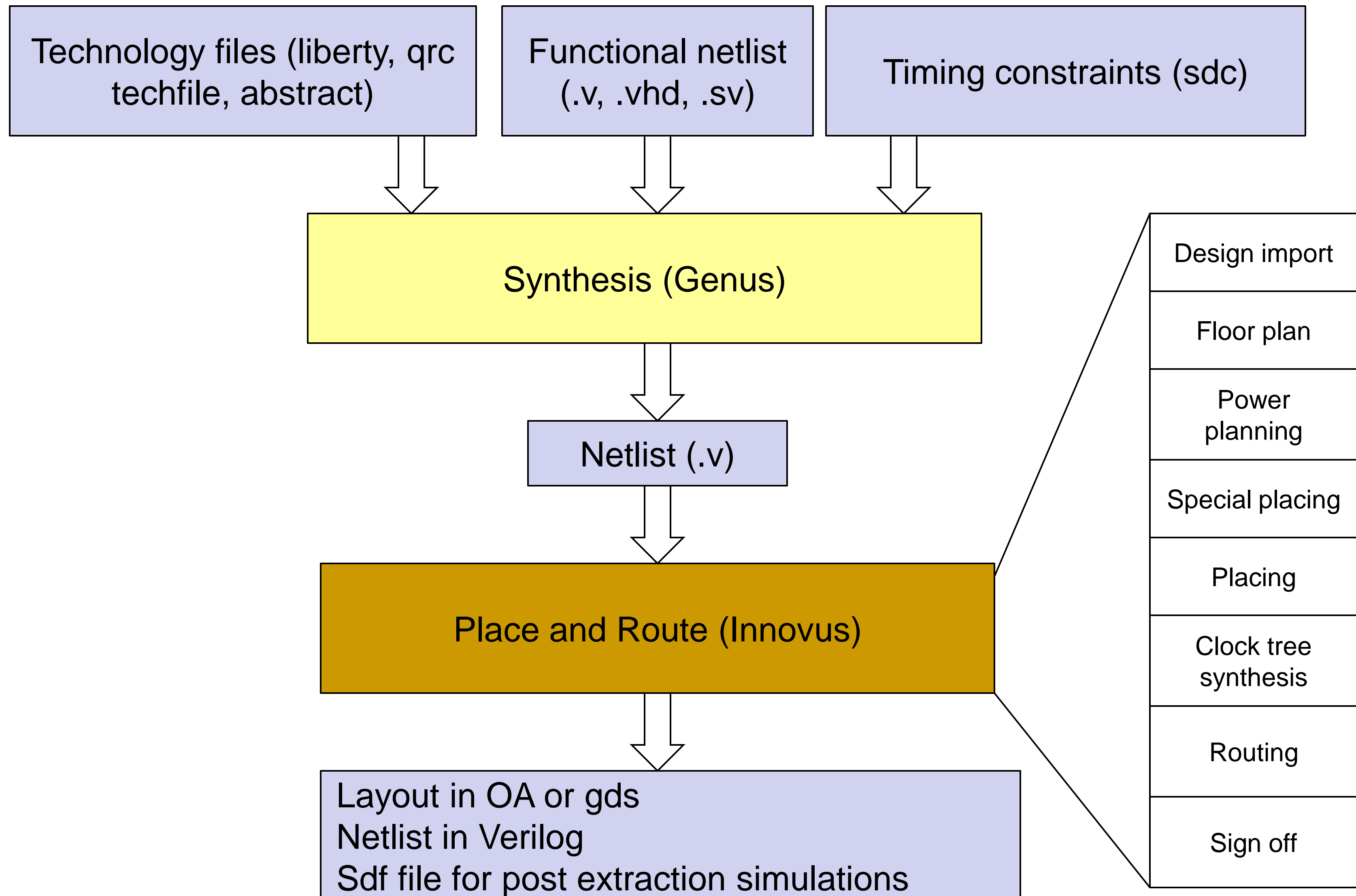
Place and route

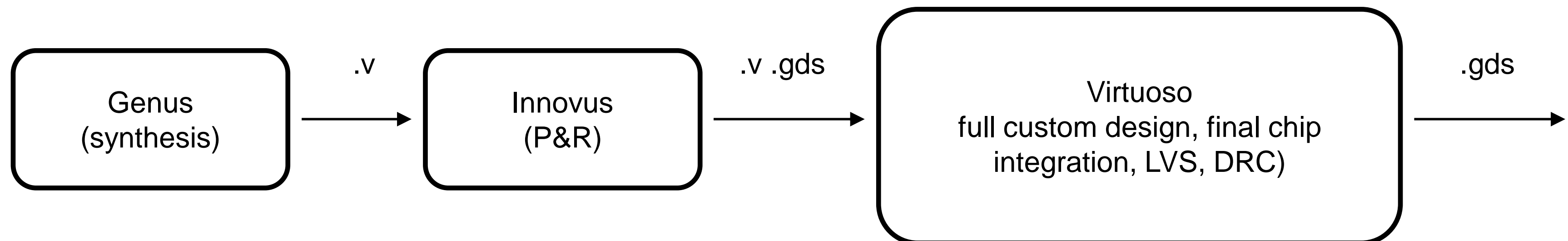
- Get an overview of place and route steps
- Place and route follows the synthesis

Design flow



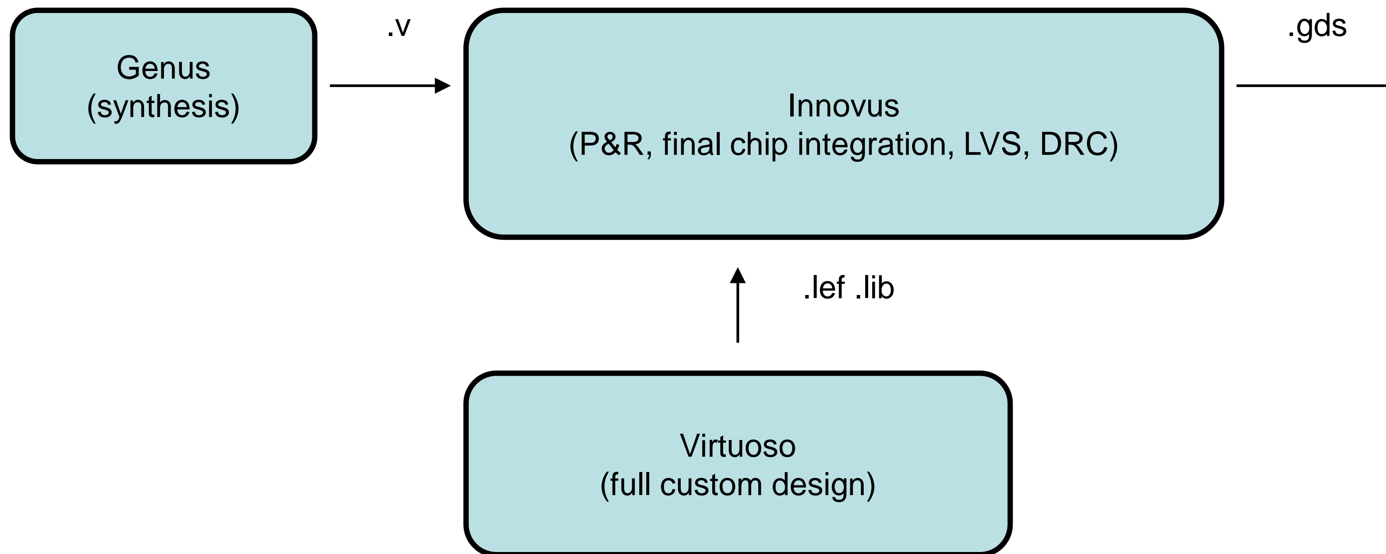
Analog on top

- Quite often chips contain analog and digital circuits. In this case, we are talking about a mixed mode chip.
- There are two ways how a mixed mode chip can be designed. The first possibility is “analog on top”. The digital parts are automatically synthesized and layout is generated by the tools “Genus” and “Innovus”.



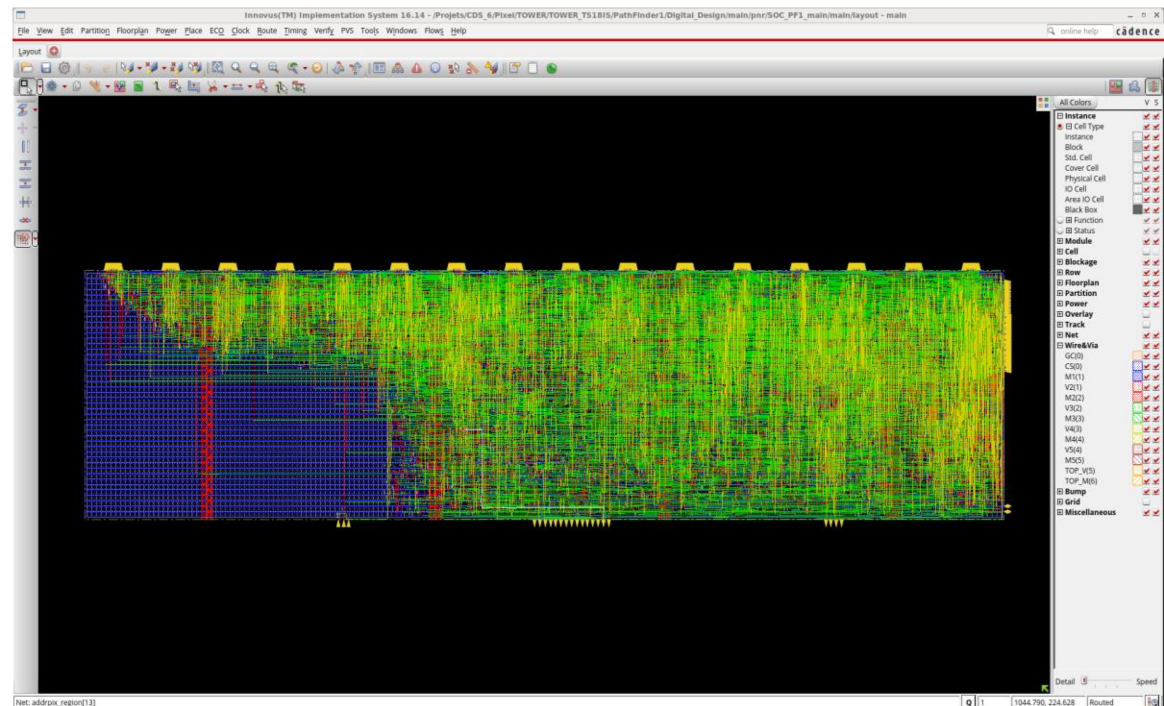
Digital on top

- In the case of “digital on top” design flow, analog blocks, which are designed in Virtuoso, are imported into Innovus and placed automatically. This design flow avoids manual routing of lines between analog and digital blocks. It checkes timing automatically. Analog blocks are taken by Innovus as an abstract view. Sdc file for the top module is needed. Lib file of the analog block could be provided as well.



GUI (manual input of commands) vs. script

- P&R tool Innovus has graphical user interface (GUI) and allows performing of steps by mouse click and form filling. GUI can be used for learning, but it is impractical when whole P&R has to be repeated many times.
- Alternative to GUI use is performing of P&R from TCL script.
- The first step in P&R is importing of design. The netlist should be imported, top cell name specified, power and ground nets defined. Process node should be specified to tune the extraction of capacitances and resistances of metal lines.



Step 2: Floor planning

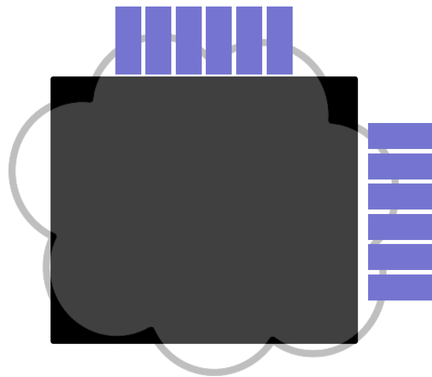
- The next step in P&R is floorplanning.
- The floor planning include following tasks: defining of design dimensions and area for input-output- (IO-) cells, pin placement, placing of reusable pieces of logic (hard-macros or intellectual property (IP) cells), defining rows of standard cells with preliminary placement of standard cells from synthesis output
- The output of the floorplan is the design exchange format “def” file. This file gives information about macros in design, placement, pin location, metal blockages, orientation. The def file can be also used as input for Genus tool, to perform a new iteration of synthesis. In this way, Syntesis will have idea about design size and optimize the netlist accordingly

Floor planning

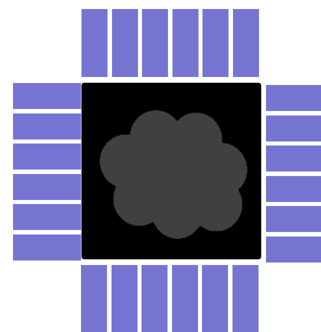
- Choosing of correct size for floorplan is important. Floorplan should be big enough so that that there is space for all needed logic cells
- The ratio of floorplan area and the space used for standard cells is called utilization
- Both too high and tool low utilization are not good. There should be enough free space so that the clock buffers can be placed and that the tool has the freedom to move the cells and optimize timing.
- On the other hand, the floorplan should not be too big, because it may be too “empty” and the lines will be unnecessarily long which increases their capacitances and make circuits slow

Floor planning

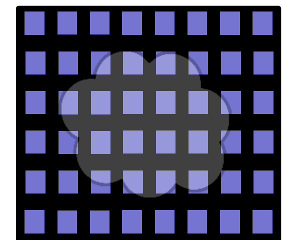
- The maximum area for floorplan is also limited by production costs. Big chips are more expensive. Also, not entire chip area can be used by logic circuits. The input-output- (IO-) contacts (pads) take quite a lot of space.
- We distinguish here between the IO-bound designs (number of IO-pads is limiting factor) where the pads take the most of chip-space, because there are many input signals, are the core-bound designs where the core logic takes the most of the space.
- In the following slides (9 - 30), we will discuss several topics related to floor planning: The choice of the chip area, IO pads, IP-blocks, hard-macros, well-tap cells…



Core Area Bound



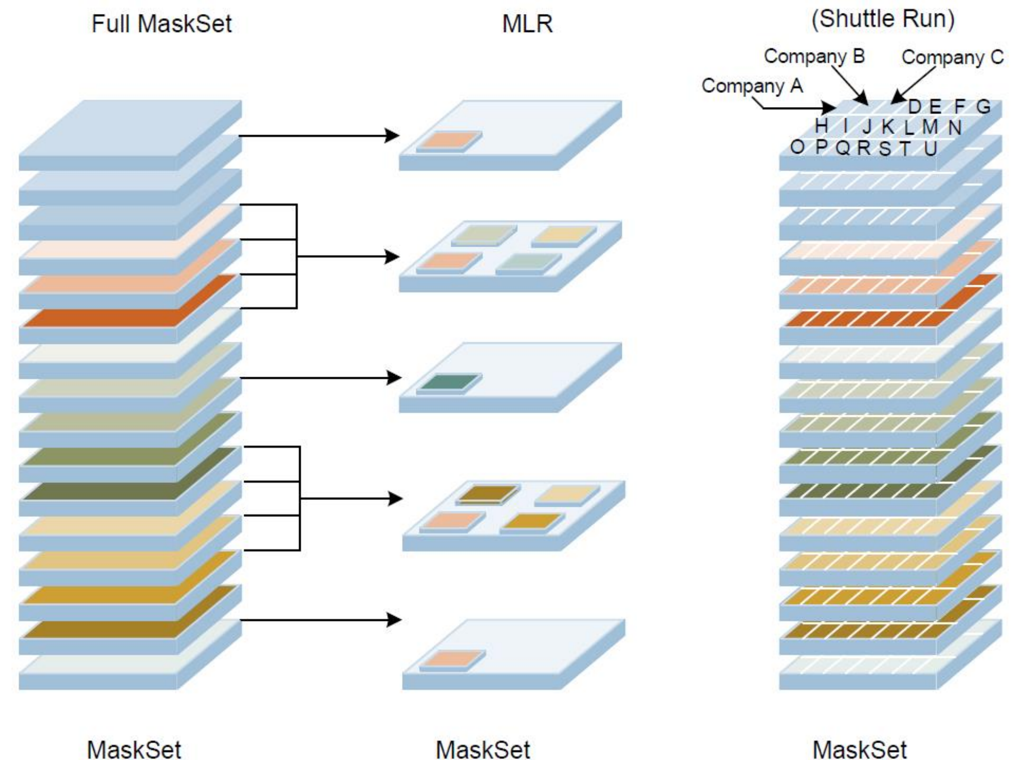
I/O Bound



I/O Bound

Three possibilities for chip production

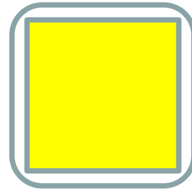
- To understand the production costs, let us introduce the three possibilities for chip production:
- In the case of **full mask set run**, one customer uses the entire mask set for his design or designs. The size of the mask projection on the wafer, called the reticle, limits the maximum chip size
- In the case of **MPW shuttle** run, many customers pay for a mask set and the reticle area is shared by many designs. The advantage is that the run cost is quite small for one customer.
- In the case of **MLM** run, one mask is used for several layers. One customer pays for the mask set, and since less masks are needed than in the case of the full mask set run, the mask cost is reduced

