SPACEWIRE NETWORK SIMULATOR

Session: SpaceWire Networks and Protocols

Short Paper

Artur Eganyan, Liudmila Koblyakova, Elena Suvorova
Saint-Petersburg University of Aerospace Instrumentation. 67, B. Morskaya, Saint-Petersburg, Russia
E-mail: artfla@rambler.ru, luda_o@rambler.ru, wildcat15@yandex.ru

1 INTRODUCTION

The simulation is an important task for device design, distributed systems building and network protocols development. Therefore, previously we have developed the SpWNM [1] intended for simulation of SpaceWire networks. In the course of further development on this project we have created SpaceWire Network Simulator (SpaNSim).

The SpaNSim is intended for designing, modeling and analyzing the SpaceWire networks of any topologies, and contains basic models of terminal node, routing switch and link. The SpaNSim has all features described in [1]: it allows designing and simulation of SpaceWire networks; implements wormhole routing, time flow and distributed interrupts mechanisms, generation and transmission of data packets; estimates workload of channels and devices; provides graphical networks design with MS Visio.

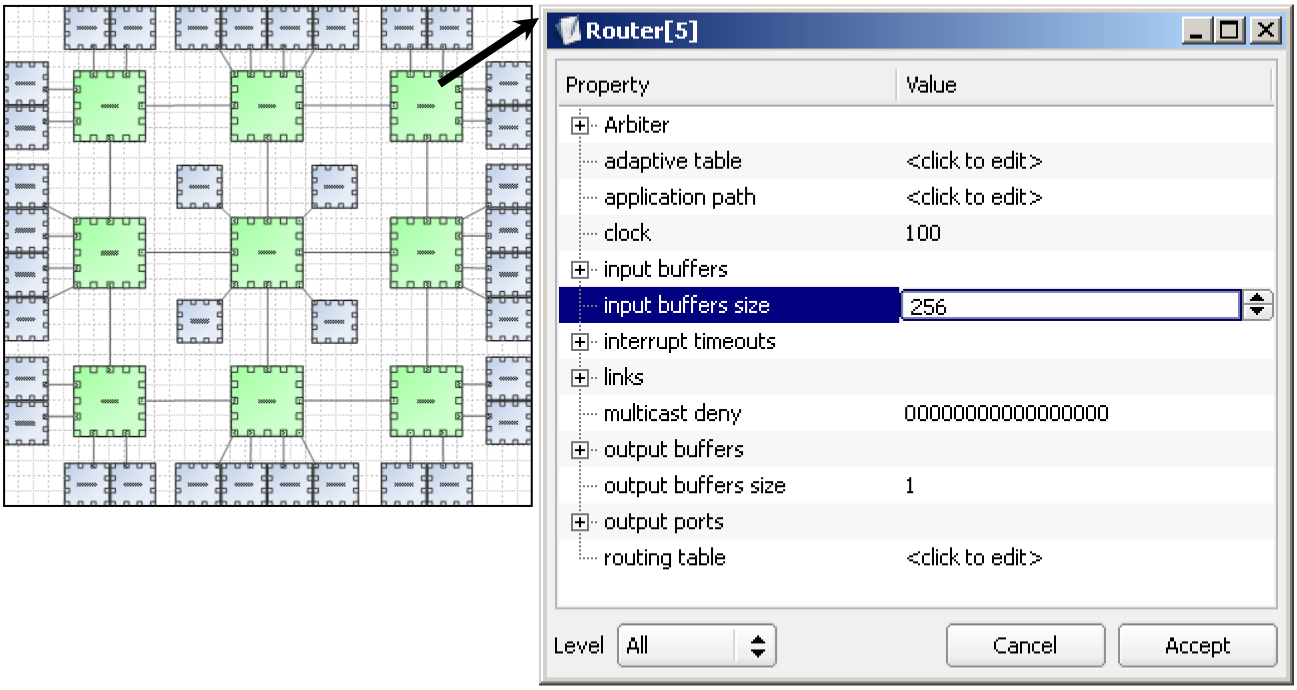
Unlike the SpWNM, SpaNSim:

- allows the user to create new device models and to write applications for them;
- allows to add new types of devices operating on different transmission standards and to connect these devices with SpaceWire networks;
- displays state of any device’s internal element: buffers content, register values and so on;
- contains a lot of modules for network analyzing: for example, one can see an in-depth information for each packet; can analyze in detail the network at level of bits, symbols, packets or particular events;
- displays all bits transferred through the channel and shows distorted bits;
- allows to describe the networks not only graphically, but with usual C++ projects, for example, in Visual Studio.

The SpaNSim is implemented in C++ and based on Qt and SystemC.

2 NETWORK DESIGN AND DEVICE SETTING

With SpaNSim you can graphically design the networks, using MS Visio. Also, the SpaNSim provides you to configure parameters and state of any device’s internal elements in interactive mode: buffers, registers, memory, clocks, and so on. It can be done during the network design in MS Visio or during the modeling. For example, you can check or change device’s buffers content after the modeling performed for some period.



Property	Value
Arbiter	
adaptive table	<click to edit>
application path	<click to edit>
clock	100
input buffers	
input buffers size	256
interrupt timeouts	
links	
multicast deny	00000000000000000
output buffers	
output buffers size	1
output ports	
routing table	<click to edit>

Fig.1. Network design and device setting in MS Visio.

3 CREATING NEW DEVICES

The SpaNSim provides you to create new device models. The new model can be implemented on basis of the existing one or without usage of existing models. For example, new device can be created by changing links of the terminal node or changing the routing switch’s arbitration or buffering scheme, channels number, and so on. For fast development, SpaNSim provides a set of ready-to-use functional units, so you can create a new device, connecting them with one another and setting their parameters.

Also you can write applications for the devices. For example, you can create an application operating inside the terminal node and sending/receiving data packets and control codes. For this, you should not know the node in details, you will only implement the application algorithm and use an interface between the application and node. Using this interface, your application will send or receive data packets, change transfer rate and so on. Each application will be presented as a independent program unit written in C++ or SystemC and attached to the node, so you can write various applications and estimate how they operate in the SpaceWire network.

In SpaNSim it is possible to add new types of devices. For example, one can implement a bridge binding the SpaceWire network with another ones: MIL-STD-1553, CAN, RS422-485 and so on. Hereby, networks of devices operating on different transmission standards can be implemented and modeled together.

4 ANALYZING SIMULATION RESULTS

The SpaNSim includes a lot of modules for analyzing modeling results. Using them the user can: estimate minimum, average and maximum propagation time of data packets, interrupts and time-codes; see information about any packet or control code transfer; see symbols flow transferred through the channels; see bit flows and find out bits distorted under the noise; get information about all occured timeouts. Also, the SpaNSim automatically builds charts displaying minimum, average and maximum propagation time for data packets and control codes.

