

# RMAP protocol SystemC model: detailed description and modelling features

Nikita Martynov - student of SUAI

# Importance of modelling

Modelling takes an important role in the development process.

## Modelling abilities:

- Detailed check of the specification
- Verification of the project
- Allow to save efforts, time and money
- Combining of a number of specified protocols into a single model

# Benefits of using SystemC

- Includes essential possibilities for functional modelling
- Based on C++ language and provide ability to use already implemented C++ high-level models for further hardware modelling

# SystemC hardware modelling abilities

- Modules and Hierarchy
- Hardware Data Types
- Methods and Threads(backbone for hardware modelling, represent processes in SystemC)
- Events, Sensitivity
- Interfaces and Channels

# SpaceWire protocol

SpaceWire is a spacecraft communication embedded network, which is developed and supported by large companies like ESA, NASA, JAXA.

The purposes of SpaceWire are:

- to facilitate the construction of high performance on board data handling systems;
- to reduce system integration costs;
- to promote compatibility between data handling equipment and subsystems;
- to encourage reuse of data handling equipment across several different missions.

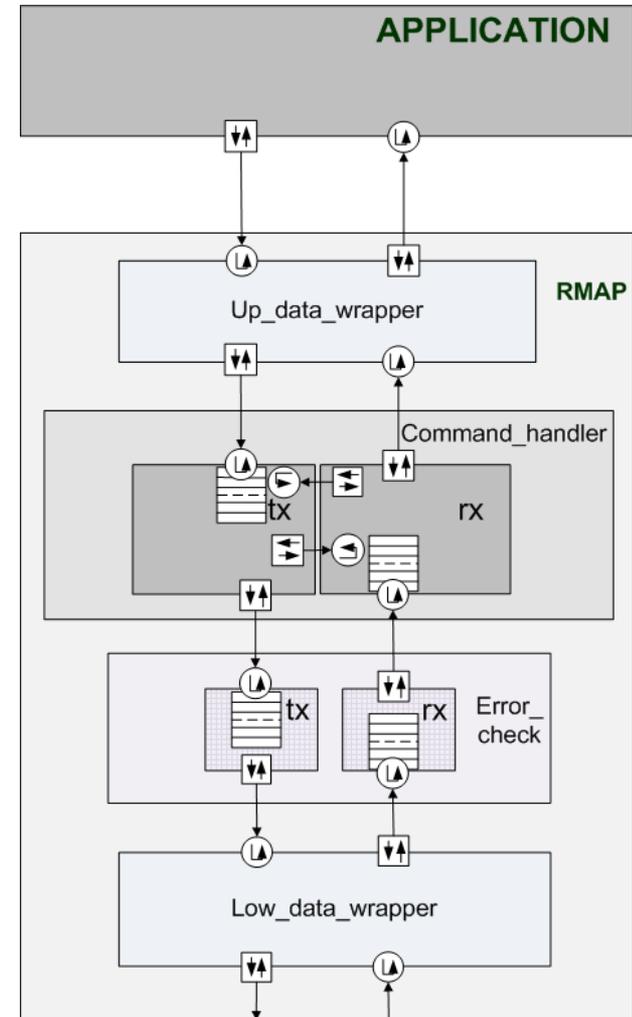
# Remote Memory Access Protocol (RMAP)

## Features:

- supports wide range of SpaceWire applications;
- configures SpaceWire network;
- controls SpaceWire units;
- gathers data and status information from those units;
- may operate alongside other communications protocols running over SpaceWire;

# RMAP Model

- Implementation based on SystemC modelling abilities;
- Event-oriented;
- Implemented with the accurate respect to RMAP specification;

