Parallel Communications PCIe

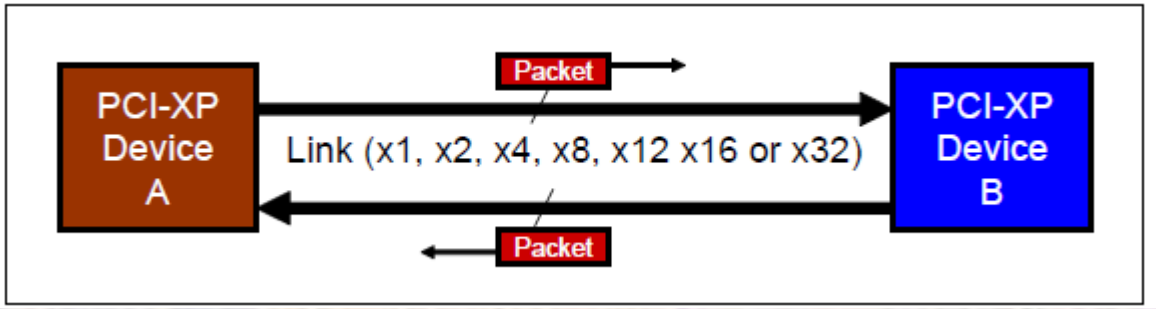
Last updated – 4/6/20

PCIE

- PCIe
  - Peripheral Component Interconnect - Express
  - Preceded by PCI and PCI-X
    - But completely different physically
  - Logical configuration separate from the physical configuration
    - Logical configuration is backward compatible → SW reuse
  - On-chip or on-board
  - Packet based – point to point architecture

PCIE

- PCIe
  - Point to point
    - Fixed – hardwired direct connection (LINK)

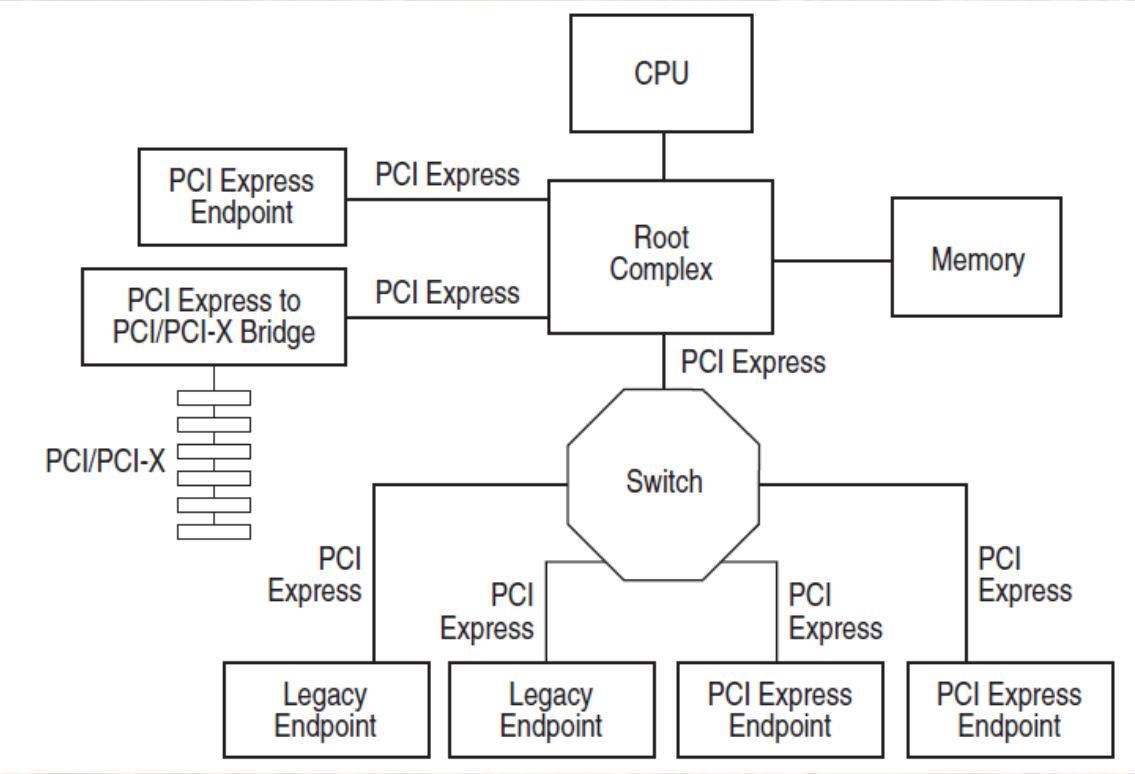


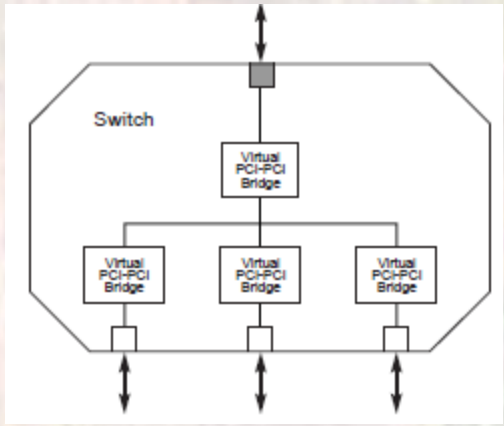
PCIE

- PCIe
  - Why point to point?
    - Reduces the RC delay on the signals
    - Allows for very fast transfer speeds
    - 2.5GT/s, 5GT/s, 8GT/s, 16GT/s, 32GT/s with 64GT/s in 2021
      - GT/s – Giga-Transfers per second
      - Assumes simultaneous Rx/Tx

PCIE

- PCIe
  - Networked - uses switches to route messages





PCIE

- PCIe
  - Lane
    - 1 transmit path and 1 receive path
    - Differential signaling → 4 wires/Lane
  - Link
    - Collection of Lanes
    - 1x, 2x, 4x, 8x, 12x, 16x, 32x
    - Symmetric – same both directions
  - No Clock !

PCIE

- PCIe
  - Lanes and Links

PCI Express version	Introduced	Line code	Transfer rate[i]	Throughput[i]				
				x1	x2	x4	x8	x16
1.0	2003	8b/10b	2.5 GT/s	250 MB/s	0.500 GB/s	1.00 GB/s	2.0 GB/s	4.0 GB/s
2.0	2007	8b/10b	5.0 GT/s	500 MB/s	1.000 GB/s	2.00 GB/s	4.0 GB/s	8.0 GB/s
3.0	2010	128b/130b	8.0 GT/s	984.6 MB/s	1.969 GB/s	3.94 GB/s	7.88 GB/s	15.75 GB/s
4.0	2017	128b/130b	16.0 GT/s	1969 MB/s	3.938 GB/s	7.88 GB/s	15.75 GB/s	31.51 GB/s
5.0	2019	128b/130b	32.0 GT/s[ii]	3938 MB/s	7.877 GB/s	15.75 GB/s	31.51 GB/s	63.02 GB/s
6.0 (planned)	2021	128b/130b	64.0 GT/s	7877 MB/s	15.754 GB/s	31.51 GB/s	63.02 GB/s	126.03 GB/s

But wait !

5GT/s x 8 links = 40GT/s

40GT/s / 8 bits = 5GB/s

Why the difference?

PCIE

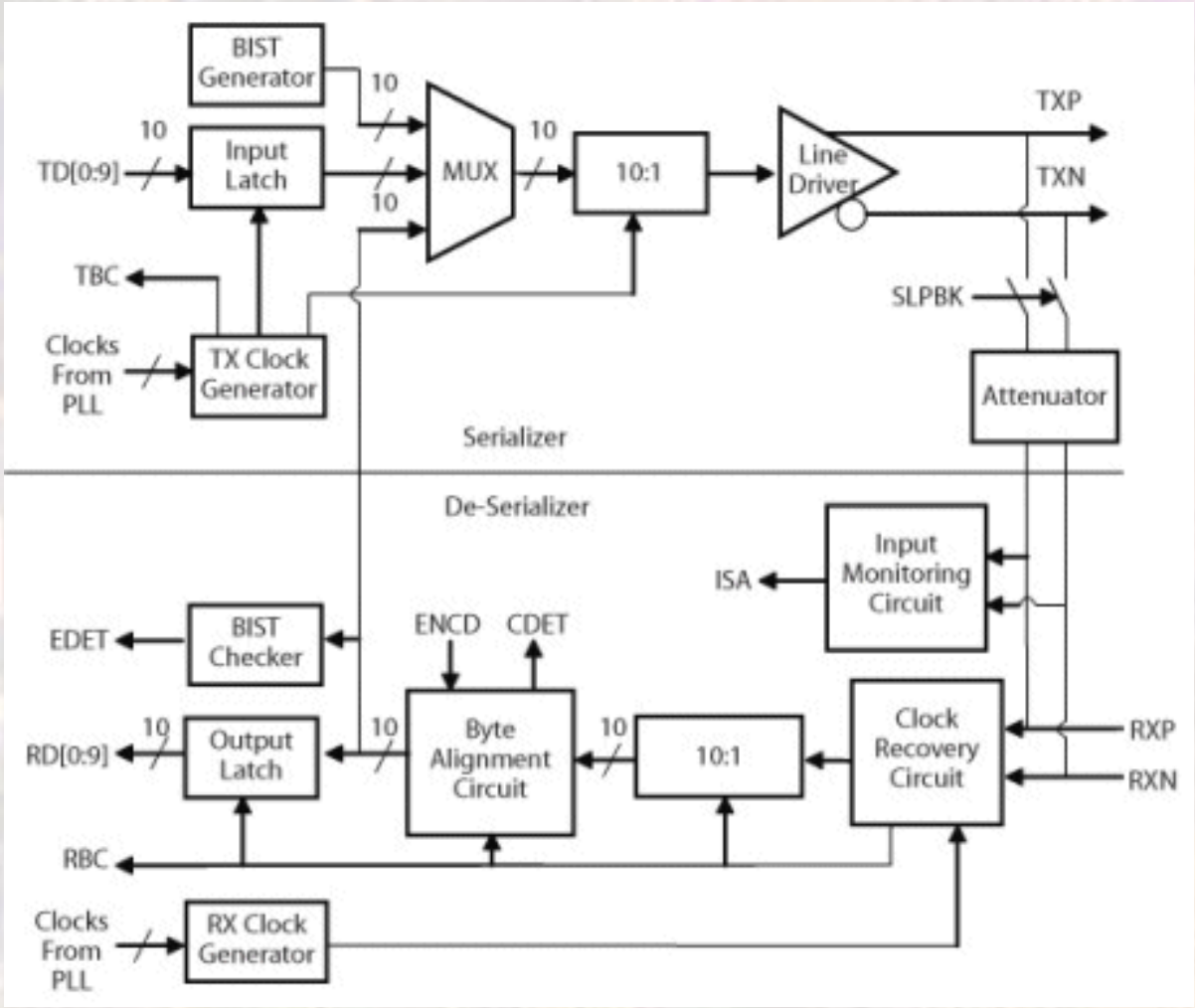
- PCIe
  - Encoding
    - PCIe 1.x and 2.x use 8b/10b encoding
    - PCIe 3.x - 6.x use 128b/130b encoding
  - 8b/10b Encoding
    - Maps 8 bit symbols to 10 bit symbols
    - Coding ensures # of 1s and # of 0s differ by ≤ 2 for any string of 20 bits
    - Coding ensures no more than 5, 0s or 1s in a row
    - Provides enough transitions to do clock recovery
    - Provides for DC balance

