

Digital ASIC Backend Process Tutorial for Frontend Engineers

Rev 1.0

SD-RTL-DGN in San Diego, CA

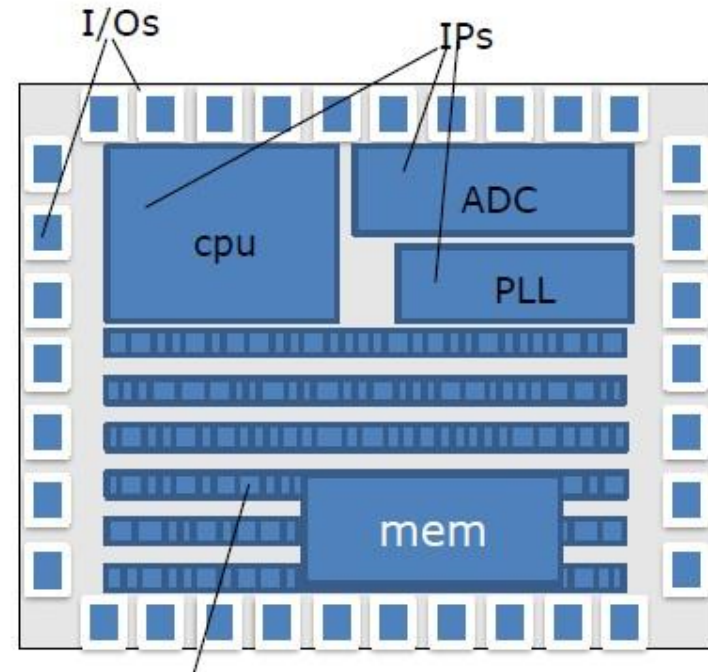
Visit us: <http://www.valpont.com/members/sd-rtl-dgn/>

Overview

- Standard cell based IC design with hard macros such as analog PLL/ADC and memories (CPU is implemented in standard cells too)
- GDS(Geometry Data Standard), chip layout sent to foundry, contains info about transistors and interconnects/wires, binary file.



GDS

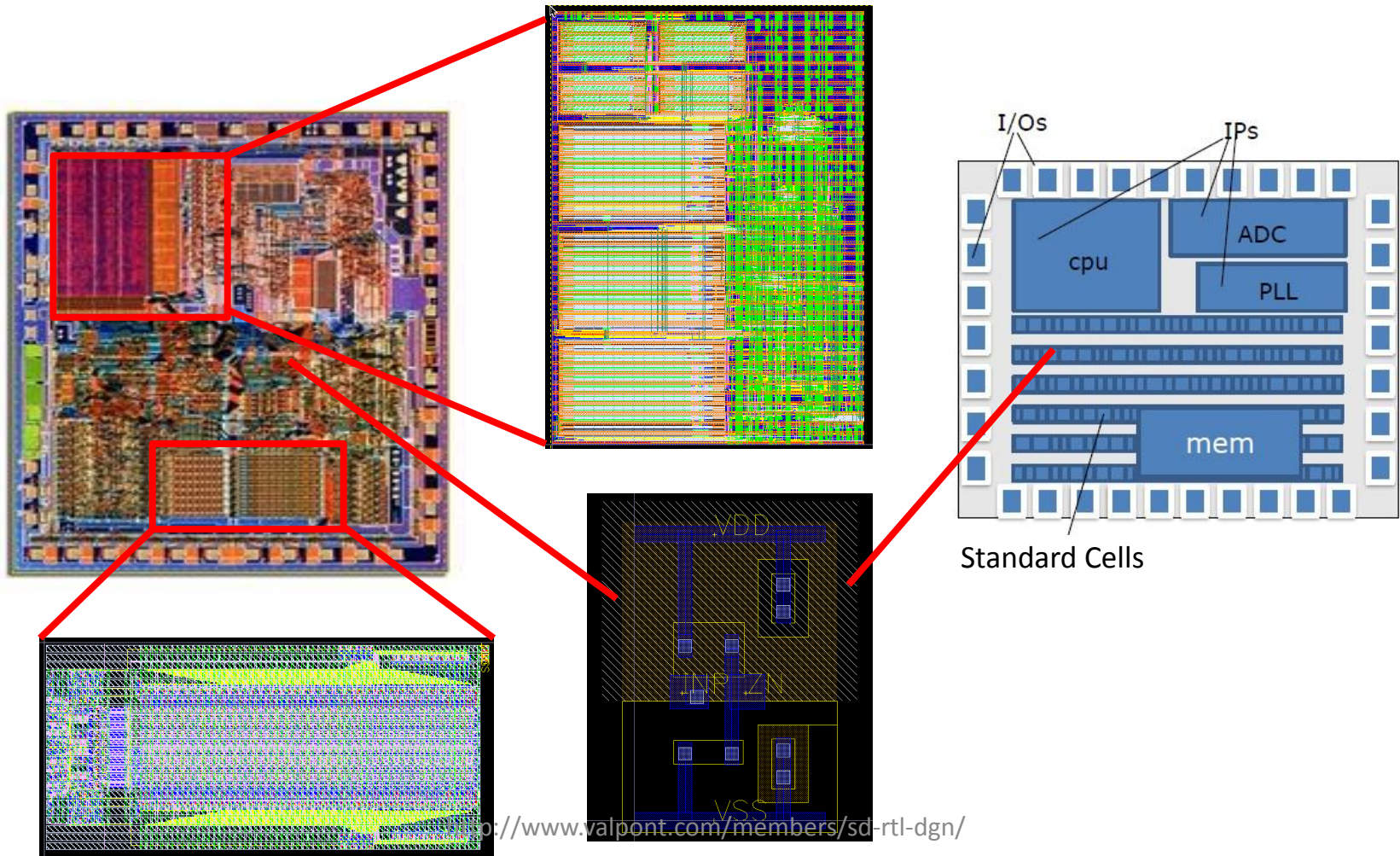


Standard Cells

Concept Diagram

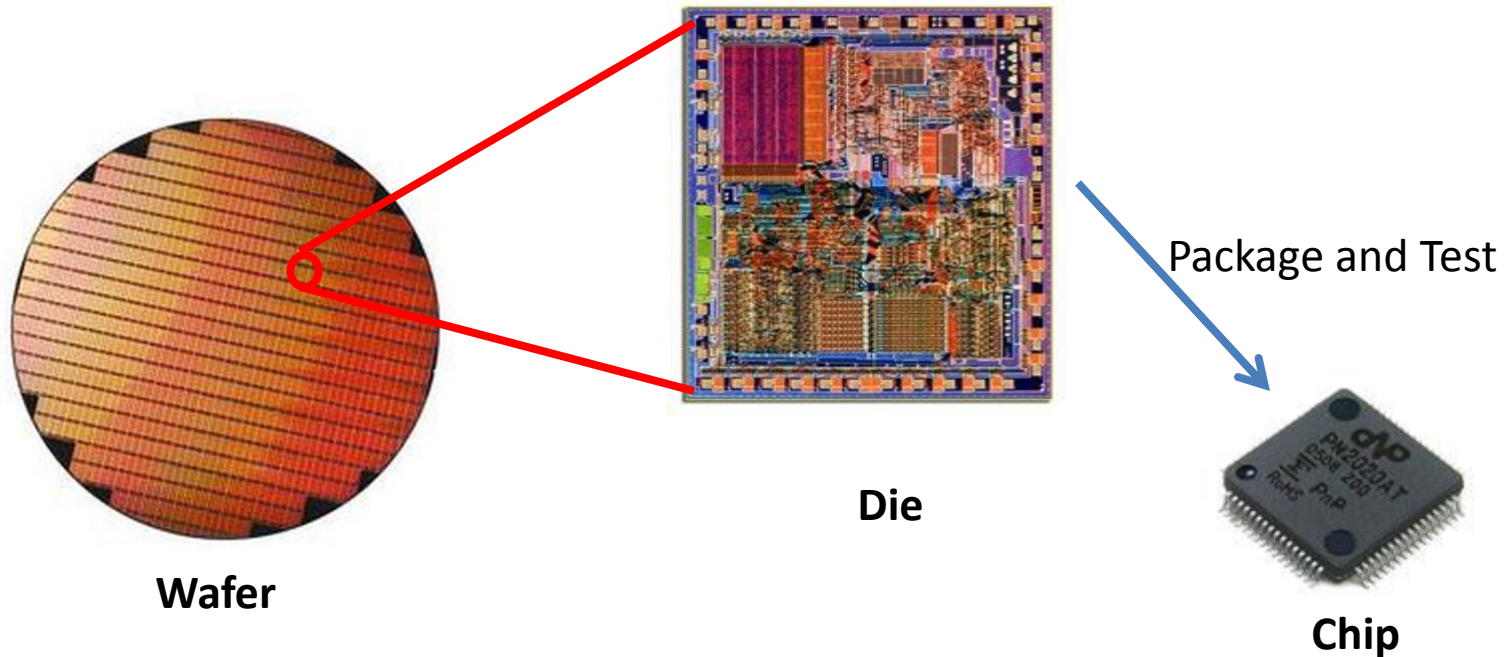
Overview

- Std cells, memories, macros in GDS



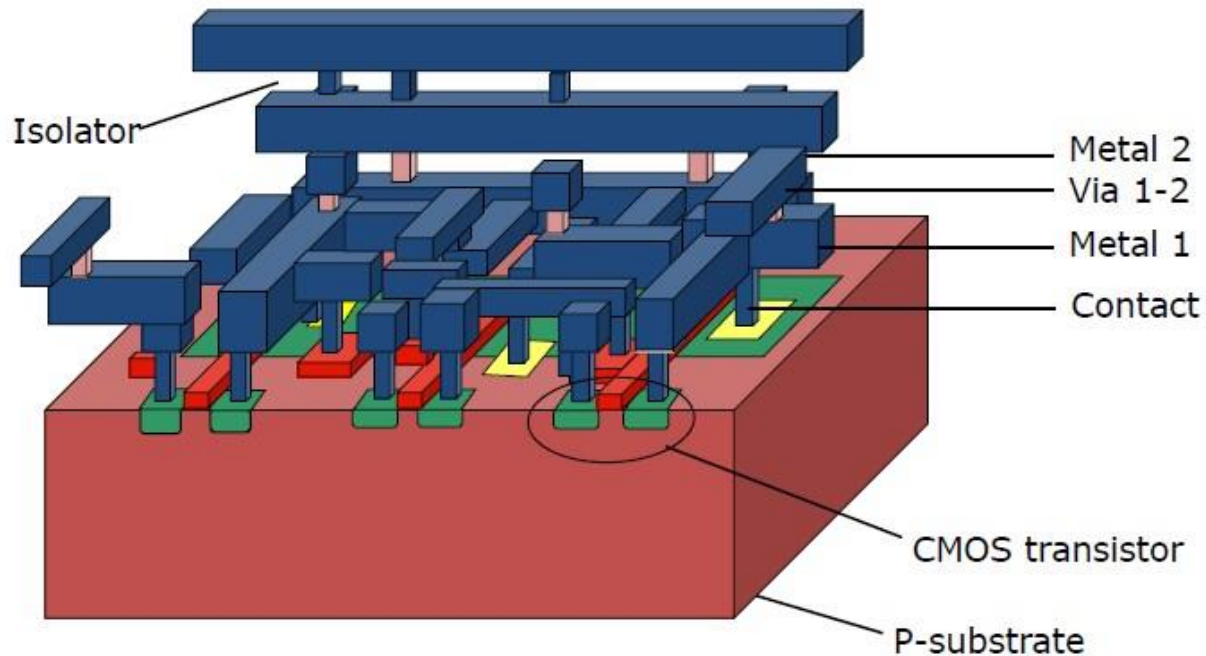
Overview

- From foundry to customer - wafer, die, and chip

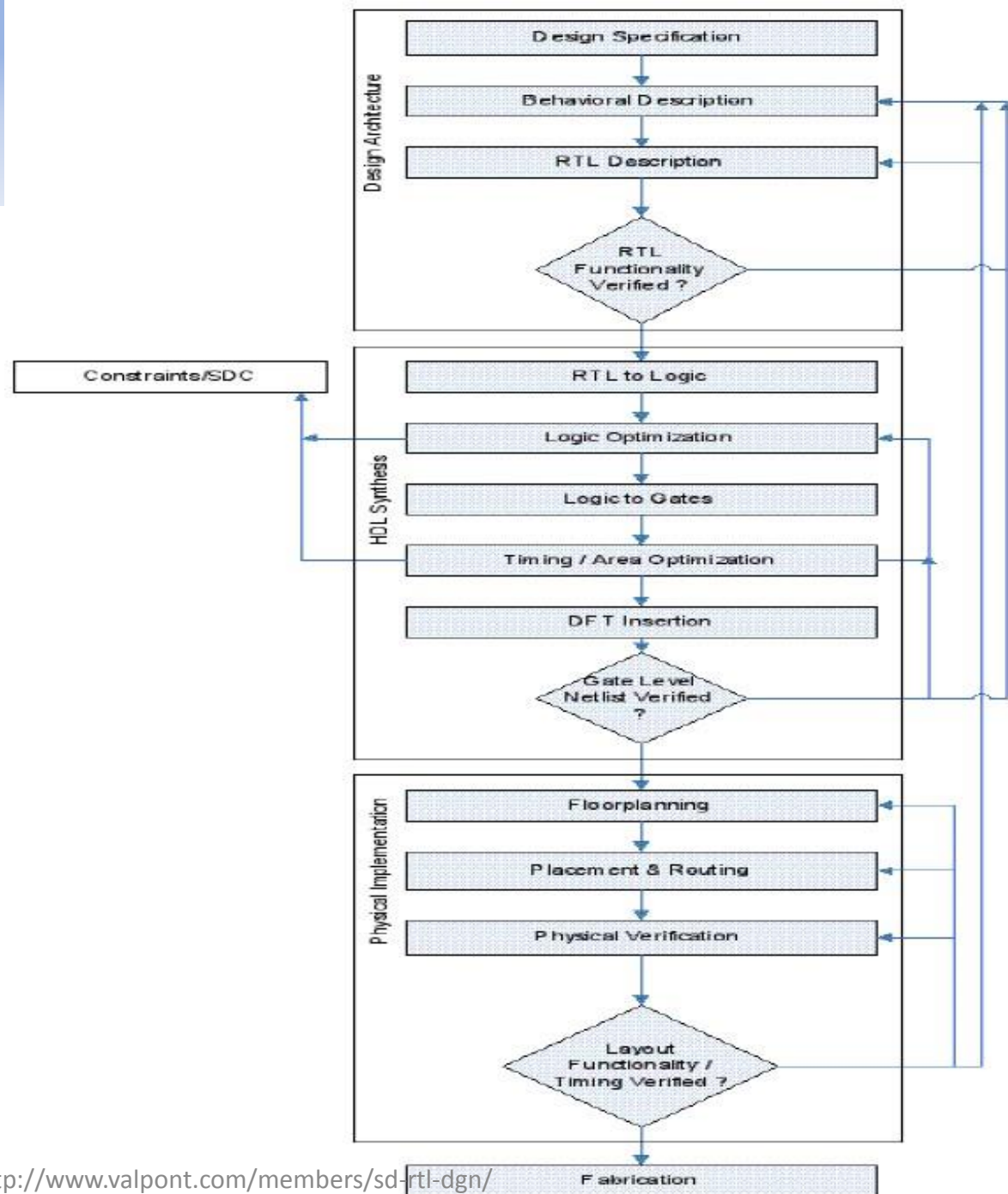


Overview

- Zoom in: 3D drawing of transistor/base layer and metal layers



- Overall ASIC design flow from frontend RTL design, verification to backend process (Synthesis, floorplan, PnR, timing)



