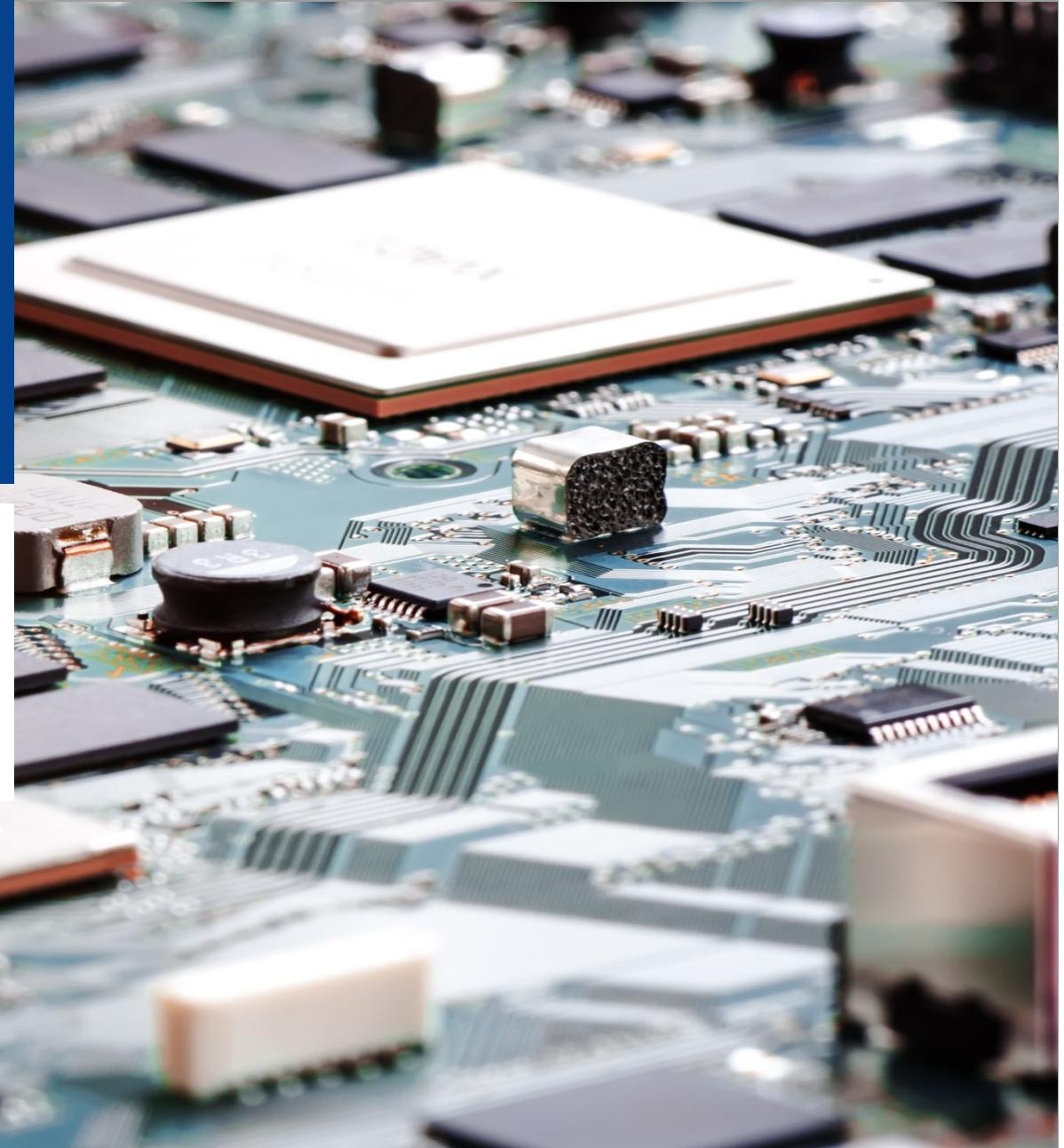


LSI LOGIC DESIGN

CHAPTER 6 Layout Design

JUNE 30, 2020
PHAM TUONG HAI
QUALITY ASSESSMENT & TRAINING DEPARTMENT
RENESAS DESIGN VIETNAM CO., LTD.
RENESAS ELECTRONICS CORPORATION

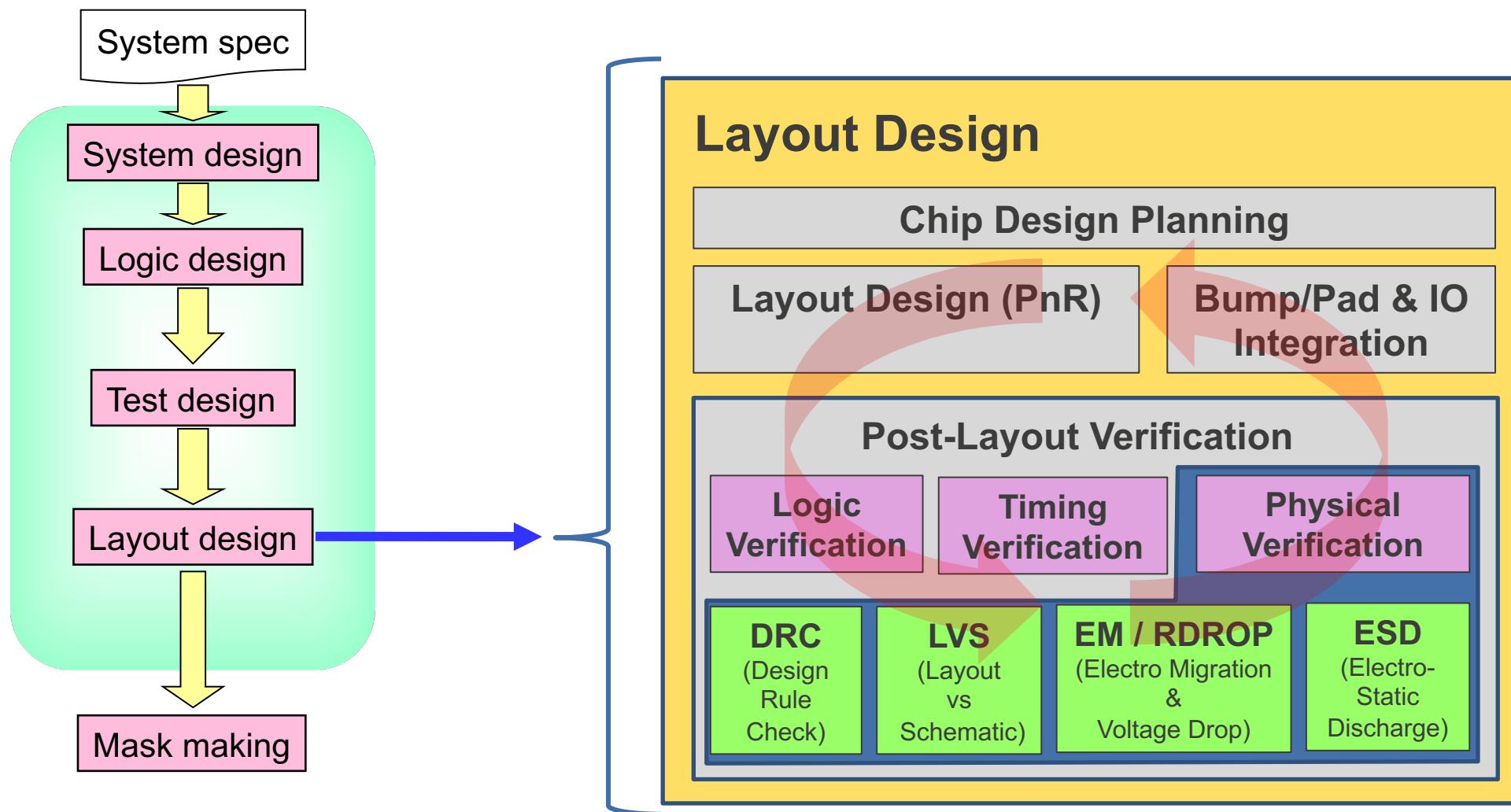


CHAPTER 6. Layout Design

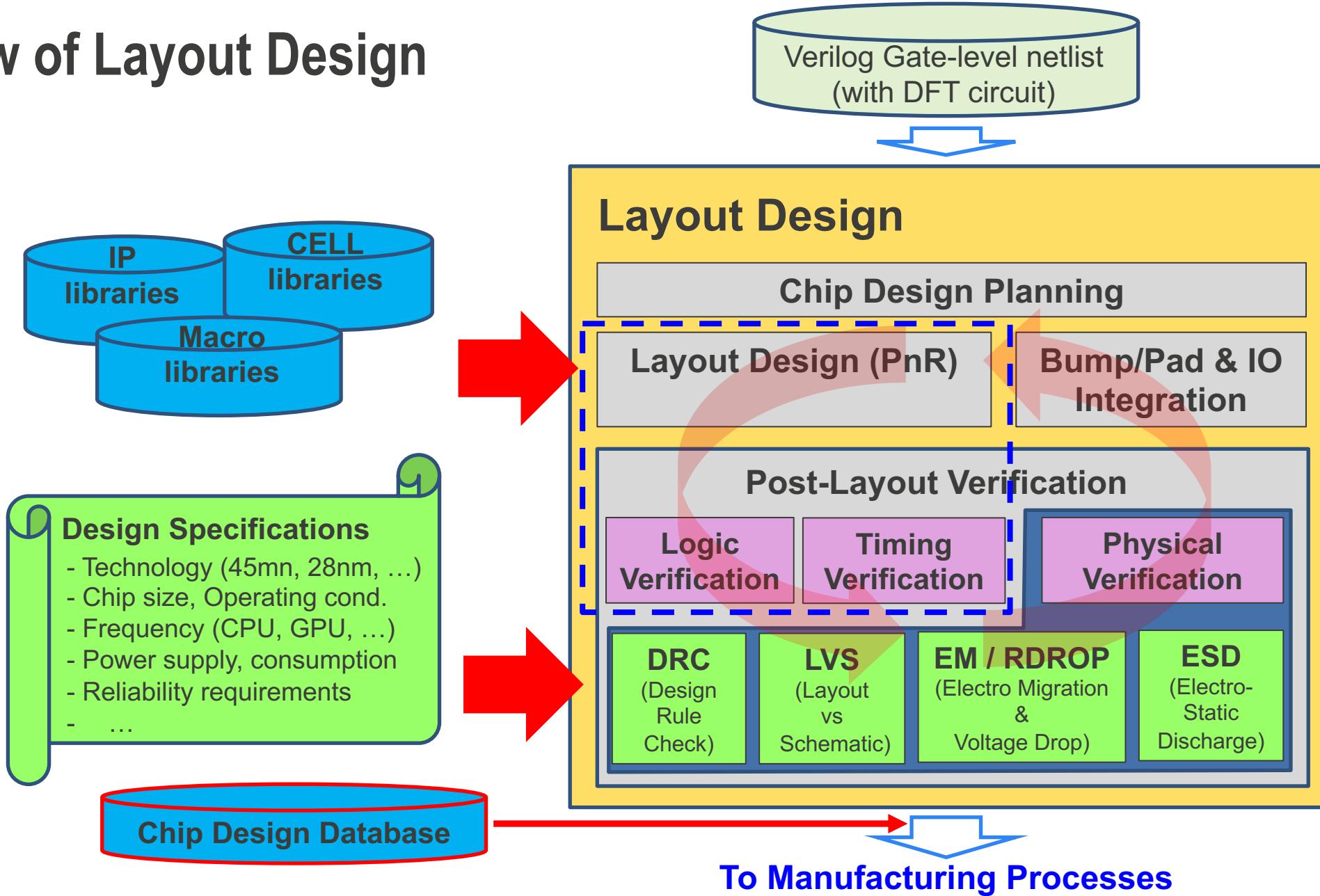
- 6.1. Overview of Layout Design.**
- 6.2. Cell and Library.**
- 6.3. Understand about Layout Design.**
- 6.4. Chip Design Planning.**
- 6.5. Block and TOP Level Layout Design.**
- 6.6. Process Defect and Yield.**

6.1 Overview of Layout Design

LSI Design Flow



Overview of Layout Design



6.2 Cell and Library

What is a Cell?

A **cell** is **macro** or **functional unit** that performs common operations and is used to build more complex logic blocks.

Examples:

- Standard cells: Inverter, NAND, NOR, Flip-Flop, Latches and Buffers.
- Macro cells: ADC (Analog to Digital converter), DAC, THS (Thermal sensor), USB, PCI
PLL (Clock pulse generator)

What is a (Cell) Library?

A cell library often refers to a collection of cells:

- Standard cell library (Primitive cell Library)
- I/O cell library
- Memory (Compiled memory, Fixed size memory)
- Analog (ADC, DAC, PLL, USB, THS)

Cell library consists of :

- Schematics (Transistor level circuit. Usually, not released)
- Frontend model (Verilog, VHDL, Liberty(.DB), other support libraries)
- Backend Model (LEF, ASTRO/Milkyway, GDS(Layout), CDL)

LEF and ASTRO are cell information which shows locations of ports, cell size, wiring prohibition area, ... used for Place-and-Route.

