OSCI Update

Guido Arnout
OSCI Chief Strategy Officer
CoWare Chairman & Founder

Chief Strategy Officer charter

- Ensure that OSCI strategy is created, coordinated, communicated & executed
- Identify OSCI technical activities & provide direction + strategic guidance to the working groups
- Keep technical & business objectives in-sync & appropriately prioritized

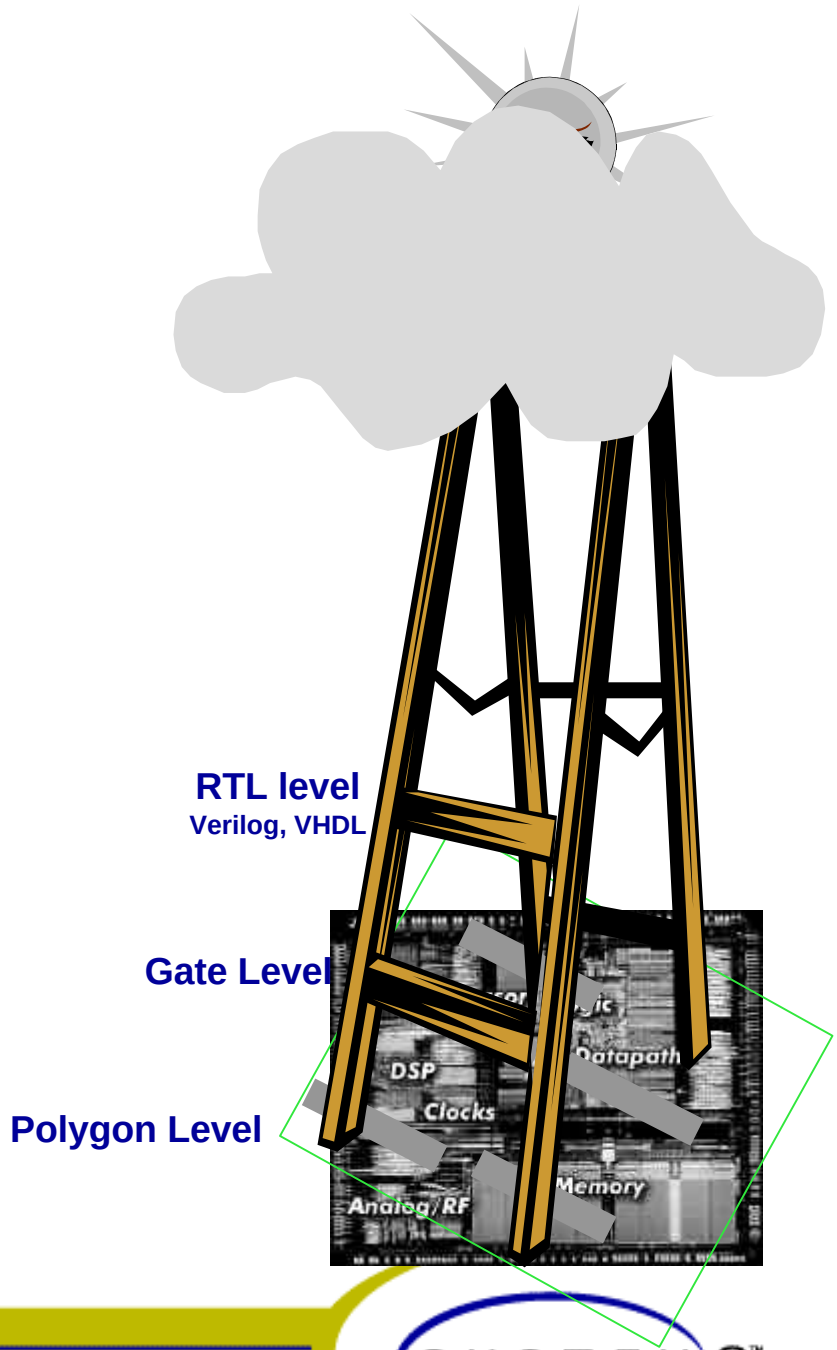
“Concept to RTL”

flowing into

“RTL to GDSII”

