SoC Design & It's Challenges

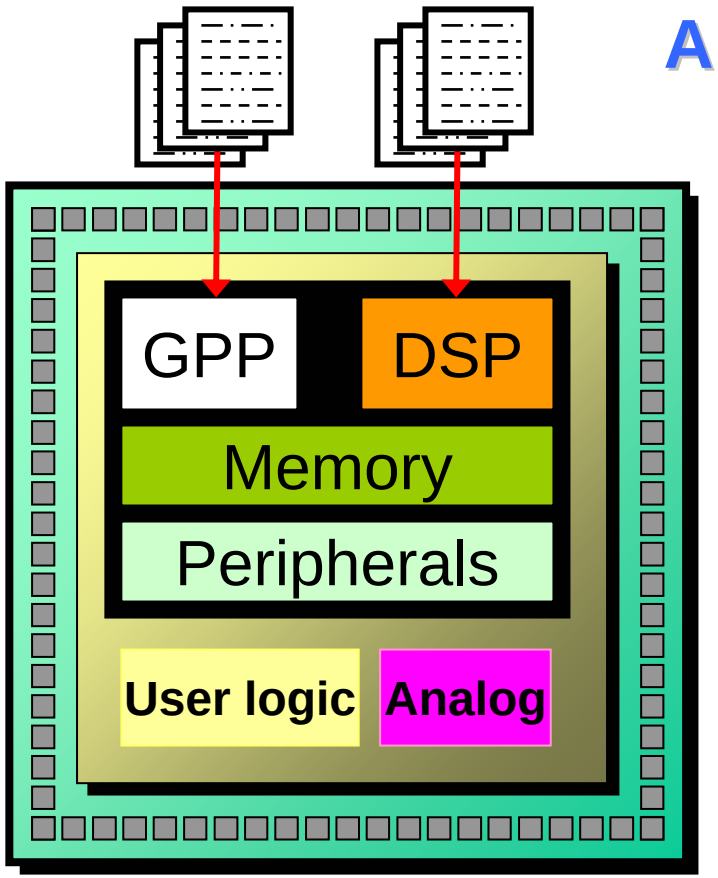
Atul Jain

November 11, 2004

Outline

- What is SoC?
- Why SoCs?
- Why SoC Methodology?
- Design Challenges
- Summary

What is System on a Chip (SoC)?



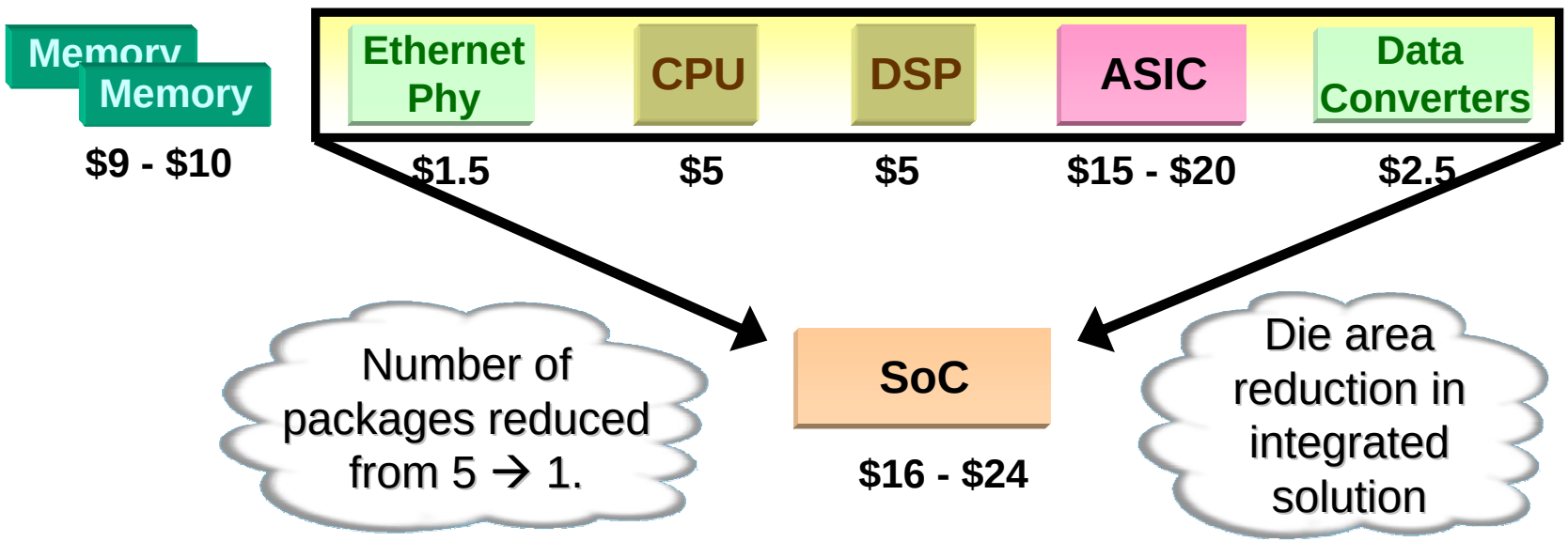
Application Specific Chip with:

- Embedded hard/soft/firm processor cores
- Embedded memories
- Multiple peripherals
- Modules interconnected via "standard" busses
- User logic 500K+ gates
- Analog Modules (Phys, ADC, DAC)
- Application S/W for the processor core(s)

Design Reuse a MUST for SoC success

Why SoCs: Potential chip-level savings?

