## LEF File for HP 0.5 Process (HP14TB)

- A partial LEF file is linked to the class page for the HP14TB process
- This contains the header information that defines the routing layer rules (width, spacing, grid)
  - M1 grid = M3 grid = $9\lambda = 2.7\mu$ , horizontal routing
  - M2 grid = $8\lambda = 2.4\mu$ , vertical routing.
- When you output LEF from your library cell abstract views, delete the header information and replace it with the contents of this file if you are using the HP14TB process



## Generating Abstracts

- The 'autoAbgen' program will generate an *abstract* view from a *layout* view
- You will need to add some lines to your *~/.cdsinit* file for autoAbgen to be available:

```
;## Info for autoAbgen
aabsInstallPath = "/opt/ecad/cadence/97a/tools/autoAbgen/etc/autoAbgen" ;
load("/opt/ecad/cadence/97a/tools/autoAbgen/etc/autoAbgen/aaicca.ile")
```

- This will work for the ECE installation

## *.autoAbgen* file

- In your Cadence library directory (should contain directories for each of your cells), need to have an *.autoAbgen* file
  - This defines abstract-generation rules that are dependent upon both the library and the process
- Linked to the class WWW page is an *.autoAbgen* file that for HP 0.5u process that uses the routing grid we have defined
  - Place this file in your cadence library directory and make sure it is named *.autoAbgen*

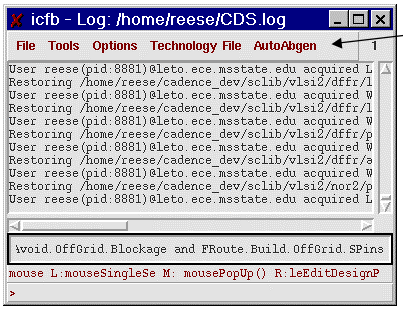


## Preparing your layouts

- Use the *prBoundary (dg)* layer and place a rectangle that defines your cell boundary
  - You may have to use the layer menu to make this layer a valid layer (use Edit -> Set Valid Layers, check box next to *prBoundary (dg)).*
- Make sure your terminals on on-grid (both horizontal and vertical) and you have a pin defined in M1 for each terminal
  - use the 'create pin' command in Virtuoso to create the pin
  - Create a 'shape' pin, define the access to the pin as 'any'
  - For VDD/GND, the shape pin needs to be one rectangle that covers the entire VDD/GND bus
- Add a string property to the cell called *prCellType* with value *standard* . Add a string property called *prCellClass* with value *core* .
  - use the 'Q' hotkey command to bring up the cell properties template and use the 'add' command to add these properties

## Generating an Abstract view

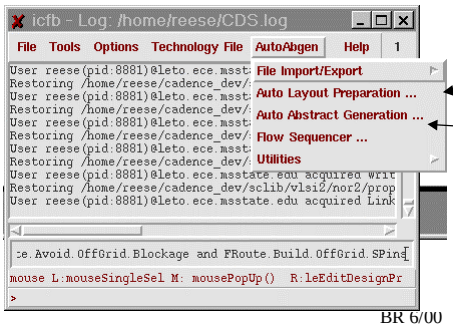
- If you have modified the *~/.cdsinit* file correctly, then the *icfb* main window should have a AutoAbgen choice






## Generating an Abstract (cont.)

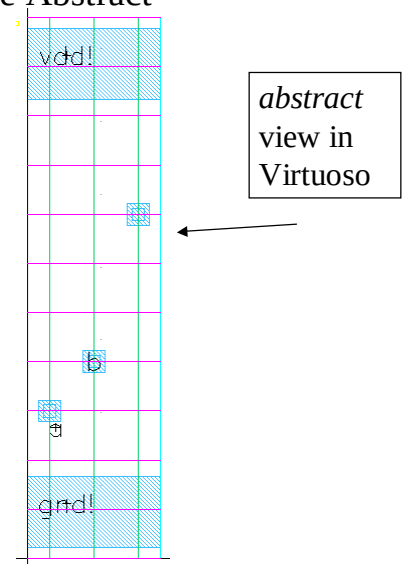
- Abstracts can be generated for all cells in a library, or for only the currently active cell/view open in Virtuoso
  - Suggest that you generate the abstracts one at a time so you can fix any errors that occur
- Open a cell layout view in Virtuoso, then access the AutoAbgen menu and execute the choices “Auto Layout Preparation”, “Auto Layout Generation” in sequence



## Viewing the Abstract

The text log in the *icfb* window will indicate if abstract generation is successful or failure. Be sure to see if the text log indicates if your pins or on grid or off grid.

Viewing the abstract in Virtuoso will show the grid and you can see if the terminals are on-grid




*abstract* view in Virtuoso



## Creating the *.lef* file

Once you have created an abstract view for each of you cells, export your library to a .lef file.

Create a text file called *lefout.list* . Each line of the file is of the format:

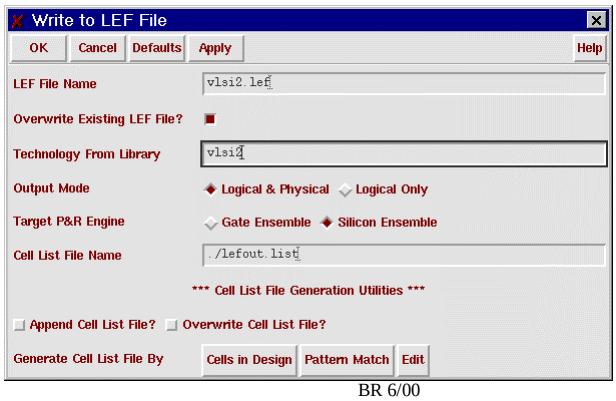
libname cellname viewname

You should have an entry for each cell in your library, i.e.:

vlsi2 inv abstract
vlsi2 nand2 abstract
vlsi2 nor2 abstract
vlsi2 dffr abstract

## Creating the *.lef* file (cont.)

From icfb, select the File → Export → LEF menu option. This will popup the menu below. Select 'Silicon Ensemble' as the target P&R engine. The LEF file below will be written to 'vlsi2.lef'.






```
MACRO inv
    CLASS CORE ;
    FOREIGN inv 0.00 0.00  ;
    ORIGIN -0.00 -0.00 ;
    SIZE 4.80 BY 29.70 ;
    SYMMETRY x y ;
    SITE core ;
    PIN y
        DIRECTION OUTPUT ;
        PORT
        LAYER met1 ;
        RECT  3.26 18.56 3.94 19.2
;
        END
    END y
    PIN a
        DIRECTION INPUT ;
        PORT
        LAYER met1 ;
        RECT  0.86 7.76 1.54 8.44
        END
    END a
    PIN vdd!
        DIRECTION INOUT ;
        USE power ;
        SHAPE ABUTMENT ;
        PORT
        LAYER met1 ;
        RECT  0.00 25.20 4.80 29.1
;
        END
    END vdd!
    PIN gnd!
        DIRECTION INOUT ;
        USE ground ;
        SHAPE ABUTMENT ;
        PORT
        LAYER met1 ;
        RECT  0.00 0.60 4.80 4.50
        END
    END gnd!
    OBS
        LAYER met1 ;
        RECT  3.00 18.30 4.20 19.5
;
        RECT  0.60 7.50 1.80 8.70
    END
END inv
```

Typical cell entry in lef file generated by the Export→LEF option.

## Silicon Ensemble Place/Route

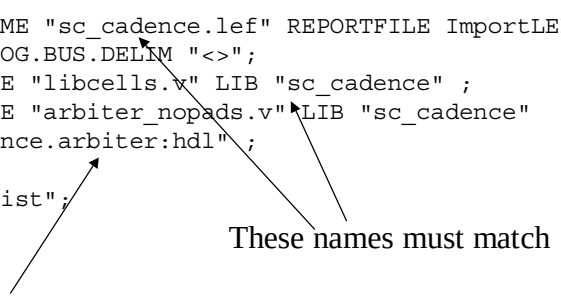
- Once you have an LEF file your library, a verilog netlist file for your design, and Verilog library file, you can run SE
- Doing Place/Route in SE has the following steps:
  - Netlist parsing
  - Floorplanning
  - IO Placement
  - Power planning
  - Cell Placement
  - Power Routing
  - Detailed Routing



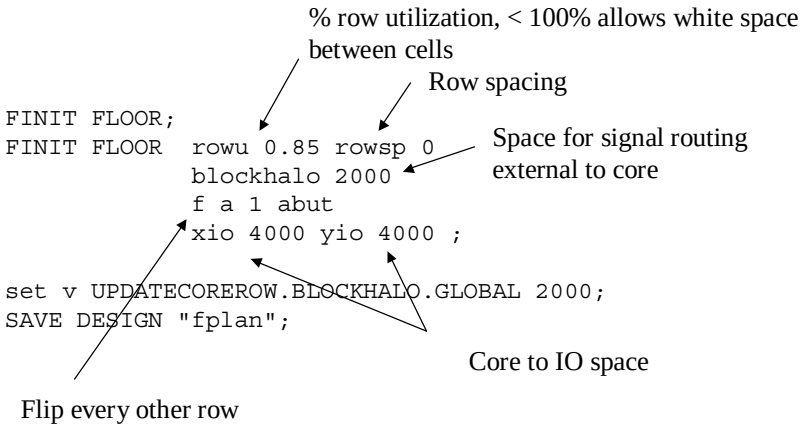
## Netlist Parsing (SE V5.2, no_pads example)

```
# Reading in the LEF and TLF files, the verilog netlist
#
FINPUT LEF FILENAME "sc_cadence.lef" REPORTFILE ImportLEF.rpt ;
SET V INPUT.VERILOG.BUS.DELIM "<>";
INPUT VERILOG FILE "libcells.v" LIB "sc_cadence" ;
INPUT VERILOG FILE "arbiter_nopads.v" LIB "sc_cadence"
  DESIGN "sc_cadence.arbiter:hdl" ;

SAVE DESIGN "netlist";
```



## Floorplanning




Units are in centi-microns, so 2000 = 20.00 μ



## IO Placement

```
IOPLACE AUTOMATIC STYLE ABUTCENTER ;
```

If there are no IO cells in the design, the above line is all that is needed. The IO pins will be brought out to all sides, and spaced evenly (regardless of the 'STYLE' setting).

Will discuss placement of IO cells in a later slide.

## Power Planning

```
#
# Power Planning
#
SET VAR DRAW.SWIRE.AT ON ;
SET VAR DRAW.CHANNEL.AT ON ;
BUILD CHANNEL ;


CONSTRUCT RING NET "gnd!" NET "vdd!" LAYER metal1 CORERINGWIDTH 1000
SPACING CENTER BLOCKRINGWIDTH 500 LAYER metal2 CORERINGWIDTH 1000
SPACING CENTER BLOCKRINGWIDTH 500  ;

# in this design, do not add vertical stripe
#ADD STRIPE    NET "gnd!" NET "vdd!" DIRECTION Vertical LAYER metal2
WIDTH 500 SPACING 200 LOWERMARGIN 10000 STEP 20000 ALL ;
DISPOSE CHANNEL ;
SET VAR DRAW.CHANNEL.AT OFF ;
SET VAR DRAW.SWIRE.AT SMALL ;
```



## Comments on Power Planning

- *Construct Ring* command is used to construct power/gnd ring around the core and also in the IO cells
  - *CORERINGWIDTH* specifies width of core ring
  - *BLOCKRINGWIDTH* specifies width of ring in I/O cells
  - Horizontal and Vertical routing layers must be specified
- Add Stripe command can be used to create vertical power/gnd busses
  - Needed for large cell arrays to distribute power throughout the array
  - Command has required option that specifies the spacing between the vertical power stripes

## Cell Placement

```
QP FILENAME "qpcells.cfg" ;
OUTPUT DEF FILENAME "qplace.def" ;
SAVE DESIGN "qplace" ;
REPORT SUMMARY FILENAME "qplace.rpt";
REPORT CONSTRAINTS DETAILED FILENAME "qplace.constraints";
```

QP (quick place) is the cell placement command.

The *qpcells.cfg* file can be used to specify different options to QP, the default options work fine for us.

To see a listing of all options, do:

% qp -h



## Power Routing

```
# Connecting the power
#
CONNECT RING NET "gnd!" NET "vdd!"
 STRIPE BLOCK ALLPORT
 IOPAD ALLPORT WIDTH 1000
 IORING FOLLOWPIN WIDTH 150 ;
```

IOPAD width is the width of the VDD/GND pins in the IO cell rings.

IORING width is the width of the Vdd/GND pins the in standard cells.

If you specify the wrong widths, then router will not find the pins and will fail.

## Detailed Signal Routing

```
#
# Do routing
#

SET VAR WROUTE.FINAL TRUE ;
SET VAR WROUTE.GLOBAL TRUE ;
SET VAR WROUTE.INCREMENTAL.FINAL FALSE ;
WROUTE NOCONFIG ;

SAVE DESIGN "routed" ;
#
OUTPUT DEF FILENAME "arbiter.def" ;
```

WROUTE (Warp Route) is the detailed router used in SE v5.2. GROUTE/FROUTE was used in SE v5.0.



## IO placement with IO Cells

```
IOPLACE FILENAME "arbiterTop.ioc" STYLE EVEN ;
```

External file used to specify placement of IO cells. IO Cells separated into 4 groups (LEFT, RIGHT, TOP, BOTTOM). A sample grouping is shown below: