Design Service PDK group, Marketing & Sales,
Semiconductor Manufacturing International Corporation

SMIC Process Design Kit (PDK)

Reference Manual

PDK Code	SPDK18EE_183350155_CDBA_CDS
EDA Platform	Cadence
Title	SMIC 0.18um EEPROM 1.8/3.3/5.0/15.5V (2P3M~2P6M) Cadence PDK
PDK Revision	1.6
Release Date	2010-8-10

Document Level: (For Engineering & Quality Document)

☐ Level 1–manual ☑ Level 2–procedure/SPEC/report ☐ Level 3–operation instruction

Security Level:

☐ Security 1 –SMIC confidential ☑ Security 2–SMIC restricted ☐Security 3–SMIC Internal

TABLE OF CONTENTS

1. DISCLAIMER

SMIC hereby provides the quality information to you but makes no claims, promises or guarantees about the accuracy, completeness, or adequacy of the information herein. The information contained herein is provided on an "AS IS" basis without any warranty, and SMIC assumes no obligation to provide support of any kind or otherwise maintain the information. SMIC disclaims any representation that the information does not infringe any intellectual property rights or proprietary rights of any third Parties. SMIC makes no other warranty, whether express, implied or statutory as to any matter whatsoever, including but not limited to the accuracy or sufficiency of any information or the merchantability and fitness for a particular purpose. Neither SMIC nor any of its representatives shall be liable for any cause of action incurred to connect to this service.

2. STATEMENT OF USE AND CONFIDENTIALITY

The following/attached material contains confidential and proprietary information of SMIC. This material is based upon information, which SMIC considers reliable, but SMIC neither represents nor warrants that such information is accurate or complete, and it must not be relied upon as such. This information was prepared for informational purposes and is for the use by SMIC's customer only. SMIC reserves the right to make changes in the information at any time without notice. No part of this information may be reproduced, transmitted, transcribed, stored in a retrieval system, or translated into any human or computer language, in any form or by any means, electronic, mechanical, magnetic, optical, chemical, manual, or otherwise, without the prior written consent of SMIC. Any unauthorized use or disclosure of this material is strictly prohibited and may be unlawful. By accepting this material, the receiving party shall be deemed to have acknowledged, accepted, and agreed to be bound by the foregoing limitations and restrictions. Thank you.

