Point-to-point protocol exploration

Nayib Boukadida

n.boukadida@nikhef.nl

F. Schreuder

A. Borga

P. Jansweijer

Hogeschool van Amsterdam

W.E. Dolman

Content

- About Nikhef
- Internship assignment
- Structure of communication protocols
- Requirements
- Survey of existing protocols (vendor dependent and independent)
- The Interlaken protocol
- Implementation
- Status
- Conclusion

Nikhef

- National Institute for Subatomic Physics (Nationaal Instituut voor Subatomaire fysica)
- Among others, Nikhef is closely involved in the development of the LHC ATLAS experiment at CERN
- Internship at the Electronic Technology department

Internship assignment

- Large particle detector facilities often require a point-to-point link at a certain stage of the DataAcQuisition (DAQ) chain to connect custom tailored electronics
- As the demand of data processing increases and larger data-sets are to be transported through these links their performance in terms of speed and bandwidth must scale accordingly
- Because of this:

I was assigned the task to research and implement a novel high-speed point-to-point link targeting FPGA communication

Internship assignment

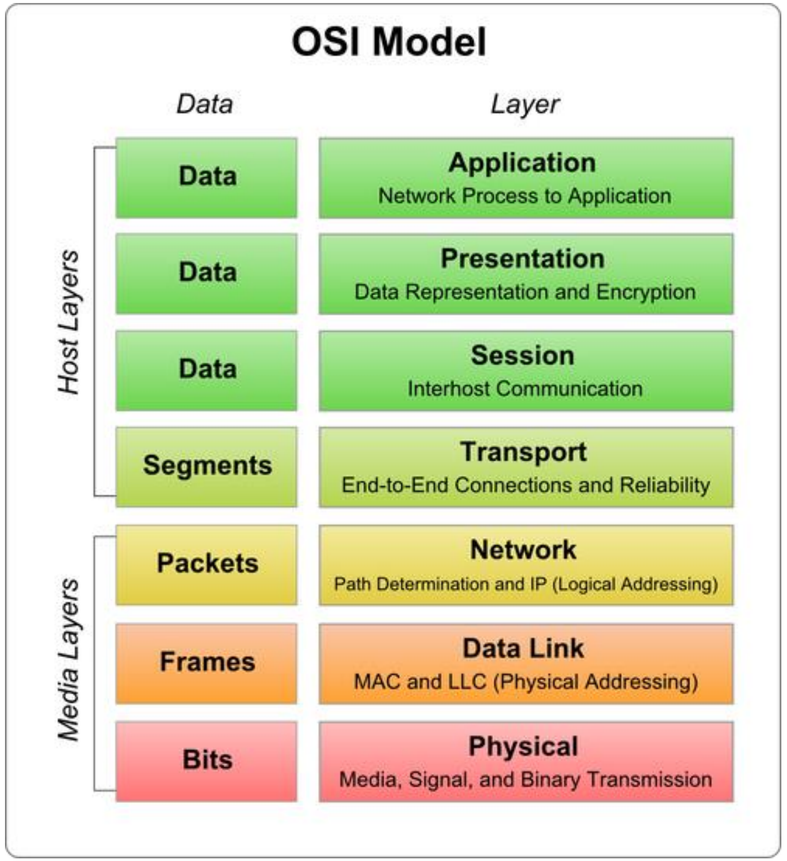
- **Explore** how point-to-point communication protocols are built up
- **Survey** the currently existing point-to-point protocols
- **Find out** if there is an existing protocol that matches a set of clear requirements
- **Describe** the protocol and **implement** a proof of concept that can run on an FPGA [possibly vendor independent]
- **Publish** specifications and implementation as Free and Open Source on specialized code hosting platforms like OpenCores

OpenCores
www.opencores.org

sharing is done with the aim of promoting
the dissemination and broad adoption of the design