LEF is often used for Cadence P&R tool

ASTRO/MILKYWAY is often used for Synopsys P&R tool

GDS is a design database format for design transfer

CDL is format of design circuit netlist, used for LVS verification (*Layout-versus-Schematic*), or ESD verification, ...

Example of Library Model (INVERTER)

Verilog (.v)

```
`ifdef TS_OFF
`else
`timescale 1ps/1ps
`endif
`celldefine
`ifdef verifault
`suppress_faults
`enable_portfaults
`endif

`ifdef FAST_FUNC
`delay_mode_zero
`else
`delay_mode_path
`endif

module TCAINVXC( A,YB );
output YB;
input A;
reg notifier;

not (YB,A);

`ifdef FAST_FUNC
`else
....
```

Liberty (.lib)

```
cell(TCAINVXC){
area : 3.0;
cell_leakage_power : a;
cell_footprint : INVX_;
pin(YB){
function : "!(A)";
max_fanout : 50;
max_capacitance : b;
capacitance : 0.000000;
direction : output;
internal_power() {
related_pin : "A";
fall_power(pwr_tin_oload_3x3){
index_1 (" , , ");
index_2 (" , , ");
values(" , , ", \
"- , , ", \
" , , ");
}
rise_power(pwr_tin_oload_3x3){
index_1 (" , , ");
index_2 (" , , ");
values(" , , ", \
" , , ", \
" , , ");
}
....
```

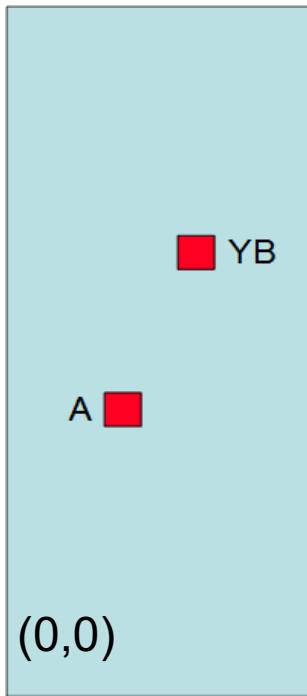
LEF (.lef)

```
MACRO TCAINVXC
CLASS CORE ;
FOREIGN TCAINVXC 0.000 0.000 ;
SIZE 0.840 BY 2.520 ;
SYMMETRY Y X ;
ORIGIN 0.0 0.0 ;
SITE CORE009 ;
PIN A DIRECTION INPUT ;
USE SIGNAL ;
AntennaGateArea 0.14 LAYER M1 ;
AntennaPartialMetalArea 0.104 LAYER M1 ;
PORT
LAYER M1 ;
RECT 0.170 0.970 0.350 1.550 ;
END
END A
PIN YB DIRECTION OUTPUT ;
USE SIGNAL ;
AntennaPartialMetalArea 0.242200 LAYER M1 ;
AntennaDiffArea 0.331200 LAYER M1 ;
PORT
LAYER M1 ;
RECT 0.490 0.380 0.630 2.110 ;
END
END YB
.....
```

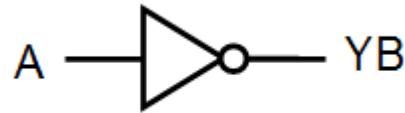
Actual numbers are deleted from original .lib

Example of Library Model (INVERTER) (cont'd)

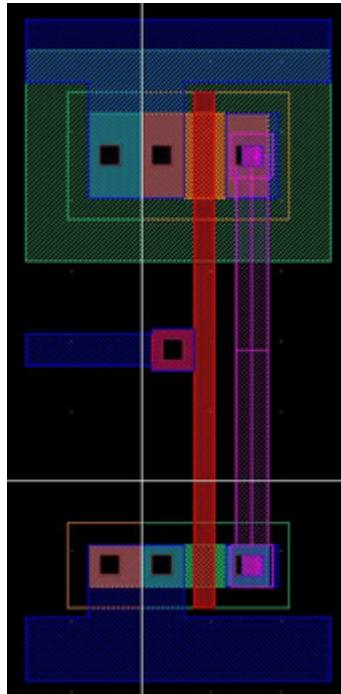
PnR Layout view



Logic symbol



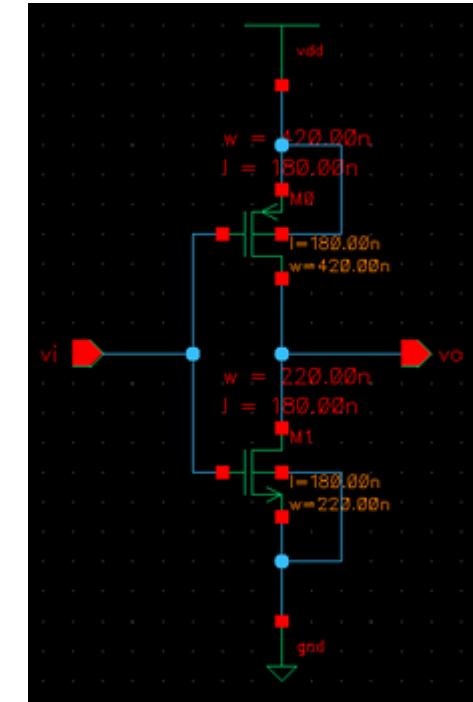
Detailed layout



CDL model (.cdl) (aaa~ddd is MOS size)

```
.SUBCKT TCAINVXC A YB  
X12 A YB VDD VSS / TCXINV wn=aaa u wp=bbb u  
XI0 A YB VDD VSS / TCXINV wn=ccc u wp=ddd u  
.ENDS
```

Schematic view (Circuit)



6.3 Understand about Layout Design

Layout Methodology

1) Individual design method

Design with interactive mode EDA tool (interactive design).

□ Eg: applied in custom layout design, standard cell / Analog layout design, ...

2) Master slice method (Gate Array Method) P&R

Common master layout with buried transistors is ready and automatic wiring tool is applied to do layout.

The development cost is small thanks to the buried gate-array.

Since base wafers are already prepared, the manufacturing period is short.

3) Cell-based method P&R

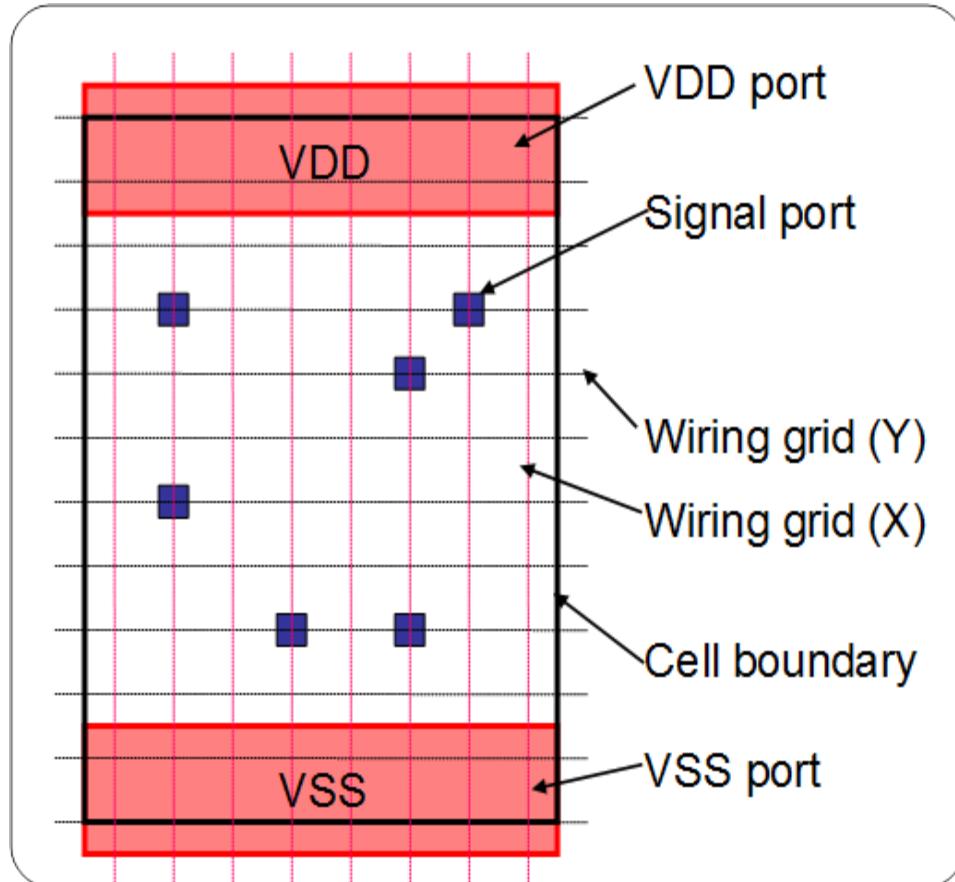
P&R area of cell/module are changeable.

It is necessary to minimize the chip size.

4) Hierarchical cell-based method P&R

A hierarchical design of method (3) where logic is split from the top down, and layout is done from the bottom-up.

Standard Cell Structure of Cell-based Method



The width of the cell is changeable.
(integral multiples in X wiring grid)

The height of the cell is normally constant (in a specific library of a specific technology, and is integral multiples in Y wiring grid)

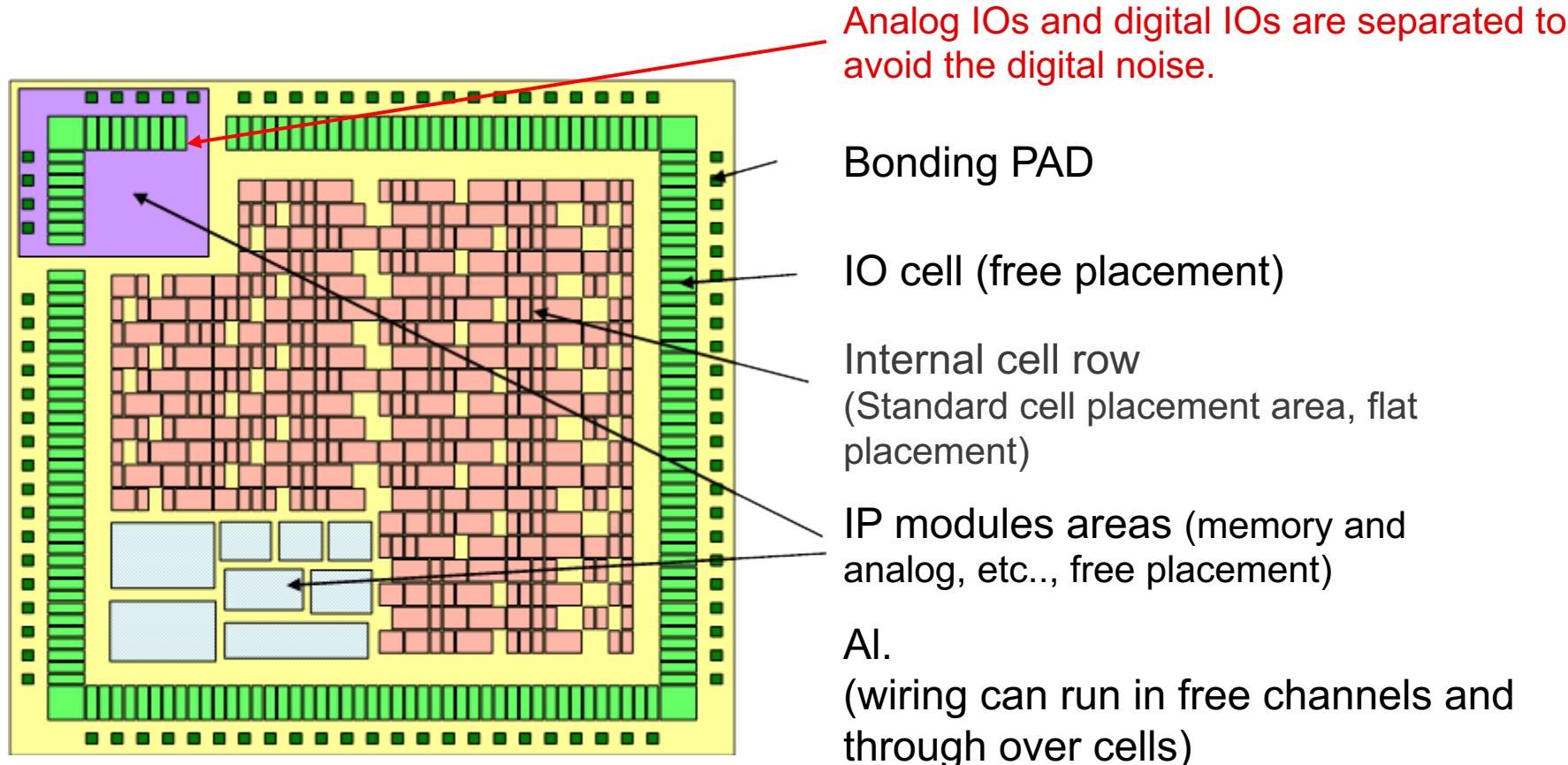
The signal port is “on grid”.
(for route-ability)

Power supply ports are in fixed locations in the cell (normally at top and bottom of the cell). Y location is same to all cells.