PCIE

- PCIe signaling
  - Clock Recovery
  - Both Rx and Tx generate a common clock frequency
    - 2.5Ghz, 5GHz, 8GHz, 16GHz
    - Only at the Rx/Tx interfaces
  - Phase between Rx and Tx is uncorrelated
  - Rx side uses a PLL to phase align the receiver clock to the incoming differential signal
    - This is where encoding helps – ensures sufficient transitions to keep PLL locked

PCIE

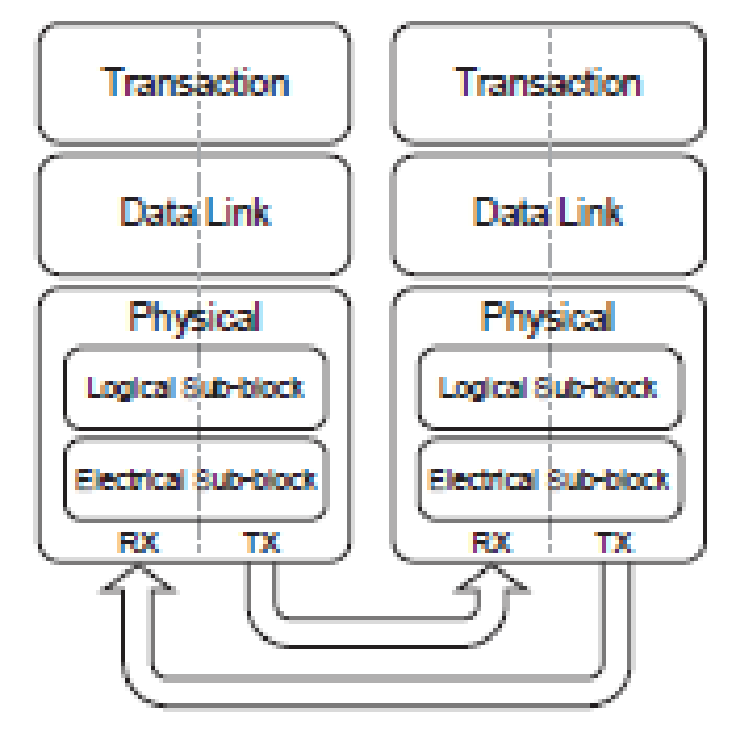
- PCIe signaling



src: design reuse

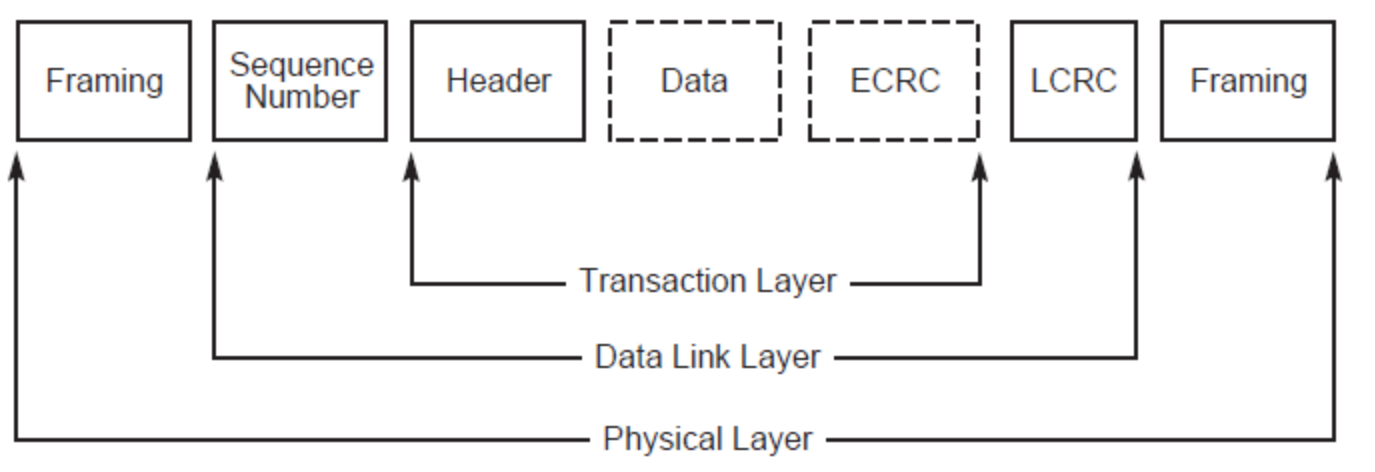
PCIE

- PCI Layering



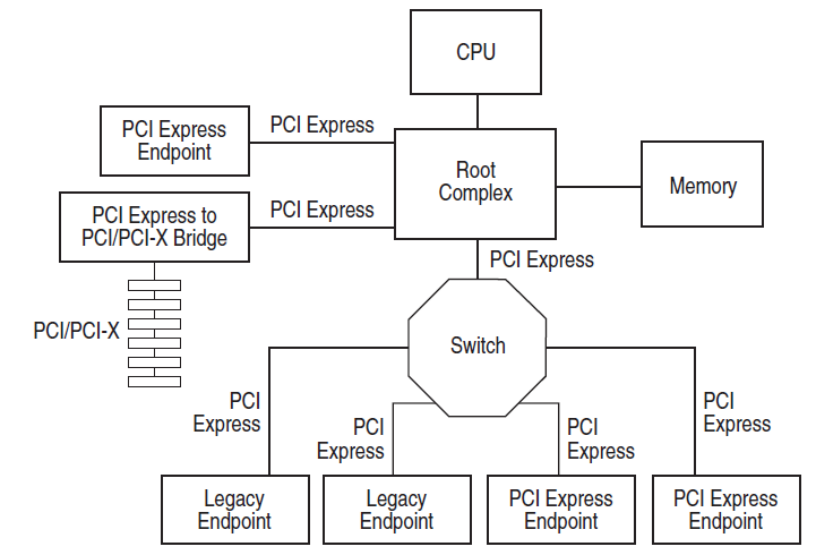
PCIE

- PCI Packets



PCIE

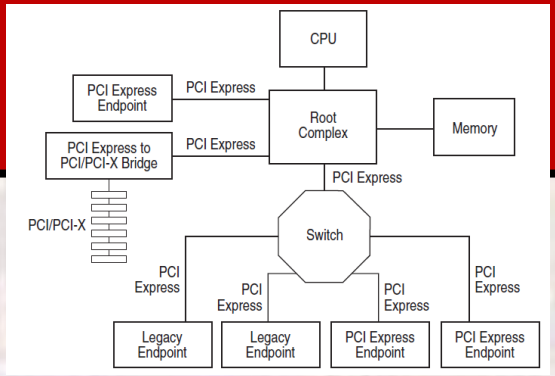
- Primary Functional Blocks
  - Root Complex
  - Endpoint
  - Switch
  - Bridge



- Upstream Port (only 1) - points toward the root complex
- Downstream Port(s) – point away from root complex

- Ingress Port – incoming port (receiver)
- Egress Port – outgoing port (transmitter)

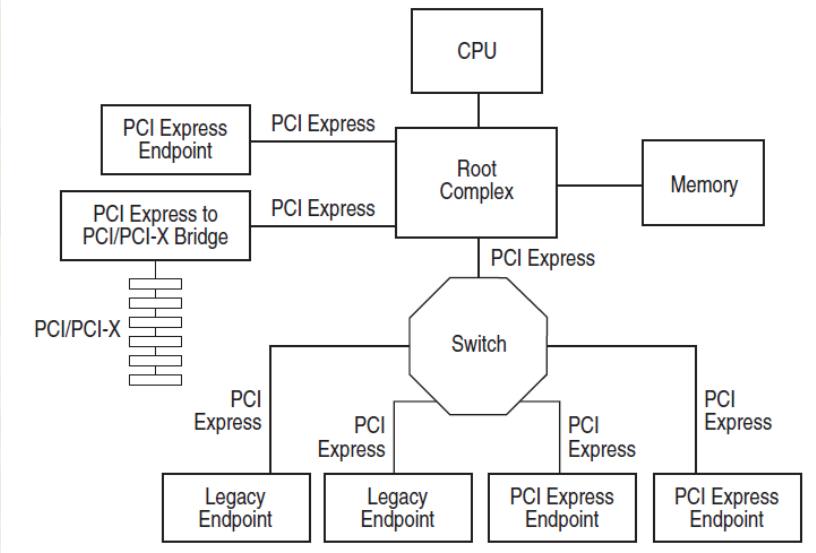
PCIE



- Root Complex
  - Connects CPU(s) and Memory to the PCI fabric
  - Multiple PCIe ports
  - Generates transactions based on commands from the CPU
  - Can act as power controller, interrupt controller, error detection response

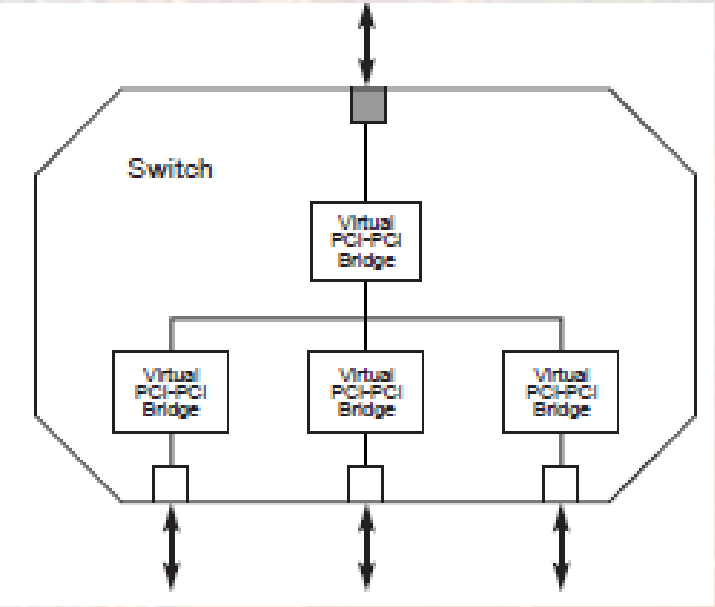
PCIE

- Endpoint
  - Peripheral or bridge to another protocol
  - An endpoint can support multiple different functions



