

REAL-TIME SIGNALING IN SPACEWIRE NETWORKS

Session: SpaceWire Networks

Short Paper

Yuriy Sheynin, Sergey Gorbachev, Ludmila Onischenko

St. Petersburg University of Aerospace Instrumentation

67 B. Morskaya str., 190 000, St. Petersburg, Russia

E-mail: sheynin@aanet.ru

ABSTRACT

In the paper we consider a complementary mechanism of Distributed Interrupts as additional type of control codes for the SpaceWire standard extension. Give main ideas of its operation, recovery in case of errors. Estimate latency characteristics of control codes distribution in SpaceWire networks, reason about timeout values for error recovery mechanism.

1. INTRODUCTION

SpaceWire is a prospective networking technology for building on-board interconnections in prospective satellites and spacecrafts. Based on it integral interconnection infrastructure can efficiently substitute several (3-5) separate interconnections that are used in modern distributed on-board systems: *sensor buses* for data streams from sensors and instruments; *command buses* for commands from control units to instruments and spacecraft equipment; *telemetry busses* for telemetry data; *data buses* for data exchange between computing modules in the course of data and signal processing. Time-codes - a specific SpaceWire standard feature that distinguishes it from one of its predecessors the IEEE 1355, give a facility to incorporate into SpaceWire interconnections on-board clock synchronization also, thus eliminating customary separate *time synchronization buses*.

Still, sideband extensions custom signals are often used for hard real-time signaling and control: system event signals (e.g. interrupts, error notification), data sampling, data transfer coordination, etc. To integrate such signal into SpaceWire interconnections also we proposed a complementary mechanism of Distributed Interrupts. Implemented as additional types of specific control codes, Interrupt codes, together with Time-codes of the origin SpaceWire standard, they form an extended set of control codes for low latency real-time signaling in SpaceWire interconnections.

The SpW control codes, due to their specification at the low levels of the SpaceWire protocol stack, are distributed by the same cables, channels that data packets are sent and switched, but their distribution does not depend on data packets flow intense and can traverse even blocked by data channels and paths. This core feature differs the SpaceWire from other high-rate interconnection standards and makes it most appropriate for real-time distributed systems interconnections.

2. DISTRIBUTED INTERRUPTS

To implement Distributed Interrupts we extend control codes by 2 additional ones. As the Time-Code, they are formed from ESC followed by a single data character, Fig.1.

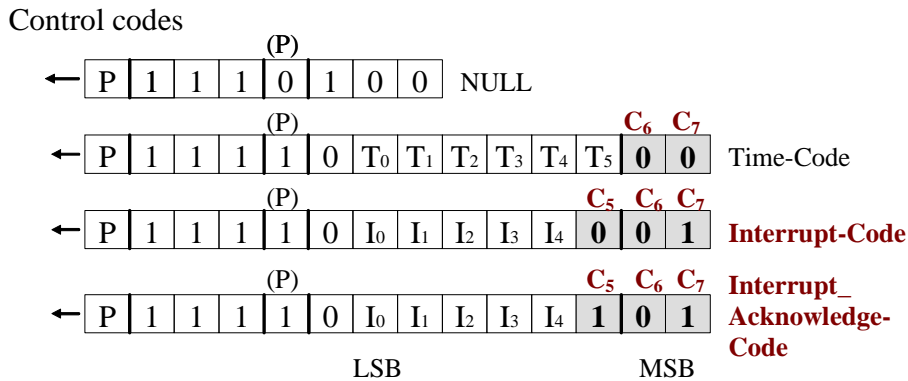


Fig.1. Extended control codes, with Interrupt/IntAck codes

Five bits of interrupt information are held in the least significant five bits of the Interrupt-Code (I₀-I₄); three most significant bits (C₅=0, C₆=0, C₇=1) contain control flags that are distributed isochronously with the Interrupt-Code. Five bits of interrupt acknowledge information are held in the least significant five bits of the Interrupt_Acknowledge-Code (I₀-I₄); three most significant bits (C₅=1, C₆=0, C₇=1) contain control flags that are distributed with the Interrupt_Acknowledge-Code.

Interrupt-Code represents a system signal request, e.g. an interrupt that is formed by a node in a SpaceWire network. It is issued by a node link that will be considered as the source node for this interrupt (Interrupt Source). It is distributed over the network to other nodes. An Interrupt-Code should be accepted for handling in some node of the SpaceWire network, which will be called the Interrupt Handler. The host of the node is supposed to implement some interrupt processing routine. One of 32 interrupt request signals (interrupt source identifiers) could be identified by the Interrupt-Code.

Interrupt_Acknowledge-Code represents a confirmation that the Interrupt-Code has reached some Interrupt Handler and has been accepted by it for processing. The Interrupt Handler node should send an Interrupt_Acknowledge-Code with the same five-bit interrupt source identifier as in the accepted Interrupt Code.

The node that sends an Interrupt-Code does not know where an Interrupt Handler in the network is. The Interrupt-Code is broadcasted to find an Interrupt Handler node. To eliminate infinite cycling of the broadcasted control code specific mechanisms for its handling in nodes and routers are provided. Each link controller of a node and each router contains one 32-bit Interrupt Source Register (ISR). When the link interface receives from its host an interrupt request with a five-bit interrupt identifier it sets appropriate bit to '1' in the 32-bit ISR. Then it sends out the Interrupt-Code with the five-bit interrupt source identifier field. If the correspondent bit in the ISR is in '1' state already then the Interrupt-Code is not sent out. A subsequent Interrupt-Code with the same interrupt source identifier can be sent by the link only after receipt of an Interrupt Acknowledge with the correspondent interrupt source identifier.

When a node, which host system can be an Interrupt Handler, receives an Interrupt-Code it checks the correspondent bit in the 32-bit ISR to be reset to '0'; in this case it

sets the ISR bit to '1' and assert its INTR_OUT output signal at its interface with the host. When the link controller in a node receives from its host an of Interrupt Acknowledge signal with a five-bit interrupt source identifier it shall reset appropriate bit in the 32-bit ISR to '0' and then sends out an Interrupt_Acknowledge-Code.

In a router, when a link interface receives an Interrupt-Code it checks the correspondent bit in the ISR. If the bit is '0' it sets the ISR bit to '1' and the input port asserts its INTR_OUT output signal at the link controller interface; it is accompanied by the five-bit interrupt source identifier of the incoming Interrupt-Code. The signal propagates to all the router output ports (except the port that have issued the signal) so that they all shall emit the Interrupt-Code with the same five-bit interrupt source identifier field, which was received by the router. But if the correspondent bit in the 32-bit ISR is equal to '1' the Interrupt-Code will be ignored (to prevent repeated Interrupt-Code propagation in networks with circular connections); the router shall not retransmit the Interrupt-Code to its output ports.

In a link an a node or in a router the Interrupt -Code and Interrupt_Acknowledge-Code control codes shall be sent out as soon as the current character or control code has been transmitted. However, the Time-Code has priority for transmission over an Interrupt_Acknowledge-Code; an Interrupt_Acknowledge-Code has priority for transmission over an Interrupt-Code.

3. INTERRUPT CODES DISTRIBUTION RECOVERY IN CASE OF ERRORS

In a SpaceWire network faults and errors may occur: link disconnect error or parity error can cause an Interrupt -Code/Interrupt_Acknowledge-Code loss; there may be spontaneous change of an ISR bit state as a result of intermittent faults in a node or in a router. The Interrupt mechanism is made to be tolerant to them. SpaceWire networks with redundant links and circular connections (e.g. mesh, torus, fat tree) are tolerant to an Interrupt -Code/Interrupt_Acknowledge-Code loss: such an error will not stop the Interrupt control codes distribution to network nodes. To ensure tolerance against multiple faults and spontaneous changes in ISR special timers are used.

Each ISR in a node or in a router has a timer per ISR bit. A timer starts at the receipt of an Interrupt-Code with correspondent five-bit interrupt source identifier and resets at receipt of an Interrupt_Acknowledge-Code with the same interrupt source identifier. In case of timeout T_i before the timer is reset, the ISR timeout event arises; the correspondent ISR bit should be reset to '0'. In the Interrupt Source link, the link also sends an Interrupt_Acknowledge-Code with the five-bit interrupt source identifier that corresponds to the ISR bit. ISR reset timeouts recover Interrupt -Codes distribution for following interrupt requests, both after Interrupt-Code and Interrupt_Acknowledge-Code losses.

4. CONTROL CODES DISTRIBUTION LATENCIES AND TIMOUTS

Efficiency of SpaceWire real-time signalling mechanisms depends on latency characteristics. They depend on several factors: link bit rates, router architecture, topology of the network interconnection, control codes flow rates. Taking some indexes for these factors we can estimate the control codes characteristics. Let T_D be worst propagation time in the network with diameter D (depends upon the SpaceWire network interconnection topology); T_{bit} – one bit transfer time; T_{wtc} – Time-code

transport through router delay (ignoring interference with previous characters/codes; depends upon implementation); T_H – delay in an Interrupt Handler node that should send an Interrupt_Acknowledge-Code (depends upon implementation).

The Time-code has highest priority among control codes and in a hop it can wait only completion of a previous character or code transmission. Its maximum time-code delivery delay, T_{Tmax} , is:

$$T_{Tmax} = (T_{wtc} + 13 T_{bit}) * (D-1) + (14 T_{bit}) * D = T_{wtc} (D-1) + T_{bit} (27 D - 13).$$

For $D=5$, $T_{wtc} = 200$ ns it will give $T_{Tmax} = 1,1 \mu s$.

Several Interrupt codes (up to 32 ones) can run concurrently in the network, and theoretically can come simultaneously to a router, thus forming a queue. They will have to let pass ahead a Time-code that can also appear at the moment. So the maximum Interrupt-code delivery delay, T_{Imax} , will be rather pessimistic:

$$T_{Imax} = (D-1) * (T_{wtc} + 13T_{bit} + 14T_{bit} + 31 * 14 * T_{bit}) + (14T_{bit}) * D =$$

$$(D-1) * (T_{wtc} + 461T_{bit}) + 14T_{bit} * D. \quad \text{For } D=5, \text{ at } 400 \text{ Mb/s it will give } T_{Imax} = 5,6 \mu s.$$

However, such situation is quite artificial. As a worst propagation time T_D it is reasonable to take more realistic estimations. Let q be an average number of waiting control codes in a router at the moment of the Interrupt code arrival. Then an estimation will be $T_D = (D-1) * (T_{wtc} + 27T_{bit} + 14 q * T_{bit}) + (14T_{bit}) * D$. For the same data, it will give more realistic picture, fig. 2; for q less then 6, T_D will be under $2\mu s$.

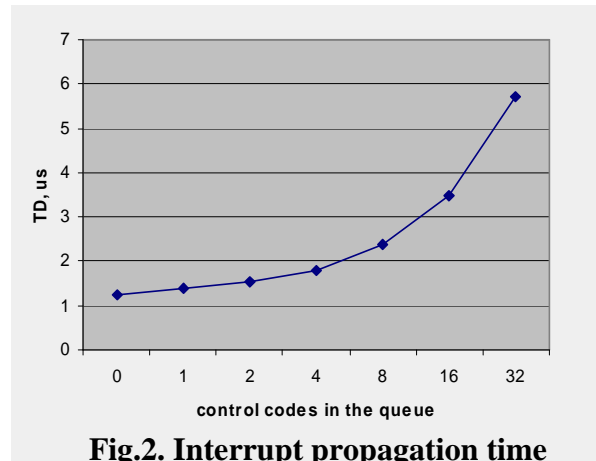


Fig.2. Interrupt propagation time as a function of control codes queue

Timeout intervals for SpW RT signals should be enough to ensure that right Interrupt/IntAck codes will reach their destination. Thus, for an Interrupt Source link the reset timeout $T_i = T_1$ shall be $T_1 > 2 * T_D + T_H$. Reset timeout for routers and non-Interrupt-Source nodes can be set $T_i = T_2$, where $T_2 \geq T_1$. Specific constraints are set on relationship between T_D and T_H :

$T_H > 2 * T_D$. It is required to prevent cycling in some particular cases with possible Interrupt-code and Interrupt_Acknowledge-code with the same Interrupt ID simultaneous distribution in networks with cycles.

5. REFERENCES

1. ECSS-E-50-12A “SpaceWire - Links, nodes, routers and networks”, European Cooperation for Space Standardization (ECSS), 2003, 124 p.
2. Sheynin Y., Solokhina T., Petrichkovitch Y., SpaceWire Technology for Parallel and Onboard Distributed Systems. “Electronika”, 2006, №5, pp.64-75, 2007, No.1, pp.38-49 (in Russian).