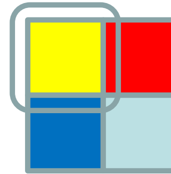


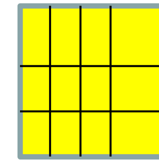
Three possibilities for chip production

Full mask run | MLM | MPW shuttle run

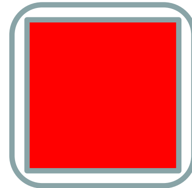
Step 1
production of yellow layer

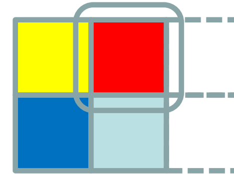


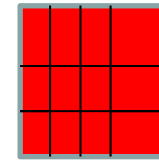




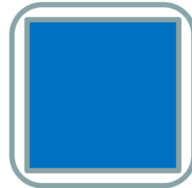
Step 2
production of red layer

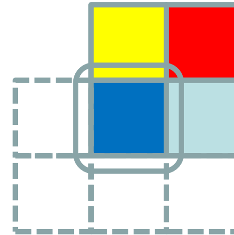


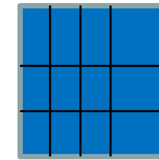


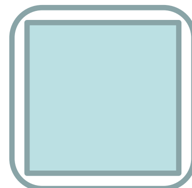


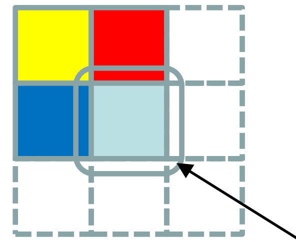
Step 3
production of blue layer

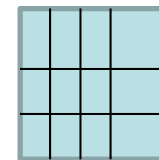








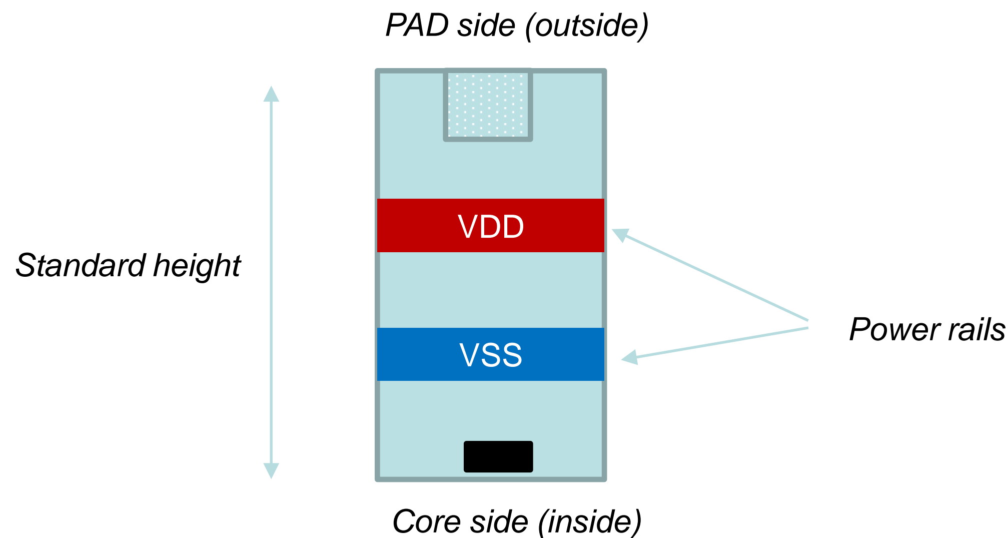




Only this segment has correct structures

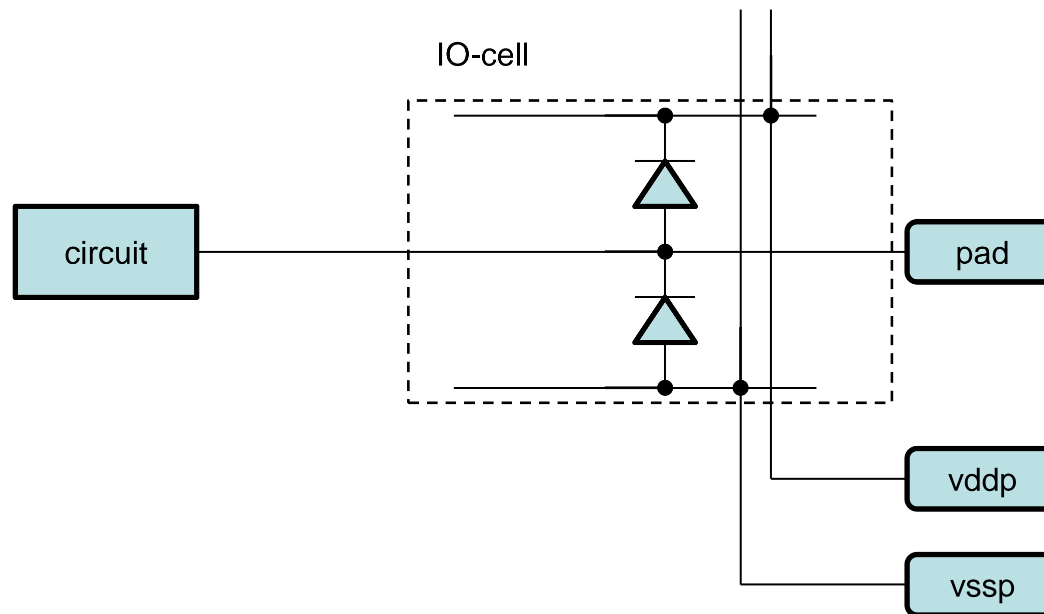
IO pads

- IO-pads provide interface to external world. The IO-pad consists of the metal contact (the contact pad) and the IO-cell with electrostatic discharge (ESD) protection circuits and buffers. IO cells are in layout quite big. Like standard cells, they are designed to be abutted to each other.

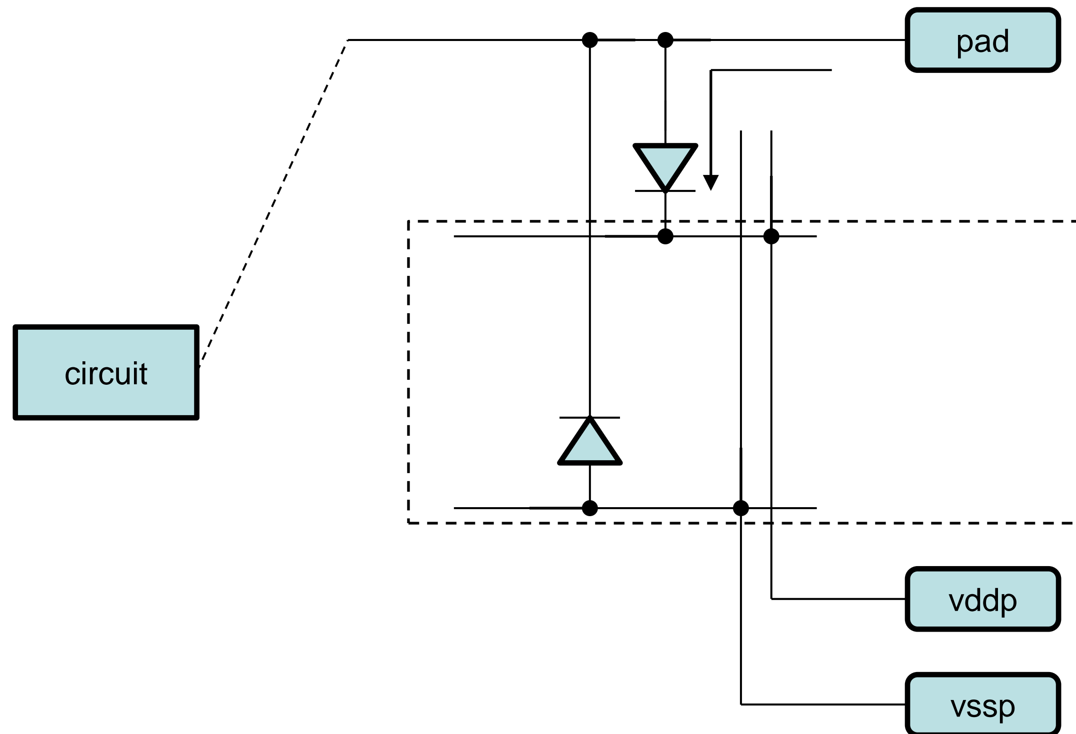


Analog pad

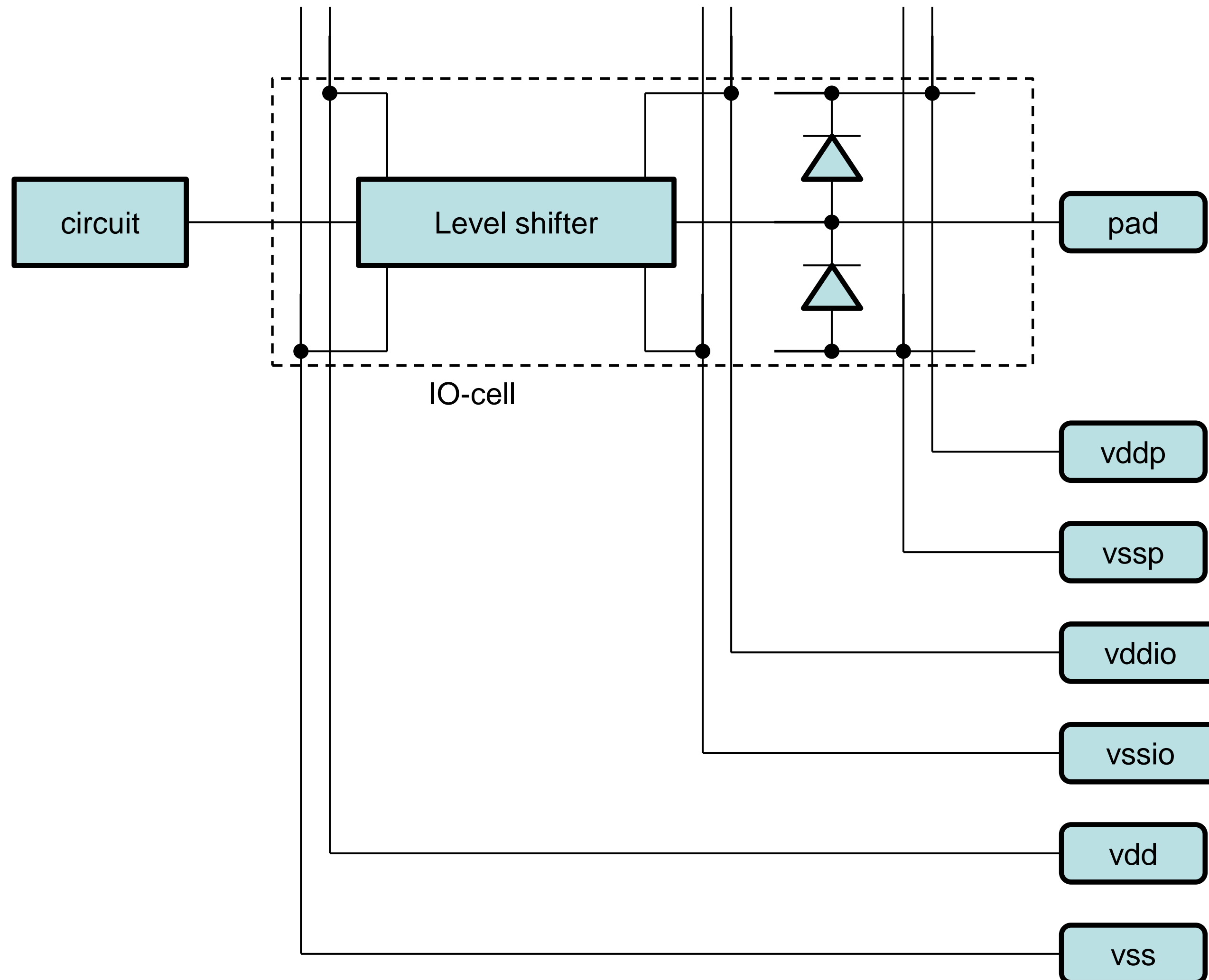
- The slide shows the analog pad. The ESD protection is usually done with two diodes that conduct current when the pad voltage exceeds the level vddp (typically 1.8V) or drops below the level vssp (typically 0V).



Analog pad

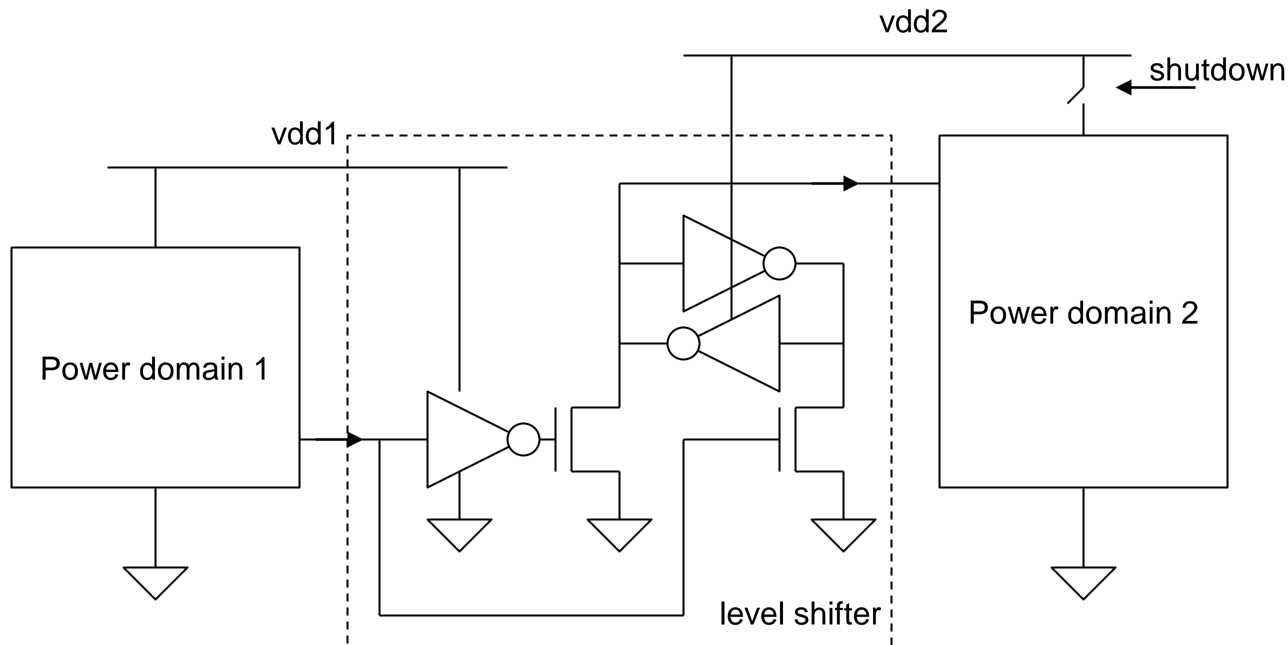


- This slide shows the digital pad. Its structure is more complicated. It contains ESD protection and a level shifter, whose purpose is to translate the logic levels from (vss to vdd) to (vssio to vddio)



Power domains

- Power domains are used to separate various supplies. Typically we have analog and digital domains, but also multiple digital power domains are possible. Digital and analog domains are separated to avoid power supply cross talk. One digital domain can have a lower supply voltage to save power and another a higher voltage for better performances. Advanced options: Digital power domains can interface using level shifters placed by the tool. One digital power domain can be “shutdown” and put into a sleep mode.



Power domains

- IO cells have specific rules for power rail connections. When the cells are abutted the power rails will be shorted. In the case of IO cells that belong to different domains, spacing cells (like IDDBREAK) should be inserted to disconnect the rails.

