APPLICATION NOTE

Building Efficient and Scalable Extraction Flows in IC-CAP

KEYSIGHT

Automated Model Extraction

Accurate and scalable compact model extraction is essential for developing reliable semiconductor Process Design Kits (PDKs). As device geometries, operating conditions, and technology nodes evolve, extracting compact model parameters across multiple device geometries and temperatures can become increasingly complex. Manual parameter tuning and improperly organized extraction sequences can make the process time-consuming, difficult to reproduce, and challenging to maintain. Keysight Integrated Circuit Characterization and Analysis Program (IC-CAP) Model Generator (MG) provides a structured environment for developing automated compact model extraction flows. By organizing extraction into sequential steps, defining parameter ranges, and using optimization algorithms, engineers can establish repeatable workflows for calibrating compact models against measured device characteristics.

This application note presents a novel systematic approach to building efficient and scalable extraction flows for CMC standard BSIM-BULK v107.1 and PSP v103.8.2 compact models in MG using Keysight Machine Learning (ML) optimizer. The proposed flow focuses on parameter values and ranges initialization, extraction-step organization, geometry and temperature scaling, and useful tips on debugging the extraction flow. The flow chart in Fig. 1 shows the steps taken during the implementation of these new flow examples in IC-CAP 2027.

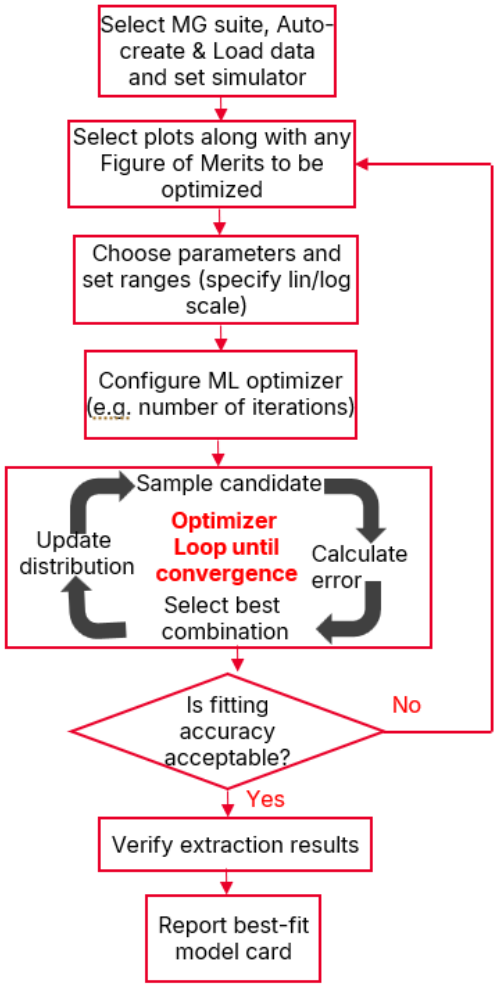


Figure 1. Flowchart showing the steps during the flow implementation in MG

Challenges in Compact Model Extraction

A robust compact model extraction flow must address several practical challenges:

- **Parameter complexity:** Selecting appropriate model parameters and defining valid search ranges for each extraction stage.
- **Multi-geometry characterization:** Extracting baseline physical parameters and subsequently optimizing length, width, and minimum-length device scaling parameters.
- **Temperature dependence:** Separately extracting nominal-temperature extraction from temperature-scaling parameter extraction.
- **Optimization stability:** Maintaining consistent starting values for each step and avoiding unacceptable parameter combinations.

- **Flow scalability:** Adapting the extraction methodology to different device geometries, technology nodes, and complex measurement datasets for a variety of PDKs
- **Debugging and repeatability:** Identifying the source of fitting errors and ensuring that changes to one extraction step do not affect other steps.

A structured MG flow addresses these challenges by dividing the extraction process into a few manageable stages, each with clearly defined parameter selections, ranges, and fitting objectives.

Developing an Automated Extraction Flow in Model Generator

Prepare Measurement Data

The first step is to prepare the measured device data for loading into MG. The measurement datasets should be organized in .mdm files so that the required device geometries, electrical characteristics (IV/CV), and global operating conditions can be associated with the corresponding extraction steps. MG allows for automated data loading using “Auto Create and Load” button once required .mdm files have been prepared.

