How to create adaptors for modeling abstraction levels

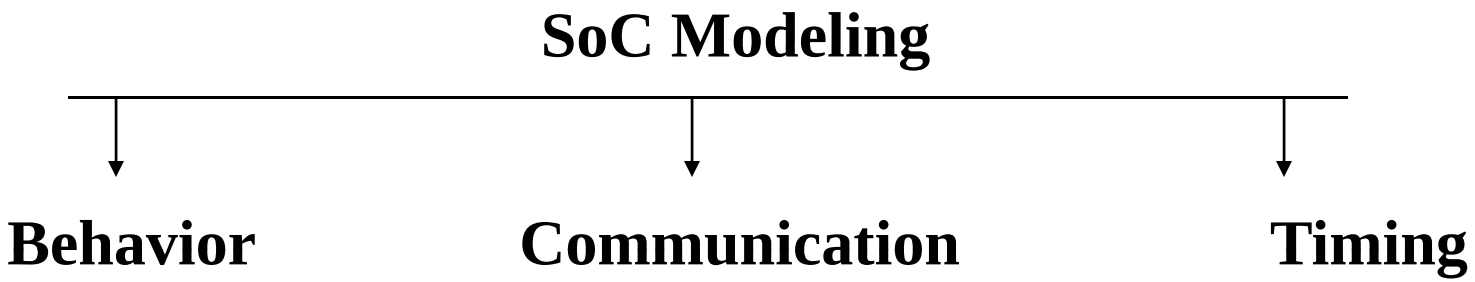
Umesh Sisodia, usisodia@circuitsutra.com
Presenting on behalf of:
Ashwani Singh, ashwani@circuitsutra.com

CIRCUIT SUTRA TECHNOLOGIES
Software Partner to Semiconductors

Agenda

- SoC Modeling
- SoC Modeling Abstraction Levels
- Requirement of Adaptors
- Role/Functionality of Adaptors
- Adaptor Modeling Challenges
- Queries

SoC Modeling



Abstraction required to speed up things

- Timing granularity
- Data granularity

CIRCUIT SUTRA TECHNOLOGIES
Software Partner to Semiconductors

SoC Modeling Abstraction

Speed

OCP-IP Terminology	Other Terminology	
TL4	PV (Programmer's view)	•No timing •TLM2.0 B.P. (LT)
TL3	AV (Architect's view)	•Interburst or no timing •TLM-2.0 B.P. (AT)
TL2		•Intra burst or no timing •Extends TLM-2.0 (AT)
TL1	VV (Verification view)	•Fully cycle accurate •Extends TLM-2.0 (AT) •Supports clock cycle synchronization and combinatorial paths
TL0	RTL	Signal level (not transaction level)

Timing and signal details

Requirements of Adaptors

Mixed-Abstraction-Layered System

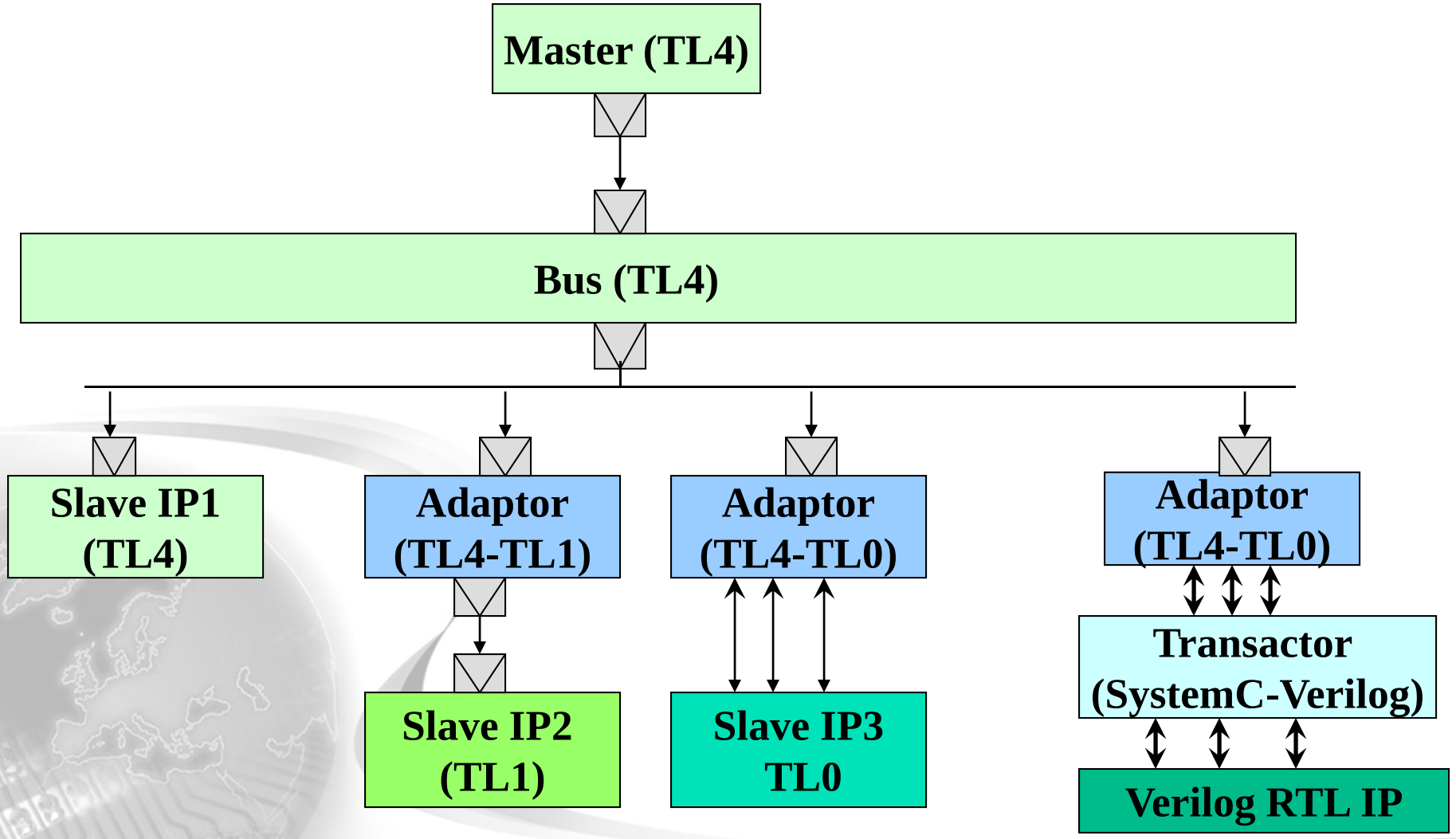
- While creating the VP at higher abstraction level, sometime it is necessary to take the pin-level RTL model of a specific IP block. These RTL models might be automatically generated by some tool (like Carbon), thus saving the model creation time.