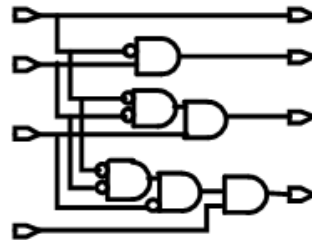
Synthesis from RTL to Netlist

Synthesis = Translation + Optimization + Mapping

```
residue = 16'h0000;  
if (high_bits == 2'b10)  
    residue = state_table[index];  
else  
    state_table[index] = 16'h0000;
```

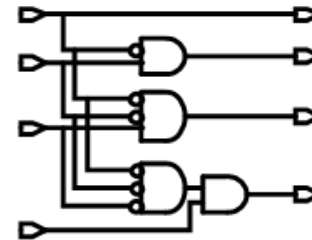
Translate

HDL Source



**Generic Boolean
(GTECH)**

Optimize + Map



Target Technology

Synthesis Timing

- Use wire-load model for timing calculation

Example: Standard Format

```
Name           : 160KGATES
Location        : ssc_core_slow
Resistance      : 0.000271 ← R per unit length
Capacitance    : 0.00017 ← C per unit length
Area           : 0
Slope          : 50.3104 ← Extrapolation slope
Fanout   Length
-----
1      31.44
2      81.75
3     132.07
4     182.38
5     232.68
Time Unit      : 1ns
Capacitive Load Unit : 1.000000pf
Pulling Resistance Unit : 1kilo-ohm
```

RTL vs Netlist Equivalence Check

- Synopsys Formality and Cadence Conformal/LEC
- Mixed usage: use LEC for RTL vs Netlist and formality for netlist vs netlist

```
module top (a,b,c,d,e,y);  
input a,b,c,d,e;  
output y;  
  
wire net;  
assign net = (a==b)?c:d;  
assign y = (net==d)?net:e;  
endmodule
```

```
module top ( a, b, c, d, e, y );  
input a, b, c, d, e;  
output y;  
wire n6, n7, n8, n9;  
  
A022X1 U6 ( .IN1(d), .IN2(n6), .IN3(n7), .IN4(e), .Q(y) );  
NOR2X0 U7 ( .IN1(n8), .IN2(n9), .QN(n7) );  
INVX0 U8 ( .INP(c), .ZN(n8) );  
OR3X1 U9 ( .IN1(c), .IN2(e), .IN3(n9), .Q(n6) );  
XOR2X1 U10 ( .IN1(b), .IN2(a), .Q(n9) );  
endmodule
```



Formal Check

Physical Process Flow

- Backend process, place and route, from netlist to GDS
- Tools
 - Synthesis: Synopsys DC/PC
 - PnR: Synopsys Astro, Cadence Encounter/icfb
 - DRC&LVS: Cadence Dracula, Mentor Calibre
 - STA: Synopsys Primetime

Floorplanning

Placement

Clock Tree Insertion (CTS)

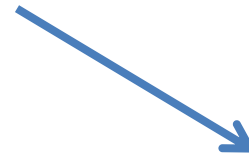
Routing

Finishing

Physical Implementation

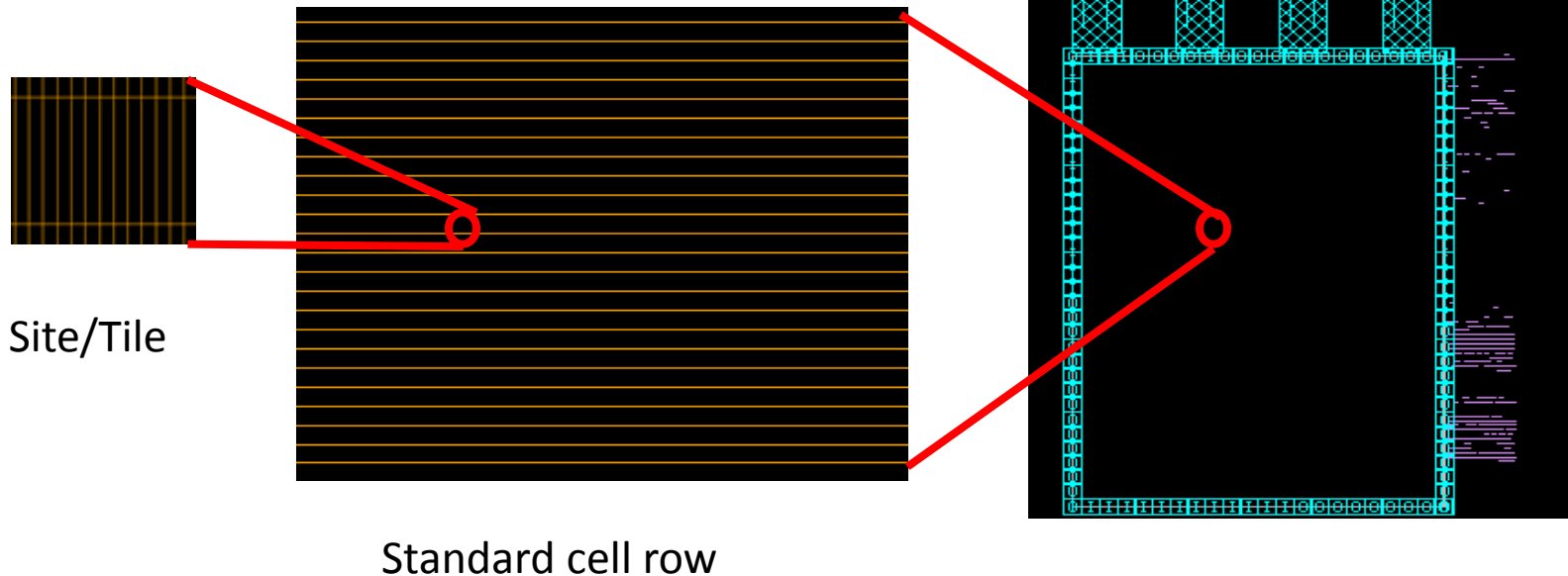
- From netlist to GDS

```
module top ( a, b, c, d, e, y );  
  input a, b, c, d, e;  
  output y;  
  wire n6, n7, n8, n9;  
  
  A022X1 U6 ( .IN1(d), .IN2(n6), .IN3(n7), .IN4(e), .Q(y) );  
  NOR2X0 U7 ( .IN1(n8), .IN2(n9), .QN(n7) );  
  INVX0 U8 ( .INP(c), .ZN(n8) );  
  OR3X1 U9 ( .IN1(c), .IN2(e), .IN3(n9), .Q(n6) );  
  XOR2X1 U10 ( .IN1(b), .IN2(a), .Q(n9) );  
endmodule
```



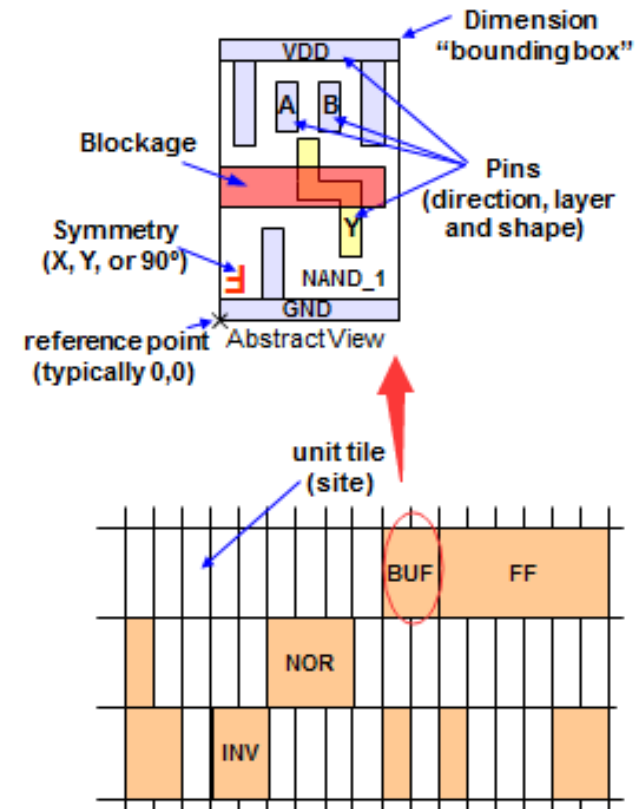
Standard Cell based ASIC Design

- Cells are of equal height in a row but of different widths based on functionality, ie. 2x1 NAND gate or FF.
- Std cell height is normally in terms of metal layer track width. 8T, 10T, 12T refers to std cell of height that is 8, 10, or 12 metal track wide.



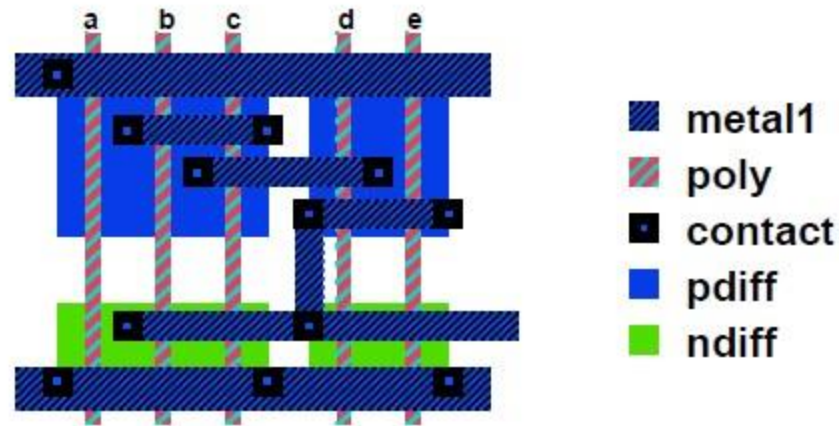
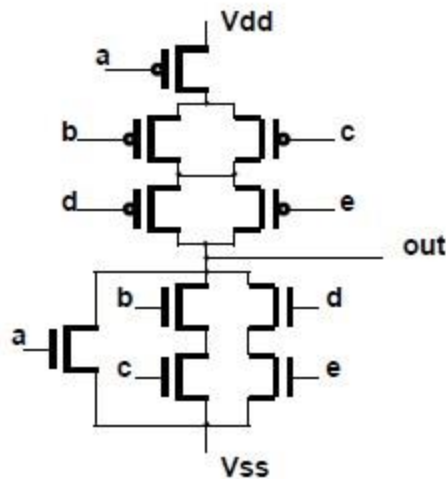
Row-based Std Cell Layout

- One chip may contain cells of different height but they are in different rows. Taller/larger std cells tend to faster but consume more area and power.
- Power and ground are connected through abutment in a row
- With a sizeable library of cells, also called technology library
 - Technology library is normally developed and distributed by foundry. But large companies normally has library group to develop their own customized library.