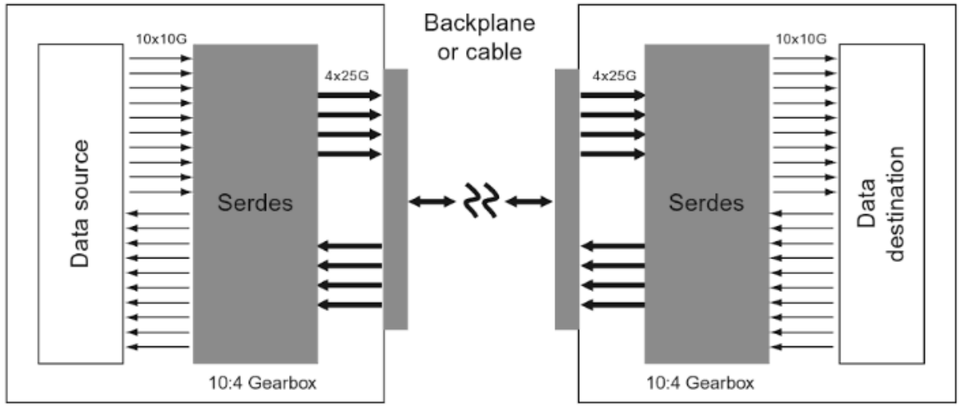
Point-to-point protocols

- Open Systems Interconnection (OSI) model
- Describes how data communication between two peers should take place
- Focused on FPGA
- The media layer will be of most interest to us
  - Layer 1 : Physical (PHY)
  - Layer 2 : Data link (MAC)



Point-to-point protocols

- Data structure and framing (OSI Layer 2)
  - Adds overhead for the peers to properly handle payload data
- Error detection and correction (OSI Layer 2/1)
  - Checks on erroneous/flipped bits
- Encoding of data (OSI Layer 1)
  - Ensures the data will arrive “electrically” correct
- Serialization and deserialization of data (OSI Layer 1)
  - Long distance communication happens over serial link [stating the obvious]

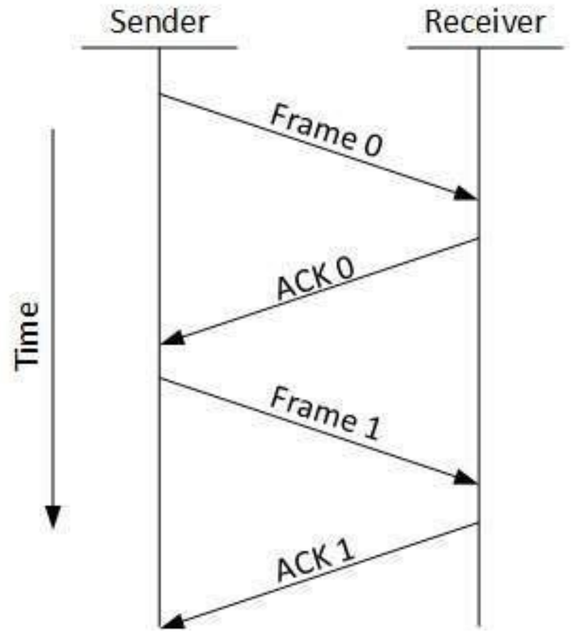


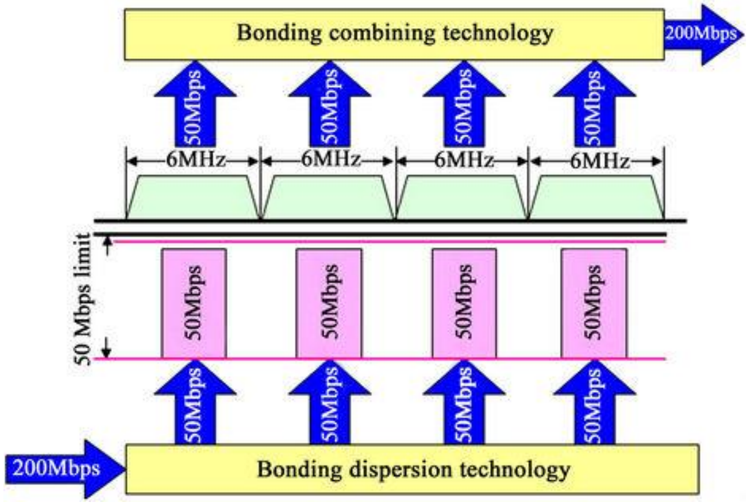
Line encoding and decoding

- Encoding adds sync headers and randomizes the binary data
  - Prevents baseline wander, constant EMI
  - Prevents errors in delineation of words
  - Essential for clock recovery
- Scrambler
  - Randomizes the data according to an algorithm
  - Can be reversed by the receiver [this is not encryption]
  - Prevents long rows of continuous binary values
  - Can be self-synchronous (on payload) or independent synchronous (per lane)
- Commonly used:
  - 8b/10b - DVI / HDMI / Gbit Eth SGMI / SATA / USB 3.0
  - 64b/66b - Ethernet / InfiniBand / Thunderbolt
  - 128b/130b - PCIe 3.0/4.0
  - 128b/132b - USB 3.1
  - 256b/257b - Fibrechannel
  - 64b/67b - Interlaken

Flow control and bonding

- Flow control
  - Start/stop transmitting at command of the receiver
  - Prevents overwhelming the receiver
- Channel Bonding
  - Combining multiple serial links in parallel
  - Increases throughput





Requirements

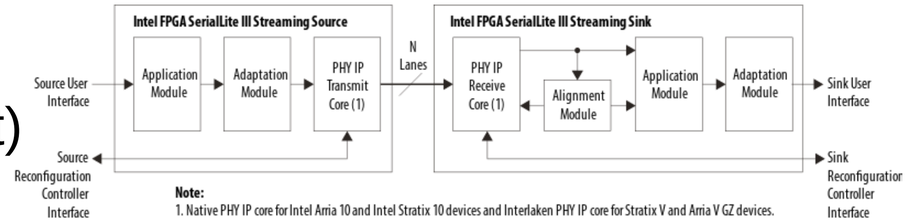
- Mandatory:
  - Line rate must be at least 10 Gbit/s (1,25 GB/s)
  - Flow control must be present
  - Range distance coverage has to be 10 - 200 meters (normal distance)

- Optional:
  - Cyclic Redundancy Check
  - Forward Error Correction
  - Channel bonding

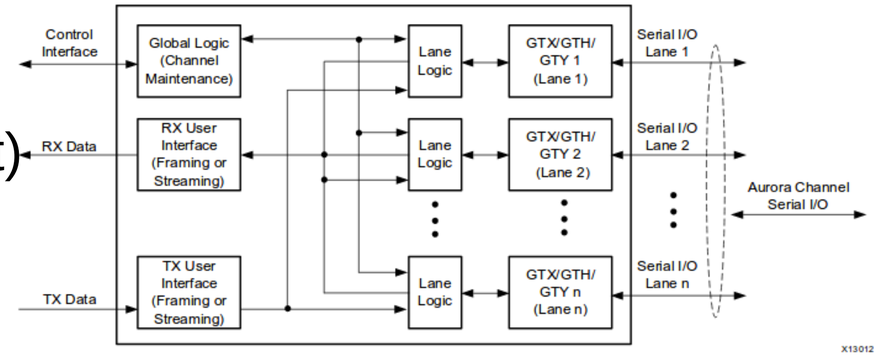
Survey

Vendor **dependent** protocols

- Altera SerialLite II & III
  - Speeds up to 28 Gbit/s (Transceiver dependent)
  - SLII supports flow control / SLIII not very clear
  - CRC-32 Error Correction
  - FEC not mentioned
  - 64b/67b encoding



- Xilinx Aurora 8b/10b & 64b/66b
  - Speeds up to 26 Gbit/s (Transceiver dependent)
  - Flow control
  - CRC-32 Error correction
  - FEC not mentioned
  - 64b/66b encoding



- Microsemi litefast – Somewhat older

Survey

This slide is explicitly full of stuff! ☺

Vendor **independent** protocols

- Interlaken
  - Clear documentation
  - meets all requirements by a fair margin
- SATA
  - Lane rate is not sufficient
  - Older 8b/10b encoding
  - Includes flow control / CRC-32
  - Not open source
  - No bonding
- CPRI
  - Unclear documentation
  - Sufficient lane rate and encoding
  - Channel bonding not supported
- HyperTransport
  - High speeds
  - Parallel bus
- Fibre Channel
  - No documentation found
  - Only old presentations to be found
  - Not open
- XAUI
  - Combines multiple slower line to 10 Gbps link
  - Used in combination with Ethernet