- A typical broadband application consists of following chips: DSP, CPU, Data Converters, ASIC/FPGA (Peripherals and Custom Logic), Ethernet PHY, and Memories.



Typical chip-level Savings of 30% - 45% using SoC!!!

Why SoCs?

- Advanced technologies enabling more integration on a chip ➔ Reduced Product Cost.
- Physical size of products shrinking ➔ Minimize number of parts on a board.
- Consumer electronics requiring low cost products ➔ More integration.
- Minimize number of silicon vendors for a product ➔ Reduced Business Cost.
- Improved Reliability.

Why SoC Methodology?

- Time-to-Market extremely important for business.
- Easy integration of Intellectual Properties (IPs) from multiple sources.
- Resource management
  - Minimize number of Resources Required to complete a design.
  - Efficiently manage resources across projects.
  - Share domain expertise across multi-site design teams.
  - Make effective use of design automation.
- Advancement in technology ➔ Handle Increased Complexity of Integration.
- Maximize Performance.
- Manage Feature Creep and Engineering Changes quickly.

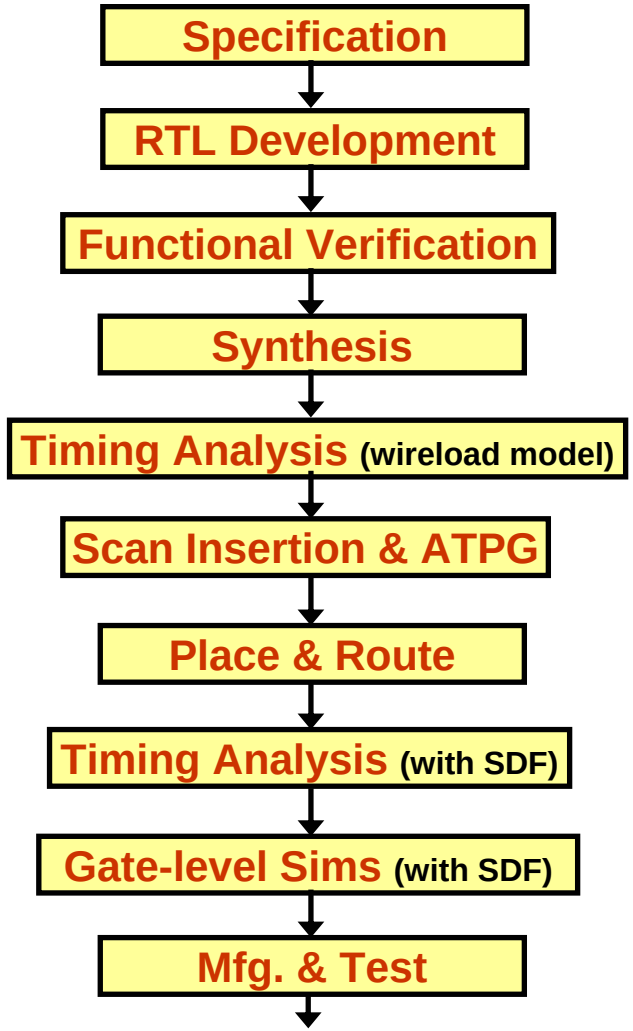
Typical Methodologies

- Waterfall Model*
- Spiral Model*
- Construct by Correction*

* *Source*: "Reuse Methodology Manual For System-On-a-Chip Designs," Michael Keating and Pierre Bricaud, Kluwer Academic Publishers.

Waterfall Model

- Based on traditional ASIC flow
- Serial design flow, design transition phases in a step function.
- Assumes all steps are Perfect and NO need to revisit a completed design phase.
- Specification is Golden, NO changes permitted without having a major impact on design schedule.
- Expects clean handoffs from one team to the next.
- Works well for ~1M gate designs.
- Not suitable for large complex **D**eep **S**ub**M**icron (**DSM**) designs.



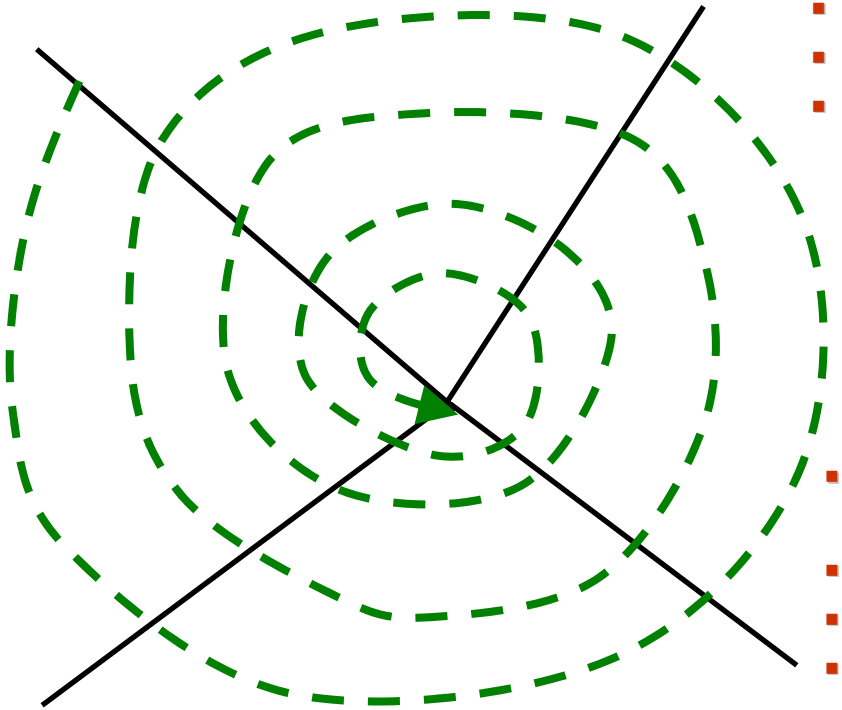
Spiral Development Model

RTL & Verification

- Architecture
- Block Partitioning
- RTL Coding
- Module & Chip-level verification
- DFT Planning

Software

- Application Prototype
- Prototype testing
- Application development
- Application testing



Physical Design

- Library selection
- Package selection
- Floorplan
- Clock expansion
- Placement
- Routing
- Signal Integrity
- IR Drop Analysis

Synthesis & Timing

- Constraints and exceptions definition
- Block-level Synthesis
- Top-level Synthesis
- Scan Insertion / ATPG
- Static Timing Analysis

Construct by Correction

- Used for development of UltraSPARC.
- Single team took design from Architectural definition through Place and Route.
- Engineers had to learn required tools.
- Team understood impact of architectural decisions on Area, Power, and Performance of final design.
- Planned multiple passes from architecture to layout to learn more about design.
- Multiple iterations allowed the team to learn from their mistakes and correct them in next iteration.
- UltraSPARC development project was one of the most successful in Sun Microsystems history.

Design Start

- Design Specification
- Library selection
  - Target technology node
  - Frequency and Power limits
  - Layout density entitlement
- IP Selection
  - **Hard** vs. **Soft** Macro
  - **Internal** vs. **external** IP
  - Embedded RAMs/ROMs
  - PLLs & Analog blocks
  - Custom Cells (if needed)
- Chip-level Power estimation

Design Start (contd…)

- Selection of verification environment.
- Selection of configuration management environment.
- IO Selection.
- Package Selection.
- Die size estimate tradeoffs
  - Preliminary floorplan
  - **IO limited** vs. **Core limited** vs. **Megamodule** limited vs. **Package** limited

Challenges for DSM designs

- Reusable IP deliverables.
- Design prototyping & partitioning.
- Process variations impact on timing analysis.
- Cross talk noise.
- Voltage drop & power management.
- Design management.

Design Planning essential for all of the above.

Hard IP

- Need to support the following models:
  - Functional
  - Timing
  - Synthesis
  - DFT
  - Floorplanning
  - Physical Design
  - Software tools for processor core
- Scan hookup requirements
- Clock insertion delay numbers and skew requirements
- Power requirements
- Integration/Test documentation

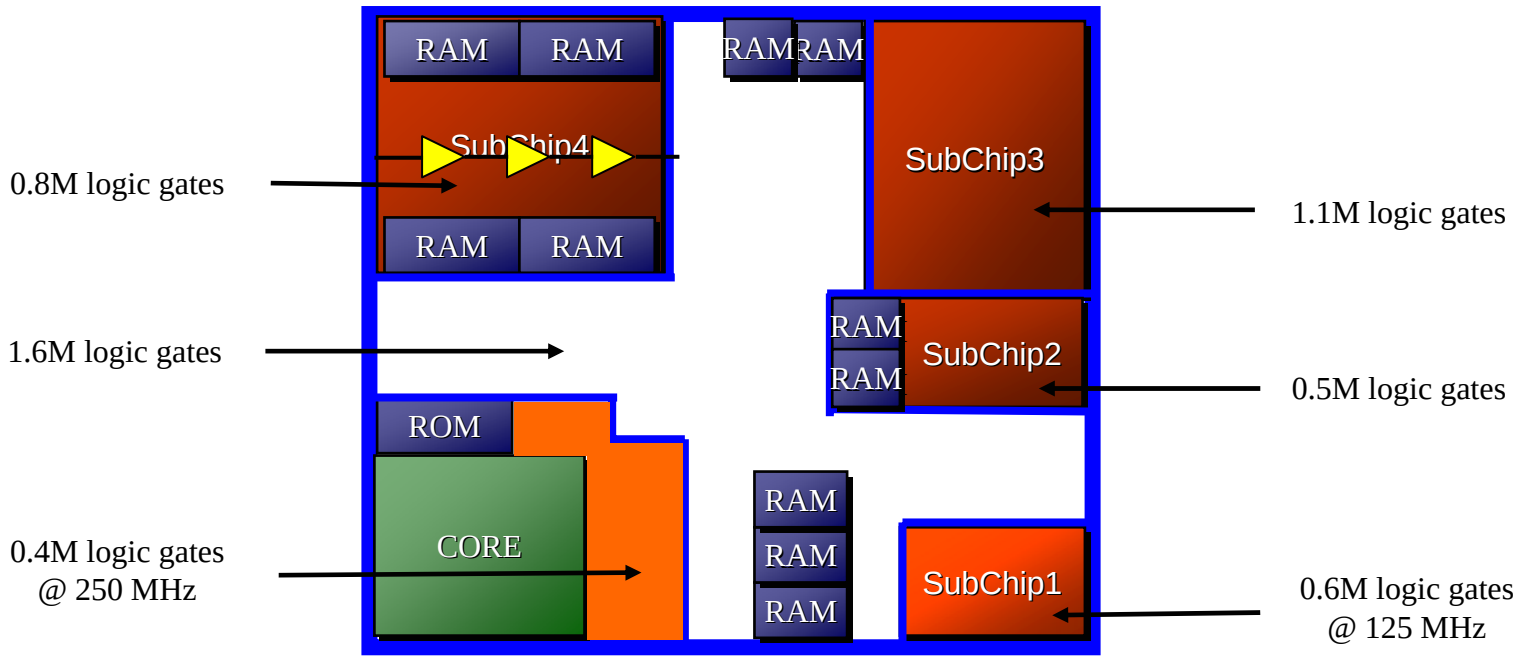
Soft IP

- Synthesizable RTL
- Functional Verification
  - Coverage metrics
  - models to speed-up verification
- Synthesis / Timing Analysis
  - Timing exceptions and constraints
  - Clock and Reset requirements
  - Critical paths should be documented
- DFT scripts
- Physical design
  - Memory placement guidelines
  - Special clock handling requirements
- Software tools for processor cores
- Implementation and integration documentation

Design Prototyping & Partitioning

- **Hardware** vs. **Software** tradeoffs
- **Prototyping** – "Early" assessment of design planning decisions
- Partitioning designs into subchips or hierarchical modules
- **Flat** vs. **Physical hierarchy** decisions
- Partitioning is needed to
  - Meet **Area**, **Timing**, and **Power** budgets.
  - Minimize top-level logic and routes.

Design Partitioning: Example



"**Divide and Conquer**" approach for partitioning the design into several smaller blocks.

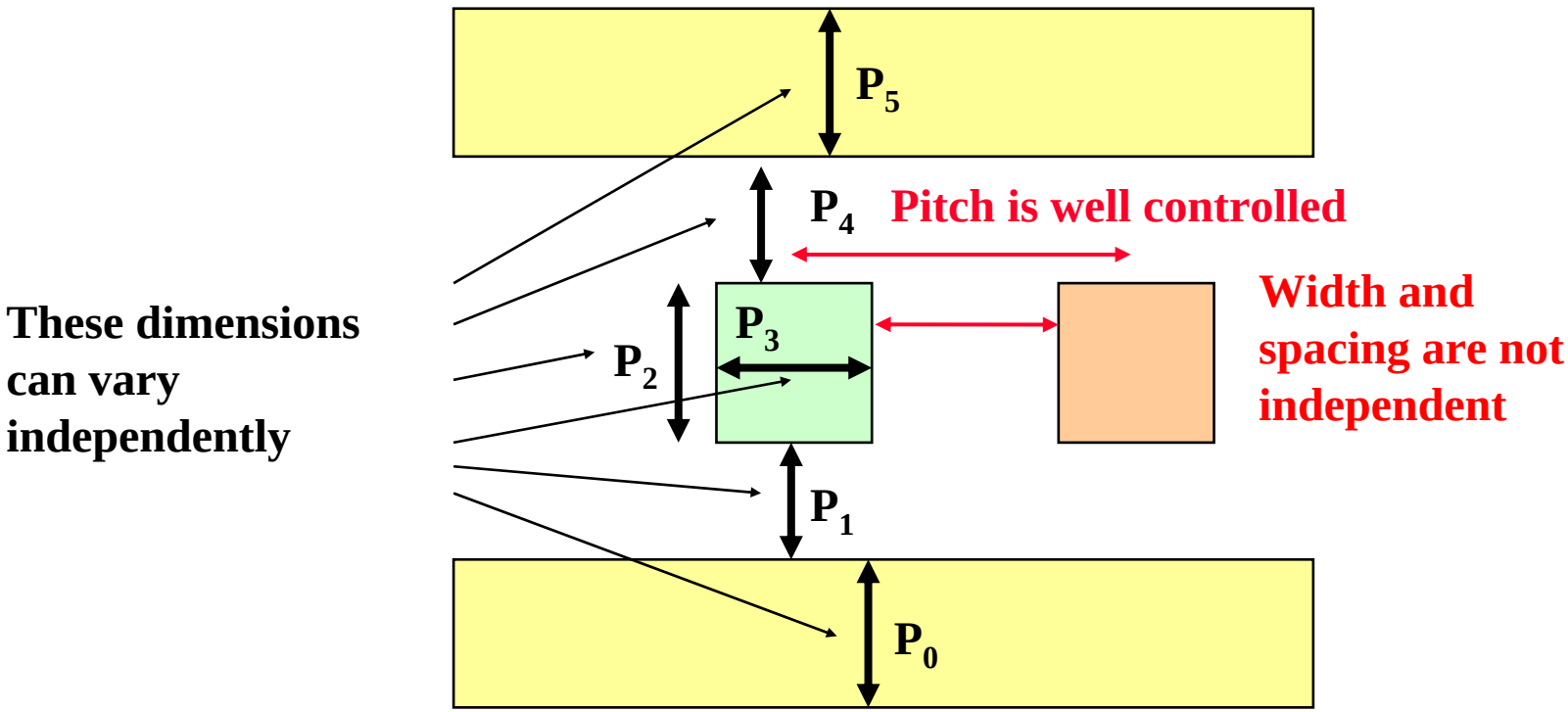
Timing Analysis for Process Variability

- Typical observations in DSM and sub-nanometer regime:
  - Parameters (L_{eff}, V_T, T_{ox}, etc.) across chips (inter-die) are different.
  - Parameters within the same chip (intra-die) are also different.
  - Why?
    - Feature size is getting smaller than "Lithography Wavelength" ➔ Some mask information is lost.
    - Critical dimensions are scaling faster ➔ Less Control.
    - **B**ack **E**nd **O**f **L**ine (BEOL) variability.
    - **A**cross **C**hip **L**ine-width **V**ariations (ACLV).
    - Across-the-chip temperature and VDD variation.
- Designers and Business goals require as many chips as possible to "work".

"Nano-CMOS Circuit and Physical Design," Ban Wong, Anurag Mittal, Yu Cao, Greg Starr, ISBN:0-471-46610-7

Interconnect variation - Net delays, Path delays are f(P)

- $C_{NET} = f_{cap}(P_0, P_1, P_2, \ldots)$
- $DELAY_{NET} = f_{net}(P_0, P_1, P_2, \ldots)$

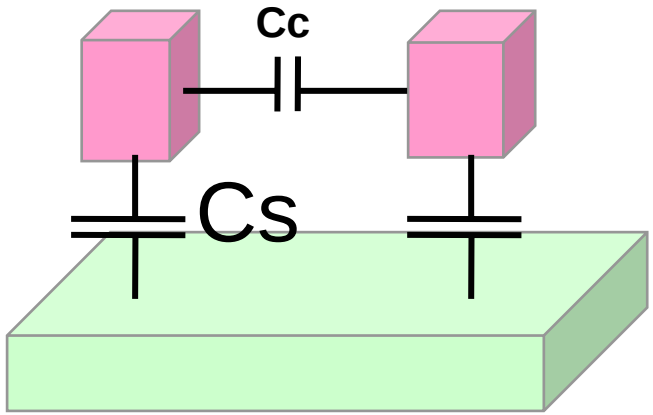


OCV Analysis

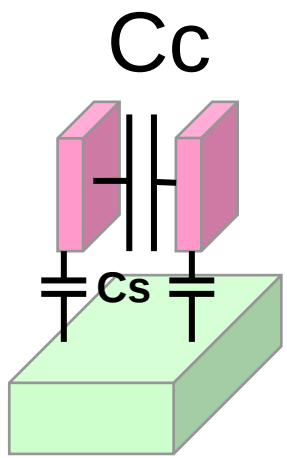
- Cannot assume constant PTV across die ➔ Essential to comprehend impact of these variations in timing analysis.
- STA tools support OCV analysis mode
  - Compute min and max delays for cells and nets by multiplying annotated delay with min and max timing de-rate value, respectively.
  - Apply min and max delays to different paths simultaneously.
  - For setup check, annotate worst case SDF. Use max delay for launch path and min delay for capture path.
  - For hold check, annotate best case SDF. Use min delay for launch path and max delay for capture path.

Crosstalk Analysis

- Wire aspect ratio changes for DSM technologies.



$\geq$ **0.5 *µm***

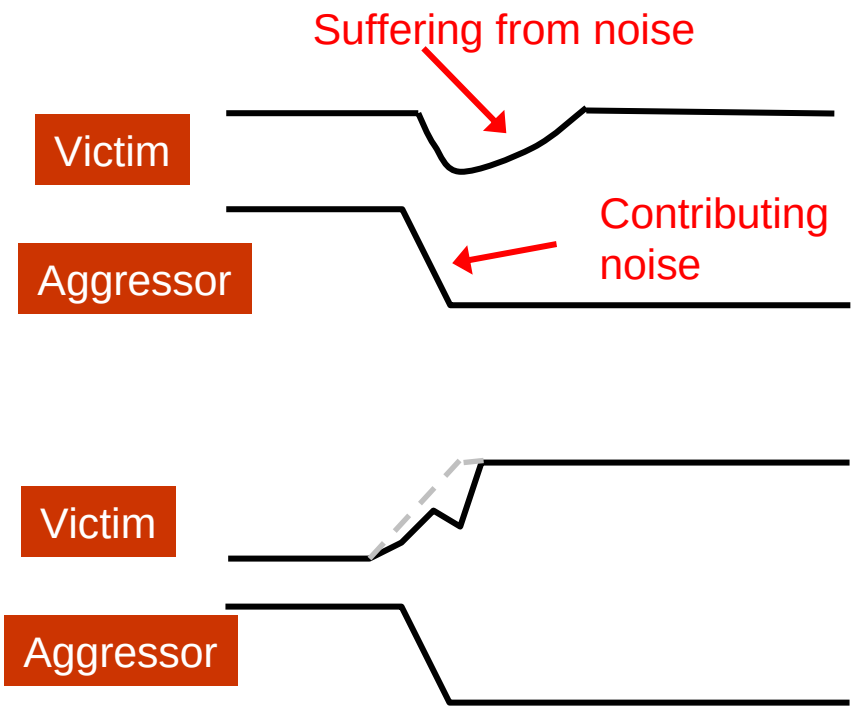


$\leq$ **0.13 *µm***

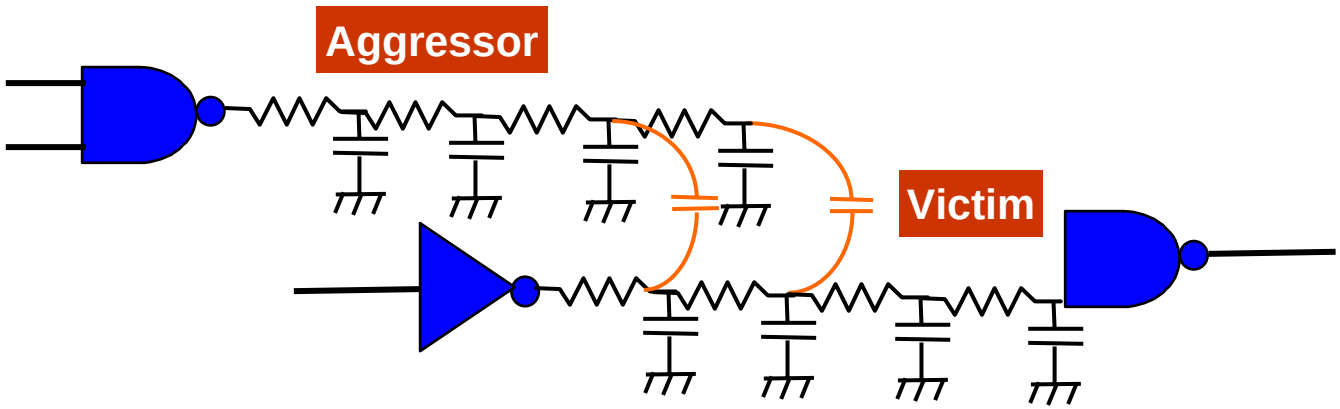
- Coupling capacitance has gone up with shrink in feature size.

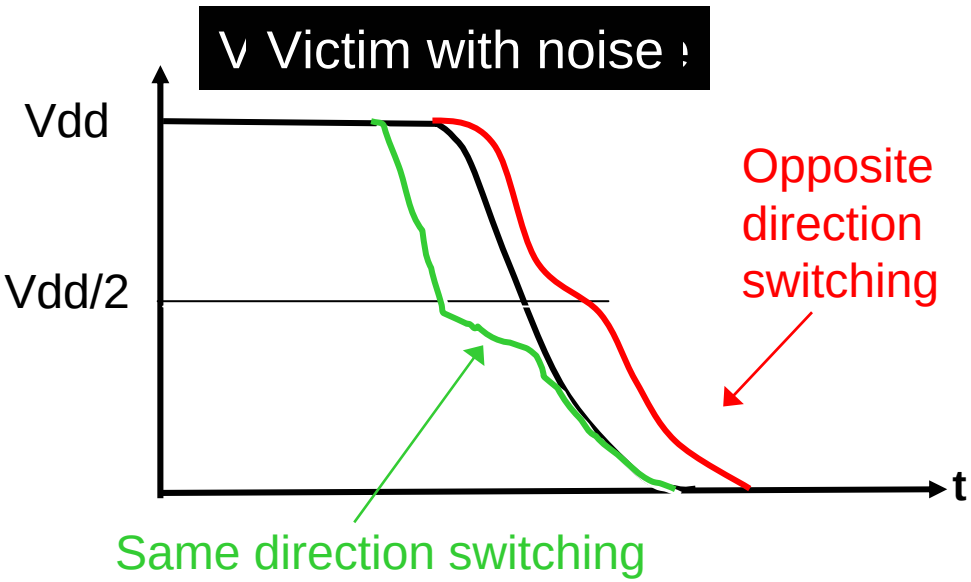
Crosstalk Effects

- Crosstalk affects the circuits in two ways: **Functionality** and **Timing.**
  - Glitch propagation problem.
  - Glitches on Clocks can be a real problem.
  - Delay Variation can be an issue.



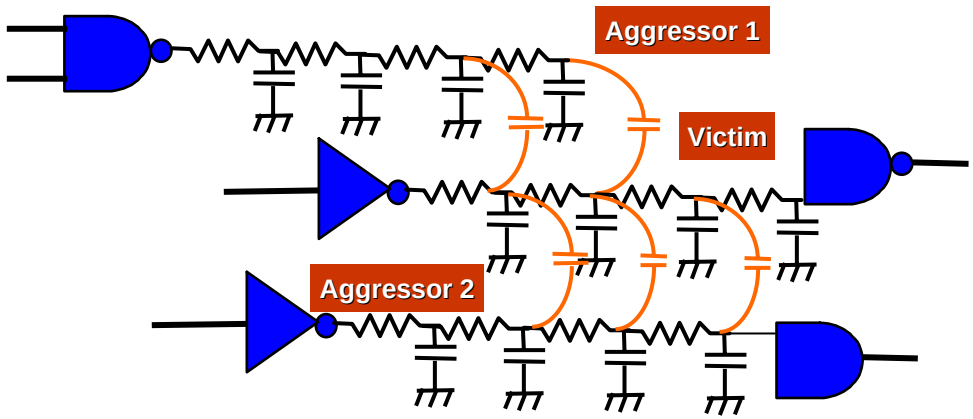
Crosstalk Delay Variation: Example



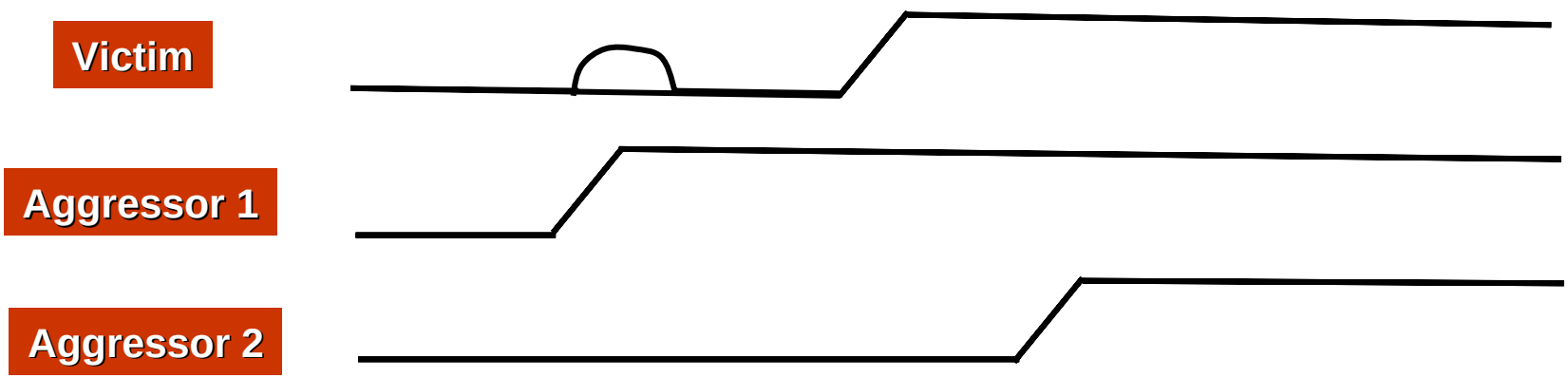


- Aggressor/Victim net switching in same direction leads to reduced signal delay through the victim net.
- Aggressor/Victim net switching in opposite direction leads to increased signal delay through the victim net.

