# High Accuracy Time Synchronization over SpaceWire Networks


**Aeroflex Gaisler AB**

**Kungsgatan 12, SE-41119 Göteborg, Sweden**

**sales@gaisler.com**

**+46 31 775 86 50**

**www.Aeroflex.com/Gaisler**

# Overview

- Definitions
- Time message over SpaceWire
  - Time formats
  - Protocol formats
- Time synchronization over SpaceWire
  - SpaceWire Time-Codes and Interrupts
  - Accuracy of SpaceWire Time-Codes
  - Improved accuracy of Time-Codes

# Definitions

- **Source:** Distributes time to destinations
- **Destination:** Receives time from a source

- **Time message:** Carries time value
- **Time synchronization:** Qualifies time value

# Time message over SpaceWire – time code

- CCSDS Recommendation for Time Code Formats, CCSDS 301.0-B-4
  - Preamble Field (P-Field)
    - Bit 0: Extension flag
    - Bit 1-3: Time code identification
    - Bit 4:7 Detail bits for information on the code
    - Note that the P-Field can be extended by an octet, bit 0 always being the Extension flag for the next octet.
  - Time Field (T-Field)
    - One or more octets

# Time message over SpaceWire - time code

- **CCSDS Recommendation for Time Code Formats, CCSDS 301.0-B-4**
  - CCSDS UNSEGMENTED TIME CODE (CUC)
    - TAI based, no leap second correction
  - CCSDS DAY SEGMENTED TIME CODE (CDS)
    - UTC based, leap second corrections
  - CCSDS CALENDAR SEGMENTED TIME CODE (CCS)
    - UTC based, leap second corrections
  - CCSDS ASCII CALENDAR SEGMENTED TIME CODE (ASCII)
    - UTC based, leap second corrections
  - AGENCY-DEFINED CODES

    - COORDINATED UNIVERSAL TIME (UTC)
    - INTERNATIONAL ATOMIC TIME (TAI)
    - GREENWICH MEAN TIME (GMT)

# Time message over SpaceWire - time code

- **CCSDS UNSEGMENTED TIME CODE (CUC)**
- **Octet 1 (mandatory if P-Field is used)**
  - **Bit 0 = P-Field Extension (‘zero’: no extension; ‘one’: field is extended)**
  - **Bit 1 - 3 = Time code identification**

    **001 — 1958 January 1 epoch (Level 1 Time Code)**

    **010 — Agency-defined epoch (Level 2 Time Code)**
  - **Bit 4 - 5 = Number of octets of the basic time unit minus one**
  - **Bit 6 - 7 = Number of octets of the fractional time unit**
- **Octet 2 (optional—presence is signaled in Octet 1)**
  - **Bit 0 = P-Field Extension (‘zero’: no extension; ‘one’: field is extended)**
  - **Bits 1-2 = Number of additional octets of the basic time added to that specified in Octet 1**
  - **Bits 3-5 = Number of additional octets of the fractional time added to that specified in Octet 1**
  - **Bits 6-7 = Reserved for mission definition**

# Time message over SpaceWire - time code

- **CCSDS UNSEGMENTED TIME CODE (CUC)**

| CCSDS Unsegmented Code | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| P-Field | | T-Field | | | | | | |
| 1st | 2nd | Coarse Time | | | | Fine Time | | |
| | | $2^{31}$ $2^{24}$ | $2^{23}$ $2^{16}$ | $2^{15}$ $2^{8}$ | $2^{7}$ $2^{0}$ | $2^{-1}$ $2^{-8}$ | $2^{-9}$ $2^{-16}$ | $2^{-17}$ $2^{-24}$ |
| 0 7 | 8 15 | 0 7 | 8 15 | 16 23 | 24 31 | 32 39 | 40 47 | 48 55 |
| 8 bits | 8 bits | 8 bits | 8 bits | 8 bits | 8 bits | 8 bits | 8 bits | 8 bits |

# Time message over SpaceWire - time code

- **Proposal:**
  - **Overall time message format and protocol not to be limited to one time format, it should instead support <u>all</u> CCSDS time formats**
  - **CUC to be the <u>baseline</u> (most commonly used)**
  - **Each recipient of the time message to accept <u>all</u> CUC sizes (as currently defined) as input, not only a limited subset, even if only a subset is implemented by the recipient**

# Time message over SpaceWire - protocol

- Time message protocols are traditionally unidirectional, e.g. Mil-Std-1553B, GPS, CAN
- Time message protocols traditionally only carry time information, possibly status from source
- Questions:
  - Is there a need for bi-directional communication?

    E.g. to read out status of a destination.
  - Is there a need for other information than just time?

    E.g. to communicate system settings.

    E.g. to setup destination remotely?