Standard Cell

- Example to show a gate schematic and its std-cell implementation



Standard cell

- In older technology, there are explicit spaces between std cell rows as routing channels
- In modern technology, routing is done OTC (over the cell) with multiple metal layers
- Routing over macros may have restriction. Sometimes designers need to pre-allocate routing channel from blocks to macros

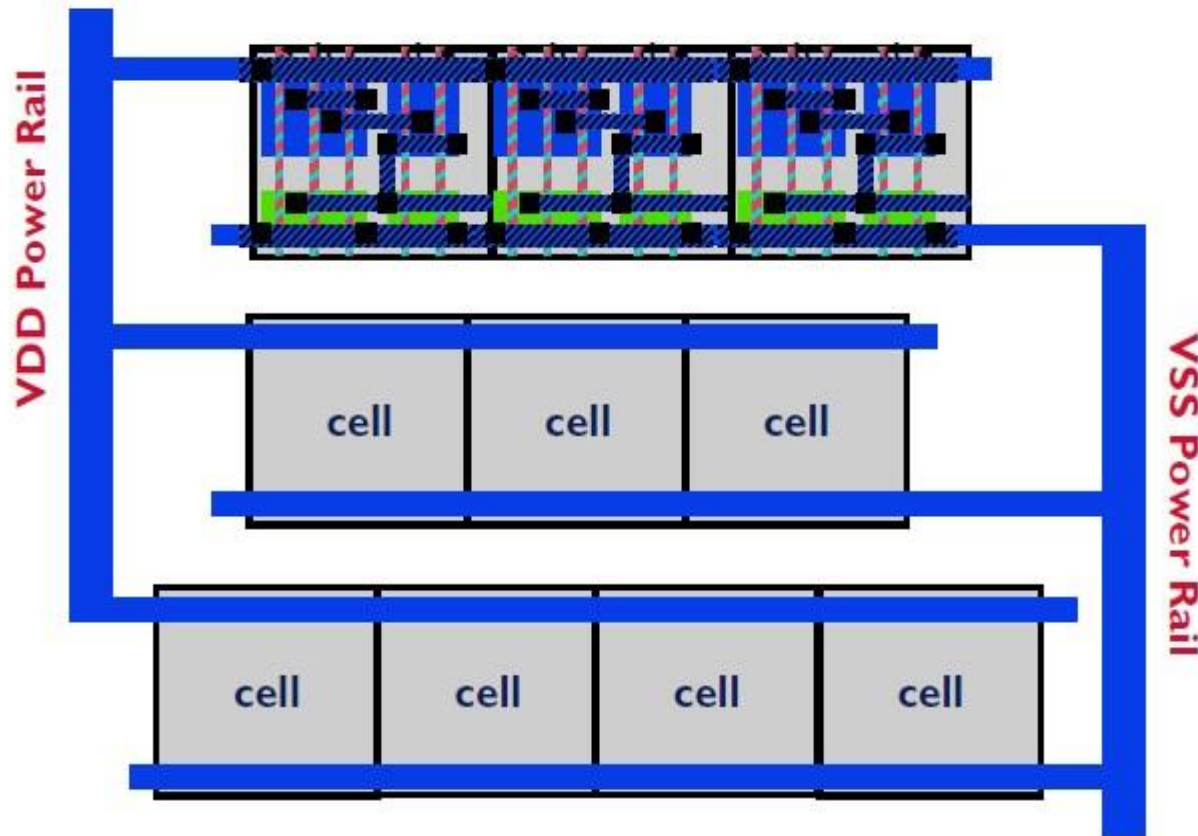


← One row of standard cells

← One channel of routing wires

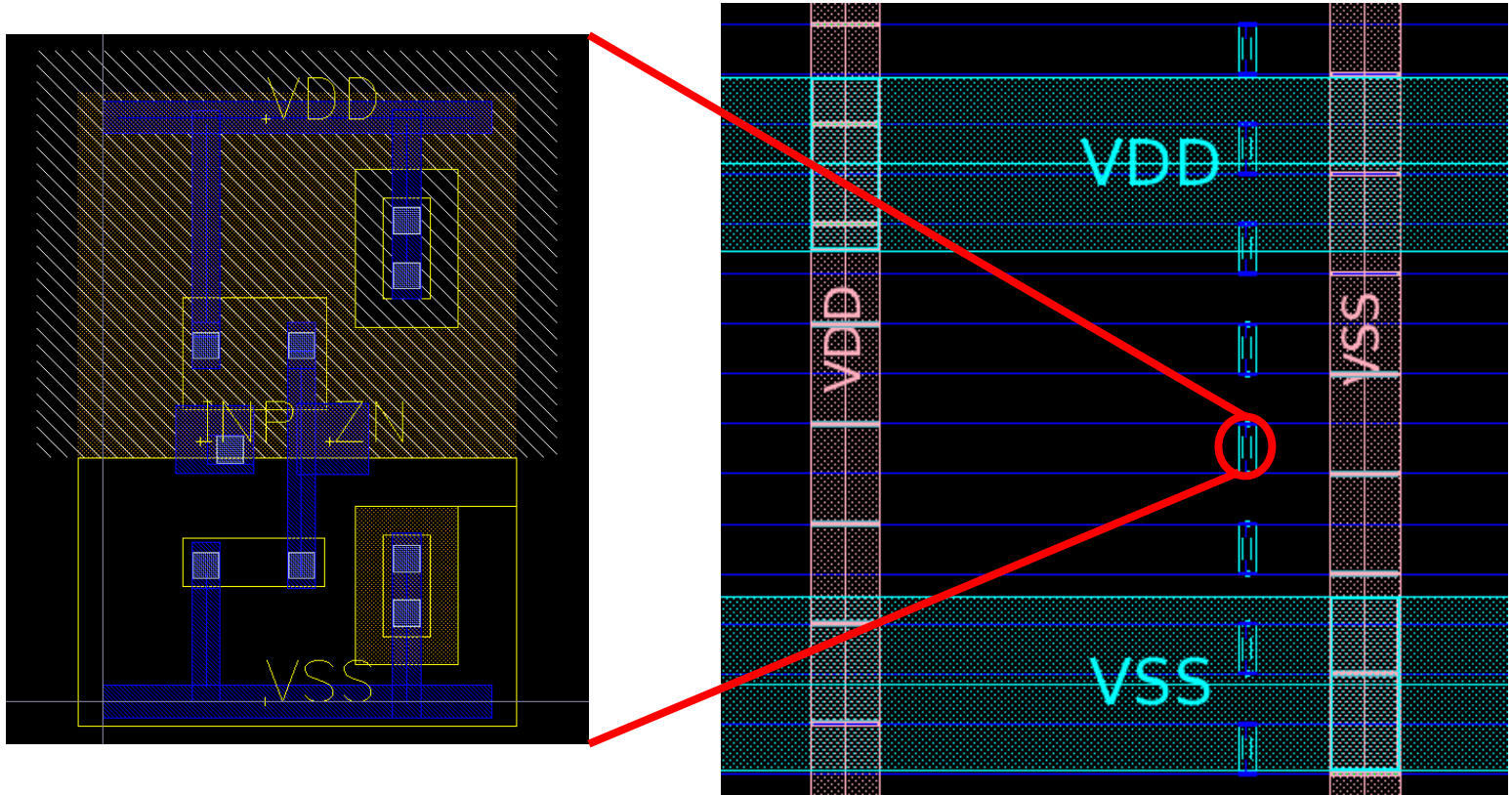
Row-based Std Cell Layout

- Power distribution – power ring/finger globally and power rail/abutment in std cell rows



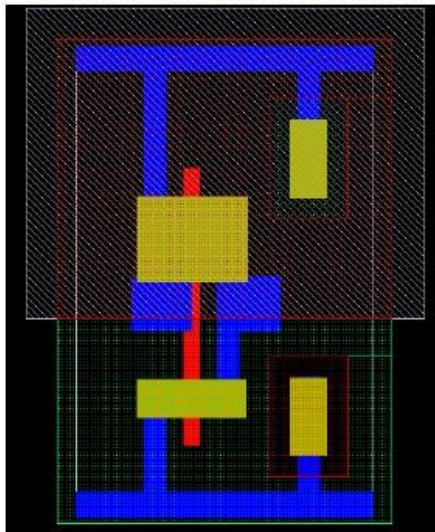
Standard Cell

- Std cell power and ground

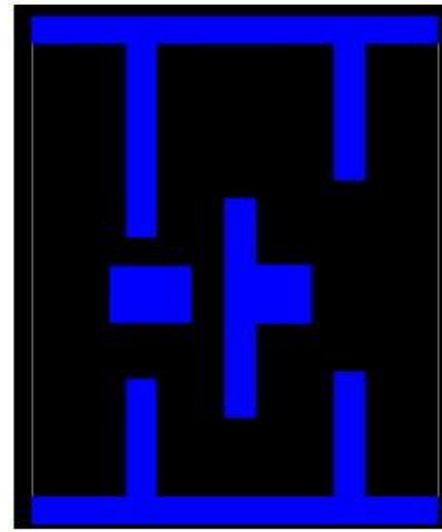


CEL View vs. FRAM View

- CEL view: all physical information about the cell. Used to output GDS and not used in PnR.
- FRAM view: contains pin location, shape, and routing blockage info. Used in PnR.
- Inverter example



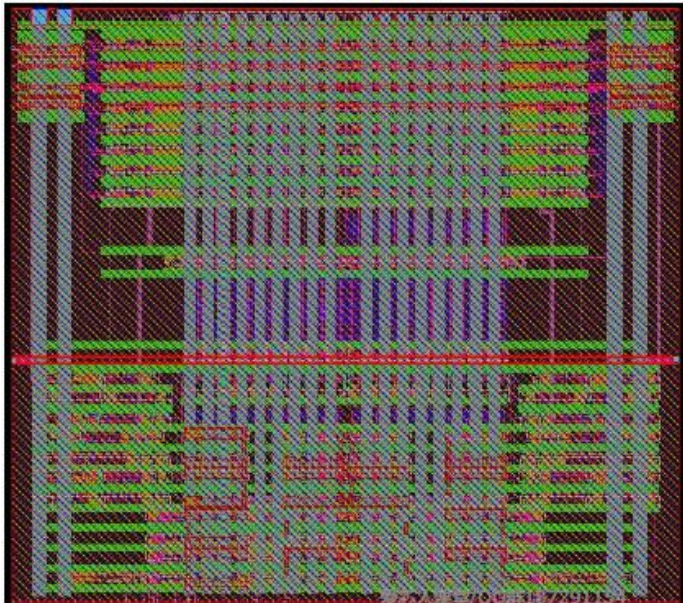
INV.CEL



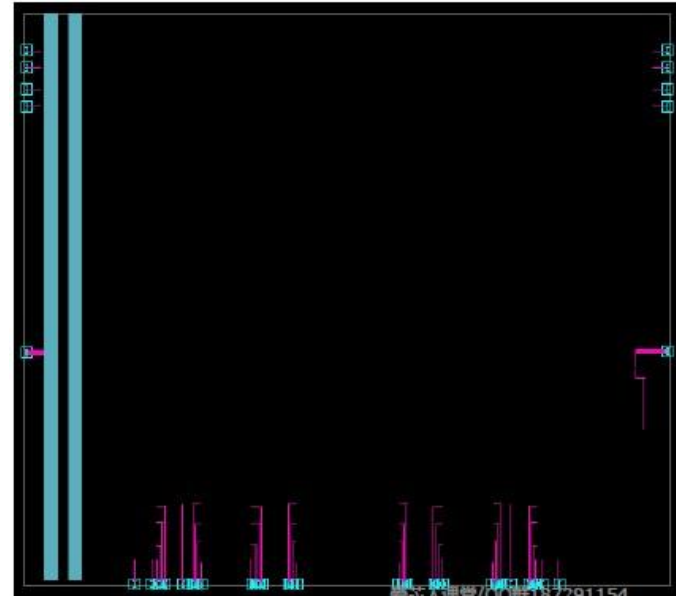
INV.FRAME

CEL View vs. FRAM View

- SRAM example



SRAM8x16.CEL



SRAM8x16.FRAME

Technology File (tf)

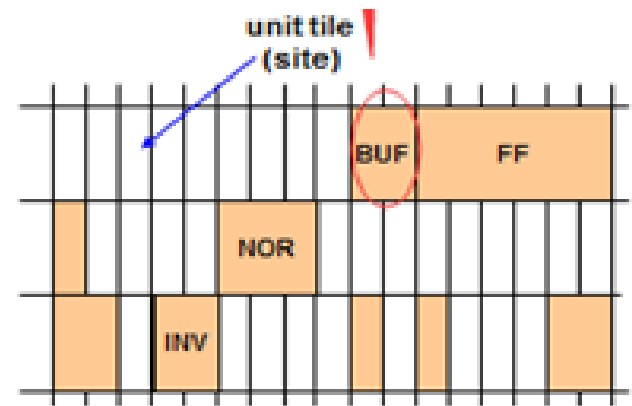
- Technology file is unique to each technology
- Contains metal layer parameters
 - Number, name designations, physical and electrical characteristics, design rules, etc., for each layer/via

```
Technology {
    name = "saed90"
    date = "May 25 2007"
    dielectric = 3.61e-05
    unitTimeName = "ns"
    timePrecision = 1000
    unitLengthName = "micron"
    lengthPrecision = 1000
    gridResolution = 5
    unitVoltageName = "V"
    voltagePrecision = 1000
    unitCurrentName = "uA"
    currentPrecision = 1000
    unitPowerName = "pw"
    powerPrecision = 1000
    unitResistanceName = "kohm"
    resistancePrecision = 10000000
    unitCapacitanceName = "pf"
    capacitancePrecision = 10000000
    unitInductanceName = "nh"
    inductancePrecision = 100
    minBaselineTemperature = 25
    nomBaselineTemperature = 25
    maxBaselineTemperature = 25
}
```

