CCS Power Library Characterization Guideline for Synopsys® Libraries

Version 5.0, August 2016

SYNOPSYS®

Copyright Notice and Proprietary Information

Destination Control Statement

All technical data contained in this publication is subject to the export control laws of the United States of America. Disclosure to nationals of other countries contrary to United States law is prohibited. It is the reader's responsibility to determine the applicable regulations and to comply with them.

Disclaimer

SYNOPSYS, INC., AND ITS LICENSORS MAKE NO WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, WITH REGARD TO THIS MATERIAL, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE.

Trademarks

Third-Party Links

Any links to third-party websites included in this document are for your convenience only. Synopsys does not endorse and is not responsible for such websites and their practices, including privacy practices, availability, and content.

Synopsys, Inc.
690 E. Middlefield Road
Mountain View, CA 94043
www.synopsys.com

Document History

Revision	Changes
1.0	Initial Version
2.0	Update syntax of library with latest version
3.0	Added guidelines for gate leakage, extended macro cell modeling and scaling
4.0	Added switching condition requirement, section 3.3.6
4.1	Strengthened the language on simulation duration, section 3.3.1
5.0	1. Revised guidelines for current waveform segmentation, section 4.6 2. Added characterization guidelines for voltage-dependent intrinsic parasitic, section 3.7 3. Added characterization guidelines for MTCMOS switch cells, section 3.8, and section 4.11

Contents

1. Introduction

Non-linear power models (NLPMs) have long been used to abstract power characteristics of cells for usage in power analysis flows. With the increase of IR drop and ground bounce in the advanced IC technology, the degradation of IC performance that is reflected on increased timing (delay or slew) becomes significant. In order to capture the effect of dynamic IR drop and ground bounce accurately, the current waveform that gates draw from the power or to the ground network is a necessary step from which to start. For those non-switching gates, the intrinsic parasitic seen by power or ground network have to be captured, too. The CCS Power, which is a combination of the current waveform and intrinsic parasitic, has been developed for advanced dynamic rail analysis. Characterization for NLDM, CCS Timing, NLPM, and CCS Power can be done simultaneously, and you can extract CCS Power library data easily from simulation results. This application note focuses on characterization of current waveform, dynamic charge (energy), and intrinsic parasitic. Topics covered include:

- CCS Power library requirements for Power Compiler, PrimeTime PX and PrimeRail
- Key items to consider in library characterization for meeting NLPM and CCS Power requirements
- Techniques that have been used by library experts to characterize libraries
- Library screening checks that are recommended to the library developer

The intended audience of this application note is library developers who characterize libraries that will be used with Power Compiler, PrimeTime PX and PrimeRail.

2. SPICE Netlist or Subcircuit Requirements

Chip-level parasitic extraction looks at standard cell geometries using FRAM view and only considers down to metal-1 geometries. Therefore, it is required that the SPICE netlists or subcircuits used during characterization should include parasitic of PG network in standard cells from metal-1 down to devices. It is recommended that you use Star-RCXT to extract the required SPICE subcircuit in the transistor level.

3. Library Requirements

This section describes the library requirements for power and rail analysis and follows mostly the requirements for NLDM and CCS Timing (see reference[1]). The impacts of these requirements on accuracy are also described. Section 4 provides suggestions for characterization techniques that you can use to generate an accurate CCS Power library.

This section includes the following topics:

- 3.1 Syntax of PG Current Waveforms
- 3.2 Delay and Transition Time Trip Points
- 3.3 Current Waveform Requirements

3.1 Syntax of PG Current Waveforms

The syntax used to describe the PG port current is very similar to the one used in CCS Timing, which is a three-dimensional current table with the time as the third index. This syntax is applied for gates with either non-propagating (that is, internal current or internal power) or propagating events (such as, switching outputs). The syntax used to describe the leakage currents for static gates is described in section 3.4.

```
cell ( cell_name ) {
  dynamic_current ( ) {
    when : state_1;
    related_inputs : related_input_1;
    related_outputs : related_output_1;
    typical_capacitances(“float, ..”);
   switching_group() {
      input_switching_condition(input_sense);
      output_switching_condition(output_sense_1, \
                                 output_sense_2, ...);
      pg_current(pg_pin_1) {
        vector(template_name) {
          reference_time : ref_time_1;
        index_output(index_output_pin_1);
        index_1(index_variable_1);
        index_2(index_variable_2);
        index_3(index_variable_3);
          values(val_list_1);
     }…
      }…
    }…
  }…
}…
```

3.2 Delay and Transition Time Trip Points

For transition time and delay trip points in CCS Power libraries, you must follow the requirements specified in NLDM and CCS Timing characterization guidelines.

3.3 Current Waveform Requirements

The current is measured toward the cell. For power pins, the current is positive if it is drawn to the cell. For ground pins, the current is negative, indicating that current flows out of the cell.

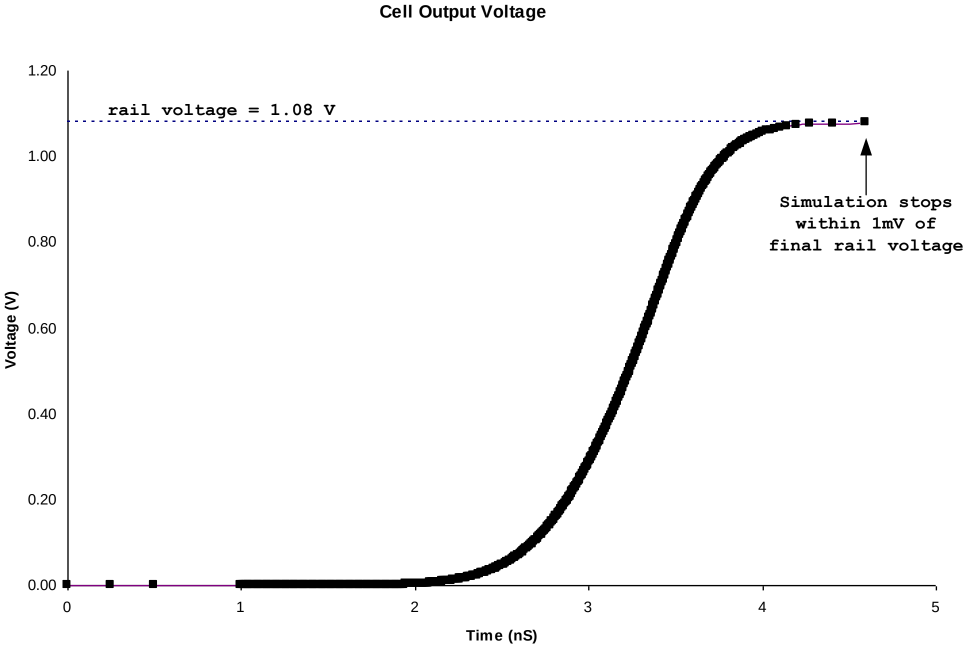
3.3.1 Simulation Autostop and Full Transition

For CCS Power, two approaches are used to determine the simulation duration, depending upon whether there is an output switching event being simulated.

When the output switches in response to an input change, the simulation captures the switching power. It is strongly recommended that a simulation auto-stop value is set when the output is within 1mV of its final value, regardless of the end current. Figure 1 shows simulation results for the output voltage of a cell, which has an auto-stop value within 1mV of the final rail voltage of 1.08V.

When there is no output switching in response to an input change, the simulation captures internal power only. In this situation, the duration of the simulation is recommended to be a factor of 10 times longer than the maximum input transition time.

Figure 1. Simulation Results With the Auto-Stop Feature



3.3.2 Current Vector Variables

The template specifying CCS Power must always have time as the value for `variable_n+1`. You can assign the `input_net_transition` and `total_output_net_capacitance` values to any of the first `n` variables (`variable_1` to `variable_n`), as shown in Example 1. If `variable_n+1` does not have the value time, Library Compiler (version X-2005.09 and later) will issue an error message for this condition, as shown in Example 2.

Example 1:

Following are the examples of the correct CCS Power templates:

```
pg_current_template ( CCS_power_1 ) {
  variable_1 : input_net_transition ;
  variable_2 : total_output_net_capacitance ;
  variable_3 : time ;

pg_current_template ( CCS_power_2 ) {
  variable_1 : total_output_net_capacitance ;
  variable_2 : input_net_transition ;
  variable_3 : time ;
```

Example 2:

Here is an example of the Library Compiler error message when `variable_3` is defined with the `input_net_transition` value:

```
Error: Line 198, The index of 'input_net_transition' can define only
one value. (LBDB-653)
```

3.3.3 Library Voltage

The nominal voltage and the voltage specified in the default operating condition of the library must be identical (see Example 3).

Example 3:

The following example shows the nominal voltage and operating condition must be identical in the library:

```
library (EXAMPLE) {
       nom_voltage : 1.050000;
       operating_condition ( MAX ) {

                process : 1.000000;
                temperature : 125.000000;
            voltage : 1.050000;
        }
    Default_operating_conditions : MAX;
```

If an operating condition does not exist in the library, Library Compiler will issue a warning message, and create an operating condition. This operation condition will be specified as the default operating condition that matches the nominal voltage in the library as shown in Example 4.

Example 4:

Here is an example of the warning message when the default operating condition is missing in Library Compiler:

```
Warning: Line 13, The default_operating_conditions is not defined.
operating_conditions "nom_pvt" is created and set as the
default_operating_conditions. (LBDB-662)
```

3.3.4 Current at the Start Point

Before the start of the event (which is identified by the initial voltage change on the input pin), the cell is in the steady state. The power/ground current in this state corresponds to the initial leakage current. Therefore, the current value at the start of the current waveform should match the leakage current for the initial state. It is not required to capture this data point in the piecewise linear waveform. When not captured, the first data point corresponding to the initial power/ground leakage current at the start time is implied.

3.3.5 Current at the End Point

At the end of the switch, all inputs and outputs stay at the steady state. The PG current flow at this moment is actually leakage current. It is required that the current value at the end point of the current waveform should match the leakage current at the same steady state specified in the leakage current group. The reason for keeping the leakage current table is that the number of states used to characterize the leakage current can be much larger than the number of switch events. In other words, the number of steady states at the end of switch is chosen for characterizing the dynamic currents. Therefore, you should have leakage currents characterized first such that you can use those values later when calibrating the dynamic currents at the end point.

3.3.6 Switching Conditions

The `dynamic_current` waveforms need to be characterized for switching conditions under all legal states. Missing tables for the switching conditions under legal states might result in inaccurate power analysis results unless the power dissipation is negligible.

3.4 Steady State Current

Use the following syntax to describe the power/ground and input current when gates are in steady state (such as subthreshold and gate leakage current).

```
cell ( cell_name ) {
  leakage_current() {
    when : state_1;
    pg_current(pg_pin_1) {
      value : val_1_1;
    }...
    gate_leakage(input_1) {
      one_value : val_one_1_1;
      zero_value : val_zero_1_1;
    }...
  }...
}...
```

Similar to the current polarity specified in section 3.3, the leakage current is also measured toward the cell. For ground pins and input pins pulled low, the current is negative, indicating that current flows out of the cell. In absence of gate leakage, current conservation must hold such that if all PG pins are specified with leakage currents, the sum of all leakage currents should be zero within the same `leakage_current` group.

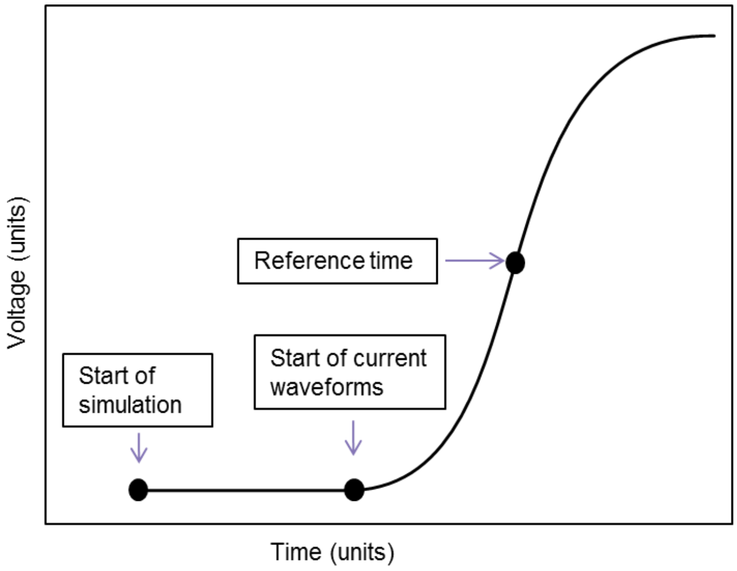
At most one power/ground pin can be omitted, and it will be derived from the other leakage currents.

When gate leakage is also modeled, the situation is more complex. To allow a compact library representation for cells with many inputs, two-level state encoding is allowed where both the currents for the input driven high (`one_value`) and pulled low (`zero_value`) are specified. Unless fine state partitioning is used (such that for all state conditions the value of the input pins is uniquely specified), the set of all currents can no longer sum up to zero. Hence, in presence of gate leakage, leakage currents for power/ground pins must not be omitted.

3.5 The reference_time Attribute

The `reference_time` attribute is the time at which the input voltage characterization waveform crosses the input delay threshold. The start time for a CCS Power characterization is the time at which the voltage transition on the input to the cell starts. This precedes the `reference_time` attribute.

Figure 2. The Definition of the reference_time Attribute



Because `reference_time` is related to physical circuit behavior, Library Compiler will issue an error if a negative `reference_time` is used in the library.

3.6 Intrinsic Parasitic Requirements

Use the following syntax to describe the intrinsic parasitic for gates in the steady state. It consists of two parts: one is intrinsic resistance and the other is intrinsic capacitance.

```
cell ( cell_name ) {
  intrinsic_parasitic() {
    when : state_1;
    intrinsic_resistance(pg_pin_1) {
      related_outputs : output_pin_1;
      value : res_1_1;
    }…
    intrinsic_capacitance(pg_pin_1) {
      value : cap_1_1;
    }…
  }…
}
```

Rail analysis requires state-dependent intrinsic parasitic. The state dependency is similar to that of leakage current. Both intrinsic resistance and intrinsic capacitance are output load independent. Under any circumstance, intrinsic resistance and capacitance values should not be negative or zero.

For any specific state, you should specify intrinsic capacitance for each PG pin. If the capacitance is missing for one PG pin, it is assumed that the capacitance to that PG pin is either negligible or equals to zero.

For any specific state, both leakage current and intrinsic parasitic should be provided in the library.

3.6.1 Intrinsic Resistance

The intrinsic resistance represents a channel connection between a pair of a PG pin and an output pin. A channel resistance is not necessary if it represents an off channel (e.g., if its value is greater than 1MΩ). See Example 5 for defining intrinsic parasitic in an inverter.

Example 5:

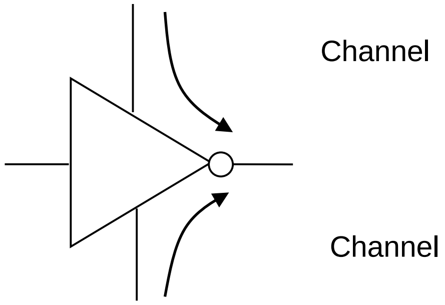
Here is an example of defining intrinsic parasitic of an Inverter where one channel is on and the other is off:

```
cell ( inverter1 ) {
  intrinsic_parasitic ( ) {
    when : !in
    intrinsic_resistance ( VDD ) {
        related_output_pin : out;
        value : 650.0;
    }
    intrinsic_resistance ( VSS ) {  /* not necessary */
        related_output_pin : out;
        value : 100.0e6;
    }
  }
}
```

Each PG pin can have multiple channels where the number of channels is equivalent to the number of outputs. If there are multiple off-channel resistances specified for the same output pin in the same intrinsic parasitic group, Library Compiler will perform sanity checking to make sure that there is no open channel from power to ground through the same output pin.

For the same output, there might also exist a group of channels and the number of channels is equivalent to the number of PG pins. Within the group, some are connected to the power pins while the rest are connected to the ground pins. For any pair of output-to-power-pin channel and output-to-ground-pin channel in the same group, the ratio of the channel resistances should be of different orders of magnitude. The recommended order of magnitude checking is 3, which means one should be at least 1,000 times greater than the other. In other words, under any specific state condition, there should never exist open channels from a power pin to an output pin, and from a ground pin to the same output pin. These two resistances can be of the same order of magnitude only if both channels are off. Only the on channel can be specified in the Liberty file. Library Compiler will issue an error if multiple channels to the same output pin exist within the same group (for the same state condition). See Figure 3 for an example.

Figure 3. Channel Resistances in an Inverter. The Resistance Ratio Is Greater Than 1000.



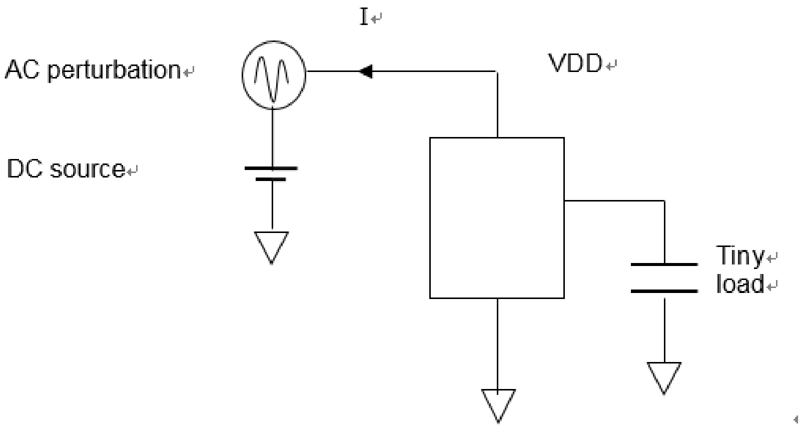
3.7 Voltage-Dependent Intrinsic Parasitic Requirements

Use the following syntax to describe the voltage-dependent intrinsic parasitic required for the wake-up process of standard cells in a switched power network.

```
lu_table_template (template_name) {
variable_1 : pg_voltage | pg_voltage_difference ;
index_1 ( "float, ... float" );
}
cell (cell_name) {
...
  intrinsic_parasitic() {
    when : "boolean expression";
    intrinsic_resistance (pg_pin_name) {
      related_output : output_pin_name;
      value : float;
      reference_pg_pin : pg_pin_name;
      lut_values (template_name) {
        index_1 ("float, ... float");
        values ("float, ... float");
      }
    }
    intrinsic_capacitance(pg_pin_name) {
       value : float;
      reference_pg_pin : pg_pin_name;
      lut_values (template_name) {
        index_1 ("float, ... float");
        values ("float, ... float");
      }
    }
  }
}
```

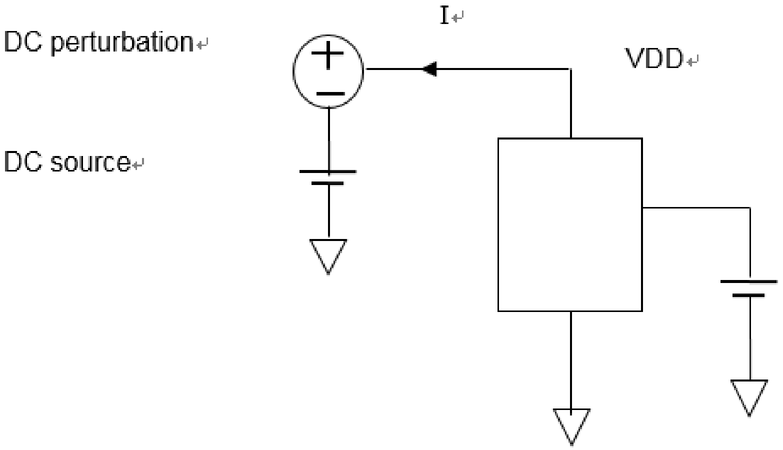
To generate voltage-dependent intrinsic capacitance, swap the voltage value on the DC voltage source where the AC perturbation is connected, and measure each capacitance correspondingly. To choose a list of swapped values (e.g., 10 values) for the DC voltage source, divide the full voltage swing into 10 intervals and swap the voltage from zero to the full swing accordingly using those 10 different values.

Figure 4. An Example of Generating Voltage-Dependent Intrinsic Capacitances



Generating voltage-dependent resistance is similar. You need to swap the voltage value on the DC voltage source where the perturbation DC voltage source is connected. At the same time, the voltage value on the output load should be simulated (or calculated in DC analysis), based on the given DC voltage value at the PG pin. In other words, for each swapped value on the PG pin DC source, there is also a certain DC voltage value on the output pin.

Figure 5. An Example of Generating Voltage-Dependent Resistances



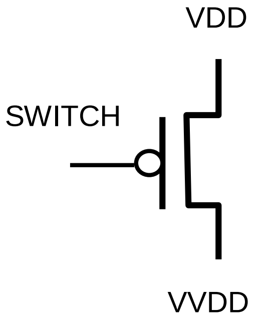
3.8 MTCMOS Switch Cell Requirements

Use the following syntax to describe DC IV curves required for MTCMOS switch cell characterization.

```
cell ( sw_cell) {
  dc_current ( <template_name> ) {
            related_switch_pin : CTL;   /* switch pin is CTL */
             related_pg_pin : VDD;            /* main VDD */
             related_internal_pg_pin : VVDD; /* virtual VDD */
             values("<float>, …");
           }
     }
```

DC IV curve is used to capture the current from actual PG pin (i.e., `related_pg_pin`) down to the virtual PG pin (i.e., the virtual PG pin) when the switch cell is in a transient state from off to on, or from on to off. A simple MTCMOS cell consists of a PMOS or NMOS transistor, as illustrated in the following picture.

Figure 6. A Simple MTCMOS Cell

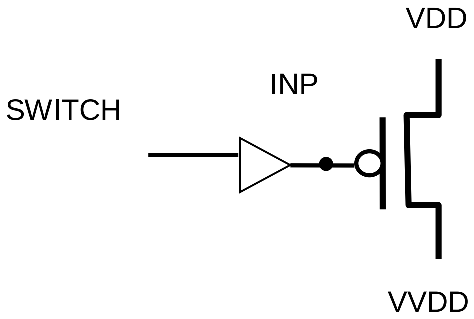


In Figure 6, the `related_pg_pin` is VDD; the `related_internal_pg_pin` is VVDD; and the `related_switch_pin` is SWITCH. The current direction is defined as follows:

- The `dc_current` is positive if the current is flowing from the `related_pg_pin` to the `related_internal_pg_pin`. An example of this kind of `dc_current` can be seen on HEADER cells.
- The `dc_current` is negative if the current is flowing from the `related_internal_pg_pin` to the `related_pg_pin`. An example of this kind of `dc_current` can be seen on FOOTER cells.

For MTCMOS cells with an internal logic in front of the PMOS or NMOS switch transistors, timing information has to be captured for the internal logic part. An example of such MTCMOS cells is as illustrated in the following picture.

Figure 7. An MTCMOS Cell with Internal Logic in Front of the PMOS Switch Transistor



In this example, CCS timing is captured for timing arc from the SWITCH pin to the internal pin INP. At the same time, DC IV curve is captured for a triplet of pins, (INP, VDD and VVDD) instead of the original pins (SWITCH, VDD, and VVDD). INP is serving as an actual switching pin for the PMOS transistor. The DC voltage source is connected to this pin, not SWITCH pin.

