SystemC Simulation in Cadence Design Environment for Protocols and Networks Verification and Estimation

Sheynin Y.E., Suvorova E.A., Onischenko L.V.

Development of new protocols and their further implementation in switch/router chips require verification and parameter estimation of a protocol and its implementations. Due to the complexity of problems, formal models of a switch or a mesh of switches are complemented by building functional models and their verification and investigation.

In the paper we consider SystemC and Cadence tools (Incisive^{TM} Unified Simulator) and its application for building scalable models of routing switches and of reconfigurable networks of switches. Specific protocol features functional verification, discovering potential interlocks and deadlocks, network topologies characteristics estimation are performed on simulated testbeds. The results are used also to determine a switch buffers' size, to select an arbitration algorithms.

The new European standard SpaceWire for spacecraft on-board communications is the case study in the paper. A network of switch model is developed for evaluation of network parameters, for compilance to the SpaceWire standard verification, for elaboration of new features of the SpaceWrie standard.

The network model includes 3 types of components: the routing switch model, the terminal nodes model, the interconnect lines model. The functions of the terminal nodes models are generation and monitoring of data and control codes flows. The parameters of flows are specified by a user.

This models can include any number of switches and terminal nodes. We use two network topology specification methods: a universal – network described by interconnection graph, and a method for regular topologies specification (e.g. different meshes, hypercubes). In the second method a network structure is described analytically; topology type and number of nodes are user defined. The examples of networks are showed on the figure 1.

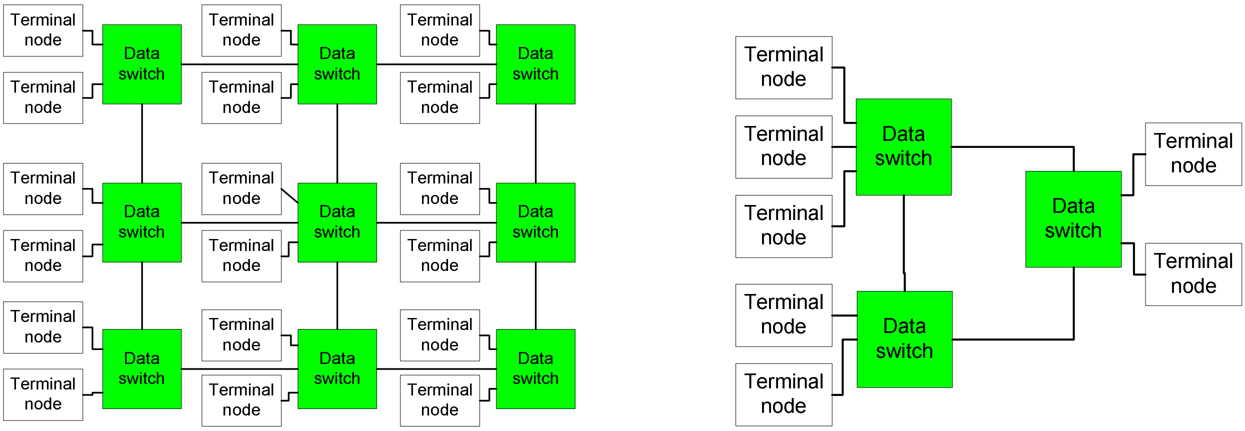


Figure 1. Examples of switch based networks

Let's consider analytical description of network from

Network topology type is 2D-grid with two terminal nodes connected to every switch (2DGrig), figure 1-a. The 2-D Grid parameters are N_n=3 (number of switches in one row), Nm=3 (number of switches in one column), N_t=2 (number of terminal nodes, connected to one switch), N_p=6 (number of ports in switch). Number of switches in the network model depends on N_n and N_m ; number of terminal nodes in the network model depends on number of switch and Nt, the number of interconnection lines depends on topology and number of nodes:

```
      Switch(Topology_type)
      {
      …
      Case 2DGrid:
      Switch_Number=Nn*Nm;
      Terminal_number=Switch_Number*Nt;
      Lines_number= Terminal_number*2+Switch_Number*4+Nm*2+Nn*2;
      Case …
      …
```

The network description:

```
      Sc_signal<interconnection_lines> int_lines[Lines_number];

      Terminal_node *tns[Terminal_number];
      For (i=0; i<= Terminal_number -1; i++)
      {
      Tns[i]= new terminal_node(node_name,tx_rate, rx_rate…)
      …
      Tns[i]->port_in(int_lines[f_terminal(i, topology_type, Nn, Nm, Nt,
Np, 0]);
      Tns[i]->port_out(int_lines[f_terminal(i, topology_type, Nn, Nm, Nt,
Np, 1]);
      …
      }

      Switch_01 *sw_01[Switch_Number];
      For (i=0; i<= Switch_Number -1; i++)
      {
      sw_01[i] = new Switch_01(switch_name, Np, tx_rate, rx_rate…)

      for (j=0; j<=Np-1; j++)
      {
      sw_01[i]->port_in[j](int_lines[f_switch_01(i, j, topology_type, Nn,
Nm, Nt, Np, 0]);
      sw_01[i]->port_out[j](int_lines[f_switch_01(i,   j,   topology_type,
Nn, Nm, Nt, Np, 1]);

      }
      }
```

The description of function `f_terminal` is

```
      Switch(topology_type)
      {
      …
      Case 2DGrid:
      If (direction==0)
      {
      Return(2*i);
      }
      Else
```

```
    {
    Return(2*i+1);
    };
    …
```

The description of the function `f_switch_01` is

```
    Switch(topology_type)
    {
    …
    Case 2DGrid:
    If (direction==0)
    {
    If ((j==4) or (j==5)) {return (i*4+(5-j)*2+1);}
    Else
    {
    If (i==0) {return(Terminal_number*2 + j*2);}
    Else
    {
    If (i<Nm) {return (Terminal_number*2 +(Np-2) *2+(Np-4)*2*(i-1) +
j*2);}
    Else
    {
    If ((I mod Nm)==0) {(Terminal_number*2 +(Np-2) *4+(Np-4)*2*(Nm-2) +
((I div Nm)-1)* ((Np-3) *4 + (Np-4)*2*(Nm-2)) +j*2);}
    Else
    {
    (Terminal_number*2 +(Np-2) *4+(Np-4)*2*(Nm-2) + ((I div Nm)-1)*
((Np-3) *4 + (Np-4)*2*(Nm-2))+ ((I mod Nm)-1)* ((Np-3) *4 + (Np-4)*2*(Nm-
2)) +j*2);
    }
    }
    }
    }
    }
    Else
    {
    If ((j==4) or (j==5)) {return (i*4+(5-j)*2);}
    Else
    {
    …
    }
    };
    …
```

We can include into network models not only system level models of switches (written in SystemC) but RTL-level models (written in VHDL) also. The Cadence design environment allows integrating of SystemC, VHDL and Verilog components' models.

Lets consider system level simulation of switch based network. The included into the network components could be described with different detalization. This allowed us choose between the resultant accuracy and reasonable simulation time. In compliance with the SpaceWire standard the length of different symbols t (number of bits in a symbol) could be different. Every symbol transmitted between nodes in our model is represented by a structure that includes a symbol type and a symbol value. The transmission of every symbol is accompanied by a strobe; the receiving node confirms receiving by a ready signal.

Every interconnection line model is characterised by set of parameters, that determines transmission rate and its variations and probabilites of errors. A transmission rate in compilance with the SpaceWire standard could vary between 2 and 400 MGz.

The most high-rate variant of a network model is operated fully synchronously; all transmission rates of the interconnection lines is equal the maximal transmission rate. The simulation time unit is determined by transmission rate. The unit value for 400 MGz is 2,5 ns. An average processing time for every symbol type is determined in terms of transmision time units. For simulation of other swithces with clock period not divisible 2,5 ns could be used smaller time unit. But this lead to an increase of the simulation time.

The average symbol processing time could be determined with real parameters of particular switches or their RTL models. It allows us to evaluate parameters of whole network for specifued switch parameters values, to determine are these values satisfactory or not on early stages of working on a project and thus to decrease project development time. Furthermore, parameter value could be overstated. In tis case we could not only reduce development time but also reduce hardware costs.

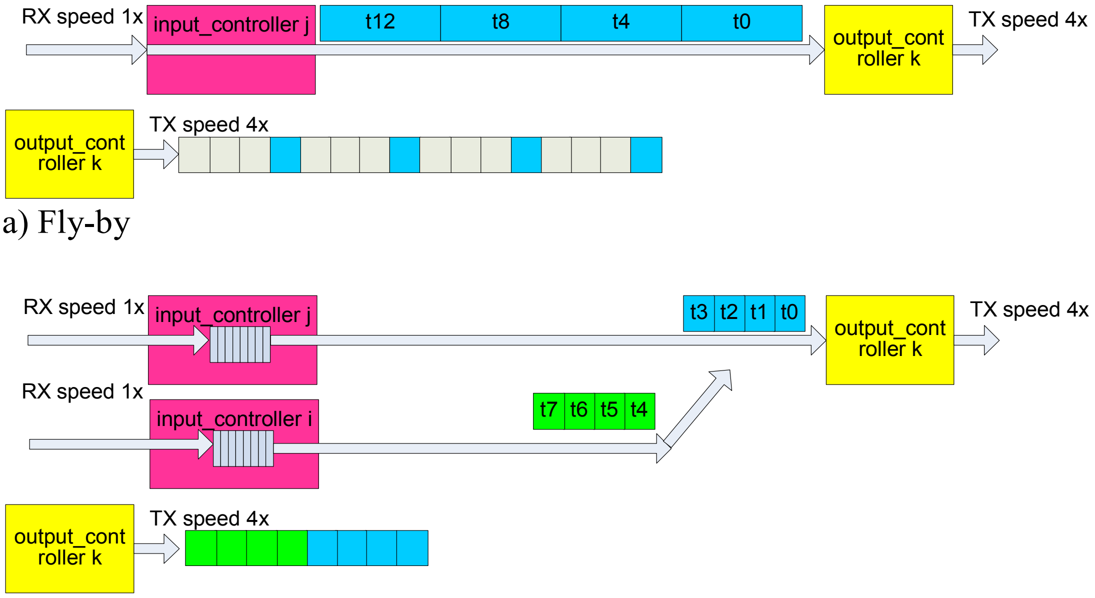
The model of a switch includes two main parts – the data processing block and the control code processing block. Buffers can be included in these blocks for evaluation of actual queues length. However this model could be used only for very preliminary evaluation of buffer size, because main load of buffer is determined when transmission rates in different interconnection lines are differ.

The switch model do not include routing table. Two approaches is used for packet routing: every packet is annotated by a number of destination port (or sequence of destination ports numbers) – the model of path addressing; for every port’s pair (i,j) is a probability of packet transmission from port i to port j and the probability distribution law are specified. This switch model allows to estimate qualitative switch parameters.

For more detailed evaluation a model in which for every interconnection line the parameters are assigned independently is used. The symbol transmission time is determined as a function of line transmission rate and a symbol type. More detailed switch models are used in this case.. The switch model structure in this level more precisely corresponds to a real structure of the RTL model. While the routing table is not used for determination of destination ports (configuration of these tables for whole network is too long and complex process) nevertheless in this model an access to routing table is simulated. It allows to evaluate access collision influence on a header processing time.

This model is used for buffer size evaluation, for selection of effective bufferisation scheme, for arbitration algorithm’s selection. The model could be used also for evaluation and selection of a necessary switch fabric channel throughput relative to switch ports throughput (interconnection lines throughput).

Another task is a choice of effective buffering scheme when port load is asymmetric with using of this model. The rate of source ports is several times less than the rate of port connected to handler port. If the fly-by commutation is used in this case then interconnection line to the handler will stand idle and waiting time for packets from other sources will increase (figure 2 a). The source rate in four times less than the handler transmission rate. As a result the line to handler will stand idle during time enough for three symbols transmission after transmission of every one symbol.



a) Fly-by

b) Switching with buffering

Figure 2. The data packet transmission with using of different bufferisation schemes.

Using of switching with buffering allows to decrease of high rate lines idle time due to preliminary data accumulation in a buffer (figure 2-b). However efficiency of buffering is essentially decreased when the packet size is greater than the buffer size due to the packet tail that is not placed into buffer when its transmission to destination port is started wold be transferred with slow rate with it is arrived into swith.

The developed switch model allowed to evaluate maximal acceptable packet size when buffer size is fixed or evaluation of necessary buffer size for a given packet size (average packet size and distribution law).

In the example, fig.3., maximal acceptable packet size for assymetric data transmission through 8-port switch with input buffers size 64 symbols (data packets from ports 1, 2, 3 are transmitted to the port 7, and data from ports 4, 5, 6 are transmitted to port 8, transmission rate from sources is in three times smaller than transmission rate to handlers). The diagram at the figure shows dependency between average packet transmission time and packet size when input port load is about 95%.

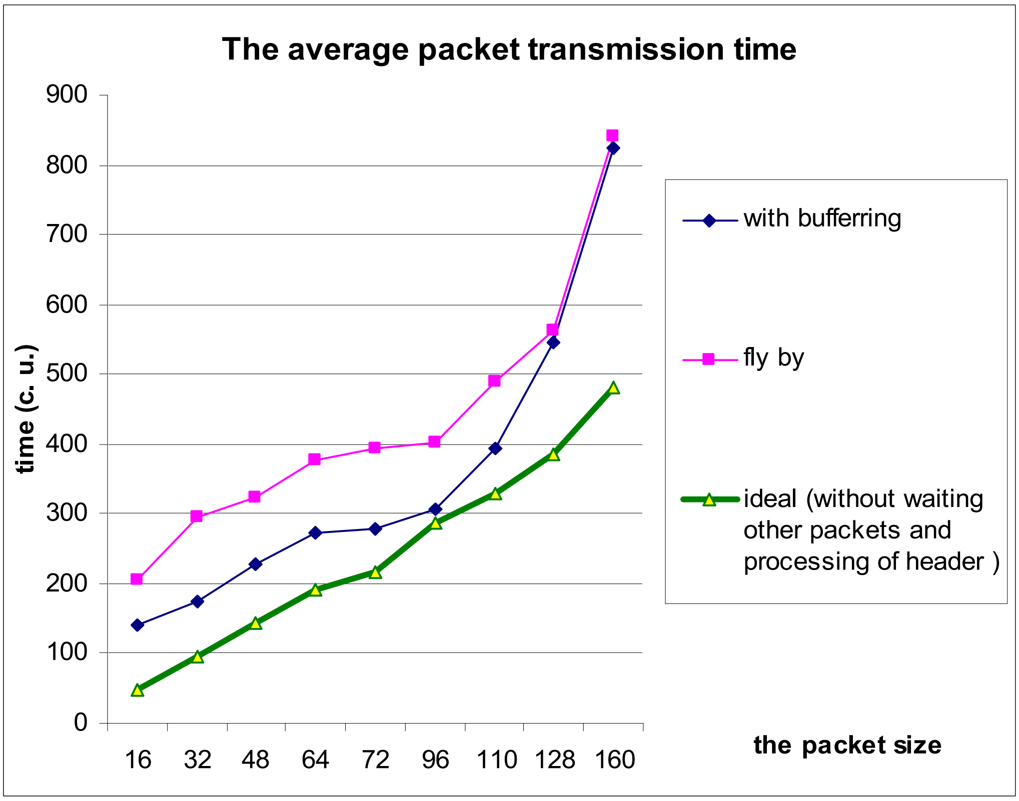


Figure 3 – the average packet transmission time

The model could be used for evaluation of ratio between the switch fabric data channel throughput and the port throughput (interconnection lines throughput). Increasing of switch fabric data channel throughput could be reached as due to increacing of lockal clock frequency as due to increasing of bytes number parrallelly transmitted in switch fabric data channel.

Increasing of switch fabric data channel throughput in combination with packet buffering leads to increasing packet transmission rate through switch fabric, but the port throughput will be limitation factor. Let's consider an 8-ports switch. The packet transmission from every port to one of the other ports is equiprobable. The average packet size is 64 symbols, the buffers size is 64 symbols. The diagram (figure 4) showes the dependence between average packet transmission time and average input port load for different ratios of switch fabric data channel throughput and port throudhput. The throughput ratio essentially affects the average packet transmission time when input port load is more than 95%. When the input port load is 99% increasing of throughput ratio from 1:1 to 4:3 allow decreasing the average packet transmission time to 11%. Further increasing the throughput ratio is less effective: an increase of throughput ratio from 4:3 to 2:1 leads to only 5%decrease of the average packet transmission time.

