

BACKEND TERMINOLOGIES

Contents

71 What is meant by "xx nm technology"?	5
77 What are technology files which used in PnR flow?	5
76 What is ETM? Distinguish .lib and .db library?	6
90 What is NLDM and CCS library? When to use CCS and when to use NLDM?	7
89 Library with Tap – Library without Tap (HoangTrinh-san – Add purpose)	10
79 What is Sigma (σ)? Six Sigma Process (6σ)? What are the differences among various Renesas's sigma libraries?	11
88a Single and double height cells	12
88b In-bound library and Out-bound library (HoangTrinh-san- Add characteristic, Merit/Demerit)	13
36 How many types of cell based on Voltage Threshold?	14
37 Advantages and disadvantages of using different Vth cells types?	14
39 What is the difference between SPEF, DSPF, SDF?(DatThai-san)	15
50 Explain ECR, ECO	16
73 What does PrimeTime, Formality, Calibre, Hspice, HLDRC, PP check, Conformal –LEC , Conformal-LP do? (Lien-san, Dam-san- PV tool list)	16
87 What is DEF, LEF? (TrungMai-san)	18
59 Don't touch, don't use, size-only setting?	18
96 Spare cell - filler cell – power switch cell	19
101. Explain N-well, DN-well	20
80 Power Consumption (Tri-san)	21
1 Explain input/output and meanings in Floorplan step? (Ref: ICC2 student guide)	22
74 Describe the required input library for Layout implementation?	23
2 Explain input/output and meanings in Placement step?	23
3 Explain input/output and meanings in CTS & postCTS step?	23
4 Explain input/output and meanings in Routing step and route_opt step?	24
5 What is DFM & Chip finishing step?	24
6 Explain Yield? How to increase yield? (In chip finishing stage)	25
7 What is track, row, grid, unitTile, pitch, process?	25
62 Prefer routing direction and non-prefer routing direction	26
63 What is routing track?	26
8 Distinguish Link lib and Target lib (in ICC – ICC2 use)?	27
9 Design view (Layout view), Abstract view and FRAM view (ICC/ICC2 use)?	27
10 Hard-macro/Soft-macro and Standard cell	28

11 Sign-off constraint.....	29
14 DFT Circuit	30
15 What is Hierarchical Design? Advantage & Disadvantage?	31
16 What is timing interface constraint (I/O Budget)?	33
21 What is Die, Core? What is the difference between die size and core size?.....	33
85 Pad limited, core limited and package limited (Dam-san).....	34
17 What is utilization? Which factors will affect utilization? Relationship between Die size and utilization.	34
18 Why we don't make design with utilization 100% at first time?	35
82 What is UPF format?	36
72 Explain Power domain, Power Island, voltage area?	36
99. What is multi power design?	37
20 Bound, Module Boundary, ... ? Hoang-san	37
23 Placement blockage	41
65 What is Route guide? What is Route Blockage?	43
83 What is power-spec, IO pin assign? Bump assign? Purpose of these file? (Dam-san)	43
86 Abutted rows and non-abutted rows	44
97. PG strap/rail/ring	44
32 What is Repeater / HFS?	46
35 What are cell types for CTS and why we choose them?.....	46
28 Global skew vs Local skew	46
29 Sink pin vs Ignore pin	47
30 Stop pin vs Float pin vs Exclude pin	Error! Bookmark not defined.
25 Explain Clock Uncertainty and latency.....	49
61 Distinguish Detail route (DR) and Search & Repair (S&R).....	53
60 What is routing layer? Global and semi global layer?	54
98. Via vs contact (mat cat dung)	55
69 What is routing congestion? How to check and improve? (TriPhung-san)	55
66 What is Scan method, Scan FF, Scan chain? (Chua dang)	56
76 Why we need to make Scan chain reorder?	59
40 What is SDC file? Why we need SDC file?	60
34 Clock Duty Cycle, clock period?	60
24 Create_clock vs create_generated_clock (When create_clock? Experienced? STA document? virtual clock?) (HoangHoang-san)	61

42 Explain OCV value?.....	64
27 Distinguish Elmore, Arnoldi vs AWE mode	65
43 Cell delay and net delay?	67
26 Explain Slack, Slew (Transition time) vs Skew.....	67
31 Jitter vs Margin.....	69
41 What is operating condition and how it effects to the design?	70
45 MCMM (ICC2)? (Chua dang)	72
44 Clock re-convergence pessimism? (HungNguyen) (Chua dang)	73
46 PVT? Give 3 examples (Removed)	Error! Bookmark not defined.
51 Timing report, ideal mode and propagated mode	Error! Bookmark not defined.
48 What is size cell/ doping cell?	74
55 False path and True path	74
56 Launch path and Capture path	75
57 Min condition, max condition?	71
58 Explain timing “critical path” and “non-critical path”	75
70 What is timing path, ideal mode, propagate mode and how to read it?	76
64 What is timing driven & Congestion driven placement?	78
47 What will we do to meet timing/drive requirements? (Removed)	Error! Bookmark not defined.
38 How to improve setup by changing Vth? (Removed)	Error! Bookmark not defined.
52 What is setup/hold time? How to fix these violation	78
53 Transition time? How to fix transition violation	81
54 How to calculate the transition time	82
91 Explain DRC?	82
92 Explain LVS?	82
93 Explain Antenna?	82
94 Explain Density?	82
95. What is GDSII format? Why we need GDS merge?.....	82
100. What is PP check?	Error! Bookmark not defined.

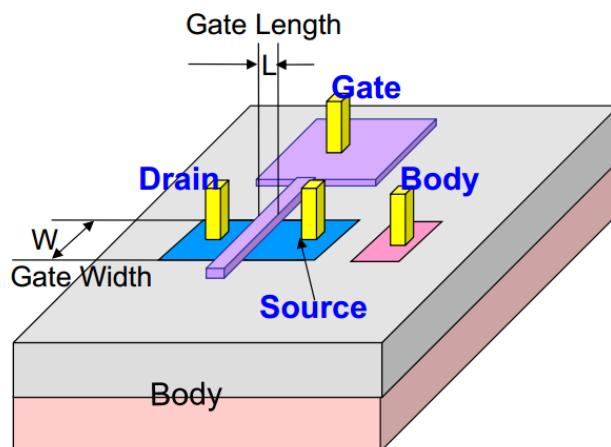
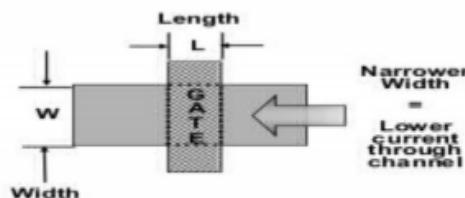
(Bổ sung sơ đồ khói cho PnR flow – ThyPhan)

71 What is meant by "xx nm technology"?

=> xx is the minimum length of the gate of the transistor. It means in the gate length of library cell which used in design, must be equal or greater than xx nanometer.

- ❖ In CMOS Technology the um or nm dimension refers to the channel length, and a minimum dimension which is fixed for most devices in the same library.
- ❖ Current flow or drive strength of the device is proportional to W/L; Device size or area is proportional to W x L.
- ❖ The “xx nm technology” is “used library cell” with the minimum gate length is xx nm and all of the rule included with library such as spacing, pitch ...

Example: 16 nm technology → the length of the transistor is 16nm



77 What are technology files which used in PnR flow?

There are 2 kinds of technology files: Physical technology file (.tf/.tlef/ ...) & Timing technology file (TLU+ file).

- ❖ **Physical technology file:** unique of each technology. It contains:
 - Layer and via definitions (number and name designations for each layer/via).
 - Dielectric constraint for each technology (Ex: dielectric = 3.7).
 - Process design rule for each layer/via (example: min wire width, wire-to-wire spacing).
 - Units and precision for electrical unit (Ex: time (unit of time (ns), timeprecision(1000), unitlengthname(micron), unitVoltage (v)...), capacitance, resistor).
 - GUI display info (colors and fill of layers).
- ❖ **Timing technology file** (TLU+: Table Look Up):

- PnR tool calculate delay of every cell and net. To do that, PnR tool needs to know each net's parasitic Rs and Cs.
- PnR tool uses the net's geometry and Look-Up Table to estimate the parasitic Rcs.

Example: The content of this file looks like the below information:

```

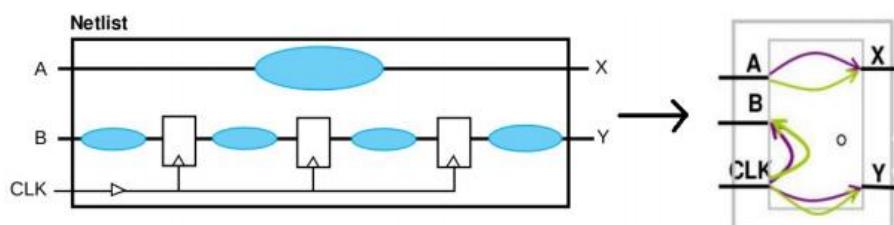
ETCH_VS_WIDTH_AND_SPACING {
    SPACINGS { 1.7640 3.5280 5.2920 7.0560 }
    WIDTHS { 1.7640 2.6460 5.2920 10.5840 13.2300 }
    VALUES {
        0.049000 0.049000 0.049000 0.049000
        0.049000 0.049000 0.049000 0.049000
        0.049000 0.049000 0.049000 0.049000
        0.049000 0.049000 0.049000 0.049000
        0.049000 0.049000 0.049000 0.049000
    }
}

RHO_VS_WIDTH_AND_SPACING {
    SPACINGS { 0.0490 0.1333 }
    WIDTHS { 0.0392 0.0529 }
    VALUES {
        0.3759 0.3759
        0.2823 0.2823
    }
}

```

76 What is ETM? Distinguish .lib and .db library?

- ❖ **ETM:** By using ETM, the original circuit will be modeled by another design containing a single leaf cell, the core cell is connected directly to the input and output ports of the model design. This cell contains the pin-to-pin timing arcs of the extracted model.



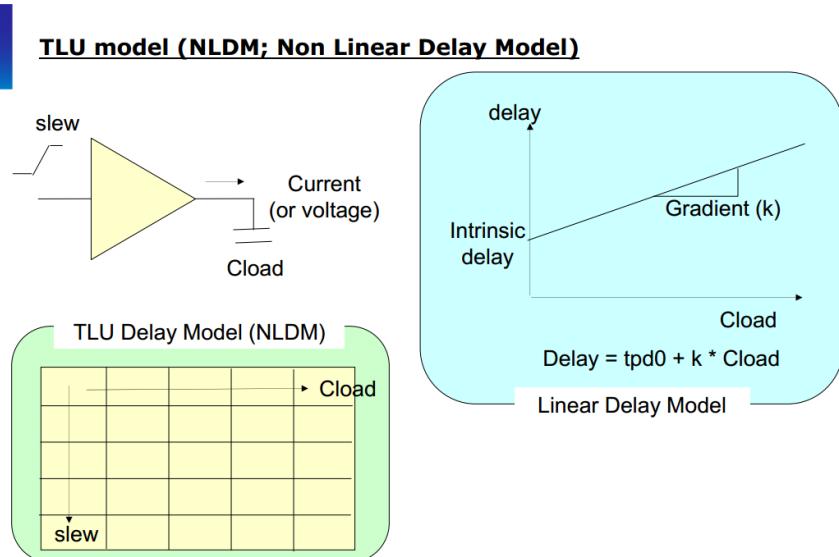
- ❖ **Distinguish .lib and .db library:**

These libraries have the same content just different format: .lib use text format while .db use binary format.

These libraries contain the logic information, timing arc of the standard cells, macros ...

90 What is NLDM and CCS library? When to use CCS and when to use NLDM?

- ❖ **Timing models:** Are intended to provide accurate timing for various instances of the cell in the design environment. The timing models are normally obtained from detailed circuit simulations of the cell to model the actual scenario of the cell operation. The timing models are specified for each timing arc of the cell.
- ❖ **NLDM:** Non-Linear Delay Model



Most of the cell libraries include table models to specify the delays and timing check for various timing arcs of the cell. The table models are referred to as NLDM (Non-Linear Delay Model) and are used for delay, output slew, or other timing check. The table models capture the delay through the cell for various combinations of input transition time at the cell input pin and total output capacitance at the cell output.

An NLDM model for delay is presented in a two-dimensional form, with the two independent variables being the input transition time and the output load capacitance, and the entries in the table denoting the delay. An example of NLDM for a typical inverter cell:

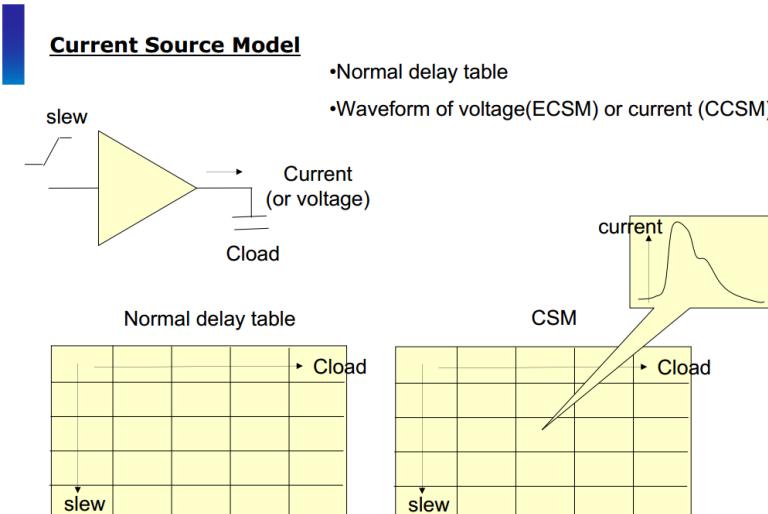
```

pin (OUT) {
    max_transition : 1.0;
    timing() {
        related_pin : "INP1";
        timing_sense : negative_unate;
        cell_rise(delay_template_3x3) {
            index_1 ("0.1, 0.3, 0.7"); /* Input transition */
            index_2 ("0.16, 0.35, 1.43"); /* Output capacitance */
            values ( /* 0.16 0.35 1.43 */ \
                /* 0.1 */ "0.0513, 0.1537, 0.5280", \
                /* 0.3 */ "0.1018, 0.2327, 0.6476", \
                /* 0.7 */ "0.1334, 0.2973, 0.7252");
        }
        cell_fall(delay_template_3x3) {
            index_1 ("0.1, 0.3, 0.7"); /* Input transition */
            index_2 ("0.16, 0.35, 1.43"); /* Output capacitance */
            values ( /* 0.16 0.35 1.43 */ \
                /* 0.1 */ "0.0617, 0.1537, 0.5280", \
                /* 0.3 */ "0.0918, 0.2027, 0.5676", \
                /* 0.7 */ "0.1034, 0.2273, 0.6452");
        }
    }
}

```

Example: An input fall transition time of 0.3ns and an output load of 0.16pf will correspond to the rise delay of the inverter of 0.1018ns.

❖ **CCS:** Composite Current Source



Higher accuracy by modeling the output stage of the driver **by an equivalent current source**.

As the feature size shrinks, the effect of interconnect resistance can result in large inaccuracy as the waveforms become highly non-linear. Various modeling approaches provide additional accuracy for the cell output drivers. Broadly, these approaches obtain higher accuracy by modeling the output stage of the driver by an equivalent current source.

The CCS timing models provide **the additional accuracy for modeling cell output drivers by using a time-varying and voltage-dependent current source**. The timing information is provided by specifying detailed models for the receiver pin capacitance and output charging currents under different scenarios.

In the CCS model, the non-linear timing is represented in terms of output current. The output current information is specified as a lookup table that is dependent on input transition time and output load.

The output current is specified for different combinations of input transition time and output capacitance. For each of these combinations, the output current waveform is specified. Essentially, the waveform here refers to output current values specified as a function of time. An example of the output current for falling output waveform, specified using *output_current_fall*, as shown below:

```
pin (OUT) {
    .
    .
    .
    timing () {
        related_pin : "IN"-;
        .
        .
        .
        output_current_fall () {
            vector ("LOOKUP_TABLE_1x1x5") {
                reference_time : 5.06; /* Time of input crossing
                    threshold */
                index_1("0.040"); /* Input transition */
                index_2("0.900"); /* Output capacitance */
                index_3("5.079e+00, 5.093e+00, 5.152e+00,
                    5.170e+00, 5.352e+00"); /* Time values */
                /* Output charging current: */
                values ("-5.784e-02, -5.980e-02, -5.417e-02,
                    -4.257e-02, -2.184e-03");
            }
            .
            .
            .
        }
        .
        .
        .
    }
}
```

The *reference_time* attribute refers to the time when the input waveform crosses the delay threshold. The *index_1* and *index_2* refer to the input transition time and the output load used and *index_3* is the time. The *index_1* and *index_2* (the input transition time and output capacitance) can have only one value each. The *index_3* refers to the time values and the table values refer to the corresponding output current. Thus, for the given input transition time and output load, the output current waveform as a function of time is available. Additional lookup tables for other combinations of input transition time and output capacitance are also specified.

(Picture)

When to use CCS and when to use NLDM?

The accuracy of NLDM library is highly dependent on EDA tools. Because the indices in NLDM are not completely match with the actual design, the EDA tools will do the interpolation to find the suitable timing value.

And each EDA tool has different interpolation algorithm. It means that different tools may export different timing value from the same NLDM library.

CCS library can help to avoid the differences between EDA tools.

Therefore, if we need high accuracy design, we should use CCS.

In some library (such as I/O library), CCS is not required. Because the timing margin for I/O area is usually large. And the number of I/O cell in 1 design is limited.

Therefore, the effect of accuracy is not much. NLDM library can be used instead.

89 Library with Tap – Library without Tap (HoangTrinh-san – Add purpose)

Content	Library with Tap	Tap-less library
Description	- The bulk terminals of the MOS are drawn inside each cell layout.	- The bulk terminals of the MOS are NOT drawn inside of the cell layout. They are drawn in a separated cell, so-called Tap cell.
Merit	- no need to make additional Types of cells - Low risk about latch-up	- Area reduction
Demerit	- Area cost	- some additional Types of cells are required :Tap cell, Row cell (or horizontal boundary cell) → it lead to complicated layout - High risks about latch-up → Tap cells must be implemented carefully
example (cross-section)		
Reference	1/. You can check which libraries using Tap or Tap-less structure in below link: (up to 28HPL process):	
	http://www.europрактиce-ic.com/libraries_TSMC.php	
	2/. latch-up-improvement-for-tap-less-library	



79 What is Sigma (σ)? Six Sigma Process (6σ)? What are the differences among various Renesas's sigma libraries?

- ❖ Sigma (σ) represents the population standard deviation, which is a measure of the variation in a data set collected about the process.
- ❖ Six Sigma Process (6σ) is a disciplined, statistical-based, data-driven management techniques intended to improve business processes by reducing the probability that an error or defect will occur.
 - If a defect is defined by specification limits (accept vs. reject outcomes) of a process, then a six sigma process has a process mean (average) that is six standard deviations from the nearest specification limit. This provides enough buffer between the process natural variation and the specification limits. (Figure 1)

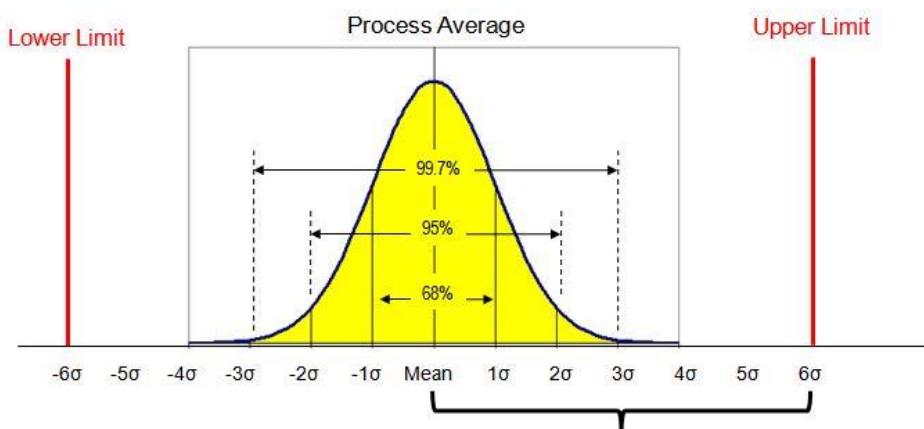


Figure 1: Sig Sigma Process with Mean and Upper/Lower Specification Limits

- ❖ Generally, processes or projects with higher sigma level requirements contain lower errors/defects, hence higher quality of product. However, higher sigma level requires stricter controls and more initial costs to produce.
- ❖ Renesas's sigma libraries are libraries with different sigma levels ($2\sigma/3\sigma$). Par default, Renesas products must satisfy 3σ level, equal 99.73% acceptance rate. In case the design cannot match timing constraints with 3σ libraries, 2σ libraries (which have more relaxed-timing constraints) will be used instead. However, this relaxation will increase the probability of defects in our designs (Probability of defects = 95.46%), reduce the quality of products.

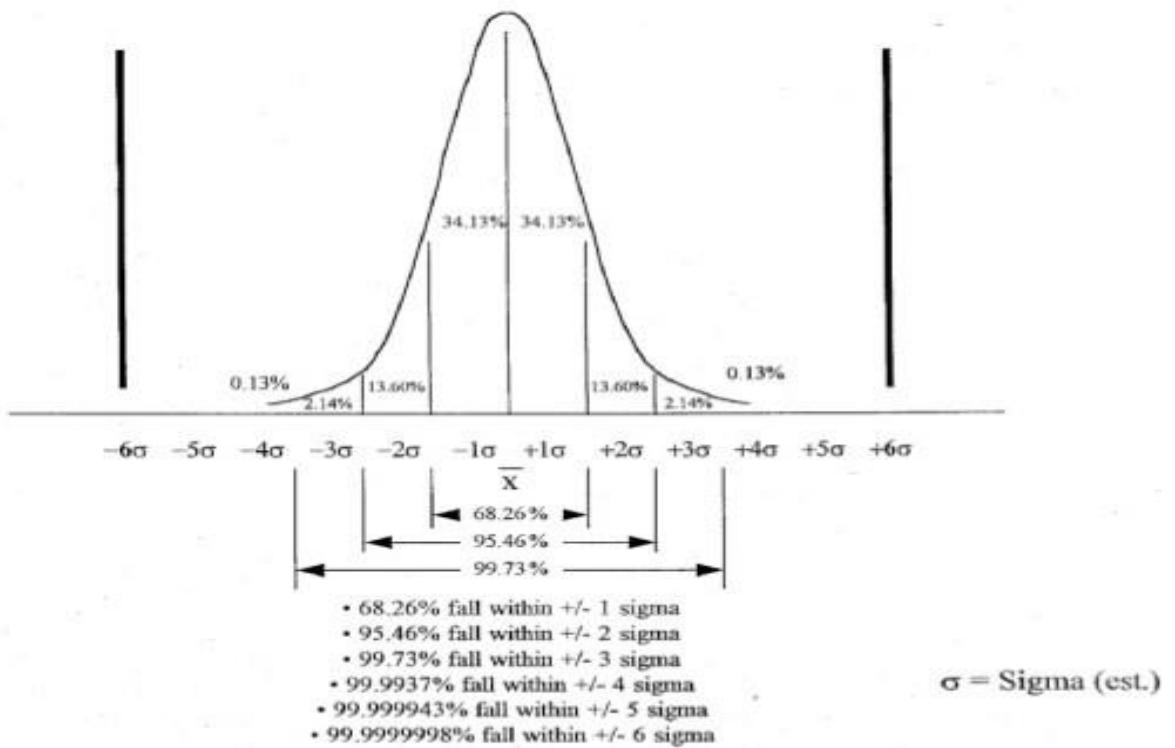
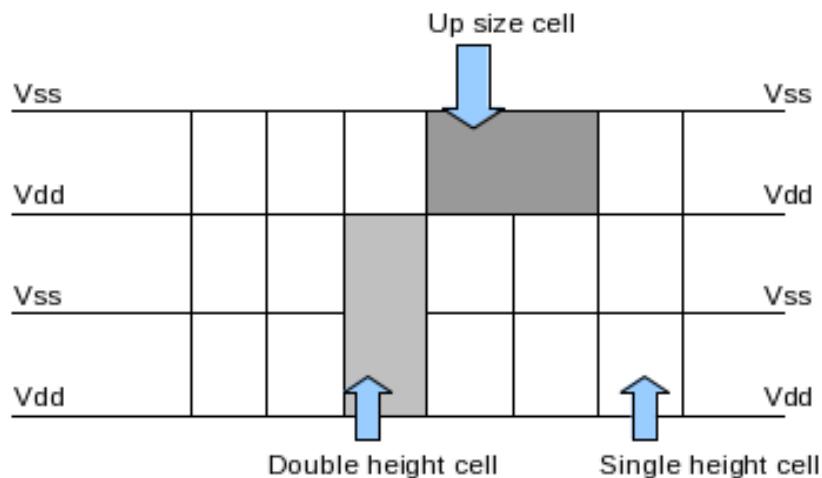


Figure 2: Sigma Levels and corresponding Probability of Success Rate.

88a Single and double height cells



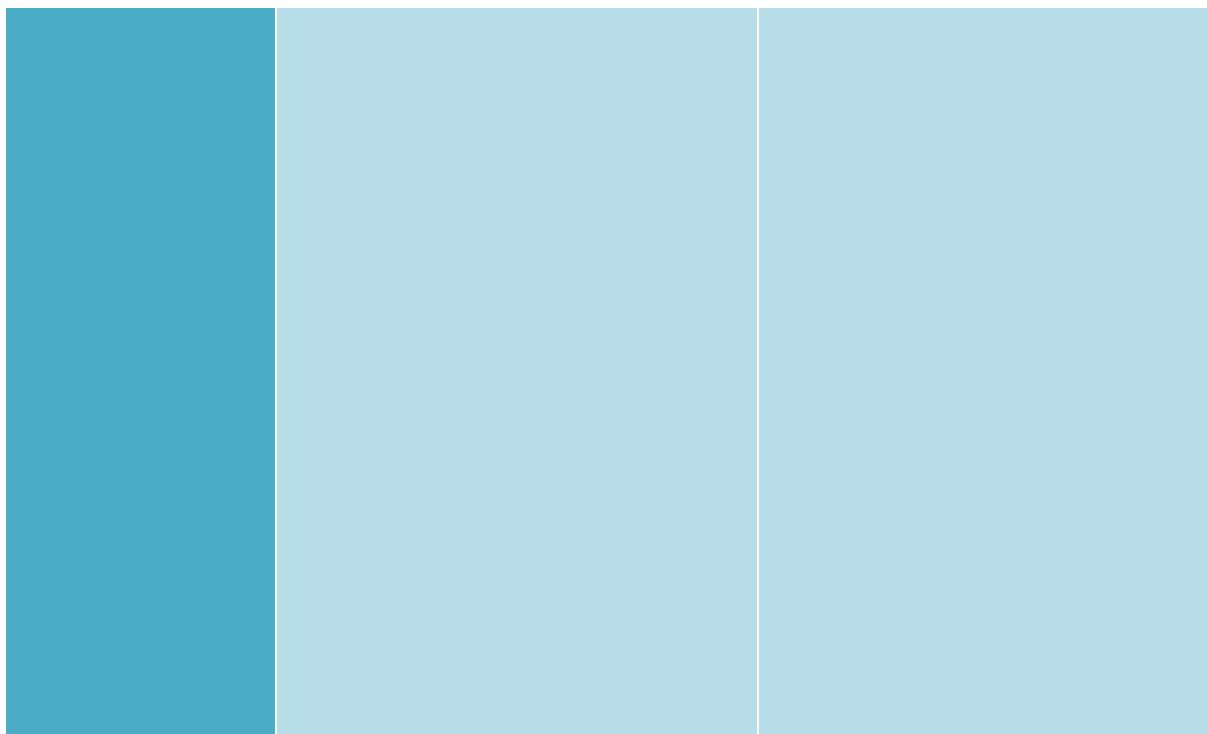
Single height cells are normal standard cells which height is matched with 1 row.

Double height cells are cells which height is double than single height cells.

Normally we use the double height cell for performance improvement (frequency) but it may cost the area or power.

88b In-bound library and Out-bound library (HoangTr-san- Add characteristic, Merit/Demerit)

Content	In-bound library	Out-bound library
Description	<ul style="list-style-type: none"> - All of the layout pattern MUST be inside of the cell boundary. - Normally, in-bound library is DRC-clean. Cells can stand alone. 	<ul style="list-style-type: none"> - Layout pattern CAN be stretched out of the boundary area in order to share the resource with other cells. - Normally, the out-bound library still remain DRC errors. Cells can NOT stand alone, they must be put abutment together to clean the DRC errors.
Merit	<ul style="list-style-type: none"> - DRC is clean for individual cells → PnR is easier. - MAY have more space to enlarge the MOS sizes → Speed of cells MAY be improved. - MAY have more space to improve the parasitic elements → better for noise prevention. 	<ul style="list-style-type: none"> - Using shared resource → big effect to area reduction.
Demerit	<ul style="list-style-type: none"> - Large area 	<ul style="list-style-type: none"> - DRC is NOT clean at cell level → PnR work will be complicated to clear the DRC. - Effect of parasitic elements may be large → sensitive to noise
Typical library	I/O, LSC, VSWC, Hard macro, some standard cell libraries	Power switch, most of standard cell libraries
Image (example)	<p>"Inbound" Cells</p> <p>VDD</p> <p>VSS</p> <p>Half row Fragmentation</p> <p>Boundary</p>	<p>"Outbound" Cells</p> <p>VDD</p> <p>VSS</p> <p>Boundary</p>



36 How many types of cell based on Voltage Threshold?

- ❖ Basically, we have some cell types as below:
 - ULVT (Ultra Low VT)
 - LVT (Low VT)
 - SVT (Standard VT) (In some processes, it's MVT- Middle VT)
 - HVT (High VT)
 - UHVT (Ultra High VT)
- ❖ The Speed of cell: UHVT < HVT < SVT < LVT < ULVT.
- ❖ The Leakage power of cell: UHVT < HVT < SVT < LVT < ULVT.
- ❖ The Driver ability of cell: UHVT < HVT < SVT < LVT < ULVT.
- ❖ Base on various technology, naming rule of each cell types are different. So, at the beginning of project, please study and investigate project's specification to have correct recognition.

