

WCH CH569 SerDes Reverse Engineering

Table of Contents

1	Introduction.....	2
2	Hardware Setup.....	3
3	Software Setup.....	4
3.1	glscopeclient configuration.....	4
3.1.1	Filter Graph.....	5
3.1.2	Main view.....	5
3.1.3	Export 8b10bIBM_Hex to csv.....	5
4	WCH CH569W SerDes Reverse Engineering.....	6
4.1	SerDes Idle sequence (no data).....	7
4.2	SerDes data sequence.....	8
4.2.1	SOF.....	8
4.2.2	Align.....	8
4.2.3	EOF.....	9
4.3	SerDes 180Mbps data capture and details.....	10
5	Appendix.....	11
5.1	SATA specification.....	11
6	Revision History.....	12

1 Introduction

The WCH CH569W is the first public MCU including a [SerDes](#) PHY (Serializer/Deserializer) which can be configured at up to 1.20Gbps officially and up to 1.38Gbps non officially.

It is a very convenient interface as it requires only 2 wires (one differential pair like [CAT 5E](#) pair) to communicate at very high speed with external world (mainly FPGA or other CH569).

It is also very interesting as this interface support network cable transfer distance of 90 meters.

Unfortunately WCH CH569W SerDes is not documented and not even present in latest examples available on github (<https://github.com/openwch/ch569> on 31 July 2022).

The SerDes Example available in CH569EVT_v1_4 (or before) is not available anymore and it was using "binary blob/static library" with very basic example (not supporting any speed configuration of the SerDes link which is by default configured at 1.2Gbps).

- CH569EVT with SerDes example is available on
<https://github.com/SoCXin/CH569/tree/master/src/EVT/EXAM/SERDES>

After lot of reverse engineering (on firmware side & analog side with oscilloscope) and with the help of multiple SerDes captures and check of data (with the help of glscopeclient and Andrew Zonenberg hints) the fact is CH569W SerDes is based on Serial ATA (SATA) specification, especially related to "SATA Primitives" used mainly for Idle Sequence and Data sequence and also related to scrambler and CRC32.

For more details with example captures and tools used in this document see following GitHub links:

- Main repository related to that document / research (including tool with source to descramble the glscopeclient csv data exported)
 - <https://github.com/hydrausb3/wch-ch569-serdes>
- HydraUSB3 SerDes example/code firmware (can be used with only one board too)
 - https://github.com/hydrausb3/hydrausb3_fw/tree/main/HydraUSB3_DualBoard_SerDes

2 Hardware Setup

All the captures are done on Development Board called HydraUSB3 V1 (available on <https://hydrabus.com/shop> or on **Digi-Key** <https://www.digikey.com/en/products/detail/benjamin-vernoux/HYDRAUSB3-V1/16651677>) and generated by WCH CH569W SerDes configured at 1.08Gbps & 180Mbps (SerDes firmware driver / examples released with open source license on <https://github.com/hydrausb3>)

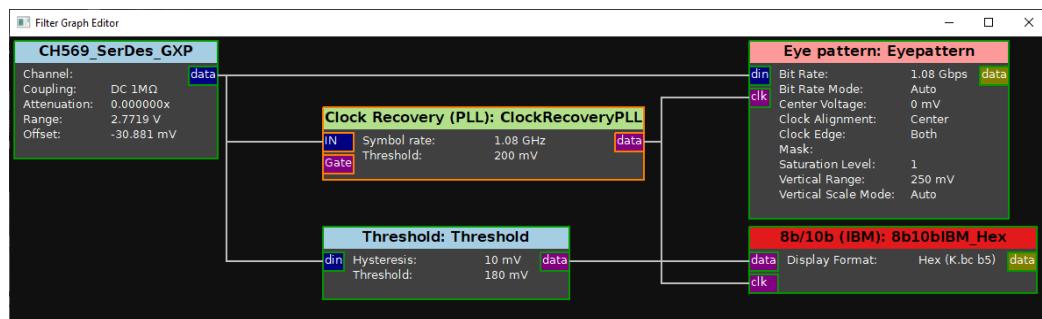
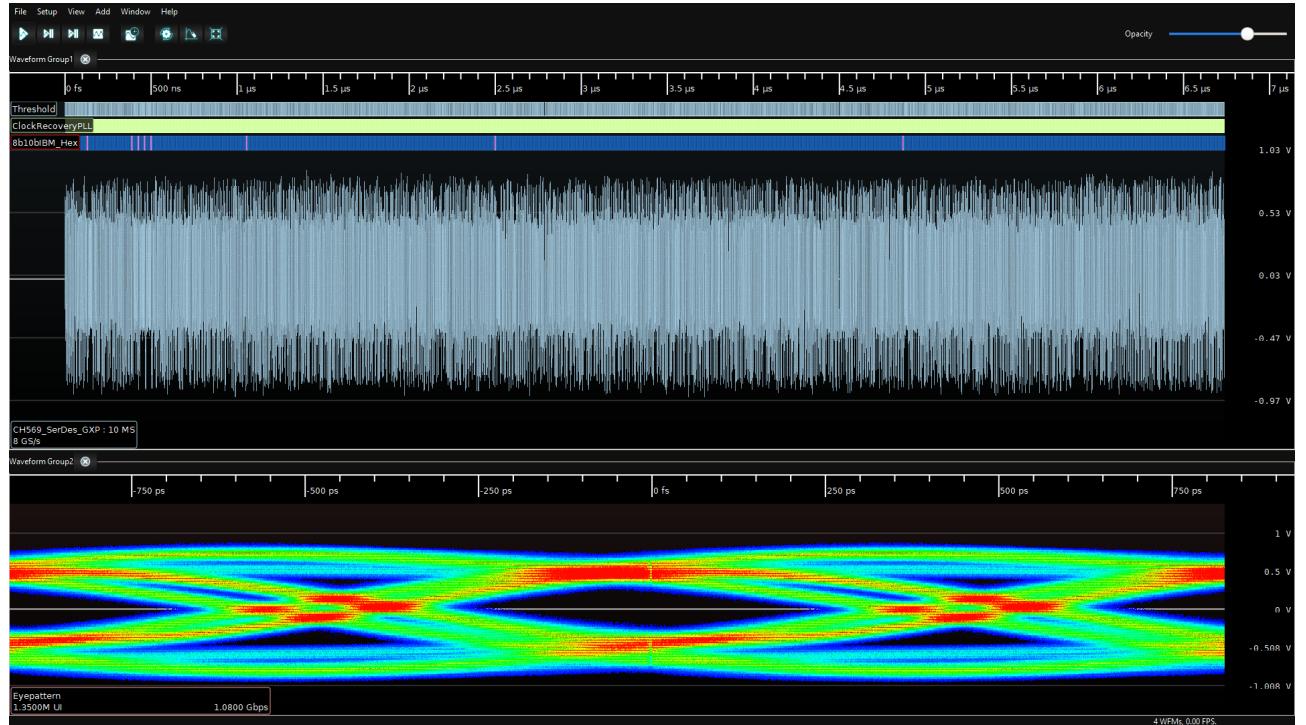
The CH569W SerDes captures are done on GXP pin with a Rigol MSO5K and using original probes with ground spring (mandatory to have a good capture) then exported to bin (to an USB key) and imported with open source glscopeclient (<https://github.com/glscopeclient/scopehal-apps>)

- It is recommended to use glscopeclient from 30 July 2022 (commit [a896eca](#))
 - See direct link <https://github.com/glscopeclient/scopehal-apps/actions/runs/2764516663>
 - There is a known issue related to recentfiles.yml (updated/corrupted when glscopeclient exit) on that version see <https://github.com/glscopeclient/scopehal-apps/issues/460>
 - Solution on Windows is to delete (xxx is user) C:\Users\xxx\AppData\Roaming\glscopeclient\recentfiles.yml before to start glscopeclient

3 Software Setup

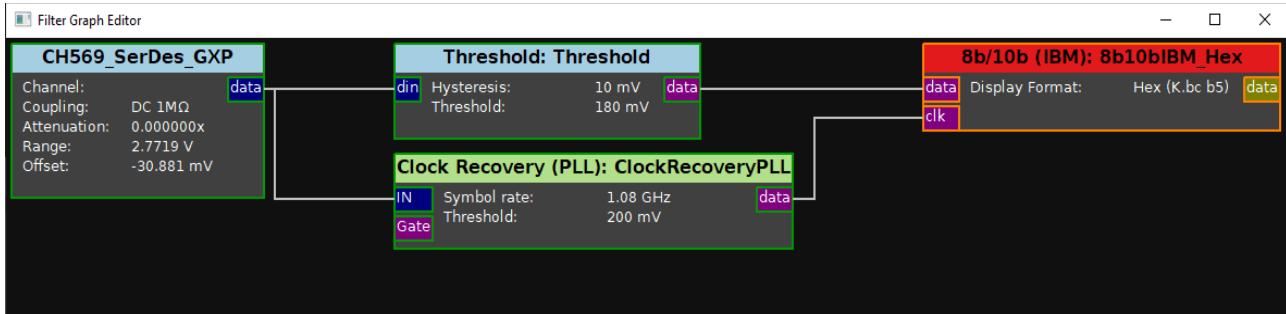
3.1 glscopeclient configuration

The "Threshold" view & "ClockRecoveryPLL" Thresholds (Hysteresis of Threshold can be adjusted too) shall be adjusted to have best results with less error in **8b10bIBM_Hex** view
To be sure the data can be recovered you shall check the signal integrity with Eye Pattern which shall have an "eye" as wide open as possible.