mid 70’s to mid 90’s

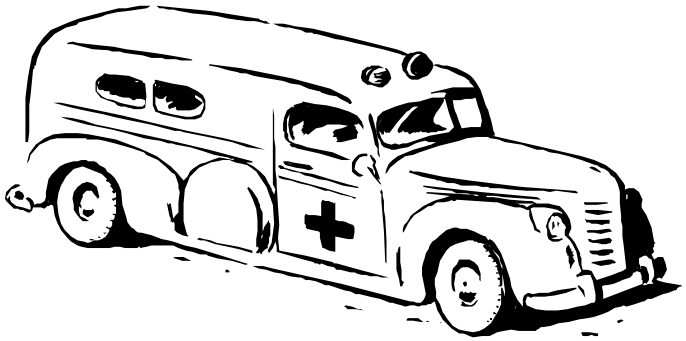
- Design abstraction moves up
- New levels of IP emerge
- EDA tools move up…
- … to enable path to silicon

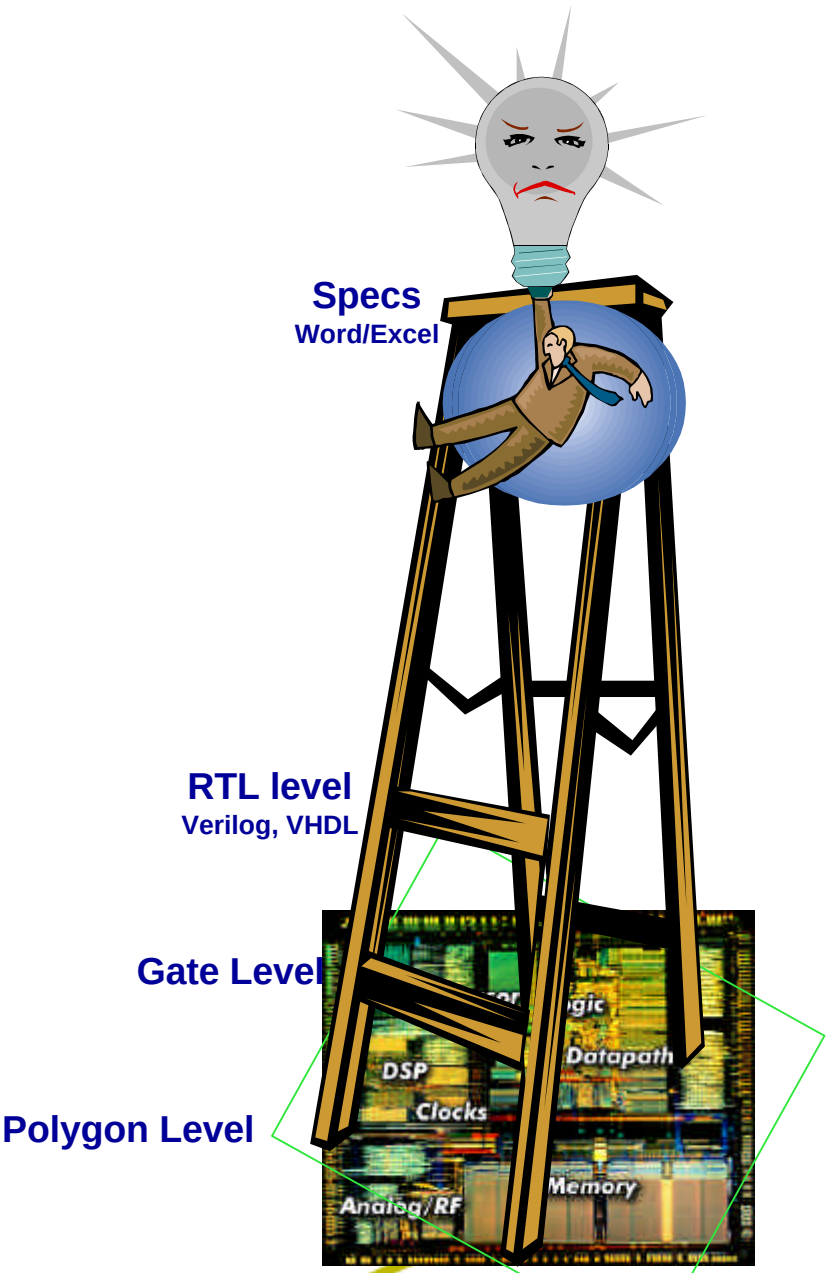


SYSTEMC™

mid 90's

- **EDA confronted with:**
  - **System = software running on silicon!**
  - **Gap between system design & RTL**

