1 CONTROL CHARACTERS AND CONTROL CODES

a. A control character shall be formed from a parity bit, a data-control flag and a two bit control code with the data-control flag set to one to indicate that the current character is a control character.

NOTE. The different control characters and control codes are illustrated in Figure 1.

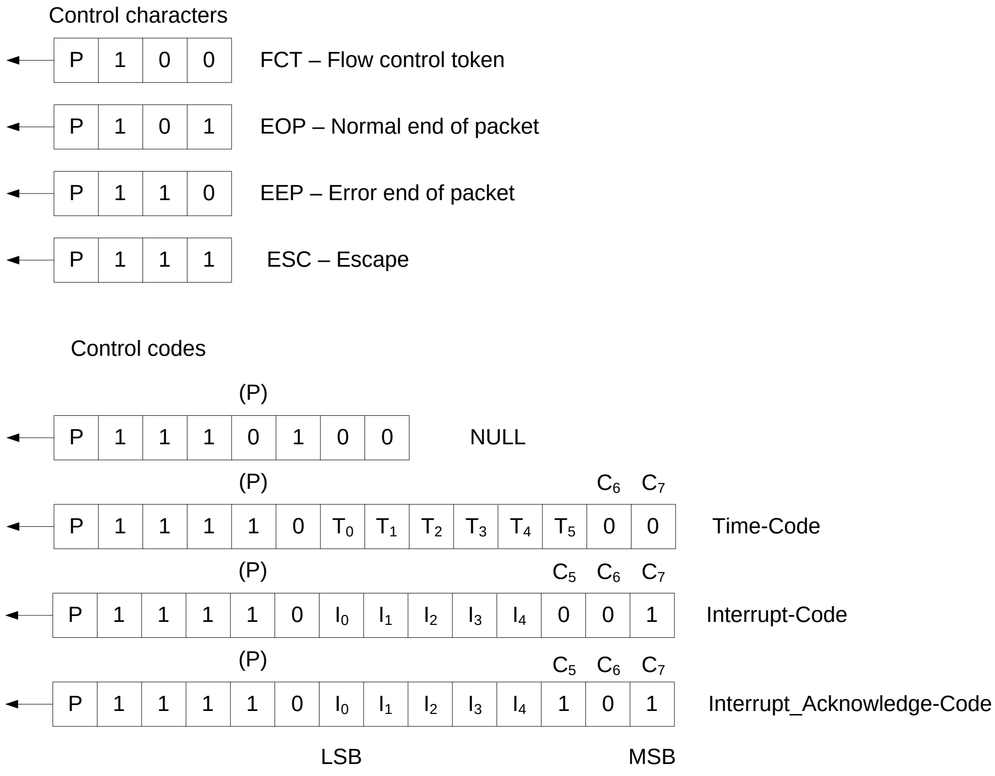

Figure 1: SpaceWire control characters and control codes

b. The NULL control code shall be formed from ESC followed by the flow control token (FCT).

NOTE 1. The parity bit (P) in the middle of the control code is zero, in accordance with clause 7.4 of the SpaceWire specification.

NOTE 2. NULL is transmitted whenever a link is not sending data or control tokens, to keep the link active and to support link disconnect detection (see clause 8).

c. The other three control codes (Time-Code, Interrupt-Code and Interrupt_Acknowledge-Code) shall be formed from ESC followed by a single data character.

NOTE 1. The parity bit (P) in the middle of these control codes is one (in accordance with clause 7.4 of the SpaceWire specification).

NOTE 2. The Time-Code is used to distribute system time information (see clause 8.12 of the SpaceWire specification) and control flags isochronous with the time-code distribution.

NOTE 3. The Interrupt-Code and Interrupt_Acknowledge-Code are used to distribute real-time interrupt signals from nodes that are sources of interrupts to nodes that can perform interrupt processing procedures (see clause 2 of this document). Interrupt-Codes and Interrupt_Acknowledge-codes can eliminate system-wide sideband signals for low latency control signals distribution.

d. Six bits of time information shall be held in the least significant six bits of the Time-Code (T_0-T_5) and the two most significant bits (C_6=0, C_7=0) shall contain control flags that are distributed isochronously with the Time-Code.

e. Five bits of interrupt information shall be held in the least significant five bits of the Interrupt-Code (I_0-I_4) and the three most significant bits (C_5=0, C_6=0, C_7=1) shall contain control flags that are distributed isochronously with the Interrupt-Code.

