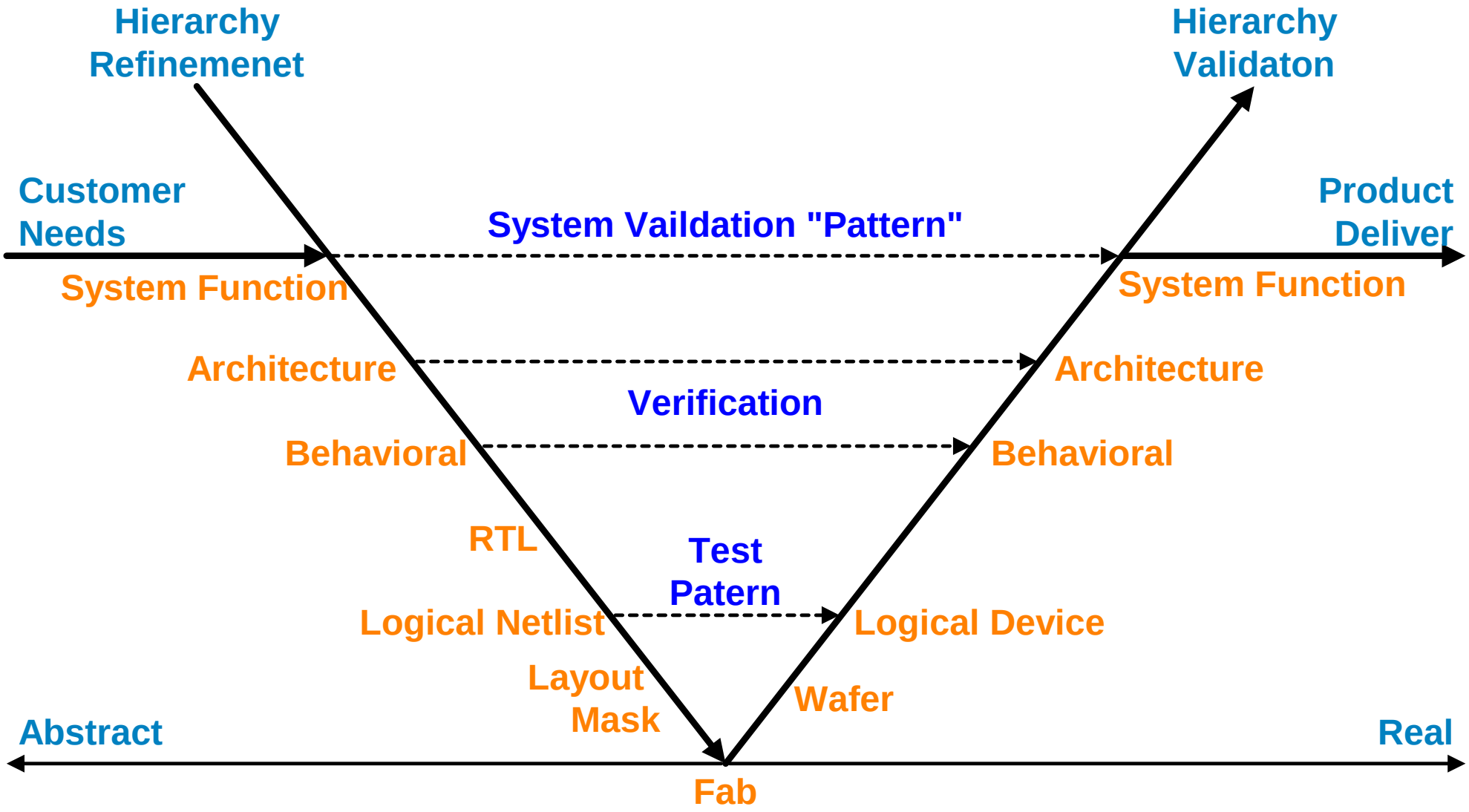
SOC Design Flow

Challenges of SoC Era

- Deign complexity
  - Validation & Verification
  - Design space exploration
  - Integration
  - Timing & power
  - Testing
  - Packaging
- Time to market
- The cost

From Requirement to Deliverables



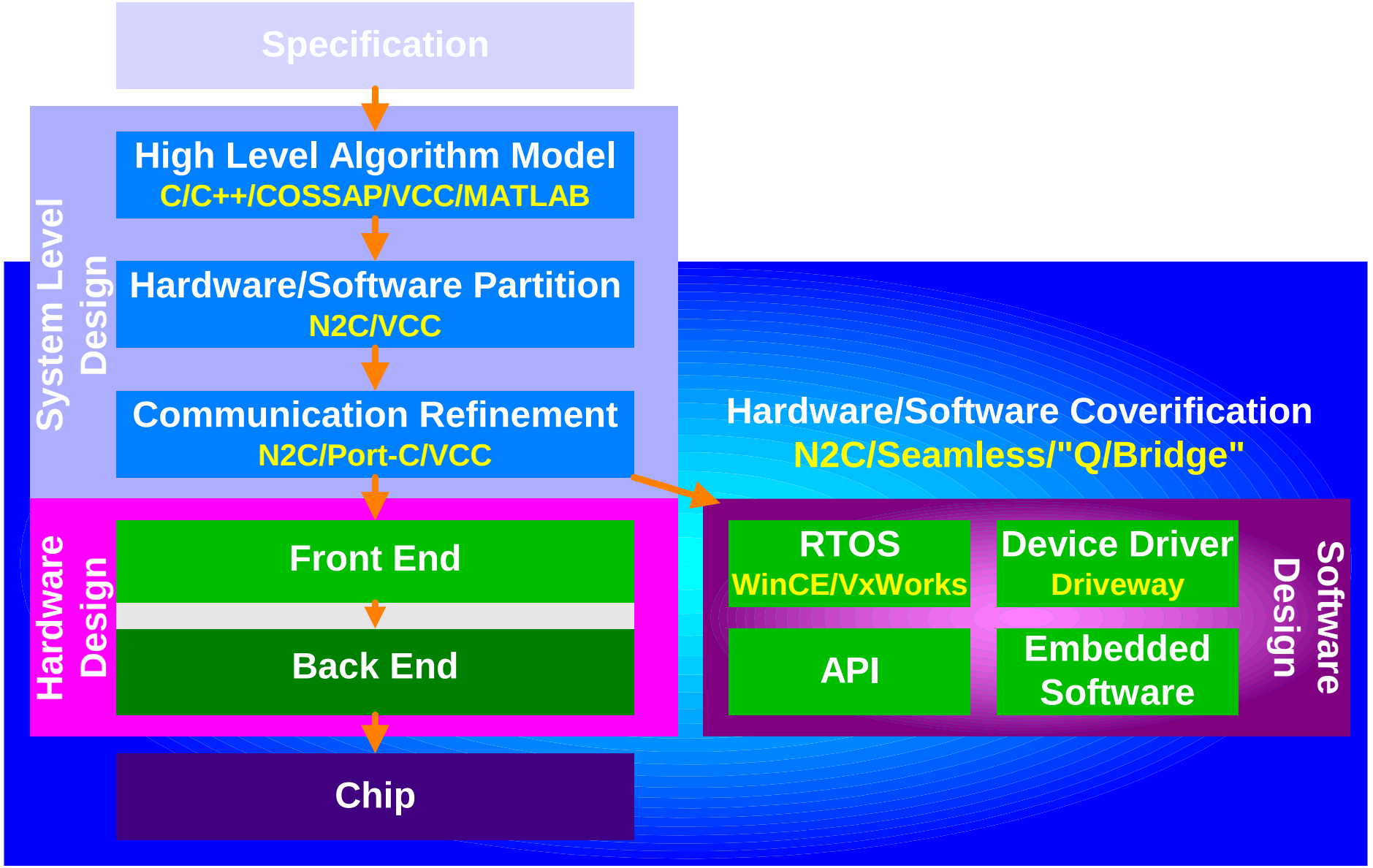
Five SoC Design Issues

- To manage the design complexity
  - Co-design
  - IP modeling
  - Timing closure
  - Signal Integrity
  - Interoperability

How to Conquer the Complexity

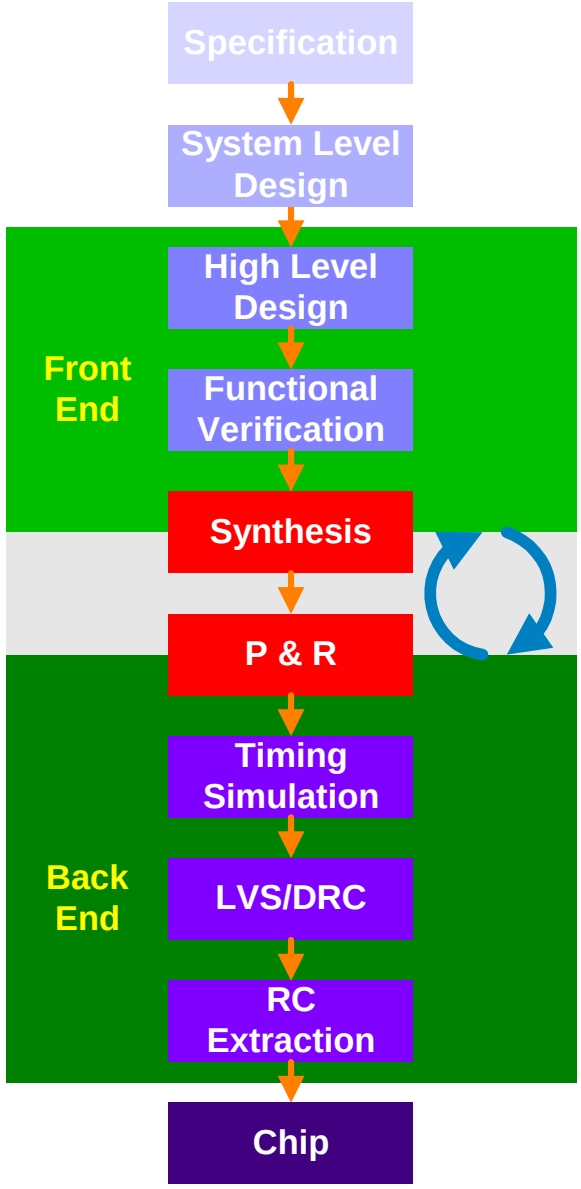
- Use a known real entity
  - A pre-designed component (reuse)
  - A platform
- Partition
  - Based on functionality
  - Hardware and software
- Modeling
  - At different level
  - Consistent and accurate

