FPGAtors

An Overview of PCIe

Omar Elbasri • 10/16/2026 • LAR234



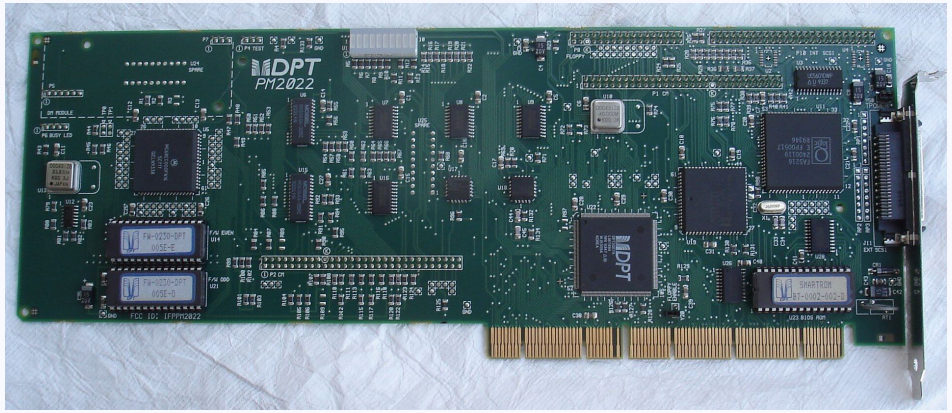
Today’s roadmap

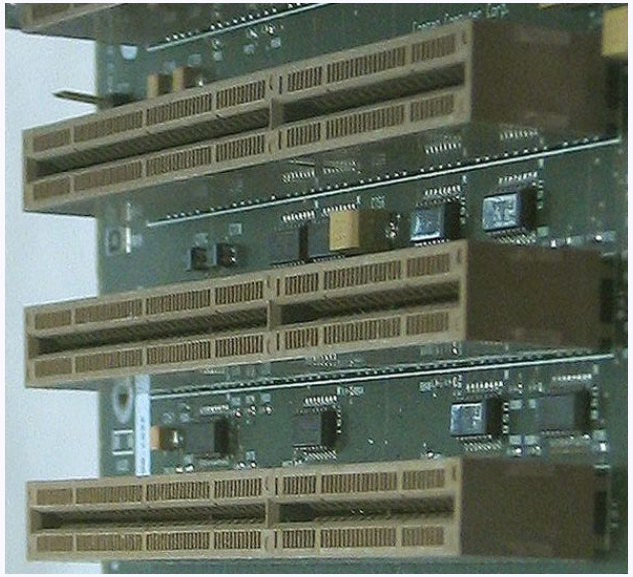
01 History	Where did PCIe come from?
02 Use Cases	When would I ever use PCIe?
03 Technical Overview	How does PCIe work?
04 A small demo	What can I do with PCIe?



Let's set the scene…

- The 1980's was the wild west of computing, everybody wanted expansion cards
- ISA (Industry Standard Architecture) was the… industry standard
- It was an expansion bus that could theoretically reach 8 MB/s (very impressive) but was typically much slower
- Not only that, but it was very painful to work with
  - Jumpers and DIP switches were used to manually set interrupt request numbers and I/O addresses
- Graphics, networking, and faster CPUs quickly outgrew it
- The alternatives were proprietary, expensive, or barely faster

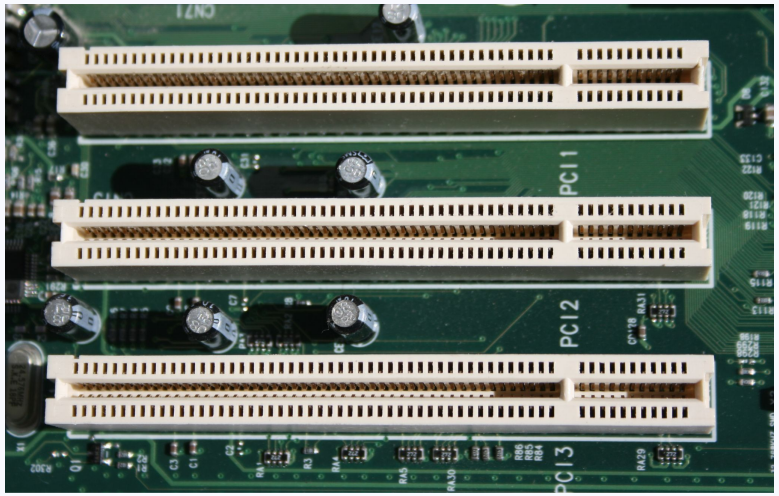




A hero to the rescue!

