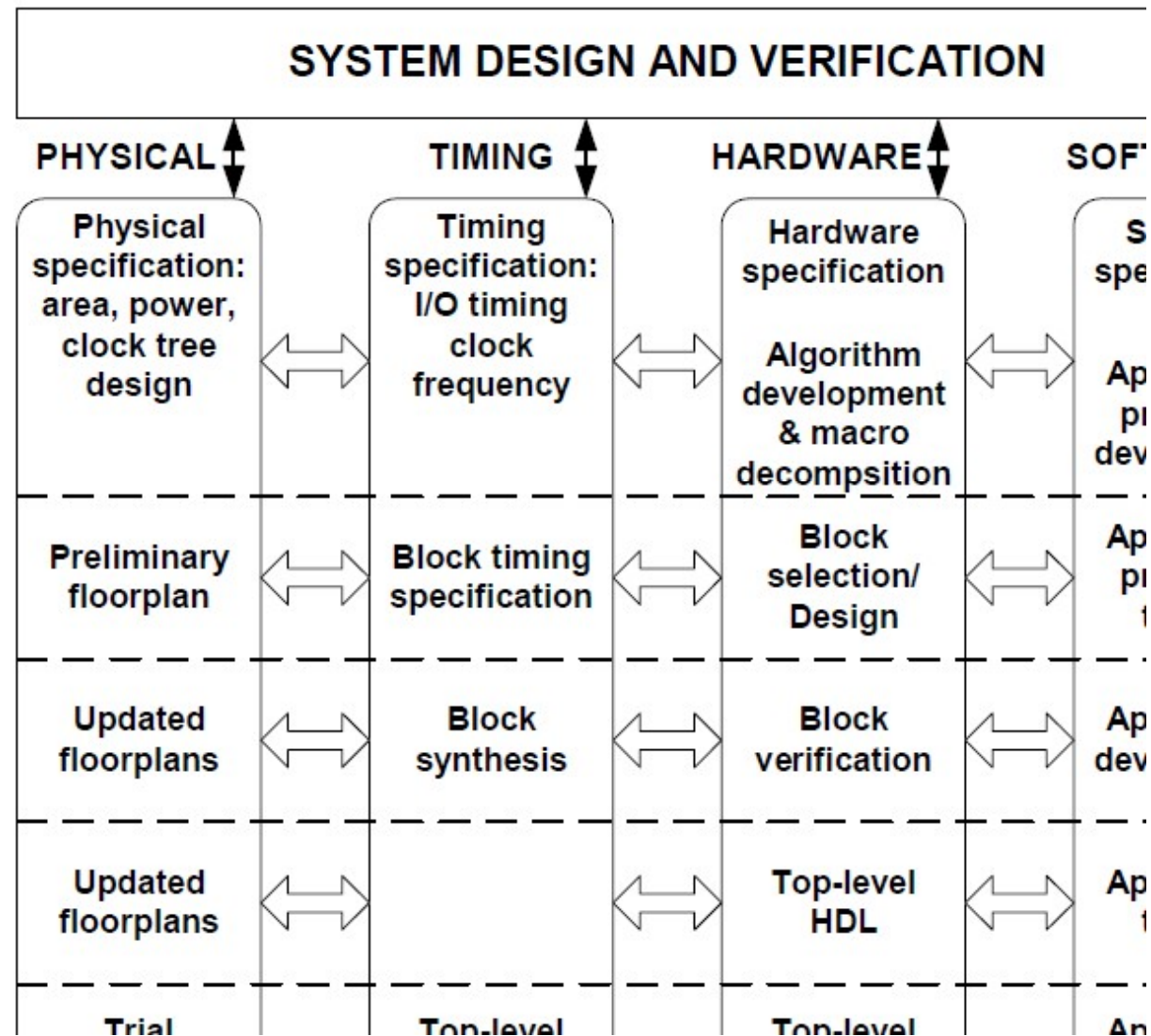




SoC MODELS

SPIRAL MODEL

Goal : Maintain parallel interacting design flows



WATERFALL V.S. SPIRAL

- Waterfall

- Work well up to 100K gate and down .5u
- Serial H/W and S/W development

- Sprial

- For large, deep submicron designs
- Parallel development of H/W & S/W
- Parallel verification and synthesis
- Floorplaning and P & R in synthesis process
- Use predesigned Macros (Hard/Soft)
- Planned iteration throughput

*“H/W and S/W development
concurrently : functionality, timing,
physical design, and verification”*



TOP-DOWN VS. BOTTOM-UP

- Classical top-down
 - Begin with spec and decomposition
 - End with integration and verification
 - Assuming lowest level block, pre-designed
 - Too ideal to be easily broken and cause unacceptable iteration
- Real-world design team
 - **Mixture** of top-down and bottom-up design
 - Building critical low-level blocks early
 - Libraries of reusable hard and soft macros helps this process



“CONSTRUCT BY CORRECTION”

- Construct by correction
 - Made the **first pass ASAP**, and refine later
 - Why
 - allow for multiple iterations
 - Used in Sun Microsystem’s UltraSPARC design methodology
 - “One of the most successful in Sun Microsystem’s History”
 - Take from architecture definition through P & R
 - Foresee impact of architectural decision on final design: area, power, performance
 - Target
 - larger, complex designs
- Correction by construct
 - Make the first pass completely right
 - Target
 - small designs



KEY TO SOC DESIGN PROCESS

- **Iteration** is an inevitable part of the design process
- The problem is **how large the loop is**
- Goal
 - **Minimize the overall design time**
- But How
 - Planned for iterations
 - Minimize iteration numbers
 - especially major loops (Spec to chip)
 - Local loop is preferred
 - coding, verifying, synthesizing small blocks
 - IP clearly help due to pre-verified
 - Parameterized blocks offer more tradeoff between area, performance and functionality
- Carefully designed **spec** is the best way to minimize the loops

