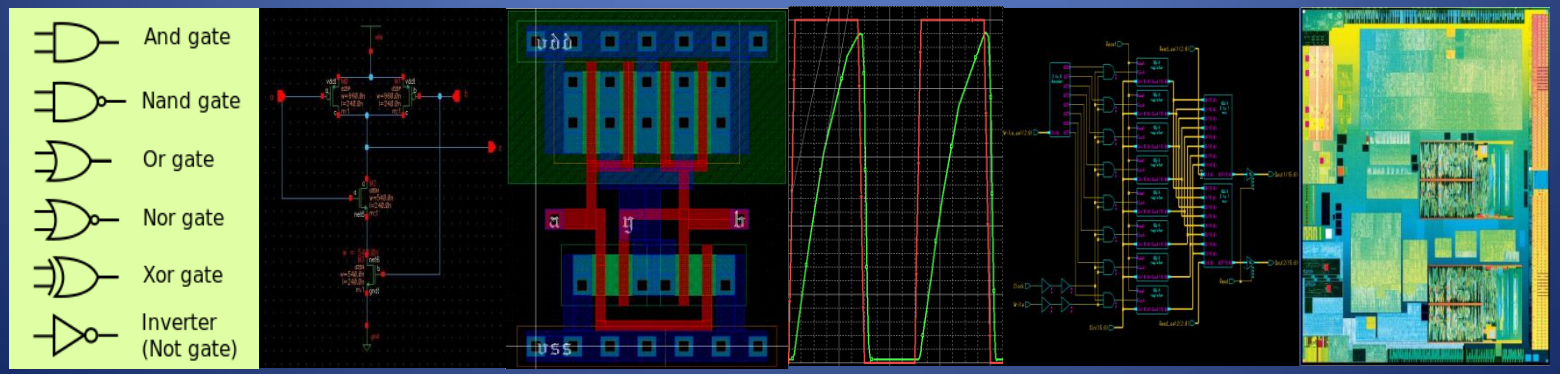
Std cell library: a key TTM enabler

KS Ramesh
Intel Corporation



Designing with leading edge process tech at Intel

- Have unbridled access to the latest process technology
- Can tailor your design to process and vice versa, since you control both!

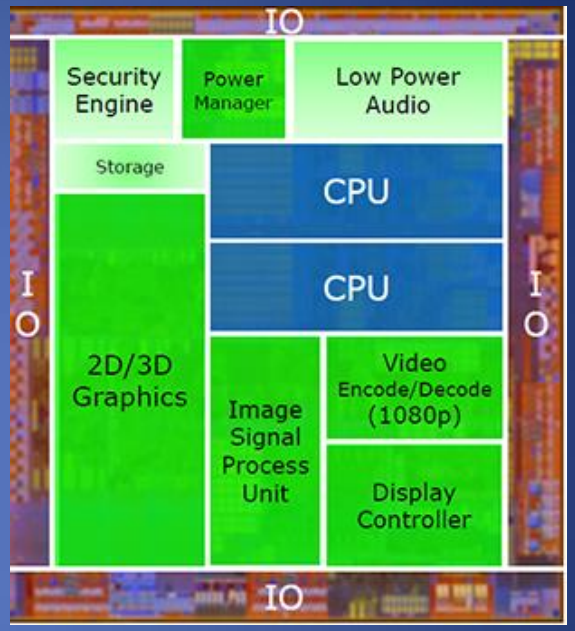
- Does this make life easier or harder for design teams ?

  - Product design runs in parallel with Process development
  - We do not always have a stable and mature process to start with compared to most others
  - Product and process targets change mid-course while lifecycle and TTM are shrinking for the industry at large
  -

- How can library help deliver good yielding Si products taking advantage of and maintaining our process lead?

STD CELL LIB requirements

- Both high perf CPU and SoC designs include IP's operating at:
  - Multiple fixed V power planes
  - Dynamic range of V (core turbo)
  - Wide range of temperature
  - Different corners on the same process node
- IP's need verification across all targeted segments/corners and PVT
  - Often, these change mid-course during project execution
  - People at and outside Intel have used MCMM for this
- As a customer, ask is always the same:
  - Ability to (re)generate accurate library collateral in the shortest possible time!
  - It takes too long! (10K cells, 4 PVT, 4 wks)



Library development

- Content: standard cells for digital, analog, cache
  - jointly developed by Design Technology and FAB Adv Design teams
  - optimized specifically for Intel manufacturing processes to extract max performance and yield.

- Engine: A full-featured internal library characterization tool which generates timing, power and RV model files.
  - Generates tool views for Synopsys' Design Compiler, Intel STA tool, Primetime, Cadence, Mentor Graphics' ATPG, SALT, Caliber, FEV/Conformal, Verilog simulators, etc.