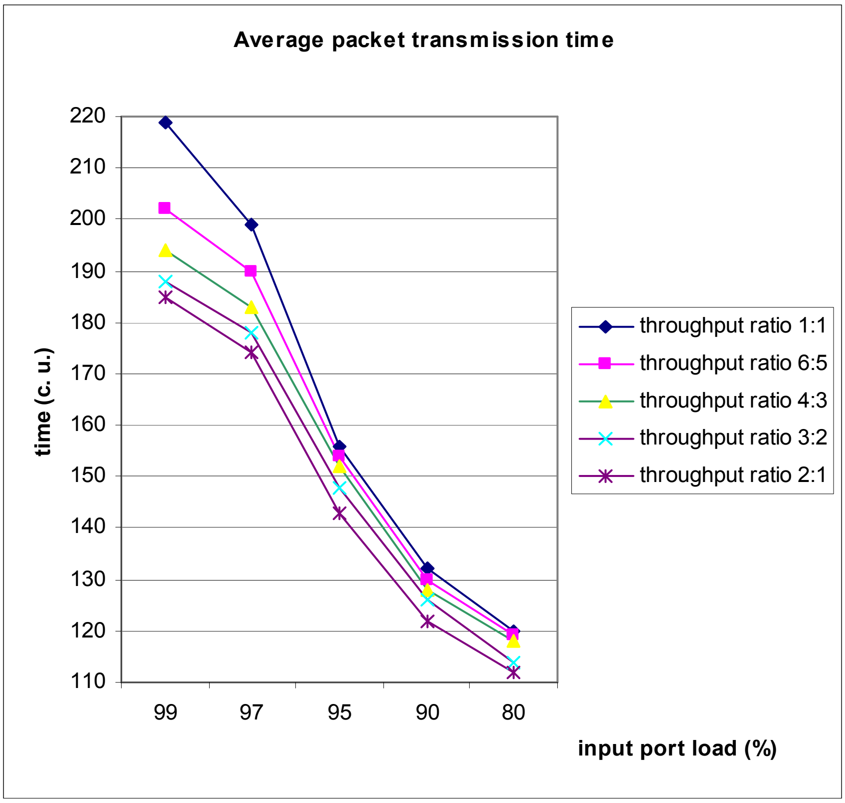


Figure 4. - Dependency between the average packet transmission time and the average input port load for different throughput ratios

For simulation of initialization process. we use more detailed models. These models includes routing tables with the same format as routing tables in switch devices. These models could be used for verification of RTL- or post-synthesis models and for verification of an arbitration algorithm.

In this case the terminal node model includes the character level interface component, that support interface between terminal node and RTL or post-synthesys model of the switch, fig.5.

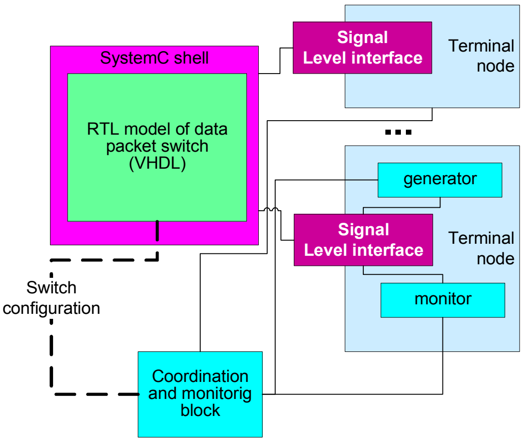


Figure 5. The structure of test shell for RTL or post-syntesys models of switches.

The test process includes next three stages:

1. The initial configuration stage. On this stage the coordination and monitoring block executes initial settings for switch model and terminal nodes models.

2. The data packet transmission stage. On these stage data packets are transmitted between the switch and the terminal nodes. The duration of these stage could be assigned as absolute value or user could assign number of packets that are transmitted from every terminal node. The terminal nodes execute preliminary control of data packet transmission and timing parameters evaluation at this stage.

3. The global data packet control and collects statistics.

During data packet transmission testing we need to control: packet header must be excluded from the packet or not excluded according routing table settings; packet contents and end-of-packet symbol must follow without changes. The set of ports, that reseived this packet must correspond to its address and adaptive routing settings. For transmission control every packet contents includes special information. The packet generator writes every transmitted packet into a log file. When the monitor receives the packet it uses log files for control packet contents and end-of-packet symbol, control of packet header processing. But only the coordination and monitorig block executes final packet header processing control. It controls multicast transmission and adaptive group routing.