3. OVERVIEW

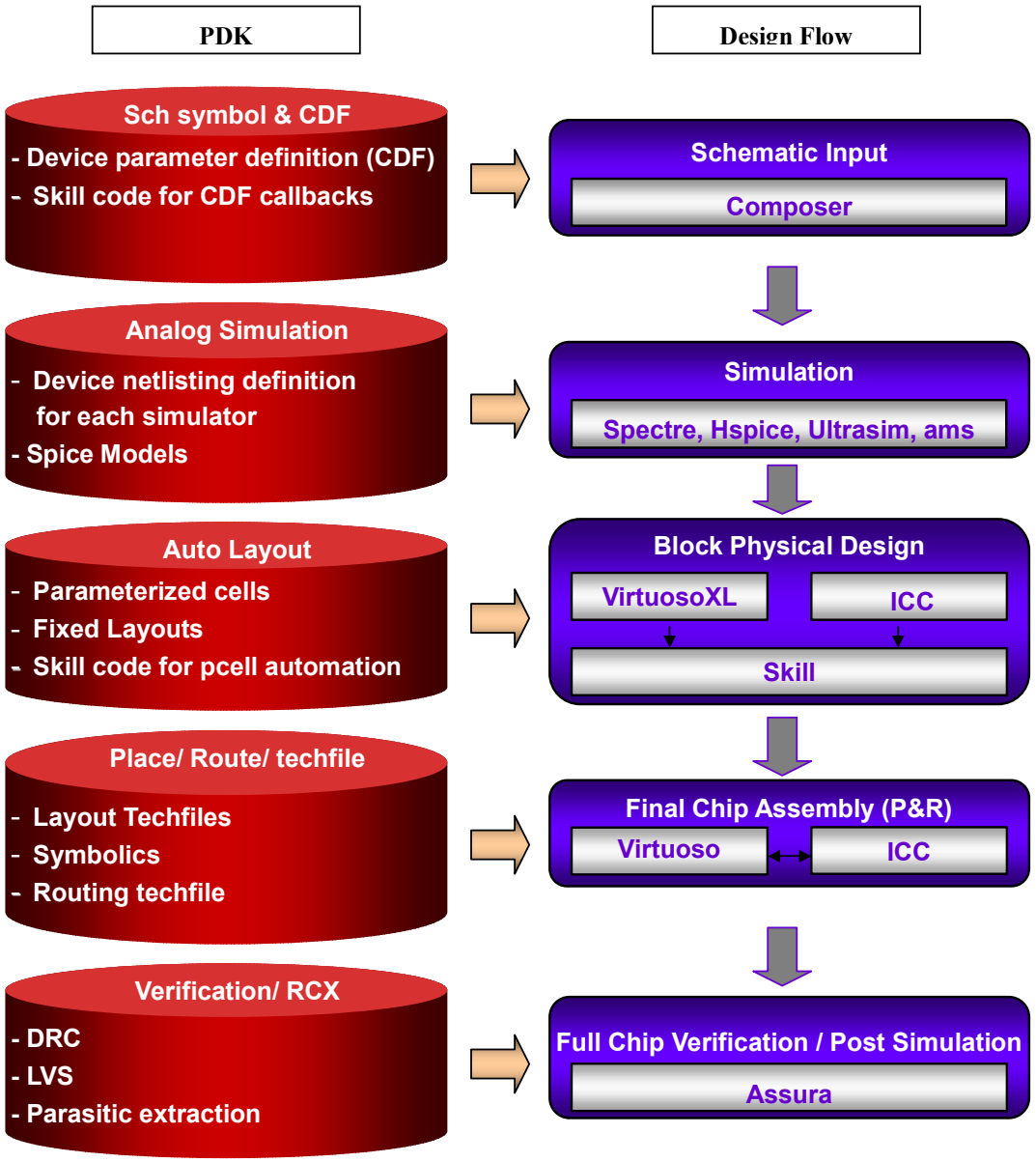
The Purpose of this document is to describe the technical details of the SMIC 90nm Process Design Kit (PDK) provided by Cadence Design Systems, Inc. (Cadence). This PDK requires the UNIX environment variable CDS_Netlisting_Mode to be set to Analog. Training is not provided as part of this PDK.

4. Supported Design Tools

Cadence Design Framework II, version 5.10.41_USR2 Virtuoso Custom Design Platform:
Virtuoso Schematic Editor
Virtuoso Analog Design Environment, including Spectre, AMS, UltraSim, and HspiceD
Virtuoso Aptivia
Technology file support for Preview
VirtuosoLE, VirtuosoXL, Virtuoso CCAR & Chip Editor
Assura DRC, LVS, and RCX
CDL netlisting
Stream In/Out

5. What makes up a PDK?

PDK stands for Process Design Kit. A PDK contains the process technology and needed information to do device-level design in the Cadence DFII environment.



6. Installation of the PDK

The PDK is distributed in compressed tar format. The distribution file name contains the PDK name and release time stamp:

smicxxxxx_cds_xPxM_*YYYYMMDD*_vx.x.tar.gz

The PDK name for 2P2M, 2P3M, 2P4M, 2P5M, and 2P6M processes follow the same naming conversion as above.

To install the PDK, login to the computer as the user who will own and maintain the PDK.

Choose a disk and directory under which the PDK will be installed. This disk should be exported to all client machines and must be mounted consistently across all client machines.

Change working directory to the location where the PDK will be installed:

cd <*pdk_install_parent_directory*>

Extract the PDK from the archive using the following commands:

gtar zxvf smicxxxxx_cds_xPxM_*YYYYMMDD*_vx.x.tar.gz

This will produce a single directory with a name similar to following format:

smicxxxxx_cds_xPxM_*YYYYMMDD*_vx.x

This is a PDK installation directory for a certain SMIC process.

The default permission on the PDK have already been set to allow only the owner to have write, read and execute access. Other users will have only read and execute access.

