An Animated Graphical Simulator for the IEEE 1596 Scalable Coherent Interface With Real-Time Extensions

Attention: Please Read This Page

This paper was originally presented in August of 1997. While we believe that the material is important and will be interesting to current users of Foresight Systems software, we have been through some changes since the paper was prepared.

Nu Thena Systems, Inc. was a predecessor company to Foresight Systems M&S. All references to Nu Thena are obsolete. Readers should assume that Foresight Systems M&S replaces all instances of Nu Thena Systems, Inc.

The current contact information for Foresight Systems M&S is:

Foresight Systems M&S
12550 N. 89th St.
Scottsdale, AZ 85260

http://www.foresightsystems-mands.com
fs_marketing@foresightsystems-mands.com

Some of the links to CERN sites are no longer active. We apologize for any potential confusion. The email address for support at Foresight is:

fs_support@foresight-mands.com

If you have questions, concerns, or issues, please contact:

Dean Stevens
President, Commercial Systems Group
Foresight Systems M&S
deans@foresightsystems-mands.com
(408) 278-3983

An Animated Graphical Simulator for the IEEE 1596 Scalable Coherent Interface With Real-Time Extensions

W. A. Windham, C.E. Hudgins, J.E. Schroeder,
Harris Corporation, Government Aerospace Systems Division
Mike Vertal, Nu Thena Systems, Inc.

As part of the Very High Speed Optical Networks (VHSON) Demonstration Program, a simulator has been developed. The Scalable Coherent Interface Network Interface Functional Simulator (SCINIFS) addresses the Institute of Electrical and Electronic Engineers (IEEE) standard 1596-1992 for the Scalable Coherent Interface (SCI) [1] . In addition, SCINIFS addresses the evolving work of IEEE project P1596.6, entitled SCI Real-Time (SCI/RT). The SCI/RT project addresses extensions to IEEE Standard 1596 that seek to enhance SCI's utility for real-time fault-tolerant computing applications. SCINIFS addresses the functional behavior and performance of SCI/RT nodes when connected in a ring. The development of the simulator, early results and future work to model SCI switches are described.

To find out more about SCI point your browser to:
`http://sunshine.cern.ch:8080/SCI/`

INTRODUCTION

The Scalable Coherent Interface Network Interface Functional Simulator (SCINIFS) is a graphical simulation model and simulation environment supporting multiple objectives:

- SCINIFS provides a tool for use in evaluating and selecting algorithms for bandwidth and latency management, in support of the IEEE P1596.6 SCI/RT project.
- SCINIFS provides a tool for investigating candidate hardware architectures and evaluating system performance for specific SCI/RT implementations.
- SCINIFS provides building blocks for use in creating the SCI Switch Architecture Simulator (SCISAS) and the SCI System Simulator (SCISIM), future work that will provide the overall simulation capabilities shown in Figure 1.

SCINIFS focuses on aspects of SCI implementation and system analysis not addressed by earlier SCI-related simulators such as SCILab and the IEEE Std 1596 SCI C code. In addition SCINIFS' structure is closely analogous to a generalized microcircuit implementation. SCINIFS provides an animated, highly graphical, model that yields an interactive perspective of system behavior.

SIMULATION ENVIRONMENT

SCINIFS provides a complete graphical environment for modeling, simulating,

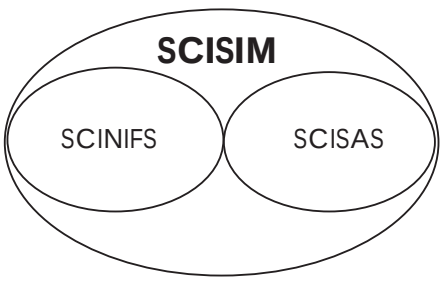

Figure 1: Interaction of SCI Simulation Environments

Paper submitted to the National Aerospace & Electronics Conference, July 14–18, 1997, Dayton, Ohio, USA.

and analyzing results. The simulation system consists of a preprocessing Graphical User Interface (GUI) and simulation and postprocessing elements based on models and tools provided by the Foresight simulator from Nu Thena Systems, Inc. The GUI was developed at Harris using Motif and provides a method for specifying the parameter values to be used in simulations (such as host loading).

With both graphical and textual modeling constructs, Foresight [2] provides a rich simulation environment. Foresight models are built hierarchically from a collection of modeling constructs. Foresight's primary modeling construct is a hierarchical flow diagram. These allow systems to be described as a network of communicating processes, and are well suited for modeling embedded system functionality. Animation capabilities exist at this level that allow "real-time" observation of system performance as a simulation proceeds.

Process blocks within a Foresight flow diagram may be additional flow diagrams, or with Foresight primitive, executable modules: state machines, minispecifications, library elements, or user-defined reusable elements. This allows detailed modeling of the primitive elements while still allowing visualization of the dynamic behavior and performance characteristics of the system at the top levels.

TOP-LEVEL MODEL DESCRIPTION

SCINIFS addresses both the functional behavior and temporal performance of arbitrary configurations of SCI/RT nodes when interconnected in a ring topology. Arbitrary single-ring topologies may be constructed graphically by replicating and interconnecting reusable elements.

At the top level in the model are the experimental frames. Each experimental frame consists of an interconnected set of nodes, one of which also acts as a scrubber, and a separate block for parameter entry. A four-node experimental frame is shown in Figure 2.

Each node is modeled as a host subblock and a network interface subblock.

Additionally, one node is assigned to do the scrubber function. The scrubber function is performed in a separate subblock that only contains scrubber logic. The subblocks contained in the model for the scrubber node are shown in Figure 3.

The host subblock represents the application hardware being supported by the network and is independent of the SCI standard. In the simulator, the host is necessary to generate requests and responses for specific simulations, hence packets are generated at a user-specified rate within the host. The host also gathers performance statistics at run time during the simulation.

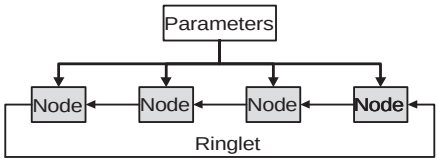

Figure 2: Four Node Experimental Frame

The node interface subblock consists of a Node Interface Unit (NIU) and a Processor Interface Unit (PIU). The NIU manages the interface to the SCI ringlet and consists of Receiver and Arbiter logic, a Bypass Queue and an Idle Buffer. The PIU contains the Transmit and Receive Queues, and the logic for transferring packets between the NIU and the Host.

The protocol primitives (i.e., packets) are modeled to the symbol field level (modeling at the bit level would be computationally inefficient and would eliminate most of the productivity gains of using a higher-level simulator rather than a VHDL-based tool). Packets are generated and transferred as such (i.e., simulation events are for the transfer of the packet as a whole, although all of the fields of the packet will be represented).

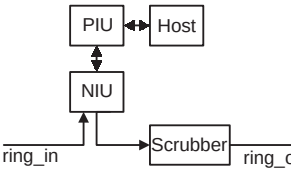

Figure 3: Scrubber Node Subblocks

Because the simulation can operate in one of two modes, base SCI fair-only mode or SCI/RT mode, there are two models for each element that is affected by the differences in the two modes. In particular, elements associated with idles (Receive Idle, Arbiter and Scrubber) are affected because the format and use of idles are different in the two modes.

An Animated Graphical Simulator for the IEEE 1596 Scalable Coherent Interface With Real-Time Extensions

Idles and Flow Control

When there are no packets being transmitted on a link, idle symbols are transmitted to keep the transmitting node and the receiving node in synchronization. (To minimize simulation time, an idle is transmitted in the simulation only when there is a change in the contents of the idle, i.e., if the pending idle transmission is different from the last one sent.) One or more idle symbols are also inserted between each packet.

"Thank You" to the authors for making the SCI model described in this paper available for CERN users. For additional information, please send an email to: foresight.support@cern.ch

Flow control in SCI is based on information contained in the idle symbols, with different implementation details for the fair-only SCI mode and SCI/RT. The flow control mechanisms are simulated by the Idle Processor subblock in the NIU. The low go (lg) bit in the idles is the low priority bandwidth allocation-control flag. For the base SCI version, all of the Idle Buffers are initialized with the lg bit set (since the lg extension causes the lg bit to fill the ring). For the real-time version, there are separate storage and processing for the fair and unfair idles.

Packet Transmission

A packet is queued for transmission from a node when the PIU transfers from the host to the appropriate Transmit Queue. In the real-time version, the Transmit Queues are priority ordered based on the send priority (spr) field. Once a packet is inserted into a Transmit Queue, it stays there until explicitly deleted by one of several mechanisms.

Information on system-modeling and simulation with Foresight is available at: http://wwwinfo.cern.ch/ce/ms/foresight/

When a packet is inserted into a Transmit Queue, an echo timeout is started for it. When the timeout expires, if the packet is still in the Transmit Queue, the packet is deleted. For requests, a response timeout is scheduled (and rescheduled if a busy status is received for the request).

Each packet in the Transmit Queues has a transmit status associated with it. When a packet is first inserted into a Transmit Queue, it is marked as not transmitted. When a packet is transmitted, it is so marked. The PIU also provides the capability to mark a packet as not transmitted (when a busy echo is received) and to replace a packet (when a Start Broadcast must be translated into a Resume Broadcast).

Access to the output link from a node is controlled by the Arbiter subblock in the NIU. Packet transmission is initiated once there is a packet in a Transmit Queue and the appropriate go-bit conditions have been met, when a previous transmission has been completed and the Bypass Queue is not empty or when the node receives a packet for retransmission from the previous node. The packet selection and the packet transmission are pipelined (i.e., the transmission time for one packet overlaps the arbitration time for the next).

The Bypass Queue always has precedence over the Transmit Queues. In the real-time version, fair-packet transmission has precedence over the unfair (since the fair is limited by the bandwidth allocation). The request and response queues are selected in round-robin order. After a processing delay, the packet is considered ready for transmission. Another packet is not selected until transmission of the last packet selected begins. Information in the transmission buffer is transferred to the next node after a time delay simulating the physical propagation delay of the link.

Received Packet Processing

When a packet arrives at a node, the receiver subblock in the NIU determines whether it should be stripped by the node, whether it is an echo or an idle or whether it should be forwarded (or transferred to the Bypass FIFO if the node is already transmitting). Packet forwarding and transfers to the Bypass FIFO begin after a receiver processing delay, without waiting for the end of the packet to arrive. The receiver will change the CRC status to "stomped" if the CRC indicates an error. In the case of Start Broadcast commands, the receiver may

transfer a packet to both a Receive Queue and to the Bypass FIFO for forwarding.

The packet is transferred to the appropriate Receive Queue in the PIU depending on the A/B Retry state and the command phase of the packet. If the packet requires an echo, the receiver generates an echo packet (indicating busy status if the packet was not accepted) and passes it to the Bypass FIFO. The transfer of the echo packet does not begin until the entire incoming packet is received, since the receiver discards the packet if the CRC indicates an error. The receiver also will translate a Resume Broadcast into a Start Broadcast if necessary.

If the received echo indicates a busy status, then the receiver signals the PIU to mark the corresponding packet in the appropriate Transmit Queue for retransmission and a new Echo Timeout is scheduled. If the echo indicates a not busy status, then the receiver signals the PIU to delete the corresponding packet in the Transmit Queue.

If the parity is good, idles are saved in the Idle Buffer and the “lg” bit is or’ed with the previously saved value (also the “hg” bit for the base SCI version). For the real-time version, if the “full” or “more” indications are set and are equal to the current node’s ID, these fields are cleared.

The PIU transfers data from the PIU's Receive Queues to the host in round-robin order, transferring messages from the Receive Queues to the Host as long as packets are available. There is a fixed processing delay plus a delay based on the number of symbols in the packet. If the packet was a response, the PIU updates its count of outstanding requests. The host in turn generates responses for received Requests after a time delay. End-to-end latency and queue waiting time are calculated and displayed for each received packet.

As noted earlier, the scrubber functions are performed by one node on the ringlet. When the Scrubber receives a packet, if its old bit is already set then the Scrubber discards the packet. If it was a Send packet, the Scrubber replaces the discarded packet with an Echo and an Idle. If it was an Echo packet, the Scrubber replaces the discarded packet with an Idle. (As noted earlier, multiple idles are sent in the real system, but for purposes of the simulation, a single idle is more efficient.)

EARLY RESULTS

The first simulation runs on SCINIFS were for verification of correct functional operation of the model for both base SCI and for the two-bit priority approach proposed for SCI/RT. Correct behavior was observed and the functional implementations were used as the starting point for the design of a dual port node interface chip. Additional functional simulation runs were made to evaluate different internal data path options for the node chip.

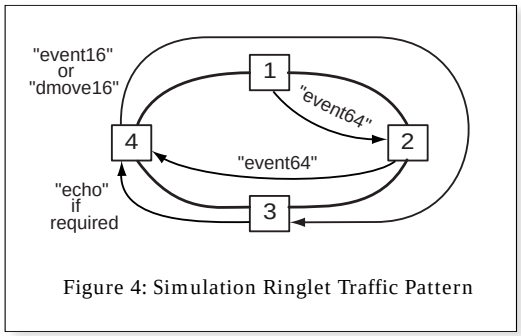

Figure 4: Simulation Ringlet Traffic Pattern

Performance of a given SCI network depends both on arbitration for the transmit link and on the insertion and removal of packets from both the transmit and receive queues. Parameters of interest include bandwidth utilization, latency and system behavior under heavy loading conditions. Performance simulations are ongoing to evaluate these parameters for a variety of network configurations and loadings.

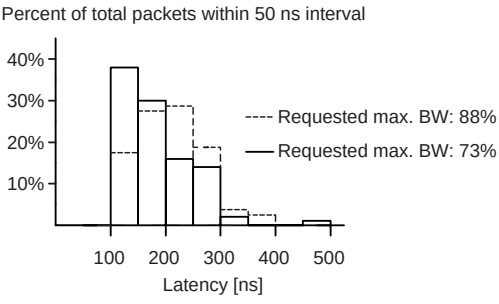

Figure 5: Effect of Increased Ringlet Loading

The “event_xx” class of operations were used for the initial bandwidth utilization and latency tests. Since “events” are not echoed, they are discarded from the transmit queue immediately after transmission and are minimally affected by transmit queue size. Similarly, a

An Animated Graphical Simulator for the IEEE 1596 Scalable Coherent Interface With Real-Time Extensions

destination node simply discards incoming "events" if its receive queues are full.

The data transfers simulated are shown in Figure 4. Multiple node-to-node transfers can occur simultaneously, giving an aggregate data transfer capability significantly greater than the individual link bandwidths. Figure 5 shows the increase in latency for packet transfers from node 4 to node 3 as ringlet loading is increased. With the higher loading, the probability increases that an incoming packet will arrive at a node that is already transmitting. The latency for incoming packets is thus increased by the time spent waiting for access to the output link. To minimize this increase in latency for high priority packets, a two bit priority approach has been proposed for SCI/RT. In this approach, priority bits in the idle symbols are used to inhibit transmissions from "fair" nodes when higher priority ("unfair") traffic is waiting to transmit. Figures 6 and 7 compare the results when all traffic is fair and when the transfers from node 4 to node 3 are given priority.

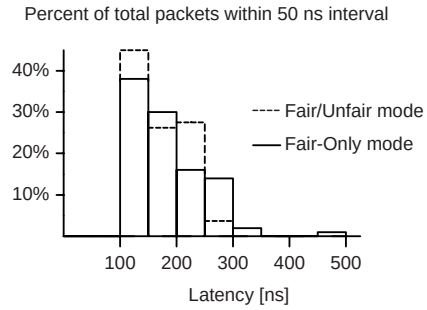

Figure 6: Comparison of Fair-Only and Fair/Unfair Modes (Requested Maximum Link Bandwidth: 73%)

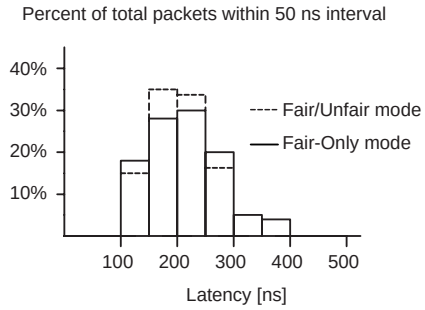

Figure 7: Comparison of Fair-Only and Fair/Unfair Modes (Requested Maximum Link Bandwidth: 88%)

Table 1 summarizes the "event_xx" simulation results. The effectiveness of the SCI/RT priority approach can be seen by the decrease in the maximum latency observed. For both SCI and SCI/RT, the use of idles for flow control reduces the bandwidth available for actual packet transmission. As shown in the table, the priority-based flow control approach requires slightly more overhead when the ringlet loading is high.