Initialize the Model

The initialization step establishes the starting point for the extraction flow. In the BSIM-BULK and PSP examples, this step loads the model parameter initial values and their corresponding ranges, along with the circuit file containing a nmos DUT, used for simulation.

The initialization step uses two files:

- **bsimbulk~with_ranges.mps** — Contains model parameter values and their min/max ranges.
- **BSIMBULK.cir** — Defines the circuit used for model simulations. It can represent a standalone model or reference a larger PDK library section.

```
# update: explicit
from icutils import icmg
mytools = icmg.IcMgExtractUtilities()
mytools.prepare_extraction()
```

Parameter ranges can be defined programmatically in the .mps file or entered through MG's Main Parameter Table and saved for subsequent use. Whenever possible, existing model parameter values and documented ranges provide a useful starting point. These ranges can then be refined based on practical extraction experience and the characteristics of the target device.

For BSIM-BULK and PSP, these initialization files must correspond to the selected compact model and simulation setup. Additionally, you must configure scaling plots in MG for key Figure-of-Merits (FOMs) such as Vtlin, Vtsat, Idlin, Idsat, Gmpeak, Gds0 etc.

Configure Parameter Ranges

Parameter ranges directly influence the solutions explored by the optimizer. Ranges that are excessively broad can increase the difficulty of finding a suitable fit, while ranges that are too restrictive can exclude useful parameter combinations. A practical approach is to begin with established parameter values and ranges such as default values from CMC model documentation, evaluate the extraction results, and progressively refine the search space.

Recommended practices include:

- Use physically reasonable parameter ranges.
- Preserve known-good parameter values as initial conditions where appropriate.
- Review parameters that repeatedly approach their range limits.
- Narrow ranges during final fine-tuning when supported by the fitting results.
- Revisit ranges when migrating the flow to a different technology node or device type.

Organizing the Extraction Sequence

The organization of extraction steps is critical when fitting a large number of compact model parameters in a single optimization step using ML optimizer. Separating baseline wide-long geometry extraction from geometry and temperature scaling helps establish a consistent progression from foundational device behavior to more complex dependencies. Below we discuss the extraction sequence as shown in Fig. 2

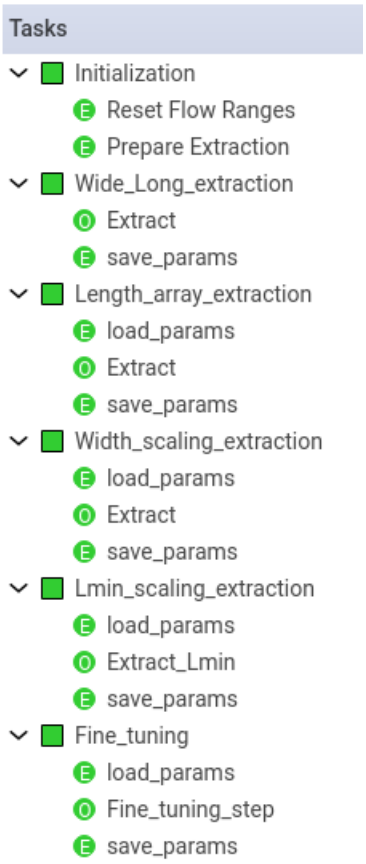


Figure 2. Typical extraction sequence shown for BSIM-BULK model in MG

Recommended BSIM-BULK/PSP Extraction Sequence

Step 1: Initialize Parameters

Load the initial model parameter values and define the corresponding optimization ranges using the built-in python functions.

Step 2: CV Extraction

Begin with capacitance-voltage (CV) extraction if CV data is available. This step focuses on charge-capacitance parameters and helps establish appropriate initial values for the physical model parameters. For BSIM-BULK flow, it is important to establish the physical values of these parameters since no CV data is available for their extraction in this example shipped in IC-CAP 2027.

Step 3: Wide-Long Extraction

Perform the initial DC current-voltage (IV) extraction using the widest and longest device geometry. This step establishes the baseline parameters with highest sensitivity before geometry-

scaling extraction begins. To guide the optimizer to obtain reasonable results, it is best to use the FOM scaling plots as optimization targets in addition to device characteristics.

