Cadence Design Systems

Analog Mixed-Signal Foundation Flow (AMSFF)

Cadence 45nm Generic Standard Cells User Guide

2014 – April

CONFIDENTIAL NOTICE

Cadence Design Systems, Inc., 555 River Oaks Parkway, San Jose, CA 95134, USA

Table of Contents

1 OVERVIEW 4

2 DATABASE STRUCTURE 5

2.1 UNIX/LINUX DATABASE FOLDERS 5

2.2 VIRTUOSO LIBRARY AND CELLVIEWS 5

2.2.1 The Standard Cells with Standard VT: gsclib045 6

2.2.2 The Standard Cells with High VT: gsclib045_hvt 7

2.2.3 The Standard Cells with Low VT: gsclib045_lvt 8

2.2.4 The Standard Cells with Back-Bias: gsclib045_backbias 9

2.3 RECOMMENDED LOCATION IN IC DESIGN DATABASE 10

2.4 RECOMMENDED VIRTUOSO TECHNOLOGY BINDING 11

3 DATABASE CONTENTS 12

3.1 VIRTUOSO FRONT-TO-BACK DATA 14

3.1.1 Virtuoso Front-End Design Data (symbol, functional, schematic, extracted) 14

3.1.2 Virtuoso Back-End Design Data (layout, abstract) 18

3.2 ENCOUNTER DIGITAL DATA 23

3.2.1 The Verilog Files (.v) 23

3.2.2 The Liberty Timing Files (.lib) 24

3.2.3 The LEF Files (.lef) 25

3.2.4 The GDS Files (.gds) 25

3.3 PHYSICAL VERIFICATION DATA 26

3.3.1 The CDL Netlist Files (.cdl) 26

3.3.2 The QRC Extraction Technology File 26

4 SUMMARY 27

5 APPENDIXES 28

5.1 STANDARD CELLS DEVELOPMENT 28

5.1.1 The GSCLIB045 common development area 29

5.1.2 Library Migration (Symbol and Schematic): gsclib090 -> gsclib045 30

5.1.3 Physical Implementation – layout 30

5.1.4 The Netlist Generation from 'schematic' view - CDL and Spectre Netlist 31

5.1.5 The Cell Characterization - Liberty Timing and Verilog 31

5.1.6 The Physical Verification and Extraction – DRC/LVS/EXT 31

5.1.7 The Cell Abstract Creation - 'abstract' view and Macro LEF 31

5.2 CADENCE 45NM GENERIC PROCESS DESIGN KIT – GPDK045 32

1 OVERVIEW

The Cadence 45nm generic standard cells are for demonstrate Cadence tools features and capabilities. The physical data is based to Cadence 45nm generic PDK that mimics to the 45nm silicon process. Therefore, the accuracy in device size, extracted parasitics, and timing may significantly differ from the actual 45nm process data.

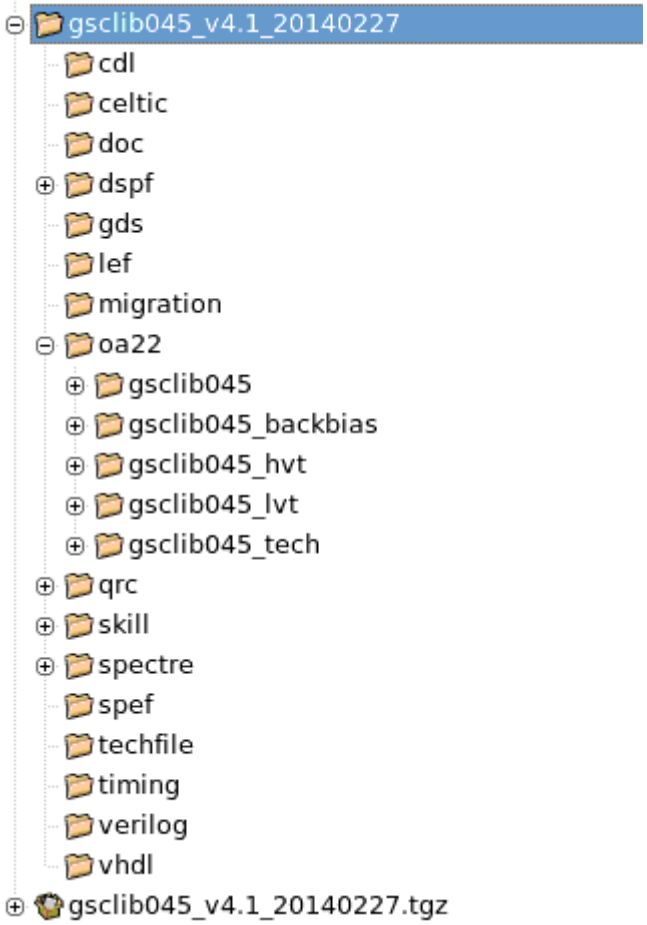
The Cadence 45nm generic standard cell database is often referred to as "GSCLIB045" in this document and among Cadence users. The Virtuoso database or library name is often referred to as "gsclib045" library. The main objective of this user guide document is to help users to know the GSCLIB045 database structure and its contents.

The GSCLIB045 supports to mixed-signal design that has both of analog and digital contents. The datasets for Virtuoso IC platform are the data sources; and the datasets for Encounter Digital platform are derived data.

2 DATABASE STRUCTURE

The GSCLIB045 data structure in Unix/Linux folders and Cadence Virtuoso Library Manager are as following.

2.1 Unix/Linux Database Folders



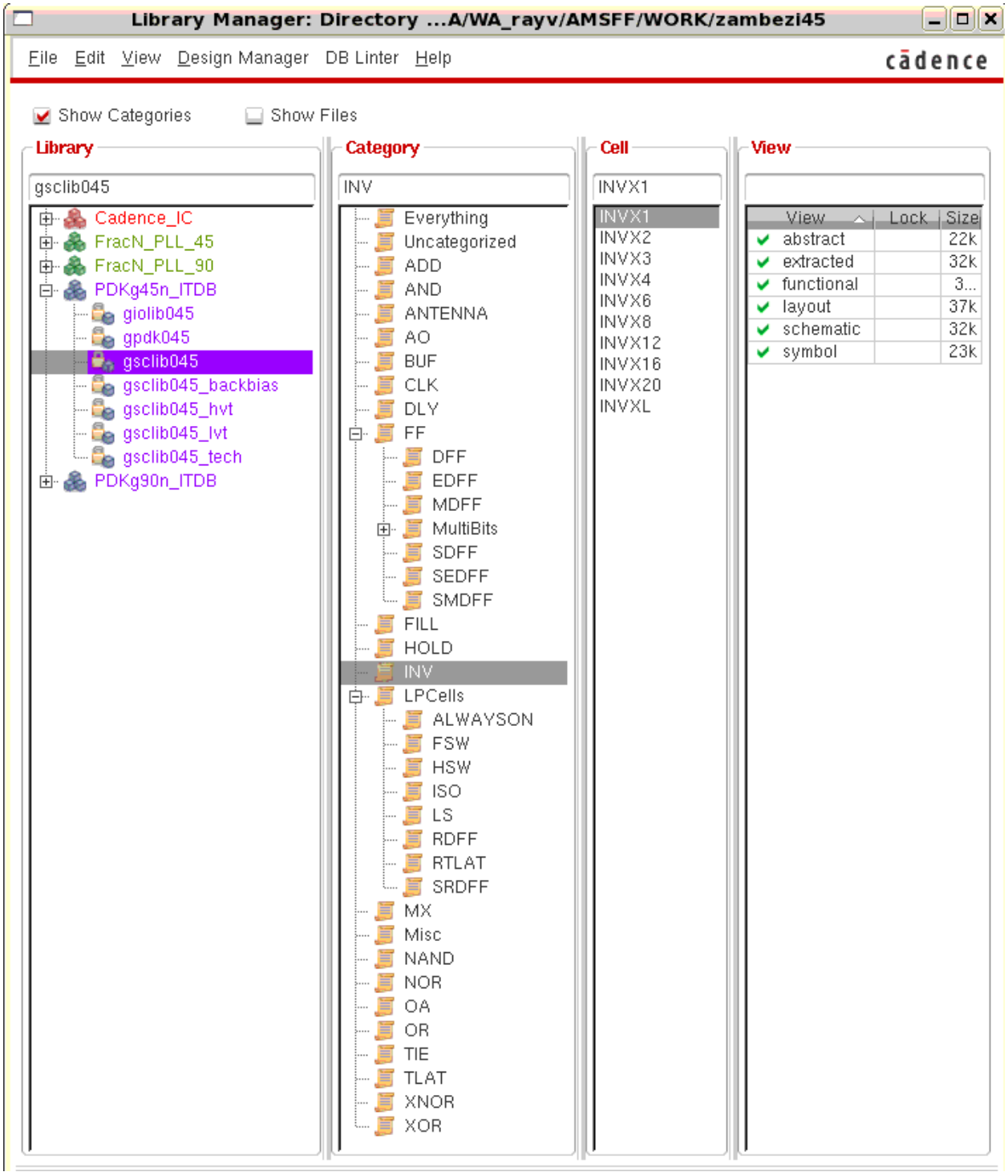
⇔ The inflated data

⇔ The compressed tar file

2.2 Virtuoso Library and Cellviews

The Virtuoso library 'gsclib045' binds to the 'gsclib045_tech' which an incremental technology database (ITDB). The standard cells are arranged into categories; and each standard cell has multiple views for supporting to front-end and back-end design.

