Reference Manual

For

Generic 90nm Salicide 1.2V/2.5V 1P 9M

Process Design Kit (PDK)

Revision 4.4

Date: 10/17/08

Table of Contents

1 Overview

The purpose of this Reference Manual is to describe the technical details of the 90nm Generic Process Design Kit (“GPDK090”) provided by Cadence Design Systems, Inc. (“Cadence”).

Software Environment

The GPDK090 has been designed for use within a Cadence software environment that consists of the following tools –

GPDK090 Cadence IC5141 Database (CDB)

Software Release Stream	Key Products
IC5141	Cadence Virtuoso Design Environment, Analog Design and Simulation, Physical Design
ICC11241	VCAR
IUS81	AMS Designer, AMS/Ultra
MMSIM70	Spectre, Ultrasim
ASSURA32	DRC, LVS, RCX
NEOCKT34	NeoCircuit
NEOCELL34	Neocell
SOC71	SOC Encounter

GPDK090 Cadence IC61 Database (OA22)

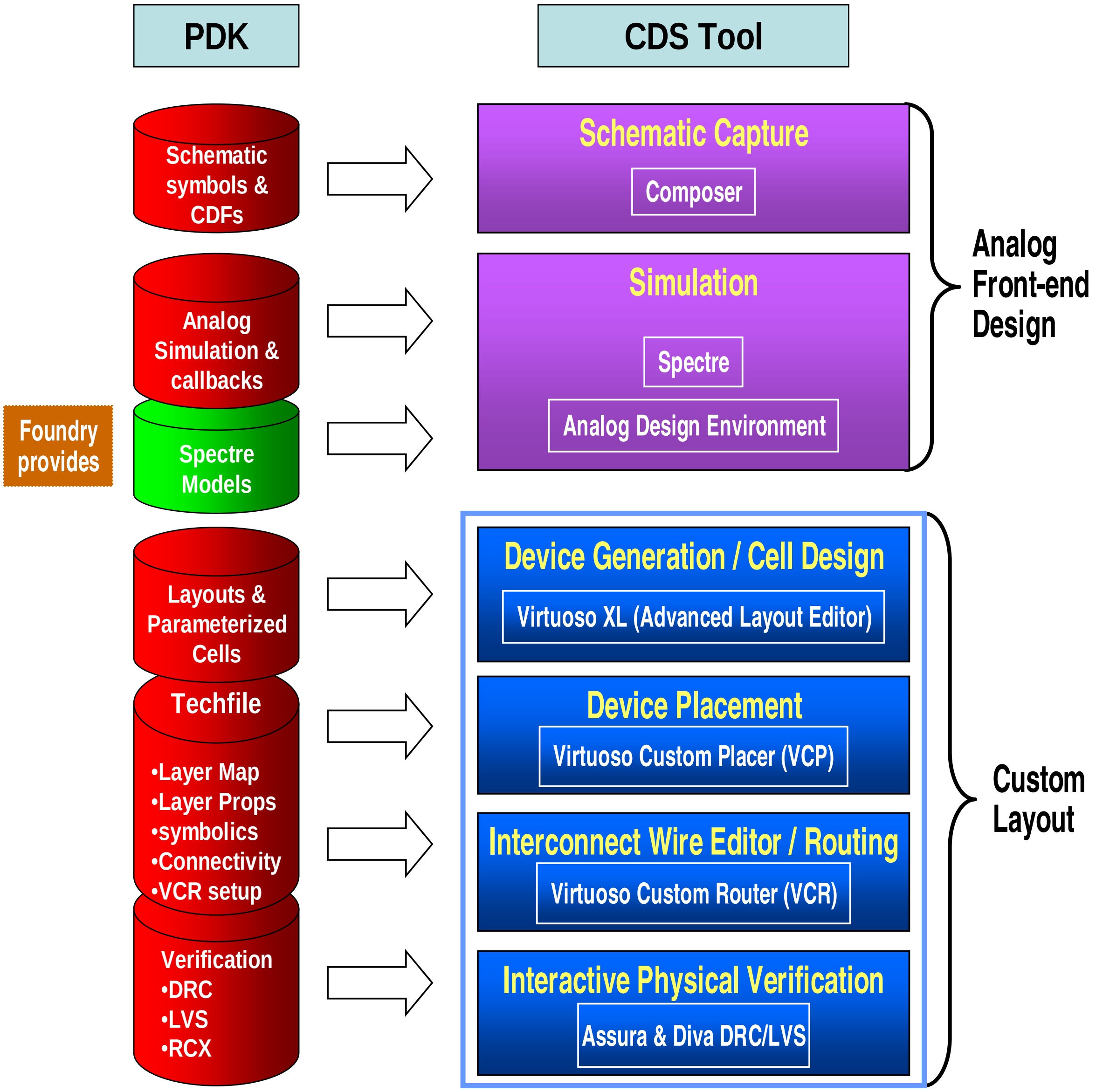
Software Release Stream	Key Products
IC613	Cadence Virtuoso Design Environment, Analog Design and Simulation, Physical Design
FINALE72	Cadence Precision Router
IUS81	AMS Designer, AMS/Ultra
MMSIM70	Spectre, Ultrasim
ASSURA32	DRC, LVS
EXT71	QRC Extraction (L, XL, GXL)
ANLS71	VoltageStorm
SOC71	SOC Encounter

