Next release of the SpaceWire standard - some requests for change

Yuriy Sheynin

St. Petersburg State University of Aerospace Instrumentation
St.Petersburg, RUSSIA
sheynin@aanet.ru

General

Limits for the SpaceWire standard evolution.

- What to do and what not to do in the SpaceWire standard evolution.

SpaceWire outside Space applications.

- Support in the standard

Standardisation & diversity of implementations

- Standard to describe the technology
- Standardisation for compatibility
- Not binding for implementations difference
- Clearly and explicitly divide normative and informative parts of the standard document

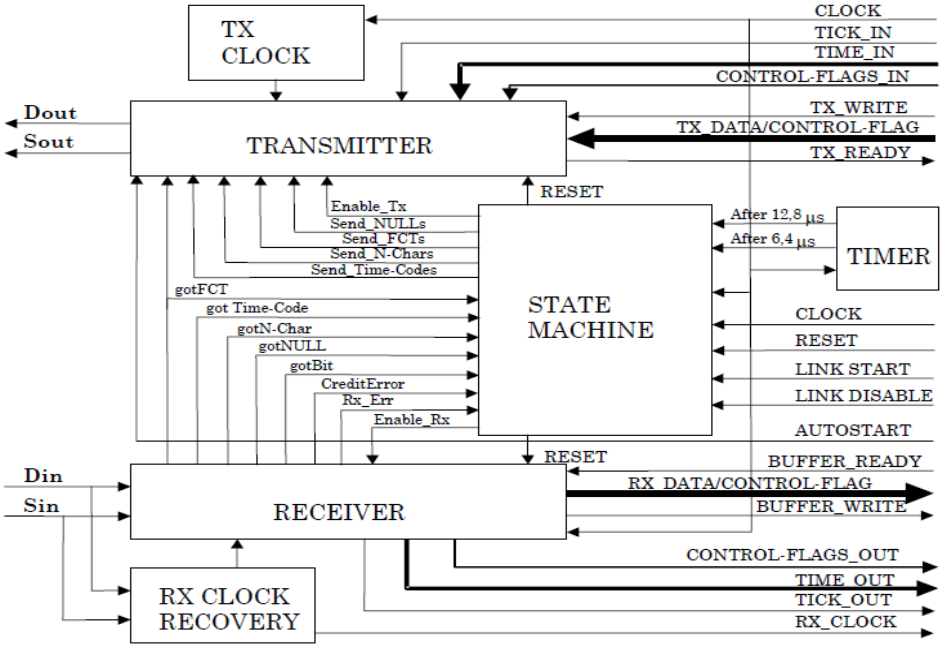


Figure 8-1: Example SpaceWire link interface block diagram

- (e.g. “Figure 8□1: Example SpaceWire link interface block diagram”, what host interface signals are mandatory, normative part, and what are informative or illustrative?)

Physical level/Signal level

- PHY level
  - Variety of cables
  - Variety of distances
  - Cable assembly with cable/cable connectors inside
- Signal level
  - Higher bit rates for limited distances
  - Longer distances with lower upper rates limits
  - Lower link start rate ?
  - Higher minimal link rate.

    Set 10 Mb/s both as the start and the minimum link rate (for the current PHY/Signal link)

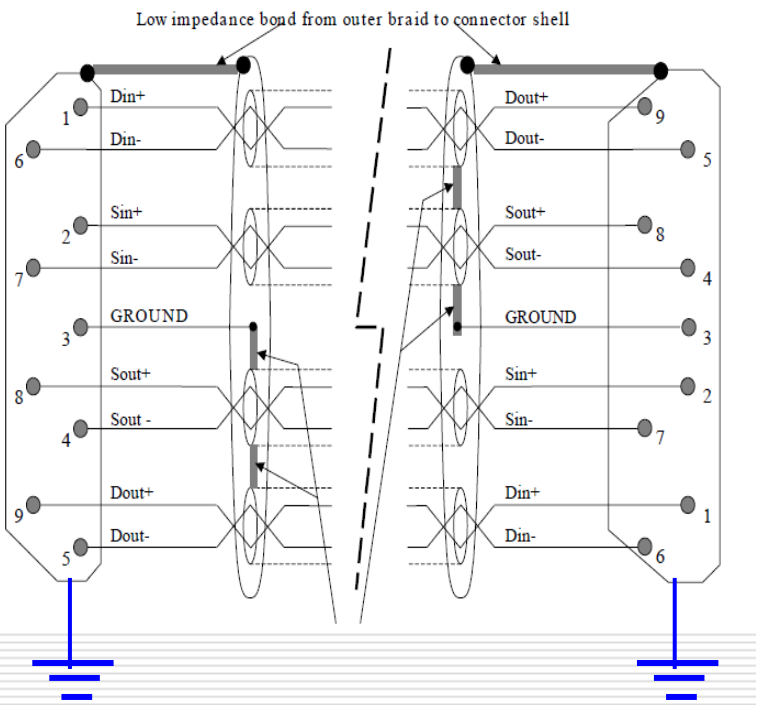
("The minimum data signalling rate is set by the disconnect timeout (clause 8.9.2.1 and 8.11.2) to greater than 1,18 Mb/s, i.e. 1/850 ns ." [p.46])

Physical level/Signal level

- Adaptable link rate
  - duplex link rate matching procedure by negotiation and/or by sequence of attempts

Conductivity isolation problems with SpaceWire links

- How to use a SpaceWire link to connect galvanic isolated blocks;
  - outer shield grounding?
- How to ensure true conductivity isolation for a SpaceWire link;
  - versions of PHY/Signal/Symbol levels for galvanic isolated links



Character level

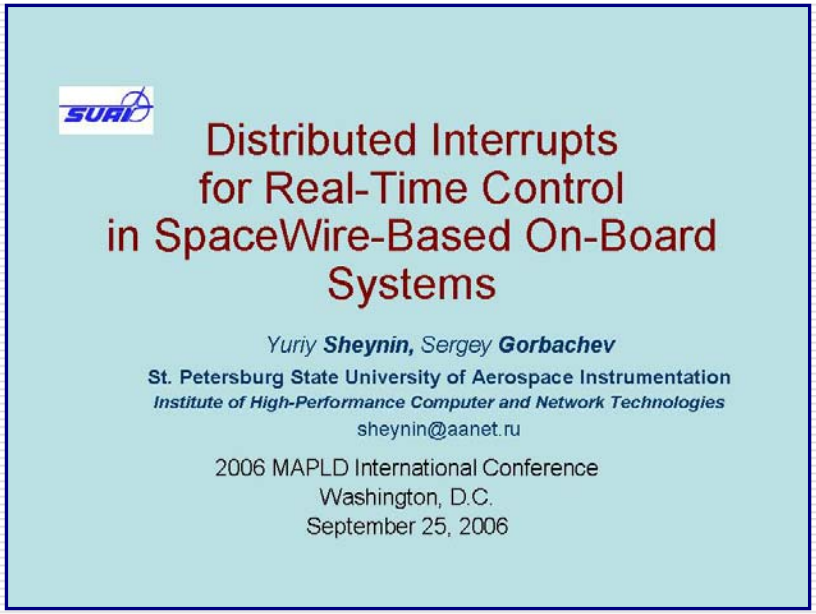
- Additional control codes for Distributed Interrupts

Sideband signals in SpaceWire interconnections

- The time codes distribution is a SpaceWire feature to substitute sideband signals for distributed systems clock synchronization.
  It is a clear advantage of the SpaceWire over other interconnection standards
- Other signals, besides the time codes, are needed to be distributed in SpaceWire also
  (interrupts, hard RT event signals, Linux RT-signals, etc.)
- To have them at the low protocol layer, in order to get
  - high priority distribution,
  - minimum latencies,
  - to traverse blocked by data links, etc.

Updates in the Distributed Interrupts proposal

- Minor updates in the Distributed Interrupt basics:
  - **One** “spare” 2-bits Control Flags combination used instead of two code combinations
  - **5**-bit coding of Interrupt identifiers instead of 6-bit (**32** interrupts instead of 64)
- Error recovery procedures specification extended