For example,

Project starts with 28nm will use from UHVT to LVT.

Project starts with 16nm, will use from SVT to ULVT.

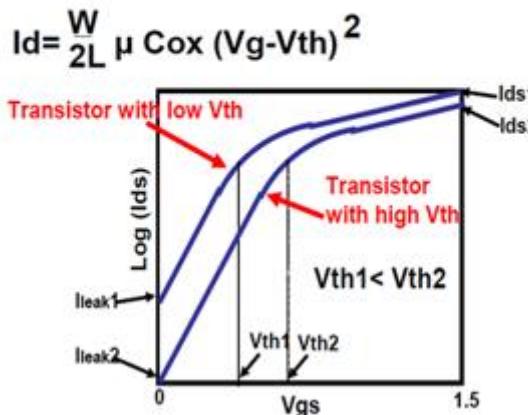
37 Advantages and disadvantages of using different Vth cells types?

- ❖ Advantages :
 - We can balance leakage power, speed and area by using multi VTH.
- ❖ Disadvantages:
 - Costly for manufacturing because we have many Vth types in design.

- Physical Verification have design rules accordingly.
- Notice about the stability of the design when using many VTH types. The best case when using 3 types.

Note: Depend on the purpose and requirement of the design for the power leakage or speed, we can choose the cell which have the best performance.

$$T_d = \frac{C_L \cdot V_{DD}}{\beta \cdot (V_{DD} - V_{th})^2}$$



39 What is the difference between SPEF, DSPF, SDF?(DatThai-san)

SPEF (Standard Parasitic Extraction Format)

DSPF (Detailed Standard Parasitic Format)

SDF (Standard Delay Format)

	SPEF	DSPF	SDF
Purpose	- Transfer design delay timing between design phases or EDA tools		
Content	- Resistance, capacitance parasitic for net - Coupling capacitance between different nets.	- Resistance, capacitance parasitic for net. - No coupling capacitance.	- Net delays , cell delays - Timing check , environment - Design, technology specific.
Size	- Smaller than DSPF because the names are mapped to integers to reduce file size.	- Bigger than SDF.	

50 Explain ECR, ECO

ECR: Engineering Change Request

- ❖ ECR is a file which contain commands for changing the netlist to fix logic bugs or other purpose. These changes may cause the logic function to change (In few cases, it doesn't change the logic function).
- ❖ The ECR is released mostly from FE designers, and sometimes from ME designers.
- ❖ BE designers apply these requests to the design and then timing-fixing and/or physical verifications are redone to validate these changes.

ECO: Engineering Change Order

- ❖ Engineering Change Order (ECO) contains two types of files: Logic ECO (same meaning as ECR) and Timing ECO:
 - Logic ECO contains the commands for changing the logic structures of the design. These changes may cause the logic functions to be different from the original design.
 - Timing ECO contains commands for changing the netlist to fix timing or other purposes. Timing ECO does not change logic functions of the original design.
- ❖ The ECO can be made by FE, ME or BE departments as necessary.

73 What does PrimeTime, Formality, Calibre, Hspice, HLDRC, PP check, Conformal –LEC , Conformal-LP do? (Lien-san, Dam-san- PV tool list)

- ❖ **PrimeTime:** is a tool to check timing.
 - PT is a full-chip, gate level static timing analysis tool targeted for complex, multimillion-gate. Checking all possible paths of design for timing violations without logic simulation or test vectors. PT finds any timing violations, the design needs to be re-synthesized using new timing constraints (generated by PT) to fix the conditions that are causing the timing errors. Input: netlist, library, AC feature.
 - Using on STA.
- ❖ **Formality:** Check logic function between RTL-RTL, RTL-gate, gate-gate.
 - Checks the functions of two designs or two cell libraries are equivalent or not. Performs RTL-to-RTL, RTL-to-gate, and gate-to-gate design verifications. Performs Verilog-to- DB, Verilog-to-Verilog, DB-to- DB, Verilog-to- SPICE, and DB-to-SPICE cell library verifications.
 - Using on checking before and after synthesis design.
- ❖ **Calibre:** is a tool to check DRC and LVS.
 - DRC (design rule check): check the design rule after layout, example: check the distance between 2 gate and source, spacing errors...
 - LVS (layout versus schematic): check the equivalent between layout and schematic.
 - Using for Physical verification.
- ❖ **Hspice:** Simulation Program with Integrated Circuit Emphasis. It's used to simulate many input and constraint to check output.
 - Using for Circuit design verification (schematic and layout). Check function design.

❖ **PP Check** is a Renesas In-house tool used to check power supply-on/off rules. It checks:

- Power supply connection: Check whether the netlist has appropriate PG supply and connection, as well as whether those PG connections agree with the specified PG specification.
- Power supply isolation: Check the isolation of connection between different power domains, power on-off sequence, and X-propagation prevention (indeterminate signal propagation).

For example:

- Check if a cell is placed in correct power domain, and whether it is connected to correct Power/Ground.
- Check whether uIO, LS, or isolation cells are used and connected to correct PG (these cells are used as signal interface (conversion) between different power domains).

❖ **What is uIO cell?**

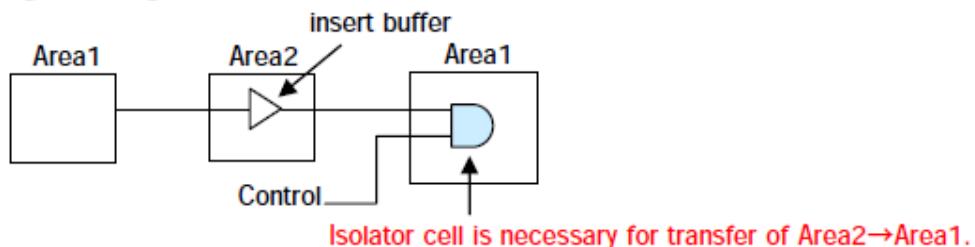
"uIO (micro IO)"s are circuits used for signal interface. uIO includes level-shifters and core isolation cells.

- Level shifter cells are used for interfaces between different power domains having different power supply.
- Core isolation cells are used for prevention of unknown value propagation (or X-propagation) between different domains with same power supply voltage.

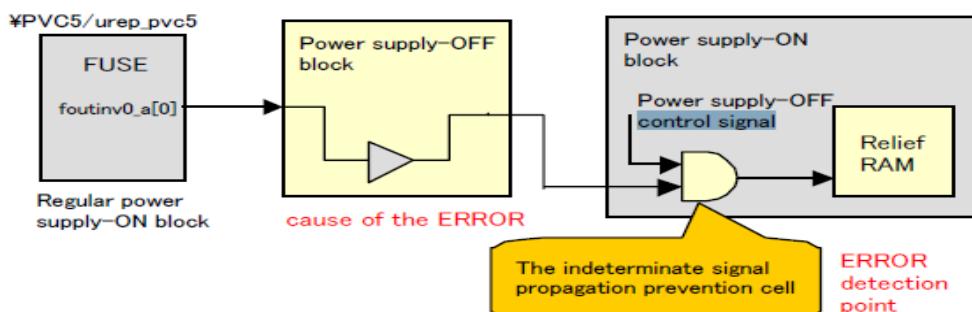
❖ **What is control signal?** Control signals are referred to as:

- Signals to control the operation of uIO cells (eg. CDSN, CDRN, SE18, RE)

<Wrong buffering1>



- Signals to control power supply-on/off:



87 What is DEF, LEF? (TrungMai-san)

59 Don't touch, don't use, size-only setting?

- ❖ **Don't touch:** to prevent modification (up size, down size, doping (change Vth)) or replacement of these objects during PnR optimization.

The common command when we want to don't touch an object is set_dont_touch (ICC/ICC2)

set_dont_touch: Set the dont_touch attribute on cells, nets, designs, and library cells to prevent the tool from replacing or modifying them during optimization.

EXAMPLES

The following commands prevent modification or removal of the block1 and analog1 cells during optimization, but allow the N1 net to be modified.

```
icc2_shell> set_dont_touch [get_cells {block1 analog1}]
```

```
icc2_shell> set_dont_touch [get_nets N1] false
```

- ❖ **Don't use:** to exclude them from the target library during optimization, we usually make dont_use for cells which have issue relating to routing (too many pin), leakage (too big size) and fabrication (easy to violated DRC) ...

The common command when we want to exclude is set_lib_cell_purpose (ICC2), set_dont_use (ICC/ICC2).

set_lib_cell_purpose: Specifies valid purposes for library cells.

EXAMPLES

The following command specifies that the lib_cells called 'G1' and 'G2' from the 'tech_lib' library should not be inserted during any optimization:

```
icc2_shell> set_lib_cell_purpose -include none {tech_lib/G1 tech_lib/G2}
```

```
icc2_shell> set_dont_use [ get_lib_cells {tech_lib/G1 tech_lib/G2}]
```

When we set don't touch and don't use: before placement and optimization

- ❖ **Size-only** to allow the tool to upsize, down size, change VTH only during optimization, don't allow to be removed. Normally, we set this setting for instances to allow them optimize but not be removed by the tool.

The common command when we want to size-only an object is set_size_only (ICC/ICC2)

EXAMPLES

The following command applies a size-only setting of true to the cell named U1.

```
icc2_shell> set_size_only U1 true
```

96 Spare cell - filler cell - power switch cell (Thac mac)

- ❖ **Spare cells** are extra gates (AND, NOR, XOR, INVERTER ...) which are included in the netlist or inserted into a layout. They are NOT functionally connected and are RESERVED for later design changes. For example after **tape out**, if some errors are found in logic function, spare cells can be used by modifying routing metals. So we can reduce cost because only metal layers (BEOL) are modified while FEOL layers are unchanged.

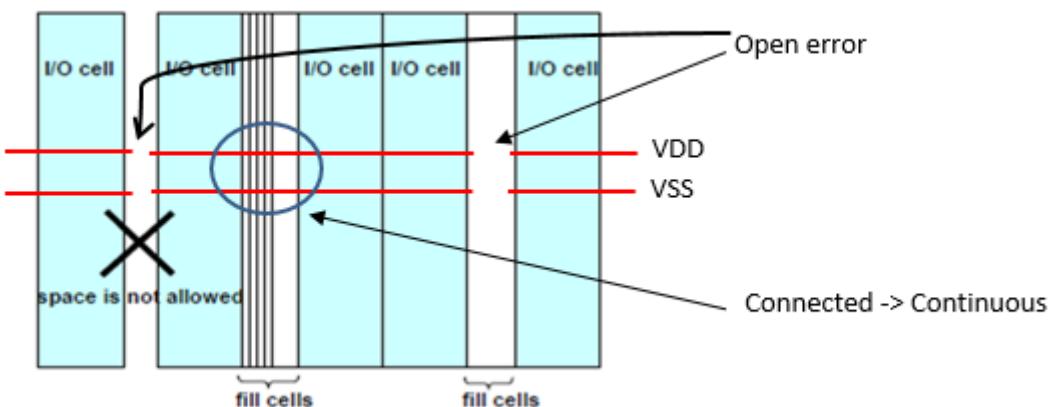
In RVC, we include spare cells after **post_route** phase.

Input of spare cells are connected to Vss and/or Vdd (tied).

Cells are identified as spare cells if:

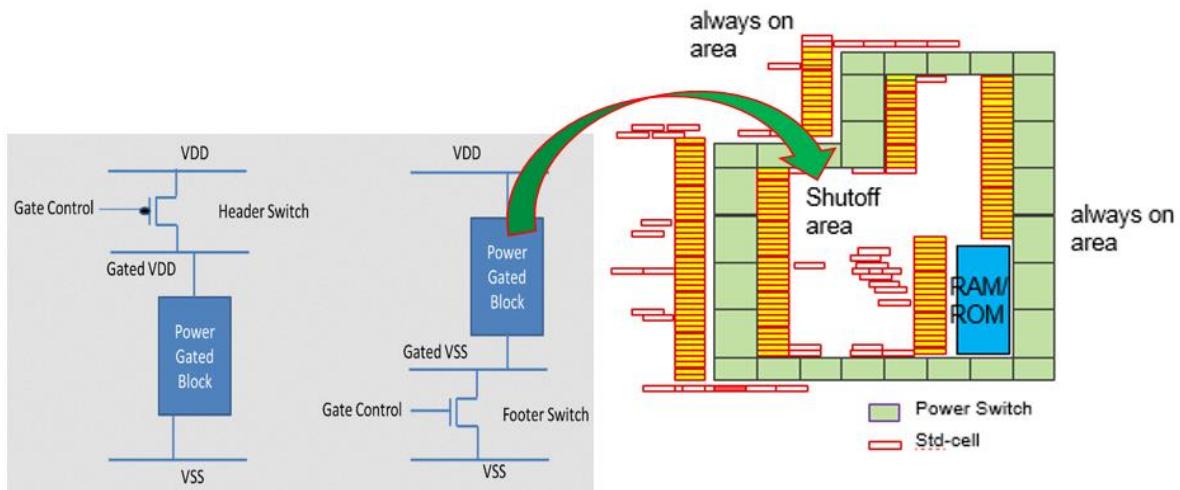
- Cell is not a physical-only cell (Only PG pins or No-Pin).
- All inputs are tied.
- All output are floating.

- ❖ **Filler cell** is 1 kind of cell in library (IO, PSW, STD ...) which is used to insert to fill up the empty area among cells. Filler cell has no logic information (Non-logic) Ex: filler cell to insert between IO cells -> this filler cell is called **IO filler**.



Role of filler cell

- Connect cells in a row to ensure that power supply lines are continuous.
- Ensure the continuity of other physical layers (eg. N-well, implant layers)
- Avoid **planarity(???)** problems when manufacturing (eg. Low density ...)
- ❖ **Power Switch cells** are used to shut-off the power supply to a power domain. There are 2 types of switching cells:
 - PMOS PSW (Header Switch) to shut off Power line.
 - NMOS PSW (Footer Switch) to shut off Ground line.



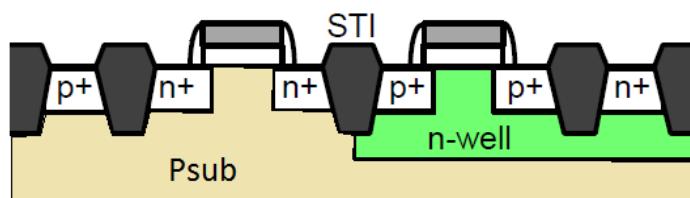
101. Explain N-well, DN-well

In CMOS (Complementary Metal Oxide Semiconductor) technology, we need both NMOS and PMOS fabricated on the same silicon substrate.

PMOS is built on N-type substrate and NMOS is built on P-type substrate.

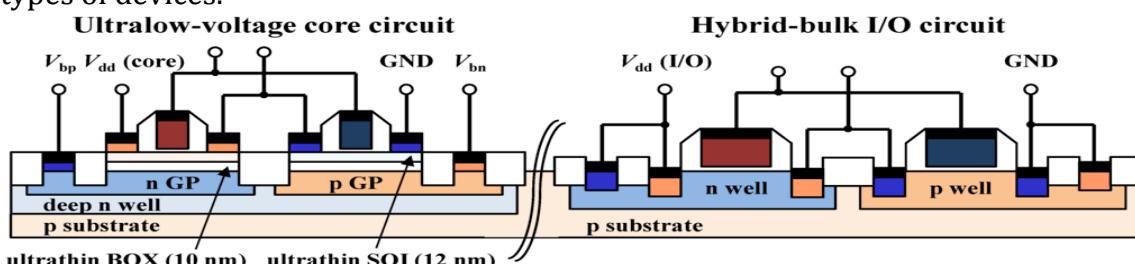
Our wafers are mostly p-type substrate, so we can build NMOS on it.

However to build PMOS on these wafers, we need to dope a selected region to be N-type, forming an N-WELL on a p-substrate wafer.

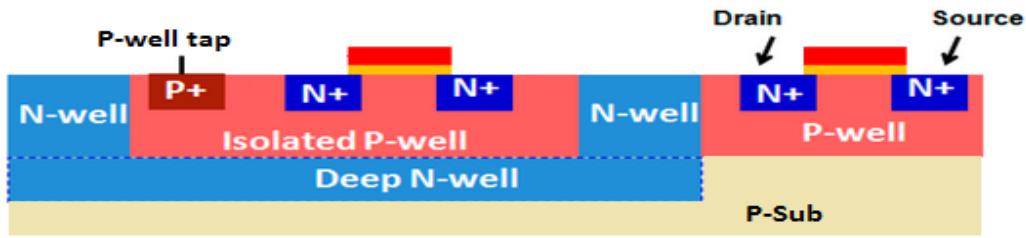


A Deep N Well (Deep N-Well or DNW) is the N doping into a P-type substrate that gets implanted deeper than a “standard” N well. (www.triadsemi.com).

DNW is used to separate substrates of NMOS (PSUB or P-Well) from each other, e.g. for different power domains. It is also often used in Analog layout or to separate different types of devices.



Because DNW is formed in a deeper position than NWELL, it is necessary to have a ring of NWELL around the boundary of DNW to make sure the isolation of the enclosed P-substrate (or PWELL).

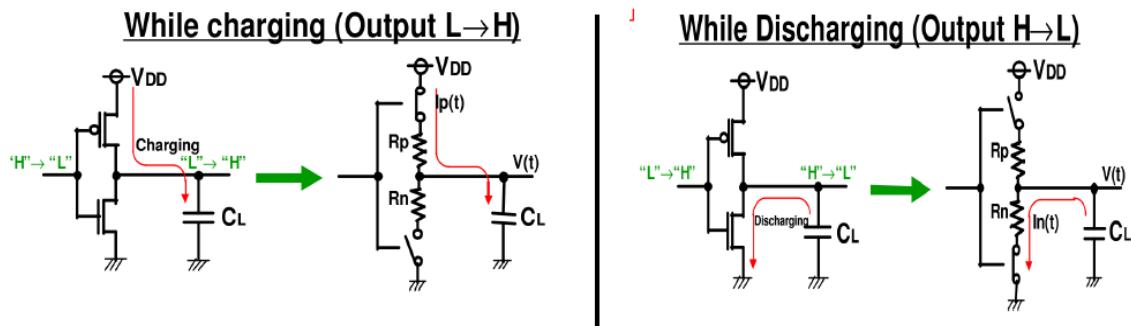


80 Power Consumption (Tri-san)

Have 3 types of power consumption:

❖ Dynamic Power

Dynamic power is the power which is generated by charging/discharging of load capacitance,



This is the energy dissipated due to the voltage or logic transitions in the design objects, such as cells, pins, and nets. The dynamic power consumption is directly proportional to the number and frequency of transitions in the design,

❖ Static Power (Leakage Power)

This is the energy dissipated even when there are no transitions in the circuit. This is also known as leakage power and depends on the device characteristics. The main contributor to leakage power is the sub-threshold-voltage leakage in the device. At lower technology nodes, leakage power consumption contributes significantly to the total power consumption of the circuit.

❖ Short-Circuit Power (Included in Dynamic Power)

Is power which is generated while switching the gate.

For example: when V_{in} changes, there is a period of time when pMOS and nMOS are at On state --> a current flows from V_{DD} to GND.

⇒ Power consumption of CMOS LSI :

$$P = P_{\text{dynamic}} + P_{\text{static}} + P_{\text{sc}}$$

Power Consumption is not good if P high or not meet the requirement,

➔ Dynamic Power Optimization

- Low Power placement (LPP):

- Clump registers closer to reduce clock-tree capacitance,
- Shorten high-activity nets,

- Gate-level power optimization (GLPO).

- Optimize gates to reduce switching activity,

→ Leakage Power Optimization

- Size cell
- Doping Vth.
 - Need to balance performance & low leakage.
 - Automated flow desired for optimal design closure.

Their effect: We have power consumption of CMOS: $P = P_{dynamic} + P_{leak} + P_{sc}$.

The P_{sc} , we must accept it and we must control $P_{dynamic}$ and P_{leak} .

$P_{dynamic}$: reduce CL or f to reduce $P_{dynamic}$.

P_{leak} : increase threshold voltage to reduce leakage current.

1 Explain input/output and meanings in Floorplan step? (Ref: ICC2 student guide)

Floorplan: It's the first step on P&R flow.

A floorplan of an integrated circuit is a schematic representation of tentative placement of its major functional blocks. Floorplans are created during the floorplan design stage and floorplan can be adjusted many times after checking timing and congestion -but before the final netlist comes. Key steps in floorplan design stage are defining block shape/size, row/track creation, macro cell placement, IO ports placement, placement blockage creation, voltage area shape/location, power planning.

Input:

- ❖ Netlist: is a verilog file. This file is synthesized from RTL code. It is maybe preDFT netlist, postDFT netlist, postJTAG netlist or postISO netlist.
- ❖ Library: difference layout tool uses difference type of library. In common, the library contain many information such as the logic function, timing characteristics, power characteristics and the physical information such as cell dimensions, border, pin locations, and mask layers, as well as technology information such as antenna rules, and electro-migration data. (Refer to the next question for more detailed)
- ❖ Block shape: provided by product team (CHIP TOP) or CHIP TOP team (IP).
- ❖ Design Block diagram: logic block connection diagram provided by FE team if available.
- ❖ Physical constraint: analog macro, BUMP, IO placement (TOP), pin arrangement requirement (IP).
- ❖ Layout instruction: analog constraint, placement constraint, routing constraint.
- ❖ Timing Constraints (SDC, optional): The main purpose of constraint is check initial timing.
- ❖ Technology information: Technology file, TLUPlus file, mapping file, Signoff ...
- ❖ Power planning: Power specifications, PG, Power domain...

Output:

- ❖ Database ready for placement with shape/size, site/row/track, placed ports, placed Hard/soft macro, placement/routing blockage, indicator for cell placement (bound) if any layout instruction.

DEF/LEF file as input for another tool if require.

74 Describe the required input library for Layout implementation?

- ❖ Required input library for layout are physical and logical library.
- ❖ The layout tool performs physical implementation, including placement, clock tree synthesis, routing, and optimization of a chip design. To perform these tasks, it needs to read in a design netlist and both physical and logic libraries. These libraries contain information about the cells used in the design netlist.
- ❖ A physical library contains information about the geometry of the cells that are placed in the design and connected with power, ground, clock, and signal routes. This library information includes the cell dimensions, border, pin locations, and mask layers, as well as technology information such as wire tracks, antenna rules, and electro-migration data.
- ❖ A logic library contains functional information about these cells, including the logic function, timing characteristics, and power characteristics. The Layout tool needs this information to perform analysis and optimization of the design based on the physical characteristics and the timing, power, noise, and signal integrity requirements.

2 Explain input/output and meanings in Placement step?

Placement: is step which locating various circuit components (standard cell, macro, blockage, bound, tap cell, boundary cell,...) within the chip's core and optimizing a number of objectives to ensure that a circuit meets its performance demands. Typical placement objectives include:

- ❖ Timing.
- ❖ Congestion. (Total wire lengths.)
- ❖ Power (Leakage & Dynamic - based on each project to be low or high priority).
- ❖ Area (based on each project to be low or high priority).
- ❖ Cell , pin density

Input:

- ❖ Database from Floorplan step.
- ❖ SDC.
- ❖ Some of requirements must be followed such as: keep cells, don't use cells, layout instruction, meta stable ...

Output: Placement database, reports (timing, Vth type use, power), log file, checker result (Formality, HLDRC, PP check...), checklist.

Output quality checks: Timing, congestion, leakage, area, module distribution, cell density.

3 Explain input/output and meanings in CTS & postCTS step?

CTS = Clock Tree Synthesis.

In this step, we will build a clock tree to supply clock signal for each flip-flop or hard/soft macro. The clock tree is very important so that we need to build and route clock signal follow requirement.

CTS priority:

- ❖ Timing correlation (both setup & hold)between preCTS & just after CTS
- ❖ DRV (clock tran/ clock cap/ clock fanout)
- ❖ Latency/Skew-in case: the longest latency on Chip -> higher priority than DRV (Will effect to chip top timing due to OCV)
- ❖ Dynamic power (Calculated based on the toggle rate -> control insertion area)

- ❖ Route ability (Adjust NDR – Non Default routing Rule)

postCTS = Design optimization after CTS step (timing, power, area,...)

Input:

- ❖ Database with all components are placed in the core area and legalized.
- ❖ Cell type/clock DRV target/routing rule spec.
- ❖ Latency/skew target.

Output:

- ❖ A design has finished postCTS step.
- ❖ Timing result
- ❖ Checker result (Formality, HLDRC – VTH clock line type)
- ❖ Checklist

4 Explain input/output and meanings in Routing step and route_opt step?

Routing: Routing creates physical connections for logic nets on design. There are three steps in routing phase:

- ❖ Global routing: **routing track estimation.** (Thuong-san) – Not finished yet
- ❖ Track assignment: assign routing track for each routing net.
- ❖ Detail routing: metal wiring for all nets.

Note: Having some another routing steps for other flexible purposes:

- ❖ Optimize route ability
- ❖ ECO routing: continue metal wiring to fix remained DRC from "detail routing" step.
- ❖ Routing verification: do routing verification to confirm DRC after routing.

Route_opt: Do design optimization (timing, power, area,...) and incremental routing (ECO route).

Input:

- ❖ Output from postCTS phase (or placement phase just for preCTS routing ability check)
- ❖ Routing rule such as min/max layer constraint, NDR rule if any, layout instruction routing rule, antenna rule, ...

Output:

- ❖ A design has finished postRoute step.
- ❖ Timing result.
- ❖ Routing verification result (DRC, LVS, open, short, antenna ...)
- ❖ Checker result (Formality, HLDRC, PP check)
- ❖ Checklist

5 What is DFM & Chip finishing step?

DFM-Design For Manufacturing: Technologies to solve various fabrication related problems (Antenna Effect, Short & Open error, Metal Erosion, Metal Liftoff, Metal Over-etching ...) in design phases ahead. Its purpose is to improve Yield.

Refer to: [Documents/1.General Documents/010 ENG/030 Training/Renesas Technical Training/02 Intermediate Course/IB001](#)

Required actions for Chip finishing:

- ❖ Insertion tap/boundary (Based on process)
- ❖ Finding and fixing antenna violation.
- ❖ Insert redundant via (if redundant via has requirement).
- ❖ Insert filler/cap cell.
- ❖ Insert metal fill.

Input: A design which released after postRoute step (clean timing/routing).

Output: A design meet requirement of manufacture, satisfy rule check of Physical Verification.

6 Explain Yield? How to increase yield? (In chip finishing stage)

Concept: Yield is the factor to evaluate how good of manufacturing process.

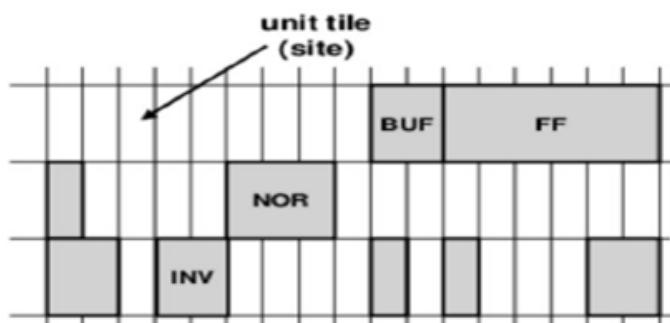
Formula: Yield is the ratio of number of good chips to number of total chips on wafer.

$$\text{Yield} = \frac{\text{Num}_{\text{good chips}}}{\text{Num}_{\text{total chips}}}$$

To increase yield, we must make sure the quality of timing, constraints and physical verification rules (include DFM).

7 What is track, row, grid, unitTile, pitch, process?

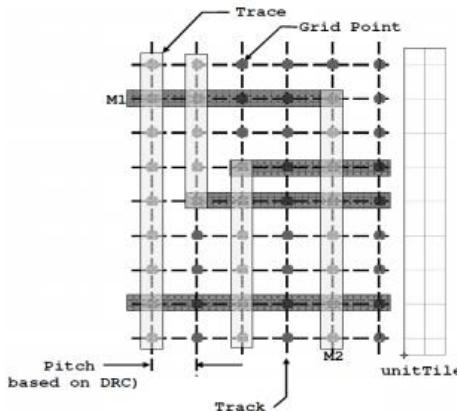
- ❖ **Track:** Standard vertical and horizontal line which to be used for indicating metal wiring (routing). (Refer to the next Question for more detailed)
- ❖ **Row:** Standard Horizontal line to be used for indicating standard cell placement.
- ❖ **Manufacturing Grid:** Is the smallest resolution of your technology process. The value can be found in technology file. Beside that we also have another kind of grids like placement grid (Site, unitTile), routing grid.
- ❖ **UnitTile (Site):** is a minimum height and width of the cell to be placed .These measures are available in the technology library. Unit tile is also known as site.



The standard cell will place follow site and row. It mean the width of cell will multiple with site and the height of cell will multiple with row.

- ❖ **Pitch:** the center-to-center distance between features of an integrated circuits such as interconnect lines.
- ❖ **Process:** The **technology node** (also **process node**, **process technology** or simply **node**) refers to a specific semiconductor manufacturing process and its design rules. Different nodes often imply different circuit generations and architectures. Generally, the smaller the technology node means the smaller the feature size, producing smaller transistors which are both faster and more power-efficient.

Refer to: https://en.wikichip.org/wiki/technology_node



62 Prefer routing direction and non-prefer routing direction

- ❖ **Prefer routing direction:** Metal layers have routing direction which are defined in the technology file of each technology.
- ❖ **Non-prefer routing direction:** Metal layers have routing direction which is defined based on user, When fixing short or DRC, we sometime use non-preferred routing direction.

For example: when fixing DRC, we draw M2 in both vertical and horizontal direction.

Note: By default, the IC Compiler II tool derives the preferred routing directions from the unit tile of **the first standard cell library in the reference library. (???)**

If the preferred routing direction is not set, the tool sets defaults.