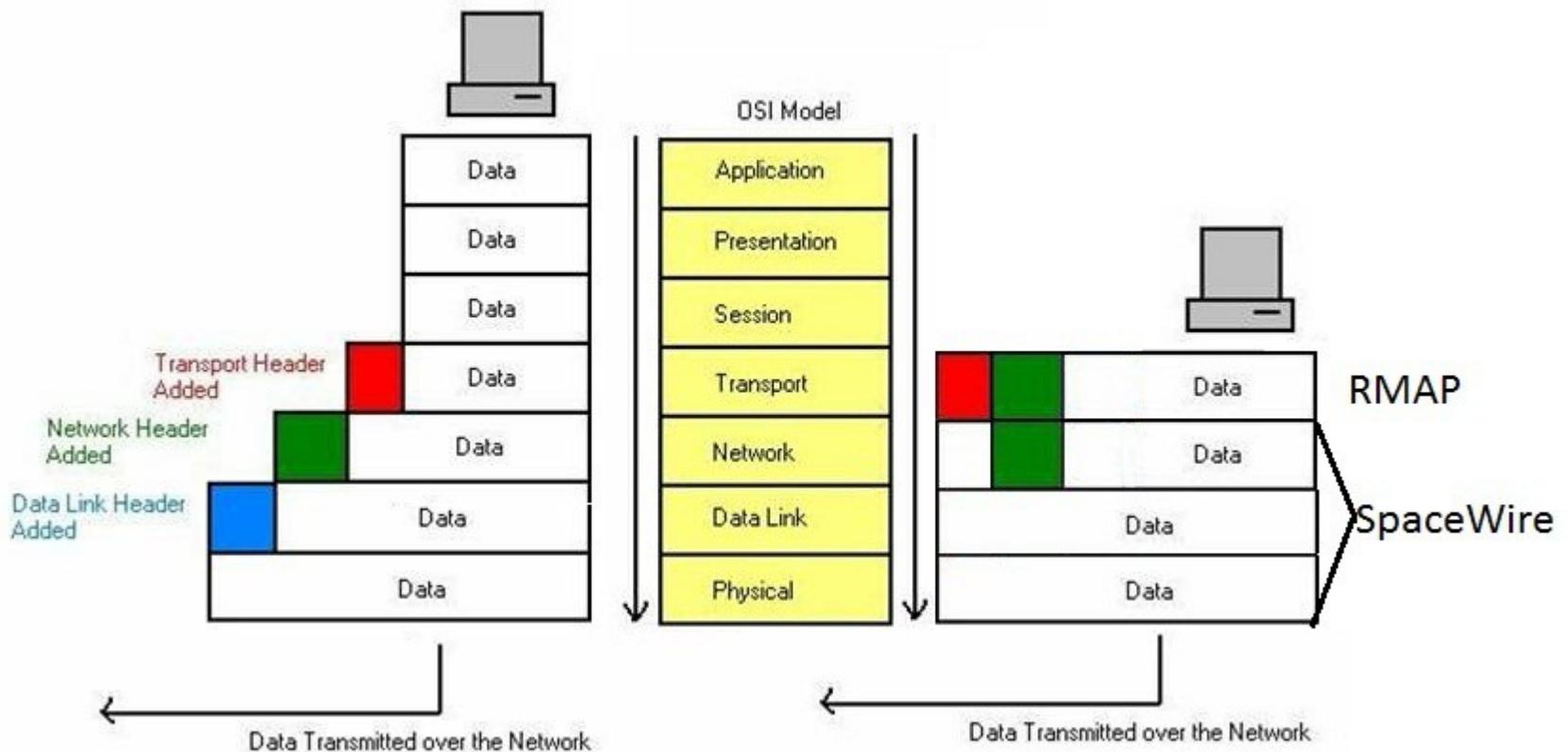


# RMAP modelling features

- A vast number of tests
- Variety of functions provided by RMAP interface to upper layer
- Modelling and testing was based on iterative approach to development

# RMAP modelling features

- A vast number of tests



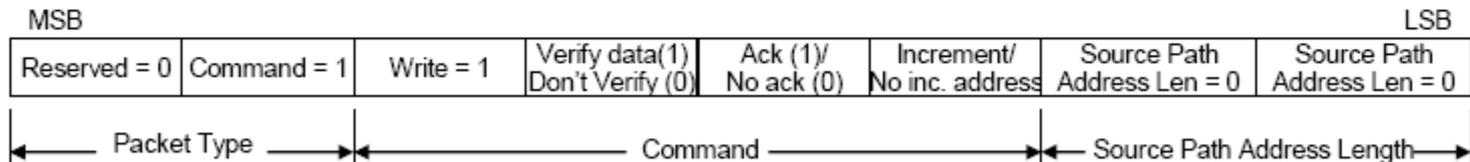
# RMAP modelling features

- A vast number of tests

*First byte transmitted*

Destination Logical Address	Protocol Identifier	Packet Type, Command, Source Path Addr Len	Destination Key
Source Logical Address	Transaction Identifier (MS)	Transaction Identifier (LS)	Extended Write Address
Write Address (MS)	Write Address	Write Address	Write Address (LS)
Data Length (MS)	Data Length	Data Length (LS)	Header CRC
Data	Data	Data	Data
Data	Data	Data	Data
Data	Data CRC	EOP	<i>Last byte transmitted</i>

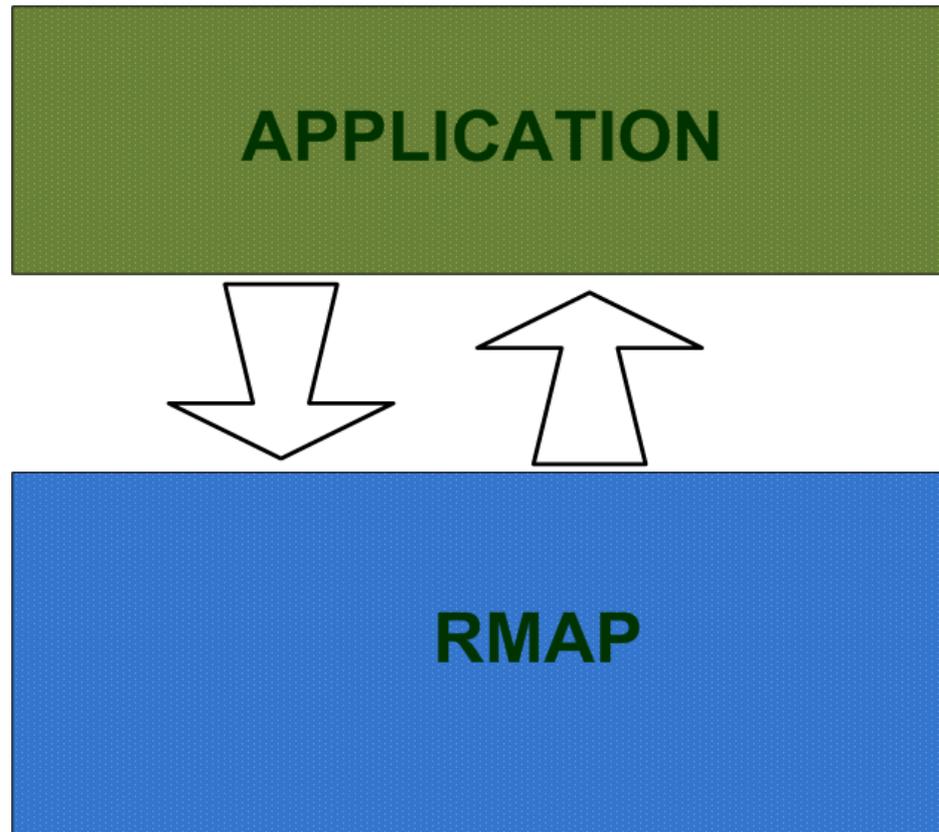
Bits in Packet Type / Command / Source Path Address Length Byte



**Figure 6-1 Write Command Format (Logical Addressing)**

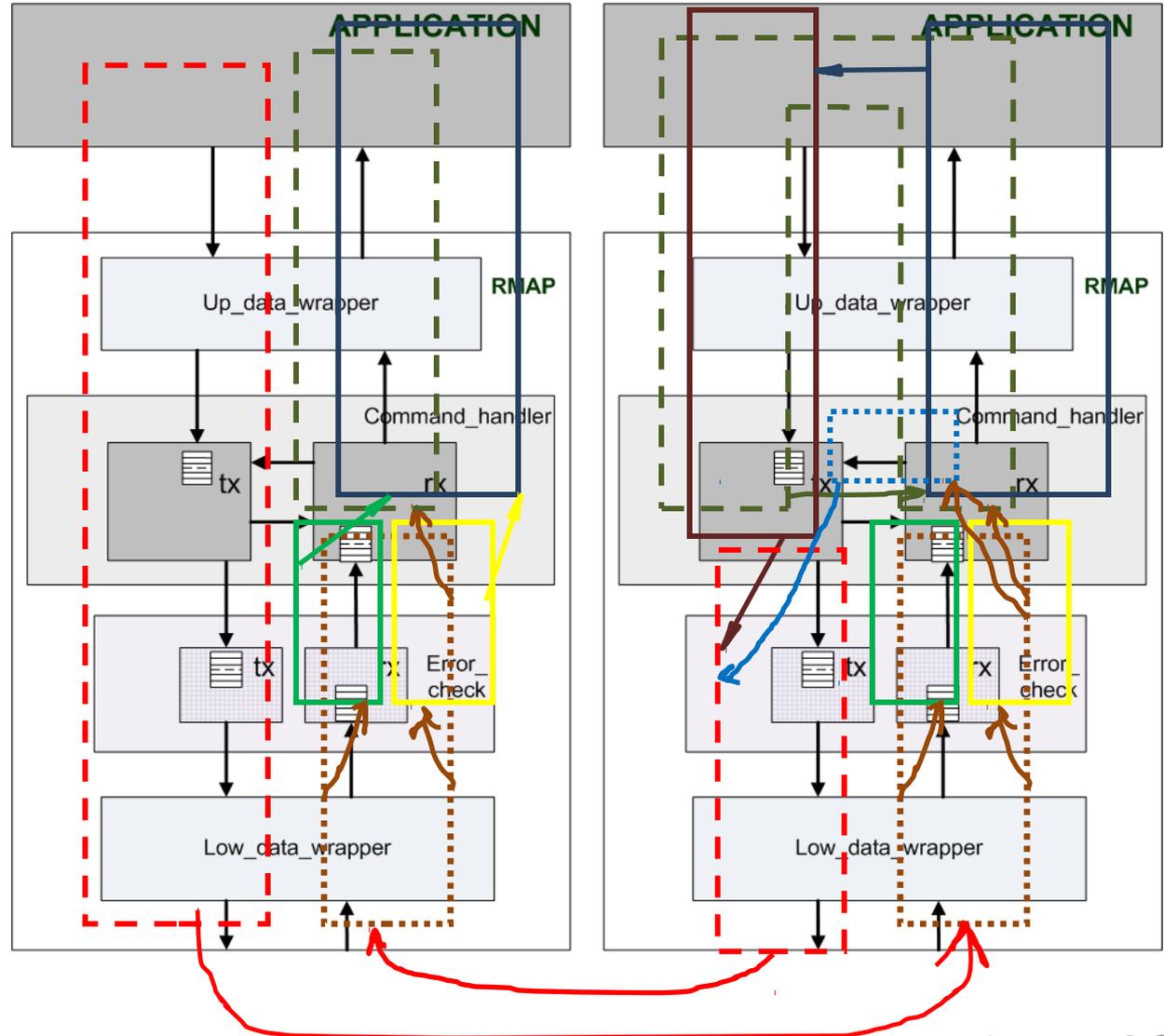
# RMAP modelling features

- Variety of functions provided by RMAP interface to upper layer



# RMAP modelling features

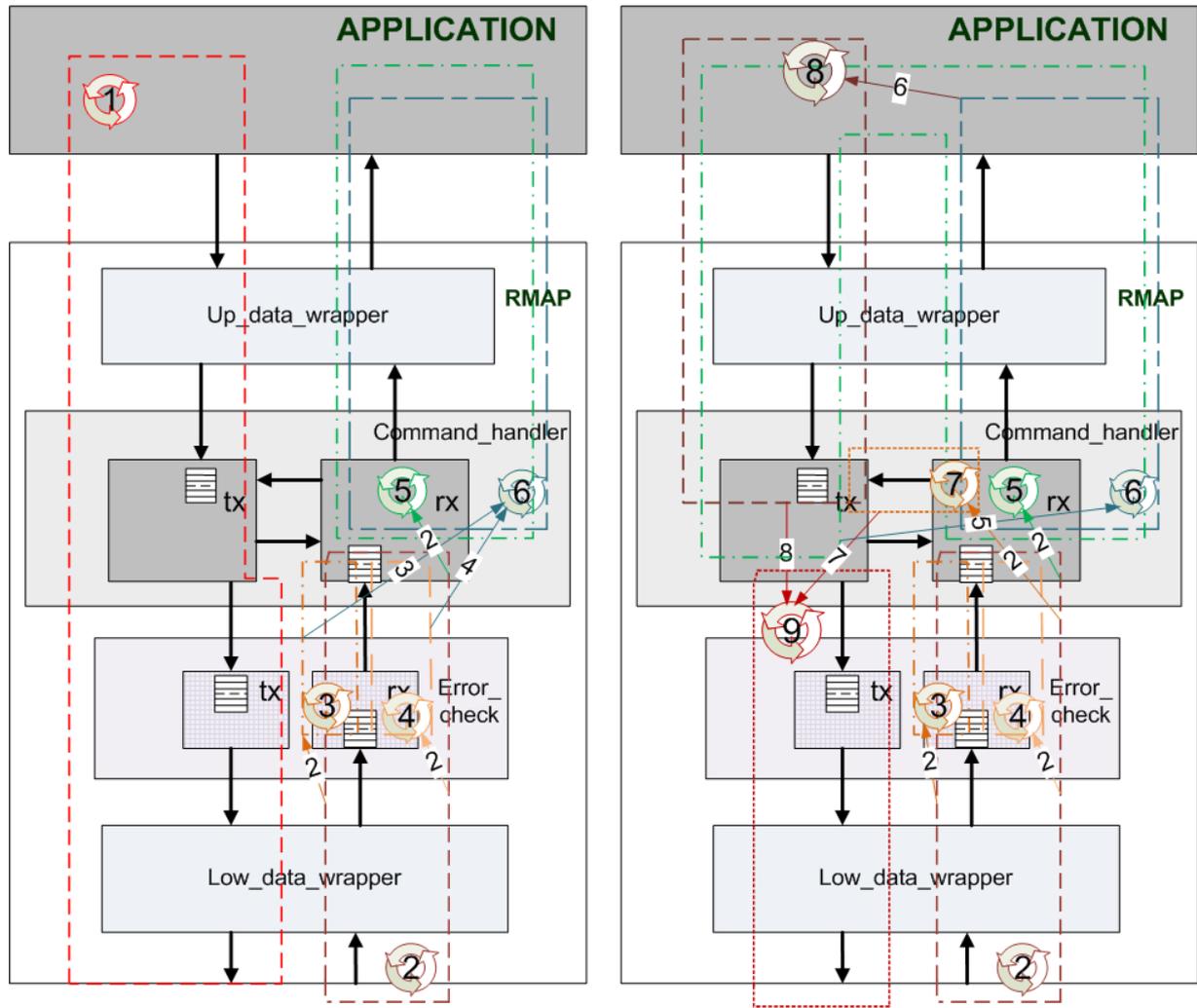
- Modelling and testing was based on iterative approach to development



# Conclusion

- The ambiguous places and inconsistencies were found
- Also bugs in specifications were found
- RMAP & SpaceWire models were combined and tested
- An essential point in checking correctness of errors processing is generation errors in channel with certain probability
- the usability of the iterative approach was affirmed

