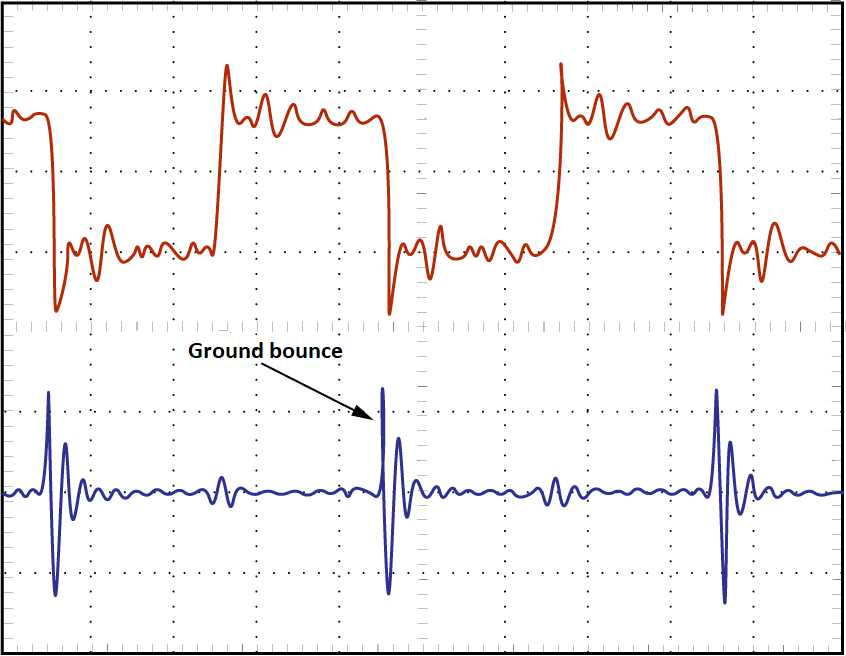
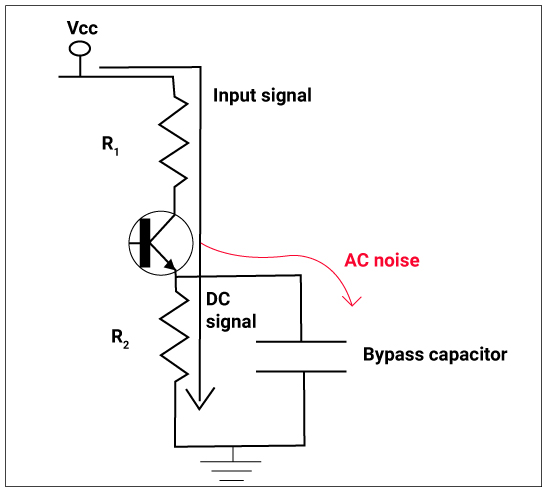
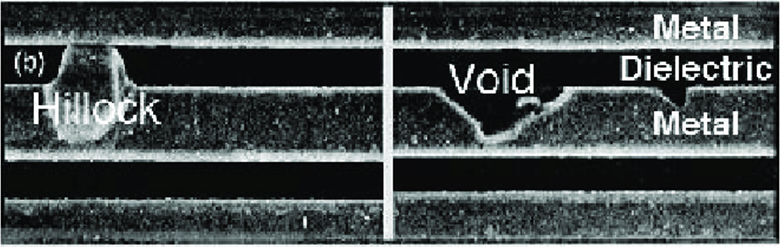
# Signal Integrity

* Concept: Signal integrity is the ability of an electrical signal to carry information reliably and resist the effects of high-frequency electromagnetic interference from nearby signals
* Turn on SI check if SI option in sign-off have YES, command option -ptsi
* Signal Integrity may be affected by various reasons, but major reasons which make chip failure are:
  1. Crosstalk (Delay and Noise )
     + Concept : Crosstalk is the undesirable electrical interaction between two or more physically adjacent nets due to capacitive cross-coupling *(Solvnet)*  
       🡪 switching of a signal in one net can interfere in the neighboring net. The affected signal is called the **victim**, and the affecting signals are termed as **aggressors**
     + Crosstalk has two major effects:
       1. ***Crosstalk glitch or crosstalk noise***
          1. Case-1: Aggressor net is switching low to high and victim net is at a constant low
          2. Case-2: Aggressor net is switching high to low and victim net is at a constant high
       2. ***Crosstalk delta delay or crosstalk delay***
          1. Case-3: Aggressor and victim net switch in opposite directions
          2. Case-4: Aggressor and victim nets switch in the same direction
     + Effect of Crosstalk
       1. Unbalance clock tree: increase or decrease the delay of clock buffers in the clock path
       2. Timing:
          - increase of delay in the data path or launch clock path 🡪 setup violation
          - decrease in the delay of any cells in the data path and launch clock or there is an increase of delay of cells in the capture clock path due to crosstalk delay 🡪 hold timing violation
  2. Ground bounce
     + Concept:
     + Effect of Ground bounce: 
     + How to fix:
       1. The easiest way to prevent ground bounce is to use a bypass capacitor to provide local charge storage for the transistor. The bypass capacitor prevents voltage from building up between local ground and true ground. It can also be helpful to reduce inductance between ground and true ground by improving the connection with multiple ground pins.
       2. Use multiple ground pins to split the current demand across multiple paths and thereby reduce the voltage drop across any one path. This is one reason that high-pin-count ICs are designed with lots and lots of ground pins.
  3. IR Drop
     + Concept: IR Drop in VLSI is known as an “**Intermediate Resistance Drop**” at “**Very Large Scale Integration**”. It refers to the variation in electrical potential between the two ends of a conducting wire when current flows through it. This potential difference is determined by the voltage drop across a resistance, which can be calculated by multiplying the current (I) passing through the resistance by its resistance value (R).
     + The IR Drop have two type:
       1. Static IR Drop
       2. Dynamic IR Drop
     + Effect of IR Drop
       1. Decreased performance.
       2. Increased Power Consumption
  4. Antenna effect
     + Concept: The antenna effect in the VLSI design course, also known as plasma-induced gate-oxide damage or plasma-induced damage, occurs when unwanted charges accumulate on exposed conductors during certain fabrication processes, like plasma etching.
  5. Electromigration
     + Concept: When a high current density passes through a metal interconnect, the momentum of current-carrying electrons may get transferred to the metal ions during the collision between them. Due to the momentum transfer, the metal ions may get drifted in the direction of motion of electrons. Such drift of metal ions from its original position is called the electromigration effect.
     + Effect of EM
       - Once the metal ions get started to shifting from its original position, these will create problems in the interconnect. It could result in an excess of ions accumulation in a particular location of deficient of ions. So either Hillocks or Void could occur in the metal interconnect.
         1. Void: If the incoming ion flux is lesser than the outgoing ion flux, It will create a void in interconnect. A void can lead a discontinuity in the interconnect and result an open circuit.
         2. Hillocks: If incoming ion flux is greater than the outgoing ion flux, It will cause the accumulation of ions and create a hillock in the interconnect. A hillock can increase the width of a metal interconnect and touch the neighbouring metal interconnect which may result in a short circuit.



* + - The following techniques could be used to prevent the EM issue.
      1. Increase the metal width to reduce the current density (PI )
      2. Reduce the frequency
      3. Lower the supply voltage
      4. Keep the wire length sort
      5. Reduce the buffer size in clock lines

# Uncertainty

* Concept:
  1. The clock uncertainty is the margin given to the clock so that ur setup or the hold window is changed. since u need to subtract/plus this uncertainty value.
  2. Clock uncertainty is used to model some factors (margin skew, jitter, … ) which could affect the clock period
     + Clock jitter: refers to the temporary change of the clock cycle at a given point of chip, which my lengthen or shorten the clock cycyle at different cycles. Jitter is generated inside the clock generator and is related to the crystal oscillator or PLL internal circuit
     + Skew: Caused by different wiring lengths and loads, resulting in the same clock signal arriving at two adjacent timing units at different times
* Command:
  1. set\_clock\_uncertainty –setup 1.5 [get\_clocks CLK]
  2. set\_clock\_uncertainty –hold 1 [get\_clocks CLK]
* Effect:
  1. In setup timing check, clock uncertainty is defined as minus (-)
  2. In hold timing check, clock uncertainty is defined as plus (+)
* Setting:
  1. STA\_USERSPECIFIED\_MARGIN = ON

🡪 To use **user uncertainty** value (value set in SDCs)

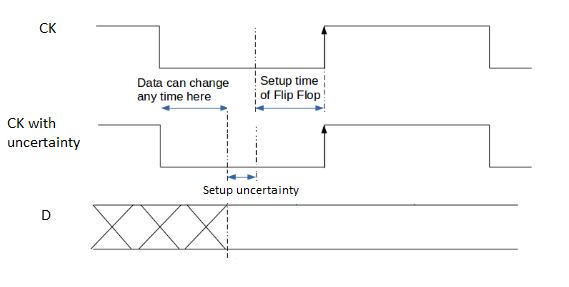
* 1. STA\_USERSPECIFIED\_MARGIN = OFF

🡪 To use **sign-off uncertainty** value

**Question: How to check the value uncertainty in fstaH env?**

Answer:

* 1. We can check in **out.log.gz** file of DFT.sdc timing report (shift, capture, OCC)
  2. in HoldChk\_reg2reg.rpt/HoldChk\_in2reg.rpt/HoldChk\_reg2out.rpt file



**Question: How clock uncertainty effect ?**



High pulse width = half pulse width of clock signal – (rise delay – fall delay)

 = 0.5 - (0.055 - 0.048) - (0.039 - 0.032) - (0.025 - 0.022) - (0.048 - 0.043) - (0.058 - 0.054) = 0.474ns

Low pulse width = half pulse width of clock signal + (rise delay – fall delay) =0.526

Assume required value of Min pulse width is 0.410ns, Uncertainty = 90ps = 0.09ns

Then high pulse width = 0.474 - 0.090 = 0.384ns

The slack is 0.384 - 0.410 = - 0.026ns

**Question: If we have 2 uncertainty value, user uncertainty and sign-off uncertainty, which one should be used?**

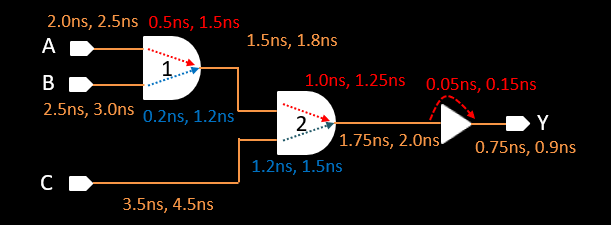
* Function SDC: follow user value.
* DFT SDC:
* Setup: follow user value.
* Hold: follow sign-off value.

**Question: Why SDC *shift* and *capture* only perform hold uncertainty ?**

# GBA (Graph base Analysis) –PBA (Path base Analysis)

PBA (path base analysis): the timing engine computes the worst case delays of all standard cells assuming the worst case slew for all the inputs of a gate

GBA (graph base analysis): the tools take into account the actual slew for the arcs encoutered while tranversing any particular timing path



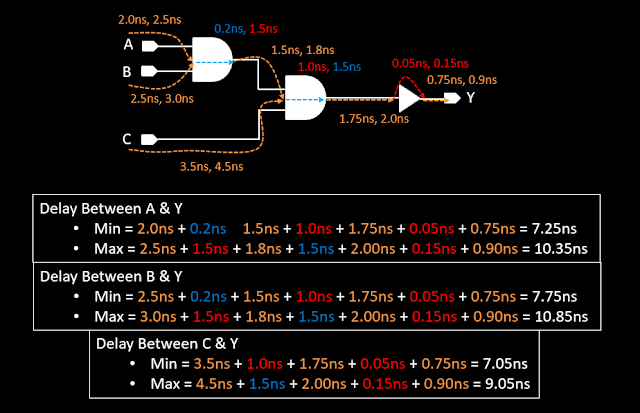
AND gate (1) has 2 input. 2 set of input-output delay combination:

* Min delay = 0.5ns, Max delay = 1.5ns
* Min delay = 0.2 ns, Max delay = 1.2ns

Similarly, for other logic gates

**Delay Calculation in Case of GBA (Graph base Analysis):**

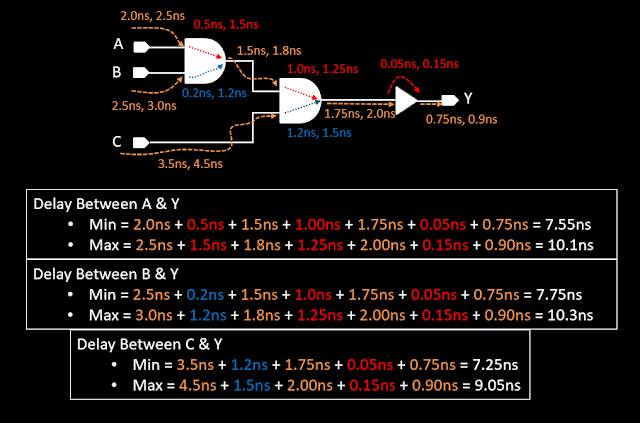
In place of choosing 2 combinations of AND gate (1), i.e (Combination\_1: 0.5ns, 1.5ns ; Combination\_2: 0.2ns, 1.2ns) we choose extreme boundaries, i.e. **min delay = 0.2ns** and **max delay = 1.5ns**



**Delay Calculation in Case of PBA (Path base Analysis):**

We are using actual delay between input pin and output pin combination (means choosing both combination of delay):

* Combination\_1: 0.5ns, 1.5ns
* Combination\_2: 0.2ns, 1.2ns



**Compare GBA & PBA result:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Path** | **Min (ns)** |  | **Max (ns)** |  |
| **GBA** | **PBA** | **GBA** | **PBA** |
| Delay Between A & Y | 7.25 | 7.55 | 10.35 | 10.1 |
| Delay Between B & Y | 7.75 | 7.75 | 10.85 | 10.3 |
| Delay Between  C & Y | 7.05 | 7.25 | 9.05 | 9.05 |

We can see: **Min\_delay\_GBA <= min\_delay\_PBA**

**Max\_delay\_GBA >= max\_delay\_PBA**

|  |  |  |
| --- | --- | --- |
| **Method** | **Advantage** | **Disadvantage** |
| Graph base analysis | Fast running time | More pessimism |
| Path base analysis | More accurate | Long running time |

In summary, we can see that the GBA-based timing calculation method can be pessimistic. Whereas, timing in PBA mode is more relistic. In this case, it is not very good to calculate the timing directly with PBA. Because the calculation method of PBA mode timing is relatively complex, the runtime will take a long time. In large-scale design, if we use PBA mode for timing signoff, it wil inevitably cause the run time to increase exponentially. By default, the timing is calculated in GBA. GBA is simple to calculate and has a fast runtime, but it is relatively pessimistic. If you find that there is a setup violation of about 5ps in the final stage of Timing signoff, and it has been optimized to the extreme (the cell cannot be upsized, and the clock tree has no possibility of doing manual eco). At this point, we can report whether the setup can meet in PBA mode.

# OCV – AOCV - POCV

On Chip Variation

**Concept:**

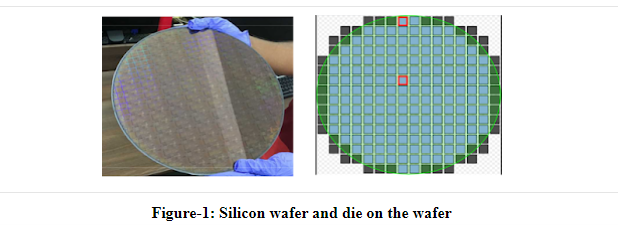
+A wafer includes many dies.

+Each die contains an individual Integrated Circuit.

+The characteristics of all the ICs are not the same.

+The transistors inside a single IC are not similar.

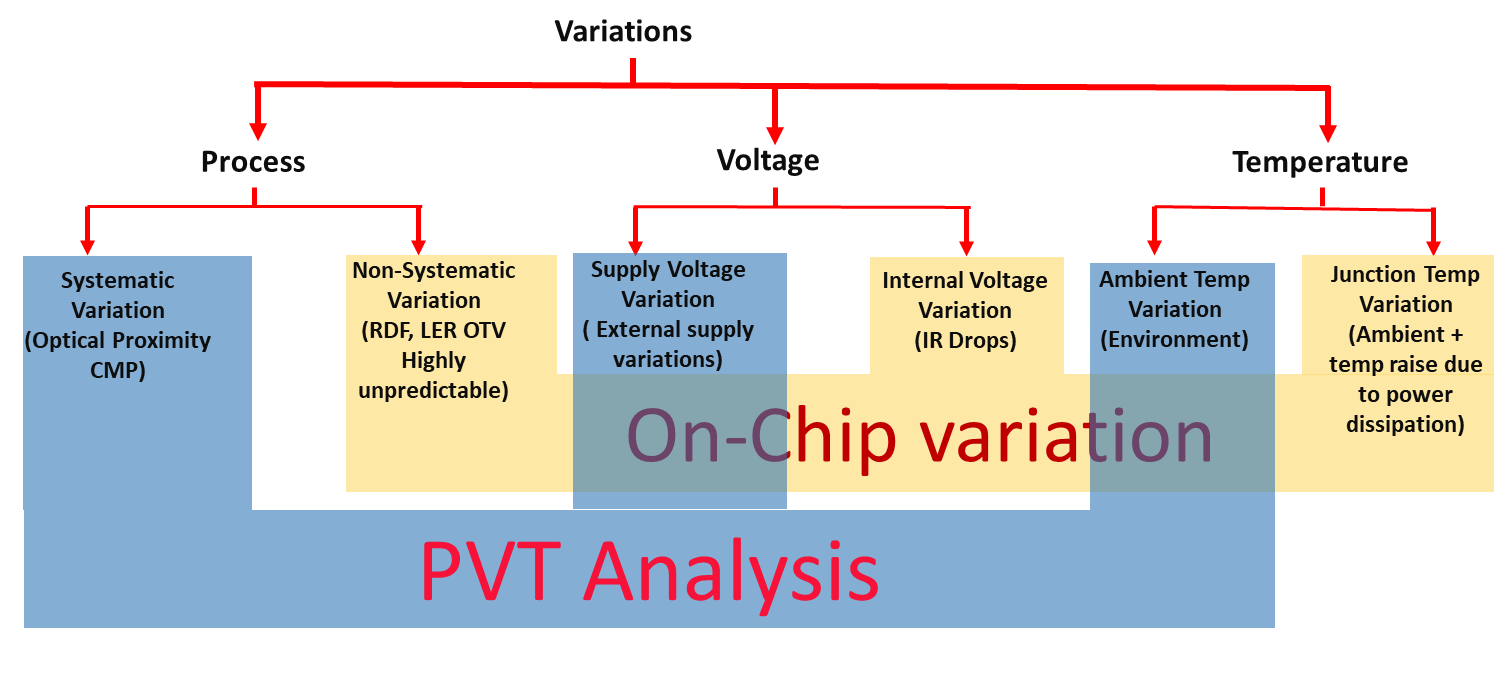
+The root cause of these variations is the fabrication process.