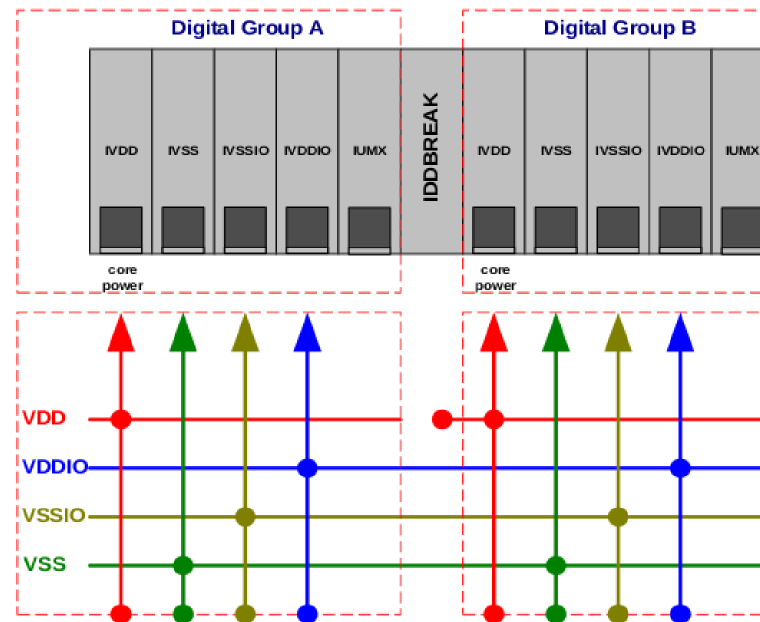
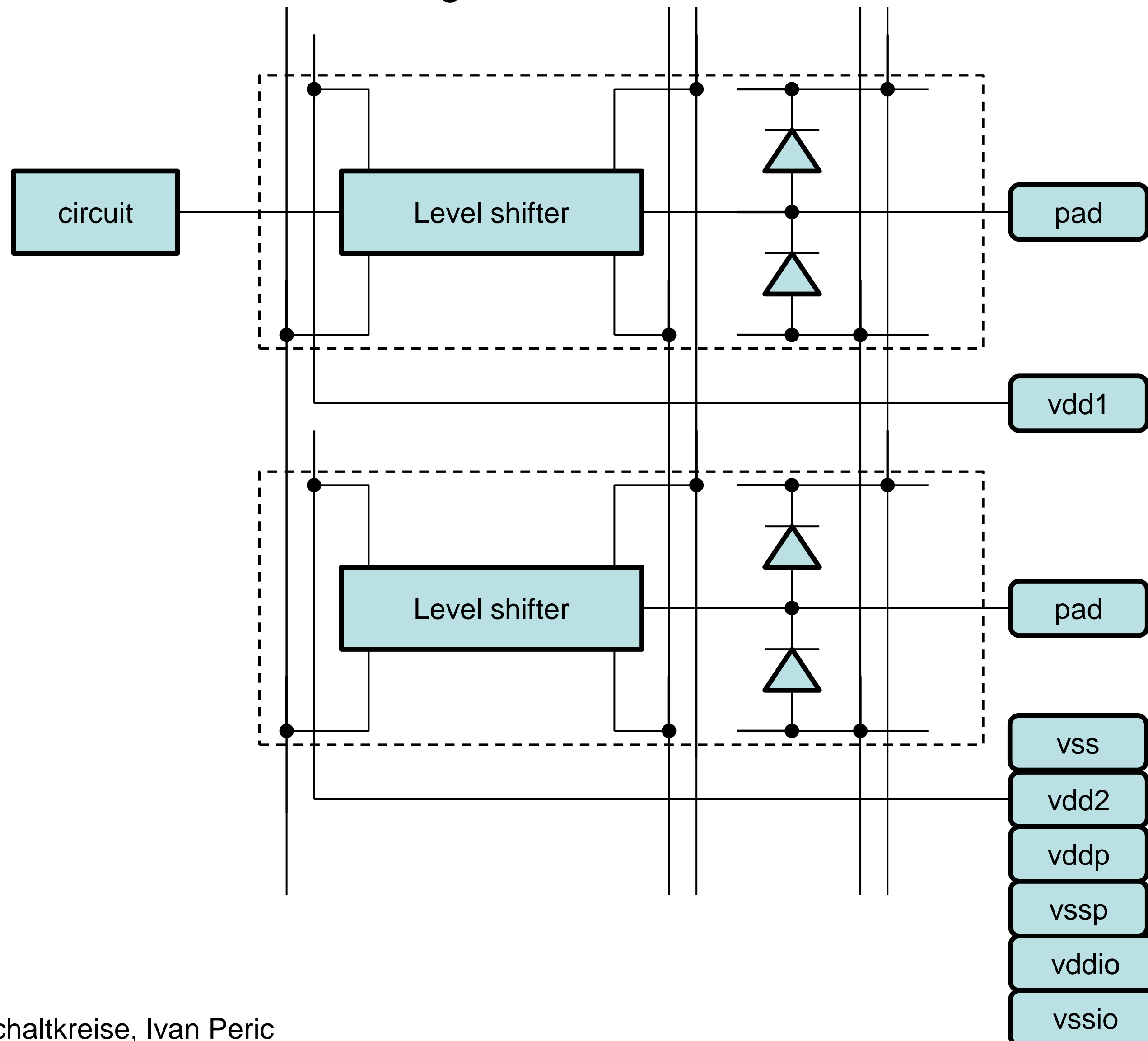


Figure 4-3-A. An Example of a Digital-to-Digital Interface

Power domains

- The slide shows as an example two digital pads that belong to two power domains with the core voltages vdd1 and vdd2. The vdd rail is cut.

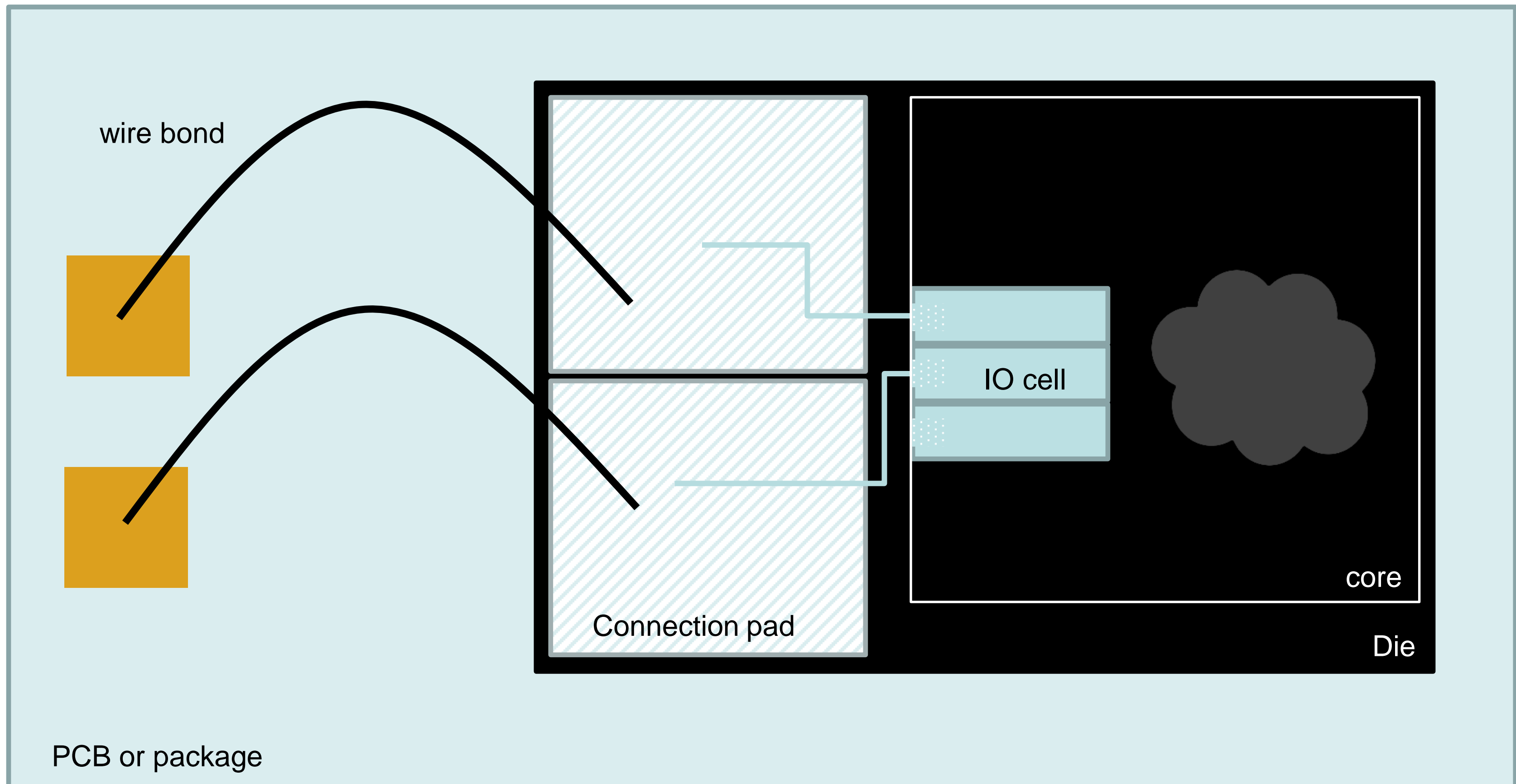


IO-cell and pad placement types

- Slides 18 - 21 explain the IO-cell and pad placement types
- There are 4 possibilities:
- IO cells can be placed outside the core area. In this case, we are talking about the ring IOs.
- IO cells can be also placed inside the core area, this type of cells are called area IOs.
- Connection pads can be placed around the die, which is typical for pads for wire bonding.
- Connection pads can be also distributed all over the chip area for flip-chip connection.

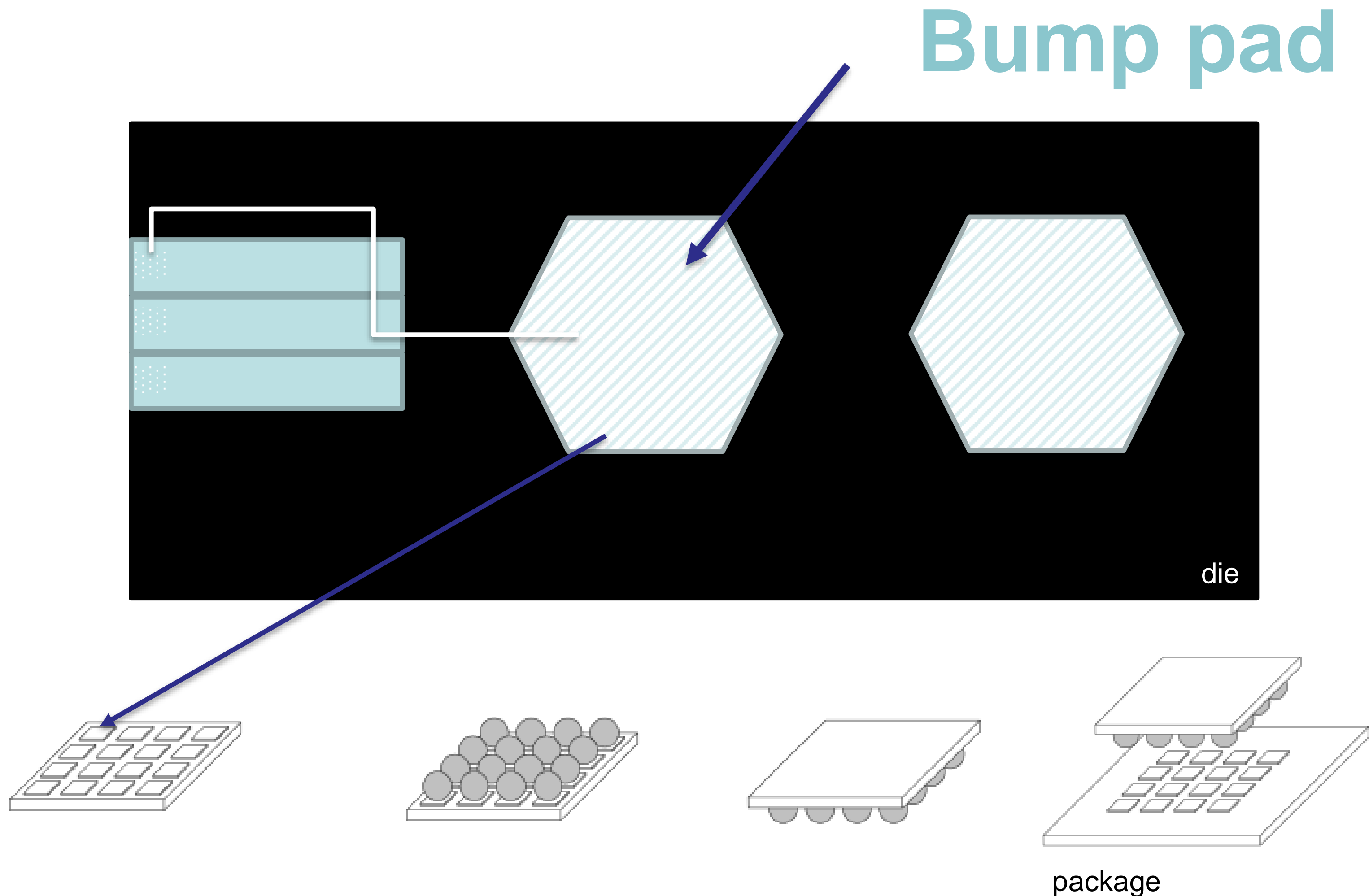
IO-cell and pad placement types

- Example: Area IOs, connection pads for wire bonding



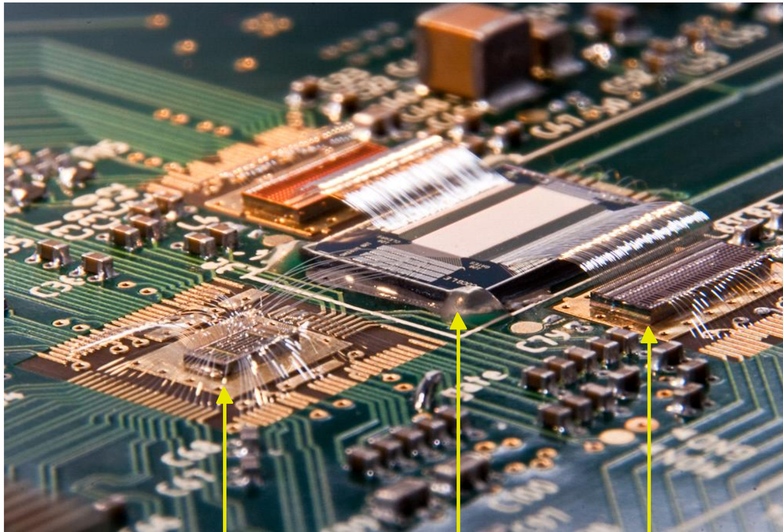
IO-cell and pad placement types

- Example: Ring IOs, connection pads for bump bonding

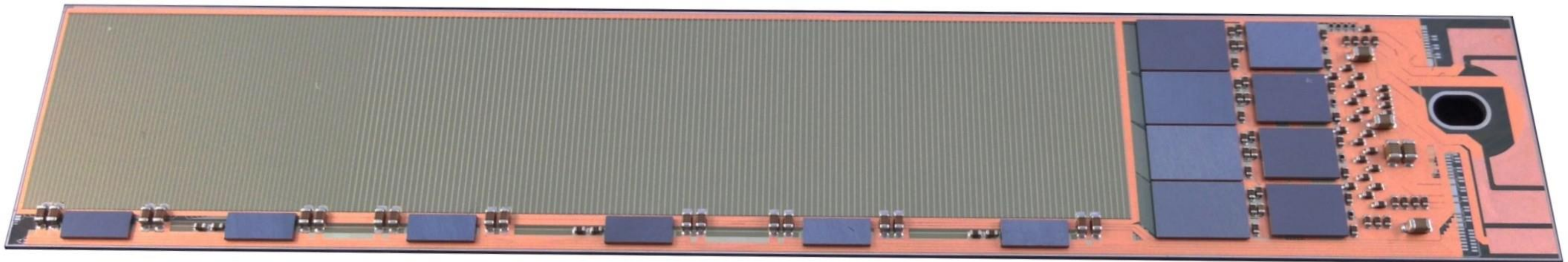


IO-cell and pad placement types

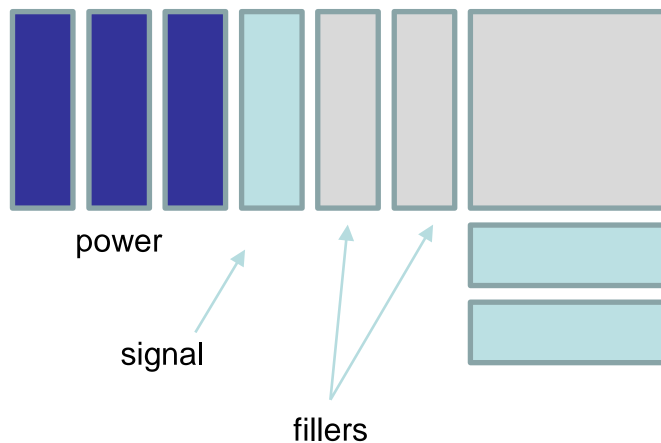
- Example: Complex wire bonding - Belle II detector prototype

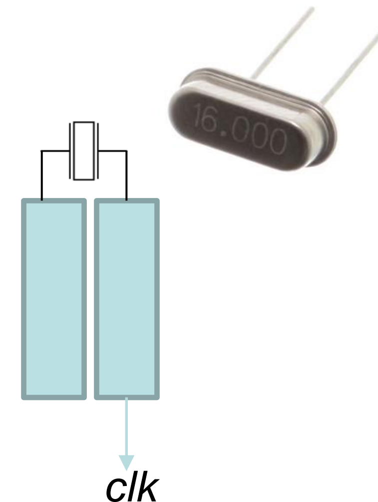


- Example: Bump bonding - Belle II detector final design



- IO Cells can be of various types, depending on the technology library
- Standard types:
  - Power Connections: VDD,VSS,VDDIO,VSSIO
  - Fillers: FILLER1, FILLER10, CORNER
  - Signal Cell for Digital/Analog Signals
- Special Cells:
  - Differential Signal (LVDS)

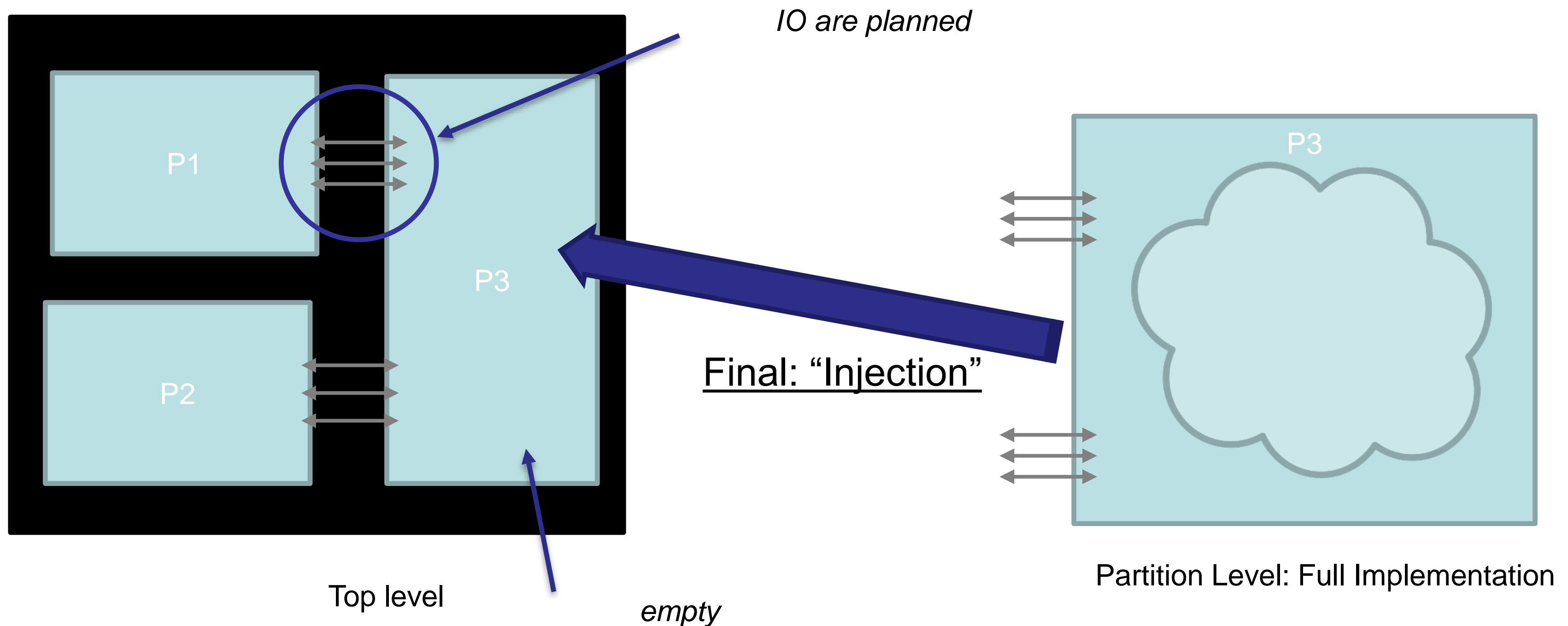




Special Cells example: LVDS input pad

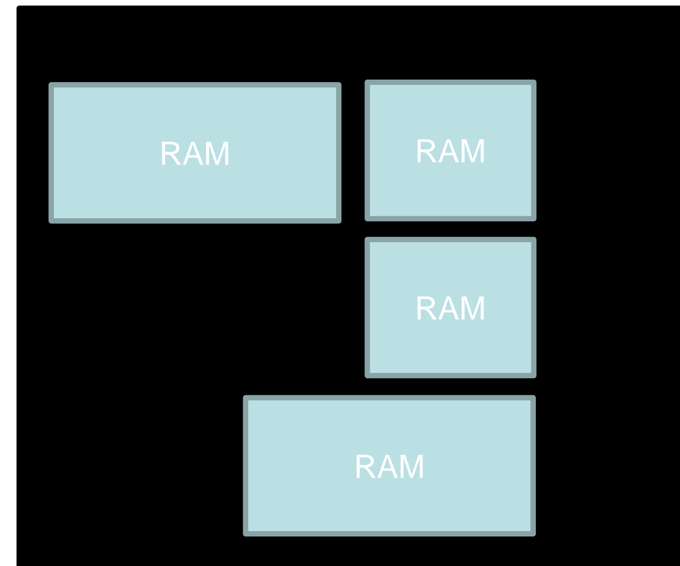
Partitions

- Partitions are used to cut-off the design in smaller subset. Each subset is synthesized and implemented separately.



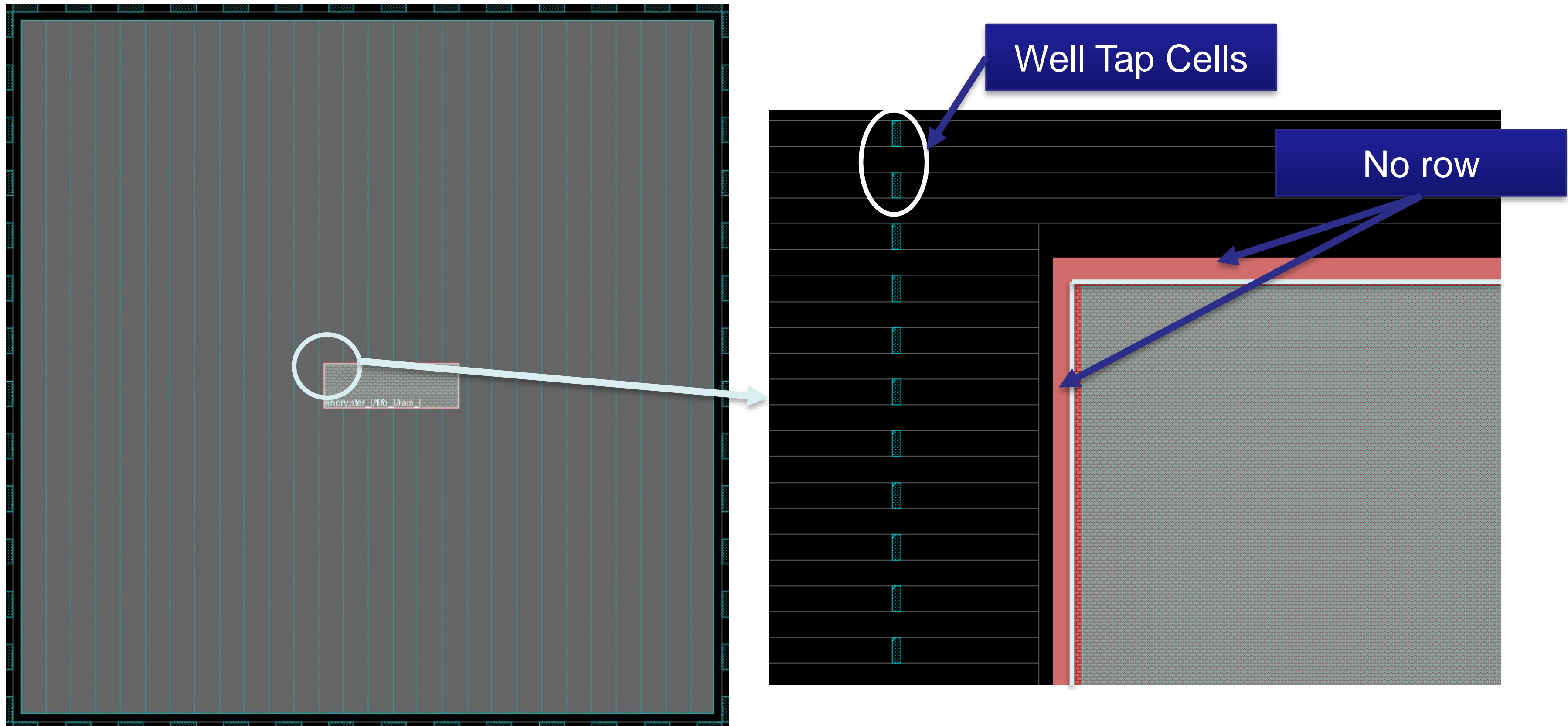
Hard macros or IP blocks

- Hard macros or IP blocks are design blocks similar to partitions, but provided by companies and ready to use. The hard macros are described by lef files that provide size and contact information. Most common hard macros are SRAMs and PLLs. They are placed like any other standard cell.



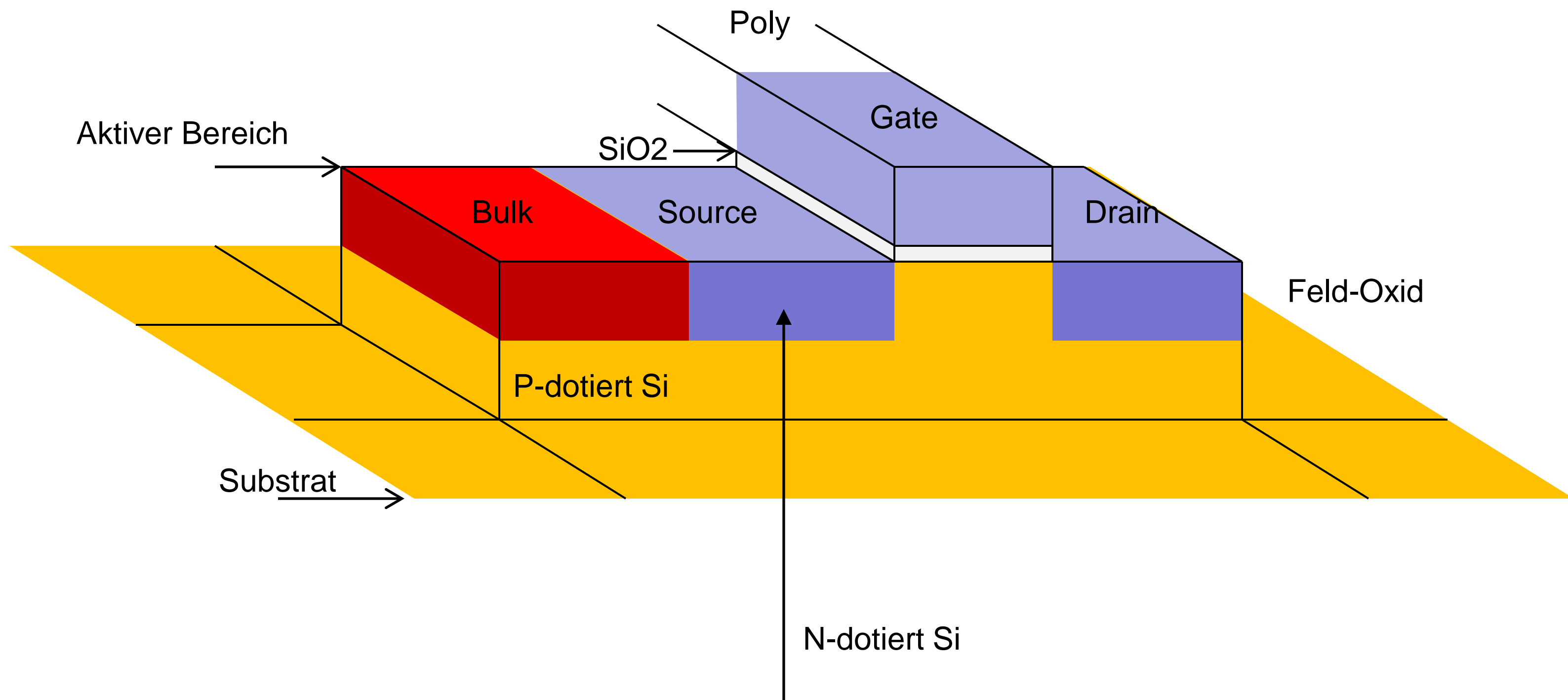
Sometimes design have many rams

Hard macros or IP blocks



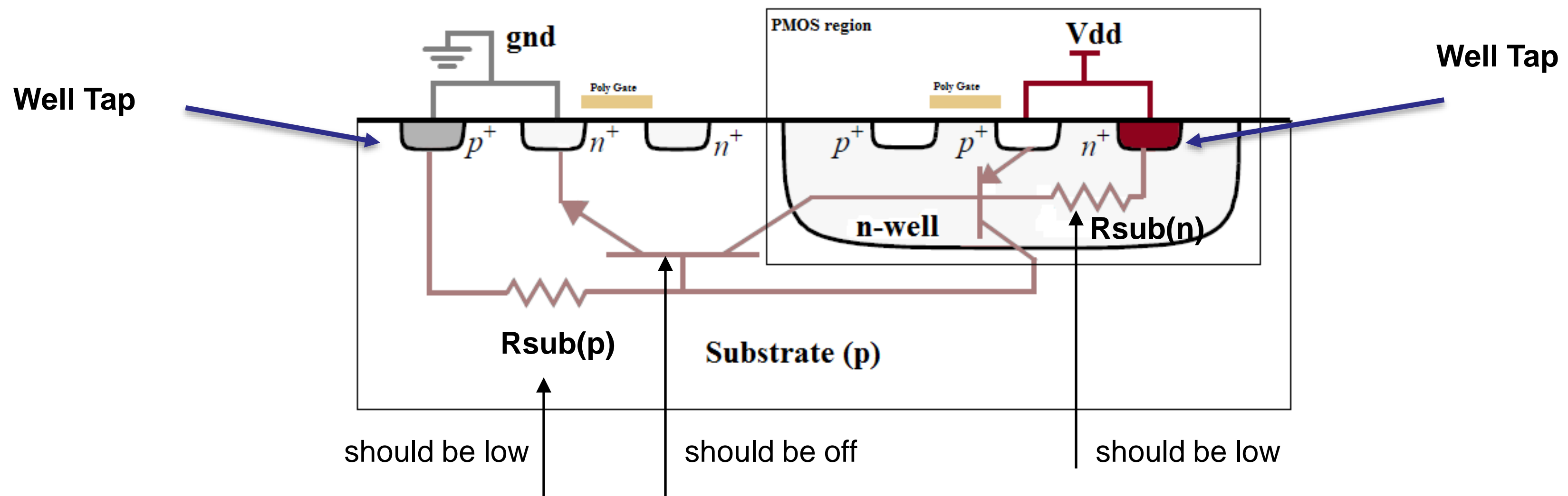
Well- and substrate contacts

- Bulk contact is used to connect the p-type substrate (or p-well) with the negative supply voltage VSS or ground GND. Similarly, a PMOS transistor requires the n-well contact that is shorted with the positive supply voltage VDD.