Metal wirings can run over cells.

Cell-based Method

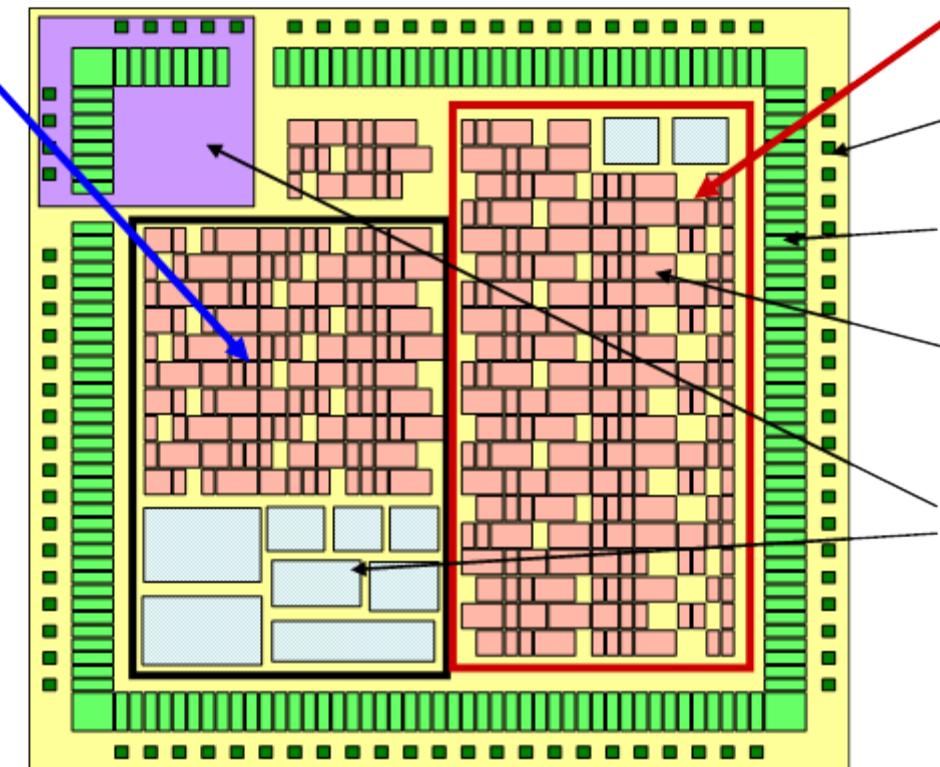
Cells can be arbitrarily placed - Automatic P&R is applied - Multi-layer metals are used.



Hierarchical Cell-based Method

The standard cell placement area is done layout hierarchically depending on the hierarchical logical structure of the gate level netlist - Automatic P&R is applied.

Hierarchical logic B



Hierarchical logic A

Bonding PAD

IO cell (free placement)

Internal cell row
(Standard cell placement area, flat placement)

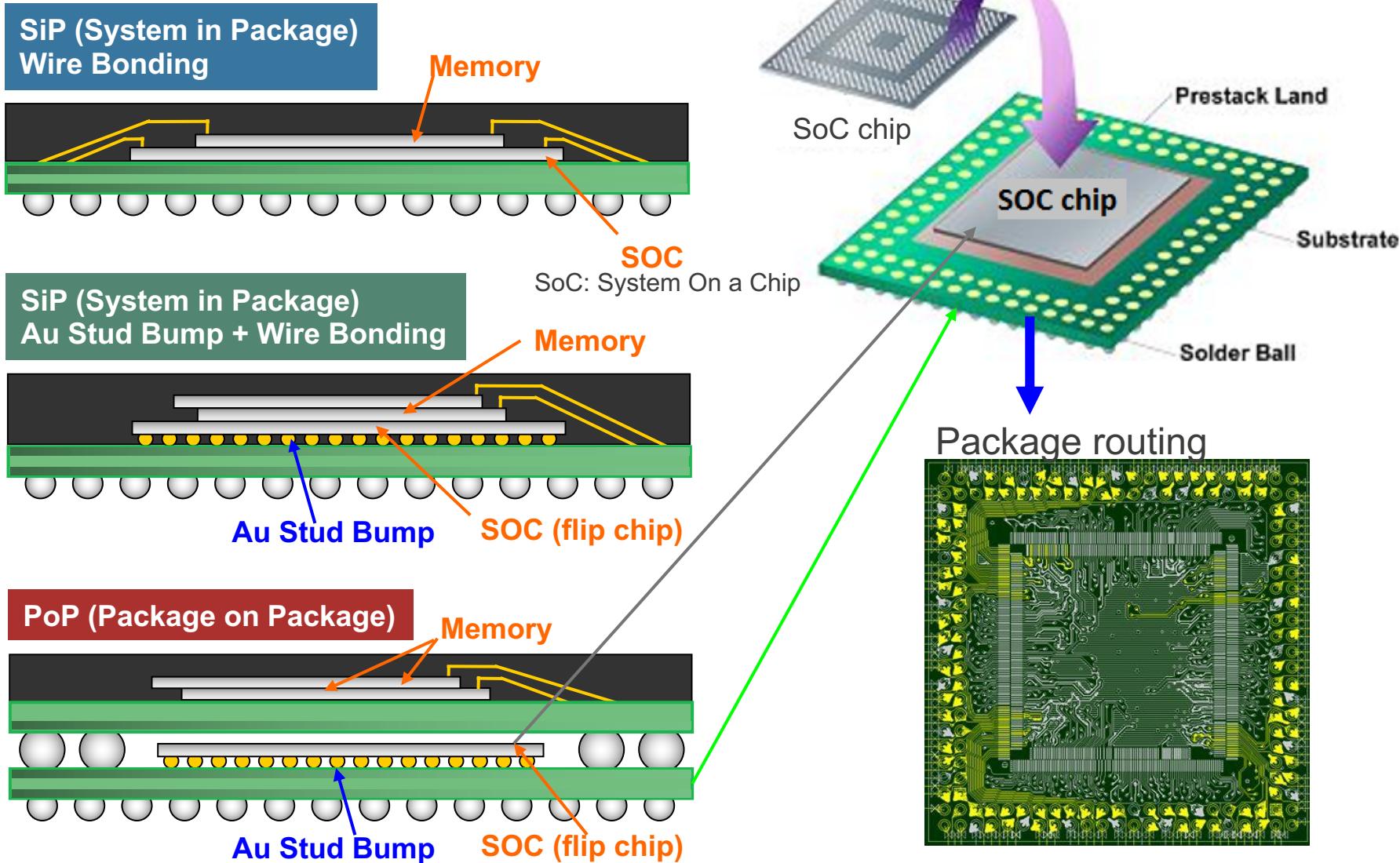
IP modules areas (memory and analog, etc., free placement)

All areas are for the wiring areas.
(wiring can run in free channels and through over cells)

Hierarchy block A and B are done PnR separately and are called into TOP design as a cell (Macro, block)

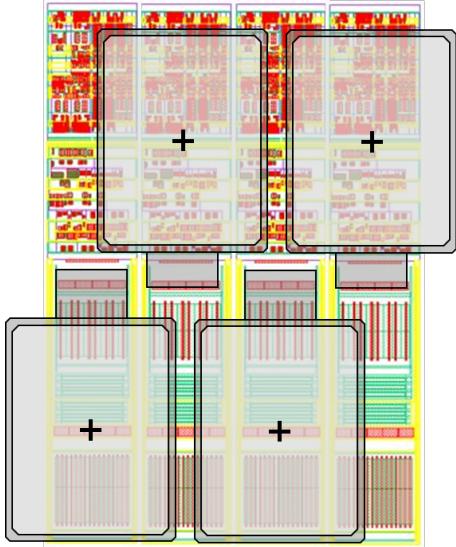
6.4 Chip Design Planning

Package Types



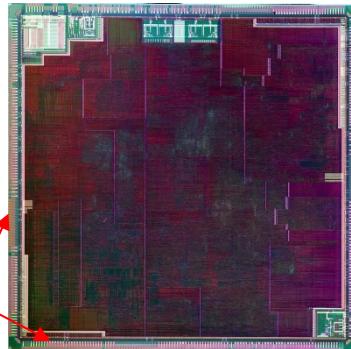
Flip Chip PAD Design

“STAGGER” Pad placement

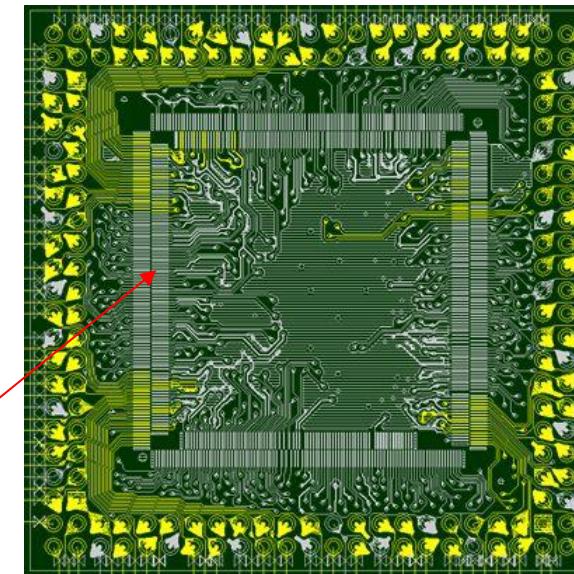


PAD is arranged around the chip

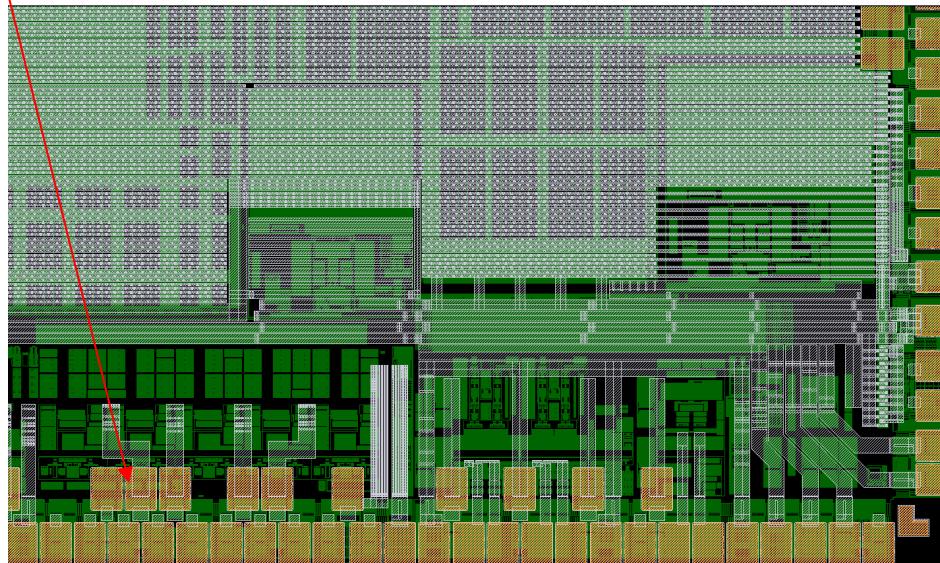
SoC using Stagger PAD



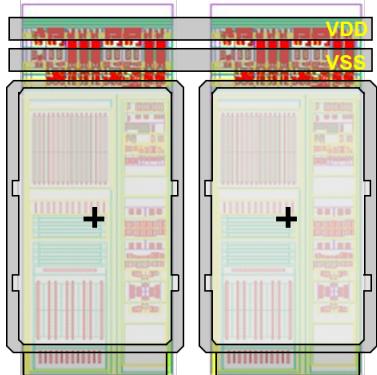
Example of package routing for Flip Chip PAD



