Routing

- Automatic Place/Route of systems blocks is integral design of modern VLSI devices
  - ⇒Cell Libraries and macro blocks need to be designed with system level routing issues in mind
- Typically, design macro blocks to use as few routing layers as possible so that higher level routing can go completely over-the-top of the block for inter-block routing.
  - ⇒Not unusual for a macro block to only use local interconnect (silicided poly) and MET1; this reserves MET2 and higher for inter-block routing.

Standard Cell Design and Routing

- Standard cell implementations use a library of standard cells that implement basic logic functions.
- Standard cells share the same height, and the placement tool creates rows of cells, each cell abutting with its neighbor
  - ⇒power connections between cells formed by abuttment
- The router creates connections by routing in available metal layers
  - ⇒Routing can be over the top of the cell or in channels between rows

Routing

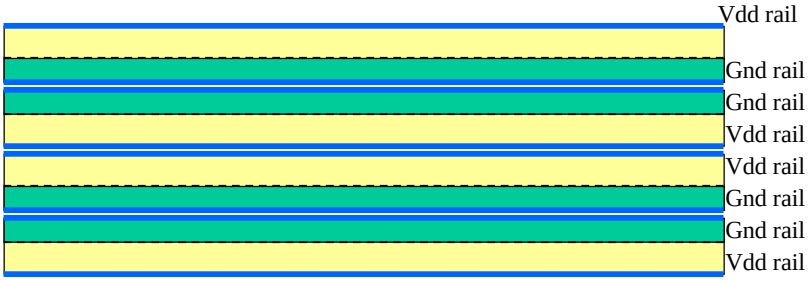
- Routers typically prefer that the pins of cells lie on some grid for routing efficiency purposes
  - ⇒Routers can usually route to off grid pins but requires more execution time and memory.
- Grids are defined for each routing layer
  - ⇒Grid spacing should be at least line-on-via, and is usually via-on-via
- A HVH routing styles means that there are three layers; with the first layer running horizontally, 2nd vertically, 3rd horizontally (M1 H, M2 V, M3 H)

Cadence Standard Cell P&R Particulars

- We will use a tool called Silicon Ensemble for standard cell placement and routing
- Requirements
  - ⇒Pins should align at a grid point (aligned with both vertical and horizontal grids)
  - ⇒Rows are placed to align with vertical and horizontal routing grids
  - ⇒Cell height should be multiple of Horizontal grid
    - →If multiple horizontal grids (i.e, HVH), then cell height should a common multiple of the grid spacing. If the M1 grid spacing is 6 lambda, and the M3 grid spacing is 8 lambda, then cell height should be an multiple of both 6 and 8 (ie., 24 lambda, 48 lambda, etc)
    - →For this reason, the ratio of M1 to M3 grid spacing should be kept simple (i.e 1:1, 2:1, NOT 11:7)

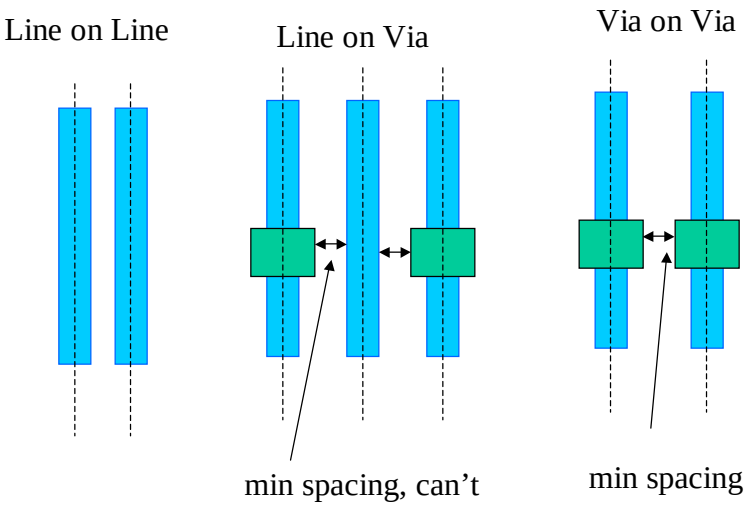
Cadence Standard Cell P&R Particulars

- Alternate rows can be flipped vertically so that powers rails of neighboring rows can abut



- Cells can be flipped horizontally to reduce routing lengths.

