CCSDS Unsegmented Code Transfer Protocol (CUCTP)

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 spacecraft is becoming increasingly important. E.g. instruments and navigational on-board resources can now be combined for establishing scientific observations and therefore need to be synchronized in time.

This has been done via dedicated signals or deterministic on-board buses (e.g. MIL-STD-1553 or OBDH).

With the advent of SpaceWire point-to-point links and routing switches being used for critical control functions the need for accurate time synchronization via this network has arisen.

Current time synchronization support in SpaceWire

- 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
- No means for handling drift caused by unstable oscillators or crystals
- No support for automatic time message and pulse distribution
- Rudimentary time-code transmission
- No means for handling delays and jitter caused by routing

CCSDS Unsegmented Code Transfer Protocol (CUCTP)

New protocol for maintaining synchronization within a SpaceWire network developed in cooperation with ESA, SciSys and Astrium.

Uses SpaceWire packets for high-level synchronization, based on CCSDS Unsegmented Code (CUC).

Uses SpaceWire Time-Code time-information for low-level synchronization, which is coupled to CUC. A new Protocol Identifier (PID) based transfer protocol (CUCTP) is defined for SpaceWire packets carrying CUC.

Destination Logical Address	Protocol Identifier	CCSDS Unsegmented Code	CRC	EOP

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
- Often used in Elapsed Time (ET) counters on board spacecraft
- 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

CUCTP packet fields

Destination Logical Address as per ECSS-E-ST-50-51C.

Protocol ID as per ECSS-E-ST-50-51C, programmable, in range 0xF0 to 0xFE.

CRC is the same as for RMAP as per ECSS-E-ST-50-52C.

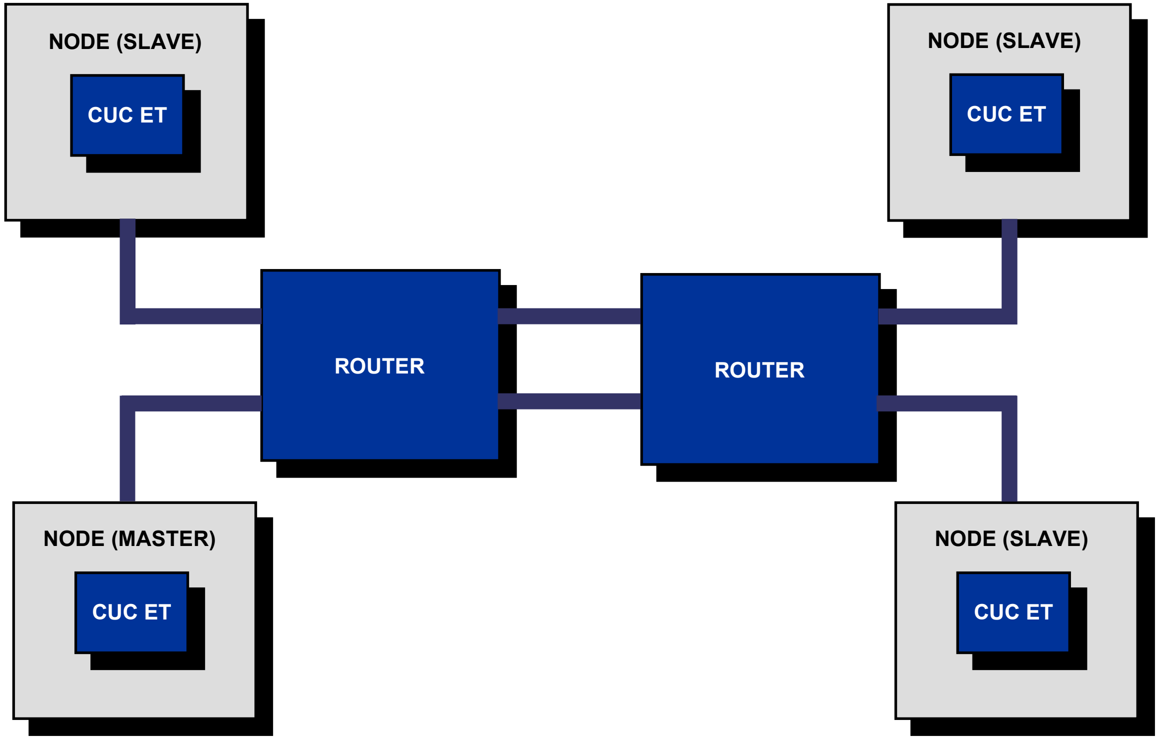
EOP as per ECSS-E-ST-50-12C.