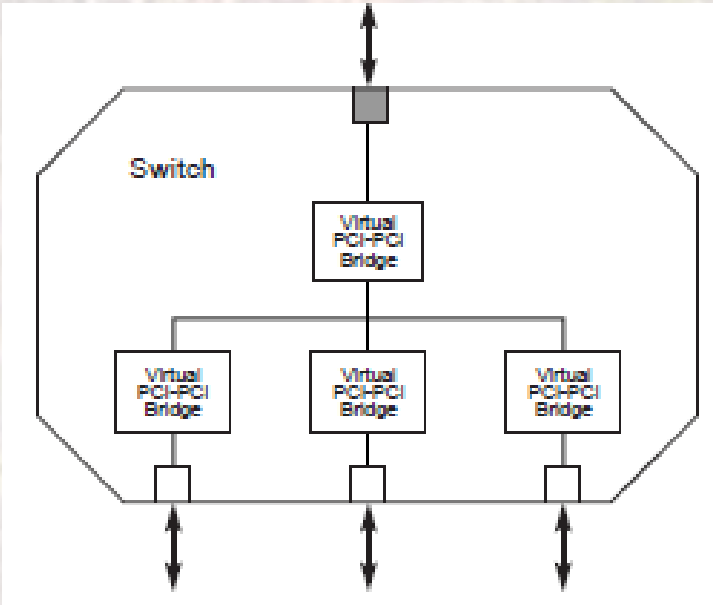
PCIE

- Switch
  - Acts like multiple virtual bridges – PCIe to PCIe
  - Hierarchy of bridges allowed
  - Configuration includes information on
    - Downstream devices
    - Downstream buses
  - Forwards based on
    - Address routing
    - ID routing
    - Implicit routing
      - Up one level, root device, …



PCIE

- Switch – cont'd
  - Perform port arbitration
    - Multiple transactions to the same port
  - Perform Virtual Channel arbitration
    - Priority decisions



PCIE

- Elements
  - Each element has a device ID
    - Bus #
    - Device #
    - Function #
  - Root: 0,0,0 (b,d,f)
  - Endpoint: x,0,x
    - Each endpoint is defined as Device # 0
    - Multifunction endpoints must have a Function # 0

PCIE

- Transactions
  - Transaction
    - 1 or more packet transmissions
    - Each transaction has a Tag associated with it
      - Allows split transactions

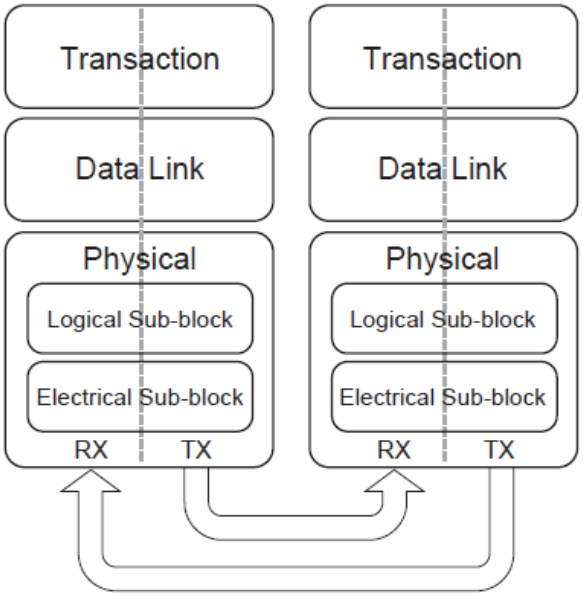
Address Space	Transaction Types	Basic Usage
Memory	Read Write	Transfer data to/from a memory-mapped location
I/O	Read Write	Transfer data to/from an I/O-mapped location
Configuration	Read Write	Device Function configuration/setup
Message	Baseline (including Vendor–defined)	From event signaling mechanism to general purpose messaging

PCIE

- Transactions
  - Non-posted transaction
    - A response is expected at some future time
    - E.g. memory read
  - Posted transaction
    - No response is required or expected
    - E.g. non-posted memory write
  - Requesters and Completers instead of Master/slave

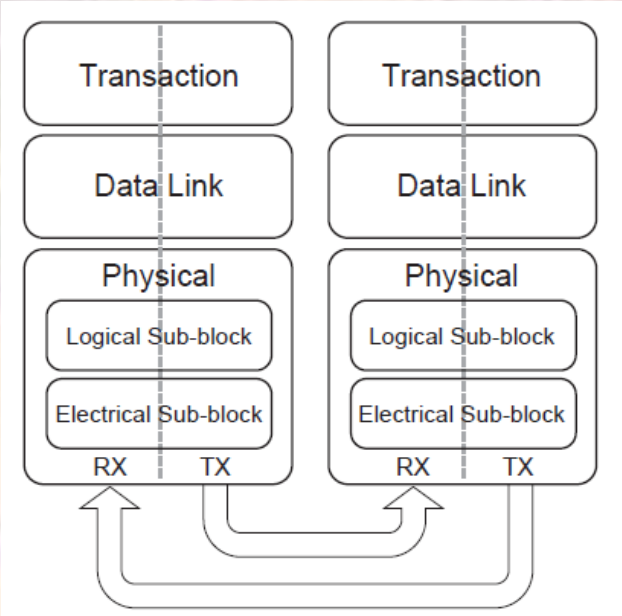
PCIE

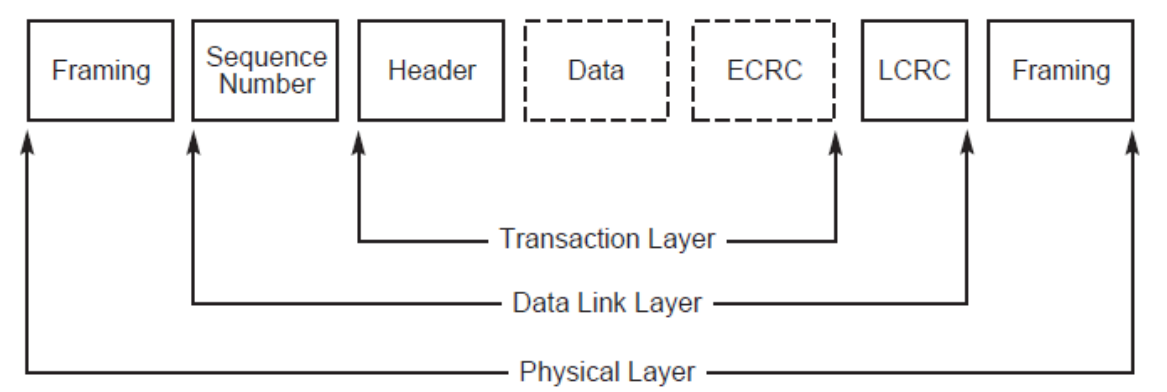
- Multi-Layer Protocol
  - Transaction Layer
  - Data Layer
  - Physical Layer
  - TLP
    - Transaction Layer Packet
    - Originates/terminates at the Transaction Layer
  - DLLP
    - Data Link Layer Packet
    - Originates/terminates at the Data Link Layer



PCIE

- Multi-Layer Protocol





PCIE

- Transaction Layer
  - Creates TLPs
  - Interfaces to the peripheral to make requests or provide results
    - Read – address, device, …
    - Write – address, device, data
    - Flow control
      - Buffer overflow, underflow
  - Each packet has a unique identifier
  - Where the Memory, IO, Config and Messages are created and consumed

PCIE

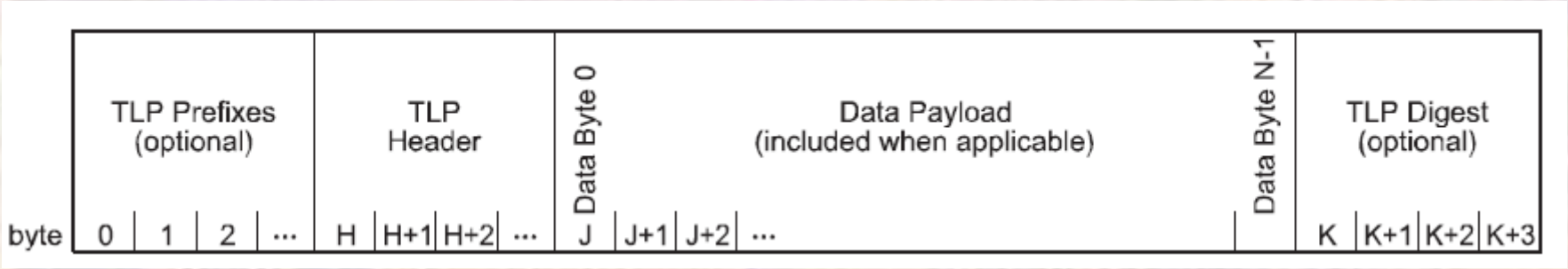
- Data Link Layer
  - Takes packet info from the Transaction Layer, adds to it, and transmits to the Physical Layer – and reverse
  - Provides Link Management
    - Supports data integrity
      - Error detection & correction
    - Requests re-transmits
  - Creates and uses packets for link management separate from transferring Transaction Layer information (DLLPs)

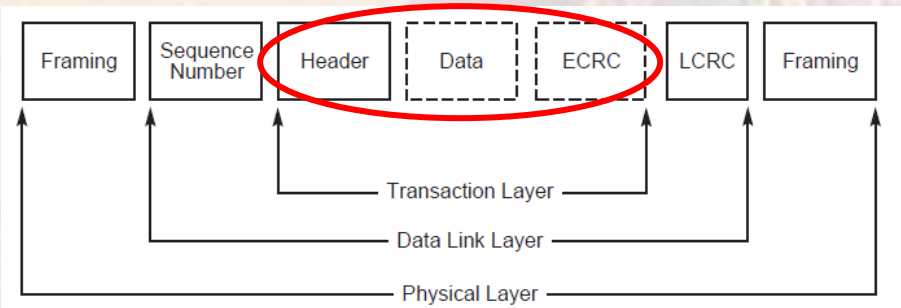
PCIE

- Physical Layer
  - Separated into two sections
  - Logical
    - Formats data
    - Adds framing information
  - Electrical
    - RX/TX circuitry

PCIE

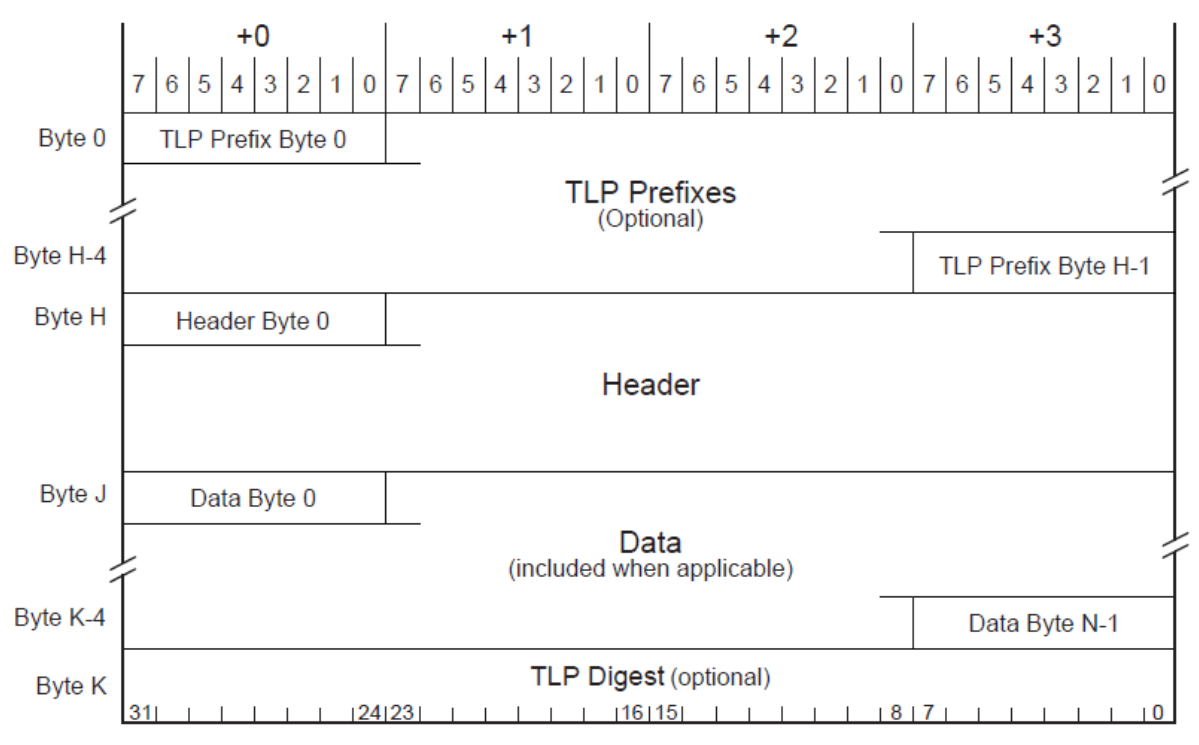
- Transaction Layer Packets
  - Left most bit transferred first





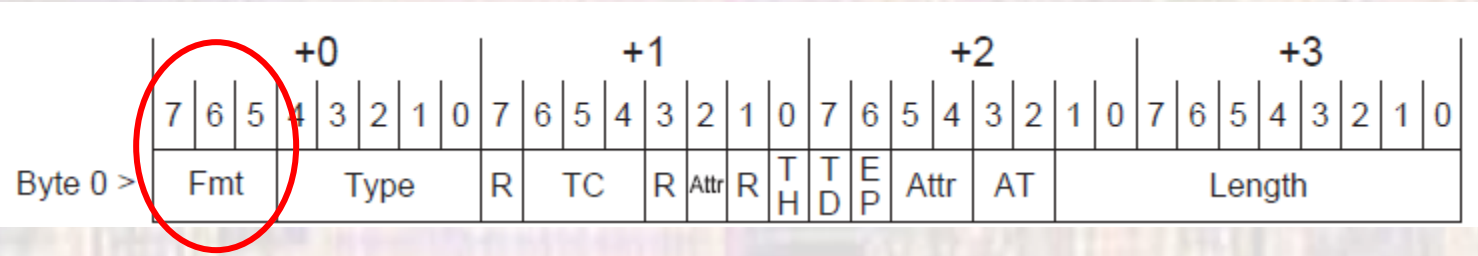
PCIE

- Transaction Layer Packets
  - Left most bit transferred first



PCIE

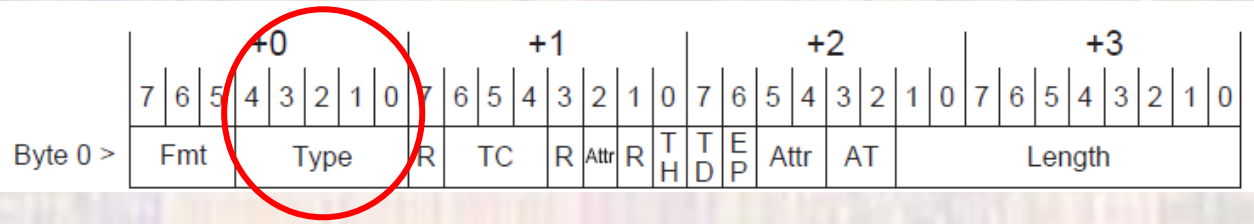
- Transaction Layer Packets
  - Header portion



Fmt[2:0]	Corresponding TLP Format
000b	3 DW header, no data
001b	4 DW header, no data
010b	3 DW header, with data
011b	4 DW header, with data
100b	TLP Prefix

PCIE

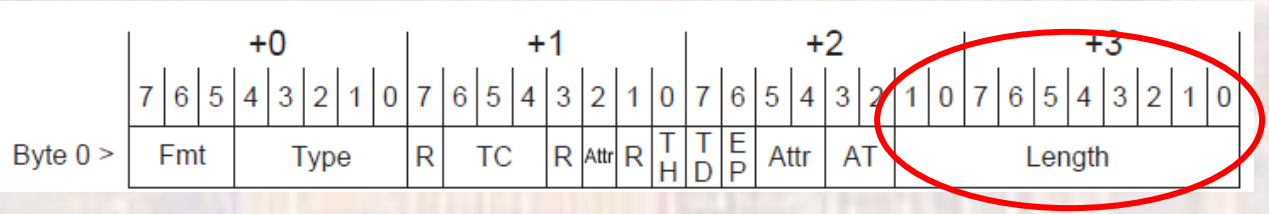
- Transaction Layer Packets
  - Header portion



TLP Type	Fmt [2:0][2] (b)	Type [4:0] (b)	Description
MRd	000 001	0 0000	Memory Read Request
MRdLk	000 001	0 0001	Memory Read Request-Locked
MWr	010 011	0 0000	Memory Write Request
IORd	000	0 0010	I/O Read Request
IOWr	010	0 0010	I/O Write Request
CfgRd0	000	0 0100	Configuration Read Type 0
CfgWr0	010	0 0100	Configuration Write Type 0
CfgRd1	000	0 0101	Configuration Read Type 1
CfgWr1	010	0 0101	Configuration Write Type 1

PCIE

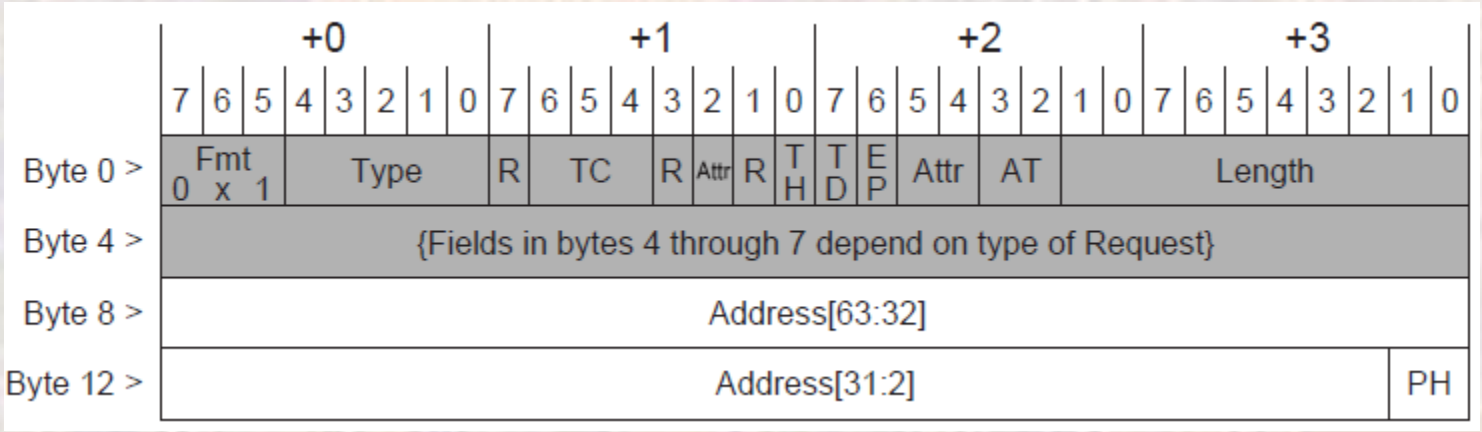
- Transaction Layer Packets
  - Header portion



Length[9:0]	Corresponding TLP Data Payload Size
00 0000 0001b	1 DW
00 0000 0010b	2 DW
...	...
11 1111 1111b	1023 DW
00 0000 0000b	1024 DW

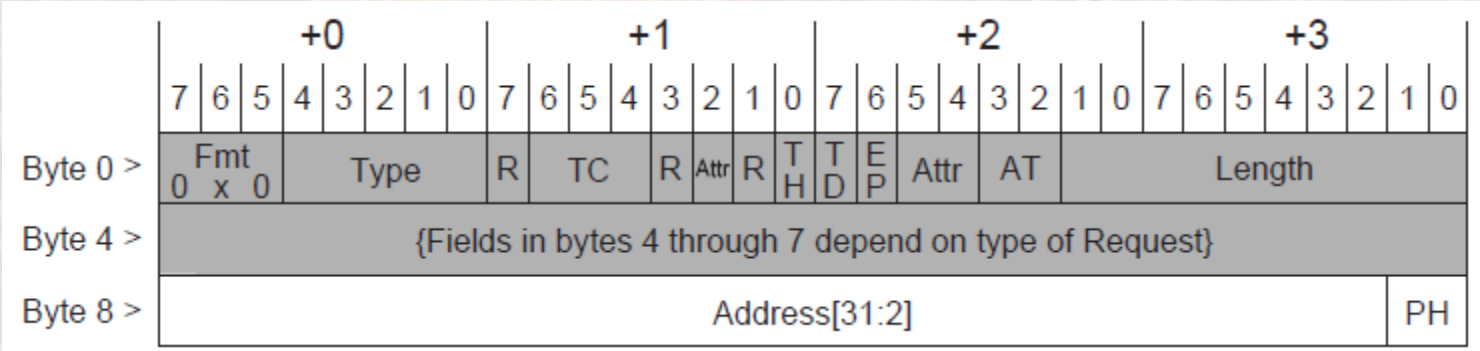
PCIE

- Transaction Layer Packets
  - Address Based Routing
    - 4 DW header
    - 64 bit addressing



