P D K Training

Calibre user guide

Y.Liu

2006-01-05

- Introduction
- Invocation
- Calibre DRC usage
- Calibre lvs usage

Introduction

- What can Calibre do
- Calibre Feature
- Calibre Utility

What can Calibre do

The Calibre Verification applications operate on rule files written in Standard Verification Rule Format (SVRF).

- DRC -- Design Rule Check
- ERC -- Electronic rule Check
- LVS -- Layout Versus Schematic
- XRC -- parasitic extraction
- OPC -- Optic Proximity Correction
- EB Fracture
- Utility something else

Calibre Feature

- **Calibre DRC / DRC-H / MT DRC-H**
  - **Flat** — Calibre DRC performs design rule checking by reading the input layout database flat and operating on the geometry.
  - **Hierarchical** — Calibre DRC-H performs design rule checking hierarchically, which minimizes redundant processing. It stores, analyzes, and processes data once per cell instead of once for every flat placement of the cell.
  - **Multi-threaded** — This configuration of Calibre DRC-H allows you to take advantage of processing with multiple CPUs.
- **Calibre LVS / LVS-H / MT LVS-H**
  - **Flat**—Calibre LVS performs flat layout versus schematic netlist checking.
  - **Hierarchical**—Calibre LVS-H performs hierarchical layout versus schematic netlist checking. Like Calibre DRC-H, it also stores, analyzes,and processes data once per cell instead of once for every flat placement of the cell.
  - **Multi-threaded** — This configuration of Calibre LVS-H allows you to take advantage of multiple CPUs

Calibre Feature (cont.)

- ✓ **Calibre RVE，Query Server**
  - **Calibre RVE and Query Server** are a licensed graphical user interface and server that allow you to investigate, debug, and highlight design errors.
- ✓ **Calibre XRC**
  - **Calibre XRC** performs layout parasitic extaction
- ✓ **Calibre Interactive**
  - **Calibre Interactive** is a licensed user interface environment for Calibre. You can invoke it from the command line or through layout editors. It is used to set up and execute DRC and LVS runs. It can also invoke Calibre RVE. Calibre Interactive works with the same layout editors as RVE.
- ✓ **Calibre Connectivity Interface**
  - The **Calibre Connectivity Interface (CCI)** is a set of licensed functionality associated with the Query Server. Calibre LVS SVDB results database into standards-based file formats (GDSII and SPICE), which can be used by downstream flows that need to access LVS extraction and comparison results (for example, backannotated netlisting)

Calibre Utility

- **EDIF-to-LVS**
  - EDIF-to-LVS (E2LVS) is a converter that translates an EDIF structural netlist into a SPICE-like netlist for use as input to Calibre LVS/LVS-H.
- **Verilog-to-LVS**
  - Verilog-to-LVS (V2LVS) is a converter that translates a Verilog structural netlist into a SPICE-like netlist for use as input to Calibre LVS/LVS-H.
- **Dracula to SVRF converter**
  - The Dracula converter allows you to convert a Dracula rule file into a Standard Verification Rule Format rule file.
- **Compare Two GDSII Database**
  - The Compare GDS utility allows you to compare two GDSII databases (flat). This utility produces an ASCII DRC results database based on a layer-by-layer analysis.
- **Rules Syntax Checker**
  - The Rules Syntax Checker utility allows you to check the syntax of a rule file without actually compiling it for a Calibre run.

Invocation

- Environment Setup
- Adding Interface into cadence virtuoso
- Necessary Condition

Environment Setup

1. In a C shell window, enter:

   % setenv MGC_HOME INSTALL_DIR

   % set path = ($path $MGC_HOME)

2. Verify the Environment variable

   % echo $MGC_HOME

3. Starting Calibre

   % calibre [options]

Adding Interface into cadence virtuoso

In order to set up your Virtuoso layout editor with a Calibre pulldown menu, we must be add the following line to the default .cdsinit file , loads Skill functions that enable Calibre within the Cadence environment.

```
load( strcat( getShellEnvVar("MGC_HOME")
      "/shared/pkgs/icv/tools/queryskl/calibre.skl" ))
```

Necessary Condition

Before you invoke a Calibre Verification tool, the following data must exist:

- Rule file
  - DRC rule
  - LVS & ERC rule
  - XRC rule
- Layout database

System Format	DRC	DRC-H	LVS	LVS-H	MGC
CIF	X	X	X	X	X
GDSII	X	X	X	X	X
SPICE			X	X	X
ASCII	X		X		X
Binary	X		X		X
CNET database			X		X
V8.x EDDM					X

Necessary Condition (cont.)

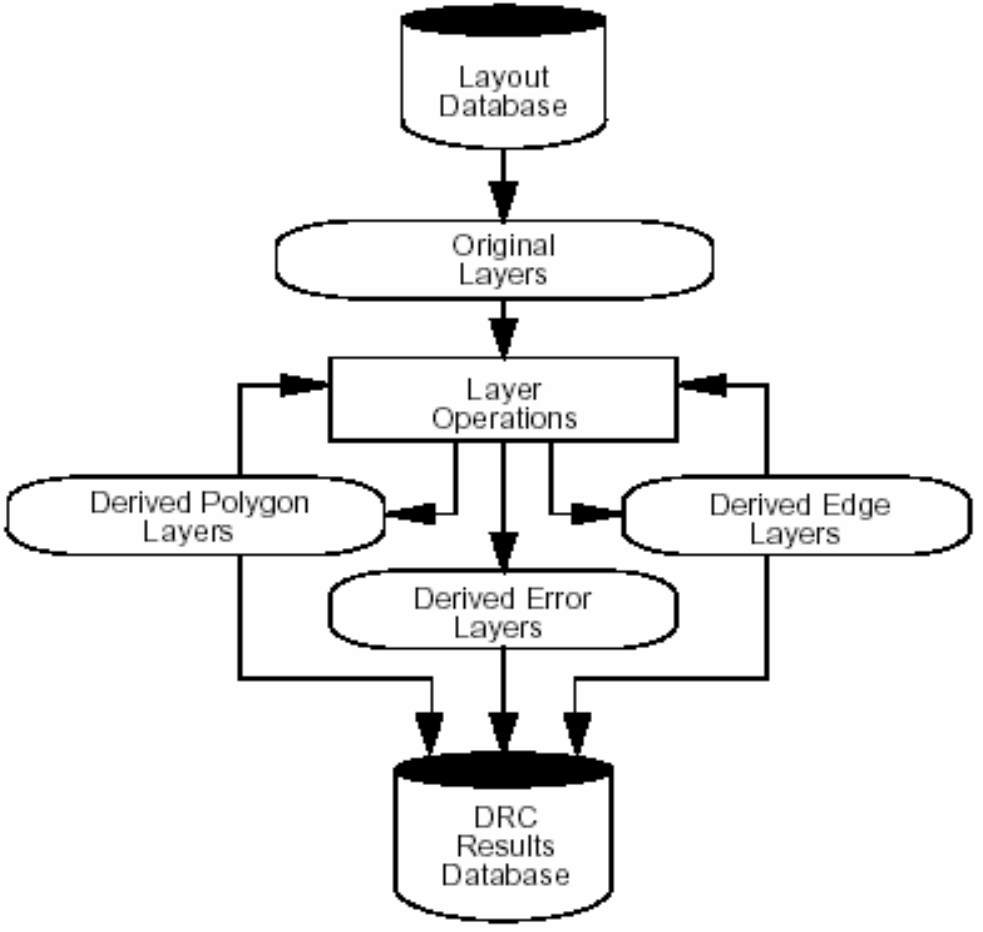
- Source database, as applicable

System Format	LVS	LVS-H	MGC
SPICE	X	X	X
CNET database	X		X
V8.x EDDM			X

Calibre DRC usage

- Operation in the DRC system
- Input and Output
- Calibre DRC command line
- Frequently used command

Operation in the DRC System



Operation in the DRC System (cont.)

- Original layer
  - Original layers (or drawn layers) are layers that represent original layout data

    LAYER M1 2 // simple layer
- Derived Polygon Layers
  - Derived polygon layers represent merged polygons generated as the output of layer operations such as Boolean functions, polygon-directed functions, and certain dimensional check operations.

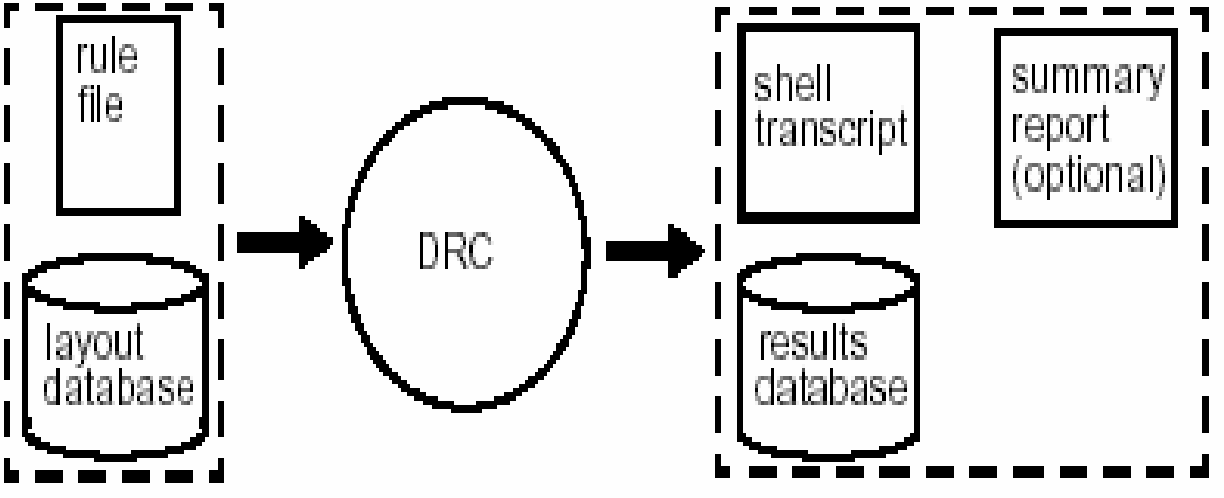
    GATE = POLY AND DIFF
- Derived Edge Layers
  - Derived edge layers represent edges or edge segments of merged polygons generated as the output of layer operations such as topological edge operations and edge-directed dimensional check operations

    long_metal_edge = length metal > 5
- Derived Error Layers
  - Derived error layers represent clusters of one, two, three, or four edges.

    rule check { copy M1 }

Input and Output

- Data flow chart



Input and Output (cont.)

- Specify the input in rule file
  - Layout System
    - Specifies the format of the layout data
  - Layout Path
    - Specifies the location of the layout data
  - Layout Primary
    - Specifies the top-level cell within the layout data

Example:

```
LAYOUT SYSTEM      GDSII
LAYOUT PATH        "GDSII_FILE"
LAYOUT PRIMARY     "TOPCELL"
```

Input and Output (cont.)

- Specify the output in rule file
  - DRC Results Database
    - Specifies where to save the results
    - The database for graphic debug including error pattern informaiton
  - DRC SUMMARY REPORT (optional)
    - Specifies the DRC summary report filename
    - The error summary information in text file

Example:

```
DRC RESULTS DATABASE     "DRC.db"
DRC SUMMARY REPORT       "DRC.rep"
```

Calibre DRC command

- Flat DRC

  calibre -drc rulefile > drc.log

- Hierarchical DRC

  calibre –drc –hier rulefile > drc.log

- Multi cpu

  calibre –drc -turbo 4 rulefile > drc.log

  calibre –drc –hier -turbo 4 rulefile > drc.log

Note:

- These commands are used frequently
- drc.log is the former shell transcript specified by yourself
- The number of cpu can be specified and in the former case it is given 4.
- Multi-cpu drc need multi-licenses.

Frequently used command

- Rule Check Selection
  - **DRC SELECT CHECK**

    Calibre DRC selects only those rule checks specified in DRC Select Check specification statements
  - **DRC UNSELECT CHECK**

    Calibre DRC does not select any rule checks specified with DRC Unselect Check specification statements in the rule file

Example: Adding the following command into rule file to control the drc check

```
DRC SELECT CHECK      contact
DRC UNSELECT CHECK contact
```

rulefile :

```
……
contact{ @ contact 0.4 X 0.4
          rectangle ==0.4 by == 0.4
           }
……
```

Frequently used command(cont.)

- Cell Exclusion
  - Exclude Cell

    Specifies one or more cells to exclude from the layout database—that is, layout cells that are not to be processed by the verification application. Excluding a cell means that no objects from any placement of the cell are processed. This includes all hierarchy within the placement

Example: Adding the follow command into rule file

Exclude Cell "FLASH" "SRAM"

cell name in gdsII

Frequently used command(cont.)

- Handling Duplicate Cells
  - Layout Allow Duplicate Cell

    Specifies whether multiple GDS records for the same layout cell are allowed for the input layout database.By default, all records after the first are discarded with a warning or error message. If YES is specified, then multiple cell records are treated as if the data were combined into a single record with no warning or error message.This can be useful when the database is split into multiple files by layer.

Example: Adding the follow command into rule file

LAYOUT ALLOW DUPLICATE CELL NO → default

LAYOUT ALLOW DUPLICATE CELL YES

Frequently used command(cont.)

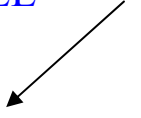
- DRC Use of Hcells
  - HCELL layout_name source_name

    Specifies the names of a pair of cells that correspond in layout and source.DRC-H also inhibits expansion of any layout cell in a rule file Hcell specification statement.

Example: Adding the follow command into rule file

HCELL “inv1” “inv1x”

HCELL “*” “*”



Wildcard character

Frequently used command(cont.)

- Define maximun output error number
  - **DRC MAXIMUM RESULTS {*maxresults* | ALL}**

    Specifies the DRC RuleCheck maximum result count for DRC. You can only specify this statement once in a rule file

Example: Modified the follow command into rule file

DRC MAXIMUN RESULTS **1000**

This is default value in rule file

NOTE: If you want output all error , please use **ALL**

DRC MAXIMUN RESULTS **ALL**

Frequently used command(cont.)

- Define maximun vertex of output shapes
  - **DRC MAXIMUM VERTEX {numbers | ALL}**

    Specifies the maximum vertex count of any polygon DRC result to be written to the DRC Results Database (ASCII or binary) by Calibre DRC/DRC-H

Example: Modified the follow command into rule file

DRC MAXIMUN VERTEX **199**

This is default value in dummy rule file

NOTE: In order to be compatible to old version GDSII ,

you had better designating 199**.**

Frequently used command(cont.)

- output error shapes to gds
  - **DRC CHECK MAP *rule_check***

    This statement is used to control the database output structure for DRC RuleChecks

Example: Added the follow command into rule file

DRC CHECK MAP **cont_size**

Pay attention to relation to following command

DRC MAXIMUN RESULTS 1000

DRC MAXIMUN VERTEX 199

DRC Results Database "drc.db" GDSII

Frequently used command(cont.)

- output error shapes to gds (cont.)

Case 1 DRC Results Database "drc.db" GDSII

DRC CHECK MAP **cont_size** GDSII

---- output all rule check error shapes((1000 errors) into layer 0 of drc.db(gds format) file

Case 2 DRC CHECK MAP **cont_size** GDSII 1 2 new.gds

DRC Results Database "drc.db" GDSII

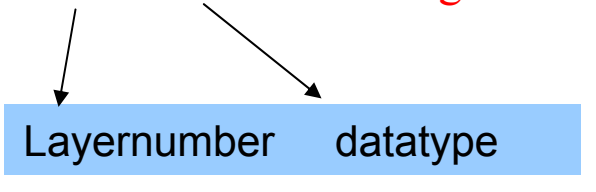
---- output cont_size rule check(1000 errors) to "new.gds" and others to "drc.db"

Case 3 DRC CHECK MAP **cont_size** GDSII 1 2 new.gds

DRC MAXIMUN RESULTS ALL

DRC MAXIMUN VERTEX 199

DRC Results Database "drc.db" GDSII

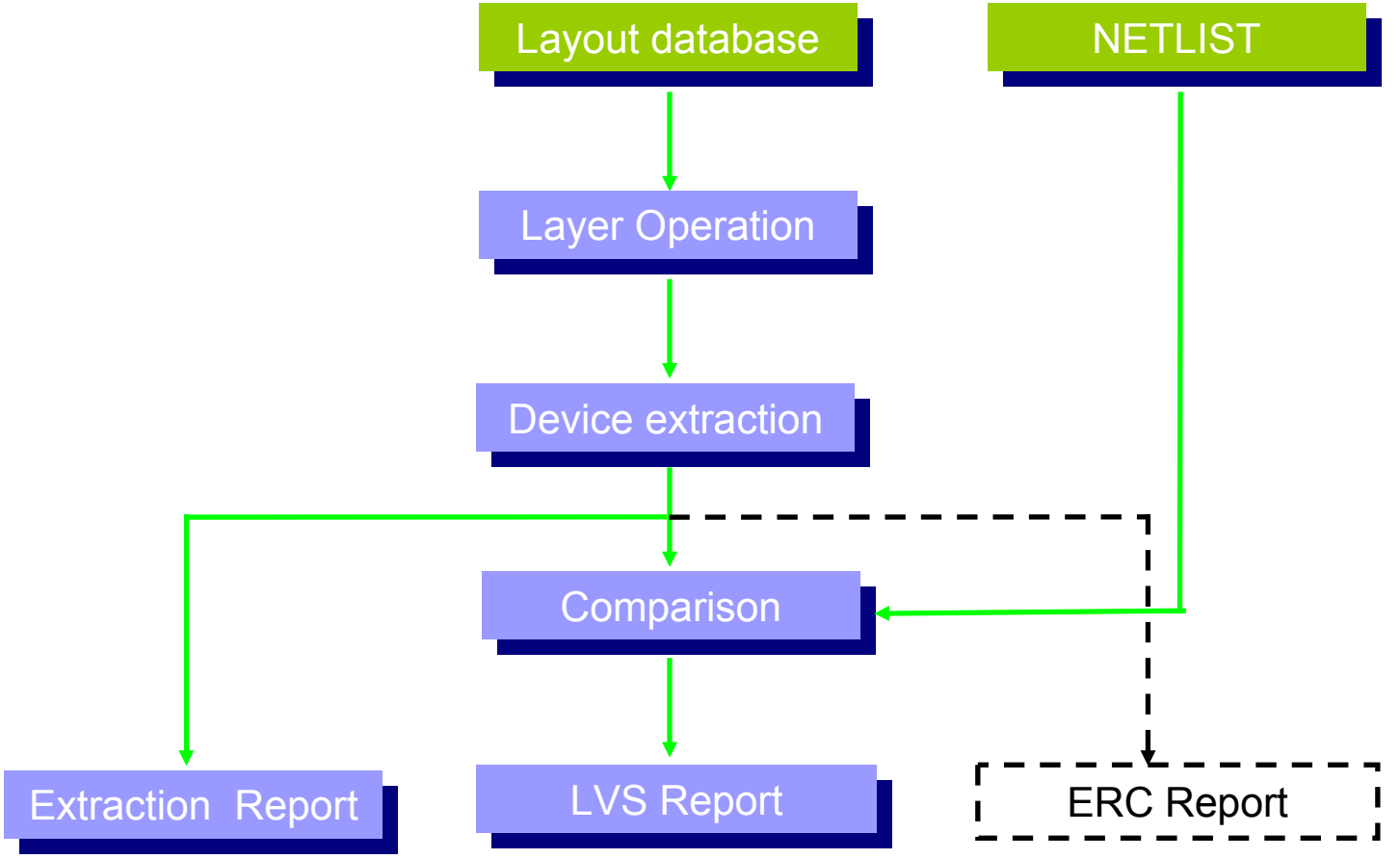


---- output cont_size rule check(all errors) to "new.gds" and others to "drc.db"

Calibre LVS usage

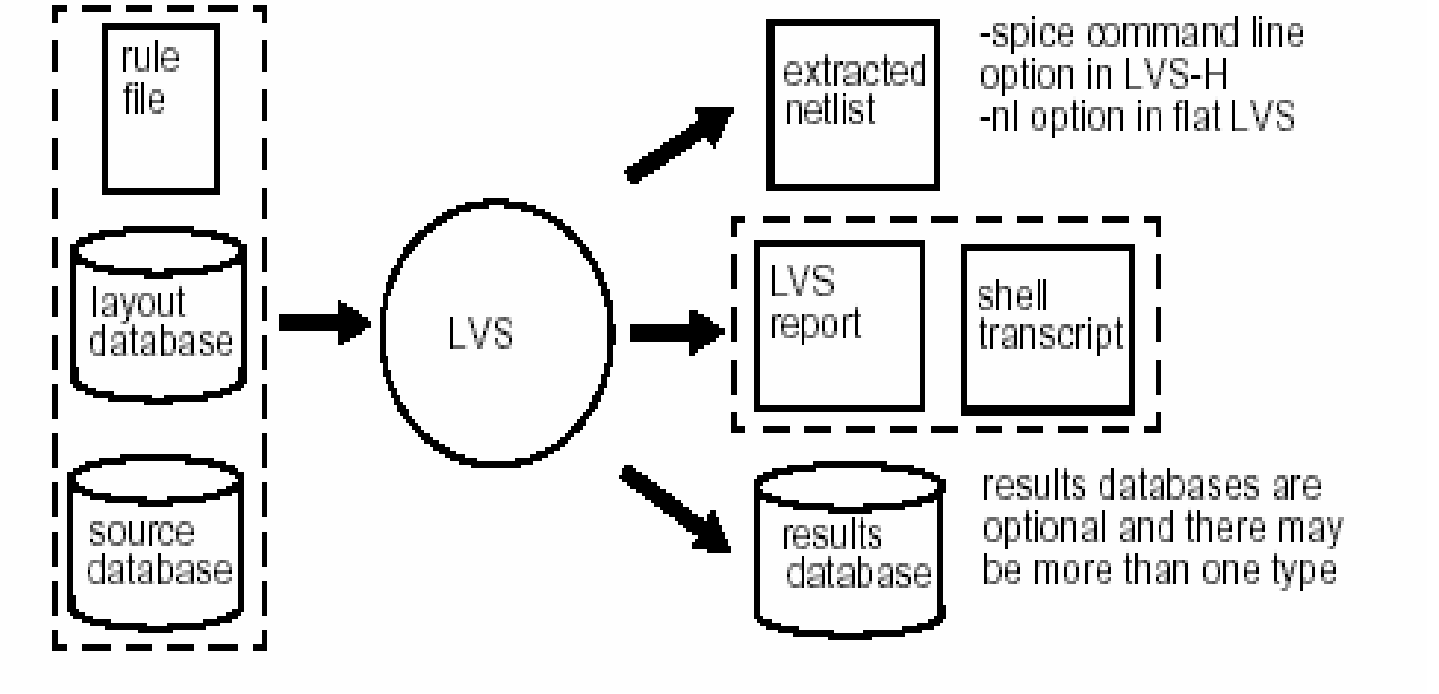
- Operation in the LVS system
- Input and Output
- Calibre LVS command line
- Frequently used command

Operation in the LVS System



Input and Output

- Data flow chart



Input and Output (cont.)

- Specify the input in rule file
  - Layout System
    - Specifies the format of the layout data
  - Layout Path
    - Specifies the location of the layout data
  - Layout Primary
    - Specifies the top-level cell within the layout data

Example:

```
LAYOUT SYSTEM        GDSII
LAYOUT PATH          "GDSII_FILE"
LAYOUT PRIMARY       "TOPCELL"
```

SVS： LAYOUT SYSTEM must be SPICE

LAYOUT PATH must be netlist file

Input and Output (cont.)

- Specify the input in rule file
  - SOURCE System
    - Specifies the format of the netlist
  - SOURCE Path
    - Specifies the location of the netlist
  - SOURCE Primary
    - Specifies the top-level cell within the netlist

Example:

```
SOURCE SYSTEM        SPICE
SOURCE PATH          “netlist"
SOURCE PRIMARY       "TOPCELL"
```

Input and Output (cont.)

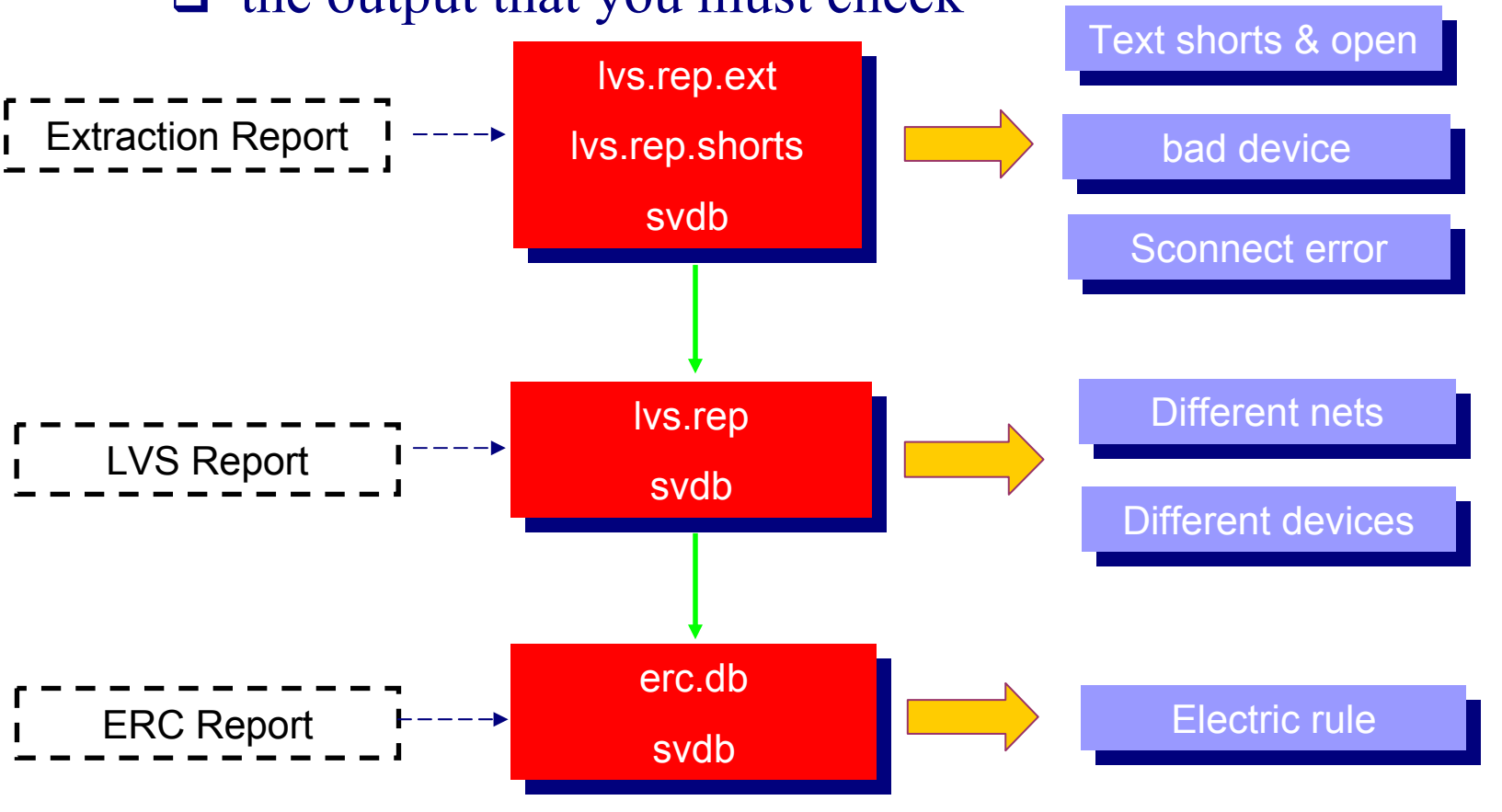
- Specify the output in rule file
  - **LVS REPORT**
    - Specifies the LVS report filename
  - **MASK SVDB DIRECTORY "svdb“** (optional)
    - Specifies the types of files generated in the standard verification database (SVDB) by LVS and PEX applications.
    - The database for graphic debug including error pattern informaiton(default for RVE)

Example:

```
LVS REPORT      “lvs.rep"
MASK SVDB DIRECTORY “svdb”
```

Input and Output (cont.)

- the output that you must check



Calibre LVS command

- Flat lvs

% **calibre –lvs rulefile | tee log**

- Hierarchical LVS
  - -spice option

    extracts a hierarchical SPICE netlist from the layout system

% **calibre –spice layout.spi rulefile | tee log**

  - -lvs option

    specifies to run Calibre LVS

% **calibre –lvs –hier –spice layout.spi rulefile | tee log**

  - -auto option

    Calibre compares cells with the *same name* in the layout and source include those specified by the -hcell option or in an Hcell rule file specification statement, if specified) as hierarchical entities

% **calibre –lvs –hier –spice layout.spi –auto rulefile | tee log**

Calibre LVS command(Cont.)

- -hcell option

  specifies a cell correspondence file for hierarchical LVS comparison

% calibre –lvs –hier –spice layout.spi -auto –hcell hcellfile rulefile | tee log

- -turbo option

  instructs Calibre LVS-H to use multi-threaded parallel processing

% calibre –lvs rulefile –turbo integer | tee log

% calibre –lvs –hier –spice layout.spi –turbo integer rulefile | tee log

❖ SVS

% calibre –lvs –hier rulefile | tee log

% calibre –lvs –hier –auto rulefile | tee log

% calbire –lvs –hier -auto –hcell hcellfile rulefile | tee log

Frequently used command

TEXT

- Define text layer
  - **PORT LAYER TEXT layer [...layer]**
    - Define the port in the layout
  - **TEXT LAYER layer [...layer]**
    - Specifies the layers in the database from which text is read for connectivity extraction. EXAMPLE.
  - **ATTACH layer1 layer2**
    - Attach connectivity information from a ***layer1*** object onto a ***layer2*** object

Example:

```
PORT  TEXT  LAYER     121   122
TEXT LAYER            121 122
ATTACH                121  ME1
ATTACH                122  ME2
```

Frequently used command(cont.)

TEXT (cont.)

- Text Specification
  - ✓ Labeling Text in layout
  - ✓ Including text file into rule file

LAYOUT TEXT name x y layer [texttype] cell_name

The text object ***name*** behaves exactly as if it were in the layout database in ***cell_name*** at coordinates ***x y*** in the cell coordinate space on ***layer***

EXAMPLE:

LAYOUT TEXT VDD 200 300 121 INV

You can write all text in a file and include into rule file!!!

TEXT (cont.)

- Reading text
  - **TEXT DEPTH [PRIMARY | ALL | number]**
    - Specifies hierarchical depth for selecting text objects from the layout database to be used as top-level text
    - The default is PRIMARY if you do not include this statement in the rule file
  - **PORT DEPTH [PRIMARY | ALL | number]**
    - Specifies hierarchical depth for reading port objects from the layout database for use in the top-level cell
    - The default is PRIMARY if you do not include this statement in the rule file

Example:

```
PORT DEPTH   PRIMARY
TEXT DEPTH   PRIMARY
```

PORT CHECK

- **LVS CHECK PORT NAMES** [**NO** | **YES**]
  - Specifies whether the tool checks the names of matched ports.
  - When you specify YES, the tool verifies that (in the top-level cell) the layout port name matches the corresponding source port name
  - NO is the default behavior if you do not include this statement
  - The matched layout port and source port are unnamed, such as not having user-given names
- **LVS IGNORE PORTS** [**NO** | **YES**]
  - Specifies whether the LVS comparison algorithm should ignore layout and source pins

Example: in our rule file

LVS CHECK PORT NAMES YES

→ It is necessary for the circuit with multi similar structure , Analog IP especially

LVS IGNORE PORTS NO

→ It is necessary for hcell IP comparison

Error: Different numbers of ports

BLOCK CELL

- LVS BOX [SOURCE LAYOUT | SOURCE | LAYOUT] cell
  - Specifies cells with contents to be ignored in LVS/LVS-H circuit comparison, default is SOURCELAYOUT.
- HCELL layout_name source_name
  - Specifies the names of a pair of cells that correspond in layout and source

Example:

```
LVS BOX   IP1 IP2 IP3
HCELL      IP1   BGR
HCELL      IP2   ADC
```

You can write hcell into a file , and use it with calibre lvs command

Example : A hcell file including the following

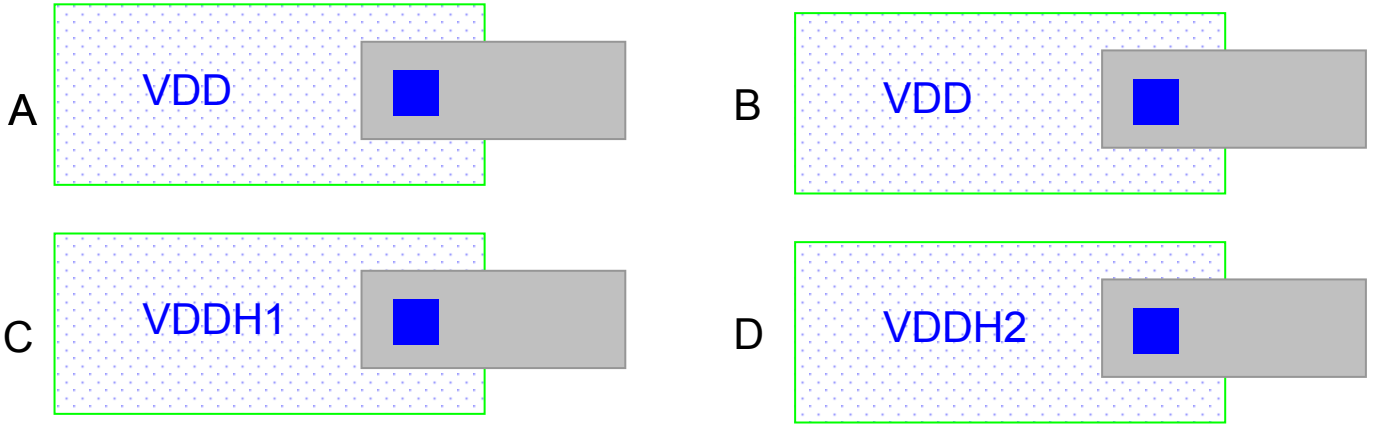
```
IP1     BGR
IP2     ADC
…       …
```

Virtual connect statement

- Virtual connect Name name
  - Specifies virtual connections for the specified net names

Example:

Virtual connect Name VDD VDDH?



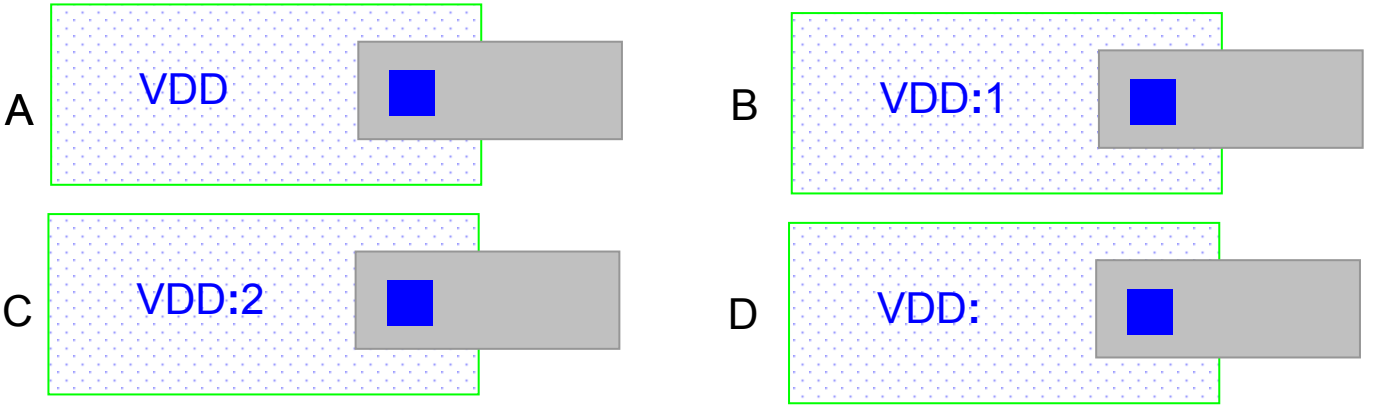
A and B are connected
C and D are connected
A and D are not connected
B and C are not connected

Virtual connect statement (cont.)

- Virtual connect colon [**NO** | **YES**]
  - create virtual connections for net names containing a colon character (**:**)
  - The default is NO

Example:

Virtual connect Name YES



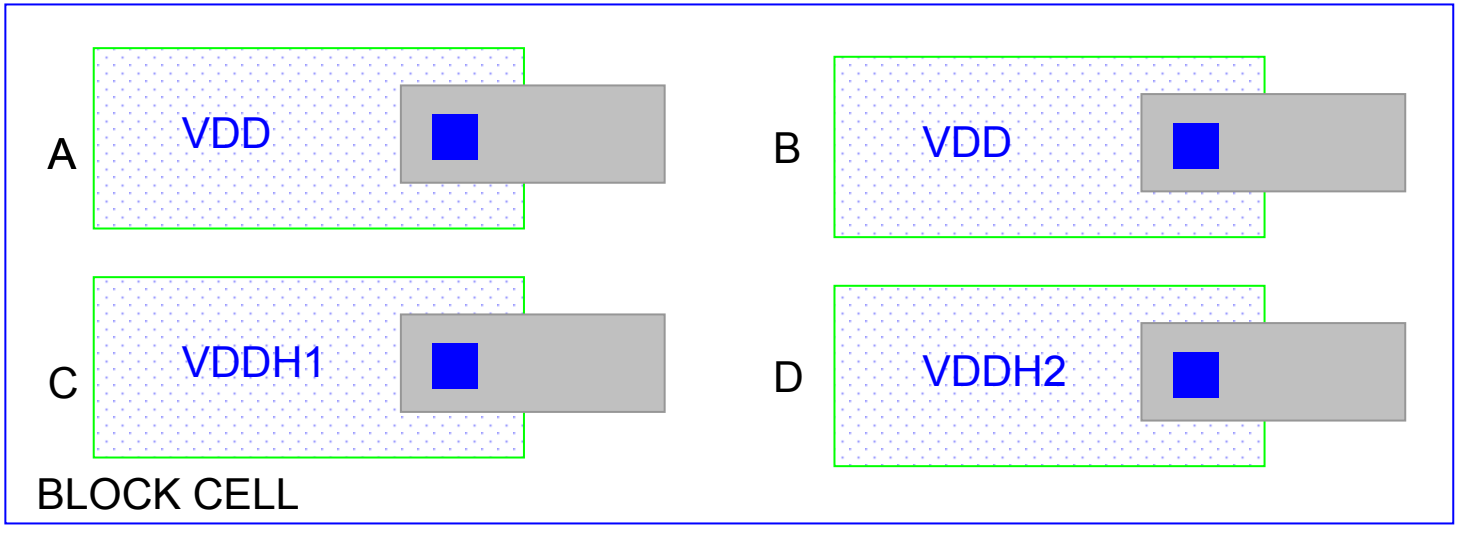
A , B ,C ,D are connected

Virtual connect statement (cont.)

- Virtual connect Box Name [**NO** | **YES**]
  - Specifies virtual connections for net names in box cells

Example:

Virtual connect Box Name VDD VDDH?



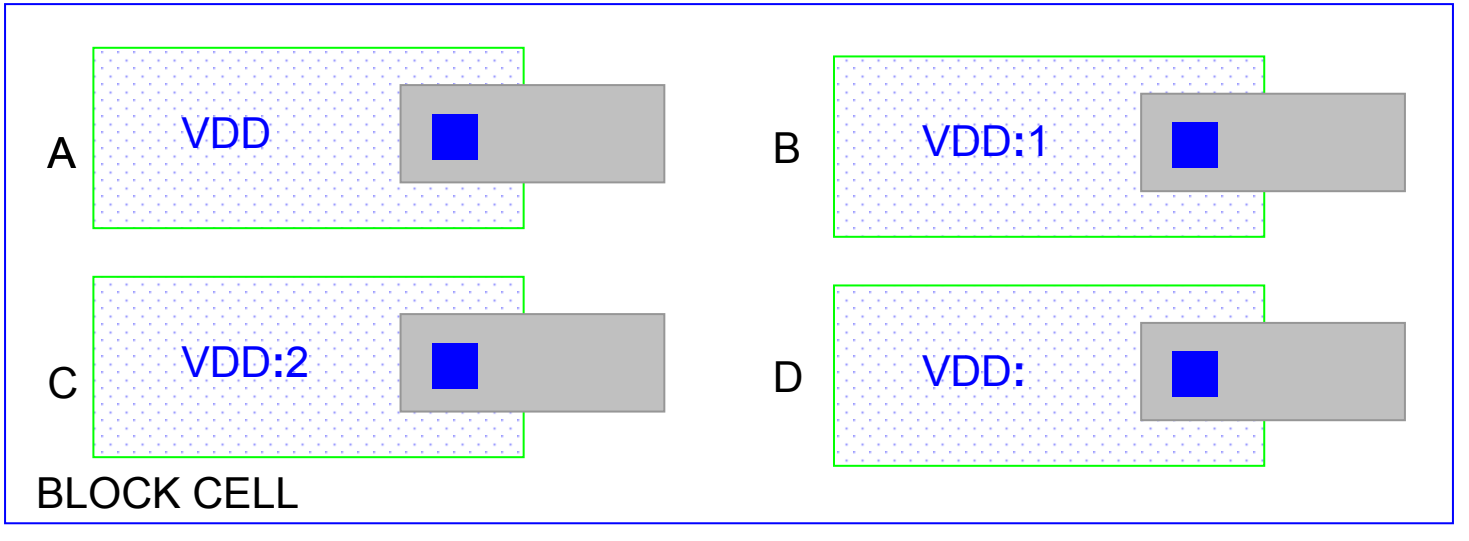
A ,B,C,D are connected

Virtual connect statement (cont.)

- Virtual connect Box Colon name
  - Specifies virtual connections for net names containing the colon character (:) in box cells
  - The default is NO

Example:

Virtual connect Box Colon YES



A and B are connected

Power & Ground Name

- LVS POWER NAME
- LVS GROUD NAME

Specifies a list of zero or more power and ground net names.Power and ground net names are used by LVS in ***logic gate recognition, in filtering of unused MOS transistors***, and ***ERC check***

For example: in your rule file

LVS POWER NAME "VDD" "VCC" "VDD5"
LVS POWER NAME "?VDD" "VDD?"
LVS GROUD NAME "VSS" "VSSPRT" "GND"
LVS GROUD NAME "VSS?"

Logic Gate Recognition

- LVS RECOGNIZE GATES {ALL | SIMPLE | NONE}

Specifies whether to recognize logic gates from transistor-level data ,Logic gate recognition allows you to swap the order of logically equivalent pins and devices in transistor level implementations of logic circuits. The default is ALL , if you do not include this statement in the rule file.

ALL—Specifies that all gates are recognized.

SIMPLE—Specifies that simple gates are recognized, prevents the formation of complete AOI and OAI gates and higher level series-parallel structures

NONE— Specifies that no gates are recognized

For example: in your rule file

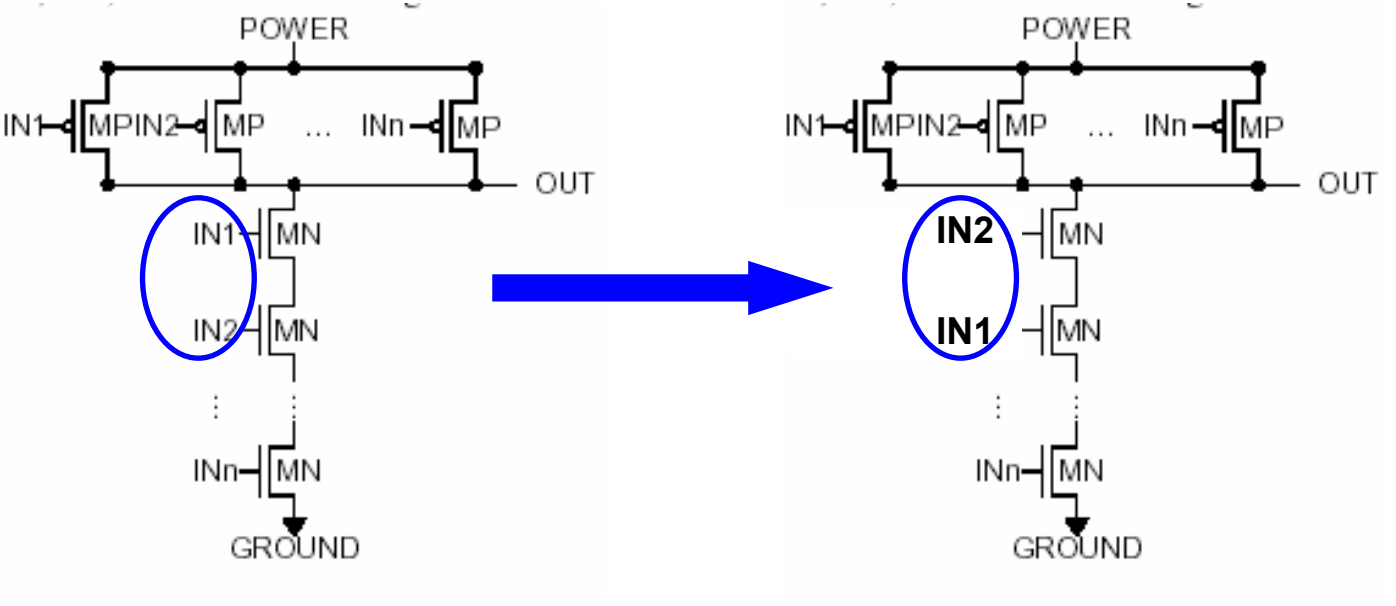
LVS RECOGNIZE GATES ALL

you must specify the power name by lvs power name command and ground name by lvs ground name command

Logic Gate Recognition (cont.)

LVS RECOGNIZE GATES SIMPLE

NANDn



LVS can match

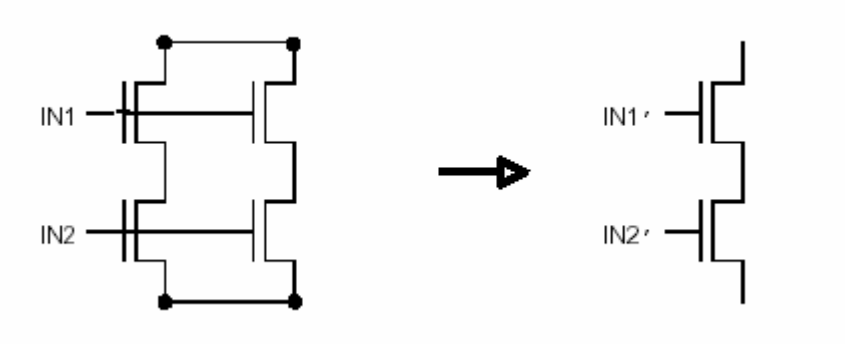
Device Reduction

- LVS REDUCE SPLIT GATES {YES | NO}
  [SEMI ALSO] [SAME ORDER] [WITHIN TOLERANCE]
  [
  [TOLERANCE {*property_name tolerance_number*}
  [*effective_property_computation*]
  ]

The default is YES , if you do not include this statement in the rule file.

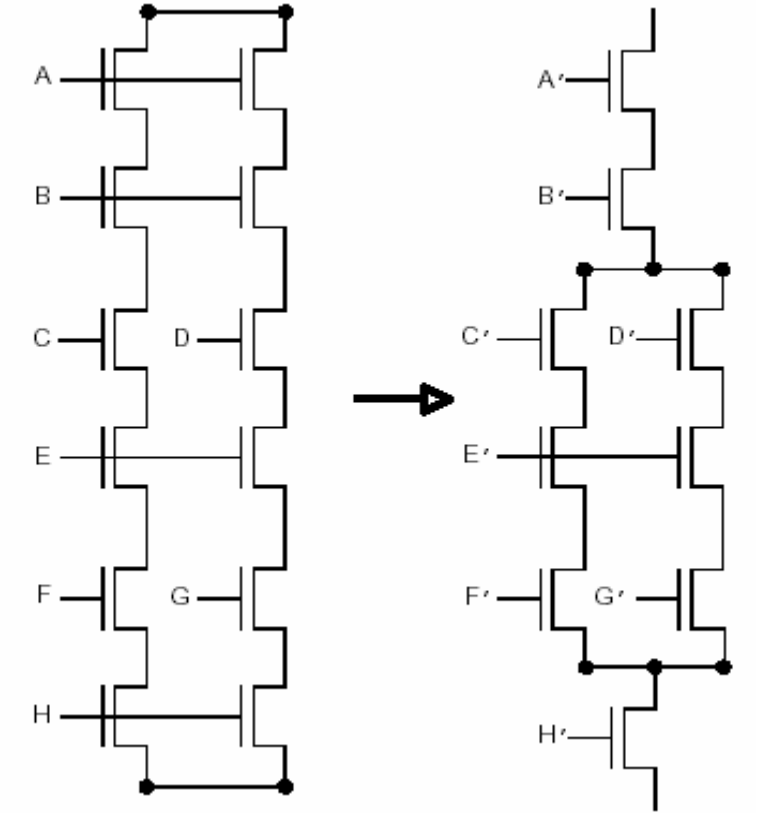
Example:

```
LVS REDUCE  SPLIT GATES YES [TOLERANCE w 1]
    [ effective M, W, L
      M = sum(M)
      p = sum (W * L)
      q = sum (W / L)
      W = sqrt (p * q)
       L = sqrt (p / q)
      ]
```

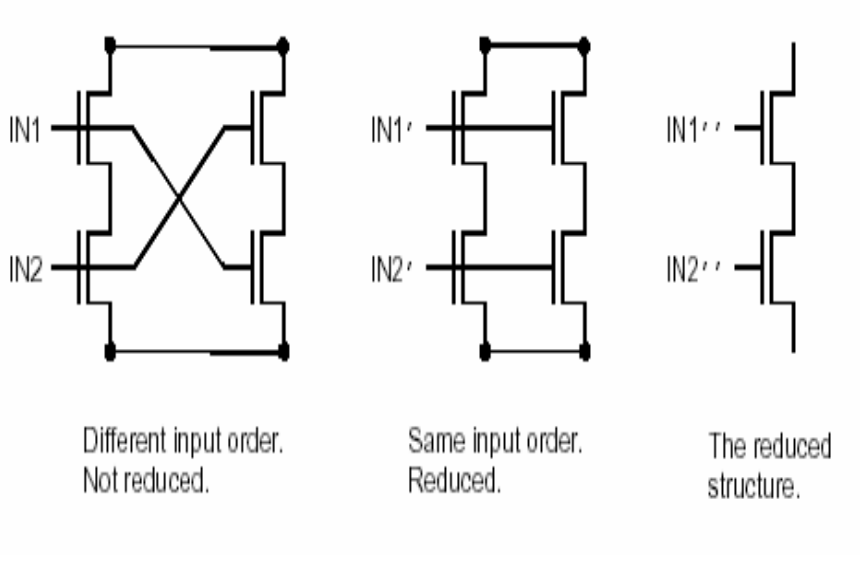


Device Reduction (cont.)

SEMI ALSO



SAME ORDER



Device Reduction (cont.)

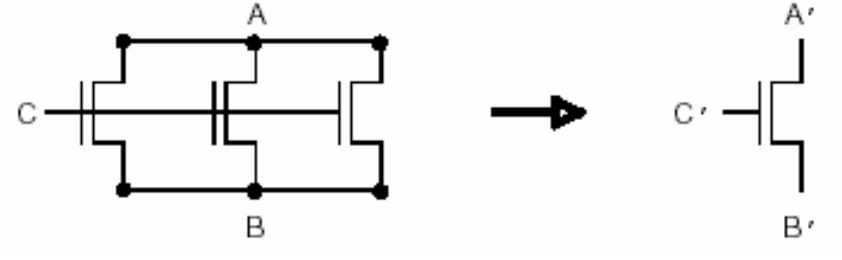
- LVS REDUCE PARALLEL MOS {YES | NO}

  [
  [TOLERANCE {*property_name tolerance_number*}
  [*effective_property_computation*]
  ]

Specifies whether to reduce MOS transistors connected in parallel into single transistors. The default is YES , if you do not include this statement in the rule file

Example:

LVS REDUCE PARALLEL MOS YES [TOLERANCE w 1]
[effective M, W, L
M = sum(M)
p = sum (W * L)
q = sum (W / L)
W = sqrt (p * q)
L = sqrt (p / q)
]



Device Reduction (cont.)

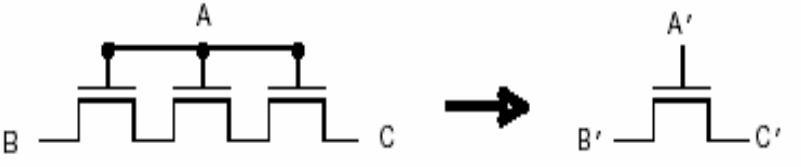
- LVS REDUCE SERIES MOS {YES | NO}

  [
  [TOLERANCE {*property_name tolerance_number*}
  [*effective_property_computation*]
  ]

The default is NO , if you do not include this statement in the rule file.

Example:

LVS REDUCE SERIES MOS YES [TOLERANCE w 1]
[effective M, W, L
M = sum(M)
p = sum (W * L)
q = sum (W / L)
W = sqrt (p * q)
L = sqrt (p / q)
]



Device Reduction (cont.)

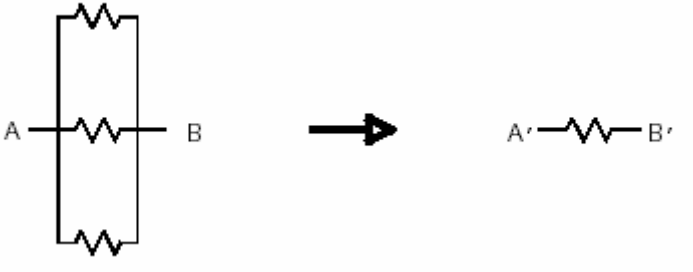
- LVS REDUCE PARALLEL RESISTORS {YES | NO}

  [
  [TOLERANCE {*property_name tolerance_number*}
  [*effective_property_computation*]
  ]

The default is YES , if you do not include this statement in the rule file.

Example:

LVS REDUCE PARALLEL RESISTORS YES [TOLERANCE R 1]
[effective R
R = 1 / sum (1 / R)
]



Device Reduction (cont.)

- LVS REDUCE SERIES RESISTORS {YES | NO}

  [
  [TOLERANCE {*property_name tolerance_number*}
  [*effective_property_computation*]
  ]

The default is YES , if you do not include this statement in the rule file.

Example:

```
LVS REDUCE PARALLEL RESISTORS YES [TOLERANCE R 1]
    [ effective R
      R = sum ( R )
    ]
```

Device Reduction (cont.)

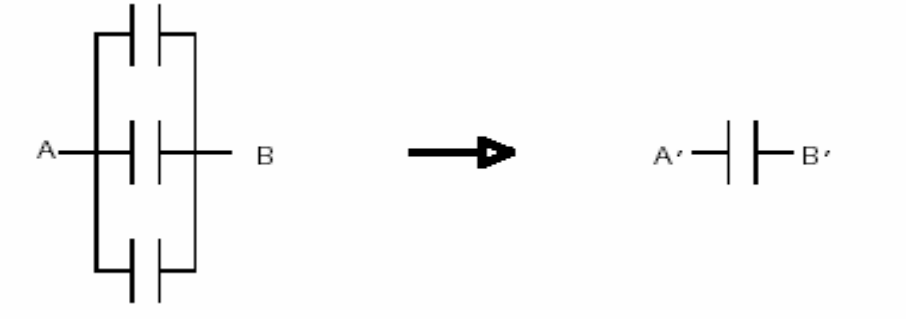
- LVS REDUCE PARALLEL CAPACITORS {YES | NO}

  [
  [TOLERANCE {*property_name tolerance_number*}
  [*effective_property_computation*]
  ]

The default is YES , if you do not include this statement in the rule file

Example:

LVS REDUCE PARALLEL CAPACITORS YES [TOLERANCE C 1]
[effective C
C= sum (C)
]



Device Reduction (cont.)

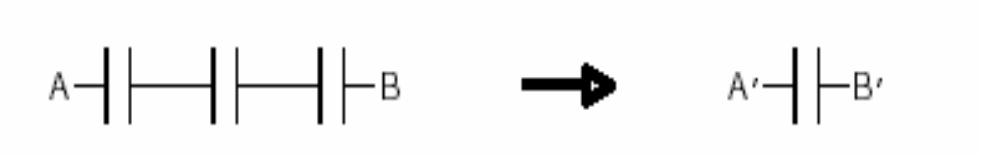
- LVS REDUCE SERIES CAPACITORS {YES | NO}

  [
  [TOLERANCE {*property_name tolerance_number*}
  [*effective_property_computation*]
  ]

The default is YES , if you do not include this statement in the rule file

Example:

LVS REDUCE SERIES CAPACITORS YES [TOLERANCE C 1]
[effective C
C= 1/ sum (1 / C)
]



Device Reduction (cont.)

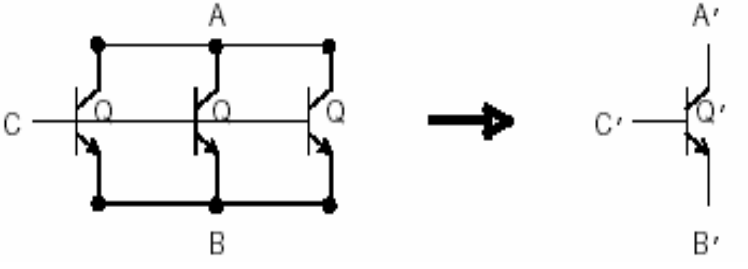
- LVS REDUCE PARALLEL BIPOLAR {YES | NO}

  [
  [TOLERANCE {*property_name tolerance_number*}
  [*effective_property_computation*]
  ]

The default is YES, if you do not include this statement in the rule file.

Example:

LVS REDUCE PARALLEL BIPOLAR YES [TOLERANCE A 1]
[effective A
A= sum (A)
]



Device Reduction (cont.)

- LVS REDUCE component_type [*(component_subtype)*]
  {PARALLEL | SERIES *pin_name_1 pin_name_2*}
  [YES | NO]
  [
  [TOLERANCE {*property_name tolerance_number*}
  [*effective_property_computation*]
  ]

The default is YES.

Example:

```
LVS REDUCE PARALLEL MOS YES   // lowest precedence
LVS REDUCE M PARALLEL NO      // next precedence
LVS REDUCE M(N) PARALLEL [    // highest precedence
EFFECTIVE w
w = sum ( w )
]
```

Device Filtering

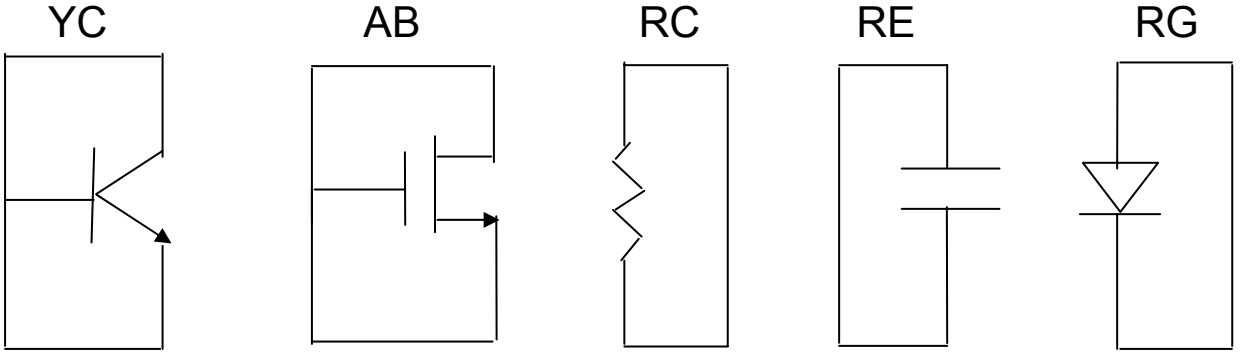
- LVS FILTER UNUSED OPTION *option*
  [SOURCE LAYOUT | SOURCE | LAYOUT]

option no default and case-insensitive

SOURCE LAYOUT is default if you include this into rule file

Example : In our rule file

LVS FILTER UNUSED OPTION YC AB RC RE RG



More option detailed, please refer to the calibre manual

Device Filtering(cont.)

- LVS FILTER UNUSED MOS {NO | YES}
- LVS FILTER UNUSED CAPACITORS {NO | YES}
- LVS FILTER UNUSED RESISTORS {NO | YES}
- LVS FILTER UNUSED DIODES {NO | YES}
- LVS FILTER UNUSED BIPOLAR {NO | YES}

Note

1. All defaults are NOT , if these commands are not included rule file
2.

	UNUSED DEVICE	UNUSED OPTION
MOS	YES	AB AC AD F G J L
CAPACITOR	YES	RD RE
RESISTOR	YES	RB RC
DIODE	YES	RF RG
BIPOLAR	YES	YB

3. **NO** specification of these commands are overridden by
   LVS FILTER UNUSED OPTION

Device Filtering(cont.)

- LVS FILTER component_type [(component_subtype)]
  [property_name [(spice_value)] [filter_constraint]] {SHORT | OPEN}
  [SOURCE LAYOUT | SOURCE | LAYOUT]

  Filters out source and layout instances that conform to the criteria specified by the statement's parameters

EXAMPLE:

LVS FILTER r(x) instpar(r) == 0 SHORT SOURCE

Verilog to Spice

- Transfer top level Verilog netlist to Spice

  v2lvs –v top.v -o top.spi

- Include the spice library file

  Added the following lines into top.spi

  .include "hnstd.spi"

  .instude "IO.spi"

  .include "IP.spi"

  ……

Spice control statements(cont.)

- .GLOBAL < node1 node2 … >

*.GLOBAL < node1 node2 … >

- specifies nodes are globally shared by all subcircuits
- provides a convenient means of communicating power supplies and clocks through the netlist

Example

```
.GLOBAL 4 7 VDD VSS
.subckt  INV A Y
   MN    Y A VSS VSS     N    W=2u   L = 1u
   MP    Y A VDD VDD     P    W=4u   L=  1u
 .ends
```

*** If the cell is block cell by using LVS BOX command and it's spice netlist only have input and output pins, you must add the global pins into the .subckt line.

for example:

```
.GLOBAL  4 7 VDD VSS
.subckt  IP1   VDD VSS A B C
 .ends
```

Spice control statements(cont.)

- *.connect net1 net2
  - Shorts the specified nodes together into one net
  - Apply to global nets anywhere in the netlist
  - Non-global nets in the top-level subcircuit

Example

```
*.CONNECT VSS VSSPST
*.CONNECT GND VSS
*.CONNECT A B
*.GLOBAL VSS  VSSPST
.SUBCKT SUB
  C1 VSS VSSPST
  C2 A B
.ENDS
.SUBCKT  TOP $ top level subcircuit
  C3 A B
   X1 SUB
.ENDS
```

Global VSS and VSSPST are connected

A and B in sub cell are separated

A and B in top cell are connected

Spice control statements

- *.EQUIV newname oldname
  - Specifies equivalence between different source and layout names
  - Can not be used to connect the different nets

Example:

specifies equivalence between the model name nenh of spice and the model name n of layout

*.EQUIV n nenh

Spice control statements(cont.)

- *.LDD
  - When you specify the LVS Spice Conditional LDD YES specification statement in the rule file, $LDD designators in the netlist are processed only if a *.LDD comment-coded statement is also present in the netlist
  - When you specify the LVS Spice Conditional LDD NO, $LDD designators in the netlist are processed unconditionally, whether or not *.LDD is present in the netlist
- *.diode
  - Instructs LVS to ignore diode (D) elements in the netlist
- *.CAPA
  - Instructs LVS to ignore capacitor (C) elements in the netlist.
- *.MEGE
  - Tells the LVS SPICE parser to interpret uppercase “M” scale factors as “Mega” (1E+6) instead of the usual “milli” (1E-3)
  - Lower case “m” scale factors are interpreted as milli
  - The default in the absence of *.MEGA statements is to interpret both uppercase M and lower case m as milli (1E-3)

Thank you!