D3.1 SpaceWire-RT Simulation and Validation Plan

Lead Beneficiary:	SUAI		
Author(s):	Yuriy Sheynin, Elena Suvorova, Valentin Olenev, Irina Lavrovskaya		
Work Package:	WP3	**Task:**	Task 3.1
Version:	2.00	**Last modified:**	26/09/2012
Status:	Released		
Approved by:	Steve Parkes	**Date:**	03/10/2012

Table of contents

1 Introduction

This document gives the SpaceWire-RT simulation and validation plan, describing the core features of SpaceWire-RT that should be simulated, test plan and main simulation and validation principles which are to be used during the WP3. This plan is prepared taking into account the following documents:

- Annex I - "Description of Work" to the SPACEWIRE-RT project, Grant Agreement # 263148 [1].
- D2.1 SpaceWire-RT Outline Specification, version 2.0 [2].
- SpaceFibre Standard Draft D v1.14 [3].
- D 1.1 Consolidated set of Requirements for SpaceWire-RT [4].

The WP3 is considered to have the following main objectives and targets:

1. Development of the formalised SpaceWire-RT specification as the layered SDL model that follows the SpaceWire-RT layered protocol stack architecture with specified Service Access Points (SAPs) between the layers and perform validation, which should cover consistency of the specification and completeness of test features. All these steps are specified is details in Task 2 and Task 3 of WP3.
2. Development of the reconfigurable SpaceWire-RT networks model in the SystemC language. The model should be investigated proving the network level target features and characteristics of the novel SpaceWire-RT technology. All these steps are specified is details in Task 4, Task 5 of WP3.
3. Analysis of the SpaceWire-RT SDL model validation and the SpaceWire-RT networks simulation, identification of inconsistencies of incompleteness in the specification, bottlenecks in the specified mechanisms at the network level. This should result in updates and improvements to the SpaceWire-RT Outline Specification. This point corresponds to Task 6 of WP3.

The WP3 covers two general approaches for the stacks of protocols modeling: modeling per layer and modeling of devices. Each of the approaches provides different opportunities for validation and simulation. The following two subclauses define their features and opportunities in details.

1.1 SDL specification and modeling

The SDL (Specification and Description Language) [5] is a language for unambiguous specification and description of the telecommunication systems behavior. The SDL model covers the following five main aspects: structure, communication, behavior, data, and inheritance. SDL language is intended for description of structure and operation of the distributed real-time systems. Writing an SDL model on the basis of the specification is itself a test of the specification for completeness and unambiguousness. As a result the consistent readable textual description and formalised specification in SDL are produced.

SDL language is the most reasonable solution for modeling and validation on per layer basis. The objective of development of the formalised specification of SpaceWire-RT assumes that the target model should describe all mechanisms, interactions and functionality which are stated in the SpaceWire-RT specification.

Each layer of the SpaceWire-RT protocol stack should be implemented in SDL, tested and validated separately. This will prove that the layer provides defined in the specification services. Moreover, the SDL model will check the consistency of the specification. In turn, the SDL model of the whole SpaceWire-RT protocol stack can be used for checking, how all mechanisms operate in common in one node. This way, SDL layered SpaceWire-RT model can be used for validation of consistency of the specification and checking of functional requirements, defined for the SpaceWire-RT.

By use of the SDL language it is possible to develop an asynchronous model only. That means that the SDL model cannot be used for checking and proving any performance requirements.

1.2 SystemC modeling and network simulation

The SystemC [6] modeling is one of the most efficient and widely used methods for studying, analysis and constructing multi-component systems, such as stacks of protocols, embedded networks of a large number of nodes, systems-on-chip, networks-on-chip, etc [7].

SystemC is a set of C++ classes and macros that provide an event-driven simulation engine. It is specifically designed for modeling parallel systems. This library allows describing multi-component systems and program components, and modeling their operation. By using the internal mechanism of events it allows to model operations distributed in time of the modeled system.

The SystemC part of WP3 is focused on modeling devices and their operation in a network. This means that interactions of components and processes inside the device (e.g. between levels of a stack) should not be considered. The model is primarily focused on the mechanism of devices' communications, such as transfer of packets, routing, performance characteristics of the network.

According to the WP3, SystemC is used to model SpaceWire-RT networks. The model should enable the building and configuration of SpaceWire-RT networks with multiple nodes and routing switches. Such models will provide the basis for investigating and proving the network level features and characteristics of the novel SpaceWire-RT technology: scalable performance, responsiveness, robustness, provision of quality of service, and ultimate low latency signaling [1].

SystemC modeling of the SpaceWire-RT will be performed over different network topologies which will help to investigate different properties of possible mechanisms for arbitration, buffering, planning/scheduling, packet forwarding.

Section 2 of this document provides SpaceWire-RT SDL validation and investigation plan, including tasks definitions and examples of schemes of target models. It also provides general simulation principles which are to be used while simulation and testing of the developed models. Section 3 provides SpaceWire-RT SystemC validation and investigation plan. It gives main simulation principles, internal structure of a SpaceWire-RT SystemC node and a switch and describes possible solutions for testing network operation on different topologies. Section 4 provides the testing plan, which considers the requirements to the SpaceWire-RT technology and gives description of tests to be performed in order to prove or disprove requirements. This section gives non-nominal cases which should be verifies during modeling in SDL and SystemC also. And finally, section 5 gives a conclusion to the document.

2 SDL Validation and Investigation Plan

2.1 SDL specification and investigation principles

2.1.1 Development of the formalized SpaceWire-RT specification

WP3, Task 2: Development of the formalized SpaceWire-RT specification as the layered SDL model that follows the layered SpaceWire-RT protocol stack architecture, with specified Service Access Points (SAP) between the layers. The structure of the model is given in Figure 1.

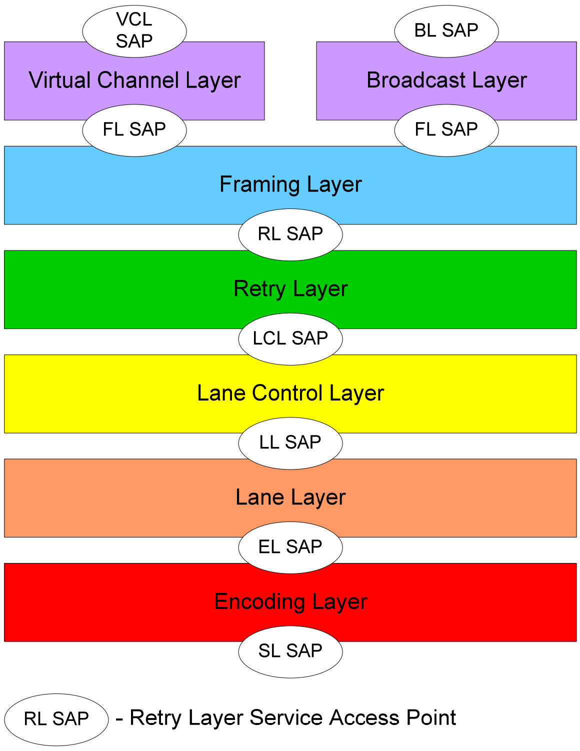

Figure 1. SpaceWire-RT Protocol Stack in SDL

Note: the Serialization Layer is out of scope of SDL specification due to the significant difference between the functionality of the layer and the features of SDL. However, if it is necessary, some simplified model of the Serialization Layer can be implemented in SDL. Another layer, which is out of scope of SDL specification, is Network Layer. The main reason for this is that SDL part of WP3 is primarily focused on point-to-point interconnection of two SpaceWire-RT nodes, which does not require Network Layer. Moreover, the Network Layer description, which is given in the SpaceWire-RT Outline Specification [2], is not sufficient for development of SDL specification and it cannot be applied to formal specification.

2.1.2 Verification of each layer in IBM Rational SDL Suite

WP3, Task 3: Verification of each layer of the SpaceWire-RT protocol stack by simulation in IBM Rational SDL Suite. According to this task, each layer defined in Figure 1 will be simulated in IBM Rational SDL Suite. This simulation gives an ability to check internal mechanisms of investigated layer and verify it.