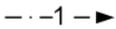
Thank you!



Legend:

 - thread

 - thread's borders

 - invoke's line with number of a thread

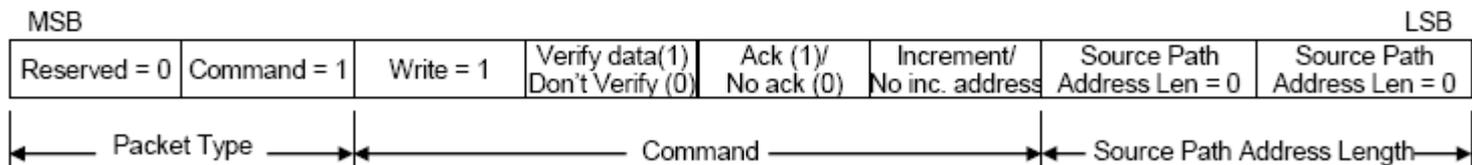
 - thread 1 invoke event of thread 2

# RMAP write command

*First byte transmitted*

Destination Logical Address	Protocol Identifier	Packet Type, Command, Source Path Addr Len	Destination Key
Source Logical Address	Transaction Identifier (MS)	Transaction Identifier (LS)	Extended Write Address
Write Address (MS)	Write Address	Write Address	Write Address (LS)
Data Length (MS)	Data Length	Data Length (LS)	Header CRC
Data	Data	Data	Data
Data	Data	Data	Data
Data	Data CRC	EOP	<i>Last byte transmitted</i>

Bits in Packet Type / Command / Source Path Address Length Byte



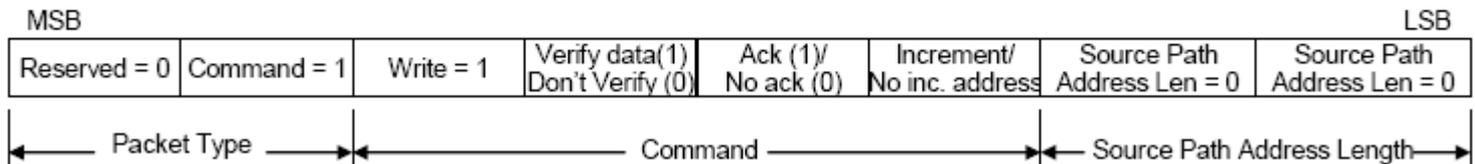
**Figure 6-1 Write Command Format (Logical Addressing)**

# Specification Bug

*First byte transmitted*

Destination Logical Address	Protocol Identifier	Packet Type, Command, Source Path Addr Len	Destination Key
Source Logical Address	Transaction Identifier (MS)	Transaction Identifier (LS)	Extended Write Address
Write Address (MS)	Write Address	Write Address	Write Address (LS)
Data Length (MS)	Data Length	Data Length (LS)	Header CRC
Data	Data	Data	Data
Data	Data	Data	Data
Data	Data CRC	EOP	<i>Last byte transmitted</i>

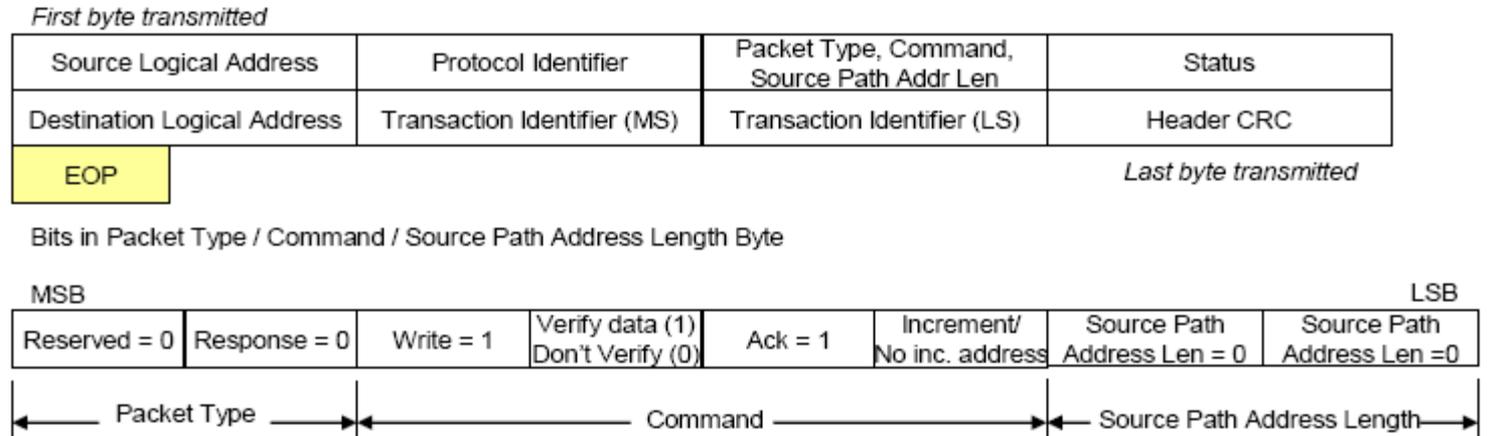
Bits in Packet Type / Command / Source Path Address Length Byte



**Figure 6-1 Write Command Format (Logical Addressing)**

The Extended Write Address byte holds the most-significant 8-bits of the memory address to be written to. This extends the 32-bit memory address to 40-bits allowing access to 1 Terabyte of memory space in each node. The Extended Write Address may be used to identify different banks of memory or registers to be written to, **to specify a target application for the data**, or to reference a specific mail box.

# Specification Bug

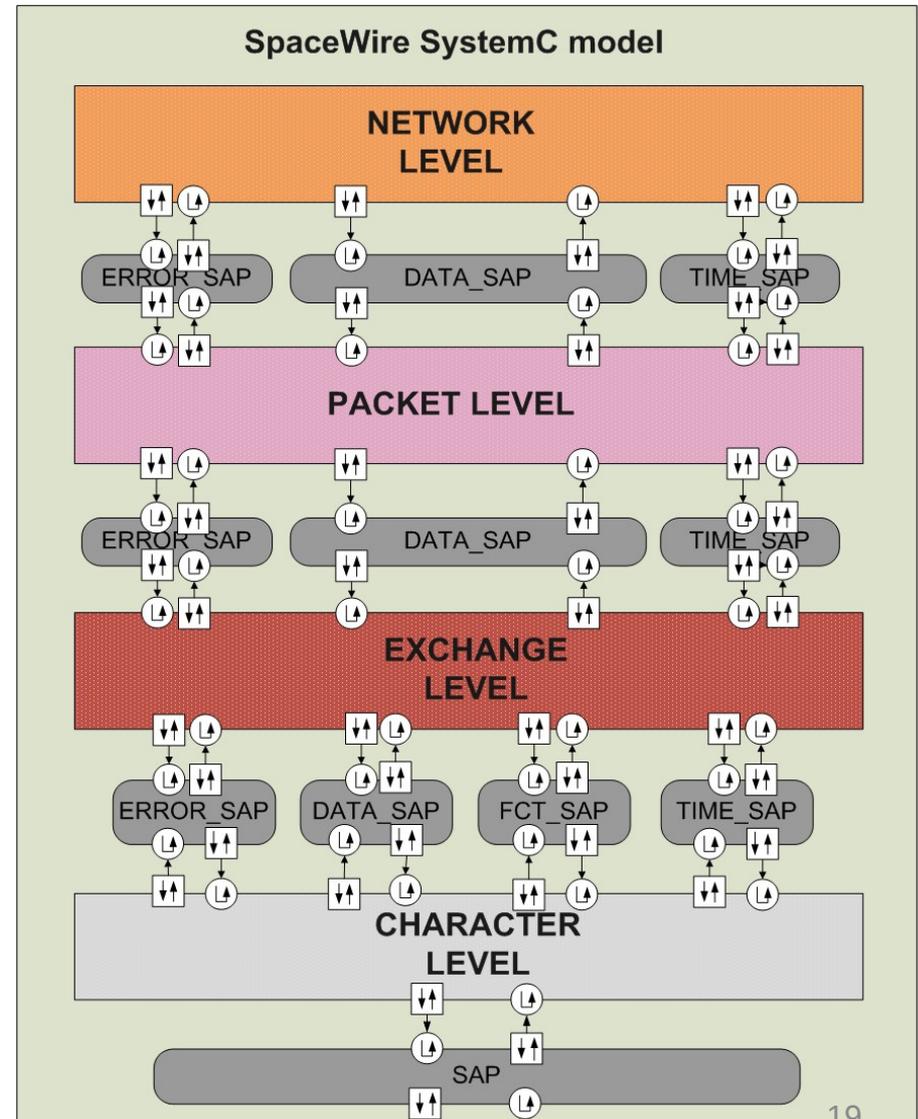


**Figure 6-2 Write Reply Format (Logical Addressing)**

# SpaceWire model overview

## Model features:

- Without signal and physical layer
- Service access points (SAPs)
- The modelling per layer method is used

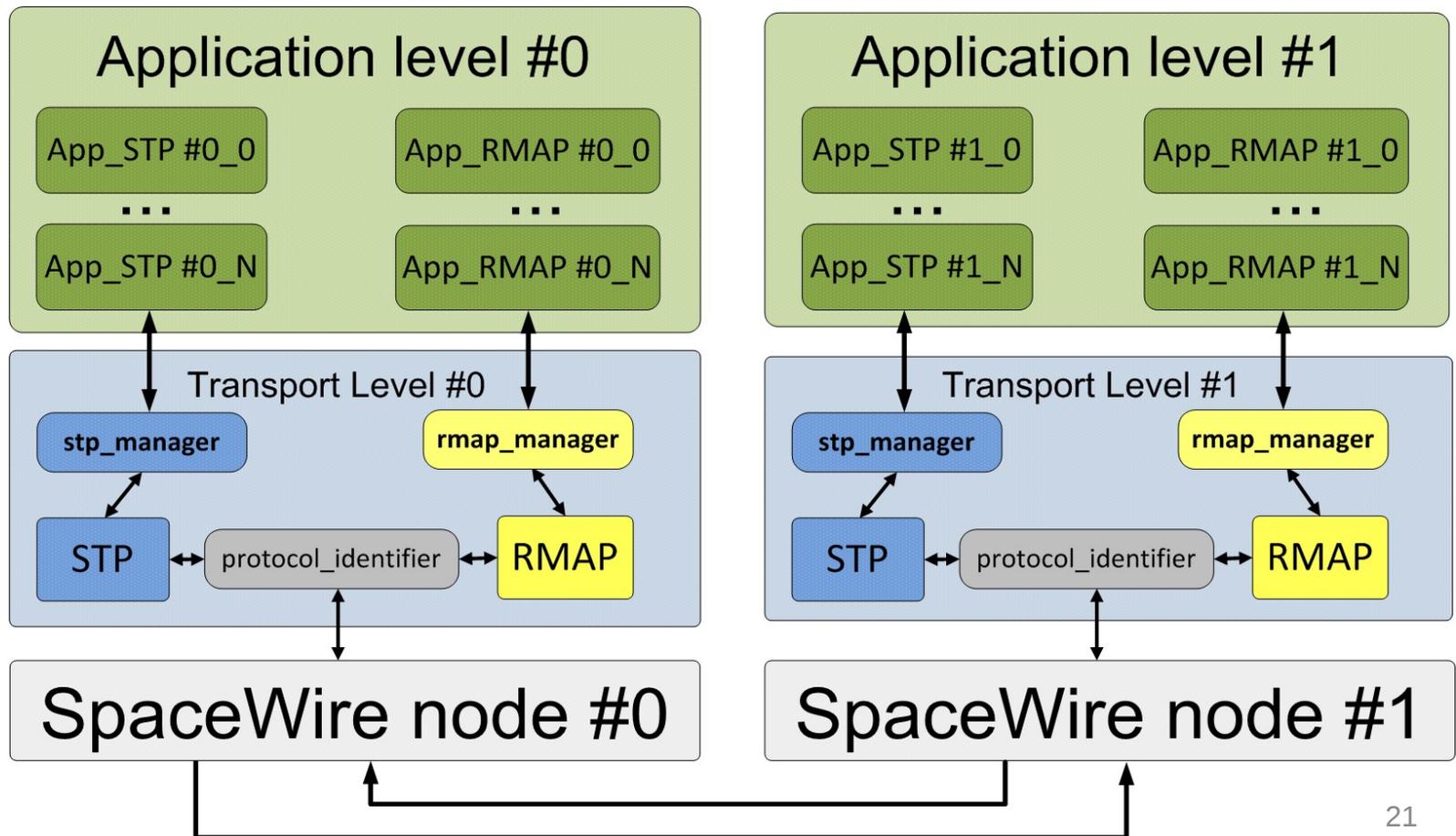


# Project overview

Total SystemC model architectural diagram:

# Testing method

Diagram of point-to-point connection for models testing:



# Testing method

## Number of benefits provided by this method:

- The model could test all the internal mechanisms
- There is a possibility to trace the whole data exchange process in both directions
- The traffic analysis are done by the model itself according to the specification
- There is a possibility to check the joint work of protocols

# Application level simulation

During the complex models step-by-step development it is necessary to have a testing system.

## Model's application layer functionality:

- generation of different kinds of traffic;
- received data processing;
- response traffic generation;
- error and critical situations handling.

Application level has two modules App\_RMAP and App\_STP. They perform an application role and they can simultaneously work with STP and RMAP transport protocols.