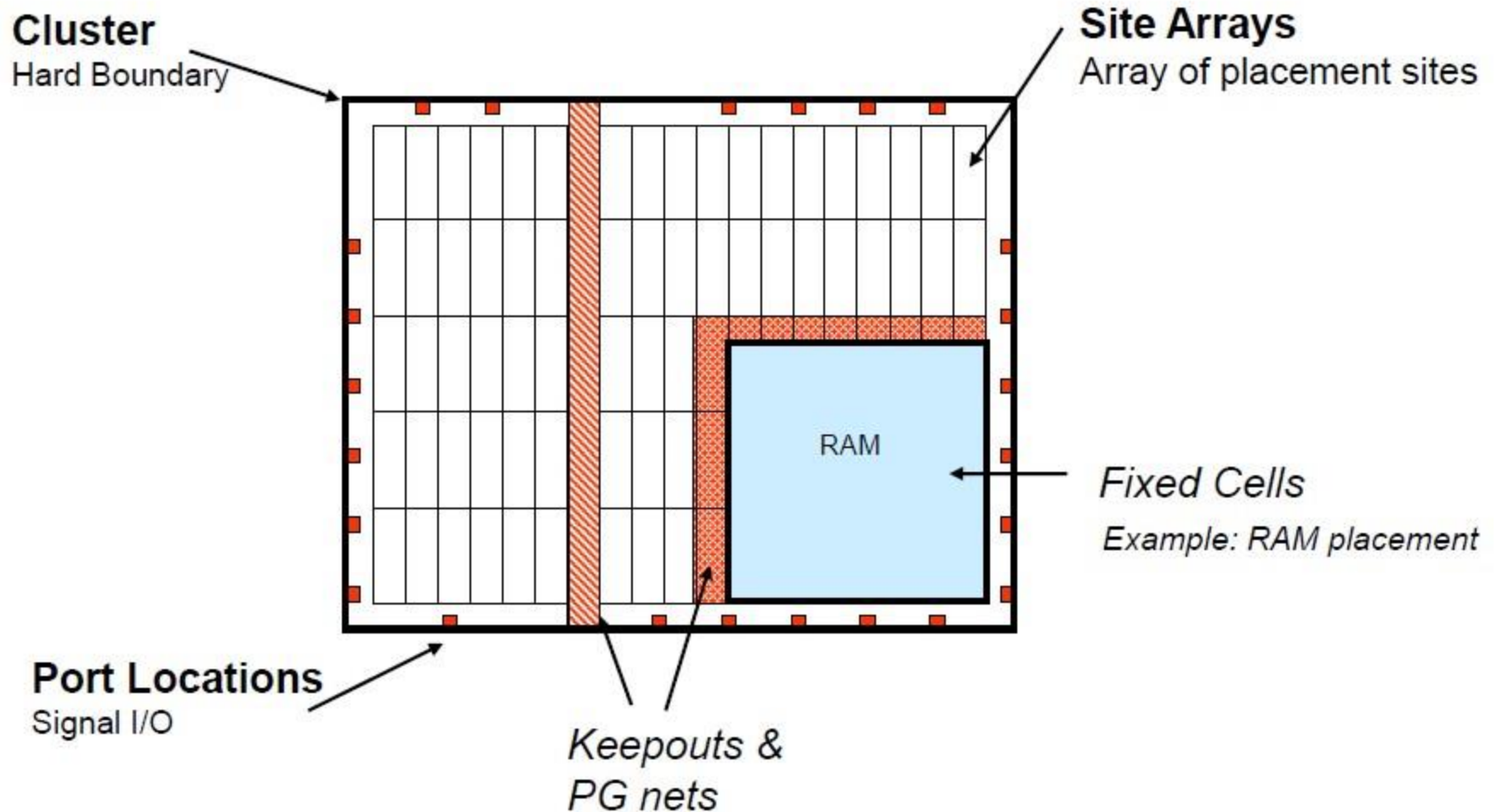
Technology File (tf)

```
Layer      "M1" {  
    layerNumber      = 11  
    maskName         = "metall1"  
    isDefaultLayer   = 1  
    visible          = 1  
    selectable       = 1  
    blink           = 0  
    color            = "blue"  
    lineStyle        = "solid"  
    pattern          = "rectangleX"  
    pitch            = 0.32  
    defaultWidth     = 0.14  
    minWidth         = 0.14  
    minSpacing       = 0.14  
    maxWidth         = 1000  
    minArea          = 0.06  
}
```

```
Tile      "unit" {  
    width           = 0.32  
    height          = 2.88  
}
```

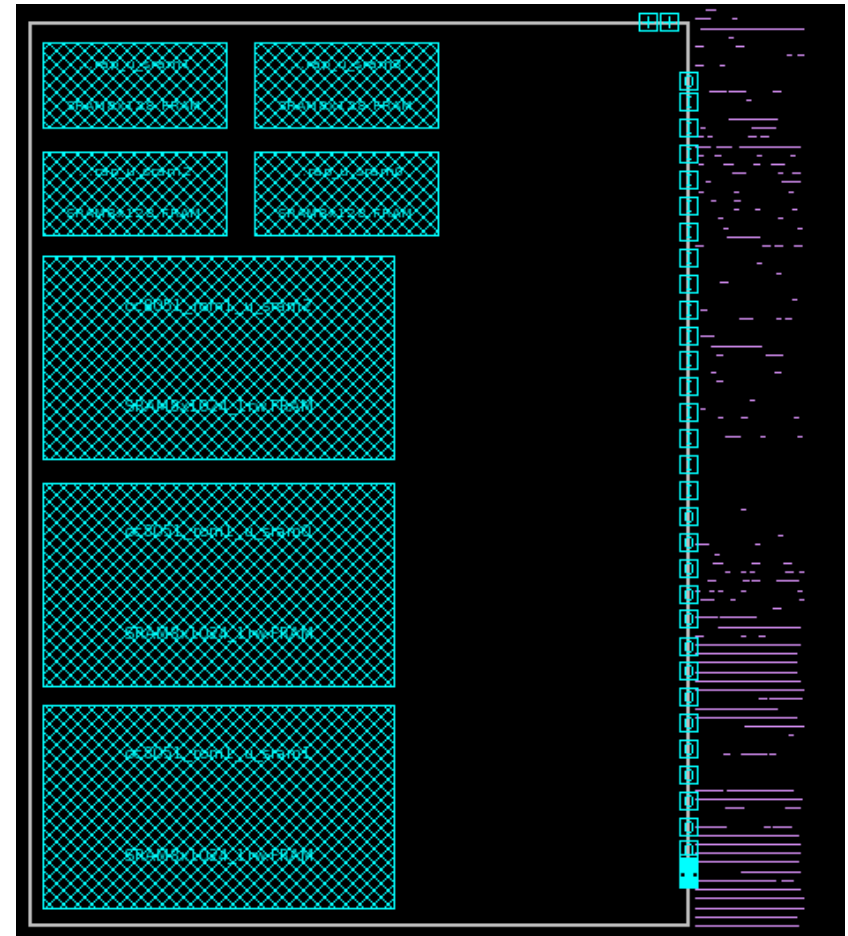


Floorplan – set target



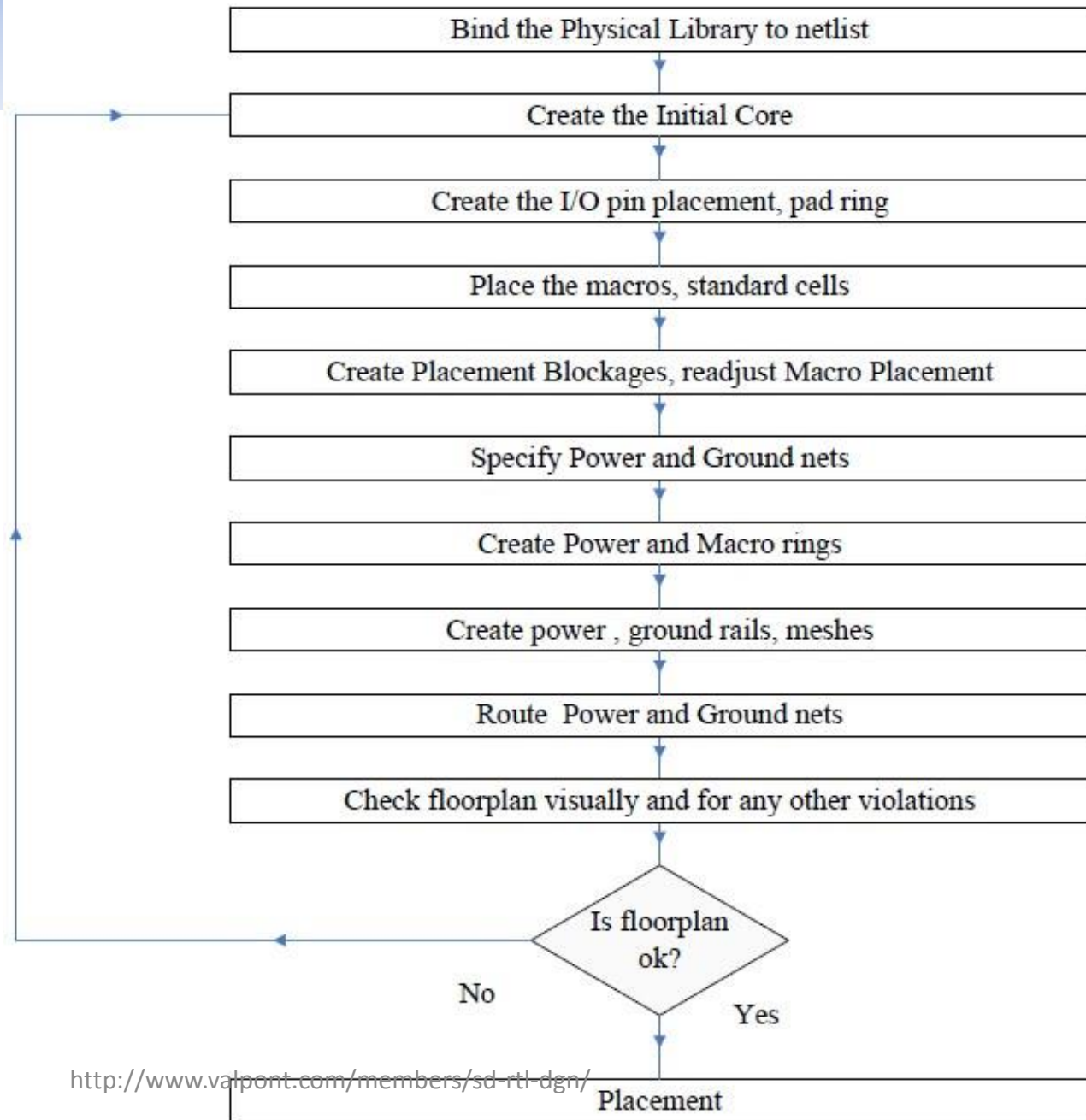
Floorplan

- Initialize_floorplan
- Size of chip
- Place main blocks, macros, pins, and IO ports
 - Normally hard macros such as memories are placed near the sides of cores or chip
- Power planning to give enough power to gates
- Type of clock distribution
- A good floorplan is to minimize the connections between logic groups/blocks



Floorplanning Summary

- A typical floorplan flow



Floorplan

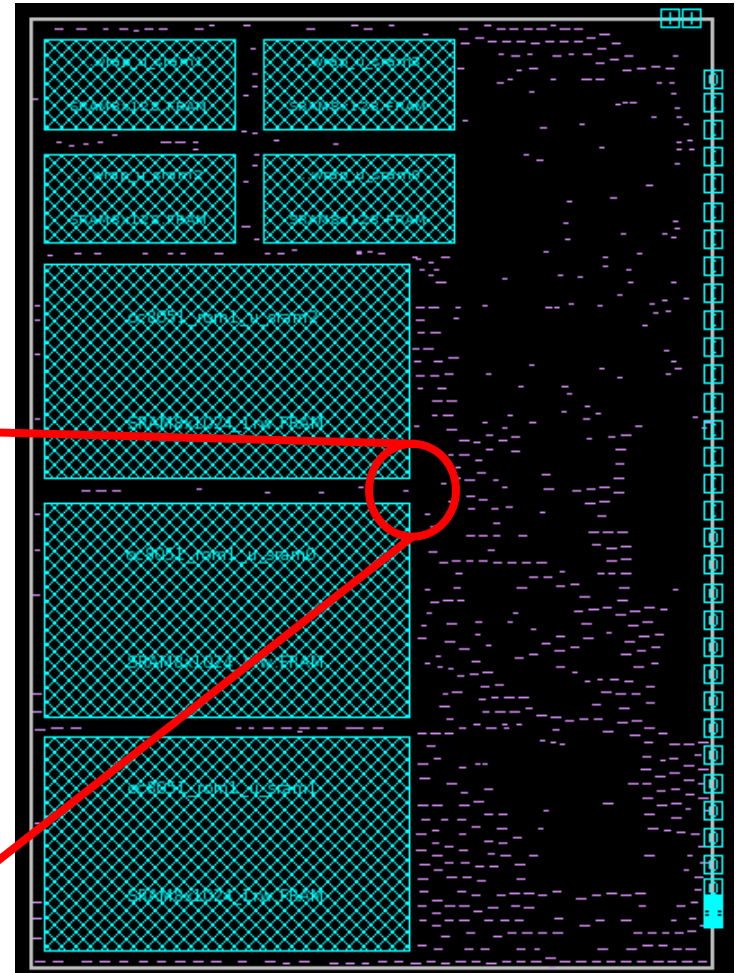
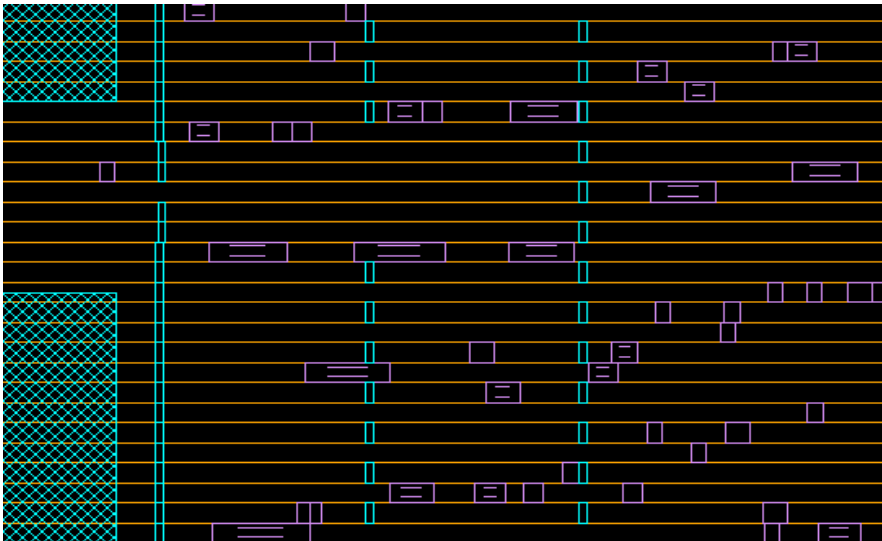
- Aspect ratio (AR)
 - Ratio of chip width and length
 - Take into account routing resource, if more horizontal layers, rectangle should be long and vice versa. Normally metal1 is used by std cells. Odd number metal layers are horizontal and even are vertical.
- Utilization
 - The percentage of area used, at chip level it is $(\text{std-cell-area} + \text{macro-area} + \text{pad-area}) / \text{chip-area}$
- Blockage
 - Standard cell blockage (no std cell allowed), non buffer blockage (only buffer allowed), and blockage below power lines