2 Documents

	Documents Used	Rev
Design Rule Document	gpdk090_DRM.pdf	4.4

3 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.



PDK
CDS Tool
Schematic symbols & CDFs
Schematic Capture
Composer
Analog Simulation & callbacks
Simulation
Spectre
Foundry provides
Spectre Models
Analog Design Environment
Analog Front-end Design
Layouts & Parameterized Cells
Device Generation / Cell Design
Virtuoso XL (Advanced Layout Editor)
Techfile
•Layer Map
•Layer Props
•symbolics
•Connectivity
•VCR setup
Device Placement
Virtuoso Custom Placer (VCP)
Interconnect Wire Editor / Routing
Virtuoso Custom Router (VCR)
Verification
•DRC
•LVS
•RCX
Interactive Physical Verification
Assura & Diva DRC/LVS
Custom Layout

4 Installation of the PDK

The user who will own and maintain the PDK should logon to the computer.

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.

Connect to the directory where the PDK will be installed:

```
cd <pdk_install_directory>
```

Extract the PDK from the archive using the following commands:

```
zcat <path_to_pdk_tar_file>/gpdk090_<version>.tar.gz | tar xf -
```

The default permissions 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.

This PDK requires the following UNIX environmental variables:

“CDS_Netlisting_Mode” to be set to “Analog”
“CDSHOME” to be set to the Cadence DFII installation path
“useAltergroup” to be set to “nil” in the .cdsenv for corner analysis to work properly with Verilog-A models as follows:

```
spectre.envOpts useAltergroup  boolean nil
```

5 PDK Install Directory Structure/Contents

Within the <**pdk_install_directory>** directory there are several directories to organize the information associated with the PDK.

assura - Directory containing the Physical Verification Rule Decks for Assura

assura_tech.lib - File containing the Cadence Assura PV initialization path

cds.lib.cdb - File containing the Cadence library definition file. This file

has the CDB version of the PDK library defined

cds.lib.oa22 - File containing the Cadence library definition file. This file

has the OA2.2 version of the PDK library defined

cds.lib – Symbolic link to the CDB version of the PDK

dfIItechFiles - Directory containing the ASCII version of the CDB and the

OA2.2 DFII techfiles

diva - Directory containing the Physical Verification Rule Decks for Diva

docs - Directory containing the Cadence PDK documentation and the Process

design rule manual

fireIce – Directory containing the technology file and layer maps for Fire & Ice

lef – Directory containing the technology LEF file.