Metal Spacing Grids



Routing Grid for HP14TB process (λ = 0.3u)

- Minimum Spacings are:
  - ⇒Via on Via horizontal (via width + M1 spacing) = 4 + 3 =7 lambda
  - ⇒Via on Via vertical (via width + M2 spacing) = 4 +3 = 7 lambda
  - ⇒Via2 on Via2 horizontal (via width + M3 spacing) = 6 + 3 = 9 lambda
- Using minimum spacings give a M1:M3 ratio of 7:9
  - ⇒Odd ratio, Cadence recommends that in this case M1 and M3 grids should be the same (i.e, 1:1 ratio).

Standard Cell Guidelines

- Will be conservative and use vertical routing grid of 8 lambda
  - ⇒Cell widths must be multiple of vertical routing grid (16, 24, 32, etc).
  - ⇒First vertical routing track will start at one-half vertical grid inside of cell
    - →First valid terminal location is one-half vertical grid into cell.
- Will use horizontal routing grid of 9 lambda
  - ⇒Cell heights must be multiple of horizontal routing grid

What should Cell Height be?

- When routing was limited to just two layers of metal, wanted short cells
  - ⇒Routing was M1 horizonal, M2 vertical
  - ⇒Most of the routing was done in the channels between cell rows
  - ⇒Short cells minimized Row to Row spacing
- With over the cell routing (HVH), tall cells now slide under the routing, and channels have disappeared.
  - ⇒Taller cells allow for multiple drive strength cells which is important for modern synthesis tools.

Cell Heights we have used

- SCMOS library (0.8u, 1.2 u library)
  - ⇒ 68 lambda - designed for 2 layer routing, cell height was kept small
- GCMOS library (0.5u)
  - ⇒ 100 lambda - designed for 3 layer routing, wanted multiple drive strengths
- RadTolerant Library (0.5u)
  - ⇒ 150 lambda - required guard rings around PMOS, NMOS transistor sites which pushed cell height up

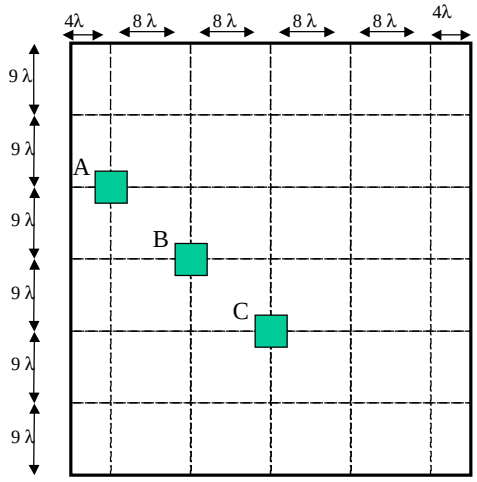
A Routing Comparison

- For the GCMOS library (100 lambda) and RadTolerant library(150 lambda), did a P&R for a 1932 cell design
  - ⇒GCMOS area : 1591 x 813 u = 1293483 u**2
  - ⇒RadTol area: 1716 x 938 = 1609608 u**2
  - ⇒ Area Ratio of 1.2 (RadTol/ GCMOS) – routing limited
  - ⇒ Cell Height Ratio RadTol/GCMOS = 1.5
- For a smaller design (325 cells), the area was:
  - ⇒GCMOS: 619 x 385 = 238315
  - ⇒RadTol: 806 x 464 = 373984
  - ⇒Ratio of 1.6 (RadTol/GCMOS) – cell area limited

Standard Cell Template

- Vdd/Gnd Rail width rule of thumb is 5x minimum width
  - ⇒ 15 lambda
- Sample Cell library provided with Cadence had cell heights that provided 9 horizontal routing tracks
  - ⇒We will be somewhat conservative and use a cell height that gives 11 horizontal routing tracks (99 lambda).
- Transistor areas should be positioned so that routing tracks in middle of cell are free

Standard Cell Routing Template (cont)



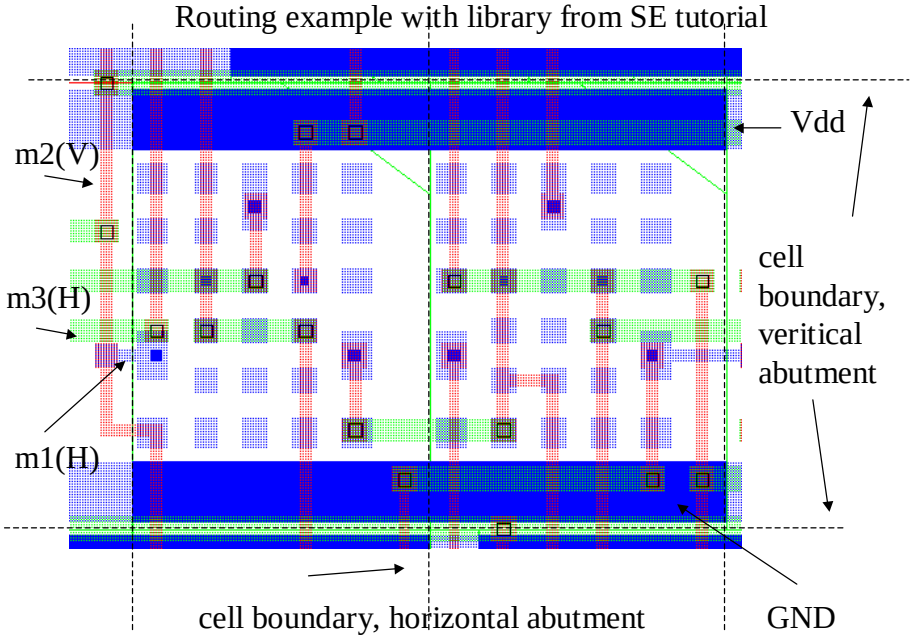
Valid terminal locations are at grid intersections. If possible, terminals should staggered horizontally. This allows easier horizontal routing access.

Can also designate multiple entry points via metal 1 to a terminal along the veritical direction

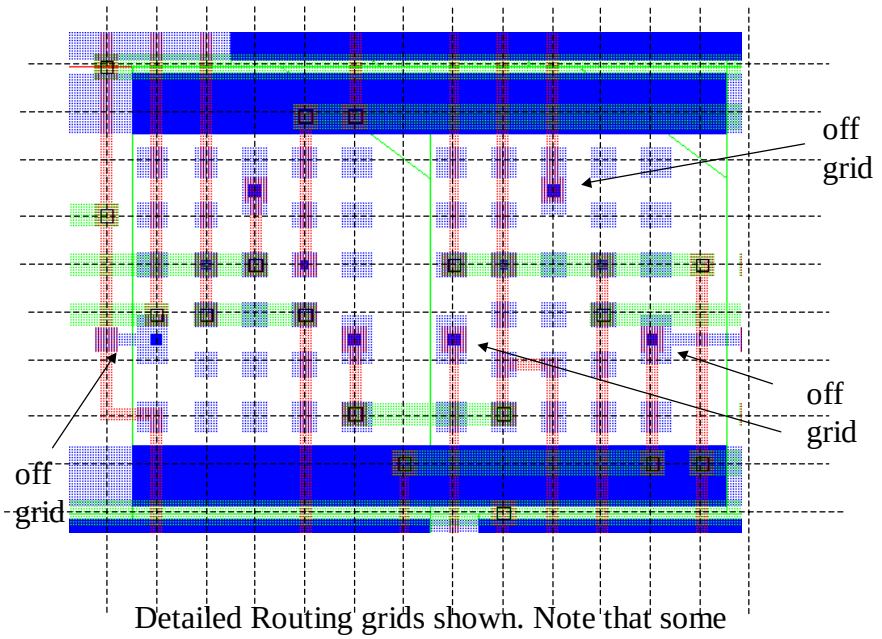
Standard Cell layout guidelines

- Orgin of cell at 0,0 in lower left hand corner
- Cell width multiple of 8 lambda, height is 99 lambda
- Vdd/Gnd rails 15 lambda wide, start at 1.5 lambda vertical spacing inside cell boundary
  - ⇒This way, if router does not support flipping rows, then cells can be abutted vertically without DRVs
- PTAP, NTAP contacts under power rails
- Cells can be abutted horizontally without DRV's.
- Grow transistors from cell bottom towards cell center
  - ⇒Use poly routing wherever possible, limit to 30-45 um
  - ⇒If have to block horizontal MET1 route, block from one side only

Routing example with library from SE tutorial



11/14/00 15

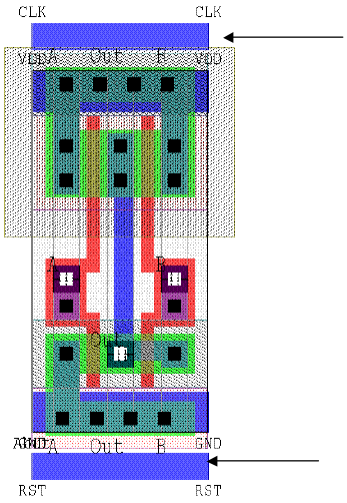


Detailed Routing grids shown. Note that some routes are off grid (off grid routing allowed by SE)

11/14/00 16

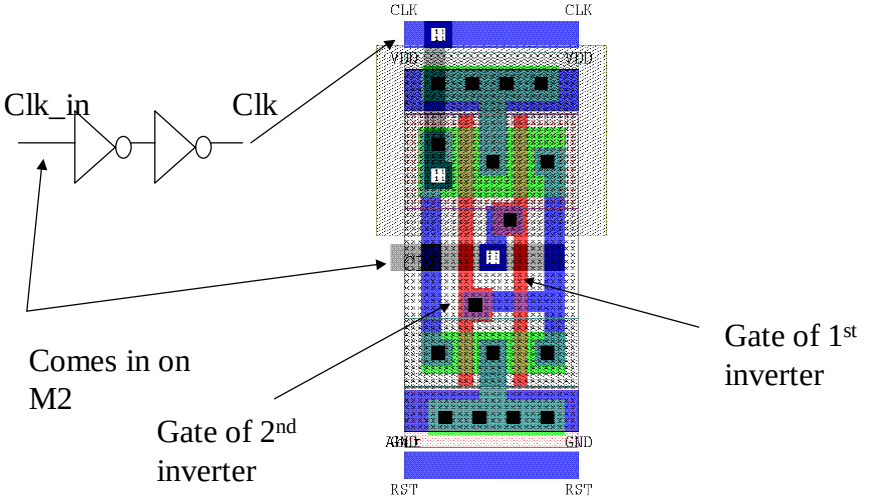
Global Net routing (Clock, Reset)

Tanner Standard Cell libraries for Leda 0.35, 0.25 processes available from MOSIS specify Clock, reset line routing in Standard Cell Template.

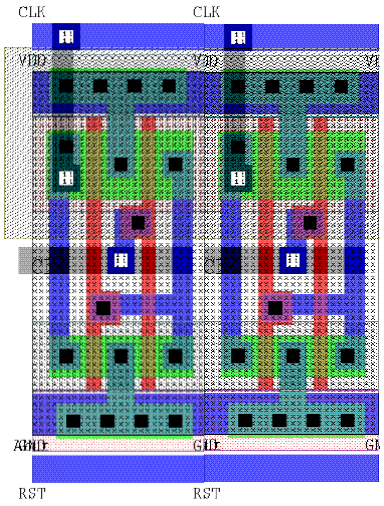


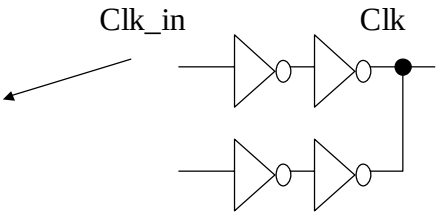
Clock Buffer Cells

Buffer clock left



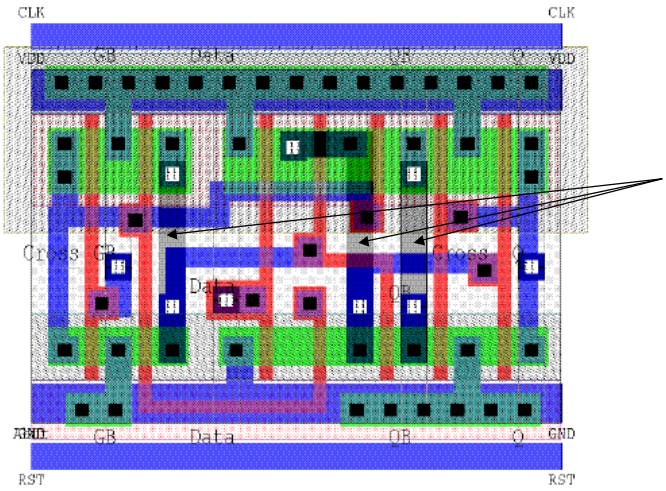
Abutment of clock buffer cells





Multiple clock buffers inserted in row can give arbitrary drive strenght

Latch Cell



M2 usage in cell always runs vertically so as to block fewest routing channels.

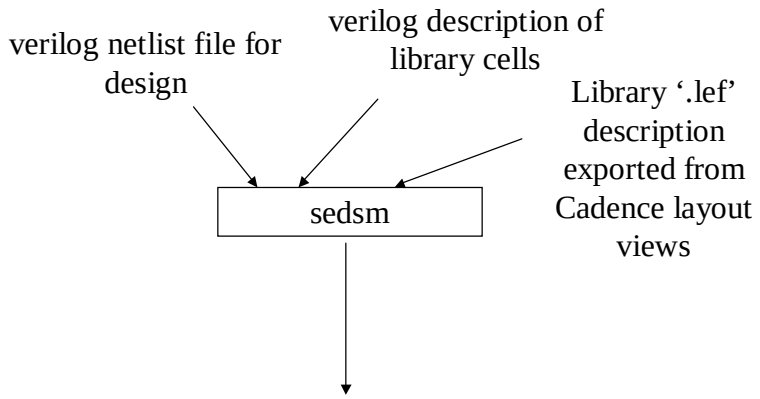
Does not tie into clock routing, this must be done via external router

If you want to try out Cadence

- Grab cds.lib, display.drf from 'misc links' off WWW page.
- 'swsetup cadence' to place tools on path
- Make a temp directory, change into it, and execute:
  - ⇒ % icfb &
  - ⇒The 'Help' button on the menu will access the online help
  - ⇒ Use the 'Go -> Main Menu' to get to list of all documents
  - ⇒Choose 'IC tools'
    - →Layout Design -> Cell Design Tutorial (manual layout tutorial)
    - →Place and Route -> Silicon Ensemble Tutorial (Stdcell)
    - →Layout Design -> Virtuso Layout Synthesizer Tutorial

Arbiter Examples

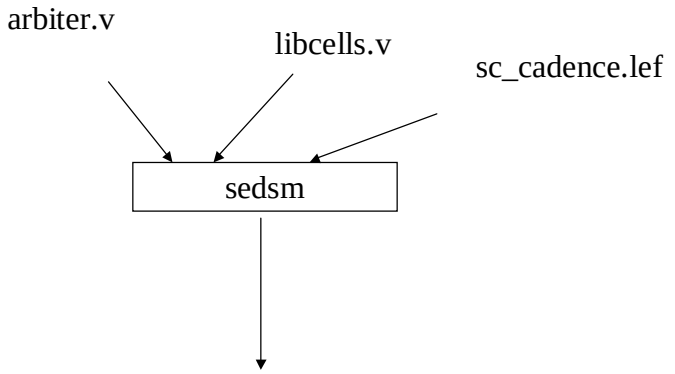
- ZIP archive attached to this lecture has three examples of using 'sedsm' (Silicon Ensemble, Deep sub-micron)
- Example uses a design called 'arbiter' that is about 300 cells mapped to the example library provided with the SE tutorial (synopsys synthesis lib *sc_cadence.lib*)
  - ⇒ *arbiter_pads* -- places pads along with the standard cell core
  - ⇒*arbiter_nopads_reese* – standard cell core only, pins routed to edges
  - ⇒arbiter_nopads_scott – example provided by Scott Jennings than only routes the standard cell core. Uses a difference set of *sedsm* commands than 'reese' example.



Final placed/routed design
(binary file *.dhd format)
(ascii file *.def format)

*.def placed/routed design file is then imported back into Virtuoso Layout tool

11/14/00 23



Final placed/routed design
(binary file *.dhd format)
(ascii file *.def format)

*.def placed/routed design file is then imported back into Virtuoso Layout tool

11/14/00 24

Running sedsm

sedsm can be run in interactive mode (as in tutorial) or in automated mode via script file

To run a script file do:

% sedsm –m=200 –b –gd=ansi “EXECUTE script.mac;” &

Example command files for *sedsm* have a *.mac* extension

To run in interactive mode do:

% sedsm

Generating Input files for *sedsm*

- Verilog netlist file of design generated by Synopsys
- Library verilog file simply has a verilog description of each library cell
  ⇒ look at *libcells.v* in ZIP archive for an example.
- The LEF (library exchange format) file is the most important file and the most difficult to generate
  ⇒ Contains the spacing rules (metal, VIAs) for your technology
  ⇒Contains a cell *abstract* for each cell in your library
  ⇒the LEF file is created via the ‘export’ option under *icfb*

Cell *abstracts*

- The LEF abstract of your cell does not contain all of the layout of your cell
  - ⇒It simply contains the layers/vias that are important from a routing point of view
- The *layout* view of your cell (*in Virtuoso)* contains the detailed layout of your design
- The *abstract* view of your cell (*in Virtuoso)* contains only the routing obstructions, pins of your cell
  - ⇒Abstracts can be generated automatically from a layout view by a tool called *autoabgen.* Abstract views generated by *autoabgen* may need to be tweaked a bit before exporting to a LEF file.
  - ⇒Abstracts can also be generated manually, simply more work to do it