Well- and substrate contacts

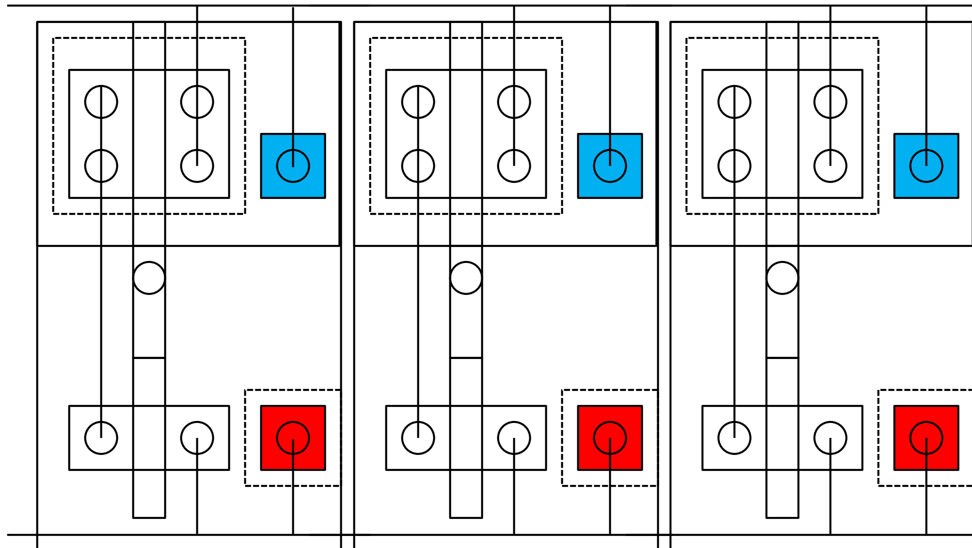
- If the p-well or n-well are not connected, their potentials can change and turn on parasitic bipolar transistors. This may lead to the "latch up", an effect that causes a high current between VDD and VSS and may damage chip.



https://en.wikipedia.org/wiki/Latch-up

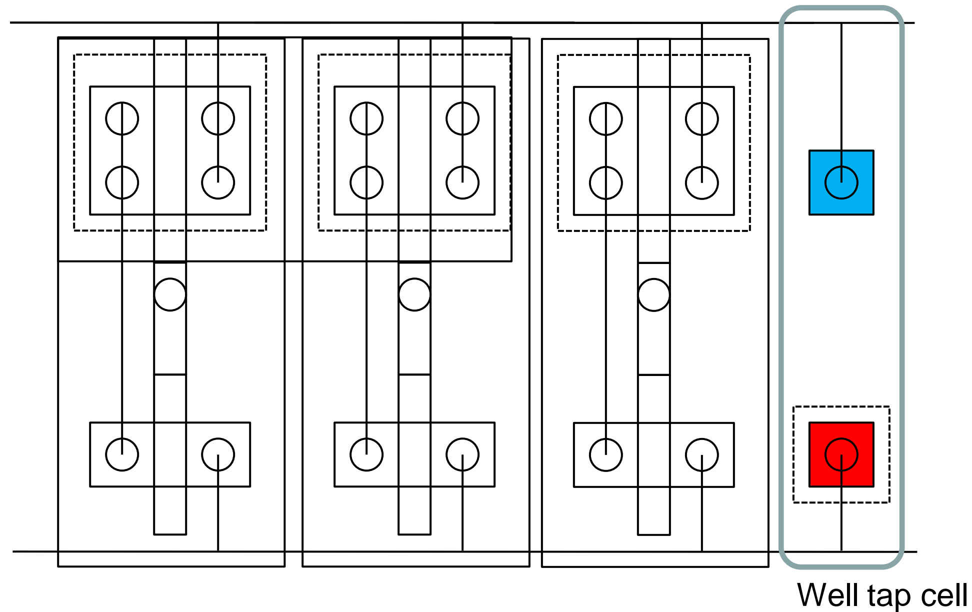
Well- and substrate contacts

- The p-well and n-well contacts can be placed in each standard cell



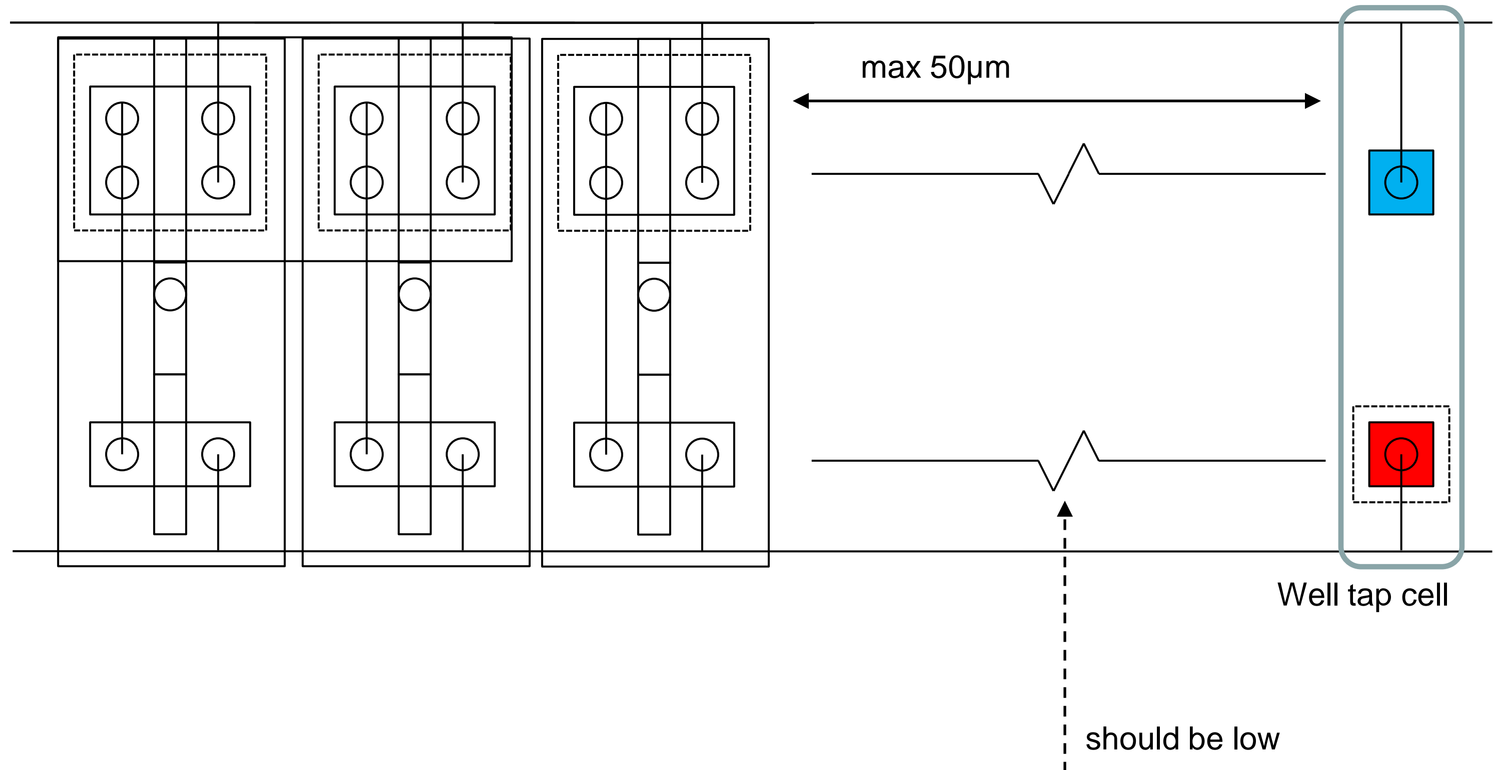
Well- and substrate contacts

- or they can be shared by several standard cells



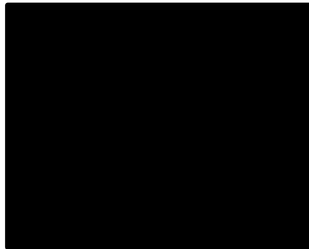
Well- and substrate contacts

- The well tap cell must not be too far from a MOSFET, otherwise the latch up may occur. To assure this the distance between well tap cells must not exceed typically 50µm

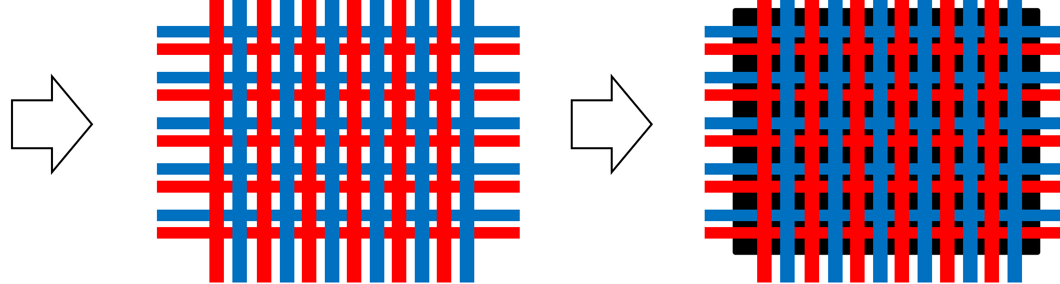


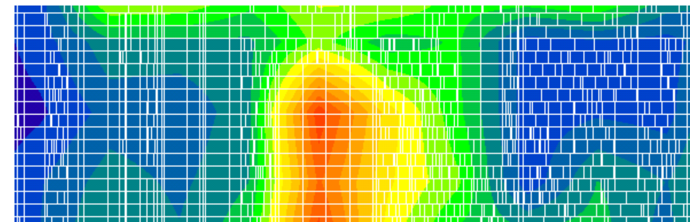
- The next step in P&R is power planning.
- Power connections are important, if they are too thin the voltage drop across the circuit can be large. This can affect the performances. Functional power analysis can be done but the best solution is to make power connection as good as possible and in this way avoid problems.

floorplan



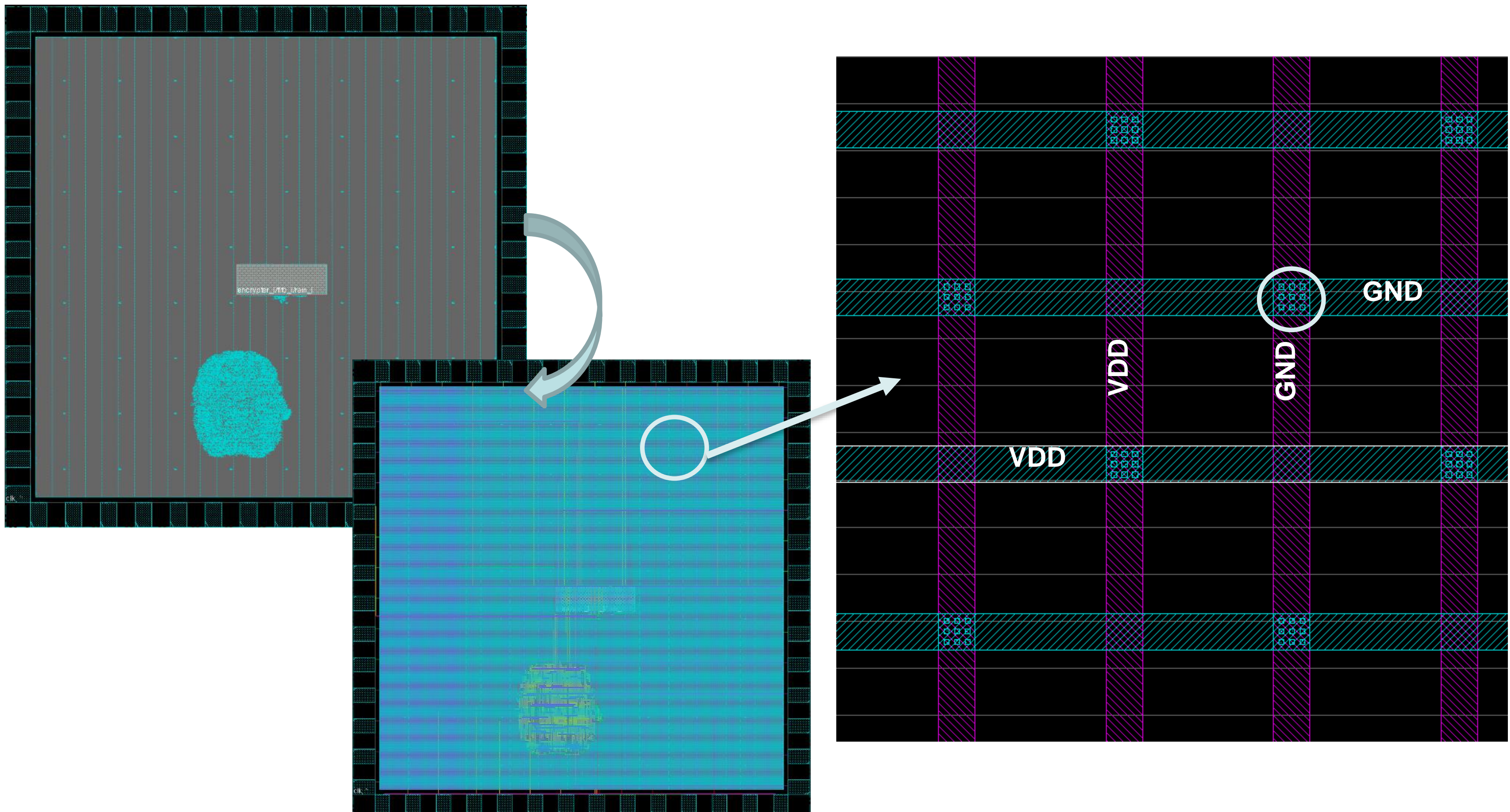
power mesh



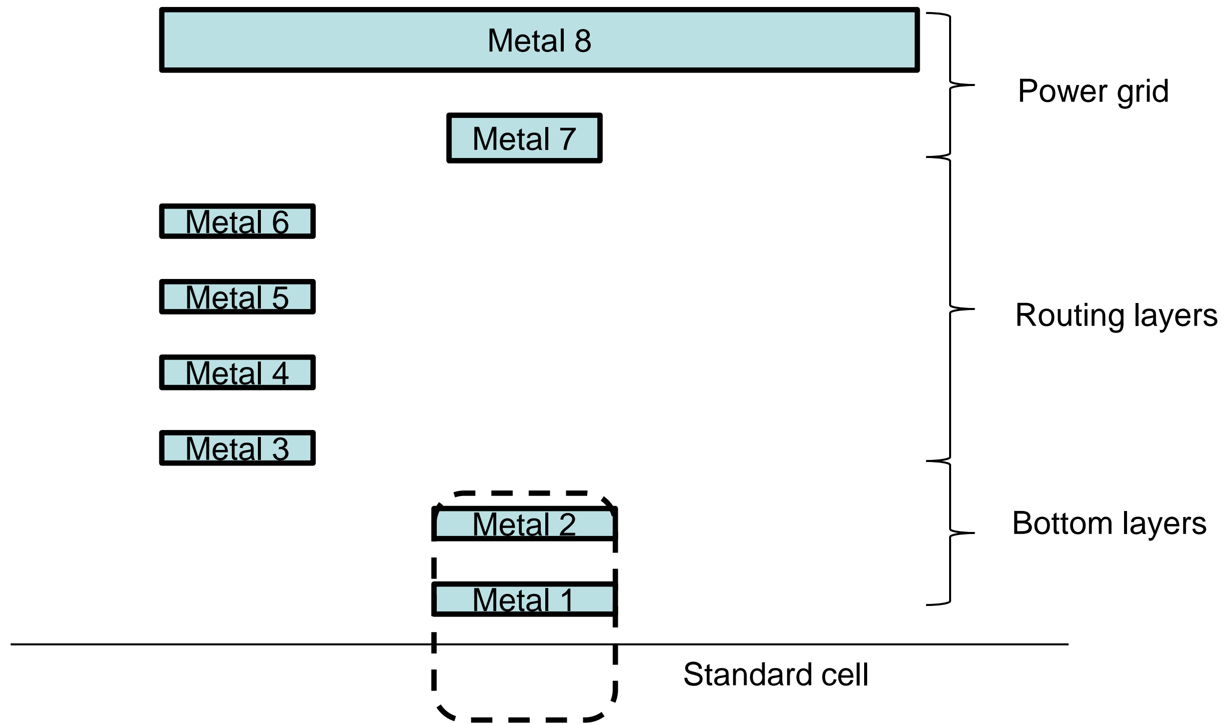


IR Drop example

- Power connections are made using command addStripe to add power stripes on high metal layers