(Layout view) PAD arrangement in Chip Layout

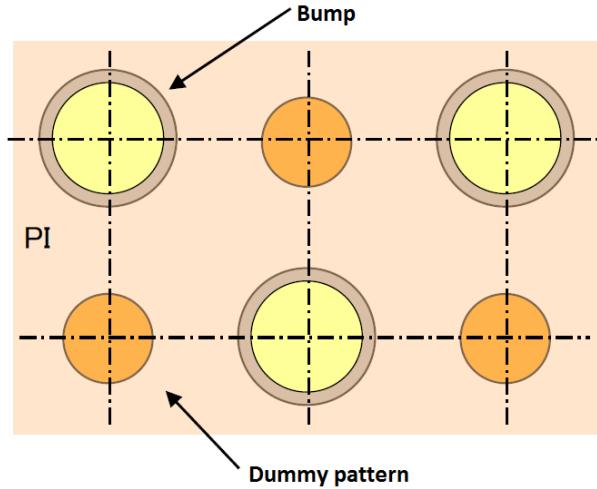


“STRAIGHT” Pad placement

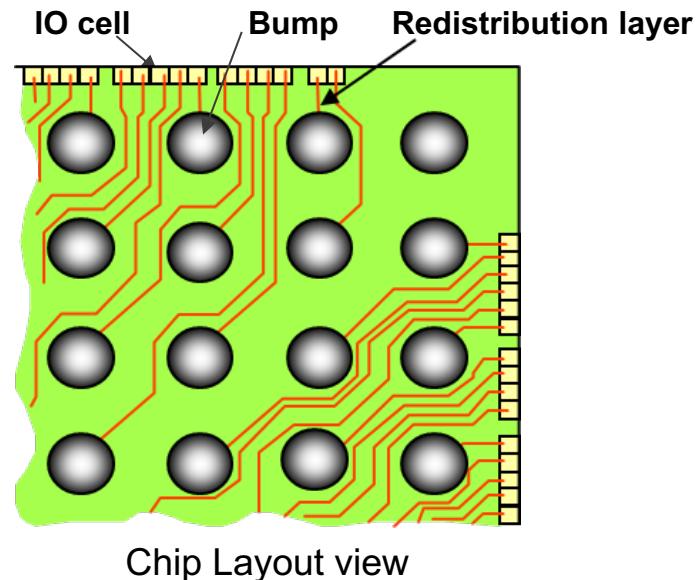
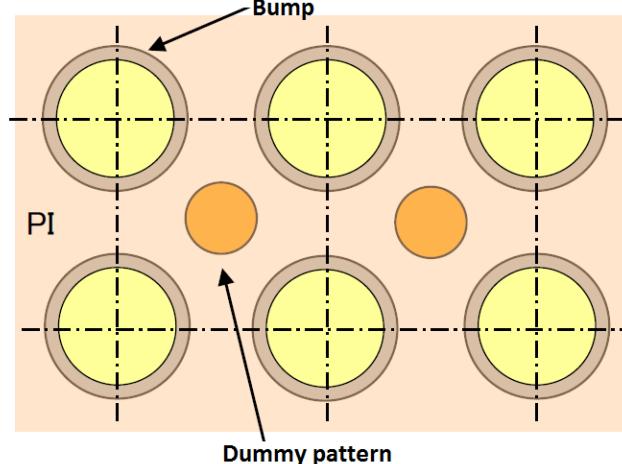


Flip Chip BUMP Design

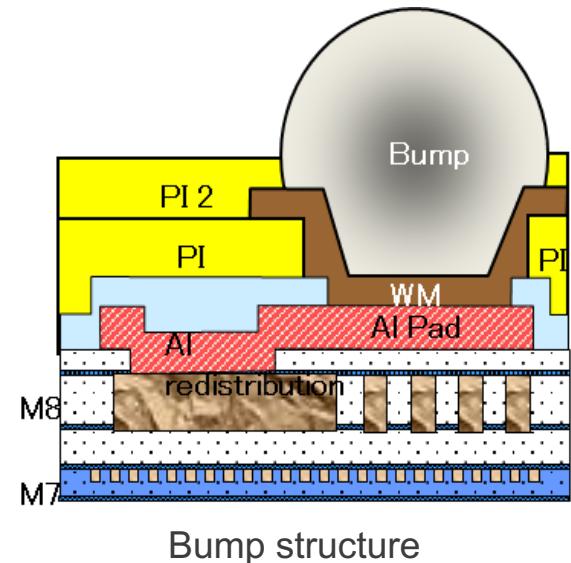
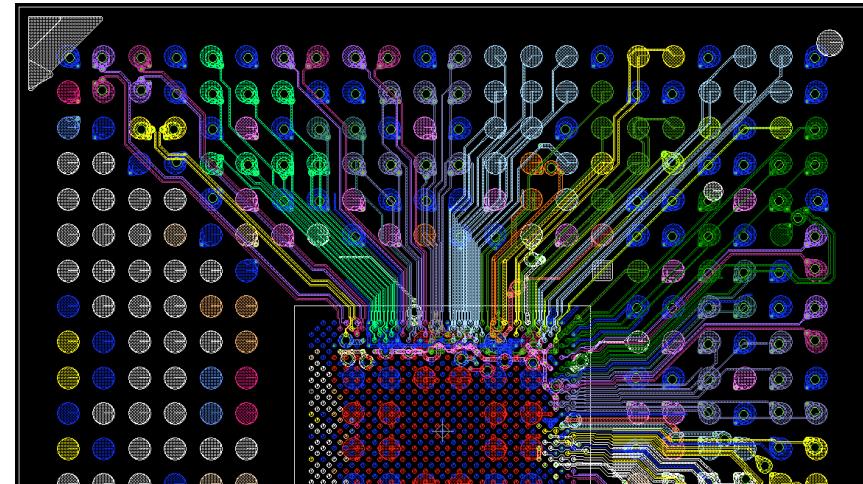
“STAGGER” Bump placement



“STRAIGHT” Bump placement

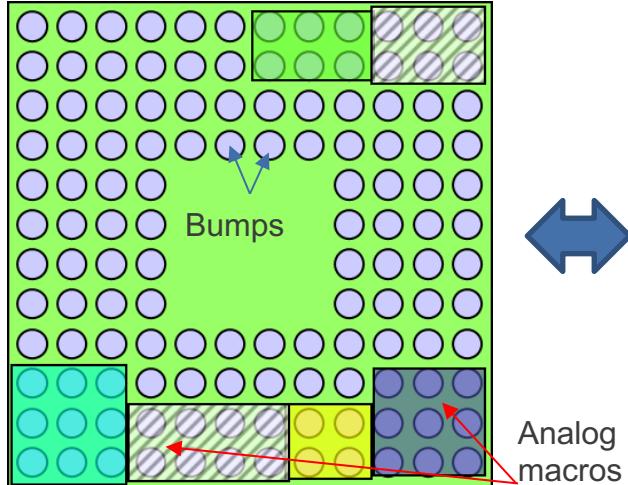


Example of Package routing for Flip Chip Bump design

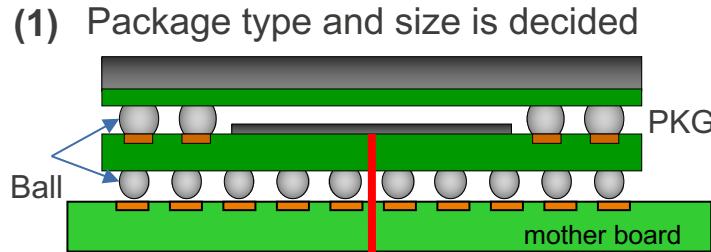
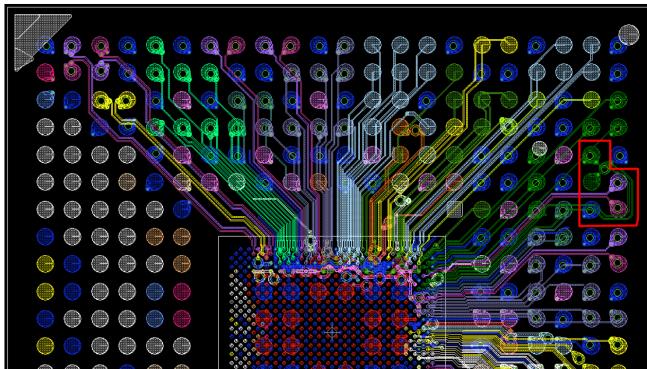


Chip Size and BUMP/PAD Planning

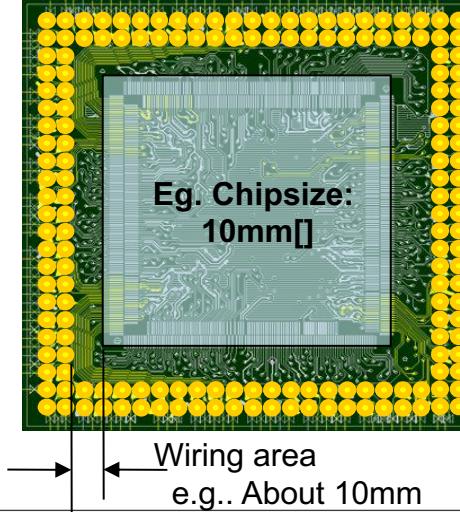
- (4) Block size is estimated to feedback to chip size decision
- (3) Analog Hard macro is planned. Bump/Pad is roughly arranged.



(Package routing is trialed and feedback to bump/pad arrangement)



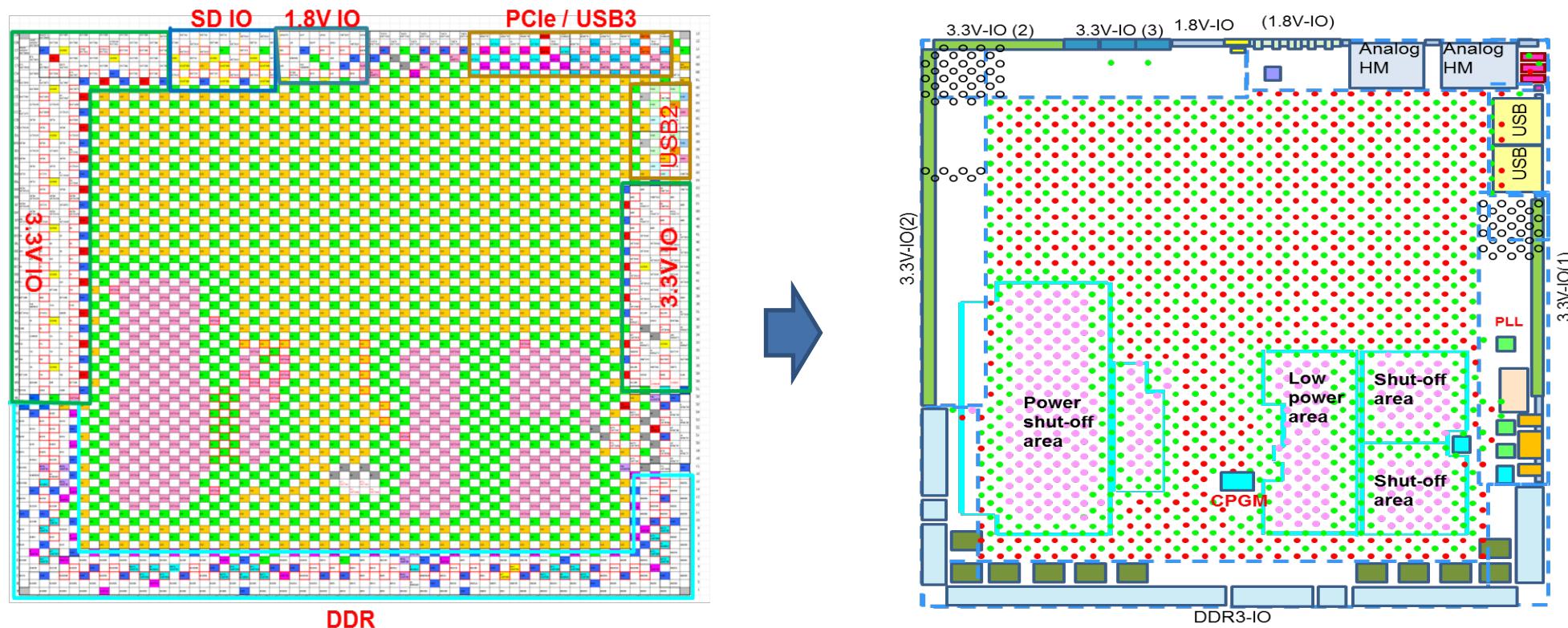
- (1) Package type and size is decided
- (2) Chip size is targeted. Package routing is planned



BUMP Assignment - Block Placement - Power Arrangement

After chip size is decided and position of Analog macros are planned:

- Make BUMP assignment and Block placement
- Arrange Power areas and their position



Early Power Analysis

Power consumption for each block is estimated.

Rough power analysis (IR drop, power distribution, ...) is executed.

- Bump assignment may be adjusted
- Power structure and Power supplying plan are made

- Big voltage drop:
- More PG bumps?
 - Block position?
 - PG supply?
 - How to reduce drop?