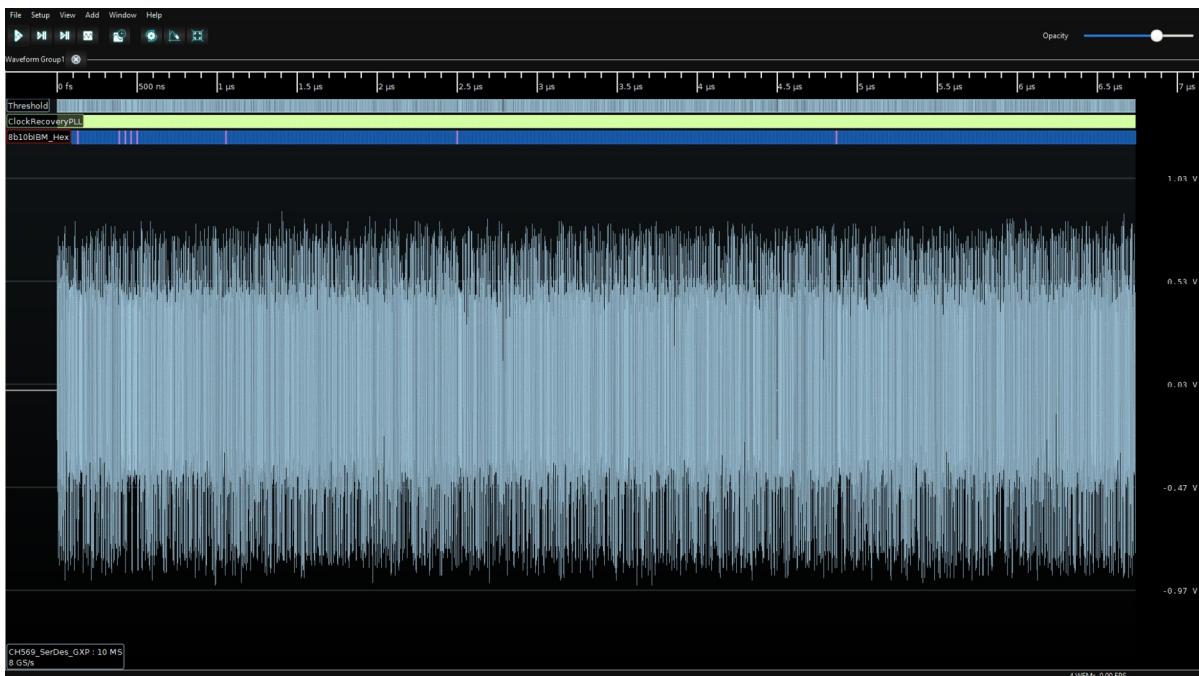
3.1.1 Filter Graph

Filter graph example minimal configuration to export 8b10IBM_Hex to csv

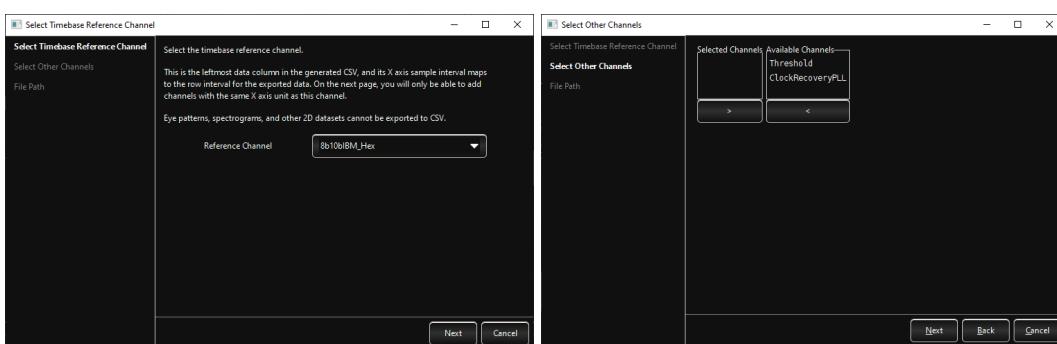


3.1.2 Main view

Main view example minimal filter configuration to export 8b10IBM_Hex to csv



3.1.3 Export 8b10bIBM_Hex to csv



Then click next and choose a name for the csv file to export (example **8b10bIBM_Hex.csv**)

4 WCH CH569W SerDes Reverse Engineering

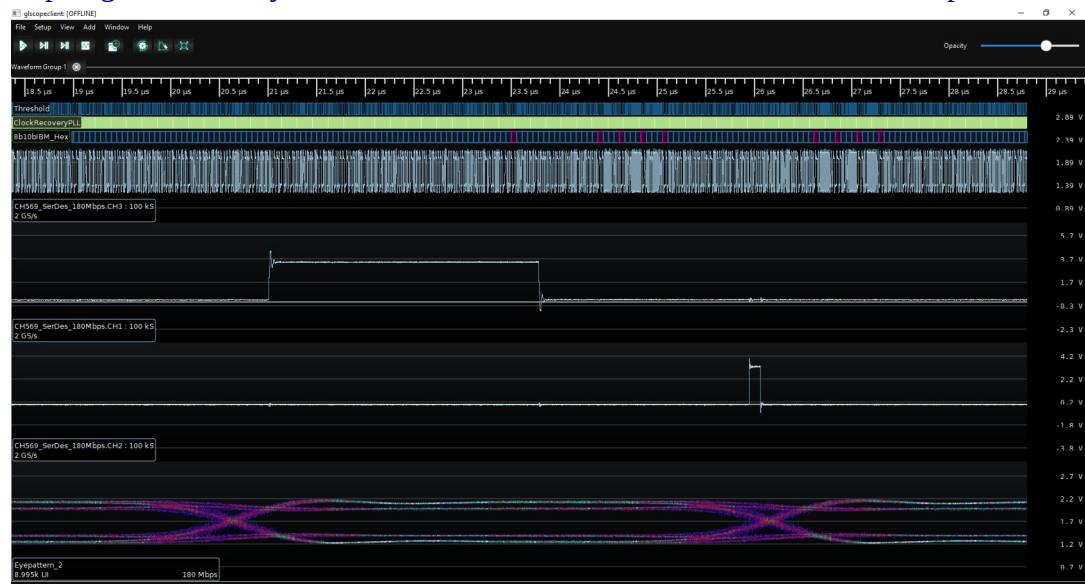
This chapter describes the data format of WCH CH569W SerDes captured with glscopeclient with the help of Serial ATA specification (See Appendix for links).

The SerDes on WCH CH569W have different states:

- SerDes Idle sequence (no data) is transmitted continuously when we are not sending any data.
- SerDes data sequence is when data are transmitted.
 - The Serdes data sequence start by a SOF and finish by an EOF

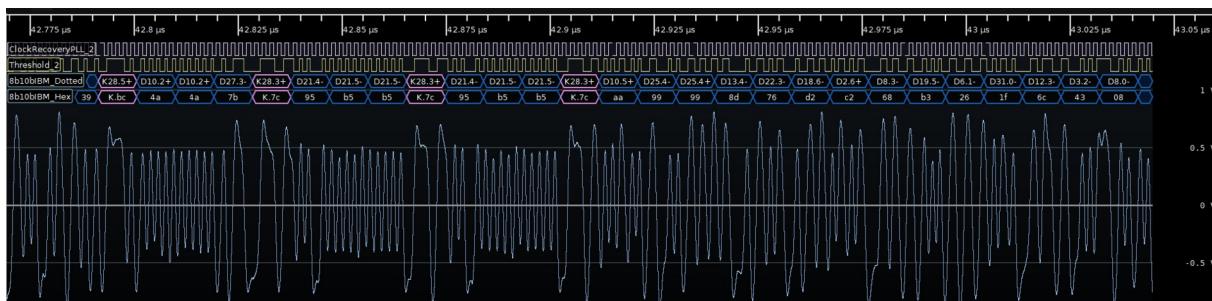
Example files (CH569 SerDes at 180Mbps) available on GitHub:

- https://github.com/hydrausb3/wch-ch569-serdes/CH569_SerDes_180Mbps

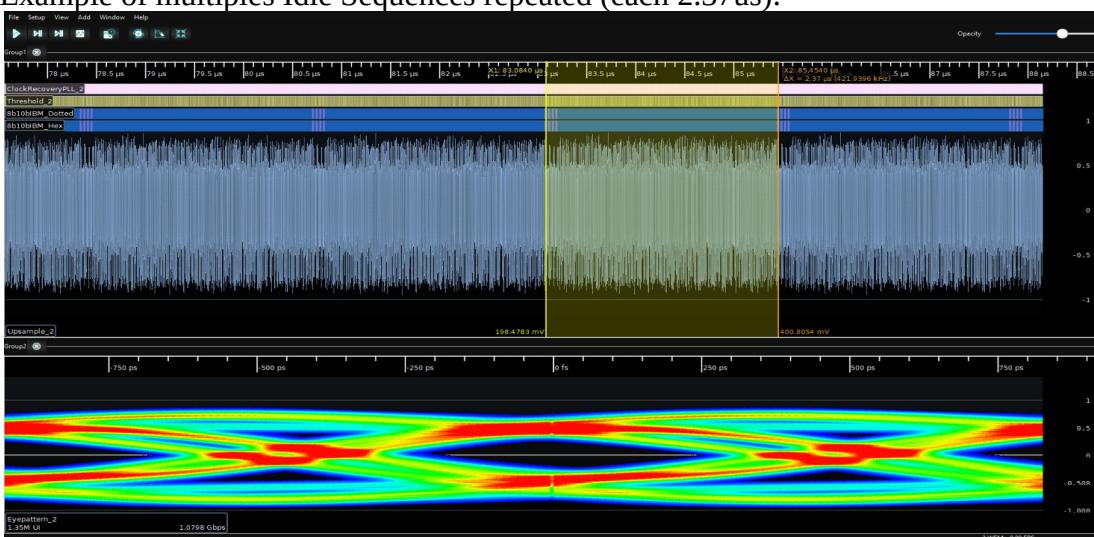


4.1 SerDes Idle sequence (no data)

- 1x K28.5 D10.2 D10.2 D27.3 => 1xALIGN
 - 1x K.bc 4a 4a 7b
 - The Align primitives establish dword boundaries within the serial bit stream.
 - They are sent as data during OOB, and every 2048 dwords there after
 - Extract Serial ATA Specification Revision 1.0 29-August-2001
- Table 20. – Description of primitives:
- Upon receipt of an ALIGN, the physical layer readjusts internal operations as necessary to perform its functions correctly.
 - This primitive is always sent in pairs - there is no condition where an odd number of ALIGN primitives shall be sent (except as noted for retimed loopback).
- 2x K28.3 D21.4 D21.5 D21.5 => 2xSYNC
 - 2x K.7c 95 b5 b5
 - Used when in idle to maintain bit synchronization.
 - Extract Serial ATA Specification Revision 1.0 29-August-2001
- Table 20. – Description of primitives:
- Synchronizing primitive – always idle.
- 1x K28.3 D10.5 D25.4 D25.4 => CONT
 - 1x K.7c aa 99 99
 - Extract Serial ATA Specification Revision 1.0 29-August-2001
- Table 20. – Description of primitives:
- The CONT primitive allows long strings of repeated primitives to be eliminated.
 - The CONT primitive implies that the previously received primitive be repeated as long as another primitive is not received



Example of multiples Idle Sequences repeated (each 2.37us):



4.2 SerDes data sequence

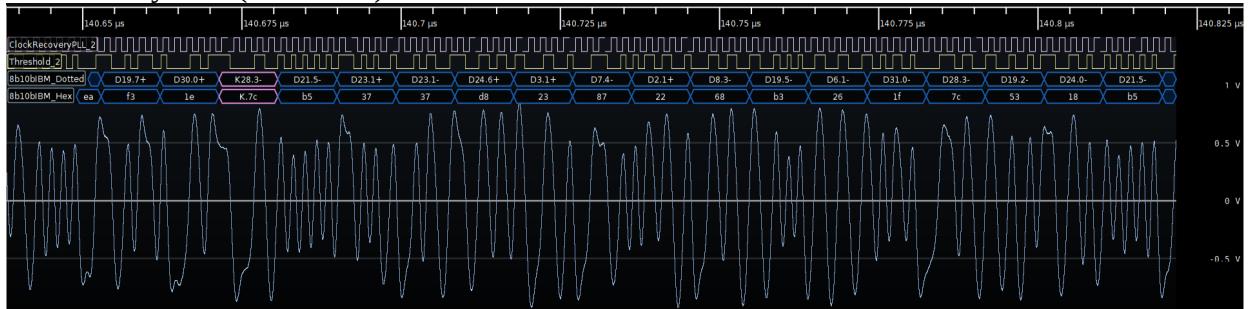
4.2.1 SOF

A frame starts with a Start of Frame primitive (SOF).

The frame ends with a CRC dword and an End of Frame primitive (EOF).

In between the two primitives are dwords that make up the information content

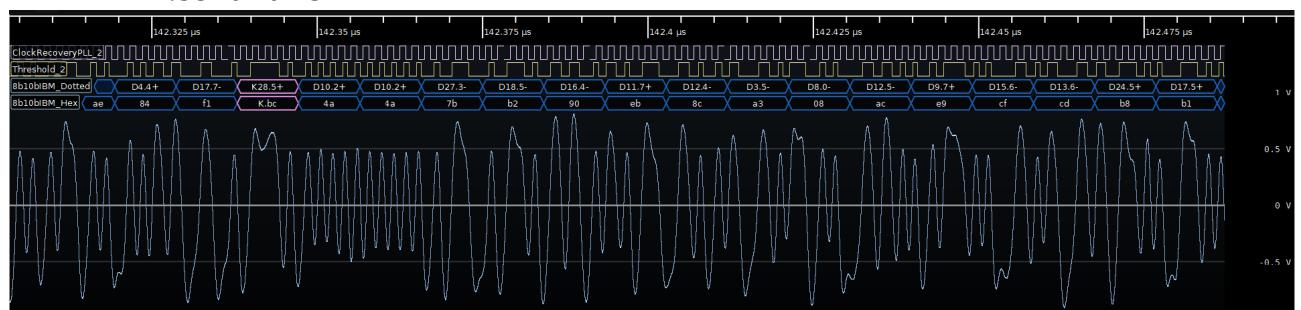
- 1x K28.3 D21.5 D23.1 D23.1 => 1xSOF
- 1x K.7c b5 37 37
- Followed by data (until EOF) scrambled like SATA format



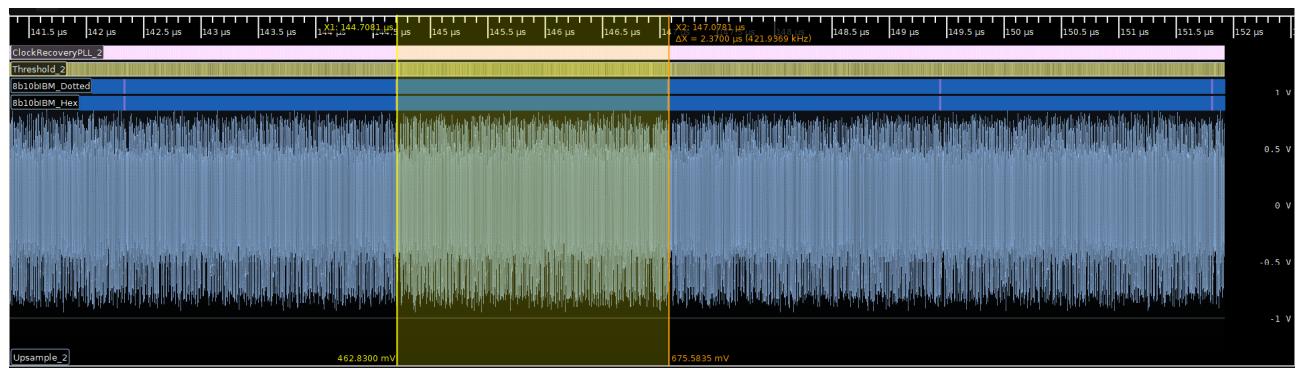
4.2.2 Align

Regularly (during data exchange)

- 1x K28.5 D10.2 D10.2 D27.3 => ALIGN
- 1x K.bc 4a 4a 7b



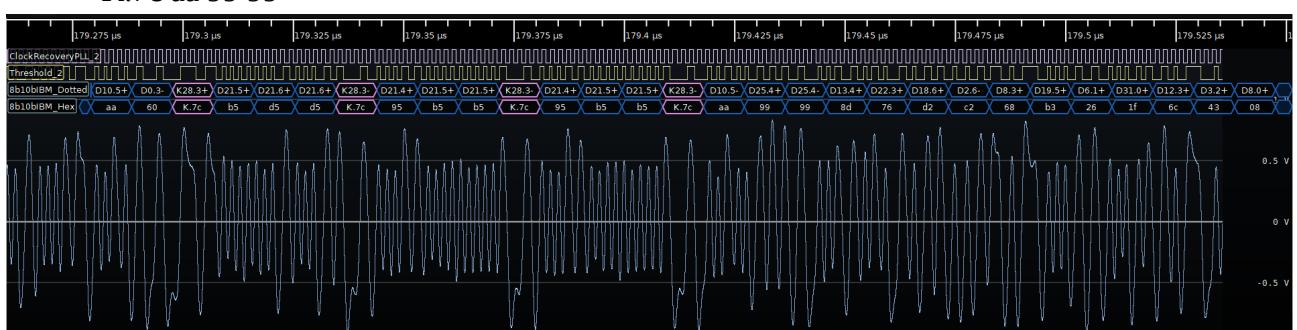
Example of multiples ALIGN repeated (each 2.37μs):