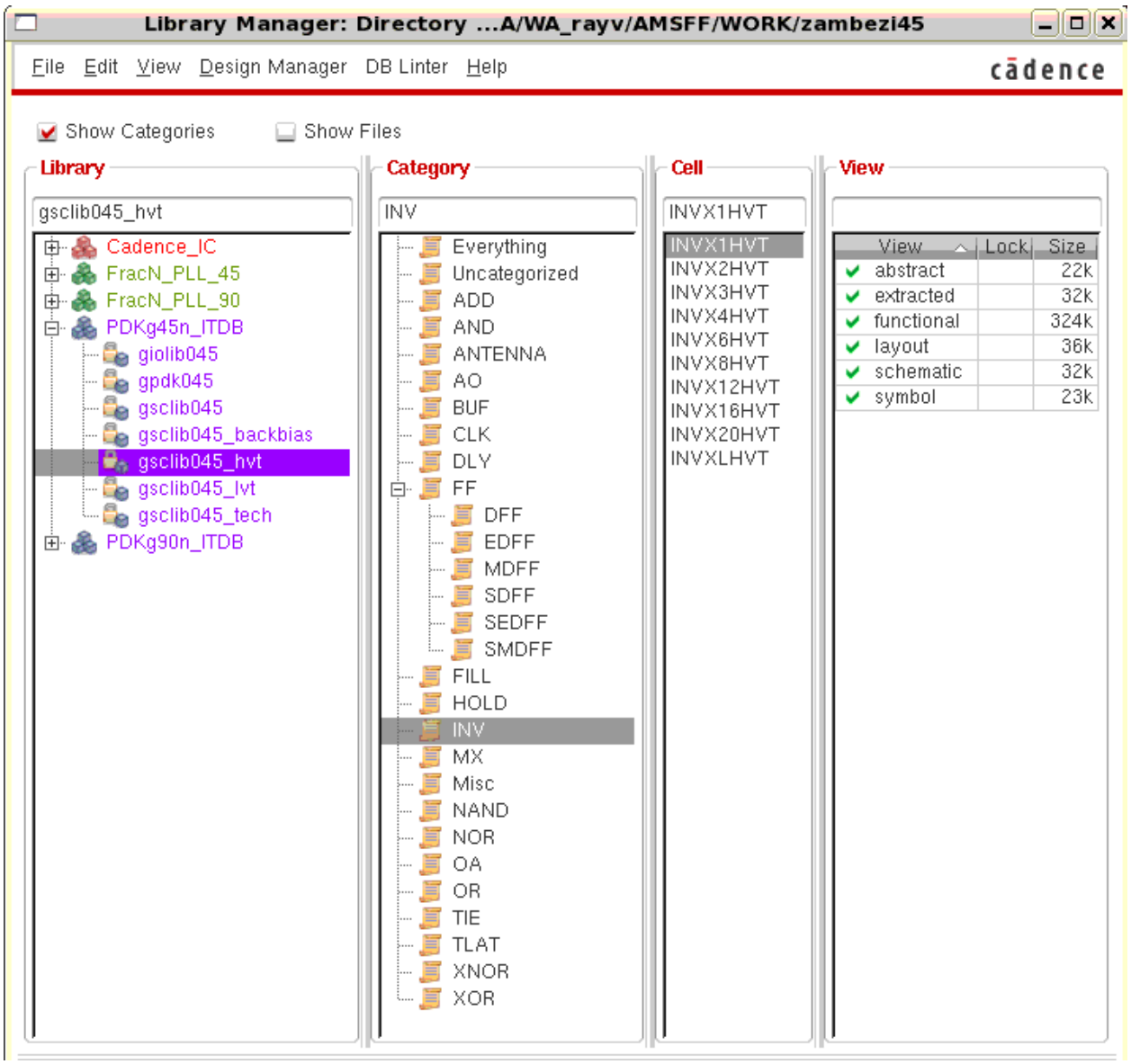
2.2.1 The Standard Cells with Standard VT: gsclib045



Note:

- The standard cells in *gsclib045* library use the MOSFET devices with standard threshold voltage (Vt).
- All cell names are with no suffix.

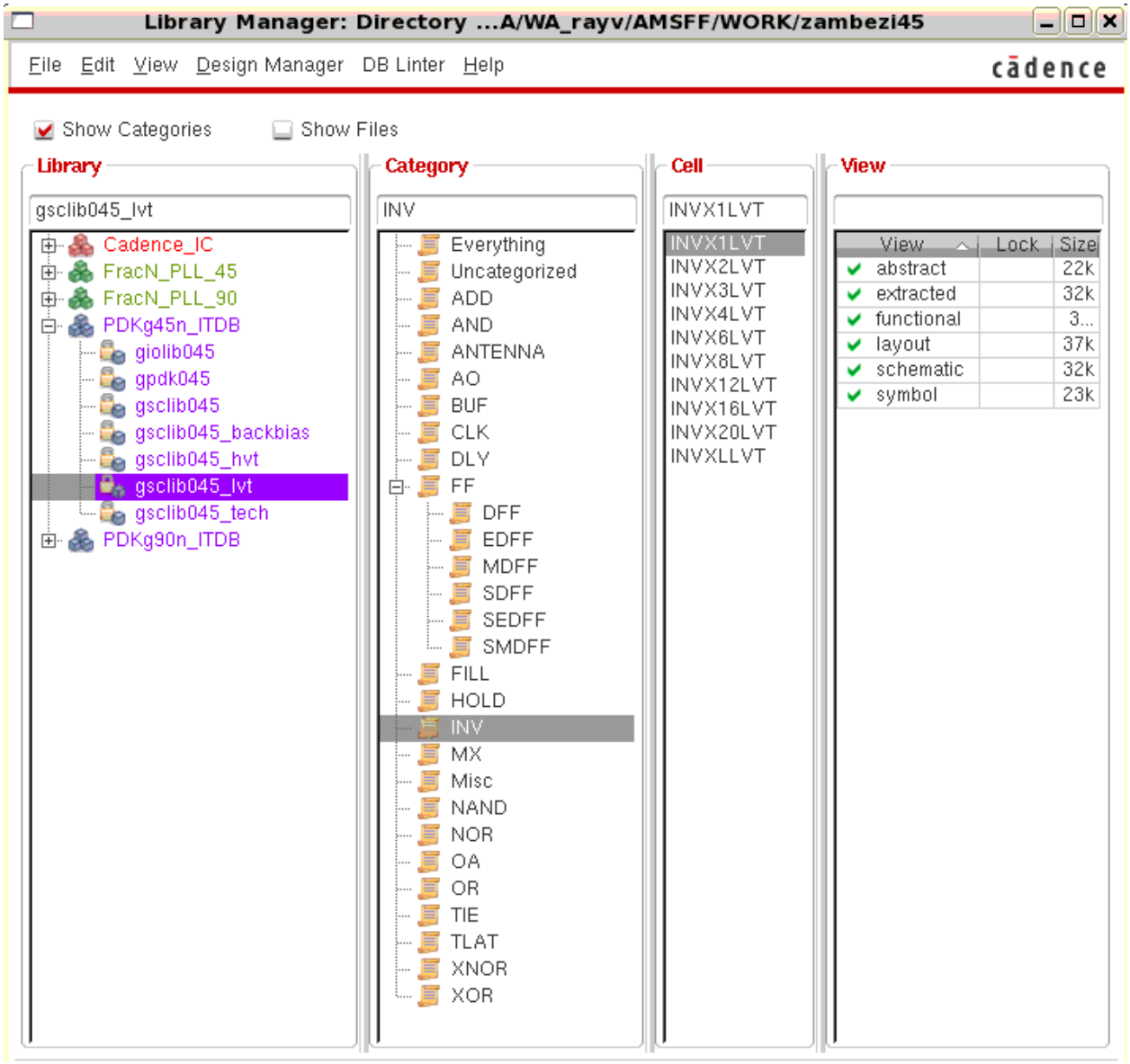
2.2.2 The Standard Cells with High VT: gsclib045_hvt



Notes:

- The standard cells in *gsclib045_hvt* library use the MOSFET devices with high threshold voltage (Vt).
- All cell names are with "HVT" suffix.

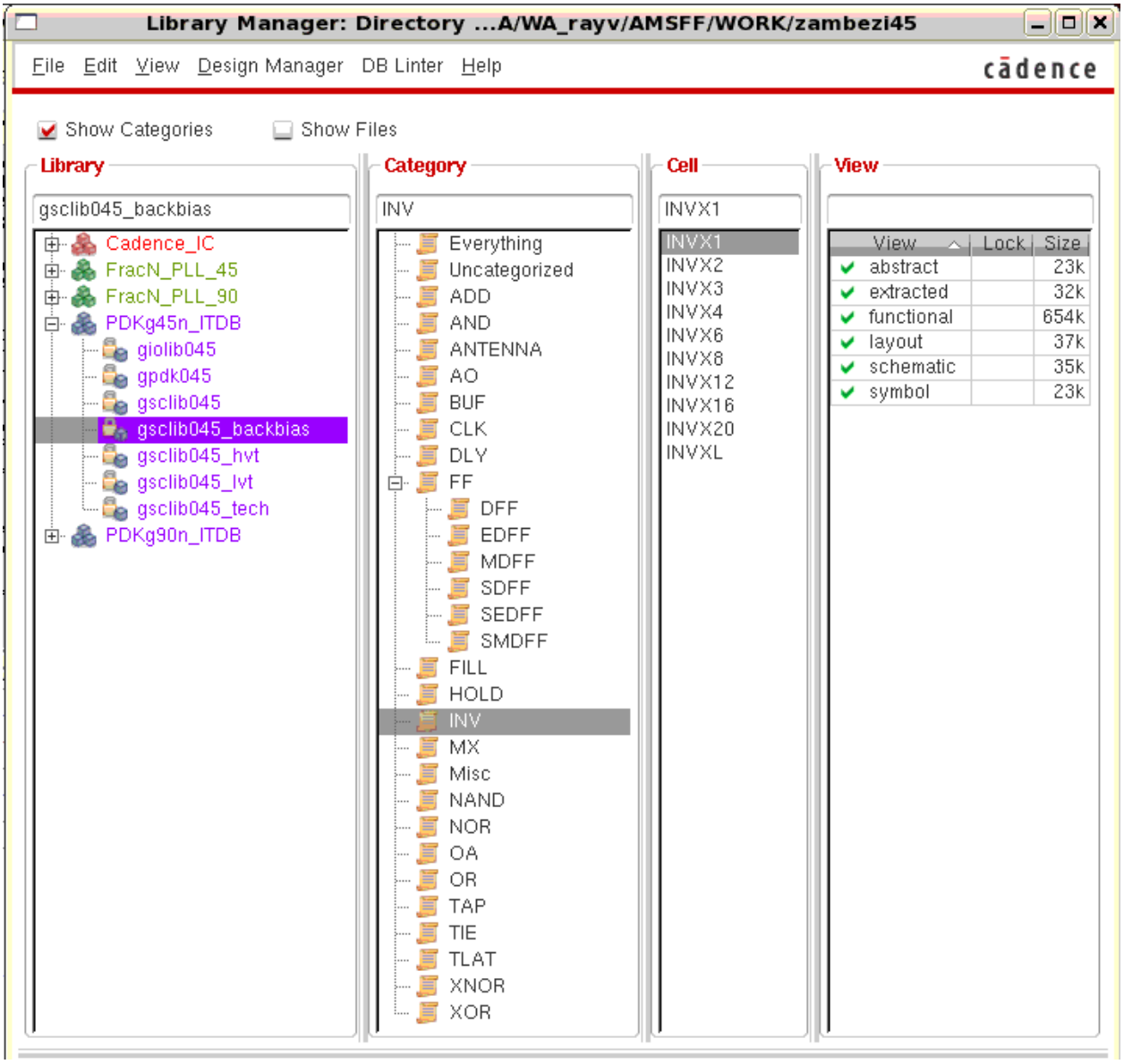
2.2.3 The Standard Cells with Low VT: gsclib045_lvt



Notes:

- The standard cells in *gsclib045_lvt* library use the MOSFET devices with low threshold voltage (Vt).
- All cell names are with "LVT" suffix.

2.2.4 The Standard Cells with Back-Bias: gsclib045_backbias

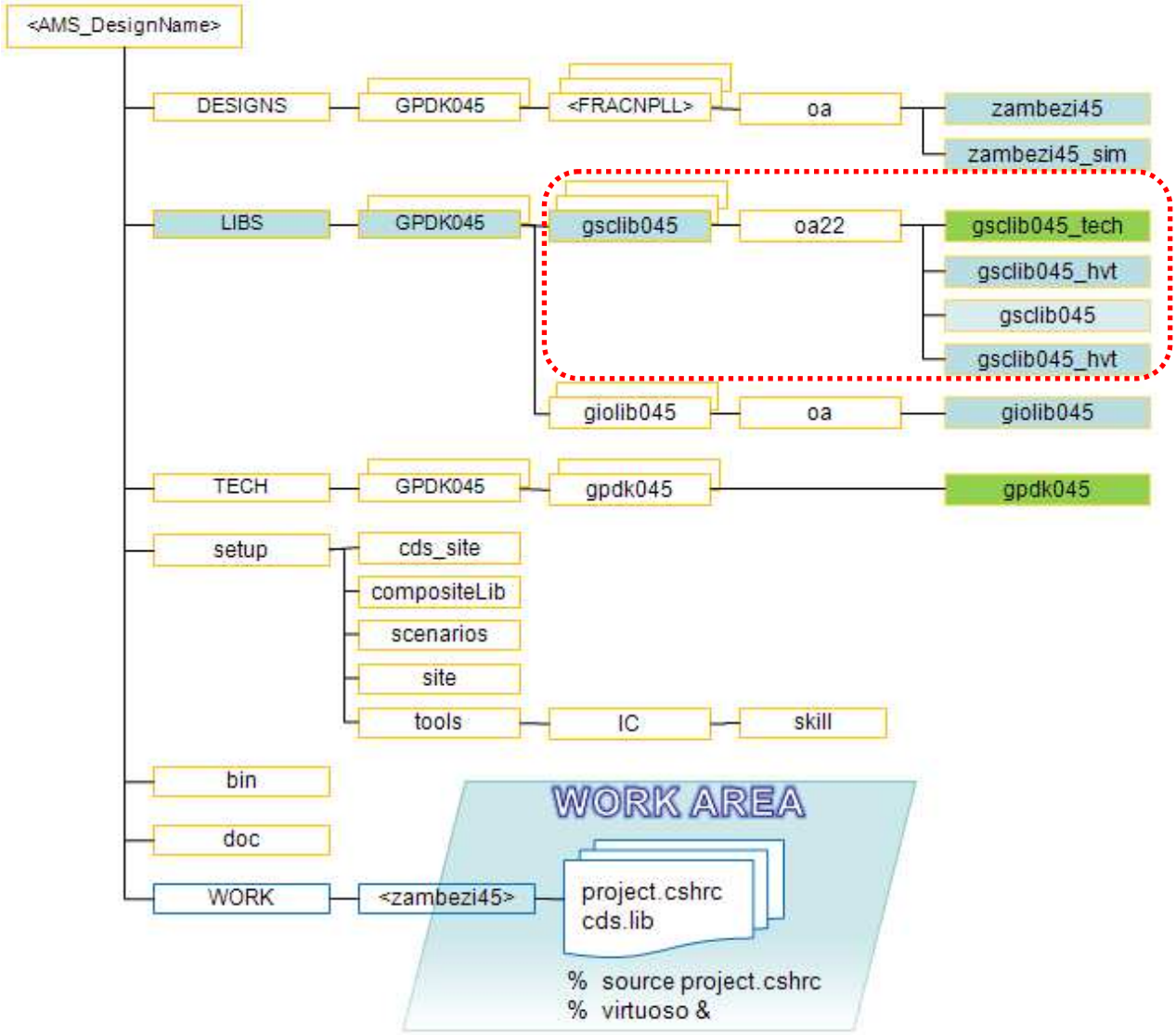


Notes:

- The standard cells in *gsclib045_backbias* library use the MOSFET devices with standard threshold voltage (Vt).
- All cell names are with no suffix.

2.3 Recommended Location in IC Design Database

The recommended location for *GSCLIB045* database in an IC design database is a sub-directory to in the LIBS/GPDK045 directory as in below database diagram. The 'gsclib045' is a symbolic link to the specific version of 'gsclib045_vxx' database.

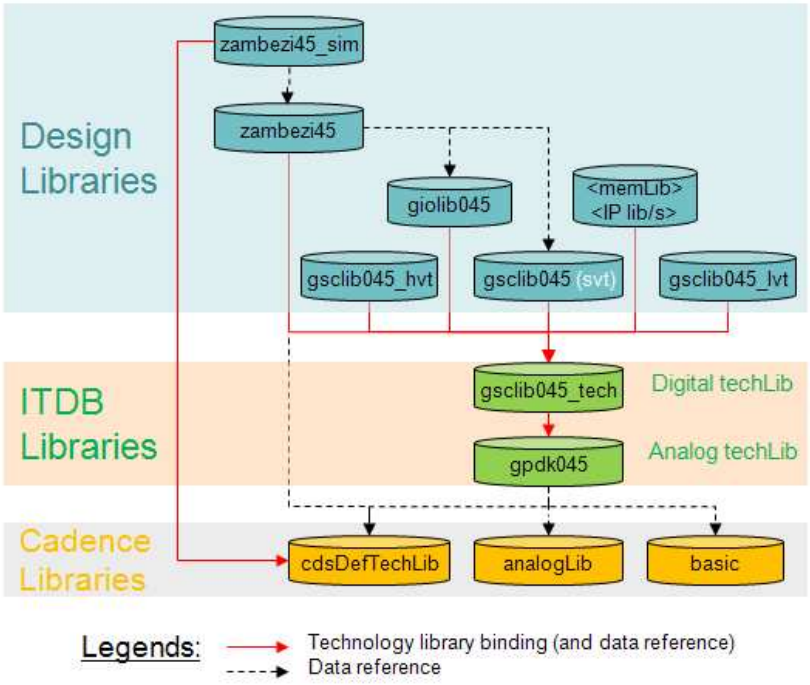


Important Note:

The *gpdk045* technology library is not bundle in *gsclib045_vxx.tgz* file; and user should obtain it separately.

2.4 Recommended Virtuoso Technology Binding

The Virtuoso '*gsclib045*' library is one of design libraries. The *gsclib045* library binds to *gsclib045_tech* technology library. The '*gsclib045_tech*' library is an incremental technology database (ITDB) that binds to *gpdk045* PDK library as shown in below diagram.



Notes:

- Digital techLib contents:
  - refTechLibs() for ITDB
  - P&R custom vias
  - P&R constraint groups
- Analog techLib contents:
  - Layer and display definitions
  - Foundry process constraints
  - Via definitions (standard vias)
  - Device, path, and guardring definitions
- Abbreviation Used:
  - gpdk: Generic PDK
  - gsclib: Generic Std. Cell Lib
  - giolib: Generic IO Library
  - _hvt: High Vt
  - _lvt: Low Vt
  - zambezi45: PLL design

Important Notes:

- The *gpdk045* technology library is not bundle in *gsclib045_vxx.tgz* file; and user should obtain it separately.
- User should update the *<path>/gsclib045/cds.lib* file with the proper path of *gpdk045* library.
- The *gpdk045* is a base library and the *gsclib045_tech* is a derived library in ITDB structure. Therefore, the *gsclib045* standard cell library needs both of *gsclib045_tech* and *gpdk045* library defined.
- The *gsclib045_tech* library is bundle in *gsclib045_vxx.tgz* file.

3 DATABASE CONTENTS

The *GSCLIB045* standard cell database contains multiple datasets for using in the mixed-signal IC design. This database supports to both of functional design and physical implementation. In Cadence tool environment, we categorize the datasets into two main categories as following.

- Virtuoso custom design datasets
- Encounter digital design datasets

The Virtuoso datasets (Eg: symbol, schematic, layout, extracted, abstract) is the source data; and the Encounter datasets (Eg: .v, .lib, .lef, .gds) is the derived data from the Virtuoso dataset.

The Virtuoso "gsclib045" library provides a collection of low-level logic functions such as AND, OR, inverter, buffers, flip-flops, latches, etc. We arrange the standard cells into multiple libraries and cell categories as in this table.

Cell Category		**Suffix To '*gsclib045*' and Cell Count**				**Drive Strength**
		(svt)	_hvt	_lvt	_backbias	
ADD		12	12	12	12	X1, X2, X4, L
AND		18	18	18	18	X1, X2, X4, X6, X8, L
ANTENNA		1	1	1	1	
AO		48	48	48	48	X1, X2, X4, L
BUF		18	18	18	18	X2, X3, X4, X6, X8, X12, X16, X20
CLK		33	33	33	33	X1, X2, X3, X4, X6, X8, X12, X16, X20
DLY		8	8	8	8	X1, X4
FF	DFF	45	45	45	45	X1, X2, X4, X8, L
	EDFF	12	12	12	12	X1, X2, X4, X8, L
	MDFF	4	4	4	4	X1, X2, X4, X8
	SDFF	44	44	44	44	X1, X2, X4, X8, L
	SEDFF	13	13	13	13	X1, X2, X4, X8, L
	SMDFF	4	4	4	4	X1, X2, X4, X8
	MultiBits	16				X1, X2
FILL		7	7	7	7	1, 2, 4, 8, 16, 32, 64
HOLD		1	1	1	1	X1
INV		10	10	10	10	X1, X2, X3, X4, X6, X8, X12, X16, X20, L
LPCells	ALWAYSON	2				X1, X2
	FSW	2				X1
	HSW	4				X1
	ISO	8				X1
	LS	20				X1
	RDFF	16				X1
	RTLAT	3				X1
	SRDFF	16				X1
MX		6	6	6	6	X1, X2, X4, L
Misc		1	1	1	1	X2

NAND	34	34	34	34	X1, X2, X4, X6, X8, L
NOR	34	34	34	34	X1, X2, X4, X6, X8, L
OA	48	48	48	48	X1, X2, X4, L
OR	18	18	18	18	X1, X2, X4, X6, X8, L
TIE	2	2	2	2	
TLAT	32	32	32	32	X1, X2, X3,X4, X6, X8, X12, X16, X20,L
XNOR	6	6	6	6	X1, X2, X4, L
XOR	6	6	6	6	X1, X2, X4, L
Sum:	567	481	481	481	