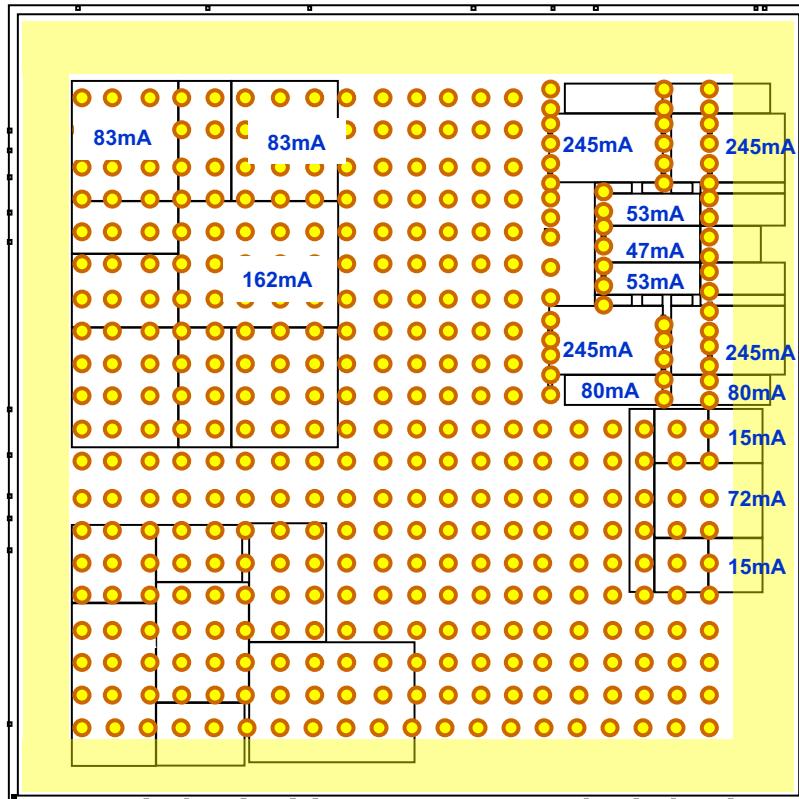


Fig VDD/VSS Core bump Image

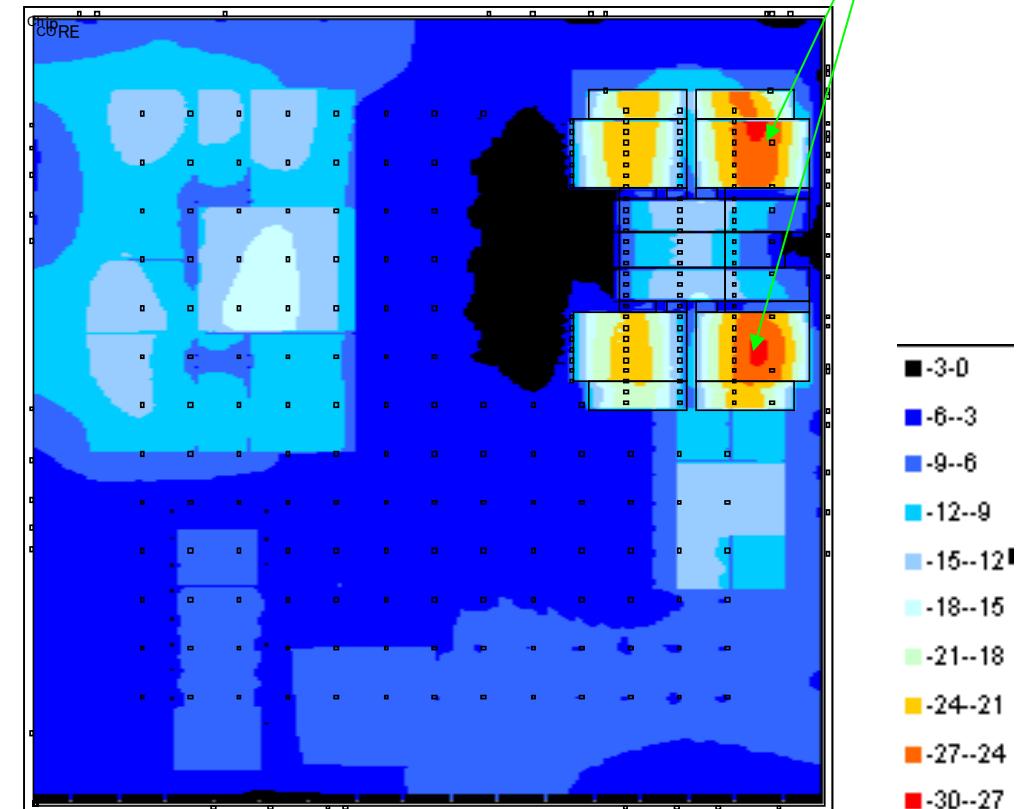
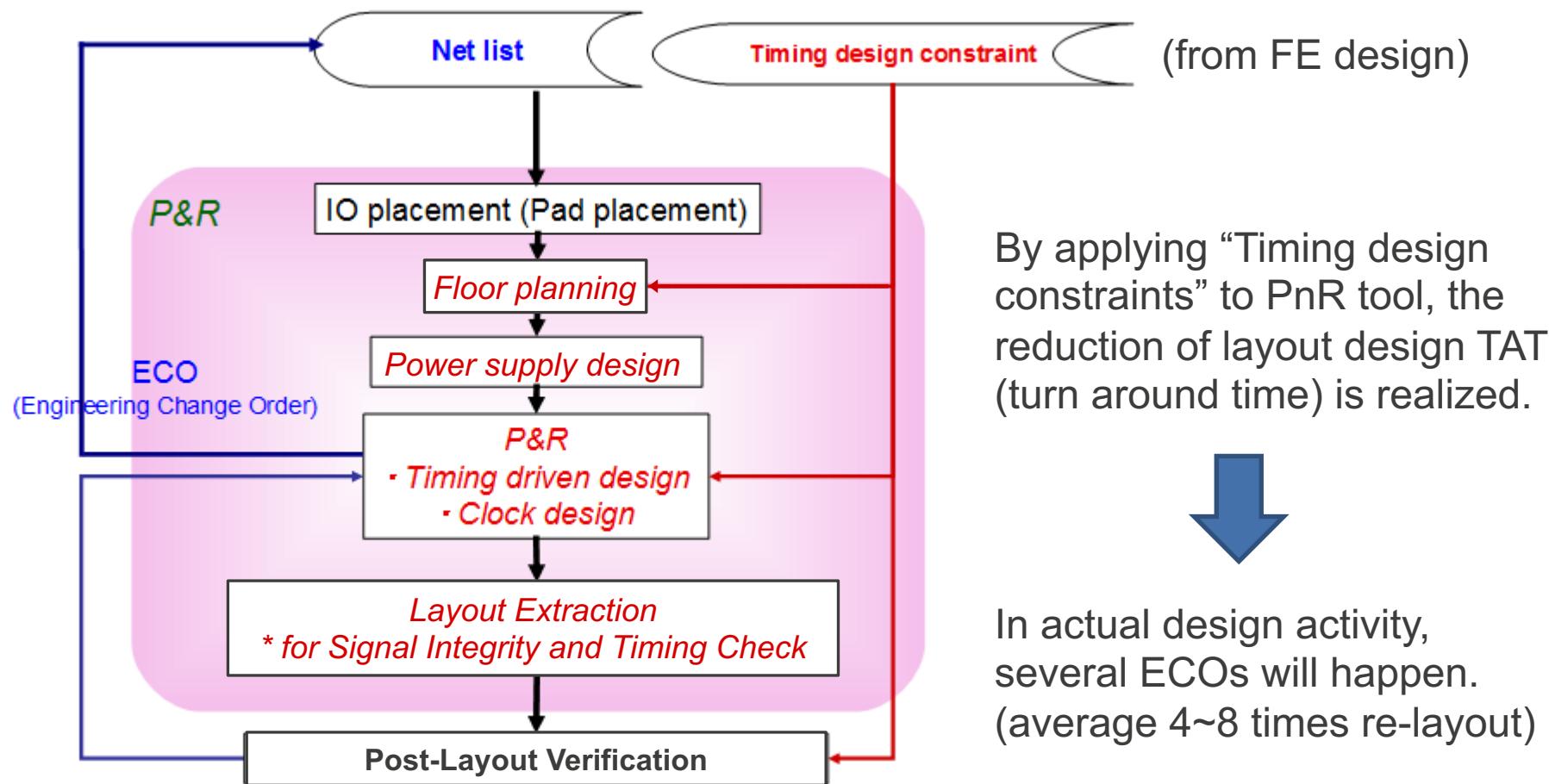


Fig temporary IR-Drop

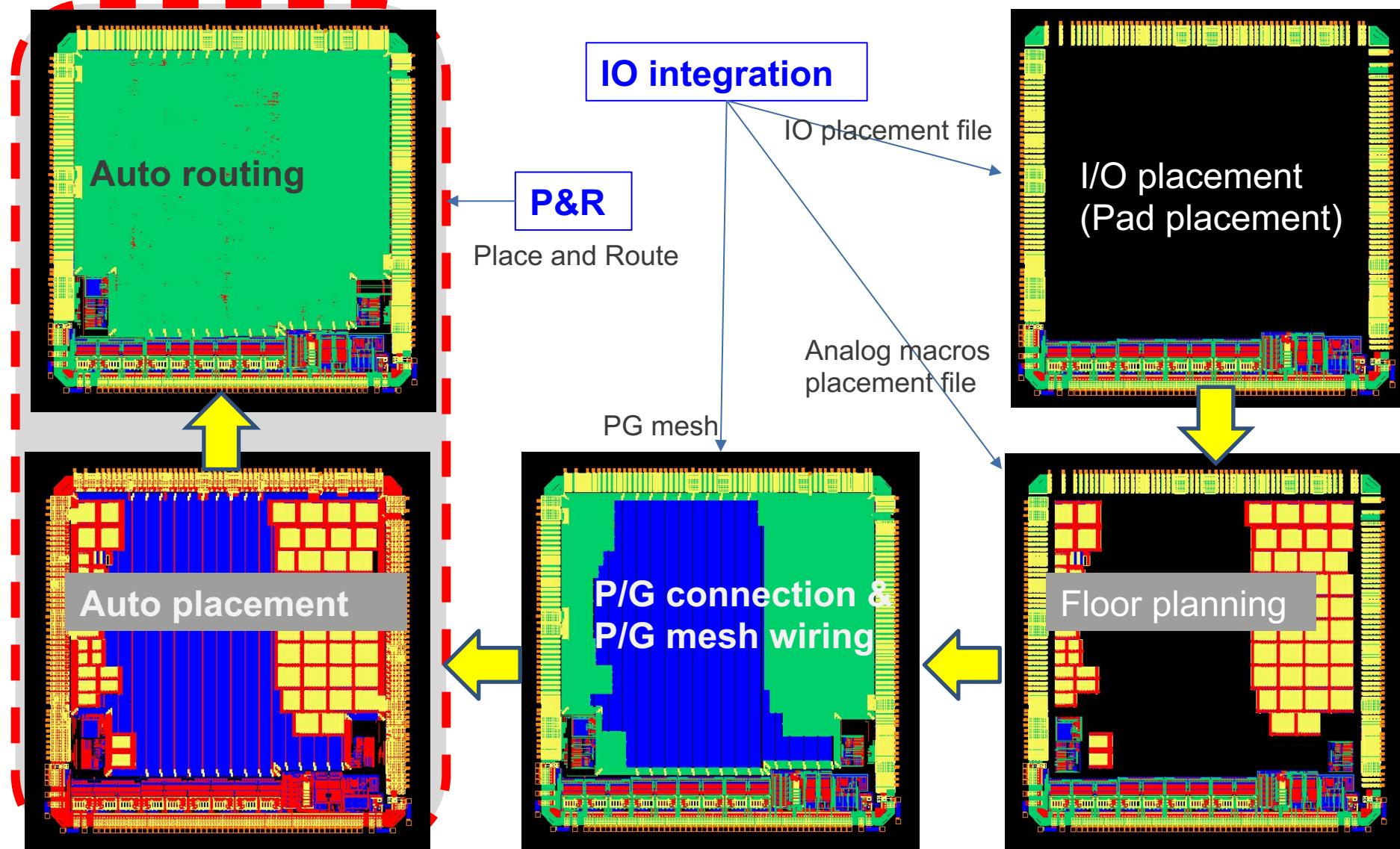
6.5 Block and TOP Level Layout Design

PnR (Placement & Routing) General Flow



PnR General Flow

(cont'd)



Floor Planning

In case of Hierarchical PnR design:

- Chip TOP Floor Planning: decide size (area) and shape for each child block.
- Each block is applied the whole design cycle (FP → place → route → timing and signal integrity check) .
- **Block interface model** is made for each block. Block size and shape is updated to/from TOP design.
- TOP design will use **block interface model** for its design cycle (FP → Place → Route → Timing and Signal integrity check).