Why does it take so long?

- Extraction, Initialization, cell simulation, model generation, validation and regression.
- Init/Sim are accuracy-runtime decisions
  - WC (dynamic vector initialization) vs static user defined
  - i/p waveform (pwl @intel vs continuous) – impacts specific topologies like pg/dcn
  - Container/zone model for wc extraction
  - #Points in 2D matrix (slope, Cmax)
  - MIS for max/min
  - Separate model for setup/hold
- Breadth of tool views needed -> validation time
- Repeat all of the above for each PVT corner :
  - V and T flip many init vectors

Statistical modelling

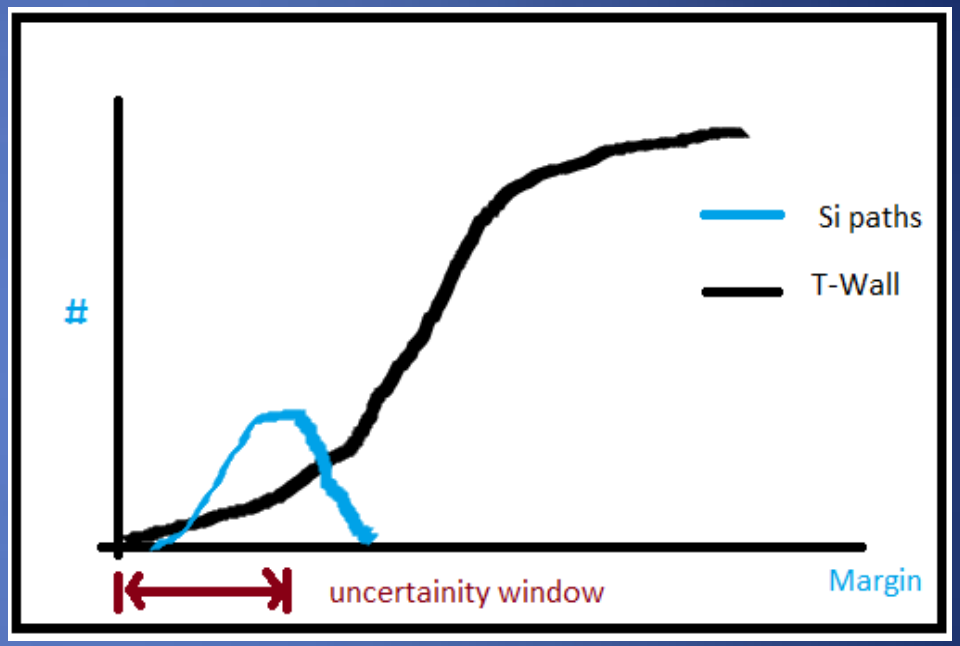
We do statistical modelling of variation effects.

- **Advanced On-Chip Variation Modeling**
  - For SOC and PT-based HIPs
  - 2-D Tables for delay de-rate factor for setup and hold as function of *#of stages* and *spatial separation*
  - Stage-count for std. cell and device level info is included as part of the collaterals
  - Could be MonteCarlo or Nova/MPP based algorithm
  - Users: PT and ICC

- **Le/Vt adjustments to std cell devices based on statistical failure models**
  - Used for our custom designs with internal STA tool
  - Can be different for setup/hold
  - Can be edge and device specific

Design and Si

- PV Model has not been a good indicator of silicon speed paths.
- Majority of speed paths do not come from the PV model convergence tail.
- Just pushing the PV convergence target alone is not a sound way to maximize silicon frequency.
- There is an *uncertainty window* around which most silicon speed paths are found.
- Goal is to augment PV model with silicon learnings and carry knowledge forward to future projects.



Sources of PV-Si miscorrelation?

- Transistor size (z) related Si miscorrelation
- Transistor layout topology induced differences in finfet performance
- Dynamic V droop
- Sensitive Cell topologies (gate i/p vs diff i/p)
- Cell loading (lumped vs distributed(pi))
- RC dominated vs device dominated load
- Changes in pn skew pre to post Si
- Aging
- Cmax – variation limits Vmin

Work around or make it faster?

- Workarounds used (runtime-accuracy tradeoff?)
  - Simulation step size
  - Apply generic vs cell specific Guardbands
  - Use RC scalars to cover for device ‘V’ scaling
  - Use MIS where needed – high vs low freq/perf
  - Bypass power modelling?
  - Flat scaling the timing models

But:

- Can we come up with a RSM that takes V and T as parameters to cut setup/simulation time?
- Would really like a smart algorithm for input vector initialization across PVT.
- Abstracting more parameters into STA (AOCVM)?
- Consistent language between lib, STA, and Spice!