2.1.3 Verification of each layer in SDL/SystemC tester

WP3, Task 3: Verification of each layer of the SpaceWire-RT protocol stack by load testing by means of simulation within SDL/SystemC tester. An example of SDL/SystemC tester scheme for the Retry Layer is shown in Figure 2.

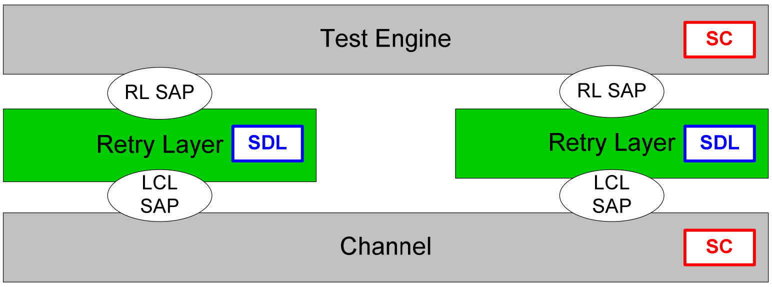

Figure 2. SDL/SystemC tester for the Retry Layer

2.1.4 Validation of the SpaceWire-RT SDL mode

WP3, Task 3: Validation of the SpaceWire-RT SDL model. The model shall be simulated within SDL/SystemC tester. In case of necessity between some pairs of adjacent layers an inter-layer can be inserted in order to trace or modify the data flow between the layers. The tested model shall consist of two SpaceWire-RT nodes, which are connected by a channel implemented in SystemC. This channel shall be capable to corrupt transmitted data in a given way. The following types of error will be used during the validation:

- bit inversion – the value of a chosen bit shall be inversed (0 → 1, 1 → 0);
- bit insertion – after a chosen bit one more bit with the same value shall be inserted;
- bit deletion – a chosen bit shall be omitted.

The scheme of the tester is shown in Figure 3.

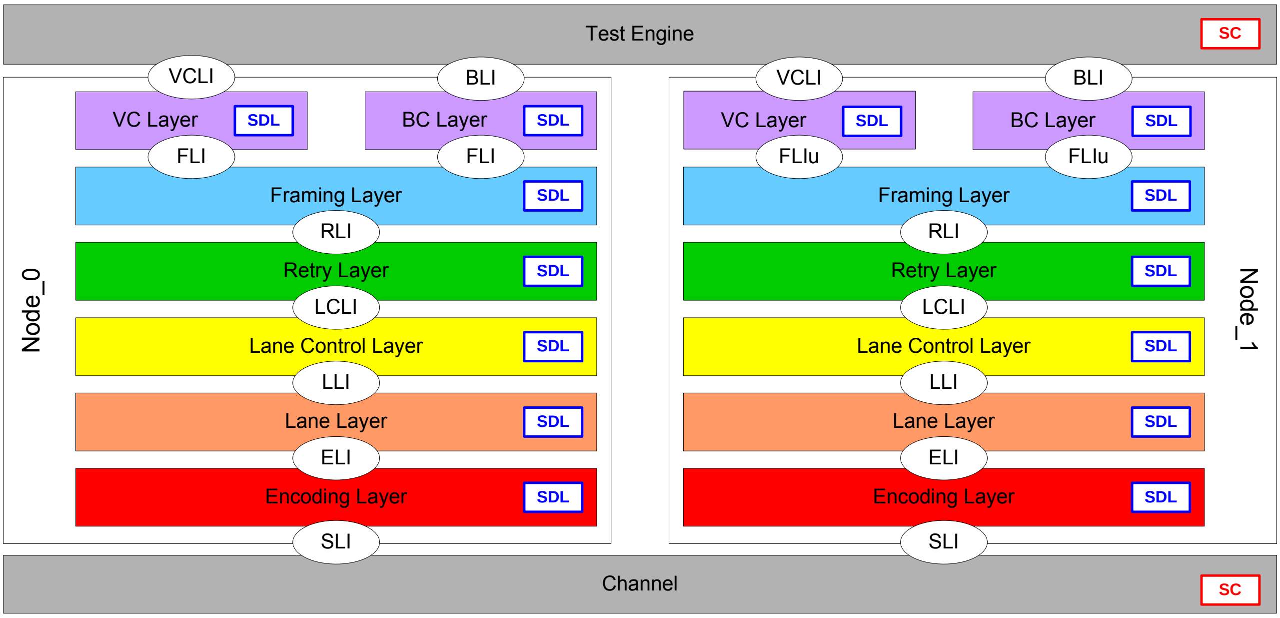

Figure 3. SDL/SystemC tester for the SpaceWire-RT protocol stack model

Protocol functions, which should be the objects of particular interest while the specification and investigation of SpaceWire-RT, are given in the table below:

Function	Layer	Description
Medium Access Control algorithms	Virtual Channel Layer	The following features should be investigated: • Bandwidth credit counter • Priority-based arbitration • Scheduled arbitration
Retry mechanism	Retry Layer	The mechanism of re-sending corrupted data should be tested under different conditions (corrupted frame, loss of ACK or NACK, etc.)
Lane Initialization State Machine	Lane Layer	A lane initialization sequence as well as Standby state should be carefully investigated
Receive Synchronization State Machine	Encoding Layer	The cases of symbol synchronization and the loss of word synchronization should be simulated under different types of errors

2.2 Explanation of SDL/SystemC tester principles

In order to implement interconnection between the SpaceWire-RT SDL model and the test environment an SDL/SystemC co-modeling approach should be used [8]. This approach assumes that a wrapper will be implemented for conversion of data from SDL representation to SystemC representation and vice-versa. Figure 4 presents more detailed scheme of the retry layer tester which includes SDL/SystemC wrapper.

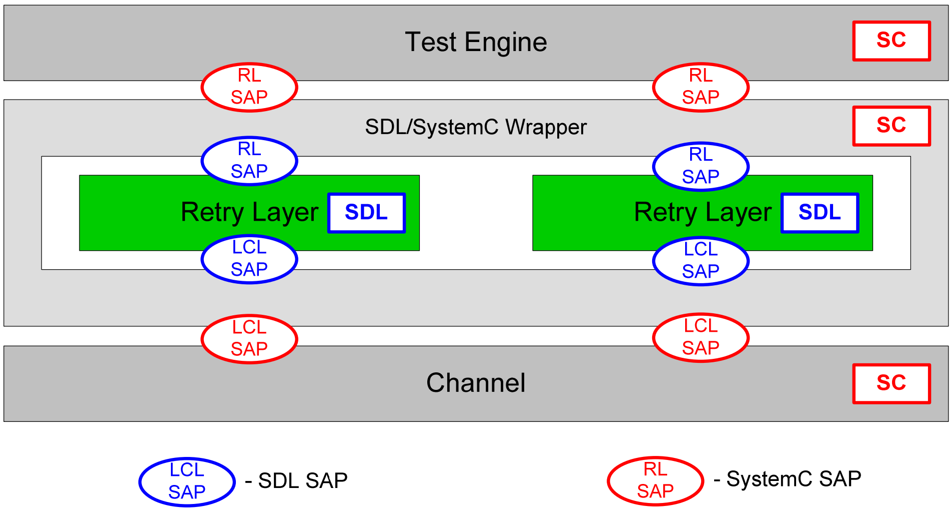

Figure 4. SDL/SystemC Wrapper

The deployment of the SDL/SystemC tester for the SpaceWire-RT SDL model verification and investigation leads to the following benefits in terms of simulation and modeling:

- The increase in the flexibility of the test environment modules. This in turn provides an opportunity to implement sophisticated test sequences to analyze low-possible test cases.
- A possibility to implement load testing. The test length can reach millions of packets in order to test the protocol under the required bit error rate (10e-15).
- The decrease in the testing time due to the pure C/C++ implementation of the tester.