7. PDK Install Directory Structure/Contents

PDK_ReleaseNote&RevisionHistory.pdf – PDF file containing the PDK release note &revision history.

cds.lib – Cadence library definition file.

display.drf – Cadence Display Resources File

assure_tech.lib - file containing the Cadence Assura initialization path

icc.rules -Virtuoso Custom Router rules file

smicxxxxx/ – SMIC PDK Library

smicxxxxx/techfile.tf – ASCII version of technology file

smicxxxxx/.cdsenv -file containing common design tool environment settings

models/ - directory that will store the PDK models

stream/ - directory containing the cadence stream in and out maps

docs/ - directory containing the PDK documentation

assura_smicxxxxx_tech/ – directory containing the Assura verification files. This directory contains the techRuleSets file that initializes the Assura environment for the user.

8. Creation of a Design Project

A unique directory should be created for each circuit design project. The following command can be executed in UNIX:

mkdir ~/circuit_design
cd ~/circuit_design

All work by the user should be performed in this circuit design directory. The following file should be copied from the PDK install directory to begin the circuit design process. The following command can be used:

cp <*pdk_install_directory*>/display.drf .

Next the user should create a "cds.lib" file. Using any text editor the following entry should be put in the cds.lib file

INCLUDE <*pdk_install_directory*>/cds.lib

Where "pdk_install_directory" is the path to where the smic90nm PDK was installed.
The following UNIX links are optional but may aid the user in entering certain forms with the Cadence environment. In UNIX the following command can be used:

ln –s <*pdk_install_directory*>/models .
ln –s <*pdk_install_directory*>/stream .

Where, again, "*pdk_install_directory*" is the path to where the SMIC PDK was installed.

9. Techfile Methodology

The smic90nm library techfile will be designed as the master techfile. This techfile will contain all required techfile information. There is an ASCII version of this techfile shipped with the PDK. This ASCII version represents the techfile currently compiled into the smic 90nm library.

The attach method should be used for any design library that is created. This allows the design database techfile to be kept in sync with the techfile in the process PDK. To create a new library that uses an attached techfile, use the command File->New->Library from either the CIW or library manager and select the "Attach to an existing techfile" option. Select the smic 90nm library when asked for the name of the Attach to technology library.

10. Customizing Layer Display Properties using the display.drf File

The display.drf can be auto loaded at Cadence start-up time or manually loaded during the Cadence session. For the file to be auto loaded, the display.drf file must be located in the Cadence start-up directory. To manually load the display.drf file (or load a new version), choose Tools->Display Resources->Merge Files... from the CIW and enter the location of the display.drf file that you want to use. If the display.drf file is not autoloaded and you do not manually load it, you will get error messages about missing packets when you try to open a schematic or layout view and you will not be able to see any process specific layers. The display.drf file can be found in the smic 90nm directories of the PDK.

Listed below are the packet, color, lineStyle, and stipplePattern definitions for a metal3 drawing layer. The packet info references predefined color, lineStyle, and stipplePattern definitions. Any of these can be changed to suit an individual user's preferences in the project copy of the display.drf file.

```
drDefinePacket(
;( DisplayName   PacketName   Stipple    LineStyle   fill      outline)
(display         m3           dots       solid       green     green ) )

drDefineColor(
;(DisplayName    ColorName    Red        Green     Blue     Blink )
(display         green        0          204       102       nil   ) )

drDefineLineStyle(
;(DisplayName     LineStyle      Size        Pattern)
 (display         solid          1           (111)   ) )
```

```
drDefineStipple(
;(DisplayName      StippleName       Bitmap)
  ( display         dots        (
                                (0 1 0 0 0 1 0 0 0 1 0 0 0 1 0 0)
                                (0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0)
                                (0 0 0 1 0 0 0 1 0 0 0 1 0 0 0 1)
                                (0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0)
                                (0 1 0 0 0 1 0 0 0 1 0 0 0 1 0 0)
                                (0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0)
                                (0 0 0 1 0 0 0 1 0 0 0 1 0 0 0 1)
                                (0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0)
                                (0 1 0 0 0 1 0 0 0 1 0 0 0 1 0 0)
                                (0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0)
                                (0 0 0 1 0 0 0 1 0 0 0 1 0 0 0 1)
                                (0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0)
                                (0 1 0 0 0 1 0 0 0 1 0 0 0 1 0 0)
                                (0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0)
                                (0 0 0 1 0 0 0 1 0 0 0 1 0 0 0 1)
                                (0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0)
```

11. Schematic Design

The user should follow the guidelines listed below while building schematics using Composer:

Project libraries should list the primitive PDK library as a reference library in the library properties form.
Users can add instances from the PDK library to designs stored in the project libraries.

When performing hierarchical copy of schematic designs care should be taken to preserve the references to the PDK libraries. These references should not be copied locally to the project directories and the references set to the local copy of PDK cells. This would prevent your designs from inheriting any fixes done to the PDK library from an upgrade.

Users should exercise caution when querying an instance and changing the name of the cell and replacing it with a reference to another cell. While like parameters will inherit values, callbacks are not necessarily executed. This would cause dependent parameters to have incorrect values.

Schematics should be designed with schematic driven layout methodology in mind. Partitioning of schematics, hierarchical design, input and output ports, should be done in a clean and consistent fashion.

Usage of pPar and iPar in a schematic design Contactext is discouraged. While this works fine in schematic design, this could lead to problems while performing schematic driven layout.

12. Library Device Setup

12.1 MOSFETs

All mosfets in the PDK library are 4 terminal devices, with the body terminal explicitly connected.

Units:
Length and width are in meters, with areas and perimeters in meters squared and meters, respectively. Design variables are allowed for Length and Width entries.

Calculation:
The area and perimeter parameters for the sources and drains are calculated from the width and the number of fingers used. This calculation assumes that the drain will always have the less capacitance (area) when there is an even number of fingers (odd number of diffusion areas). The Width per finger is calculated by dividing the width by the number of fingers. This parameter is for viewing by the designer.

Simulation:
These mosfets are netlisted as their predefined device names for simulation purposes. SMIC's model definitions are used for these devices.

12.2 Resistors

The resistors in the library consist of three types: *metal, diffused and insulated*. The diffused types include p+, n+, are 2 or 3 terminal resistors with diode back plates. The insulated resistors are those that are isolated from silicon by an insulator (oxide) such as poly resistors. These resistors are 2 or 3 terminal devices. The metal resistors are 2 terminal resistors with the resistive material being the appropriate metal process layer. Serpentine resistor layouts are not allowed.

Units:
The width is specified in meters for schematic simulation. All parameters entered into the resistor form must be integers or floating-point numbers. No design variables are supported due to the calculations that must be performed on the entries.

Calculation:
The length of the resistor segment is calculated from the resistance value, the width, and the numbers of parallel or series segments specified. These calculations are based on the SMIC data provided in the design rule manuals and SPICE models provided by SMIC. See the Foundry Documents in PDK Release Note for document names.

The width and length are snapped to grid, and the resistances are recalculated and updated on the component form based on actual dimensions.

Simulation:
SMIC supplied models are used to model the resistors.

12.3 Bipolar Transistors

All BJT's in the PDK library are 3 terminals.

Units:
Only fixed size devices are allowed. A cyclic is used to enter the desired size.

Calculation:
The area is calculated from the emitter size cyclic.

Simulation:
These BJTs are netlisted as their predefined device names for simulation purposes. SMIC's model definitions are used for these devices.

12.4 Diodes

All diodes in the PDK library are 2 terminal devices.

Units:
Length and width are in meters. Design variables are not allowed for length and Width entries.

Calculation:
The area is calculated from the width and length entered.

Simulation:

These diodes are netlisted as their predefined device names for simulation purposes. SMIC's model definitions are used for these devices.

12.5 Capacitors

Capacitors in SMIC are mimcaps ， pipcaps and varactors. The mimcaps are two terminal metal capacitors ，pipcaps are two terminal poly capacitors and the varactors are two terminal voltage controlled capacitors.

Units:

The length and width of mimcaps and pipcaps are specified in meters for schematic simulation. The varactors are specified by the number of fingers or groups of unit-varactors. All parameters entered into the capacitor form must be integers or floating-point numbers. No design variables are supported due to the calculations that must be performed on the entries.

Calculation:

The area is calculated from width and length of the mimcaps and pipcaps. The varactors allow the user to enter multiplier values as well as the number of fingers. The appropriate model and number of instances is subsequently netlisted in the simulation environment.

Simulation:

SMIC's model definitions are used for these devices.

12.6 Varactors

SMIC provides MOS Varactors and junction diode Varactors.
All varactors in SMIC PDK are 2 terminal devices.

Units:

The length and width are in meters. Number of fingers must be an integer.

Simulation:

SMIC's model definitions are used for these devices.

13. Supported Devices

13.1 MOSFETs

n18e2r	1.8V nominal VT NMOS transistor
n33e2r	3.3V nominal VT NMOS transistor
nz50e2r	5.0V ZMOS transistor
nz18e2r	1.8V native NMOS transistor
n50e2r	5.0V high VT NMOS transistor
p18e2r	1.8V nominal VT PMOS transistor
p33e2r	3.3V nominal VT PMOS transistor
p50e2r	5.0V highl VT PMOS transistor
n155e2r	15.5V high VT NMOS transistor
nz155e2r	15.5V ZMOS transistor
p155e2r	15.5V high VT PMOS transistor

13.2 Resistors

rndif	N+ diffusion resistor with salicide
rpdif	P+ diffusion resistor with salicide
rndifsab	N+ diffusion resistor without salicide
rpdifsab	P+ diffusion resistor without salicide
rnpo	N+ poly resistor with salicide
rnpo_3t	3-terminal N+ poly resistor with salicide
rppo	P+ poly resistor with salicide
rppo_3t	3-terminal P+ poly resistor with salicide
rnposab	N+ poly resistor without salicide
rnposab_3t	3-terminal N+ poly resistor without salicide
rpposab	P+ poly resistor without salicide
rpposab_3t	3-terminal P+ poly resistor without salicide
rhrpo	High resistance poly resistor
rhrpo_3t	3-terminal High resistance poly resistor
rnwaa	Nwell resistor under active area
rnwsti	Nwell resistor under STI
rmx	Metal resistors
rndif_ckt	N+ diffusion resistor with salicide subckt model
rpdif_ckt	P+ diffusion resistor with salicide subckt model

rndifsab_ckt	N+ diffusion resistor without salicide subckt model
rpdifsab_ckt	P+ diffusion resistor without salicide subckt model
rnpo_ckt	N+ poly resistor with salicide subckt model
rnpo_3t_ckt	3-terminal N+ poly resistor with salicide subckt model
rppo_ckt	P+ poly resistor with salicide subckt model
rppo_3t_ckt	3-terminal P+ poly resistor with salicide subckt model
rnposab_ckt	N+ poly resistor without salicide subckt model
rnposab_3t_ckt	3-terminal N+ poly resistor without salicide subckt
rpposab_ckt	P+ poly resistor without salicide subckt model
rpposab_3t_ckt	3-terminal P+ poly resistor without salicide subckt
rhrpo_ckt	High resistance poly resistor subckt model
rhrpo_3t_ckt	3-terminal High resistance poly resistor subckt model
rnwaa_ckt	Nwell resistor under active area subckt model
rnwsti_ckt	Nwell resistor under STI subckt model
rtimsabe2r	Buried As-implanted n-type diffusion resistor
rtimsabe2r_ckt	Buried As-implanted n-type diffusion resistor subckt model

13.3 Bipolar Transistors

pnp18	1.8V PNP bipolar transistor
pnp33	3.3V PNP bipolar transistor

13.4 Diodes

ndio18e2r	1.8V N+/Pwell diode
pdio18e2r	1.8V P+/Nwell diode
ndio33e2r	3.3V N+/Pwell diode
pdio33e2r	3.3V P+/DNwell diode
nwdioe2r	Nwell/P substrate diode
dnwdioe2r	1.8V DNWell/Pwell diode
ndio50e2r	5.0V N+/Pwell diode
ndio155e2r	5.0V N+/Pwell diode
nzdio50e2r	5.0V NZ+/Psub diode
nzdio155e2r	15.5V NZ+/Psub diode
pdio50e2r	5.0V P+/DNwell diode
pdio155e2r	15.5V P+/DNwell diode

13.5 Capacitors

Mime2r	Topmetal–MiM capacitor, with Cspec=1 fF/um^2
Mime2r_ckt	Topmetal–MiM capacitor subckt model
Pipe2r	5.0v PIP capacitor
Pipe2r_ckt	5.0v PIP capacitor subckt model

13.6 Varactors

pvar18e2r_ckt	1.8V N+ poly/Nwell MOS Varactor

14. Views Provided

14.1 MOSFETs

- Four terminals (D, G, S, B)
- Five terminals for RF mosfets (D, G, S, B, T)
- symbol, spectre, ams, UltraSim, hspiceD, auLvs, auCdl, ivpcell, layout(Pcells)

14.2 Resistors

- Two terminals (PLUS, MINUS)
- Three terminals for sub-circuit resistor model (PLUS, MINUS, B)
- symbol, spectre, ams, UltraSim, hspiceD, auLvs, auCdl, ivpcell, layout (Pcells)

14.3 Bipolar Transistors

- Three terminals(C, B, E)
- symbol, spectre, ams, UltraSim, hspiceD, auLvs, auCdl, ivpcell, layout (Fixed layouts)

14.4 Diodes

- Two terminals (PLUS, MINUS)
- symbol, spectre, ams, UltraSim, hspiceD, auLvs, auCdl, ivpcell, layout(Pcells)

14.5 Varactors

- Two terminals (PLUS, MINUS)
- symbol, spectre, ams, UltraSim, hspiceD, auLvs, auCdl, ivpcell, layout(Pcells)

15. CDF parameters

15.1 MOSFETs

- Multiplier – number of Parallel MOS devices
- Model Name – spectre model name (non-editable)
- Length (M) – gate length in meters
- Total Width (M) – gate width in meters (sum of all fingers)
- Finger Width – width of each gate finger/stripe
- Fingers – number of poly gate fingers/stripes used in layout
- Threshold – finger width at which to apply device folding of the layout
- Apply Threshold –button to apply threshold or not

- Gate Connection –allow shorting of multi-finger devices and addition of contact heads to gate end.

- S/D Metal Width – width of metal used to short sources/drains.

- Switch S/D – source is defined as left-most diffusion region and alternating regions to the right. Pins are not automatically permuted and can be switched using this parameter.

- Bodytie Type – None, Detached, or Integrated (butting source).
  – For Detached, user may select Left, Right, Top, and/or Bottom to specify the located of bodyties. Selection of all four creates a guardring.
  – For Detached, the user may specify Tap Extension(in microns) which sets the distance from the bodytie to the device. Maximum distance is 100 microns.
  – For Integrated, the user may select Left or Right for a device with an odd number of fingers(1,3,5,). The user may select Left and Right for an even fingered device. This option does not apply to native and medium Vth devices.

- Drain diffusion area, etc – several simulation parameters are presented. The area and perimeter parameters are calculated and netlisted in accordance with the foundry models.

MOSFET ADD INSTANCE FORM

15.2 Resistors

- Model name (model) – Spectre model name (non-editable)
- Multiplier (m) – number of parallel resistor devices.
- Segments (segments) – Number of resistor segments
- Segment Connection (connection) – Connection between segments (Parallel, Series)

- Calculated Parameter (calculatedParam) – Parameter to be calculated (Resistance, Length)

- Resistance (r) – Entry or calculated value for resistance.
- Segment Width (segW) – Entry value for segment width in meters.
- Segment Length (segL) – Entry or calculated value for segment Length in meters, not including interface resistance.

- Effective Width (effW) – Calculated value based on number of segments, segment width and segment connection (non-editable)

- Effective Length (effL) – Calculated value based on number of segments, segment width, and segment connection (non-editable)

- Subckt resistor calculated Width (cktW) – Calculated value based on segment width.