- The industry needed something **fast**, **CPU-independent**, **self-configuring**, and **open**
- Introducing… PCI (Peripheral Component Interconnect) from Intel in 1992
- 32-bit, 33 MHz, about 133 MB/s. Later revisions went to 64-bit and 66 MHz
- PCI was not only faster, but brought some real innovations to the table:
  - A host bridge sat between the CPU and the bus, so the bus outlived CPUs
  - Configuration space allowed devices to identify themselves and request resources
  - Bus mastering allowed cards to DMA straight to memory
- Shared bus architecture, so devices had to take turns transmitting data (remember this)
- This software model is still alive today! Your OS sees PCIe devices the exact same way it sees PCI devices, more than 30 years later.

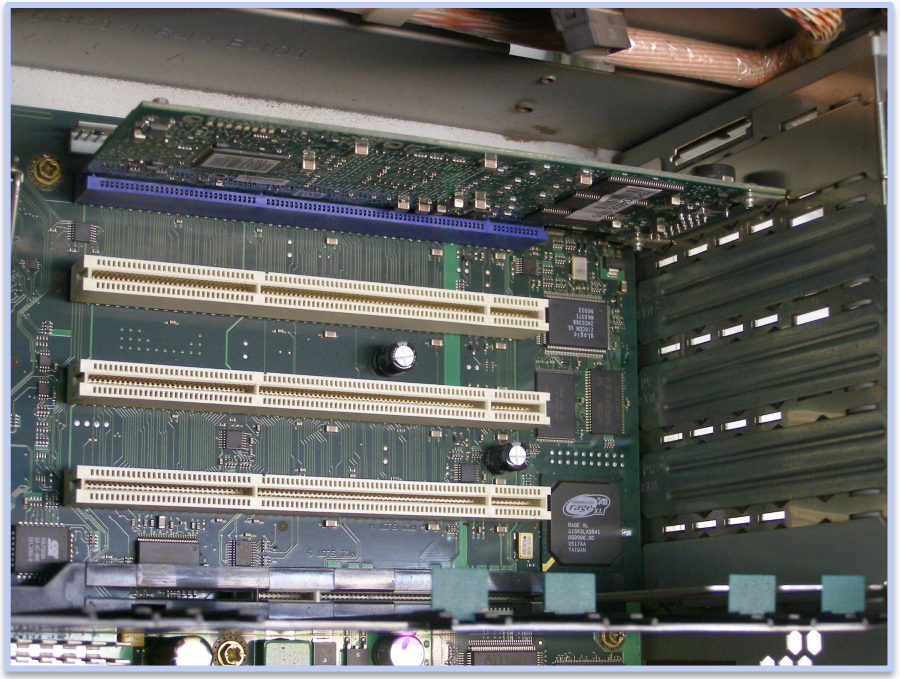




PCI… again

- Then in 1998, PCI-X was released: basically PCI on steroids for servers
- 64 bits at 133 MHz gave transfer rates of up to 1 GB/s!
- Then PCI-X 2.0 in 2003 used DDR/QDR (double/quad data rate) to transmit 2 or 4 bits per clock cycle instead of just 1
- It was also backwards compatible, and PCI cards worked in PCI-X slots
- The real catch was that the faster it went, the fewer slots it could support
- Why was this the case?

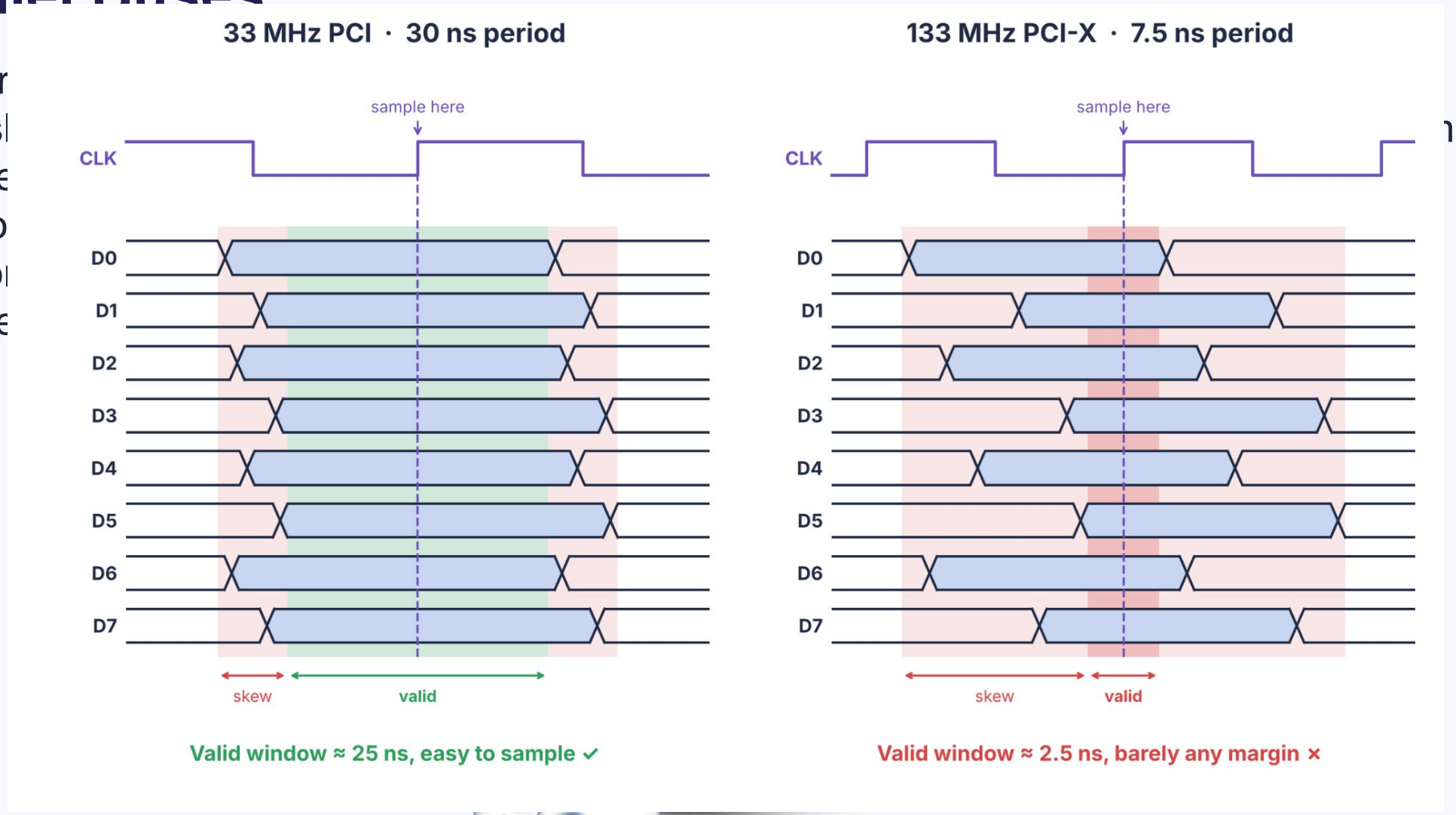
Bus Speed	Slots
66 MHz	4
100 MHz	2
133 MHz	1



Parallel Buses

- Par
- A s
- The
- Clo
- Mo
- The

ne



The REAL hero



- We all know it, we all love it, it’s PCIe!
- What changed?
  - Serial instead of parallel, each lane sends one bit at a time
  - Point-to-point, no more shared bus
  - Differential pairs cancelled out noise
  - Embedded clock, the clock rides with the data
  - Full duplex, both directions at the same time (similar to SPI rather than I2C)
- What was the result?
  - No more shared bus, no more slowest device wins
  - Bandwidth could be expanded by adding lanes (x1, x2, x4, x8, x16)
  - Data travels in packets like a tiny network
  - Same configuration space as PCI, so software didn’t need to be adjusted much

How fast does it go?

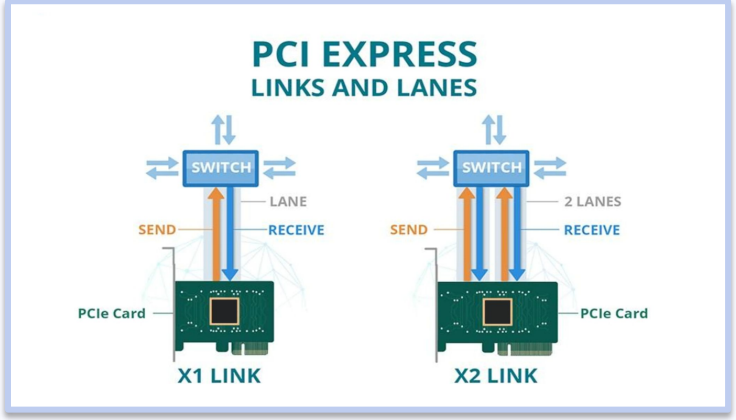
Generation	Year	Data rate	Encoding	x1 (approx)	x16 (approx)	Headline Feature
1	2003	2.5 GT/s	8b/10b	250 MB/s	4 GB/s	Serial, lanes, packets
2	2007	5 GT/s	8b/10b	500 MB/s	8 GB/s	De-emphasis, double speed
3	2010	8 GT/s	128b/130b	985 MB/s	15.8 GB/s	128b/130b, equalization
4	2017	16 GT/s	128b/130b	2 GB/s	31.5 GB/s	Retimers, lane margining
5	2019	32 GT/s	128b/130b	3.9 GB/s	63 GB/s	Alternate Protocols
6	2022	64 GT/s	PAM4 + FLIT	7.6 GB/s	121 GB/s	PAM4, FEC, FLIT
7	2025	128 GT/s	PAM4 + FLIT	15 GB/s	242 GB/s	Optical
8	~2028	256 GT/s	PAM4 + FLIT	—	484 GB/s	In draft

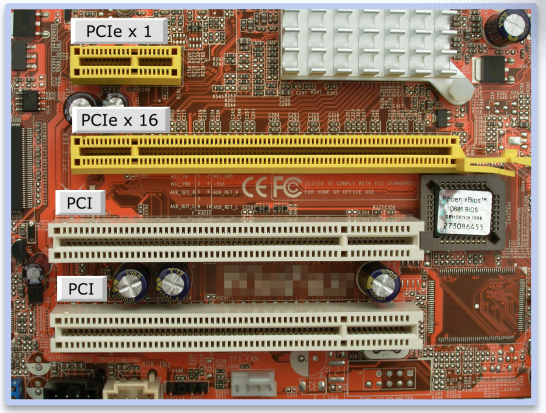


What is it used for? (A lot)

Device	Typical Lanes	Why?
GPUs	x16	Huge bandwidth to and from CPU
NVME SSDs	x4	Replaced SATA's ~550 MB/s ceiling
Network Cards	x8 to x16	400 Gb (that's gigaBIT) needs Gen 5 x16 for full bandwidth
Wi-Fi cards	x1	Laptop Wi-Fi chips
Thunderbolt/USB4	x4 (tunneled)	External GPUs, docks, drives
Phones	x1 to x2	Wi-Fi and modem chips
Accelerators	x4 to x16	FPGAs, AI cards, capture cards, software-defined radios







What is it good at doing?



- PCIe is almost similar to a freight train: it's good at moving a LOT of data
- Benefits:
  - Can move a huge amount of data, tens to hundreds of GB/s
  - DMA allows devices to read/write RAM without bothering the CPU
  - Many requests in flight at once (pipelining, FPGA reference) hide latency
  - Standard discovery process means less setup time per-device
- Drawbacks:
  - AWFUL at tiny transfers, each packet has ~24 bytes of overhead (address, type, CRC) so a 4 byte write would have 86% overhead (yikes!)
  - CPU reading from device = thousands of cycles waiting (which is why DMA is used)
  - Only travels a couple inches on a PCB without extra components/special cables
  - Very complex protocol, requires high speed transceivers + big protocol stack

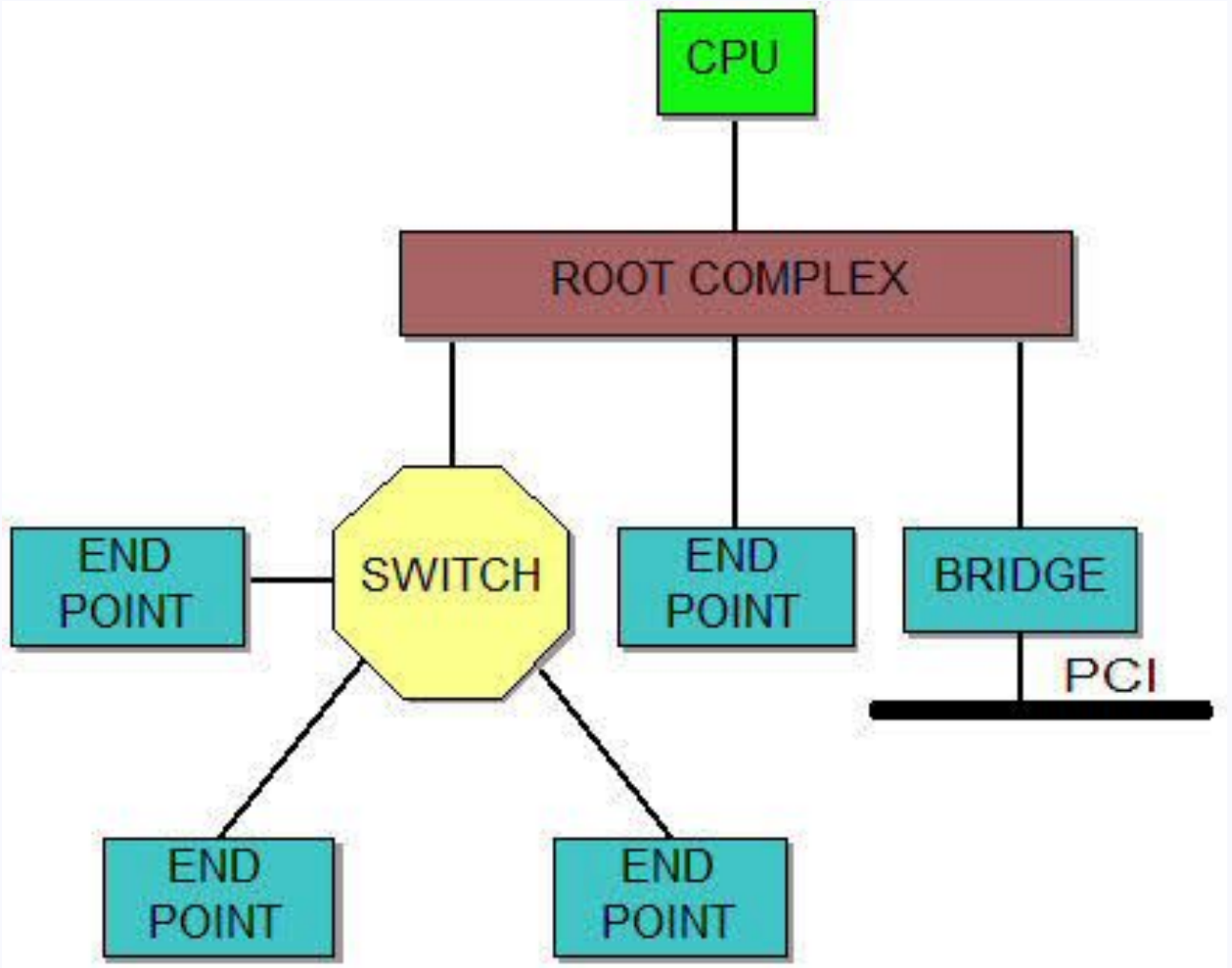
How do I know what to use?

- Generally, your communication protocol will be decided by the hardware you have available
  - Can’t use PCIe without all the necessary components!
- But in some cases (like working with FPGAs) you will have a lot more flexibility
- Other factors (# of devices, compatibility, familiarity, etc.) can also influence these decisions

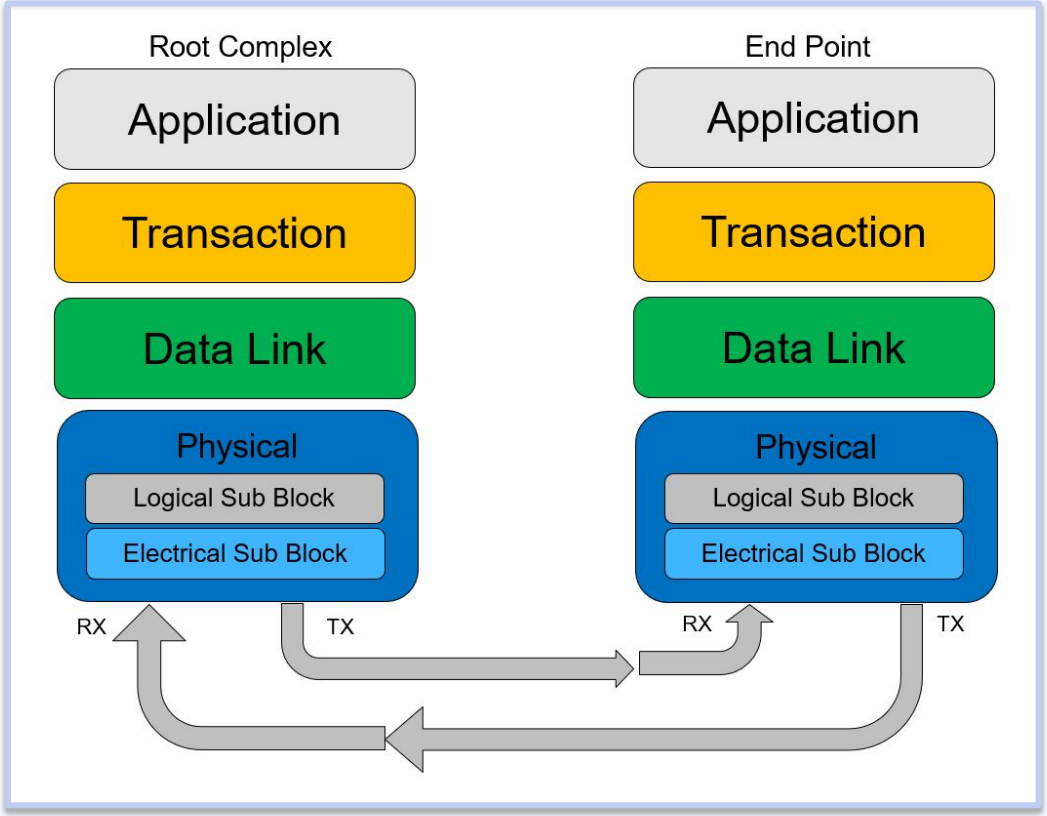
Protocol	Wires	Speed	Reach	Hardware Cost
I2C	2	100 kHz - 1 MHz	~ 1m	Trivial
SPI	4+	~ 10 - 100 MHz	Centimeters	Trivial
UART	2	kbps - Mpbs	Up to ~ 1 km (radio)	Trivial
USB	4+	0.5 - 10GB/s	~ 1 - 5 m	Moderate
PCIe	4 per lane + extras	100+ GB/s	~ 30 cm on board	High
Ethernet	4 - 8	1 - 100 GB/s	100 m copper, km fiber	Moderate to high

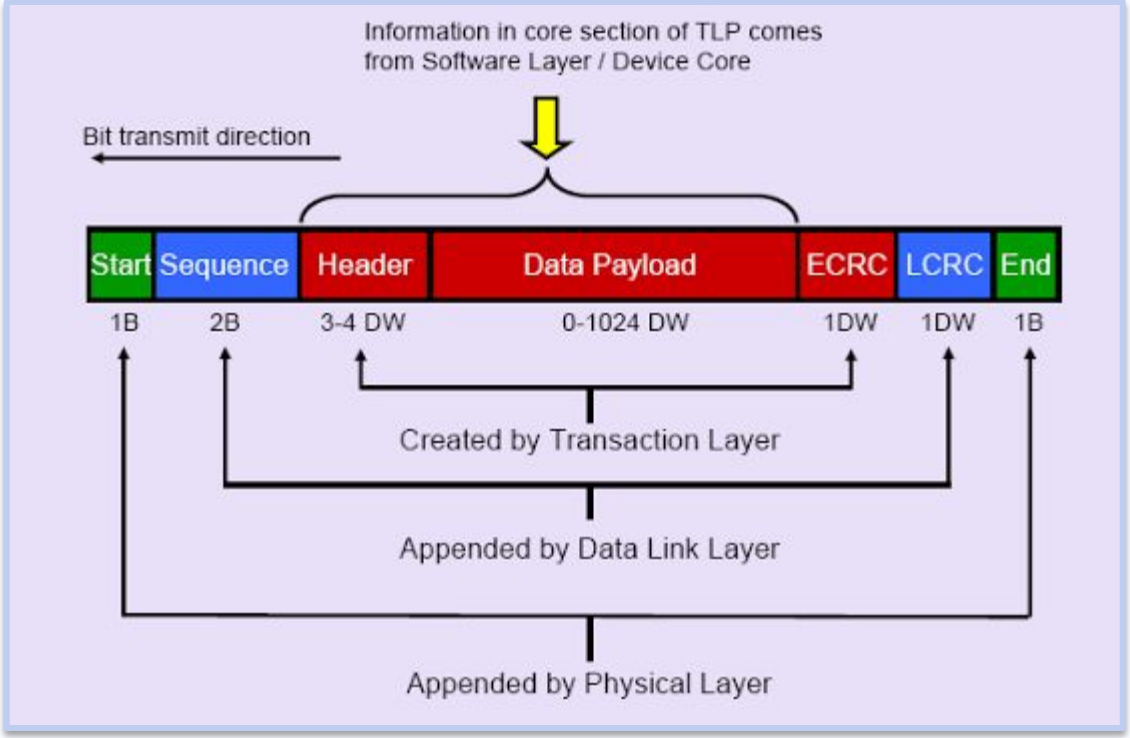
How does it work? (less text more pictures from here on)



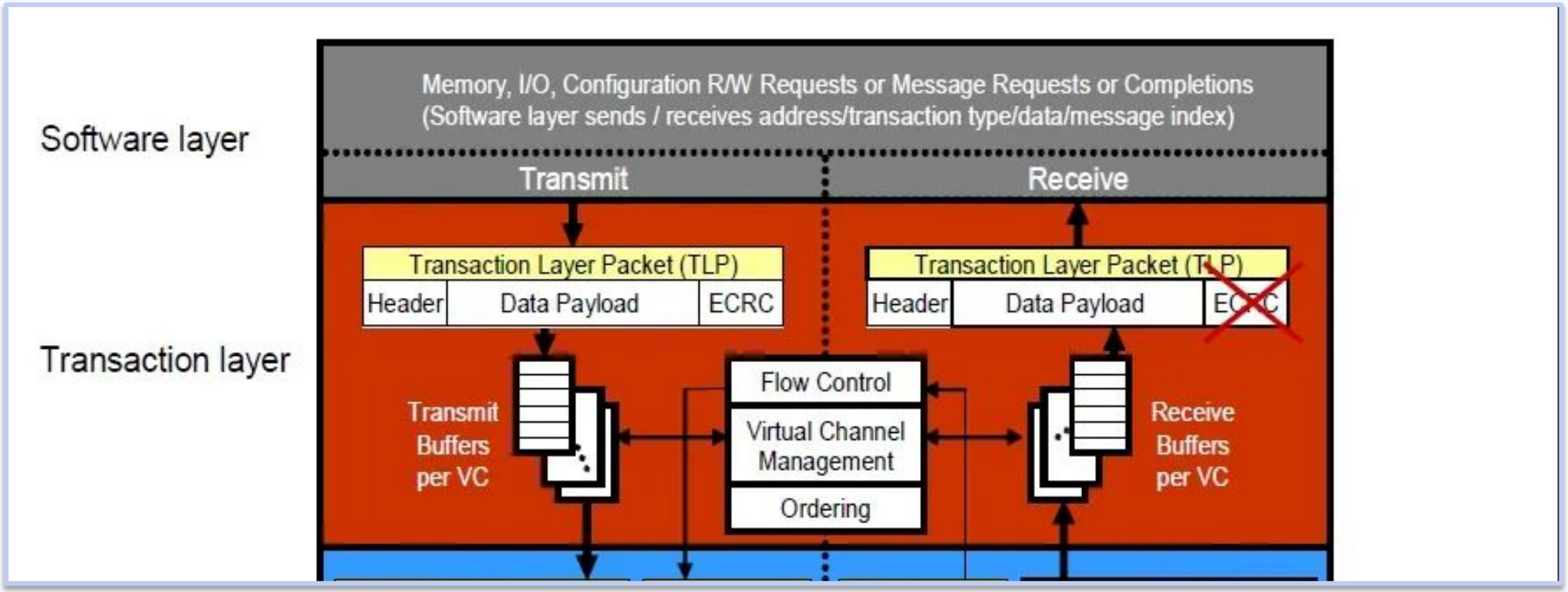


How does it work?