3 SystemC Validation and Investigation Plan

3.1 SystemC network simulation principal

In this task we plan to use the “modeling of devices” simulation principal. That means that the model will consist of the number of modules - Nodes, Switches and Channels models. The internal logic of each device doesn’t need to be fully simulated and the layered structure of the protocol inside the Node or Switch module is not necessary to implement. The functions of main interest are those functions, which are in order for the network communication. So a number of features of the protocol could be implemented without the full conformity to the specification structure. This means that there is no need to implement Layers, SAPs and some handshakes between them because internally a Node or a Switch model should be as simple as possible, but it should operate correctly, e.g. generate data and process it as specification says. Finally, the network will be made and configured using Node, Switch and Channel modules and each module will be used as a black box with a number of settable parameters.

Examples of network models with such a method are shown in Figure 5:

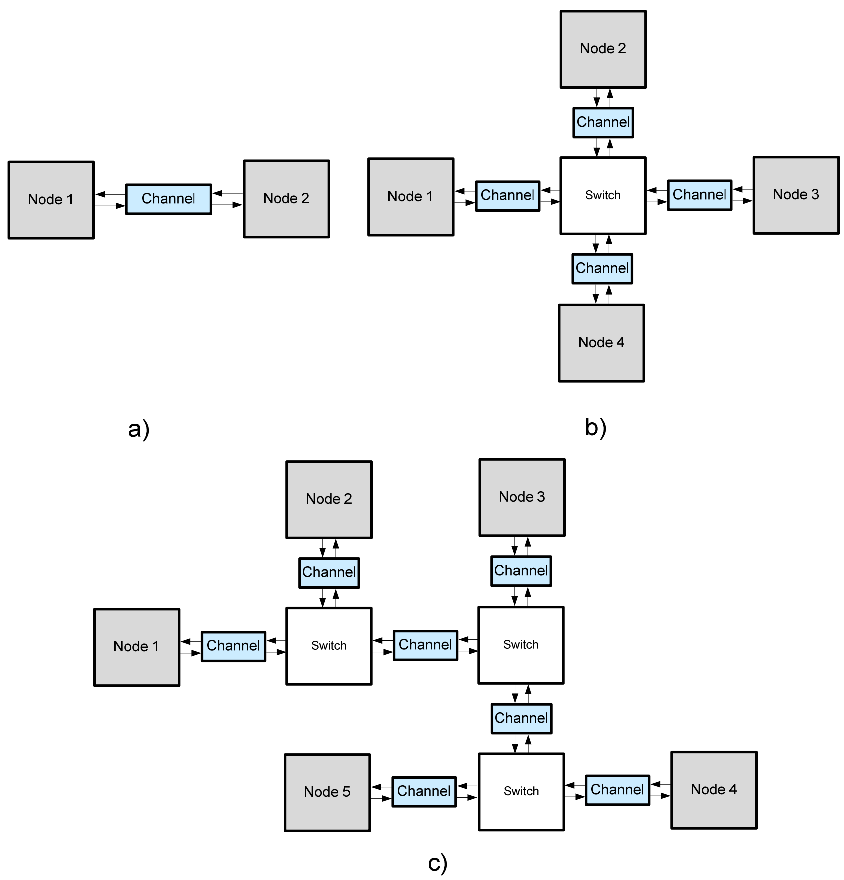

Figure 5. Examples of the “modeling of devices” method

3.2 Internal structure of the SpaceWire-RT SystemC Node

As it was mentioned above, the internal structure of the Node should not follow the layered structure of the protocol, but it should have the whole functionality needed for the Network communication.

Each Node will consist of a SpaceWire-RT Module and TestGenerator on top of it (Figure 6). SpaceWire-RT Module is a model of a SpaceWire-RT controller which takes SpaceWire packets from the TestGenerator and pushes encoded 10-bit symbols to the Channel. TestGenerator is a model of a host Application (Application Layer) which generates SpaceWire packets for the transmission through the network.

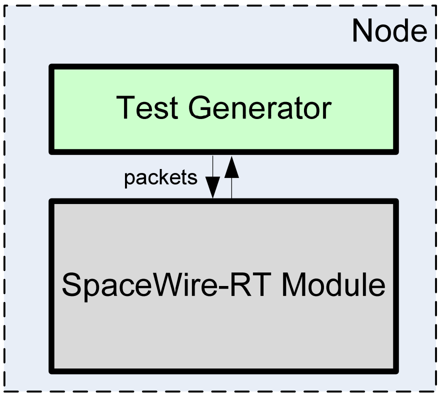

Figure 6. Node Structure

The internal structure of the SystemC Module is planned to look like it is shown in Figure 7.

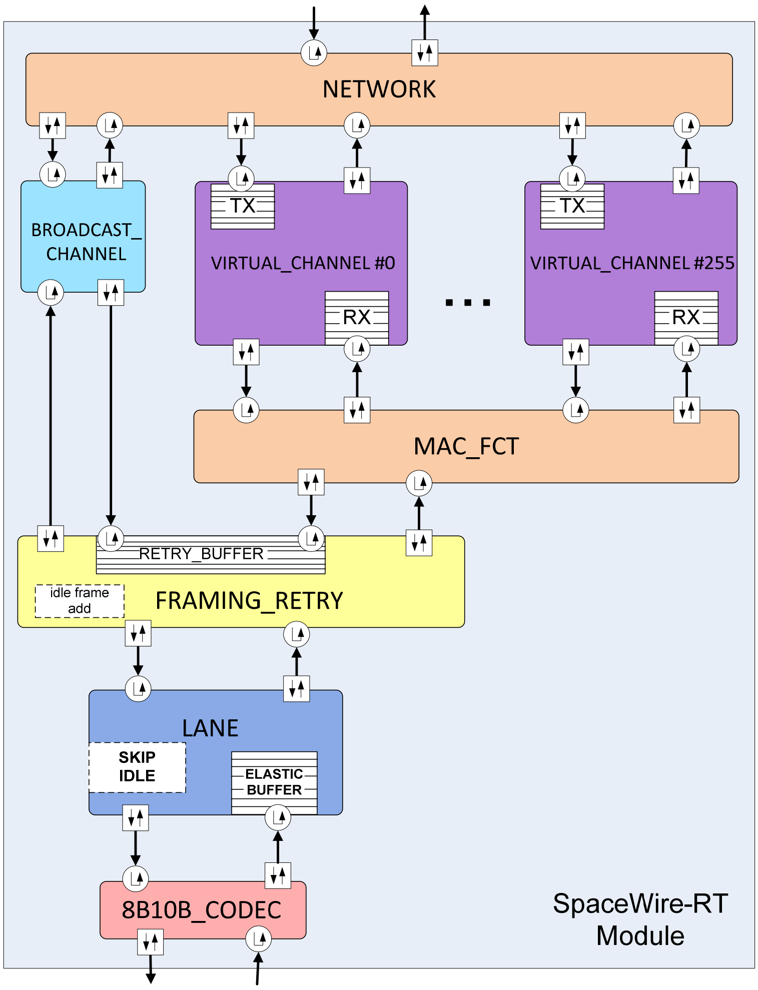

Figure 7. Architectural diagram of the SpaceWire-RT Module

The SpaceWire-RT Switch structure is different to a Node structure, because Switch structure ends at the Network layer, which should be able to operate with a number of ports, and the Network Layer itself would be much more complex. The internal structure of the SystemC Switch is planned to look like it is shown in Figure 8.