libs.cdb – The CDB version of the PDK library

libs.oa22 – The OpenAccess 2.2 version of the PDK library

pvs - Directory containing the Physical Verification Rule Decks for PVS

models - Directory containing the device spectre models

neocell - Directory containing the Neocell technology files

neocircuit - Directory containing the Neocircuit technology file sna – Directory containing the technology files for Seismic SNA

soce – Directory containing the Capacitance tables for SOC Encounter

stream - Directory containing the GDSII stream layer map file

vavo - Directory containing the Virtuoso Analog Voltagestorm data file

vcr – Directory containing the Virtuoso Custom Router data file

vlm – Directory containing the Virtuoso Layout Migrate data file

6 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>/dfIItechFiles/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 GPDK090 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 GPDK090 PDK was installed.

7 Technology File Methodology

The GPDK090 Library techfile will be designated as the **master** techfile. This techfile will contain all required techfile information. An ASCII version of this techfile is shipped with the PDK. This ASCII version represents the techfile currently compiled into the gpdk090 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 gpdk090 library when asked for the name of the *Attach To Technology Library*.

Note: This PDK is using 2000uu/dbu for all layout views.

8 Customizing Layer Display Properties

The display.drf file is automatically loaded by the libInit.il file whenever the gpdk090 library is opened.

To auto-loaded your own display.drf file at Cadence start-up time put the display.drf file 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 auto-loaded 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.

A display.drf file for the GPDK090 can be found in the PDK install directory under techFiles directory.

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           (1 1 1) )
)