Note: The abbreviations used in the category name

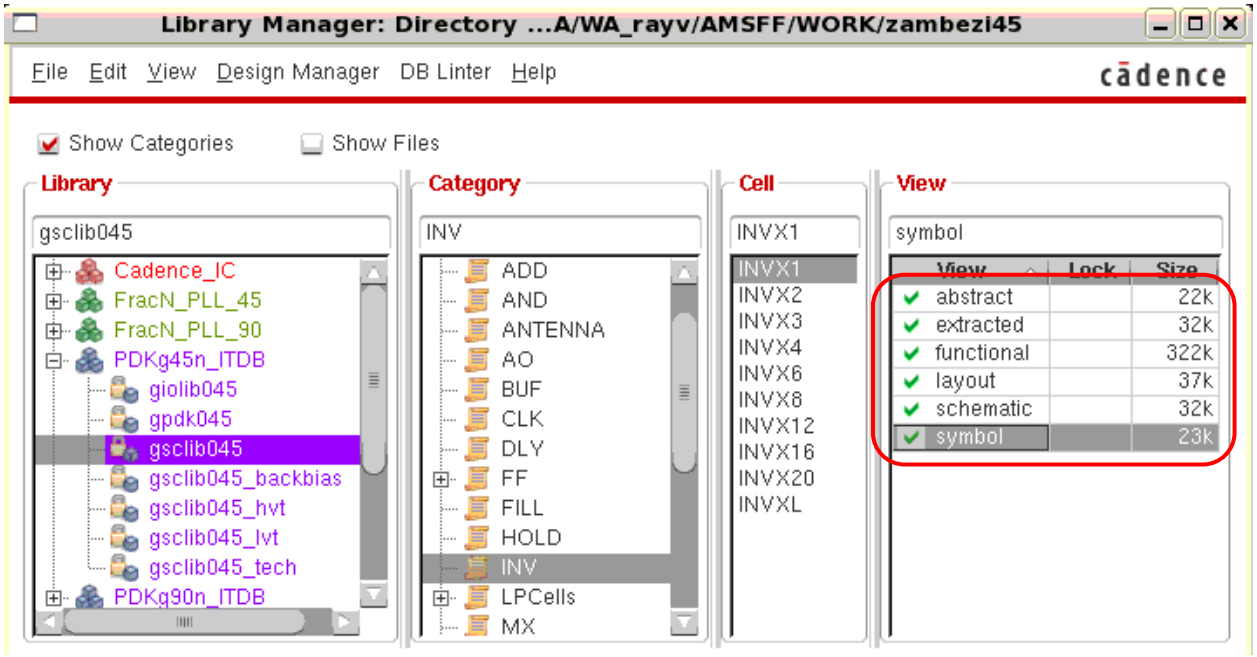
ADD = Adder, AO = AND OR, BUF = Buffer,
CLK=Clock, DLY = Delay, FF = Flip-Flop,
FILL = Filler, INV =Inverter, LPCells = Low Power Cells,
MX = Mixer, OA = OR AND, TLAT = Transmission Latch,
XNOR = Exclusive NOR, XOR = Exclusive OR

In mixed-signal IC chip design, the standard cell provides the basic logic functions to both of custom and digital design tools. Each standard cell has its own unique digital logic functions. In transistor level simulation and physical implementation, the digital logics are derived to transistor scheme in 'schematic' view; then the layout is created with transistors specified in the *schematic* view. Each standard cell uses a common cell grid factors of height and width for maximize the P&R density, routability, and consistency.

3.1 Virtuoso Front-to-Back Data

Cadence Virtuoso IC platform provides the front-to-back design tools for analog, custom digital, and mixed-signal design. The analog and MS circuitry design needs standard cells with multiple datasets for rapid simulations with different accuracy levels by using the *functional*, *schematic*, and *extracted* view. The Virtuoso Layout suite provides the capability of implementing the physical layout with using the *schematic* as guide; and later it generates the *abstract* view.

For providing datasets to the Virtuoso IC tool suites, each standard cell in *gsclib045* library has multiple views as in this sample snapshot of "INVX1"inverter cell.

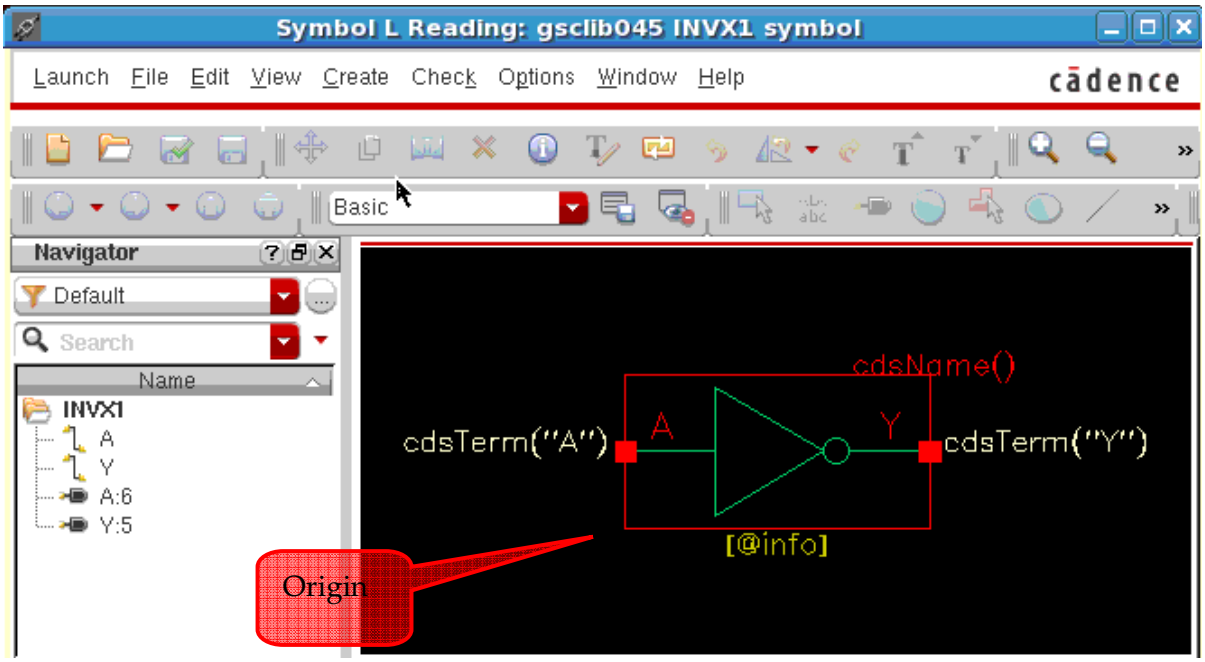


3.1.1 Virtuoso Front-End Design Data (symbol, *functional*, *schematic*, *extracted*)

The Virtuoso Front-End design tool provides the design capability of circuitry composing and simulation. The simulation run-time varies with the using view; and the accuracy is reciprocal to the run-time. The *functional* view provides the fast logic function with shortest simulation run-time. The *schematic* view provides more detail in transistor level with longer run-time and bigger result data. The *extracted* view provides additional RC parasitics in the layout to the design transistors; and the simulation run-time is longest one.

3.1.1.1 The 'symbol' view

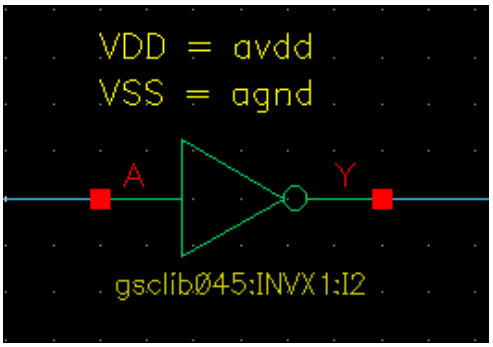
In graphical circuitry design, the *symbol* view is a representative of *schematic* view for use in hierarchical structure. The *symbol* view must have pins match to the schematic terminals (unless the schematic terminal has *netExpression* for inherited connection).



The above inverter symbol has two terminals or pins "A" and "Y". The supply pins "VDD" and "VSS" are excluded in *symbol* view; and this inverter symbol is the *implicit* pin symbol. The symbol origin is on the low-left corner of the selection box (red rectangle).

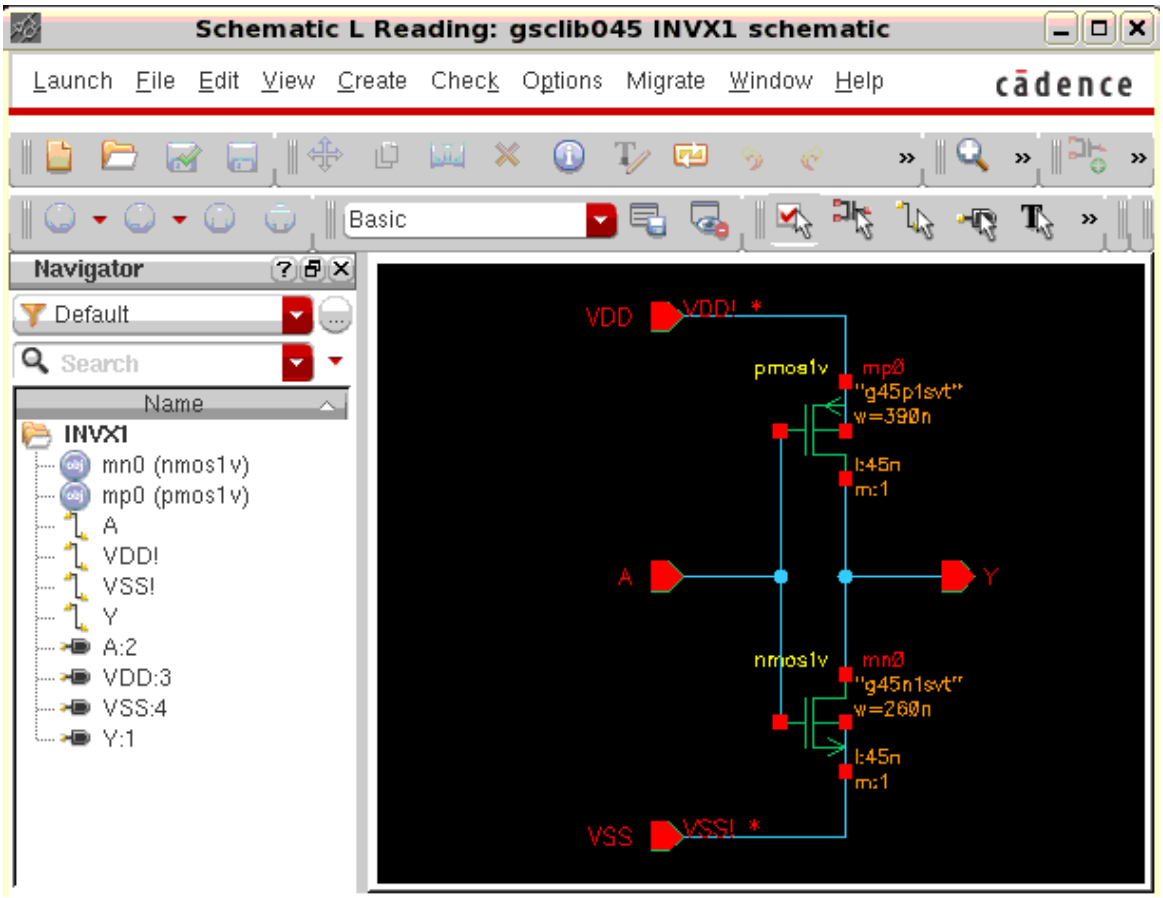
Notes:

- When the *schematic* view of INVX1 cell has *netExpression* defined to VDD terminal and VSS terminal; use can assign the specific supply to the symbol instance via adding *netSet* property in the '*Edit Object Properties*' form.
- Sample snapshot of an inverter instance that user sets 'avdd' to VDD terminal and 'agnd' to VSS terminal.



3.1.1.2 The 'schematic' view

The *schematic* view, which derived from the cell logic functions to transistor structure, has the MOSFET symbol instances with device size, connection wires, and terminals. This *schematic* view guides the layout work; and is a source for Spectre and CDL netlist. The standard cell must have matched LVS (layout vs. schematic) and clean DRC. The terminals between *schematic*, *symbol*, and *layout* view must match and correspond.



The above snapshot is the *schematic* view of INVX1 cell. The INVX1 schematic has four terminals that include two supply terminals for power (VDD) and ground (VSS). The supply terminals are with *netExpression* inherited-connection that defaults to the global VDD! and VSS! net. Therefore, user can assign the VDD and VSS terminal to a desired net via the *netSet* property to the symbol instance. Without specifying or in defaulted symbol instance, the VDD and VSS supply terminal will automatically connect to the global VDD! and VSS!.

In standard cell library development, the schematic view is a mandated item for layout guidance and for source of CDL and Spectre netlist. In standard cell IP package, the *schematic* view may be optional; and user can use the Spectre or CDL netlist for simulation and physical verification. The GSCLIB045 database provides Spectre and CDL netlist in these directories:

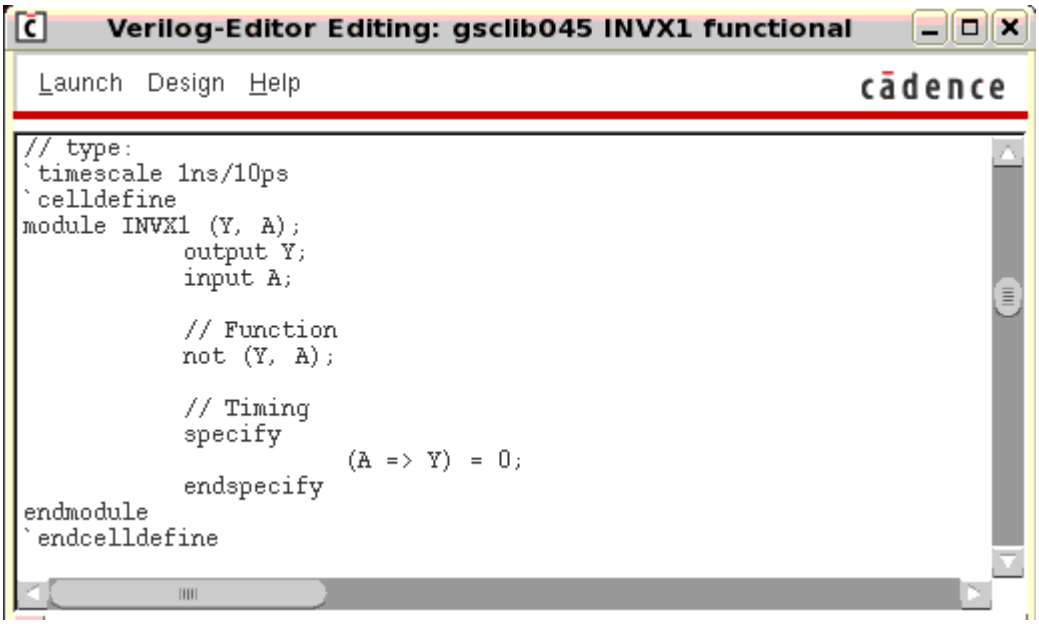
- Spectre netlist: `<path>/gsclib045/spectre`
- CDL netlist: `<path>/gsclib045/cdl`

3.1.1.3 The 'functional' view

The *functional* view provides digital logic functions to Spectre simulator and Encounter RTL synthesis. The Verilog compiling builds this *functional* view by link to a Verilog file with either all standard cells or a single cell. For maintenance efficiency, we recommend a single Verilog file (.v) with all standard cells. For creating the *functional* view to a specific Verilog file, please refer this shell script.

```
<path>/gsclib045/verilog/compile_gpdk045_functional.csh
```

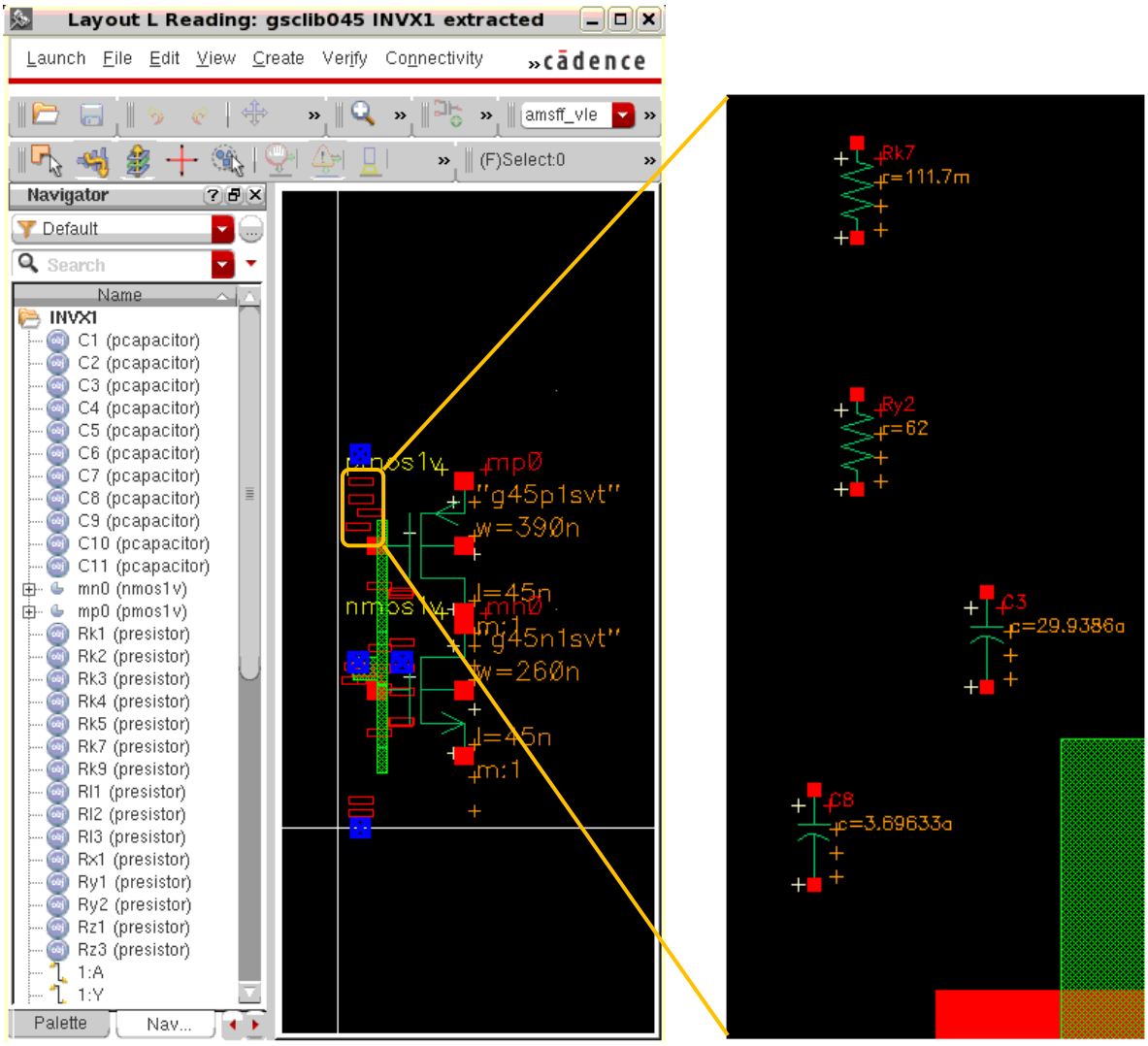
The *functional* view in Virtuoso database is mainly for use in the mixed-signal simulation. The below snapshot is the *functional* view of INV1X cell.



It is obvious that this inverter has zero delay time in *functional* view; due to the delay time is given in the associated liberty-timing file (.lib).

3.1.1.4 The 'extracted' view

The *extracted* view contains the extracted design devices and the R/C parasitics in the *layout* view. Including the extracted parasitics to the detail simulation is crucial to the critical block design. The below snapshot is the *extracted* view of INVX1 cell.



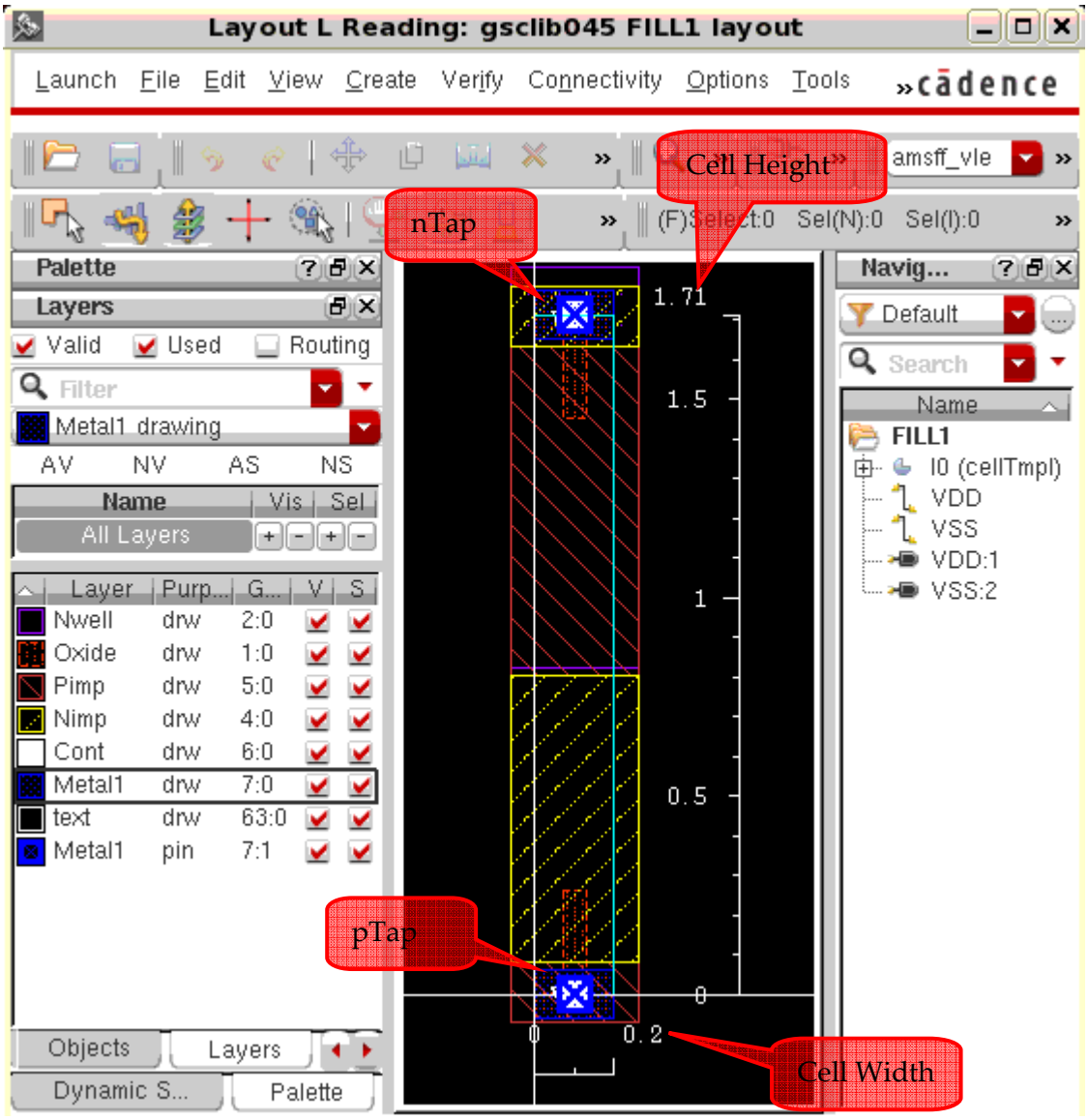
3.1.2 Virtuoso Back-End Design Data (layout, *abstract*)

The Virtuoso datasets for back-end design is in the *layout* and *abstract* view. The *layout* view provides the complete dataset for tapeout and generating the *abstract* view.

3.1.2.1 Cell Grid (height and width)

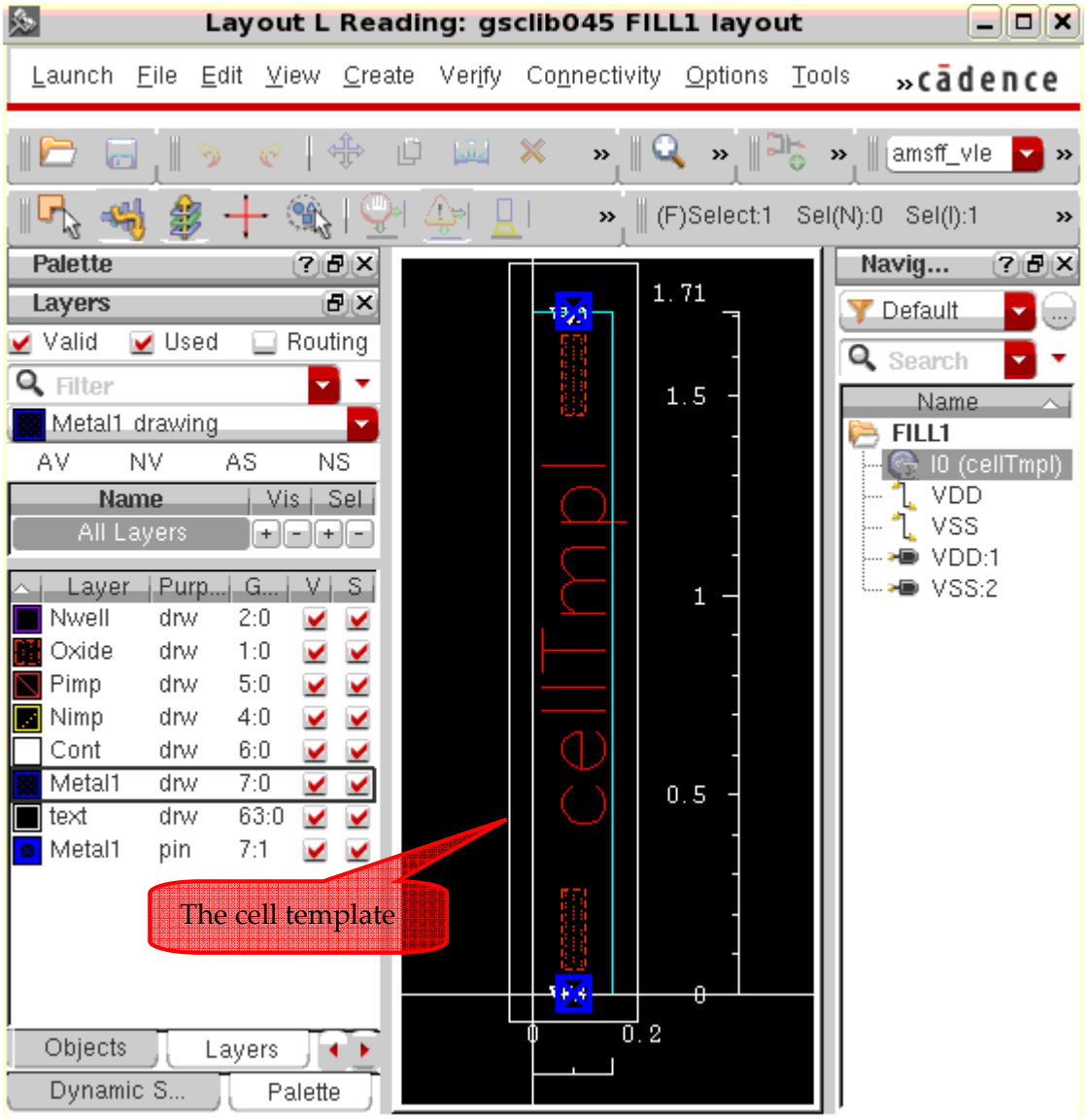
The *gsclib045* standard cells use a common grid factors that defined in a filler cell called "FILL1" as following:

- Height = 1.71u (Ref: The boundary box height - middle of VSS rail to VDD rail)
- Width = 0.20u (Ref: The boundary box width)



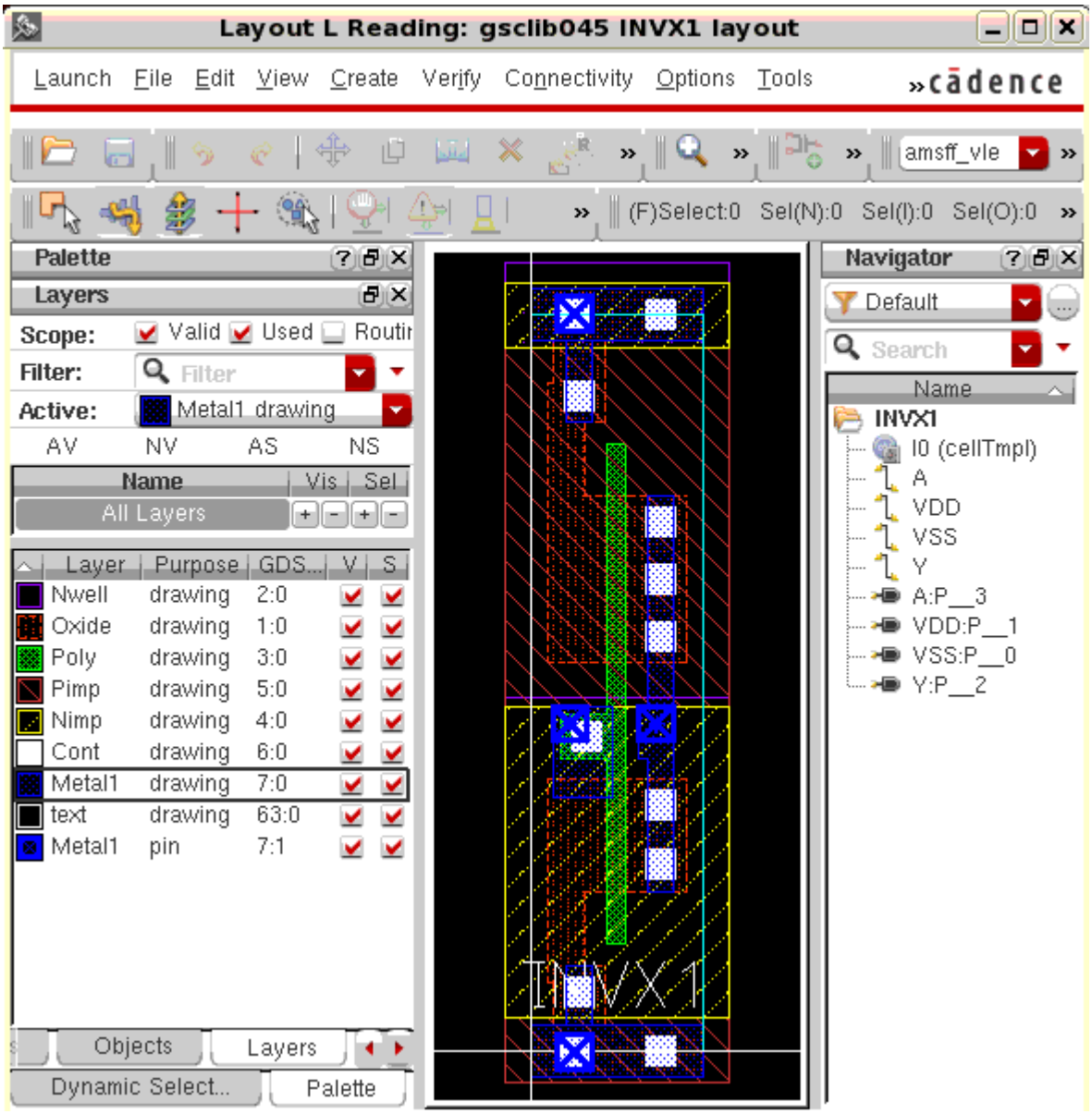
All standard cells in this library use this common grid as a multiply factor for specify the cell height and width. For instance, the INVX cell has cell height of 1.71u (which is from 1 x 1.71u height) and cell width of 0.40u (which is from 2 x 0.20u width). The double height cells such as the low-power cells (Eg: PINVX1) have cell height of 3.42u which is from '2 x 1.71u height'.

It is obvious that the *gsclib045* cells use a template cell called "cellTmpl" for define the common polygons such as Nwell, Pimp, and Nimp. The template cell also provides the *nTap* to the VDD rail and *pTap* to the VSS rail.



3.1.2.2 The 'layout' view

The *layout* view contains the final and detail polygons for mask making. This view must comply with all process design rules and have the matched LVS. The terminals in *layout* view must locate in the routing grid and same name as in the *schematic* and *symbol* view. The *layout* view is a source or supper set of the *abstract* view. The below snapshot is the *layout* view of *INVX1* cell.

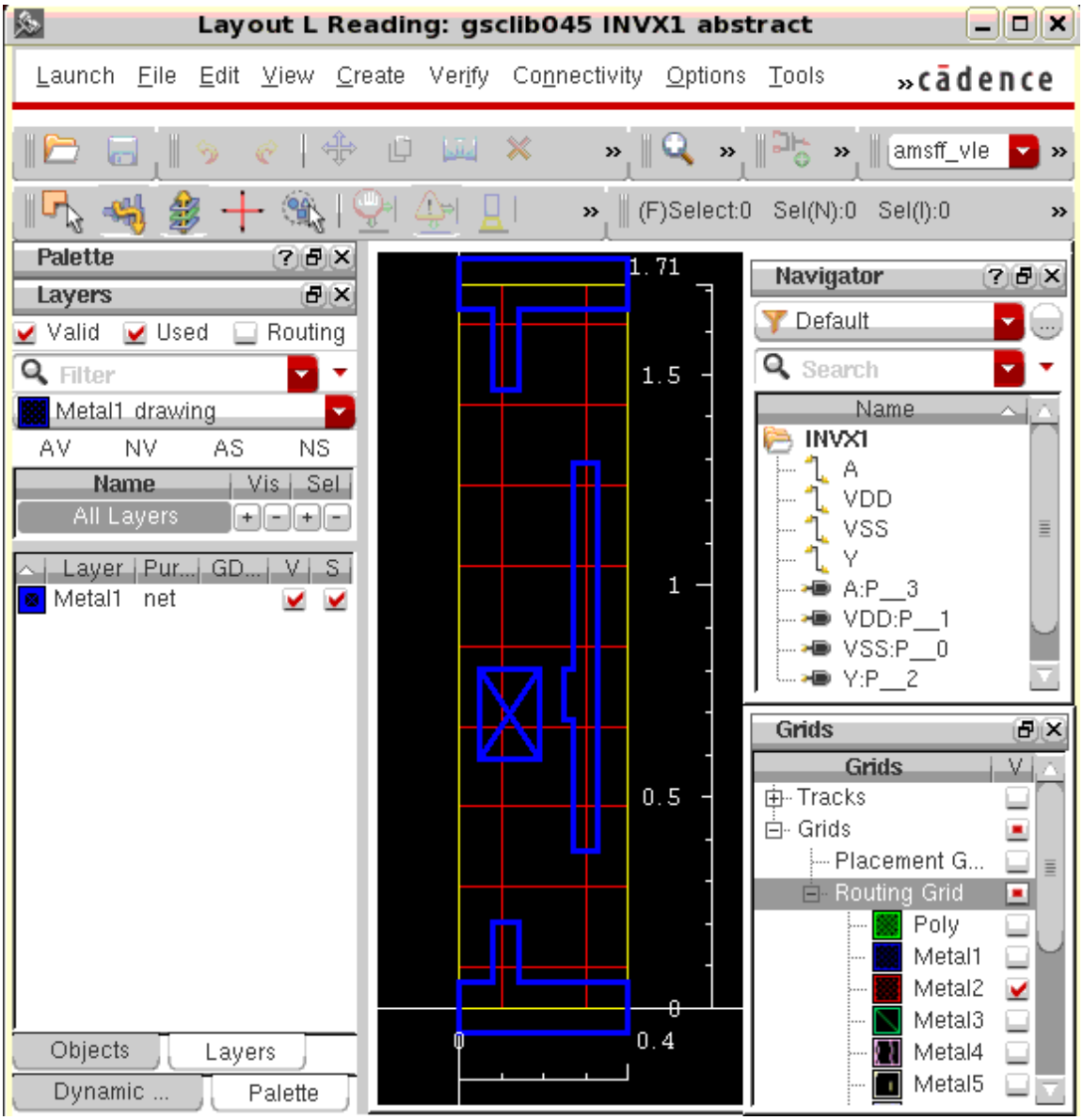


Note:

- The *layout* view is a source of *abstract* view and the GDS data.
- The GDS files for all cells in gsclib045 library are available in this directory: `<path>/gsclib045/gds`

3.1.2.3 The 'abstract' view

The *abstract* view is derived from the *layout* view. This view contains minimal data such as cell boundary box and terminal polygons for faster display in P&R tool. The *abstract* view is a source or master of the macro LEF cell. The below snapshot is the *abstract* view of *INVX1* cell.



Notes:

- The 'Metal1.net' layer is utilized in pin polygon; but it does not have GDS map entry. This means the *abstract* view must be remastered to *layout* view in the final data. Then, the 'Metal1.net' polygon will be replaced with 'Metal1.drawing' layer in the *layout* view.
- All pins must locate in Metal2 grid (red lines) for the possibility of drop via horizontally or vertically by the routing tool.
- The *abstract* view is a source of macro LEF cell which is available in this directory:
  `<path>/gsclib045/lef`
- Case of having the macro LEF file but does not have the Virtuoso 'abstract' view, user can import the macro LEF cell into Virtuoso 'abstract' view via the *Virtuoso® LEF In* utility (ref: CIW menu: File -> Import -> LEF).

3.2 Encounter Digital Data

Cadence Encounter platform provides tool suites for digital design. The digital synthesis needs logic models from standard cells; and the P&R needs the standard cell layout from standard cells. The GSCLIB045 database supplies the logical and physical building block of standard cell to the RTL synthesis tool and the P&R tools.

3.2.1 The Verilog Files (.v)

The 45nm standard cell specified its digital logic models in a Verilog file or ".v" file. The digital logic function is the first source of standard cell development; and it evolves gradually to schematic, layout, abstract, timing, and macro LEF.

In *GSCLIB045* database, the Verilog files are generated from Altos Liberate software and are available as following.

- Path: <path> /gsclib045/**verilog**
- **Files:**
  - **Slow**
    slow_vdd1v0_basicCells.v
    slow_vdd1v0_basicCells_hvt.v
    slow_vdd1v0_basicCells_lvt.v
    slow_vdd1v0_dNW0v0_basicCells_backbias.v
    slow_vdd1v0_dNW0v3_basicCells_backbias.v
    slow_vdd1v0_extvdd1v0.v
    slow_vdd1v0_extvdd1v2.v
    slow_vdd1v0_multibitsDFF.v
    slow_vdd1v2_basicCells.v
    slow_vdd1v2_basicCells_hvt.v
    slow_vdd1v2_basicCells_lvt.v
    slow_vdd1v2_dNW0v0_basicCells_backbias.v
    slow_vdd1v2_dNW0v3_basicCells_backbias.v
    slow_vdd1v2_extvdd1v0.v
    slow_vdd1v2_extvdd1v2.v
    slow_vdd1v2_multibitsDFF.v

  - **Fast**
    fast_vdd1v0_basicCells.v
    fast_vdd1v0_basicCells_hvt.v
    fast_vdd1v0_basicCells_lvt.v
    fast_vdd1v0_dNW0v0_basicCells_backbias.v
    fast_vdd1v0_dNW0v3_basicCells_backbias.v
    fast_vdd1v0_extvdd1v0.v
    fast_vdd1v0_extvdd1v2.v
    fast_vdd1v0_multibitsDFF.v
    fast_vdd1v2_basicCells.v

fast_vdd1v2_basicCells_hvt.v
fast_vdd1v2_basicCells_lvt.v
fast_vdd1v2_dNW0v0_basicCells_backbias.v
fast_vdd1v2_dNW0v3_basicCells_backbias.v
fast_vdd1v2_extvdd1v0.v
fast_vdd1v2_extvdd1v2.v
fast_vdd1v2_multibitsDFF.v

3.2.2 The Liberty Timing Files (.lib)

The liberty-timing file or ".lib" file contains the timing, delay and power parameters associated with a specific standard cell. The timing is obtained from multiple analog simulations under variety of conditions to a specific standard cell.

Cadence Encounter will use the liberty timing file (.lib) along with the Verilog model file (.v) in synthesizing the circuit based on the timing and load information of the standard cells.

The liberty timing files in *GSCLIB045* database are characterized from Altos Liberate software. The timing are based on 2x2 constraint-table-template for tool demonstration purpose (shorten characterization time and small timing file). For actual silicon process, we recommend the 7x7 constraint-table-template for high accuracy. The library timing files of *gsclib045* library are available as following.

- **Path**: <path> /gsclib045/timing
- **Files**:
  - **Slow:**
    slow_vdd1v0_basicCells.lib
    slow_vdd1v0_basicCells_hvt.lib
    slow_vdd1v0_basicCells_lvt.lib
    slow_vdd1v0_dNW0v0_basicCells_backbias.lib
    slow_vdd1v0_dNW0v3_basicCells_backbias.lib
    slow_vdd1v0_extvdd1v0.lib
    slow_vdd1v0_extvdd1v2.lib
    slow_vdd1v0_multibitsDFF.lib
    slow_vdd1v2_basicCells.lib
    slow_vdd1v2_basicCells_hvt.lib
    slow_vdd1v2_basicCells_lvt.lib
    slow_vdd1v2_dNW0v0_basicCells_backbias.lib
    slow_vdd1v2_dNW0v3_basicCells_backbias.lib
    slow_vdd1v2_extvdd1v0.lib
    slow_vdd1v2_extvdd1v2.lib
    slow_vdd1v2_multibitsDFF.lib

  - **Fast**
    fast_vdd1v0_basicCells.lib

fast_vdd1v0_basicCells_hvt.lib
fast_vdd1v0_basicCells_lvt.lib
fast_vdd1v0_dNW0v0_basicCells_backbias.lib
fast_vdd1v0_dNW0v3_basicCells_backbias.lib
fast_vdd1v0_extvdd1v0.lib
fast_vdd1v0_extvdd1v2.lib
fast_vdd1v0_multibitsDFF.lib
fast_vdd1v2_basicCells.lib
fast_vdd1v2_basicCells_hvt.lib
fast_vdd1v2_basicCells_lvt.lib
fast_vdd1v2_dNW0v0_basicCells_backbias.lib
fast_vdd1v2_dNW0v3_basicCells_backbias.lib
fast_vdd1v2_extvdd1v0.lib
fast_vdd1v2_extvdd1v2.lib
fast_vdd1v2_multibitsDFF.lib

3.2.3 The LEF Files (.lef)

In GSCLIB045 database, the library-exchange format files (.lef) are generated from the Virtuoso 'abstract' view. The macro LEF file is an ASCII data format that describes a specific standard cell. The LEF files of *gsclib045* standard cells are available as following.

- Path: <path>/gsclib045/**lef**
- Files: gsclib045_tech.lef
  gsclib045_macro.lef
  gsclib045_hvt_macro.lef
  gsclib045_lvt_macro.lef
  gsclib045_backbias_macro.lef
  gsclib045_multibitsDFF.lef

3.2.4 The GDS Files (.gds)

The graphic database system files (.gds) of *gsclib045* standard cells are translated from the Virtuoso 'layout' view and available as following.

- Path: <path>/gsclib045/**gds**
- Files: gsclib045.gds
  gsclib045_backbias.gds
  gsclib045_hvt.gds
  gsclib045_lvt.gds

3.3 Physical Verification Data

3.3.1 The CDL Netlist Files (.cdl)

The circuit design language files (CDL) of *gsclib045* standard cells are generated from the Virtuoso 'schematic' view and are available as following.

- Path: <path>/gsclib045/**cdl**
- Files: gsclib045.cdl
  gsclib045_backbias.cdl
  gsclib045_hvt.cdl
  gsclib045_lvt.cdl

3.3.2 The QRC Extraction Technology File

The GSCLIB045 database provides the QRC extraction technology file for Encounter tool as following.