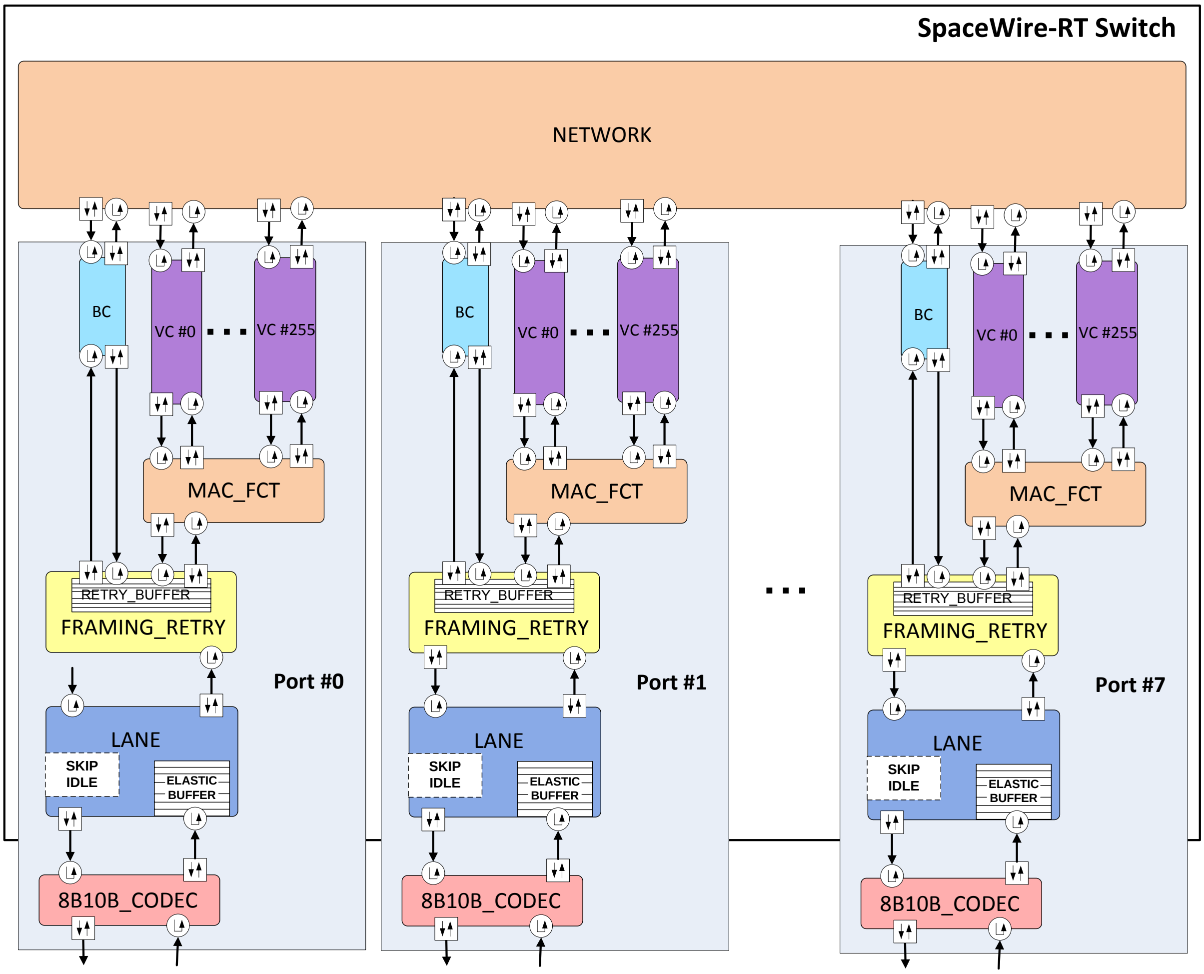

Figure 8. Architectural diagram of the SpaceWire-RT 8-port Switch

Each Test Generator, Node, Channel and Switch will log all the incoming and outgoing data. It will help to analyse the results of the simulation. Every log-file will contain text information about data transmission, certain moment of time of the particular operation, length and type of data, etc.

3.3 Network tests

To test basic mechanisms of the SpaceWire-RT on a network three different network configurations are planned to be used:

1. Point-to-point configuration
2. Tree configuration
3. Circular configuration

The first step in testing would be simulation of a point-to-point configuration of a network. This will additionally test the internal mechanisms of the node and basic communication mechanisms.

The next step would be the including of the Switch into a network structure (a tree configuration). Doing it we will additionally test a switch, then we will test the routing and switching mechanisms, circulation of data between the nodes and all QoS mechanisms defined in SpaceFibre specification [3].

The third step is modeling of a circular network configuration, which has a “cycle” in the network structure. This will validate that a “cycle” would not cause any problems for the SpaceWire-RT.

These three configurations will give an ability to test all the required parameters and internal mechanisms of the protocol.

The last step would be testing of the different combinations of the 2 and 3 configurations. More research could be done with the developed network framework model studying practical network configurations, which are closely related to some real configurations of the onboard networks. It will give an ability to see the models of real onboard networks, test and research them. But in this case each particular model should be developed more properly, with additional blocks and test mechanisms that are out of scope of current modeling work.

3.3.1 Nodes and Switches modules

Before the start of the network simulation we have to be sure that the Node and Switch modules are working in fully respect to the specification: correctly get and send the data, correctly process it, process errors. For this purposes each Node and Switch will be tested independently using different Test Generators as it is shown in Figures 9 and 10.

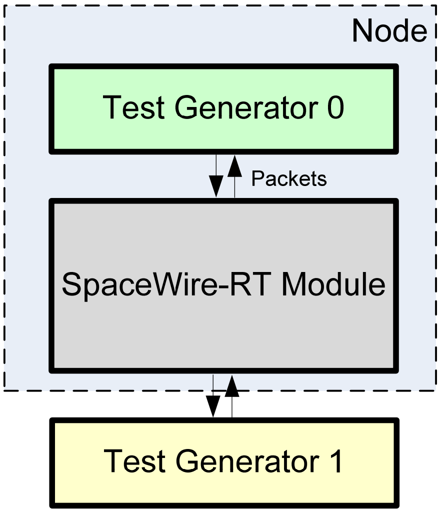

Figure 9. Testing of the SpaceWire-RT Node

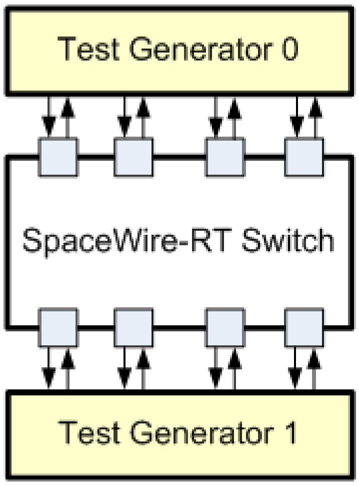

Figure 10. Testing of the SpaceWire-RT 8-port Switch

Such way of testing gives an ability to test:

- internal mechanisms of the Node that forms data traffic and process it;
- basic testing of the routing table workability;
- the serial loop-back mechanism;

To be sure, that modules operate in full compliance with the SpaceWire-RT specification, a number of tests with different kinds of traffic generation is planned to be performed. This just will prove that Node and Switch modules operate as expected and each unit of the network will have no errors in the internal logic. The difference with SDL is that using SDL we will test the validity of these mechanisms, and on SystemC we just ensure that modules work according to the specification.

3.3.2 Point-to-point configuration testing

This is the simplest way to test the communication features of SpaceWire-RT Nodes. It will help to test all the specified mechanisms of the protocol (excluding the broadcasting and network mechanisms), such as fault tolerance, quality of service, retransmission mechanisms, etc. Point-to-point configurations will be used for checking correctness and consistency of SpaceWire-RT technology. It is the same work that is planned to be done on SDL; it will double-check the correctness of the SpaceWire-RT mechanisms and check the simplest network communication, which should be the first step in the SystemC network model development.

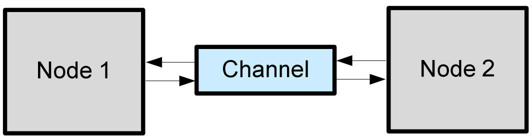

Figure 11. Point-to-point network configuration

3.3.3 Tree configuration

In this configuration we use one switch, then a number of switches, to test the network communication. In this test we start to use the network layer of the SpaceWire-RT specification, [2, section 12]. For example, this test gives an ability to validate (in addition to previous ones) broadcasting, network addressing, routing and switching, etc.