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)
)
```

9 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 similar 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.

10 Library Device Setup

Resistors

The resistors in the library consist of three types; *diffused*, *insulated,* and *metal*. The diffused types include p+ and n+ and come in three-terminal varieties. The insulated resistors are those that are isolated from silicon by an insulator (oxide) such as poly resistors. These resistors are two-terminal devices. The metal resistors are those resistors that are used as interconnect and feed-through; they are also 2-terminal devices. Serpentine resistor layouts are not allowed.

Units:

The length and width are specified in meters for schematic simulation. Design variables are supported for both the length and width parameters.

Calculation:

The user has two choices in determining how the final resistor configuration is calculated. The user may request the calculation of either the resistor length or the resistor value. In both cases, the calculated values are determined based upon a combination of the length, width, resistance value, number of resistor segments (series or parallel), and contact resistance.

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

Simulation:

Subcircuit definitions are used to model the resistors

Mosfets

All mosfets in the PDK library are 4 terminals, 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 specifying parameter values on mosfet devices.

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 are an even number of fingers (odd number of diffusion areas). The finger width is calculated by dividing the total width by the number of fingers. Depending upon which value is entered into the form by the user, either the total width or the finger width will be calculated using the aforementioned calculation.

Simulation:

These mosfets are netlisted as their predefined device names for simulation purposes.

Bipolar Transistors

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

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.

Diodes

All diodes in the PDK library are two-terminal.

Units:

Length and width are in meters. Design variables are 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.

11 Supported Devices

Mosfets

- nmos1v - 1.2 volt nominal Vt NMOS transistor
- nmos1v_3 - 1.2 volt nominal Vt NMOS transistor with Inherited Bulk Node
- nmos1v_hvt - 1.2 volt high Vt NMOS transistor
- nmos1v_hvt_3 - 1.2 volt high Vt NMOS transistor with Inherited Bulk Node
- nmos1v_iso – 1.2 volt isolation NMOS transistor
- nmos1v_nat – 1.2 volt native NMOS transistor
- nmos2v – 2.5 volt nominal Vt NMOS transistor
- nmos2v_3 – 2.5 volt nominal Vt NMOS transistor with Inherited Bulk Node
- nmos2v_nat – 2.5 volt native NMOS transistor
- pmos1v – 1.2 volt nominal Vt PMOS transistor
- pmos1v_3 – 1.2 volt nominal Vt PMOS transistor with Inherited Bulk Node
- pmos1v_hvt – 1.2 volt high Vt PMOS transistor
- pmos1v_hvt_3 – 1.2 volt high Vt PMOS transistor with Inherited Bulk Node
- pmos2v – 2.5 volt nominal Vt PMOS transistor
- pmos2v_3 – 2.5 volt nominal Vt PMOS transistor with Inherited Bulk Node

Resistors

- resnsndiff - N+ diffused resistor w/o salicide
- resnsndiff_3 - N+ diffused resistor w/o salicide with Inherited Bulk Node
- resnspdiff - P+ diffused resistor w/o salicide
- resnspdiff_3 - P+ diffused resistor w/o salicide with Inherited Bulk Node
- ressndiff - N+ diffused resistor w/i salicide
- ressndiff_3 - N+ diffused resistor w/i salicide with Inherited Bulk Node
- resspdiff - P+ diffused resistor w/i salicide
- resspdiff_3 - P+ diffused resistor w/i salicide with Inherited Bulk Node
- resnwsti - N-Well resistor under STI
- resnwsti_3 - N-Well resistor under STI with Inherited Bulk Node

- resnwoxide - N-Well resistor under OD
- resnwoxide_3 - N-Well resistor under OD with Inherited Bulk Node
- resnsnpoly - N+ Poly resistor w/salicide
- resnsnpoly_3 - N+ Poly resistor w/salicide with Inherited Bulk Node
- resnsppoly - P+ Poly resistor w/salicide
- resnsppoly_3 - P+ Poly resistor w/salicide with Inherited Bulk Node
- ressnpoly - N+ Poly resistor w/o salicide
- ressnpoly_3 - N+ Poly resistor w/o salicide with Inherited Bulk Node
- ressppoly - P+ Poly resistor w/o salicide