NOTE. The Interrupt-Code is used to distribute interrupt request information and control flags isochronous with the Interrupt-Code distribution.

f. Five bits of interrupt acknowledge information shall be held in the least significant five bits of the Interrupt_Acknowledge-Code (I_0-I_4) and the three most significant bits (C_5=1, C_6=0, C_7=1) shall contain control flags that are distributed isochronously with the Interrupt_Acknowledge-Code.

NOTE. The Interrupt_Acknowledge-Code is used to distribute interrupt acknowledge information and control flags isochronous with the Interrupt_Acknowledge-Code distribution.

g. An escape character (ESC) followed by ESC, EOP or EEP is an invalid sequence and shall be noted as an escape error (see clause 8.9.2.3 of the SpaceWire specification).

1.1 Distributed Interrupts Interface (informative)

a. The distributed interrupts interface to the host system should comprise four signals, INTR_IN, INTR_OUT, INTR_ACK_IN and INTR_ACK_OUT, a five-bit interrupt input port, a five-bit interrupt output port, a three-bit control flag input port and a three-bit control flag output port.

b. When INTR_IN is asserted and the link interface is in the Run state it should cause the transmitter sending an Interrupt-Code.

c. INTR_OUT should be asserted whenever the link interface is in the Run state and the receiver receives a valid Interrupt-Code.
d. When INTR_ACK_IN is asserted and the link interface is in the Run state it should cause the transmitter sending an Interrupt_Acknowledge-Code.
e. INTR_ACK_OUT should be asserted whenever the link interface is in the Run state and the receiver receives a valid Interrupt_Acknowledge-Code.
f. A five-bit interrupt output port should be provided for passing Interrupt Identifier from the link receiver to the host system.
g. A five-bit interrupt input port should be provided for passing Interrupt Identifier from the host system to the link transmitter.
h. A three-bit input and output control flag port should be provided for passing control flag information. The Interrupt-Code control flags are C_5=0, C_6=0, C_7=1, and the Interrupt_Acknowledge-Code control flags are C_5=1, C_6=0, C_7=1.

2 INTERRUPTS DISTRIBUTION

2.1 Introduction

a. The Distributed Interrupts mechanism may be supported by SpaceWire nodes and routers. Rules and recommendations defined by this section are assumed to be applicable to only those SpaceWire nodes and routers, which are aimed to support the Distributed Interrupts mechanism.

b. The Distributed Interrupts mechanism provides a low-latency signaling among nodes in SpaceWire networks.

c. Interrupt distribution is not affected by data packets transmitted simultaneously with an Interrupt-Code over the same links. Interrupt distribution can be performed even over links that are blocked by data packets.

2.2 General

a. Interrupt-Code represents an interrupt request. It may be issued by a node which is assigned to be an Interrupt Source for this type of interrupts.

b. Each Interrupt-Code shall be identified by an Interrupt Identifier. In a network there may be up to 32 Interrupt Identifiers ranging from 0 to 31.

c. Depending on the operation mode defined for an Interrupt Identifier, there may be one or more Interrupt Sources of Interrupt-Codes with this Interrupt Identifier.

d. An Interrupt-Code should be accepted for handling in some node of the SpaceWire network. The node is called an Interrupt Handler. The host system of the Interrupt Handler is supposed to implement some interrupt processing routine for the accepted Interrupt-Code.

e. In a network there may be one or more Interrupt Handlers for Interrupt-Codes with a particular Interrupt Identifier.

f. Interrupt_Acknowledge-Code represents a confirmation that the Interrupt-Code has reached a correspondent Interrupt Handler and has been accepted for processing.

> NOTE. Issuing Interrupt_Acknowledge-Code does not mean that the execution of the correspondent interrupt processing routine has already been completed or even started.

g. The distribution of Interrupt-Codes with a particular Interrupt Identifier has two operation modes: the Acknowledgement mode and the Un-acknowledgement mode. Both modes may be operated concurrently in a SpaceWire network.

NOTE. This means that there may be Interrupt-Codes which are distributed in the Acknowledgement mode and Interrupt-Codes which are distributed in the Un-acknowledgement mode.

h. In the Acknowledgement mode, defined in clause 2.3, any Interrupt Handler which accepts an Interrupt-Code for processing, shall send the Interrupt_Acknowledge-Code as it is defined in clause 2.3.3. The Acknowledgement mode may be supported by implementations of the Distributed Interrupts mechanism.

i. In the Un-acknowledgement mode, defined in clause 2.4, any Interrupt Handler which accepts for processing an Interrupt-Code, shall not send the Interrupt_Acknowledge-Code. The Acknowledgement mode may be supported by implementations of the Distributed Interrupts mechanism.

j. The order of priority for transmission of characters in a link shall be as follows:

   1. Time-Code, highest priority;
   2. Interrupt_Acknowledge-Code;
   3. Interrupt-Code;
   4. FCT;
   5. N-Char;
   6. NULL, lowest priority.

k. If a node or a router supports the Distributed Interrupts mechanism, each link interface in this node or router shall manage Interrupt-Codes. If a node or a router supports the Acknowledgement mode of the mechanism, each link interface in this node or router shall manage Interrupt_Acknowledge-Codes.

l. If a node or a router supports only the Un-acknowledgement mode, a received Interrupt_Acknowledge-Code should be ignored and deleted.

m. If a node or a router does not support the Distributed Interrupts mechanism, a received Interrupt-Code or Interrupt_Acknowledge-Code should be ignored and deleted.

n. A request for the transmission of an Interrupt-Code or Interrupt_Acknowledge-Code shall be ignored, if the link interface is not in the Run state.

o. An Interrupt-Code or Interrupt_Acknowledge-Code received when the link interface is not in the Run state shall be interpreted as an error.

p. A 32-bit Interrupt Source Register (ISR) shall be used to prevent repeated Interrupt-Code and Interrupt_Acknowledge-Code propagation in networks with circular connections. Each *i-th* bit in the ISR corresponds to the Interrupt Identifier with the same number.

q. Each router shall contain one single ISR for all link interfaces.

> NOTE. Several Interrupt-Codes with the same Interrupt Identifiers can be received by different ports of a router at the same time. In order to prevent an erroneous situation, a router should check and set or reset ISR bits through atomic operations.

r. Each node shall contain one or more ISRs.

s. One single ISR in a node may be assigned for several link interfaces. If a node contains more than one ISR, the host system in the node should manage the reception of redundant Interrupt-Codes and Interrupt_Acknowledge-Codes.

> NOTE. While a node contains more than one ISR, it is not able to prevent the reception of redundant Interrupt-Codes and Interrupt_Acknowledge-Codes transmitted through redundant links and received at different ports for which different ISRs are assigned.

t. After reset of a node or a router the ISR bits in this node or router shall be set to zero.

u. A link interface in a node or a router may be assigned whether it is permitted to send or to receive Interrupt-Codes. Similarly, a link interface in a node or a router may be assigned whether it is permitted to send or to receive Interrupt_Acknowledge-Codes.

v. If a link interface in a node or in a router is not permitted to send Interrupt-Codes or Interrupt_Acknowledge-Codes, it shall ignore requests from the host system for the transmission of the codes.

w. If a link interface in a node or in a router is not permitted to receive Interrupt-Codes or Interrupt_Acknowledge-Codes, it shall ignore receptions of the codes from the link.

2.3 Acknowledgement mode

2.3.1 General

a. In the Acknowledgement Mode in a network there shall be not more than one Interrupt Source for Interrupt-Codes with a particular Interrupt Identifier.

b. In the Acknowledgement Mode an Interrupt Source gets a confirmation that the issued Interrupt-Code has reached an Interrupt Handler by the reception of the correspondent Interrupt_Acknowledge-Code.

c. Clearing the network after the propagation of an Interrupt-Code and allowing the distribution of the next Interrupt-Code with the same Interrupt Identifier may be implemented in two ways: Interrupt_Acknowledge-Codes are the primary way and ISR reset timers are the secondary way, which is used in case of error.

2.3.2 Interrupt-Code distribution

a. When a link interface of a node receives a request for the transmission of an Interrupt-Code accompanied by an Interrupt Identifier, the following actions shall be performed:

   1. Check the correspondent bit in the 32-bit ISR.
   2. If the bit is ‘0’, it shall be set it to '1'. Then the link interface shall send out an Interrupt-Code with the Interrupt Identifier and the other three control bits set to the C_5=0, C_6=0, C_7=1 values.
   3. If the bit is ‘1’, the request shall be ignored.