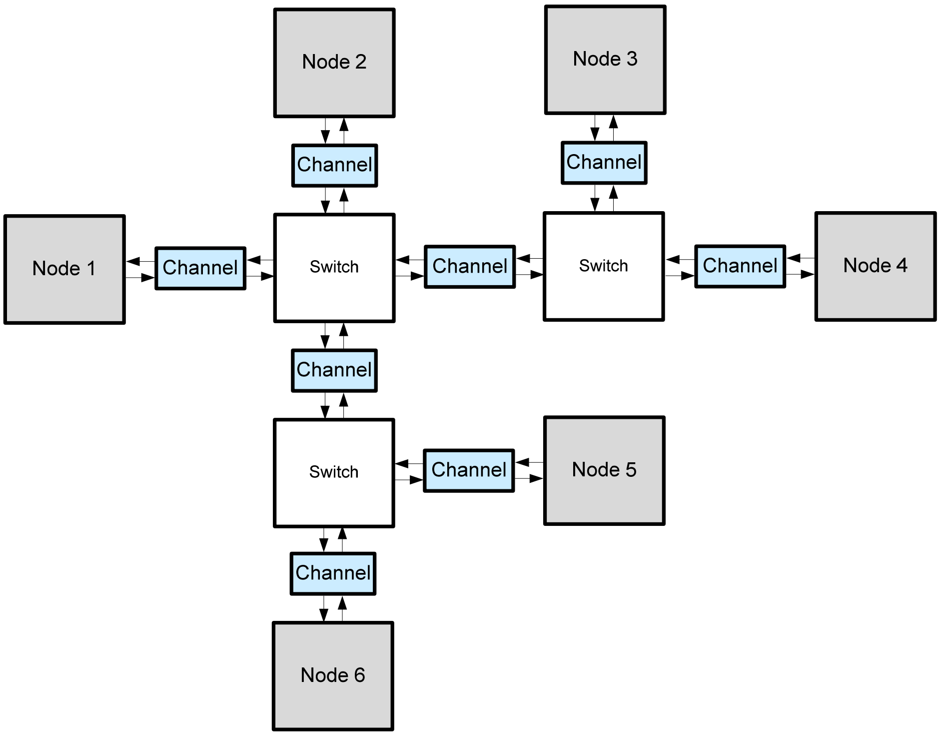

Figure 12. Tree network configuration

3.3.4 Circular configuration

This test is different to the previous because it has a “cycle”, it means that packet can be passed to the destination over the network using different ways. This test will show that such kind of a situation is correctly processed in the SpaceWire-RT specification.

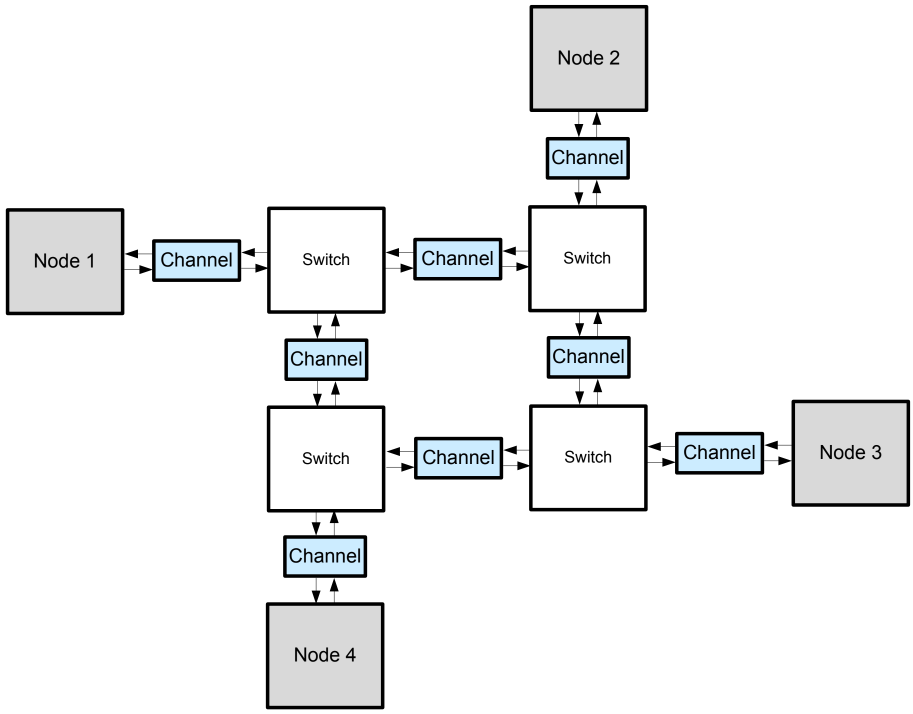

Figure 13. Circular network configuration

3.3.5 Mixed configurations

Mixed configurations are different combinations of the 2nd and 3rd configurations. During this step firstly we plan to test a SpaceWire-RT network, which consists of a bigger number of Switches and Nodes (as an example – Figure 14).

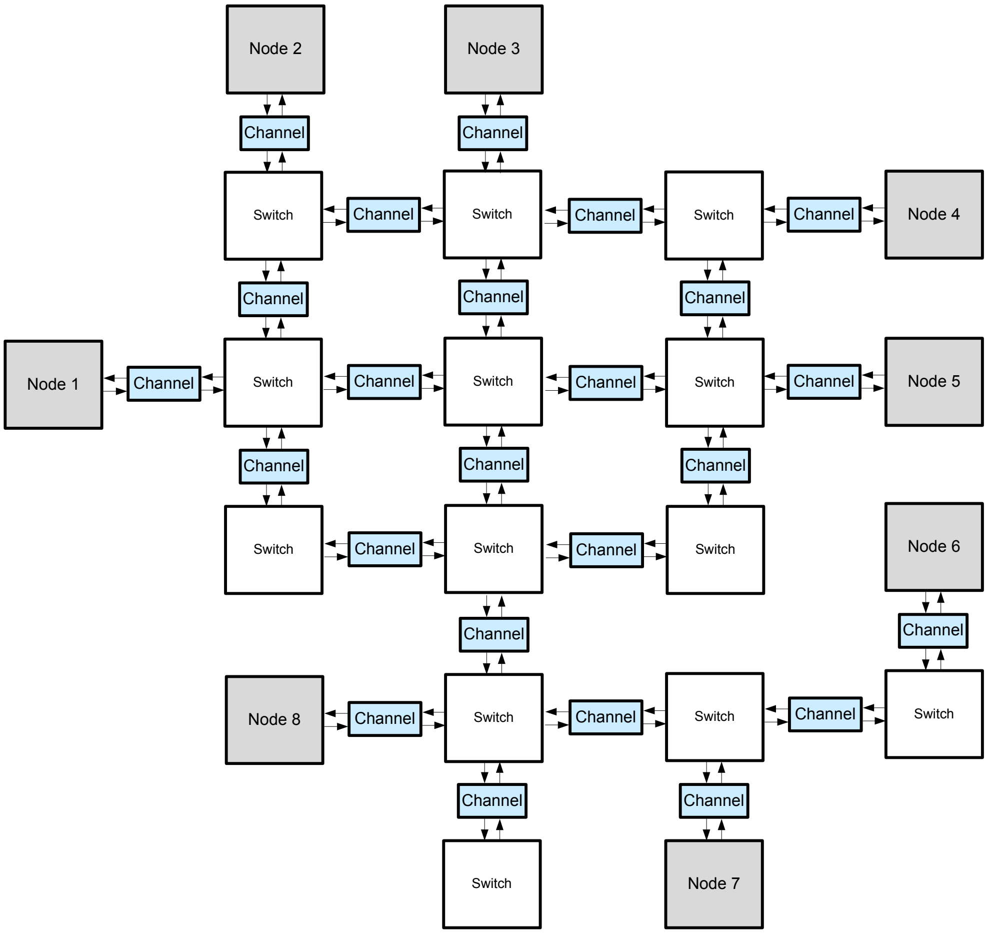

Figure 14. Mixed configuration

3.3.6 Real onboard network structures

Additionally, it is possible to test SpaceWire networks that are closely related to the real configurations of the onboard networks (as an example – Figure 15, Figure 16). As examples of real onboard network structures some reference data-handling networks could be taken, e.g. of the Mercury Polar Orbiter (see Figure 15) or ASNARO SpaceWire Networks (see Figure 16) [9].

