"Use and re-use considerations when creating SystemC TLM models"

An Approach presented by:
Sameer Arora
Project Manager
HCL Technologies Ltd.
Noida
(sameera@hcltech.com)

HCL

Purpose

This presentation talks about "Use" and "Re-use" considerations when creating SystemC TLM Models.

- What is abstraction, Why abstract design models.
- Various abstraction levels, and their use cases.
- Re-use from one abstraction level to another, and from one model to another.

Focus is on the most typical models – SOC peripherals – that connect to the host bus, have a SW programmable register set, and one or more I/O interfaces.

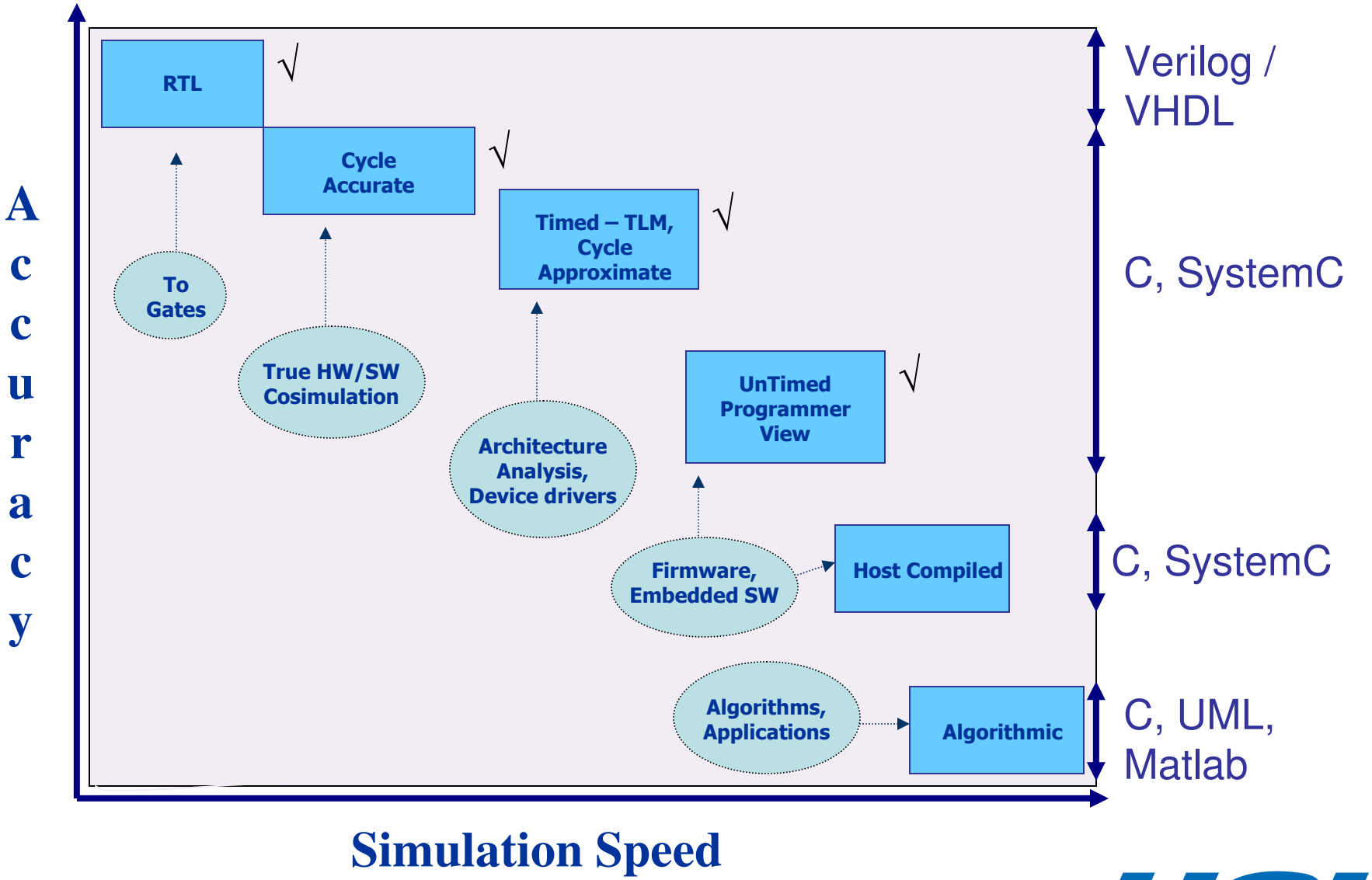
HCL

Abstraction

- ***Abstract: Conceptual, Intangible (without form), Extract (remove)***
- ***Abstraction: Generalization, Removal***
- Applied to modeling, this means that at higher abstraction we create models that are less tangible (without form), more generalized, and have some features taken out or removed.
- Advantage:
  - Higher simulation speed, leading to new applications.
  - Generalized, hence high reuse, easy to develop and maintain, and can be committed to different implementations.
- What is abstracted in a model?