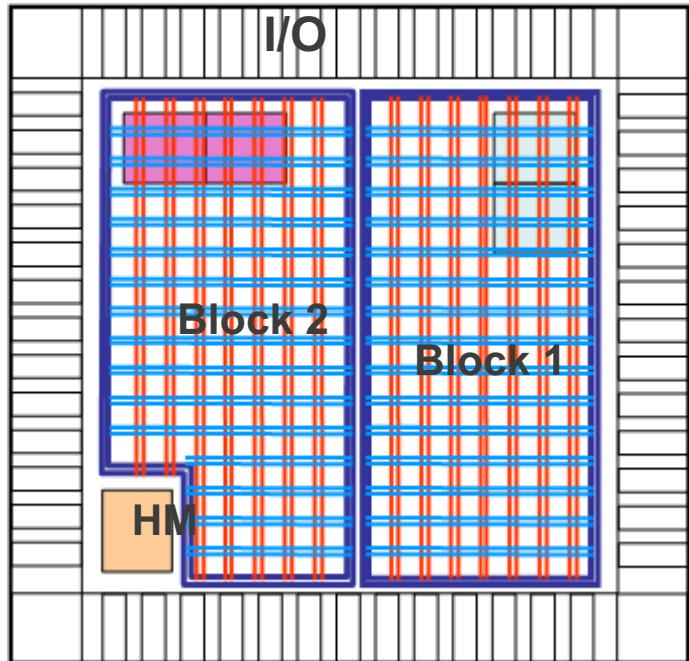


Top Floor Plan:

- Hierarchical blocks, Hard macros, and I/O cells are placed.
- The remained area (in gray) is reserved for logic cells on TOP level.

Power Supply Design

“Power supply design” targets to establish power routing to keep specifications of design rule (for voltage drop, electro-migration, etc..) and to supply sufficient current to each cell, considering to minimize the chip size.



<Design approach>

1) Manual design

The designer decides **number of wire straps** and their **width**, then uses layout editor and the P&R tool to draw PG mesh **manually**.

2) Semi-automatic design

Use Angel@Ring (Renesas in-house tool) and pre-determined power routing cells.

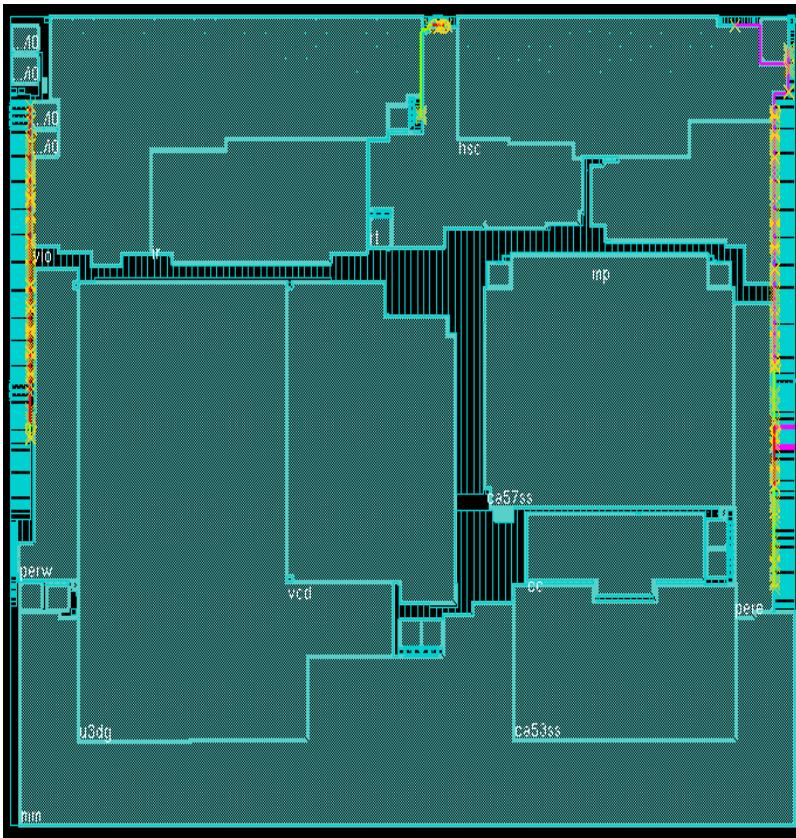
3) Automatic power design tool

Developed and implemented to EDA tools
EDA tools will decide PG mesh pattern, number of wire straps, metal width, ... and do the PG routing **automatically**

PnR - Placement

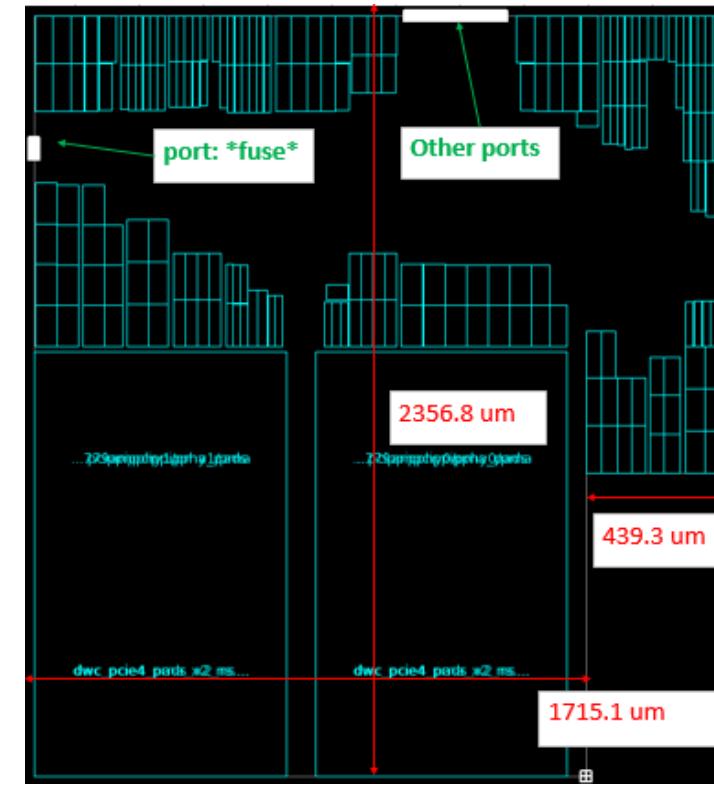
TOP Placement

- IO, Analog macros, and RAM/ROM are placed
- Block shapes are decided.
- Power/Ground grid is designed.
- Position to place pins for each block is planned.



Block placement

- RAM/ROM, Analog macros are placed
- Module ports are placed
- Power/Ground grid is implemented.



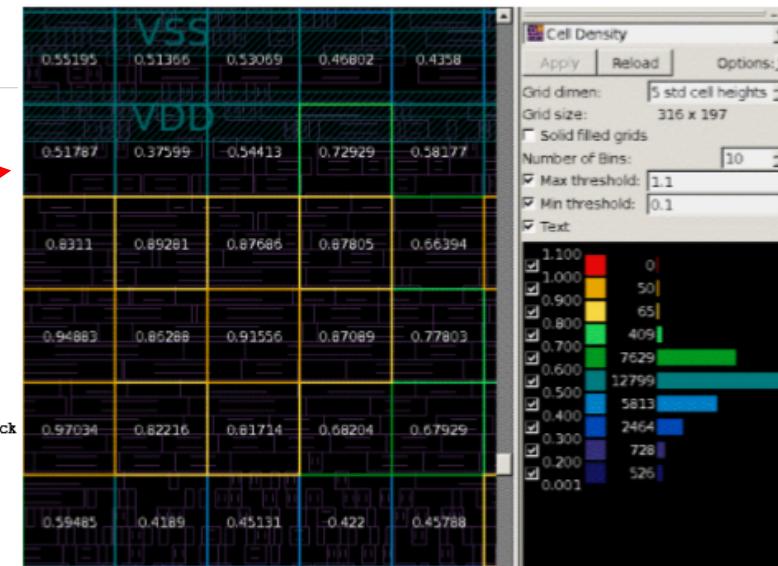
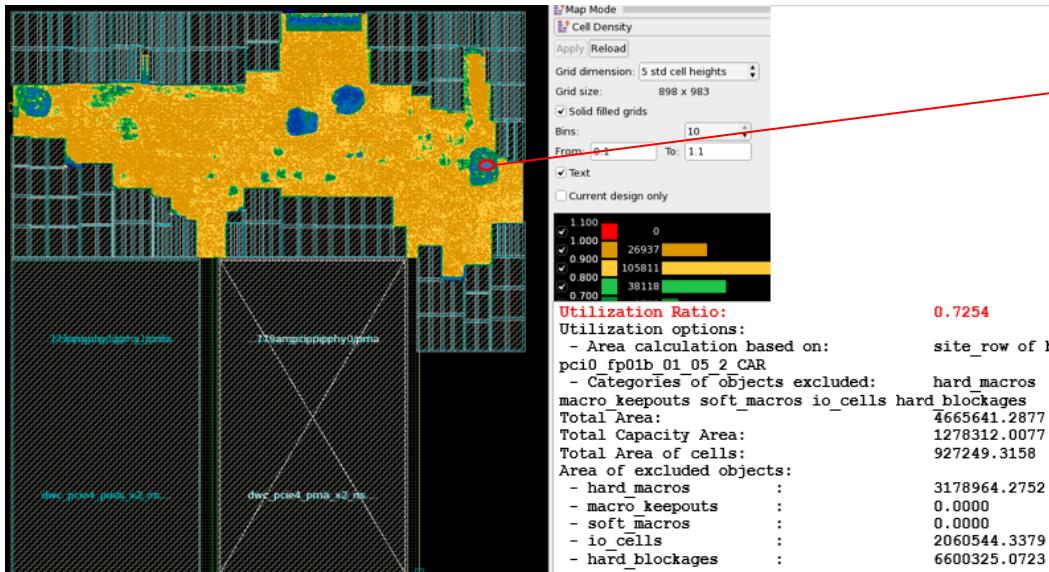
PnR - Placement Optimization

Prerequisite: Design after floor-planning.

In a **timing-driven design**, placement and optimization step performs:

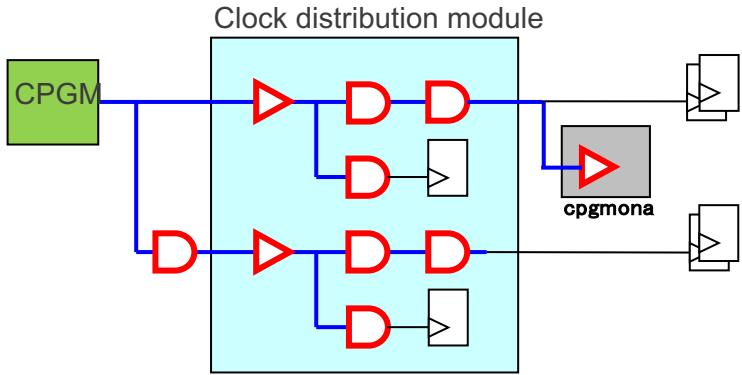
- Logic standard-cells are automatically placed by EDA tool.
- A virtual ideal-clock-tree is build for timing optimization.
- Connections are virtually routed
- Logic can be optimized and standard cells can be changed and/or moved to other places to resolve routing congestion, reduce timing violations, or to perform power optimization.
- Design rule check (DRC) and timing check are performed and violations are resolved automatically, e.g.. by replacing or moving cells .