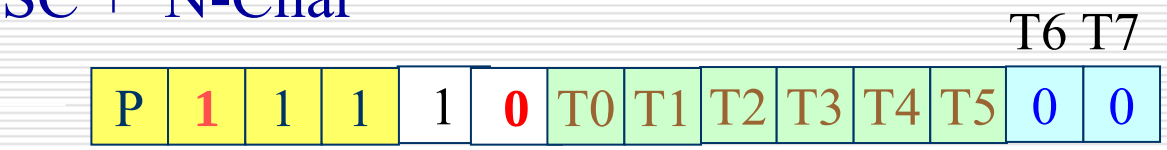


Distributed Interrupts coding

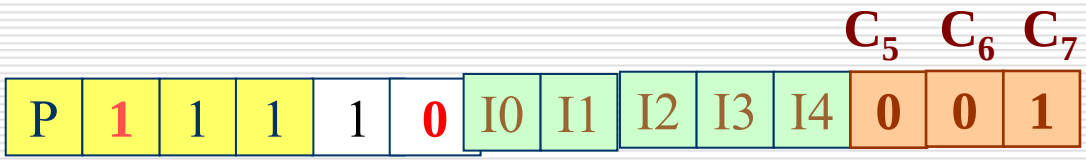
a) ESC+L-char

NULL=ESC+FCT

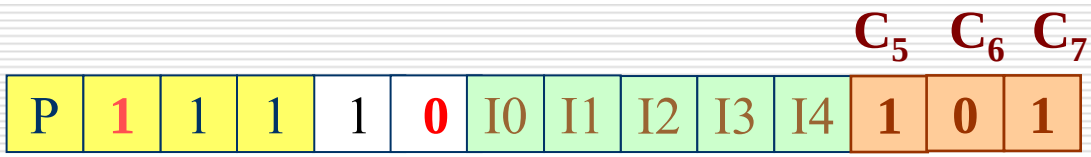
b) ESC + N-Char



Time-code



Interrupt-code

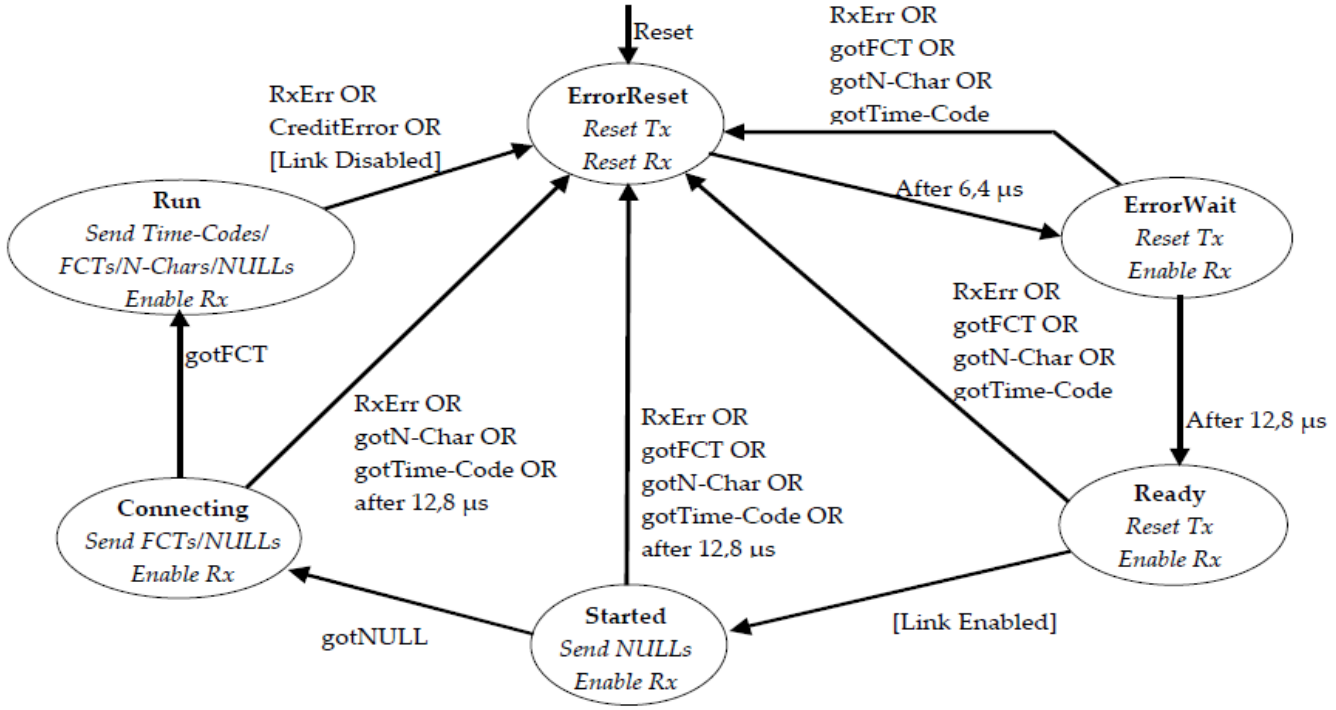


Interrupt_Acknowledge code

Exchange level

- Error corrections in the current state machine specification
- Timeouts
- State machine modification for the Simplex mode of link operation

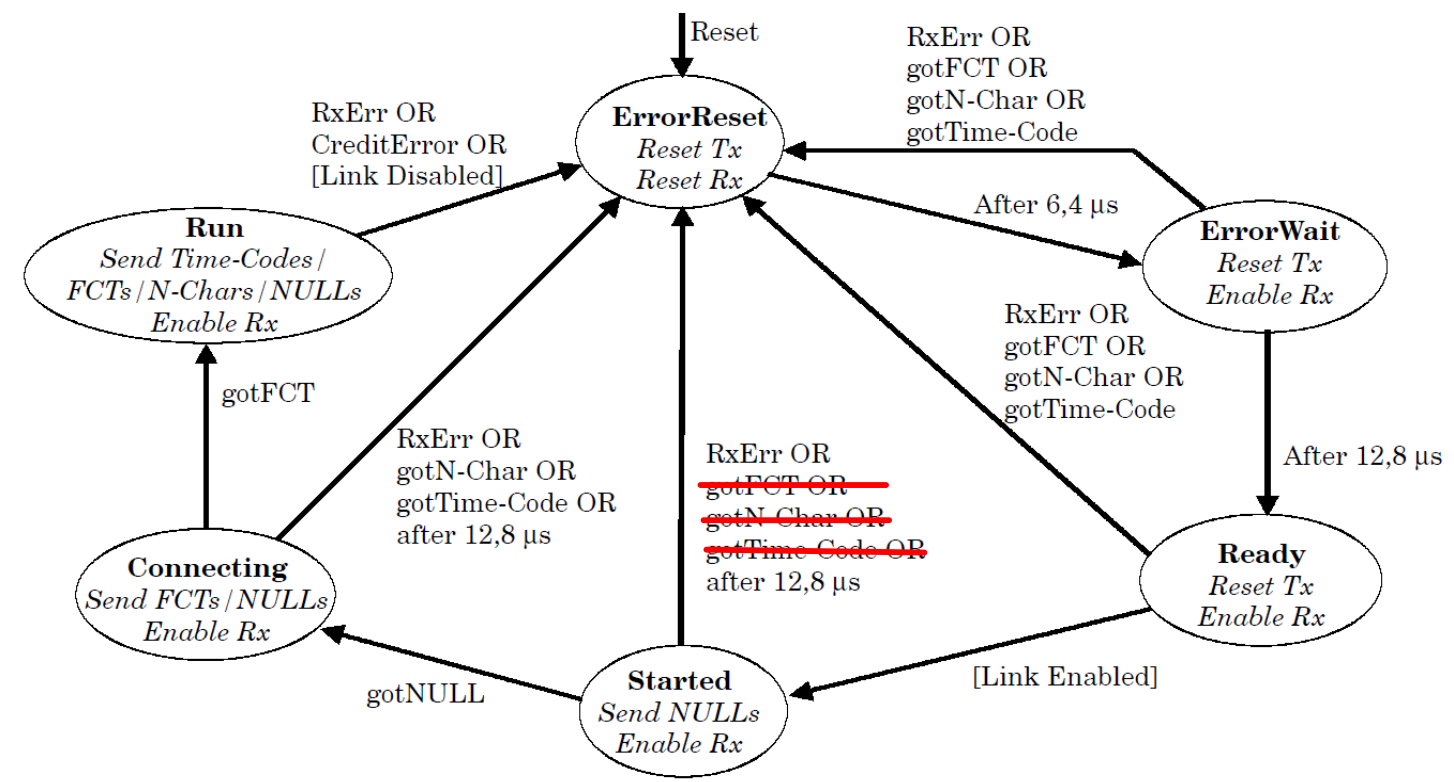
“Started” state and got characters/codes



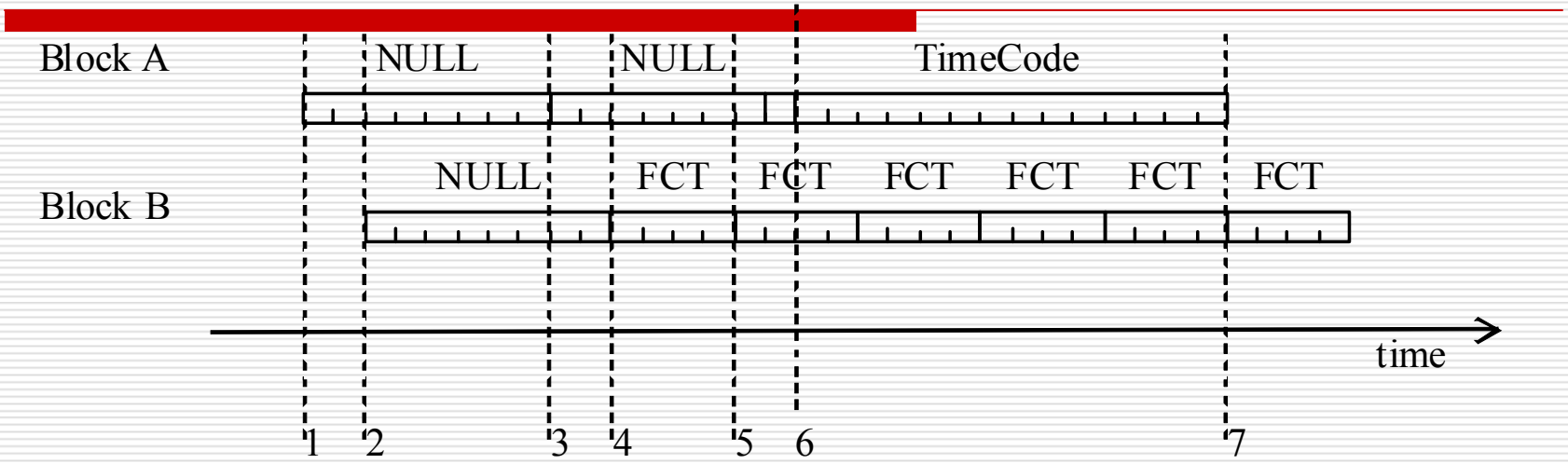
RxErr = Disconnect error OR Parity error OR Escape error (ESC followed by EOP or EEP or ESC).

NOTE Disconnect error only enabled after First Bit Received. Parity Error, Escape Error, gotFCT, gotN-Char, gotTime-Code only enabled after First NULL Received (i.e. gotNULL asserted). Thus RxErr OR gotFCT OR gotN-Char OR gotTime-Code is really RxErr OR (gotNULL AND (gotFCT OR gotN-Char OR gotTime-Code)).

In *Started* ignore got characters until gotNULL

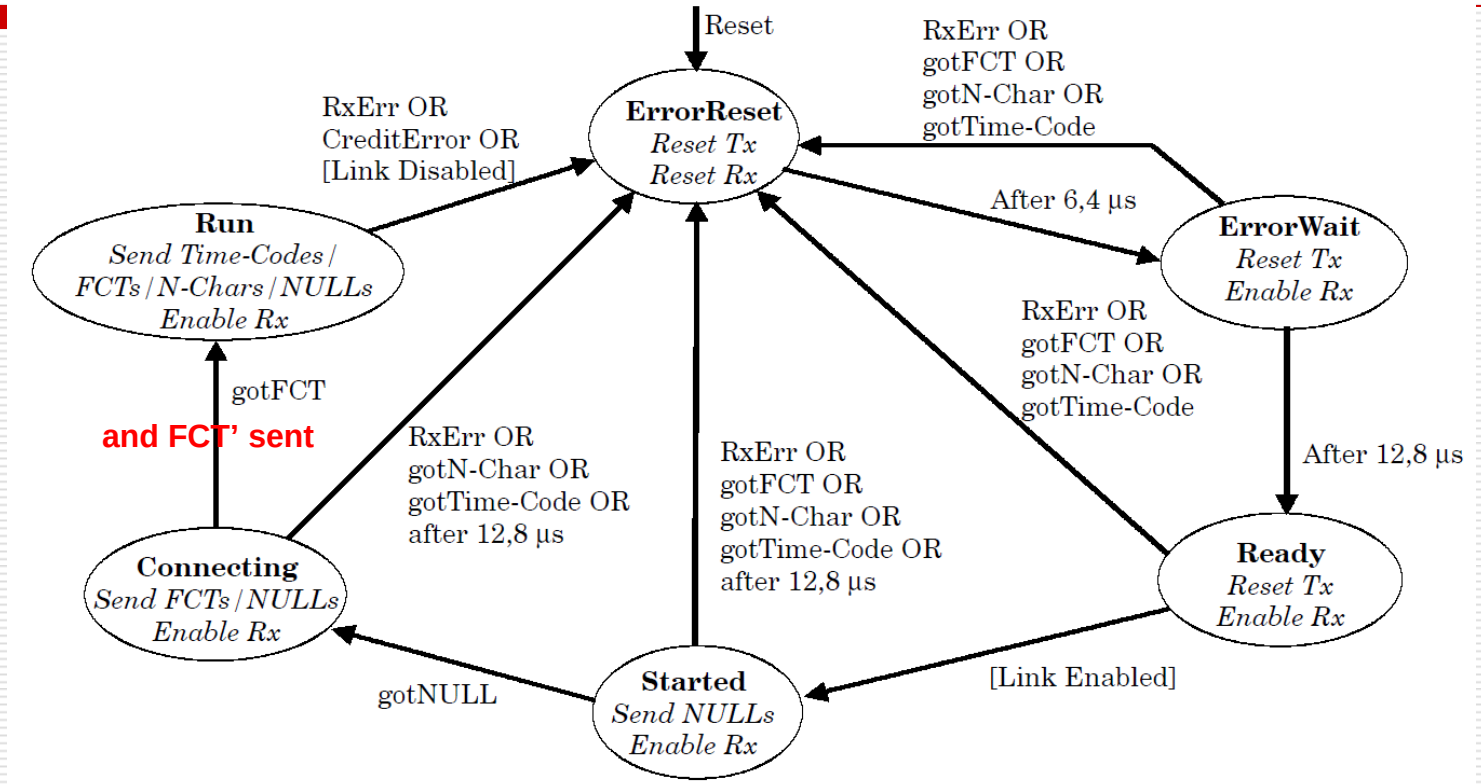


State Machine error



1) Block A is in *Started*, sending NULL symbols.
2) Block B is in *Started* , sending NULL symbols.
3) Block A received NULL-symbol, going to *Connecting* state, sending **NULL symbols**.
4) Block B received NULL-symbol, going to *Connecting* state, sending **FCT symbols**.
5) Block A received FCT-symbol, going to *Run* State.
6) Block A in *Run* state, sending **TimeCode symbol**.
7) Block B received **TimeCode** in **Connecting** state, going to ***Reset*** State

Resolving problem



FCT'sent shows that SpW block *<u>has</u>* sent FCT symbol by himself.
Thus, the state machine goes to *Run* state only after FCT symbols has been **sent and received.**

Exchange timeout periods

8.11.3 Exchange timeout periods (page 76)

a. The 6,4 µs (nominal) timeout period shall be from 5,82 µs (i.e. 64 cycles of 10 MHz + 10 % clock) to 7,22 µs (i.e. 65 cycles of 10 MHz -- 10 % clock).
b. The 12,8 µs (nominal) timeout period shall be from 11,64 µs (i.e. 128 cycles of 10 MHz + 10 % clock) to 14,33 µs (i.e. 129 cycles of 10 MHz -- 10 % clock).

For example.

Side A detects channel's error. Side A stop tx.

After 1 µs side B detects disconnection error. Side B stop tx.

Both sides stop tx. Why the sides A,B can't start connection right now?

Sides A,B needs time for reinitialize theirs hardware. Why 64/128 cycles?

We suggest to *reduce* these timeouts;
or make them link operation parameters,
between 2 and 6,4 us and 4 and 12,8 us, respectively

Packet/Network level

Packet level

- Delete as a separate level
- Include its information into the Network level specification

Network level

- Broadcast/multicast modes in SpaceWire interconnections
- SpaceWire nodes with multiple links
  - (Not a router!)
  - Nodes with embedded packet routing (routers)
- Configuration space specification (basic) for a router and for a node (with regard to SpaceWire links)

Multi-level issues.
Reset specification

The current standard doesn't specify Reset as a systematic procedure that covers operation of several, if not all, protocol levels.
We can only find one clause (8.5.3.1) on it and a pour of separate words and sentences with "reset".

8.5.3.1 Reset.

Reset represents power on reset, other hardware reset or software commanded reset.

Reset specification

- How a reset should be done in respect to different protocol layers?
- What levels should be involved in a Reset?
- What levels should reset only together, and what levels could be reset individually also?
- For instance, the basic FSM is specified at the Exchange level. How should operate the Character level ?
- How should the data transmission stop on Reset or Link Disable?
  Should the transmitter immediately stop the current character transmission? Or finish the transmission before stopping transmission and putting DS to zero?
- What are Reset RX and Reset TX in the FSM? What actions correspond to them?

Link Disabled condition

The statements from **ECSS-E-50-12C**:

- **8.5.3.8 (p.64)** [Link Disabled] is a condition set by external hardware or software in order to disable and stop the link interface.
- **8.5.2.7 (p.63) Run** b). If the link interface is disabled, or if a disconnect error, parity error, escape error or credit error is detected, while in the *Run* state, then the state machine shall move to the *ErrorReset* state.

The problem:

- The current received packet is not ended with EEP while state machine moved from RUN to *ErrorReset* state due to [Link Disabled] condition.

The decision:

- To add **11.4 (p.98)** with information about insertion of EEP (in receiving packet) and delete data in the transmitter buffer when [Link Disabled] condition occurs.
- For Example:
- Rename chapter : 11.4 Link error/**Link disabled** recovery
- Rewrite : 11.4 a) Detect error (disconnect, parity, escape sequence, character sequence, credit) **or Link disabled condition.**

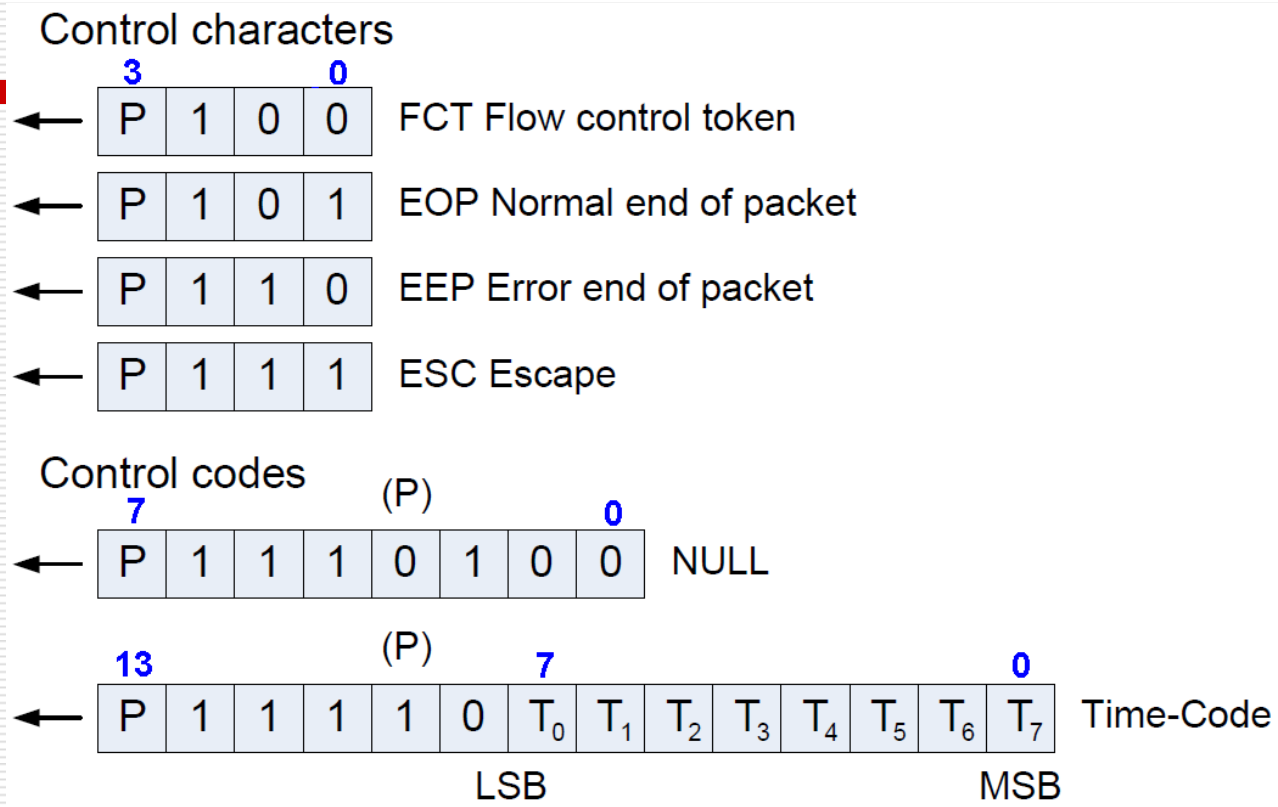
Conclusion

- An updated release of the basic SpaceWire standard is needed
- It should be done ASAP
- Official approval will take 6-9 month
- Prepare the draft to the next SpaceWire WG meeting.
  By June, to use ISC-2010 Conference event as SpaceWire WG/Steering Committee meeting also ?
- Organise intermediate contributions and discussions by e-mail and teleconferences (organised by the ESTEC?)

Thank you!

Back up

To add bit numbers (blue)



Following pictures are to be changed:

- Figure 4-4: Data and control characters
- Figure 7-2: SpaceWire control characters and control codes
- Figure 7-3: Parity coverage

[Link Disabled] condition and adding EEP to the buffer

The statements from **ECSS-E-50-12A**:

- **11.4 (p.98)** If any form of error is detected within the link interface then c) …add EEP (error end of packet) to the receiver buffer
- **11.4. (p.98)** d. Delete data in the transmitter buffer (i.e. transmit FIFO in Figure 33) until
- the next EOP (End of Packet).
- **8.5.3.8 (p.64)** [Link Disabled] is a condition set by external hardware or software in order to disable and stop the link interface.
- **8.5.2.7 (p.63) Run** b). If the link interface is disabled, or if a disconnect error, parity error, escape error or credit error is detected, while in the *Run* state, then the state machine shall move to the *ErrorReset* state.

The problem:

- The current received packet is not ended with EEP while state machine moved from RUN to *ErrorReset* state due to [Link Disabled] condition.

The decision:

- To add **11.4 (p.98)** with information about insertion of EEP (in receiving packet) and delete data in the transmitter buffer when [Link Disabled] condition occurs.

For Example:

- Rename chapter : 11.4 Link error/**Link disabled** recovery
- Rewrite : a) Detect error (disconnect, parity, escape sequence, character sequence, credit) **or Link disabled condition.**

To change the following picture:

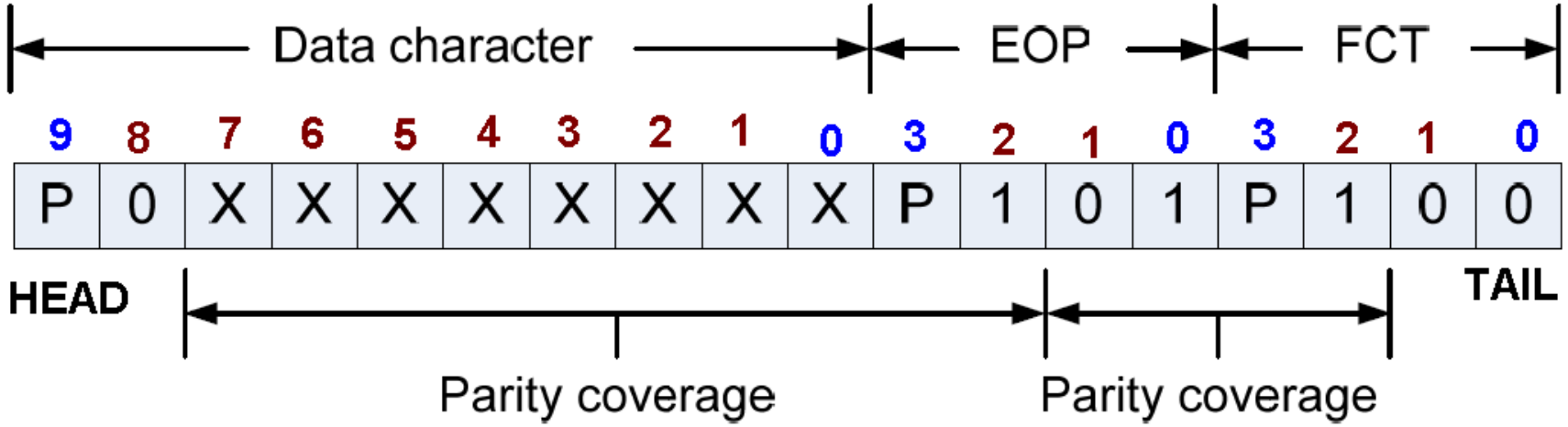


Figure 7-3: Parity coverage

Somehow have to be indicated parity bit to which the parity coverage corresponds.

Example 1: to write EOP P-bit parity coverage

Example 2: to mark each coverage and parity bit with one color

Physical level/Signal level

"The minimum data signalling rate is set by the disconnect timeout (clause 8.9.2.1 and 8.11.2) to greater than 1,18 Mb/s, i.e. 1/850 ns ." [p.46]