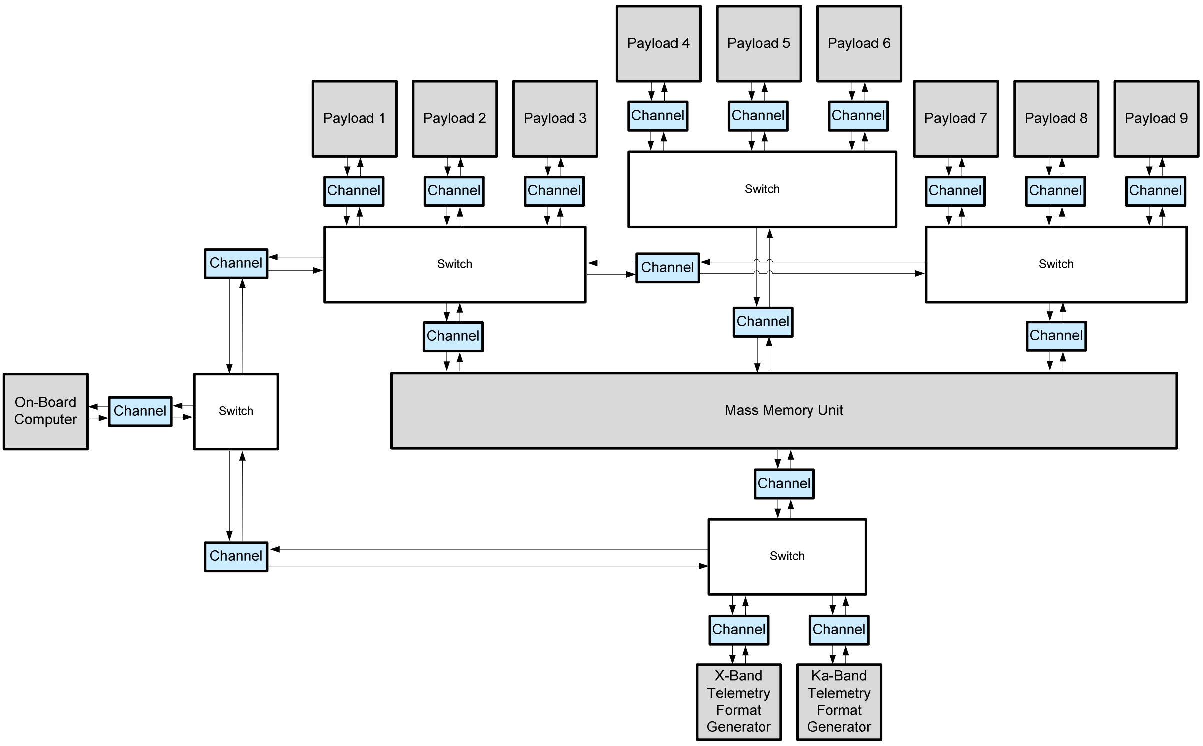

Figure 15. Mercury Polar Orbiter data-handling architecture

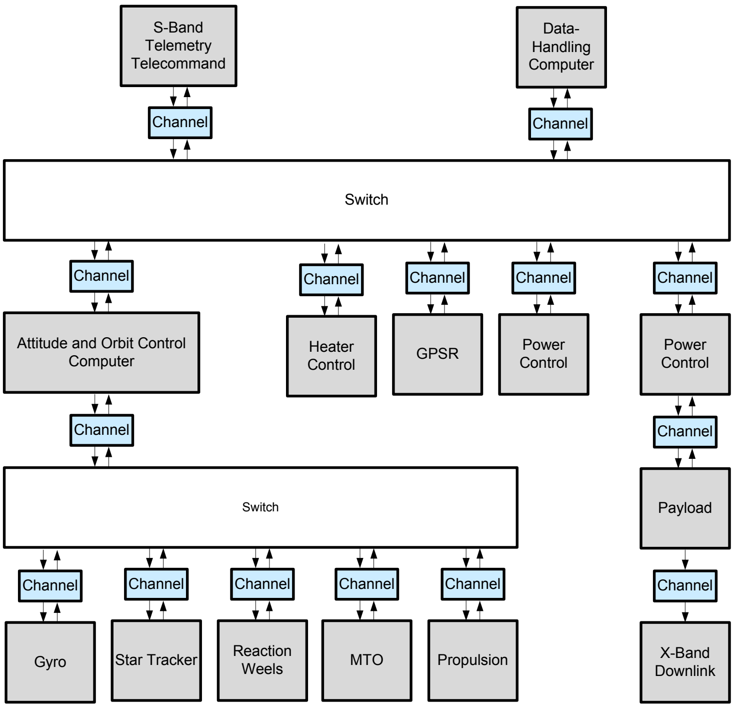

Figure 16. ASNARO SpaceWire Networks

Such kind of testing of realistic onboard systems models causes additional problems and leads to additional work. Firstly, we'll need to study the main principles of system operation and the purpose of each node in the Network. Depending on the purposes of the nodes, we will need to develop the additional modules of the model such as Mass Memory Units or some other Nodes with multiple incoming and outgoing ports. Additionally, for the testing purposes special configuration of each Node and generation of the specific traffic sequences possibilities should be developed; only the strictly planned queues of the packets can cause a correct operation of the system and correct results.

So such kind of testing needs additional work effort which is not planned for the current contract. It could be done during the further work on SpaceWire-RT standard and can give important research results.

4 Test Plan

First of all the developed SDL and SystemC models should be used for checking correctness and consistency of SpaceWire-RT specification. After this step is performed, the developed SDL and SystemC models should be used for checking the SpaceWire-RT technology against the requirements evaluated in WP3 and summarized in Section 6 of the D1.1 document [4]. And finally, the SpaceWire-RT standard has still some problems and issues that are points of special interest and are to be tested carefully.

This section starts with the description of how SpaceWire-RT will be checked on correctness and consistency. Then it takes the list of the RF and EU consolidated requirements, analyses them and describes, how the particular requirement is planned to be tested and which model (SDL or SystemC) will be used for it. Additionally, we describe non-nominal cases which will be verified during simulation on SDL and SystemC.

4.1 Checking correctness and consistency of SpaceWire-RT

One of the main objectives of the WP3 is checking correctness and consistency of SpaceWire-RT technology specification. This will be done by specification and simulation on SDL point-to-point model and by simulation of SystemC network models. SDL model will be primarily focused on checking internal functionality and how all mechanisms of all layers work in common in one node. Main steps for SDL specification and simulation are given in Section 2 of this document.

SystemC network models will be primarily focused on simulation and checking of network aspects of SpaceWire-RT, performance parameters (e.g. latency) and some features that cannot be covered by SDL point-to-point simulation (e.g. broadcast distribution, QoS, etc.). Main steps and general principles for SystemC simulation are given in Section 3 of this document.

4.2 Proving requirements tests

4.2.1 Packet size

REQ-50 packet size (data handling, computer bus)

SpW-RT shall support application packet sizes up to at least 32 Mbytes.

REQ-51 packet size (control bus, telemetry)

SpW-RT shall support packet sizes in the range from 8 bytes to 64 Kbytes.

Test solution:

The packet size requirement should be checked at both SDL and SystemC models. According to the SpaceWire-RT Outline specification, SpaceWire-RT accepts SpaceWire packets to transmit over a specific virtual channel. SpaceWire packets are not limited in size. Consequently, application is able to send packets of various lengths over SpaceWire-RT network. SDL model can be used for testing and proving a possibility of transmission of 32 Mbytes packet over a single link. SystemC network model can be used for evaluation of latencies in dependence on the packet length. For example, the SystemC network model simulation can provide the results for the following set of packet lengths: 32 Mbytes, 16 Mbytes, 8 Mbytes, 512 Kbytes, 64 Kbytes, 32 Kbytes, 8 byte.

4.2.2 Maximum latency

REQ-60 Maximum latency (control bus)

SpW-RT shall support a maximum latency of less than 100 µs.

REQ-61 Maximum latency (time synchronization bus)

SpW-RT shall support a maximum latency of up to 100 ns.

REQ-62 Maximum latency (computer bus)

SpW-RT should support a maximum latency of less than 100 ns over a single link.

Test solution:

The latency requirements should be checked at the SystemC network model. During simulation we plan to evaluate latency values for broadcast messages and data packets in different network topologies and configurations. As a result, we will get an estimation of the network size which fits the latency requirements.

4.2.3 Reliability

REQ-70 reliability

SpW-RT shall provide a capability for reliable data delivery

Test solution:

The reliability requirement should be checked on both SDL and SystemC models. SpaceWire-RT standard provides automatic acknowledgements (ACK) and negative acknowledgements (NACK), which are used for indication about the validity of the received data. If the received data frame, broadcast frame or FCT is not valid due to some reasons (wrong frame sequence number, wrong CRC, etc.) a NACK shall be sent, resulting in retransmission of lost information. To check this mechanism a model of channel with errors will be used. Data, transmitted over the channel will be corrupted, causing NACK transmission. Therefore, this way the reliability of the SpaceWire-RT will be checked. For the purpose of acceleration of getting testing results we plan to increase BER for the channel (e.g. assume BER = 10^{-5}).

4.2.4 Determinism

REQ-80 determinism

SpW-RT shall provide determinism.

Test solution:

The determinism requirement should be checked on SystemC models only. SpaceWire-RT provides deterministic data delivery using a scheduling mechanism. Therefore, scheduling QoS should be tested on the network model to prove this requirement.

4.2.5 Automatic acknowledgement

REQ-100 automatic acknowledgement (control bus)

SpW-RT should support configurable automatic acknowledgement.

Test solution:

The automatic acknowledgement requirement should be checked on both SDL and SystemC models. Automatic acknowledgement is the feature of the SpaceWire-RT standard and it is defined in the Retry Layer of the SpaceFibre part of the SpaceWire-RT technology. Automatic acknowledgement mechanism primarily will be validated by the SDL model of the Retry Layer simulation as one of its main features. On reception of a data frame, broadcast frame of FCT the receiving side should send ACK or NACK in dependence on the validity of the received data.

4.2.6 Automatic fault detection and identification

REQ-110 Automatic fault detection

SpW-RT shall support automatic fault detection.

REQ-111 Automatic fault identification

SpW-RT may support automatic fault identification.

Test solution:

The automatic fault detection and identification requirement should be checked on both SDL and SystemC models. SpaceWire-RT provides automatic fault detection at different layers. During simulation at the SDL model all the defined in the specification mechanisms will be tested and checked under different error conditions, which can be caused by inserting errors into data transmitted through the channel. SystemC network model will be used for testing network layer features responsible for network fault detection. For example, automatic discard of a packet in case of mismatch of the path address and the virtual channel assignment.

4.2.7 Failure and fault tolerance of network

REQ-120 failure and fault tolerance of network (data handling network)

SpW-RT network should be able to automatically recover from faults.

Test solution:

The failure and fault tolerance of network requirement should be checked on SystemC models only. The Network Layer of the SpaceWire-RT provides mechanisms for failure and fault tolerance of network. These mechanisms will be checked by testing the following error situations:

- Deadlock
- Babbling idiot

4.2.8 Multi-path transmission

REQ-130 multi-path transmission (control bus)

SpW-RT shall support multi-path transmission.

Test solution:

The multi-path transmission requirement should be checked on SystemC models only. In order to test multi-path transmission routing switches will be configured in such a way that packets to one destination can be transmitted by different paths.

4.2.9 Broadcast data transfer

REQ-140 broadcast data transfer (time synchronisation bus)

SpW-RT shall support broadcast data transfer.

Test solution:

The broadcast data transfer requirement should be checked on SystemC models only as only a network model can be used for proving this requirement. The SpaceWire-RT defines Broadcast Layer, which is responsible for transmission of broadcast messages over a link. The SystemC network model with the following topology will be used for checking broadcast mechanisms:

- tree configuration (see p. 3.3.3);
- circular configuration (see p. 3.3.4).
- Mixed configuration (see p. 3.3.5)

4.2.10 Multi-cast data transfer

REQ-150 multi-cast data transfer (computer bus)

SpW-RT shall support multi-cast data transfer.

Test solution:

The multi-cast data transfer requirement should be checked on SystemC models only as only a network model can be used for proving this requirement. SpaceWire-RT uses SpaceWire network layer, consequently, it provides all SpaceWire network features, particularly multi-cast. For multi-cast testing switches will be configured in such a way that particular logical address corresponds to multi-cast and, consequently, switch provides a set of output ports for transmitted multi-cast packet.

4.2.11 Out-of-band signals and information

REQ-160 out-of-band signals

SpW-RT shall transfer time-ticks and interrupts with very short latency.

Test solution:

The out-of-band signals and information requirement should be checked on SystemC models only. Addressing SpaceWire-RT this requirement corresponds to the latency of broadcast messages, which are provided by the broadcast layer of SpaceFibre part of SpaceWire-RT stack. During testing, the latency for the broadcast messages delivery will be evaluated on different topologies.

4.2.12 Communication

REQ-190 Communication

SpW-RT shall support the communication requirements as described in the table below.

	Distance	Rate	Latency	Packet size	QoS
Data-handling network	Short to long	Low to high	Not important	Short to long	Reserved bandwidth
Control bus	Short to long	Low	Low	Short to long	Deterministic delivery
Telemetry bus	Short to long	Low	Low	Short	Reserved bandwidth
Computer bus	Short	Very high	Low	Short to long	Reserved bandwidth
Time-sync bus	Short to long	Low	Very low	Short	High priority

Side-band	Short	Low to high	Very low	Short	High priority

Test solution:

The communication requirements should be checked by means of SDL simulation, SystemC simulation, as well as by analytic methods. The distance requirement is impossible to prove and test by modeling and simulation neither in SDL, nor in SystemC. Data rate requirement will not be tested as long as it more refers to the Physical Layer and user application characteristics. Physical Layer characteristics would be investigated in the WP4. The latency requirement will be checked as described in pp. 4.2.2 and 4.2.11. Packet size requirement will be checked as described in 4.2.1. The QoS requirement will be tested while specification and modeling in SDL and SystemC as SpaceWire-RT provides several types of quality of service (best effort, bandwidth reserved, priority, scheduled).

4.2.13 Non testable requirements

The consolidated RF and EU requirements include some requirements, which are impossible to prove and test by modeling and simulation neither in SDL, nor in SystemC. Here is the list of these requirements, which are out of scope of WP3:

- Data rate (REQ-10, REQ-11)
- Distance (REQ-20, REQ-21)
- Galvanic isolation (REQ-30, REQ-31)
- Transmission medium (REQ-40, REQ-41)
- Validity (Bit error rate) (REQ-90)
- Mass interconnect (REQ-170)
- Power consumption (REQ-180)

4.3 Non-nominal test cases

SpaceWire-RT standard has some problems and issues that are points of special interest and are to be tested carefully. Particularly, its operation should be checked in case of some errors which can occur during data transmission. The following subsections describe possible non-nominal cases for SDL and SystemC investigation of SpaceWire-RT.

4.3.1 Test cases for SDL simulation

4.3.1.1 Encoding Layer test cases

The Encoding Layer Receive Synchronisation State Machine should be carefully investigated. To check workability of the state machine the following types of errors should be simulated:

- ">4 Four RX" errors in CheckSync state
- Word realignment in CheckSync state
- Word realignment in Ready state

4.3.1.2 Lane Layer test cases

The Lane Layer Initialisation State Machine is the object of special interest as it has a large number of states and complicated transitions. Therefore, this state machine should be carefully tested in order to check its behavior under different conditions. The following cases will be simulated on the point-to-point SDL model:

1. One node is asserted LaneStart flag. Another node is configured to AutoStart.
2. For both nodes LaneStart flag is asserted. This case is intended to check the situation, when flag in one of the nodes is de-asserted and then AutoStart flag is asserted.
3. For both nodes LaneStart flag is asserted. While in Initialise state LoS occurred. This case is intended to check Lane Layer operation in case of loss of signal by receiver during initialisation phase. LoS signal will be modeled by the channel.
4. For both nodes LaneStart flag is asserted. LoS is detected after connection is established (i.e. in the Active state). This case is intended to check Lane Layer operation in case of loss of signal by receiver during normal operation. LoS signal will be modeled by the channel.
5. Modeling ">8RXERR" during different phases of operation (initialise, active). For this case the channel will introduce errors in the transmitted data with the specified BER.

