K. Kasinski[1], W. Zabolotny[2], R. Szczygiel[1]
[1]AGH University of Science and Technology Cracow
[2]Warsaw University of Technology

v.0.4 – Modifications: proposal of special registers, added epoch_var3 as a proposal for epoch/status subframe

v.1.0 – changed K.28.1 to K.28.5 comma used to simplify the encoder design (Cadence chipware). Revised frame contents according. Added address space. Added short link synchronization procedure. Added short verbal description of uplink and downlink. Other minor changes.

v.1.1 – modified link synchronization procedure (different commas, different bit alignment), added defined state after reset

v.1.2 – fixed typo in uplink timestamp bits (thanks Joerg)

v.1.3 – changed SOS and EOS characters to make it more robust (thanks Robert), clarified description of TS_MSB frame generation (thanks Joerg), changed uplink frame picture in overview (thanks Joerg), added status bits on missing sequence number(thanks Joerg). Added proposed address space of STSXYTER. Added comment on address validity {WRaddr} and many other modifications (Joerg).

v.1.4 – (2014.10.14) – minor typos and fixes (Joerg).

v.1.5 – (2014.10.22) – added appendix on timestamp data handling and resorting considerations.

v.1.6 – (2014.10.24) – modifications suggested by W. Zabolotny. Clarified uplink/downlink direction. Changed pictures in overview. Added worknote in downlink part. Clarified ARQ behavior. Added worknote on possible transmission of current TS value when no hits are transmitted. Typos. Clarified Post_reset frame. Reformatted uplink frame table. Other minor changes. Modified appendix C.

v.1.7 – (2014.10.25) – added way the incoming traffic is distributed onto the links. Removed “min. 800ns period” from uplink description and pictures (Robert).

v.1.8 – (2015.01.05) – added modifications requested by Joerg.

v. 1.9 – (2015.01.09) - Row=192, Col=25 register description change. Decision made on way of solving problem with correct hit assignment within larger timeline -> Dummy hits will contain actual values of TS (part of it).

v.1.10 – (2015.01.12) – Added CRC polynomials, Corrected numeration of registers in *Row=192*

v.1.11 – (2015.01.15) – Added register mnemonics.

v. 1.12 – (2015.01.22) - Few corrections by Joerg: 3.1, Table 3 (updated dummy hit -> Timestamp <13:6>), Updated figure in Appendix C.

v.1.13 – (2015.01.29) – Added byte/bit ordering information for uplink and downlink frames on the link, added CRC poly description

v.1.14 – (2015.01.30) – Added exemplary frames

v.1.15 – (2015.02.03) – added 3-bits of sequence number to RDdata_ack frame (CRC reduced from 7 to 4 bits).

v.1. 16 – (2015.02.06) – modified register functionality: CMD_SOFT_RESET
- CMD_TS_RESET register renamed to CMD_TS_LOAD
- register C22_R192 removed (functionality moved to STATUS)
- register C23_R192 renamed to FIFO_AFULL_THR
- register C24_R192 renamed to FIFO_AFULL_COUNT
- modified functionality of the register C25_R192 ELINK_MASK (hard and soft masking modes introduced)
- 11 bits defined in the STATUS register
- new register: STATUS_MASK, C28_R192
- new register: FE_EM_THR, C29_R192
- new register: FE_EM_COUNTER, C30_R192
- new register: SEU_COUNTER, C31_R192

8b/10b version of protocol

1. Protocol Overview

Regular Downlink frames (from DPB to ASIC):

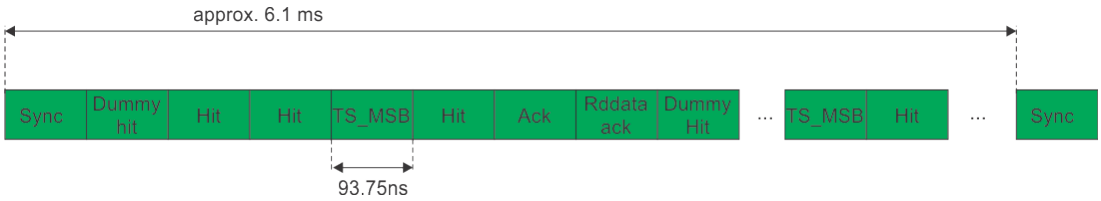
Example of 5 frames sent via downlink

- downlink transfer via GBT frame
- shared downlink (8 chips/shared link)
- multidrop data signal (e-link) & clock (p-clock)

Note: important to keep clock and data lines identical so that optimal phase alignment of the clock vs. data developed during link initialization is valid for all 8 chips on FEB.

- Clock received from GBTx is 160MHz
- downlink rate: 160 Mb/s
- link bandwidth 2.6 Mframes/s (375ns/frame)
- constant frame length: 60 bits (6 8b/10b characters)
  10-bit K28.5 comma character + 5 data characters
- 1 frame type with 4 request types

Regular Uplink frames (from ASIC to DPB):



Example of frames sent via uplink

- uplink transfer via wide frame mode (up to 14 e-links/GBTx @ 320Mbps),
- constant frame length: 30 bits (3 8b/10b characters),
- 1, 2 or 5 uplinks per STSXYTER (programmable)

Note: It will be possible to select any subset of the 5 available links (Reg address: Row=192, Col=24).

- uplink rate: 320 Mb/s,
- link bandwidth: 9.41 Mhits/s (assuming TS_MSB is sent every 800ns),
- 6 various frame types,

Link Synchronization Downlink / Uplink Frames

During link synchronization three additional frames are used:

- SOS – start of synchronization
- EOS – end of synchronization
- K28.1 comma character

Post-Reset Uplink Frames

After chip is reset it keeps on sending K28.5 comma character until the link synchronization is triggered.

2. Downlink (DPB to ASIC) Control Frame

The downlink is to transmit the following control requests:
No_op – No operate. Empty control frame. No effect, no acknowledge required.
WRaddr – Write address. Transmits address used for register write and read commands.
RDdata – Read data from previously set address.
Wrdata – Write data to previously set address.
(last three are referred in the text as “acting commands”).
For the error correction scheme the modified selective repeat ARQ mode is proposed: control acknowledge messages must contain the 4-bit sequence number, not-acknowledged control requests are retransmitted, the register values can be verified by readback. Timeout for receiving ack should be implemented in the DPB. DPB should not transmit new acting command unless it receives an acknowledgement.

```
Worknote (Joerg): The "ack" will not reach the DPB before the next possible slot for a control
request.  The  GBTx  core  latency  alone  is  166ns(down)  +  254ns(up)  =  420ns;
The  protocol  implementation  either  has  to  be  able  to  handle  multiple  pending  "acks"
(which  should  be  fine  with  the  4bit  sequence  number),  or  in  case  of  waiting  for  the  "ack"
before sending the next control request we limit the max. rate of control sequences by a factor of at
least 3 (eventually more).
```

Decision: Commands “no_op”, “WRaddr” and “WRdata” can be transferred continuously.
Only command “RDdata” should wait for Rddata_ack frame (which does not contain sequence number)

For error detection, a 16-bit CRC polynomial 0xC86C is used. It provides a Hamming Distance of 6 for less than 135 data bits. 14-bit address space covers physical registers of the front-end chips but also virtual registers triggering special functions (like FIFO flush, front-end reset etc.). Since downlink channel is shared by 8 ASICs the chip addressing is required. 4-bit chip address was used to enable also broadcasting commands when address FF is set. All special commands like DAQ_start, DAQ_stop, Sync, WRtime are done by accessing registers.

2.1 Regular control frame

- constant length (60 bits after 8b/10b encoding)
- control frames sent continuously (after link initialization procedure)
- 4 request types, 14-bit payload, 16-bit CRC
- CRC-16 digest: polynomial = 0xC86C = $(x^{16})+x^{15}+x^{12}+x^{7}+x^{6}+x^{4}+x^{3}+1$, HD=6 for <135 data bits

BYTE<0> frame_bits<47:40> bits_8b10b<59:50>	BYTE<1> frame_bits<39:32> bits_8b10b<49:40>	BYTE<2> frame_bits<31:24> bits_8b10b<39:30>	BYTE<3> frame_bits<23:16> bits_8b10b<29:20>	BYTE<4> frame_bits<15:8> bits_8b10b<19:10>	BYTE<5> frame_bits<7:0> bits_8b10b<9:0>
Comma character	Frame ID & Chip Address	Request type / Payload	Payload	CRC	CRC
K28.5 001111 1001 110000 0110	<7:4> chip address (0..7, 15) <3:0> sequence number (0..15)	<7:6> Request type (0..3) <5:0> Payload <13:8>	<7:0> Payload <7:0>	<7:0> CRC <15:8>	<7:0> CRC <7:0>

Byte order on the link: **BYTE<0>** first (then 1,2,3,4 and 5)
Bits order on the link: **bits_8b10b<59>** first (then 58, 57, ..., 1, 0)

Chip address	Comment
0x0 … 0x7	Individual chip addressing (8 chips per FEB). Addresses are assigned by bonding of 3 pads on FEB. Default address value: 0 (on-chip pull-down)
0x8 … 0xE	Reserved
0xF	Broadcast

Request Type	Code	Ack req?	Payload	Comment
no_op	0x0	No	X	No operation. Empty control frame. No effect.
WRaddr	0x1	Yes	14’b address	Write address (for WRdata). Address remains valid for consecutive WRdata requests. But RDdata overwrites it with register address used in RDdata frame.
WRdata	0x2	Yes	8’b data	Write data to register (address set previously by WRaddr)
RDdata	0x3	Yes	14’b address	Read data from register.

14-bit address covers all STSXYTER internal registers and leaves margin for additional, off-channel registers.
Details of address space will be provided in a STSXYTER2 documentation.

Downlink frame example #1:
Downlink data for register MASK_13_0 readout, chip address = 0, sequence number = 0 (skipping K28.5):
0x00c4c0cf2c [0, 196, 192, 207, 44]
Downlink frame example #2
Downlink data for register MASK_13_0 readout, chip address = 0, sequence number = 1 (skipping (K28.5):
0x01c4c0f75e [1, 196, 192, 247, 94]

Uplink (ASIC to DPB) Frames

The uplink data contain mainly hit data but also control responses (e.g. register value, acknowledgements) and status data (e.g. Timestamp MSBs, status bits). The optimization of the frame format with respect to the maximum link throughput was done. As a data compression feature, hit data can contain only 8 LSBs of the Timestamp (plus 2 additional bits overlapping with remaining timestamp part) while the remaining bits will be transferred separately (in the TS_MSB frame). Such a frame structure results in:
- constant, 30-bit frame (after 8b/10b encoding)
- average number of hits per 9th TS bit change: 25.6 (assuming 250 khit/s/channel)
- link throughput: 9.41 MHit/s/link
- link occupancy: 62.34% (when all 5 links are used and assuming 250khit/s/chanel)

Dummy Hit is an empty frame used to keep the link synchronous in case nothing is to be transferred. It differs from a regular Hit subframe in ADC value being equal to 0 (if the hit was registered ADC value is always higher than 0) and timestamp value also equal to 0.
The TS_MSB is a data compression frame transmitted every time the hit timestamp bits <13:8> change. This is the MINIMUM period for TS_MSB transmission. It might be larger in case there are no hits for a longer period of time. It contains 6 most significant timestamp bits triplicated and CRC-protected. Note: TS_MSB generation is based on the hits coming out from the sorter so in case no hits are registered the TS_MSB will also be not transmitted. The TS_MSB is sent before the new hits with modified <13:8> TS bits.
TS_MSB frame is generated and transmitted on each link separately (whenever TS <13:8> bits change in any link).
Two overlap bits <9:8> are used to enable data resorting at the edges of TS_MSB generation moment (due to the intrinsic uncertainty of 200ns resulting from frame generation time vs. pulse amplitude relation).

In case no hits are transmitted in a longer period of time there might be a problem with tracing the absolute time the new incoming hit belongs to (see Appendix C). For this purpose dummy hits in the TS field will contain TS<13:6> bits.

There are two types of acknowledgement frames. *Ack* is a regular, short acknowledgement. 2-bit ACK encodes acknowledgement status while 4-bit sequence number enables identification of acknowledged command. In addition, for diagnostic purposes 4-bit status, CP- configuration parity bit (detecting SEUs on configuration registers) and 6 LSBs of timestamp are transmitted. *RDdata_ack* however contains 14-bit register content and 7-bit CRC protection (polynomial: 0x5B providing Hamming Distance of 4). 14-bit payload is required to read back the addresses which are 14-bit wide. Acknowledgement message can be transferred on any link.

Sync frame contains three consecutive comma characters K28.5. It is transferred roughly at the rate of 166 Hz (6.114 ms : 2^16 frames) to enable monitoring of the link synchronization (it is not used to resynchronize the link on-the-fly). If the DPB does not see these frames periodically it may decide to resynchronize the link. It is not used to dynamically adjust the bit order in the FIFO!.

Alert cases: STSXYTER can trigger sending *Ack* frames without appropriate request if one of the alert cases occur:
- Throttling alert (more than specified number of channel FIFOs are almost full)
- Sync alert (downlink comma characters arrived out of sync)
- Incorrect sequence number in downlink occurred.

2.2 Frame

Table. 3. Structure of the uplink frames.

Structure of the uplink frames (before 8b/10b encoding)																								
	BYTE<0> frame_bits<23:16> bits_8b10b<29:20>								BYTE<1> frame_bits<15:8> bits_8b10b<19:10>								BYTE<2> frame_bits<7:0> bits_8b10b<9:0>							
Type	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Dummy Hit	0	7-bit channel address = 0x00							5-bit ADC = 0x00					0x0		0x0 (TODO: Timestamp<13:6> (actual state of counter))								0
Hit	0	7-bit channel address							5-bit ADC > 0x00					TS<9:8> (overlap)		Timestamp<7:0>								EM
TS_MSB	1	1	Timestamp<13:8>						Timestamp<13:8>						Timestamp<13:8>						4-bit CRC poly 0x9 = (x^4)+x+1			
RDdata_ack	1	0	1	14-bit register content																	3-bit sequence number (LSB)		4-bit CRC poly 0x9 = (x^4)+x+1	

Ack	1	0	0	ACK	4-bit sequence number	CP	4-bit status value	0x00 or Timestamp<7:2> depending on CONFIG<1> register setting	4-bit CRC poly 0x9 = $(x^4)+x+1$

Byte order on the link: **BYTE<0>** first (then 1 and 2)
Bit order on the link: **bits_8b10b<29>** first (then 28, 27, …, 1, 0);

Structure of the uplink frames (represented only in 8b/10b encoding			
Type	10-bit frame (8b/10b)	10-bit frame (8b/10b)	10-bit frame (8b/10b)
Sync	K28.5 comma character	K28.5 comma character	K28.5 comma character
POST_Reset	K28.5 comma character		

Description:
EM – event missed flag,
CP – config parity,
ACK<1:0>
0x0 – reserved,
0x1 – acknowledge (command represented by 4-bit seq number is acknowledged),
0x2- not acknowledge (command represented by 4-bit seq number is not acknowledged: CRC not matched)
0x3 – alert (check 4-bit status value)
STATUS <3:0>
<0> - Throttling alert (indicates that channel FIFOs start to assert almost full, threshold is configurable)
<1> - Sync alert (indicates that downlink comma characters arrive out of sync)
<2> - ~~Incorrect sequence number in downlink~~
<3> - Other error (need to check error registers)
Register Content:
Data can originate from:
- ADC counters (12-bit)
- 8-bit Front-End registers (DICE cells)
- 14-bit Register File registers
Unused MSbs are equal 0.

Uplink frame examples
RDdata_ack for a register with content = 0: 0xa00025 [160, 0, 37]
RDdata_ack for a register with content = 2: 0xa0014b [160, 1, 75]
Note: there are no fixed ACK frames, as the ACK frame contains timestamp.

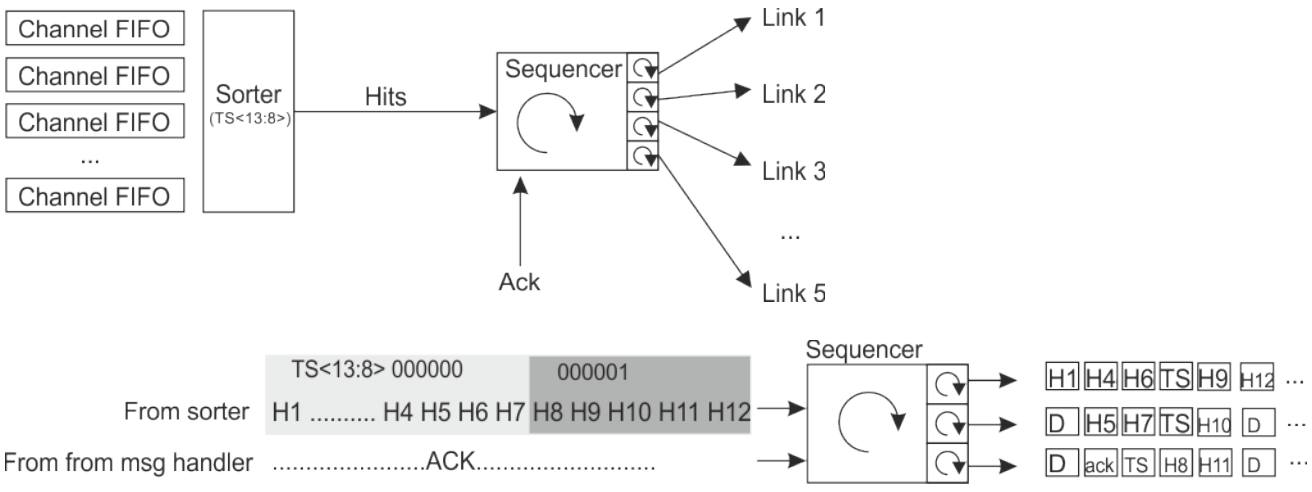


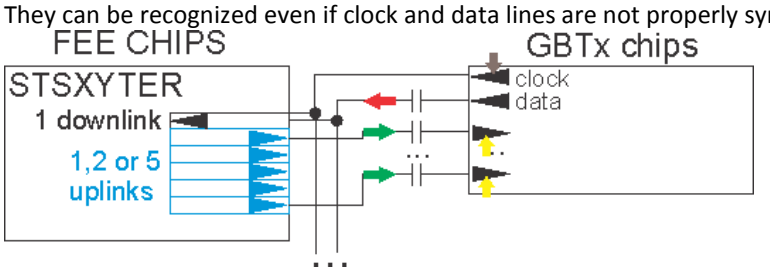
Fig. The way the incoming traffic is distributed onto the link (uplink). Important note: Each link monitors the incoming hits on its own and generates TS_MSB if necessary.