- Path: <path>/gsclib045/**qrc/qx**
- Files: gpdk045.tch

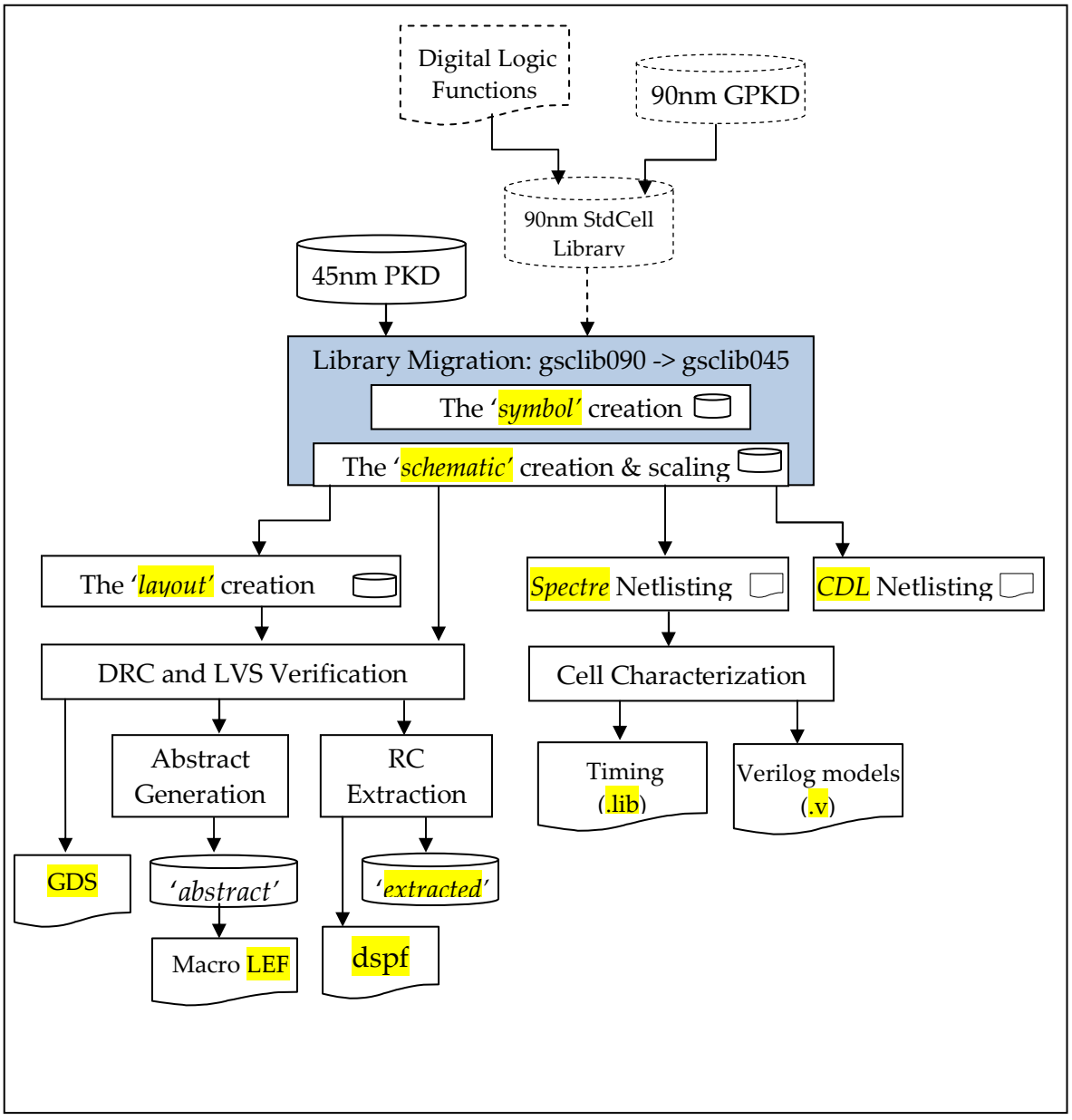
4 SUMMARY

- The GSCLIB045 database provides the 45nm generic standard cells to the mixed-signal IC design.
- For support to the Virtuoso Custom IC platform, the GSCLIB045 database provides these datasets: *symbol*, *schematic*, *functional*, *layout*, and *abstract*.
- For support to the Encounter Digital platform, the GSCLIB045 database provides these datasets: Verilog models (.v), Liberty timing (.lib), Technology and Macro LEF (.lef), Layout in GDS stream (.gds), CDL netlist (.cdl), and RC extraction techgen (.tch).

5 APPENDIXES

5.1 Standard Cells Development

The GSCLIB045 database was developed as in below diagram. The *symbol* and *schematic* are migrated from the 90nm GPDK node.



The GSCLIB045 database development had excluded the cell digital logic functions and the cell logic-to-schematic derivation. We assume that all cell schematics in 90nm node generate the accurate digital logic functions and have good well validation via its ancestor nodes such as 130nm, 180nm, 250nm, etc.

5.1.1 The GSCLIB045 common development area

We categorize the standard cell database as a design database. Therefore, we have the common development area of GSCLIB045 database inside of *Zambezi45* database as below:

<path>/WORK/zambezi45/**LPMS_WS**

We utilize the 'Makefile' program for automate the cell characterization, physical validation, and dependent data generations. We break the development tasks into two main buckets as following.

5.1.1.1 Cell Characterization Flows (Outputs: .scs, .cdl, .v, .lib)

These are top directories for cell characterization to specific standard cell library:

- **Path**: <path>/WORK/zambezi45/**LPMS_WS**
- **Directory**: basicCells_Char_flows

  basicCells_backbias_char_flows
  basicCells_hvt_char_flows
  basicCells_lvt_char_flows
  comboPad_Char_flows
  LPCells_Char_flows
  multibitsDFF_Char_flows

  Each directory above has a main 'Makefile' file; and its sub-directories for specific task have a '*Makefile*' file as below.

  Makefile
  _netlist/*Makefile*
  _netlist_scs_extracted /*Makefile*
  _netlist_cdl/*Makefile*
  _char_ldb/*Makefile*
  _lib_update/*Makefile*
  _v_update/*Makefile*
  _package/*Makefile*

5.1.1.2 Cell Implementation Flows (Outputs: abstract, .lef, .gds, extracted, .dspf)

These are top directories for cell implementation to specific standard cell library:

- **Path**: <path>/WORK/zambezi45/**LPMS_WS**
- **Directory**: basicCells_imp_flows

  basicCells_backbias_imp_flows
  basicCells_hvt_imp_flows
  basicCells_lvt_imp_flows
  LPCells_imp_flows
  multibitsDFF_hvt_imp_flows

multibitsDFF_imp_flows
multibitsDFF_lvt_imp_flows

Each directory above has a main 'Makefile' file; and its sub-directories for specific task has a '*Makefile*' file as below.

Makefile
_drcLVS/*Makefile*
_gds/*Makefile*
_abstract/*Makefile*
_macroLEF/*Makefile*
_dspf _extract/*Makefile*

5.1.2 Library Migration (Symbol and Schematic): gsclib090 -> gsclib045

We utilized VLM tool to migrate 'gsclib090' cells to 'gsclib045' cells via programs in this directory:

<path>/gsclib045/**migration**

The top program is '*runmigrate.csh*' that calls to a SKILL replay file '*migrateCells.il*'. The result is the Virtuoso *gsclib045* library with 481 cells that have *symbol*, *schematic*, and *layout* views. The MOSFET device width (w) in schematics is scaled down by a multiply factor of '0.6' via the '*migratemap_sch.il*' file. For more detail of *gsclib045*- to-*gsclib045* VLM migration, please consult with this document.

<path>/gsclib045/migration/*gpdk90to45StandardCellMigration.pdf*

We added the "LPCells" category (71 cells) and "MultiBits" category (16 cells) into *gsclib045* library later on (without VLM migration).

5.1.3 Physical Implementation – layout

The *gsclib045* library provides two layout types as following.

- Flatten and polygon base layout
  The polygon base layout or flatten layout is without using pCells. There is no device instance binding directly between the *schematic* and *layout* view. There are 481 out of 567 cells in *gsclib045* library (exclude cells in *LPCells* and *MultiBits* category) with this layout flavor. These 481 cells were migrated via VLM tool; and the *gpdk90to45StandardCellMigration.pdf* document provides more detail about the layout migration setup and steps.

- Hierarchical and pCell base layout
  The seventy-one *LP* cells are hierarchical pCell base layout; and sixteen *MultiBits* cells are hierarchical layout. In the hierarchical cells, the *schematic* and *layout* view have bound device instances via the Virtuoso Layout XL.

5.1.4 The Netlist Generation from 'schematic' view - CDL and Spectre Netlist

We use the hierarchical 'Makefile' programs in the *Cell Characterization Flows*. We can generate all of CDL and Spectre netlist from the top *Makefile* program or individually from a specific *Makefile* program. For instance of *gsclib045* cells:

All: <path>/WORK/zambezi45/LPMS_WS/ basicCells_Char_flows/Makefile

Spectre : <path>/WORK/zambezi45/LPMS_WS/ basicCells_Char_flows/_netlist/Makefile

CDL: <path>/WORK/zambezi45/LPMS_WS/basicCells_Char_flows/_cdl_netlist/Makefile

5.1.5 The Cell Characterization - Liberty Timing and Verilog

We use the hierarchical 'Makefile' programs in *Cell Characterization Flows*. We can run all sub-flows from the top *Makefile* program or run individually sub-flow from a specific *Makefile* program. For instance of *gsclib045* cells:

All: <path>/WORK/zambezi45/LPMS_WS/ basicCells_Char_flows/Makefile

Char : <path>/WORK/zambezi45/LPMS_WS/ basicCells_Char_flows/_char_ldb/Makefile

Timing: <path>/WORK/zambezi45/LPMS_WS/ basicCells_Char_flows/_lib_update/Makefile

Verilog: <path>/WORK/zambezi45/LPMS_WS/basicCells_Char_flows/_v_update/Makefile

5.1.6 The Physical Verification and Extraction – DRC/LVS/EXT

We use the hierarchical 'Makefile' programs in *Cell Implementation Flows*. We can run all sub-flows from the top *Makefile* program or run individually from a specific *Makefile* program. For instance of *gsclib045* cells:

All: <path>/WORK/zambezi45/LPMS_WS/ basicCells_imp_flows/Makefile

DRC/LVS : <path>/WORK/zambezi45/LPMS_WS/ basicCells_imp_flows/_drcLVS/Makefile

Extraction: <path>/WORK/zambezi45/LPMS_WS/basicCells_imp_flows/_extract/Makefile

5.1.7 The Cell Abstract Creation - 'abstract' view and Macro LEF

The cell abstract and macro LEF generation are parts of hierarchical 'Makefile' programs in *Cell Implementation Flows*. We can run all sub-flows from the top *Makefile* program or run individually from a specific *Makefile* program. For instance of *gsclib045* cells:

All: <path>/WORK/zambezi45/LPMS_WS/ basicCells_imp_flows/Makefile

Abstract: <path>/WORK/zambezi45/LPMS_WS/ basicCells_imp_flows/_abstract/Makefile

LEF: <path>/WORK/zambezi45/LPMS_WS/basicCells_imp_flows/_macroLEF/Makefile

5.2 Cadence 45nm Generic Process Design Kit – GPDK045

The 45nm generic standard cell database, GSCLIB045, is associated to 45nm generic PDK called 'GPDK045'. The GPDK045 provides the 'gpdk045' technology library with symbols and pCells, Spectre models, PVS DRC and LVS rule file, and QRC extraction rule file.

The GSCLIB045 database does not bundle the associated GPDK045 database. User should download the GPDK045 database separately and put it into the IC design environment as in '*2.3 Recommended Location in IC Design Database*' section.