- While creating the VP for a specific use case, to speed up things, it may be a good idea to reuse the existing models at other abstraction levels. This will enable to get the platform working quickly, and then models can be replaced one by one with the correct abstraction.

- Adaptors and transactors will also be required for HW/SW co-verification while using the virtual platform at a higher abstraction level along with the advanced RTL verification environment.

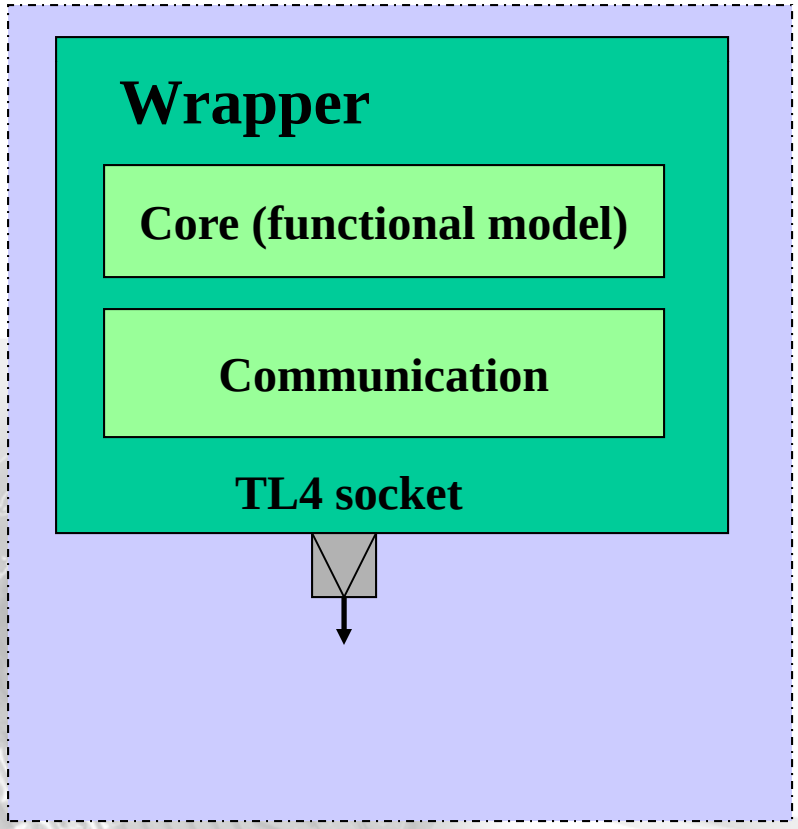
Requirements of Adaptors

Mixed-Abstraction-Layered System



Requirements of Adaptors

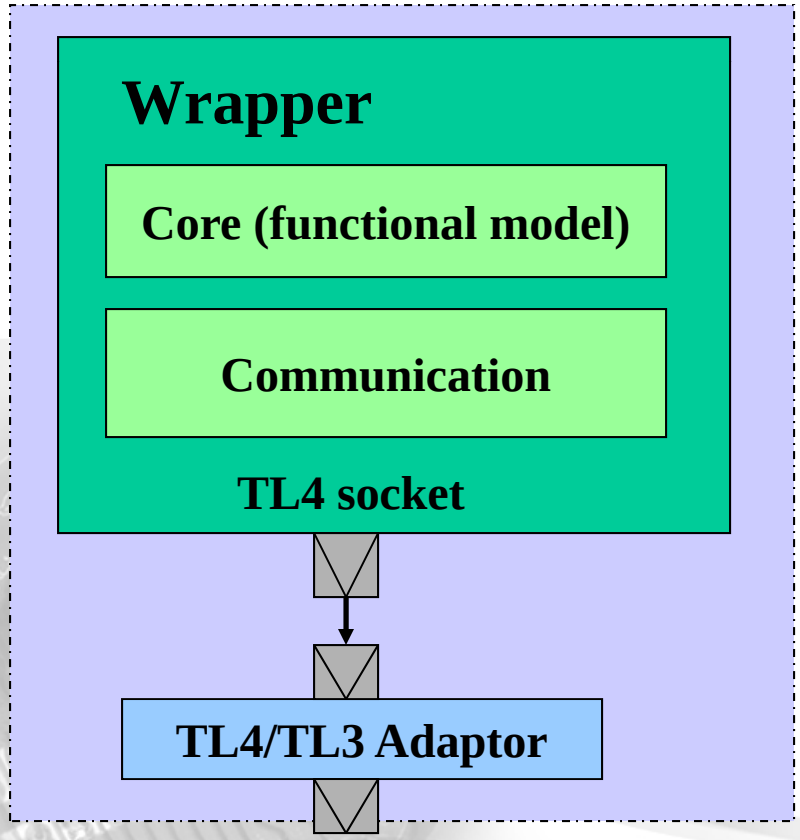
Encourages the reusability of code: The separation of computation and communication allows the reusability of code across abstraction levels. The TL4 model can be used in combination with the proper adaptor to work at different abstraction levels.