# Time message over SpaceWire - protocol

## Protocol structure:

- Simple unidirectional, e.g. CCSDS Unsegmented Code Transfer Protocol (CUCTP)

| Destination Logical Address | Protocol Identifier | CCSDS Unsegmented Code | | | | | | | | | CRC | EOP |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | P-Field | | T-Field | | | | | | | | |
| | | Default | Extended | Coarse Time | | | | Fine Time | | | | |
| | | | | $2^{31}$ $2^{24}$ | $2^{23}$ $2^{16}$ | $2^{15}$ $2^{8}$ | $2^{7}$ $2^{0}$ | $2^{-1}$ $2^{-8}$ | $2^{-9}$ $2^{-16}$ | $2^{-17}$ $2^{-24}$ | | *weight* |
| | | 0 7 | 8 15 | 0 7 | 8 15 | 16 23 | 24 31 | 32 39 | 40 47 | 48 55 | | *index* |
| 8 bits | 8 bits | 8 bits | 8 bits | 8 bits | 8 bits | 8 bits | 8 bits | 8 bits | 8 bits | 8 bits | 8 bits | *no. of bits* |

- Remote Memory Access Protocol (RMAP)
  - Supports write (uni-dir) and read (bi-dir)
- Custom protocol based on RMAP
  - Subset of RMAP for time distribution only
  - New Protocol ID (PID)

# Time message over SpaceWire - protocol

- **Proposal:**
  - Add a first control byte to enable future extensions (can be integrated in RMAP address)
  - Investigate the need for bi-dir communication, if so select RMAP (-like) protocol
  - Possible system settings:
    - Time-Code rate, e.g. 10 ms
    - Source for Synchronization, e.g. 63 to 0 wrap, or Interrupt #0
  - Possible destination setup:
    - Latency (i.e. distance from) wrt source

# Time synchronization – Time-Codes

- **SpaceWire Time-Codes - general**
  - 2 bits used for status flags, but to be changed to identifier bits instead (T7:T6)
- **SpaceWire Time-Codes with T7:76 = 00**
  - 6 bits used for counter (with values 0 to 63)
  - Legacy operation called time-codes hereafter
- **SpaceWire Time-Codes with T7:76 = ??**
  - 5 bit used for interrupt number
  - 1 bit used for interrupt/acknowledgement selection
  - New operation called interrupts hereafter

# Time synchronization – Time-Codes

- **SpaceWire Time-Codes with T7:76 = 00**
  - Proposed to be used in SpaceWire-D as time-slot delimiters, being sent every 10-100 ms (approximately)
  - Could be used for synchronization, but might be difficult to align 64 codes with e.g. 1 second boundary
- **SpaceWire Time-Codes with T7:76 = ??**
  - If priority is specified e.g. Interrupt #0, then short latency could be obtained (as goods as for time-codes)
  - Interrupt acknowledge #0 (from a destination) could be used for measuring round-trip delays (i.e. latency*2)
  - Separated from time-code usage in SpaceWire-D

# Time synchronization – Accuracy

- Time-Code distribution suffers from:
  - Latency:

    time it takes to transfer a time-code from source to destination
  - Jitter:

    variation of the above time
- Latency, theoretical:
  - Best case 14 bit periods
  - Worst case 14 bit periods + synchronization
- Jitter, theoretical:
  - Best case 10 bit periods
  - Worst case 12 bit periods + synchronization

# Time synchronization – Accuracy

- Latency, empirical:
  - Design dependent:
    - E.g. highly pipelined SpaceWire codec supporting 200 Mbits in an anti-fuse FPGA has high latency due to intermediate buffer stages
  - Varies with system frequency, transmitter/receiver frequency and actual bit rate:
    - Need to characterize at several frequencies and bit rates
- Jitter, empirical:
  - Add two bit periods for quantatization effects

# Time synchronization – improvements

- **Jitter reduction techniques have been proposed:**
  - Stretch the clock period (up to 12.5 Mbps)
  - Add fixed time to latency that can compensate jitter, requires that the jitter for each time-code is distributed with the time code. Requires a change to SpaceWire standard.
- **For extremely high requirement applications:**
  - Use a separate synchronization signal!
- **For high requirement applications:**
  - There might be hope…

# Time synchronization – improvements

- **Jitter reduction techniques have been proposed:**
  - Stretch the clock period (up to 12.5 Mbps)
  - Add fixed time to latency that can compensate jitter, requires that the jitter for each time-code is distributed with the time code. Requires a change to SpaceWire standard.
- **For extremely high requirement applications:**
  - Use a separate synchronization signal!
- **For high requirement applications:**
  - There might be hope…

# Time synchronization – improvements

- **Jitter reduction techniques based on statistical methods under investigation:**
  - Measure the delta between ideal (wrt local time-keeper) and actual occurrence of a time-code arrival
  - Store delta with sign and build statistics
  - Calculate appropriate correction
  - Correct ideal time for next expected occurrence of time-code arrival
- **Method also corrects drift (or frequency wander)**
- **Does not affect standardization**
- **Work in progress…**

# Contact information

- IP cores information:

  http://www.Aeroflex.com/Gaisler

- Board information:

  http://www.Aeroflex.com/Gaisler

- Software and tools information:

  http://www.Aeroflex.com/Gaisler

- Contact:

  sales@gaisler.com