- Subckt resistor calculated Length (cktL) – Calculated value for sub-circuit resistor length (total resistance including the interface resistance).

- Left Dummy (leftDummy) – Toggle for placement of dummy resistor segment on the left side

- Right Dummy (rightDummy) – Toggle for placement of dummy resistor segment on the right side

- Contact Rows (cntRows) – Entry value for number of contact rows in each segment head

- Contact Columns (cntCols) – Calculated value for number of contact columns in each segment head (non-editable)

RESISTOR ADD INSTANCE FORM

15.3 Bipolar Transistors

- Model name – Spectre model name (non-editable)
- Emitter Size – cyclic of the “width x length” of the emitter (microns).
- Area – calculated emitter area based on emitter width & length in meters squared.
- Multiplier – Number of Parallel Bipolar devices
- Estimated Operating Region – Cyclic of Spectre model operating region

BIPOLAR ADD INSTANCE FORM

15.4 Diodes

- Model name – Spectre model name (non-editable)
- Device Area – Calculated junction area in meters squared (non-editable)
- Width (M) – Entry value for diode width in meters.
- Length (M) – Entry value for diode length in meters.
- Multiplier – Number of Parallel Diode devices
- Periphery of junction – Calculated junction periphery in meters (non-editable).

DIODE ADD INSTANCE FORM

15.5 Capacitors

- Model name – Spectre model name (non-editable)
- Multiplier – Number of Parallel capacitor devices.
- Calculate – Parameter to be calculated (Capacitance, Length, Width, width & length).
- Total Capacitance – Calculated capacitance based on width, length and multiplier.
- Capacitance – Calculated capacitance based on width and length in Farad.
- Width (M) – Entry value for capacitor width in meters.
- Length (M) – Entry value for capacitor length in meters.
- Bottom Connection – Choice for capacitor bottom plate connection.

CAPACITOR ADD INSTANCE FORM

15.6 Varactors

- Model name – Spectre model name (non-editable)
- Multiplier – Number of Parallel varactor devices.
- Fingers – Number of poly gate fingers or AA stripes used in layout.
- Width (M) – Varactor width in meters.
- Length (M) – Varactor length in meters.

VARACTOR ADD INSTANCE FORM

16. Component Label Defaults

16.1 MOSFETs

- component parameters: model, l, w, m
- operating point: ids, vgs, vds, vth, vdsat
- model: vto, kp, gamma
- instance name prefix: NM, PM

16.2 Resistors

- component parameters: r, w, l, segments, connection Type
- operating point: v, i, res
- model:
- instance name prefix: R, X

16.3 Bipolar Transistors

- component parameters: model, area, m
- operating point: betadc, ic, Vce
- model: bf, is, va
- instance name prefix: Q

16.4 Diodes

- component parameters: model, area, m
- operating point: id, vd, reg
- model: is, rs, n
- instance name prefix: D

16.5 Capacitors

- component parameters: c, w, l, m
- operating point: cap
- model:
- instance name prefix: C

16.6 Varactors

- component parameters: w, l, nf, m
- operating point:
- model:
- instance name prefix: C

17. Simulation Models

The following simulators are supported in this PDK:

Spectre
hspiceD
AMS
UltraSim

The following model library setup for Spectre is done automatically when the user first opens the SMIC PDK library or uses a component from that library. The user may disable this feature by editing the libInit.il file found in the SMIC PDK library.

<*pdk_install_directory*>/models/spectre/xxxx_spe.lib	tt
<*pdk_install_directory*>/models/spectre/xxxx_spe.lib	res_tt
<*pdk_install_directory*>/models/hspice/xxxx.lib	TT
<*pdk_install_directory*>/models/hspice/xxxx.lib	RES_TT

Where <*pdk_install_directory*> is the path where the SMIC PDK library is installed, and the second entry is the simulation *corner* that the user wants to simulate.

The user should follow the instructions provided with the device models from SMIC to ensure the proper selection of sections for simulation.

18. Add Wire Utility

Certain devices in the library have bulk pins (i.e. n18, p33, etc.). A utility has been added to the PDK to automatically wire these bulk terminals to user specified signals (i.e. gnd!, vdd!, etc.). This will help reduce schematic clutter while maintaining required circuit hookup. The following picture shows the result of executing the *smic18ee_addWire* routine.