An example of the above MTCMOS cell is shown as follows:

```
cell(<cell_name>) {
  switch_cell_type : coarse_grain;

  pg_pin ( VDD ) {
    pg_type : primary_power;
  }
  pg_pin ( VVDD ) {
    pg_type : internal_power;
  }

  pin ( SWITCH ) {
   direction : input;
  }

  pin ( INP ) {
   direction : internal;
  …
  }

  dc_current ( iv1 ) {
    related_switch_pin : INP;       /* switch pin is INP */
    related_pg_pin : VDD;           /* main VDD */
    related_internal_pg_pin : VVDD;     /* virtual VDD */
    values("0.50, 0.40, 0.0",\
       "0.45, 0.35, 0.0",
         "0.01, 0.01, 0.0");        /* from IV characterization */
  }
}
```

3.9 The Accuracy of Library

The library should have sufficient indices such that the interpolation error compared to the reference circuit simulator is within acceptable tolerance.

3.10 The Range of Indices

The range of indices for input transition time and output capacitance must expand across the entire range of allowable values when the cell is used in a design. Make sure that the minimum input transition time and output capacitances extend to the minimum values allowed in designs. If the library is used for signal integrity analysis, consider extending the characterization range. Do not use cells outside the characterization range, as this can cause large differences in delay and output transition time, compared to the reference circuit simulator. Avoid extrapolations outside the table.

The range of indices for switch cell IV curves depends on the switch cell's operating voltage. If the switch cell is operating from 0.0V to VDD (e.g., 1.0V), the voltage range for both switch pin and virtual PG pin is recommended to be from 0.0V to 1.0V, too. It is also recommended that a logarithmic scale is used for choosing indices from 0.0V to 1.0V. The recommended number of indices is 10.

3.11 The Number of Significant Digits

- CCS Power Current Waveforms

  The values of currents in the tables should have at least 3 significant digits. Adding trailing zeros to values does not meet the criteria for significant digits. This guideline might need to be more stringent, depending on process and closeness of indices in the table.

  Current vector data can be represented with scientific notation in the `index_n+1` variable and the `values` attributes. While not essential, it will preserve digits in the current waveform with a small load. The recommendation is to use at least 4 significant digits.

- Intrinsic Parasitic

  The intrinsic resistance and capacitance in the library should have at least 2 significant digits. If the off channel resistance (that is, greater than 1 MΩ) is present in the library, the required significant digit can be smaller.

- Switch Cell IV Curves

  The values in the DC IV curves should have at least 3 significant digits.

4. Library Characterization for CCS Power

During library characterization and generation, there are many factors that you must consider to ensure accurate timing information in the library. This section discusses several key items, as described in the following topics:

- 4.1 Circuit Simulator Settings

4.1 Circuit Simulator Settings

During the characterization process, a reference or “golden” circuit simulator, such as HSPICE, is used to simulate the cell. It is important to run simulation with appropriate settings to achieve the best accuracy.

4.1.1 Control of Simulation Time Step

For HSPICE, you should use adaptive or dynamic time stepping. For more details, please see reference [2]. This method ensures enough points are captured when voltages in the circuit are changing quickly, and reduces the number of points when voltages in the circuit change slowly. You can achieve this by setting a large simulation time step in the .tran statement and the `absvar` and `relvar` variables in the .option command to control the simulation time steps. Example 6 shows the simulation commands that should be included in an HSPICE deck to enable dynamic time stepping. Segmentation tolerance is discussed in Section 4.5.

Example 6: Using Adaptive Time Step in HSPICE

```
.param segmentation_tolerance=0.010
.param vswing=’vdd-vss’
.param absolute=’segmentation_tolerance/vswing’
.option accurate relvar=segmentation_tolerance absvar=absolute
.tran 1.0n 1.0u
```

4.1.2 Simulation Speed

If you use the .tran simulation setting in Example 6, include autostop in the SPICE deck to minimize simulation runtime. Otherwise a total time of 1 microsecond will be simulated. The autostop feature terminates the simulation after all .meas statements have been completed. Similar to characterizing for CCS Power, you should add the .meas statements within 1mV of the final rail voltage on the input and output of the cell to the SPICE deck. Example 7 shows the autostop and .meas statements for an inverter with a rising output. If the cell has previously been characterized for NLDM, autostop might already be set at the second transition time threshold and should be changed when characterizing for CCS Power.

Example 7: Using autostop to Decrease Simulation Time

```
.option autostop
.meas tran tstopi  when v( input)=’vss+0.001’    fall=1
.meas tran tstopo  when v( output)=’vdd-0.001’   rise=1
```

4.1.3 Simulation Algorithm

Simulator transient analysis has several methods for numerical integration. The trapezoidal algorithm should be used for library characterization. It provides the highest accuracy and fastest simulation time. The gear method should be avoided, because it is targeted for inductive or lossy circuits and is not accurate. The trapezoidal method is the default numerical integration algorithm in HSPICE.

4.2 Input Characterization Waveforms

In gate-level static timing, the input characterization waveform used to generate the library plays a significant role in the results obtained. To characterize the library, you should use a waveform that emulates on-chip voltage curves. It is recommended that you use the same Synopsys pre-deriver waveform in CCS timing characterization as the input characterization waveform. See CCS timing characterization guidelines in reference[1].

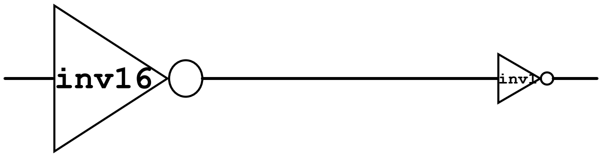
4.3 Operating Range of Cells

You should consider the operating range of cells to determine the minimum and maximum values of indices in the table. The methods described here show you how to determine the operating range of a cell and library developers can use different methods if needed. No matter which method you choose to use, the chosen method always applies to NLDM, CCS Timing, and CCS Power.

Typically, the operating range is constrained by the electrical rules for the library and the process. The following guideline describes how you can determine the operating range:

- The minimum transition time value can be determined by the output transition time of the inverter with the maximum size (small Rd) driving the minimum size inverter (large Rd) with no wire capacitance (see Figure 8).

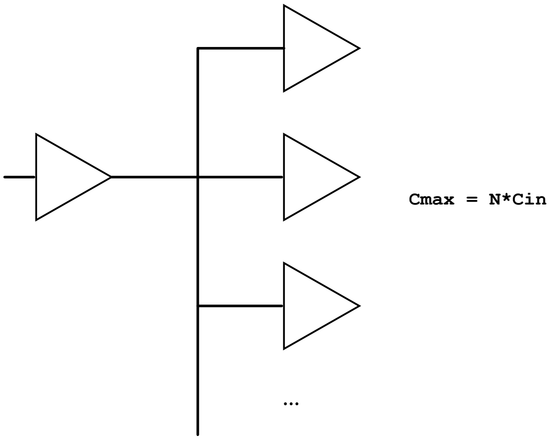
Figure 8. Determining Library Minimum Transition Time



- The minimum output capacitance can be determined by the load of the minimum size inverter. This typically entails evaluating the inverter with the smallest transistor size (width and length), and therefore the smallest value of gate capacitance at the input port.

- The maximum transition time and maximum output capacitance are interdependent. The maximum transition time is dependent on process technologies, design rules, and design applications. A cell's maximum capacitance can be determined by its maximum fan-out load without exceeding the maximum transition time (see Figure 9).

Figure 9. Cell Driving Its Maximum Load, Cmax



Important note:

You should set the `max_transition` and `max_capacitance` attributes in the library to make sure that cells are not used outside the characterization range.

4.4 Selecting Characterization Points

The operating range sets the minimum and maximum indices in the table. Once you have determined the operating range, the next step is to select the intermediate characterization points. The following decisions must be made:

- How many indices are needed in the table?
- What are the values of the indices in the table?

As a general rule-of-thumb, start with a 7-by-7 table as the minimum table size. The decision on intermediate indices should be made to minimize the interpolation errors. Library characterization experts have developed different techniques to determine the indices. In general, selecting indices that are not equally spaced can minimize interpolation errors. Typically, in regions where the input transition time is small (fast) and the output load is small (light), delay and output transition time tends to increase non-linearly with the increasing input transition time and output load. Thus, in these regions, more characterization points are needed to minimize interpolation errors and to achieve better correlation with reference circuit simulators. Sections 4.4.1 and 4.4.2 discuss the methods for determining indices.

4.4.1 Geometric Spacing and Over-Sampling

Given the minimum and maximum operating range, you use a number generator program to generate the indices using geometric spacing. This approach ensures that in regions with small values of input transition time and output capacitance values, the indices are placed closer together in order to minimize the interpolation errors. Using this approach also over-samples by generating tables from 5-by-5 indices to 21-by-31 indices.

Data for each table (delay and transition time) are plotted and evaluated to determine non-linear regions in the table. Based on this, a smaller number of indices are selected, while following the requirements to minimize the interpolation errors.

For more information on using a number generation problem to generate the indices, see reference [4].

4.4.2 Plotting of Interpolation Errors

Using the method of plotting the interpolation errors also generates a set of indices that are not equally spaced for minimizing the interpolation errors. However, at the start of the cell characterization, indices are equally spaced. For each quadrant of the table, a matrix of 10-by-10 equally spaced points are simulated using the reference simulator. The interpolation errors are measured and plotted. Based on these plots, you might observe that in regions where input transition time and/or output capacitances are small, the interpolation errors tend to be larger, compared to the other regions of the table. You can apply automation to re-position the indices in the table while reducing interpolation errors.

For more information on plotting the interpolation errors to generate indices, see reference [5].

4.5 Choosing Indices for Multiple-Output Cells

For cells with multiple outputs, it is not feasible to enumerate all possible variations on the capacitance for each output pin and to carry all combinations of SPICE simulations. The current waveform table will thus become an (m+1) dimensional dense table, where m is the number of outputs and “1” comes from the input slew index. (The timing index is not counted.) It is recommended that you change the capacitance for each output one by one, and use the typical capacitance values for the rest of the outputs. In other words, the current waveform table will still be two-dimensional and sparse. The typical capacitance values can vary for different outputs. However, the capacitance value chosen for an output should be the same when you change the capacitance for the other outputs. This is demonstrated in Example 8.

Example 8: The Current Waveform Table for a Multiple-Output Cell

```
pg_current_template ( CCS_power_1 ) {
  variable_1 : input_net_transition ;
  variable_2 : total_output_net_capacitance ;
   variable_3 : time ;
          }
       …
Cell ( FFHQ )
```

```
dynamic_current() {
   when: “D”
   related_inputs : “CP” ;
    related_outputs: “Q QN” ;
   input_switching_condition(rise);
       output_switching_condition(rise, fall);
       typical_capacitances(10.0, 10.0);
    pg_current ( pg_1 ) {
     vector ( CCS_power_1 ) {
           reference_time : 0.43;
       index_output : Q;
       index_1 ( 0.1 );
       index_2 ( 1.0 ); /* Q */
       index_3 ( “0.000, 0.0873, 0.135, 0.764”)
           values ( “0.002, 0.009, 0.134, 0.546”)
         }…
     vector ( CCS_power_1 ) {
           reference_time : 0.43;
       index_output : QN;
       index_1 ( 0.1 );
       index_2 ( 1.0 ); /* QN */
       index_3 ( “0.000, 0.0812, 0.146, 0.813”)
           values ( “0.003, 0.010, 0.151, 0.423”)
         }…
        }…
   }…
```

In Example 8, there are two output pins: Q and QN. Because a three-dimension table is not preferred, you sweep the input transition time and load the capacitance value one at a time, and then keep the other with a typical (and fixed) value. For example, when the load capacitance of Q is swept, the load of QN is fixed at 10.0 of some capacitance unit. Similarly, when you sweep capacitance on QN, the load on Q is fixed. You can always choose a typical value for the capacitance load. This is further illustrated in Figure 10. The order of the indices (except the last one “time”) of the table is mapped to the same order of pins in the `related_inputs` and `related_outputs` groups.

Figure 10. Indices Selection of Multiple-Output Cells

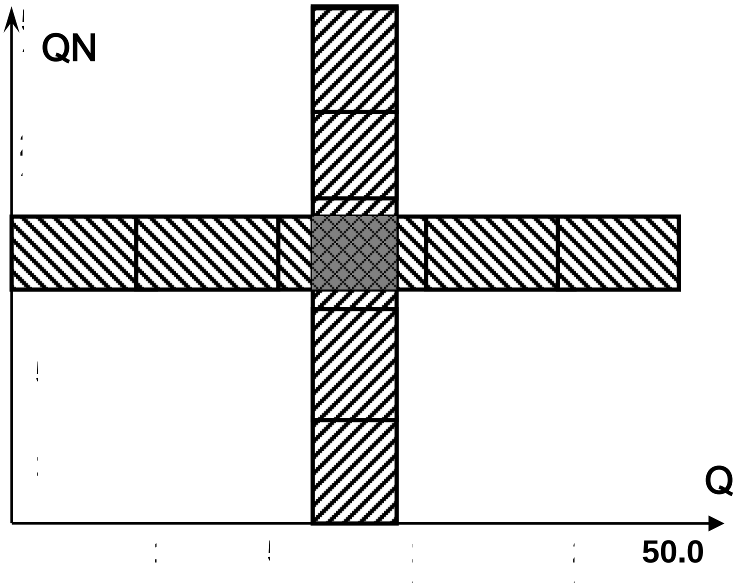


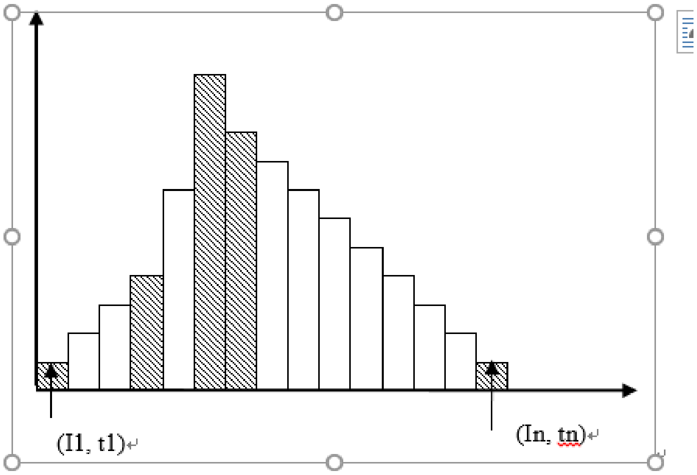
Figure 10 shows that when the load of Q is changed, the load of QN is fixed at 10.0. When you sweep the load on QN, the load of Q does not change. Only one intersection point is required in cases where multiple sweeping variables exist in the design.

4.6 Segmenting PG Current Waveforms

The PG port current waveform generated from the simulation can have hundreds of points. To reduce the library size, only a minimal number of points from the current waveform are saved.

Unlike output currents measured in CCS Timing, the PG currents have bigger variations due to bogus Miller currents and short circuit currents. Therefore, in the segmentation process, the PG currents are converted to a step-wise format first in order to minimize the variation. The original waveform is divided into a number of buckets. For each bucket the averaged current is captured as a list of current values (I, t), that is, a step-wise current waveform (see Figure 11).

Figure 11. Step-Wise PG Current Waveforms



The recommended bucket size is 10 ps. After the step-wise current waveform is obtained, follow the guideline described in sections 4.6.1, 4.6.2, and 4.6.3 to segment piecewise linear points from the waveform.

4.6.1 Dominant and non-dominant Current Waveforms

Current waveforms on PG pins can be classified into two major categories: dominant current waveforms and non-dominant current waveforms. In a single power and ground pin cell, dominant current waveforms represent the currents drawn from the power pin when the related output pin is rising, or the currents discharged to the ground pin when the related output pin is falling.

Non-dominant waveforms are the currents for every other switching condition not covered by the dominant current waveform, which includes:

- Current waveforms on the power pin when the related output pin is falling
- Current waveforms on the ground pin when the related output pin is rising
- Current waveforms on the power or ground pin when the output pin is not switching

When a cell consists of more than one power pin or ground pin, the dominant current waveform is defined as the current drawn from the power pin which provides supply power to the output, or the current discharged to the ground pin.

4.6.2 Segmenting Dominant Current Waveforms

Dominant currents are usually much bigger than non-dominant currents and require more points to capture in the piecewise linear format in order to generate accurate rail analysis results.

A waveform-template-based algorithm is applied for segmenting dominant current waveforms. Dominant current waveforms can be classified into the following waveform types:

- Current waveforms with one single peak

- Current waveforms with multiple peaks
- Current waveforms which consist of both negative currents and positive currents

A current waveform has a peak wherever its time derivative is zero. A single-peak current waveform is defined when:

- There is one point over the whole waveform region where its time derivative is zero.
- There are multiple points where the time derivative is zero, but the current magnitude at one peak is at least 10 times bigger than all the other peaks.

For single-peak current waveforms, it is recommended that you use 6 points to capture the current waveforms.

The algorithm in Figure 12. illustrates how to segment a single-peak current waveform using only 6 points:

a. Capture waveform peak point (1).

b. For characterization using Synopsys pre-driver, follow CCS timing characterization to capture waveform starting point (2) as the point where the input stimulus waveform starts.

c. Traverse down from the peak to the right-hand side of the waveform. Capture 10% and 5% of peak time. Extrapolate from 10% to 5% peak times to get waveform end time. Use this end time as the end point current (3).

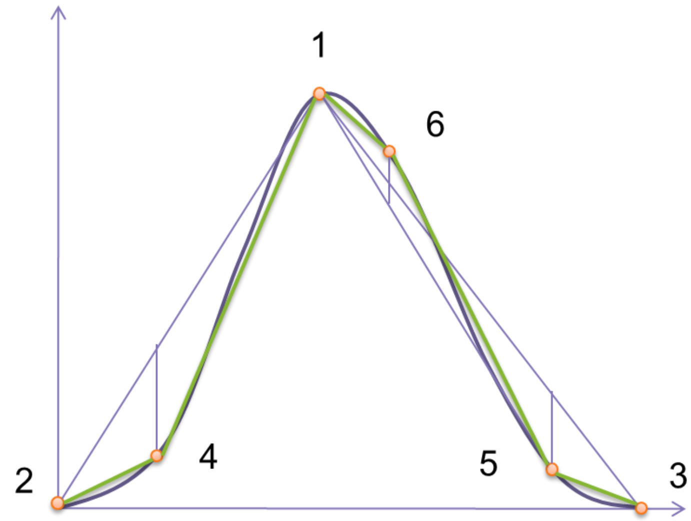
d. Draw a line from the peak point (1) to the starting point (2). Calculate the error at each waveform point as the distance from the waveform point to the line. Pick the waveform point (4) which has the largest error.

e. Draw a line from the peak point (1) to the ending point (3). Calculate the error at each current waveform point as the distance from the waveform point to the line. Pick the waveform point (5) which has the largest error.

f. Redraw lines from point (1) to point (4) and point (4) to point (2). Calculate the error at each waveform point as the distance from the waveform point to the line. Similarly redraw lines from point (1) to point (5) and point (5) to point (3). Calculate the error at each waveform point. Pick the point which has the largest error (which is point 6 on the right-hand size of the waveform).

g. Segmentation ends since all 6 points have been used.

Figure 12. Segmenting a Single Peak Current Waveform Using Only 6 Points



A current waveform has multiple peaks if there are multiple points where its time derivative is zero. A multiple peak current waveform is defined as follows:

- There is more than one point where its time derivative is zero.
- Assume the current magnitude on the biggest peak is Imax. Use the starting point, biggest peak point, and end point to construct a triangular waveform. For all the other peaks, calculate the distance between the peak and the triangular waveform. More than one peak exists only if the calculated distance is more than one tenth of Imax.

For multiple peak current waveforms, the recommended number of points to be used is equal to 5+ number of peaks.

Shown in the Figure 13. the algorithm to segment a double peak current waveform using only 7 points by running the following steps:

a. Capture waveform peak points (1) and (2).

b. Capture waveform starting at point (3).

c. Capture waveform ending at point (4).

d. Draw lines from the peak point (1) to (2), and (2) to (3). Calculate the error at each waveform point as the distance from the waveform point to the lines. Pick the waveform point (5) which has the largest error.

e. Draw a line from the peak point (1) to the ending point (4). Calculate the error at each current waveform point as the distance from the waveform point to the line. Pick the waveform point (6) which has the largest error.

f. Redraw lines from point (1) to (5), (5) to (2), (1) to (6), and (6) to (4). Calculate the error at each waveform point as the distance from the waveform point to the line. Pick point (7) which has the largest error.

g. Segmentation ends since all 7 points have been used.

Current waveform with both positive portion and negative portion is defined as: it has at least two peaks, and one peak is positive, and other peak is negative. The magnitude difference between those two peaks is less than 10X. The recommended number of points to be used to capture this kind of waveform is 5 + number of peaks (both positive and negative).

The algorithm in the below illustrates how to segment a current waveform with both positive and negative peaks:

a. Capture positive and negative peaks (1) and (2).

b. Capture starting point (3).

c. Capture ending point (4).

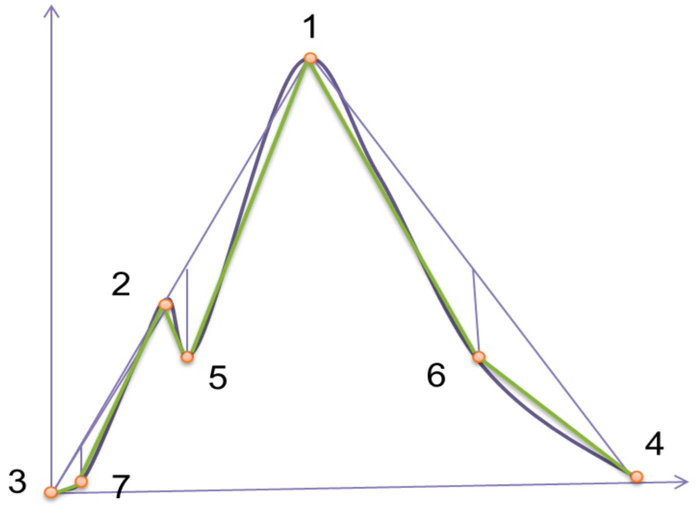
d. Draw lines from (1) to (2), (2) to (3). Calculate the error for each waveform point as the distance from the waveform point to the lines. Pick point (5) which has the largest error.

e. Draw lines from (1) to (4). Calculate the error for each waveform point as the distance from the waveform point to the line. Pick point (6) which has the largest error.

f. Redraw lines from (1) to (5), (5) to (2), and (1) to (6), (6) to (4). Calculate the error for each waveform point. Pick point (7) which has the largest error.

g. Segment ends since all 7 points have been used.

Figure 13. Segmenting a Double Peak Current Waveform Using 7 Points



4.6.3 Segmenting Non-Dominant Current Waveforms

Non-dominant current waveform usually has bigger variation than dominant current waveform. However, its current magnitude is also much smaller than the one in dominant current waveform. It is recommended to use a smaller number of points to capture the current waveform. Without losing too much accuracy, using 3 points (i.e. triangular waveform) is recommended.

The following steps illustrate how to extract the waveform using 3 points:

- Capture waveform starting point (1) as the point where the input stimulus waveform starts.
- Capture the waveform ending point (2) as the time where the waveform is integrated to get the total charge or power as in NLPM power table.
- Capture peak time (3) as the time the largest peak occurs. The peak current value is calculated as

  ```
  total_charge * 2 / ( end_time – start_time )
  ```

  The area in the triangular waveform matches the total charge in the NLPM power table.

4.7 Total Dynamic Charge Calculation

The piecewise linear waveform captured in the library consists of n time/current pairs . The integrated area or total dynamic charge of this waveform is calculated as the sum of the area of all trapezium segments:

$$\mathrm{Q}=\sum_{\mathrm{k}=1}^{\mathrm{n}}\frac{1}{2}\left(I_k+I_{k-1}\right)\left(t_k-t_{k-1}\right)$$

Note that this calculation relies on the position of the implied starting point:

$$\left(t_0, I_0\right)=\left(0, I_{initial\_leakage}\right)$$

4.8 Load Current Requirements

Besides PG currents, the load current is also needed for rail analysis. When the output does not switch, the load current is zero; when it switches, the load current can be obtained from CCS-timing library if it is measured. Therefore, it is required that if PG currents are measured and stored in CCS-Power library under the output switching condition, the load current should be measured and stored in CCS-timing library also.

4.9 Intrinsic Parasitic Characterization

There are multiple ways to characterize intrinsic parasitic, such as AC, DC or transient analysis. It is recommended to use a small AC, DC or transient signal during the characterization, which mimics power or ground variations in rail analysis. Before any measurement is made, it is required that all internal nodes inside subcircuits are properly initialized and stay at stable state.

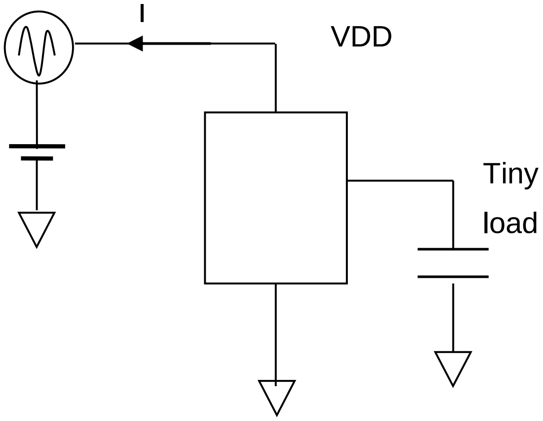
- Intrinsic Capacitance

  To characterize intrinsic parasitic, a small sinusoidal voltage perturbation is applied to each PG pin separately. All outputs are connected with tiny loading capacitance (that is, 1fF) such that the measured intrinsic capacitance is purely PG pin capacitance, and it does not depend on loading. The recommended voltage magnitude is 0.1V or 10% of power supply. By measuring the magnitude and phase of the current response, you can derive the capacitance based on the following formula,

$$C_{eff} = \frac{I_0}{\omega V_0 \cos\phi}$$

Where ω is the sinusoidal waveform frequency; *V_0* is the magnitude of the perturbation voltage waveform; *I_0* is the magnitude of the current response; and ϕ is the phase of the current response. See Figure 14 for a schematic picture of characterization.

Figure 14. Characterization Scheme for Intrinsic Capacitance

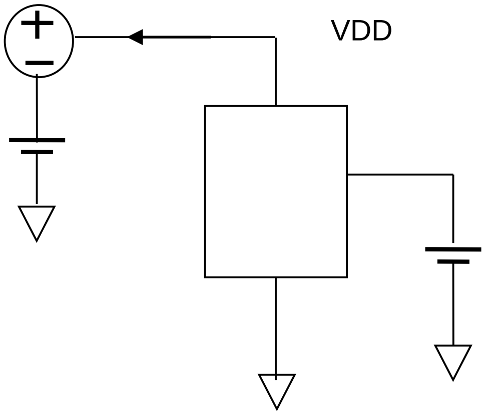


You should use various frequencies during the characterization and take the average among those values under different frequencies. The recommended frequency range is 10MHz to 1GHz.

- Intrinsic Resistance

The characterization structure for intrinsic resistance is similar to that for characterizing intrinsic capacitance, except that a small DC or transient perturbation signal is placed at the PG pin. The output has to be tied to a voltage source at its original DC voltage level (see Figure 15). The recommended voltage magnitude of the DC or transient perturbation signal is 0.1V or 10% of power supply. By measuring the current response, the intrinsic resistance is calculated as the magnitude of the perturbation voltage signal divided by the current magnitude.

Figure 15. Characterization Scheme for Intrinsic Resistance



4.10 Leakage Current Characterization

The leakage current characterization is very similar to the intrinsic parasitic characterization, except that there is no need to have perturbation signal applied to PG pins. The current on the power/ground pins and input pins needs to be measured with open outputs or ideal loads. It is important that no static output current occurs during leakage current characterization, that is, non-ideal loads with leaking gates are not suitable for leakage characterization.

4.11 MTCMOS Switch Cell Characterization

The recommended characterization method is to use DC analysis to capture the IV characteristic for a triplet of three pins: switch pin, main PG pin and virtual PG pin. Using the header cell in Figure 16 as an example, the switch pin and the virtual PG pin should be tied to DC sources whose voltage swing value is specified according to the operating voltage.

Figure 16. A Header Cell Sample for Characterizing IV Curves

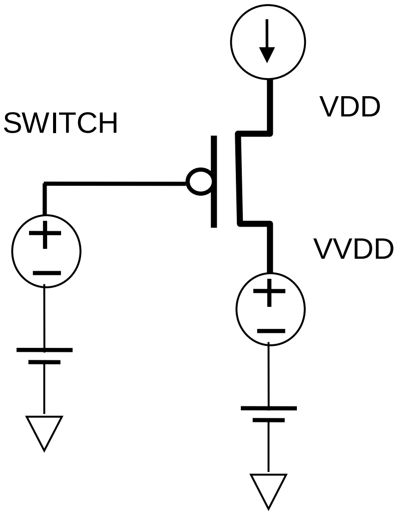


Figure 16 illustrates how IV curves should be characterized using a header cell. Usually, the maximum voltage is the VDD supply value when the block is fully charged. A current meter should be placed at VDD pin (the `related_pg_pin`) to capture the DC current as the voltage swing value changes at switch pin and virtual PG pin.

As stated in section 4.7, when there is internal logic in front of switch transistors, the external switch pin does not turn on the switch transistor directly. It is the internal pin which turns on the switch transistor directly. For this kind of switch cells, timing information has to be captured for the arc from the external switch pin to the internal pin. CCS timing is recommended for best accuracy. Then the IV curves should be captured for the triplet among internal pin, main PG pin and virtual PG pin.

4.12 Other Considerations

- Simultaneous Switching

  Simultaneously switching inputs during library characterization are not supported.

- Merging Current Waveforms

  Merging related current waveforms to produce a single current waveform can result in inaccurate rail analysis results and is not recommended.

- Level Shifters

  A cell can be characterized with the input rails different from the output rails. This is supported for CCS Power.

4.13 Significant Digits in Library Tables

See Section 3.11 for significant digit requirements for libraries.

5. Macro Cell Modeling for CCS Power

This section describes macro-cell modeling for CCS Power, mainly focusing on memory cells. While characterizing macro cells, most of the guidelines for standard cells described previously apply. This section describes what you have to know to model macro cells in CCS Power.

5.1 Simulation Setup

Generally speaking, the size and complexity of macro cells are much larger than those of standard cells. It is almost impossible to simulate a memory design with HSPICE. Fast SPICE simulation tools such as NanoSim and HSIM are usually adopted to verify the timing and power of the designs. Fast simulators take advantage of the regularity and hierarchy of large memory designs.

Including all signal net parasitics results in the most accurate power/ground current waveforms with the penalty of losing the efficiency of fast SPICE simulators. To balance the tradeoff between accuracy and capacity of the simulation, you should at least lump signal net capacitance to the netlist to preserve the total charge that is needed to operate the design.

The representation for standard cells is impractical for cells with many outputs because the power for all possible combinations of toggling outputs must be captured separately. For macro cells such as memory cells, the power contribution is mainly determined by the input events and independent from the actual output toggles. Most of their power consumption comes from the internal operation but not from charging the output loading capacitance. There is no need to sweep different loading capacitances while characterizing the current waveforms for macro cells. You should use the maximum loading capacitance that still maintains timing closure.