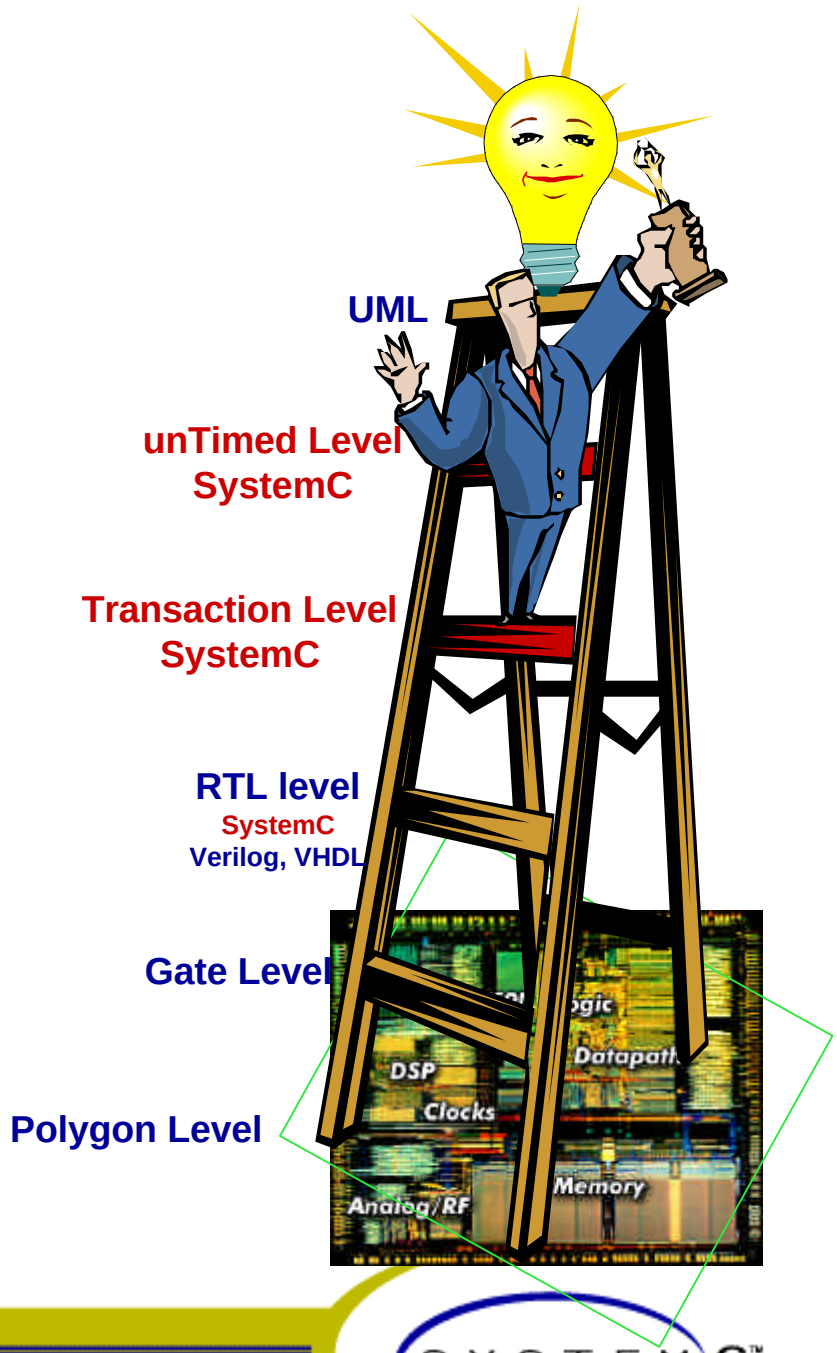




SYSTEMC™

2002 & beyond

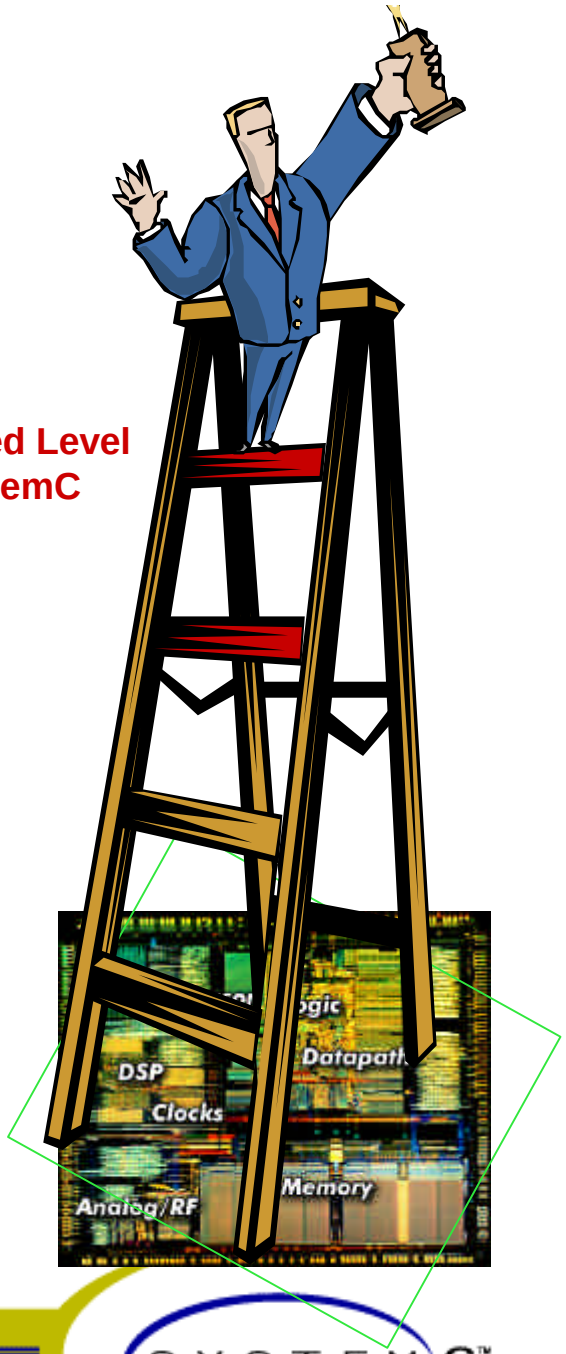
- EDA moves up, Again!
- SystemC bridges abstract modeling & RTL
- IP moves up, Again!
- EDA tools ensures path to silicon, Again!



SYSTEMC™

Value of untimed SystemC

- Design & debug system functionality
  - Prior to committing to an architecture
  - Prior to deciding what is SW or HW
- Fast enough to develop application software
- Ensures that the right system is built