      NOTE. This means that the Interrupt-Code with that Interrupt Identifier has been sent to the network but the correspondent Interrupt_Acknowledge-Code has not been received yet.

b. A subsequent Interrupt-Code with the same Interrupt Identifier shall not be sent out until one of the following events occur:

   1. In case of the reception of an Interrupt_Acknowledge-Code with the same Interrupt Identifier, the event shall be the expiration of the correspondent t_G minimum time interval.
   2. In case of a failure, the event shall be the completion of the recovery procedure (see clause 2.3.4).

c. The t_G minimum time interval is a minimum time interval between the reception of an Interrupt_Acknowledge-Code and the transmission of the subsequent Interrupt-Code with the same Interrupt Identifier. The length of the t_G minimum time interval shall be set in such a way as to ensure that the propagation the next Interrupt-Code over the network will not be corrupted by the propagation of the previous Interrupt_Acknowledge-Code.

   NOTE 1. If the propagation of an Interrupt-Code was corrupted by the propagation of the previous Interrupt_Acknowledge-Code, the correct operation of the mechanism would not be guaranteed. In this case a number of errors would be able to occur, e.g. endless propagation of a code over a network with cycles.

   NOTE 2. The length of the t_G minimum time interval depends on the wide range of parameters as followings: network topology, number of Interrupt Handlers, distance between Interrupt Source and Interrupt Handler, etc. Thus, methods of calculation of the interval length may vary from network to network and, therefore, are not specified in this document.

d. When a link interface in a router receives an Interrupt-Code the following actions shall be performed:
   1. Check the correspondent bit in the 32-bit ISR.
   2. If the bit is '0', it shall be set to '1'. Then the request for the Interrupt-Code transmission shall be propagated to all the output ports of the router (except the port that has received this Interrupt-Code) so that they all shall emit the Interrupt-Code with the same Interrupt Identifier as was received by the router.
   3. If the bit is '1', the received Interrupt-Code shall be ignored.

      NOTE. In this case the Interrupt-Code shall be ignored so as to prevent repeated Interrupt-Code propagation in networks with circular connections.

e. When a link interface in a node receives an Interrupt-Code it shall perform the following actions:
   1. Check the correspondent bit in the 32-bit ISR.
   2. If the bit is '0', it shall be set to '1'. Then the link interface may transmit the received Interrupt-Code to the host system.
   3. If the bit is '1', the Interrupt-Code shall be ignored.

      NOTE. Such case can occur due to a redundant transmission, broadcasting of the Interrupt-Code or a malfunction in the network.

2.3.3 Interrupt_Acknowledge-Code distribution

a. When an Interrupt Handler receives the Interrupt-Code, which this Interrupt Handler is responsible to process, it shall acknowledge the reception by issuing an Interrupt_Acknowledge-Code with the same Interrupt Identifier as in the received Interrupt-Code.
b. An Interrupt_Acknowledge-Code shall be sent out upon the expiration of the correspondent t_H time interval. The t_H time interval is a time interval between the reception of an Interrupt-Code and the transmission of the correspondent Interrupt_Acknowledge-Code.
c. The length of the t_H minimum time interval shall be set in such a way as to ensure that the Interrupt Handler will not receive the same Interrupt-Code after the correspondent Interrupt_Acknowledge-Code is sent out.

   NOTE 1. This means that an Interrupt_Acknowledge-Code will not be sent out until the acknowledged Interrupt-Code has propagated over the part of the network from where it can reach the Interrupt Handler again.

NOTE 2. The length of the t_H minimum time interval depends heavily on the network topology. Thus, methods of calculation of the interval length may vary from network to network and, therefore, are not specified in this document.

d. When a link interface of a node receives a request for the transmission of an Interrupt_Acknowledge-Code accompanied by an Interrupt Identifier, the following actions shall be performed:
   1. Check the correspondent bit in the 32-bit ISR.
   2. If the bit is '1', it shall be reset to '0'. Then the link interface shall send out an Interrupt_Acknowledge-Code with the Interrupt Identifier and the other three control bits shall be set to the values of the control flags C_5=1, C_6=0, C_7=1.
   3. If the bit is '0', the request shall be ignored.

e. When a link interface in a router receives an Interrupt_Acknowledge-Code, the following actions shall be performed:
   1. Check the correspondent bit in the 32-bit ISR.
   2. If the bit is '1', it shall be reset to '0'. Then the request for the Interrupt_Acknowledge-Code transmission shall be propagated to all the output ports of the router (except the port that has received this Interrupt_Acknowledge-Code) so that they all emit the Interrupt_Acknowledge-Code with the same Interrupt Identifier as was received by the router.
   3. If the bit is '0', the Interrupt_Acknowledge-Code shall be ignored.

NOTE. It prevents repeated Interrupt_Acknowledge-Code propagation in a network with circular connections.

f. When a link interface in a node receives an Interrupt_Acknowledge-Code, the following actions shall be performed:
   1. Check the correspondent bit in the 32-bit ISR.
   2. If the bit is '1', it shall be reset to '0'. Then the link interface may transmit the received Interrupt_Acknowledge-Code to the host system.
   3. If the bit is ‘0’, the Interrupt_Acknowledge-Code shall be ignored.

2.3.4 Interrupt Codes distribution recovery in case of errors

a. Causes that can lead to an Interrupt-Code or Interrupt_Acknowledge-Code distribution malfunction are:
   - A link disconnect error or a parity error occurred during the transmission of an Interrupt-Code or an Interrupt_Acknowledge-Code over a link;

- A spontaneous change of an ISR bit state as a result of intermittent faults in a node or in a router.

  NOTE. In SpaceWire networks with redundant links and circular connections (e.g. mesh, torus, fat tree) an error that causes a loss of an Interrupt-Code or an Interrupt_Acknowledge-Code will not stop the distribution.

b. Each ISR in a node or in a router shall have a reset timer per each ISR bit.

c. A reset timer shall be started in an Interrupt Source at the transmission of an Interrupt-Code with the correspondent Interrupt Identifier.

d. A reset timer shall be started in a router or a node at the receipt of an Interrupt-Code with the correspondent Interrupt Identifier.

e. A reset timer shall be stopped and reset in an Interrupt Handler at the transmission of an Interrupt_Acknowledge-Code with the correspondent Interrupt Identifier.

f. A reset timer shall be stopped and reset in a router or a node at the receipt of an Interrupt_Acknowledge-Code with the correspondent Interrupt Identifier.

g. If a reset timer expires before the correspondent Interrupt_Acknowledge-Code is received, the ISR timeout event arises and the correspondent bit in the ISR shall be reset to '0'.

  NOTE 1. An ISR bit reset timeouts recover the network for the distribution of subsequent interrupt requests.

  NOTE 2. An ISR bit reset timeouts recover Interrupt-Codes distribution after both Interrupt-Code and Interrupt_Acknowledge-Code losses.

h. The values of ISR reset timeouts in nodes ($T_{ISRreset\_N}$) and routers ($T_{ISRreset\_R}$) shall be not less than the worst propagation time of the Interrupt-Code in the network plus the maximum delay in an Interrupt Handler that should send an Interrupt_Acknowledge-Code plus the worst propagation time of the Interrupt_Acknowledge-Code in the network ($T_{ISRreset}$).

i. The timeout values in nodes and routers shall be in the following relation:

$$T_{ISRreset} \leq T_{ISRreset\_R} \leq T_{ISRreset\_N},$$

2.4 Un-acknowledgement mode

2.4.1 General

a. The Un-acknowledgement mode provides Interrupt-Codes distribution without acknowledgment.

b. In the Un-acknowledgement mode an Interrupt Source cannot gets a confirmation that the issued Interrupt-Code has reached an Interrupt Handler.
c. In the Un-acknowledgement mode ISR timers are the only way to clear the network after the propagation of an Interrupt-Code and allow the distribution of the next Interrupt-Code with the same Interrupt Identifier.
d. In the Un-acknowledgement mode in a network there may be more than one Interrupt Source for Interrupt-Codes with a particular Interrupt Identifier.

2.4.2 Interrupt-Code distribution

a. In the Un-acknowledgement mode Interrupt-Codes shall be distributed as defined in clause 2.3.2, points a, d and e.
b. An ISR in a router shall have a reset timer per each ISR bit.
c. An ISR in a node may have a reset timer per each ISR bit. If a node is an Interrupt Source and does not implement a reset timer for the correspondent ISR bit, the bit shall not be set to '1' while issuing the Interrupt-Codes. If a node is an Interrupt Handler, it shall implement reset timer for the correspondent ISR bit.

NOTE. The case when a node which is neither Interrupt Source nor Interrupt Handler does not implement reset timers is implementation dependent. Such node can either set the ISR bit to '1' once or never set the bit to '1'.

d. A subsequent Interrupt-Code with the same Interrupt Identifier should not be sent out until the expiration of the correspondent ISR reset timer.

NOTE. If an Interrupt-Code is sent before the expiration of the correspondent reset timer, this Interrupt-Code will not be propagated through the network.

e. When a reset timer expires in a node or a router, the correspondent bit in the ISR shall be reset to '0'.
f. The values of ISR reset timeouts in nodes ($T_{ISRreset\_N}$) and routers ($T_{ISRreset\_R}$) shall be more than the propagation time of the correspondent Interrupt-Code via the longest cycle in the network ($T_{MaxCycleProp}$). The timeout values in nodes shall be more than the timeout values in routers. As a result, the timeouts value shall be in the following relation:

$$T_{MaxCycleProp} < T_{ISRreset\_R} < T_{ISRreset\_N} .$$

2.5 Error protection and recovery

2.5.1 Protection from babbling idiots

a. Babbling idiot node is a node sending Interrupt-Codes or Interrupt_Acknowledge-Codes which it is not permitted to send and busy the network with it unduly.

b. Protection from babbling idiot nodes may be implemented both in nodes and routers.

c. A link interface in a node may be assigned that it is permitted to send Interrupt-Codes with particular Interrupt Identifiers only. Similarly, a link interface in a node may be assigned that it is permitted to send Interrupt_Acknowledge-Codes with particular Interrupt Identifiers only.

d. If a link interface in a node is permitted to send Interrupt-Codes or Interrupt_Acknowledge-Codes with particular Interrupt Identifiers only, it shall ignore requests for the transmission of the correspondent codes with other Interrupt Identifiers.

e. A protection from babbling idiot nodes may be provided through dedicated edge routers.

f. For a router it may be specified which Interrupt-Codes and Interrupt_Acknowledge-Codes are permitted to be received by a particular link interface of the router.

g. If the received Interrupt-Code or Interrupt_Acknowledge-Code is not permitted to be received by this link interface, it shall be ignored.

2.5.2 Protection from an unexpected code occurrence

a. Causes that can lead to an unexpected code occurrence are:
   - Wrong settings of the t_H and t_G time interval values;
   - Several Interrupt Sources for Interrupt-Codes with a particular Interrupt Identifier being distributed in the Acknowledgement mode;
   - Network malfunction which caused the occurrence of a false Interrupt-Code or Interrupt_Acknowledge-Code (e.g by bits inversion in a link due to noise);
   - A babbling idiot node if no special policies are used (e.g. policies defined in clause 2.5.1).

b. An unexpected Interrupt-Code or Interrupt_Acknowledge-Code occurrence can lead to the following situations:
   - The Interrupt Handler can receive a false Interrupt-Code, i.e. the code that was not generated by the Interrupt Source;
   - An infinite looping can occur in networks with circular connections.

c. Each ISR in a node or in a router may have a timer per each ISR bit for the protection from an unexpected code occurrence. Such timer should be set to the $T_{ISRchange}$ timeout value defining the minimum allowed time between two consecutive ISR bit changes. This means that the ISR bit value may not be changed before the $T_{ISRchange}$ timer expired.

d. In case of an attempt to change the state of a bit of ISR while the correspondent $T_{ISRchange}$ timer has not expired yet, the bit value should not be changed and the received code should be ignored.

2.6 Implementation issues

a. The mechanism of interrupts distribution, defined by this section, raises several implementation issues which are considered in this clause.

b. In an Interrupt Handler the generation of Interrupt_Acknowledge-Codes can be performed at either hardware or software level. In order to ensure correct operation of the mechanism, the generation should be performed at hardware level.

NOTE. The generation of Interrupt_Acknowledge-Codes at software level could lead to a number of problems. Firstly, the generation time of Interrupt_Acknowledge-Code may be less predictable than in the case of hardware implementation. This, in turn, makes it difficult to calculate the t_H time interval and ISR reset timeout value.

The second reason is that the generation time of Interrupt_Acknowledge-Code in software significantly exceeds the same period for hardware generation. This subsequently decreases the performance of the mechanism.

c. In a router distributed through several output ports Interrupt-Codes and Interrupt_Acknowledge-Codes should be sent out in the same order for all the ports.

NOTE. This issue ensures more predictable in time operation of the mechanism.

d. In case of the operation in the Acknowledgement mode the $T_{ISRreset\_N}$ and $T_{ISRreset\_R}$ timeouts are intended to be used for error recovery only. For this reason it is recommended that the timeout values significantly exceed the minimum interval defined as $T_{ISRreset}$.

NOTE. The calculation of the worst propagation time typically does not cover all the delays. Thus, the timers should expire only when it is certainly true that the correspondent Interrupt-Code or Interrupt_Acknowledge-Code has been lost.