5.2 Generating Input Vectors

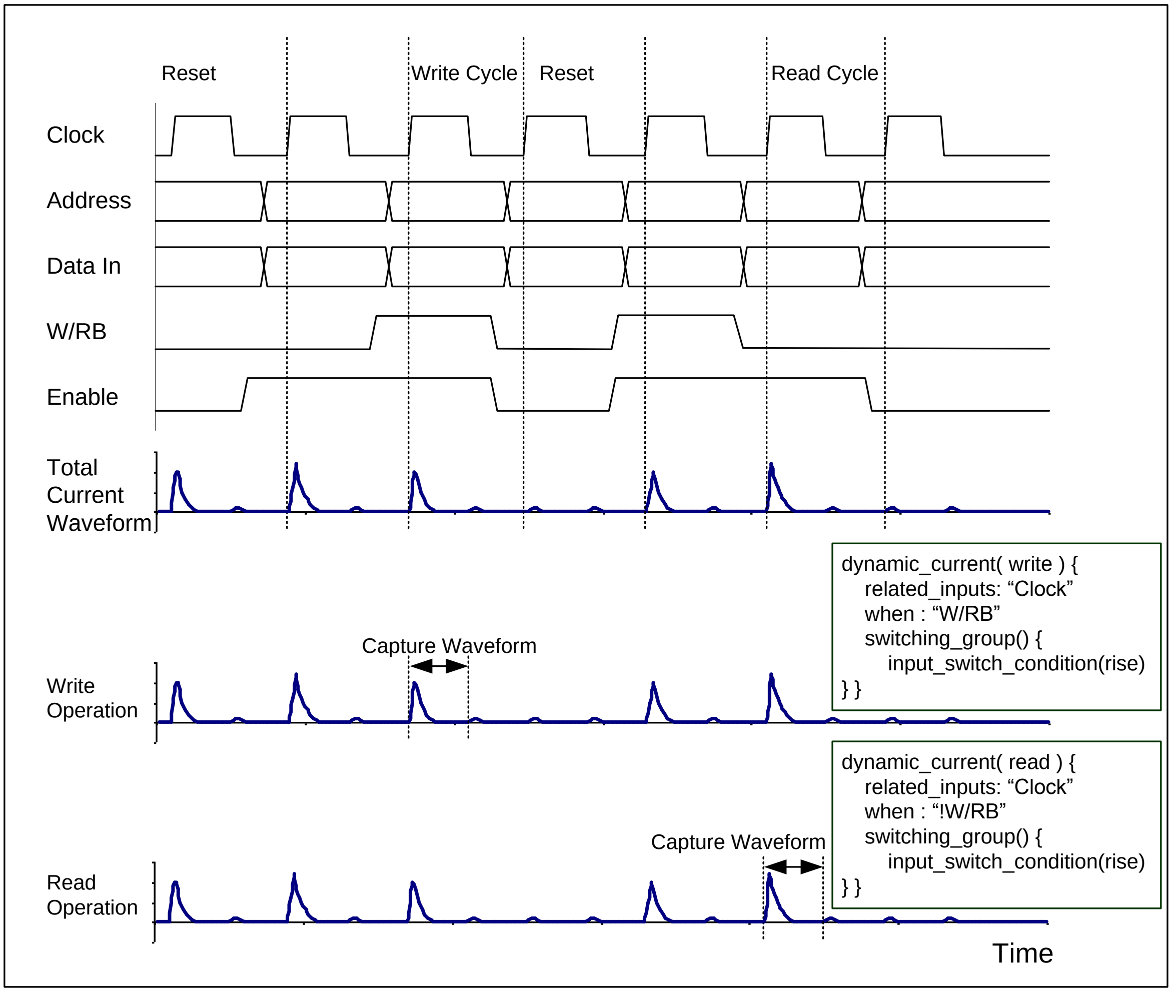
There is no need to separate different operations in different simulation runs. Run circuit simulation with all the possible operations that are needed to be captured in the library. To do this , you must design a sequence of input vectors before running the simulation.

While creating the input vectors for circuit simulation, ensure that there is enough time to initialize the circuit. Most of the fast simulation tools do not calculate the Design Compiler solution at time zero. Instead, they rely on simulating for a long enough time to have the circuit reach a steady state.

For those signals that are neither the trigger (i.e., related input) nor a control signal (i.e., any signal appearing in the when condition) of an operation, you can toggle all of them before triggering the operation to achieve the worst power consumption.

Figure 17 shows how to generate the input vector and capture the current waveform. The length of the waveform should cover the whole period starting with the time the related input pin was toggled and ending just before its next toggle. In this example, that is only half a clock cycle as shown.

Figure 17. Input Vector Generation and Waveform Capturing



```
dynamic_current( write ) {
    related_inputs: "Clock"
    when : "W/RB"
    switching_group() {
        input_switch_condition(rise)
} }
```

```
dynamic_current( read ) {
    related_inputs: "Clock"
    when : "!W/RB"
    switching_group() {
        input_switch_condition(rise)
} }
```

5.3 Characterizing Asynchronized Operation

While buses and bundle representations are used for the `related_inputs` field, such an operation is an asynchronized operation. In practice, individual input signals are usually skewed in terms of switching time. As long as the skew is within a certain timing window, CCS Power can group the toggled inputs and treat this group of events as one operation. Different applications may have their own way of determining how input signals should be grouped as one event.

When generating the input vector to characterize such an operation, you can arbitrarily select a specified number of the related inputs and toggle them. Toggle them simultaneously to achieve the maximum peak current consumption.

5.4 Obtaining Total Capacitance

The `total_capacitance` attribute specifies the total capacitance that can be seen on a power/ground net, including all possible sources of capacitance such as metal capacitance, diffusion capacitance, transistor intrinsic capacitance, and so on. It is state independent, which means no when condition can be specified with this attribute. Currently, CCS Power does not support state-dependent intrinsic capacitance for macro cells. Use estimation, transistor-level parasitic extraction and circuit simulation, or silicon measurement to obtain the total capacitance of a power/ground net.

6. Characterization Loads Between CCS Power Libraries

If the CCS Power library is intended to be used with multiple libraries for voltage or temperature scaling, each PG current group should have identical output loads for characterization. This is for accurate voltage and temperature scaling. There is no requirement that the characterization slews be identical for each group, but however it is recommended to have identical slew values.

7. Checking CCS Power Libraries

In addition to library checking stated in CCS Timing Characterization Guideline[6], the following is a summary of checks that you can use to qualify a CCS Power library for Power Compiler, PrimeTime PX, and PrimeRail. These will be implemented in the Library Compiler `read_lib` command.

CCS Power Library Requirement	Check	Section	LC Message
Current Vector Variables	Check the last variable in the dynamic `current template` is always `time`.	3.3.2	LBDB-653
End Point of Current Waveform	Check that the value of the last point of current waveform is the leakage current at the switching-ending state.	3.3.5	
Number of edges in the output switching condition group	Check that the number of rise/fall edges in the output switching condition group is equal to or smaller than the number of pins `related_outputs`.	5.5	
Both channels are not off in C-intrinsic characterization.	Check that the resistance value are both smaller than 1e+06 for power and ground.	3.6.2	

8. Summary

This paper describs CCS Power library requirements for power and rail analysis with applications such as Power Compiler, PrimeTime PX and PrimeRail. Key considerations and suggested methods for achieving accurate libraries are discussed. Beginning in the Y-2006.06 version of Library Compiler, the `read_lib` command supports CCS Power data in the library.

9. References

[1] "NLDM and CCS Timing Library Characterization Guidelines for PrimeTime® and PrimeTime *SI*", Version 1.1, from Synopsys, Inc.

[2] "CCS Timing Characterization Guideline", Synopsys, Inc.

[3] "HSPICE User Guide: Basic Simulation and Analysis", Version L-2016.03, March 2016, from Synopsys, Inc.

[4] "Using PathMill® to Characterize Library Cells", Hai Vo-Ba, System VLSI Technology Operation, Hewlett Packard, Synopsys Users' Group Conference, 2001.

[5] "Constructing Better Non-Linear Delay Models (NLDM) to Improve Timing Closure", Steven E. Start, Erik N. Comparini, American Microsystems, Inc., Synopsys Users' Group Conference, 2001.

[6] "Full Chip Power Network Analysis with Characterization of GDSII Macro Cells", Version 2.0, from Synopsys, Inc.