- ressppoly_3 - P+ Poly resistor w/o salicide with Inherited Bulk Node
- resm<k> - Metal <k> resistor (k=1..9)

Capacitor

- nmoscap1v – 1.2v Nmos cap
- nmoscap1v3 – 1.2v Nmos cap with bulk node
- pmoscap1v – 1.2v Pmos cap
- pmoscap1v3 – 1.2v Pmos cap with bulk node
- nmoscap2v – 2.5v Nmos cap
- nmoscap2v3 – 2.5v Nmos cap with bulk node
- pmoscap2v – 2.5v Pmos cap
- pmoscap2v3 – 2.5v Pmos cap with bulk node

Bipolars

- vpnp2 - 1.2 volt Vertical substrate PNP 2x2
- vpnp5 - 1.2 volt Vertical substrate PNP 5x5
- vpnp10 - 1.2 volt Vertical substrate PNP 10x10
- npn - Bipolar NPN with variable emitter area
- pnp - Bipolar PNP with variable emitter area

Diodes

- pdio – 1.2 volt P+/nwell diode
- ndio - 1.2 volt N+/psub diode

12 Views provided

The following table explains the use of the cellviews provided as part of this PDK:

symbol	Used in Composer schematics
spectre	Simulation / netlisting view for the Spectre & UltraSim simulator
schematic	Simulation / netlisting view for all simulators; Mixed-mode and logic resistors use schematic to call other simulator resistor views. It is used to implement series and parallel features in those resistors.
hspiceD	Simulation / netlisting view for the Hspice simulator
auLvs	Netlisting view for DIVA and Assura
auCdl	Circuit Descriptive Language netlisting view typically used to generate a netlist for Dracula or third party simulators.
ivpcell	Device recognition symbol used in the extracted layout for netlisting purposes with DIVA and Assura
layout	Fixed cell or pcell used in Virtuoso Layout Editor.

Mosfets

- Four terminals (D, G, S, B)
- symbol, spectre, hspiceD, auLvs, auCdl, ivpcell, layout (Pcells)

Resistors

- Three terminals (PLUS, MINUS, B) for diffused resistors
- Two terminals (PLUS, MINUS) for poly and metal resistors
- symbol, schematic, auLvs, auCdl, ivpcell, layout (Pcells)
- Resistors called in schematic views include views for all simulators, symbol, spectre, hspiceD, auLvs, auCdl, ivpcell

Capacitor

- Three terminals (really four -- S/D overlapped) (G, D/S, B) for mos caps
- symbol, spectre, hspiceD, auLvs, auCdl, ivpcell, layout (Pcells)

Diodes

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

Bipolars

- Three terminals (C, B, E)
- symbol, spectre, hspiceD, auLvs, auCdl, ivpcell,

13 CDF parameters

Mosfets

Model Name - spectre model name (non-editable)

Multiplier - number of Parallel MOS devices

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-fingered devices and addition of contact heads to gate ends

S/D Connection – allow shorting of sources and/or drains on multi-finger devices

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

Edit Area & Perim – allow Drain/Soure area and periphery be entered manually for simulation

Drain diffusion area, etc. – several simulation parameters are presented. The area and perimeter parameters are calculated and netlisted in accordance with the layouts or can be entered manually if “Edit Area & Perim” is checked

Resistors

Model Name – Spectre model name (non-editable)

Segments – number of series or parallel segments for a resistor

Segment Connection – cyclic field used for series or parallel segments

Calculated Parameter – radio button that determines whether resistance or Length is the calculated value when instantiating a new resistor device

Resistance – total resistance value equal to the sum of body resistance, contact resistance, end resistance, and grain resistance

Segment Width – resistor segment width in meters

Segment Length – resistor segment length in meters

Effective Width – effective resistor segment width in meters

Effective Length – effective resistor segment length in meters

Left Dummy – boolean value used to place a dummy resistor strip on the left side of the main resistor

Right Dummy – boolean value used to place a dummy resistor strip on the right side of the main resistor

Contact Rows – integer number of contact rows

Contact Columns – integer number of contact columns

Show Tap Params – boolean value allowing the user to set the visibility of the resistor tap properties

Left Tap – boolean value used to place a resistor tap on the left side of a device

Right Tap – boolean value used to place a resistor tap on the right side of a device