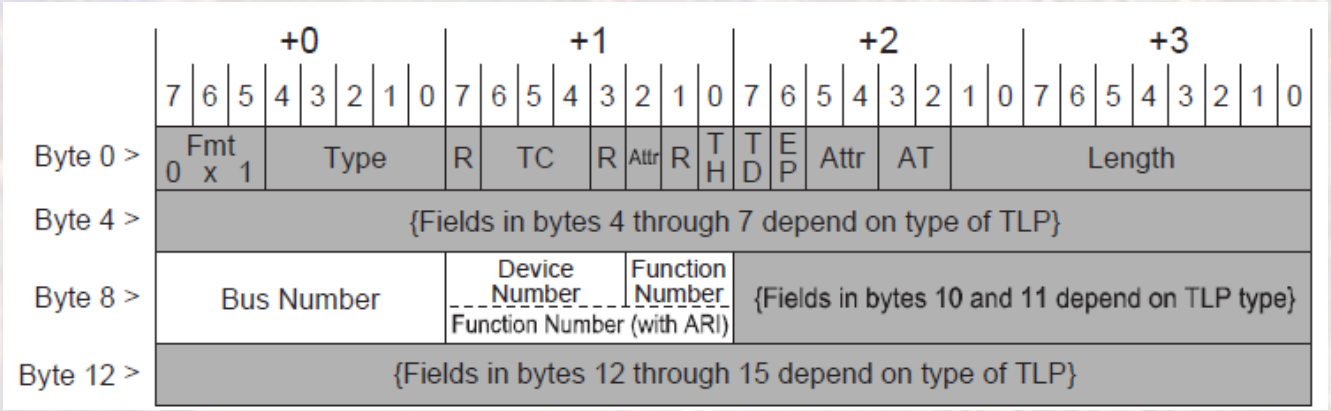
PCIE

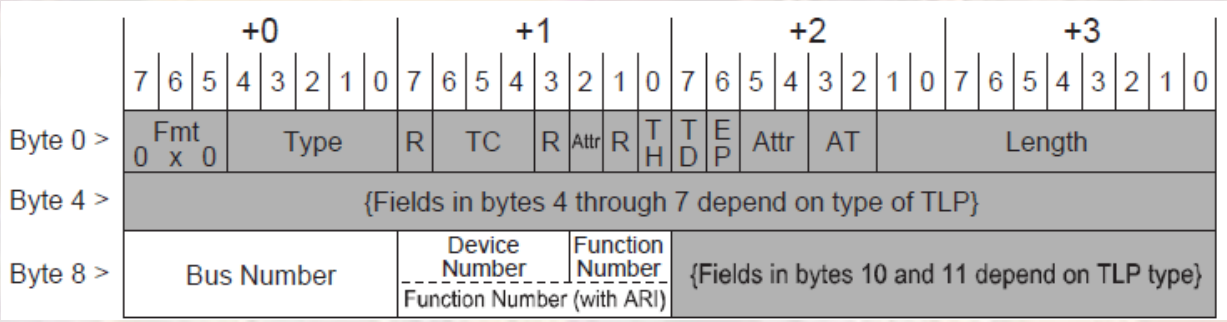
- Transaction Layer Packets
  - Address Based Routing
    - 3 DW header
    - 32 bit addressing



PCIE

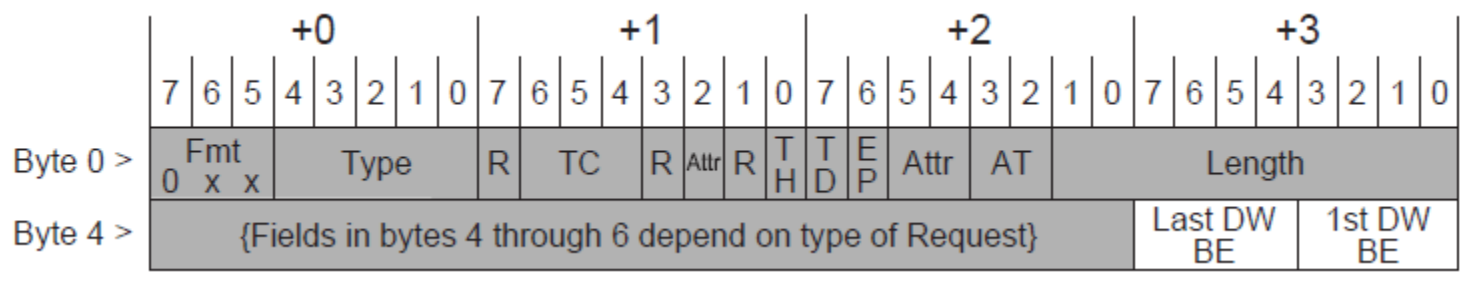
- Transaction Layer Packets
  - ID Based Routing





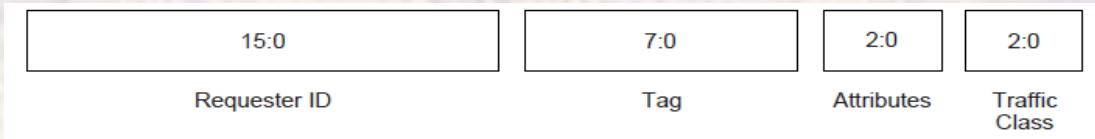
PCIE

- Transaction Layer Packets
  - Data Word Byte Enable
    - Provides Byte level enabling of the first and last data words
    - All intermediate words (assuming >2 for length) must match both the first and last DW enable
    - 32 bit words → 4 bytes
    - 1st DWenable = 0110, last DWenable = 1100
      - Enable byte 1 and 2 of the 1st word, no intermediate words, bytes 2 and 3 of the last DW



PCIE

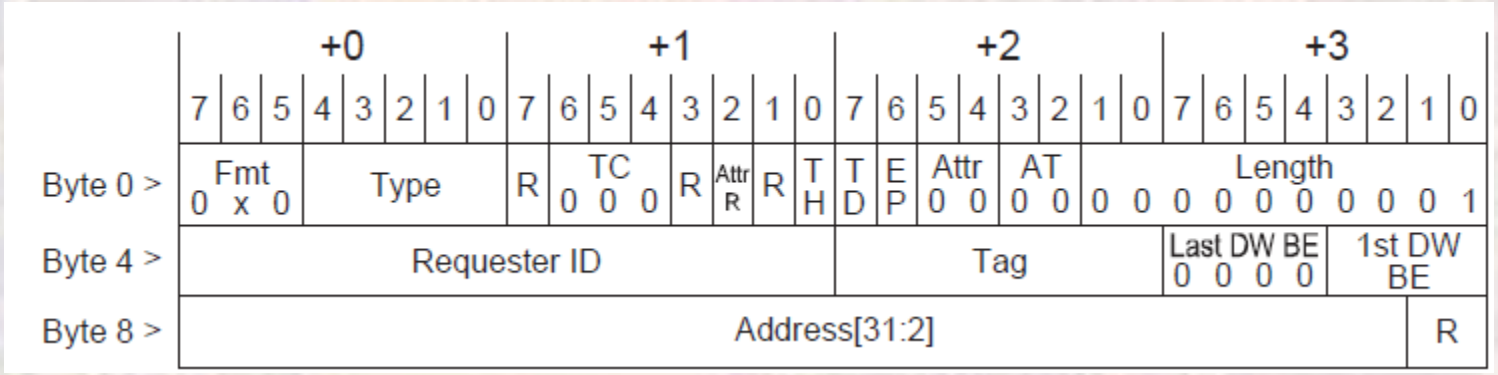
- Transaction Layer Packets
  - Transaction ID
    - Requester ID
    - Tag
    - Attributes – no-snoop?
    - Transaction class – best effort, round robin, …



	+0								+1								+2								+3							
	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
Byte 0 >	Fmt 0 x 1			Type					R	TC			R	Attr	R	TH	TD	EP	Attr		AT				Length							
Byte 4 >	Requester ID																Tag								Last DW BE				1st DW BE			
Byte 8 >	Address[63:32]																															
Byte 12 >	Address[31:2]																														PH	

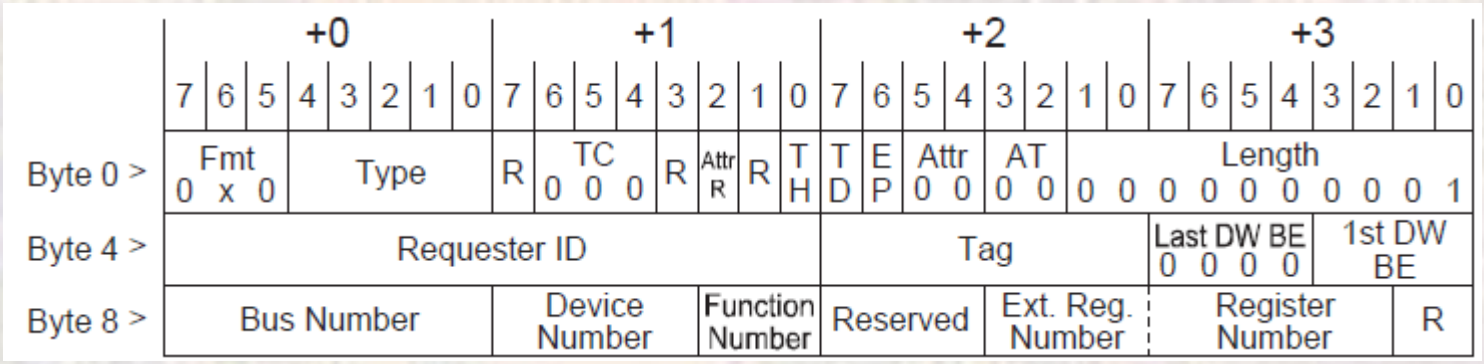
PCIE

- Transaction Layer Packets
  - IO Transaction Request Example



PCIE

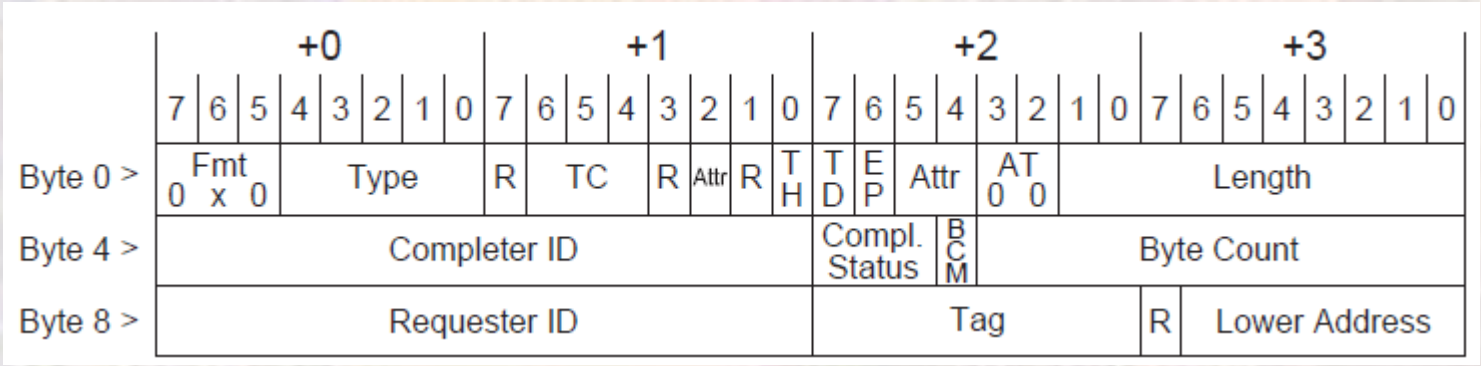
- Transaction Layer Packets
  - Configuration Transaction Request Example



	+0								+1								+2								+3							
	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
Byte 0 >	Fmt 0 x 0				Type				R	TC 0 0 0			R	Attr R	R	T H	T D	E P	Attr 0 0		AT 0 0		Length 0 0 0 0 0 0 0 0 0 1										
Byte 4 >	Requester ID																Tag								Last DW BE 0 0 0 0				1st DW BE			
Byte 8 >	Bus Number								Device Number								Function Number				Reserved				Ext. Reg. Number				Register Number								R

PCIE

- Transaction Layer Packets
  - Completion header
    - Header for any completion packets
    - E.g. results of a read



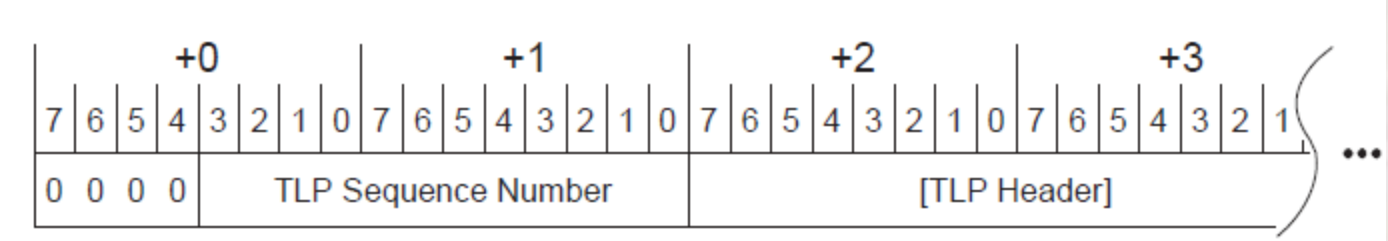
Completion Status[2:0] Field Value (b)	Completion Status
000	Successful Completion (SC)
001	Unsupported Request (UR)
010	Configuration Request Retry Status (CRS)
100	Completer Abort (CA)
all others	Reserved

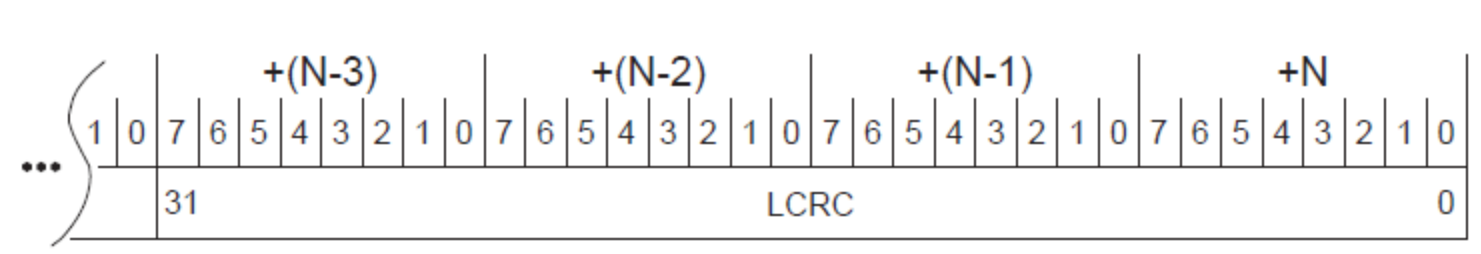
PCIE

- Transaction Layer Packets
  - Optional Digest Field
    - Can be used for additional data checking at the Transaction Layer
    - Use a CRC – Cyclical Redundancy Check on the bits of the Transaction Layer Packet

PCIE

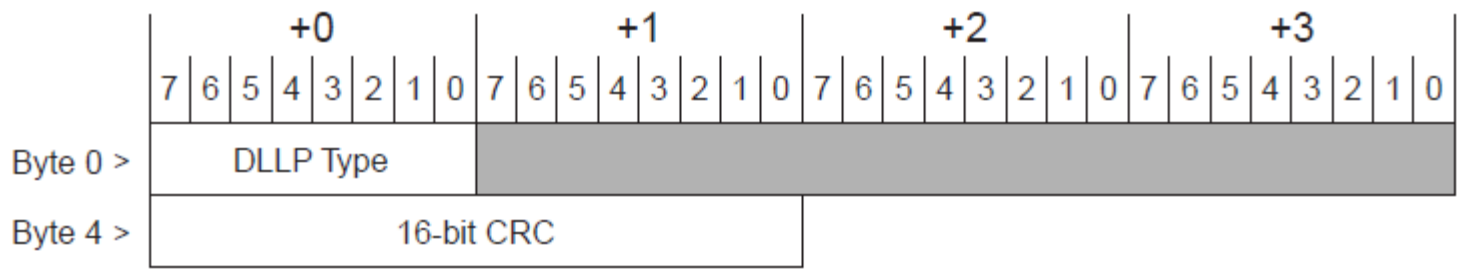
- Data Link Layer Packets - TLP
  - Prepends a sequence #
    - Used to detect entirely missing transmissions





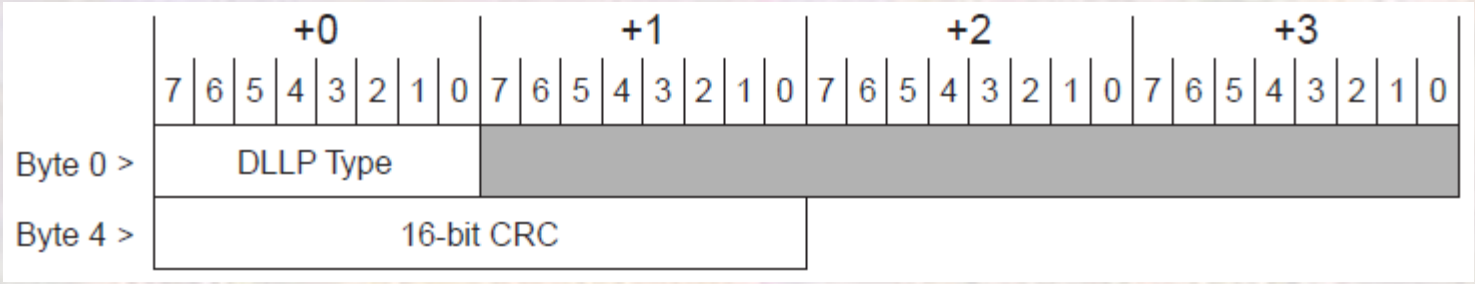
PCIE

- Data Link Layer Packets - DLLP
  - General format

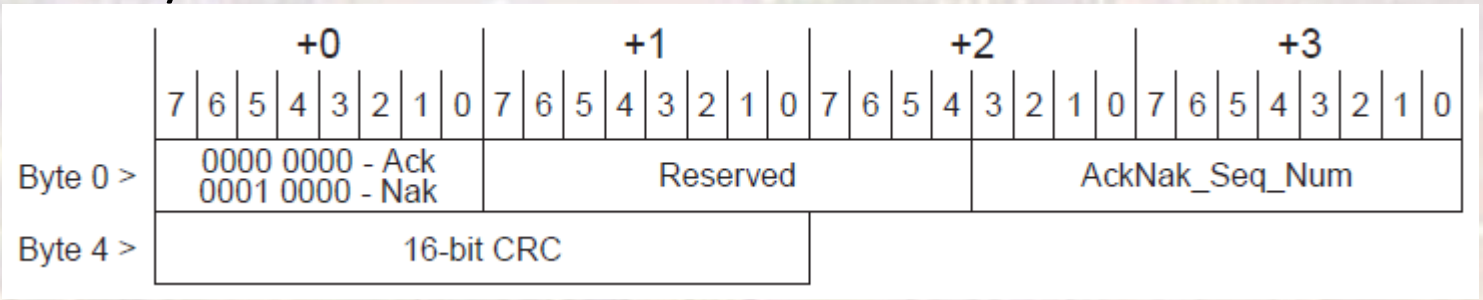


PCIE

- Data Link Layer Packets - DLLP
  - General format

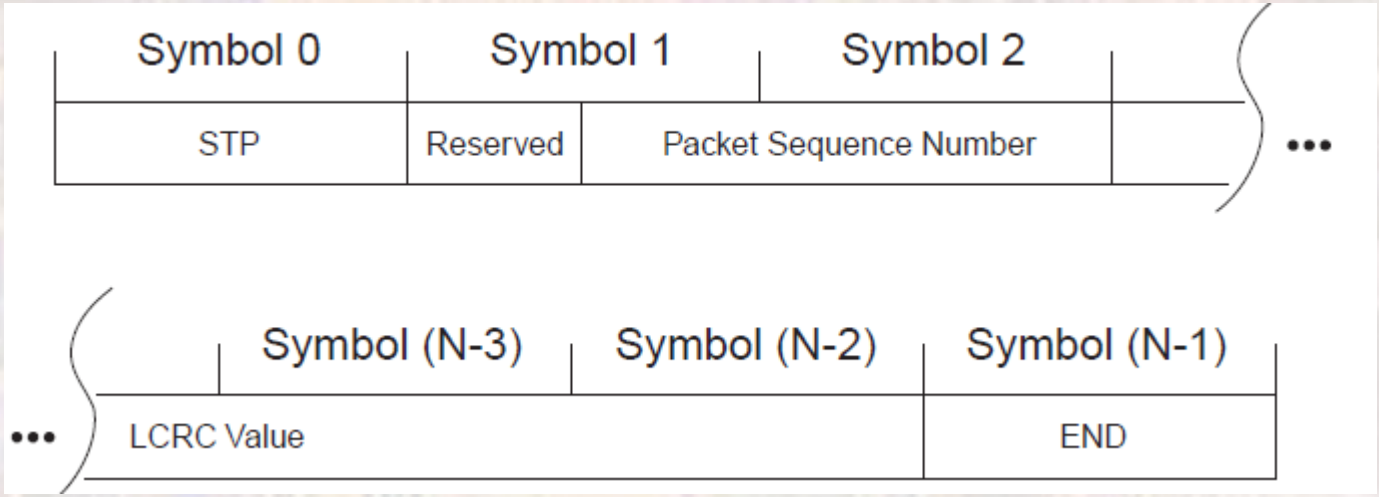


  - Ack/Nak format



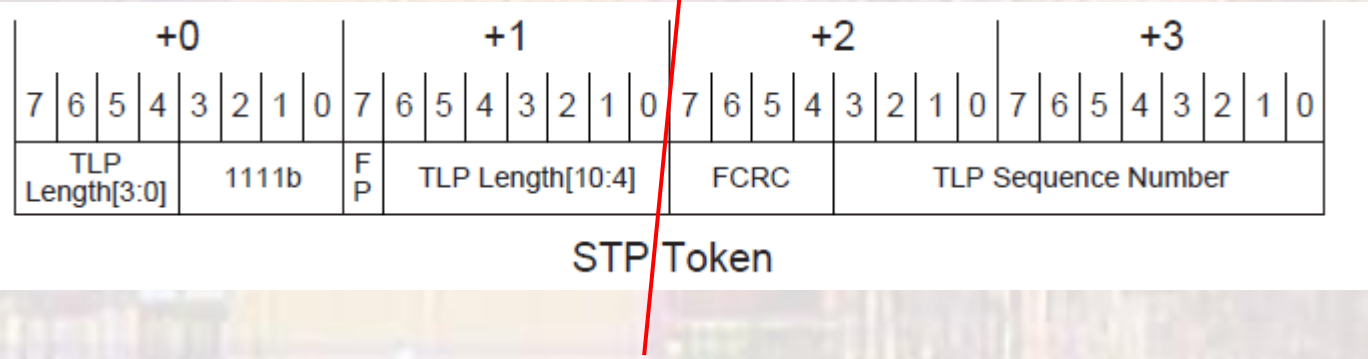
PCIE

- Physical Layer – Logical
  - Framing



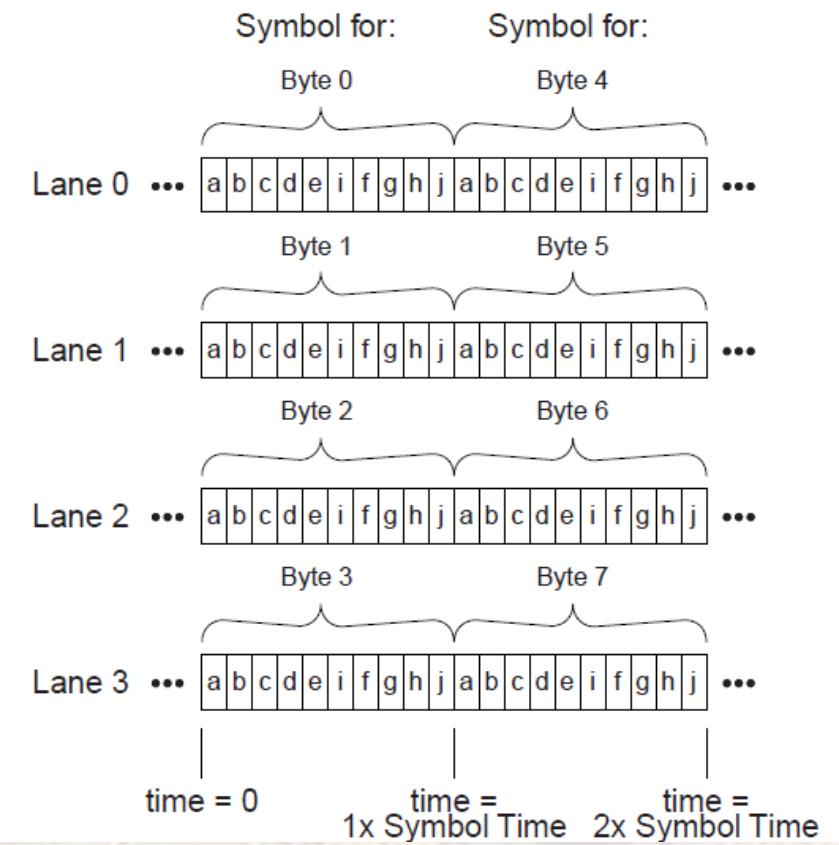
PCIE

- Physical Layer – Logical
  - Framing Token – Start of TLP



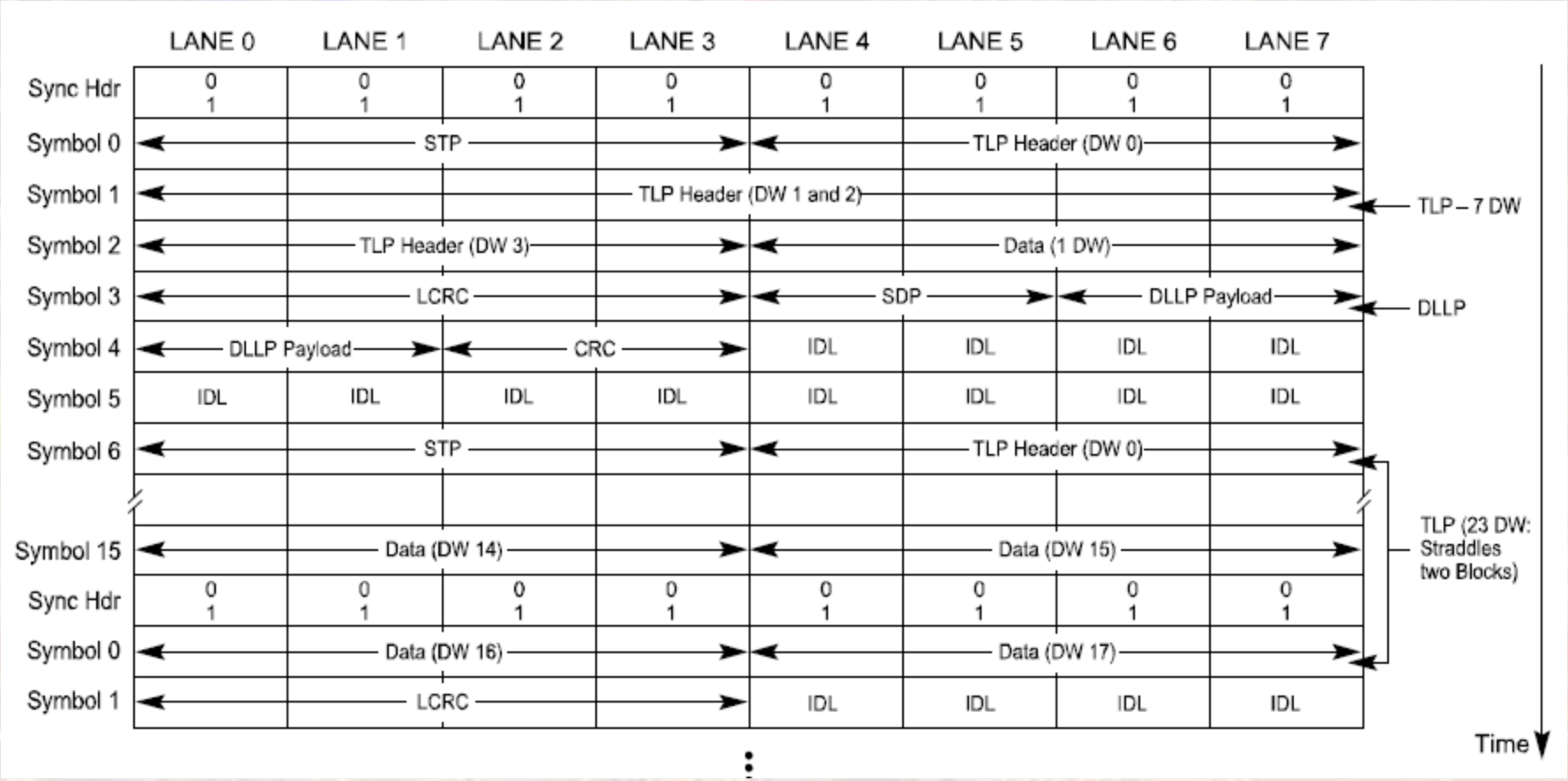
PCIE

- Physical Layer – Logical
  - Striping



PCIE

- Packet example



PCIE

- Physical Layer - Logical
  - Encoding
    - PCIe 1.x and 2.x use 8b/10b encoding
    - PCIe 3.x and 4.x use 128b/130b encoding
  - 8b/10b Encoding
    - Maps 8 bit symbols to 10 bit symbols
    - Coding ensures # of 1s and # of 0s differ by ≤ 2 for any string of 20 bits
    - Coding ensures no more than 5, 0s or 1s in a row
    - Provides enough transitions to do clock recovery
    - Provides for DC balance

PCIE

- Physical Layer - Logical
  - 8b/10b Encoding
    - Bottom 5 bits → 6 bit code

5b/6b code

Input		RD = −1	RD = +1
	EDCBA	abcdei	
D.00	00000	100111	011000
D.01	00001	011101	100010
D.02	00010	101101	010010
D.03	00011	110001	
D.04	00100	110101	001010
D.05	00101	101001	
D.06	00110	011001	
D.07	00111	111000	000111
D.08	01000	111001	000110
D.09	01001	100101	
D.10	01010	010101	
D.11	01011	110100	
D.12	01100	001101	
D.13	01101	101100	
D.14	01110	011100	
D.15	01111	010111	101000

Input		RD = −1	RD = +1
	EDCBA	abcdei	
D.16	10000	011011	100100
D.17	10001	100011	
D.18	10010	010011	
D.19	10011	110010	
D.20	10100	001011	
D.21	10101	101010	
D.22	10110	011010	
D.23 †	10111	111010	000101
D.24	11000	110011	001100
D.25	11001	100110	
D.26	11010	010110	
D.27 †	11011	110110	001001
D.28	11100	001110	
D.29 †	11101	101110	010001
D.30 †	11110	011110	100001
D.31	11111	101011	010100
K.28	11100	001111	110000

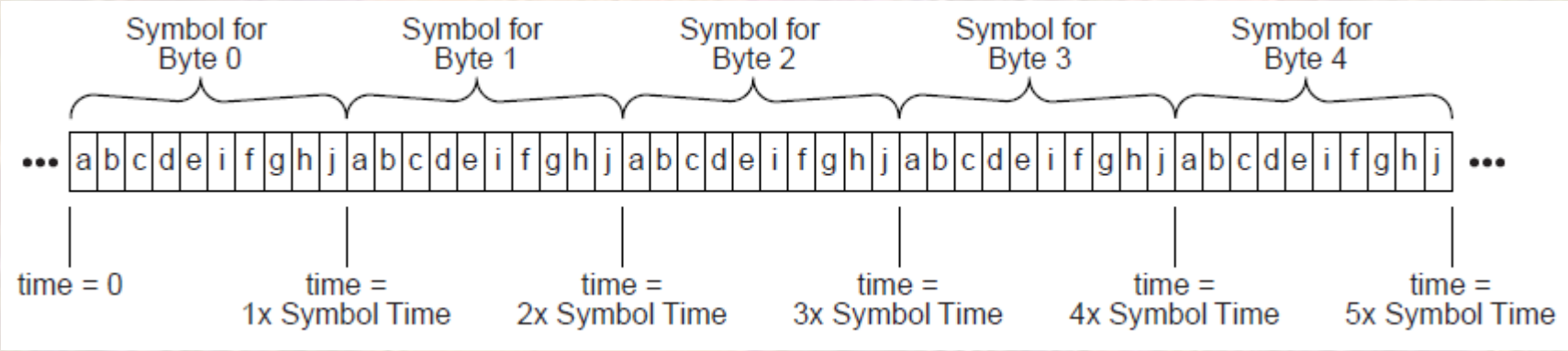
PCIE

- Physical Layer - Logical
  - 8b/10b Encoding
    - Top 3 bits → 4 bit code

3b/4b code							
Input		RD = −1	RD = +1		Input	RD = −1	RD = +1
	HGF	fghj				HGF	fghj	
D.x.0	000	1011	0100		K.x.0	000	1011	0100
D.x.1	001	1001			K.x.1 ‡	001	0110	1001
D.x.2	010	0101			K.x.2 ‡	010	1010	0101
D.x.3	011	1100	0011		K.x.3 ‡	011	1100	0011
D.x.4	100	1101	0010		K.x.4	100	1101	0010
D.x.5	101	1010			K.x.5 ‡	101	0101	1010
D.x.6	110	0110			K.x.6 ‡	110	1001	0110
D.x.P7 †	111	1110	0001					
D.x.A7 †	111	0111	1000		K.x.7 †	111	0111	1000

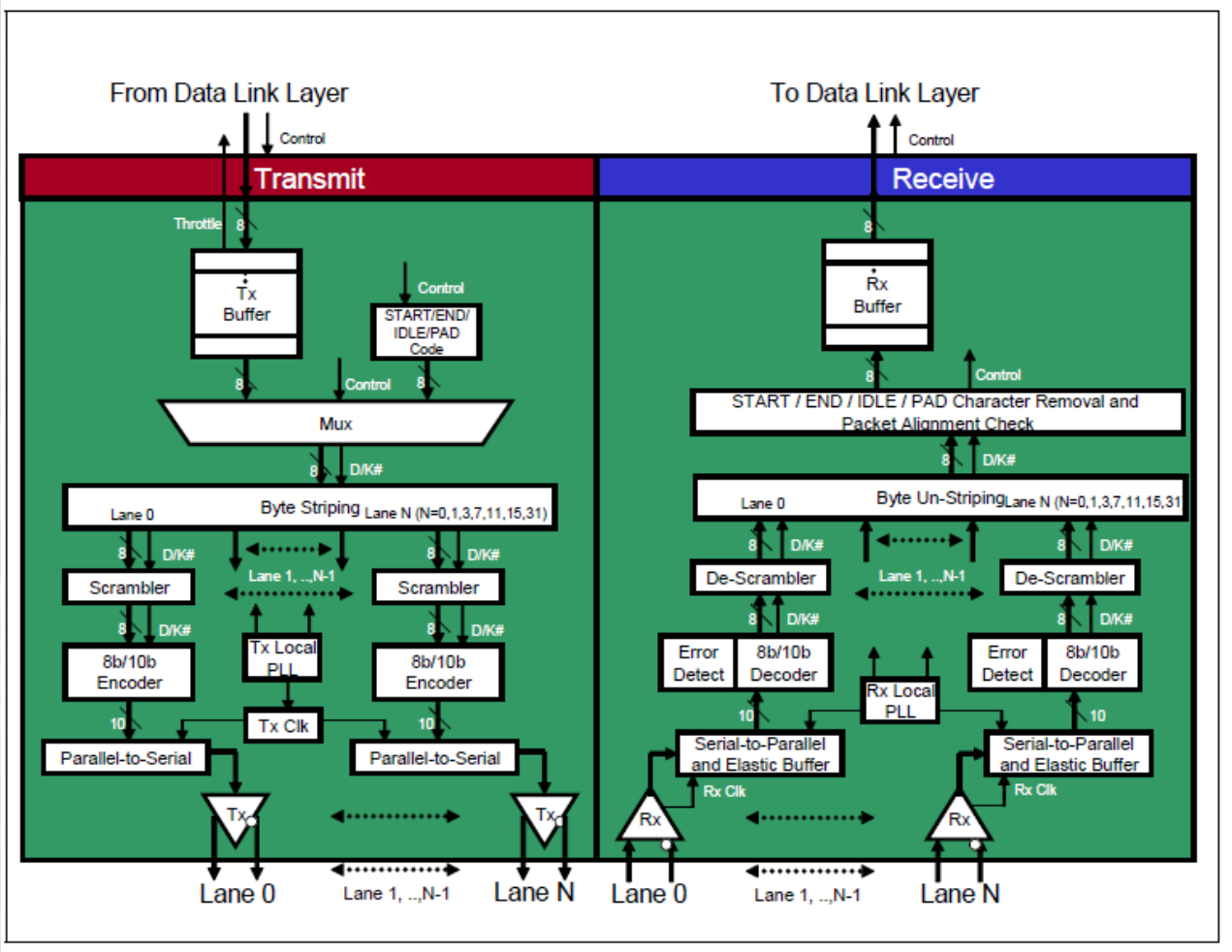
PCIE

- Physical Layer – Logical
  - Symbol Encoding



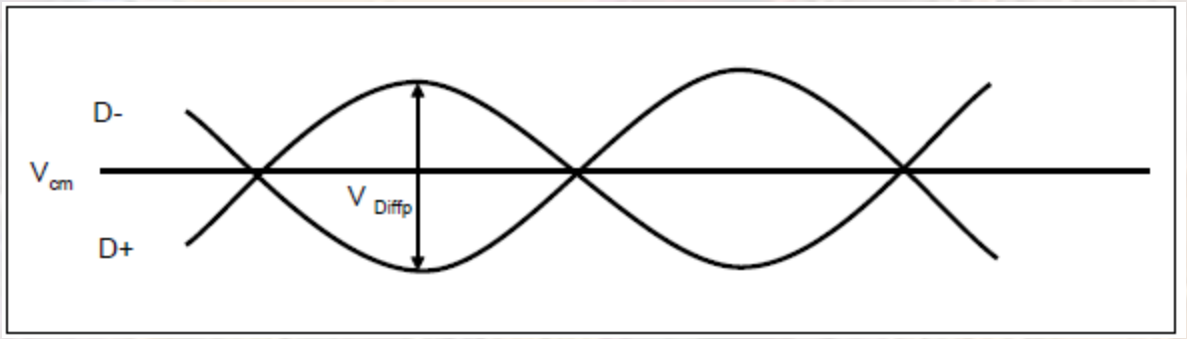
PCIE

- Physical Layer



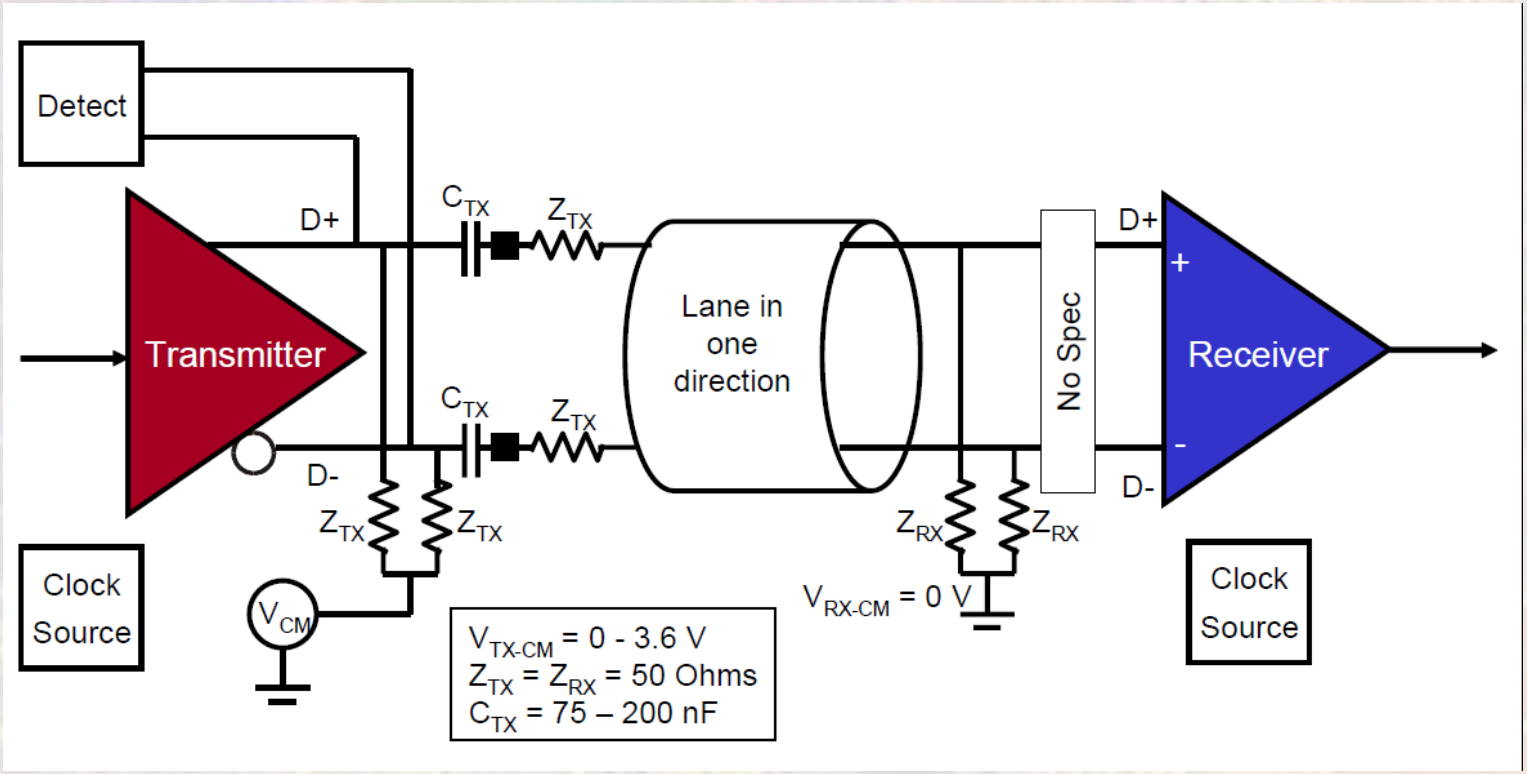
PCIE

- PCIe signaling
  - 2 wires for each transmission
  - Differential signals
    - Differential receivers and transmitters
    - AC coupled → different levels at each end
      - How do we do this ?
    - Differential Pk-Pk voltages from 800mv to 1.2v
      - + Vpp → 1 -Vpp → 0
    - Common mode range from 0 to 3.6v



PCIE

- PCIe signaling
  - LVDS – Low Voltage Differential Signalling



PCIE

- Electrical