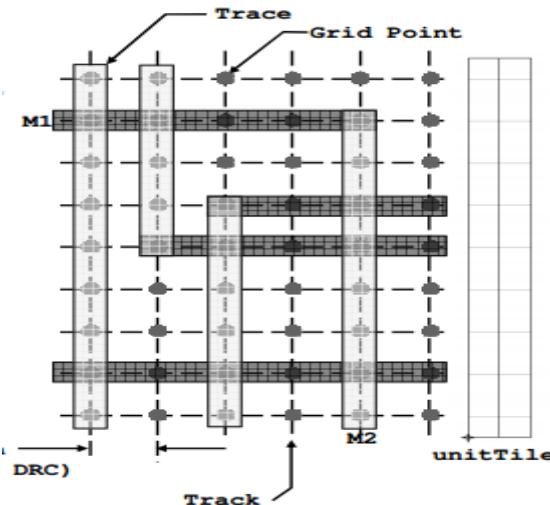
- If an adjacent layer has a preferred routing direction, Zroute sets the other direction as the preferred direction for the layer
- Otherwise, the preferred direction for the metal 1 layer is set to horizontal, the preferred direction for the metal 2 layer is set to vertical, and so on.

Example: to set the preferred routing direction to vertical for the M5 and M7 layers, use the following command:

```
icc2_shell> set_attribute -objects [get_layers {M5 M7}] \
    -name routing_direction -value vertical
```

63 What is routing track?

- ❖ Routing track: Standard vertical and horizontal line which to be used for indicating metal wiring (routing). The tracks and preferred routing directions are defined in technology file.
- ❖ Metal traces (routes) are built along, and centered upon routing tracks based on a grid.



- ❖ Metal routes must meet minimum width and spacing “design rules” to prevent open and short circuits during fabrication.
- ❖ In gridded routers these design rules determine the minimum center-to-center distance for each metal layer.
- ❖ Grid or track spacing:
 - Congestion occurs if there are more wires to be routed than available tracks.
 - *Why we use track?* → optimize routing resource, satisfy design rules.

8 Distinguish Link lib and Target lib (in ICC - ICC2 use)?

- ❖ **Link lib:** contain **all** timing information for standard cells, macro cells used in the design. Need to make sure all libraries (which are necessary for the design) are included in this link library setting.
- ❖ **Target lib:** Is the library that the tool uses to pick cells for optimization and remapping. It is typically set to only the standard cell library. We can control the cell type for the design by modify this library.

9 Design view (Layout view), Abstract view and FRAM view (ICC/ICC2 use)?

[Abstract view:

The elements are retained in abstract depend on the abstract creation way.

1. Timing abstract retains
 - . data and clock paths needed for analysis and optimization at top-level.
 - . cells are used in keeplist, timing constraint or exception such as set_size_only, set_false_path, clock pin, clock gating, set_input_delay, set_output_delay, set_case_analysis, set_disable_timing, set_clock_balance_points., set_operating_conditions, set_voltage, set_clock_sense,.....
2. Abstract support multi voltage by storing the UPF relevant to the retained interface logic. This is promoted to the top-level during linking.
3. Abstract may content power information. This enables you to perform power analysis of the complete design while at top-level

4. Abstract may be created for signal EM analysis. This enables you to signal EM analysis inside abstracts while at top-level (Source ???)

❖ Definition:

Design view is view mode when all elements of design are displayed and we can see/edit using layout tool. Data size of Layout view will be the heaviest view.

Abstract view is a modeled view of the gate-level netlist of a block. Abstract views contains ONLY the REQUIRED logics which are interfaced outside of the block. Most other internal logics are removed, except for the internal logic which related to constraints (timing constraints...). In Innovus tool, the abstract view is called **ILM**.

Placement information, timing constraints, clock tree exceptions, LPE information, trans., PG information and UPF constrains. Sometimes NDR also retained in the abstract view. (Not finished yet_Vy-san)

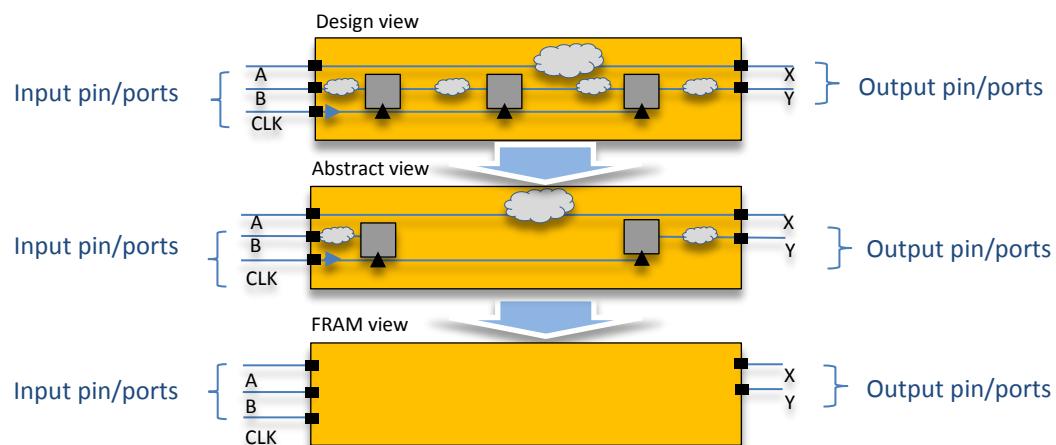
FRAM view is another format of LEF which contains mainly boundary, pins, routing blockages.

❖ Purpose:

Design view: use when the designer want to have the full timing view of sub-block. Normally use in timing debug phase.

Abstract view: mainly use in timing optimization phase for 2 main purposes: reduce running time (compare with Design view) and timing correlation between TOP-SUB (compare with FRAM view + timing Liberty)

FRAM view: used for mainly 2 steps: design planning, routing.



10 Hard-macro/Soft-macro and Standard cell

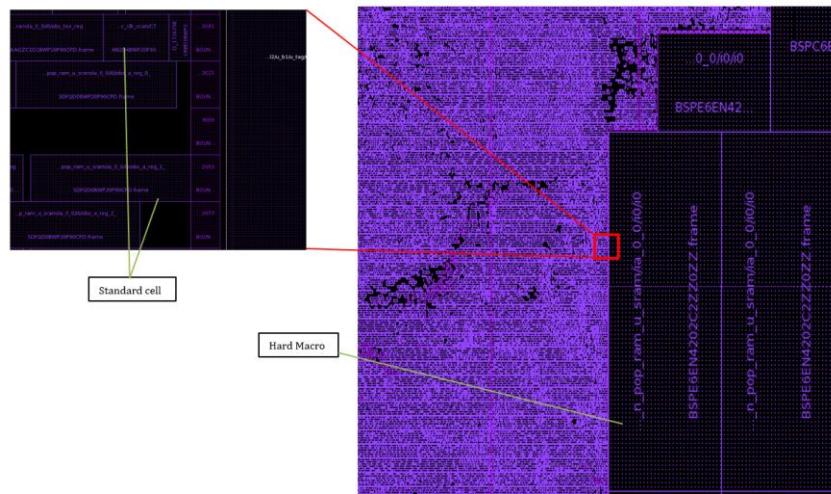
- ❖ **Hard-macro** is categorized as an Intellectual Property (IP) which is designed for specified manufacturing technology. They are block level designs (hierarchical design) which are designed for specific functional and optimized for

power/area/timing. They need to be clean both timing and physical verification beforehand.

While accomplishing physical design it is possible to only access pins of HM. We have freedom to move, rotate (some cases we could not rotate 90 degree), flip but can't touch anything inside HM.

The followings IP can be considered as HM:

- Memory: SRAM, ROM, Register File and Flash.
- Analog IP (ADC, DAC, Power Regulator, USB...). Sometimes we could consider IO/PAD as a HM.
- Pre-designed blocks.



- ❖ **Soft-macro** is the block or IP which can be touched at upper level when design like timing optimization ...
- ❖ **Standard-cell** is
 - A logic units of similar geometry. STD implement basic logic (NAND, NOR, INV, FlipFlop, Latch, and complex gates ...)
 - STD usually comes in libraries for a specific technology and depended on manufacture. (TSMC, Renesas, Samsung...)
 - [Usage flow] Hardware is described in RTL. Description is translated in Boolean logic and sequential elements. After that, the logic is mapped on standard cells. Standard cells are placed in rows. Finally, Cells are properly connected by routing channels and/or additional metal layers.
 - In Custom layout design (IO design; Analog design); we might use custom STD for other purpose (re-design STD for specific usage)

11 Sign-off constraint

Sign-off constraint is a set of constraints for all verification steps that the design must pass before it can be tape-out.

There are two types of sign - off's: front-end sign-off and back-end sign-off. After back-end sign-off the chip goes to fabrication.

Sign-off constraint includes all constraint of design for:

- ❖ Netlist constraints (Logic equivalent, Checker, ...)
- ❖ Timing constraints. (Sign-off document, SDC, SPEF, ...)
- ❖ Physical constraints. (Physical rules, ...)
- ❖ Power constraints. (Power spec, ...)

14 DFT Circuit

DFT (Design For Testing) consists of **IC design** techniques that add testability features to a hardware product design. The added features make it easier to develop and apply manufacturing tests to the designed hardware. The purpose of manufacturing tests is to validate that the product hardware contains no manufacturing defects that could adversely affect the product's correct functioning.

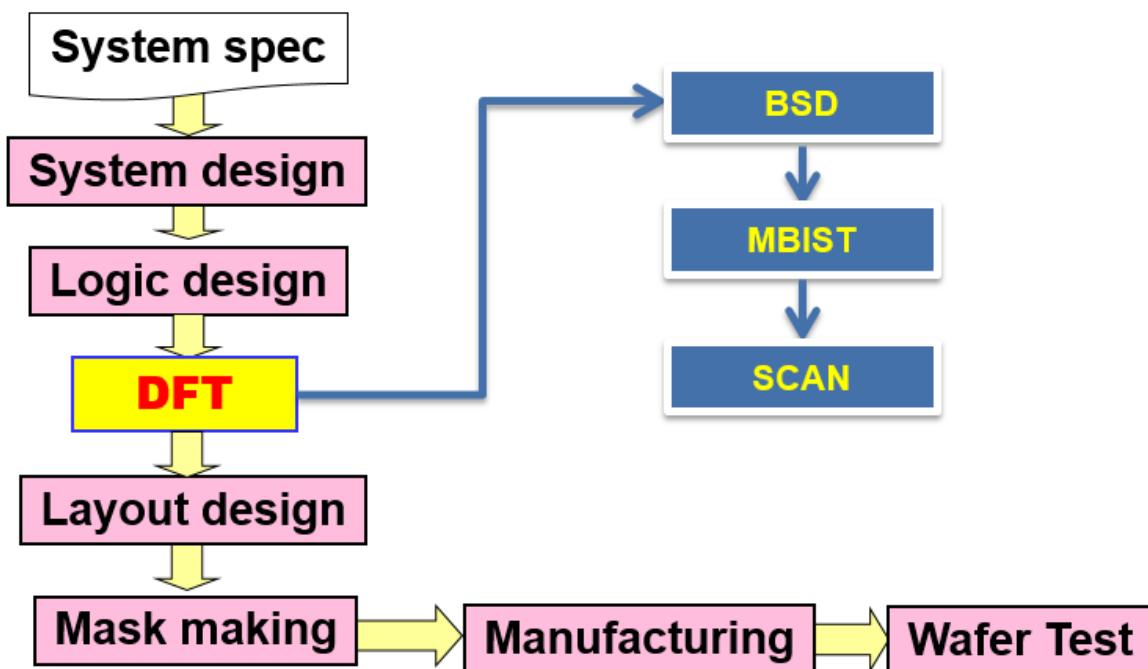
DFT is included:

Creating test circuits.

Inserting test circuits into LSI.

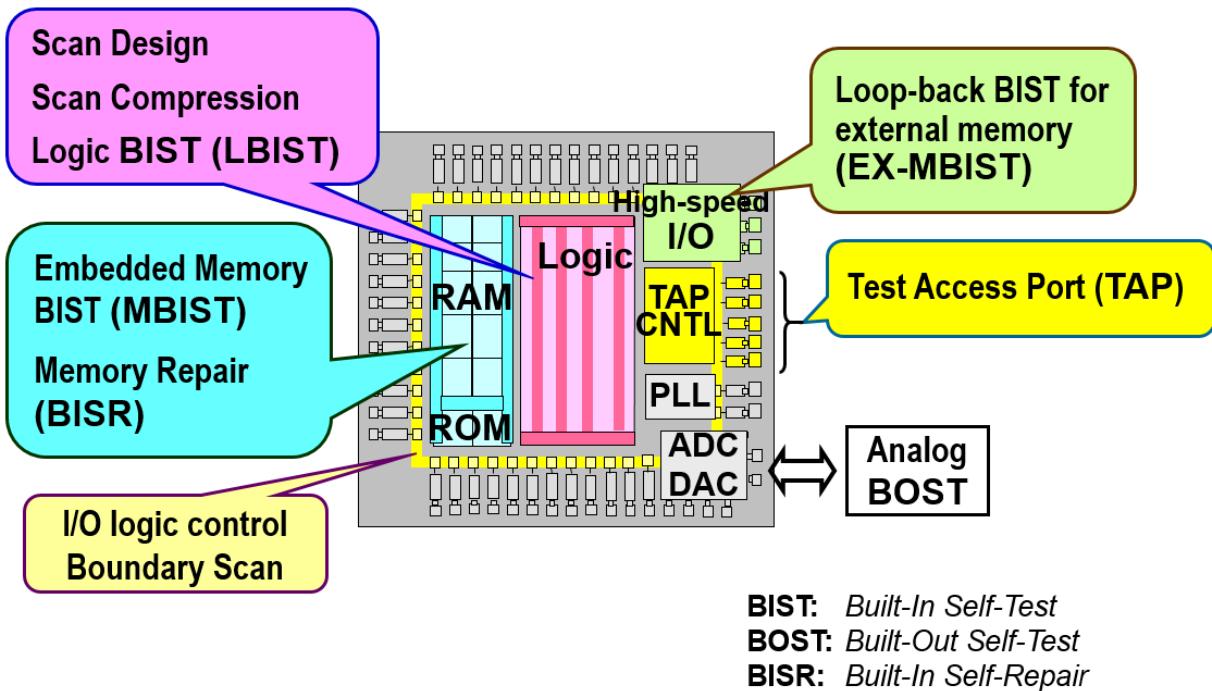
Preparing test data for physical failure analysis.

Position of DFT in LSI flow:



Picture 1: Position of DFT in LSI flow

Basic circuits in DFT include: BSD circuit, MBIST circuit, SCAN circuit.



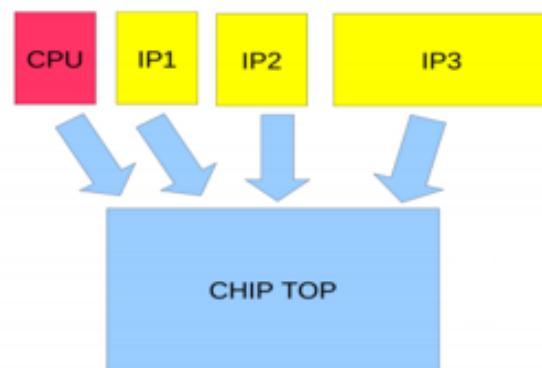
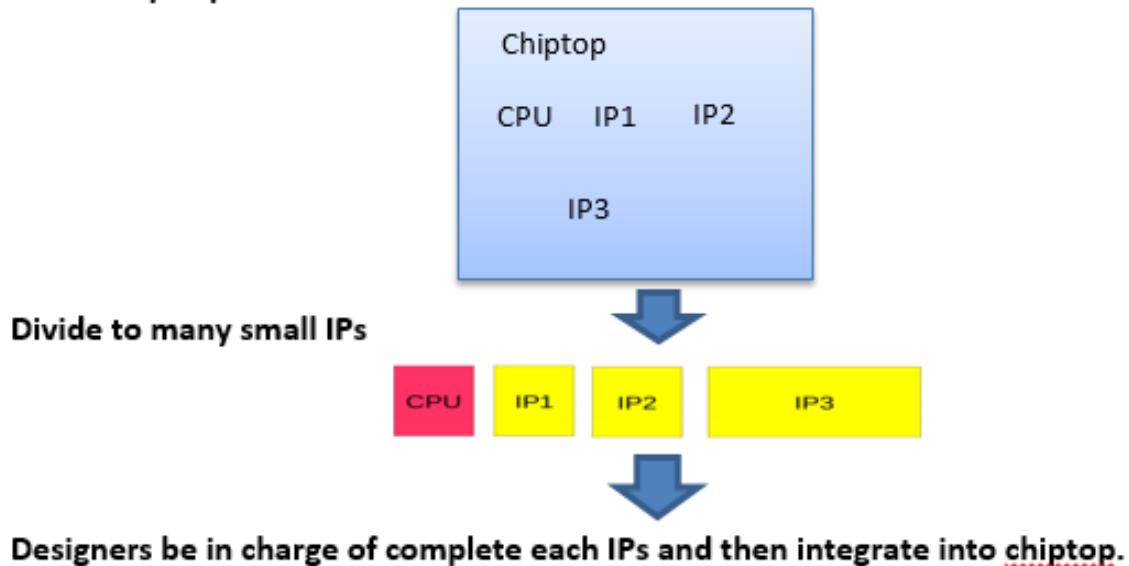
Picture 2: Application Example of DFT

Refer to:

- [Documents/1. General Documents/010 ENG/030 Training/Renesas Technical Training/01 Basic Course/BG002](#)
- <\\rvc-vnas-01\Backend\COMMON\DOCUMENT\Terminology\Reference\0 DFT Introduction-1.pdf>

15 What is Hierarchical Design? Advantage & Disadvantage?

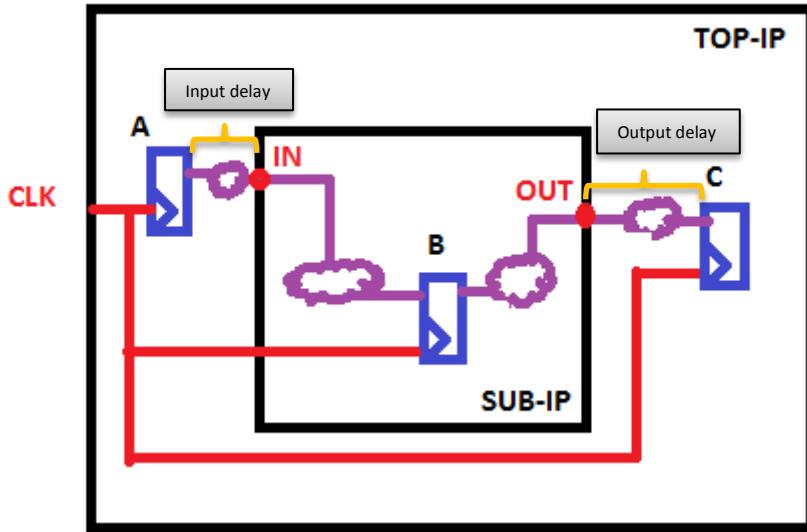
Hierarchical Design is a method to divide design into multiple blocks, designer can work on the blocks separately and in parallel from RTL to physical implementation. Targeted to shorten TAT (Turn-Around Time) and better QoR (Quality of Result).

Initial Chip-top

Advantage	Disadvantage
<ul style="list-style-type: none"> - Reuse IPs: Base on function of chip, we can re-use hier-IP of previous projects to save time and cost for re-design. - Reduce TAT, have more loop for debug and improve design quality. - Improve productivity when designing complex chip. - In case of any timing issue, you can fix individual block. 	<ul style="list-style-type: none"> - Need more HR (human resource) and machine (server). - Difficult to control all sub-blocks.

16 What is timing interface constraint (I/O Budget)?

I/O Budget is a concept related to interface timing delay when the design has timing relationship to outside (IP-IP, IP-TOP, TOP-PACKAGE).



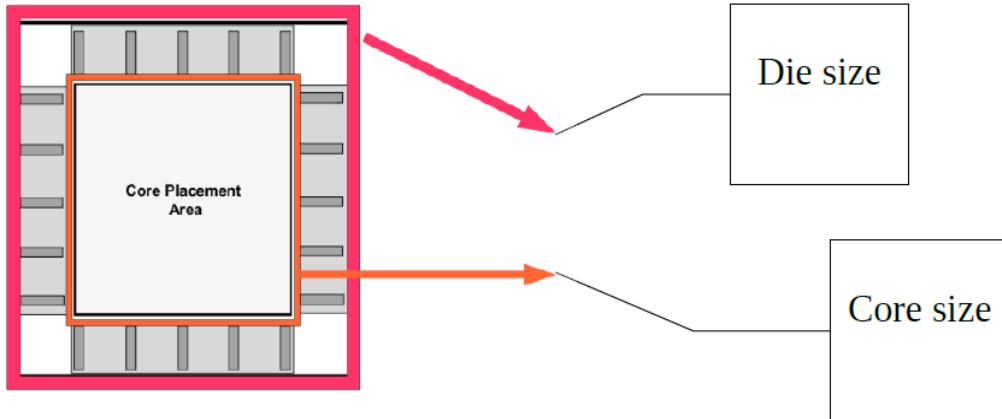
For example in picture above:

We have timing path from FF-A to FF-B (Top-IP to Sub-IP) and timing path from FF-B to FF-C (Sub-IP to TOP-IP). We need to divide timing budget for TOP-IP and SUB-IP to run to optimize separately each IP. The format setting:

```
# CLK: 8ns
set_input_delay -clock CLK 4 [get_ports {IN}]
set_output_delay -clock CLK 6 [get_ports {OUT}]
```

21 What is Die, Core? What is the difference between die size and core size?

- ❖ A die is a small block of semiconducting material, on which a given functional circuit is fabricated.
- Silicon die is a fully functional chip that has not yet been packaged. A wafer consists of many dies that are separated during the dicing process.
- ❖ Core is the place where Cells (Standard cells/Hard Macro) can be placed



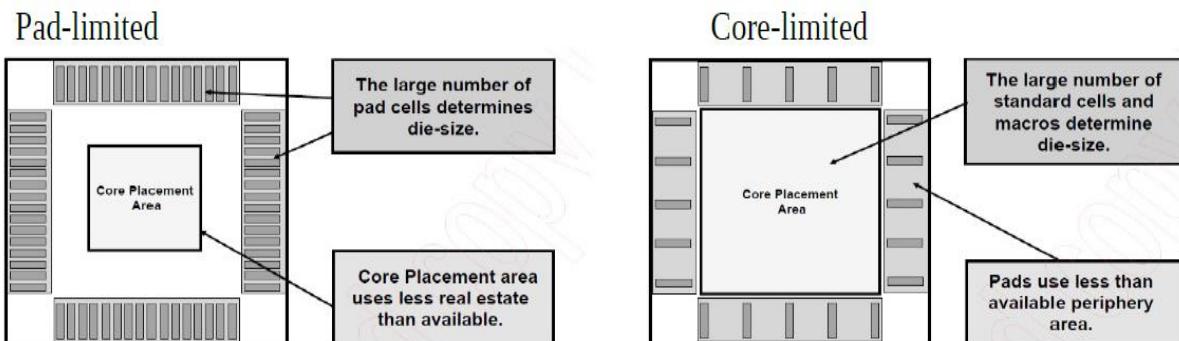
- ❖ Note: Core size \leq Die size.

DIE Area = Core Area + IO Pad Area.

[Add question] What is Pad, Bump? (Dam-san)

85 Pad limited, core limited and package limited (Dam-san)

- ❖ **Pad I/O:** Where chip contact with outside world (peripheral devices, bumps ...). Pad/IO can be placed not only around the corners or edges of chip but also core area where it can be connected to inside macro and Bump/Pin package (for Flip Chip design style) conveniently.
- ❖ **Pad limited and core limited:** Refer to the below pictures



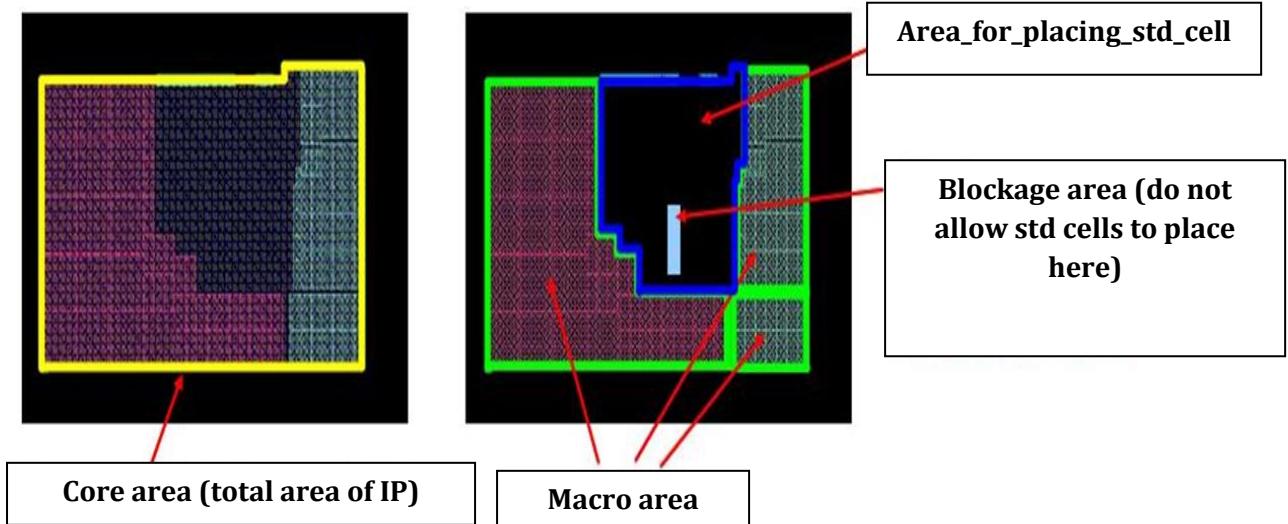
Depend on design specifications -> Find the way how to balance between pad I/O area and core placement area to smallest die size.

17 What is utilization? Which factors will affect utilization? Relationship between Die size and utilization.

- ❖ **Utilization:** is an index which shows relation between the utilized area and the available area (Capacity of core). Utilization is calculated by percentage (%).
- ❖ **Formula:** (use in Renesas)

$$\%U = \text{Demand} / \text{Capacity} = \text{std_cell_area} / \text{area_for_placing_std_cell}$$

While: $\text{std_cell_area} = \text{total area of std cells from the design}$
 $\text{area_for_placing_std_cell} = (\text{core area} - [\text{macro area} + \text{blockage area}])$



- ❖ **Factors affect utilization -> increase/decrease standard-cell area**
 - Performance
 - Power
 - Route ability
- ❖ **Relationship between Die size and utilization:** if Utilization is low, the Die size is larger.

18 Why we don't make design with utilization 100% at first time?

- ❖ Utilization is increased step by step in PnR phase because of timing/ leakage/ congestion optimization.
- Example:**
 - Initial Floorplan: Utilization = 75%.
 - PreCTS: Utilization = 77% (increase 2% by timing/leakage/congestion optimization).
 - CTS: Utilization = 78% (increase 1% by inserting clock buffers).
 - PostCTS: Utilization = 79% (increase 1% by timing/leakage/congestion optimization).
 - Route & PostRoute: Utilization= 80% (increase 1% by timing/leakage/congestion optimization).

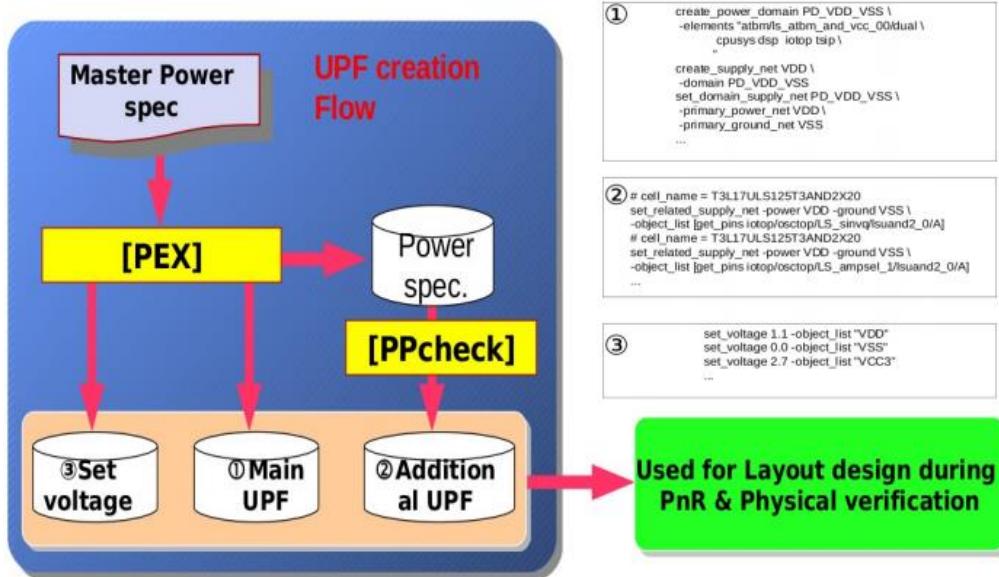
If utilization is 100% at first time, Design will not have available space for standard cells (size cell or insertion) during PnR Phase → Design is **FAILED**.
- ❖ High utilization makes difficulty in routing. There are more cells in a specific area. There are more connection (or nets) in that area.
→ Routing resource becomes more critical.
- ❖ In STA phase, design may still have timing violation and need to fix by sizing cell/inserting buffers.
- Example:**

STA phase: utilization increases more 1% by timing fixation

If utilization is 100% at the beginning of STA phase, Design will not have available space for standard cells (size cell or insertion) during STA Phase → Design is also **FAILED**.

82 What is UPF format?

- ❖ **UPF** (Unified Power Format) is “**The IEEE 1801 standard**” for Design and Verification of Low Power Integrated Circuits,
- ❖ It brings out the common rules for multi-voltage in design.