HCL

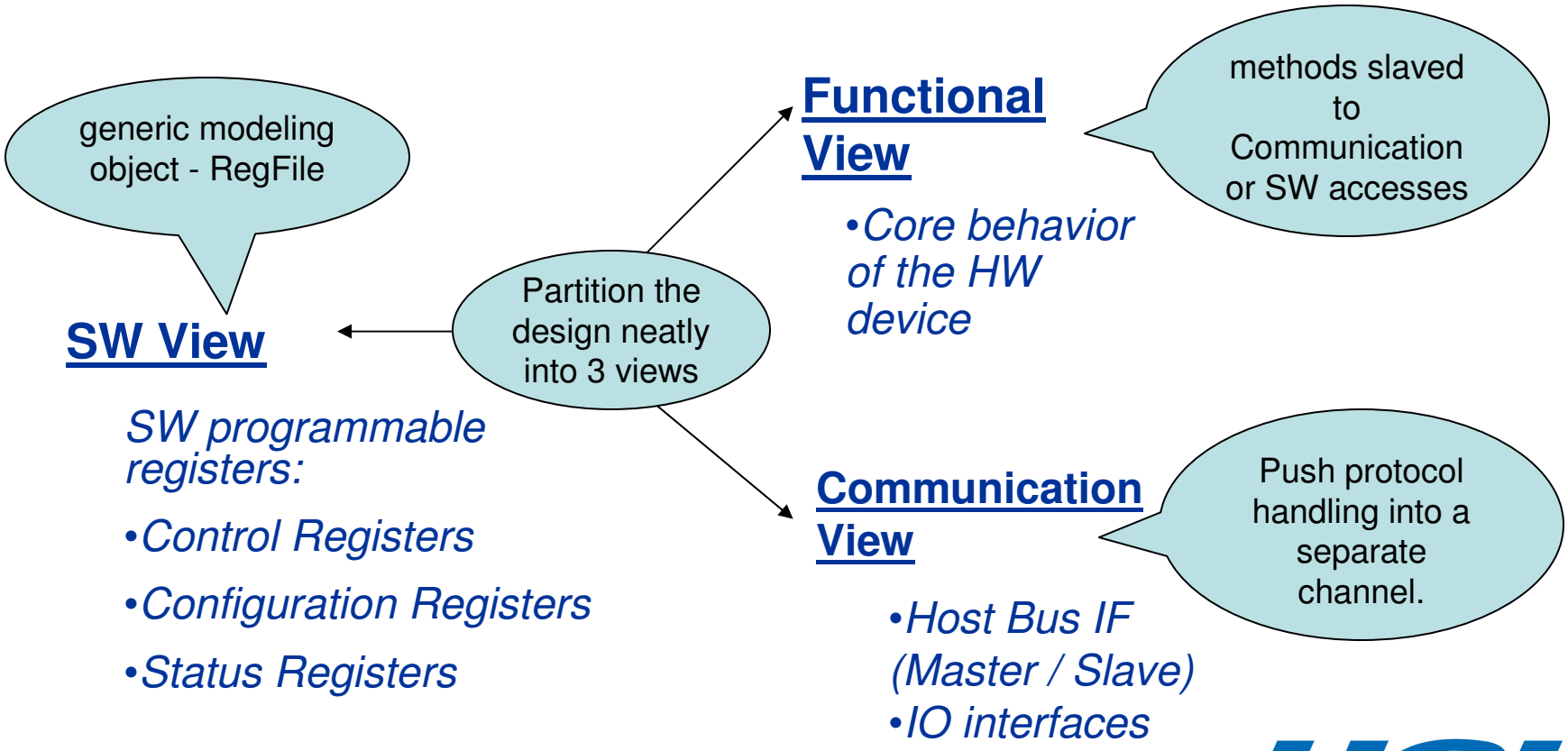
Use


Accuracy
RTL
√
Cycle Accurate
√
Timed – TLM, Cycle Approximate
√
To Gates
True HW/SW Cosimulation
Architecture Analysis, Device drivers
UnTimed Programmer View
√
Firmware, Embedded SW
Host Compiled
Algorithms, Applications
Algorithmic
Verilog / VHDL
C, SystemC
C, SystemC
C, UML, Matlab
Simulation Speed

HCL

Re-use

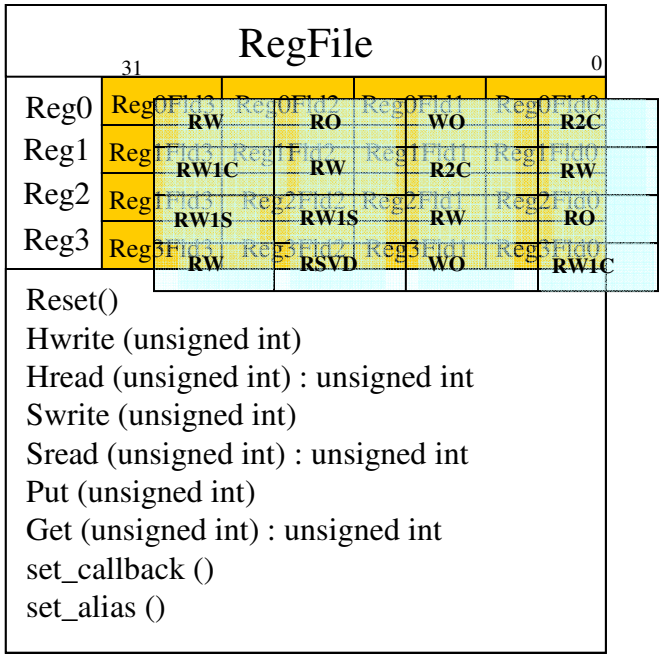
- **Reuse from one peripheral model to another**
  - **Of the same type (say. DMAC to another DMAC)**
  - **Of different types (say. Asynch. to Synch. serial communication)**
- **Reuse from one abstraction level to another**



HCL

RegFile: Generic SW View modeling object

- General purpose reusable design used to model the register set.
  - Number of registers, their R/W attributes is a property of the SW View.
  - Individual fields (that map to functionality) is a property of the SW View.
  - Reset values of the various fields is a property of the SW view.
- Functionality can register call-backs.
- Functionality can be slaved to SW accesses through methods sensitive to events generated by the RegFile

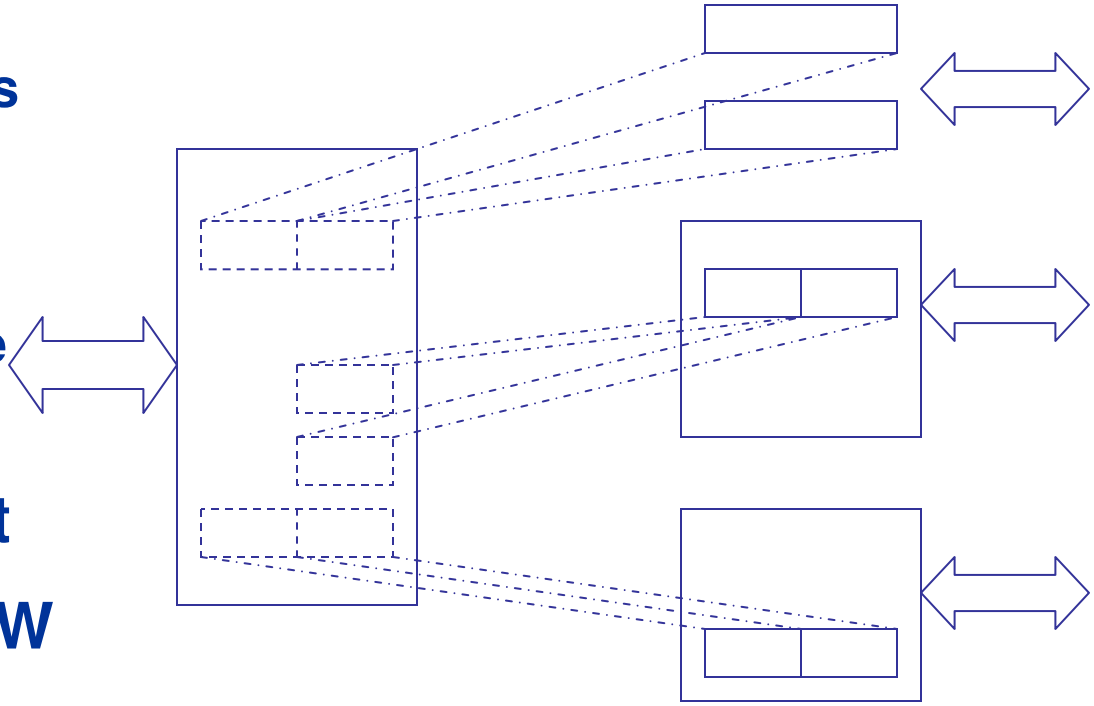


A SW view (RegFile) once configured, can be re-used across all abstraction levels.

HCL

RegFile: (cont.)

- **RegFile can Support structural partitions and hierarchical designs.**
  - ➢ **Address decoding is taken care of by the RegFile object.**
  - ➢ **Same properties are extended.**
- **RegFile can support "over-laying" one SW view over another.**

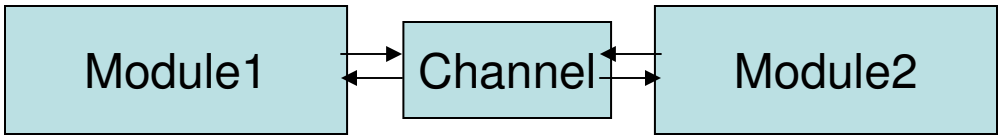


A SW view can be re-used from one peripheral model to another

HCL

Communication Modeling

- Separate the communication from functionality.
- Push communication (protocol) into a separate block (channel).

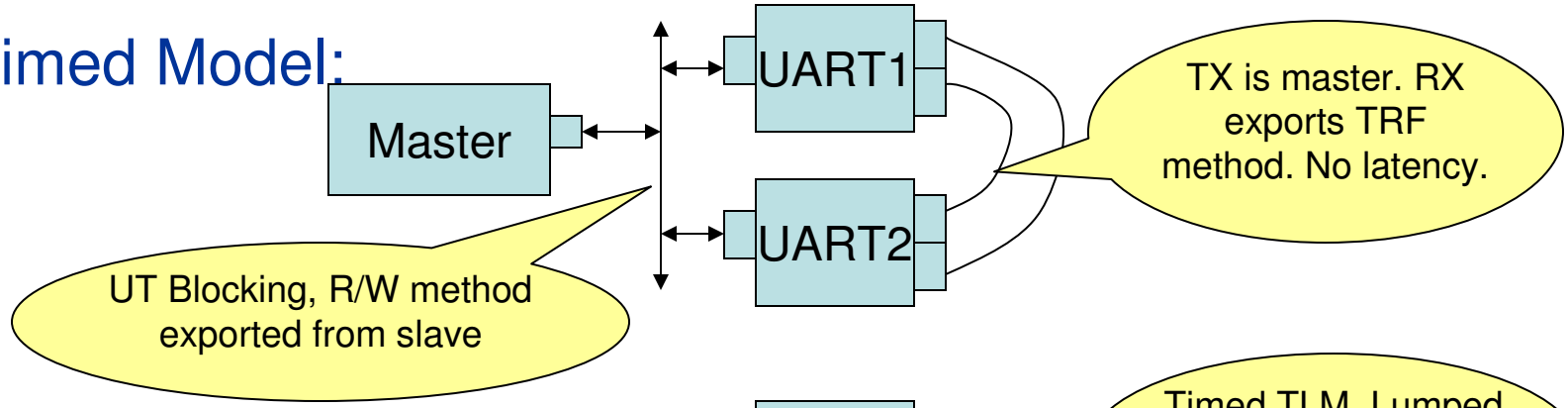


- UnTimed model: Blocking / non-blocking IF Method
- Timed TLM model: Non-blocking IF method. Slave the functionality to events available in the channel.
- BCA model: Map the transactional attributes to pin-activity.
- ***When Communication and Functionality are separated:***
  - ***Functionality can be re-used across different abstraction levels.***
  - ***Communication can be re-used from one peripheral model to another.***

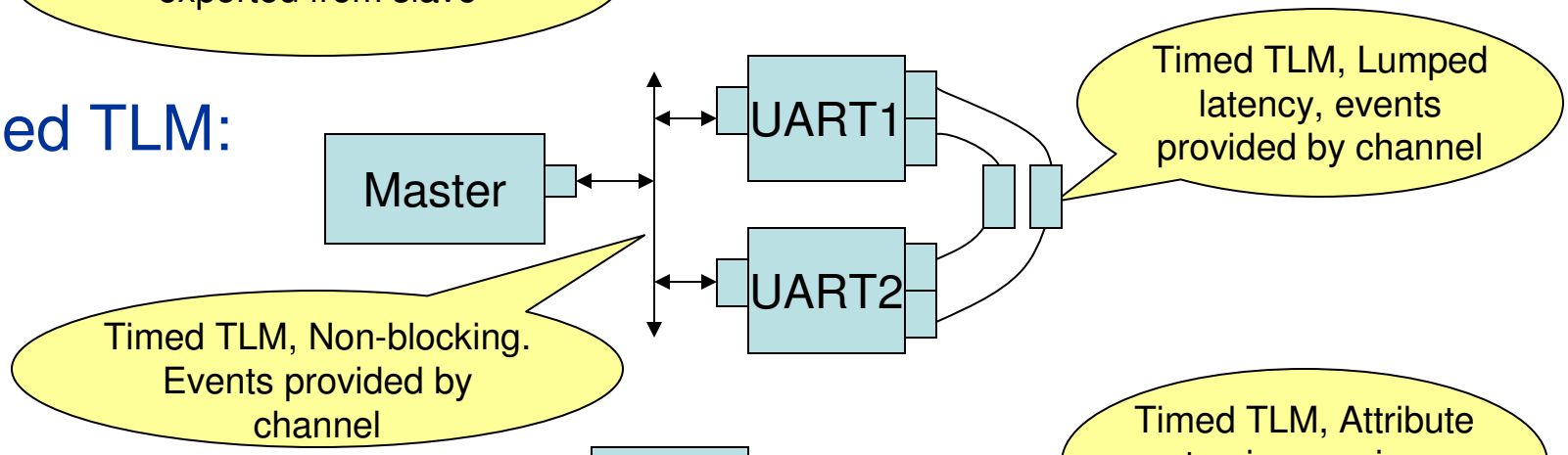
HCL

Case Study: UART model

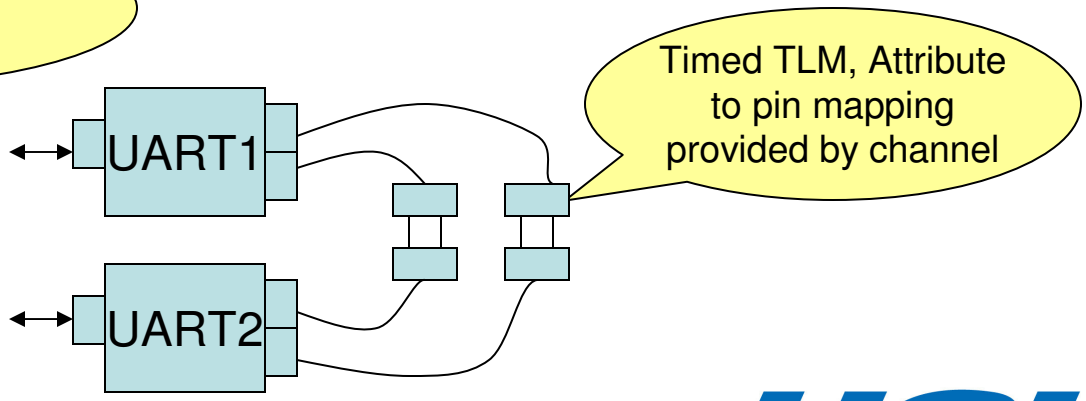
- UnTimed Model:



- Timed TLM:



- Timed TLM with BCA IF:



HCL

Case Study: UART model

- Simulation Speed:

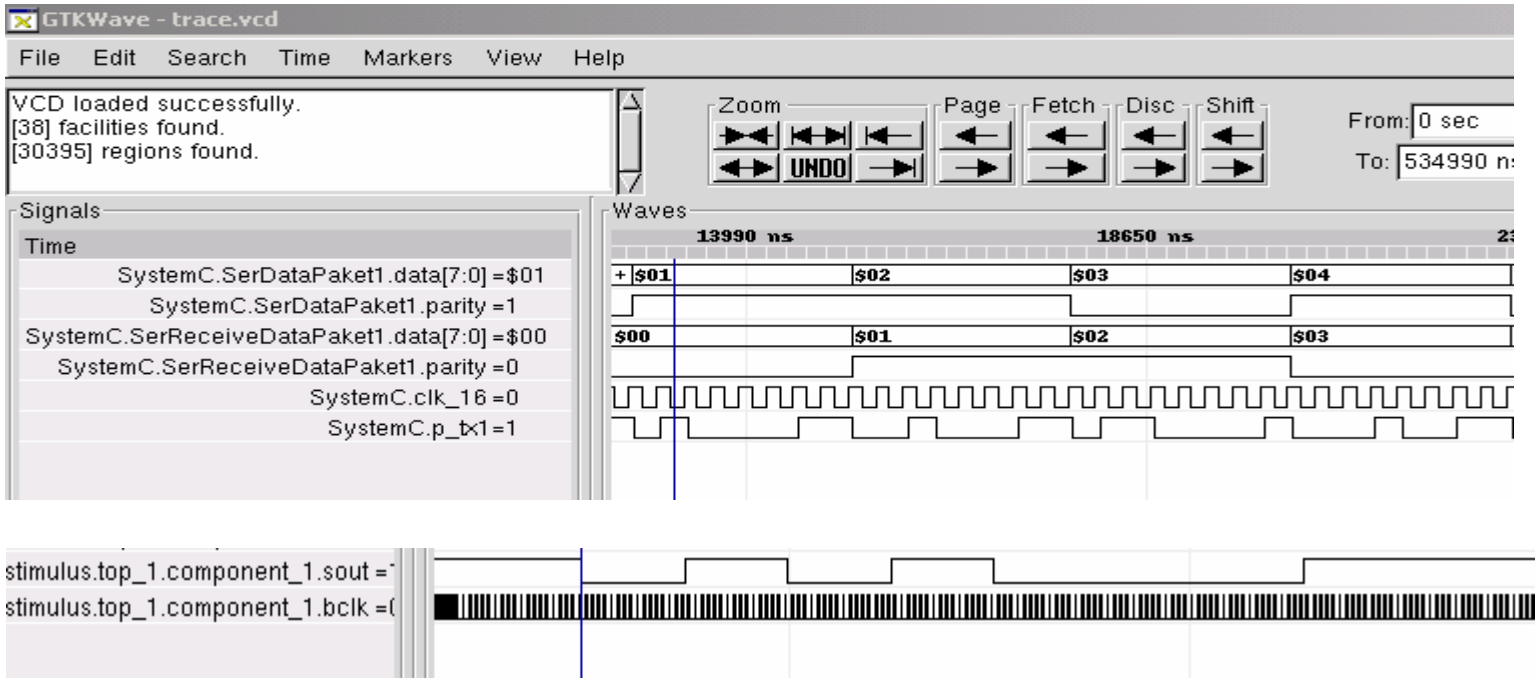
Abstraction	Speed
UnTimed	820,000 cps
Timed TLM	370,000 cps
Timed TLM with BCA IF	108,000 cps

- Baseline:
  - Single SC_Thread = 1.149 million cps
  - UT master, node, UT slave = 1.03 million cps
  - 2 SC_Threads = 775,000 cps
  - TLM master, node, TLM slave = 534,000 cps

HCL

Case Study: UART model

- Waveforms:



- Code Re-use:
  - 100% in RegFile.
  - 90% in UART methods.
  - 70% in UART channel.

HCL


HCL