SoC Design Flow



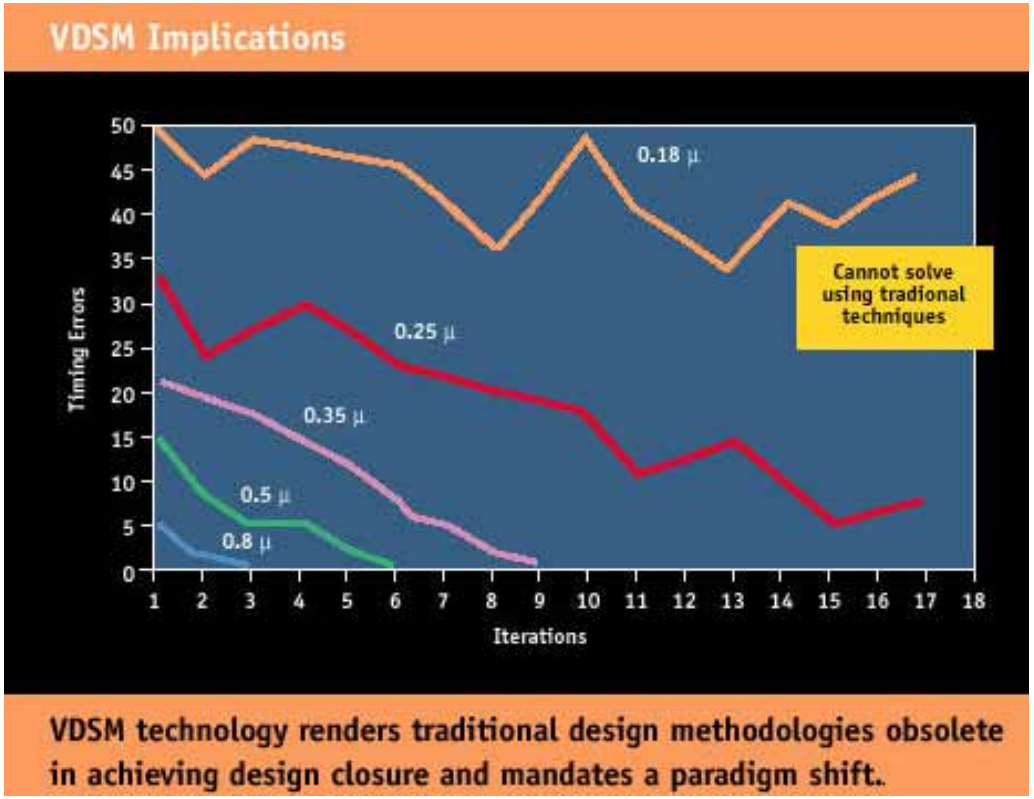
Physical Design Flow

- In VDSM
  - Interconnect dominates delay
  - Timing closure
  - Signal integrity
- Traditional design flow
  - Two-step process
  - Physical design is performed independently after logic design
- New design flow
  - Capture real technology behaviors early in the design flow
  - Break the iteration between physical design and logic design

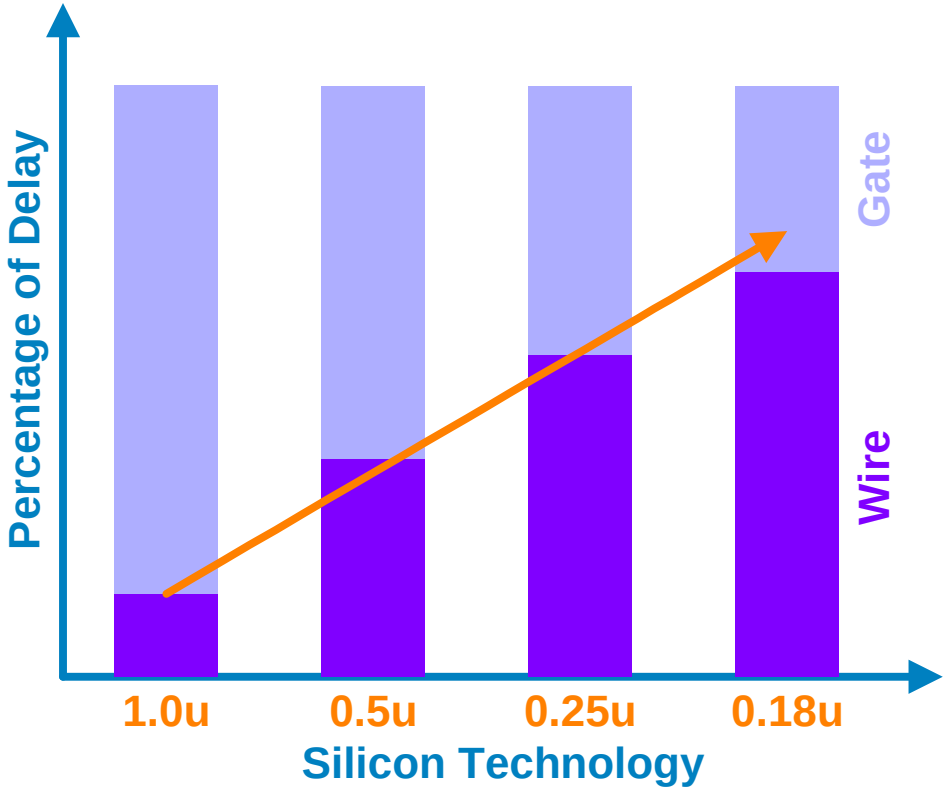


Making Sense of Interconnect

- At 0.35u, timing convergence started to become a problem.
- At 0.25u, it started to significantly impact the work of the designer.
- At 0.18u, if not accounted for, it actually causes designs to fail.



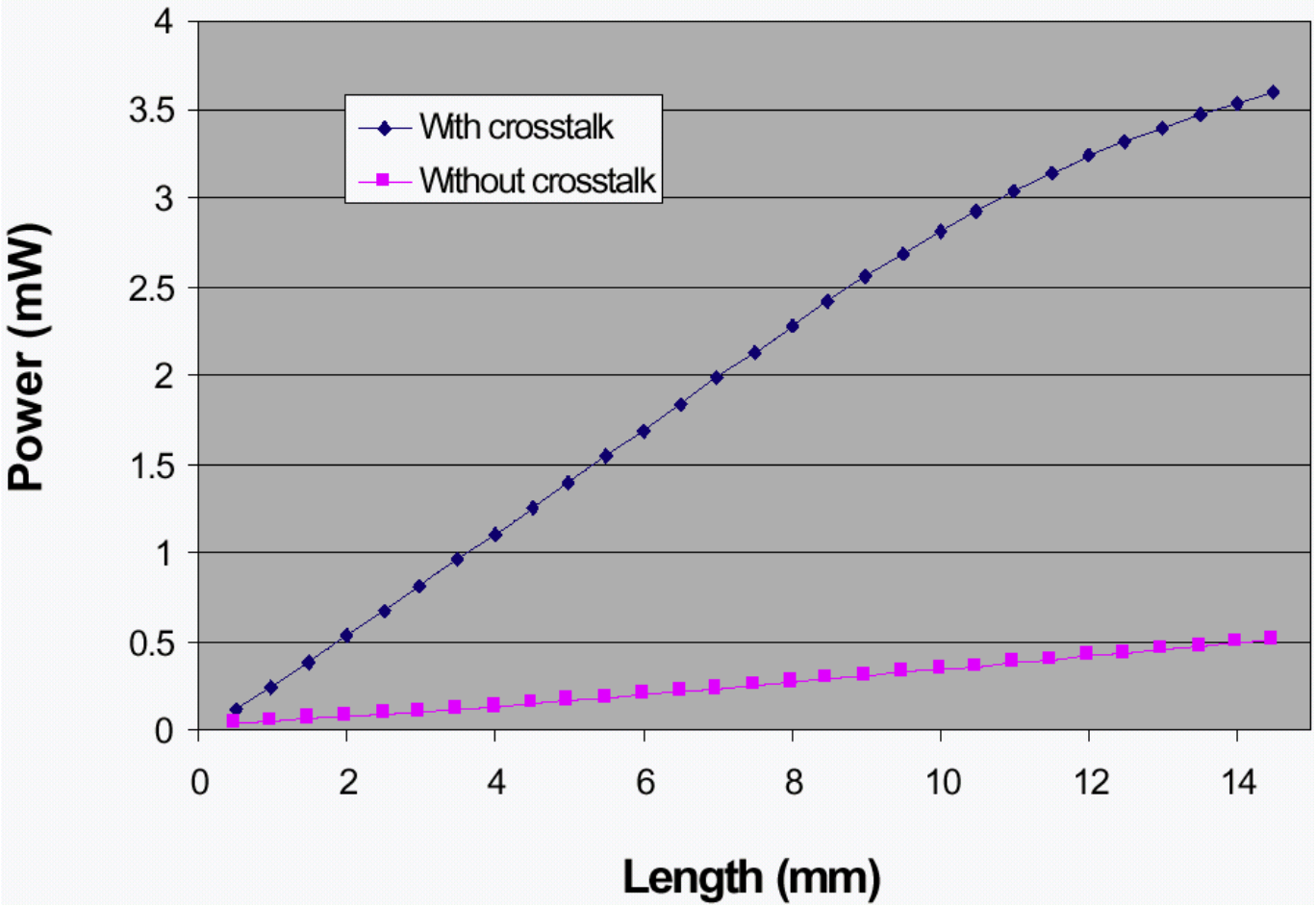
Source: Avant!



Source: Synopsys

Interconnect Power Consumption in DSM

- DSM effects in energy dissipation:
  cross-coupling capacitances



Source: Y. Zorian, S. Dey, and M. Rodgers, "*Test of Future System-on-Chips*," Proceeding of the 2000 International Conference on Computer-Aided Design, 392-398

Signal Integrity and Timing Closure

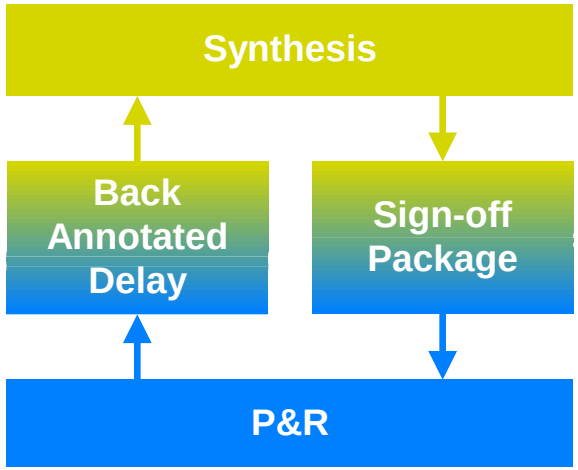
- Root causes of both Signal Integrity and Timing Closure
  - Inadequate interconnect modeling techniques
  - No effective design methodology
- Synthesis timing does not correlate with physical timing
  - Factors
    - Coupling capacitance increases
    - Interconnect resistance increases
    - Device noise margins decrease
    - Higher frequencies result in on-chip inductive effects
  - Problems
    - Signal electromigration
    - Antenna effects
    - Crosstalk delay
    - Crosstalk noise

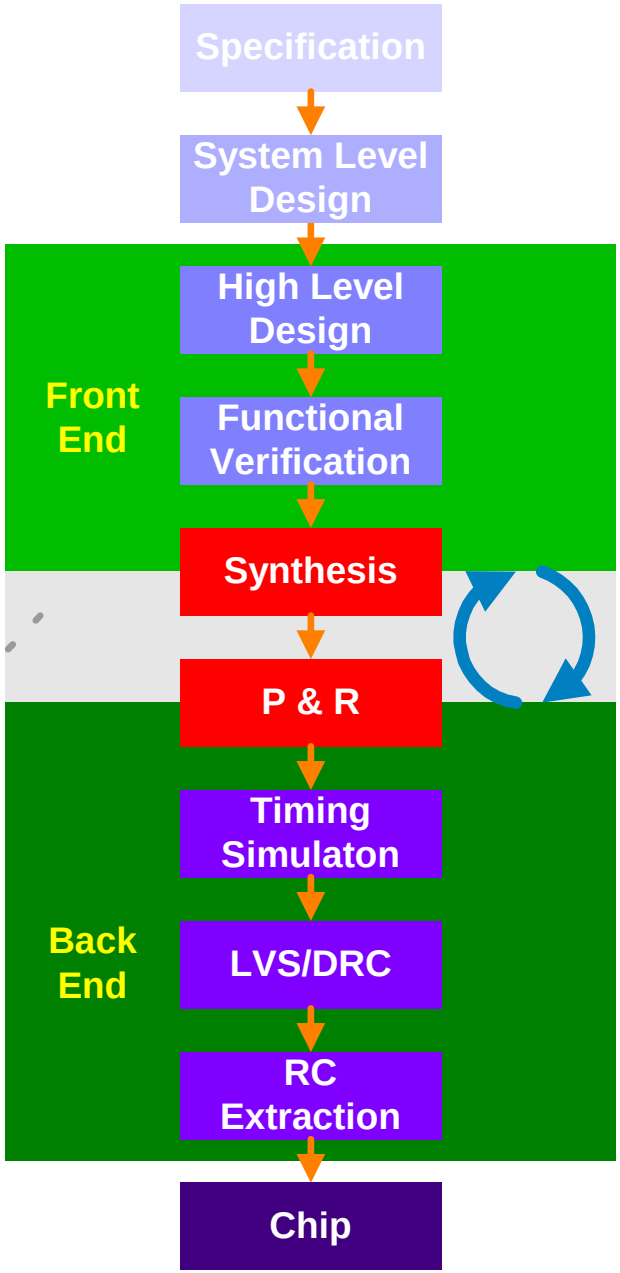
Example - Crosstalk Delay

- Net-to-net coupling capacitance dominates as a percentage of total capacitance in VDSM.
- The coupling capacitance can be multiplied by the *Miller Effect*
  - Wire capacitance can be off by 2X if the adjacent wires are switching in the opposite direction.
  - The coupling capacitance can be much less than expected if the wires are switching in the same direction
- Both have to be considered during timing analysis to fully account for setup and hold constraints.

New Physical Design Flow Needed

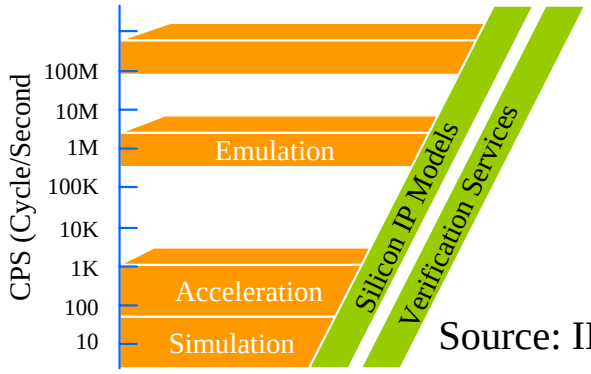
- Bring physical information into the logical design
- Overview of solutions
  - Single pass methodology
  - Synthesis-driven layout
  - Layout-driven synthesis
  - All-Integrated (optimization, analysis and layout) layout





HW/SW Cosimulation Through Emulation

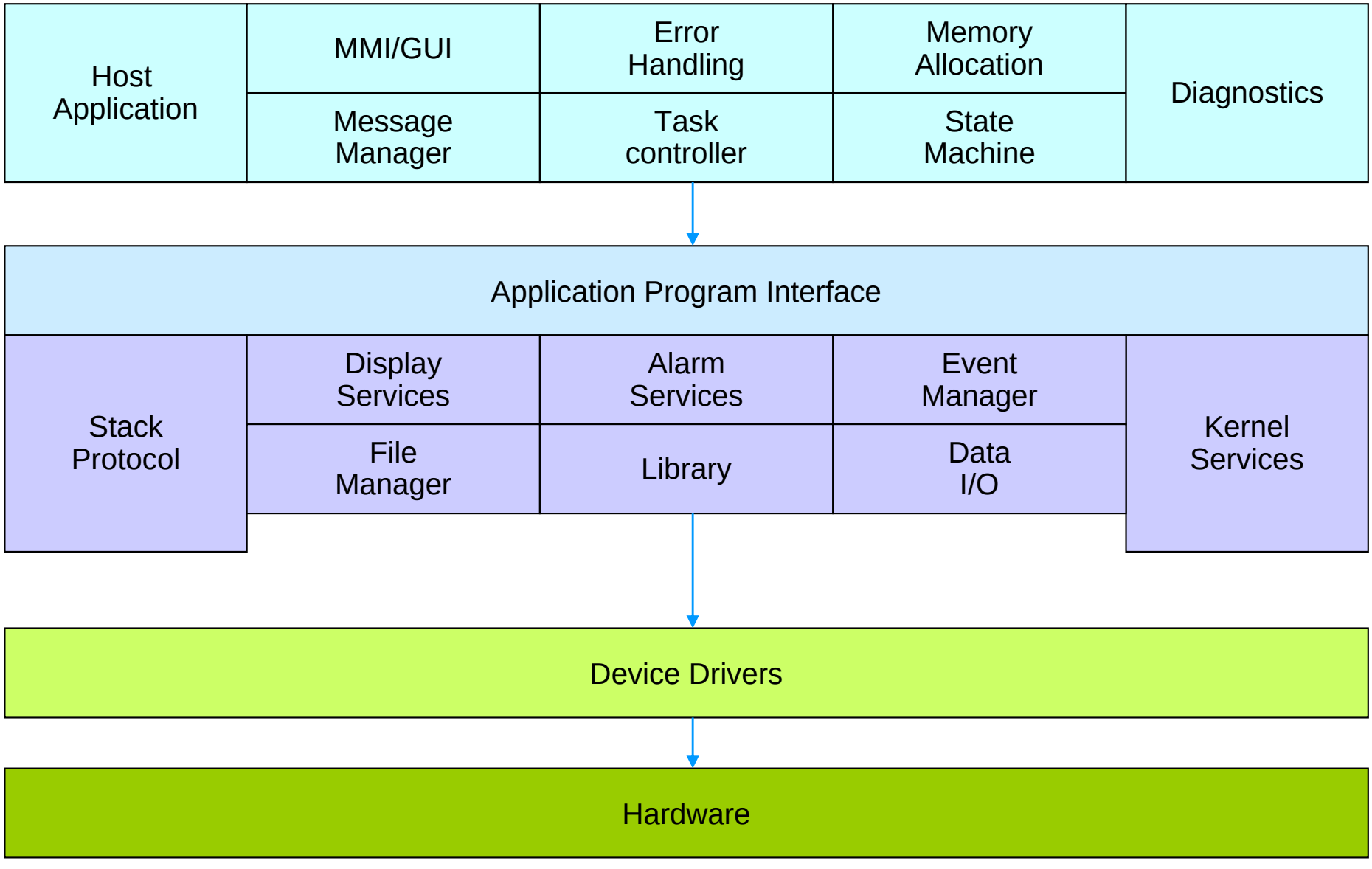
- Emulation in “virtual silicon”
  - Complete functional simulation of the chip at close to real time
  - Run real software



Source: IKOS Systems Inc

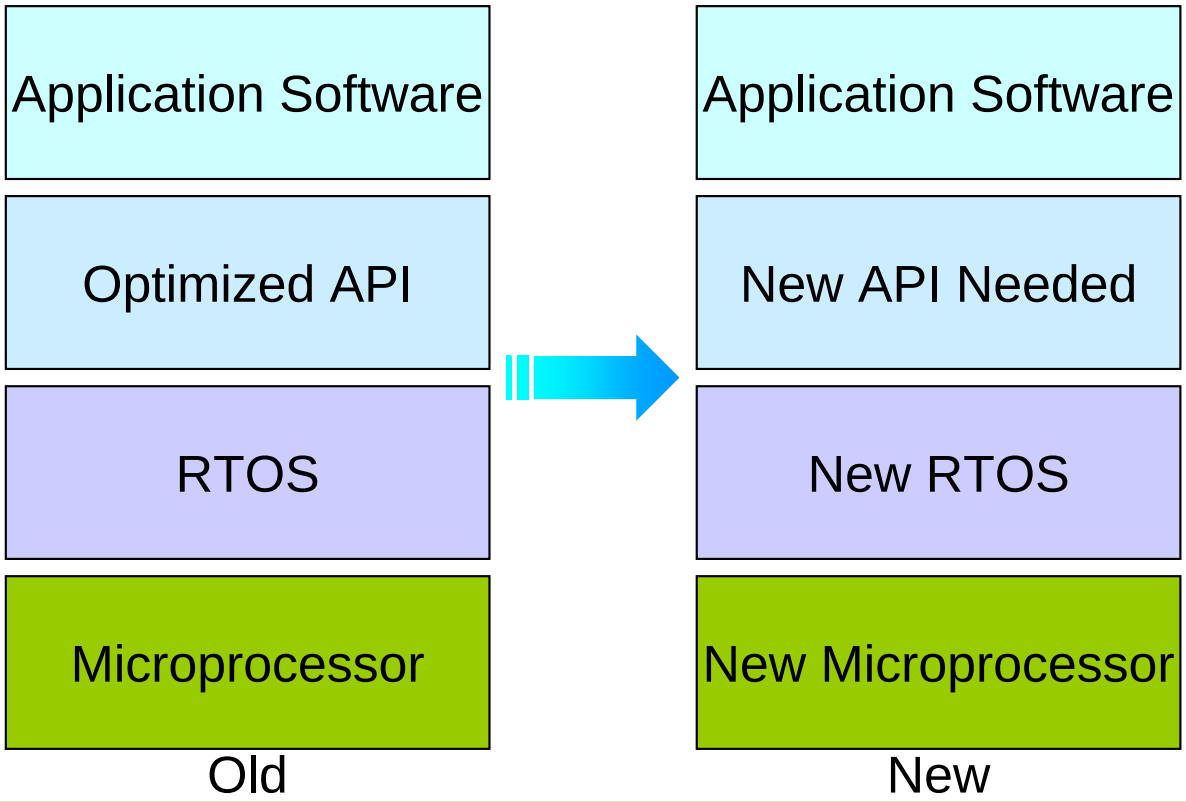
- Tools to enable simulation between EDAs and emulators
  - Cycle-based simulators
  - Full-timing simulators
  - Instruction set simulators
  - E.g. Quickturn Q/Bridge
- Expensive, long learning curve and set-up time

Embedded Software Architecture for SoC Design



Software Development

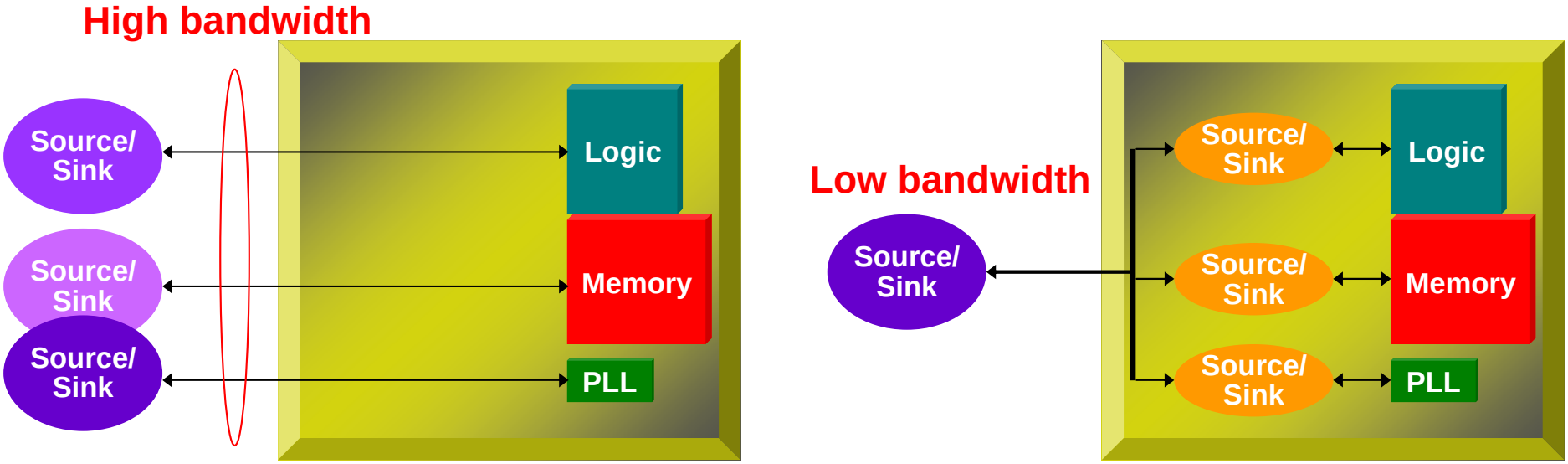
- Porting software to a new processor and RTOS
  - Using a common RTOS abstraction layer
- The evolution of embedded system in the future
  - An standard RTOS



Software Performance Estimation

- Have to take the following into account
  - Instruction set
  - Bus loading
  - Memory fetching
  - Register allocation
- Example: Cadence VCC technology
  - CPU characterized as Virtual Processor Model
  - Using a Virtual Machine Instruction Set
  - SW estimation using annotation into C-Code
  - Good for early system scheduling, processor load estimation
    - Two orders of magnitude faster than ISS
    - Greater than 80% accuracy

Tester Partitioning



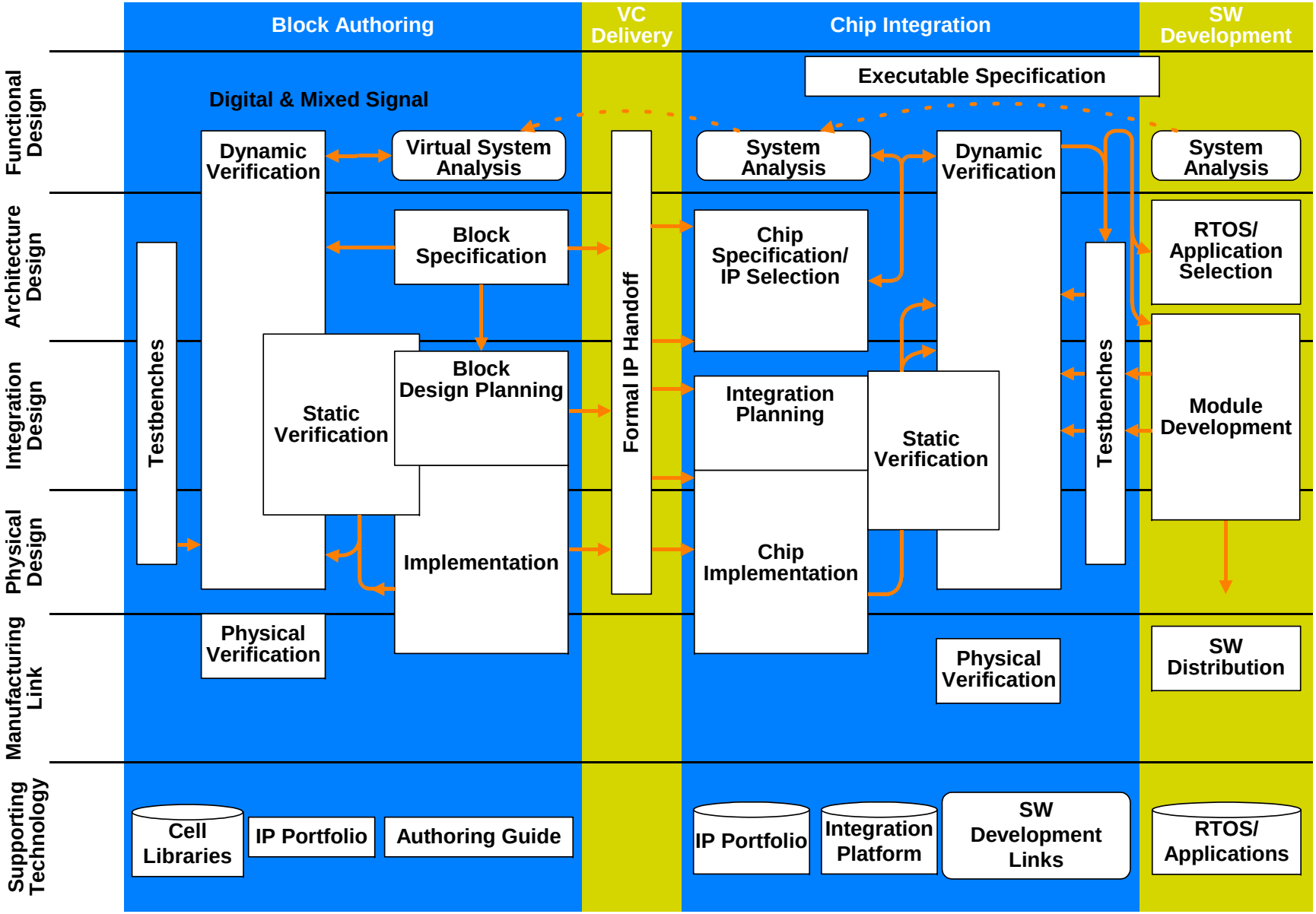
External Tester Embedded Tester External Tester Embedded Tester

Source: Y. Zorian, S. Dey, and M. Rodgers, "*Test of Future System-on-Chips*," Proceeding of the 2000 International Conference on Computer-Aided Design, 392-398

Self-Testing of Embedded Processor Cores

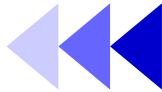
- Logic BIST
  - Based on the application of pseudo random test patterns generated by on-chip test pattern generators like LFSRs.
  - Cannot always achieve very high fault coverage for processor cores.
- Instruction-based self-test techniques
  - Rely on generating and applying random instruction sequences to processor cores.
  - The approach determines the structural test needs of processor components
  - Advantage: programmability and flexibility

Platform-based Design



Source: "Surviving the SOC Revolution - A Guide to Platform-Based Design" by Henry Chang et al, Kluwer Academic Publishers, 1999

Design Entry



Gate level

Truth table

FSM

Waveform

1K~10K

Manage Size and Run-Time

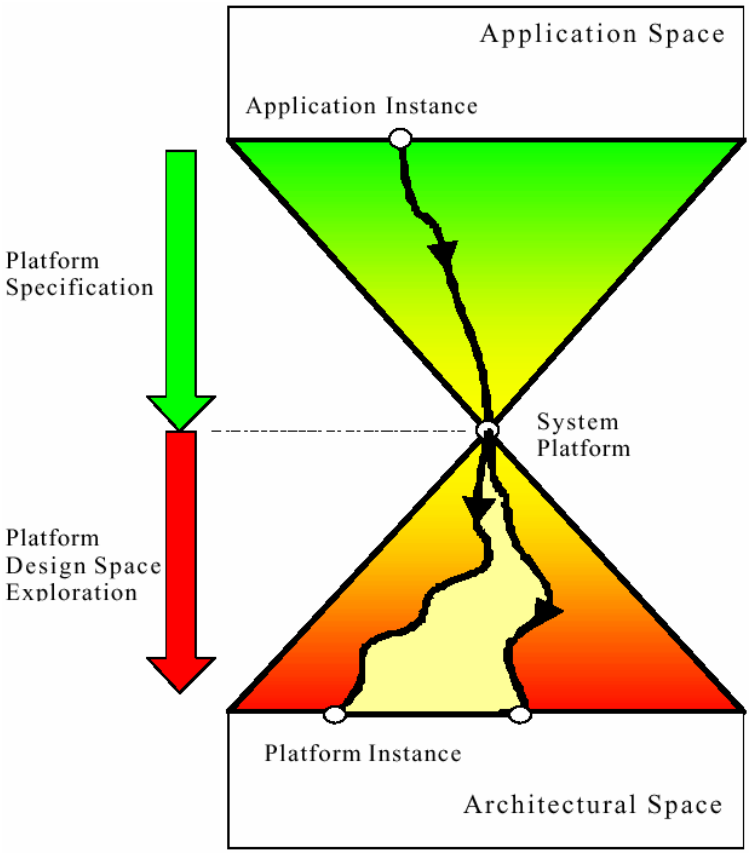
RTL level 10K~100k

System level modeling 100K~100M

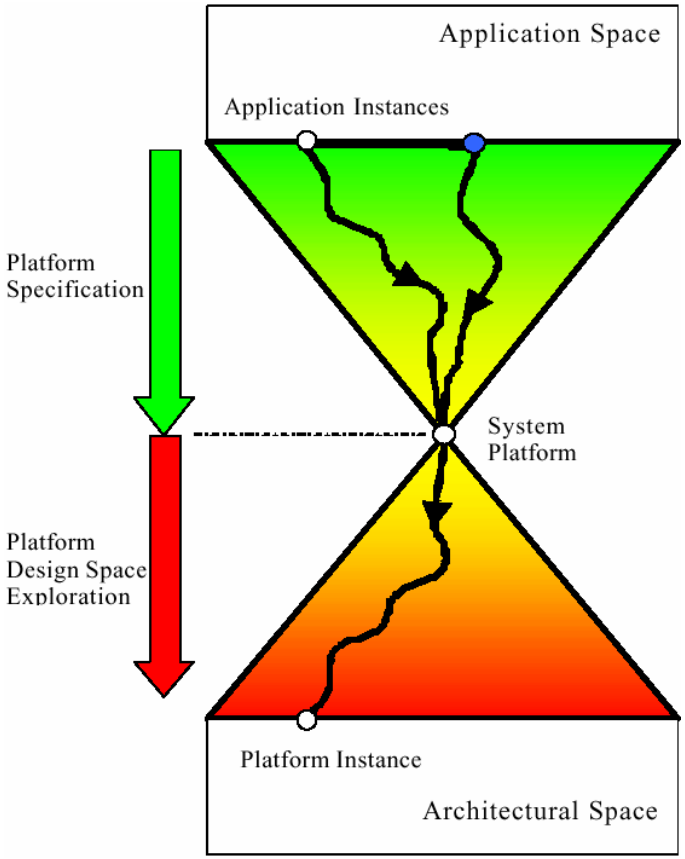
Hardware Platform-Based Design

It is a “meet-in-the-middle” approach.