- TL4 Model
- Used for eSW development

Requirements of Adaptors

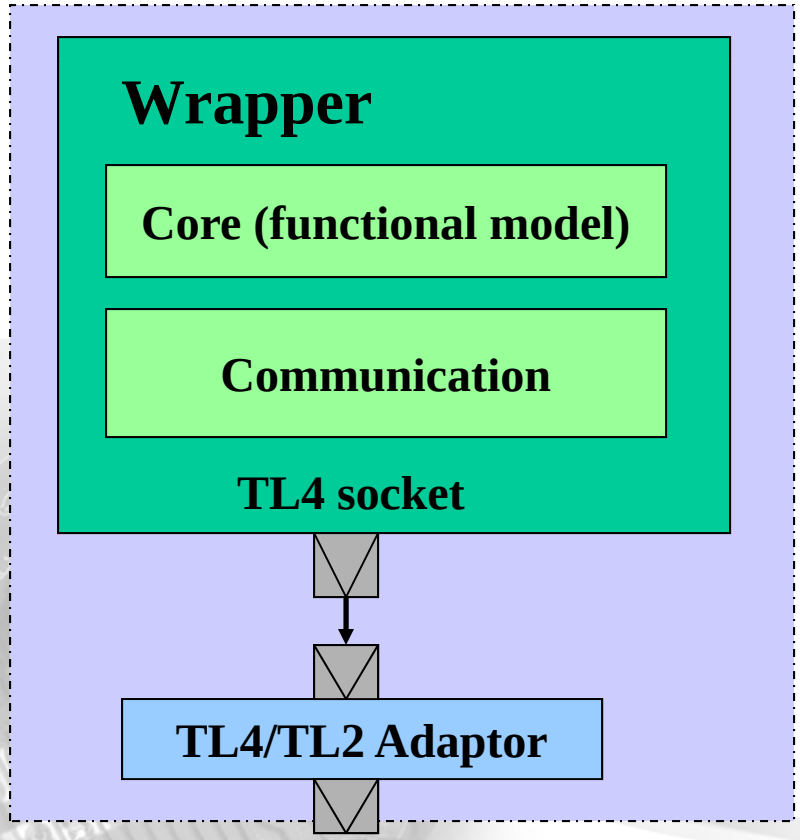
Encourages the reusability of code: The separation of computation and communication allows the reusability of code across abstraction levels. The TL4 model can be used in combination with the proper adaptor to work at different abstraction levels.



- TL3 Model
- Used for Architectural exploration
- Used for Performance analysis

Requirements of Adaptors

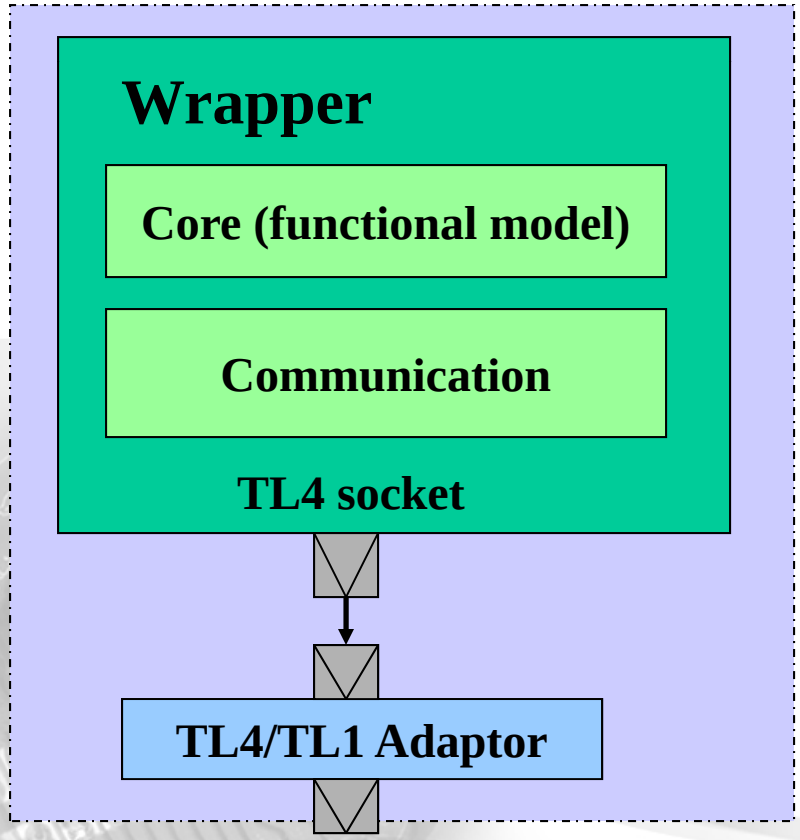
Encourages the reusability of code: The separation of computation and communication allows the reusability of code across abstraction levels. The TL4 model can be used in combination with the proper adaptor to work at different abstraction levels.



- TL2 Model
- Used for Architectural exploration
- Used for Performance analysis

Requirements of Adaptors

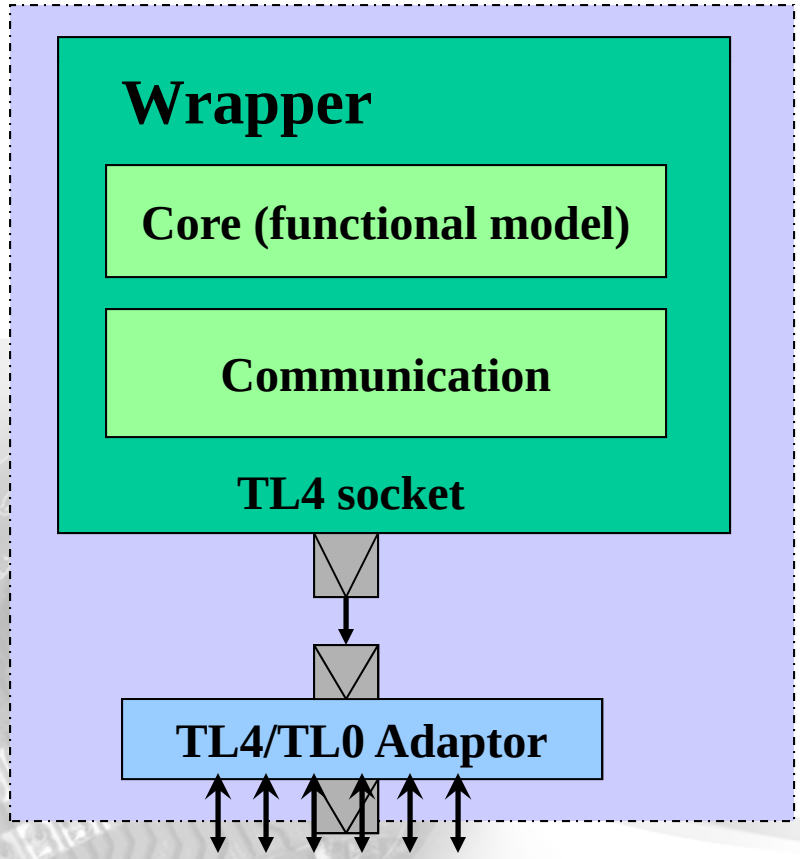
Encourages the reusability of code: The separation of computation and communication allows the reusability of code across abstraction levels. The TL4 model can be used in combination with the proper adaptor to work at different abstraction levels.



- TL1 Model
- Cycle-accurate TLM model
- Used for Architectural exploration
- Used for Performance analysis
- HW/SW co-verification

Requirements of Adaptors

Encourages the reusability of code: The separation of computation and communication allows the reusability of code across abstraction levels. The TL4 model can be used in combination with the proper adaptor to work at different abstraction levels.



- TL0 Model
- Cycle-accurate pin-level model
- HW/SW co-verification

Kinds of Adaptors

DownStream Adaptors

Master	Slave
TL4 →	TL3
TL3 →	TL2
TL2 →	TL1
TL1 →	TL0

UpStream Adaptors

Master	Slave
TL3 →	TL4
TL2 →	TL3
TL1 →	TL2
TL0 →	TL1

Adaptor's role:

- Insert the extra timing points (BP/extended phases)
- Add more payload data members (using the extensions)

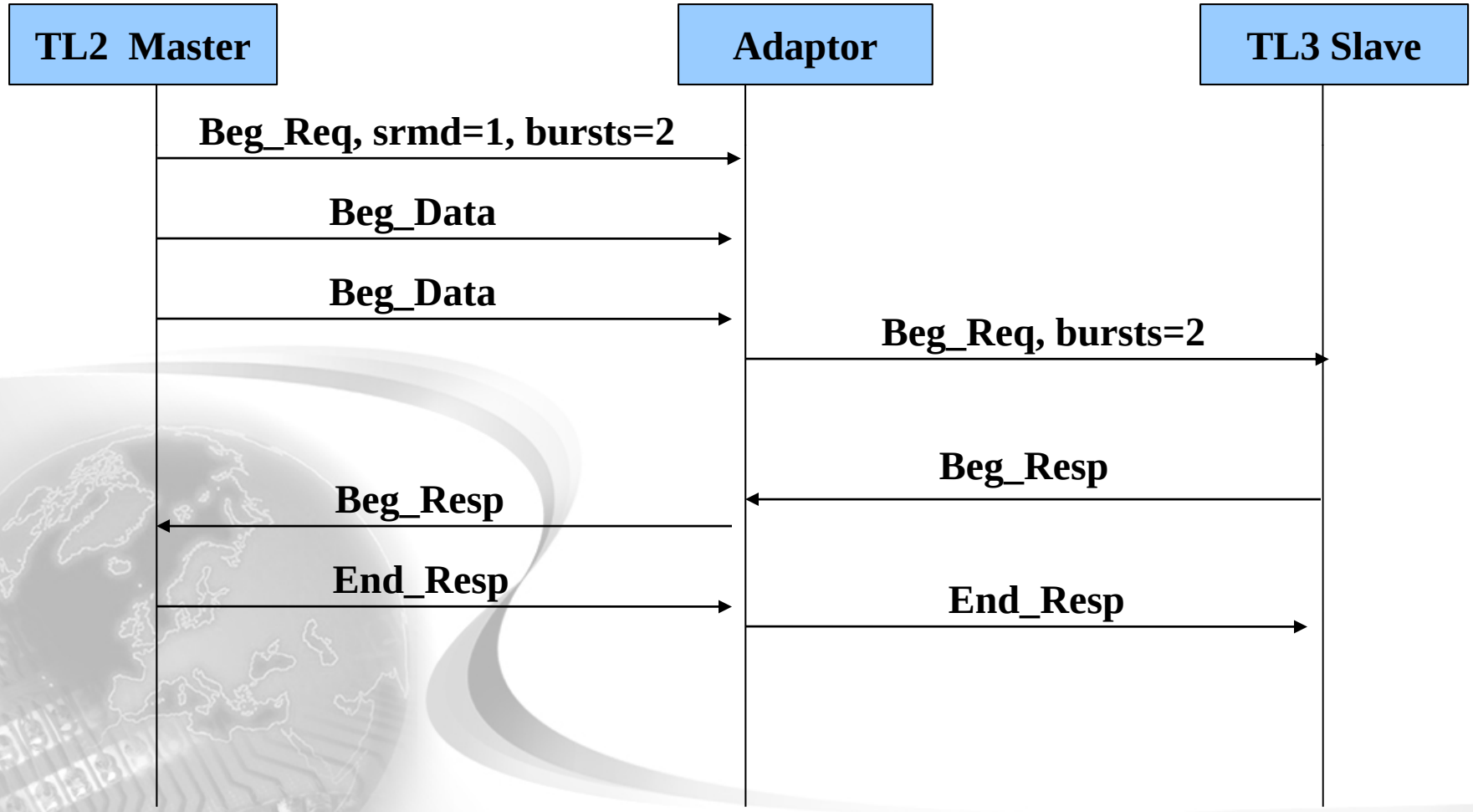
Adaptor's role:

- Abstract away the timing points (which are phases in TLM-2.0)
- Abstract away the data members (normal payload/extended) which are not necessary

Adaptor Modeling Challenges

Handling Timing Abstraction

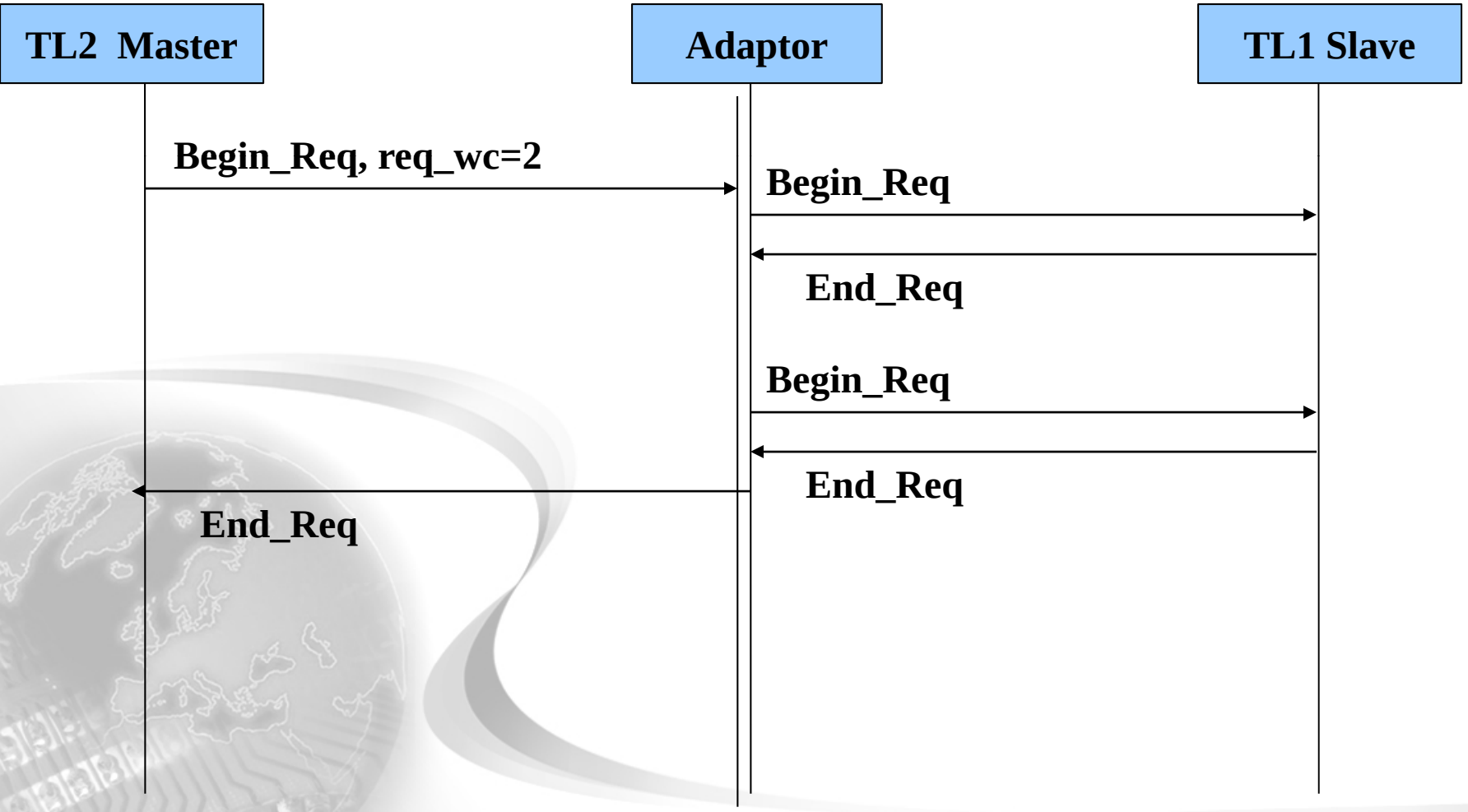
- Addition/deletion of timing points (phases)



Adaptor Modeling Challenges

Handling Timing Abstraction

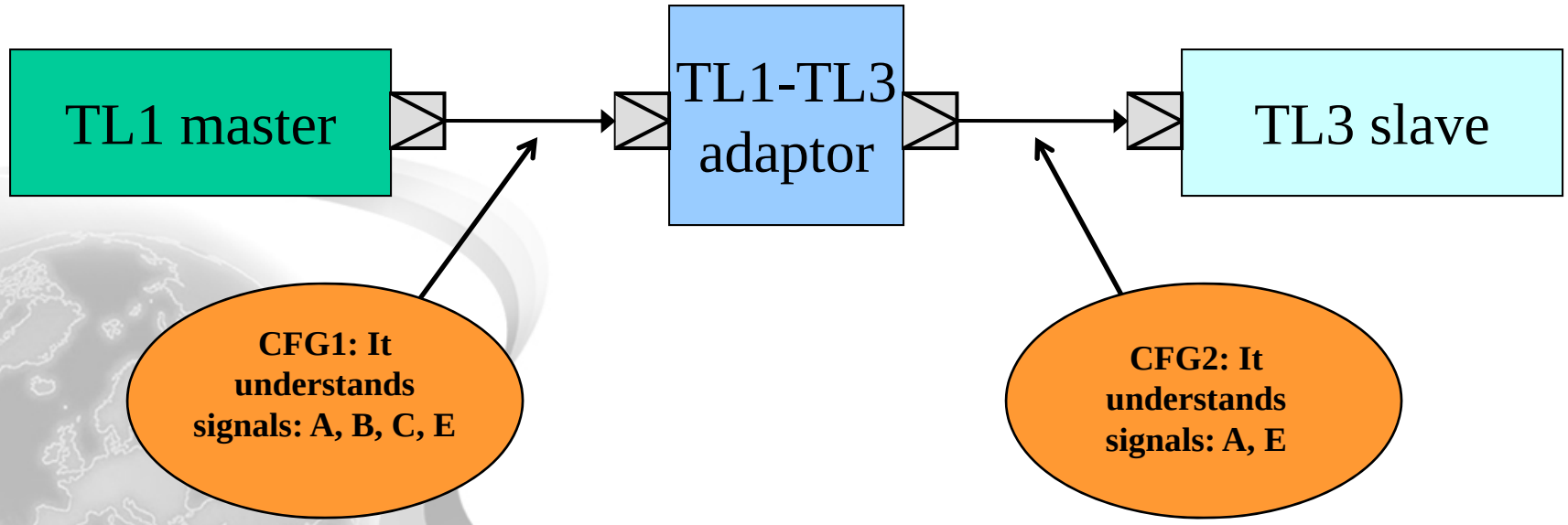
- Addition/deletion of timing points (phases)



Adaptor Modeling Challenges

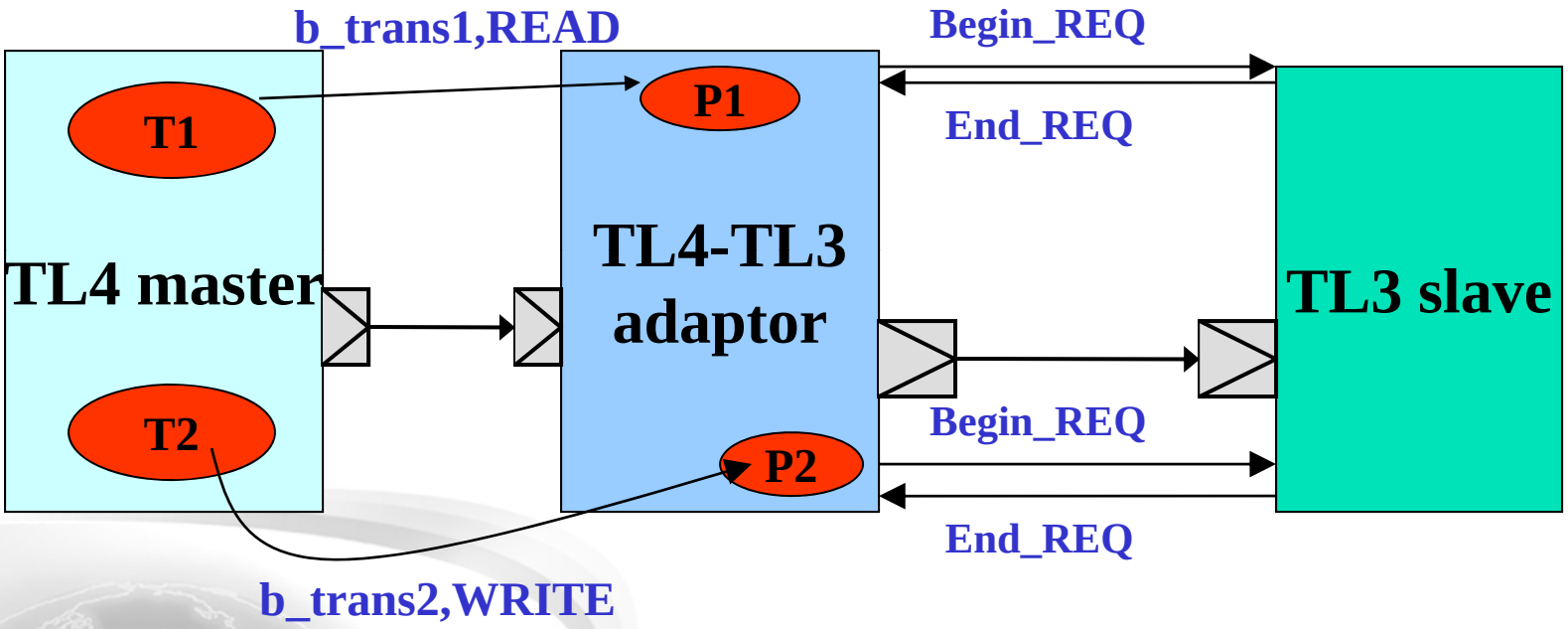
Handling Payload Extensions

Adaptors adds/deletes data extensions into the payload or validates/invalidates payload members depending upon the configuration of master or slave sockets.



Adaptor Modeling Challenges

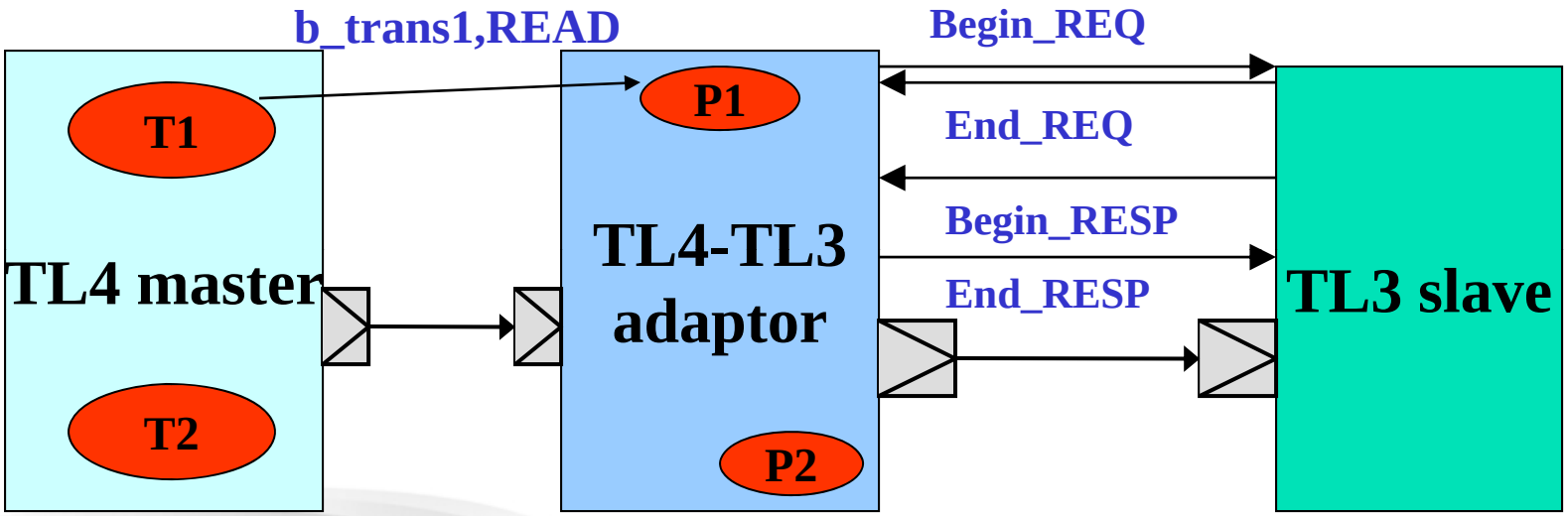
Handling Order of Transactions



Both blocking transport calls will wait until the TLM_COMPLETED for respective nb_trasport, adaptor will unblock the appropriate blocking call. TLM-2.0 ordering rules must be respected.

Adaptor Modeling Challenges

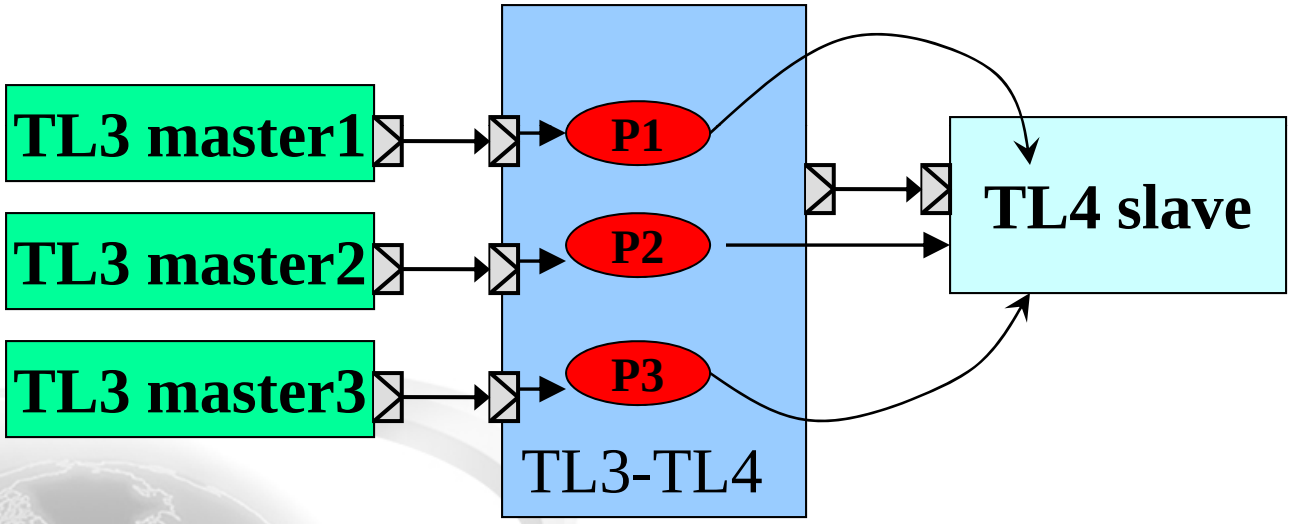
Handling Order of Transactions



Both blocking transport calls will wait until the TLM_COMPLETED for respective nb_trasport, adaptor will unblock the appropriate blocking call. TLM-2.0 ordering rules must be respected.

Adaptor Modeling Challenges

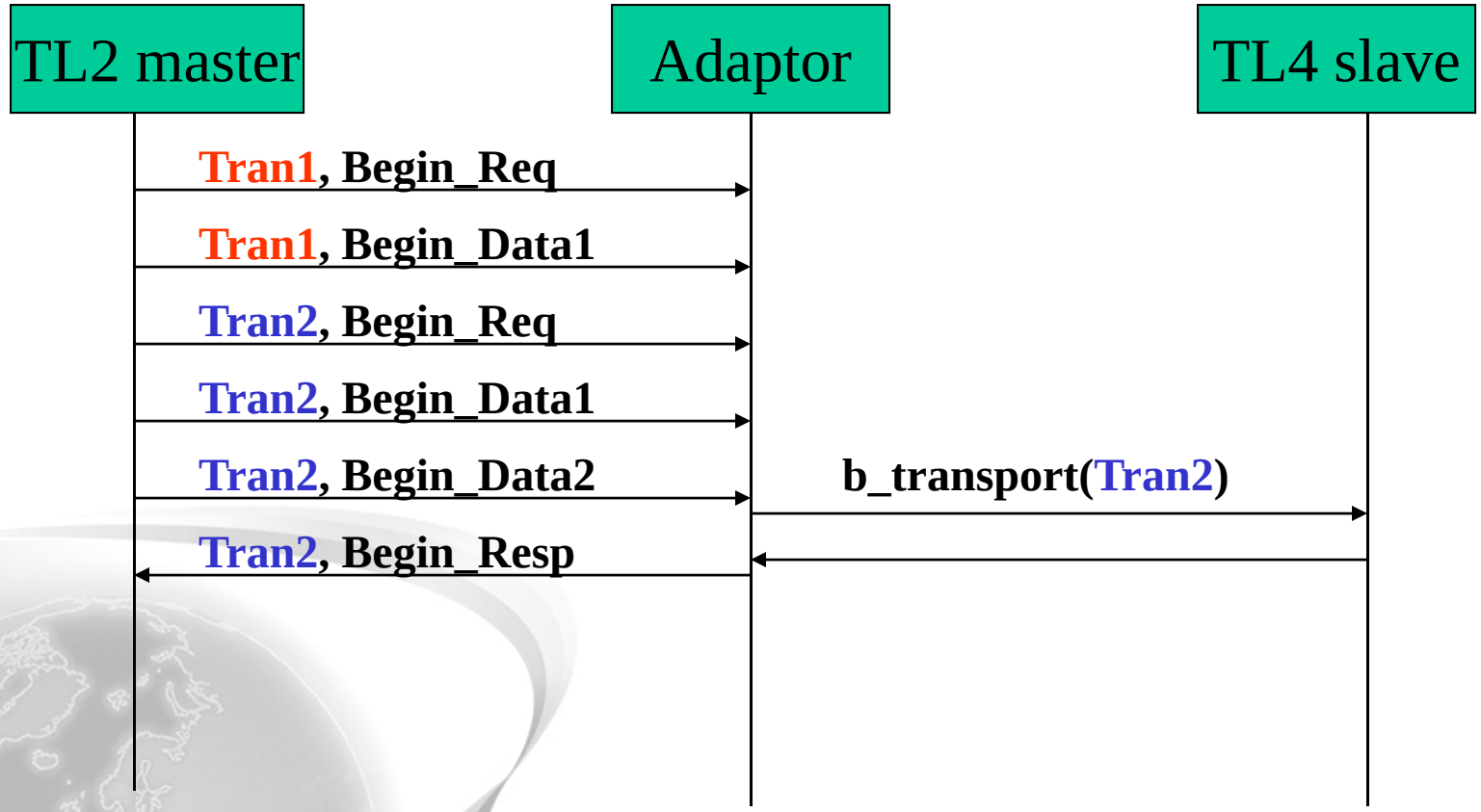
Handling b/nb and nb/b Conversion



The adapter should be able to handle concurrent non-blocking transport calls from multiple initiators. It should be able to create threads for respective blocking transport calls.

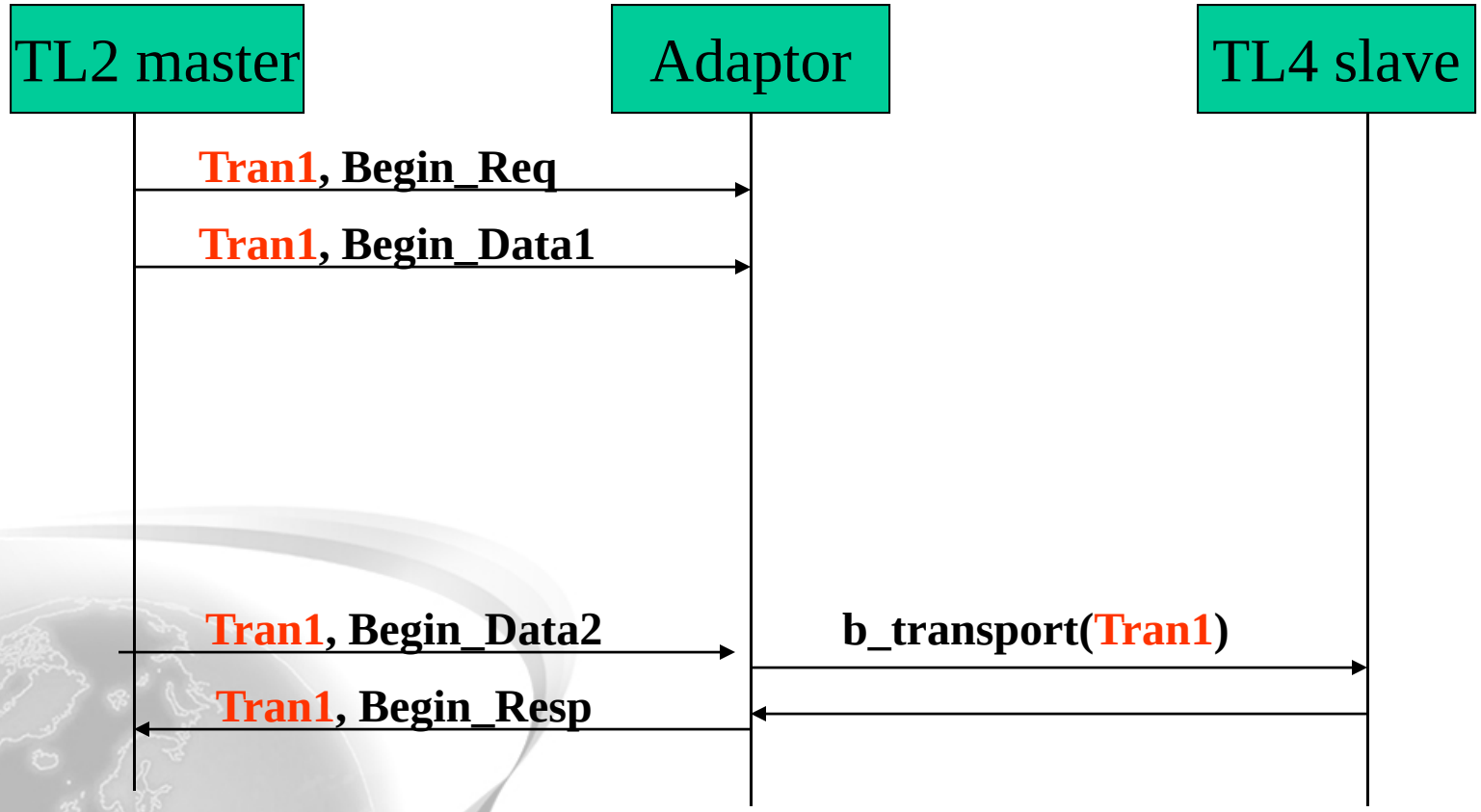
Adaptor Modeling Challenges

Handling Out-of-order/Outstanding Transactions



Adaptor Modeling Challenges

Handling Out-of-order/Outstanding Transactions



SoC Modeling Services (SystemC, TLM-2.0)
info@circuitsutra.com

Thank You

If you would like to download a copy of the presentation slides in PDF format, click the Attachments link at the top of the presentation viewer.