



# **“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](mailto:sameera@hcltech.com))**

# Purpose

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

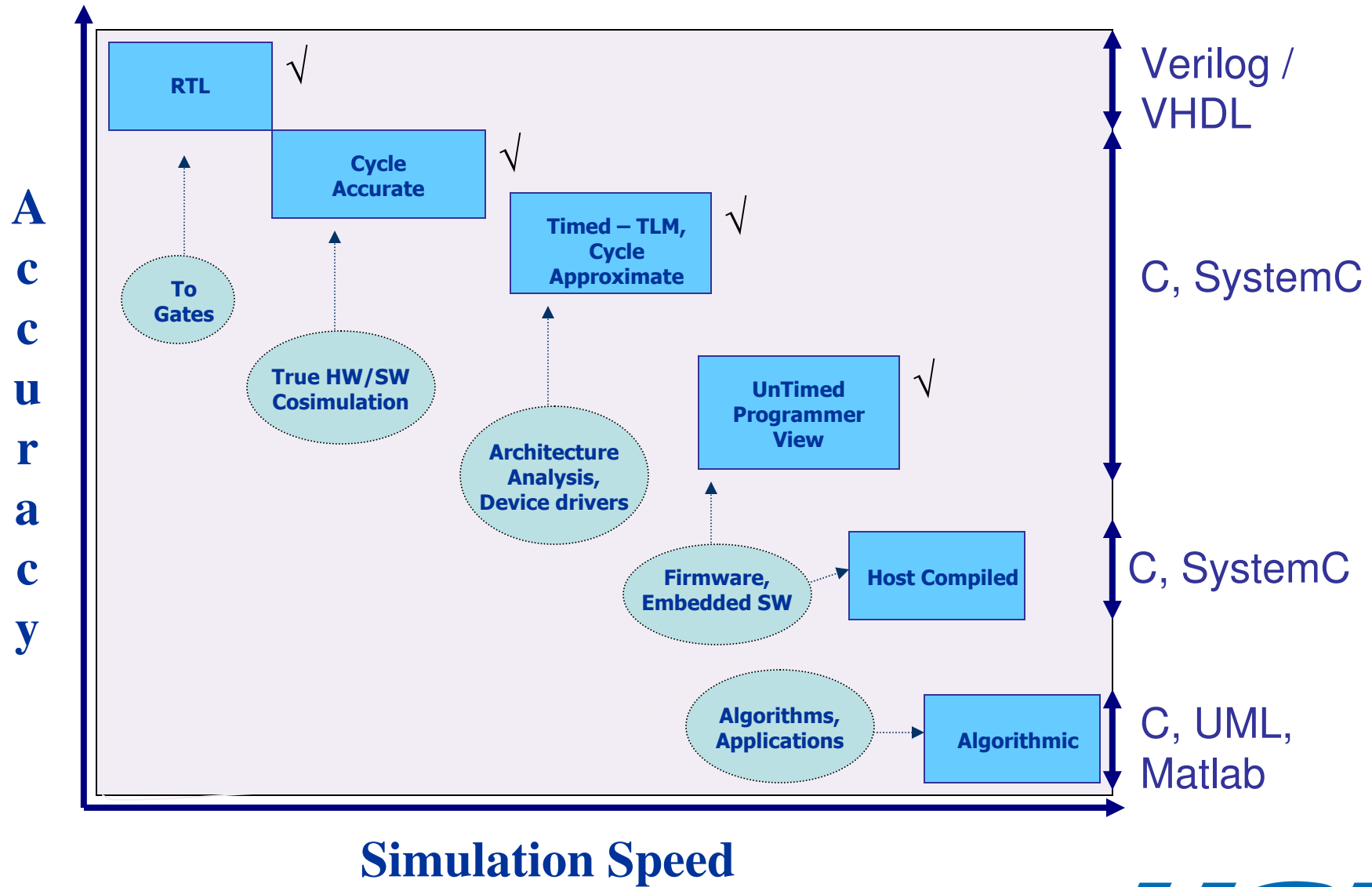
- o What is abstraction, Why abstract design models.
- o Various abstraction levels, and their use cases.
- o Re-use from one abstraction level to another, and from one model to another.

***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.***

# Abstraction

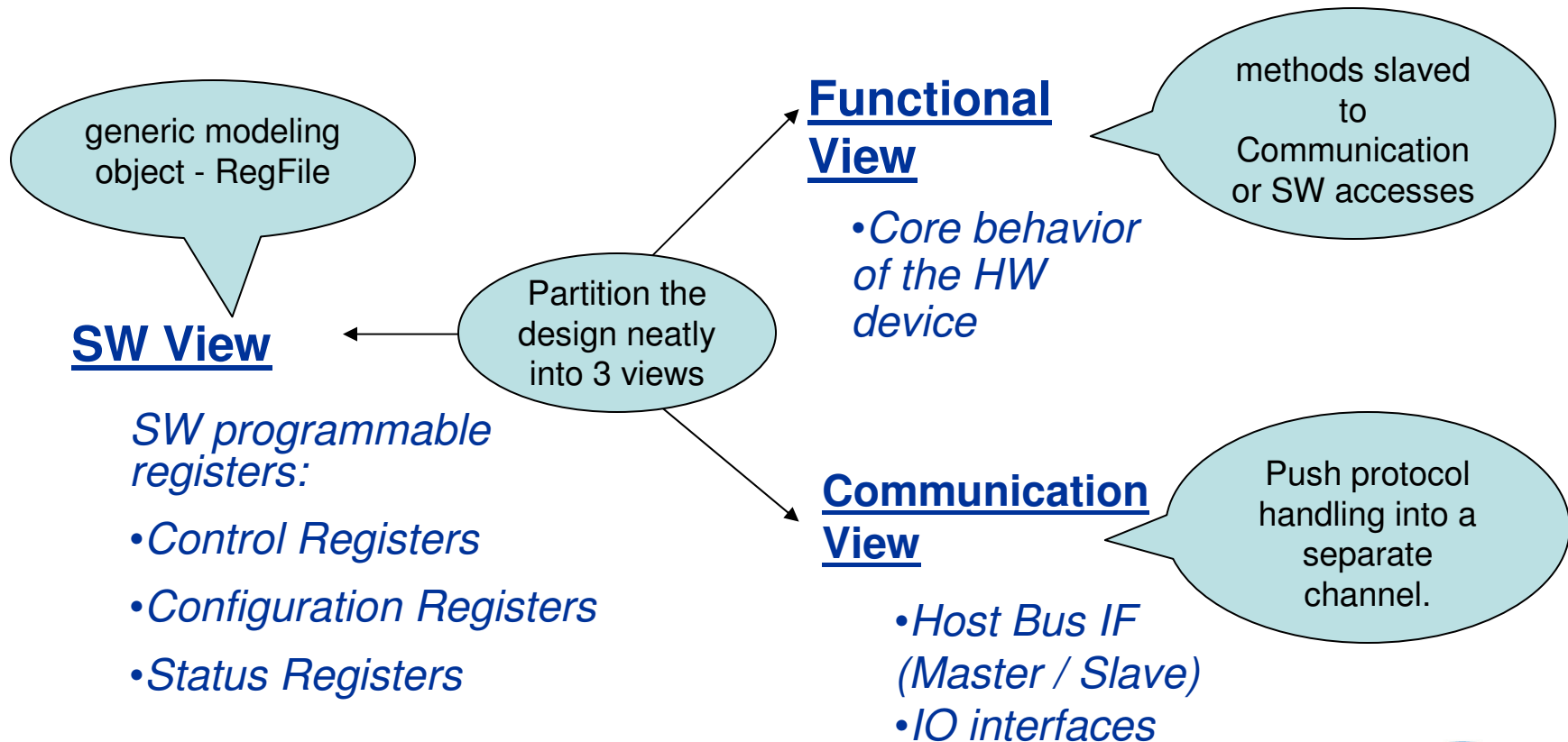
- ***Abstract: Conceptual, Intangible (without form), Extract (remove)***
- ***Abstraction: Generalization, Removal***
- Applied to modeling, this means that at higher abstraction we create models that are less tangible (without form), more generalized, and have some features taken out or removed.
- 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?

# Use



# Re-use

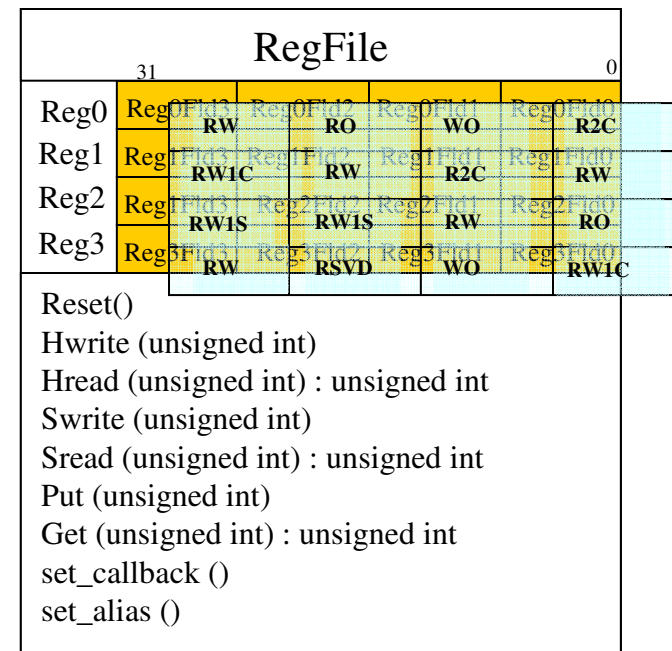
- Reuse from one peripheral model to another
  - Of the same type (say. DMAC to another DMAC)
  - Of different types (say. Asynch. to Synch. serial communication)
- Reuse from one abstraction level to another



# 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.**



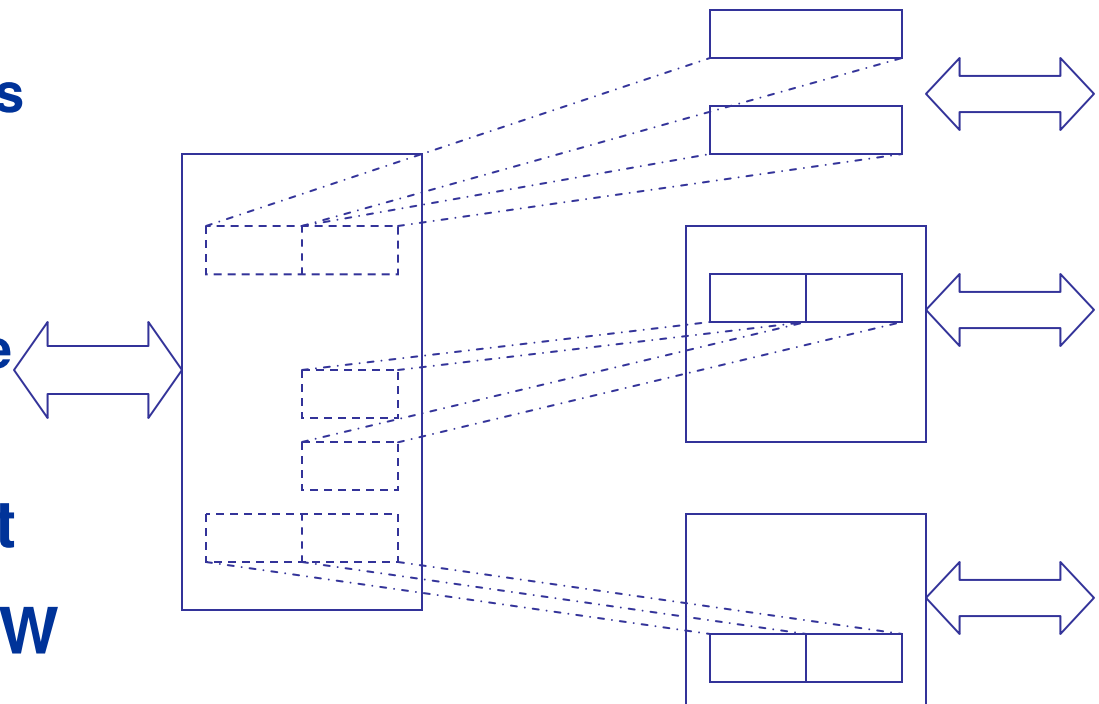
# RegFile: (cont.)

- RegFile can Support structural partitions and hierarchical designs.

- Address decoding is taken care of by the RegFile object.

- Same properties are extended.

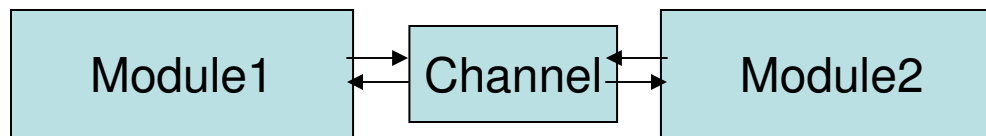
- RegFile can support “over-laying” one SW view over another.



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

# Communication Modeling

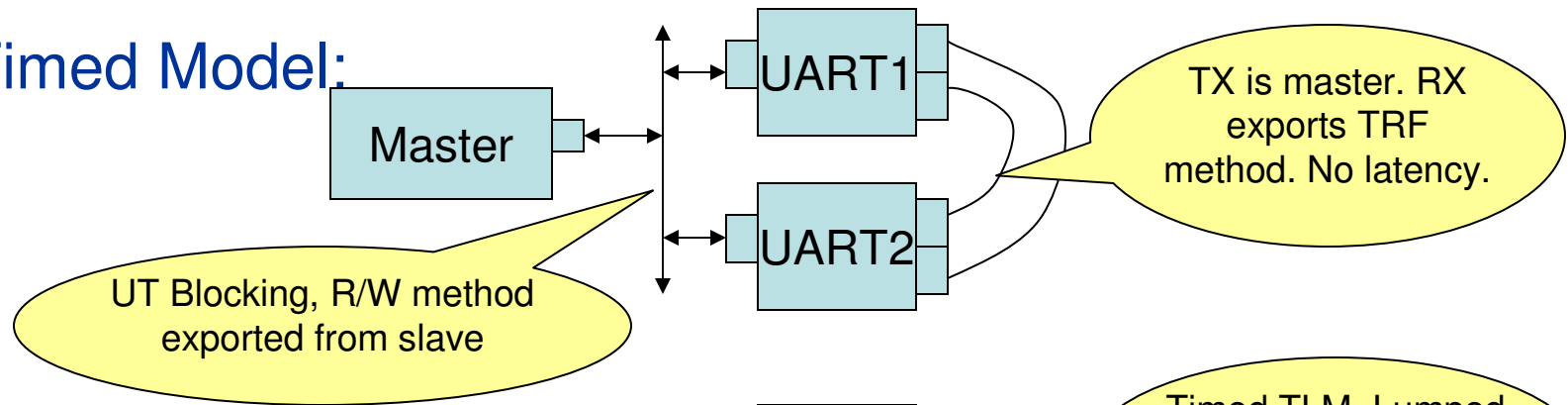
- o Separate the communication from functionality.
- o Push communication (protocol) into a separate block (channel).



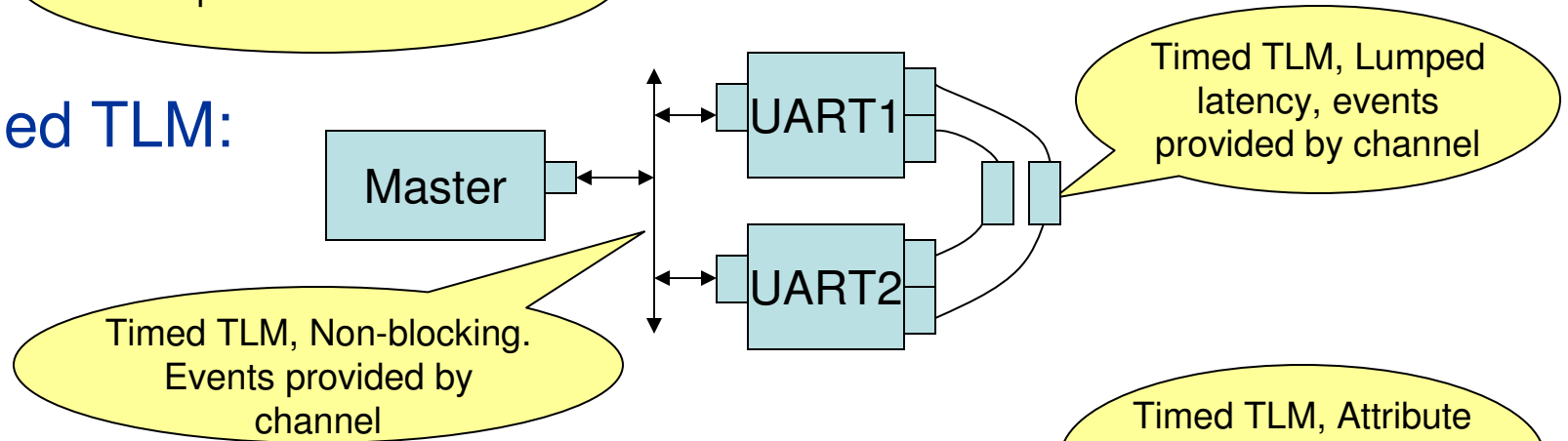
- o UnTimed model: Blocking / non-blocking IF Method
- o Timed TLM model: Non-blocking IF method. Slave the functionality to events available in the channel.
- o BCA model: Map the transactional attributes to pin-activity.
- o **When Communication and Functionality are separated:**
  - o **Functionality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

# Case Study: UART model

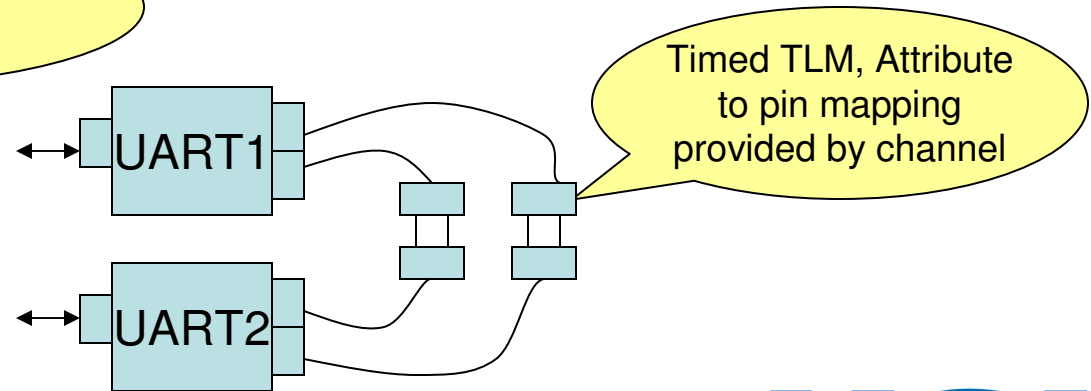
## o UnTimed Model:



## o Timed TLM:



## o Timed TLM with BCA IF:



# Case Study: UART model

## o Simulation Speed:

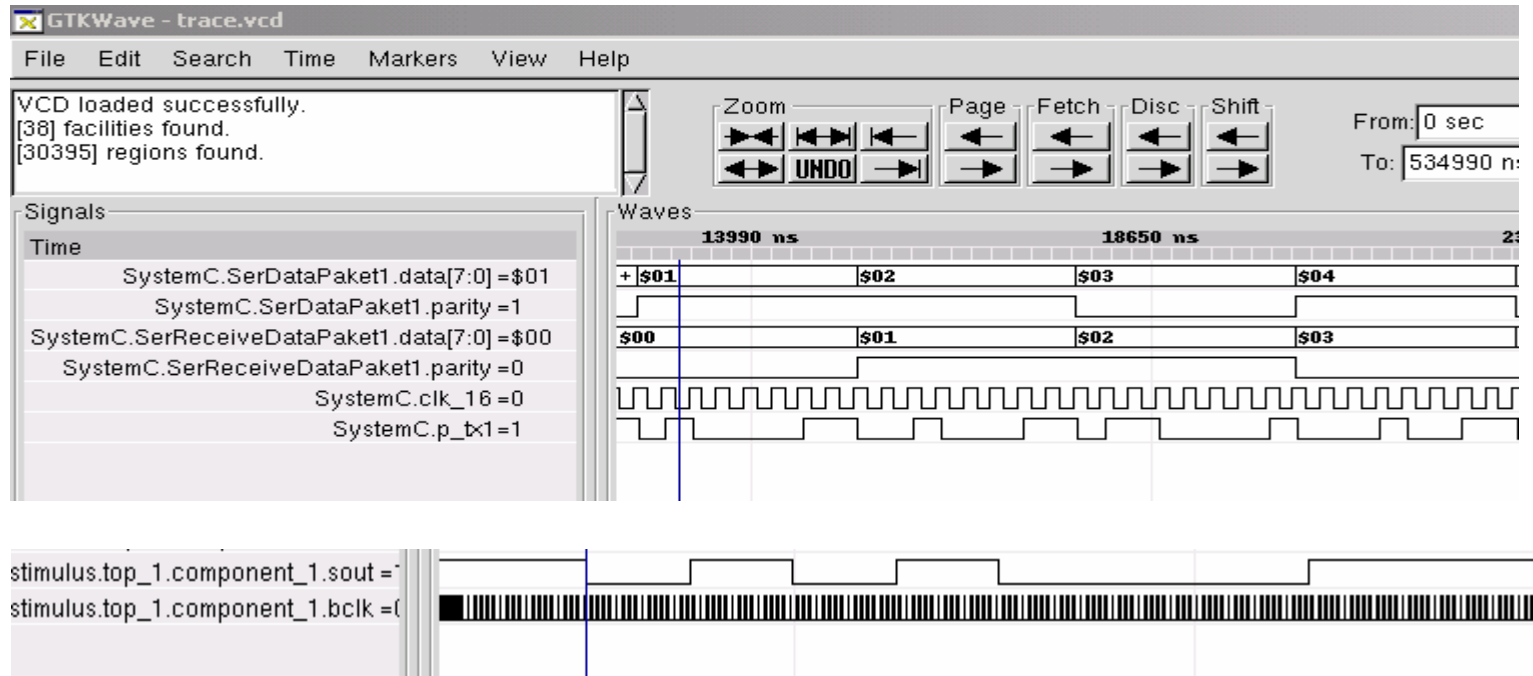
Abstraction	Speed
UnTimed	820,000 cps
Timed TLM	370,000 cps
Timed TLM with BCA IF	108,000 cps

## o Baseline:

- o Single SC\_Thread = 1.149 million cps
- o UT master, node, UT slave = 1.03 million cps
- o 2 SC\_Threads = 775,000 cps
- o TLM master, node, TLM slave = 534,000 cps

# Case Study: UART model

## o Waveforms:



## o Code Re-use:

- o 100% in RegFile.
- o 90% in UART methods.
- o 70% in UART channel.

---

***HCL***