	Interlaken	SATA	CPRI	Fibre channel
Lane rate	25,3 Gbps	6 Gbps	24,33 Gbps	12,8 Gbps
Encoding	64b/67b	8b/10b	64b/66b	256b/257b
Flow control	Yes	Yes	-	-
Range distance	Cable dependent	Short	Cable dependent	Cable dependent
CRC	CRC-24/32	CRC-32	-	Yes
FEC	RS(544,514)-Ext	-	RS(528,514)	RS(544,514)
Channel bonding	Upto 400 Gbps	-	-	-

Survey

This slide is explicitly full of stuff! ☺

Vendor **independent** protocols

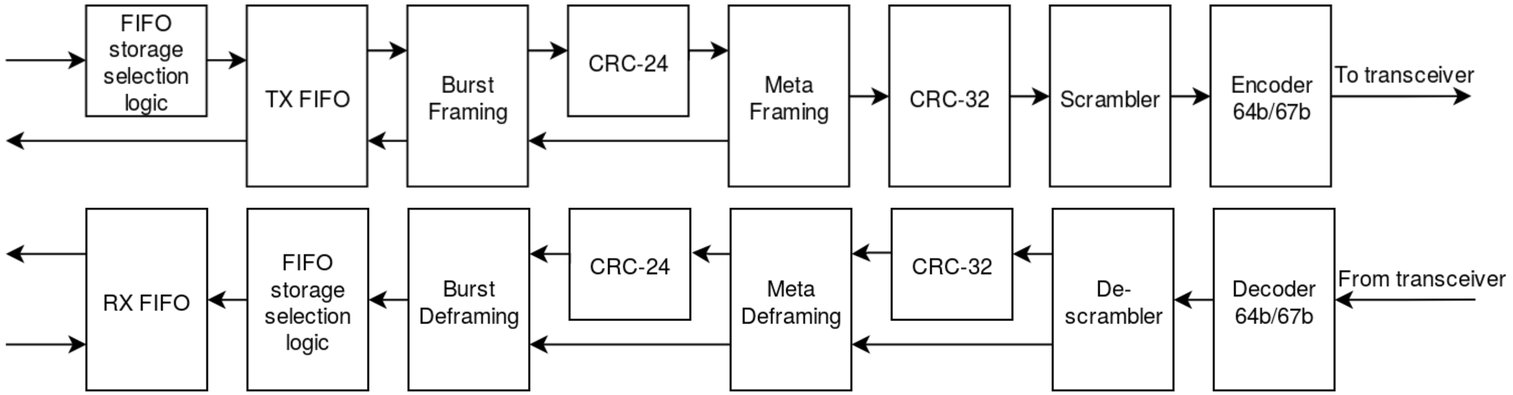
- Interlaken
  - Clear documentation
  - meets all requirements by a fair margin
- SATA
  - Lane rate is not sufficient
  - Older 8b/10b encoding
  - Includes flow control / CRC-32
  - Not open source
  - No bonding
- CPRI
  - Unclear documentation
  - Sufficient lane rate and encoding
  - Channel bonding not supported
- HyperTransport
  - High speeds
  - Parallel bus
- Fibre Channel
  - No documentation found
  - Only old presentations to be found
  - Not open
- XAUI
  - Combines multiple slower line to 10 Gbps link
  - Used in combination with Ethernet

	Interlaken	SATA	CPRI	Fibre channel
Lane rate	25,3 Gbps	6 Gbps	24,33 Gbps	12,8 Gbps
Encoding	64b/67b	8b/10b	64b/66b	256b/257b
Flow control	Yes	Yes	-	-
Range distance	Cable dependent	Short	Cable dependent	Cable dependent
CRC	CRC-24/32	CRC-32	-	Yes
FEC	RS(544,514)-Ext	-	RS(528,514)	RS(544,514)
Channel bonding	Upto 400 Gbps	-	-	-

The Interlaken Protocol

- Revision 1.2 October 7, 2008
- Property of Cortina Systems and Cisco Systems
- **Royalty-free**
- Interlaken Alliance
- Many members, among others:
  - Broadcom, Intel, Lattice, Microsemi, Xilinx
- It was made because…

Implementation



- Worked on a implementation in VHDL – target Xilinx VC707 board
- Only the transceiver and FIFOs make use of a proprietary IP Core
- Overhead (Single lane)
  - Bursts : BurstMax 32 (256 B), BurstMin 4 (32 B) -> 2/34 = 5,88% (Best case)
  - Metaframing : Metaframelength 2048 (16.384 B) -> 4/2048 = 0,20%
  - Encoding : 64b/67b -> 64/67 = 4,48%
  - Total overhead of about 10,3%
- We called it Core1990

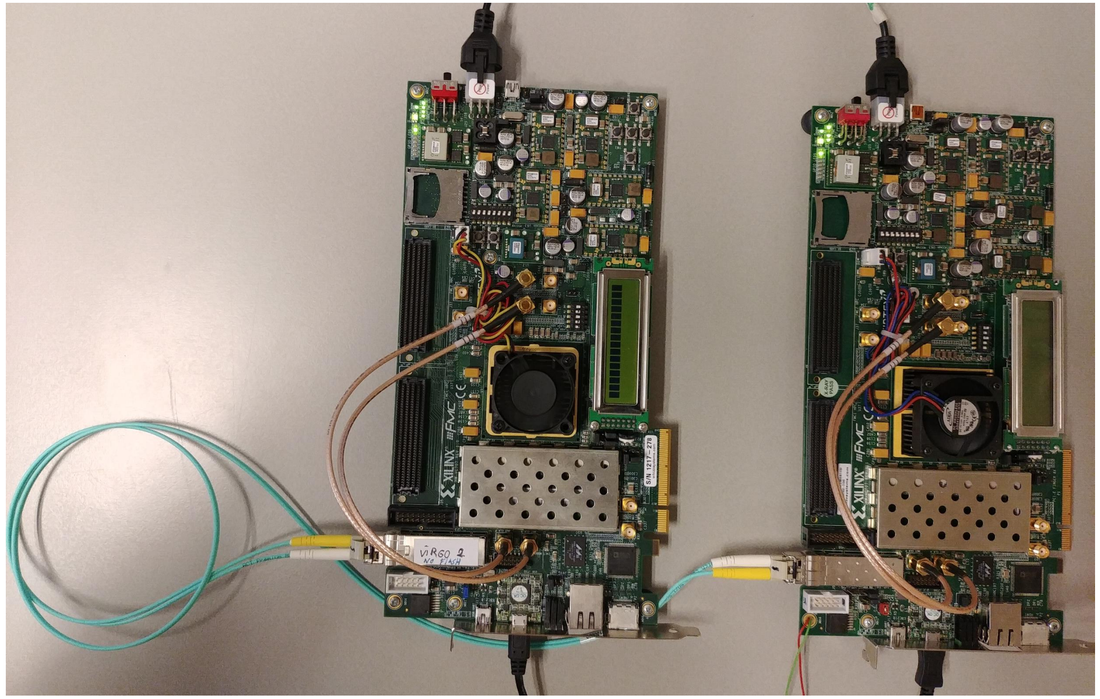
Status

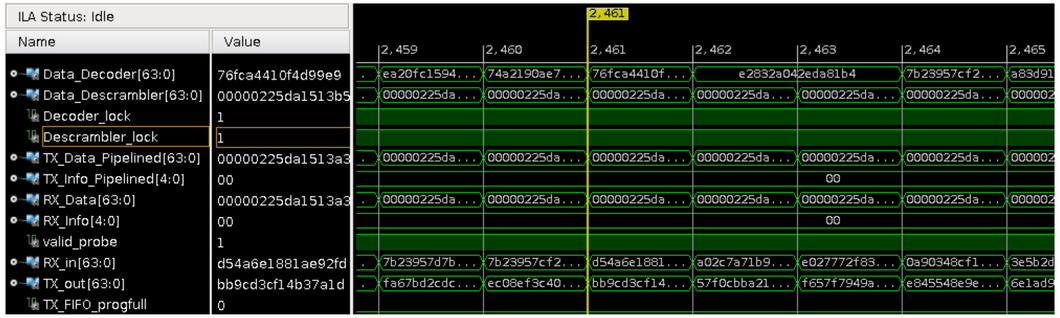
- ✓ **Found** a protocol that meets all requirements
- ✓ **Described** the protocol extensively in a structured document
- ✓ Vivado project is properly setup and **design is synthesizable**
- ✓ Design functions verified in **behavioral simulation**
  - ❑ Complete transmitter and receiver chains
  - ❑ CRC-24 and receiving FIFO need some adjustments
- ~~❑ Not yet tested in hardware~~

Status

✓ **Tested** in hardware (VC707)

- ❑ Started with single board loop-back and expanded to two boards using **fiber**
- ❑ Link is **stable** between boards
- ❑ Data constantly **generated** by counter and being **verified**
- ❑ When the fiber is unplugged **lock** is lost and when reconnecting the link **locks** again
- ❑ Transceiver operates at **10 Gbit/s**





Conclusion

- Gained knowledge
  - Much more experience in writing correct VHDL
  - Better idea of how to analyze IP Core documentation
  - More confident in designing according to specs
  - Navigate through a lot of signals in simulation
  - How to properly gather and research information
  - How to properly structure a specification document
  - Hands-on experience with “real (macho) hardware”
  - Design verification and validation in hardware

- My accomplishment is the first step
  - Others can learn from it and continue its development

- Information and documentation is uploaded to:

https://opencores.org/project/core1990_interlaken

Conclusion

- Gained knowledge
  - Much more experience in writing correct VHDL
  - Better idea of how to analyze IP Core documentation
  - More confident in designing according to specs
  - Navigate through a lot of signals in simulation
  - How to properly gather and research information
  - How to properly structure a specification document
- My accomplishment is the first step
  - Others can learn from it and continue its development
- Information and documentation will be uploaded to:

https://opencores.org/project/core1990_interlaken

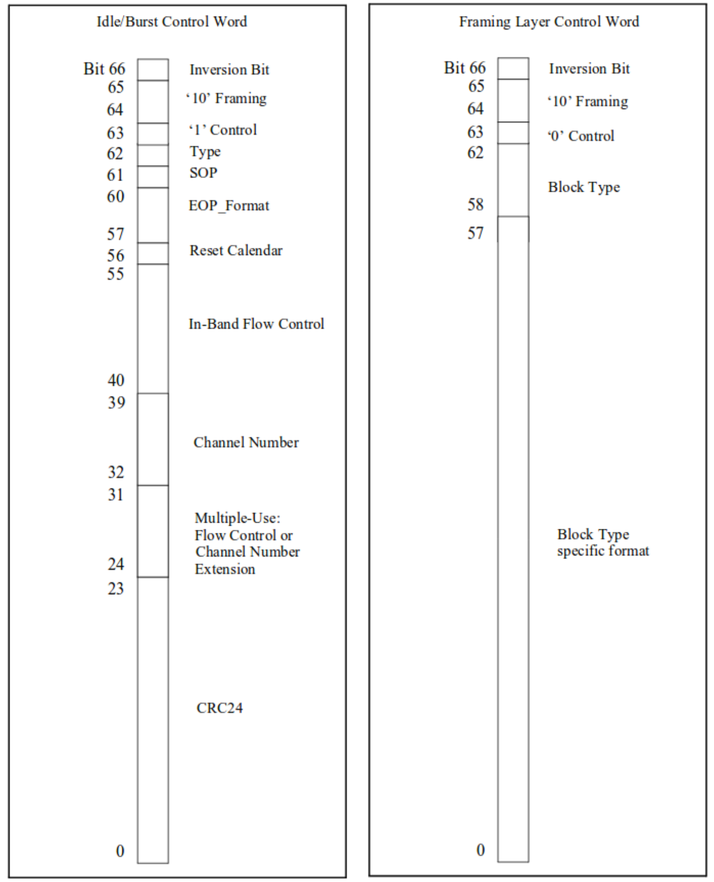
Thank you for your attention! ☺

Backup slides

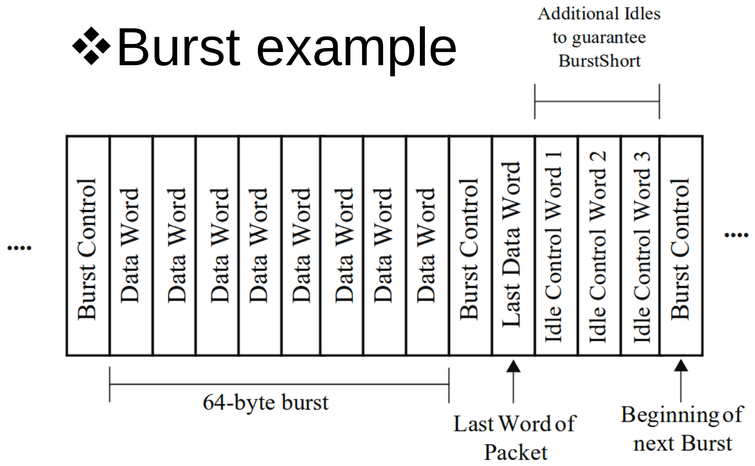
The Interlaken Protocol

This slide is a cheat sheet! ☺

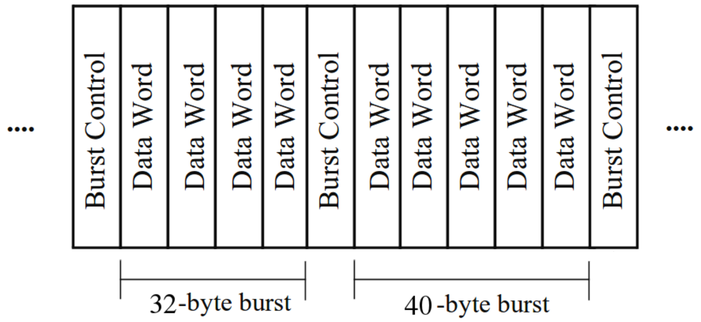
- Burst framing
  - Covered by CRC-24
  - Contains start and end
  - BurstMax
  - BurstMin
- In-band flow control
- Channel numbers
- Idle/burst and framing control words



- Burst example



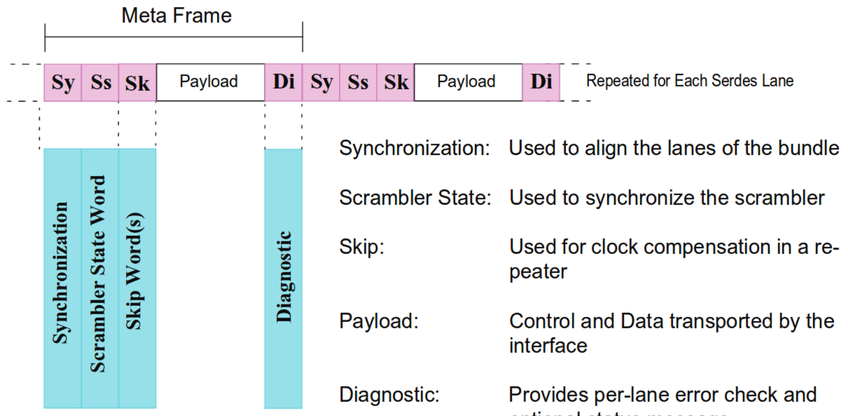
- Optional scheduling



The Interlaken Protocol

This slide is also a cheat sheet! ☺

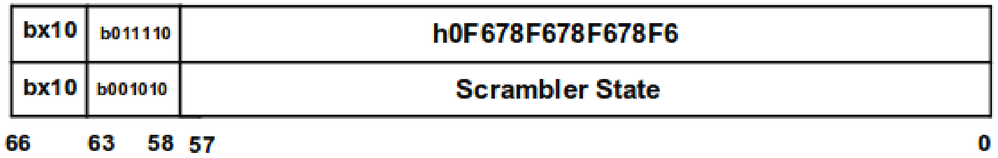
- Meta framing
  - Covered by CRC-32
  - Synchronizes the connection
  - Communicates scrambler states
  - Skip words for clock synchronization
  - Diagnostic information



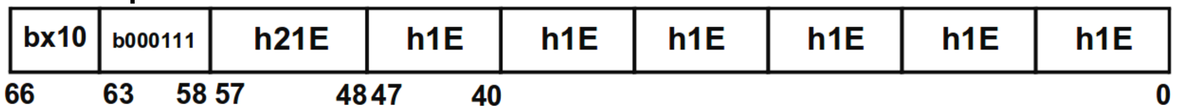
Synchronization:	Used to align the lanes of the bundle
Scrambler State:	Used to synchronize the scrambler
Skip:	Used for clock compensation in a re-peater
Payload:	Control and Data transported by the interface
Diagnostic:	Provides per-lane error check and optional status message

Meta Frame Control Word	Block Type (positive disparity)	Block Type (negative disparity)
Synchronization	011110	100001
Scrambler State	001010	110101
Skip	000111	111000
Diagnostic	011001	100110

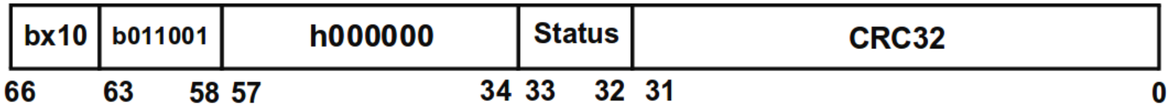
- Synchronization / Scrambler state



- Skip word



- Diagnostic word



The Interlaken Protocol

This slide is also a cheat sheet! ☺

- Flow Control
  - XON/XOFF
- In-band flow control
  - Included in burst control words
  - Intended for duplex operation
- Out-of-band flow control
  - Covered by CRC-4
  - Intended for simplex operation

- CRC-32 - Meta framing

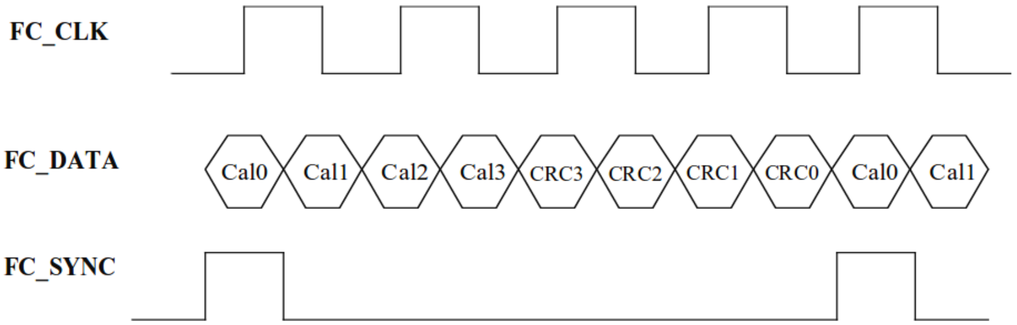
  $X^{32}+X^{28}+X^{27}+X^{26}+X^{25}+X^{23}+X^{22}+X^{20}+X^{19}+X^{18}+X^{14}+X^{13}+X^{11}+X^{10}+X^{9}+X^{8}+X^{6}+1$

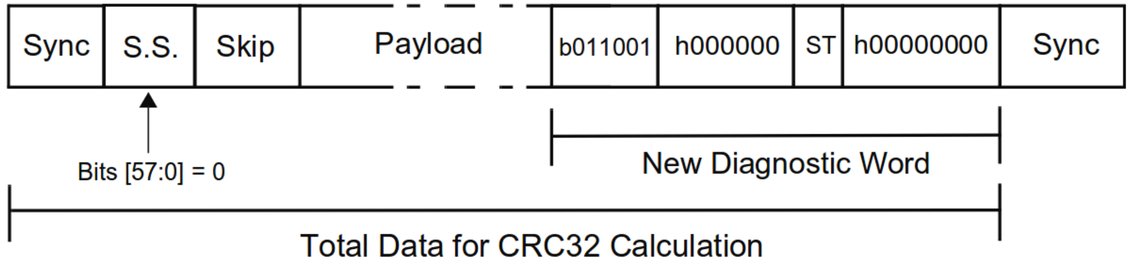
- CRC-24 - Burst framing

  $X^{24}+X^{21}+X^{20}+X^{17}+X^{15}+X^{11}+X^{9}+X^{8}+X^{6}+X^{5}+X+1$

- CRC-4 - Out-of-band flow control

  $X^{4}+X+1$





The Interlaken Protocol

This slide is also a cheat sheet! ☺

❖Scrambler

- Independent synchronous scrambler
- 58 bit polynomial : $X^{58}+X^{38}+1$
- No error multiplication through multiple frames
- In case of error, synchronizes itself

❖Encoder

- Adds a 3-bit preamble
- 50% of combinations possible
- Running disparity within 96-bit boundary
- Doesn’t rely on the scrambler
- Lower Bit-Error-Rate

Bits [66:64]	Interpretation
001	Data Word, no inversion
010	Control Word, no inversion
101	Data Word, bits [63:0] are inverted
110	Control Word, bits [63:0] are inverted
All others	Illegal states