System Integrator Perspective



Platform Provider Perspective



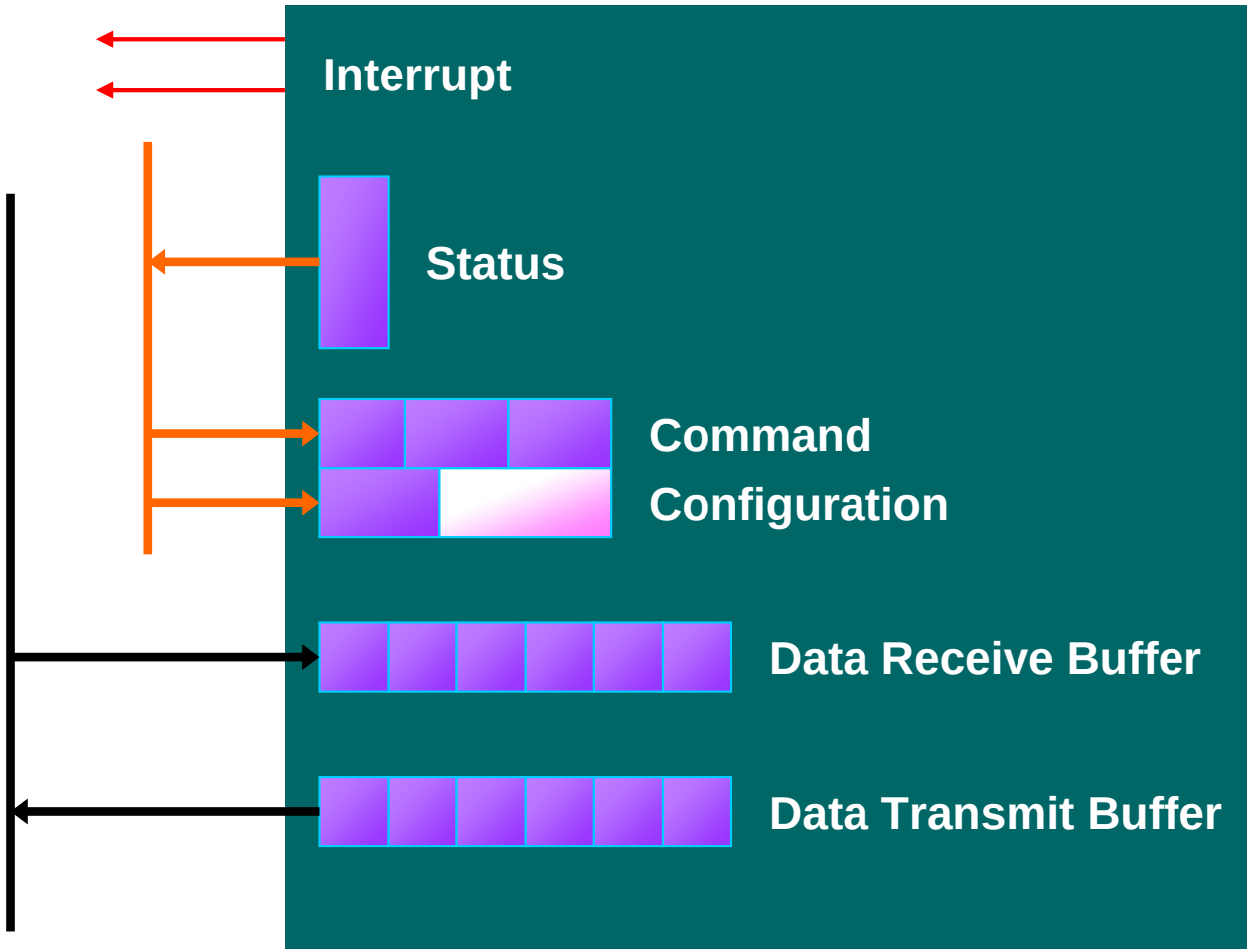
System-Level Design

- Goal
  - To define the platform that satisfies the system functions with performance/cost tradeoff
- Platform design
  - Bus structure
  - IP and their function design
    - **Customized instructions**
    - **Parallelism**
    - **Command parameters**
    - **Configurable parameters**
    - **IP parameters**
  - Control scheme
  - Data communication (bandwidth)

Control Scheme Model

Interrupts

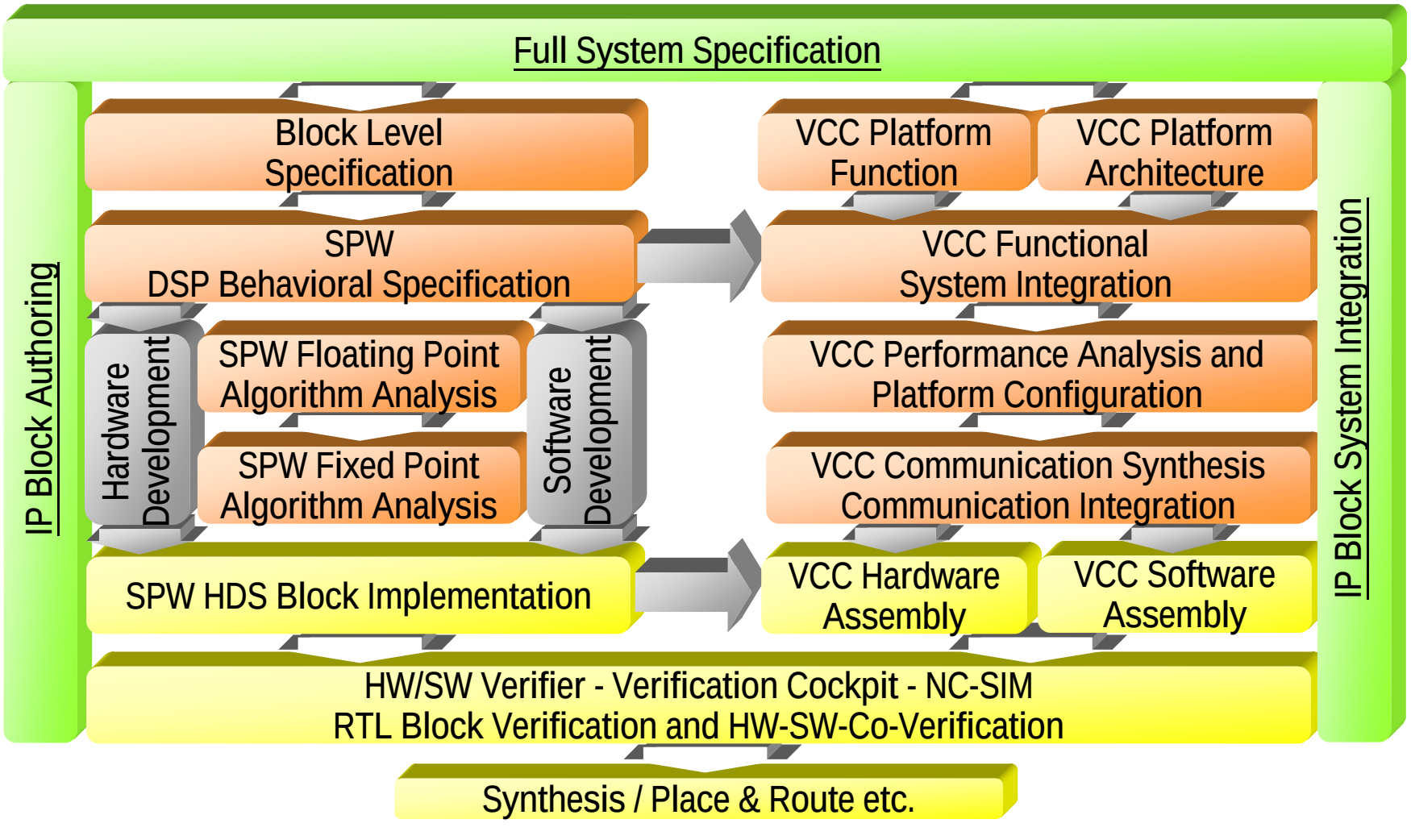
Status Polling (timer)