4.2.3 EOF

End Of Frame sequence

- 1x K28.3 D21.5 D21.6 D21.6 => 1xEOF
- 1x K7.c b5 d5 d5
 - Extract Serial ATA Specification Revision 1.0 29-August-2001
- Table 20. – Description of primitives:
 - EOF marks the end of a frame. The previous non-primitive Dword is the CRC for the frame.
- 2x K28.3 D21.4 D21.5 D21.5 => 2xSYNC
- 2x K.7c 95 b5 b5
- K28.3 D10.5 D25.4 D25.4 => CONT
- K.7c aa 99 99



4.3 SerDes 180Mbps data capture and details



Details:

- SerDes SATA SOF (compliant with SATA CRC32 spec) start at 23.50us
 - SATA SOF => Hex: 7C B5 37 37
- SerDes DATA (HDR / PAYLOAD DATA / CRC32)
 - HDR / Header (specific to WCH CH569W) start at 23.723us
 - Scrambled data => Hex: 72 89 2D CD
 - Descrambled data => Hex: 0F FF FF FF
 - 4bits = Packet Number (incremented after each packet transmitted, start from 0x0 up to 0xF then restart to 0x0 and so on)
 - 28bits = Packet Identifier configured with CH569W SerDes Register
 - PAYLOAD data start at 23.945us
 - Scrambled data => Hex: 32 E9 7C 45
 - Descrambled data => Hex: 5A 5A 5A 5A
 - CRC32 checked as OK(compliant with SATA CRC32 spec) start at 24.168us
 - Scrambled data => Hex: 3E 04 3C B0
 - Descrambled data => Hex: 15 34 47 52
- SerDes SATA EOF (compliant with SATA CRC32 spec) start at 24.39us
 - SATA EOF => Hex: 7C B5 D5 D5

5 Appendix

5.1 SATA specification

See https://www.seagate.com/support/disc/manuals/sata/sata_im.pdf

Most interesting chapter for CH569 SerDes are:

- § 7.4 Primitives (mainly related to SOF/EOF)
- § 7.5 CRC calculation and scrambling of FIS contents
- APPENDIX A. SAMPLE CODE FOR CRC AND SCRAMBLING

See extract of § 7.4.3 Primitive encoding

7.4.3 Primitive encoding

The following table defines the encoding for each primitive.

Table 21 – Primitive encoding

Primitive name	Byte 3 contents	Byte 2 contents	Byte 1 contents	Byte 0 contents
ALIGN	D27.3	D10.2	D10.2.	K28.5
CONT	D25.4	D25.4	D10.5	K28.3
DMAT	D22.1	D22.1	D21.5	K28.3
EOF	D21.6	D21.6	D21.5	K28.3
HOLD	D21.6	D21.6	D10.5	K28.3
HOLDA	D21.4	D21.4	D10.5	K28.3
PMACK	D21.4	D21.4	D21.4	K28.3
PMNAK	D21.7	D21.7	D21.4	K28.3
PMREQ_P	D23.0	D23.0	D21.5	K28.3
PMREQ_S	D21.3	D21.3	D21.4	K28.3
R_ERR	D22.2	D22.2	D21.5	K28.3
R_IP	D21.2	D21.2	D21.5	K28.3
R_OK	D21.1	D21.1	D21.5	K28.3
R_RDY	D10.2	D10.2	D21.4	K28.3
SOF	D23.1	D23.1	D21.5	K28.3
SYNC	D21.5	D21.5	D21.4	K28.3
WTRM	D24.2	D24.2	D21.5	K28.3
X_RDY	D23.2	D23.2	D21.5	K28.3

6 Revision History

Date	Revision	Changes
31-July-2022	V1.0	Initial release (written by Benjamin VERNOUX).