There are no arguments to run the program, the user must type ***smic18ee_addWire()*** in the CIW. If you don't have a schematic cellview open it will give you an error message.

If the user has instances selected when they run addWire, it will prompt them for the label name for those specific instances and will only add the wires to those instances. It does nothing to the schematic hierarchy.

If no instance is selected then the program will wire up all instances that have their bulk nodes unconnected. The user gets prompted for the label name for each type of instance, and they also have the option of running it down through the hierarchy or just at current cellview.

And, finally, if the user does not want to use gnd! or vdd! as the label name, there is an entry box for the user to type in an alternative net name. If another wire name issued, it will be added to the cyclic list of label name choices - but only for that DFII session. Once you exit the dfII session, the cyclic is reset to gnd! and vdd!.

19. Update CDFs Utility

When changes are made to a device inside the PDK, these changes often affect circuit design which has already been created. For example, a sheet resistance Rsh value may change on a resistor which affects the length of the device passed to the simulator and used to generate the layout.

These parameters are not automatically updated in each of the designer's circuit libraries. A function has been written and included as a part of the PDK to complete the update to an existing library such that all modifications made to a PDK since a previous release are reflected in each of the circuit designs inside a library.

The procedure name as typed in the CIW is ***PasFlexUpdateLib***. Executing the command in CIW brings up a GUI for updating design libraries. After specifying a design library, the procedure traverses the design library hierarchy and updates all the appropriate CDF information.

Please note, however, that possible LVS violations may arise as a result of running this routine depending upon what changes have been made to the PDK. For example, a sheet resistance Rsh change as specified earlier could cause a resistor to shrink in length in the schematic thus causing a mismatch as far as LVS is concerned. Please be sure that you verify again each design in simulation, DRC, and LVS to insure that no unintended modifications have been overlooked.

20. Techfile Layers

Techfile layers defined in the PDK (techfile.tf) are done in compliance with technology file supplied by SMIC techfile group. The technology file compiled with this PDK contains all defined layer/levels as defined in the SMIC techfile group supplied techfile. Layer names and visibility /selectability may be altered from this technology file, if necessary.

21. Virtuoso XL

The standard Cadence Virtuoso XL design flow is implemented. This includes basic connectivity of connection layers, wells, and substrate, and symbolic contacts. The M factor is used for device instance multiplier - there is no conflict with the parameter used in cell operation. Names are be displayed on the layout views to aid in schematic-layout instance correlation. Auto-abutment of MOSFET devices is supported. Pin permuting of MOSFET and resistor devices is also supported. The pcell layouts are compiled into the PDK.

The users should follow the guidelines listed below for layout design:

- The VirtuosoXL tool requires a separate license for operation.
- Users obtain maximum leverage from the PDK by doing schematic driven layout in the Virtuoso XL environment. This flow will produce a correct design layout. The Virtuoso Custom Router (IC Craftsman) can be used to finish the interconnection in the layout.
- The VCR rules file for the target process is provided with the PDK (icc.rules).
- Abutment is currently supported only for MOS transistors. Note, abutment will work only on schematic driven layouts.
- Schematic Driven Layout is recommended over Netlist Driven Layout.

NOTE: Skill pcell source code is not included in the PDK kit.

21.1 Symbolic Contacts

M1_AA	Metal-1 to Active Area Contact
M1_GT	Metal-1 to Poly Contact
M1_NW	Metal-1 to Nwell Contact
M1_SN	Metal-1 to N+ Active Area Contact
M1_SP	Metal-1 to P+ Active Area Contact
M1_SUB	Metal-1 to P+ Substrate Contact
M2_M1	Metal-2 to Metal-1 Via
M3_M2	Metal-3 to Metal-2 Via
M4_M3	Metal-4 to Metal-3 Via
M5_M4	Metal-5 to Metal-4 Via
M6_M5	Metal-6 to Metal-5 Via

CREATE CONTACT FORM

22. Known Problems & Solutions

Problem:

Solution: