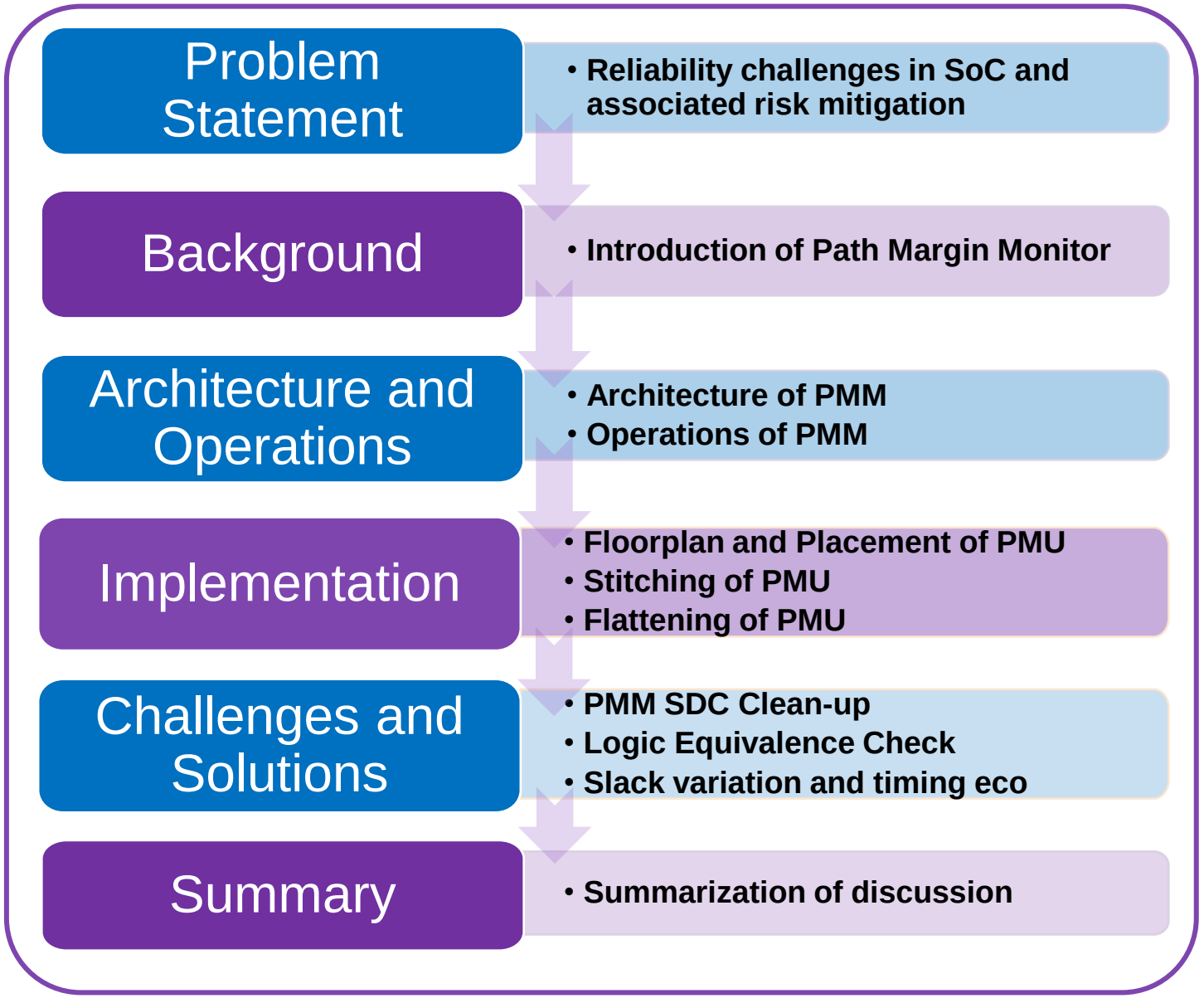
Adoption of PATH MARGIN MONITOR (PMM) and Overcoming the Implementation Challenges in SoC Design

Rajesh Yadav
Dr. Anukool Rajoriya
Ravish Shah
Shantanu Kumar

Samsung Semiconductor India Research – Bangalore (India)

Coach:
Jagadeeswaran
Application
Engineer

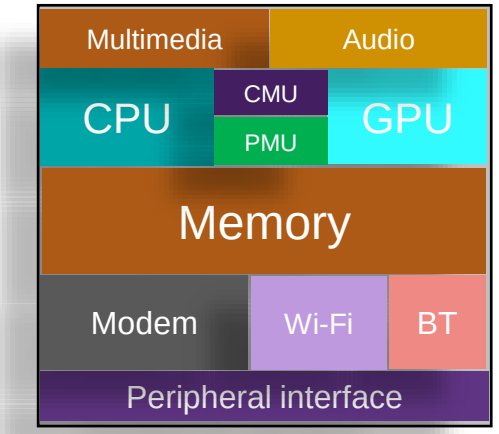
Agenda


Problem Statement
• Reliability challenges in SoC and associated risk mitigation
Background
• Introduction of Path Margin Monitor
Architecture and Operations
• Architecture of PMM
• Operations of PMM
Implementation
• Floorplan and Placement of PMU
• Stitching of PMU
• Flattening of PMU
Challenges and Solutions
• PMM SDC Clean-up
• Logic Equivalence Check
• Slack variation and timing eco
Summary
• Summarization of discussion

Problem Statement

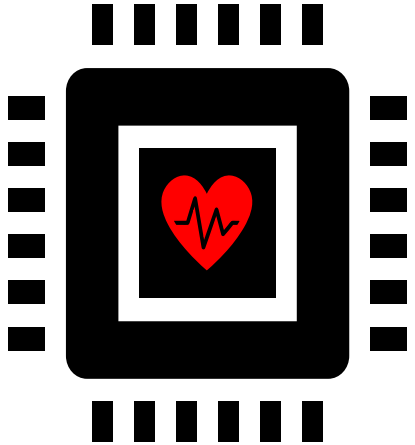
Reliability Concerns in Complex SoC

- **Complexity of Modern SoC**
  - Nowadays a single SoC contains many different modules
  - These leads higher transistor counts, complex interconnect structures, which makes SoCs incredibly complex



- **Reliability Challenges**
  - Reliability and robustness become major concerns in complex SoC
  - Performance degrades gradually which may lead to chip failure/malfunction
  - Major factors Transistor strain, Electrostatic Effects, self heating, Electro migration and IR
  - Malfunction/failure of chip, is very risky and could be a disaster

- **Risk Mitigation**
  - Need a continuous “health monitor for a chip” for preventing malfunctions
  - Traditional methods: ring oscillator and on chip temperature sensors not optimal
  - **Path Margin Monitor** can access the path margin of timing paths on real time basis and capable to mitigate the risk of chip failure due to path degradation.



Introduction of Path Margin Monitor - PMM

Path Margin Monitor - Introduction

- **About Path Margin Monitor**
  - IP developed by **Synopsys Inc.,** promising to address the reliability related risks of chip failure
  - PMM can check available setup margin of functional path while chip is in operation and can alerts before setup failure
  - It also helps to improve timing model for a technology node by comparing the pre and post silicon timing results
  - PMM acts as a **microscopic guardian** inside the SoC which continuously monitors health of the chip

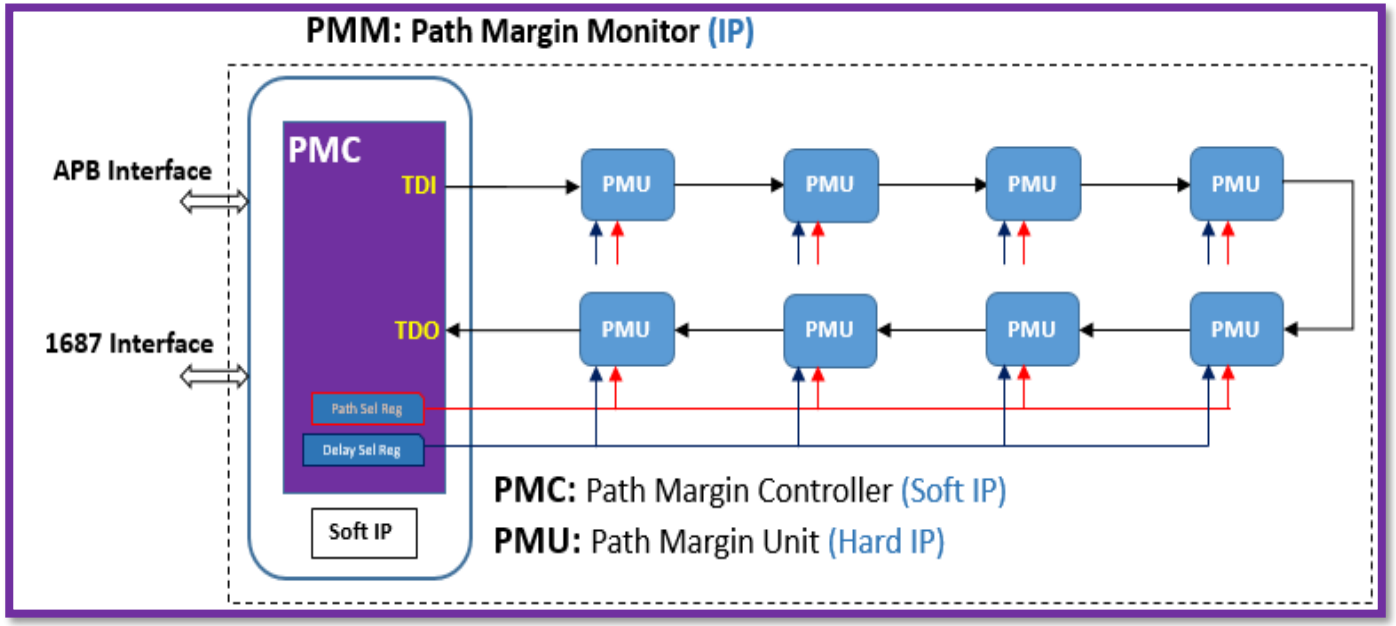
- **Path Margin**
  - The difference between the data required time and actual data arrival time at capture flop
- **Data path degradation**
  - Data path degrades due to continuous voltage application on the silicon transistors – termed as Aging effect of transistors. Which may lead to performance degradation or chip failure

A sufficient path margin makes a chip more reliable and robust

Samsung Semiconductor India Research (SSIR) have implemented the PMM in SoC in collaboration with Synopsys Inc. (IP provider)

Architecture and Operations of PMM

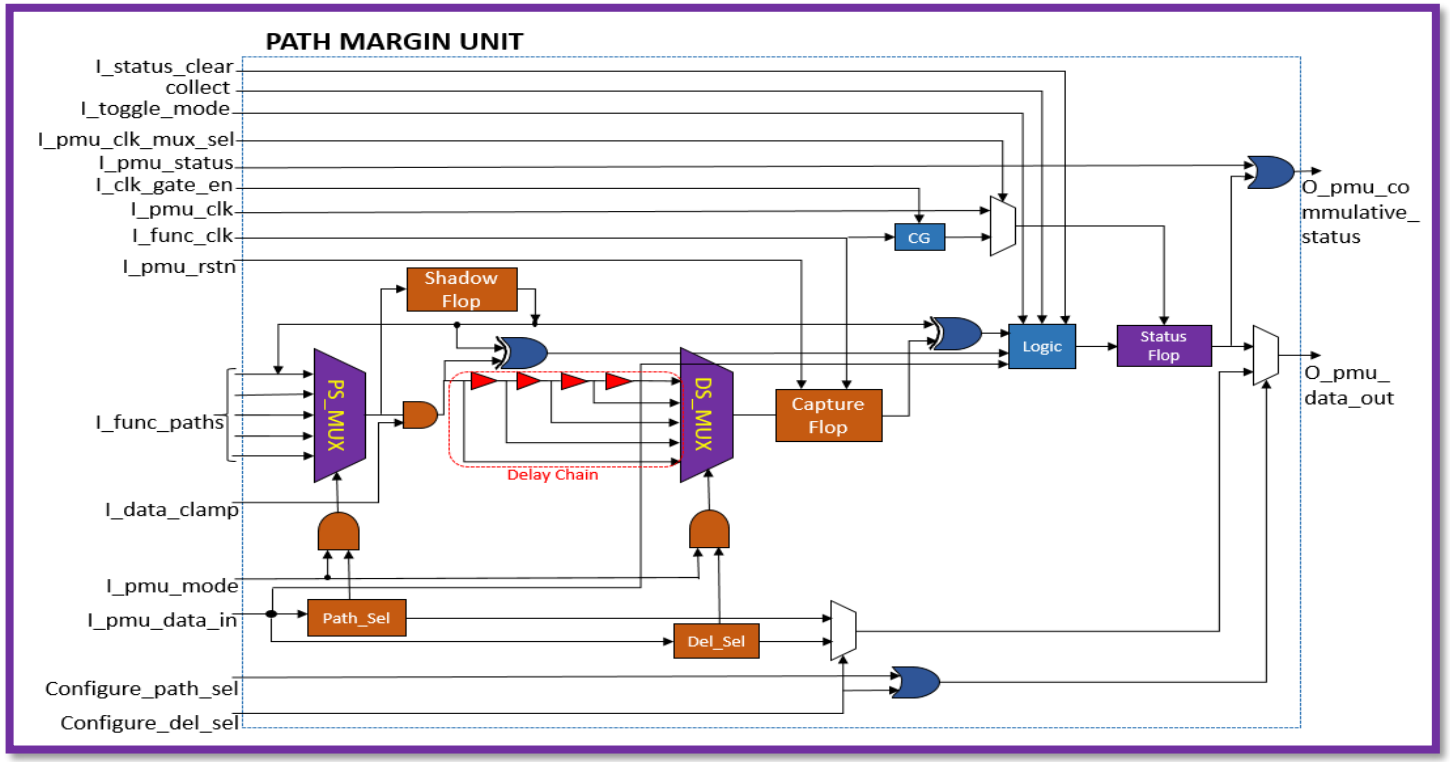
Architecture of PMM



- **Main components of PMM**
  - Path Margin Controller (PMC) - **Soft IP**
  - Path Margin Unit (PMU) - **Hard IP**

- **High level architecture of PMM**
  - All PMUs are connected to each other in a **daisy chain loop,** and loops are connected to PMC
  - PMC is the main controlling unit and it can control **15** such PMU loops
  - PMU is the main functional unit which get connected to critical functional endpoints
  - PMM has two interface : **APB Interface** (used in in-filed monitoring) and **IEEE 1687 Interface** (used in manufacturing test)
  - Total numbers of PMU in a design is decided by total core area to be monitored and coverage area of each PMU

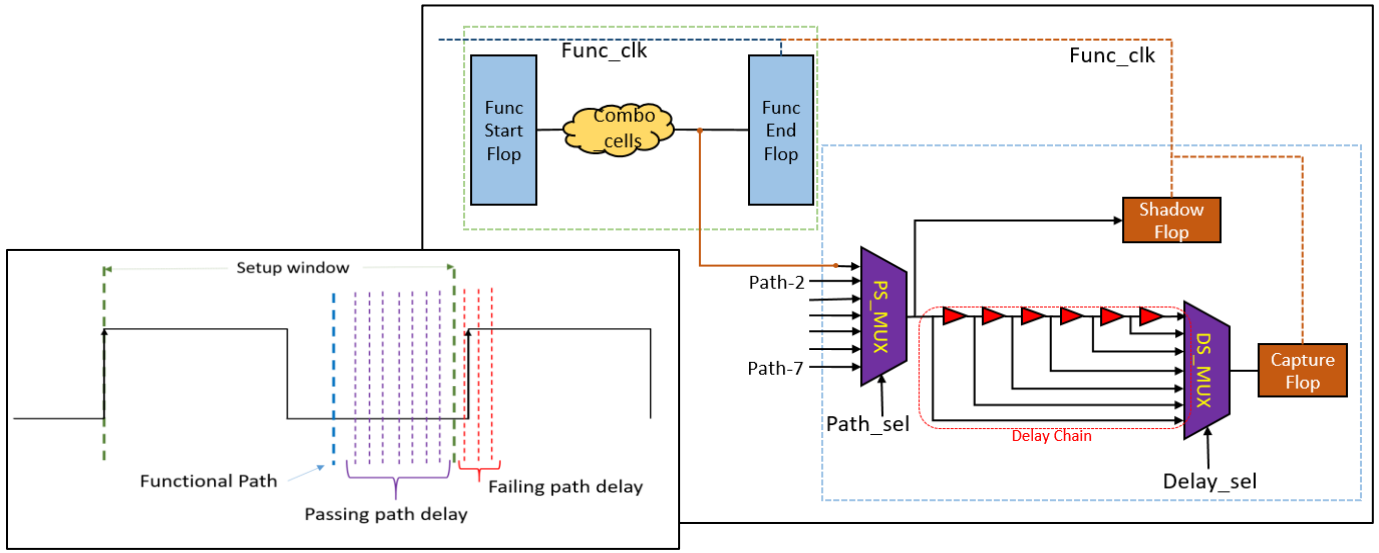
Architecture of PMU



High level architecture of Path Margin Unit

- PMU can be connected to **7** functional endpoints. **Path selection register** and multiplexer, select one endpoint to monitor
- **Delay chain** and Delay selection multiplexer add step delay before capture flop
- **Shadow flop** captures the data without any delay addition
- **Capture flop** captures the data after adding the step delay
- **Status flop** stores pass/fail status

Operation of PMM



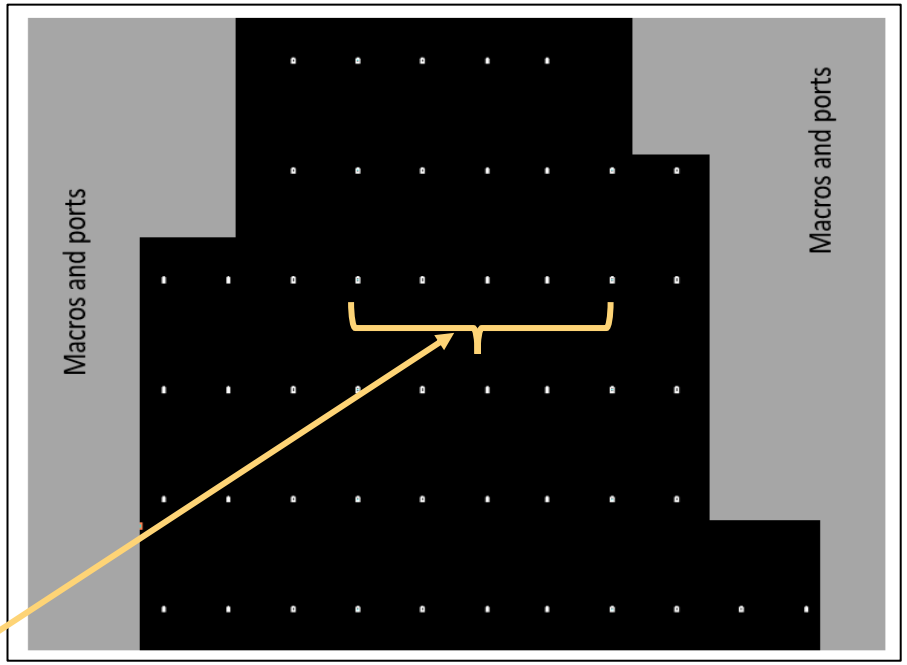
- **PMM Operations**
  - One endpoint is get configured by path selection multiplexer
  - Shadow flop captures the data without any delay addition, whereas in capture flop path, step delay is added until setup violation
  - If the final delay on capture flop is well within the setup limit, then data captured in capture flop is correct and the added delay is called **passing path delay**
  - If additional delay causes data to arrive after setup window, data will not be captured correctly in capture flop. Such added delay value is called **failing path delay**
  - The maximum delay at which data is captured correctly in capture flop is called **PMM threshold margin**
  - Initial threshold margin for all endpoint is stored, which help to quantify the data path degradation at any point of time

Implementation of PMM

Implementation of PMM

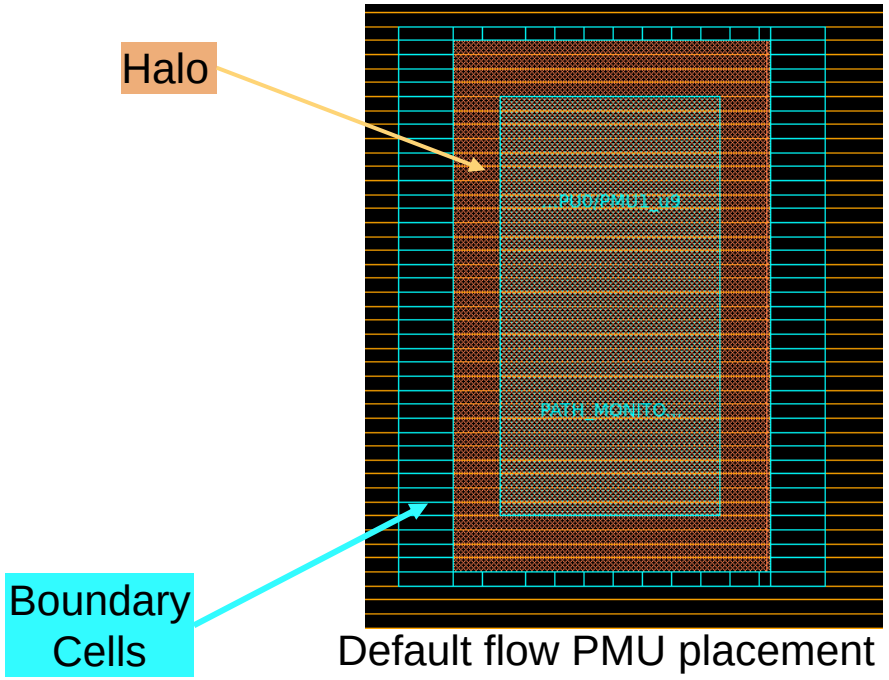
- **Floorplan of PMU**
  - PMUs are instantiated as a hard-macro using physical database and sprinkled throughout in the design
  - These are pre placed in floorplan stage after determining total placeable region and voltage areas
  - In our case, each PMU covers 50-micron radius, and based on that we had total 50 PMUs requirement
  - PMUs are snapped to the grid so that core Power-Ground nets and cell grid aligns with PMU internal structure

- **Floorplan Guidelines**
  - There are some specific requirement in floorplan as listed below
    - Size of PMU must be multiple factor of block site row
    - No boundary or tap cells to be inserted around PMU IP
    - Offset between PMU boundary and PMU core should be 0
    - Lower metals PG should align with block PG for continuity
    - First cell row orientation and PG structure of PMU should be same as block level

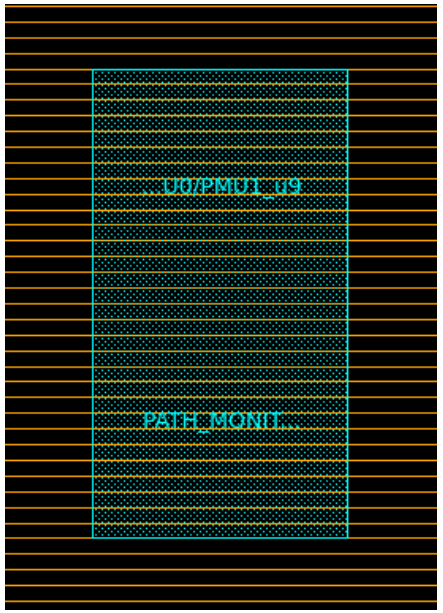


Implementation of PMM

- **Floorplan challenges**
  - By default, PnR flow treat PMU as a normal macro, so flow creates halo and add boundary cells around PMUs
  - But we don’t need halo and boundary cells around the PMU. So, we had to tweak our flow
  - After tweaking the flow, we got the PMU placement without halo and boundary cells

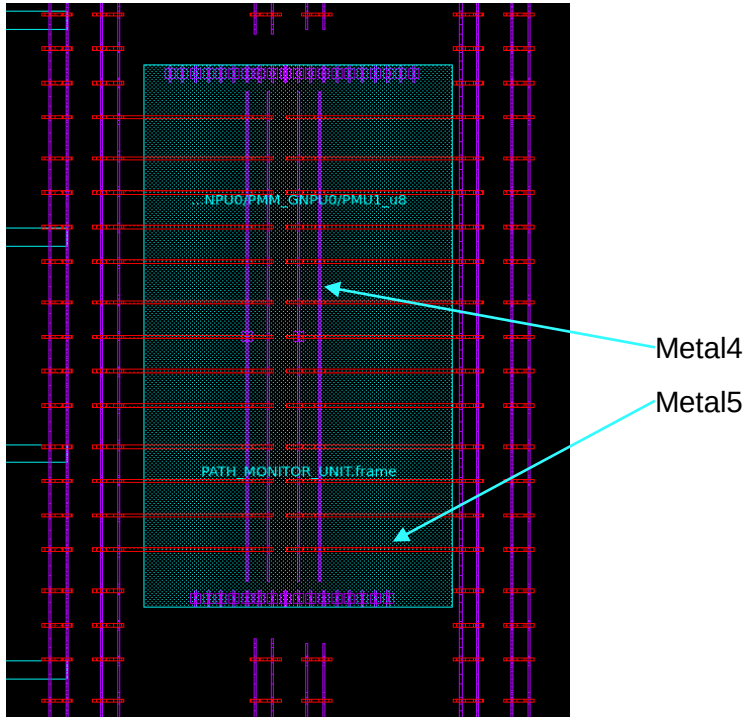


Default flow PMU placement


After flow changes, PMU placement

Implementation of PMM

- **Powerplan of PMU**
  - PMU uses till Metal4 layer , where pins are on higher 2 layers and PG terminals are on metal4
  - Metal1 at block level is trimmed without keepouts around PMU, so that VDD and VSS of PMU and block will be continuous
  - Rest of the layers can be trimmed with keepout margin
  - PMU’s Metal4 PG-pins are connected to core Metal5 PG using Via45 as shown in figure



Powerplan of PMU

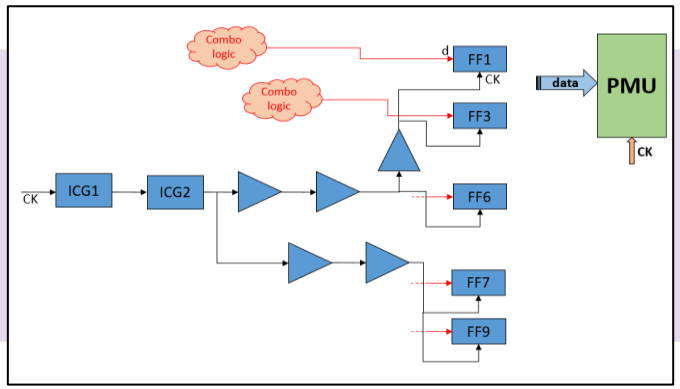
Implementation of PMM

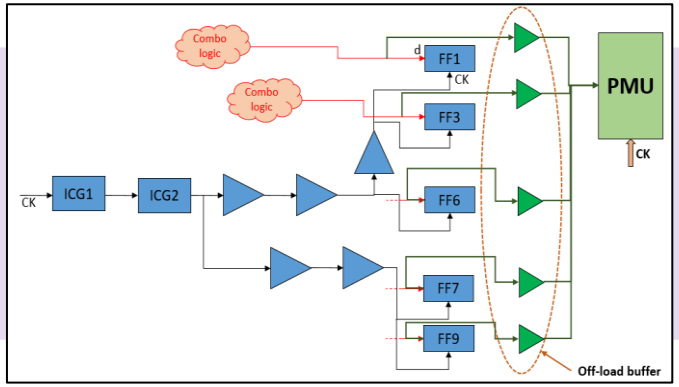
- **PMU Stitching**
  - After Clock Optimization happened, we need to assign the selected **Testable Endpoints** to nearest PMU
  - Once endpoint assignment done, we need to establishing connection between PMU and selected endpoints. which is called as PMU stitching
  - Based on search radius, setup and hold slack range, clock group, and other coverage criteria - testable endpoints are selected and stitched

- **Stitching guidelines**
  - Following guidelines must be followed in PMU stitching
    1. Endpoints must belong to defined **voltage area, and clock domain**
    2. Endpoints **setup slack** must be within defined range across all active scenarios, **hold slack** must be positive
    3. All endpoints connected to same PMU must belong to same **leaf level** ICGs and will have similar clock latency
    4. Endpoints with **smaller setup slack, higher instance coverage, higher testable rate** will have higher priority

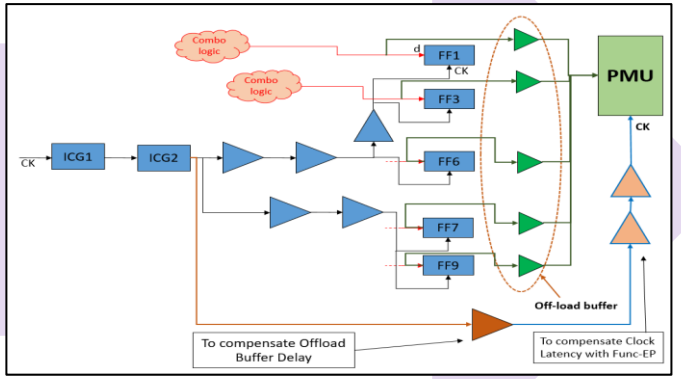
Implementation of PMM

- **PMU stitching**
  - Once functional endpoints are selected, **Offload buffers** are inserted in data path to connect the endpoints with PMUs
  - Similarly, clock path also need to be stitched. For clock stitching we need to add clock inverter pairs in clock path and extend the endpoint clock till PMU





Off-load buffer insertion in data path

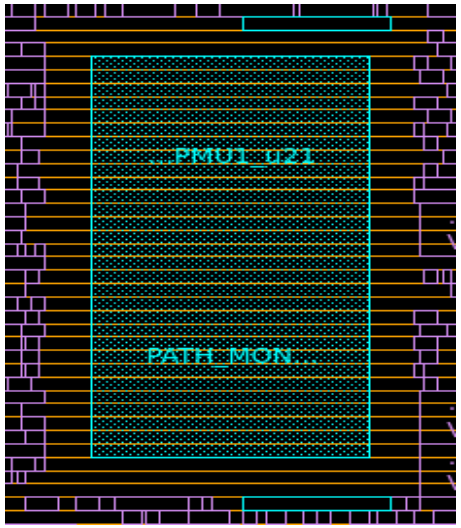


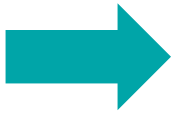
Inverter pair insertion in clock path

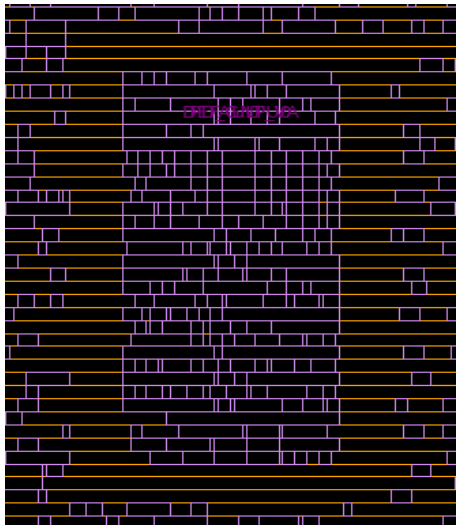
Implementation of PMM

- **PMU Flattening**
  - Once place and route is completed, the PMUs are flattened and merged into block level
  - There are some checks recommended which we need to make sure that those are correct in flatten db.
    1. PMU cell rows need to be continuous and cell orientation should match with block level
    2. Metal1 PG should be continuous between PMU and block level
    3. There should not be any boundary cell between PMU and core
    4. PG route track color should be aligned

Before PMU Flattening







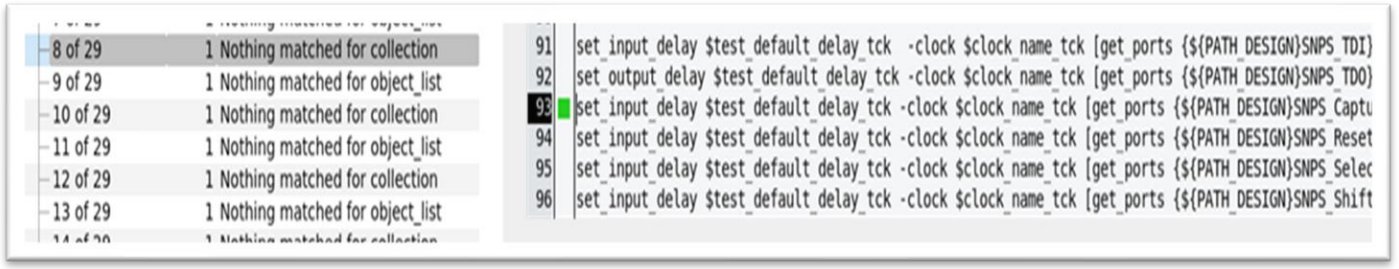
After PMU Flattening

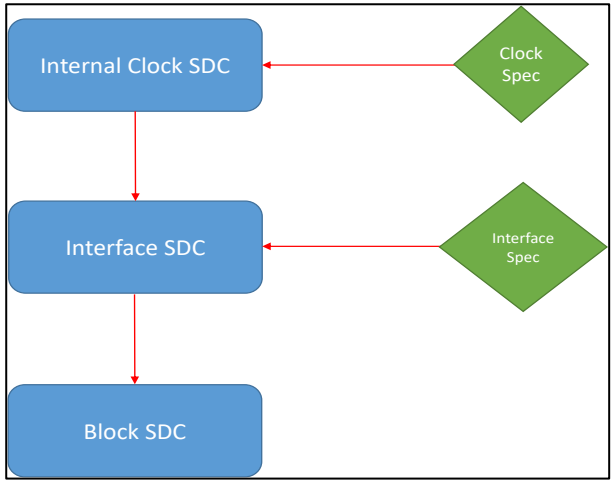
PMM Implementation Challenges and Solutions

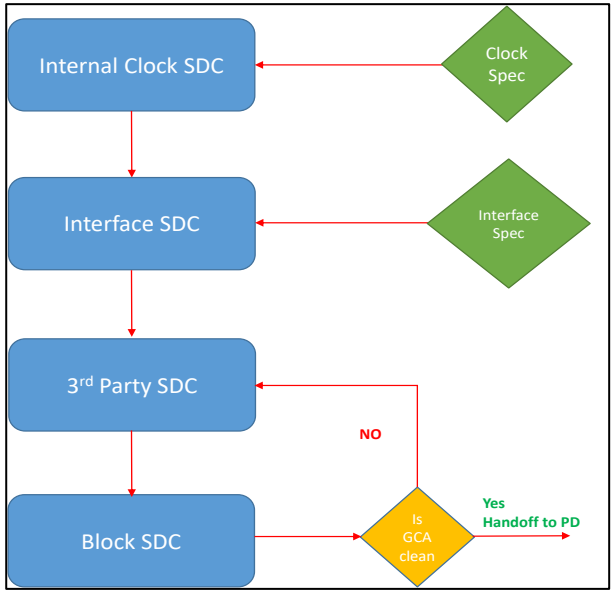
Implementation Challenges and Solutions

- **PMM SDC clean-up and Integration**
- **Samsung SDC generation flow**
  - Decentralized SDC setup, where internal and interface constraints come from different area
  - SDCs are categorized and segregated and sourced in a ordered manner
  - Challenging to integrate 3rd party SDC

- **Galaxy Constraint Analyzer (GCA)**
  - Helped in
    - Catch the SDC related issues quickly
    - Able to provide timely feedback to IP owner

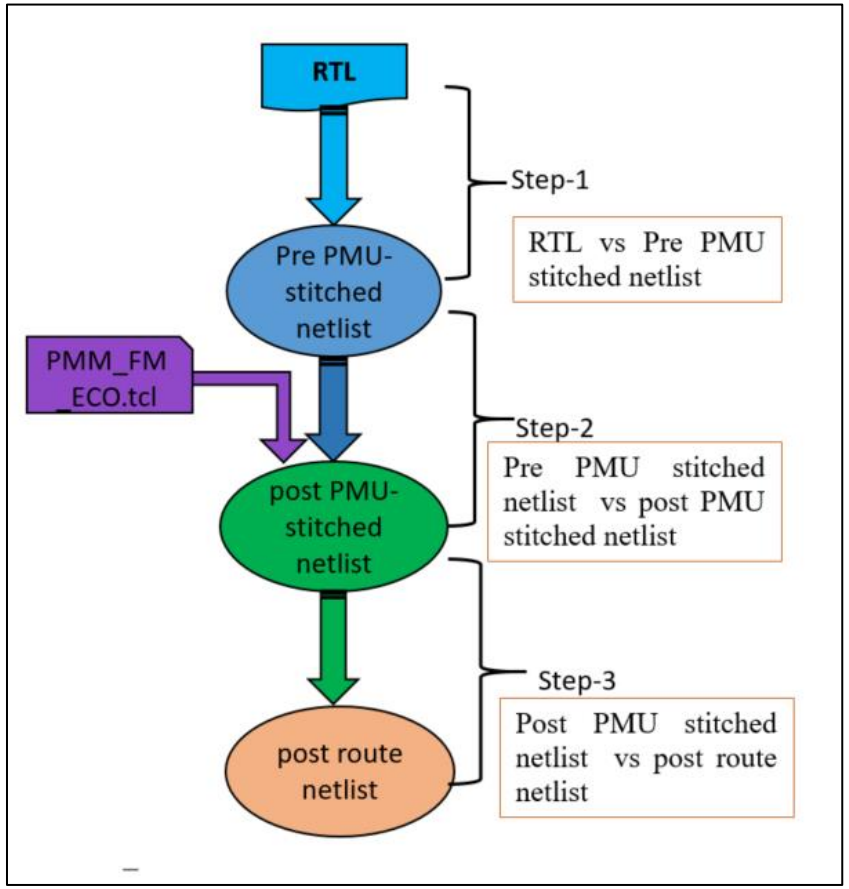






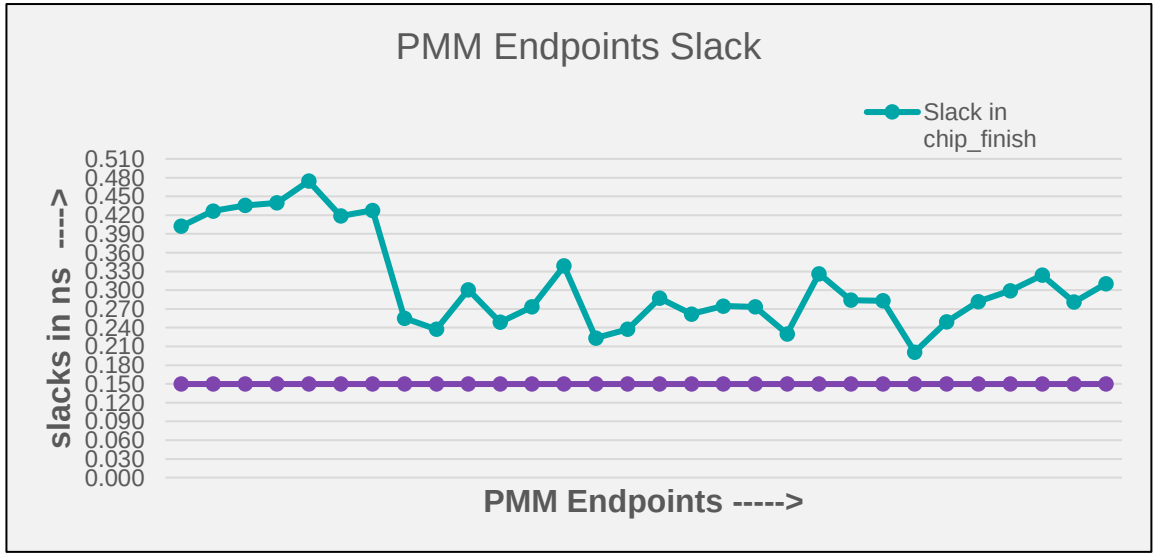
Implementation Challenges and Solutions

- **Logic Equivalence check (LEC) with PMM**
  - In case of PMM implementation, we need to follow different approach for LEC check as in post CTS stage we intentionally make changes in PMU connections
  - In PMM implemented netlist, we need to follow three-step approach for LEC run
    1. RTL vs Pre PMU stitched netlist
    2. Pre PMU stitched netlist vs post PMU stitched netlist
    3. Post PMU stitched netlist vs post route netlist



Implementation Challenges and Solutions

- **PMM endpoints slack change in optimization**
  - Considering TAT, all signoff scenarios are not enabled in post clock optimization
  - If any Vmin corner is not enabled in post clock optimization where PMU stitching happened, and enabled in route stage, it may change the endpoint slack
  - We faced a similar situation where some vmin corner was not enabled in post CTS and enabled after PMU stitching. So PMU endpoints have been over optimized and slack became highly positive (beyond the measurable range of 150ps)
  - In such case we need to do eco and reduce the end points slack to measurable range.



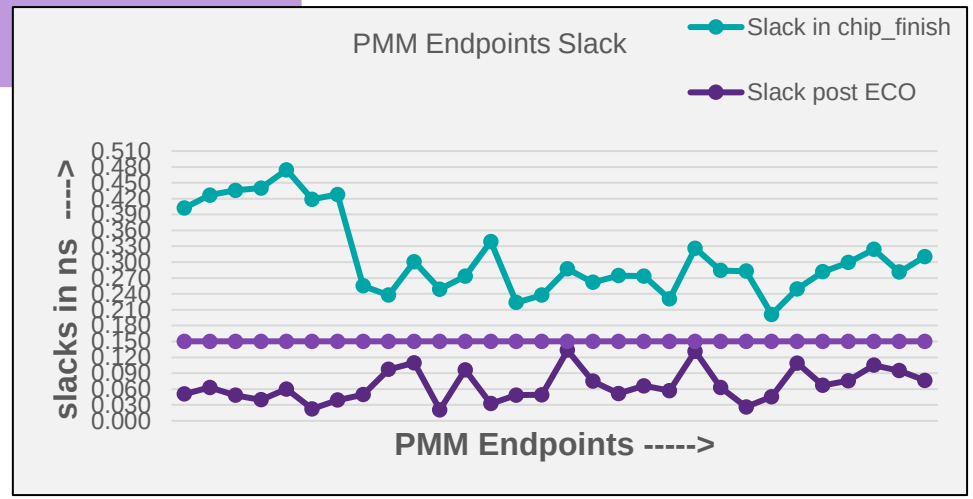
PMM END POINTS	Chip finish slack	Max acceptable slack
PMM_EP_1	0.402	0.150
PMM_EP_2	0.427	0.150
PMM_EP_3	0.436	0.150
PMM_EP_4	0.440	0.150
PMM_EP_5	0.474	0.150
PMM_EP_6	0.419	0.150
PMM_EP_7	0.428	0.150
PMM_EP_8	0.255	0.150
PMM_EP_9	0.238	0.150
PMM_EP_10	0.301	0.150
PMM_EP_11	0.249	0.150
PMM_EP_12	0.273	0.150
PMM_EP_13	0.339	0.150
PMM_EP_14	0.224	0.150
PMM_EP_15	0.237	0.150
PMM_EP_16	0.287	0.150
PMM_EP_17	0.262	0.150
PMM_EP_18	0.275	0.150
PMM_EP_19	0.273	0.150
PMM_EP_20	0.230	0.150
PMM_EP_21	0.326	0.150
PMM_EP_22	0.284	0.150
PMM_EP_23	0.283	0.150
PMM_EP_24	0.201	0.150
PMM_EP_25	0.249	0.150
PMM_EP_26	0.282	0.150
PMM_EP_27	0.299	0.150
PMM_EP_28	0.324	0.150
PMM_EP_29	0.281	0.150
PMM_EP_30	0.310	0.150

Implementation Challenges and Solutions

- **Slack fixing techniques**
  - To avoid such situation all the corners must be enabled where the PMM Stitching is done
  - Reducing the PMM endpoints slack in eco stage is challenging, However, our algorithm helps in reducing the slack without any manual fixes

TABLE 1: ALGORITHM FOR OPTIMIZING PMM SLACK

```
for i in *func_path* do {for all the functional paths for
which slack needs to be reduced}
     find first MUX which connects to both functional and
PMU path
     Check setup slack {At MUX input }
     Find delta {delta = Actual slack - desired slack}
     for j in (delta / unit_delay) do {delay of a buffer
of delay cell}
          insert buffer at input of the MUX
     end for
end for
```



PMM END POINTS	Chip_finish slack	Max acceptable slack	post ECO slack
PMM_EP_1	0.402	0.150	0.051
PMM_EP_2	0.427	0.150	0.063
PMM_EP_3	0.436	0.150	0.048
PMM_EP_4	0.440	0.150	0.040
PMM_EP_5	0.474	0.150	0.060
PMM_EP_6	0.419	0.150	0.022
PMM_EP_7	0.428	0.150	0.039
PMM_EP_8	0.255	0.150	0.049
PMM_EP_9	0.238	0.150	0.097
PMM_EP_10	0.301	0.150	0.109
PMM_EP_11	0.249	0.150	0.021
PMM_EP_12	0.273	0.150	0.096
PMM_EP_13	0.339	0.150	0.032
PMM_EP_14	0.224	0.150	0.049
PMM_EP_15	0.237	0.150	0.049
PMM_EP_16	0.287	0.150	0.133
PMM_EP_17	0.262	0.150	0.075
PMM_EP_18	0.275	0.150	0.051
PMM_EP_19	0.273	0.150	0.066
PMM_EP_20	0.230	0.150	0.057
PMM_EP_21	0.326	0.150	0.131
PMM_EP_22	0.284	0.150	0.063
PMM_EP_23	0.283	0.150	0.026
PMM_EP_24	0.201	0.150	0.046
PMM_EP_25	0.249	0.150	0.109
PMM_EP_26	0.282	0.150	0.067
PMM_EP_27	0.299	0.150	0.076
PMM_EP_28	0.324	0.150	0.105
PMM_EP_29	0.281	0.150	0.095
PMM_EP_30	0.310	0.150	0.076

Implementation Challenges and Solutions

- **PMM timing ECO flow**
  - Considering the PMU as a hard logic and PMC as a soft logic, ECO flow need to be customized
  - We developed an algorithm for quick eco implementation with below 3 goals
    - Mapping Hard IP endpoints to Soft IP pins
    - Quick timing fixes
    - Less or no manual fixing

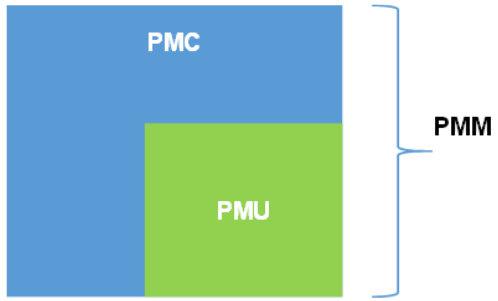


TABLE 2 : ALGORITHM FOR TIMING CLOSURE

ALGORITHM1: PSEUDOCODE FOR TIMING FIXING

```
set_dont_touch *PMU* {all PMU endpoint is set don't touch}
for i in *PMM_u* do {violating endpoint instance from timing report or
PT session}
    first instance at PMC-PMU boundary from violating PMM endpoint
    Check hold (setup) margin for setup (hold) fixing
    If margin > threshold and VT (insts) == High do
        HVT → LVT {Swap to lower VT}
        Repeat step 4 until timing met
    else
        Find next instance and repeat
    end if
end for
```

Summary

Summary

- PMM **ensures the reliability** of SoC, **avoids** sudden chip failures/malfunction
- PMM can also be used in **manufacturing test** which helps to improve the timing model of a technology node
- **PMM** is an IP, developed by **Synopsys Inc.** Which can continuously monitor the available **path margins** while chip is in operation
- PMC and PMU are two main part of PMM, PMC is soft IP whereas PMU is a hard IP
- While Implementing PMM, we need to take care of some guidelines for PMU placement, stitching and flattening
- Galaxy Constraint Analyzer (GCA) tool has helped to identify the bottleneck to improve the SDC quality
- During LEC, we followed a three-step check approach
- In ECO, ensuring the changes are made in PMC only not the PMU cells has helped in successful implementation

SAMSUNG snug

THANK YOU

Our
Technology,
Your
Innovation™

Q&A