Step 4: Length-Array Extraction

Extract length-scaling parameters using devices spanning the shortest to the longest channel lengths at a fixed, widest device width. This is typically one of the most important and longest running scaling steps in both BSIM-BULK and PSP example flows.

Step 5: Width-Array Extraction

Extract width-scaling parameters using devices ranging from the narrowest to the widest width at the longest channel length.

Step 6: Lmin-Array Extraction

Perform the minimum-length array extraction after the length- and width-scaling geometries have been addressed. This step should contain the smallest and narrowest geometries. Additionally, fine-tuning of some of the parameters extracted previously can also be combined with this step provided their ranges are further narrowed down.

Step 7: Temperature Scaling

Once the nominal-temperature devices and geometry-scaling parameters have been extracted, extract just the temperature-dependent parameters using MG scaling plots, rather than full IV curves.

This staged sequence provides a systematic path from initial parameter values to baseline electrical behavior, geometry scaling, and temperature dependence.

Using the ML Optimizer Flow Examples

MG includes BSIM-BULK and PSP ML Optimizer examples for bulk CMOS devices that can be launched from the following location:

Extract → Model Generator → Open Example → ICMG_PSP_Example.icproj

The example provides a starting point for exploring the extraction flow. Running the flow using the **Extract** button executes the configured sequence. Figure 3 and 4 below show the final extraction results for BSIM-BULK and PSP models using ML Optimizer in MG.

To adapt the example to a target device, engineers can modify the initialization settings, parameter ranges, point to corresponding model card inside library, and extraction-step configuration to match their measurement data and modeling objectives.

Starting from an existing example can reduce the effort required to establish the initial workflow and provides a consistent structure for subsequent customization.

