.

Noise Simulation



NF (Noise Figure) is calculated as:

$$NF = 1 + \frac{N_a}{G_a k T_0 \Delta f}$$

Here,

- N_a is the available noise power at the output of the two-port (due to the two-port's noise and not due to the input impedance).
- G_a is the available power gain.
- k is the Boltzmann's constant.
- T_0 is the 290 Kelvin reference temperature.
- $NFmin$ is the minimum Noise Figure.
- $gamma_opt$ means the reflection coefficient values, which if presented at the input of the noisy two-port, results in the best noise performance.
- g_as is named Associated Gain, which assumes that the in-out impedance is matched for the minimum noise figure (that is, $gamma_opt$), while the output is matched for the maximum gain.

External Simulator

MBP can call external simulators (HSPICE or SPECTRE) for simulation.

For the local one (the simulator and MBP are installed on the same PC/Server), ensure that the simulator can be run in batch mode before switching the simulator to external by choosing Simulation > Simulator > External HSPICE (or SPECTRE) from the main menu.

For the remote one (the simulator and MBP are not installed together, but you can call the simulator from remote server), choose Simulation > External SPICE options from the main menu, as shown in following figure. Ensure that the network supports Remote Call command (such as TELNET, RSH, SSH, and so on) and Remote Transfer command (such as FTP, RCP, and SCP). Input Remote IP, User name, Password, and so on. Check Call Remote option; then click the button Check . If MBP responds Success, you can switch to external simulator by choosing Simulation > Simulator > External HSPICE (or SPECTRE) from the main menu.

External SPICE options

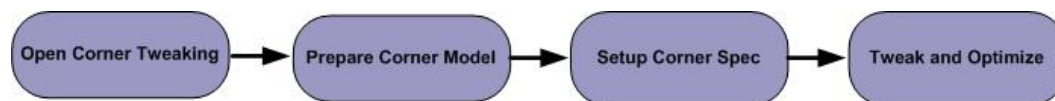
The screenshot shows the 'External SPICE options' dialog box. It features four tabs: 'Hspice', 'Spectre', 'ADS', and 'Spice3'. The 'Hspice' tab is active. The dialog includes a 'Call Remote' checkbox, which is checked. Below this are several input fields: 'Remote Call' (a dropdown menu set to 'TELNET'), 'Remote Transfer' (a dropdown menu set to 'FTP'), 'Remote IP' (text input '192.138.0.180'), 'User Name' (text input 'test'), 'Password' (text input '***'), 'Command' (text input 'hspice'), 'Prompt' (text input '\$'), and 'Directory' (empty text input). A 'Check' button is located at the bottom right of the form area. At the bottom of the dialog, there is another 'Is Delete Netlist Files' checkbox, which is also checked, and 'OK' and 'Cancel' buttons.

Corner Modeling

Corner Modeling

Load the project from `${MBP_HOME}/demo/Corner/mosfet/corner.prj` location, comprising the default settings for corner tweaking.

Corner tweaking is a four step procedure, as depicted below.



Step 1: Open the corner tweaking window

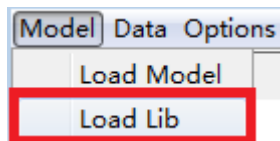
1. Select Extraction > Corner Tweaking from MBP window. This displays the corner tweaking window.

Step 2: Prepare corner model

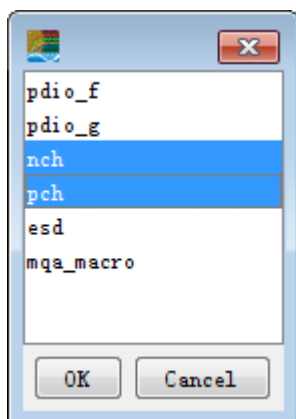
You can prepare the corner model either via a library or from TT model. Details of both the approaches are described below:

Loading Corner model from library

1. Select Model > Load Lib to open the lib file.

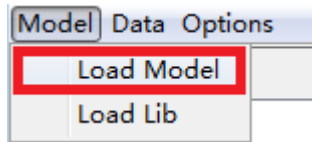


2. Select the required model, then click OK to load all the corners in MBP.

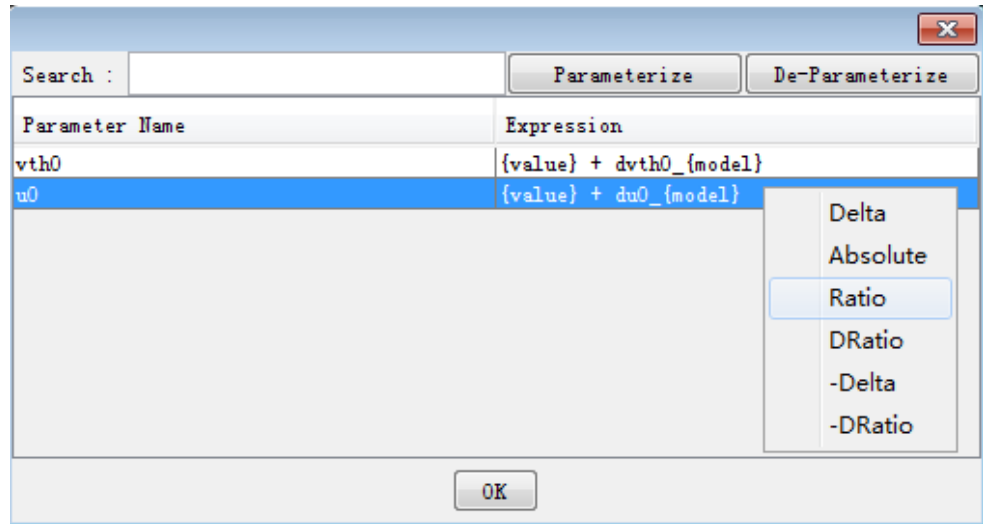


Creating corner model from TT model

1. Select Model > Load Model to select TT model files.



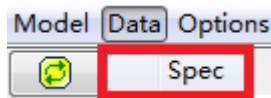
2. Edit and confirm the parameterize settings. You can edit an expression or select a predefined expression(s).



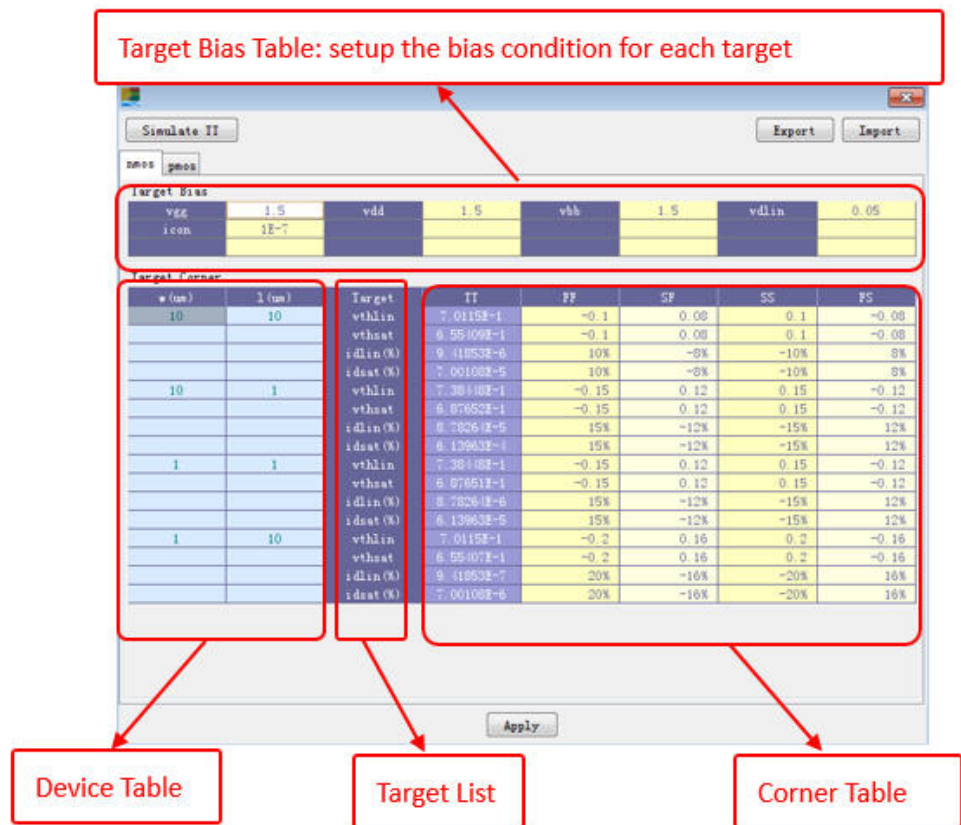
3. Click OK to create all the corners and load into MBP.

Step 3: Setup Corner Spec

1. Select Data > Spec to open the spec setup dialog box.



2. Edit the spec.



FF, SF, SS, FS columns in the Corner Table display the shift value from TT. In addition, you can also view the equation on selecting a respective cell, as depicted below:

Target	TT	FF
vthlin	7.0115E-1	-0.1
vthsat	6.55409E-1	-0.1
idlin(%)	9.41853E-6	10%
idsat(%)	7.00108E-5	TT*(1+10%)
vthlin	7.38448E-1	-0.15

You can use Ctrl+C, Ctrl+V shortcuts to copy/paste on the Corner Table.

a. Edit biases.

Double-click on the Target Bias Table to change the bias name value.

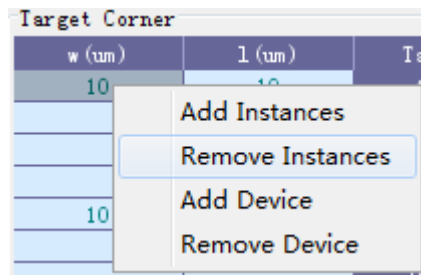
Target Bias							
vgg	1.5	vdd	1.5	vbb	1.5	vdlin	0.05
icon	1E-7						

b. Edit device.

Double-click on the Device Table to change the instance value.

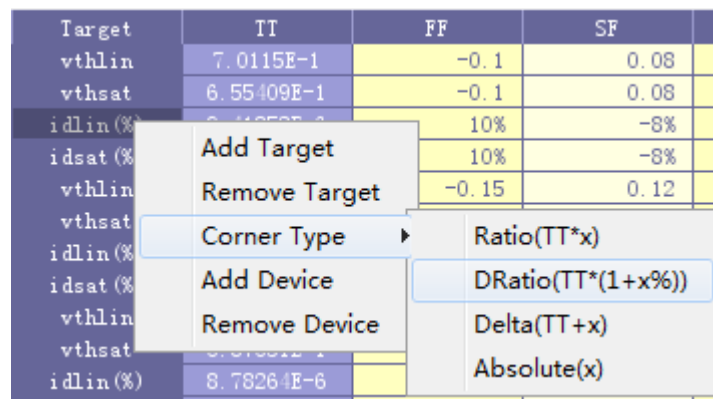
Target Corner							
w (um)	l (um)	Target	TT	FF	SF	SS	FS
10.0	10	vthlin	7.0115E-1	-0.1	0.08	0.1	-0.08

Right-click on the Device Table to add a new instance/device, or remove the selected instance/devices.



c. Edit Target.

Right-click on the Target List to add/remove targets and setup the corner equation.

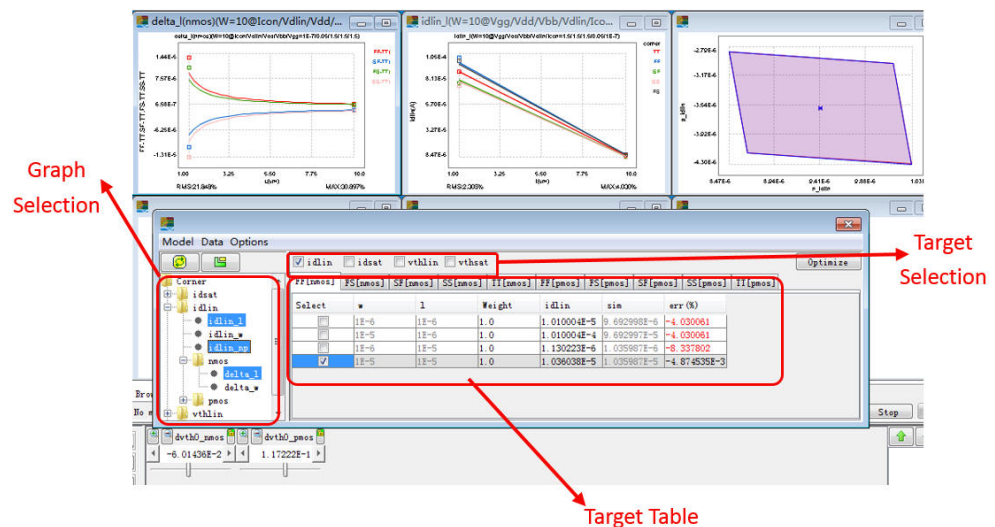


3. Apply the spec.

When the spec is ready, click Apply to generate the data.