CCSDS Unsegmented Code as per 301.0-B-3.

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*

CUCTP time synchronization in a SpaceWire network

- Each node contains an Elapsed Time counter based on the CUC format.
- One node is selected to be the time master and periodically sends time-codes and CUCTP packets to keep the other nodes synchronized



Synchronization of slave ET (1)

The 6-bit time-count in the Time Codes is 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.

If the Time Code was received within the window it is compared to the ET bits corresponding to the time-count mapping and if equal the ET is considered to be synchronized.

If no Time Code is received within the window the ET is considered synchronized but freewheeling at the Time Code level

Synchronization of slave ET (2)

A CUCTP packet is sent by the master every 64 Time Codes.

It is used for synchronizing bits with higher weight than those mapped to the Time Code.

Whenever the Time Code time-count wraps from 0x3F to 0x00 and a valid CUCTP packet has been received the higher weight bits in the ET are compared to the corresponding bits in the packet.

If there is a mismatch a wrapping error has occurred.

Synchronization of slave ET (3)

If no packet has been received the ET is still considered synchronized but freewheeling on the packet level.

The whole ET can also be initialized using the CUCTP information. This is only done occasionally.

Note that the master has to send the CUCTP packet to all slaves individually.

Routers with packet distribution could relieve this issue by assigning one or more addresses to be used as broadcast or multicast addresses for CUCTP.

The Aeroflex Gaisler SpaceWire router has support for packet distribution and tests with the CUCTP are planned.

Synchronization of slave ET (4)

Elapsed Time counter bits (T-Field)

Ideal synchronization point (OFFSET=0x00000000)

Field	CCSDS index	AMBA index	Weight	Time-Code mapping	Tolerance		Elapsed Time counter bits (T-Field)	Window of tolerance		
T-Field Coarse Time	0	31	2^{31}			Checked bits during wrap, compared with CUCTP packet	00000000000000000000000000000000	0000000000000000	0000000000000000	00000000000000000000000000000000
	...	...	...				...	...	...	...
	17	14	2^{14}				00000000000000000000000000000000	0000000000000000	0000000000000000	00000000000000000000000000000000
	18	13	2^{13}				00000000000000000000000000000000	0000000000000000	0000000000000000	00000000000000000000000000000000
	19	12	2^{12}				00000000000000000000000000000000	0000000000000000	0000000000000000	00000000000000000000000000000000
	20	11	2^{11}				00000000000000000000000000000000	0000000000000000	0000000000000000	00000000000000000000000000000000
	21	10	2^{10}				00000000000000000000000000000000	0000000000000000	0000000000000000	00000000000000000000000000000000
	22	9	2^{9}				00000000000000000000000000000000	0000000000000000	0000000000000000	00000000000000000000000000000000
	23	8	2^{8}				00000000000000000000000000000000	0000000000000000	0000000000000000	00000000000000000000000000000000
	24	7	2^{7}	31			00000000000000000000000000000000	0000000000000000	0000000000000000	00000000000000000000000000000000
	25	6	2^{6}	30	31		00000000000000000000000000000000	0000000000000000	0000000000000000	00000000000000000000000000000000
	26	5	2^{5}	29	30		00000000000000000000000000000000	0000000000000000	0000000000000000	00000000000000000000000000000000
	27	4	2^{4}	28	29		00000000000000000000000000000000	0000000000000000	0000000000000000	00000000000000000000000000000000
	28	3	2^{3}	27	28		00000000000000000000000000000000	0000000000000000	0000000000000000	00000000000000000000000000000000
	29	2	2^{2}	26	27		00000000000000000000000000000000	0000000000000000	0000000000000000	00000000000000000000000000000000
	30	1	2^{1}	25	26		00000000000000000000000000000000	0000000000000000	0000000000000000	00000000000000000000000000000000
	31	0	2^{0}	24	25		00000000000000000000000000000000	0000000000000000	1111111111111111	11111111111111111111111111111111
T-Field Fine Time	32	31	2^{-1}	23	24	Time-Code, checked during synchronization (MAPPING=18)	11111111111111111111111111111111	1111111111111111	0000000000000000	00000000000000000000000000000000
	33	30	2^{-2}	22	23		11111111111111111111111111111111	1111111111111111	0000000000000000	00000000000000000000000000000000
	34	29	2^{-3}	21	22		11111111111111111111111111111111	1111111111111111	0000000000000000	00000000000000000000000000000000
	35	28	2^{-4}	20	21		11111111111111111111111111111111	1111111111111111	0000000000000000	00000000000000000000000000000000
	36	27	2^{-5}	19	20		11111111111111111111111111111111	1111111111111111	0000000000000000	00000000000000000000000000000000
	37	26	2^{-6}	18	19		11111111111111111111111111111111	1111111111111111	0000000000000000	00000000000000000000000000000000
	38	25	2^{-7}	17	18	Checked bits during synchronization	11111111111111111111111111111111	1111111111111111	0000000000000000	00000000000000000000000000000000
	39	24	2^{-8}	16	17		11111111111111111111111111111111	1111111111111111	0000000000000000	00000000000000000000000000000000
	40	23	2^{-9}	15	16		11111111111111111111111111111111	1111111111111111	0000000000000000	00000000000000000000000000000000
	41	22	2^{-10}	14	15		11111111111111111111111111111111	1111111111111111	0000000000000000	00000000000000000000000000000000
	42	21	2^{-11}	13	14		11111111111111111111111111111111	1111111111111111	0000000000000000	00000000000000000000000000000000
	43	20	2^{-12}	12	13		11111111111111111111111111111111	1111111111111111	0000000000000000	00000000000000000000000000000000
	44	19	2^{-13}	11	12		11111111111111111111111111111111	1111111111111111	0000000000000000	00000000000000000000000000000000
	45	18	2^{-14}	10	11		11111111111111111111111111111111	1111111111111111	0000000000000000	00000000000000000000000000000000
	46	17	2^{-15}	9	10		11111111111111111111111111111111	1111111111111111	0000000000000000	00000000000000000000000000000000
	47	16	2^{-16}	8	9		11111111111111111111111111111111	1111111111111111	0000000000000000	00000000000000000000000000000000
	48	15	2^{-17}	7	8		11111111111111111111111111111111	1111111111111111	0000000000000000	00000000000000000000000000000000
	49	14	2^{-18}	6	7		11111111111111111111111111111111	1111111111111111	0000000000000000	00000000000000000000000000000000
	50	13	2^{-19}	5	6		00000000000000001111111111111111	1111111111111111	0000000000000000	00000000000000001111111111111111
	51	12	2^{-20}	4	5		11111111111111110000000000000000	1111111111111111	0000000000000000	11111111111111110000000000000000
	52	11	2^{-21}	3	4	Ignored bits (TOL=4)	00000000111111110000000011111111	0000000011111111	0000000011111111	00000000111111110000000011111111
	53	10	2^{-22}	2	3		00001111000011110000111100001111	0000111100001111	0000111100001111	00001111000011110000111100001111
	54	9	2^{-23}	1	2		00110011001100110011001100110011	0011001100110011	0011001100110011	00110011001100110011001100110011
	55	8	2^{-24}	0	1		01010101010101010101010101010101	0101010101010101	0101010101010101	01010101010101010101010101010101

AEROFLEX GAISLER

SPWCUC – IP Core

A new VHDL IP core, SPWCUC, has been developed that supports automatic reception of SpaceWire Time-Codes and CUCTP packets (i.e. slaves).

CUCTP packets are sent every 64 Time-Codes, and the reception of a Time-Code with time-information = 0x00 synchronizes the CUCTP contents with the local counter.

No software support is required in receivers (i.e. slaves).