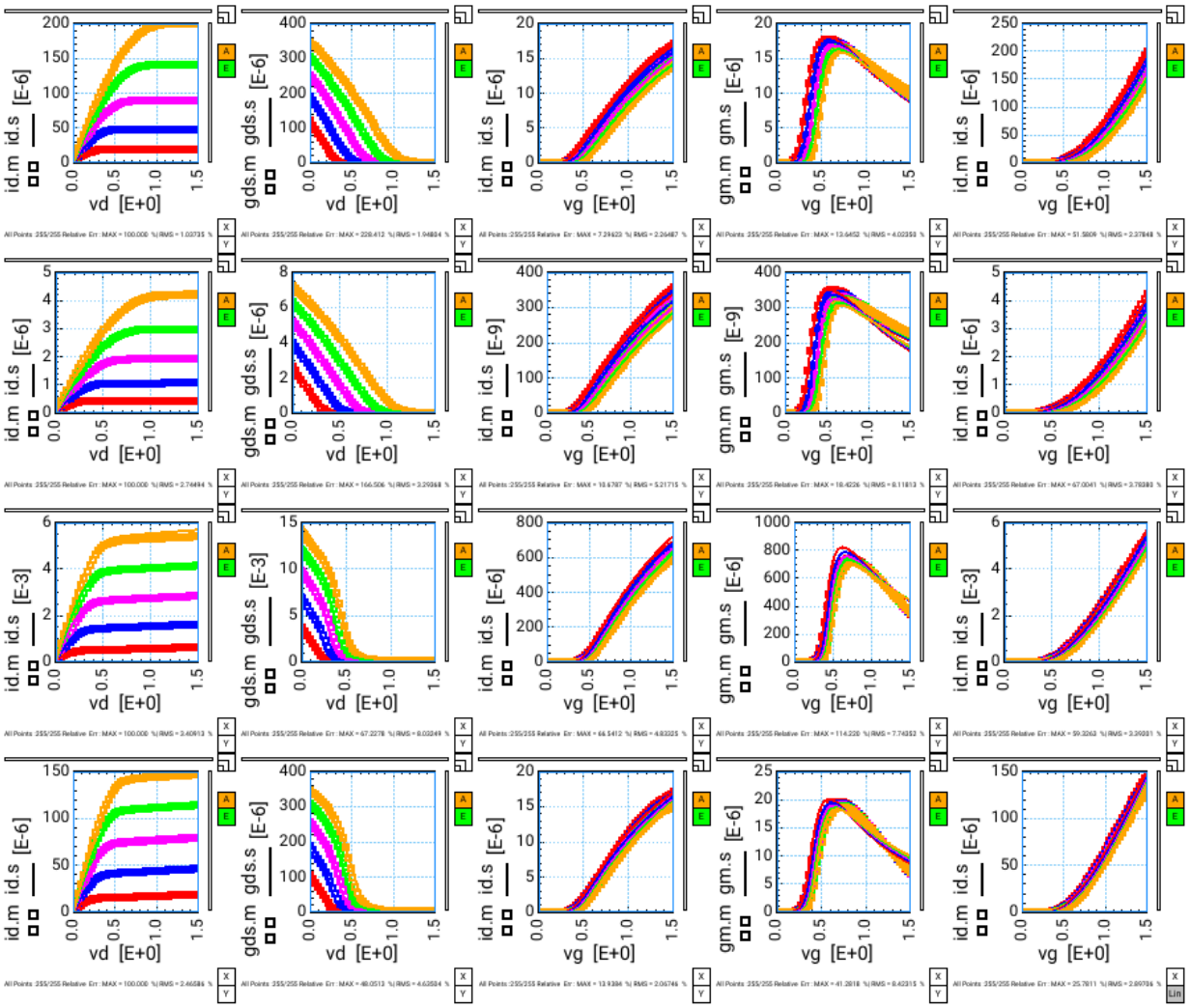


Figure 3. Four corner device fitting results for BSIM-BULK example

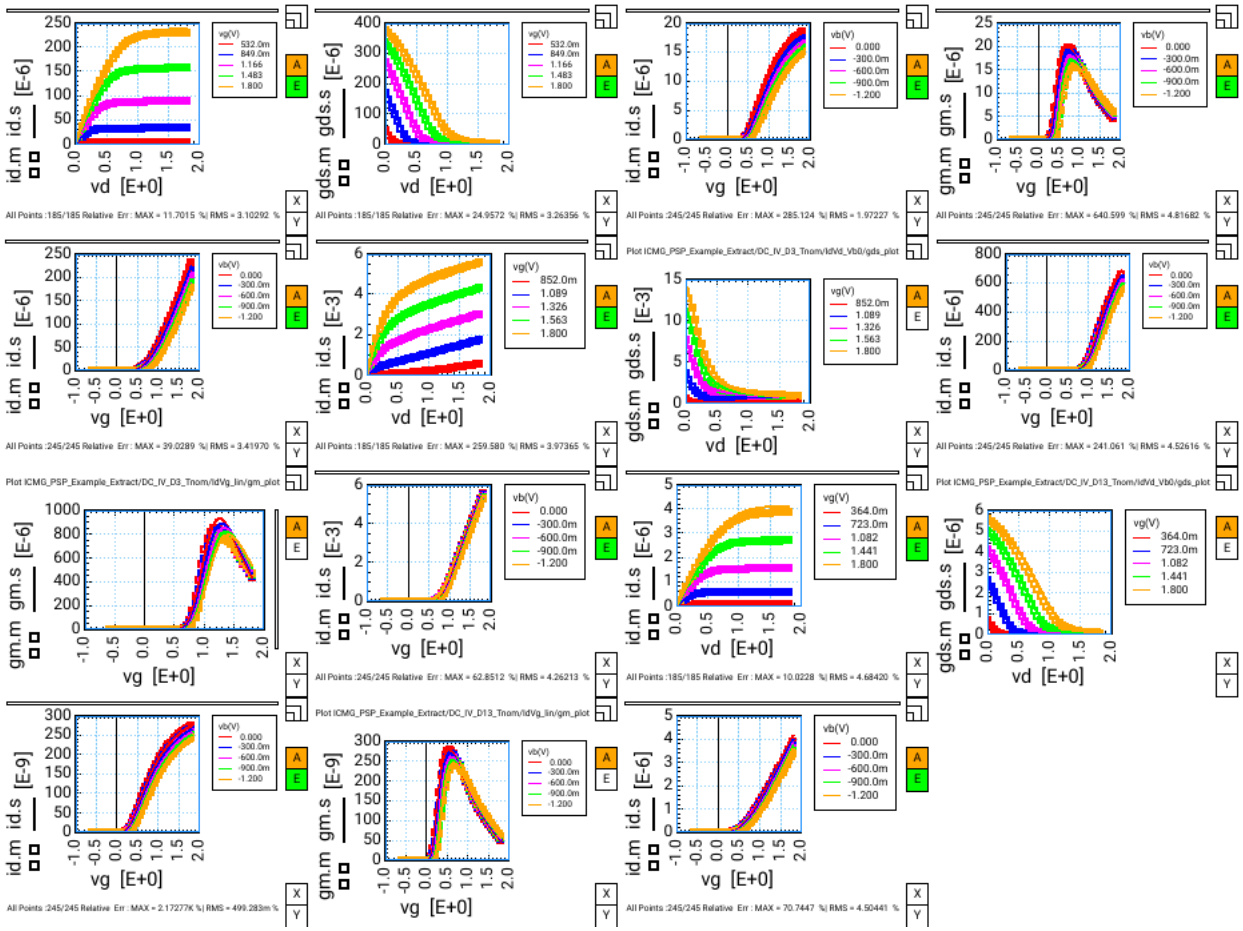


Figure 4. Final fitting results shown for PSP model example

Debugging and Optimizing the Extraction Flow

When the optimizer does not produce the expected extraction results, the following checks can help isolate the issue.

1. **Validate Parameter Ranges**

Review the parameter ranges and confirm that they allow suitable parameter combinations. Incorrect ranges can prevent the optimizer from reaching an acceptable solution.

2. **Review Plot Weights**

Check the weights assigned to scaling plots or individual curves. Adjust these weights as needed to balance the fitting objectives across the measured characteristics.

3. **Verify Extraction-Step Ordering**

Confirm that the extraction steps follow the intended sequence. For example, performing length-array extraction after width-array extraction may produce different results than following the recommended sequence for BSIM-BULK and PSP models.

4. **Revisit Configuration When Changing Technologies**

When migrating a flow to another technology node or device type, review the parameter ranges and fitting weights. Existing settings may not be appropriate for the new characterization data.

5. **Allow Sufficient Optimization Iterations**

Verify that each extraction step has sufficient iterations to converge toward a suitable solution. Other optimizer settings, including the random seed, population size, learning rate and initial error, may also be useful when diagnosing complex extraction cases.

6. **Maintain Consistent Starting Conditions**

For step-level debugging only, use provided scripts such as *load_params* and *save_params* to control parameter loading and saving before and after each. Ensuring that a step begins with the same initial parameter values makes it easier to determine whether a change in configuration improves the result.

Building Reusable and Scalable Flows

A scalable extraction methodology enables a flow to be applied to additional geometries, operating conditions, or technology variants.

The following practices support maintainability and repeatability:

- **Modularize the extraction sequence:** Keep initialization, baseline extraction, geometry scaling, and temperature scaling logically separated.
- **Centralize parameter management:** Maintain parameter values and ranges in clearly defined model configuration files or parameter tables.
- **Use consistent initial conditions:** Make extraction results easier to compare by controlling the starting parameters for each step.
- **Document fitting priorities:** Record the intended purpose of each step and the rationale behind important curve weights and parameter ranges.
- **Validate changes incrementally:** Evaluate individual step modifications before incorporating them into the complete flow.
- **Revisit technology-specific settings:** Review model ranges, weights, and extraction configurations whenever the target device or technology changes.

These practices help engineers maintain a consistent extraction methodology while accommodating different compact models and characterization requirements.

Conclusion

Efficient compact model extraction depends not only on the powerful ML optimizer but also on the quality of the parameter ranges, appropriate weights, and extraction-step organization.

Keysight IC-CAP Model Generator provides a structured environment for developing automated extraction workflows as opposed to traditional back and forth between numerous smaller extraction steps. The BSIM-BULK and PSP ML Optimizer examples shipped in IC-CAP 2027 demonstrate how initialization, CV extraction, baseline IV extraction, geometry scaling, and temperature scaling can be organized into a systematic extraction process. By modularizing these steps, flows are generic and automated thereby shortening the model extraction cycle significantly and less susceptible to variability introduced by the modeling engineer.

Keysight enables innovators to push the boundaries of engineering by quickly solving design, emulation, and test challenges to create the best product experiences. Start your innovation journey at www.keysight.com.

KEYSIGHT

This information is subject to change without notice. © Keysight Technologies, 2026, Published in USA, October 27, 2026, 3126-1852.EN