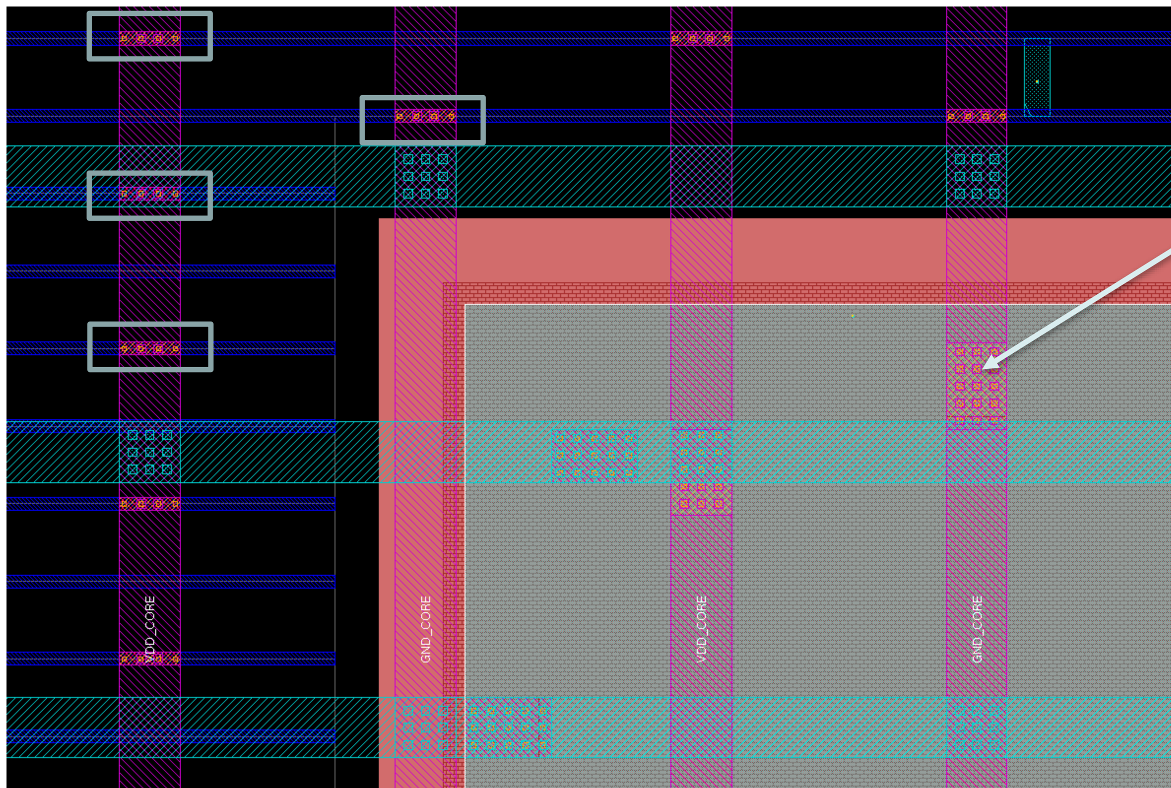


- The cross section through metal layers



Power planning

- Use the command editPowerVia to add VIA down to the power pins of the standard rows and macros. Consult the technology files (like lef) to find out on which layer the power pins are available

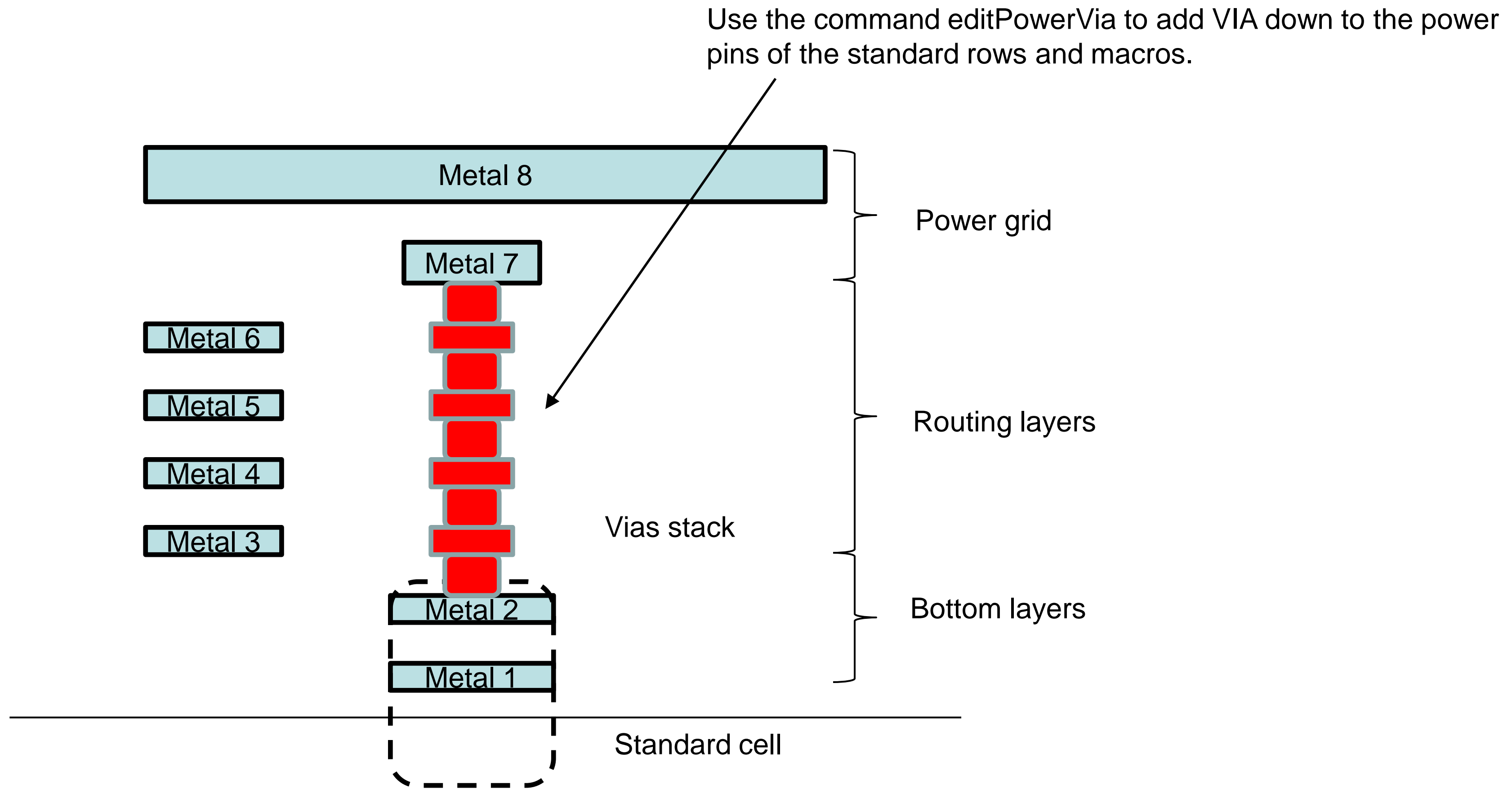


from SRAM LEF

```
PIN GND
  DIRECTION INOUT ;
  USE GROUND ;
  SHAPE RING ;
 PORT
  LAYER ME3 ;
  RECT 0.000 101.520 321.350 103.520 ;
  RECT 0.000 1.000 321.350 3.000 ;
  RECT 319.350 1.000 321.350 103.520 ;
  RECT 0.000 1.000 2.000 103.520 ;
 END
END GND
```

Power planning

- The cross section through metal layers

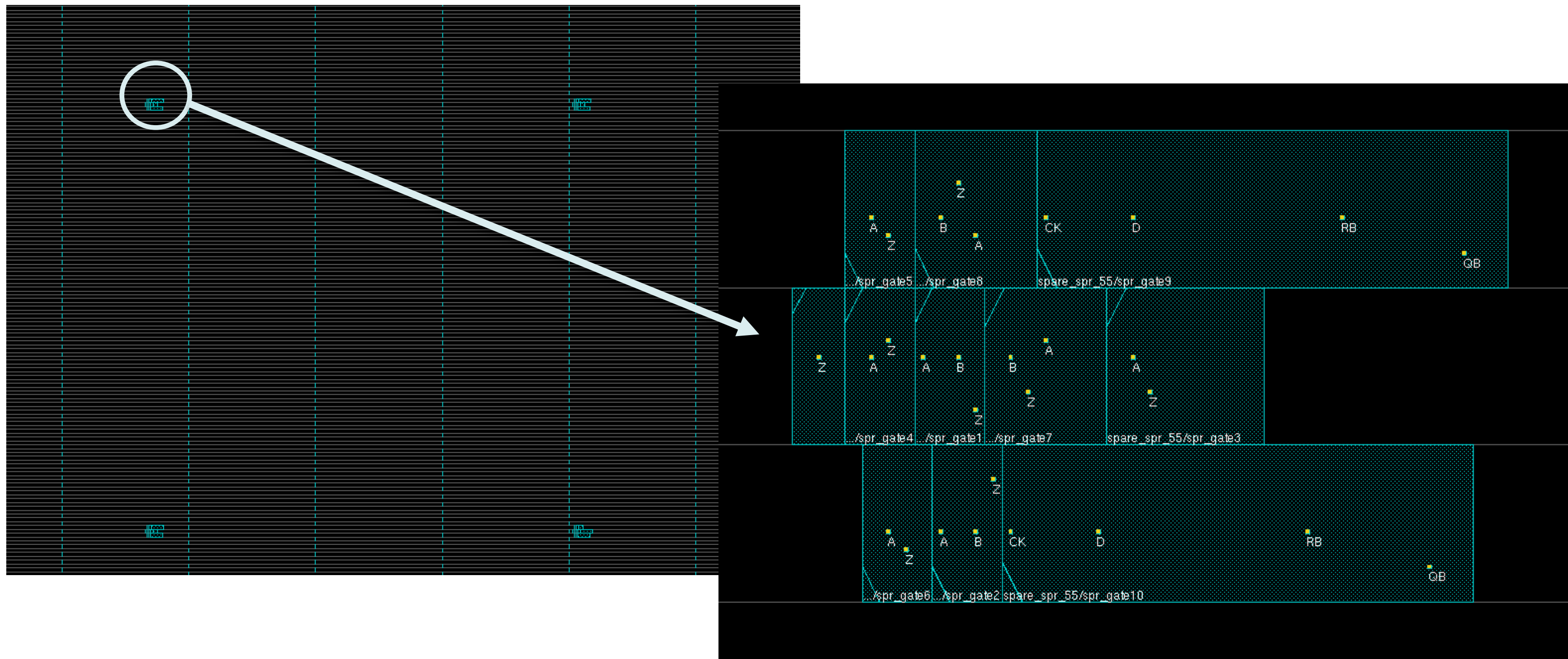


Step 4: Special cell placement

- Special cells like hard macros can be “manually” placed using command placeInstance and by specifying coordinates.
- Code example
- placeInstance <instance_name> <location> <orientation>

Placement of spare cells

- Some clusters of cells can be added on the core area for latter fixes



Step 5: Standard cell placement

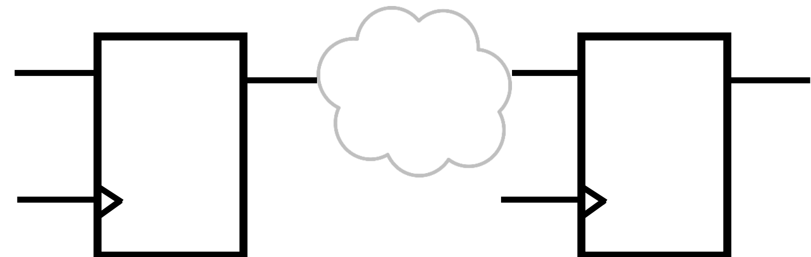
- Standard cell placement is quite trivial, the tool automatically place the cells where can be. The command is placeDesign. First routing is done to give a good idea of the feasibility of the design. The tool tries to optimize timing. This preliminary routing is then deleted. After placing the design is optimized using command optDesign –preCTS
- Code example

```
//Specify the top and bottom layer used for routing

setRouteMode -earlyGlobalMaxRouteLayer 4

setRouteMode -earlyGlobalMinRouteLayer 1

//Place your std cells

place_opt_design

//Optimize your design if necessary

optDesign -preCTS
```

- Where are we?
- Floor planning is done
- Input-outputs are set
- Power structures have been planned
- Standard cells are placed
- What remains?
- Distribute the clock
- Route the design
- Respect design rules for manufacturing
- Keep timing within acceptable margins
- Export the design data for DRC/LVS and production

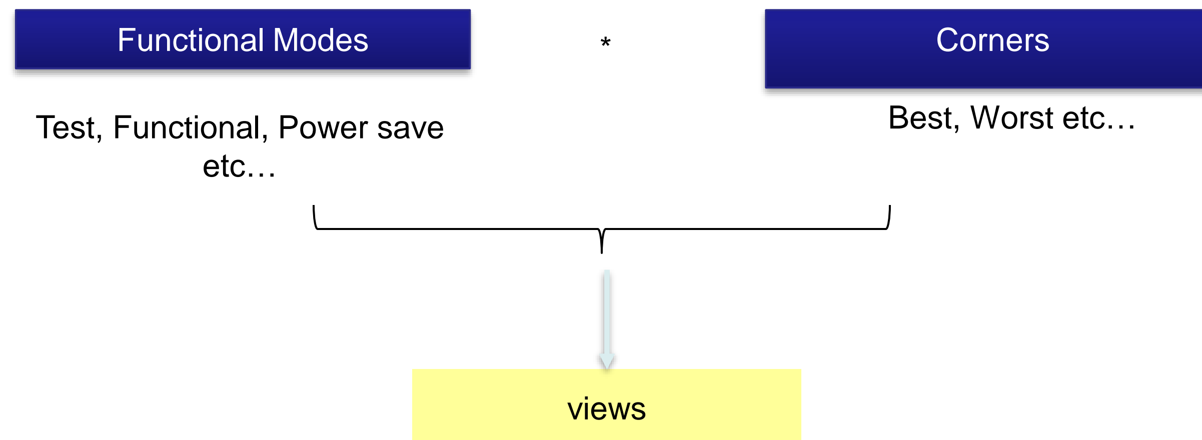
Setup and Hold

- Setup time is the time available until the next clock edge, the input signal of a flip flop should change during this time.
- Hold time is the time that is “not available” after a clock edge, the input signal of a flip flop should stay unchanged during this time.
- There are several fixes for the violations, some of them can be done in design some of them are done by the P&R tool. For instance, the tool can add skew and start the next flip flop (the flip flop that receives signal) later by adding clock buffers to avoid setup violations.
- Concerning hold violations, the tool can start the previous flip flop (the flip flop that generates signal) later or add delay buffers into the data path. This eliminates hold violations.



Multi-Mode Multi-Corner

- The timing of the cells depends on conditions such as supply voltage, temperature and on process variations. Different conditions are called corners. There is different library set for every corner. It can be fast, typical and slow corner for process variations. Each delay corner combines liberty (.lib) for the standard cells and the QRC tech files for the nets.
- The chip may also have different functional modes (e.g. the "functional mode" with fast clock and the "test mode" with slow clock), that are described by sdc files for each mode.
- It is possible to combine corners with modes in the P&R code, such combinations are called views. Further it is possible to tell the tool to use one specific view for the hold time analysis and another for the setup time analysis.

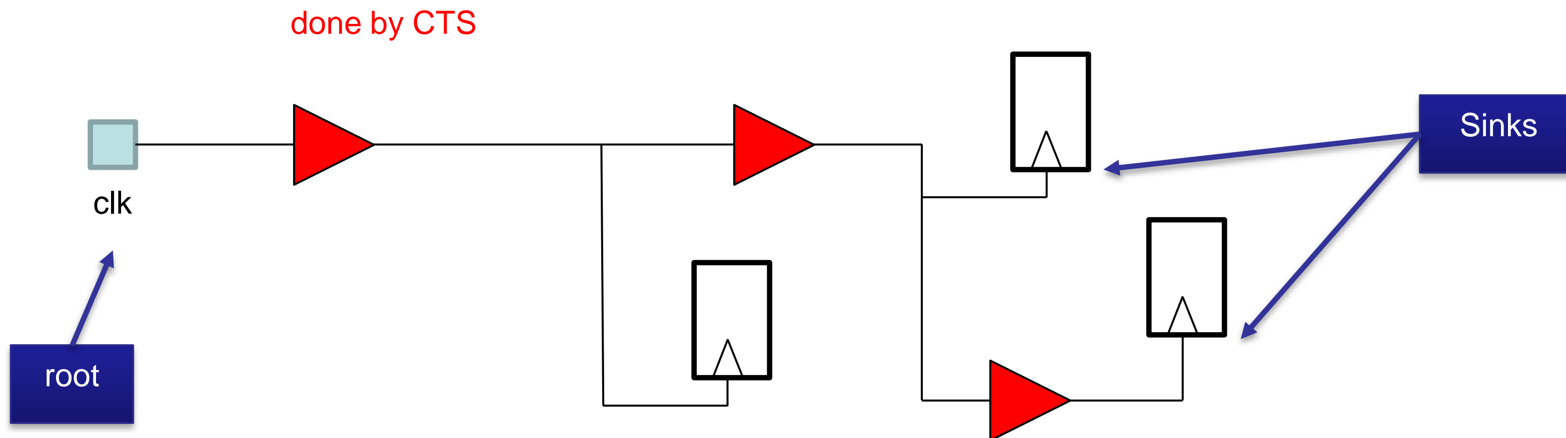


optDesign

- In Cadence tools, the optDesign command is used to optimize timing after each phase: before clock tree synthesis, before routing, after routing.
- Timing fixes are done for setup and hold separately, fixes may alter manufacturing rules (DRC), DRC may alter timing: always re-optimise the design along the implementation flow.