4.3.1.3 Lane Control Layer test cases

Current specification of Lane Control Layer [3] does not define all necessary for multi-laning mechanism specification (e.g. lane synchronisation, lane distribution, lane concentration, etc.). For this reason at the current moment it is impossible to propose any non-nominal situations.

4.3.1.4 Retry Layer test cases

The main feature of the Retry Layer is the possibility of retransmission of data in case of an error occurrence during transmission. The following situations should be checked by means of simulation:

- ACK or NACK loss or corruption during transmission over the channel.
- Retry buffer overflow. This can be the result of the previous case, as no data is removed from the retry buffer.

4.3.1.5 Framing Layer test cases

Framing layer of SpaceFibre part of SpaceWire-RT is responsible for framing SpaceWire packet data, broadcast messages and FCTs to be sent over the SpaceFibre link and scrambling SpaceWire packet data for EMC mitigation purposes. Therefore, it does not contain complicated mechanisms which should be investigated under error conditions.

4.3.1.6 Broadcast Layer test cases

SDL model can be used for testing layer operation while registering and unregistering on broadcast messages reception of particular type of channel. SDL provides only point-to-point models, therefore, it cannot be used for checking broadcast message distribution over the network. This will be checked by SystemC simulation (see p. 4.3.2.4). Thus, Broadcast Layer does not contain any complicated mechanisms which can be investigated under error conditions at SDL point-to-point model.

4.3.1.7 Virtual Channel Layer test cases

The Virtual Channel Layer Medium Access control mechanisms should be the objects of particular interest. A medium access controller shall determine which virtual channel shall be allowed to send a data frame next in accordance with the QoS of the virtual channels and their precedence. Particularly, the case of bandwidth credit counter saturation should be investigated during simulation.

4.3.2 Test cases for SystemC simulation

4.3.2.1 Logical and path addressing tests

Logical and path addressing could be tested by replacement of Node address in the packet with wrong address during the transmission of the packet through the Channel:

1. One Node generates a packet for transmission to the some other Node and sends it to the SpaceWire-RT network.

2. In one chosen Channel module the address of the destination node is changed to the wrong one.
3. The Switch gets the packet, reads the address and discards the packet, because the address does not exist.

4.3.2.2 Wrong VC numbers test

This test is done by the specific configuration of the Switch when one of the ports does not have a virtual channel with a needed number:

1. The Node generates a packet for transmission through the VC #4 and sends it to the Network.
2. One of the switches is configured such a way, that incoming port has six virtual channels, but needed outgoing port has only two virtual channels.
3. When switch gets a packet through the VC#4 it has to send it further trough the VC#4 also, but it cannot. So the packet should be discarded.

4.3.2.3 Wrong frame sequence number

This test is done by replacement of #FR_Seq in frame with wrong one during the transmission of the data through the channel:

1. One Node generates a packet for transmission to the some other Node and sends it to the SpaceWire-RT network.
2. In one chosen channel module one of the #FR_Seq is changed to the wrong one.
3. The Switch (or Node) gets the frame, checks the #FR_Seq, rejects the frame and sends NACK.

4.3.2.4 An error during the broadcast frame transmission

1. The broadcast frame is sent to the SpaceWire-RT network and during the transmission in one of the Channels the broadcast frame is deleted.
2. The Node, whish is connected with that Channel, gets the frame with wrong #FR_Seq (because the broadcast is lost), and indicates the error to the Switch by sending of NACK.
3. Switch gets the NACK and resends the broadcast frame.

So this test checks the broadcast sending mechanism, plus it gives an ability to count the latency for the broadcasts with and without transmission errors.

4.3.2.5 Endless packet circulation

Such kind of the situation can be tested on a Circular network configuration by a special configuration of Switches, which sends round the packet. So we can model an endless packet circulation, which will show, how the network will work in such a situation and what would happen if one of the Virtual Channels or more will always be busy with sending round a packet. We will check if there a possibility to block the network with such a situation and how would it influence on a network loading.

4.3.2.6 Deadlock

To test it we configure every Node and Switch in the network such a way that it has only one Virtual Channel. Then in one of the Nodes we block all the outgoing FCTs and so we make the incoming buffer full. So we can model a deadlock. And then by adding the additional VCs, we can look at how the network would work with a deadlock in one of the VCs.

5 Conclusion

This document described the SpaceWire-RT simulation and validation plan, pointing out the core features of SpaceWire-RT that should be simulated and validated, test plan and main simulation and validation principles which are to be used in the WP3. We have defined main principles of simulation and testing in SDL and in SystemC. SDL and SystemC models will be used for checking correctness and consistency of SpaceWire-RT technology specification, checking the models against the requirements to the SpaceWire-RT technology in accordance with the given testing plan as well as checking non-nominal cases. It should be pointed out that the progress of WP3 significantly depends on the changes of SpaceFibre specification, which is the basis for SpaceWire-RT technology development. The current simulation and investigation plan has been prepared taking into account the following documents:

- Annex I - "Description of Work" to the SPACEWIRE-RT project, Grant Agreement # 263148
- D2.1 SpaceWire-RT Outline Specification, version 2.0
- SpaceFibre Standard Draft D v1.14
- D 1.1 Consolidated set of Requirements for SpaceWire-RT

The current WP3 Simulation and Validation Plan is based on the enlisted documents. The list is considered to be fixed. It should be pointed out, that the current plan is primarily based on the SpaceFibre Draft D specification. The work on the WP3, simulation and validation of the Specification is running now for almost 3 months, thus we are about a half of our way with it now. However, the next draft – SpaceFibre Draft E1, [10], is to be published soon, and its additions and changes would be applied also if time and work effots would permit. It mainly depends on changes in the specification for the protocol stack layers that have been already simulated and validated. Any other updates of the SpaceWire-RT specification would be a result of analysis of the WP3 and should be released next to it.

The planned work effort estimation is 26.15 person/months which is close to the work effort estimation defined in WP3 description. It covers all necessary specification, simulation and testing tasks to investigate SpaceWire-RT standard specification according to the WP3 objectives.

More testing and investigation could be considered with case studies of real spacecraft onboard systems for different classes of spacecrafts. However, it would require much more efforts (over additional 6-12 person/months), taking into account testing of different real onboard network configurations. It is out of the scope and resources of WP3 and can be performed in future projects.

References

1. Annex I - "Description of Work" to the SPACEWIRE-RT project, Grant Agreement # 263148
2. Parkes S.M, “SpaceWire-RT Outline Specification, version 2.0” , University of Dundee, 6th September 2012.
3. Parkes SM, Ferrer Florit A, Gonzalez A, and McClements C, “SpaceFibre”, Draft D, Space Technology Centre, University of Dundee, 29th February 2012.
4. Viacheslav Grishin, Paul Rastetter, Petr Eremeev, Anatoly Lobanov, “Consolidated set of Requirements for SpaceWire-RT, version 2.0”, Submicron, Astrium, 7th September 2012.
5. International Telecommunication Union. Recommendation Z100: Specification and Description Language (SDL). 2007.
6. Open SystemC Initiative (OSCI). (2005). *IEEE 1666™-2005 Standard for SystemC.*
7. Gipper, J. (2007). SystemC the SoC system-level modeling language. Embedded computing Design
8. Co-Modeling of Embedded Networks Using SystemC and SDL / S. Balandin, M. Gillet, I. Lavrovskaya, V. Olenev, A. Rabin, A. Stepanov // International Journal of Embedded and Real-Time Communication Systems (IJERTCS), IGI Global, 2011. Pp. 24-49
9. Steve Parkes "SpaceWire User's Guide", STAR-Dundee Limited, 2012
10. Parkes SM, Ferrer Florit A, Gonzalez A, and McClements C, “SpaceFibre”, Draft E1, Space Technology Centre, University of Dundee, (to be published).