Placement

- Placement: place the gates (std cells)
- Coarse placement – approximate locations and may be not legal
- Detail placement – legalize placement
- Timing driven placement vs Congestion driven placement
 - TDP: opt critical path timing, shorten nets
 - CDP: tends to spread cells and thus lengthen nets
 - Iterative placement, set different opt effort based on congestion severity

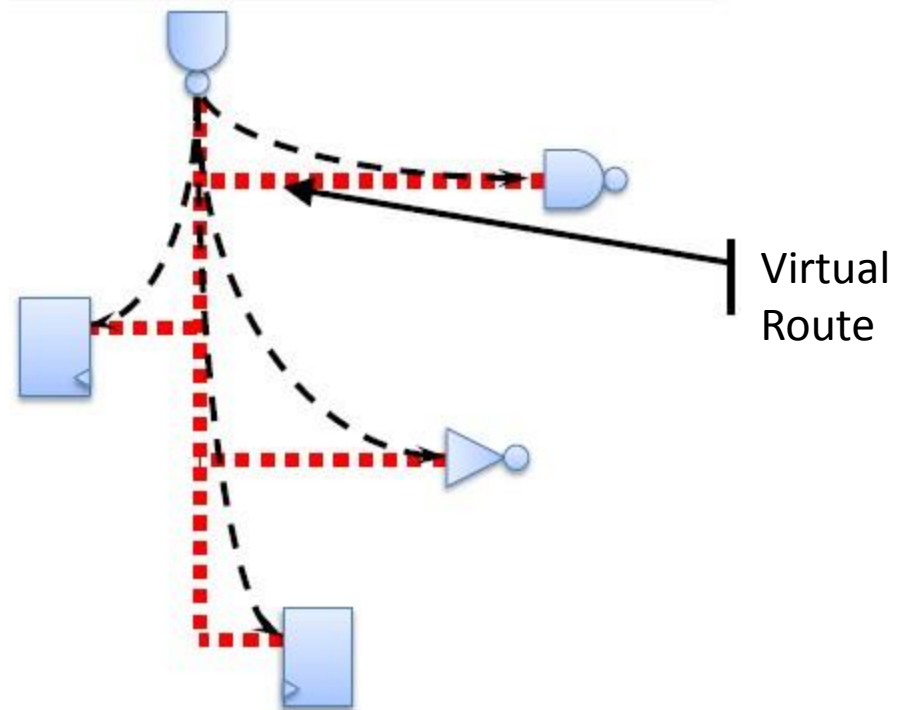
Placement

- Place_opt
 - Timing opt: add/delete buffers, resize gate, swap pins, move instances, etc.
 - min area and pwr



Placement

- Place_opt needs route timing to optimize placement. But route is not started yet, how to get route timing?
 - ICC uses virtual route (Manhattan geometry, horizontal-vertical, no diagonal routing) to estimate routing length and shape, then estimate RC parameters based on TLU+ model
 - The same way is used for clock tree synthesis (CTS)

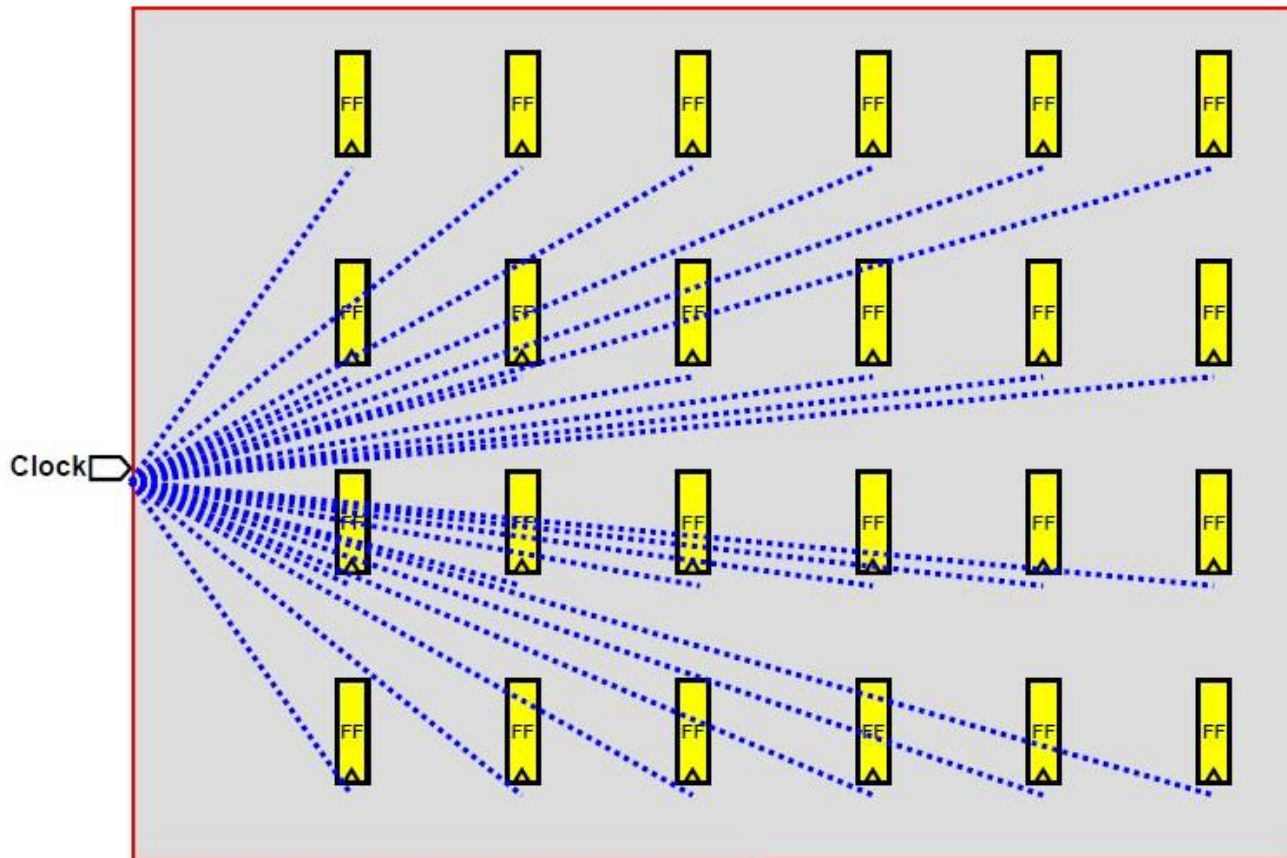


Clock Tree Synthesis (CTS)

- During placement, clock is assumed to be ideal – clock tree insertion delay is 0 from clock port to FF clock pin.
- CTS builds clock tree and routes clock nets from port to FF pins
- Need to optimize timing after CTS (called postCTS Opt) including setup and hold

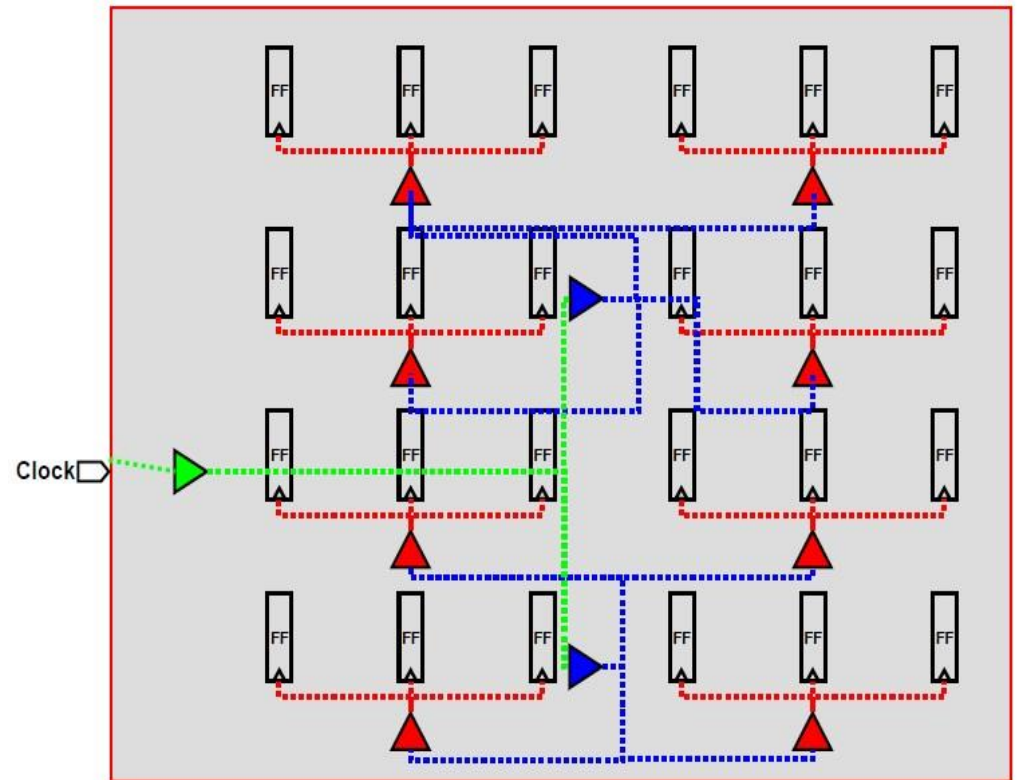
Clock Tree Synthesis (CTS)

- Ideal clock assumed before CTS



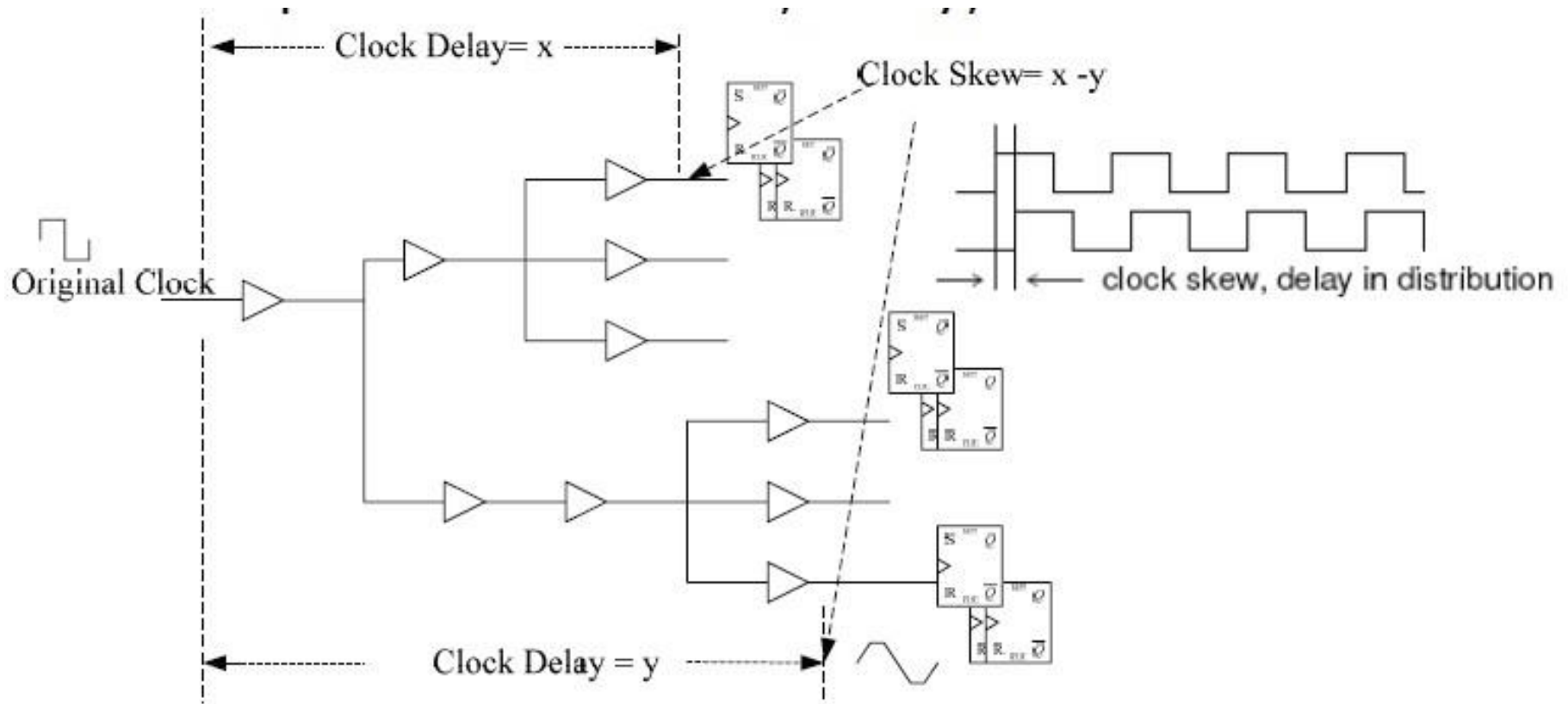
Clock Tree Synthesis

- Routed clocks after CTS.
 - Clock buffers added
 - Congestion may increase, other cells may be moved
- Normally CTS needs multiple iterations
- CTS needs frontend inputs