Step 4: Tweak and Optimize

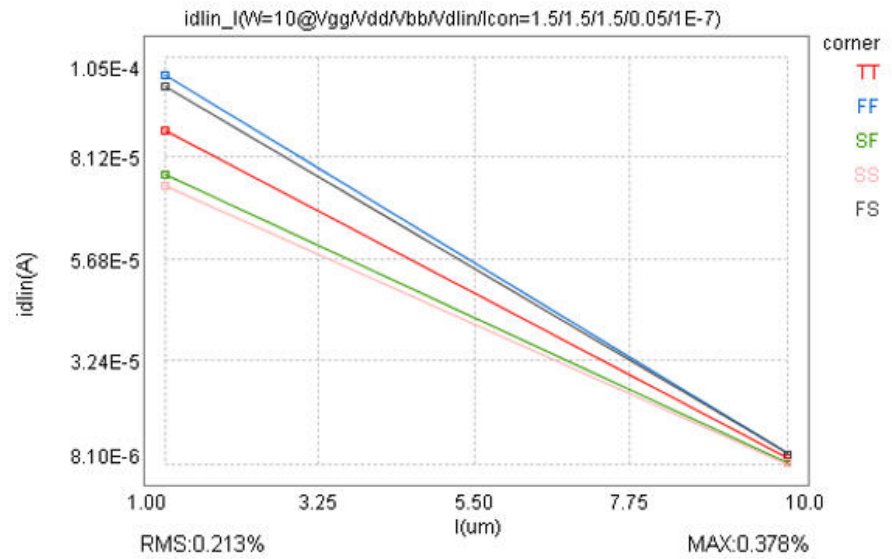
1. Overview



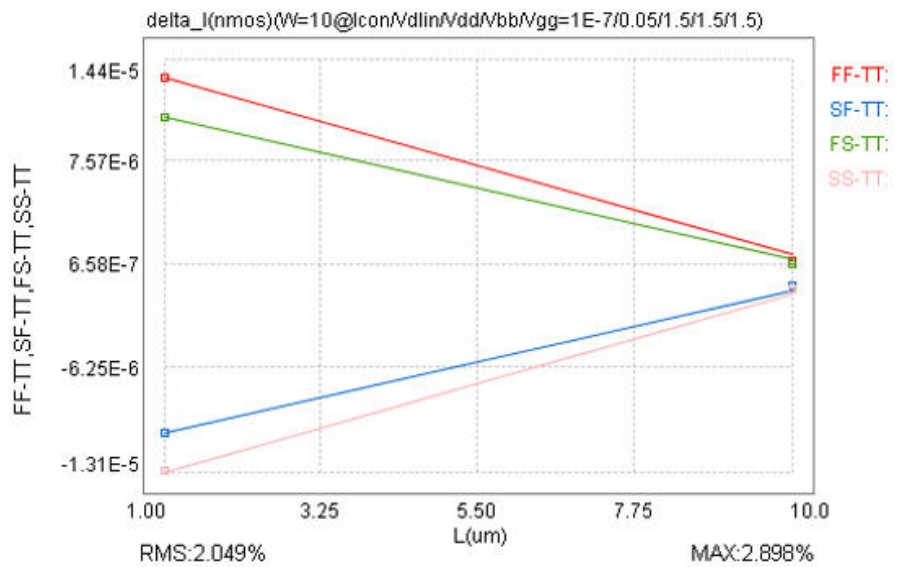
2. Graph

You can select the graph from the tree. The default setting comprises three kinds of graphs:

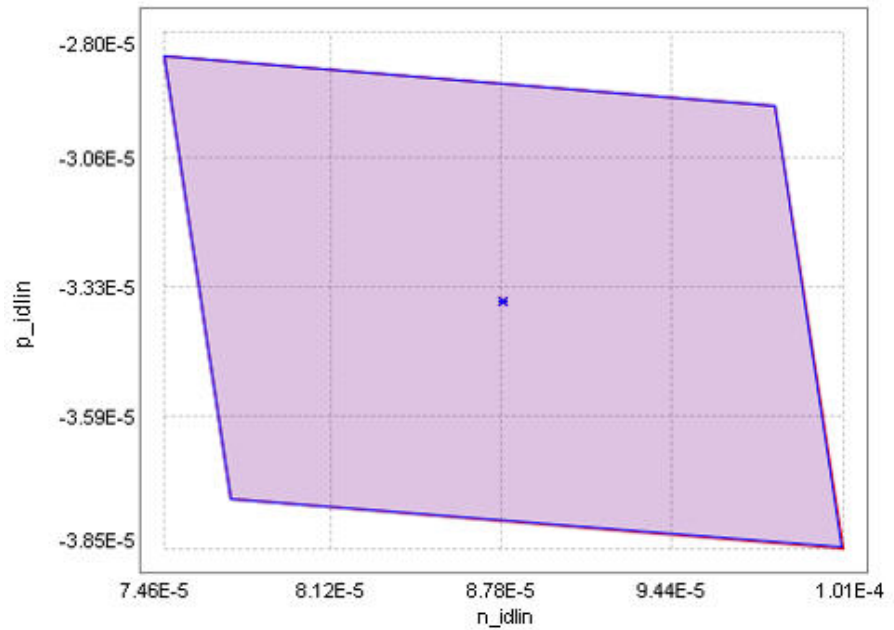
a. Target Trend



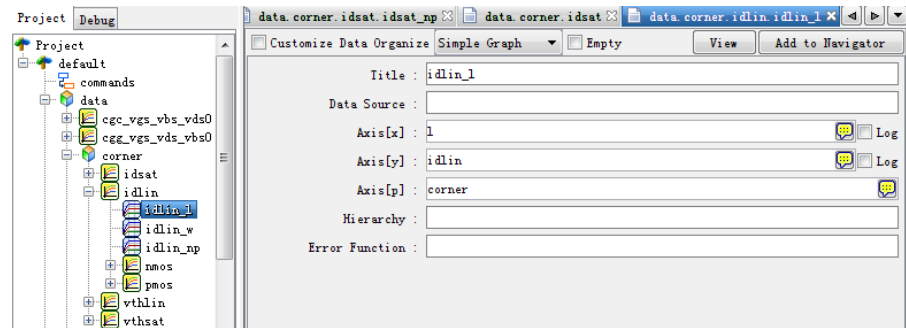
b. Corner-TT Trend



c. Polygon

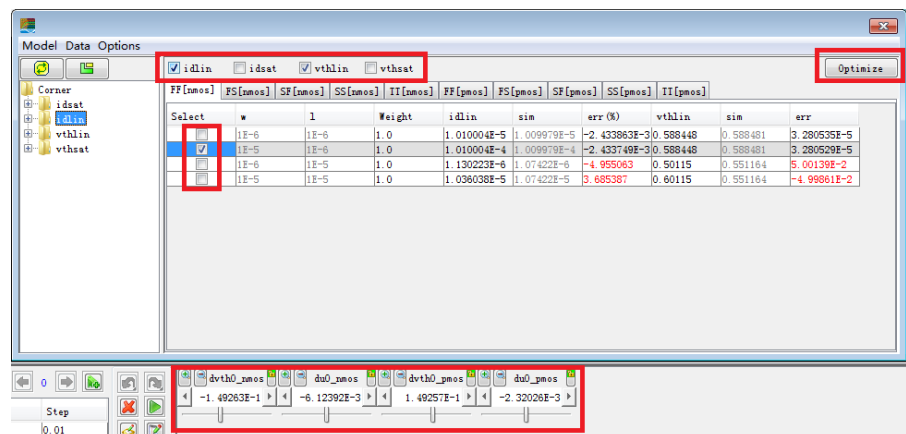


You can add more graphs and script (define the data and graphs under "data/corner"):



3. Optimization

- a. From the MBP window, select parameters.
- b. From Corner Tweaking window, select the targets and devices.
- c. Click Optimize, to optimize all the corners.



This information is subject to change
without notice.

www.keysight.com