Top Tap – boolean value used to place a resistor tap on the top side of a device

Bottom Tap – boolean value used to place a resistor tap on the bottom side of a device

Tap Extension – float values to set where the left, right, top, and bottom taps would be to its original placements. This parameter is related to the stretch handle on the taps. The input format should be “left 1.3 right 1.0 top 0.0 bottom 2.0” without the quotes. If neither pair is not present, a zero is assumed

Sheet Resistivity – sheet rho value for body of resistor (non-editable)

End Resistance – resistance value for any salicided area near the contact heads in a non-salicided resistor (non-editable)

Contact Resistance – resistance value for the contact heads of a particular resistor (non-editable)

Grain Resistance – constant resistance value for any salicided area near the contact heads in a non-salicided resistor (non-editable)

Delta Width – resistor width process variation value in meters (non-editable)

Delta Length – resistor length process variation value in meters (non-editable)

Temperature Coefficient 1 – temperature coefficient #1 for resistor (non-editable)

Temperature Coefficient 2 – temperature coefficient #2 for resistor (non-editable)

MOScaps

Model Name - spectre model name (non-editable)

Multiplier - number of Parallel MOS devices

Calculated Parameter - Calculated parameter cyclic (capacitance, length, width)

Capacitance – total capacitance

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

Gate Connection – allow shorting of multi-fingered devices and addition of contact heads to gate ends

S/D Connection – allow shorting of sources and/or drains on multi-finger devices

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

Area capacitance – Capacitance per unit area used in parameter calculations (non-editable)

Fringe capacitance – Fringe Capacitance of perimeter used in parameter calculations (non-editable)

Temp rise from ambient, etc. – several simulation parameters are presented.

Bipolars

Model name	Model name used in simulation
Device Area	Emitter area in microns squared (non-editable
Emitter width	Emitter width microns
Multiplier	Number of Parallel Bipolar devices
Estimated operating region	Simulation operating region

Diodes

Model name	Model used for simulation name
Calculate Parameter	Choices are 'area' , 'width' or 'length'
Device Area	Calculated junction area in meters squared (non-editable)
Length (M)	Diode length in meters
Width (M)	Diode width in meters
Multiplier	Number of Parallel Diode devices
Periphery of junction	Calculated junction periphery in meters (non-editable)

14 Model Setup

This PDK supports the Cadence Spectre, Ultrasim, and AMS, circuit simulators, including corner modeling of the MOSFETs.

The following model sections are defined in the <**pdk_install_directory**>/models/spectre/gpdk090.scs file.

Section
NN
FF
SS
FS
SF

15 Techfile Layers

Cadence will provide a standard display setup, and will not support desired changes to the display. The customer is free to modify the display.drf file used on-site to achieve any desired display.

CDS #	GDS #	GDS type	CDS name	CDS purpose	Description
2	1	0	Oxide	drawing	Oxide
4	24	0	Oxide_thk	drawing	Thick Oxide
6	2	0	Nwell	drawing	Nwell
10	3	0	Poly	drawing	Poly
11	18	0	Nhvt	drawing	N+ high Vt
12	4	0	Nimp	drawing	N+ implant
13	23	0	Phvt	drawing	P+ high Vt
14	5	0	Pimp	drawing	P+ implant
15	52	0	Nzvt	drawing	Native Nmos
16	72	0	SiProt	drawing	Salicide Blocking
18	19	0	Nburied	drawing	N buried
20	6	0	Cont	drawing	Contact
30	7	0	Metal1	drawing	Metal1
32	8	0	Via1	drawing	Via1
34	9	0	Metal2	drawing	Metal2
36	10	0	Via2	drawing	Via2
38	11	0	Metal3	drawing	Metal3
40	30	0	Via3	drawing	Via3
42	31	0	Metal4	drawing	Metal4
44	32	0	Via4	drawing	Via4
46	33	0	Metal5	drawing	Metal5
48	34	0	Via5	drawing	Via5
50	35	0	Metal6	drawing	Metal6
52	37	0	Via6	drawing	Via6
54	38	0	Metal7	drawing	Metal7
56	39	0	Via7	drawing	Via7
58	40	0	Metal8	drawing	Metal8
60	41	0	Via8	drawing	Via8
62	42	0	Metal9	drawing	Metal9
30	7	1	Metal1	pin	Pin purpose
34	9	1	Metal2	pin	Pin purpose
38	11	1	Metal3	pin	Pin purpose
42	31	1	Metal4	pin	Pin purpose
46	33	1	Metal5	pin	Pin purpose
50	35	1	Metal6	pin	Pin purpose
54	38	1	Metal7	pin	Pin purpose

58	40	1	Metal8	pin	Pin purpose
62	42	1	Metal9	pin	Pin purpose