SYSTEMC™

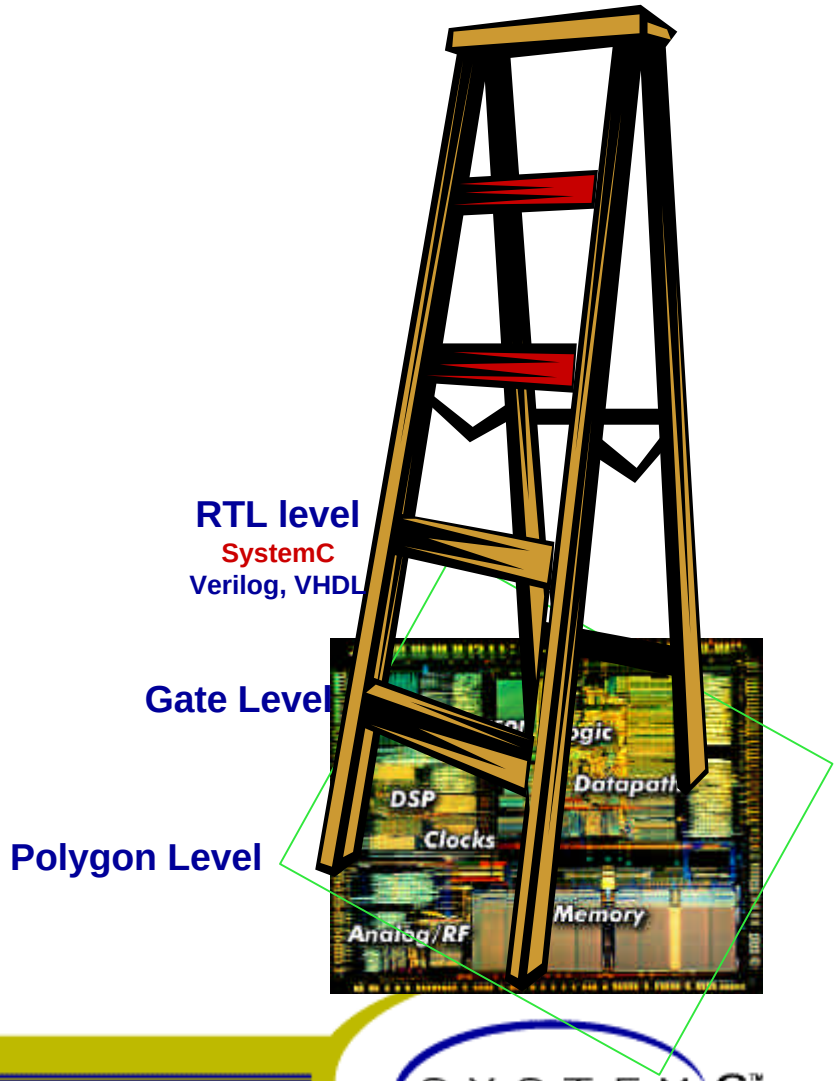
Value of transaction level SystemC

- Functional + transaction cycle accurate
- Design & explore system architectures
  - Prior to implementing “intuition” in RTL
- Fast & accurate enough to develop “hardware dependent software”
- Enables true IP re-use at the system level
- Ensures that the right system is built right

SYSTEMC™

Value of RTL level SystemC

- The familiar pin cycle accurate RTL level
- Remains valuable for detailed “component” implementation
- Not the focus of SystemC



SYSTEMC™

Significance of TLM

Design productivity = *f*(abstraction level)

- **Productivity per designer (fully documented & debugged)**
  - **30 polygons per day**
  - **30 transistors per day**
  - **30 gates per day**
  - **30 lines of RTL per day**

  while communication abstraction only moved …
  … from voltage/current to 0/1

- **What 30 “things” do we now want to do in 1 day?**

SYSTEMC™

The next 30 “things” per day!

- Put together a platform with 30 IP blocks
  - Processors, Buses, Peripherals, Memories, …
- Sprinkle some software on the processors
  - Algorithm, expected software behavior, …
- Build, run, analyze result today…
  … to make new architectural design decisions tomorrow
  - Go from single layer bus to multi layer, …
  - Add or replace processors, …
  - Change memory configurations, peripherals, …
- Until platform is “tuned” for expected SW

The goal?

- **Platform serves as a HW model for SW development**
  - **Only if 100x faster than the RTL that HW people offer now**

AND

- **Is starting point for HW implementation & verification**
  - **Only if it has 100% cycle accurate communication**
  - **Only if it has path to Block & Communication synthesis**

What does this require?

- IP at a higher level of behavior AND communication abstraction
  - Transactions are next higher level than 0 & 1; hence TLM
  - Simpler to understand and code
  - Faster to execute (100x faster than RTL)
  - Yet 100% cycle accurate to make architectural decisions
- IP that in interchangeable
  - Requires a standard TLM API that can exploit all bus functionality
- IP with a path to implementation
  - Needs a modeling style standard
  - Must enable communication synthesis
  - Must enable block synthesis (or replacement with proven RTL)

Significance of a TLM standard

- **Bridges the worlds of software and hardware**
  - **Bridge 1 = LANGUAGE**
    - **SystemC brings HW and SW in the same language base**
  - **Bridge 2 = SPEED**
    - **TLM communication abstraction 100x faster**