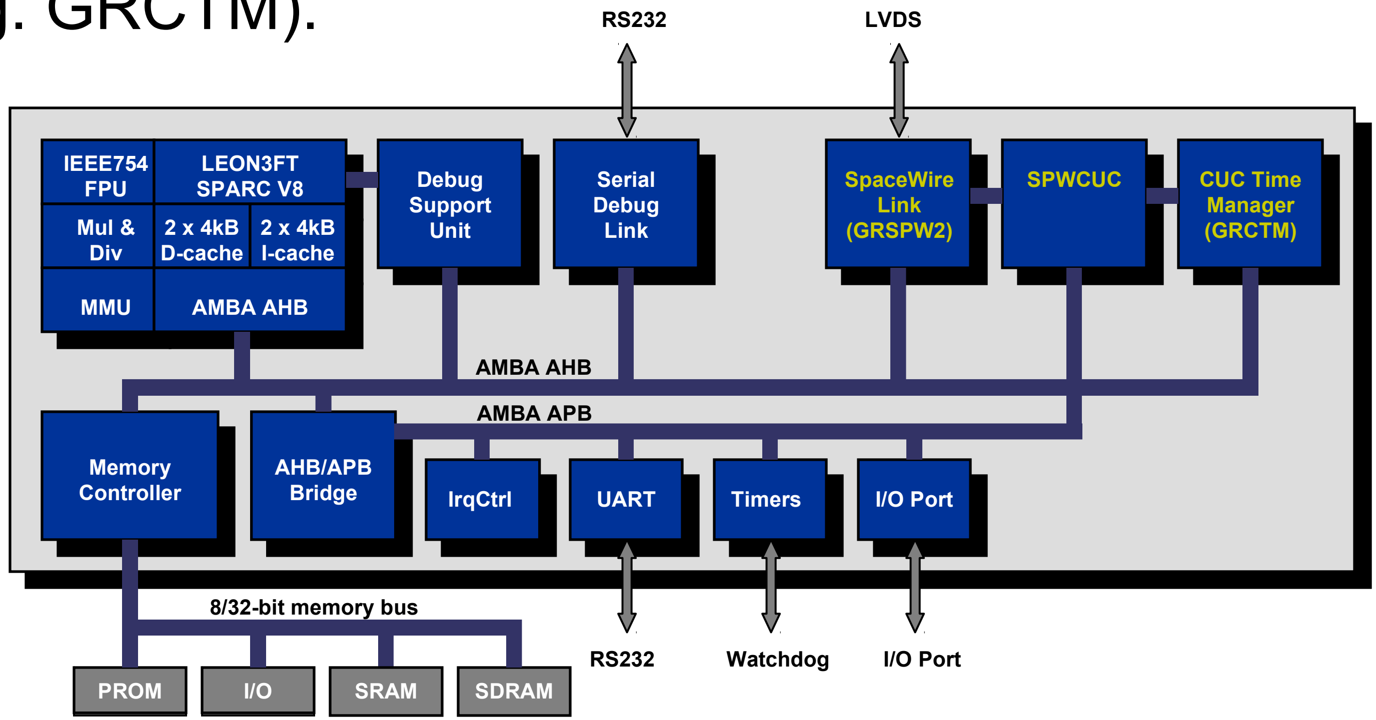
Support for automatic Time-Code transmission and CUCTP packet forming is also provided (i.e. masters).

The core is fully integrated in the GRLIB IP core library.
SPWCUC has already been deployed in RASTA systems funded by European Space Agency (ESA).

SPWCUC – IP Core in a LEON3 system

SPWCUC interfaces with other IP cores through dedicated signals for high time accuracy, as well as an AMBA APB interface for access from software etc.

SPWCUC IP core is to be used together with a SpaceWire codec (e.g. GRSPW2) and CUC time manager (e.g. GRCTM).



AEROFLEX GAISLER

Latency (1)

We now have a method for sending messages but no means for taking latency, jitter and drift into account.

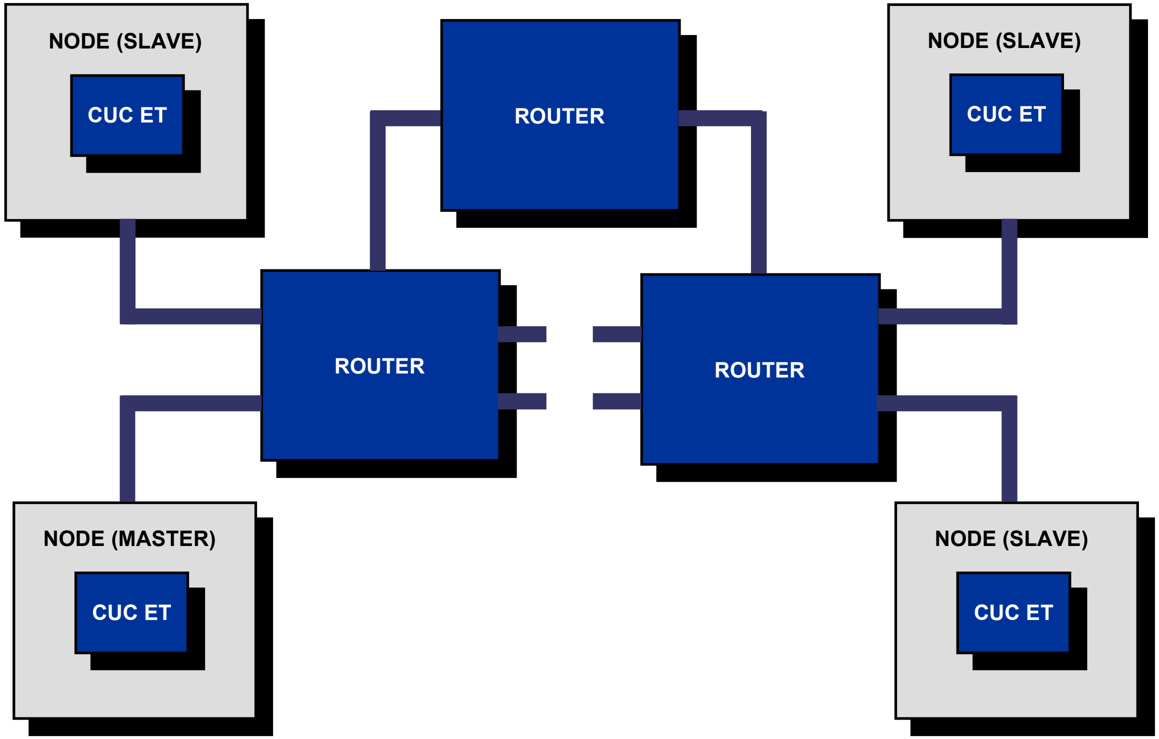
Latency is taken into account at the slaves by adding an offset to the nominal point at which a Time Code is expected to arrive.

The offset needs to be quantified. The minimum offset can be calculated as the delay for the Time Code along the shortest path in the network.

Latency (2)

The difficult task is to determine a maximum value in networks where e.g. routers can be taken offline causing the Time Code to take a different path.

One method to determine the latencies is by sending Time Codes along all different paths during a startup phase. This concept is used in the Network Time Protocol (NTP) and Precision Time Protocol (PTP).



Jitter (1)

The second thing to determine is jitter.

Jitter is caused by uncertainties in the actual time of Time Code transmission from the transmit request.

It is caused by the fact that Time-code transmission is delayed from the request with the time remaining for transmitting the current character.

The difference between the longest and shortest time depends on the character being sent. It is in the order of 10 transmission clocks and for 200 Mbps it is in the range of 50 ns. The problem is compounded for each link interface the Time Code passes.

Jitter (2)

As for latency jitter would be known in a fixed configuration, but with variable propagation paths it might be difficult to bound.

Dynamic methods for determining jitter exist in the Ethernet domain. Similar methods that would involve Time Code passing between master and slave could be used for SpaceWire.

The challenge is to move beyond the accuracy that can be achieved over the internet (10 ms) and local area networks (200 us).

Drift

The third thing to determine is the drift in the system.

It should be possible to compensate for drift if the latency and jitter has been determined accurately.

Then the average error between the received Time Code and the expected time it should arrive could be considered to be caused by drift.

This can be compensated for by changing the offset.

Summary

- The CCSDS Unsegmented Code Transfer Protocol provides the means for achieving accurate time synchronization over SpaceWire
- It is built from existing CCSDS time information recommendations and ECSS SpaceWire standards.
- It works with existing SpaceWire hardware without modification
- The accuracy depends on methods to determine system jitter and latency. These methods have not been defined