30	7	3	Metal1	label	Label purpose
34	9	3	Metal2	label	Label purpose
38	11	3	Metal3	label	Label purpose
42	31	3	Metal4	label	Label purpose
46	33	3	Metal5	label	Label purpose
50	35	3	Metal6	label	Label purpose
54	38	3	Metal7	label	Label purpose
58	40	3	Metal8	label	Label purpose
62	42	3	Metal9	label	Label purpose
30	7	4	Metal1	net	Net purpose
34	9	4	Metal2	net	Net purpose
38	11	4	Metal3	net	Net purpose
42	31	4	Metal4	net	Net purpose
46	33	4	Metal5	net	Net purpose
50	35	4	Metal6	net	Net purpose
54	38	4	Metal7	net	Net purpose
58	40	4	Metal8	net	Net purpose
62	42	4	Metal9	net	Net purpose
71	7	2	Metal1	slot	Slot purpose
72	9	2	Metal2	slot	Slot purpose
73	11	2	Metal3	slot	Slot purpose
74	31	2	Metal4	slot	Slot purpose
75	33	2	Metal5	slot	Slot purpose
76	35	2	Metal6	slot	Slot purpose
77	38	2	Metal7	slot	Slot purpose
78	40	2	Metal8	slot	Slot purpose
79	42	2	Metal9	slot	Slot purpose
80	25	0	Psub	drawing	P substrate
82	22	0	DIOdummy	drawing	Recognition layer for diodes
84	21	0	PNPdummy	drawing	Recognition layer for pnp
86	20	0	NPNdummy	drawing	Recognition layer for npn
88	17	0	IND2dummy	drawing	Recognition layer for inductor
90	16	0	INDdummy	drawing	Recognition layer for inductor
92	15	0	BJTdum	drawing	Recognition layer for vpnp
93	84	0	Cap3dum	drawing	Recognition layer for moscap
94	13	0	Resdum	drawing	Recognition layer for resistor
95	36	0	Bondpad	drawing	Recognition layer for bondpad
96	12	0	Capdum	drawing	Recognition layer for moscap
97	14	0	CapMetal	drawing	Recognition layer for moscap
98	71	0	ResWdum	drawing	Recognition layer for resistor
99	75	0	M1Resdum	drawing	Recognition layer for metal res
100	76	0	M2Resdum	drawing	Recognition layer for metal res
101	77	0	M3Resdum	drawing	Recognition layer for metal res
102	78	0	M4Resdum	drawing	Recognition layer for metal res

103	79	0	M5Resdum	drawing	Recognition layer for metal res
104	80	0	M6Resdum	drawing	Recognition layer for metal res
105	81	0	M7Resdum	drawing	Recognition layer for metal res
106	82	0	M8Resdum	drawing	Recognition layer for metal res
107	83	0	M9Resdum	drawing	Recognition layer for metal res
108	60	0	VPNP2dum	drawing	Recognition layer for vpnp
109	61	0	VPNP5dum	drawing	Recognition layer for vpnp
110	62	0	VPNP10dum	drawing	Recognition layer for vpnp
114	70	0	IND3dummy	drawing	Recognition layer for inductor
115	74	0	ESDdummy	drawing	Recognition layer for esd

16 Virtuoso XL

The standard Cadence Virtuoso XL design flow will be implemented. This includes basic connectivity of connection layers, wells, and substrate, and symbolic contacts. The M factor will be used for device instance multiplier - there will be no conflict with the parameter used in cell operation. Names will 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 device is also supported. The skill 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 by design layout. The Virtuoso Custom Router (VCR) can be used to finish the unconnected interconnect in the layout.

The VCR rules file for the target process is provided with the PDK.

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.

17 Diva Decks

These decks can be found in the extracted PDK directory tree in the ‘diva’ directory.

A link to the Diva decks is also located under the ‘libs.cdb/gpdk090’ and the ‘libs.oa22/gpdk090’ directory.

Diva DRC

Those files are based on the design rules outlined in the design rule manual. The files are:

- divaDRC.rul

Diva Extract

The DIVA Extract rule decks, divaEXT.rul

Diva LVS

The parameters checked in LVS, divaLVS.rul, include:

- MOS Devices - type, length and combination of width, "m" factor
- Resistors - type, width, and length
- Diodes - type, area, "m" factor
- BJT’s - type, emitter area, "m" factor

18 Assura Decks

Cadence has developed the Assura DRC, LVS, and RCX rule files from the documentation provided.

These decks can be found in the extracted PDK directory tree in the directory:

- assura

Assura DRC

The Assura DRC file provided is named

- drc.rul

Assura ANTENNA

The Assura Antenna file provided is named

- antenna.rul

Assura LVS

The Assura LVS files provided are located in the pv/assura directory and named

- extract.rul
- compare.rul

Assura RCX

The Assura RCX files provided are located in the following directory

- assura/rcx - Directory where Assura RCX files are provided