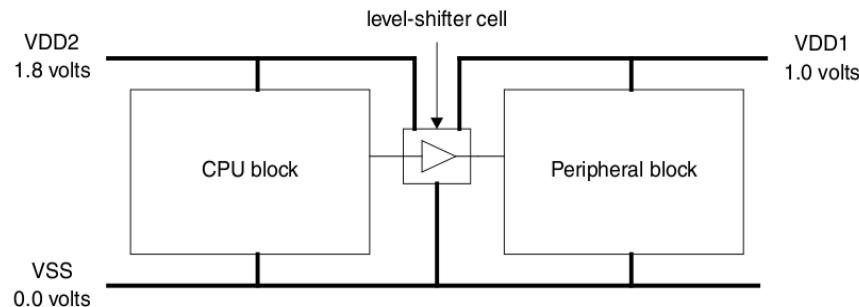


72 Explain Power domain, Power Island, voltage area?

- ❖ **Power domain**
 - Power domain is a defined group of elements in the logic hierarchy that share a common set of power supply needs.
 - All logic elements in a power domain use the same primary supply and primary ground.
 - Power domain is defined in UPF.
- ❖ **Power Island**
 - A geographically bounded collection of instances, all from the same power domain. Since they are geographical, power islands are used in the **physical level description** to "realize" power domains. A power domain may be implemented as multiple, possibly disjoint power islands, thus both, one-to-one or many-to-one relationships between power islands and power domains are possible.
- ❖ **Voltage area**
 - A voltage area is a physical placement area for the cells associated with a power domain.
 - Voltage area is defined in Floorplan stage.
 - Command :
 - To define a voltage area: `create_voltage_area` with option `-power_domains`.
 - To modify an existing voltage area: `set_voltage_area`.

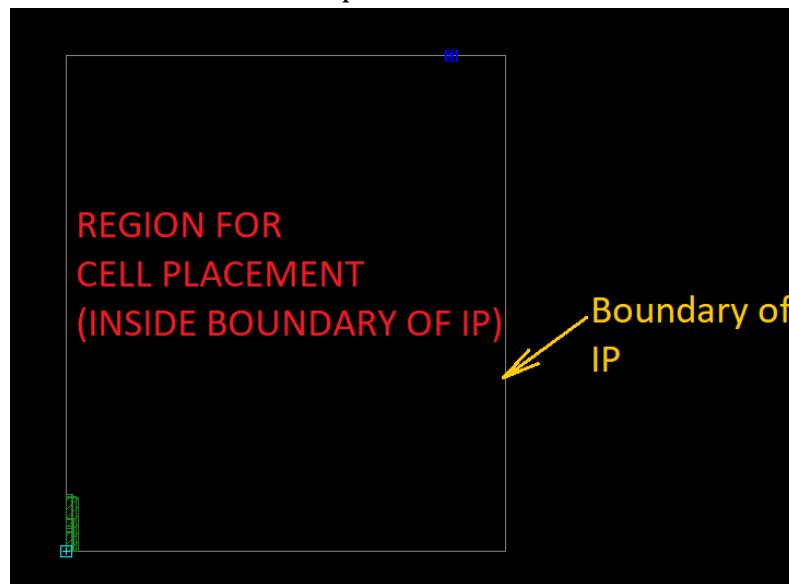
99. What is multi power design?

A multi power design is a design that has **more than one power domains with different power supply voltage**. In design below, 2 blocks (CPU block and Peripheral block) are in different power domains:



20 Bound, Module Boundary, ... ? Hoang-san

- ❖ **Region:** is area which allows to put cells on them.

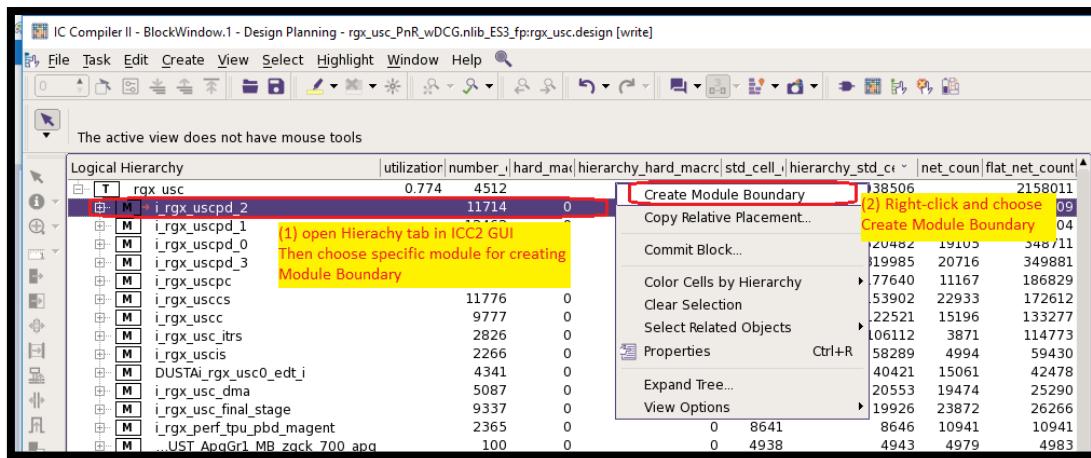


- ❖ **Module Boundary:** is used in the GUI of ICC/ICC2 to help analyze the logical hierarchy in the floorplan step.

In floorplan step, Layout Designer can create some Module Boundaries (each of which presents for each module).

Designer can modify size, location of Module Boundaries and know how many connections between Module Boundaries. Then decide where a specific Module should be placed in the PnR design.

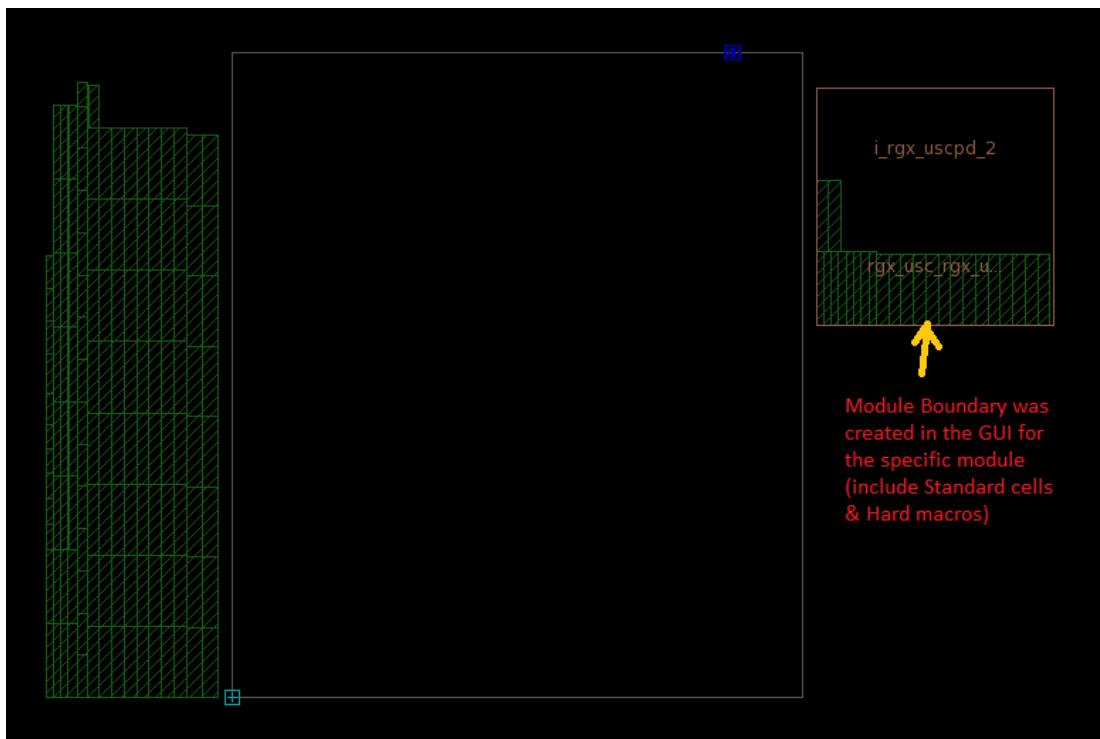
Below pictures are examples for creating Module Boundary and Analyzing Floorplan:



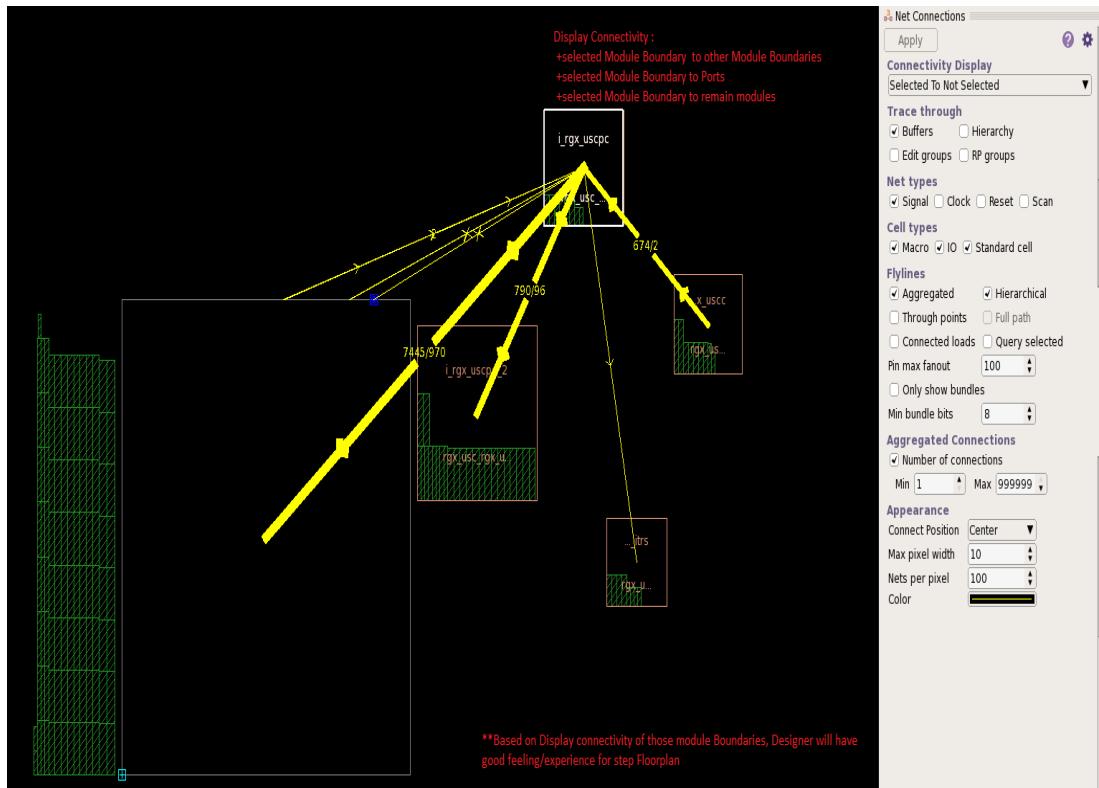
Picture 1. How to create Module Boundary in the GUI of ICC2

	utilization	number	hard_ma	hierarchy_hard_ma	std_cell	hierarchy_std_c	net_coun	flat_net_count
T rgx usc	0.774	4512	0	156	24	577	2038506	2158011
M i_rgx_uscpd_2	0.774	11714	0	24	577	156	322122	18812
M i_rgx_uscpd_1	12462	0	24	496	321456	19465	321456	349904
M i_rgx_uscpd_0	11978	0	24	675	320482	19105	19465	348711
M i_rgx_usccs	13633	0	24	573	319985	20716	349881	
M i_rgx_uscc	6260	0	8	359	177640	11167	186829	
M i_rgx_usccs	11776	0	32	2674	153902	22933	172612	

Picture 2. Display of Module Boundary in the GUI of ICC2 (Hierarchy View)

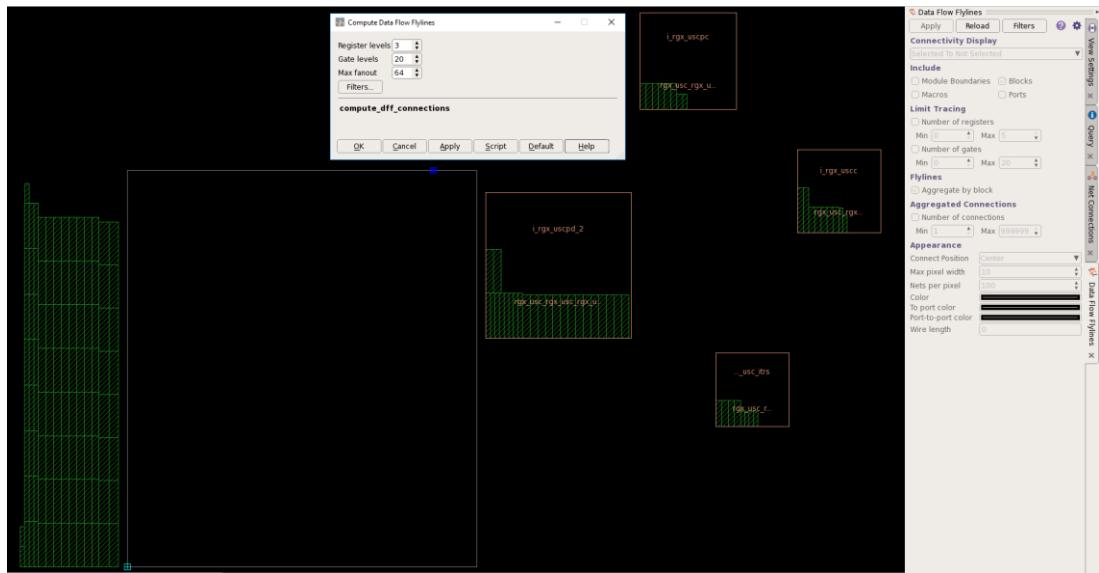


Picture 3. Display of Module Boundary in the GUI of ICC2 (Layout View)

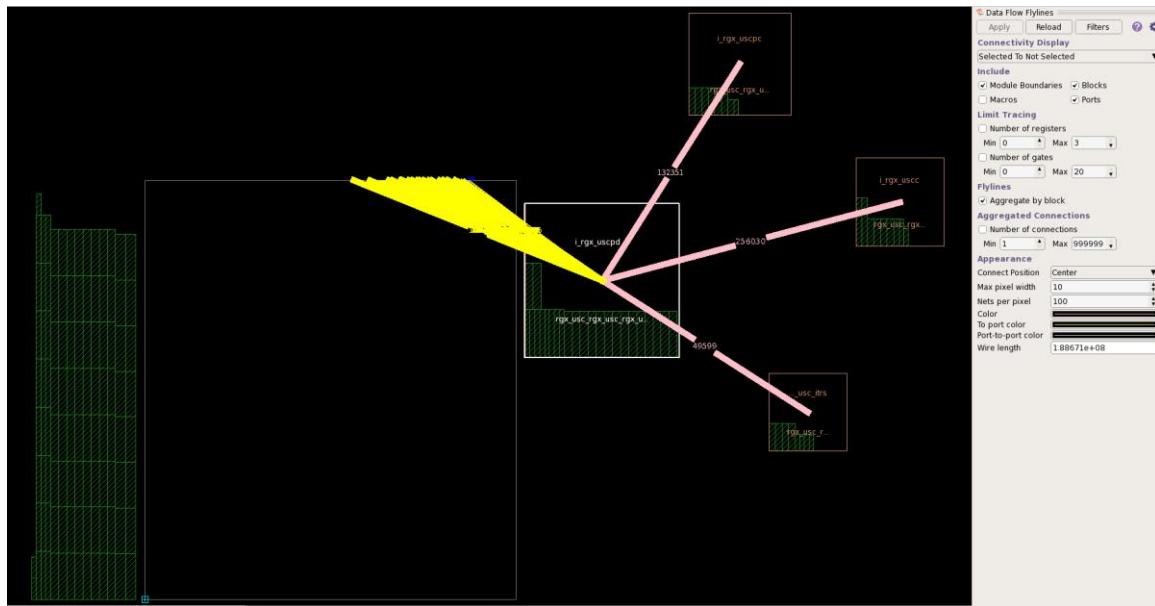


Picture 4. View Connectivity of each Module based on Module boundaries

(**Based on Display connectivity of those module boundaries, Designer will have good feeling/experience for step Floorplan)



Picture 5.1 Compute Data Flow Flylines (It takes long time to calculate connection)



Picture 5.2 View Data Flow Flylines from Module Boundaries

- ❖ **Bound:** (used in ICC2) is region-based placement constraints for coarse placement.

Bound is defined with command ***create_bounds*** (in ICC2) by Layout Designer.

- There are 3 types of bounds: move bounds, group bounds, and diamond bounds.

Move bounds restrict the placement of cells to a specific region of the core area. To create a move bound, use the **-coordinate** option.

Group bounds are floating region constraints. Cells in the same group bound are placed within a specified bound but the absolute coordinates are not fixed. Instead, they are optimized by the placer.

To create a group bound, use the **-dimension** option.

Diamond bounds are region constraints centered on a specific object, which can be fixed or floating. Other objects in the same diamond bound are placed within the specified Manhattan distance from the central object but their absolute coordinates are not fixed. Instead, they are optimized by the placer.

To create a diamond bound, use the **-diamond** option together with the **-dimension** option.

- **Soft, hard, and exclusive** bounds are defined as follows:

Soft bounds specify placement goals with no guarantee that the cells will be placed inside the bounds. If the timing or congestion cost is too high, cells might be placed outside the region. This is the default.

Soft group bound example:

```
icc2_shell > create_bounds -dimension {100 100} -name Soft_bnd1 INST1
```

Hard bounds force placement of the specified cells inside the bounds. To specify hard bounds, use the **-type hard** option along with the **-dimension** or **-coordinate** option.

However, overusing hard bounds can lead to inferior placement solutions.

In hard bounds, other cells also can be placed along with the specified cells in those bounds.

Hard move bound example:

```
icc2_shell > create_bounds -coordinate {0 0 10 10} -type hard \
-name Hard_bnd4 INST4
```

Exclusive bounds force placement of the specified cells inside the bounds.

All other cells must be placed outside the bounds. To specify exclusive bounds, use the **-exclusive** option.

Exclusive move bound example:

```
icc2_shell > create_bounds -name Ex_bnd5 -coordinate {200 200 250 250} -exclusive
```

Items	Move bounds	Group bounds	Diamond bounds
Option “-coordinate”	yes	None	None
Option “-dimension”	None	Yes	Yes
Option “-diamond”	None	None	Yes
Restrict placement of Cells to a specific region	Yes	None	None
Combine with option “-type hard” (hard bound)	Yes	None	Yes
Combine with option “-type soft” (soft bound)	Yes	Yes	Yes
Combine with option “-exclusive” (exclusive bound)	Yes	None	None

23 Placement blockage

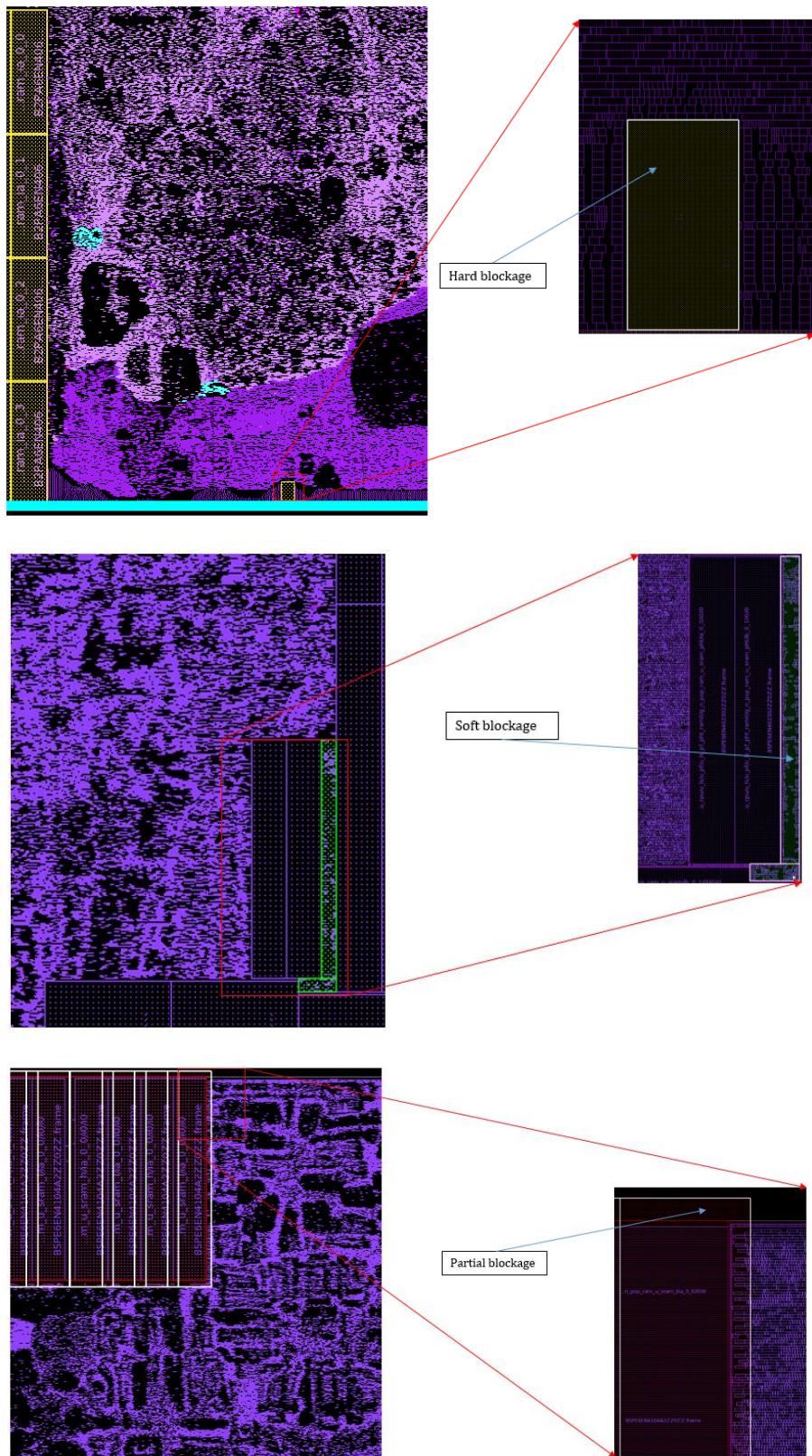
Placement blockage is an area that not permit or limited cells place in this area.

There are three main types of placement blockages:

[1]Soft blockage: Limit cells placement. Kind of cells which can be placed will be defined by command options.

[2]Hard blockage: Not permit any cells place in this area.

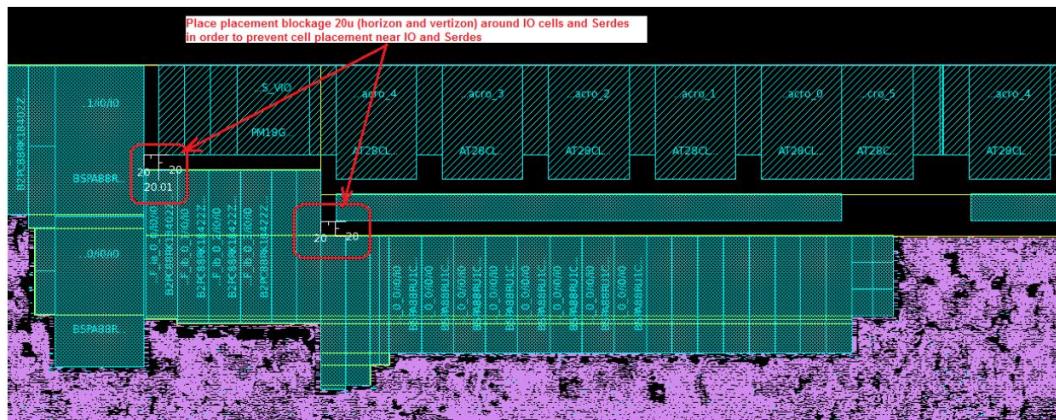
[3]Partial blockage: Use to limit quantity of cell placement in this area by percentage (%).



Purpose of placement blockage:

- ❖ Soft blockage and Partial blockage are used to limit cell insertion in a specific area. This limitation is to reserve more available placement room on that area.
- ❖ Hard blockage is used to restrict standard cell's placement in a sensitive Area (around Analog macro, IO cells, Memory Boundary ...). Below picture is an example of hard placement blockage to follow Layout instruction for IO cells and Analog cells.

*Follow layout instruction for analog macro

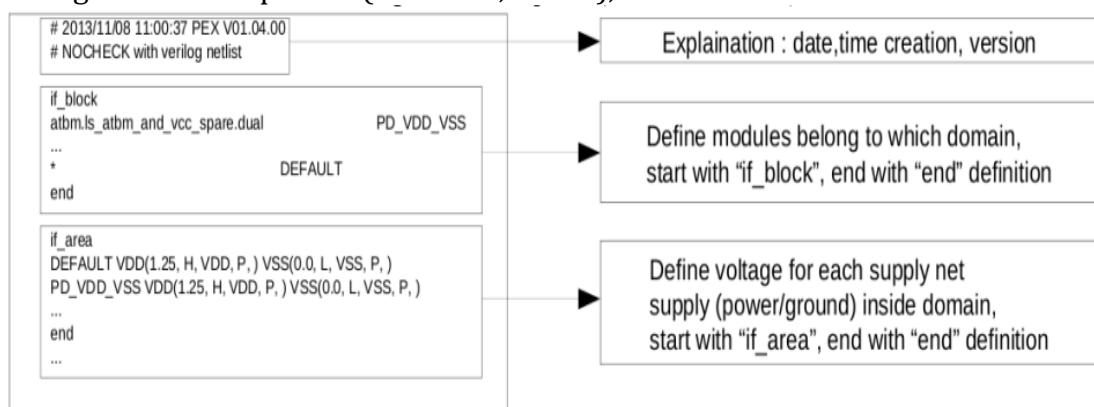


65 What is Route guide? What is Route Blockage?

- ❖ **Route guide:** Define a region for below purposes:
 - Prevent routing for signal or pre-routed nets, either completely or partially.
 - Change the wiring direction
 - Control wiring density
 - Support to fix routing violation
- ❖ **Routing blockage:** Defines a region where no routing allow on a specific layer.

83 What is power-spec, IO pin assign? Bump assign? Purpose of these file? (Dam-san)

- ❖ **Power-spec:** Also called power supply specification, is a list specifies “Power Domain” information of Design. It be created based on UPF Standard and used during verification phases (PP Check, LVS ...),



- ❖ **IO pin assign:** File that describes PAD number, PAD name, PAD position, IO cell name related to PAD etc.

This file is used as an input of EPOD (Esd PrOtection Device circuit placement check). EPOD uses the following parameters: PINNAME, IO BUF, DIRECTION, INSTANCE, WDPAD VOLTAGE and PADNAME,

VERSION:#####;	Design name
DESIGN:r7f5a009;	Define meaning of each column (format)
orderdef:WDPAD, PINNAME, IOBUF, PADNAME, INSTNAME; LB001, AVSS0_0, pf3v30tc, VSSPLL, AVSS0_0_PIO; ... LR001, P20, pf3c3bo0o2godvql, pad, iotop.pamnvcc.AUTO_IOBUF_P20; ... LT056, PC1, pf3c3bo0o2godvql, pad, iotop.pamnvcc.AUTO_IOBUF_PC1;	Order and name of each IO, LB : bottom LR : right LT : top

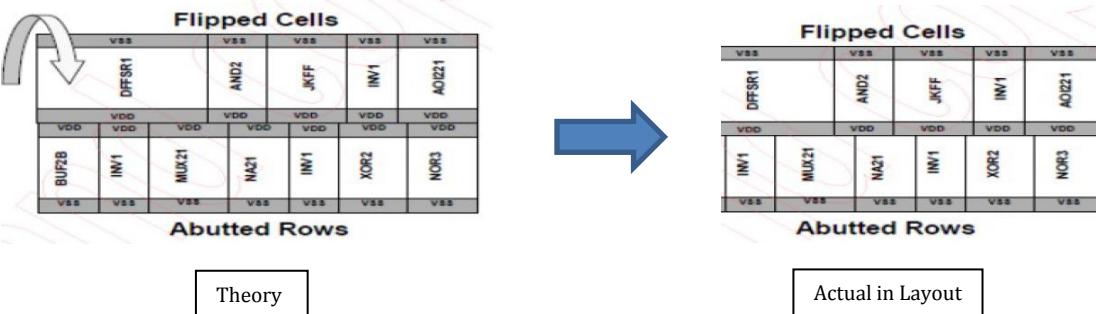
86 Abutted rows and non-abutted rows

- ❖ **Abutted rows:**

Placement rows are commonly abutted to reduce core area.

Adv: Reduce area.

Disadv: EM, IR Drop issue due to sharing power rail.



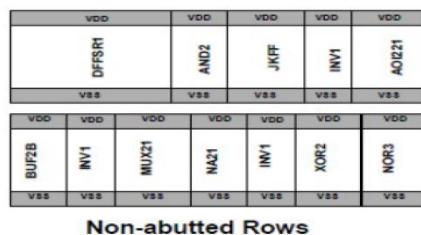
- ❖ **Non abutted rows:**

Placement rows are not abutted.

Adv: No EM, IR Drop issue.

Disadv: Large area.

Note: Nowadays, we use abutted rows (because we have methods to solve EM, IR Drop).



97. PG strap/rail/ring

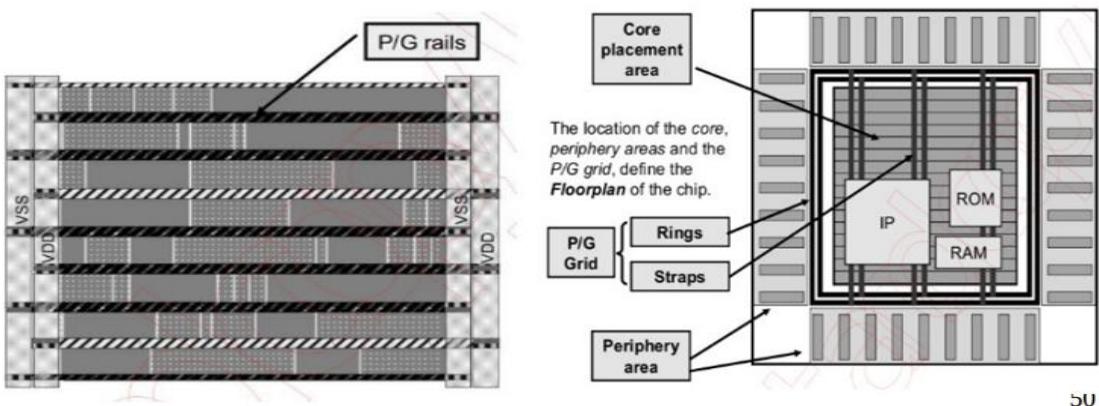
- ❖ **PG straps and PG rails** refer to Power-Ground mesh that supplies Power and Ground from PG Bumps (or PG Pads) to macro pins and/or core STD cells. Strap is the intermediate connection from Bumps or Pads to rails/Macro, when rail is the direct connection to STD cells.

Normally PG rails will be designed by M1 because STD cells are designed with VDD/VSS pins in M1.

Ex: Structure of PG mesh in T28 and T16:

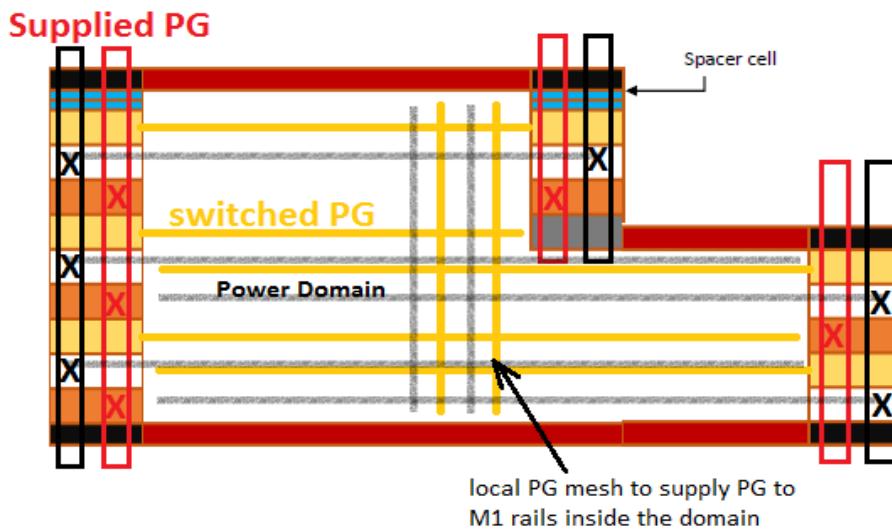
In T28 1p9M process: M1 rails, Via1-4, M5 straps, Via5-7, M8-VIA8-M9, RV-AP

In T16 1p11M process: M1 rails, Via1-4, M5 straps, Via5-9, M10-VIA10-M11, RV-AP



- ❖ **PG ring** refers to some different meanings:
 - PG ring can refer to a ring of PG strips (vertical and horizontal) surrounding a power domain or a layout area. It is formed to supply PG to the enclosed area. A local PG mesh will also be created, connecting from the PG ring and down to each std cell row.
 - PG ring can also refer to a ring of Power-Gating cells (PSW cells) to enclose a power domain.

(Picture) Thuong-san



In this case, PG strips (vertical and/or horizontal) will be drawn over and connected to PSW cells. A local P/G mesh will supply P/G from PSW to std-cells of the power domain.

32 What is Repeater / HFS?

❖ Repeater

Repeater is buffer / a pair of inverters that **receives a signal and retransmits** it to maintain the signal strength, also can reduce the total net RC since splitting the original net to shorter nets. So that, the signal can cover longer distance.

Beside timing improvement, we can use the repeater as a guidance for routing.

❖ HFS

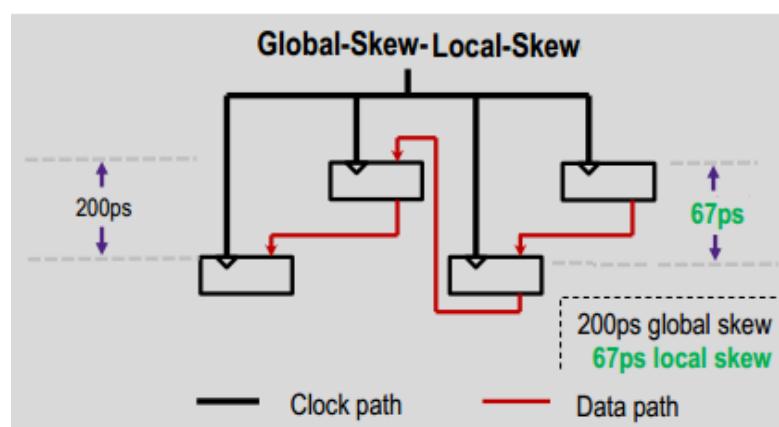
High fanout synthesis (HFS) is one step in preCTS stage, this step determine buffer trees for high fanout nets in design with purpose to guarantee the timing.

35 What are cell types for CTS and why we choose them?

- ❖ Cell types for CTS (Clock tree synthesis) mean cells which is used to synthesize the clock tree in CTS step, such as fixing DRC, inserting clock delay of design, balancing skew between clocks. Clock cell types have stable characteristic between rise edge and fall edge, in other words, small difference transition time between rise edge and fall edge.
- ❖ Please be notice about the clock cell types which will be used. It may impact to latency, skew, timing and leakage later on. For example, using weak cells for clock tree will cause large transition time. Normally, we will use clock buffer, but depends on project requirement, we can use clock inverter instead.
- ❖ In CTS environment, we need to provide CTS cell types for 4 kinds of usage: clock buffer, clock inverter, clock-gating and clock logic.
- ❖ CTS cells usually have some special key words like “*CL*”, “CK*” in the reference name (or you can get the attribute of library cell with “ref_name”).

28 Global skew vs Local skew

- ❖ **Global skew** is the difference from clock timing path of all FFs in the design within the same clock source. And when global skew optimization is run, CTS tries to match the clock delays for all FFs' clock pins.
- ❖ **Local skew** is the difference from clock timing path of only “related” FFs pairs. FFs are called “related” if one FF launches data when is captured by other.



❖ ISSUE:

The `report_clock_tree` command reports a local skew value which is larger than the global skew value.

What is the reason for this behavior?

`report_clock_tree -operating_condition max`

`report_clock_tree -operating_condition max -local_skew`

Based on the basic definition of local and global skew, local skew can never be greater than global skew.

However, if your design contains ICGs, the `report_clock_tree` command can report a local skew that is larger than the global skew.

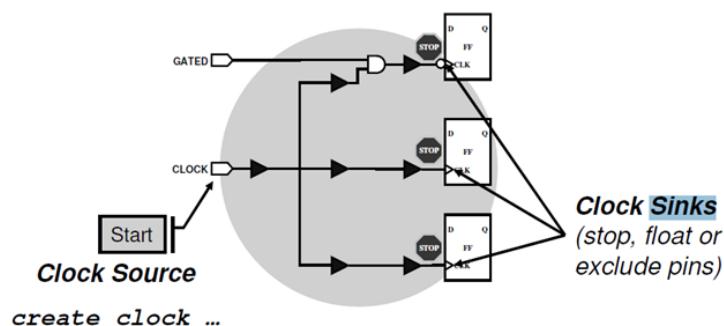
The reason for this is because the local skew report looks at timing paths and paths ending at ICGs are valid timing paths. However, ICGs are not valid endpoints for global skew reporting, since clock tracing will go through ICGs.

If you want local skew reporting to ignore ICG paths, use the switch the `local_skew_skip_icg` option of the `report_clock_tree` command.

`icc_shell > report_clock_tree -local_skew_skip_icg -local_skew`

29 Sink pin, Stop pin, Float pin vs Exclude pin

❖ Sink pin



Sink pins are the clock endpoints that are used for delay balancing. The tool assigns an insertion delay of zero to all sink pins and uses this delay during delay balancing. Sink pins are also referred to as balancing pins.

In CTS: the tool uses sink pins in calculations and optimizations for both *design rule constraints* and *clock tree timing* (skew and insertion delay).

Sink pins include:

- Stop pins.
- Float pins.
- Exclude pins.

Command: `set_clock_balance_points`

`report_clock_balance_points`

❖ Stop pin

A **stop pin** is a synchronous pin **which its phase delay is 0**. The tool assigns delay balancing among all stop pins of clock tree to achieve minimum skew (default target is zero skew).

Typically, the clock port of flip flop and macro cell is defined as a stop pin with correct clock tree delay information.

When a clock pin is not defined by library or defined incorrectly, tool assigns it an implicit exclude pin during clock tree synthesis phase.

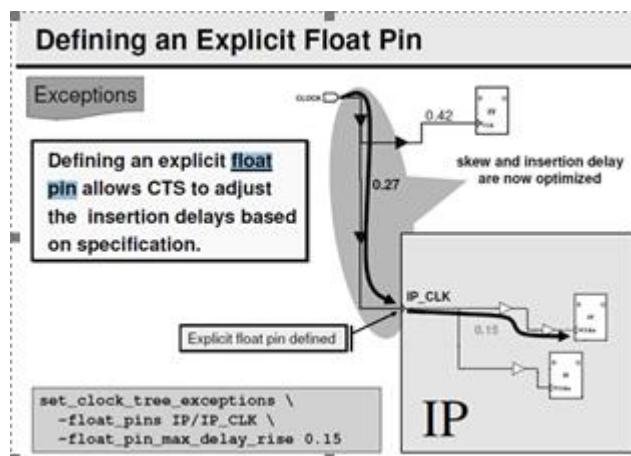
Incorrectly defined clock pins include:

- Clock pins without trigger edge info.
- Clock pins without a timing arc to the corresponding output pin.
- Clock pin without clock source (no clock)

❖ Float pin

A float pin is a synchronous pin which **its delay is user-specified value**. The tool assumes user-specified value delay at clock pin which is defined float setting and assigns delay balancing among all endpoints with assumed value.

Float pin is usually defined in case of clock pin of hard macro IP or RAM



In the figure, user-specified value is 0.15ns with assuming clock delay from IP clock pin to virtual flip flop inside IP is 0.15. Then, float value setting should be 0.15ns at IP clock pin.

❖ Exclude pin

A exclude pin is synchronous pin or asynchronous pin which **it does not need to apply delay balancing** with other endpoints. The reason may come from asynchronous timing requirement or designer intent purpose. There is 2 kind of exclude pin setting.

- **Implicit EXCLUDE (ignore) pins:** The tool assigns implicit exclude pin automatically based on design circuit or library pin attribute, for example:
 - Clock pin has floating load/ or disabled timing arc
 - Clock go to output but there is no stop/float setting of output.
 - Clock pin which has data attribute in library (library problem)

- Non-clock input pins of sequential cells (D, Set, Reset, etc).
- Pins with 3-state enable arc.
- Select/Control pin of mux used in the data path.
- Input pin of pre-existing gate, if all pins in its fan-out are exclude pins.
- Clock pins without trigger edge info.
- **Explicit exclude pin (ignore pin):** User defines clock pin as ignore pin. The tool ignores delay balancing as user setting.

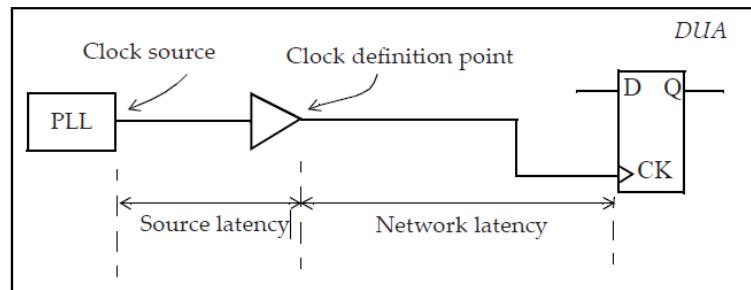
25 Explain Clock Uncertainty and latency

❖ Latency:

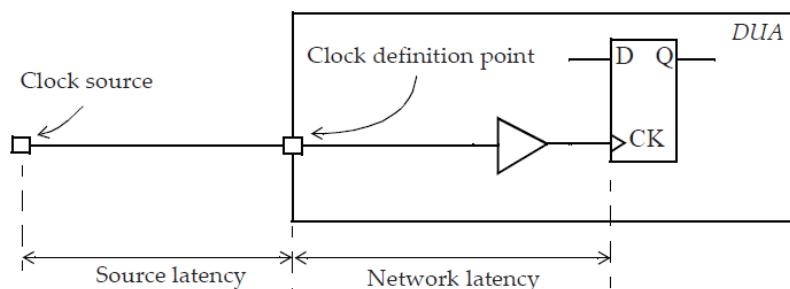
This is the amount of time a clock signal takes to propagate from the original clock source to the sequential elements in the design.

The latency consists of the following two components:

- **Source latency:** This is the delay from the clock source to the clock definition point in the design.
- **Network latency:** This is the delay from the clock definition point to the register clock pin.



(a) On-chip clock source.



(b) Off-chip clock source.

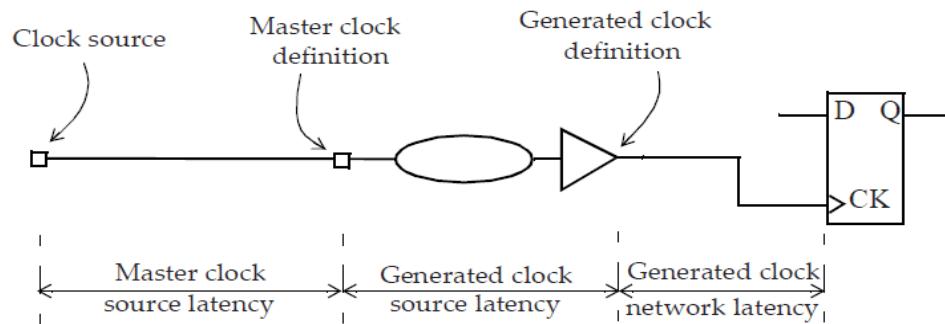
You can representing the network latency by using one of the following methods:

- Estimating and explicitly specifying the latency of each clock.

This is referred to as ideal network latency and the corresponding clocks are referred to as ideal clocks. Ideal clocks are usually used before you perform clock tree synthesis.

- Allowing the tool to compute the latency by propagating the delays along the clock network.

This is referred to as propagated network latency and the corresponding clocks are referred to as propagated clocks. Propagated clocks should be used only after you perform clock tree synthesis.



❖ Uncertainty (Setup/Hold)

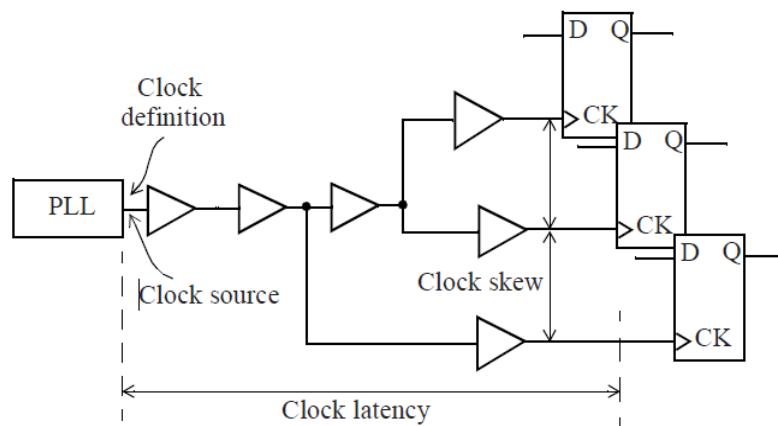
This is the maximum variation in the clock arrival time at the register clock pins of the same clock or different clocks.

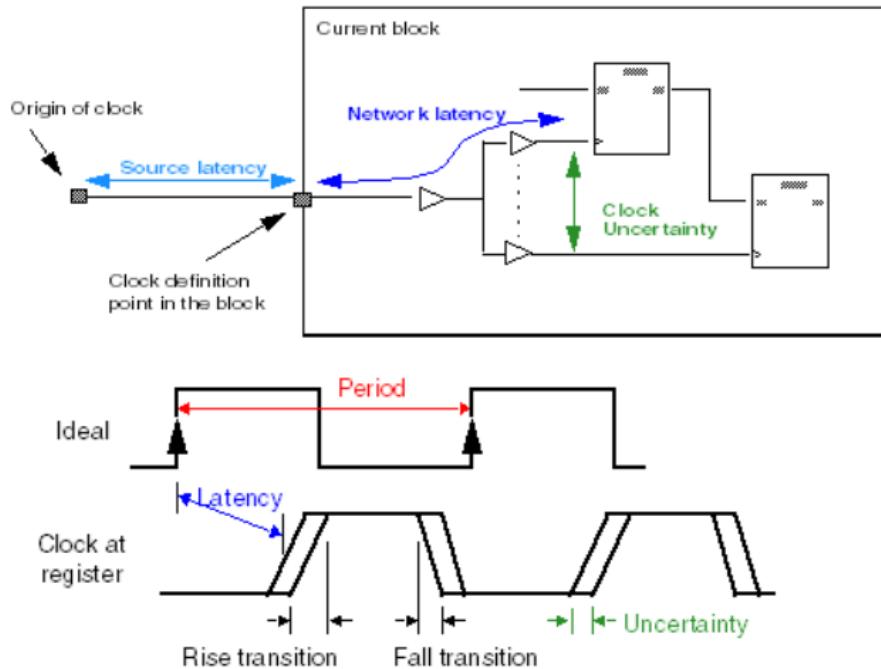
The clock uncertainty has the following two components:

- The **clock jitter**: is amount of different time between cycles of a clock. Jitter is one factor affect to Clock Uncertainty (CU).
Equation: $0.1\sqrt{T/2}$
- The **clock skew (Q26)**, which is the difference in clock arrival times resulting from different propagation delays from the clock source of the block to the different register clock pins.

This is applicable only before clock tree synthesis. After clock tree synthesis, the clock skew is accounted for by the propagated clock network latency.

Because clock uncertainty can have an additional margin built in to tighten setup or hold checks, you can specify different uncertainties for setup and hold on the same path.





❖ Specifying Clock Source Latency (ICC2 commands)

To set a source latency for both ideal and propagated clocks, use the [set_clock_latency -source](#) command.

The following commands specify the source latency of the clock CLK1 with an external clock network delay varying from 1.5 to 2.5 and having a dynamic component of 0.5:

```
icc2_shell> set_clock_latency 1.5 -source -early -dynamic 0.5 [get_clocks CLK1]
```

```
icc2_shell> set_clock_latency 2.5 -source -late -dynamic 0.5 [get_clocks CLK1]
```

To remove the source latency you specify with the [set_clock_latency](#) command, use the [remove_clock_latency](#) command.

❖ Specifying Ideal Network Latency

To specify a clock as an ideal clock, use the [set_ideal_network](#) command. Until you complete clock tree synthesis, specify all clocks as ideal.

To specify the ideal network latency, use the [set_clock_latency](#) command.

The following example sets the expected rise latency to 1.2 and the fall latency to 0.9 for the clock names CLK1 for the current design:

```
icc2_shell> set_clock_latency -rise 1.2 [get_clocks CLK1]
```

```
icc2_shell> set_clock_latency -fall 0.9 [get_clocks CLK1]
```

❖ Specifying Clock uncertainty

[set_clock_uncertainty](#) :

For designs with multiple scenarios, by default, the clock uncertainty applies only to the current scenario. To specify the clock uncertainty for:

- All the scenarios of a specific modes, use the -modes option.
- All the scenarios of specific corners and the current mode, use the -corners option.
- All the scenarios of specific modes and corners, use the -modes and -corners options

Specific scenarios, use the -scenarios option.

When you use this option, you cannot use the -modes or -corners options.

For example, to set a simple setup uncertainty of 0.21 and a hold uncertainty of 0.33 for all paths leading to endpoints clocked by the clock named CLK1, use the following commands:

```
icc2_shell> set_clock_uncertainty -setup 0.21 [get_clocks CLK1]
```

```
icc2_shell> set_clock_uncertainty -hold 0.33 [get_clocks CLK1]
```

To set an interclock uncertainty of 2 between clocks named CLKA and CLKB, for both setup and hold, use the following commands:

```
icc2_shell> set_clock_uncertainty 2 -from [get_clocks CLKA] -to [get_clocks CLKB]
```

```
icc2_shell> set_clock_uncertainty 2 -from [get_clocks CLKB] -to [get_clocks CLKA]
```

To remove clock uncertainty settings, use the `remove_clock_uncertainty [all_clocks]` command.

❖ Report clock qor

Reports largest and smallest latency sinks.

```
prompt> report_clock_qor -clocks clk -type latency
```

Reports all latency sinks.

```
prompt> report_clock_qor -clocks clk -type latency -all
```

Reports all latency sinks. The **-largest** and **-smallest** options are ignored.

```
prompt> report_clock_qor -clocks clk -type latency -all -largest 10 -smallest 10
```

```
report_clock_timing -type report_type
```

```
prompt> report_clock_timing -type summary
```

```
prompt> report_clock_timing -type latency
```

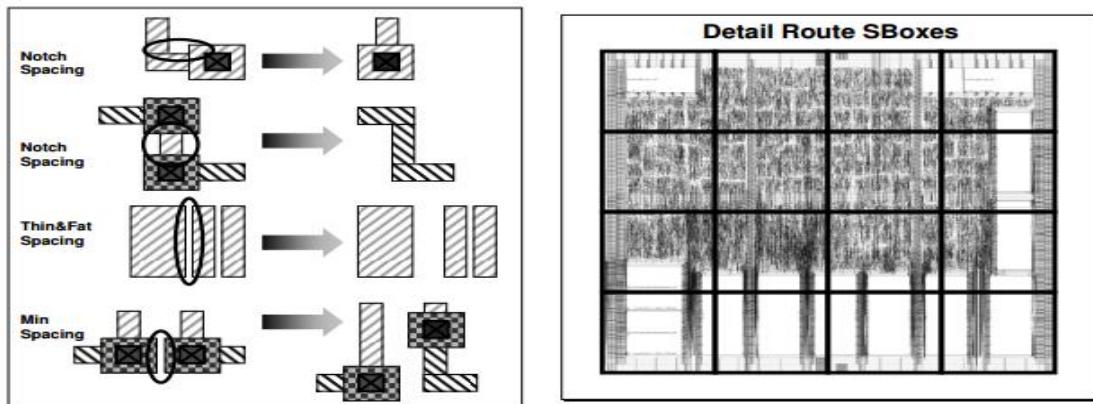
```
prompt> report_clock_timing -clock CLK1 -type skew -setup \
-nworst 5 -include_uncertainty_in_skew
```

Before performing clock tree synthesis, you must use the `remove_ideal_network` command to remove the ideal setting on the fanout of the clock trees.

61 Distinguish Detail route (DR) and Search & Repair (S&R)

❖ DR:

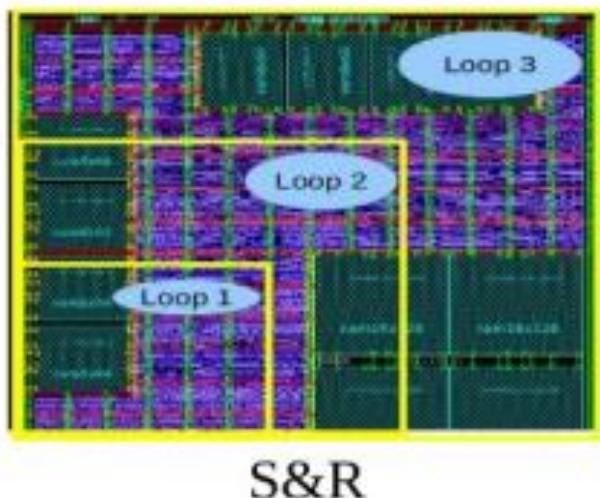
- Using in initial routing step. Initial routing has 3 main steps: global routing → track assignment → detail routing.
- Detail route: This step will draw the real metal for all of connection in design and attempts to clear DRC violations using a **fixed size Sbox**. The detail router does not work on the entire chip at the same time. Instead it works by rerouting within the confines of a small area called a “SBox”. **Due to the fixed Sbox size, detail route may not be able to clear all DRC violations.**
- Concurrent optimization of design rules, antenna rules, wire optimization, and via optimization during detail routing.
- You can restrict the routing to a specific area of the block by using the - coordinates option (or by specifying or selecting the bounding box in the GUI).
- Some command in ICC2:
 - % route_auto (will run global routing → track assignment → detail routing)
 - % route_detail (perform only detail routing)
 - % route.detail_options (For information about the application options that affect detail routing)
 - % route_group -all_clock_nets command runs global routing, track assignment, and detail routing on the clock nets.
 - %route_eco (This command performs ECO routing. It should be used after signal routing has been completed. It also runs global routing, track assignment, detail routing when the design still has open nets)



❖ S&R:

- Search & Repair: Fixes the short and DRC violations through multiple loops **using progressively larger SBox sizes.**
- In ICC2, by default, the detail router performs a maximum of 40 search and repair iterations.
- Some command or option in ICC2:

% -max_detail_route_iterations (Option to modify the maximum number of detail routing iterations).



60 What is routing layer? Global and semi global layer?

- ❖ **Routing layer:** is the layer which we use for metal connection, nowadays the chip is more compact, so we need more layers to deal with this.
- ❖ **Global layer:** at the top layer of chip. This layer contain only PG (Power and Ground) or TOP connection (RDL routing, ...)
- ❖ **Semi global layer:** the wider/thicker layers than fine layers. We can use this semi-global layer with some purposes like clock line routing (to reduce clock latency), signal routing (to reduce wire delay), or PG which required small resistance.
- ❖ **Fine layer:** based on the manufacture and each technology, the manufacture such as TSMC will offer a package of fine layer according to each product (low performance CHIP/high performance chip/ASIC/GPU will have difference fine layers). Fine layer is defined by M_x layers. M₁ is NOT M_x layer. In order to determine the M_x layer, please check the process name.

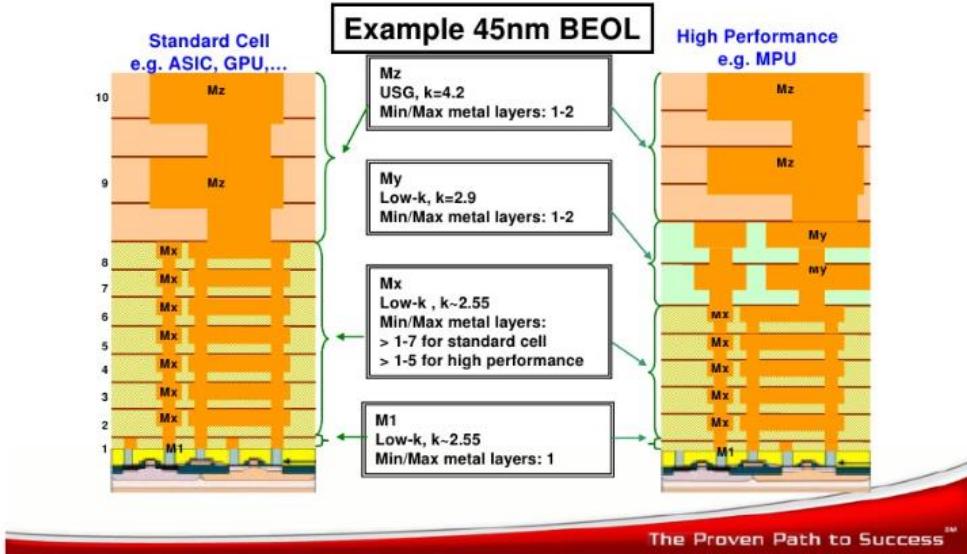
Example: 4x2y2z or 4x4y1z1r. In this example, 4x means there are 4 M_x in the process (M_x will be started by M₂ and stopped at M₅); or 4y means there are 4 semi-global layers and so on. For more detailed, please check at cross-section of metal/via in each Design manual.

- ❖ Below is example picture which describe the routing layer:

Customized Interconnect Offerings

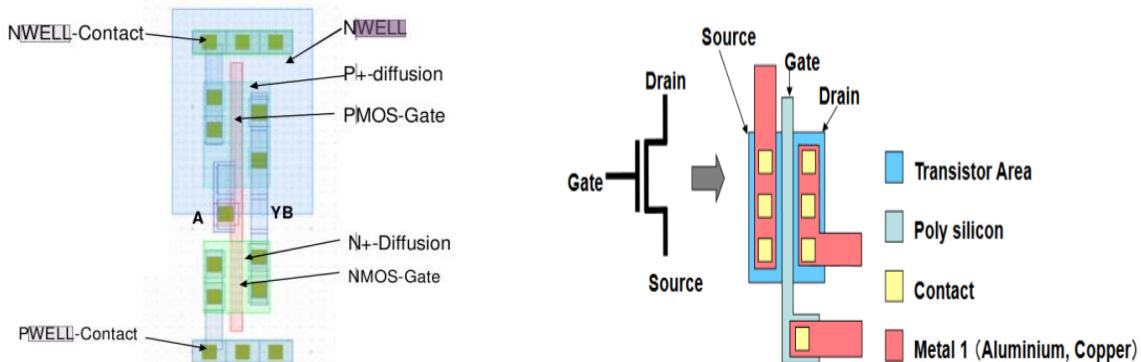


➤ TSMC offers customized interconnect to meets the needs of high packing density, high performance (low-K, low-R), and low cost.



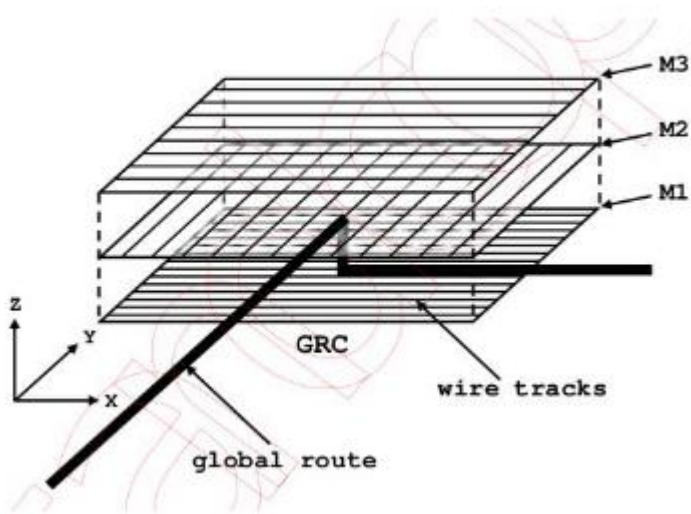
98. Via vs contact (mat cat dung)

- ❖ **Via:** connector between metal layer.
- ❖ **Contact:** used to connect metal 1 to Gate, Source, Drain of transistor.



69 What is routing congestion? How to check and improve? (TriPhung-san)

- ❖ **GRCs:** is a small box that is divided by each track. Its height equal to STD cell.



- GR determines whether each assigned GRC (Global Routing Cells) along a path has enough wire tracks for the assigned nets through the edges of that GRC.
- If there are not enough wire tracks, GR reassigns metal layers or GRCs accordingly.
- Check GRC: Example, 1 GRC allow 5 net. If this GRC has 7 net, overflow is 2.

❖ **Congestion:**

- When cells are closed together at the area.
=> It's hard for routing and may cause short.
- Congestion occurs if there are more wires to be routed than available tracks.
=>Congestion (rout-ability) is a function of the number of available tracks in a given area compared to the number of signals that need to be routed through that area.

Congestion, timing and max cap/transition are estimated based on virtual route or a throw away global route.

66 What is Scan method, Scan FF, Scan chain?

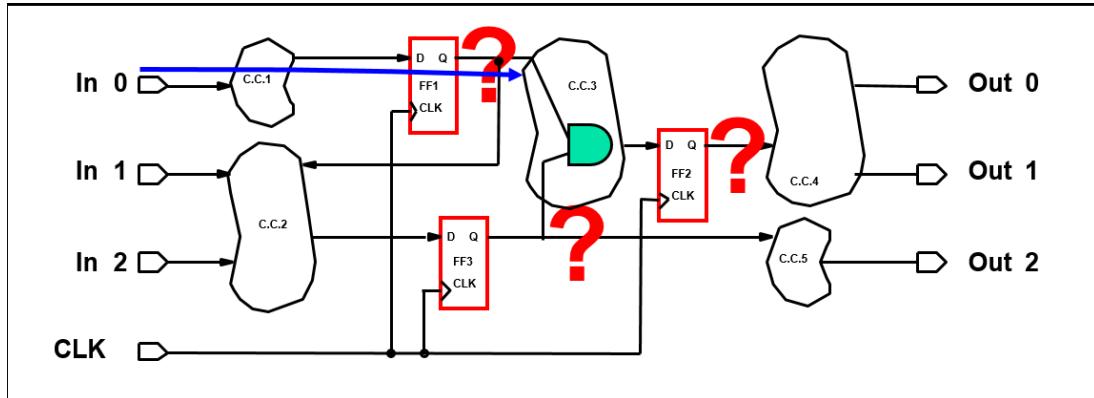
DFT is a technique, which facilitates a design to become testable after production. It's the extra logic which we put in the normal design, during the design process, which helps its post-production testing, and **SCAN** is one kind method of DFT.

❖ **Scan method:**

Memory element (FF/Latch) in the sequential logic cannot be controlled /monitored if the procedure is not carried out sequentially ([Picture 1](#)). This leads to decrease in testability, and auto-generation of the test pattern becomes difficult. The port of each memory element is modeled as a virtual I/O port ([Picture 2](#)) by monitoring/controlling the internal state of memory element, and the entire unit is treated as a combinational circuit as a method of facilitating auto-generation of the test pattern and improving the testability. This method of improving the Testability by monitoring / controlling the memory element in the sequential logic

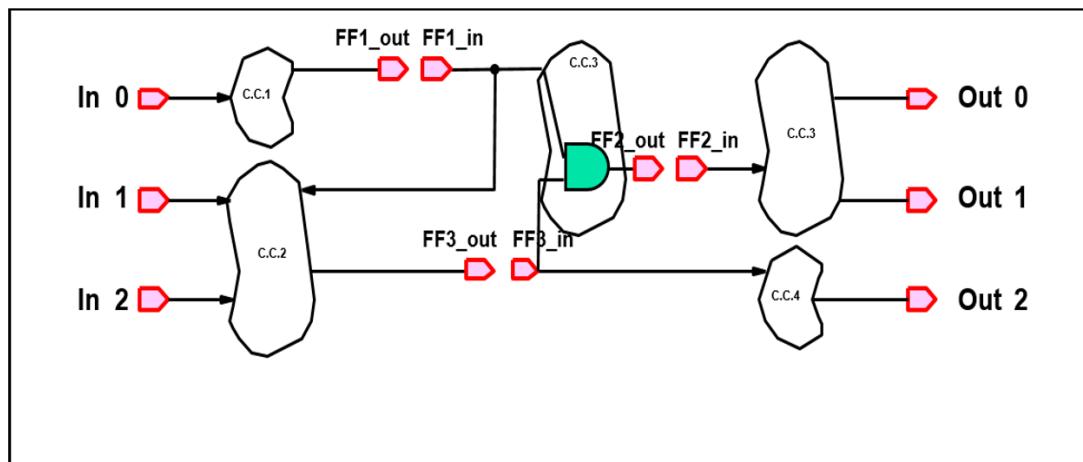
is called "Scan Method" and change / modification of the circuit to achieve this method is called "Scanning".

Example:



Picture 1: Cannot test combinational logics if FF operates as a sequential circuit

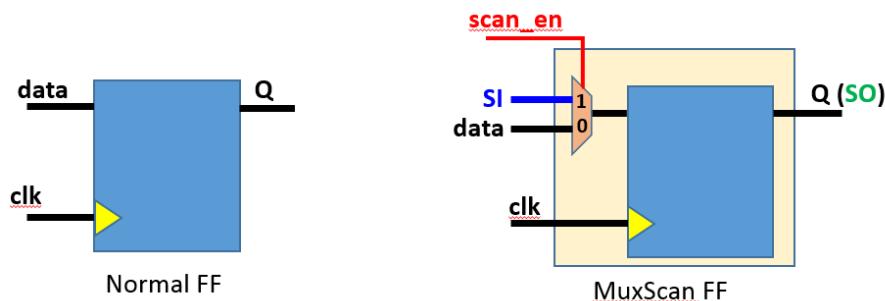
Testability is lowered because flip-flops (FF₁, FF₂ and FF₃) is a memory unit.



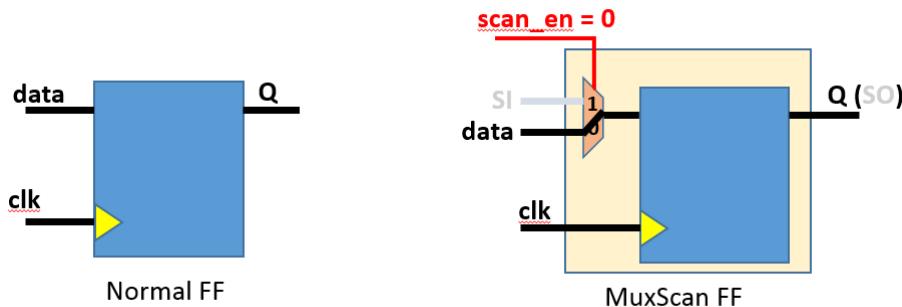
Picture 2: Treat FF as virtual input-output port

Treat FF₁ - FF₃ as virtual input-output ports and handle the entire circuit as combination circuits.

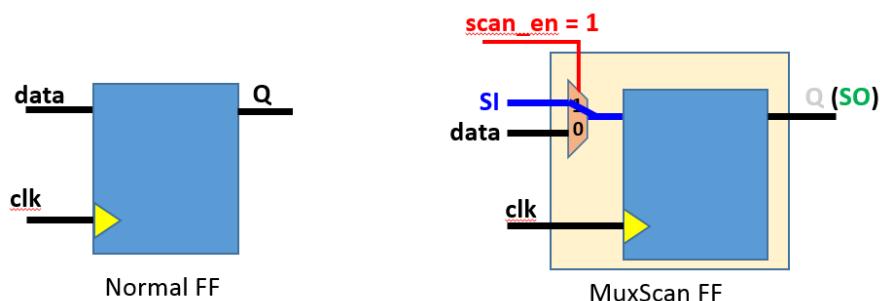
❖ Scan FF:



- **scan_en = 0:** The data input to FF from data port (ScanFF operates as a Normal FF)



- **scan_en = 1:** The data input to FF from SI port

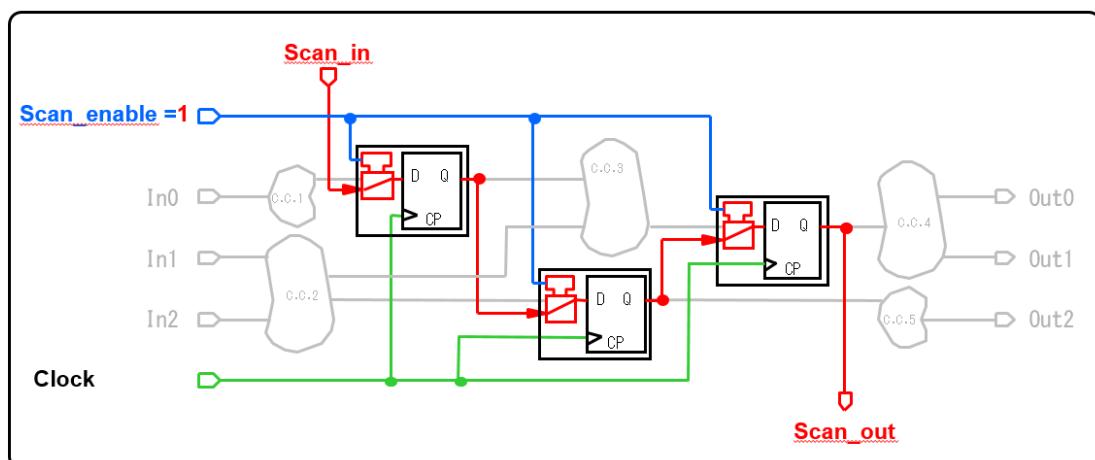


❖ Scan chain:

Scan chain is the path that serially connects FF through SI (scan-in) port and SO (scan-out) port.

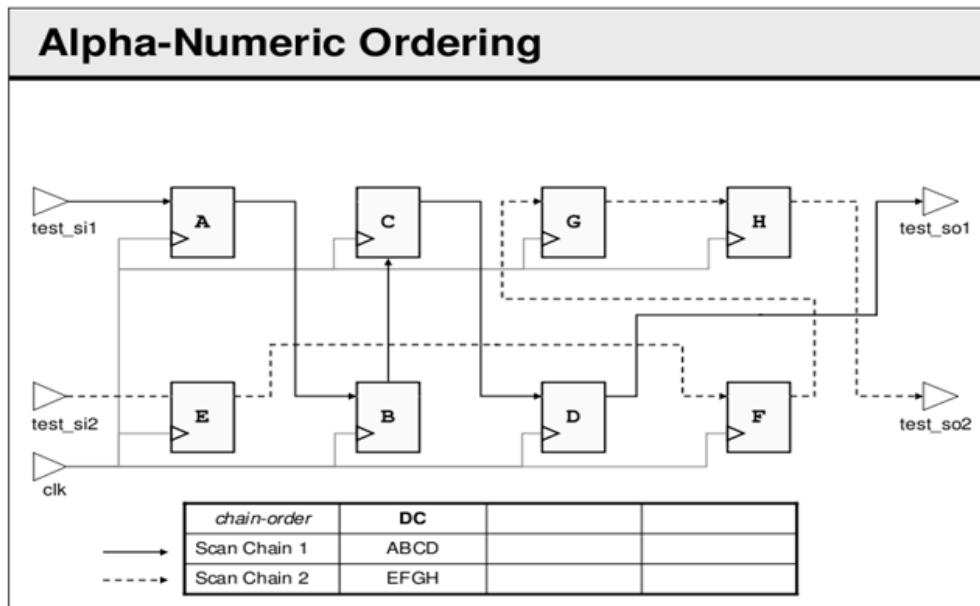
Blue Normal/SCAN operation switching circuit

Red Scan chain

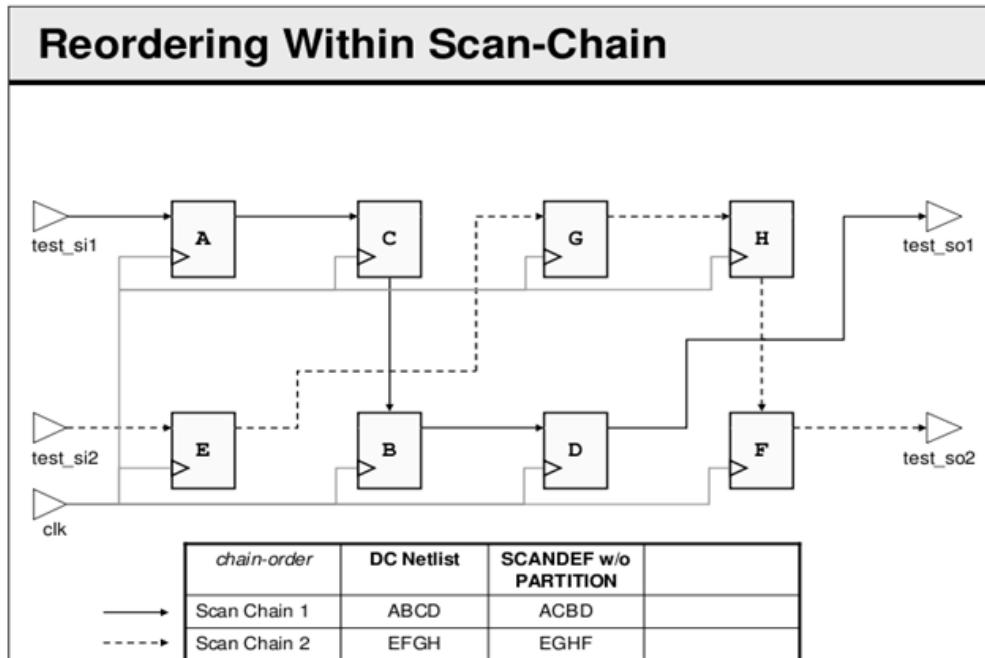


76 Why we need to make Scan chain reorder?

- ❖ Scan chain reordering is applied during placement optimization and clock optimization steps. It helps optimizing routing more easily, saves routing resources.
- ❖ Before reordering: Default scan chain is created sequential by Alpha-Numeric => If FF A is far away from FF B => waste resources.



- ❖ After reordering: FF is reconnected to get better routing.



40 What is SDC file? Why we need SDC file?

- ❖ **SDC file:** contains the rules that are written are referred to as constraints and are essential to meet designs goal in terms of Area, Timing and Power to obtain the best possible implementation of a circuit.

There is a common format, for constraining the design, which is supported by almost all the tools, and this format is called **SDC - Synopsis Design Constraints** format.

The file is saved with an **.sdc** extention.

SDC syntax is a TCL based format, that is, all commands follow the TCL syntax. SDC contains mainly following constraints that are very essential for design.

- Clock definition
- Generated clock
- Input/Output delay
- Min/Max delay
- False path
- Multi cycle path
- Case analysis
- Disable timing arcs

The constraints are the following types according the commands types:

- Basic commands
- Object Access Commands
- Timing Commands
- Environment Commands
- Multi-Voltage Commands

- ❖ **Why we need SDC file?**

The EDA (Design Compiler, IC Compiler, and PrimeTime (from Synopsys vendor) or Innovus (from Cadence vendor) ...) tools use the SDC description to synthesize, analyze and optimize design. Base on the design specification, SDC will guide EDA tools to understand and achieve the target specification such as timing, power based on condition that we made of.

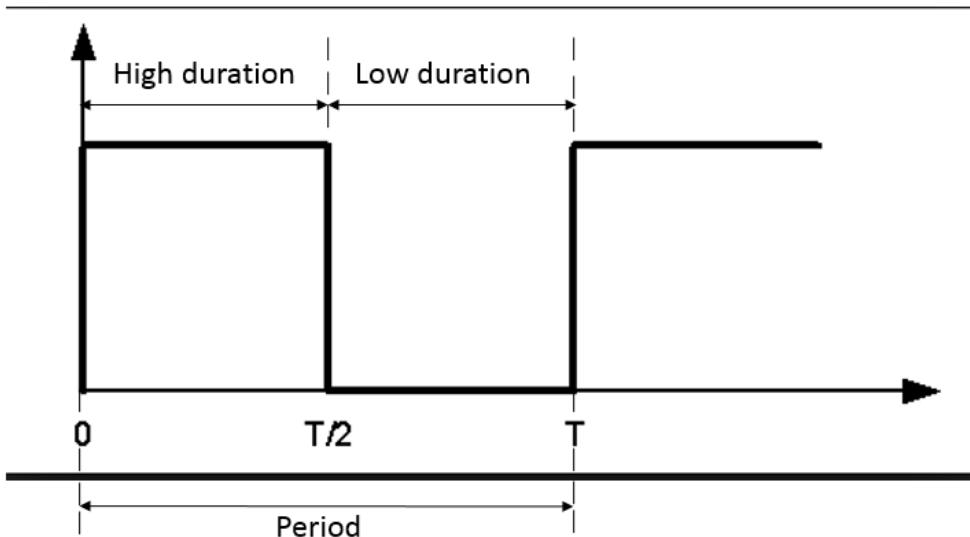
34 Clock Duty Cycle, clock period?

- ❖ To define a clock, we need some information from Specification:
 - Clock source: It can be a port of the design, or be a pin of a cell inside the design (typically that is part of a clock generation logic).
 - Clock period: The time period of the clock.
 - Clock duty cycle: Refer the below formula:

$$\text{Clock duty cycle (\%)} = \frac{\text{High Duration}}{\text{Clock Period}} * 100\%$$
 - Edge period: The period for the rising edge and the falling edge.

Example: a 50% clock duty cycle means the signal is on 50% of the time and off 50% of the time in 1 clock period.

Clock Period: Time of one period of clock. Usually count by ns (nanosecond).

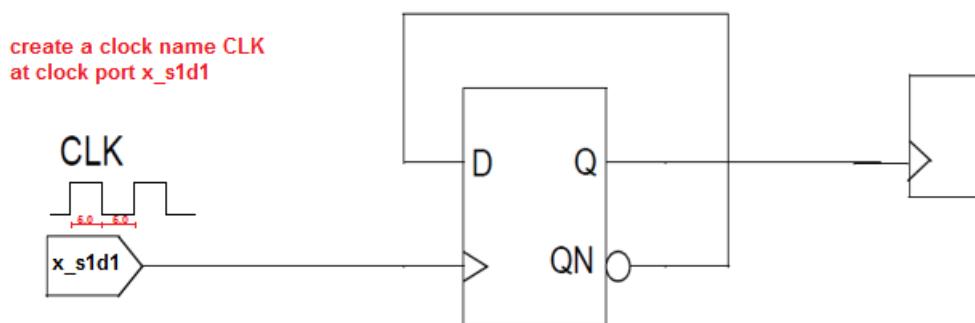


24 Create_clock vs create_generated_clock (When create_clock? Experienced? STA document? virtual clock?) (HoangHoang-san)

- ❖ All of the clocks are identified in the SDC files by below commands:
 - **create_clock**
 - **create_generated_clock**
- ❖ Description of **create_clock**: Creates a clock object. It is created in the current design and is applied to the specific sources.

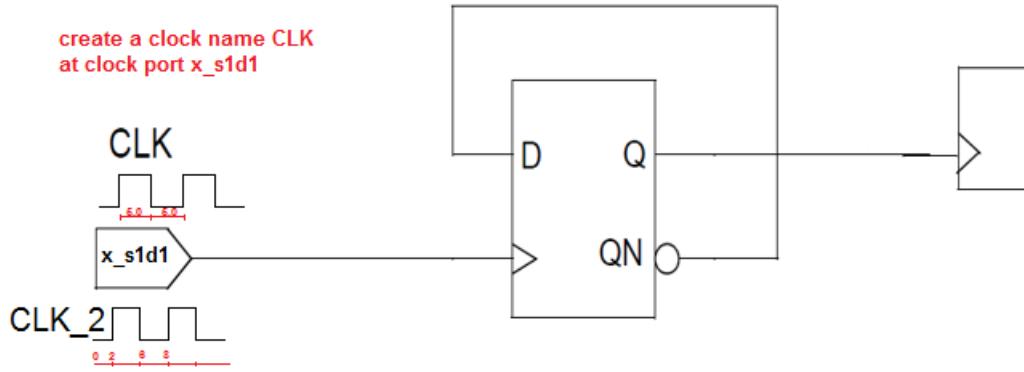
The following example creates a clock named CLK on port x_s1d1 with a period of 10.0

- `icc2_shell> create_clock -name CLK -period 10.0 [get_ports x_s1d1]`



The following example creates another clock named CLK_2 on port x_s1d1 with a rising edge at 2.0 and a falling edge at 8.0 on the same port (period =8.0)

- `icc2_shell> create_clock -name CLK_2 -period 8 -waveform [2.0 6.0] [get_ports x_s1d1]`



- ❖ Description of **create_generated_clock**: Created a generated clock object. Creates a generated clock in the current design.

You can specify a pin or a port as a generated clock object. The command also specifies the clock source from which it is generated.

The advantages of using this command is that whenever the master clock changes, the generated clock automatically changes.

The generated clock can be created as:

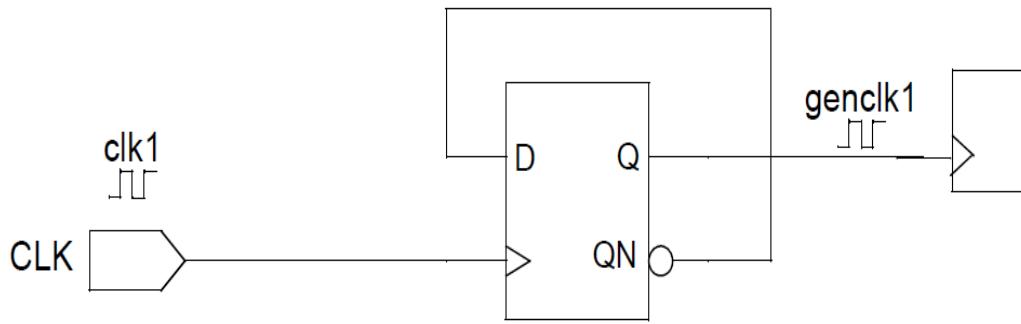
- A frequency divided clock (by using the **-divide_by** option),
- Frequency multiplied clock (by using the **-multiply_by** option),
- Special divide by one (by using the **-combinational** option),
- Or an edge-derived clock (by using the **-edges** option).

The frequency-divided or frequency-multiplied clock can be inverted by using the **-invert** option.

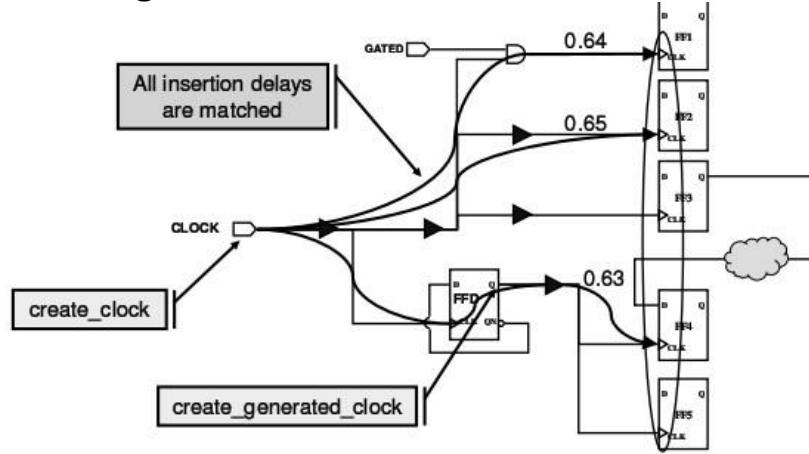
The shifting of edges of the edge-derived clock is specified by using the **-edge_shift** option. The **-edge_shift** option is used for intentional edge shifts and not for clock latency.

The following commands create a generated clock named genclk1 on output pin Q of FF port named CLK, clock source is clk1 (on clock port CLK)

- **icc2_shell> create_generated_clock -name genclk1 -source [get_port CLK] -divide_by 2 [get_pins FF/Q]**

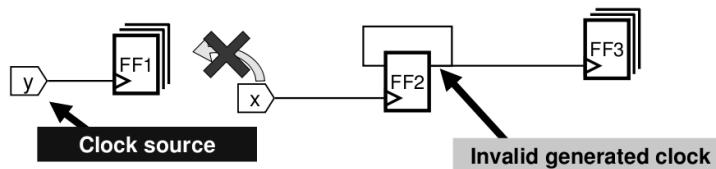


Some notes for generated clock:



- A generated clock must be able to trace back to its clock source as identified with -source option

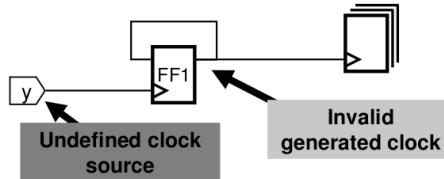
```
create generated_clock -name clk -source [get_ports y] \
    -divide_by 2 [get_pins FF2/z]
```



- In this case, the sinks of generated clock (FF3) will not be balanced with the sinks of its source (FF1)

- The source of a generated clock must be defined

```
create_generated_clock -name div2 \
-source [get_ports y] -divide_by 2 [get_pins FF1/z]
```



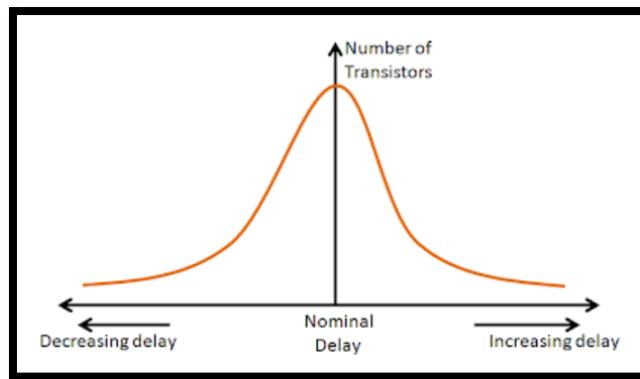
- Tool can not synthesize a clock tree for an invalid generated clock

If the master-clock source is not a clock source defined by the `create_clock` or `create_generated_clock` command, the tool cannot synthesize a clock tree for the generated clock or its source.

Use the `check_clock_trees` command to verify that your master-clock sources are defined correctly.

42 Explain OCV value?

Static timing analysis of a design is performed to estimate its working frequency after the design has been fabricated. Nominal delays of the logic gates as per characterization are calculated and some pessimism is applied above that to see if there will be any setup and/or hold violation at the target frequency. However, all the transistors manufactured are not alike. Also, not all the transistors receive the same voltage and are at same temperature. The characterized delay is just the delay of which there is maximum probability. The delay variation of a typical sample of transistors on silicon follows the curve as shown in figure 1. As is shown, most of the transistors have nominal characteristics. Typically, timing signoff is carried out with some margin. By doing this, the designer is trying to ensure that more number of transistors are covered. There is direct relationship between the margin and yield. Greater the margin taken, larger is the yield. However, after a certain point, there is not much increase in yield by increasing margins. In that case, it adds more cost to the designer than it saves by increase in yield. Therefore, margins should be applied so as to give maximum profits.



We have discussed above how variations in characteristics of transistors are taken care of in STA. These variations in transistors' characteristics on as fabricated on silicon are known as OCV (On-Chip Variations). The reason for OCV, as discussed above also, is that all transistors on-chip are not alike in geometry, in their surroundings, and position with respect to power supply

27 Distinguish Elmore, Arnoldi vs AWE mode

Overview of Delay Calculation

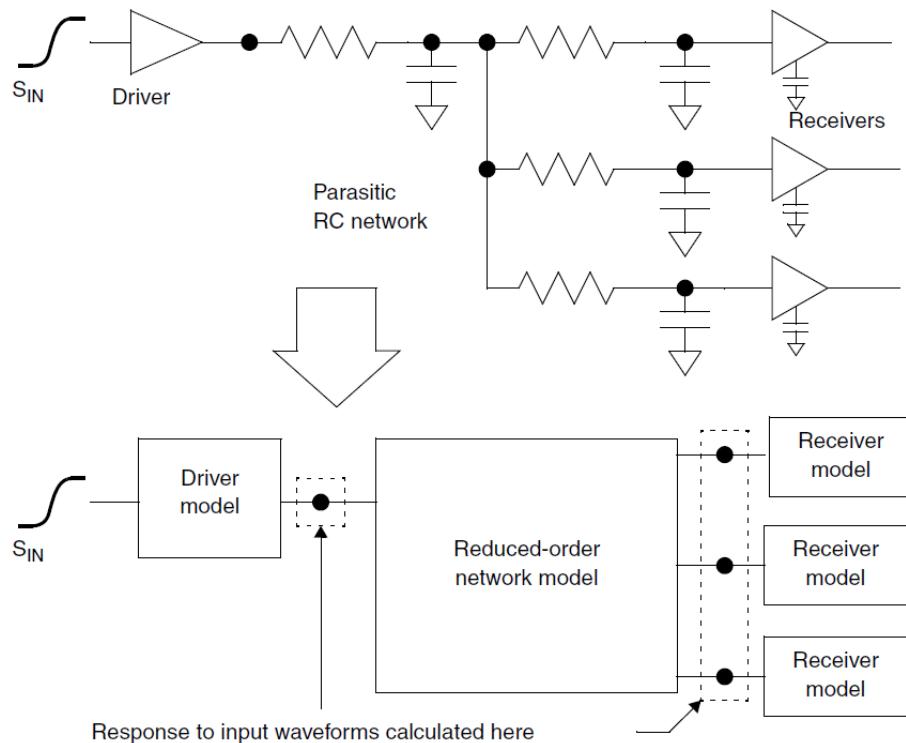
To perform static timing analysis, tool must accurately calculate the delay and slew (transition time) at each stage of each timing path. A stage consists of a driving cell, the annotated RC network at the output of the cell, and the capacitive load of the network load pins. The goal is to compute the response at the driver output and at the network load pins, given the input slew or waveform at the driver input, using the least amount of runtime necessary to get accurate results.

To perform stage delay calculation accurately and efficiently, tool uses models to represent the driver, RC network, and capacitive loads on the net.

The driver model is intended to reproduce the response of the driving cell's underlying transistor circuitry when connected to an arbitrary RC network, given a specific input slew.

The receiver model is intended to represent the complex input capacitance characteristics of a cell input pin, including the effects of rise/fall direction, the slew at the pin, the receiver output load, the state of the cell, and the voltage and temperature conditions.

The reduced-order network model is a simplified representation of the full annotated network that has nearly the same response characteristics as the original network. Tool can use Elmore, Arnoldi and AWE models.



By default, ICC uses the Elmore delay model for preroute delay calculation and the Arnoldi delay model for routed clock and postroute delay calculation. To change the delay calculation model, use the [set_delay_calculation_options](#) command.

The Elmore model is faster than but not as accurate as the Arnoldi model. The asymptotic waveform evaluation (AWE) model is also available as an option; it has most of the speed of the Elmore model and most of the accuracy of the Arnoldi model.

To determine which delay model is being used, check for the symbol used next to the incremental delay in the report generated by the [report_timing](#) command.

If you are using AWE delay models, the report would contain a w symbol, as shown in the following example:

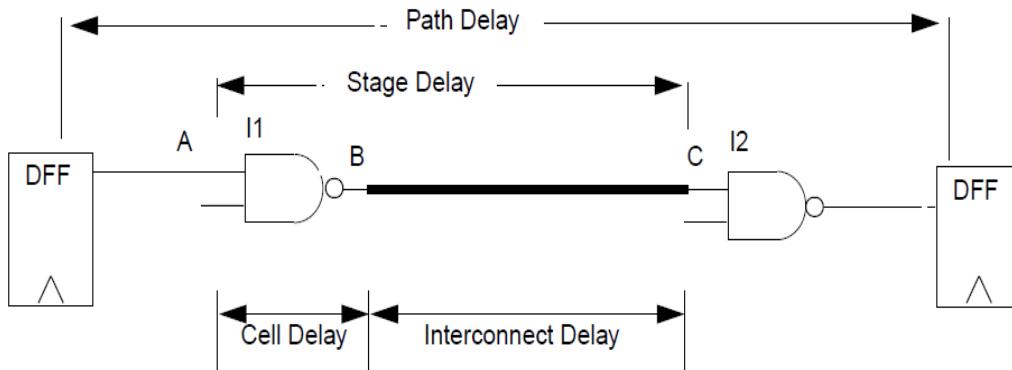
Startpoint: sdram_clk (clock source 'SDRAM_CLK')	Endpoint: sd_DQ_out[15] (output port clocked by SD_DDR_CLK)	Point	Incr	Path
clock SDRAM_CLK (fall edge)			3.75	3.75
sdram_clk (in)			0.00	3.75 f
I_SDRAM_TOP/sdram_clk (SDRAM_TOP)			0.00	3.75 f
sd DQ out[15] (out)			0.01 w	4.49 f

If you are using Elmore delay models, the report would contain a * symbol, as shown in the following example:

Startpoint: sdram_clk (clock source 'SDRAM_CLK')	Endpoint: sd_DQ_out[15] (output port clocked by SD_DDR_CLK)	Point	Incr	Path
clock SDRAM_CLK (fall edge)			*	3.75
sdram_clk (in)			*	3.75
I_SDRAM_TOP/sdram_clk (SDRAM_TOP)			*	3.75
sd DQ out[15] (out)			*	4.49

clock SDRAM_CLK (fall edge)	3.75	3.75
sdram_clk (in)	0.00	3.75 f
I_SDRAM_TOP/sdram_clk (SDRAM_TOP)	0.00	3.75 f
sd_DQ_out[15] (out)	0.01 *	4.49 f

43 Cell delay and net delay?



The cell (t_{cell}) delay is the delay through the logic element as determined by the cell's intrinsic delay, the load on the cell, and input edge rate to the cell. The delay from A to B is an example of a cell delay.

$$T_{cell} = t_{intrinsic} + t_{load_dep}$$

The intrinsic delay ($t_{intrinsic}$) is defined as the delay between an input and output pin pair of a cell, when the output pin does not see any load (no-load condition).

The load-dependent delay (t_{load_dep}) is the additional delay that exists between an input and output pin pair of a cell, when the output pin of the cell is driving a certain load. In above, the extra delay between A and B, when B is connected to the net, is an example of a load-dependent delay.

When looking up the value in library, you just see 1 value which is calculated by a formula of {input transition, output capacitance}. You can check the **Question ... : "What is NLDM and CCS library? When to use CCS and when to use NLDM?"** for more detail.

The interconnect delay (or net delay) is the delay between an output pin of one cell to the input pin of another cell. The delay from B to C is an example of an interconnect delay.

26 Explain Slack, Slew (Transition time) vs Skew

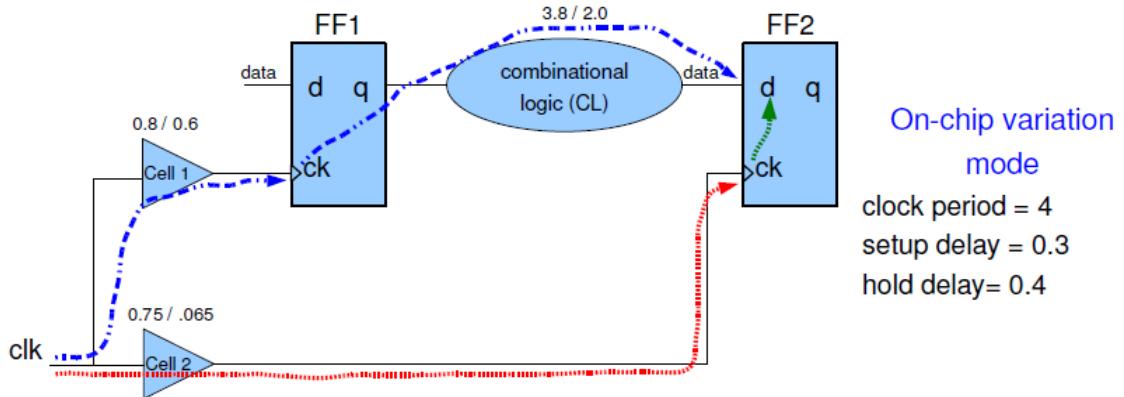
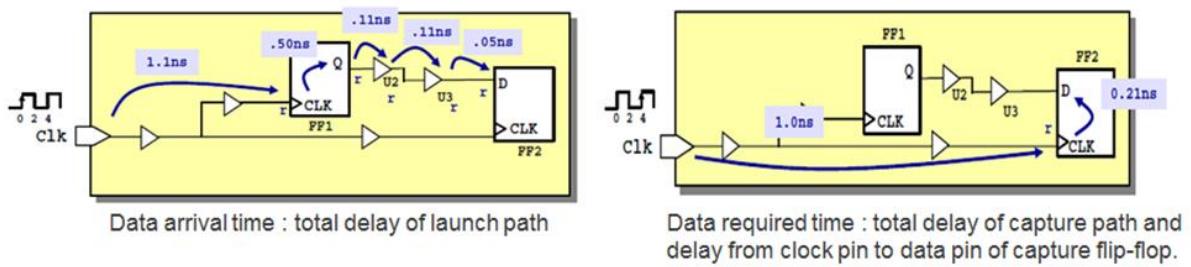
- ❖ **Slack:**

The amount of time by which a violation is avoided. Slack is calculated by the difference between arrival time and required time of data. (setup and hold)

$$\text{Slack} = \text{data required time} - \text{data arrival time} \quad (\text{for setup check})$$

$$= \text{data arrival time} - \text{data required time} \quad (\text{for hold check})$$

A slack of 0 means that the timing constraint is just barely satisfied. A negative slack indicates a timing violation.



★ Setup check:

$$\begin{aligned} \text{data required time} &= \text{Cell2(min)} + \text{clk period} - \text{setup delay} \\ &= 0.65 + 4 - 0.3 = 4.35 \end{aligned}$$

$$\begin{aligned} \text{data arrival time} &= \text{Cell1(max)} + \text{FF1(max)} + \text{CL(max)} \\ &= 0.8 + 0 + 3.8 = 4.6 \end{aligned}$$

$$\begin{aligned} \text{slack} &= \text{data required time} - \text{data arrival time} \\ &= 4.35 - 4.6 = -0.25 \Rightarrow \text{Violated} \end{aligned}$$

★ Hold check:

$$\begin{aligned} \text{data required time} &= \text{Cell2(max)} + \text{hold delay} \\ &= 0.75 + 0.4 = 1.05 \end{aligned}$$

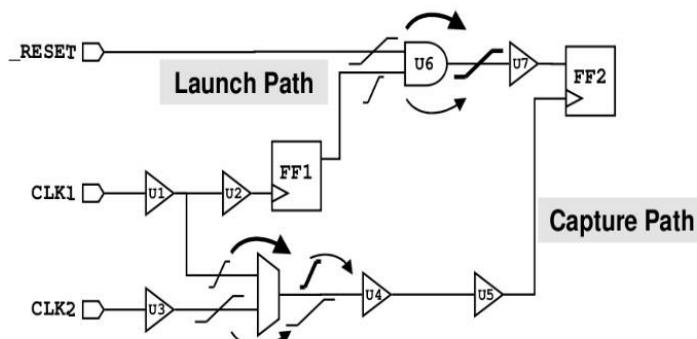
$$\begin{aligned} \text{data arrival time} &= \text{Cell1(min)} + \text{FF1(min)} + \text{CL(min)} \\ &= 0.6 + 0 + 2.0 = 2.6 \end{aligned}$$

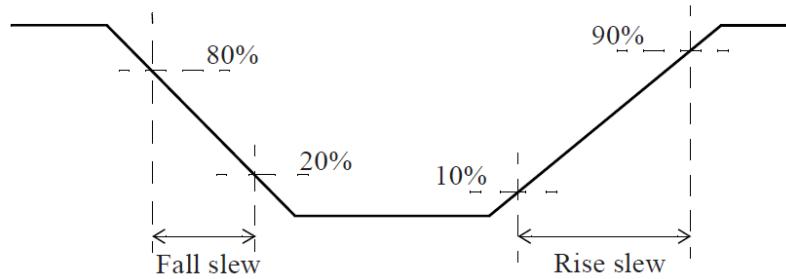
$$\begin{aligned} \text{slack} &= \text{data arrival time} - \text{data required time} \\ &= 2.6 - 1.05 = 1.55 \Rightarrow \text{Met} \end{aligned}$$

❖ Slew:

This is the amount of time it takes for a signal to change from logic low to logic high (rise time), or from logic high to logic low (fall time), also known as [transition time](#).

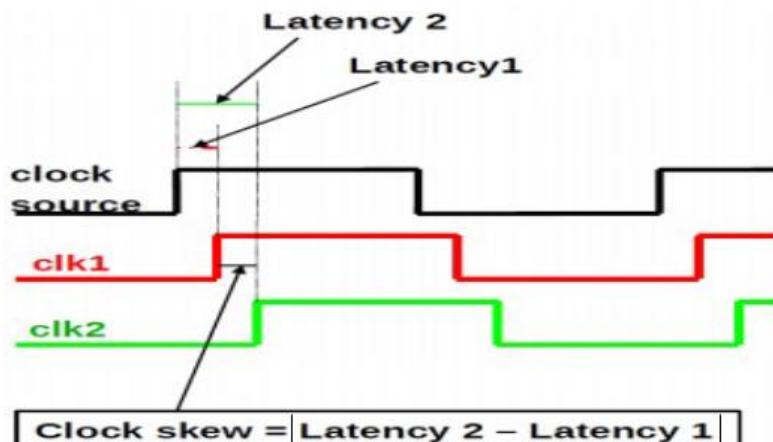
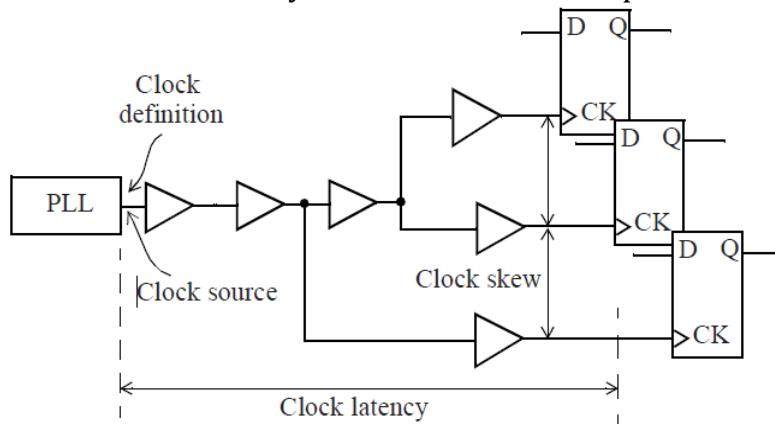
Slew is defined as a rate of change. In static timing analysis, the rising or falling waveforms are measured in terms of whether the transition is slow or fast.





❖ Skew:

The difference in clock latency between two different points in a clock network.



31 Jitter vs Margin

❖ Jitter

Jitter is amount of different time between cycles of a clock.

Jitter is one factor affect to Clock Uncertainty (CU).

$$\text{Equation: } 0.1\sqrt{T/2}$$

❖ Margin

Margin is an estimated value, depending on operating condition (process, voltage, temp).

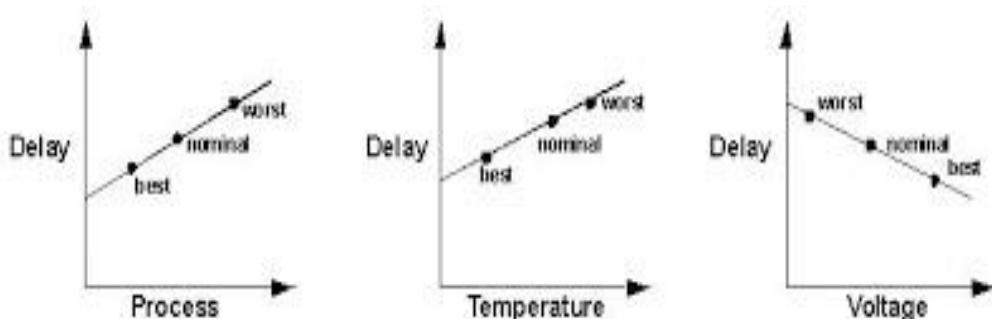
Purpose: margin is an additional value to make constraint more strictly for assuring meet timing in the worst case.

Margin value is given by producer and defined in specification.

41 What is PVT (operating condition) and how it effects to the design? (Nagasan)

The operating conditions of a design include the process, voltage, and temperature parameters under which the chip is intended to operate. The EDA tools analyze and optimize the design under the conditions you specify. Operating conditions are usually specified in the logic library of the design.

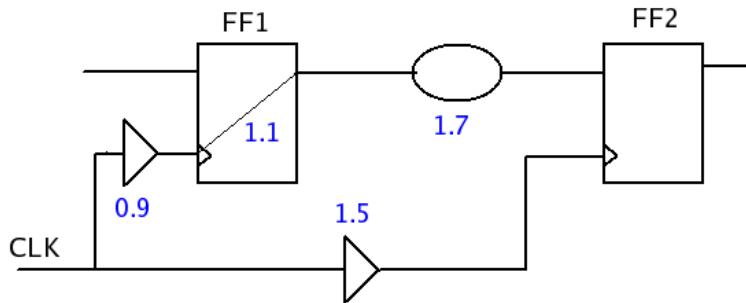
- ❖ **Process:** the outcome of the actual semiconductor manufacturing steps, typically 1.0 for most technologies. Relating to the scaling of the device parameters, small process number results smaller delay.
- ❖ **Supply voltage:** Voltage using; delay of cell dependent on the saturation current. Power supply inflects the propagation delay. (switching speed – IR drop). Higher voltage results smaller delay.
- ❖ **Temperature:** temperature in operation of the design. This parameter affect power consumption and operation.
- ❖ **Tree-type:** The definition for the environment interconnect model for calculating interconnect delays:
 - *best_case_tree*
Models the case in which the load pin is physically adjacent to the driver. In the best case, all wire capacitance is incurred but none of the wire resistance must be overcome.
 - *balanced_tree*
Models the case in which all load pins are on separate, equal branches of the interconnect wire. In the balanced case, each load pin incurs an equal portion of the total wire capacitance and wire resistance.
 - *worst_case_tree*
Models the case in which the load pin is at the extreme end of the wire. In the worst case, each load pin incurs both the full wire capacitance and the full wire resistance.



PrimeTime offers 3 analysis modes with respect to operating conditions, called the single, best case/worst case and on-chip variation (OCV) modes

Single operating condition: PrimeTime uses a single set of delay parameters for the whole circuit, based on one set of process, temperature, and voltage conditions.

Example for single mode:



The delay value at each element is fixed (**max delay = min delay**).

The delay (for both setup and hold checking) in launch path and capture path are calculated as follows:

$$\text{launch_delay} = 0.9 + 1.1 + 1.7$$

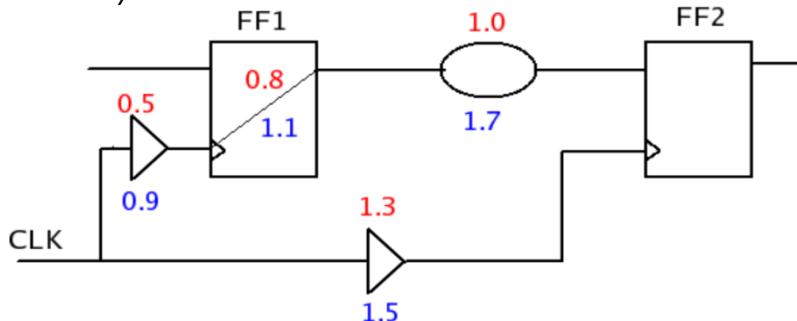
$$\text{capture_delay} = 1.5.$$

Best case/Worst case operating condition: PrimeTime checks the circuit for the two extreme operating conditions, minimum and maximum.

For setup checks, it uses maximum delays for all paths.

For hold checks, it uses minimum delays for all paths.

Example for best case/worst case mode:



The maximum delay and minimum delay at each element are different.

The delay is calculated as follows:

Setup checking (using **maximum delay value** for both paths):

$$\text{launch_delay_setup} = 0.9 + 1.1 + 1.7$$

$$\text{capture_delay_setup} = 1.5$$

Hold checking (using **minimum delay value** for both paths):

$$\text{launch_delay_hold} = 0.5 + 0.8 + 1.0$$

$$\text{capture_delay_hold} = 1.3$$

57 Min condition, max condition? (Move after Q41: PVT)

Operating conditions: Defined as a combination of PVT

PVT: Process, Voltage and Temperature

PVT refers to a variation in cell delays and cell pin capacitances due to fabrication process variance, power supply voltage variations, and temperature

❖ **Max condition:** (process : slow-slow)

- Worst operating conditions for timing check (setup, hold, transition, capacitance, ...), using maximum delay for all paths (cells delay + nets delay)
- OCV: max delay for launch path, min delay for capture path.

❖ **Min condition:** (process : fast-fast)

- Best operating conditions mainly use hold check (because setup is not critical in fast condition), using minimum delay for all paths (cells delay + nets delay)
- OCV: min delay for launch path, max delay for capture path.

In between max-min condition, we also have **Typical condition**, which is the average condition for timing check. By the way, this condition is rarely use in our company.

45 MCMM (ICC2)?

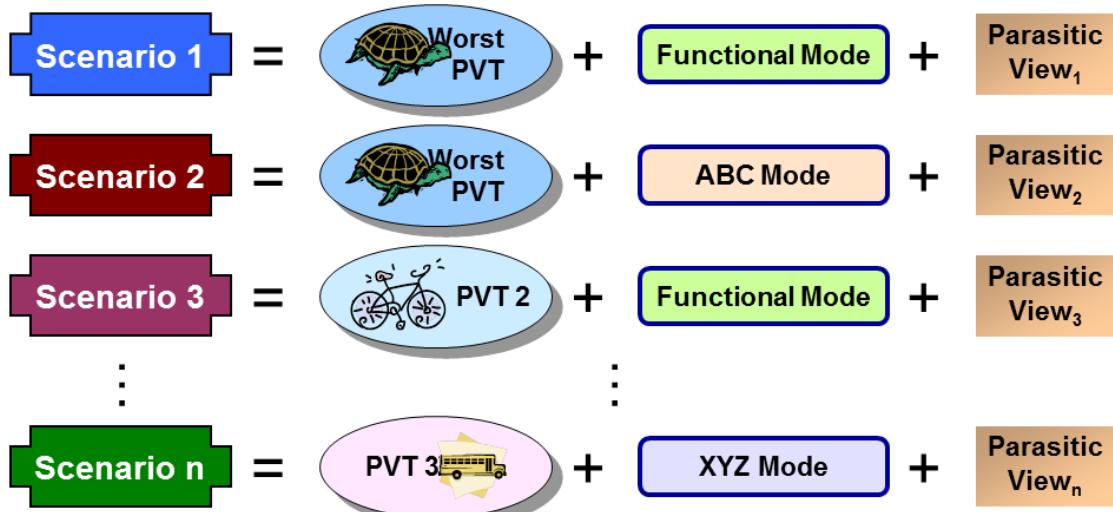
MCMM: Multiple Corners Multiple Modes.

(MMMC: Multiple Modes Multiple Corners - Innovus)

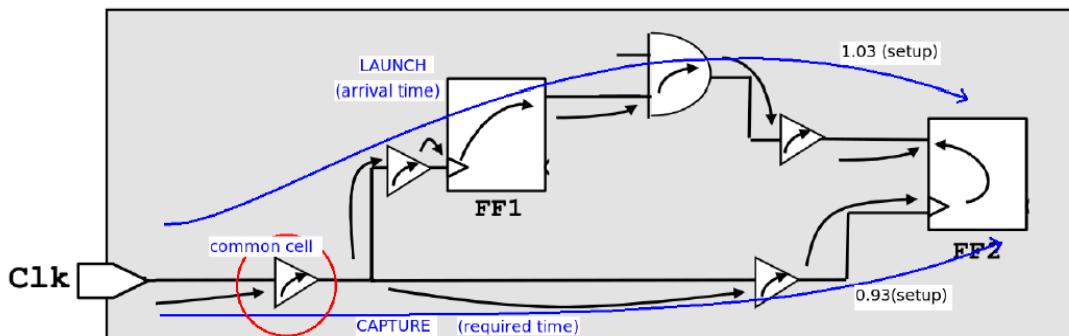
Designs are often required to operate under multiple operating conditions (“corners”) and in multiple modes (USER, DFT ...). Such designs are referred to as multicorner, multimode, or multicorner-multimode designs

When we design a chip, we have to verify a chip under MCMM to ensure that the chip can work well under multiple conditions working and difference modes.

- ❖ A scenario refers to a mode and/or corner that can be analyzed or optimized.
- ❖ A mode is defined by a set of clocks, timing constraints, and libraries. It can also have annotation data such as SDF or parasitics files.
- ❖ A corner is defined as a set of libraries characterized for process, voltage and temperature variations. Corners are not dependent on functional settings, but rather result from process variations during manufacturing along with voltage and temperature variations in the environment in which the chip will operate.
- ❖ More than 2 scenarios are used to verify a chip, we call MCMM.



44 Clock re-convergence pessimism? (HungNguyen)



When we design a chip, we have to verify a chip under MCMM to ensure that the chip can work well under multiple conditions working and difference modes.

In above figure, two flip flop FF1 and FF2 share a common clock but are placed at physically at the different places in the same die. The launch and capture path share a common cell in the clock tree till the point known as “common point”. The 2 clock path diverse from that point.

Every cell has 2 type of delay as a part of its specification, “Max delay” and “Min delay”. There are several scenarios in the design where we use either max delay or min delay of a particular cell. Such as best case analysis, worst case analysis, on chip variation (OCV) analysis during timing analysis.

If during timing analysis a condition arise where you have to use max delay for one timing path and min delay for another timing path (during OCV analysis), you have to use 2 different values for a common path. For examples, you use max delay for launch

path and min delay for capture path. But practically the delays through the common cells can never be different when calculated as part of the launch versus capture path. The delay difference that ICC II calculates for their share cells by default, create pessimistic timing called **clock re-convergence pessimism (CRP)**. The amount of CRP is defined as the difference between the latest and earliest arrival time through the shared or common cells to the common point.

In physical design, the cells along the common portion of the clock tree cannot simultaneously achieve their max and min delay values. Thus there will be a single value of delay to the common point that will be propagated to both launch and capture clock paths. So the effect of CRP should be removed

48 What is size cell/ doping cell?

Size cell is changing the drivability of master cell. Size cell includes up-size & down-size

Doping cell is changing to another cell which has the different Vth.

Ex:

Upsize cell: size_cell U9981 THNQAN21X**20** ;# THNQAN21X**10**

Doping cell: size_cell U9981 **TH**NQAN21X10 ;# **TUH**NQAN21X10

Size cell impacts:

- ❖ Up-size:
 - Advantages: Improve transition, capacitance (setup timing closure); reduce pin density to improve route ability.
 - Disadvantages: increase area, may increase dynamic & leakage power.
- ❖ Down-size:
 - Advantages: Improve hold timing, decrease area
 - Disadvantages: May make transition, capacitance worse;

Doping cell impacts: (Refer to the Question 36)

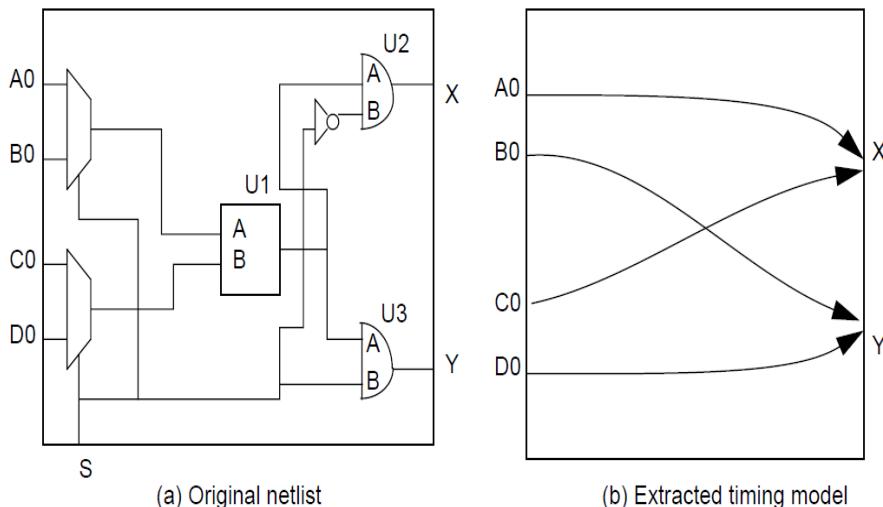
55 False path and True path

False path:

Specifies paths that are not real which implies that these paths are not checked in STA. A false path is ignored by the STA for analysis.

If one is trying to obtain a path report and the STA reports that no path is found, or it provides a path report but the slack is infinite, it may be a false path.

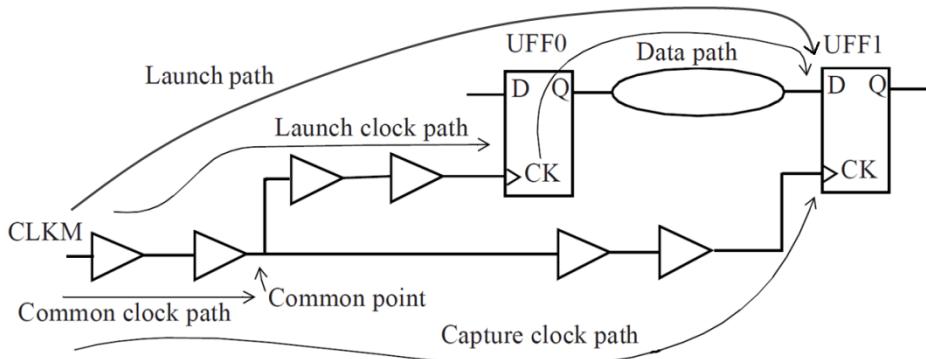
For example, a path can exist between two multiplexed logic blocks that are never selected at the same time, so that path is not valid for timing analysis. In that case, PT still calculate the path delay, but does not report it.



True path:

Is path that care the timing. The timing path that data can propagate through the path
=> We must check the timing for all path.

56 Launch path and Capture path



Launch path: Include the circuit elements from the input port of the design to the clock pin at the data launch point.

Capture path: Show the circuit elements leading up to the clock pin at the data capture point.

According to the figure:

Launch path = Launch clock path + CLK-Q timing arc of start FF + Data path.

Capture path = Capture clock path.

58 Explain timing “critical path” and “non-critical path”

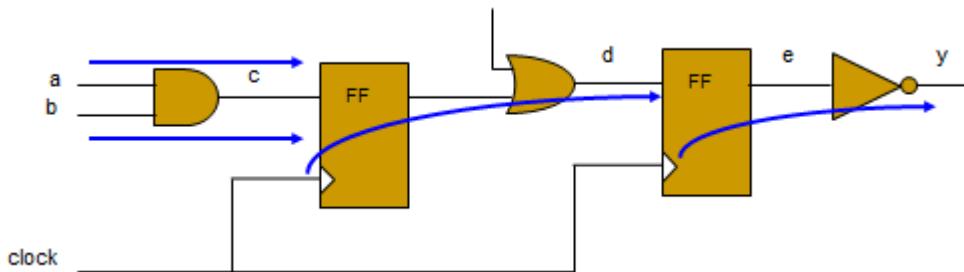
- ❖ Timing critical path: very difficult to fix timing paths, which will need very detail analysis to find the suitable way to fix. Normally these paths should be taken care in advance to have smooth auto optimization by layout tool later, like useful skew, placement care, or something else.
- ❖ Timing non critical path: the paths which we still can see the possibility to fix the timing.

70 What is timing path, ideal mode, propagate mode and how to read it?

(Merge with Q52 – Tri-san)

Timing Path Report					
Point	Fanout	Cap	Trans	Incr	Path
clock LB_AC_C_S2Dn_CA53_CPU0_STGNO_2ZCLK_T0 (rise edge)				0.0000	0.0000
clock network delay (ideal)				0.0000	0.0000
u_ca53_cpu/u_ca53_noram/u_ca53dpu/u_dpu_dbg/editr_reg_18_	0.0000	0.0000	0.0000	0.0000	(rising edge-triggered flip-flop clocked by LB_AC_C_S2Dn_CA53_CPU0_STGNO_2ZCLK_T0)
Endpoint: u_ca53_cpu/u_ca53_noram/u_ca53ifu/u_ca53ifu_pd/u_ca53ifu_pd debug dec_a64 undef_pdl_reg					(rising edge-triggered flip-flop clocked by LB_AC_C_S2Dn_CA53_CPU0_STGNO_2ZCLK_T0)
Mode: DFT					
Corner: MAXLT_typ_setup					
Scenario: DFT:MAXLT_typ_setup					
Path Group: REG2REG					
Path Type: max					
Attributes					
b - black-box (unknown)					
s - size_only					
d - dont_touch					
u - dont_use					
g - generic					
h - hierarchical					
i - ideal					
n - noncombinational					
E - extracted timing model					
Q - Quick timing model					
Point	Fanout	Cap	Trans	Incr	Path
clock LB_AC_C_S2Dn_CA53_CPU0_STGNO_2ZCLK_T0 (rise edge)				0.0000	0.0000
clock network delay (ideal)				0.0000	0.0000
u_ca53_cpu/u_ca53_noram/u_ca53dpu/u_dpu_dbg/editr_reg_18_	0.0000	0.0000	0.0000	0.0000	/CP (SDFQCK1D4BWP40P140HVT)
r u_ca53_cpu/u_ca53_noram/u_ca53dpu/u_dpu_dbg/editr_reg_18_	0.0731	0.1120	0.1120	0.1120	/Q (SDFQCK1D4BWP40P140HVT)
f u_ca53_cpu/u_ca53_noram/u_ca53dpu/u_dpu_dbg/dbg/dbg_ins_o[18] (net)	12	0.0251	0.0763	0.0092	0.1212
f u_ca53_cpu/u_ca53_noram/u_ca53ifu/u_ca53ifu_pd/U12436/A2 (NR2D1BWP40P140UHVT)					
r u_ca53_cpu/u_ca53_noram/u_ca53iflu/u_ca53iflu_pd/U12436/2N (NR2D1BWP40P140UHVT)			0.0809	0.0950	0.2162
r u_ca53_cpu/u_ca53_noram/u_ca53iflu/u_ca53iflu_pd/n6702 (net)	3	0.0029	0.0809	0.0000	0.2162
r u_ca53_cpu/u_ca53_noram/u_ca53iflu/u_ca53iflu_pd/U13237/A1 (OAI21D0P7BWP40P140UHVT)			0.0489	0.0705	0.2867
f u_ca53_cpu/u_ca53_noram/u_ca53iflu/u_ca53iflu_pd/U13237/2N (OAI21D0P7BWP40P140UHVT)			0.0489	0.0000	0.2867
f u_ca53_cpu/u_ca53_noram/u_ca53iflu/u_ca53iflu_pd/n6703 (net)	1	0.0008	0.0489	0.0000	0.2867
f u_ca53_cpu/u_ca53_noram/u_ca53iflu/u_ca53iflu_pd/U13238/A1 (ND2D0BWP40P140UHVT)			0.0489	0.0000	0.2867
f u_ca53_cpu/u_ca53_noram/u_ca53iflu/u_ca53iflu_pd/U13238/2N (ND2D0BWP40P140UHVT)			0.0396	0.0434	0.3302
r u_ca53_cpu/u_ca53_noram/u_ca53iflu/u_ca53iflu_pd/n6705 (net)	1	0.0006	0.0396	0.0000	0.3302
r u_ca53_cpu/u_ca53_noram/u_ca53iflu/u_ca53iflu_pd/U13241/A1 (ND2D0BWP40P140UHVT)			0.0396	0.0716	0.4018
f u_ca53_cpu/u_ca53_noram/u_ca53iflu/u_ca53iflu_pd/U13241/2N (ND2D0BWP40P140UHVT)			0.0778	0.0716	0.4018
f u_ca53_cpu/u_ca53_noram/u_ca53iflu/u_ca53iflu_pd/n6706 (net)	1	0.0017	0.0778	0.0000	0.4018
f u_ca53_cpu/u_ca53_noram/u_ca53iflu/u_ca53iflu_pd/U13242/B (AOI13ID1BWP40P140UHVT)			0.0552	0.0919	0.4938
r u_ca53_cpu/u_ca53_noram/u_ca53iflu/u_ca53iflu_pd/U13242/2N (AOI13ID1BWP40P140UHVT)			0.0552	0.0919	0.4938
r u_ca53_cpu/u_ca53_noram/u_ca53iflu/u_ca53iflu_pd/n6718 (net)	1	0.0011	0.0552	0.0000	0.4938
r u_ca53_cpu/u_ca53_noram/u_ca53iflu/u_ca53iflu_pd/U13250/A1 (OAI21D0P7BWP40P140UHVT)			0.0552	0.0798	0.5735
r u_ca53_cpu/u_ca53_noram/u_ca53iflu/u_ca53iflu_pd/U13250/2N (OAI21D0P7BWP40P140UHVT)			0.0814	0.0798	0.5735
f u_ca53_cpu/u_ca53_noram/u_ca53iflu/u_ca53iflu_pd/n6733 (net)	1	0.0021	0.0814	0.0001	0.5736
f u_ca53_cpu/u_ca53_noram/u_ca53iflu/u_ca53iflu_pd/U13266/A1 (AOI21D1BWP40P140)			0.0814	0.0426	0.6318
r u_ca53_cpu/u_ca53_noram/u_ca53iflu/u_ca53iflu_pd/U13266/2N (AOI21D1BWP40P140)			0.0426	0.0582	0.6318
f u_ca53_cpu/u_ca53_noram/u_ca53iflu/u_ca53iflu_pd/n6770 (net)	1	0.0016	0.0426	0.0000	0.6318
r u_ca53_cpu/u_ca53_noram/u_ca53iflu/u_ca53iflu_pd/U158/A2 (ND3D0P7BWP40P140LVT)			0.0426	0.1185	0.7503
f u_ca53_cpu/u_ca53_noram/u_ca53iflu/u_ca53iflu_pd/U158/ZN (ND3D0P7BWP40P140LVT)			0.1827	0.1185	0.7503
f u_ca53_cpu/u_ca53_noram/u_ca53iflu/u_ca53iflu_pd/n33091 (net)	1	0.0101	0.1825	0.0016	0.7519
f u_ca53_cpu/u_ca53_noram/u_ca53iflu/u_ca53iflu_pd/u_ca53ifu_reg/D (SDFQCK1D4BWP40P140LVT)			0.1825	0.7519	
data arrival time					0.7519
clock LB_AC_C_S2Dn_CA53_CPU0_STGNO_2ZCLK_T0 (rise edge)				0.9990	0.9990
clock network delay (ideal)				0.0000	0.9990
u_ca53_cpu/u_ca53_noram/u_ca53ifu/u_ca53ifu_pd/u_ca53ifu_reg/CP (SDFQCK1D4BWP40P140LVT)	0.0000	0.0000	0.0000	0.9990	
r clock uncertainty				-0.1200	0.8790
library setup time				-0.1274	0.7516
data required time				0.7516	0.7516
---				-0.7519	
---				-0.7519	
---				-0.0003	

Setup timing path is show the way how to data transfer between start point and end point. It means the data start transfer from the start point when the start clock signal active to the endpoint, the endpoint will accept the data when the end clock signal active.

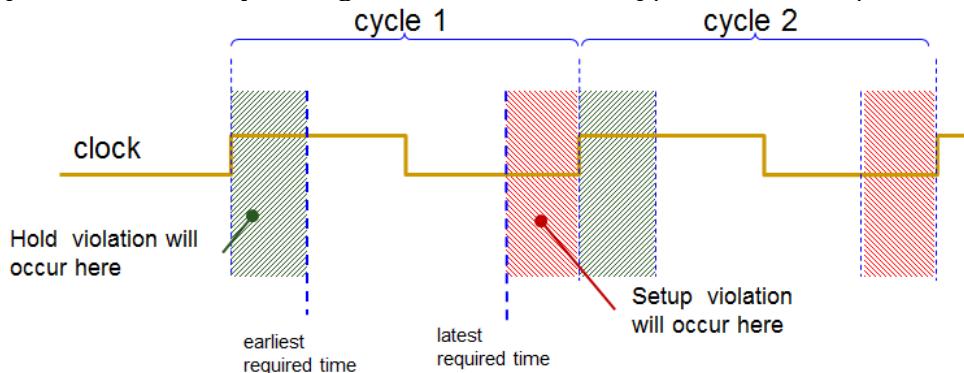


Required time specifies the time point (interval) at which data is required to arrive at end point (data is required to be stable after arrival).

Time point after which data can become unstable (change) is called earliest required time.

Time point after which data cannot become unstable (change) is called latest required time.

The requirement is set by timing constraints like setup/hold, removal/recovery, etc.



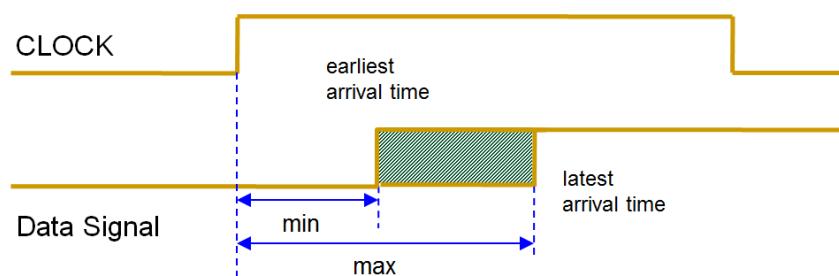
Arrival time defines the time interval during which a data signal will arrive at a path endpoint (after arrival time signal will be stable).

Data is normally triggered by clock edge.

Data arrival depend on circuit delay, which vary (depend on temperature, supply voltage, etc.).

Minimum delay, early arrival.

Maximum delay, late arrival.

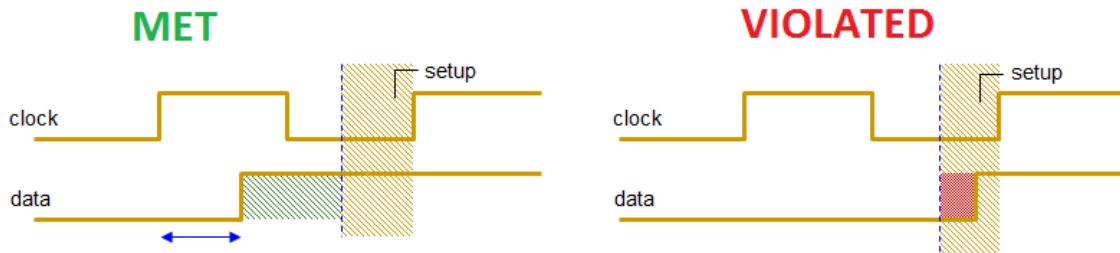


Slack is the difference between the required time and the arrival time.

Negative slack → violation.

Positive slack → constraint have been met.

Critical path is a path in design that has biggest slack if it real.



Ideal mode: the mode before clock tree is built & no clock is propagated in the circuit. In ideal mode, the report show zero clock skew. The skew need to have some assumptions in somewhere like clock uncertainty (Refer to Q25: Uncertainty) and the fixed value for clock transition. In some case, consider to apply the useful skew in ideal mode.

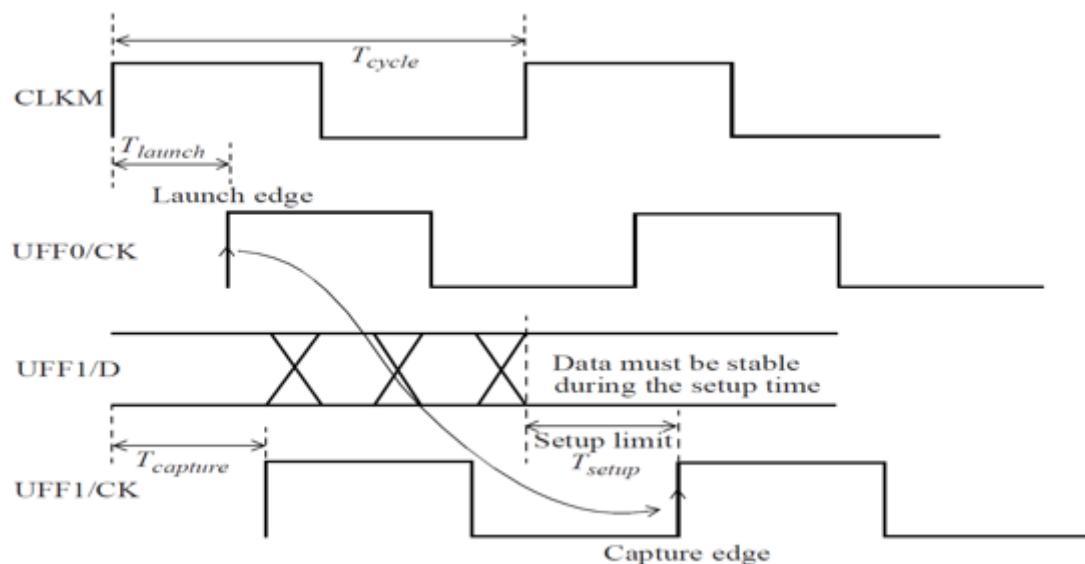
Propagated mode: the mode after the clock is propagated. The latency of real clock path is calculated. Hence, clock skew, clock latency, clock transition will be considered in timing check and optimization.

64 What is timing driven & Congestion driven placement?

- ❖ **Timing-Driven Placement:** Timing-driven placement tries to place cells along timing critical paths close together to reduce net RCs and meet setup timing.
- ❖ **Congestion-driven placement:** Congestion-driven placement runs the routing estimation, which creates congestion maps to find the congestion hot spots. Placement is then driven by the congestion map to remove hot spots.

52 What is setup/hold time? How to fix these violation

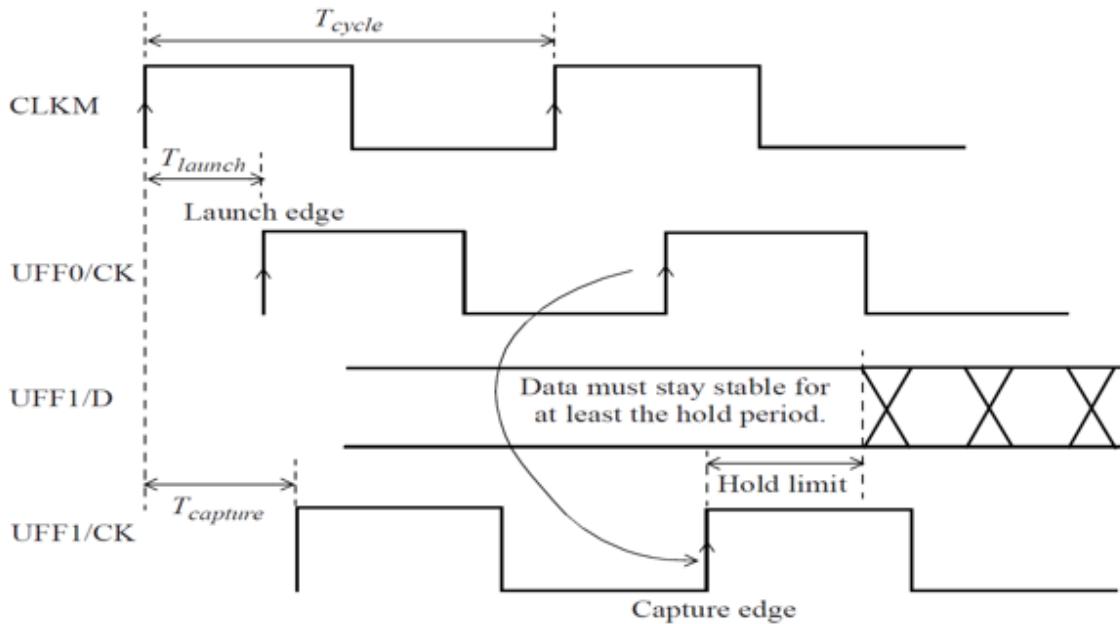
Setup time: Latest time before arrival of capture clock edge (flip flop) or de-asserting of enable line (latch), that input data is required to be stable in order for storage device to work correctly. If setup time is violated, desired input data will not be stored; input data from previous clock cycle might remain stored.



To fix setup violation, we have 2 methods:

- ❖ Optimize data path: reduce the arrival time by some ways:
 - Increase drive-ability of instance: by changing the master cell type of that instance to the bigger drive-ability one.
 - Change threshold voltage (Vth): by changing the master cell type of that instance to the lower threshold voltage type.
 - Promote the layer of routing metal (i.e. from fine layers to semi-global layers), purpose is to reduce the Resistance of net then will reduce the net delay.
 - Improve net delay caused by long wire or many loads (fanout) by using buffers (repeaters).
 - By-pass some redundant instances on data path which not effect to logic function (like buffer, inverter-pair) to reduce total arrival delay.
 - Any other methods which help to reduce the data delay, ...
- ❖ Optimize clock path: called useful-skew method.
 - By touching to the clock line to optimize the total timing, with the same purpose that reduce the total arrival time or sometimes increase the required time.
 - To reduce the arrival time on the clock line, we also need to increase the drive-ability of the clock drivers, or something else same as data path (in ideal mode, you need to use "set_clock_latency" command to set a virtual estimated clock value for this intention).
 - To increase the required time, you need to add more delay into the end-point FF clock line by using clock buffer or inverter. (In ideal mode, you need to use "set_clock_latency" command to set a virtual estimated clock value for this intention).
 - You need to aware that using useful skew will easily cause side effect to the other timing path related to the same clock line as we touched.

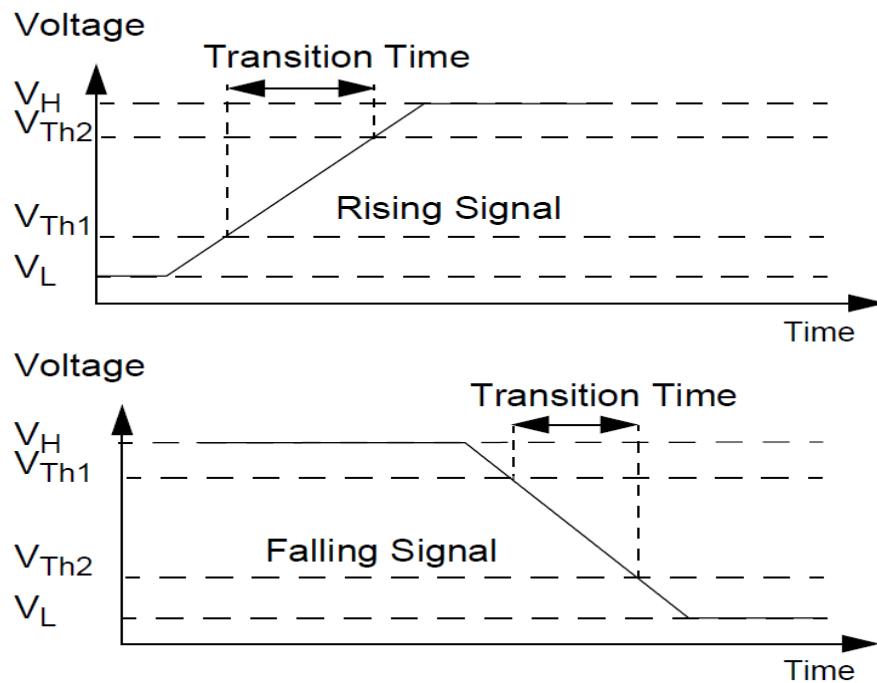
Hold time: Latest time after arrival of capture clock edge (flip flop) or de-asserting of enable line (latch) that input data is required to remain stable in order for storage device to work correctly. If hold time is violated, desired input data will not be stored; input data from next clock cycle might slip through and be stored.



To fix hold time, same as setup we will have 2 methods:

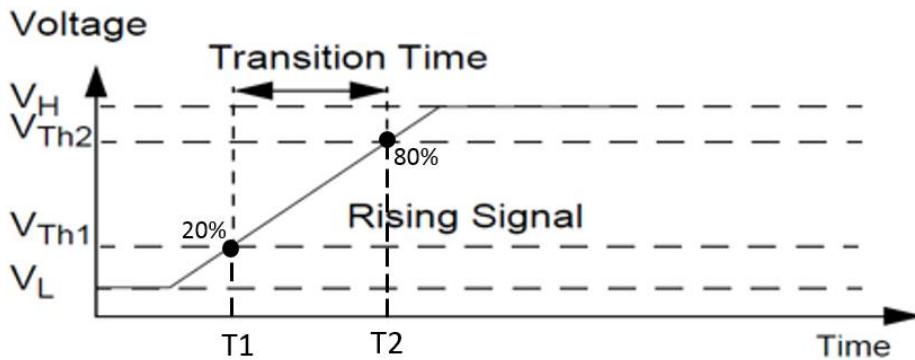
- ❖ Optimize data path: by increase the delay of the arrival time
 - Insert more delay in the data path by using buffers, inverters ... which will not cause logic function change.
 - Increase wire delay by drawing longer wire into the current wire (like detour path) or downgrade layer to lower one to increase the Resistance. BTW, please note about the side effect of this method like transition/capacitance/...
 - Other methods that can increase the data path delay (downsize, change to higher V_{th} , ...)
- ❖ Optimize clock path: opposite with the way of fix setup timing, you need to increase the clock latency of started point FF, and reduce the latency of end point FFs by using the same hint of how to optimize like above for setup. Please note that this way also causes side effect same as fix setup way, so you need to be very careful.

53 Transition time? How to fix transition violation



Transition time: the time needed to transition from one logic state to the other (falling change and rising change). Based on specific requirements, we can get values by measuring the transition time from 80% to 20% or 90% to 10% or 70% to 30% of high level supply voltage (VDD/VCC).

Example:



Transition time is measured at 2 points of time ($T_1(V_{Th1})$ & $T_2(V_{Th2})$)

$$\rightarrow T_{\text{transition}} = |T_1(V_{Th1}) - T_2(V_{Th2})|$$

While: $V_{Th1} = 20\%V_L$, $V_{Th2} = 80\%V_H$.

Ref: <\\rvc-vnas-01\\Backend\\COMMON\\DOCUMENT\\Terminology\\Reference\\Static-Timing-Analysis-for-Nanometer-Designs.pdf>

Fix transition time:

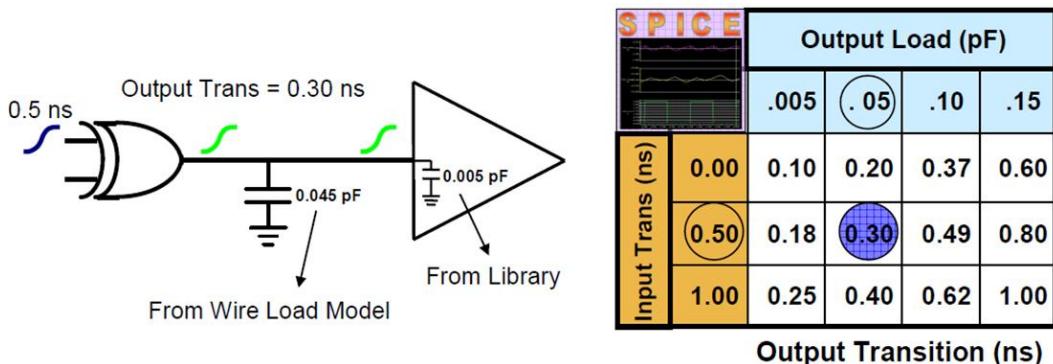
- Weak driver: -> Increase drive-ability of driver
- Long wire: -> Separate wire by repeaters

- Many fan-out: -> Divide fan-out.

54 How to calculate the transition time

CMOS delay model: Transition Time = $f(R,C)$.

Transition time from table lookup and interpolation/extrapolation.



91 Explain DRC?

Please refer to document: [PhysicalDesign_Basic_Knowledge_v1r1s2.pdf](#)

92 Explain LVS?

Please refer to document: [PhysicalDesign_Basic_Knowledge_v1r1s2.pdf](#)

93 Explain Antenna?

Please refer to document: [PhysicalDesign_Basic_Knowledge_v1r1s2.pdf](#)

94 Explain Density?

Please refer to document: [PhysicalDesign_Basic_Knowledge_v1r1s2.pdf](#)

95. What is GDSII format? Why we need GDS merge? (Trung-san-Add to PV doc)

GDS stands for Graphic Database System.

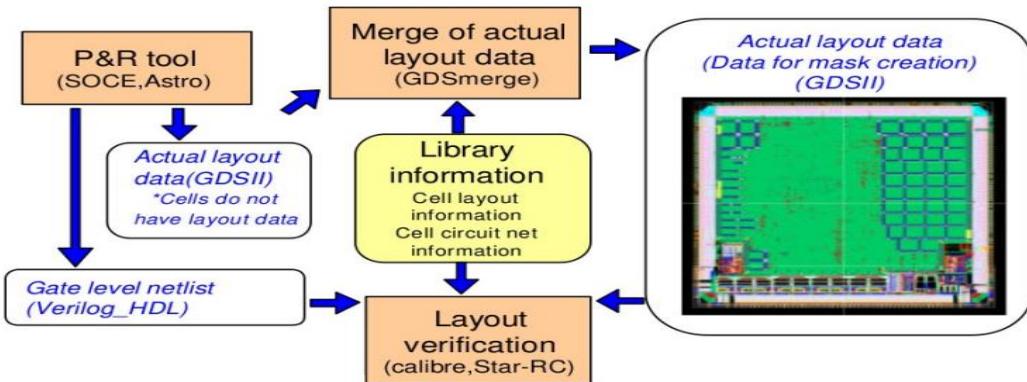
GDSII stream format, common acronym **GDSII**, is a **database file format** which is the de facto industry standard **for data exchange** of integrated circuit or IC layout artwork (design). It is a **binary file format** representing planar geometric shapes, text labels, and other information about the layout in hierarchical form. The data can be used to reconstruct all or part of the artwork to be used in **sharing layouts, transferring artwork between different tools, or creating photomasks**. *<source: en.wikipedia.org>*

What is GDS-merge?

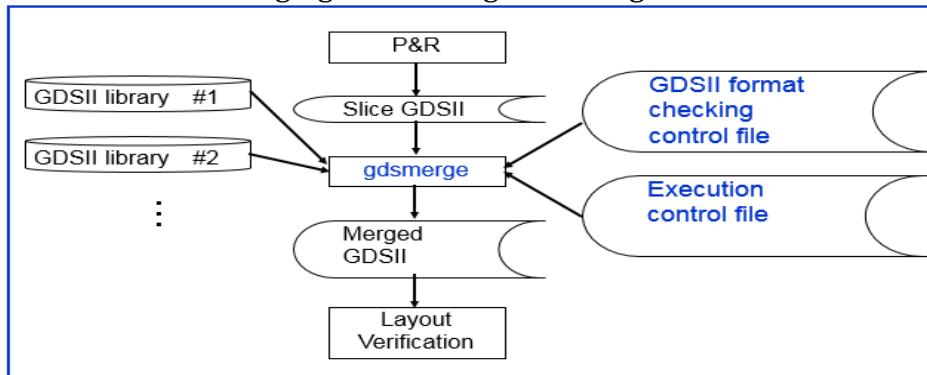
A well-known layout editor tool by Cadence, VIRTUOSO, is often used to merge cell layout patterns and block/chip layout patterns. However, layout size of hierarchical blocks and full-chip is huge because of huge number leaf cells and interconnection patterns plus other physical patterns. So it will take very long time for VIRTUOSO to merge the whole hierarchical design with all leaf cells. In addition, some of our big designs have reached the limitation that VIRTUOSO can handle.

GDS-merge is an in-house tool of Renesas used to merge GDSII files of hierarchical blocks and leaf cells to generate a full “top-to-leaf” layout design. It can handle very complicated and big size hierarchical designs.

Below is a typical flow of merging GDS and verifying the GDS after merging:



Below is a basic flow of merging GDSII using GDS-merge tool:



101. What is process corners: ff/ss/fs/sf/tt?

- 102 What is NDR?**
- 103 What is ATOM and related logic: JTAG, ATAP, CPG...?**
- 104. Basic algorithm of CTS: What the tool do? How many stages? ...**
- 105 What is difference between the design with mini CPG/mini ATOM (Rcar Gen3) and without mini CPG/mini ATOM (Rcar Gen2 & Before)? What is the adv/disadv of using mini CPG/mini ATOM**
- 106. What is EM?**
- 107. What is IR drop?**
- 108. Bump, pad?**
- 109. Package? Single/double/triple scribe? Seal-ring?**
- 110. Wire bonding, RDL**
- 111. Shield-ring?**
- 112. NBTI? TDDB?**
- 113. SMVA?**
- 114. What is input to build up a Chip?**
- 115. Timing noise?**
- 116. Double switching?**
- 117. Glitch?**
- 118. Power control by Clock gating?**
- 119. ESD?**
- 120. Wire load model?**
- 121. Always ON domain? OFF domain? Operation of PSW, ISO cell?**
- 122. DVFS? Level shifter?**
- 123. DF/CF, DN/CN?**
- 124. Asynchronous, synchronous design?**
- 125. What is End-CAP/Well-TAP?**
- 126.**
- 127. What is End-CAP/Well-TAP?**
1. What is the real effect of IRdrop/EM/Latchup/... in actual chip?
 2. What is the actual performance when we change std drivability? std Vth?
 3. What is the different of CMOS and FINFET?

4. What is ENDCAP/WELLTAP/BOUNDARY/GuardRing/... cells?
- 5.