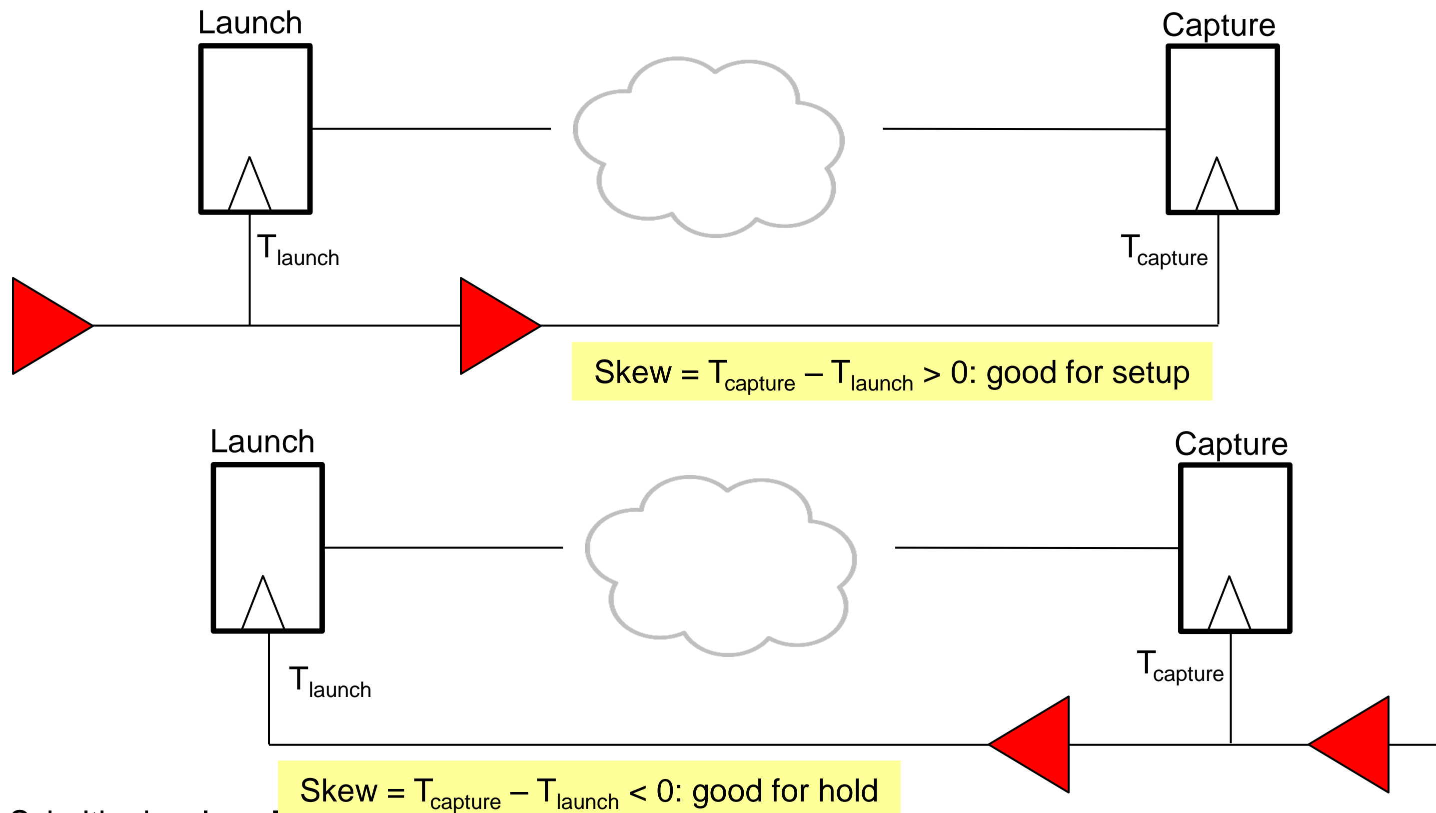
Step 6: Clock tree synthesis

- Clock is a special net, it has one clock source and many “sinks” – the clock receiving flip flops.
- Clock is in the Verilog code just a net, however on chip it has usually a tree-like structure made of clock buffers.
- Clock tree is mandatory for synchronous design: one buffer cannot drive every flip flop. The clock should arrive almost at the same time at each flip flop.
- The aim of the clock tree synthesis is to produce the clock tree and assure that there are no setup and hold violations.



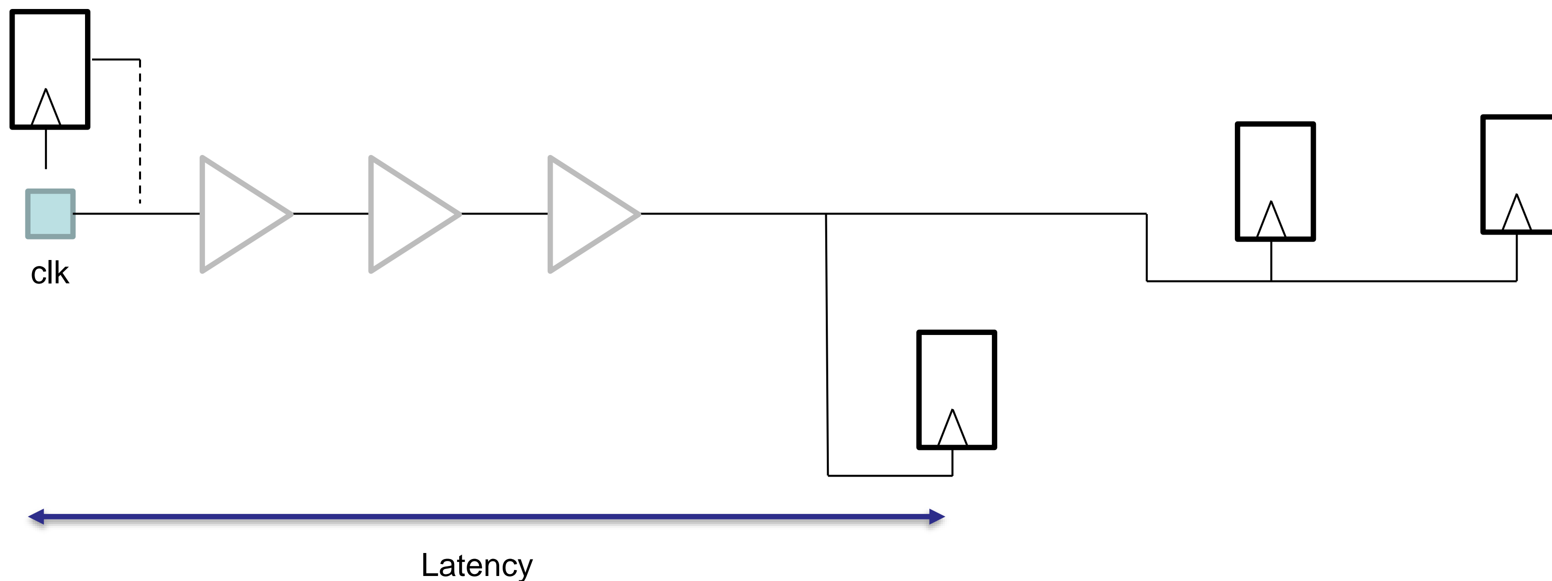
Clock tree synthesis

- Clock skew is the difference between the clock edges at two flip flops, one of them is generating the signal (the launch flip flop) and the other receiving it (the capture flip flop). Clock skew is defined as Skew = $T_{capture} – T_{launch}$.
- It can be positive and negative. Positive skew provides more time time for setup (good for setup).
- Negative skew removes time for setup (bad) and it is good for hold. The original task of clock tree synthesis is to minimize the clock skew.



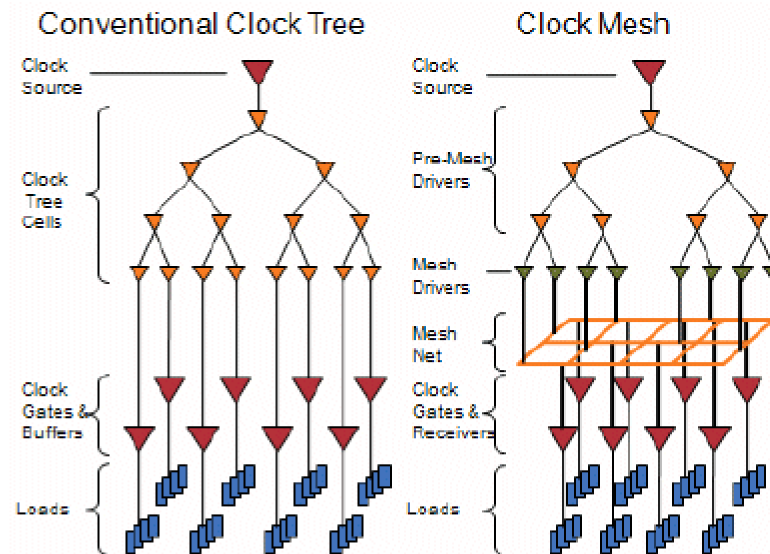
Clock latency

- Clock latency is used to describe the time clock signal needs to reach the flip flops. It is measured from the clock source to the flip flop input.



Tree vs. mesh

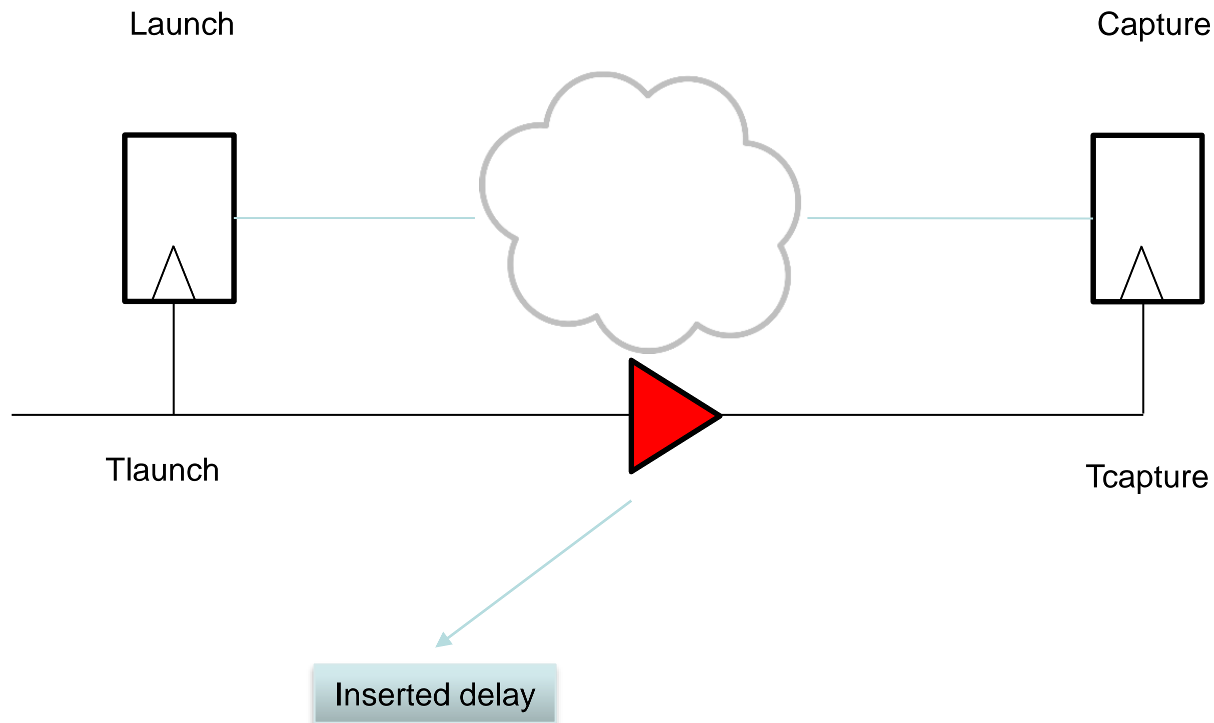
- There are two types of clock routing.
- Standard clock tree is commonly used.
- Clock mesh uses a grid structure to reduce the skew.
- Clock mesh seems to be a better solution when there is a significant on chip variation of standard cell delays.



http://www.design-reuse.com/articles/21019/clock-mesh-benefits-analysis.html

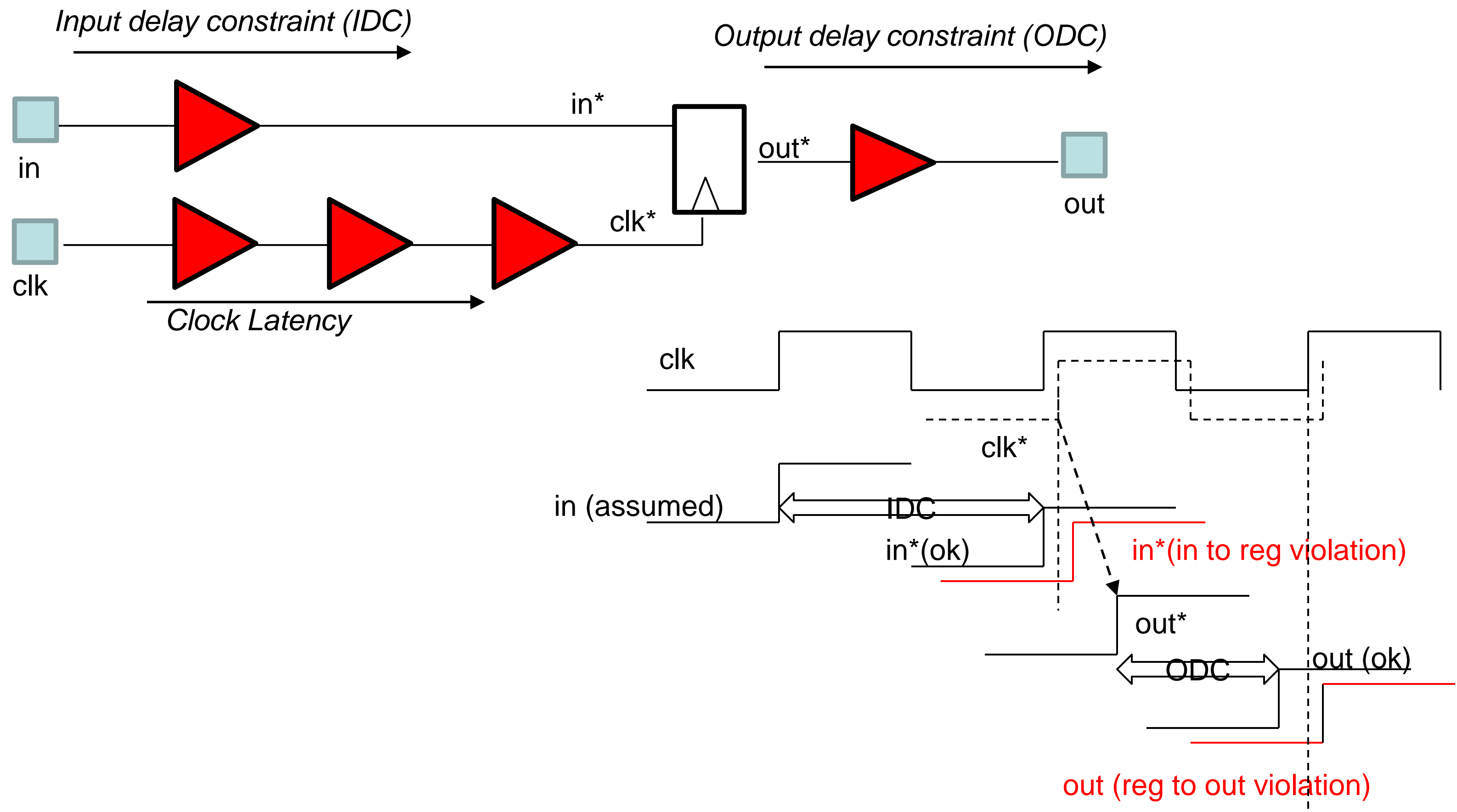
Usefull skew

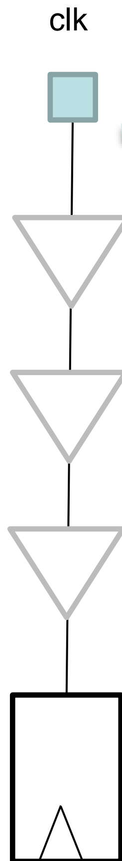
- An advanced option of clock tree synthesis is to use positive/negative skew to improve setup or hold timing. This feature should be used with precaution.



Input and output timing

- Clock tree latency influences the capturing of input signals and generation of output signals. It can cause the input to register and register to output time violations. These violations are not serious as the setup and hold violations on chip because the external circuits can be modified to cope with too late outputs or they can generate the signals for the chip earlier or later.

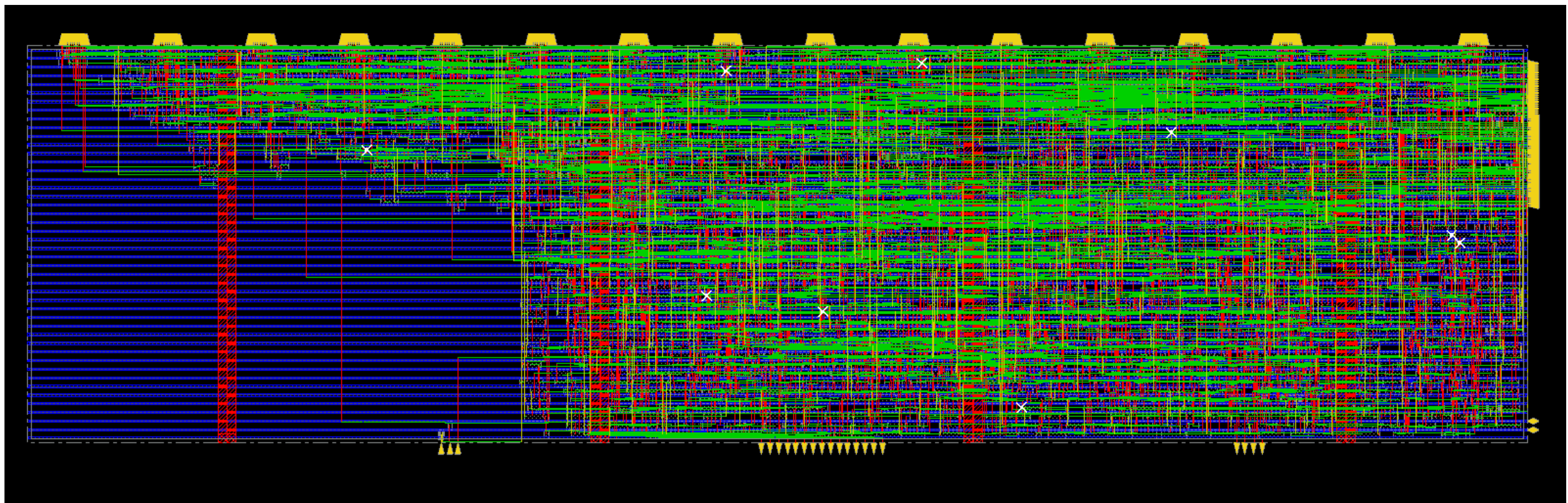





```
######################################################################
Path 1: VIOLATED Setup Check with Pin encrypter_I/rc4_I/K_reg[2]/CK
Endpoint:   encrypter_I/rc4_I/K_reg[2]/D (^) checked with  leading edge of 'clk'
Beginpoint: encrypter_I/rc4_I/i_reg[0]/Q (v) triggered by  leading edge of 'clk'
Path Groups: {reg2reg}
Analysis View: functional_worstHT
Other End Arrival Time          0.735
- Setup                         0.013
+ Phase Shift                   2.500
- Uncertainty                   0.150
= Required Time                 3.072
- Arrival Time                  3.303
= Slack Time                   -0.232
     Clock Rise Edge                      0.000
     = Beginpoint Arrival Time            0.000
     Timing Path:
     +-----------------------------------------------------------------------------------------------------
     |                    Pin                     | Edge |                 Net
     |                                            |   |  |
     |--------------------------------------------+------+----------------------------------------------
     | clk_in/DI                                  |  ^   | clk_int
     | clk_int__L1_I0/A                           |  ^   | clk_int
     | clk_int__L1_I0/Z                           |  ^   | clk_int__L1_N0
     | clk_int__L2_I0/A                           |  ^   | clk_int__L1_N0
     | clk_int__L2_I0/Z                           |  ^   | clk_int__L2_N0
     | clk_int__L3_I0/A                           |  ^   | clk_int__L2_N0
     | clk_int__L3_I0/Z                           |  ^   | clk_int__L3_N0
     | clk_int__L4_I3/A                           |  ^   | clk_int__L3_N0
     | clk_int__L4_I3/Z                           |  ^   | clk_int__L4_N3
     | encrypter_I/rc4_I/RC_CG_HIER_INST58/RC_CGIC_INST/C |  ^   | clk_int__L4_N3
     | K                                          |      |
     | encrypter_I/rc4_I/RC_CG_HIER_INST58/RC_CGIC_INST/G |  ^   | encrypter_I/rc4_I/rc_gclk_180681
     | CK                                         |      |
     | encrypter_I/rc4_I/i_reg[0]/CK              |  ^   | encrypter_I/rc4_I/rc_gclk_180681
     | encrypter_I/rc4_I/i_reg[0]/Q               |  v   | encrypter_I/rc4_I/n_31313
     | encrypter_I/rc4_I/FE_OCPC1786_n_31313/A    |  v   | encrypter_I/rc4_I/n_31313
     | encrypter_I/rc4_I/FE_OCPC1786_n_31313/Z    |  v   | encrypter_I/rc4_I/FE_OCPN1786_n_31313
     | encrypter_I/rc4_I/FE_OFC385_n_31313/A      |  v   | encrypter_I/rc4_I/FE_OCPN1786_n_31313
     | encrypter_I/rc4_I/FE_OFC385_n_31313/Z      |  v   | encrypter_I/rc4_I/FE_OFN385_n_31313
```

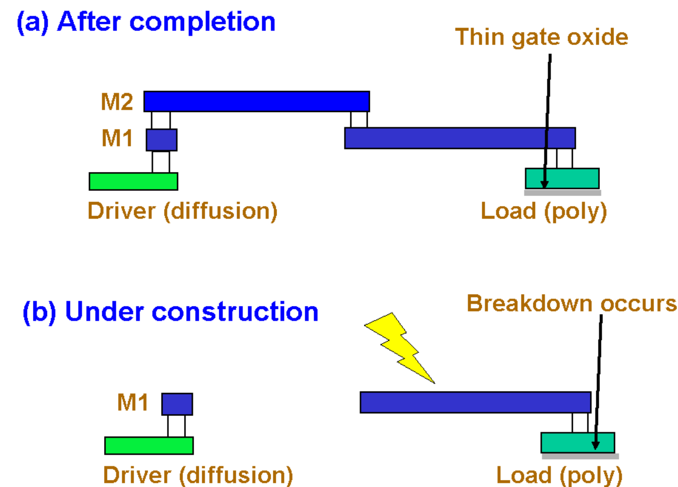
Step 7: Routing

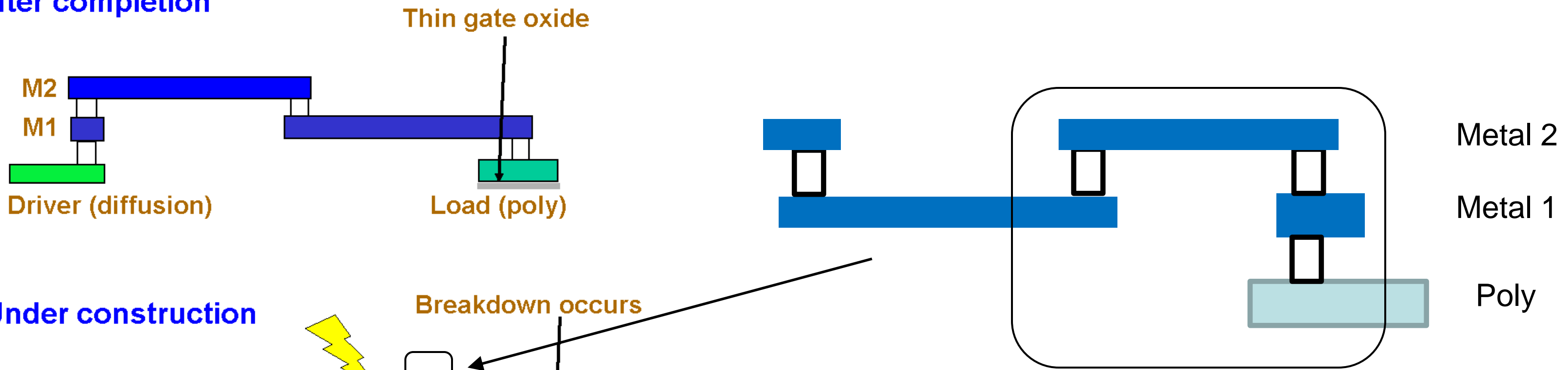
- During routing process following is done:
- Data paths are routed with accurate calculation (extraction) of line capacitances and resistances. During routing crosstalk and signal integrity are taken into account. Design rules are followed. Contacts between metal layers (the vias) are optimized by using double cuts. Antenna violations are fixed. The routing procedure can optimize the timing (timing driven routing) or try to avoid signal errors due to capacitive crosstalk (signal integrity driven routing).

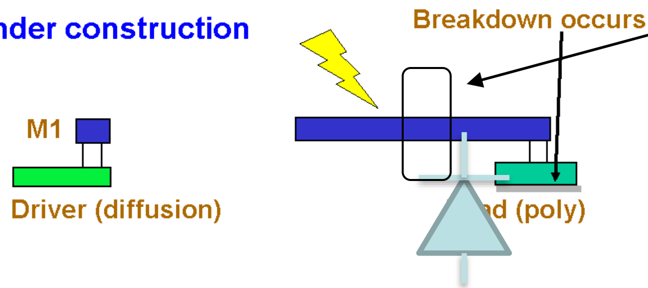


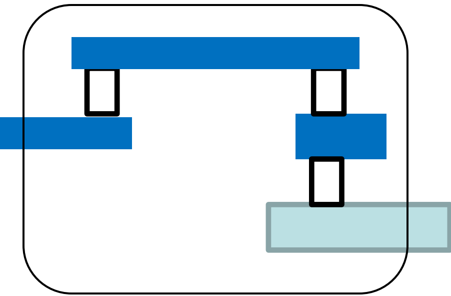
Antenna effect

- Antenna effect is a risk during manufacturing process. During production of routed lines, especially ion etching or polishing, charge is generated which increases the voltage of the lines. The high voltage can damage the transistor gates that are connected to the lines. IC would not be working if it happens. The problem can be fixed by adding electrostatic discharge protection diodes (antenna cells) or by making the metal bridges. A command verifyProcessAntenna will perform antenna violation fixing



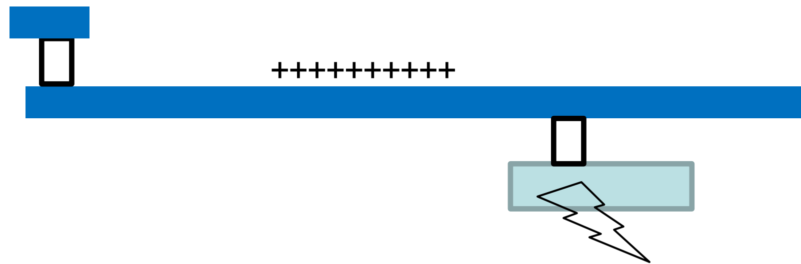






Antenna diode

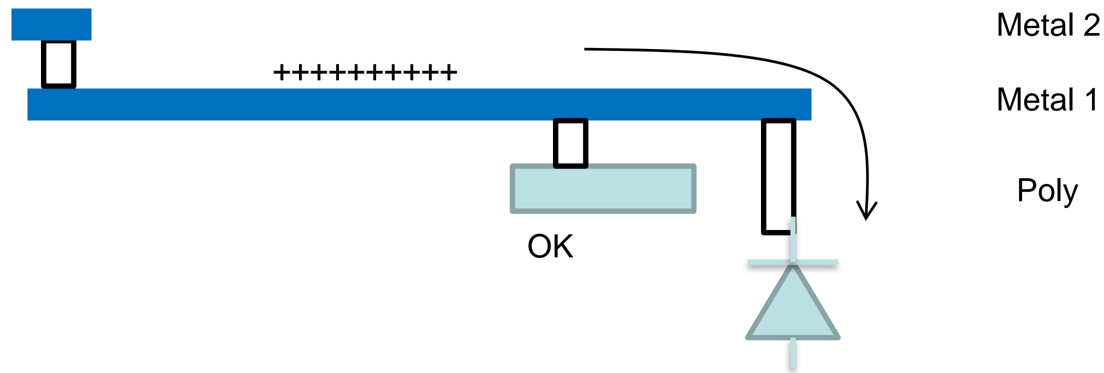
Bridge

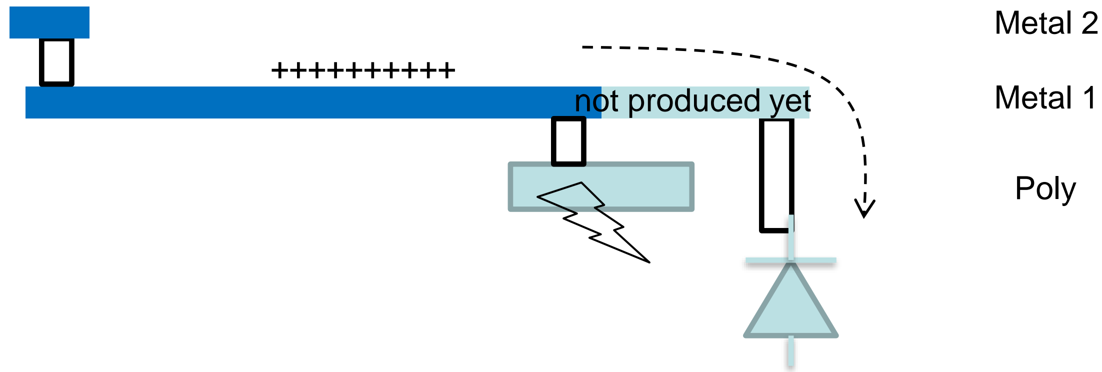


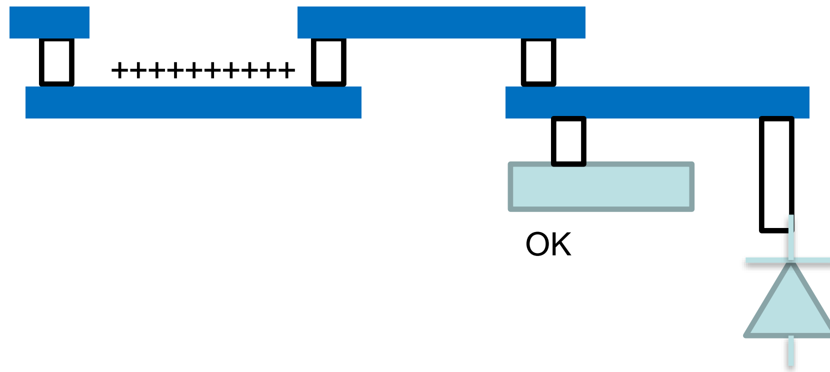
Metal 2

Metal 1

Poly


Metal 2
++++++++++
Metal 1
Poly
OK


++++++++++
not produced yet
Metal 2
Metal 1
Poly



Metal 2

Metal 1

Poly

Step 8: Signoff

- Some final steps should be done before manufacturing
- Unused area is filled with filler cells with command addFiller.
  - The filler cell names must be written in the code, they are found in technology documentation
- Verify and fix DRC
- Recalculate timing and fix setup and hold violations
- Perform metal filling to fix density rules
- Write out the GDS file
- Perform layout versus schematic checks