Time synchronization in SpaceWire networks

Marko Isomäki, Sandi Habinc

Aeroflex Gaisler AB
Kungsgatan 12, SE-411 19 Göteborg, Sweden

marko@gaisler.com

www.Aeroflex.com/Gaisler

Introduction

Time synchronization in SpaceWire networks was discussed at the 15th WG meeting. That presentation focused on the technical details of a proposed protocol (CUCTP) for time synchronization.

This presentation will give a brief summary of the major technical aspects of the protocol while the major part focuses on system level aspects, a possible integration with SpaceWire-D and some discussion on potential drawbacks mentioned at the previous meeting.

Background

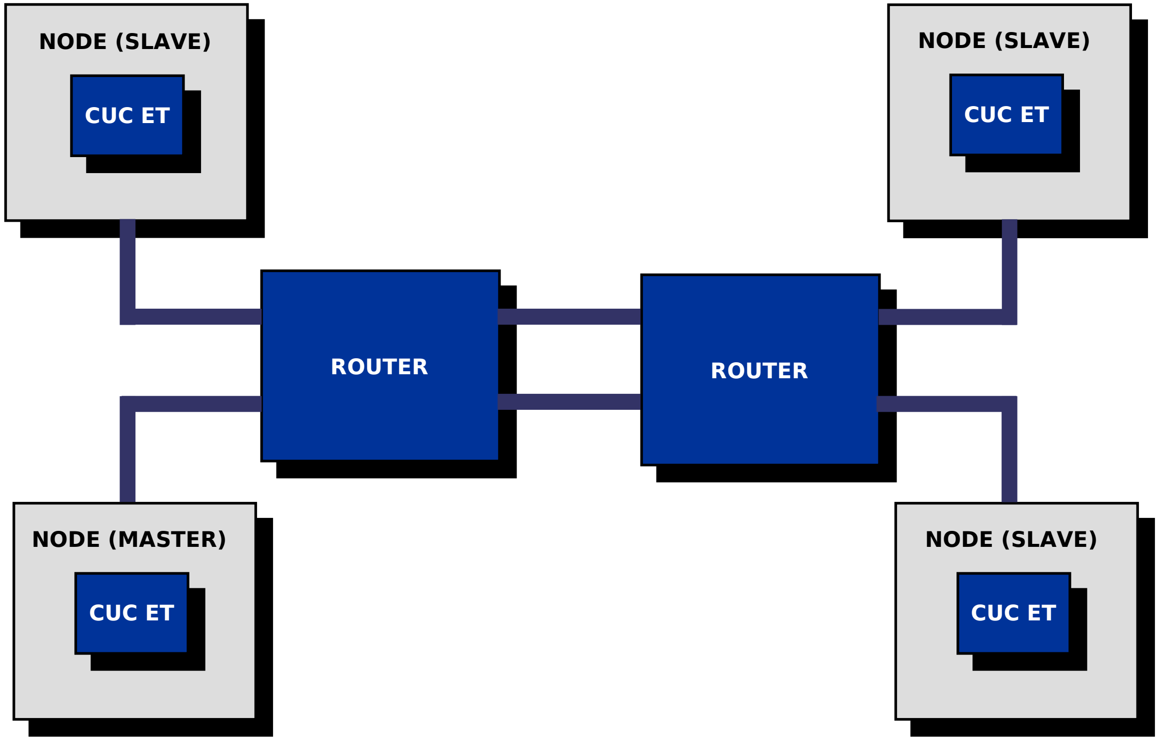
- SpaceWire is an asynchronous network i.e. there is no common clock signal being distributed for the communication meaning that each node is responsible for its own clock.
- The SpaceWire-D protocol is intended to provide means to handle real time traffic on a SpaceWire network capable of supporting control loop frequencies between 1 Hz and 1 kHz.
- Other important properties of the protocol are high throughput, simplicity and low/medium implementation cost.
- Command & Control and Data traffic are multiplexed on the same links/network.

Scheduling

- The method to achieve the requirements is to schedule transfers on the network level to avoid contention.
- This requires that all nodes in the network have a local clock which is synchronized to a common master clock.
- The node containing the master clock is responsible of distributing the master time to the other nodes.
- This is done through a time distribution protocol proposed at the previous WG meeting.

CUCTP time synchronization in a SpaceWire network

- Each node contains an Elapsed Time counter based on the CCSDS Unsegmented Code (CUC).
- One node is selected to be the time master and periodically sends time-codes and packets to keep the other nodes synchronized
- When synchronized the ET counters can be used for SpW-D scheduling.



CCSDS Unsegmented Code (CUC)

- CCSDS defines several different formats for how time should be defined in a system
- The most commonly used one is the CCSDS Unsegmented Code (CUC) defined in the 301.0-B-3 recommendation
- It supports resolutions that are higher than the minimum jitter than can be guaranteed by Time-code distribution in SpaceWire networks
- This was seen as the most suitable time-code format for use with SpaceWire time synchronization

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

AEROFLEX GAISLER

Synchronization of slave ET

The 6-bit time-count in the Time Codes are mapped to 6-bits in the CUC ET.

Time Codes are sent by the master with the frequency determined by the mapping to the CUC.

The Time Code is expected to be received synchronously with the slave ET counter i.e. at the time when the ET transitions to the expected value.

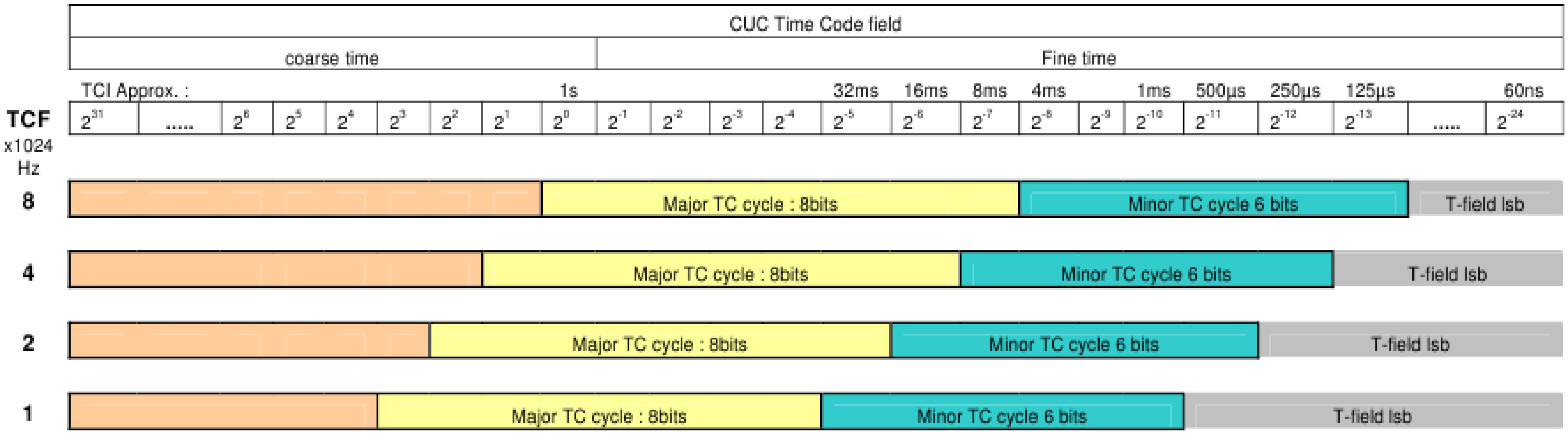
A window of tolerance around this point is allowed.

Synchronization of slave ET (2)

Time Codes keeps the bits from the msb to which it is mapped to and below synchronized

Bits with higher significance are kept in sync by periodically sending packets

The time-code master has to be the same as the master sending the packets (they are completely synchronous).



Time distribution data protocol

Needs to carry the complete CUC while the Time Codes only carry the 6-bits they are mapped to.

Several different suggestions:

1. CCSDS Unsegmented code transfer protocol (CUCTP) presented by Aeroflex Gaisler at 15th WG meeting.

Destination Logical Address	Protocol Identifier	CCSDS Unsegmented Code	CRC	EOP

2. Use the RMAP protocol with the CUC information being transferred by writing a certain (standardized?) address.

3. Use the same protocol as in RMAP but with a different PID defining a separate address space.

Time distribution data protocols pros and cons

CUCTP:

Simple, requires small hardware resources, supports packet distribution

Cannot use existing hardware

RMAP:

Can use existing hardware (CUC mapped to existing address space), supports replies (but are they needed?)

Does not support packet distribution, might not be possible to map CUC at same address, more expensive in hardware if RMAP not already present.

RMAP new PID:

Can use existing hardware with small modification, supports replies, supports packet distribution, no potential address clashes

more expensive in hardware if RMAP is not already present in target

Accuracy

Limited by three different causes: latency, jitter and drift.

The highest proposed Time Code frequency (8 kHz) has a period of 122.1 us. At 200 Mbit/s the jitter at each link is 70 ns and the latency for each link transfer is 70 ns.

In most networks this should give enough accuracy for even the shortest Time Code period.

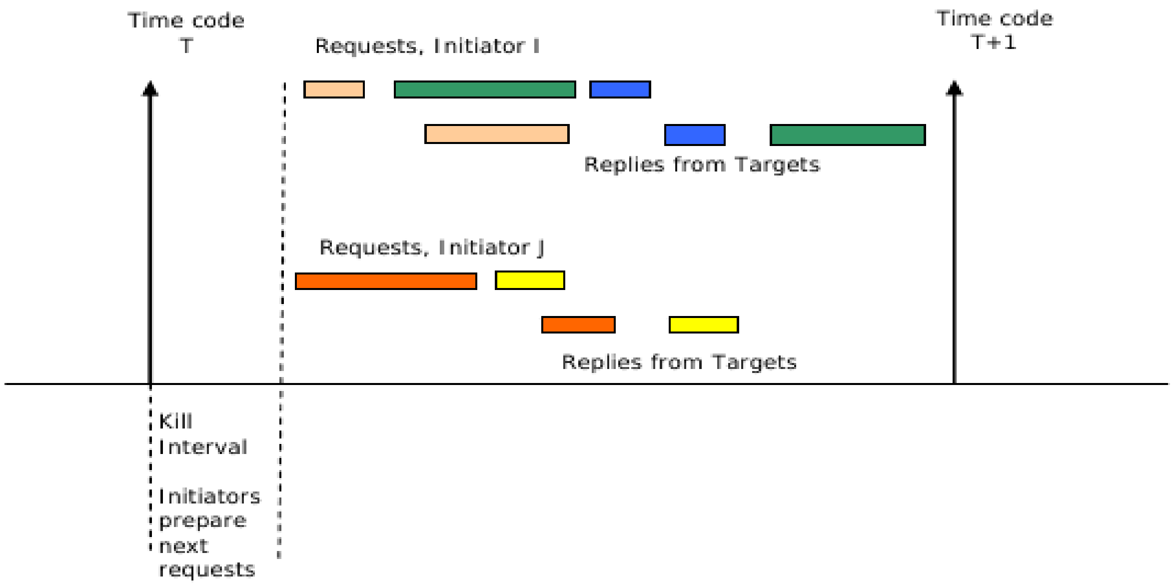
If the whole network or one or more links run at 2 Mbit/s the jitter and latency are both 7 us. This is no longer negligible. Methods to measure average jitter and latency would have to be used.

With latency and jitter estimated averaging can be used to determine an offset between master and slave ETs which should be caused by drift.

Kill period

The kill period introduced in the SpW-D requirements and trade-offs document suggests that this period is a fraction of a Time Code period.

This would require that bits in the CUC less significant than than the Time Code mapping should be synchronized. This might not be the case and packets might be erroneously killed (or not killed).



Kill period(2)

The kill action itself appears to be a non-trivial task to accomplish especially in routers.

A router for example needs to detect that an input/output port pair have exceeded their limit, stop the transfer and spill both the receiver and transmitter.

Spilling receiver and transmitter could be a difficult task to perform and to have an upper time limit.

Also should it be notified to an initiator that a packet has been killed at the target? How should this be done.

Implications for RMAP

The current SpW-D proposal also implicitly imposes timing restrictions on the RMAP layer.

As to our understanding this limits RMAP to certain lengths for a specific configuration (e.g. 10 MHz Link, 1 kHz TC). To avoid restrictions on RMAP this requires segmentation.

The current proposal seems to locate this segmentation higher than RMAP in the protocol stack resulting several layers being affected.

How will an RMAP target in hardware for example handle a long write? There would have to be a SPW-D layer tightly integrated into the core which detects timeouts and spills the packet.

This results (again) in a non-trivial layering.

Conclusions

- It is better to use either a specific protocol e.g. CUCTP or RMAP with a new protocol ID for distributing time.
- Methods for determining network timing parameters such as latency and jitter might be needed especially with low or mixed link speeds.
- Using bits with lower significance than the Time Code mapping can lead to erroneous packet dropping due to lack of synchronization.
- Kill function and RMAP over SPW-D can be tricky to implement especially in existing hardware

Questions?

AEROFLEX GAISLER