Crosstalk Delay: Timing Window



Victim Net Delay with Crosstalk



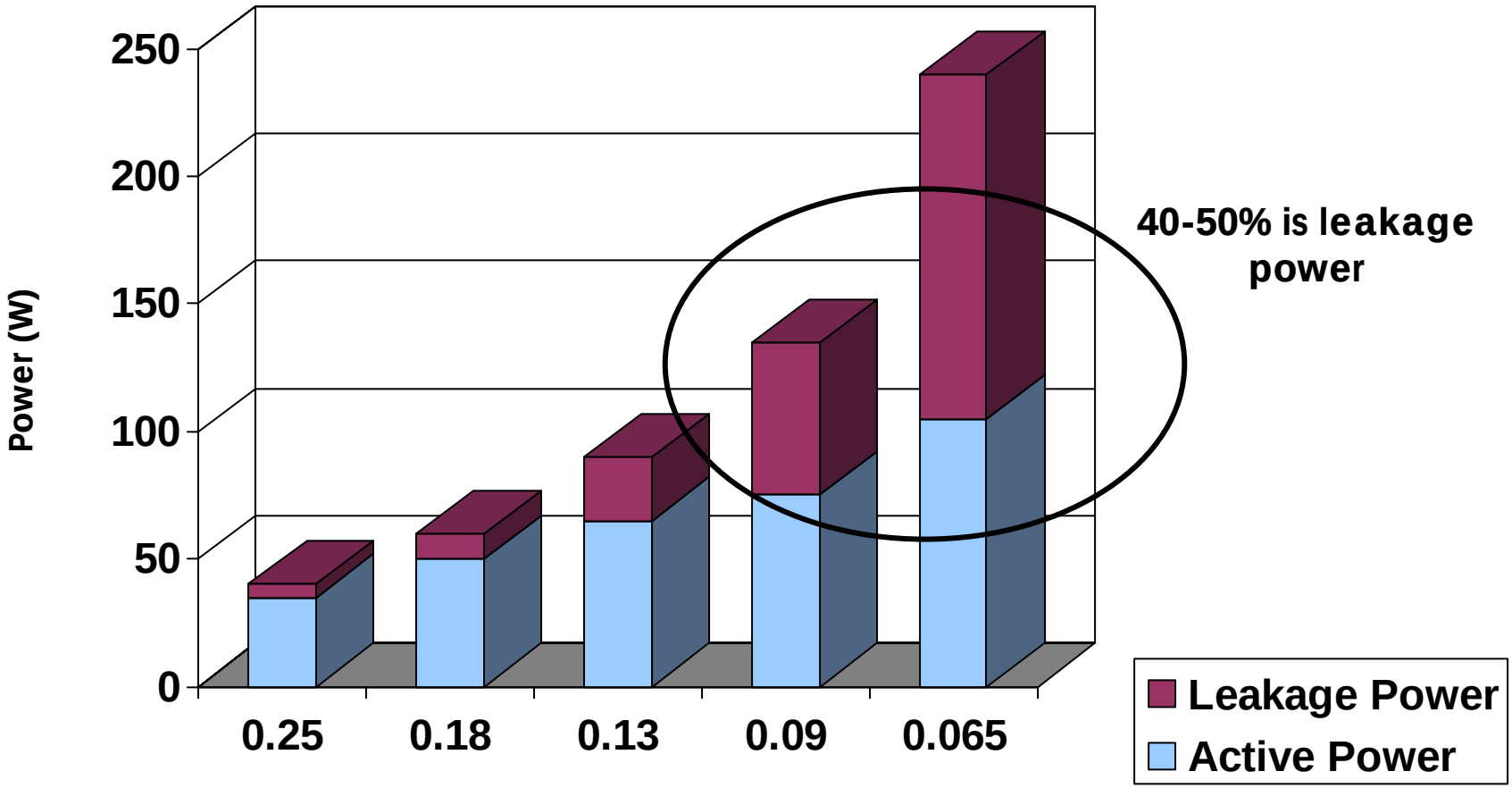
Effects of Crosstalk Delay

- Delay changes due to affect of switching aggressor net on victim net needs to be comprehended:
  - During **setup** check worst case condition
    - **capture clock** is **faster**; **data path** and **launch clock** are **delayed**.
  - During **hold** check worst case condition
    - **capture clock** is **slower**; **data path** and **launch clock** are **speedup**.

How to Manage Crosstalk?

- Ensure transition timing window for coupling nets do not overlap.
- For heavily loaded nets improve transition time.
- Minimize use of low drive cells.
- For clocks, shield to minimize coupling.
- Use Physical Design tools capabilities of avoiding crosstalk during detail route.

Power Management Trends



Power Management – Motivation

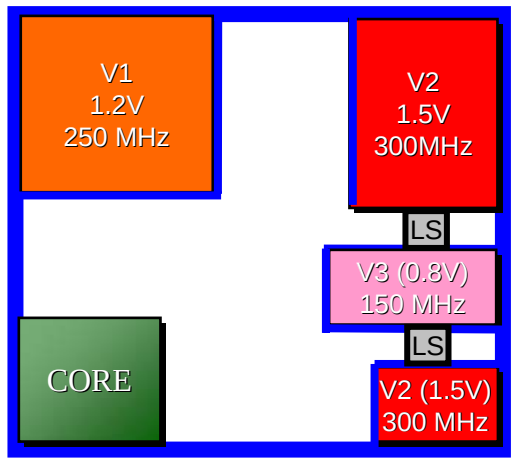
- Active or Dynamic Power
  - $P_{dynamic} = P_{switching} + P_{short\text{-}circuit}$
  - $P_{switching}$: Primarily due to actively changing states of the circuit due to constantly charging and discharging of the effective capacitive loads
    - $P_{switching}$ = **S**witching **A**ctivity x freq x C_{eff} x V_{dd}^2
    - Clock power is the largest contributor (nearly 75%)
    - Interconnect is the largest consumer
  - $P_{short\text{-}circuit}$: Momentary "crowbar" current flowing between VDD and GND when transistor stacks switch state
- Leakage or Static Power
  - Primarily contributed by source-to-drain leakage current that increases with lowering V_T (threshold) and increasing temperature.

Dynamic Power Management Techniques

- Clock Gating
  - Selectively turn OFF registers when not needed.
- Power Gating
  - Shut OFF power to blocks in stand-by mode.
  - Retention flip-flops (on an isolated power supply) can be used to save the logic state of all sequential elements when the chip is powered down.
  - Eliminates need to re-initialize the device when coming out of stand-by mode.
- Power-aware Physical design
  - Reduce capacitive loading by down-sizing gates.
  - Minimize wire length.

Voltage Islands

- Partition design into:
  - High speed blocks fed by higher voltage.
  - Low speed blocks fed by lower voltage.
- Need **L**evel **S**hifters (**LS**) between blocks operating at different voltage levels and ensure correctness of LS connectivity.



- Need **S**caleable **P**olynomial **L**ibrary (**SPM**) support
- Additional careabouts related to scan stitching and clock distribution required.

Design Management

- Configuration management
  - Paramount for Multi-site development
- Design and Scope changes
  - Feature Creep
  - Process Requirement Changes
  - Test and Reliability Requirement Changes.
- External deliverables, e.g., IPs, custom cell.
- Handoffs and milestones
  - Design Milestones, e.g., RTL freeze, Initial synthesis, tapeout.
  - Design team (or customer) handoffs.
- Optimize engineering and compute resources.

Summary

- Robust Design Methodology needed for success of SoCs.
- Design **Reuse** and **Automation** essential ingredients.
- Different methodologies followed in the industry:
  - Waterfall Model
  - Spiral Design Model
  - Construction By Correction
- SoC challenges can be handled effectively with Design Planning and Management.