There are three major sources of variations: P,V,T

But all the variations can not be taken care in PVT analysis and can not be modelled easily

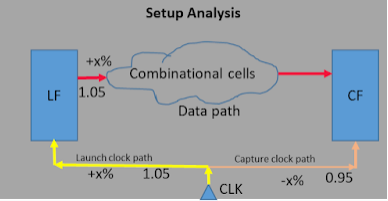
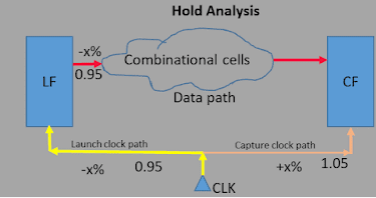
There are two types of variations: **systematic** variation and non-systematic (**random**) variation

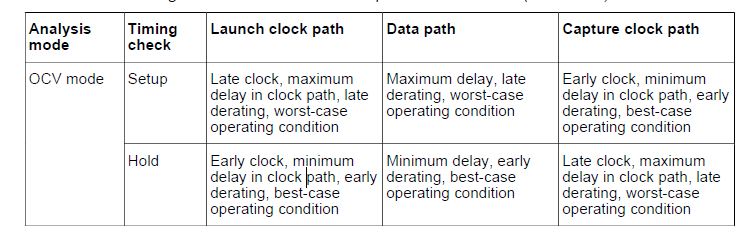
****

* **To take care of OCV we need to add some pessimism in the timing of standard cells. We basically apply ±x% of additional delay to all the standard cells. Which is called OCV derate.**

## OCV

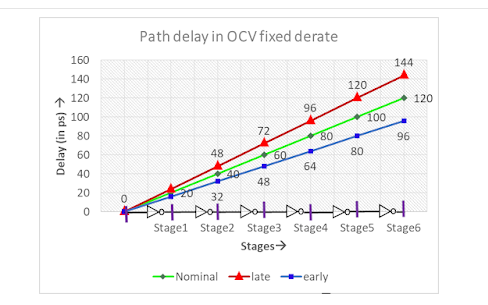
**Concept:** In OCV a fixed timing derate factor is applied to the delay of all the cells. Fab process variations could either increase or decrease the delay of cells. Tool considers early or lately derate based on the path and the type of analysis.





**Issues in OCV:**

Fixed timing derate used for all the cells in the OCV is over pessimistic. In reality, there is the cancellation of random variation effect. All the cells in a particular path could not be delayed all or early all. There is a mixed type of effect always and this causes cancellation of effect in total.

****

## **AOCV**

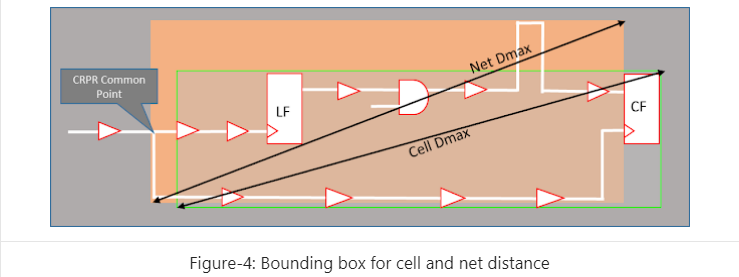
**Concept:** In AOCV (Advanced OCV) derate is applied on each cell based on **path depth** and **distance of the cell** in the timing path and it also varies with **cell type** and **drive strength of the cell**. The **depth** of the cell models the **random variation** component, while **distance** (location) of the cell in the path models **systematic variation** of the cells.

AOCV supports two analysis modes:

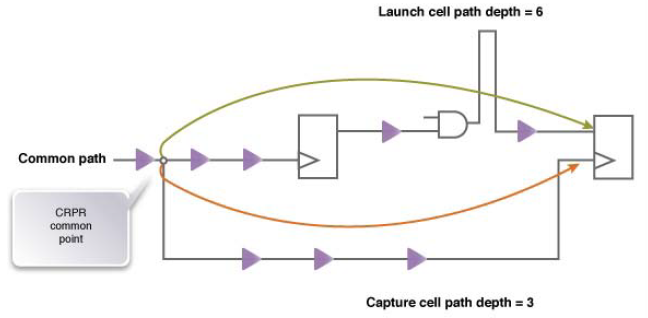
1. **Clock only**

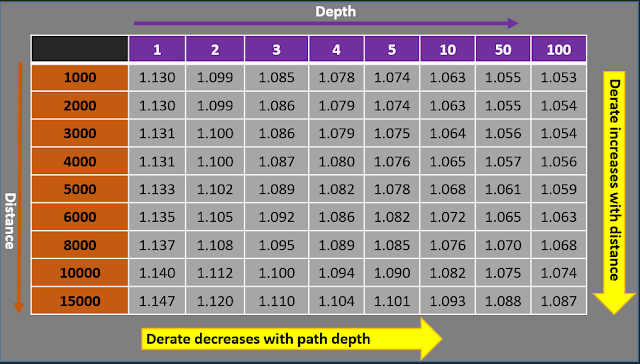
2. **Clock and data**

Distance: is define by a bounding box for net and cell:



Path depth:





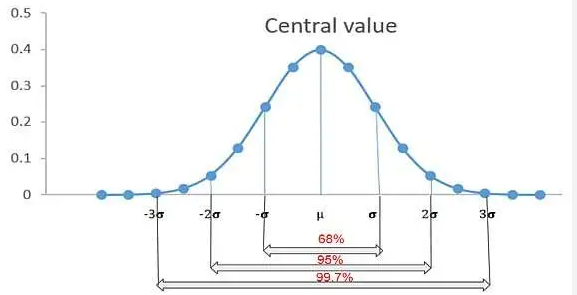
**Distance**: If the distance increases, systematic variation would increase and to mitigate the variation, we need to use higher derate value. So along with the distance, derate value increases.

**Path depth**: In the case of distance is fixed and path depth increases, systematic variation would be constant but the random variation would tend to cancel each other. Therefore as path depth increases the derate factor would decrease. Figure-5, illustrate the path depth in the timing path.

## POCV

**Concept:** In POCV, instead of applying a specific derating factor to a cell, cell delay is calculated based on a delay variation of that cell. The variation value (σ) is a unique value specific to that library cell.

For POCV, it is assumed that the nominal delay (expected delay) value of a cell follows a normal distribution curve. The delay value of the cell is calculated after many experiments and their mean is taken as the nominal delay.

****

There are 3 types of POCV:

+Single-Parameter Random Variation: POCV can model random variation in the timing behavior of each cell instance as a single parameter of that instance.

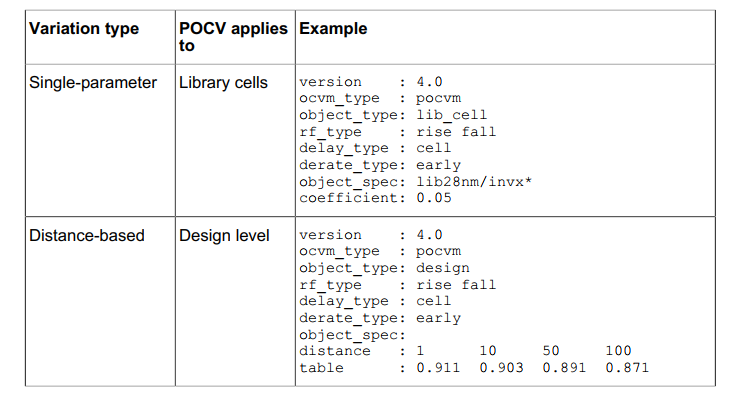
+Distance-Based Variation: POCV can model systematic variation in a timing path as a function of the distance.

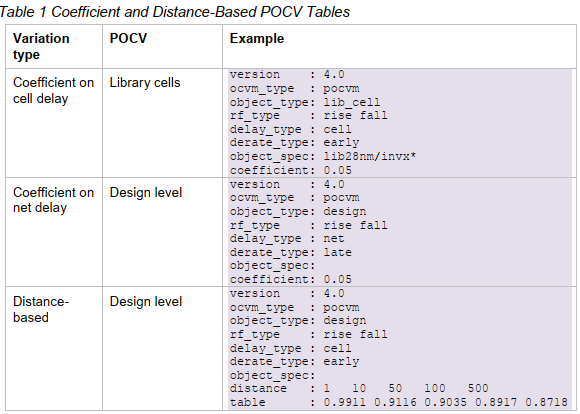
+Mixing Single-Parameter and Distance-Based Variation

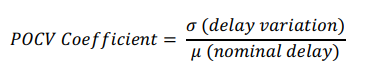
**Input**

To use parametric on-chip variation (POCV), need one of the

**• POCV Single Coefficient Specified in a Side File**







If **using distance-based** derating tables for POCV analysis, the tool **needs coordinates** to calculate the path distance,

If applying variation data from both LVF libraries and a side file, the side file takes precedence

• For single-parameter variation, this behavior can be controlled by the ***timing\_pocvm\_precedence*** variable.

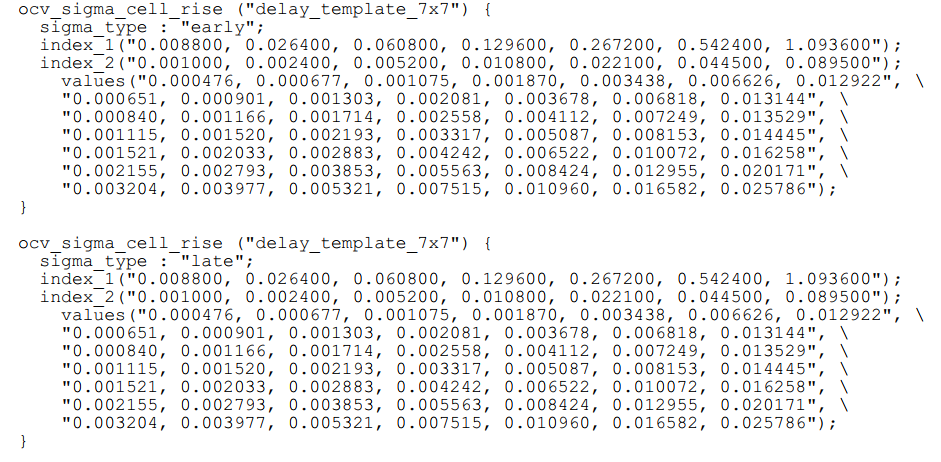
- file: default - POCV side file always takes precedence regardless what level (i.e design or lib\_cell) that the POCV is applied.

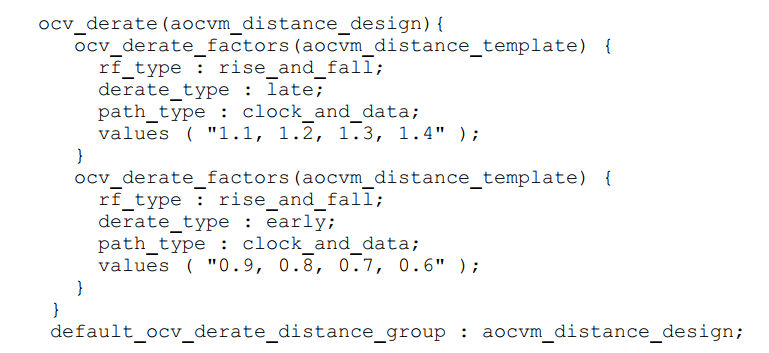
- libr

+LVF when it is applied on the same library cell

• For distance-based variation, the **side file always takes precedence**

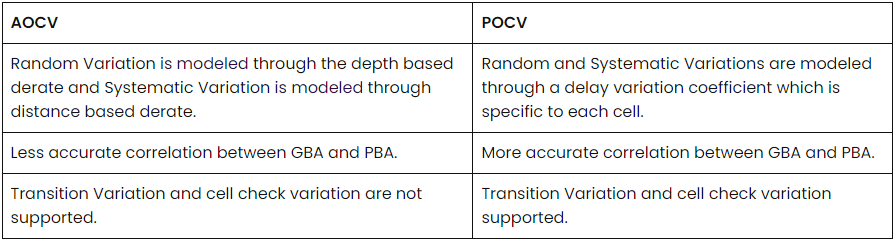
**• POCV Slew-Load Table in Liberty Variation Format (LVF lib)**





A side file with POCV single coefficient applied on a library cell has higher precedence than POCV slew-load table.

Compare AOCV and POCV:



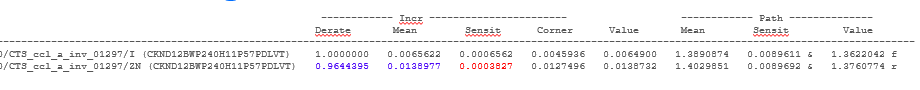
**POCV/VLF in timing report**

**In the Incr columns,**

• The Mean, Sensit, and Corner numbers provide mean, standard deviation, and corner information about each incremental delay distribution by itself, independent of other incremental delay distributions.

Corner = Mean ± K\*Sensit

K: the value of *timing\_pocvm\_report\_sigma*. The default is 3.



• The Incr Value number is not derived from the incremental delay distribution.

>> increment value= current\_path value - previous path value

**In the Path columns,**

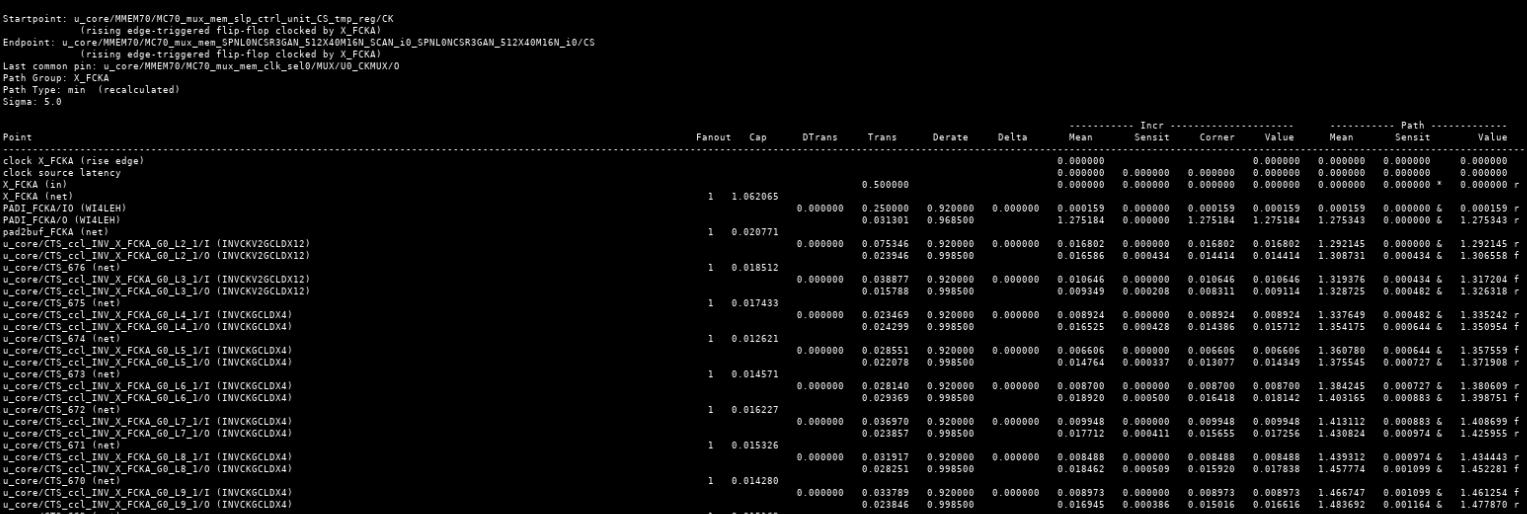
• The Mean, Sensit, and Value numbers provide mean, standard deviation, and corner information about the cumulative arrival distribution at that pin.

“Mean” column, it is simple addition of all mean stage delays up to that point “

Sensit” column, it is simple RSS (Root Sum Square) of all sigma stage delay values up to that point

For example: post-STA

**Timing Report – LVFMBCGH – Cmin125C - Hold**

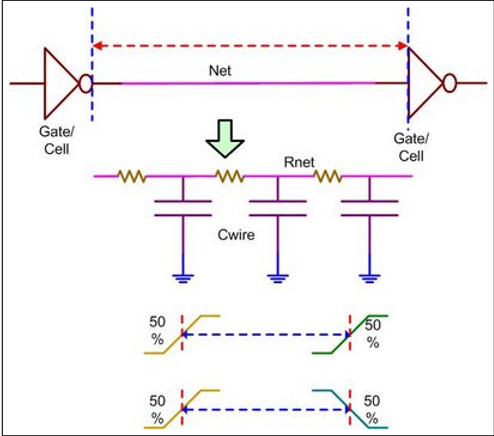
****

**“&” (after increment delay):** the delay number is calculated with RC network back-annotation

**“\*”**: for Standard Delay Format *(SDF) back-annotation*

**“H”** (hybrid annotation): after annotation, this letter will appear after we continue change delay of cell.

“+” ( for lumped RC): From the ouput cell1 -> input cell2, the net delay will model by RC network and lumped RC is 1 type of RC network. So when we change cell 1, this model will effect to delay of Input cell2, the effect will display by “+” in timing report.



**In the Incr columns,**

• The Mean, Sensit, and Corner numbers provide mean, standard deviation, and corner information about each incremental delay distribution by itself, independent of other incremental delay distributions.

**Corner** = 0.016586 - 5\*0.000434 = 0.014416

• The Incr Value number is not derived from the incremental delay distribution.

**increment value** = 1.306558 – 1.292145 = 0.014413

**In the Path columns,**

• The Mean, Sensit, and Value numbers provide mean, standard deviation, and corner information about the cumulative arrival distribution at that pin.

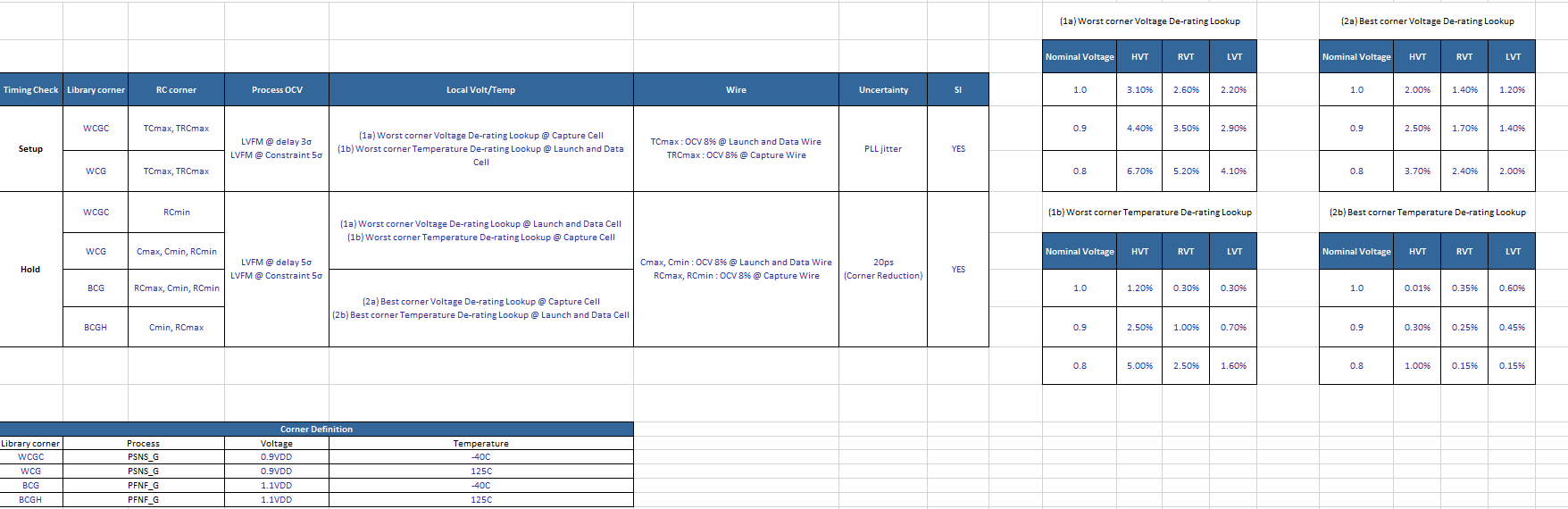
“Mean” column, it is simple addition of all mean stage delays up to that point, i.e.

**Mean** = 0.000159 + 1.275184 + 0.016802 + 0.016586 = 1.308731

“Sensit” column, it is simple RSS (Root Sum Square) of all sigma stage delay values up to that point, i.e. Sensit = sqrt(0.000434^2) = 0.000434

**Value** is equal to 1.308731 – 0.000434\*5 = 1.306561

**Derate =** 1 – 0.15% **= 0.9985**

****

**Dtrans** (delta transition) and **Delta** (delta delay): show the contribution of crosstalk to the delay per stage in the path. The collum labeled Incr (increment) already includes the calculated delta delay.  
An ampersand character (&) in the incr column indicates the presence of parasitic data. (primetime – 512)

# Extract Timing Models

Overview:

Timing model extraction is the process of extracting the interface timing of hierarchical blocks into a timing library. Typically, model extraction has the following advantages:

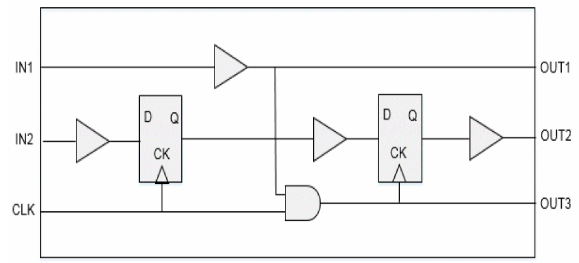
* Reduces the memory requirements by generation of extraction timing models (ETMs) for respective blocks, which may be huge in size.
* Reduces the static timing analysis (STA) run time.
* Hides the proprietary implementation details of the block from a third-party.

ETM Generation:

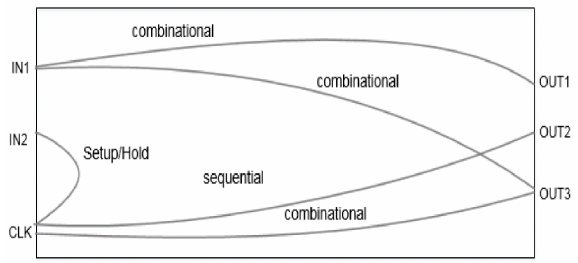
ETM represents the timing for interface paths. Input to flop/gate, input to output and flop/gate to output paths of a design are preserved as timing arcs in the ETM. If there are multiple clocks capturing the data from an input port then an arc with respect to each input port is extracted.

The flop-to-flop type of paths are not written in the ETM, as these do not affect the interface paths timing.

Equivalent ETM for a given netlist:



Extracted Model:



The extracted timing model is context-independent because it gives the correct value of arcs/delays, even with varying values of input transitions, output loads, input delays, or output delays. In such cases, it is not required to re-extract the model if some of the context is changed at a later stage of development.

However, the model depends upon the operating conditions, wire load models, annotated delays/loads, and RC data present on internal nets defined in the original design. So if these elements change at the development stage of design then we need to re-extract the model for correlation with the changed scenario.

**Basic Elements of Timing Model Extraction**

**Nets**

Nets can mainly be classified into two types – boundary nets and internal nets. The nets which are directly connected to the input or output ports of the block are termed as the boundary nets. The nets which drive and are driven by some internal instance pin of the block are termed as the internal nets. Model extraction treats both the nets differently as boundary nets are always related and connected to the context.

Boundary Nets:

The extracted model does not preserve the boundary nets of the original design. Instead, it factors the boundary net delays into the model. The boundaries are connected to the context. So the RC data for them may change if the context of the block changes at a later stage. To be better context independent, we can create the top level SPEF file for boundary nets of this block or use the default behavior to consider the boundary nets RC data for model extraction.

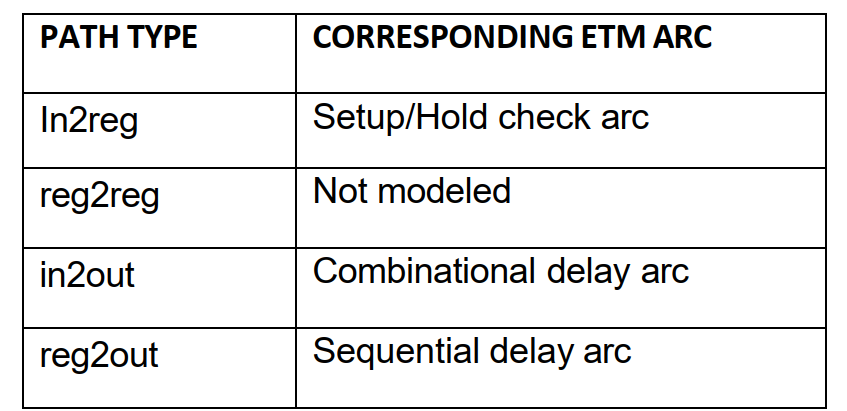
Internal Nets:

As the internal nets connect the pins for the internal instance of the block, they are in no way dependent on the context environment.

If the internal nets are annotated with detailed parasitics, the RC data defined for the internal nets is used as it is for delay calculation and is added in the extracted paths. If no RC is defined for these nets, then the wire load models are used to calculate their delays.

If internal nets are annotated with detailed parasitics, such as Reduced Standard Parasitic Format (RSPF) or SPEF, wire load model isn’t used to calculate the net capacitance and resistance. Instead, it takes the values from the annotated delay instead of the calculated delays.

Timing Paths



Timing paths are broadly divided in four types:

In2reg

Reg2out

In2out

Reg2reg: The reg2reg paths are not relevant to the model extraction process

**In2reg Paths:**

A In2reg path is a setup/hold check that starts from an input port and is captured at a flop or gating element by a clock. So the in2reg type of paths are captured in equivalent setup or hold checks. The setup/hold values to be written in ETM are calculated using the data path delay, the setup/hold value of the library cell and the clock path delay.

Setup arc value = data path delay (input to flop) + setup value of flop – clock path delay (clock source to clk pin)

Hold arc value = data path delay (input to flop) – hold value of flop – clock path delay (clock source to clk pin)

The value for these arcs is the function of the transition on the input port and the transition at the clock source. If a flop is captured by multiple clocks then separate setup/hold arcs are extracted with respect to each clock source.

Reg2out Paths:

The reg2out paths are paths starting from a register and ending up on an output port.

The delay for the arc will be equal to:

Sequential arc delay = delay (clock source to CK of register) + delay (register CK pin to out port).

The delay for these arcs is a function of the slew at the clock source and the capacitance at the output port. As the extracted model can be used for max as well as min analysis, two arcs are preserved in the model to represent the longest and the shortest path.

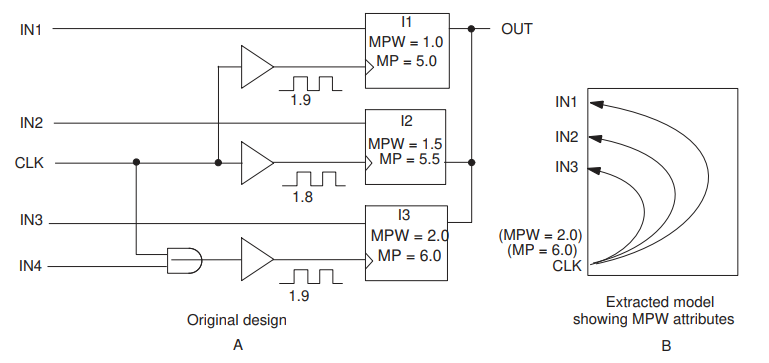
In2Out Paths

The in2out paths are the path starting from an input port and ending up on an output port. These paths are pure combinational paths. So for such paths an equivalent combinational arc is modeled in the extracted model.

The delay for the arc will be equal to:  
Combinational arc delay = delay (delay of all elements in the path)

**Minimim Pulse Width and Period Checks:**

Minimum pulse width and minimum period constraints on cell pins are propagated as attributes on the clock pins in the extracted model. Minimum pulse width and minimum period extraction:



In the original design A, all latches have minimum pulse width and minimum period attributes on their clock pins. In the extracted model B, the minimum pulse width attributes are translated into attributes on the clock port. Only the maximum of the minimum pulse width and minimum period values are used. CLK has the MPW\_HIGH = 2.0 attribute set on it. In the original design A, the input pin IN4 does not have minimum pulse width or minimum period attributes, even though it is in the fanin of the clock pin of latch I3.

Violations can occur when the clock pulse width decreases to less than the required minimum pulse width during the course of its propagation toward the latch clock pin. These violations are not reported for the model. In this figure, the clock waveform at the input latch I3 has a width of 1.9, which is less than the required 2.0. Using 2.0 for the clock port attribute without considering the slew effects causes this minimum pulse width violation to be missed.

Note:

The extracted timing model looks at the most constraining minimum pulse width attributes of the library pin and the minimum pulse width constraints applied with the set\_min\_pulse\_width command.

False path:

If the set\_false\_path constraint is applied through some internal pins/ports, then those paths/arcs are removed during extraction.

SIGN OFF

HVT: High voltage threshold => High delay, low leakage power => used in power critical functions.  
LVT: Low voltage threshold => Low delay, high leakage power => used in time critical functions.

RVT: Regular voltage threshold (SVT – Standard Threshold Voltage) offers trade-off between HVT and LVT i.e., moderate delay and moderate power consumption.

# Design rule check

* **Checking list**
  + The clock tree must use the same Vt type cell (LVT, RVT, HVT)
  + Clock path should use the CK type cell
  + Forbid the delay cell in the clock path
  + Input slew check on the clock path (Use the maximum slew of previous stage and multiplied 0.85)
  + Input slew check on the data path (Use the maximum slew of previous stage as criteria )
  + Output capacitance check
  + Output port slew check
* Syntax
  + General Syntax :
    - fstaH-group <sdc\_id> <mode 1> <mode 2> <mode ...> [-ip/-nonip] [-ptsi] [-glitch] [-gen\_aocv\_index] [-chk\_aocv\_index][-drc][-drc\_only][-acs]
  + CPF-In Flow Syntax:
    - fstaH-msv<cpf\_file\_name> <mode 1> <mode ...> [-ip/-nonip] [-ptsi] [-glitch ] [-gen\_aocv\_index] [-chk\_aocv\_index][-drc][-drc\_only][-acs]
* Option
  + [-drc]: timing check with the design rule check
  + [-drc\_only]: only analyze the design rule checking
  + [-acs]: source acs file only when the option be specified

**Notice: [-acs] should be used with [-drc/-drc\_only]**

* Example
  + Timing check with the design rule check
    - fstaH -group $SDC aocvwg -drc
  + Only analyze the design rule checking in the worst corner
    - fstaH -group $SDC w -drc\_only

# SPEF

Standard Parasitic Extraction Format (SPEF) allows the representation of parasitic information of a design(R, L, and C) in an ASCII (American Standard Code for Information Interchange exchange format). A user can read and check the values in a SPEF file.

A typical SPEF file will have 4 main sections

– a header section

– a name map section

– a top level port section

– the main parasitic description section.

**Header section**: contains basic information such as the SPEF version number, design name, and units for R, L and C

**Name map section**: To reduce file size, SPEF allows long names to be mapped to shorter numbers preceded by a \*. This mapping is defined in the name map section.

\*DELIMITER: --> Delimiter between the pin and its instance.

\*BUS\_DELIMITER [ ] ---> specifies the prefix and suffix used to identify a bit of a bus.

\*T\_UNIT 1.00000 NS---> NS | PS: specifies the time unit

\*C\_UNIT 1.00000 FF----> PF | FF: specifies the capacitance unit

\*R\_UNIT 1.00000 OHM-----> OHM | KOHM: specifies the resistance unit

\*L\_UNIT 1.00000 HENRY-----> HENRY | MH | UH: specifies the inductance unit.

**The port section**: is simply a list of the top level ports in a design. They are also annotated as input, output or bidirect with an I, O or B.

**Parasitics** :

Each extracted net will have a \*D\_NET section. This will usually consist of a \*D\_NET line, a \*CONN section, a \*CAP section, \*RES section and a \*END line. Single pin nets will not have a \*RES section. Nets connected by abutting pins will not have a \*CAP section.

The **\*D\_NET line** tells the net name and the net's total capacitance. This capacitance will be the sum of all the capacitances in the \*CAP section.

**\*CONN Section**

The \*CONN section lists the pins connected to the net. A connection to a cell instance starts with a \*I. A connection to a top level

port starts with a \*P.

The syntax of the \*CONN entries is:

\*I <pin name> <direction> \*C <xy coordinate> <loading or driving information>

Where:

– The pin name is the name of the pin.

– The direction will be I, O or B for input, output or bidirect.

– The xy coordinate will be the location of the pin in the layout.

– For an input, the loading information will be \*L and the pin's capacitance.

– For an output, the driving information will be \*D and the driving cell's type.

– Coordinates for \*P port entries may not be accurate because some extraction tools look for the physical location of the logical

port (which does not exist) rather then the location of the corresponding pin.

**\*CAP Section**

The \*CAP section provides detailed capacitance information for the net. Entries in the \*CAP section come in two forms, one for a capacitor lumped to ground and one for a coupled capacitor.

A capacitor lumped to ground has three fields,

– an identifying integer,

– a node name and

– the capacitance value of this node

1 regcontrol\_top/GRC/U9743:E 0.936057

A coupling capacitor has four fields:

2 regcontrol\_top/GRC/U9409:A regcontrol\_top/GRC/U10716:Z 0.622675

If netA is coupled to netB, the coupling capacitor will be listed in each net's \*CAP section.

**\*RES Section**

The \*RES section provides the resistance network for the net.

Entries in \*RES section contain 4 fields,

– an identifying integer,

– two node names and

– the resistance between these two nodes.

– E.g

1 regcontrol\_top/GRC/U9743:E regcontrol\_top/GRC/U9407:Z 10.7916

The resistance network for a net can be very complex. SPEF can contain resistor loops or seemingly ridiculously huge resistors even if the layout is a simple point to point route. This is due how the extraction tool cuts nets into tiny pieces for extraction and then mathematically stitches them back together when writing SPEF.

# SDF

What is SDF: SDF (Standard Delay Format) is an IEEE standard for the representation and interpretation of timing data for use at any stage of the electronic design process. The ASCII data in the SDF file is represented independent of any tool and language. It includes path delays, timing constraint values, interconnect delays, high level technology parameters and etc.

Information of SDF

* Type of timing related Information
  1. Delays:
     + Path delay.
     + Device delay
     + Interconnect delay
     + Port delay
  2. Timing checks: setup, hold, recovery, removal, skew, width, period, and nochange.
  3. Timing constraints: path, skew, period, sum, and diff
  4. Timing environment: intended operating timing environment
* Ways of Implementation:
  1. Incremental delays: Introduce delay data that is added to existing delay values in the design
  2. Absolute delays: Introduce delay data to replace existing delay values in the design.
* Others:
  1. Conditional and unconditional module path delays and timing checks
  2. Design/instance-specific or type/library-specific data
  3. Scaling, environmental, and technology parameters

SDF files can be used for BACK-ANNOTATION and also for FORWARD-ANNOTATION. It has

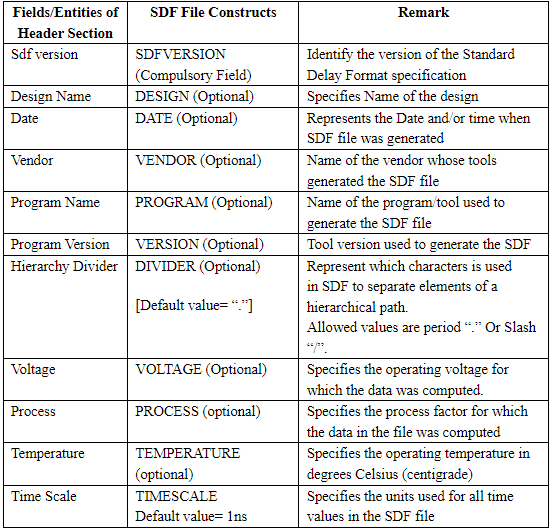
Description of computed timing data for back-annotation.