Example of Cell density map

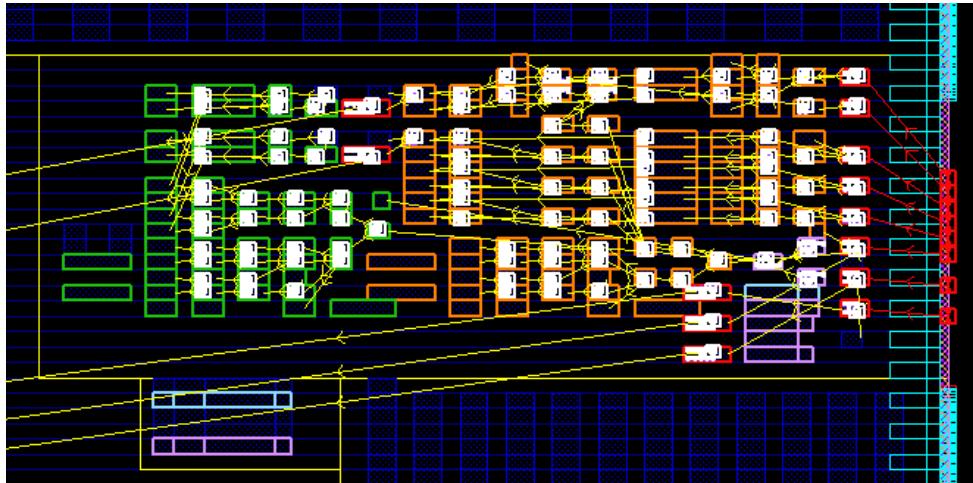


PnR - Clock Design

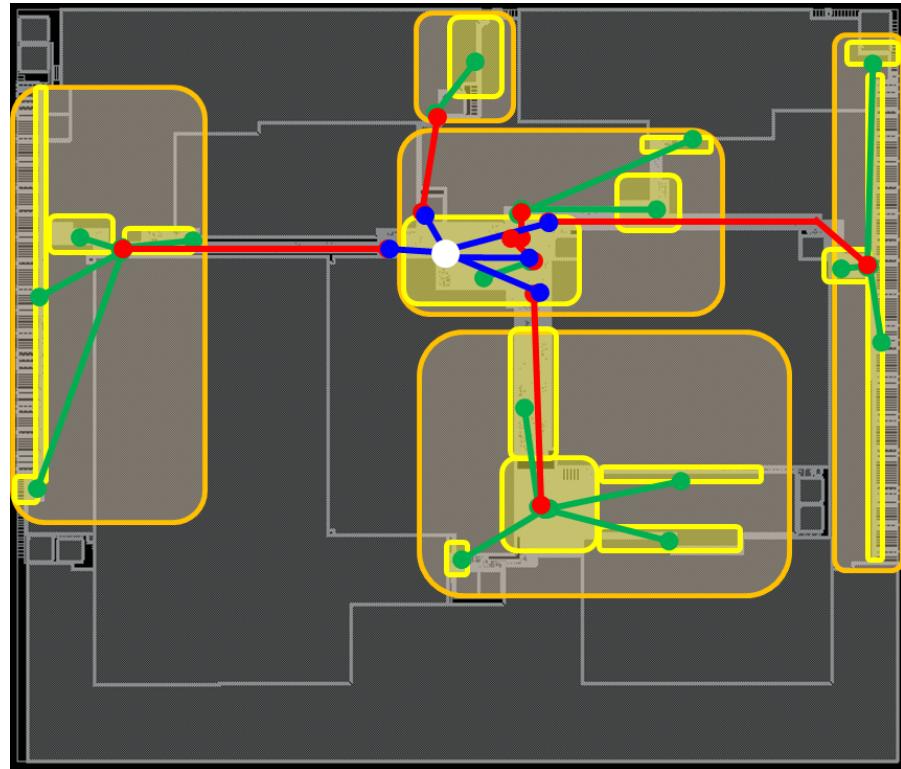
Analyze clock structure



Manually place clock cells of clock distribution modules



Insert main clock buffers of the clock tree

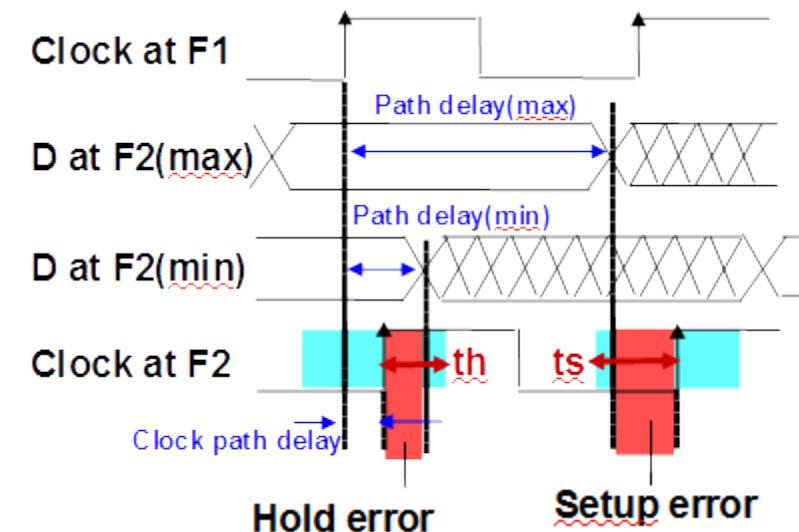
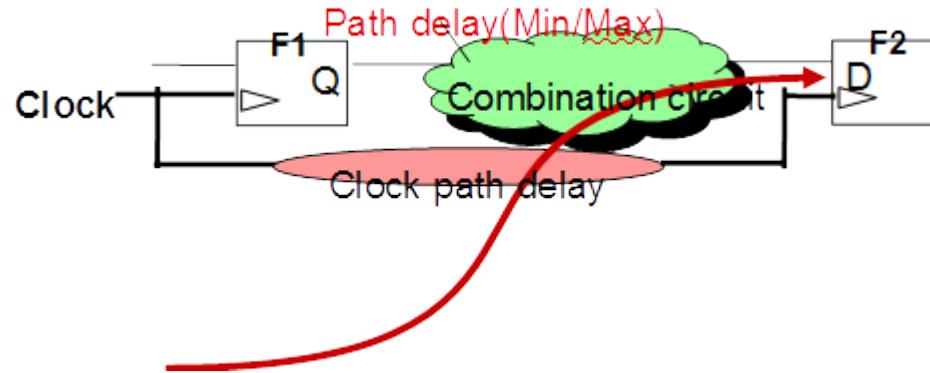


PnR - Timing Driven Design

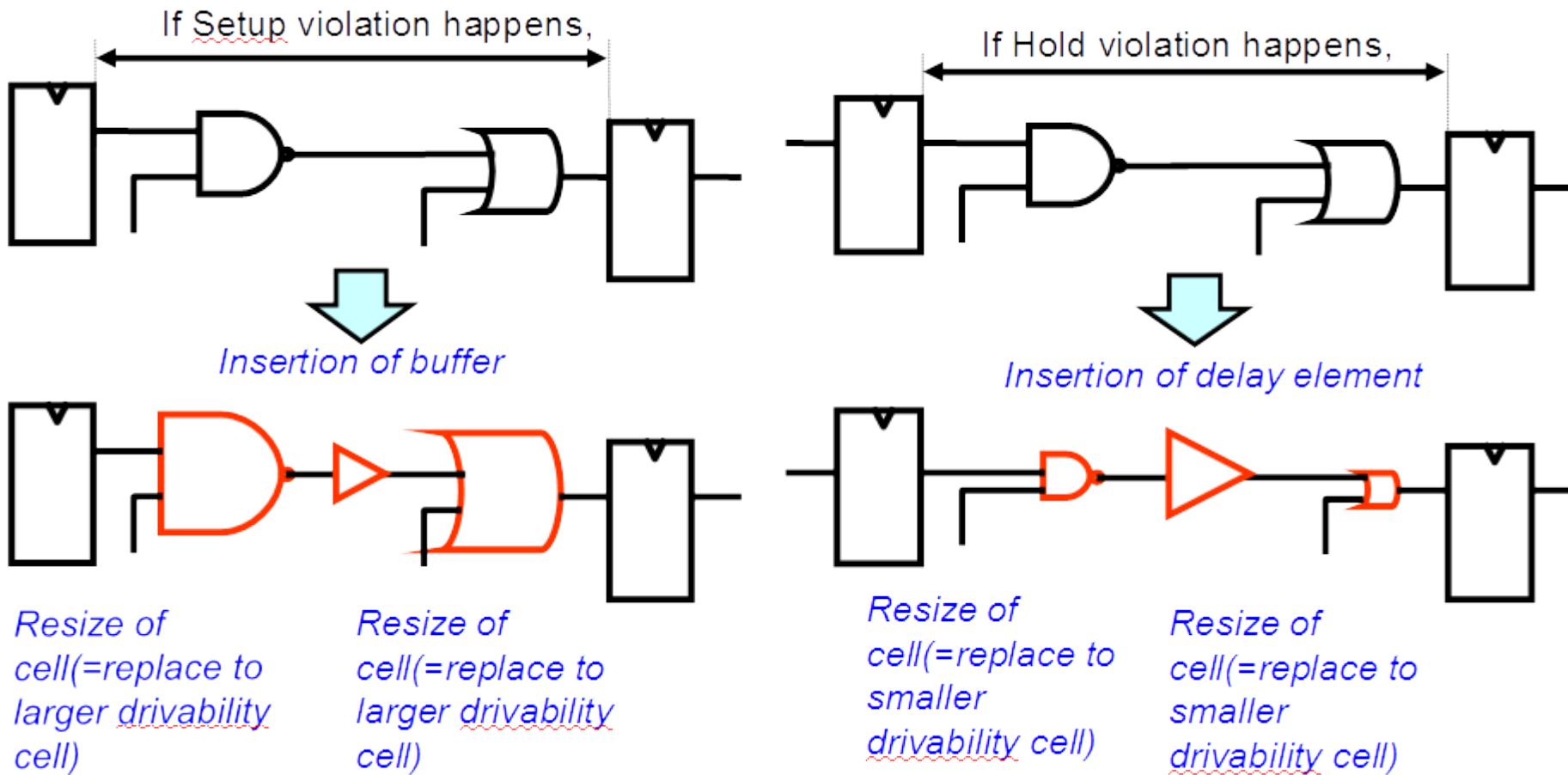
- Timing driven design means that Automatic P&R tool designs layout considering **timing constraints specified at logic design**.
- **Timing constraint** means that timing (path delay value, setup time, hold time) between clock port and data port of flip-flop.

Constraint of Setup time: Data should reach by (the reach time of clock- ts)

Constraint of Hold time: Data should not change by (the reach time of clock + th)

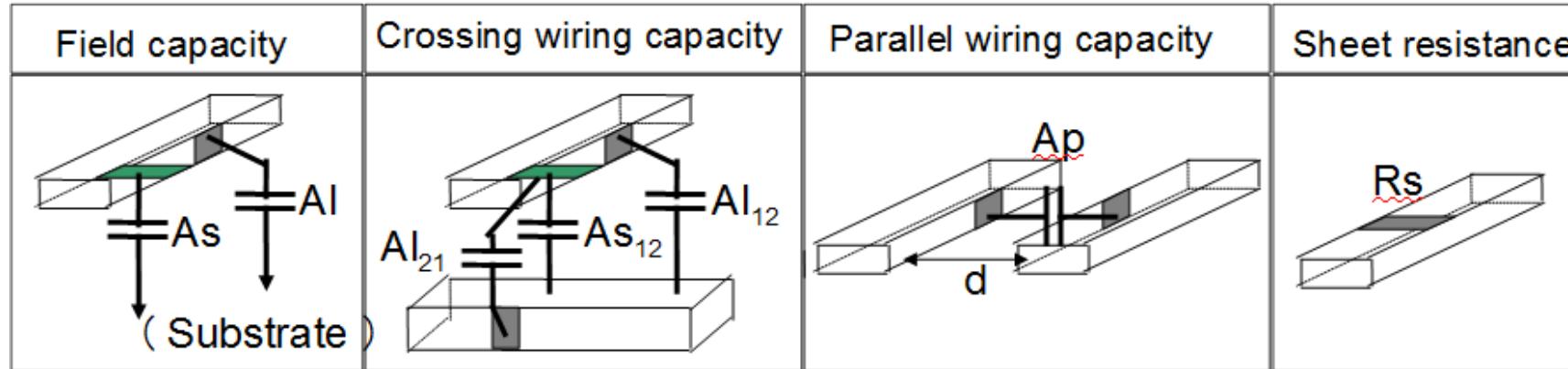


PnR - Timing Driven Design (cont'd)

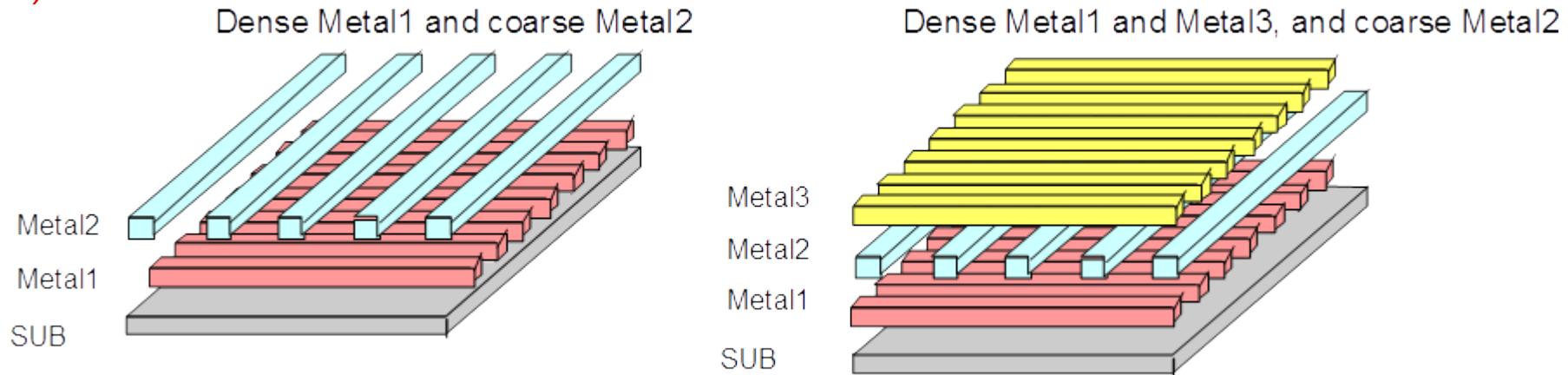


Layout Extraction (for Signal Integrity and Timing Check)

RC extraction means that the extraction of wiring loads (stray capacitance and resistance) from layout after P&R.