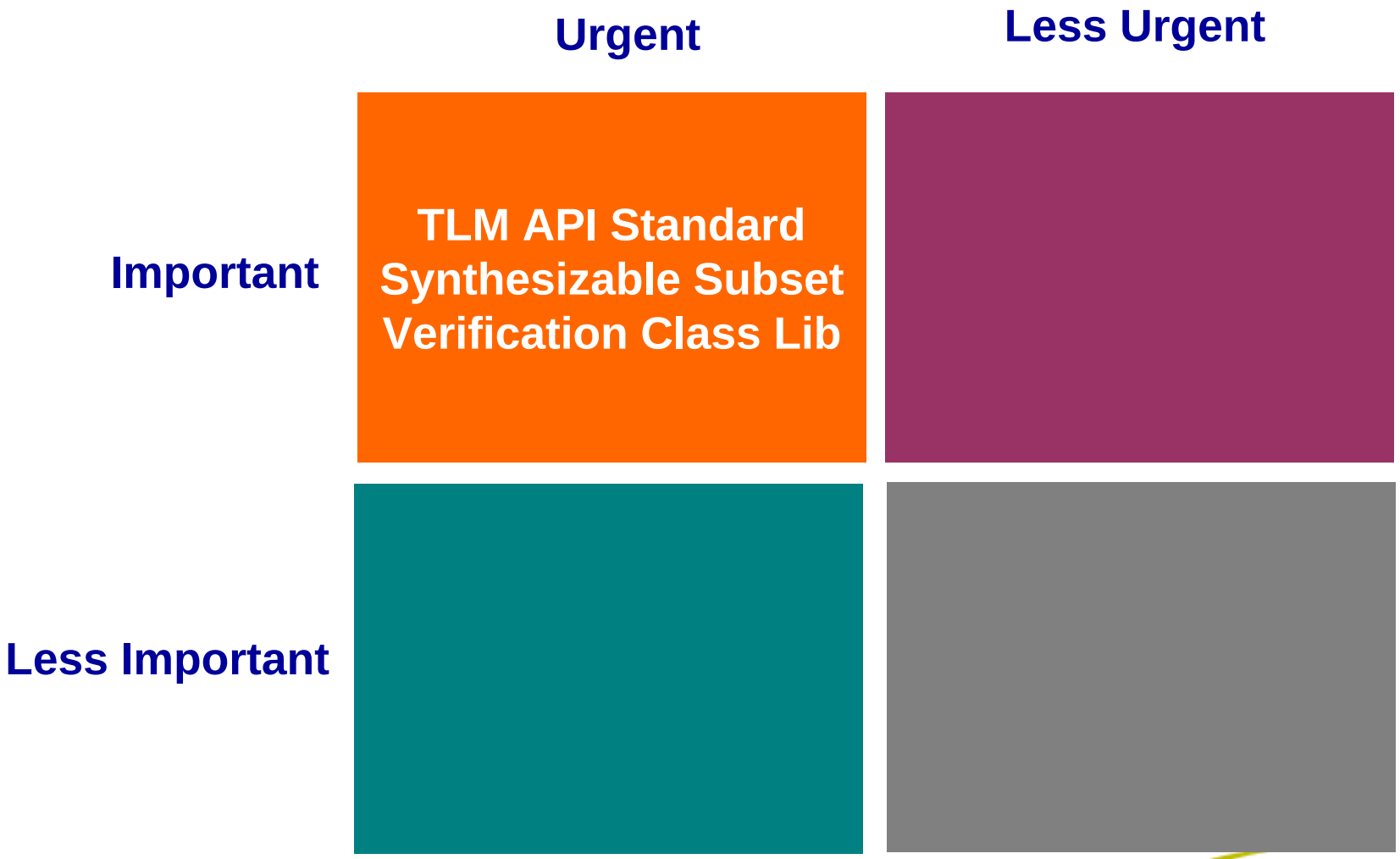
SW

HW

SYSTEMC™

Impact on OSCI direction

The “Stephen Covey” matrix



Urgent things we are doing

- Language Working Group
  - Funding LRM for SystemC 2.1 core language
  - Submit for IEEE standardization in 2003
  - Ensure consistency of “core” language
    - “Not powerful enough” is bad but “Too complex” to use is equally bad
- Verification Working Group
  - Building SystemC 2.1 compliant verification class library

Urgent things we are not doing yet

- **Standardize TLM API**
  - Different industry groups are defining first flavors
  - Without changing the core language
- **Coding style guide & subset for block synthesis**
  - Creation of Synthesis Working Group approved

3 take-aways

- The value of SystemC is in higher abstractions
- The transaction level bridges SW and HW
- SystemC core language ready for IEEE
  - Working Groups evolving compliant libraries