An advantage of this approach is that once an SDF file has been created for a design, all analysis and verification tools

can access the same timing data, which ensures consistency.

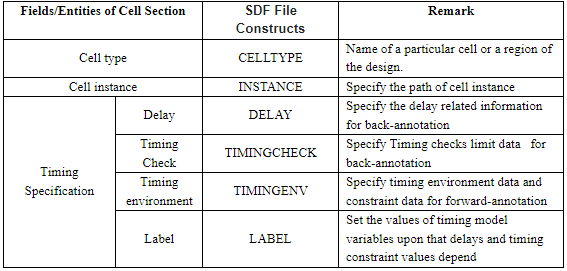
The specification of timing constraints for forward-annotation.

Syntax of SDF :         (DELAYFILE <header section>  <cell section> )

**Header** **Section** contains all the relevant information about the SDF file like design name, sdf version, tool which is used to generate this SDF file, temperature, voltage etc. Most of the information is only for documentation purpose and are optional. 

**Cell Section** has all the important information require for timing calculation like delay, timing constraints, timing environment and etc for a particular cell or part of the design.

In the SDF, its necessary that at least "1" (one) cell section should be present. There is no limit on higher side. Sequence of Cell section is also important in the SDF file. Lets suppose that there are 2 cell sections defining the timing properties/specification for same part of the design, then the information in one section can override (Replace the existing information- this will happen when the ABSOLUTE keyword is present in the cell section) Or may be cumulative with other information ( added to previous information- if INCREMENTAL keyword is present in the cell section). EDA tools read the file from top to bottom ( or say beginning to end) and apply the different constraint or information in the similar sequence.

syntax of the Cell section (CELL  <cell\_type>  <cell\_instance>  <timing\_spec> )

CELLTYPE

It represent the Name of the Cell in the design. That may be either the cell name mentioned with in the Library (e.g standard cell library) OR name of a

particular region with in the design ( e.g Name of a Block).

E.g- there are 3 types of CELLTYPE.

(CELLTYPE "MYDESIGN") >> Name of My design.

(CELLTYPE "AND") >> Name of a cell present in standard cell library.

(CELLTYPE "DFF") >> Name of a block present in my design at some hierarchy

INSTANCE

It identify the region with in the design for which timing information is present in the cell section. If you want to apply the timing constraints for a particular instance of a cell, then you have to mention of the complete path of that particular instance. Else if you want to use associated timing data with all occurrences of the specified cell type, then in the INSTANCE mention wild character ( \* ).

Path of the instance should start from the same level of design from where EDA tool (or say annotator) is instructed to apply the SDF file.

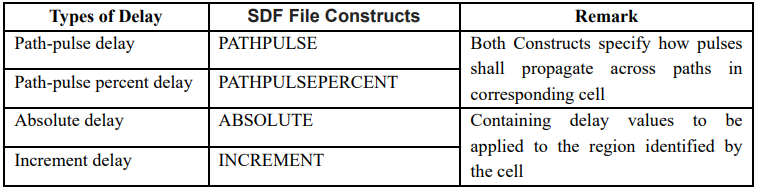
For exampleFrom the top level the path of AND gate is "Top/block1/block2/block3/AND" but you have instructed to annotator to apply a SDF from

block2 level, then the path in the INSTANCE should be "block3/AND".

Similarly if you will mention (INSTANCE \*) in the above case, then the timing information only link to the AND gate present in the same

level or in below levels, not at a level of block1.

This entry should be in consistance with the CELLTYPE. It should not be the case that Cell name is AND and the instance path is with respect to the NAND gate.

DELAY: There are 4 ways to implement the delay value on the design defined in the SDF file

Path-Pulse Delay and Path-Pulse Percent Delay:

As such both are same and only difference in terms of representation of value. In Path pulse Percent Delay every thing is in terms of percentage only.

These parameter usually apply a limit on a PULSE and determine whether a pulse of certain width can travel through a device or not and appear in which form to output.

Syntax is (PATHPULSE <input port> <output\_port> <pulse\_rejection\_limit> <X-limit>)

Note:

If input\_port and output\_port is not defined then these limits applied to all paths present in the instance defined by cell entry.

pulse\_rejection\_limit means if the width of Pulse is less then this value, nothing will appear at the output.

X-limit means if width is less then this but greater then pulse\_rejection\_limit, output will be unknown (means X stage).

If any one value is provided in place of these 2 limits, then both limits are set to that value.

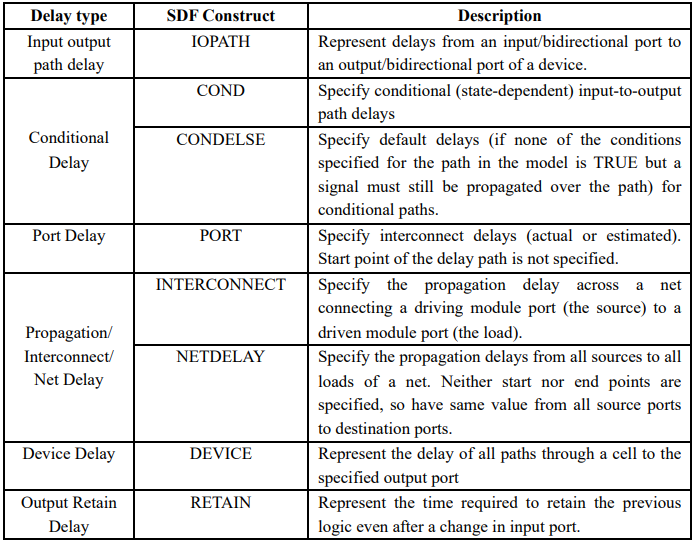
These Limits must not be in negative.

ABSOLUTE and INCREMENT:

ABSOLUTE - introduce delay data to replace existing delay values in the design during annotation.

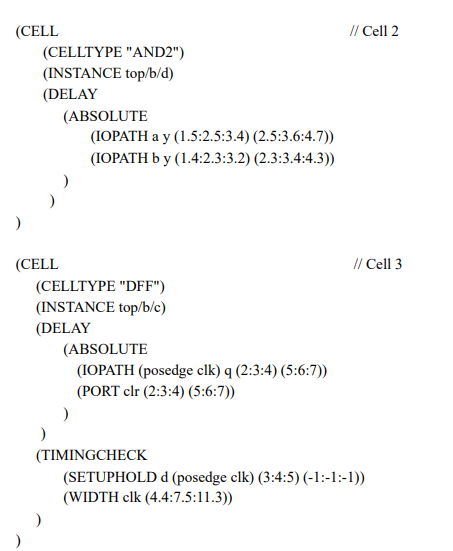
INCREMENT -introduce delay data that is added to existing delay values in the design during annotation.

Both of these can be specified with the help of same type of SDF constructs. These are based on the type of Delay in a design



**Timing check**

Specify constraint values with critical paths in the design (PATHCONSTRAINT, PERIODCONSTRAINT, or SKEWCONSTRAINT) and provide information about the timing environment (ARRIVAL, DEPARTURE, SLACK, or WAVEFORM) in which the circuit shall operate. Constructs in this sub-clause are used in forward-annotation and not back-annotation.



# RC corner

Parasitic Interconnect Corners

Parasitics can be extracted at many corners. These are mostly governed by the variations in the metal width and metal etch in the manufacturing process. Some of these are:

* Typical: This refers to the nominal values for interconnect resistance and capacitance
* Cmax: This refers to the interconnect corner which results in

maximum capacitance. The interconnect resistance is smaller than at typical corner. This corner results in largest delay for paths with short nets and can be used for max path analysis.

* Cmin: This refers to the interconnect corner which results in minimum capacitance. The interconnect resistance is larger than at typical corner. This corner results in smallest delay for paths with short nets and can be used for min path analysis
* Rcmax: This refers to the interconnect corner which maximizes the interconnect RC product. This typically corresponds to larger etch which reduces the trace width. This results in largest resistance but corresponds to smaller than typical capacitance. Overall, this corner has the largest delay for paths with long interconnects and can be used for max path analysis.
* Rcmin: This refers to the interconnect corner which minimizes the interconnect RC product. This typically corresponds to smaller etch which increases the trace width. This results in smallest resistance but corresponds to larger than typical capacitance. Overall, this corner has the smallest path delay for paths with long interconnects and can be used for min path analysis.

Based upon the interconnect R and C for various corners described above, an interconnect corner with larger C results in smaller R and a corner with smaller C results in larger R. Thus, the R partially compensates for the C across various interconnect corners. This implies that no single corner maps to an extreme value (worst-case or best-case) for path delay for all types of nets. The path delay using Cworst / Cbest corners is extreme only for short nets while RCworst / RCbest corners is extreme only for long nets.

The typical interconnect corner is often the extreme in terms of path delay for nets with average length. Thus, designers often choose to verify the timing at various interconnect corners described above. However, even the verification at each corner does not cover all possible scenarios since different metal layers can actually be at different interconnect corners independently - for example, Max C corner for METAL2, Max RC corner for METAL1, and so on. Statistical timing analysis offers a mechanism for static timing analysis where different metal layers can be at different interconnect corners.

PVT Corners

The PVT corners dictate at what conditions the STA analysis takes place.

The most common PVT corners are:

* WCG (slow process, low power supply, high temperature)
* WCGC (worst-case slow at cold - slow process, low power supply, low temperature): Below 65nm process node, this corner may be the slowest corner in some library cells.
* BCG (fast process, high power supply, low temperature)
* BCGH: (best-case slow at hot – fast process, high power supply, high temperature): Below 40nm process node, it has significant resistance effective.

The Process is divided into 5 different corners (the first letter stands for NMOS and the second letter stands for PMOS):  
TT: Typical N Typical P

FF: Fast N Fast P

SS: Slow N Slow P

FS: Fast N Slow P

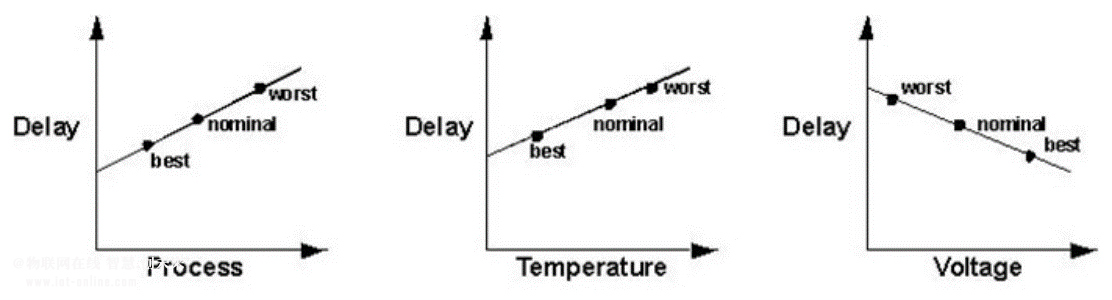
SF: The Slow N Fast P

Process Corner is for both N-type and P-type doping at different concentrations. NMOS and PMOS are process-independent and do not affect each other. By adjusting the speed of the process injection, the speed of the device is simulated, and the different levels of FF and SS are set according to the size of the deviation.

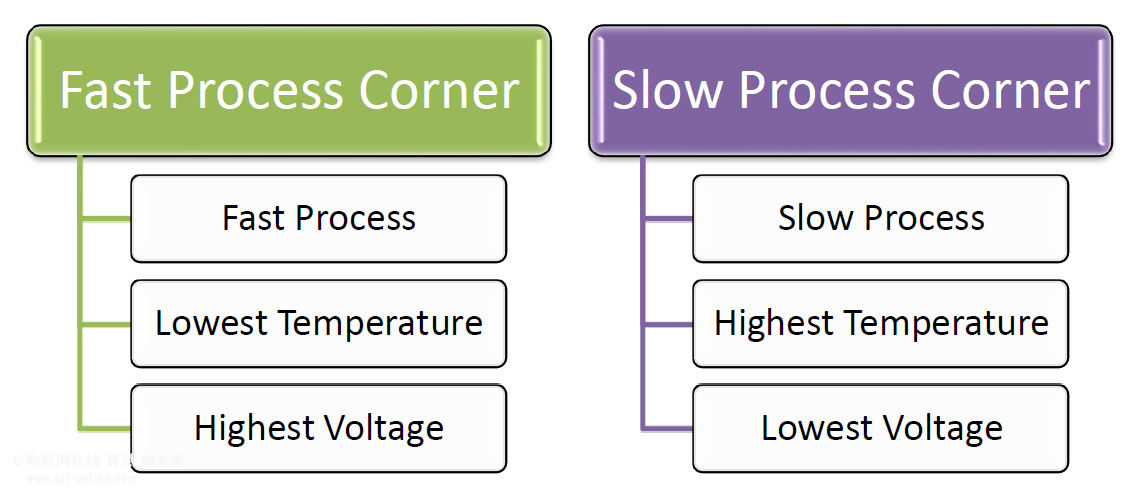
Under normal circumstances, most of them are TT and the above 5 corners can cover

About 99.73 % of the range at +/-3 sigma, and the occurrence of this randomness is in line with the normal distribution

The effect of PVT on chip performance

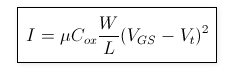


Different PVT conditions form different corners, and the influence of RC corners should be considered in the digital circuit design



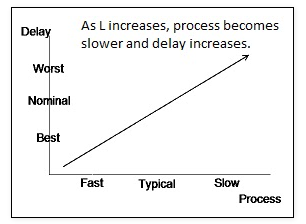
1. Process: You must have heard people talking in terms of process values like 90 nm, 65 nm , 45 nm and other technology nodes. These values are characteristic of any technology and represents the length between Source and Drain of a MOS transistor that you might have studied. While manufacturing any die, it has been seen that the dies that are present at the center are pretty accurate in their process values. But big, but can have significant impact on timing.

Recall from your undergrad courses the following formula for current flowing in a MOS transistor:



L repersents the process value. For same temperature and voltage values, current for 45nm process would be more than current for 65nm process

More is the current, faster is the charging/discharging of capacitors. And this means, delays are less.



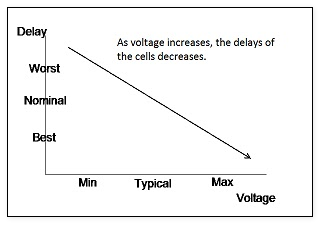
1. Voltage:

The voltage that any semiconductor chip works upon is given from outside. Recall while working on breadboards in your labs, you used to connect a 5V supply to the Vcc pin of your IC. Modern chips work on very less voltage than that. Typically around 1V-1.2V

This voltage must be the output of either a DC source or maybe the output of some voltage regulator. The output voltage of voltage regulator might not be a constant over a period of time. Let’s say, you expected your voltage regulator to give 1.2V, but after 4 years, it’s voltage dropped down to 1.08V or increased up to 1.32V. So, you gotta make sure your chip is working well between 1.08 and 1.32V

This is where the need to model Voltage variations come into picture.

From the same equation as above, it can be seen that more is the voltage, more is the current. And hence, delays are less.



1. Temperature:

The ambient temperature also impacts the timing. Let’s say you are working on a gadget in Siachen glacier where temperature can drop down to -40 degrees centigrade in winters and you expect your device to be working fine. Or maybe you are in Sahara desert, where ambient temperature is +50 degrees and your car engine temperature is +150 degrees and again you expect your chip to working fine. While designing, therefore, STA engineers need to make sure that their chip will function correctly in the temperatures between -40 to +150 degrees.

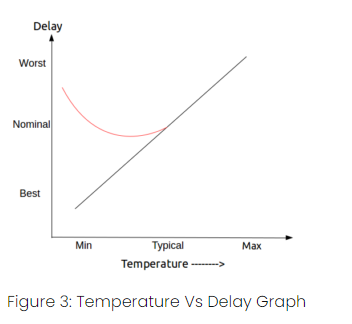
Higher is the temperature, more is the collision rate of electrons within the device. This increased collision rate forbids other electrons in the periphery to move. Since electron movement is responsible for current flowing in the device, current would decrease with increase in temperature. Therefore, delays are normaly more at higher temperatures.

For technology nodes below 65nm, there’s a phenomenon called TEMPERATURE INVERSION, where delays tend to increase with decreasing temperature. We shall talk about the same later. Don’t get confused with it here.

As temperature increases, mobility and threshold voltage start decreasing. The delay is inversely proportional to the mobility and directly proportional to the threshold voltage.

In the higher technology node, where the supply voltage is very high, the effect of VTh is very low as (VGS – VTh) value is large. Hence mobility plays major role in deciding current. So at higher technology nodes, when the temperature increases mobility decreases and as a result the delay will increase.

At the lower technology node (specifically, less than 65nm), the supply voltage is very low, so the (VGS – VTh) difference is small and the square of this value is very small resulting reduced ID current, which increases delay at lower temperature. Where at other end above 65nm delay decreases at lower temperature.



HVT, RVT, LVT, ULVT:

HVT: High voltage threshold => High delay, low leakage power => used in power critical functions.  
LVT: Low voltage threshold => Low delay, high leakage power => used in time critical functions.

RVT: Regular voltage threshold (SVT – Standard Threshold Voltage) offers trade-off between HVT and LVT i.e., moderate delay and moderate power consumption.

ULVT: Ultra low voltage threshold => These cells are utilized in aplications that need very low power consumption, such as mobile devices and other battery-powered devices and are made to work at lower voltage levels than LVT cells

The difference between different Vt cells

The lower the threshold voltage, the smaller the saturation current, so the higher the speed performance. However, the leakage current will become larger, so the leakage power will become larger.

Speed: HVT < SVT < LVT < ULVT

Latency: HVT > SVT > LVT > ULVT