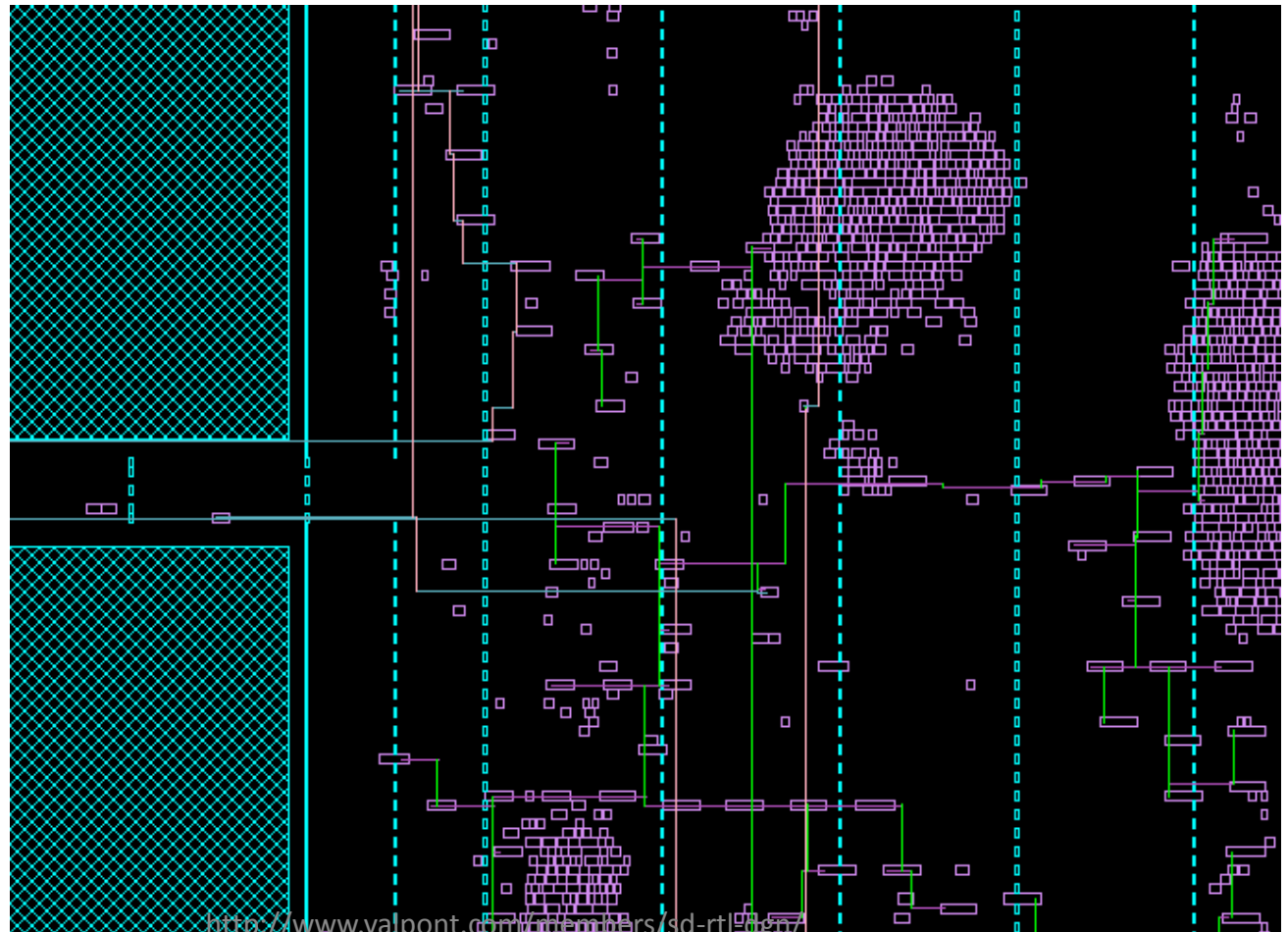
Clock Tree Synthesis (CTS)

- Main parameters: skew, delay, transition time



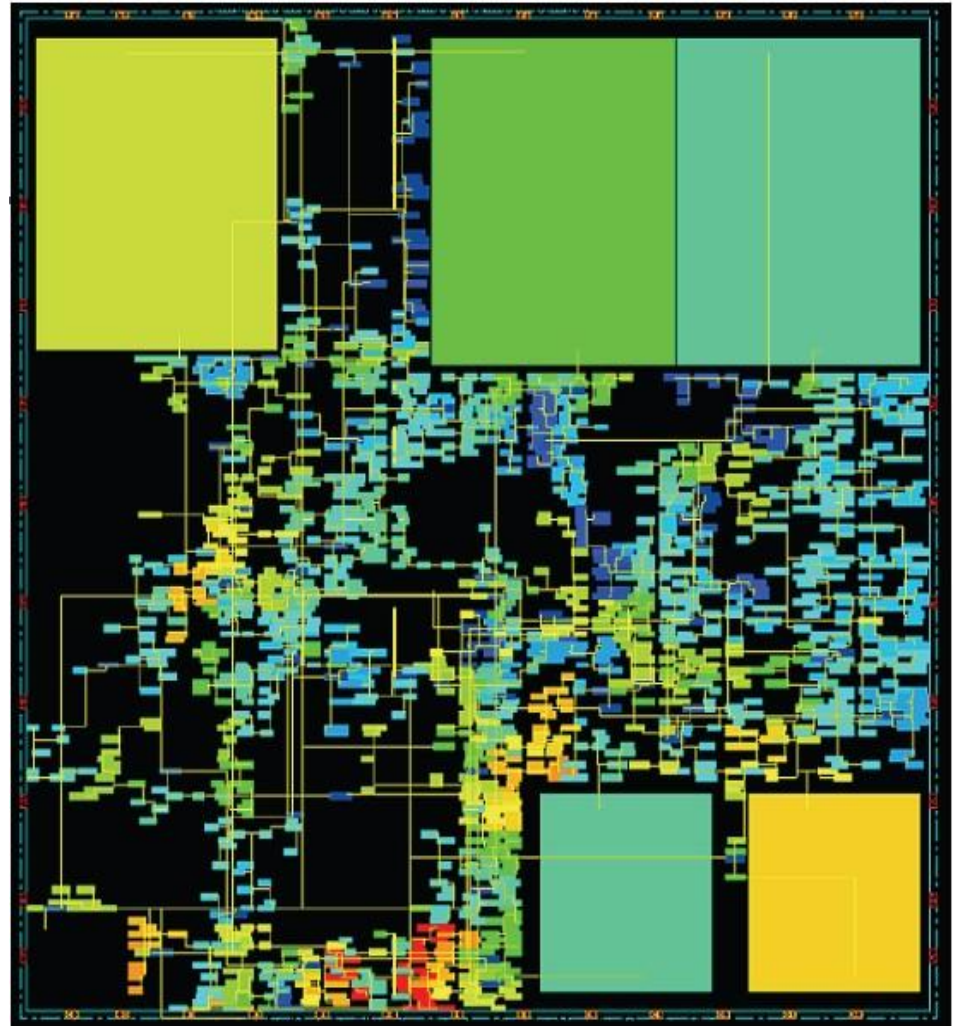
Clock Tree Synthesis (CTS)

- Route clock nets before signal nets (done in routing stage). Green colored nets are clock nets.



Clock Tree Synthesis (CTS)

- Example of clock tree in a single processor project



Routing

- Use metal/via to connect pins of std cells and macros as well as finishing up power and ground
- Output is a routed design which is DRC and timing clean



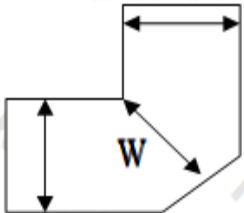
Routing

- Global routing
 - Plans overall connections between blocks, determine routing topology such as channels or routing regions nets go through. Max # of nets routed, min routing area, min total wire length, min delay
- detailed/local routing
 - Actual connection takes place, creates actual via and metal connection, min area/wire-length/delay
- Timing check after routing with accurate parasitic RC extraction of nets

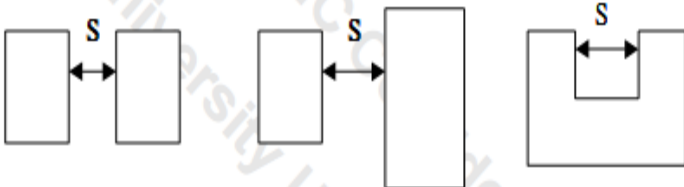
Design Rule Check (DRC)

- Design rule examples:

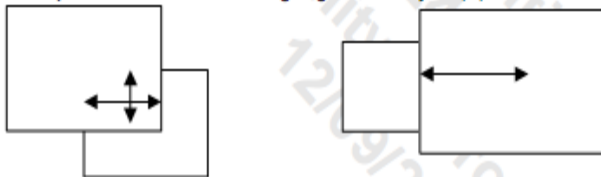
Width: Distance of interior-facing edge for single layer. (W)



Space: Distance of exterior-facing edge for one or two layer (S)



Overlap: Distance of interior-facing edge for two layers (O)



Enclosure: Distance of inside edge to outside edge (Fully inside) (EN)

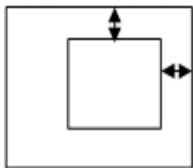


Illustration Window (CB)

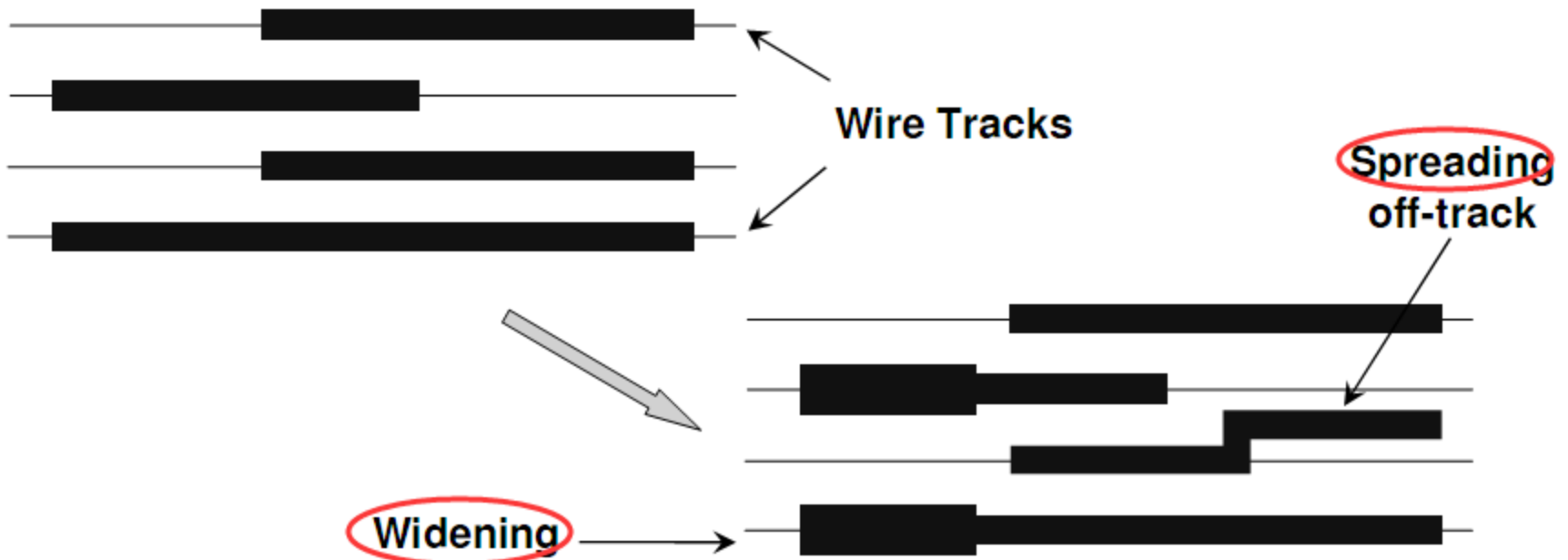
Rule No.	Description	Label	Rule
Single in-line (Figure 3.2.3.1) / Staggered (Figure 3.2.3.2) / Tri-Tier (Figure 3.2.3.3)			
#CB.W.1	Width of CB	A	≥ Table 3.2.3.1
#CB.W.2	Length of CB (the edge perpendicular to nearby chip edge)	A1	≥ Table 3.2.3.1
CB.W.3	Width of pad metal under Mtop-2 within CB region Staggered: M1, M2~Mtop-3 under inner pad Tri-tier: M1, M2~Mtop-3 under middle pad M2~Mtop-3 under inner pad C015/C018 CUP pad are excluded.	E	≥ Table 3.2.3.1
CB.W.4 ^②	Recommended total width of all metal layers connecting with bond pad. (Refer to the "I/O ESD Protection Guideline" section in the corresponding design rule manual.)	W	≥ 20
#CB.S.1	Space of two CB	B	≥ Table 3.2.3.1
CB.S.2	Space of metal pad geometries, or space of metal pad to metal line DRC check metal pad by the following definitions: (1) (Mx AND [CB sizing 3]), or (2) (Mtop/Mtop-1) AND [CB sizing 3] for C015/C018 CUP pad	F	≥ 2.5
#CB.S.3	Space of the different pad geometries [(Single to Stagger), (Single to Tri-tier) or (Stagger to Tri-tier)]	B1	≥ 15
#CB.P.1 ^④	Pitch of (4 pads at the chip corner) Note: Starting from the corners, the first four pitches must be larger than the pitch in the regularly repeating pad center area.	C1	≥ Table 3.2.3.1
#CB.P.2 ^④	Pitch of x-direction (inner and middle pad of Tri-Tier)	C2	≥ 13 ≤ 40
#CB.P.3 ^④	Pitch of x-direction [(inner and outer pad of Staggered) or (middle and outer pad of Tri-Tier)]	C3	≥ Table 3.2.3.1
#CB.P.4 ^④	Pitch of y-direction (between inner-middle-outer pad) Staggered: inner-outer pad Tri-Tier: middle-outer pad, and inner-middle pad	C4	≥ Table 3.2.3.1
CB.EN.1	CB enclosure by Mx and Mtop in pad regions Except, Staggered: M1, M2~Mtop-3 under inner pad Tri-Tier: M1, M2~Mtop-3 under middle and inner pad C015/C018 CUP pad: M1~Mtop-2	D	≥ 2 (Al process)
#CB.R.1	For pad pitch < 55μm, one pair of notches on each pad CB is required, to be the reference point for probing and bonding. Please refer to Figure 3.2.3.1 for the size and location.		

Chip Finish

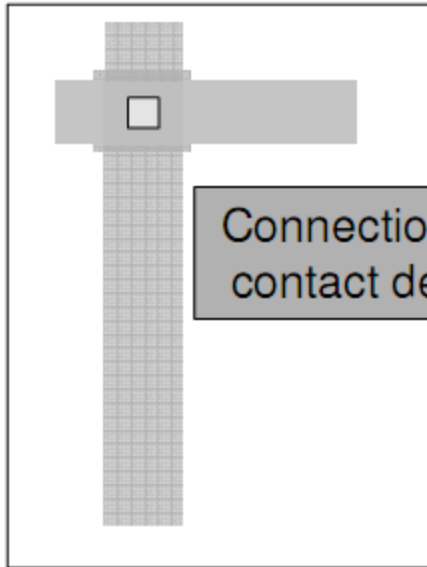
- To improve yield, so called DFM (Design for Manufacturing)
 - Wire spreading
 - Redundant via insertion
 - Filler cell insertion
 - Metal fill insertion
 - Metal slotting

Wire Spreading / Widening

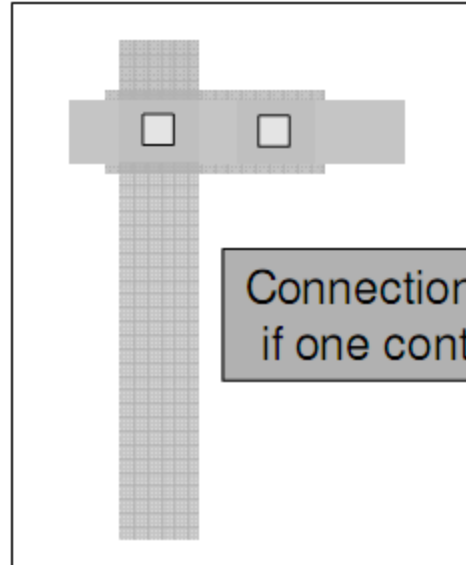
- Widen wires to reduce *open* critical area



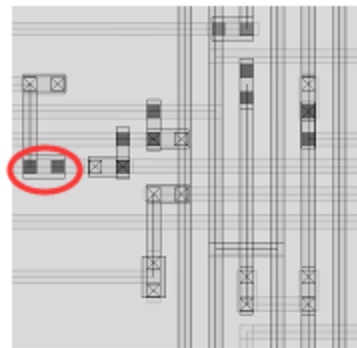
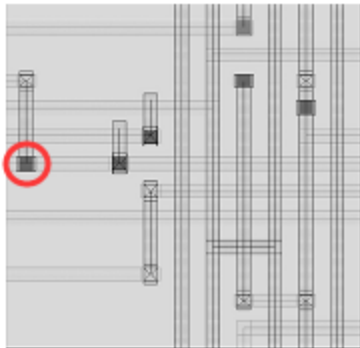
Redundant Via Insertion



Connection fails if contact defective

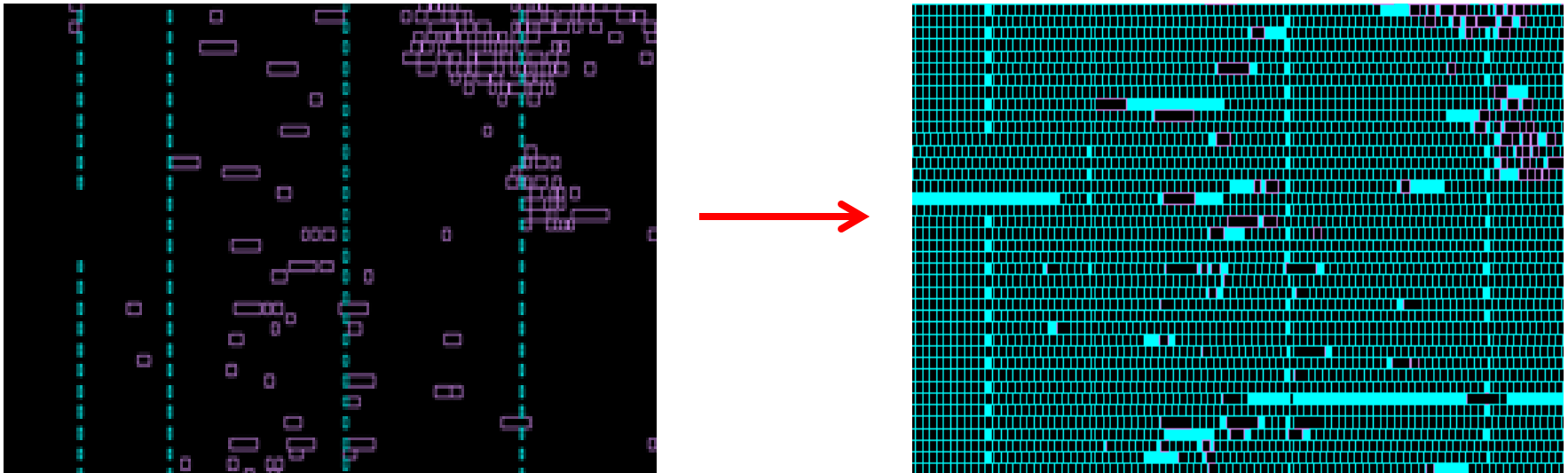


Connection is okay even if one contact defective



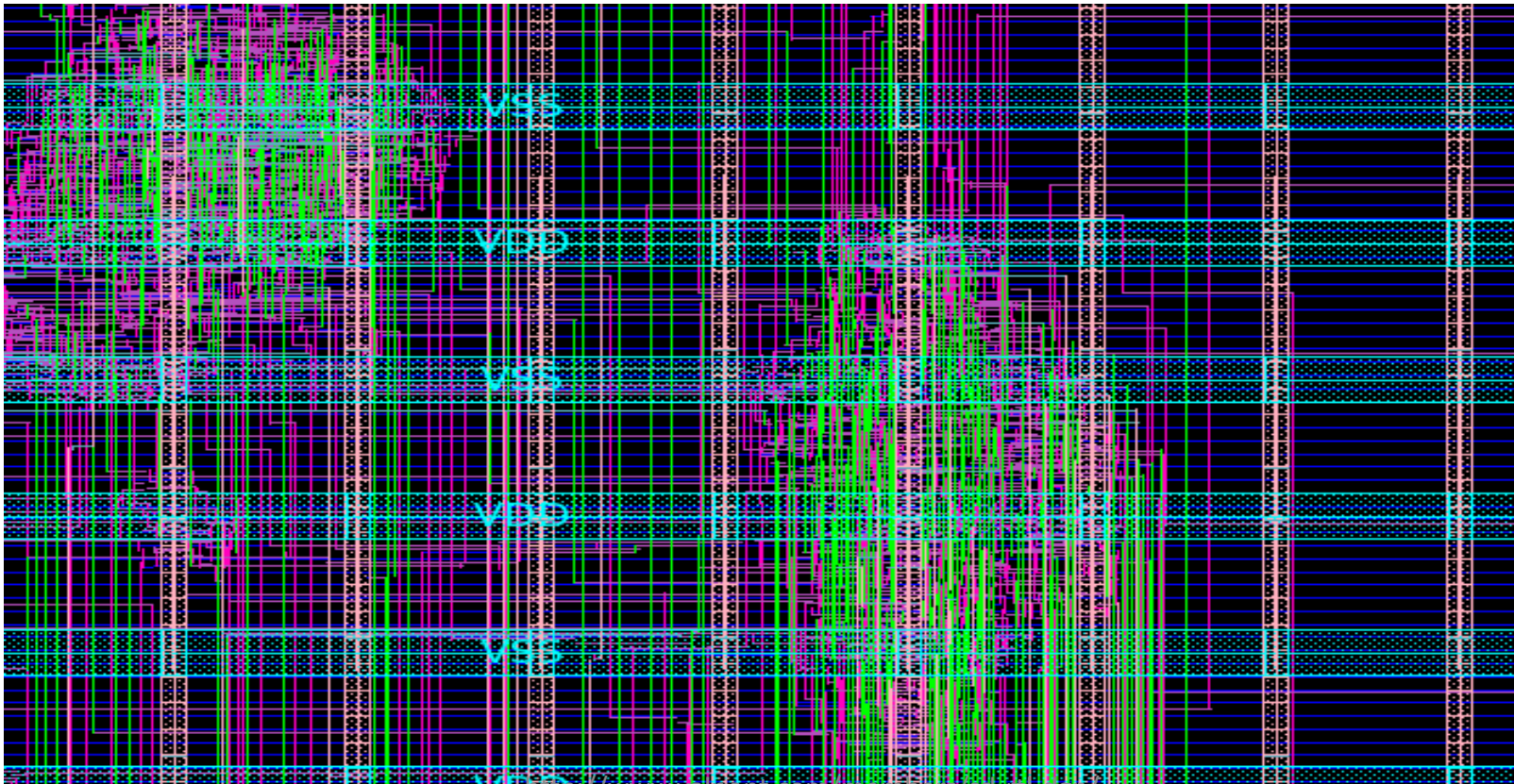
Filler Cell Insertion

- Fill up empty row/site which doesn't contain std cells
- Make Nwell/Pwell consecutive
- Can be used as decoupling cap cell

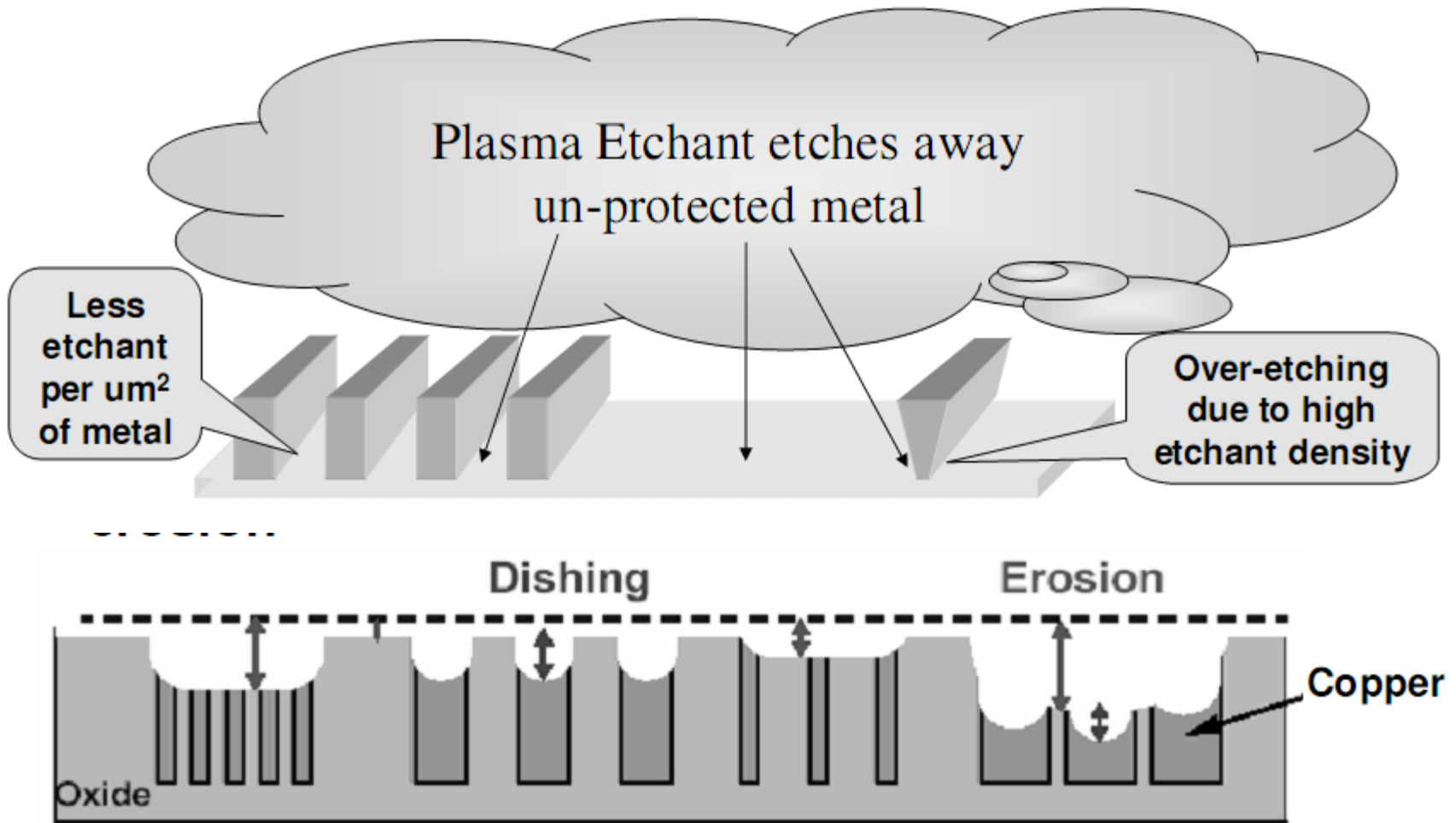


Metal Fill

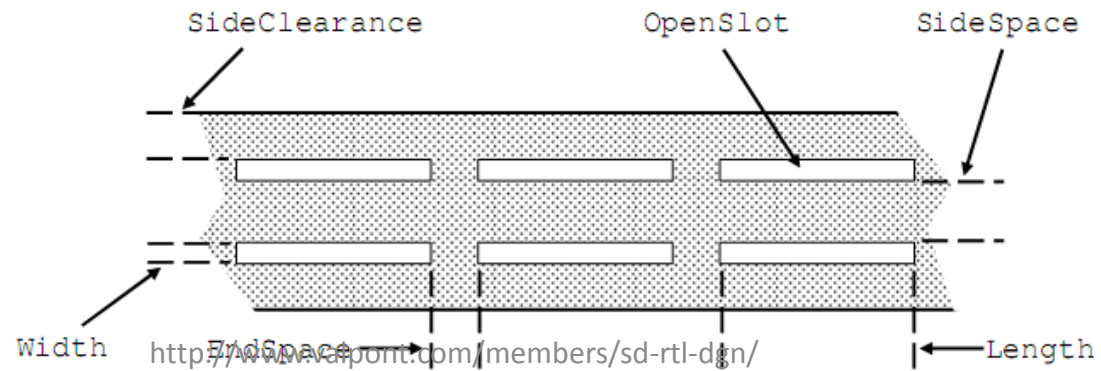
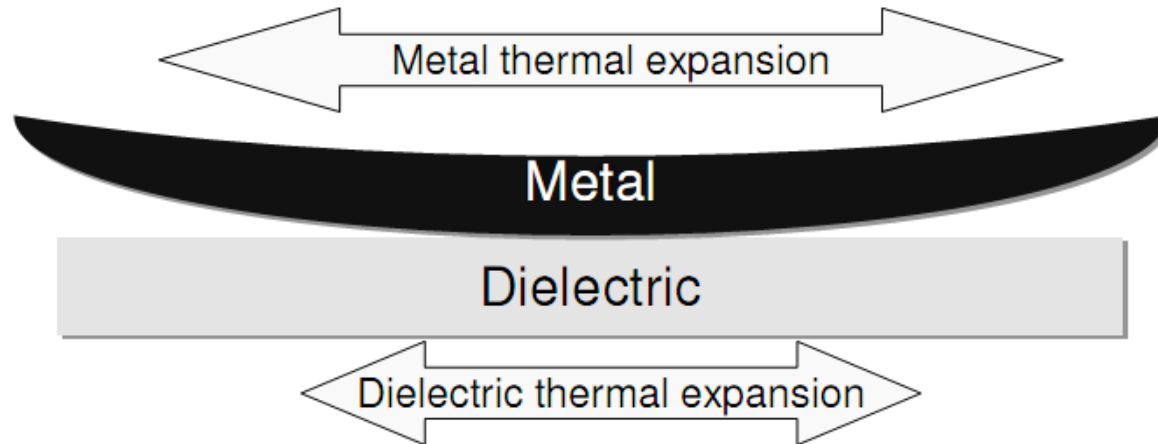
- Issued caused by small metal density in manufacturing



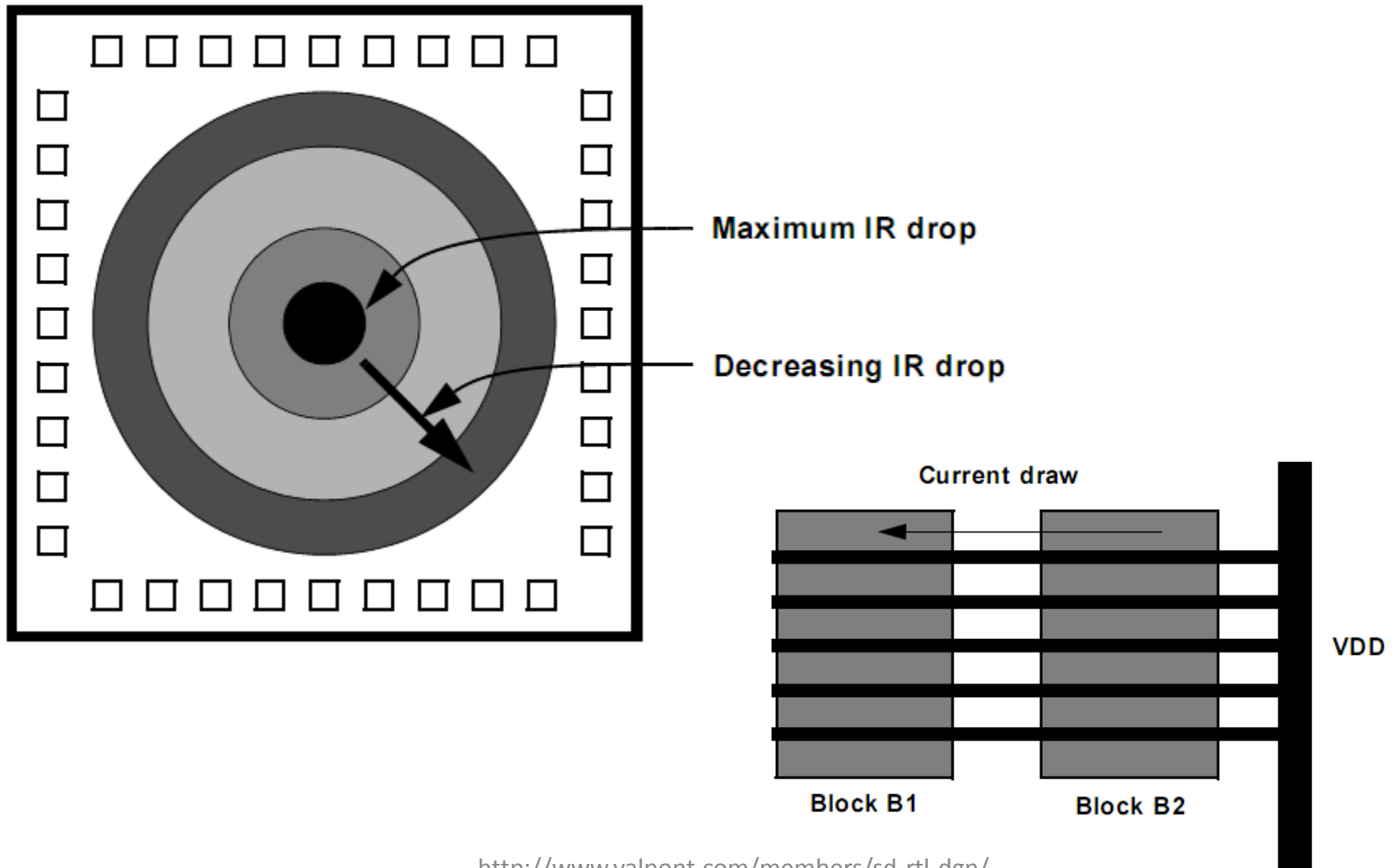
Metal Fill



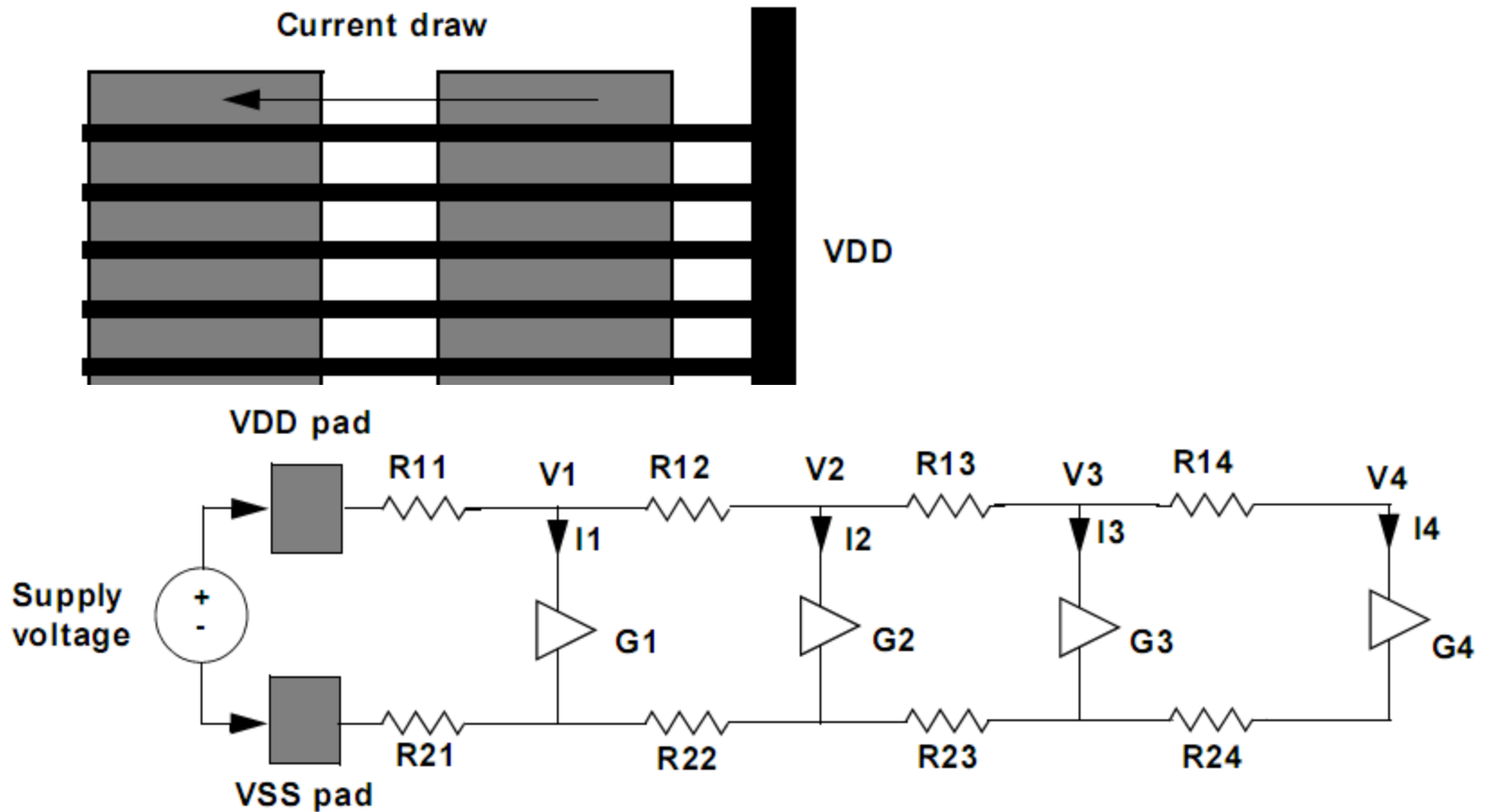
Metal Slot



Power Signoff - IR Drop



IR Drop

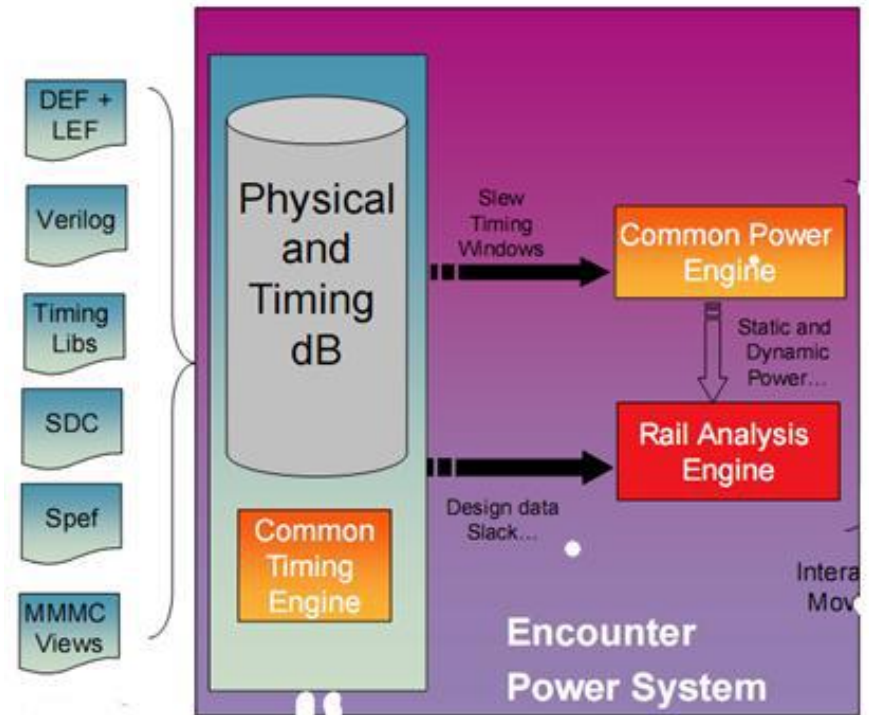


IR Drop

- Static IR drop
 - 3% for VDD + VSS (Flip chip)
 - 5% for VDD + VSS (wire bond)
- Dynamic IR drop
 - Around 5x in signoff constraint
 - Scan mode IR drop
 - Peak power usually around clock edge
 - Analyzing IR drop during small timing window when FFs are switching at the same time

Power Signoff

- Tool
 - Cadence EPS
 - Synopsys PrimeRail



Power Signoff