The effect of finite transmission queue size is seen in the results of simulations using the traffic patterns shown in Figure 4 but with (echoed) "dmove16" transfers between nodes 4 and 3. The difference between the "event16" and "dmove16" results is shown in Figure 8.

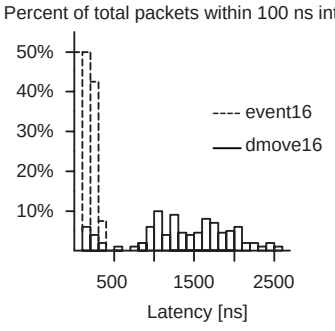

Figure 8: Comparison of "event16" and "dmove16" Results (Fair-Only Mode, Requested Maximum Link Bandwidth: 88%)

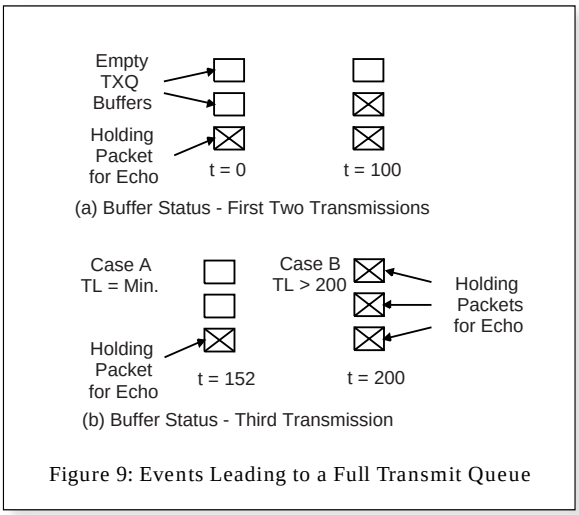

Figure 9: Events Leading to a Full Transmit Queue

The cause of the greatly increased latency for the "dmove_xx" transfers is shown in Figure 9. For the conditions simulated, a packet was ready for transmission on the ringlet every 100 ns. Best case ringlet latency from node 4 to node 3 was 125 ns. However, under heavy traffic conditions, ringlet latency increased significantly, causing the transmit queues to become full. Hence overall latency was increased by the additional time required for a transmit queue to be cleared (by a returned echo) and become available for accepting a new packet from the host.

Parameter / Mode	Max. Requested BW = 73%	Max. Requested BW = 88%
Max. Latency / Fair	480 ns	395 ns
Max. Latency / Fair-Unfair	295 ns	295 ns
Act. BW / Fair	63%	74%
Act. BW / Fair-Unfair	63%	70%

Table 1: Summary of "Event_xx" Simulations

Plans

SCINIFS is the first part of a general simulation effort and will be used for additional simulations at the ringlet level. In addition, a SCI/RT Switch Architecture Simulator (SCISAS) will address the band-functional behavior, and also performance, of an SCI/RT switch. Together, SCINIFS and SCISAS will form an SCI/RT System Integration Simulation (SCISIM) that will address the function, and performance, of an integrated SCI/RT system.

Acknowledgments

Funding and support from AAAS-1, Wright Laboratory is gratefully acknowledged.

References

[1] Scalable Coherent Interface, IEEE Std 1596- 1992, 2 August 1993, 248 pages.

[2] M. Vertal, Foresight: System Simulation for System Developers, Proceedings of the 27th Annual Simulation Symposium, IEEE Computer Society Press, La Jolla, CA, April 1114, 1994.

An Animated Graphical Simulator for the IEEE 1596 Scalable Coherent Interface With Real-Time Extensions