

DISTRIBUTED INTERRUPTS MECHANISM VERIFICATION AND INVESTIGATION
BY MODELING ON SDL AND SYSTEMC

Session: SpaceWire networks and protocols

Short Paper

Liudmila Onishchenko, Artur Eganyan, Irina Lavrovskaya

Saint-Petersburg University of Aerospace Instrumentation. 67, B. Morskaya, Saint-Petersburg, Russia

E-mail: luda_o@rambler.ru, artfla@rambler.ru, i.lavrovskaya@mail.ru

ABSTRACT

Distributed Interrupt mechanism has been proposed for next SpaceWire standard release. Interrupt codes and Interrupt_Acknowledge codes are low-latency signaling codes, according to specification their distribution does not depend on data flow. That make it useful for real-time distributed systems interconnections.

In this paper we present: the distributed interrupts mechanism verification by using model on the Specification and Description Language (SDL); investigation and analysis of this mechanism by using the SpaceWire Network Functional Model in SystemC.

SDL ToolSuite gives an ability to implement specifications. A distributed interrupt system implementation on the SDL, allows to have a reference implementation for it and check how Interrupt codes and InterruptAcknowledge codes are send through the network. This model can be verified and it is very useful to check the correctness of the specification.

The SpaceWire Network Functional model includes a description of basic SpaceWire network elements like node, routing switch and link, allows to assemble a SpaceWire interconnection system of required structure, implements wormhole routing, time flow and distributed interrupts mechanisms, generation and transmission of data packets. We use this tool for distributed interrupts mechanism investigations. It includes the following research:

- Distributed Interrupt mechanism in case of: networks with adaptive group routing, different channels rate; networks with cyclic topology like mesh, hypercube and so on; several sources and handlers of the same Interrupt code type
- The research of timeout mechanism as the instrument for protection and recovery after faults

In this paper we also give some recommendations about using the distributed interrupts mechanism and choosing the timeout parameters for efficient recovery.