3. Link Synchronization

In the downlink & uplink direction we define two special frames, which are normally forbidden:
Special frames do not use 8b/10b coding. They use a DC-balanced code not allowed by 8b/10b coding (8b/10b allows max 5 consecutive “1” or “0”) which makes it easy to detect without frame synchronization and even in case of bad clock-data phase alignment. They are used for link initialization sequence

SOS Start of synchronization	00000111111111100000 (10 “1”, 10 “0”)
EOS End of synchronization	11001111110000001100

They can be recognized even if clock and data lines are not properly synchronized.



Overview:
- downlink clock / data phase (GBT chip settings): set by DPB based on presence or not SOS in STSXYTER’s response.
- uplink data delay: by DPB based on K.28.1 character transmitted by STSXYTER during synchronization procedure.
- STSXYTER bit alignment: based on first comma received after EOS.
- DPB bit alignment: based on K.28.1

FULL LINK SYNCHRONIZATION

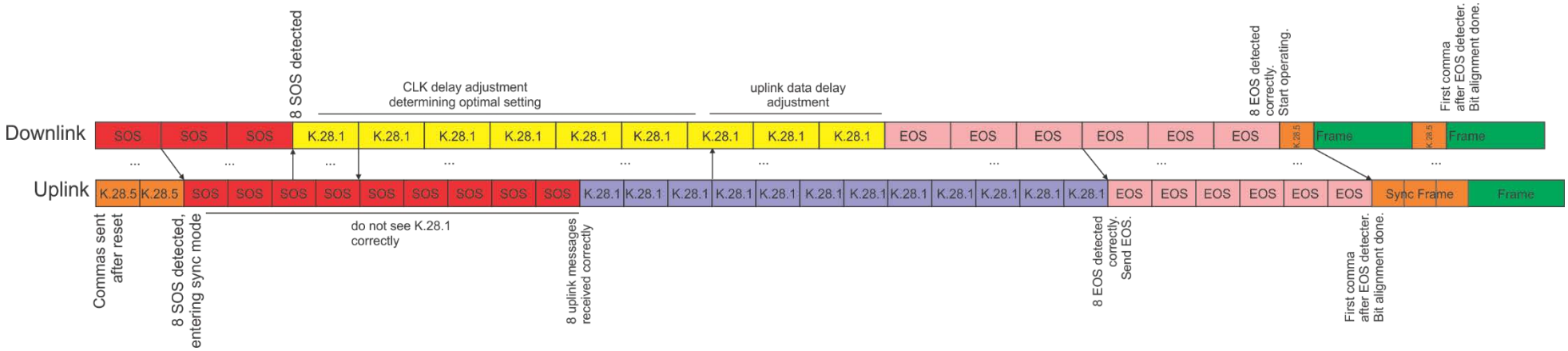
- STSXYTER after reset keeps on sending **K.28.5** character.
- DPB triggers the synchronization procedure by sending **SOS** characters.
- When STS XYTER receives **SOS**, it enters the synchronization state, and responds with the same sequence **SOS**.
- When DPB receives **SOS** from all connected STS XYTERS, it stops sending the **SOS**, and starts to send a comma character **K.28.1.** During this time, depending on the STSXYTER response it adjusts the clock / downlink data phase. The DPB determine in what range of clock delay the STSXYTER receives the correct message (the DPB does not need to adjust the data phase since SOS message can be detected without full link synchronization). STSXYTER responds with:
  - **K.28.1** (if it understands K28.1 correctly) or
  - **SOS** (if it does not understand K.28.1 correctly)
- After the above is completed, the DPB starts to adjust the uplink data delay to determine the optimum data delays to receive the uplink **K.28.1** message correctly.
  When synchronization is completed, the DPB repeatedly sends **EOS** sequence, and STS XYTERs respond with the same **EOS** sequence.
- DPB sees **EOS** response from all STSXYTERs, stops sending **EOS** and immediately starts to operate link in a normal way.
- STSXYTER does the bit alignment on first comma character after **EOS**. When it happens it starts operating in a regular way.
- Note: During normal operation, STSXYTER monitors arrival of K.28.5 comma character and if it is detected out of sync a Sync Alert message is sent to DPB.

QUICK LINK SYNCHRONIZATION

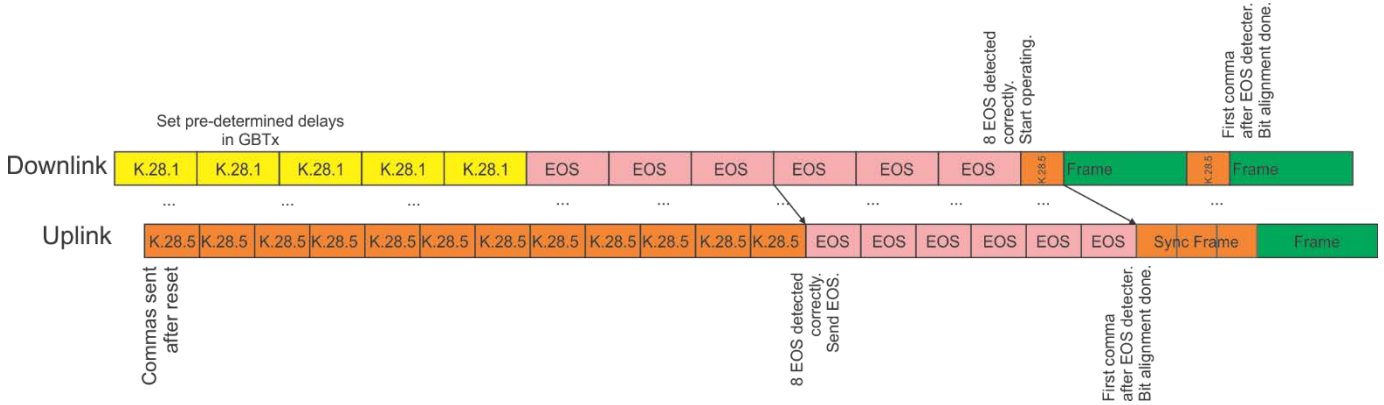
When the optimal delays are already known quick link synchronization procedure can be used for bit alignment.

- STSXYTER after reset keeps on sending **K.28.5** character.
- DPB sends a pre-determined configuration to the GBTx ASIC.
- DPB sends **EOS** sequence. STSXYTER responds with **EOS** as well.
- DPB sees **EOS** response from STSXYTER, stops sending **EOS** and immediately starts to operate link in a normal way.
- STSXYTER does the bit alignment on first comma character after **EOS**. When it happens it starts operating in a regular way.

Regular link synchronization procedure



Quick link synchronization procedure



Appendix A. STSXYTER Address space

Analog Front-End registers

Reg_address	Device type	description
Address structure (accessible through e-link)		
Address<13:0> = {6'd *col* , 8'd *row*}		
***row* in the range of 0 – 129 (Physical channels: 0..127, Test channels: 128,129)**		
***col* =0,2,4..60**	Counter (R)	12-bit ADC cell counter, 0 - the highest threshold comparator, 60 - the lowest threshold comparator. (31 counters in total per channel)
Col=62	Dice cell (R/W)	<7> channel disable (default: 0, channel enabled) <6> reserved <5:0> Trimming DAC (fast comparator)
***col* =1,3,5…61**	Dice cells (R/W)	8-bit ADC comparators trimming DACs, 1 - the highest threshold comparator, 61 - the lowest threshold comparator (typ: 128)
***col* =63**	Dice cells (R/W)	12-bit counter (fast comparator)
***row* = 130 (Global DACs address space)**		
***col* =0**	Dice cells (R/W)	6-bit DAC: : <5:0> CSA bias current (CSA_IN_REF!) (typ: 45) 0 - 0.5 mA 31 - 1.0 mA 63 - 1.5 mA
***col* =1**	Dice cells (R/W)	6-bit DAC: : <5:0> CSA feedback resistance R_F (ifed), (typ: 0) 0 - 255MΩ 14 - 10MΩ 63 - 2MΩ The resistances are given for I_{leak} = 0.
***col* =2**	Dice cells (R/W)	Mode selection: <6> pulse stretcher enable (PS_en): 0 - disabled, 1 - enabled <5> polarity (POL): 0 - electrons; 1 - holes <4:3> correction of the leakage current induced DC level shift at the CSA output (in_csap), (typ: 2) 0 - 4 µA (I_{leak} = -15nA, ifed=17, R_F=10MΩ) 2 - 16 µA (I_{leak} = 0) 3 - 62 µA (I_{leak} = 15nA, ifed=30, R_F=10MΩ) <2:0> polarity selection bias current (i_core), (typ: 3) 0 - 30 µA 3 - 70 µA 7 - 100 µA Typical: 19 for negative, 51 for positive pulses.
***col* =3**	Dice cells (R/W)	6-bit DAC: <5:0> bias current of slow and fast shaper amplifiers core (in_sh), (typ: 24) 0 - 100 µA 31 - 200 µA 63 - 300 µA
***col* =4**	Dice cells (R/W)	8-bit <7:0> calibration pulse amplitude, 0-15 fC, (calib_amplitude) (typ: 0)
***col* =5**	Dice cells (R/W)	<1:0> calibration channel group select (cal_channel_group_sel): 0 - channels: 0,4,...,124, + 129 (test) 1 - channels: 1,5,...,125 2 - channels: 2,6,...,126 3 - channels: 3,7,...,127, +128 (test)
***col* =6**	Dice cells (R/W)	<6:0> ADC high threshold (note: the ADC input signal is inverted), 875 mV (typ: 24)
***col* =7**	Dice cells (R/W)	<6:0> ADC low threshold (note: the ADC input signal is inverted), 1150 mV (typ: 40)
***col* =8**	Dice cells (R/W)	6-bit DAC <5:0> Reference current for the high speed discriminator (discr_iref_glob) (typ: 32)
***col* =9**	Dice cells (R/W)	8-bit DAC <7:0> High speed discriminator threshold (discr_th2_glob) (typ: 111)
col=10	Dice cells (R/W)	8-bit DAC <7:0> Global ADC threshold (VREF_T) (typ: ???) (new)
col = 11..63		Forbidden addresses

Register File registers:

Reg_address	Device type	description
Address structure (accessible through e-link)		
***Row=192* (registers in the register file -> synthesized in digital part)**		

Col=1	TIMESTAMP	Register (R/W)	14-bit value of the timestamp counter Gray encoded
Col=2	CMD_SOFT_RESET	Register (W/R)	Writing '1' to each bit gives the corresponding chip part reset: <0> - datapath output fifo reset <1> - channel fifos reset <2> - data latch stage reset <3> - front-end channel reset <4> - timestamp reset (load with 0) Reading results in 14'x1 value. Other values: reserved.
Col=3	CONFIG	Register (R/W)	<7:2> reserved <0> - channel mask enable (def: 0 disabled) Enables channel masking feature <1> - enable timestamp in the ACK frame (def: 0 – disabled)
Col=4..13	MASK_13_0 MASK_27_14 MASK_41_28 MASK_55_42 MASK_69_56 MASK_83_70 MASK_97_84 MASK_111_98 MASK_125_112 MASK_129_126	Register (R/W)	14-bit channel mask register (for throttling). Channel is disabled when corresponding bit is set to 1 Col=4 , bits<13:0> : mask channels 13 through 0 (def: 0) Col=5, bits<13:0>: mask channels 27 through 14 (def: 0) Col=6, bits<13:0>: mask channels 41 through 28 (def: 0) Col=7, bits<13:0>: mask channels 55 through 42 (def: 0) Col=8, bits<13:0>: mask channels 69 through 56 (def: 0) Col=9, bits<13:0>: mask channels 83 through 70 (def: 0) Col=10, bits<13:0>: mask channels 97 through 84 (def: 0) Col=11, bits<13:0>: mask channels 111 through 98 (def: 0) Col=12, bits<13:0>: mask channels 125 through 112 (def: 0) Col=13, bits<3:0>: mask channels 129 through 126 (def: 0)
Col=14	TS_RESET_VAL	Register (R/W)	<13:0> Timestamp value to sync Keeps a value that will be written into the timestamp counter during sync command.
Col=15	CMD_TS_LOAD	Register (W)	Writing FF results in writing content of TS_RESET_VAL (C14_R192) register into the timestamp counter.
Col=16	DAQ_CTRL	Register (R/W)	DAQ control register <0> - data acquisition enabled (typ: 0 disabled) blocks data at the level of channel fifo input. Note: what about event missed which may occur in the middle?
Col=17	CMD_FLUSH_FIFO	Register (W)	Flush fifo register Writing FF results in flushing all channel fifos.
Col=18	TEST_CTRL1	Register (R/W)	Test_Ctrl 1. Control register of test stimulus. Test stimulus cuts-off the channels from back-end and generates fake hits according to the variant. <0> - TEST_MODE_#0 - raw data streaming test enable (def: 0 – disabled). The output links are flooded with unformatted frames of incrementing bytes. <1> - TEST_MODE_#1 - hit data streaming test enable (def: 0 – disabled). The chips streams out hit data with a rate and randomization controlled by TEST_CTRL2 register. <7:2> - spare
Col=19	TEST_CTRL2	Register (R/W)	Test_Ctrl 2. Control parameters for the TEST_MODE_#1 test <4:0> - generated hit rate control (def: 0). The chip hit rate is hit_rate = 160e6 / 3 / 32 * 5_bit_value (for 160 MHz input clk) This covers the range from 1.(6) to 51.(6) Mhit/sec/chip (13,02 to 403.6 khit/sec/channel). The timestamp value of the hit is taken from the timestamp counter (binary value with LSB fixed to 0), channel address = 127, energy = 31, missed = 0 <7:5> - number of timestamp LSB bits to randomize (def: 0). Selected number of timestamp LSB bits are randomized using LFSR with poly = $x^8 + x^6 + x^5 + x^4 + 1$
Col=20	TEST_CTRL3	Register (R/W)	-- spare --
Col=21	MONITOR_REF	Register (W/R)	Monitor input reference <7:0> sets reference level of the monitor input
Col=22	-- spare --		
Col=23	FIFO_AFULL_THR	Register (W/R)	<8:0> Threshold channel fifo almost full flag count. When the number of the almost full flags exceeds the threshold, a bit is set in the configuration register and the FIFO_AFULL_COUNT counter is incremented (def: 0)
Col=24	FIFO_AFULL_COUNTER	Register (R/W)	<11:0> Number of "FIFO almost full" registered notifications. Writing 0 resets the counters.

Col=25	ELINK_MASK	Register (W/R)	E-link selection <4:0> e-link disable soft mask (block data sending on the link, send dummy hits and sync frames), <9:5> e-link disable hard mask (output of the masked link is stuck to zero) def: 10'b00000000000 – all links enabled. Example: 10'b0000000011 – no data transferred on links #0 and #1 Example: 10'b0001100000 – chip serial outputs #0 and #1 stuck to 0.
Col=26	LAST_ADDR	Register (R/W)	Address used for previous register access. Set by WRaddr frame and RDdata frame.
Col=27	STATUS	Register (R/W)	Information about the chip status. The bits once set to 1 remain until overwritten with the register write command. <0> **dp_fifo_throttling_alert** - set when the number of fifos in the channels loaded over a certain value is over the threshold value set in the FIFO_AFULL_THR register <1> **link_sync_alert** - set when K28.5 arrives in the middle of the frame <2> **fe_chan_trottling_alert** - set when number of EM (event missed hits in the fe_channels) exceeds a threshold value set in FE_EM_THR register <3> **cd_crc_error** - set when a CRC error was detected on the downlink data <4> **cd_fifo_full** - set when command decoder output fifo overflows <5> **cd_data_collector_error** - internal error in the command decoder, ACK and RDdata may be corrupted <6> **thg_fifo_full** - set when test hit generator output fifo overflows <7> **monitor_over_threshold** - the value monitored by the c21r92 register is higher then the register content (exceeded the threshold) <8> **dp_fifo_throtl_alert_counter_full** – FIFO_AFULL_COUNTER full <9> **fe_chan_throtl_alert_counter_full** – FE_EM_COUNTER full <10> **seu_counter_full** – SEU_COUNTER full
Col=28	STATUS _MASK	Register (R/W)	Mask for each status bit. The status bit is not reported in the ACK frame when corresponding mask bit is set to 1.
Col=29	FE_EM_THR	Register (W/R)	Threshold for front-end channels' event missed count. When the current number of the event-missed set in the channels is higher than the threshold, then a bit is set in the status register (def: 0)
Col=30	FE_EM_COUNTER	Register (R/W)	<11:0> Shows the number of times the count of event-missed flags exceeded the threshold. (def: 0) Writing 0 resets the counter
Col=31	SEU_COUNTER	Register (R/W)	<11:0> Shows the number of SEUs in the selected registers (marked with SEU_CNT in the table).

Appendix C.

Timestamp handling

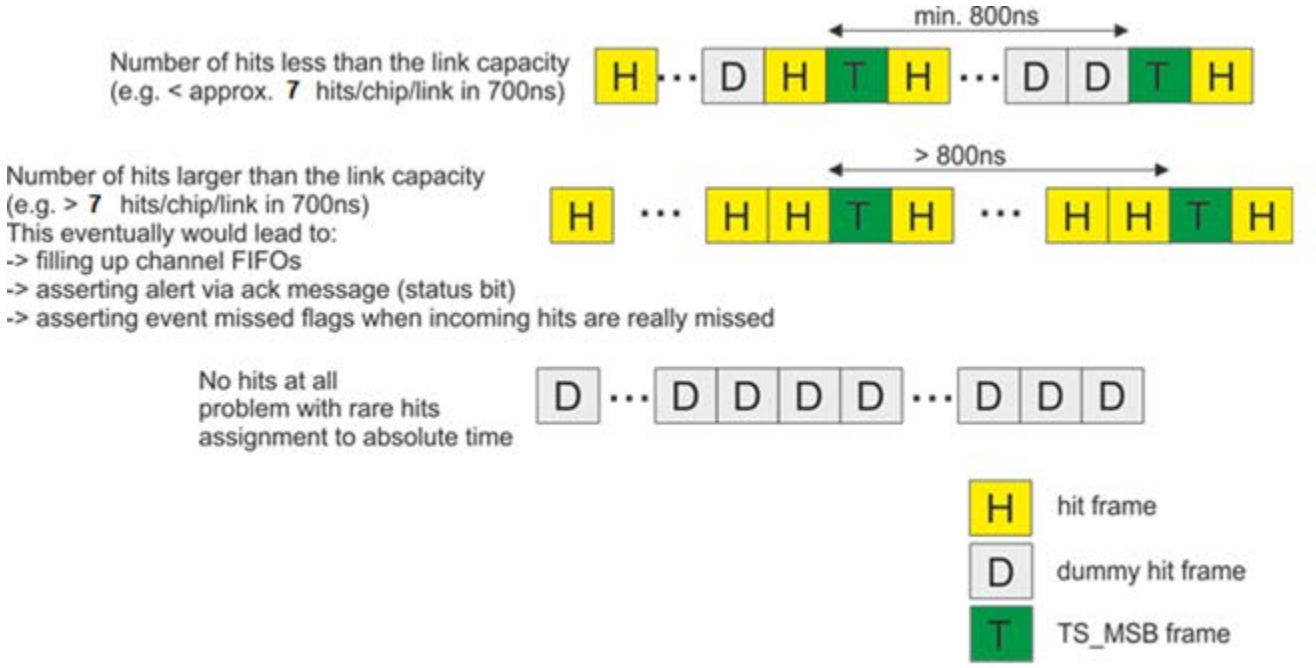


Fig. Data on link in three cases: no hits at all, number of hits within the link bandwidth, number of hits above the link bandwidth.

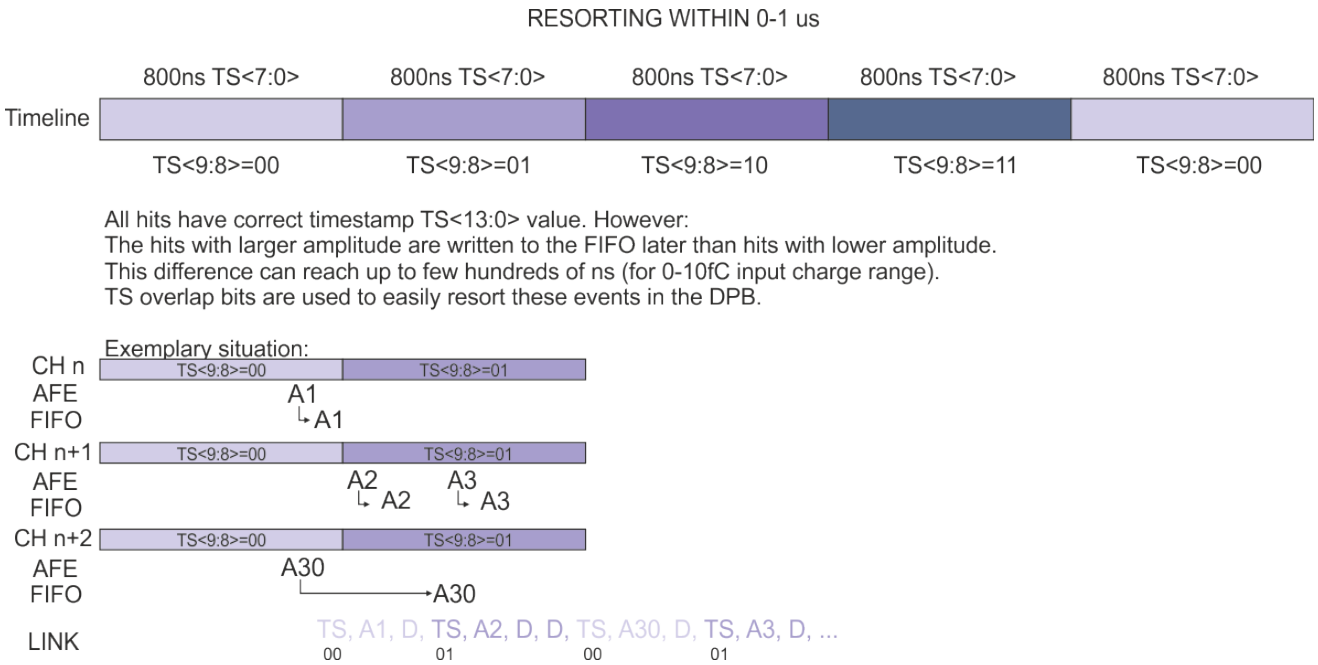


Fig. Uncertainty of hit arrival time at FIFO input and the way it appears at the link.

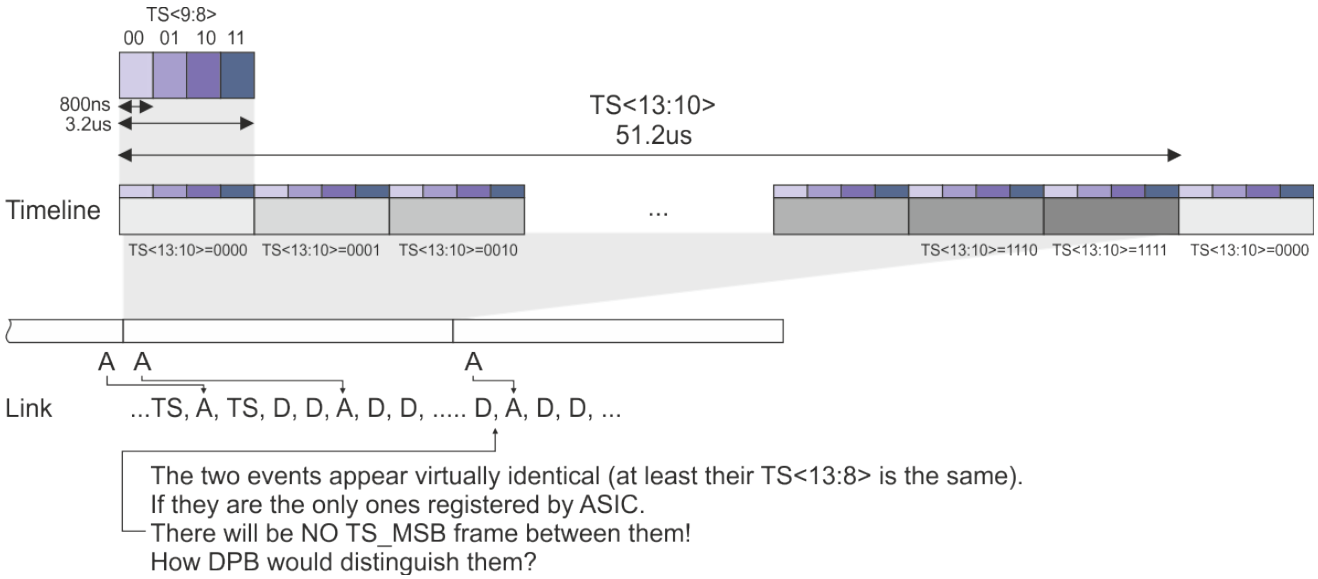


Fig. Definition of possible uncertainty of two alone hits spaced in time by approx. 51.2 us (2^14 * TS LSB)

Considered options:

1. ~~Dummy hits contain no additional information, no additional TS_MSB frames are sent. DPB keeps a mirror Timestamp counter (to be aware of the time passing in the ASIC). Two counters are synchronized by a sync message. The ASIC's channel FIFO length is adjusted depending on the number of links (to keep roughly constant minimum time to stream out hits from a completely filled-out ASIC) and avoid situation that the stream-out time is close to or longer than 51.2 us. In this way, by merging information from TS counter mirror together with the lack of TS_MSB frames the hits can be properly assigned.~~
2. **Dummy hits in the TS field will contain TS<13:8> bits. The purpose is just to keep DPB aware of the time passing in the ASIC. The TS_MSB frame is still generated based on hits only while the dummy hits will send actual value of TS counter.**
3. ~~Once every 8 consecutive dummy hits (approx. every 800ns) there will be TS_MSB frame transmitted with current TS <13:8> value.~~

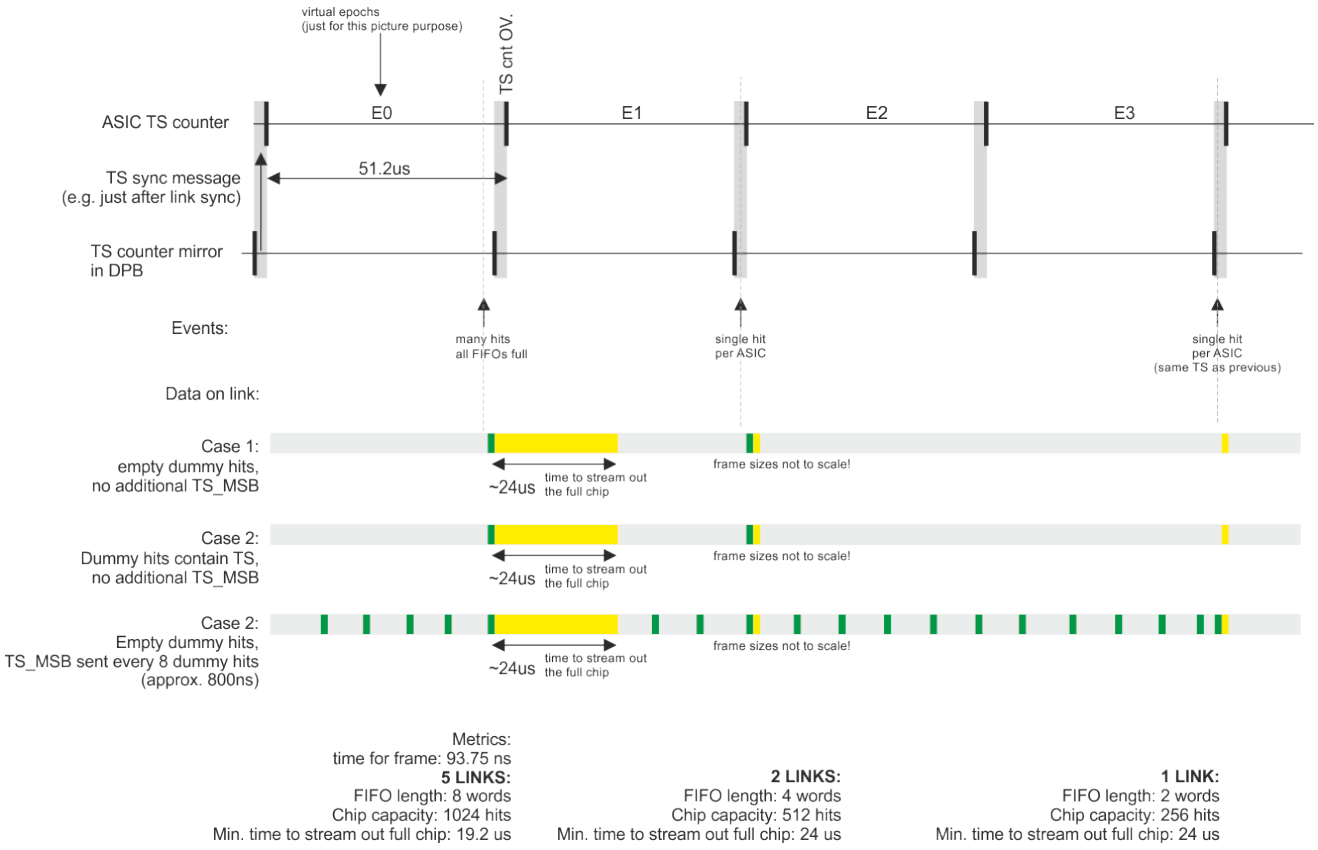


Fig. Proposal of solving problem with correct hit assignment within larger timeline. (Case 2 was selected finally)