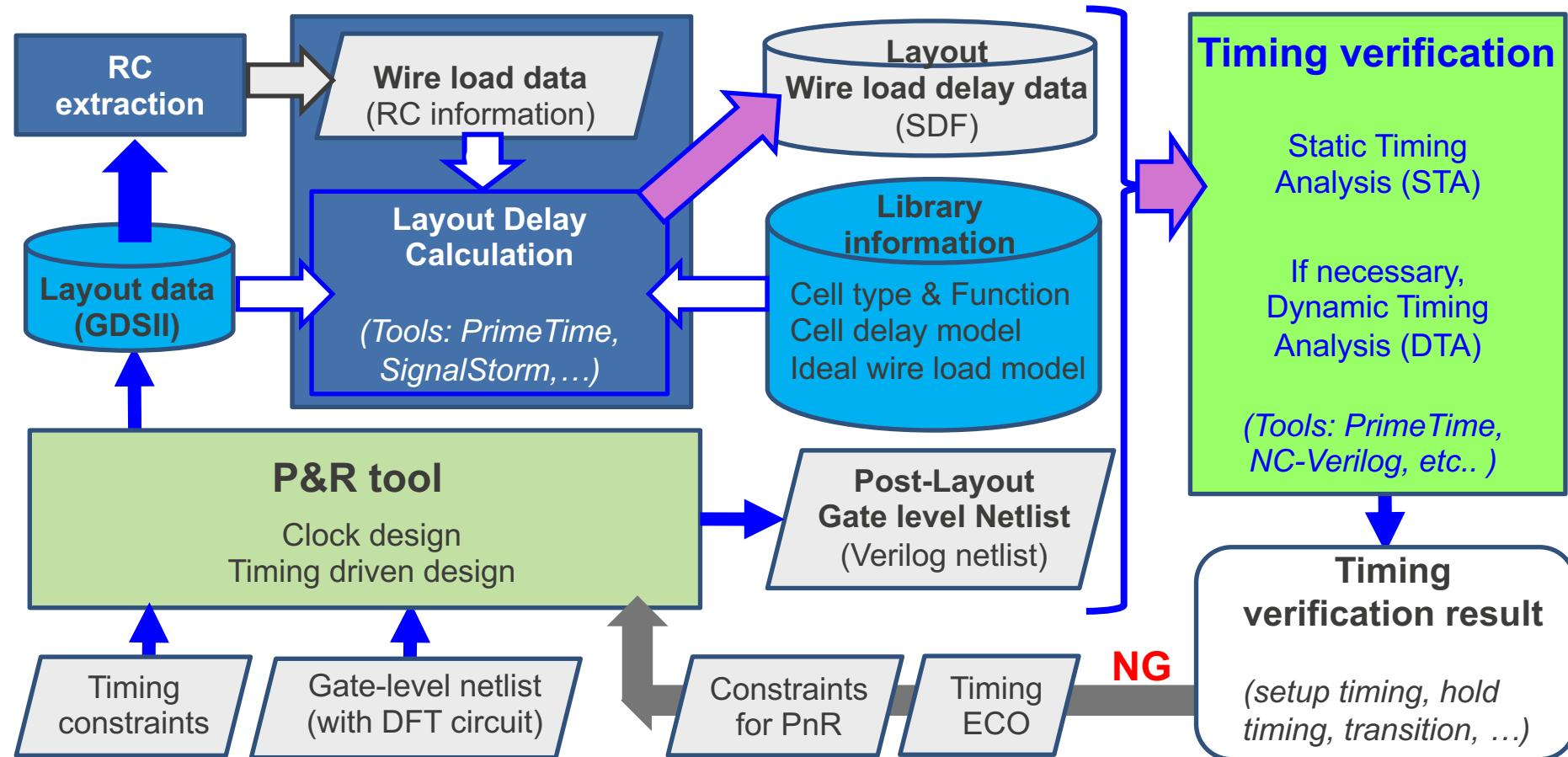


The parasitic capacitance changes depending on peripheral wiring (with/ without adjacent wire, top and bottom, coarse or dense).



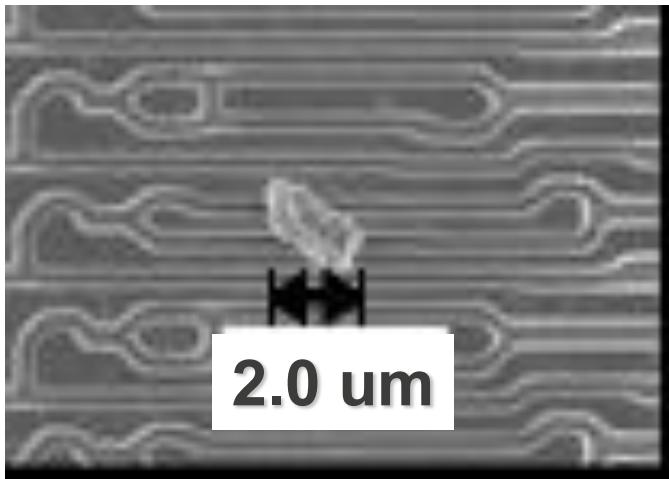
Timing Verification

Timing verification is done based on **post layout delay data** extracted from post layout wire load.
Minimum accuracy error to actual LSI (Si) is required.



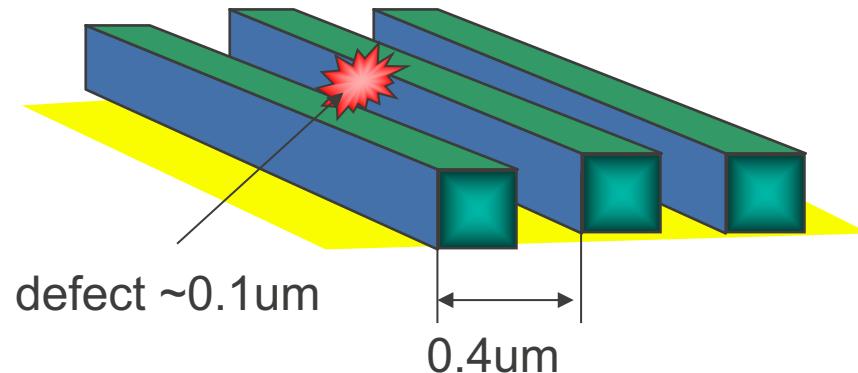
6.6 Process Defect and Yield

Patten Defect and Yield

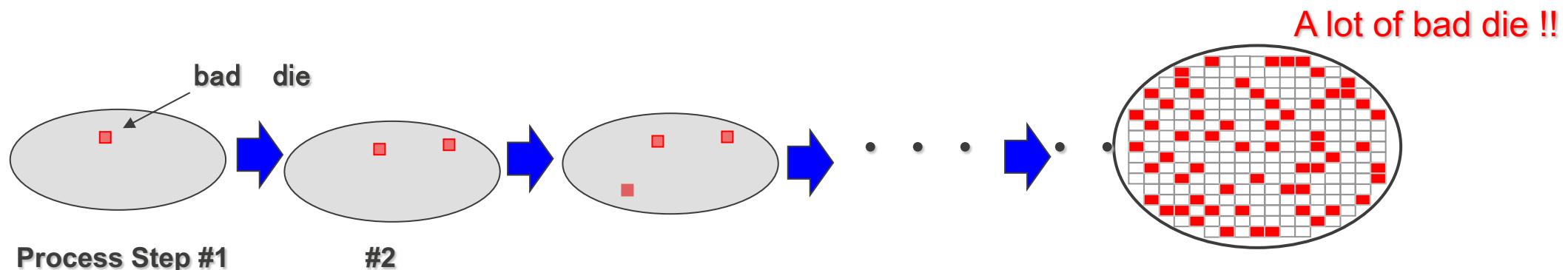


Interconnect
short !!

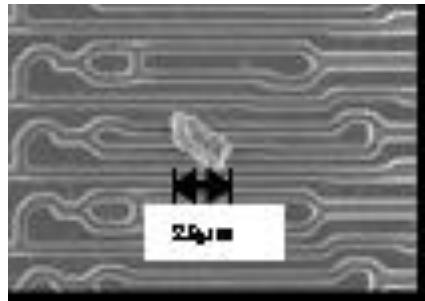
⇒ bad die



Even if we have 1 defect/wafer/step, Important steps > 100



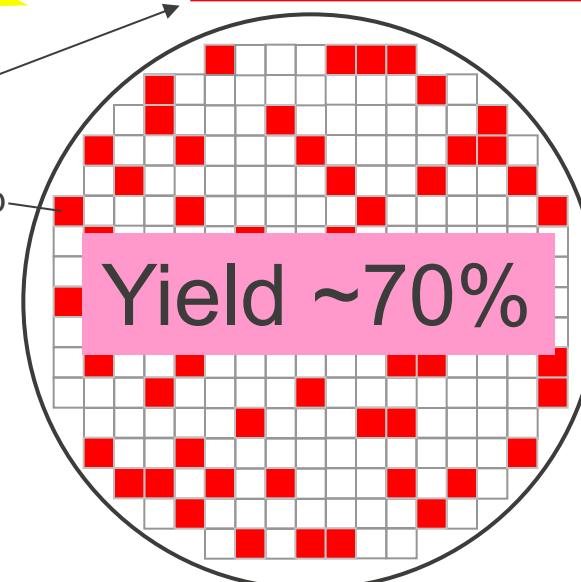
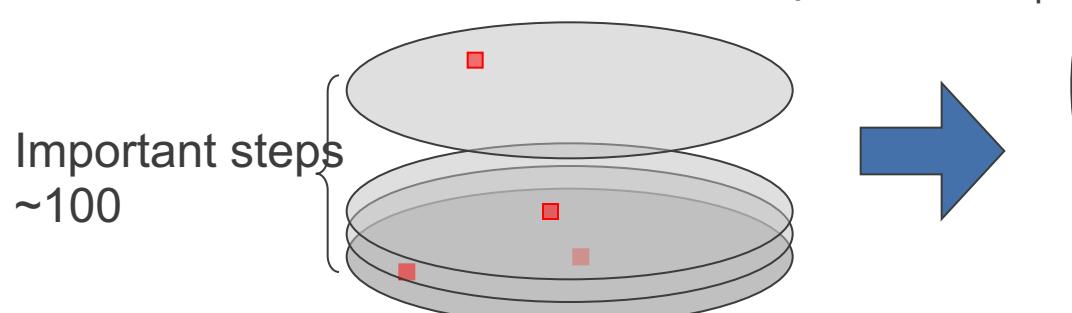
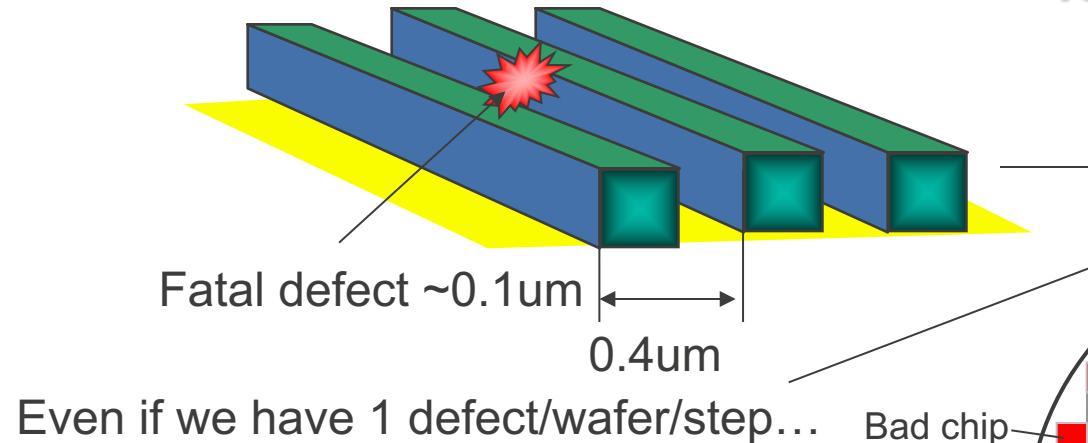
Patten Defect and Yield (cont'd)



$$\text{Yield} = \frac{\text{Number of Good Dies}}{\text{Total Number of Dies}} \times 100$$

To analogy, in defect size on a wafer:

1 piece of foreign particle such as a tip of hair dropped in the area of 216mx216m.



Contamination Reduction

Human is one of the origin of contamination.

To reduce the contamination from human, we put a special cloth and all wafer process steps are done in the **clean room**.



Figure 4-9 Configuration of typical modern cleanroom for IC fabrication. Photo courtesy of Stanford Nanofabrication Facility.

From SILICON VLSI TECHNOLOGY

Yield Expression

If the good chips can be gotten randomly in a wafer, the yield is expressed by the following formula.

$$Y = \exp[-(S \times D)]$$

Where **S** is the area of the chip (cm^2) and **D** is a defect density($/\text{cm}^2$)

Example:

$$D = 1 / \text{cm}^2$$

$$S = 6.0 \text{ mm} \times 6.0 \text{ mm} = 36 \text{ mm}^2 = 0.36 \text{ cm}^2$$

$$Y = \exp[-(0.36 \times 1)] = 0.6976 \Rightarrow \text{Yield} = 69.8\%$$

Renesas.com