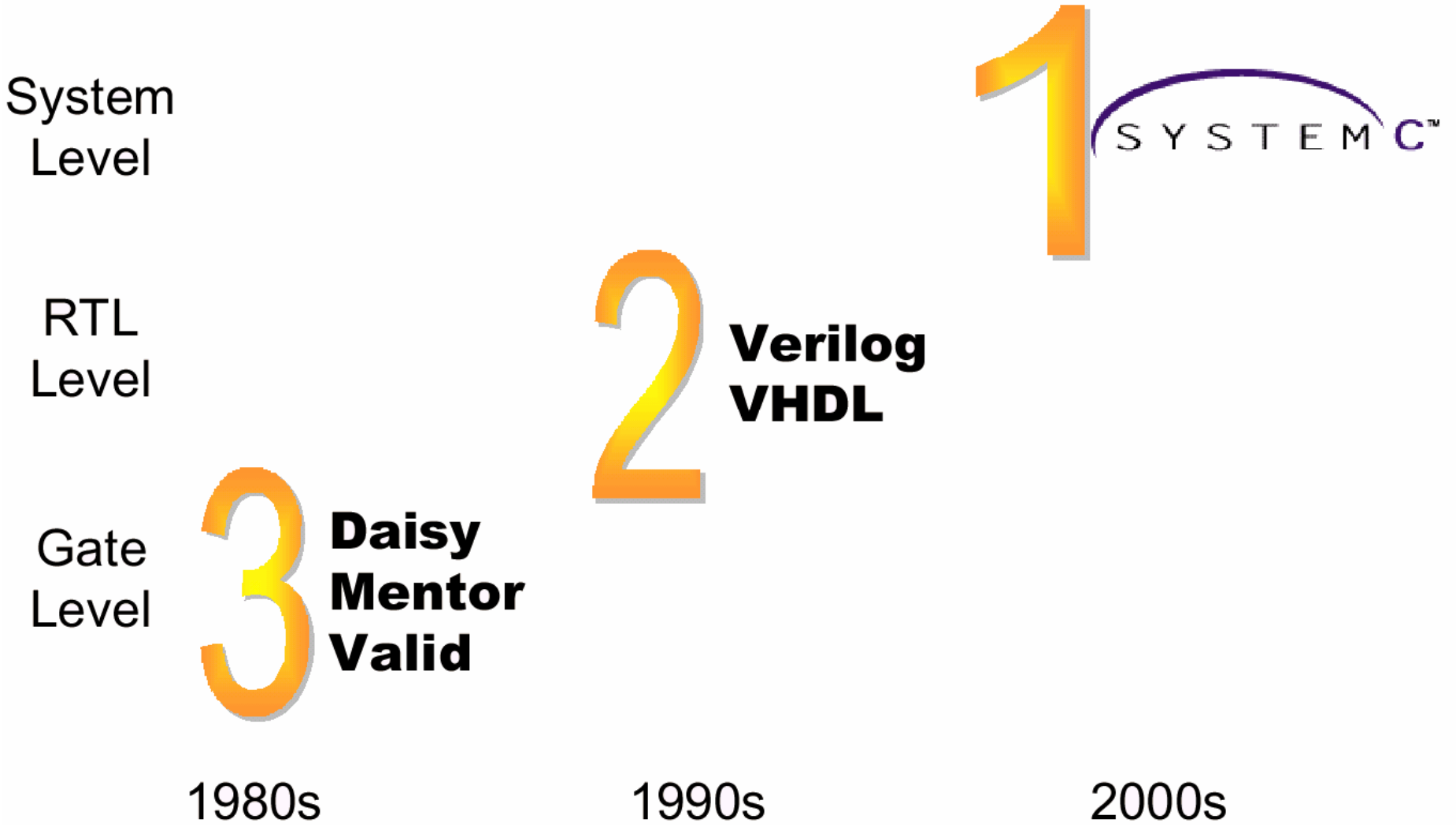
Some Helps in System-Level Design

- Cadence VCC (Virtual Component Codesign, from Cadence Berkely Labs)
  - Performance simulation
  - Communication refinement technology
- Vast Systems Technology
  - VPM (Virtual Processor Model)
  - HW/SW codesign
- CoWare N2C (Napkin-to-chip)
  - Interface synthesis
- SystemC

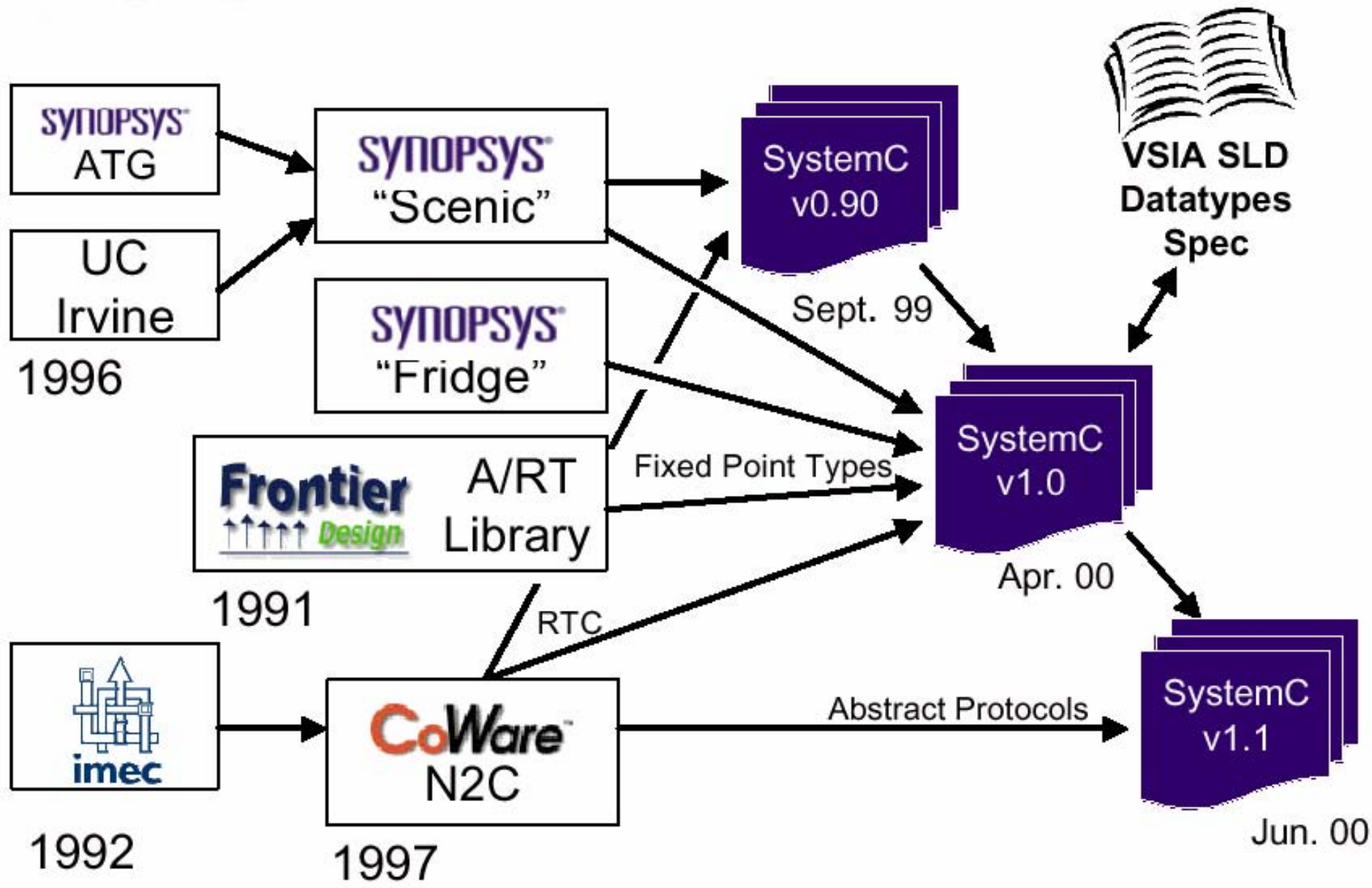
Cadence's VCC



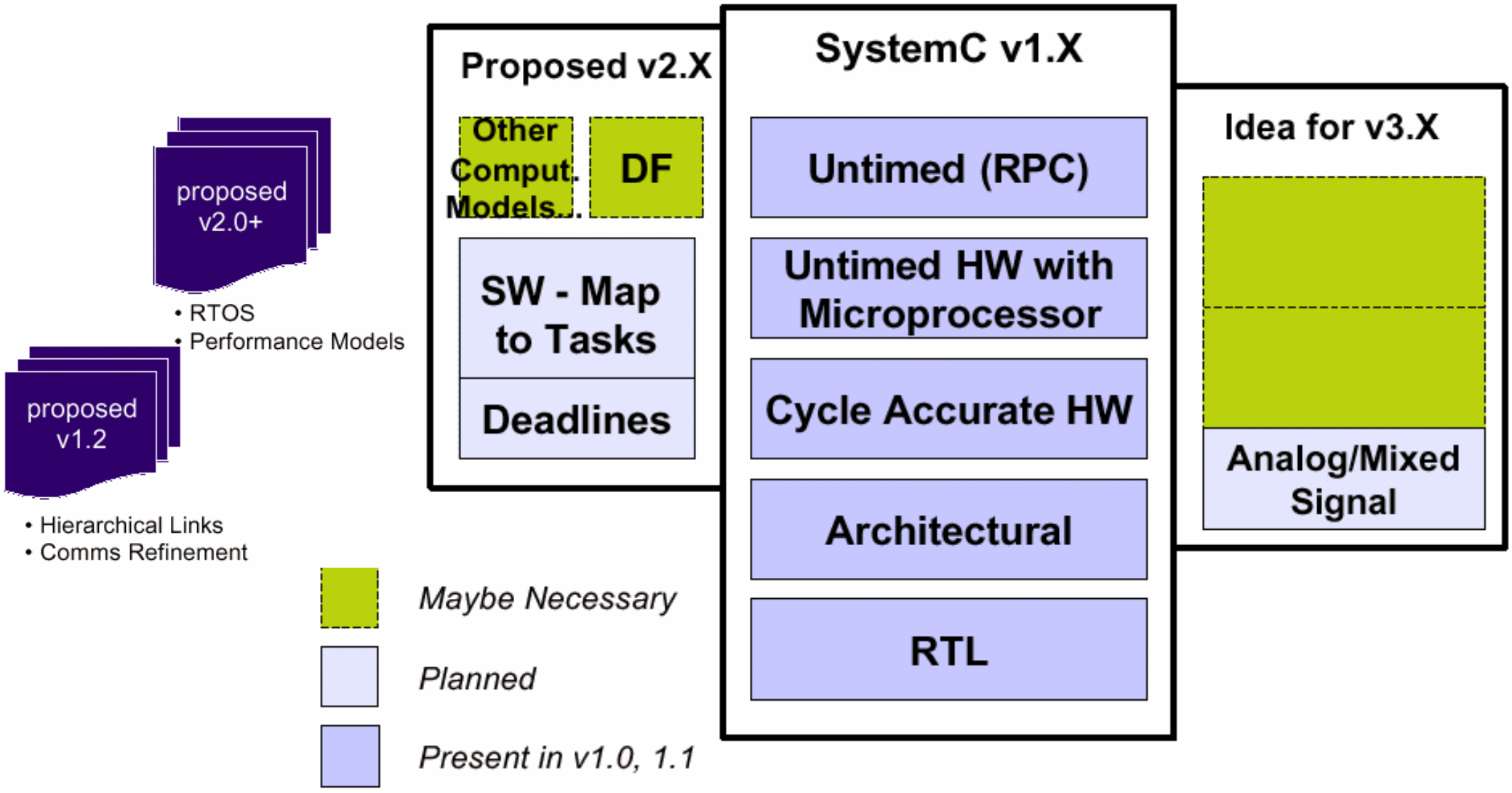
An Opportunity To Do It Right !



SystemC Heritage



SystemC Roadmap



The Intent of Different Level of Model

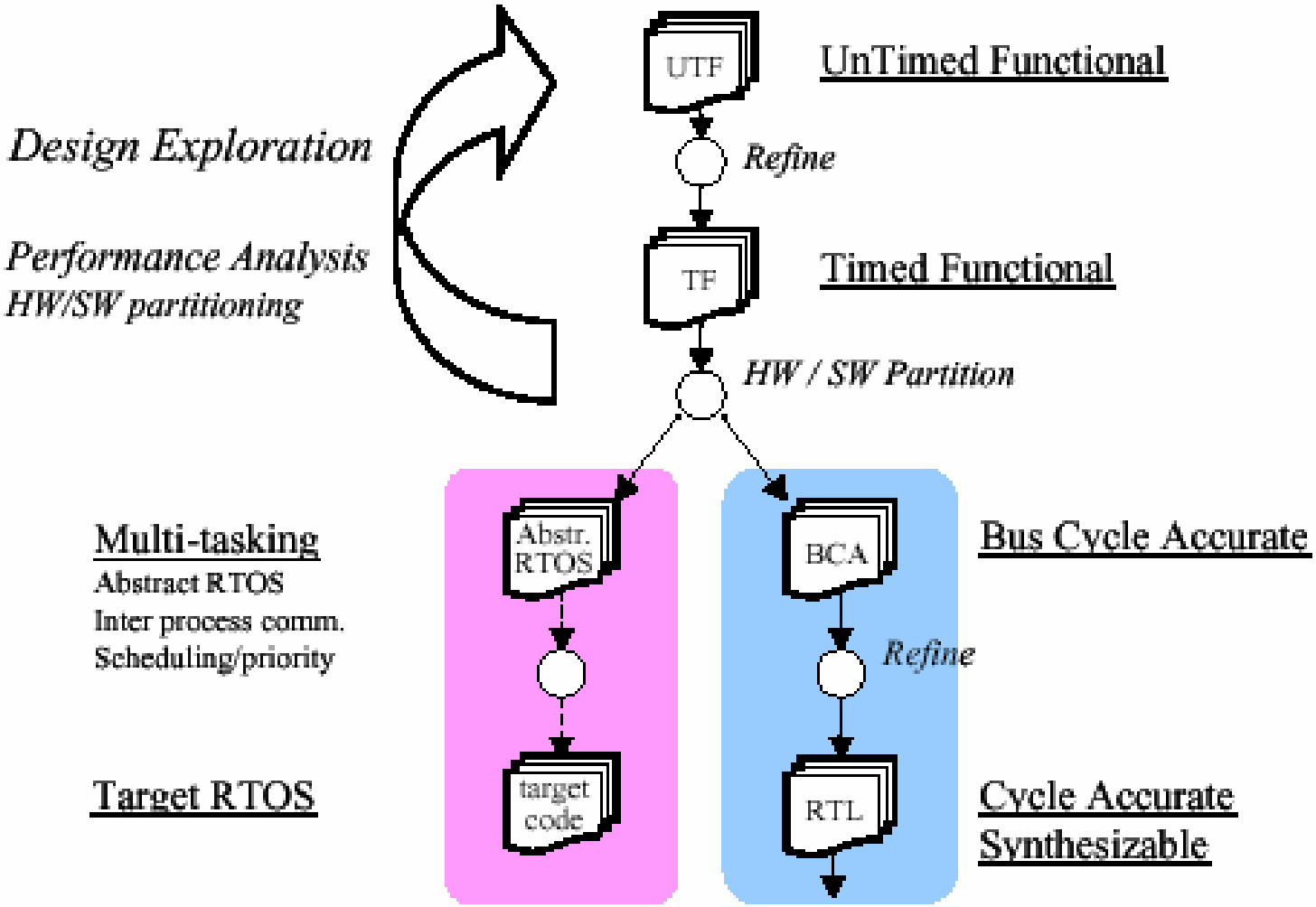
- Design exploration at higher level
  - Import of top-level constraint and block architecture
  - Hierarchical, complete system refinement
  - Less time for validating system requirement
  - More design space of algorithm and system architecture
- Simple and efficient verification and simulation
  - Functional verification
  - Timing simulation/verification
  - Separate internal and external (interface) verification
  - Analysis: power and timing
- Verification support

SystemC

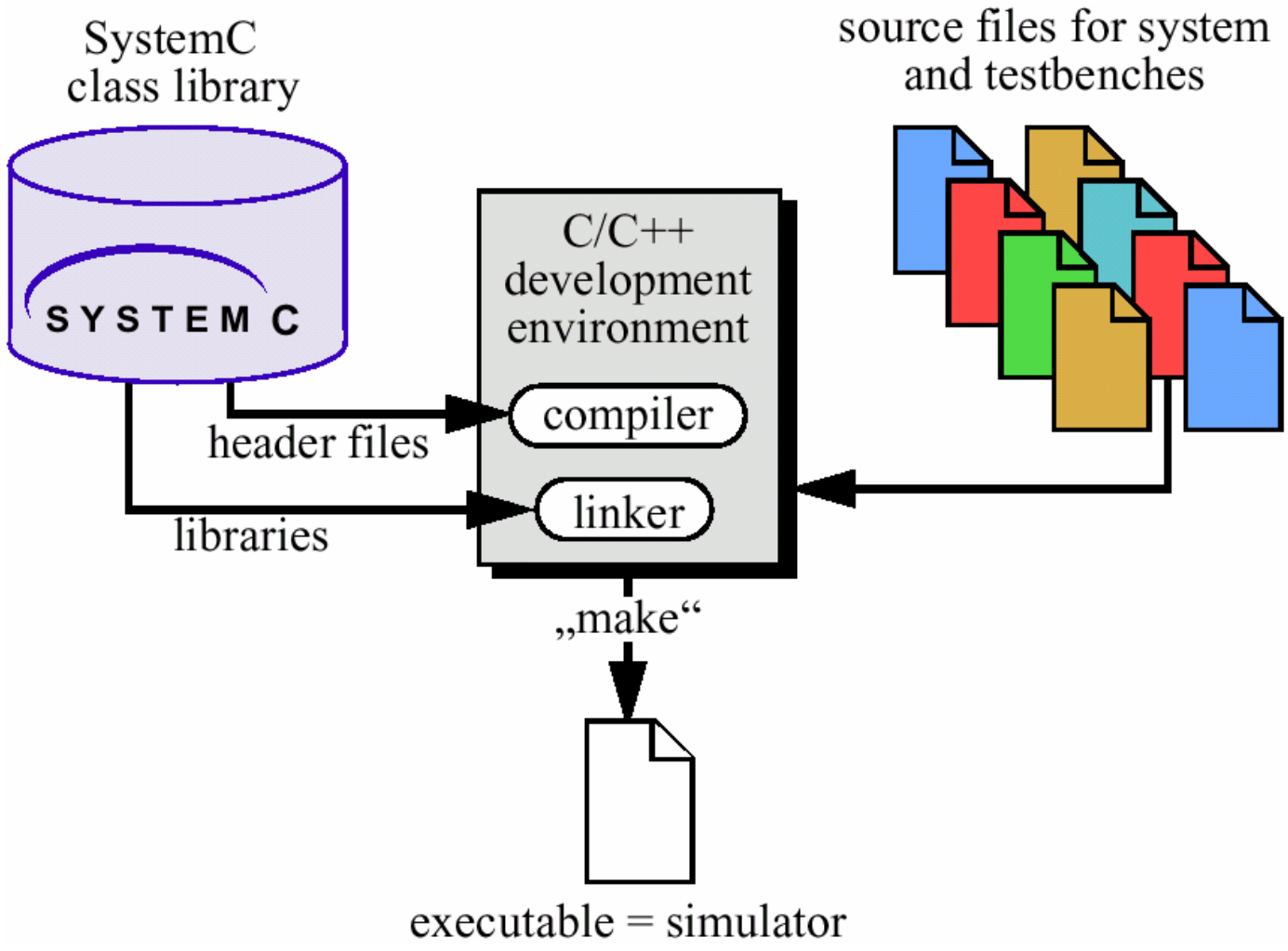
- SystemC is a modeling platform
  - C++ extensions to add hardware modeling constructs
  - a set C++ class library
  - simulation kernel
  - supports different levels of abstraction

Good Candidate for Task Level Mapping

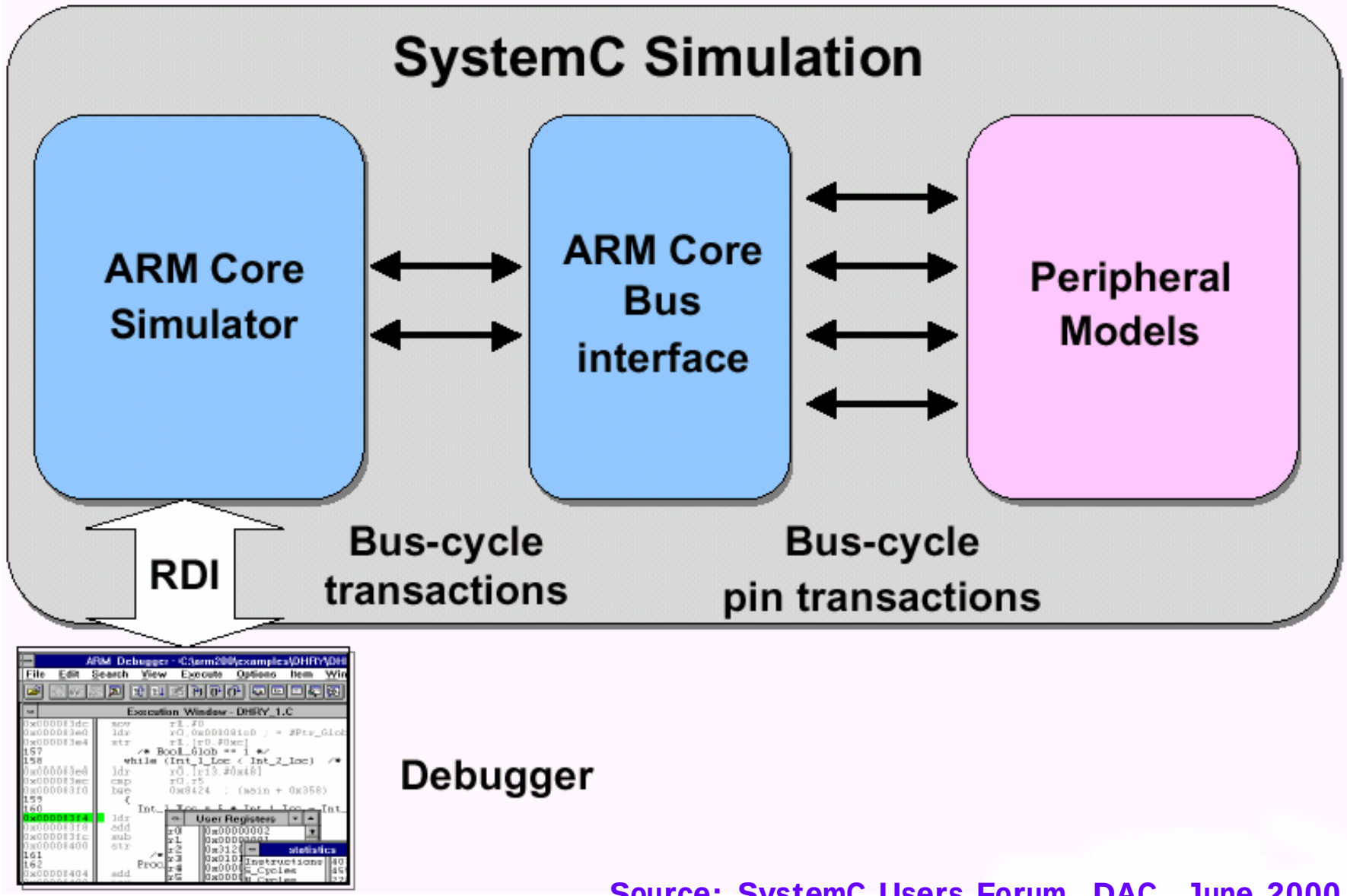
Level of abstraction in SystemC



SystemC Design Flow



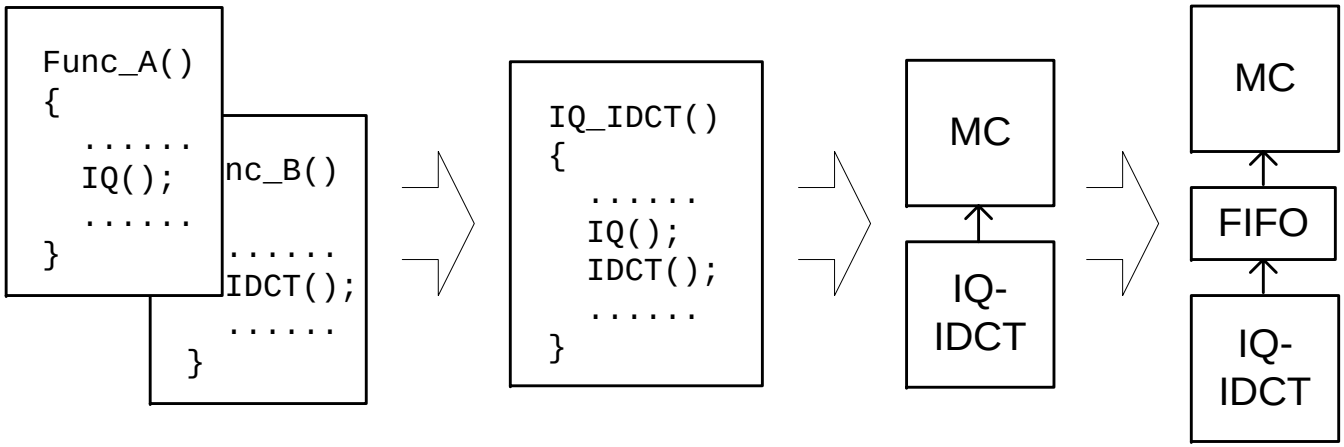
Example



Source: SystemC Users Forum, DAC, June 2000

Implementing Virtual Prototypes

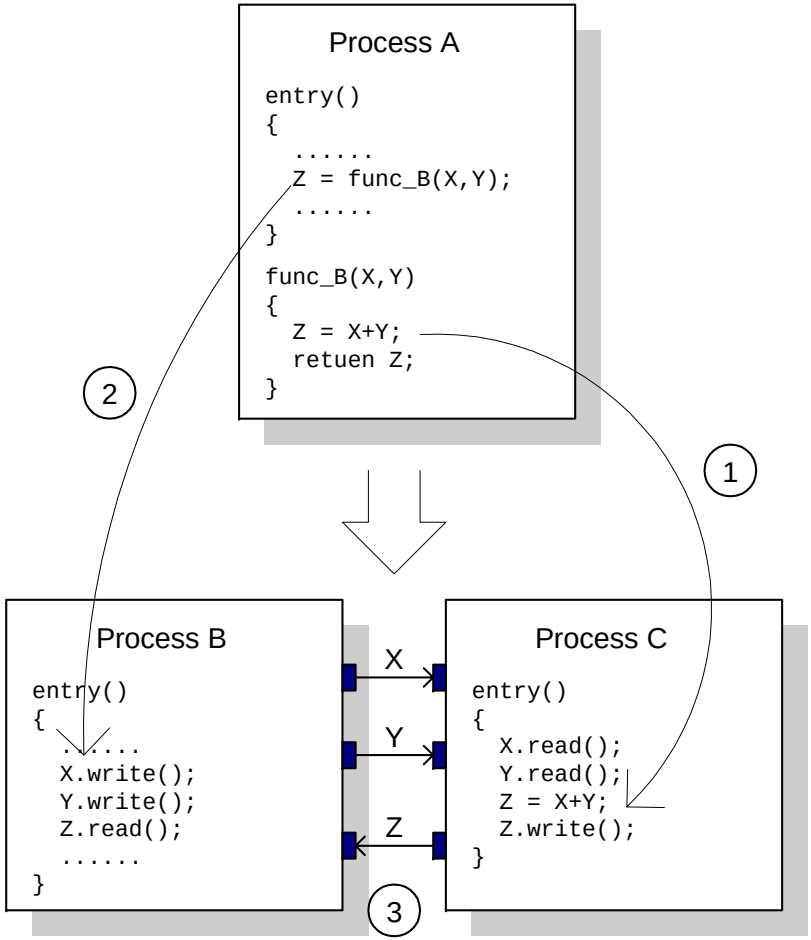
- Functionality partition
- Module specification
- Communication refinement



Functionality Partition

- Separating communication and computation
- Using hierarchy to group related functionality
- Choosing the granularity of the basic parts

Module Specification (1/2)



1. Pull out functionality into new created process
2. Replace function call with inter-process communcation.
3. Instantiate new process and define channels to connect them.

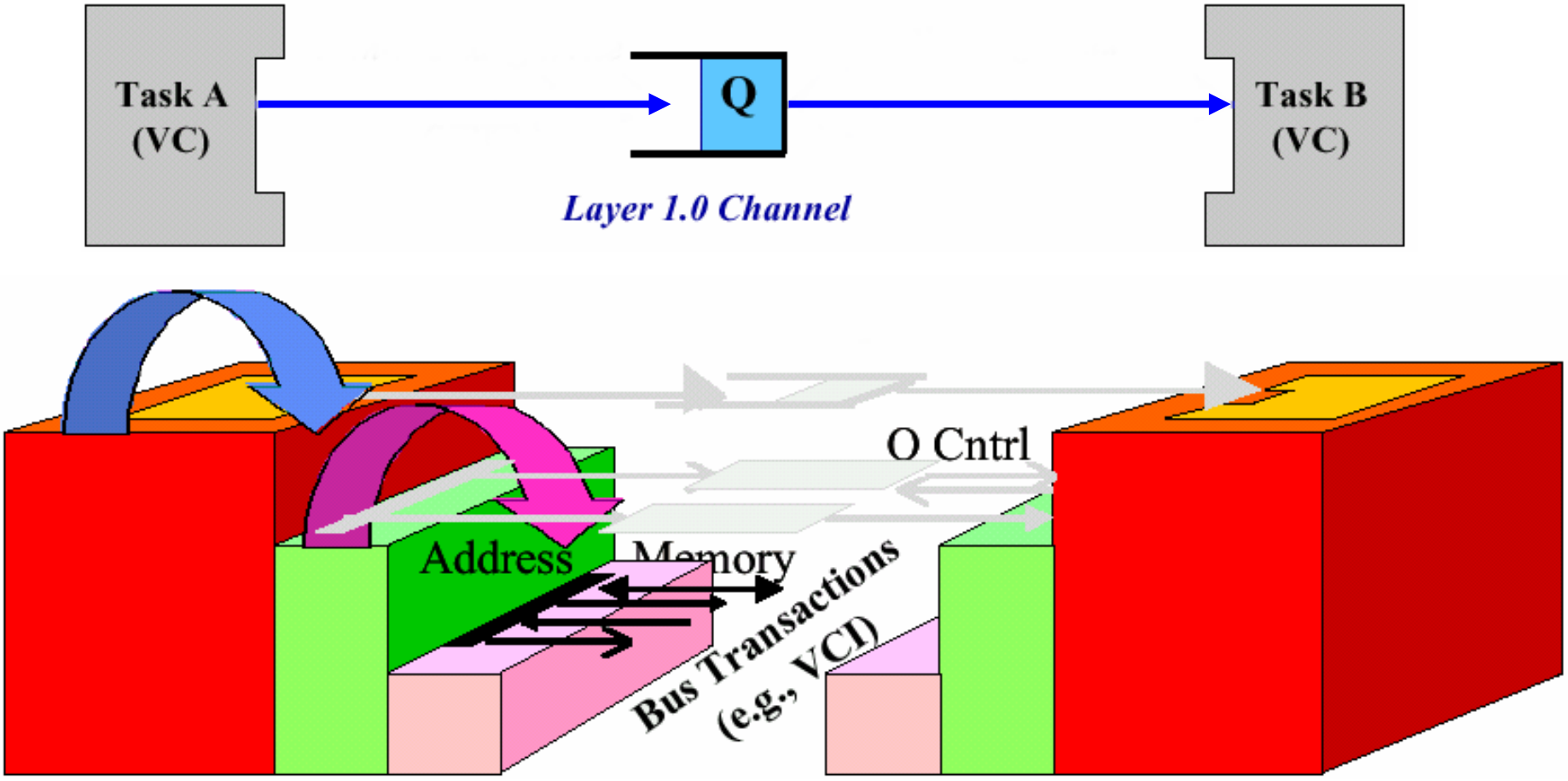
Module Specification (2/2)

- Abstraction Levels
  - Untimed Functional Level
    - Processes execute in zero time but in order
  - Timed Functional Level
  - Bus-Cycle Accurate Level
    - Transaction on bus are modeled cycle accurate
- Cycle Accurate Level
  - Behavior is clock cycle accurate

Communication Refinement

Key

Guarantee consistency of communication during refinement



Software Performance Estimation

- Have to take the following into account
  - Instruction set
  - Bus loading
  - Memory fetching
  - Register allocation
- Example: Cadence VCC technology
  - CPU characterized as Virtual Processor Model
  - Using a Virtual Machine Instruction Set
  - SW estimation using annotation into C-Code
  - Good for early system scheduling, processor load estimation
    - Two orders of magnitude faster than ISS
    - Greater than 80% accuracy

Discussion: Commonality and Differentia

- Differentiae
  - Processor core (e.g., customized inst. set)
  - IP parameterized
  - IP add/move
- Design methodology of platform
  - System-level
  - Platform-level design methodology
    - Design flow
    - Models
    - Tools (EDA venders, 3rd party or home-made)

Summary

- Platform-based design
  - From board design to SoC design
  - From executable spec., i.e., C/C++, to SystemC
- Modeling
  - Performance evaluation
  - Task mapping
  - Communication refinement