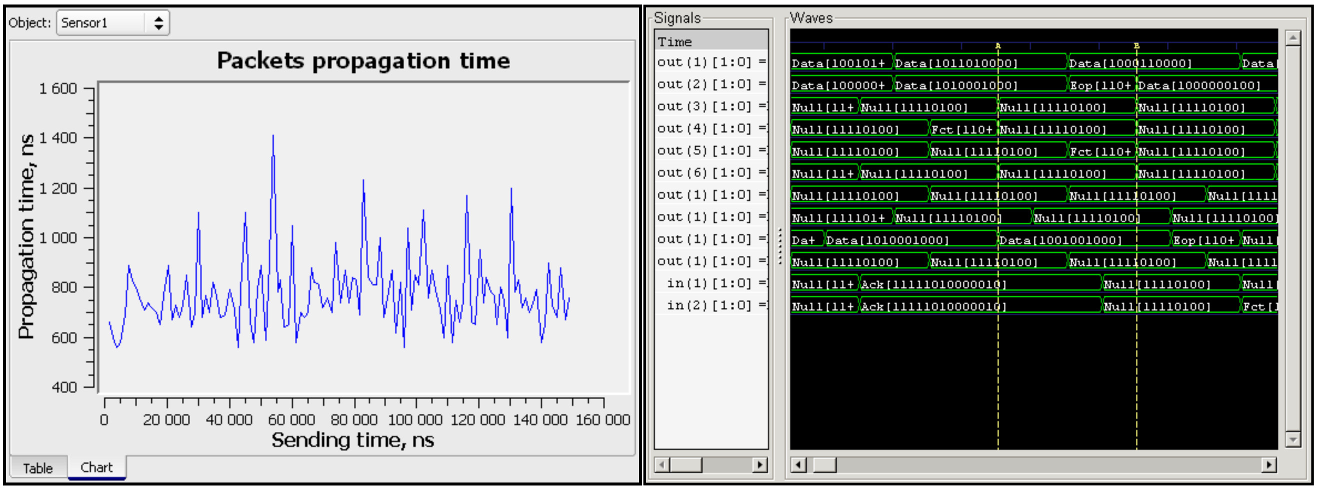

Object: Sensor1
Packets propagation time
Propagation time, ns
1 600
1 400
1 200
1 000
800
600
400
0
20 000
40 000
60 000
80 000
100 000
120 000
140 000
160 000
Sending time, ns
Table
Chart
Signals
Waves
Time
out(1)[1:0] =
out(2)[1:0] =
out(3)[1:0] =
out(4)[1:0] =
out(5)[1:0] =
out(6)[1:0] =
out(1)[1:0] =
out(1)[1:0] =
out(1)[1:0] =
out(1)[1:0] =
in(1)[1:0] =
in(2)[1:0] =
A
B
Data[100101+ Data[1011010000] Data[1000110000] Data
Data[100000+ Data[1010001000] Eop[110+ Data[1000000100]
Null[11+ Null[11110100] Null[11110100] Null[11110100]
Null[11110100] Fct[110+ Null[11110100] Null[11110100]
Null[11110100] Null[11110100] Fct[110+ Null[11110100]
Null[11+ Null[11110100] Null[11110100] Null[11110100]
Null[11110100] Null[11110100] Null[11110100] Null[1111
Null[111101+ Null[11110100] Null[11110100] Null[11110100]
Da+ Data[1010001000] Data[1001001000] Eop[110+ Null
Null[11110100] Null[11110100] Null[11110100] Null[1111
Null[11+ Ack[11111010000010] Null[11110100] Null
Null[11+ Ack[11111010000010] Null[11110100] Fct[

Fig.2. Analysis of data packets propagation time and displaying bits transferred through the SpaceWire channels.

Under development of new devices, the user can specify events and graphical dependencies he is interested in, and they will be appended to the results.

5 SPANSIM AND OPNET MODELER

The OPNET Modeler is one of the most popular software for network design and simulation. Therefore we have compared it with the SpaNSim relatively to SpaceWire networks simulation and analyzing.

In the default configuration of OPNET Modeler, there are no SpaceWire models ready to use. To simulate SpaceWire networks with OPNET, firstly you should have developed at least all basic devices: terminal node, routing switch, link and channel. For this, it is needed to write a lot of code and build finite-state machines implementing sending and receiving of data packets and control codes, supporting wormhole routing, adaptive group routing and so on. If you are going to transfer packets symbol-by-symbol (for example, to implement a wormhole routing), in OPNET each symbol should be defined as a packet itself. Bit-level transferring and analyzing (e.g., in channel with a noise) will be more difficult for implementation in the Modeler. In the SpaNSim, all of this is already implemented.

In the table below you can see comparison between some general characterictics and features of the OPNET Modeler and SpaNSim.

What is compared	SpaNSim	OPNET Modeler, v. 14.0
Network graphical design	Yes	Yes
Device graphical design	No	Yes
Setting of devices and their components	Yes	Yes
Displaying device's buffers and registers content	Yes	No by default

Programming languages used for device development	C++, SystemC	Primarily C. It is possible to use C++ and scripts.
Modeling core performance, in events per second	Approximately equal	
Information units	Symbols inside devices and bits inside the channels	Packets formatted by the user
Modeling results provided by default	Analysis of each packet and control codes. Analysis of minimum, average and maximum propagation times. Displaying transferred and distorted bits. Displaying a content of internal buffers and registers.	Animation for packets transfer. Minimum, average and maximum propagation times. Propagation time and a route for each packet.
Modeling results defined by the user	Any additional charts and text messages	Any additional charts and text messages
Modeling the networks of different protocols	Yes	Yes

By default, the OPNET Modeler does not support bit-level transfer and analysis, and does not provide detailed analysis of device buffers and registers. In SpaNSim, all these features are implemented.

In the Modeler, if you want to write an application for some device, you will usually change the device code. In the SpaNSim, devices provide convenient interfaces for applications, so you should not think about the device code, you only implement the application's algorithm.

Also, a new device in the SpaNSim is developed on the high-level language (C++ or SystemC) and with some ready-to-use modules. During programming, the user has a direct access to all device parameters, and he can use intuitive data types like «simulation time», «symbol», «packet», «routing table», and so on. In the OPNET Modeler, any device component is described with finite-state machine, and the user puts some C code in the states. Access to the device parameters is not as clear as it in the SpaNSim. So, in the OPNET Modeler it can be difficult to design a very detailed device model.

6 CONCLUSIONS

The SpaNSim provides graphical design, simulation and analysis of SpaceWire distributed systems. It has a set of ready-to-use devices and allows you to create new device models operating on SpaceWire or different standards. As a result of simulation, it is possible to estimate a wide range of characteristics required for building the distributed systems, and to define systems' parameters.

REFERENCES

1. Elena Suvorova, Liudmila Onishchenko, Artur Eganyan, "SpaceWire Network Functional Model", Session "SpaceWire networks and protocols", International SpaceWire Conference, Dundee 2007.