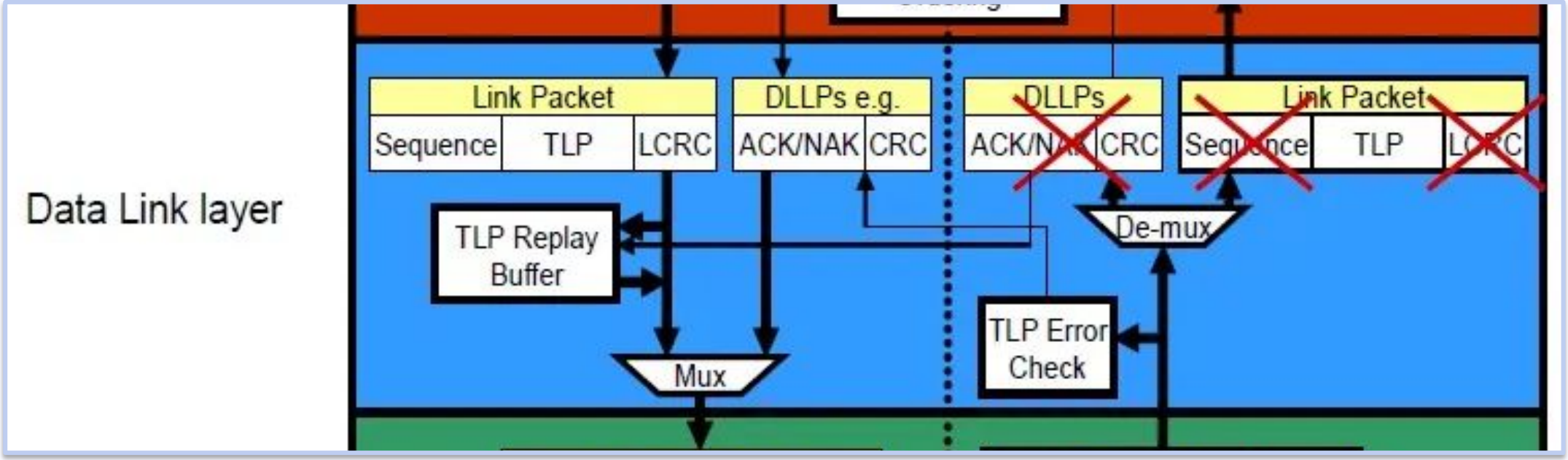




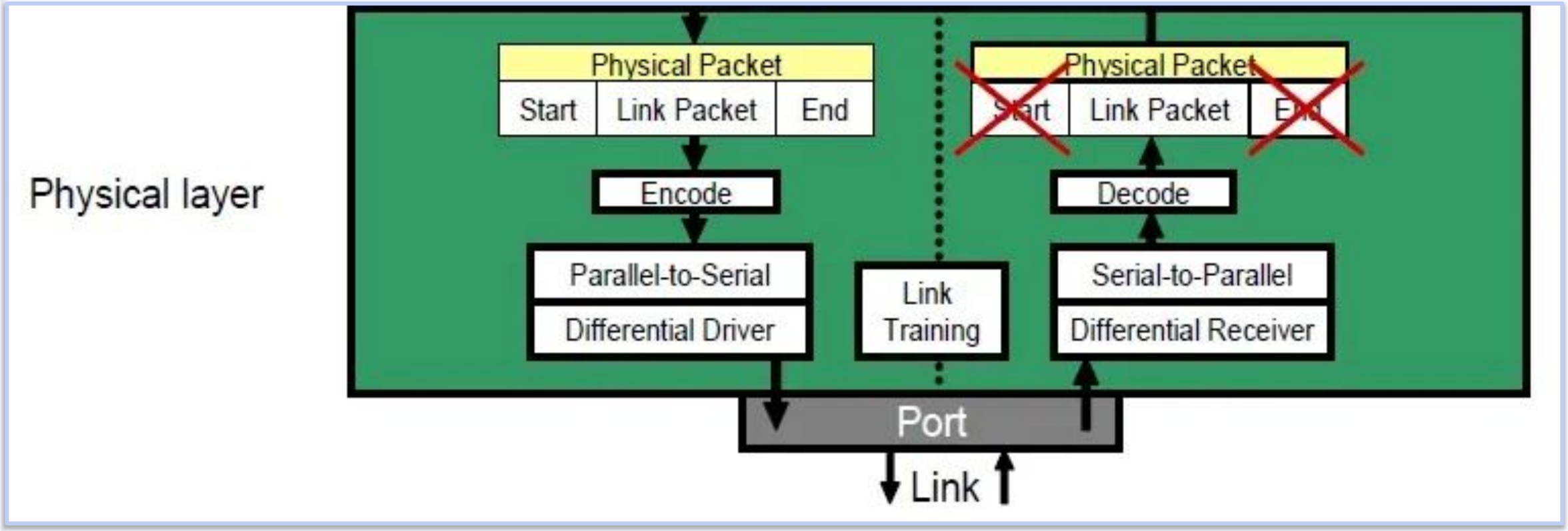
Transaction Layer



Data Link Layer

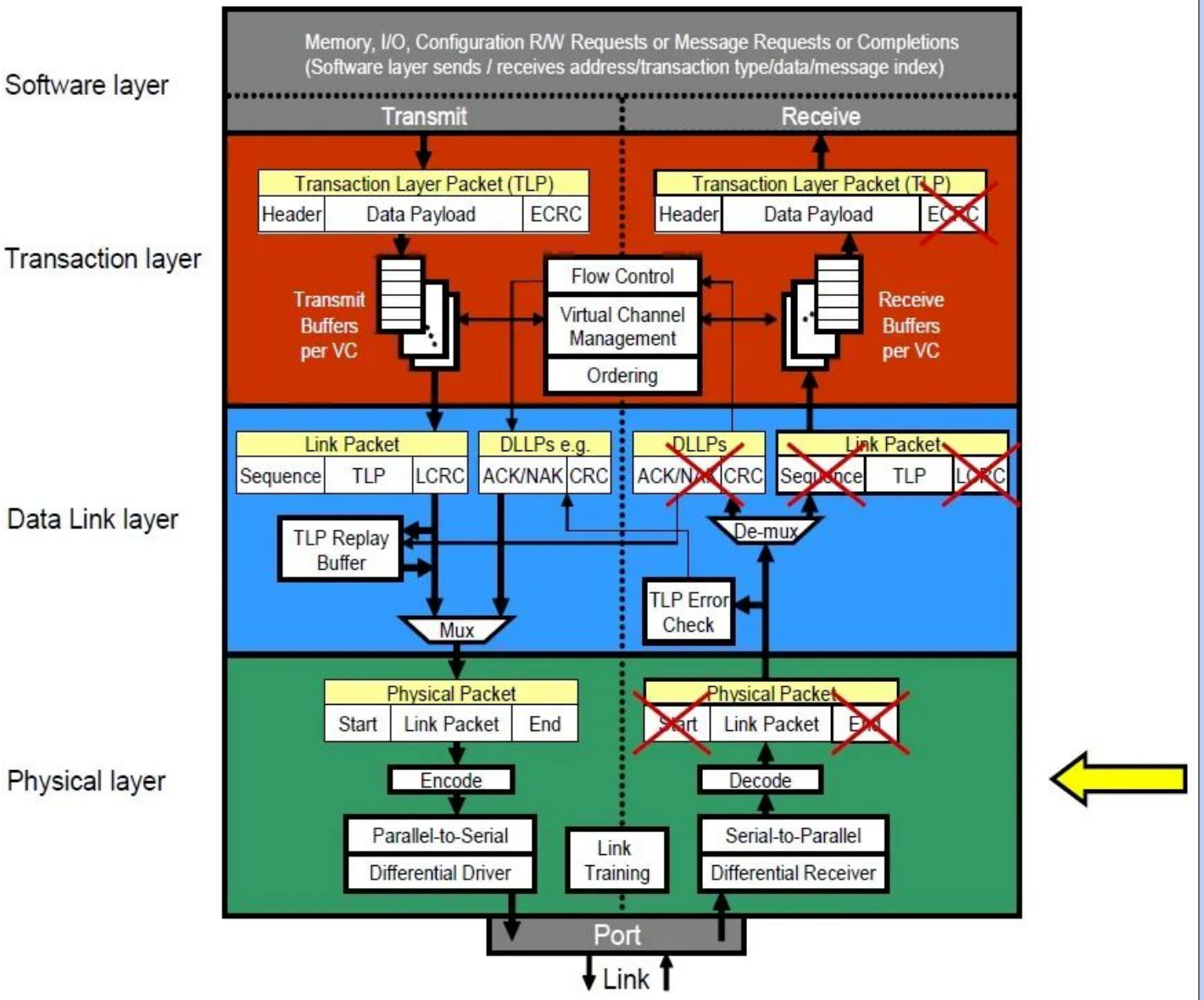


Physical Layer



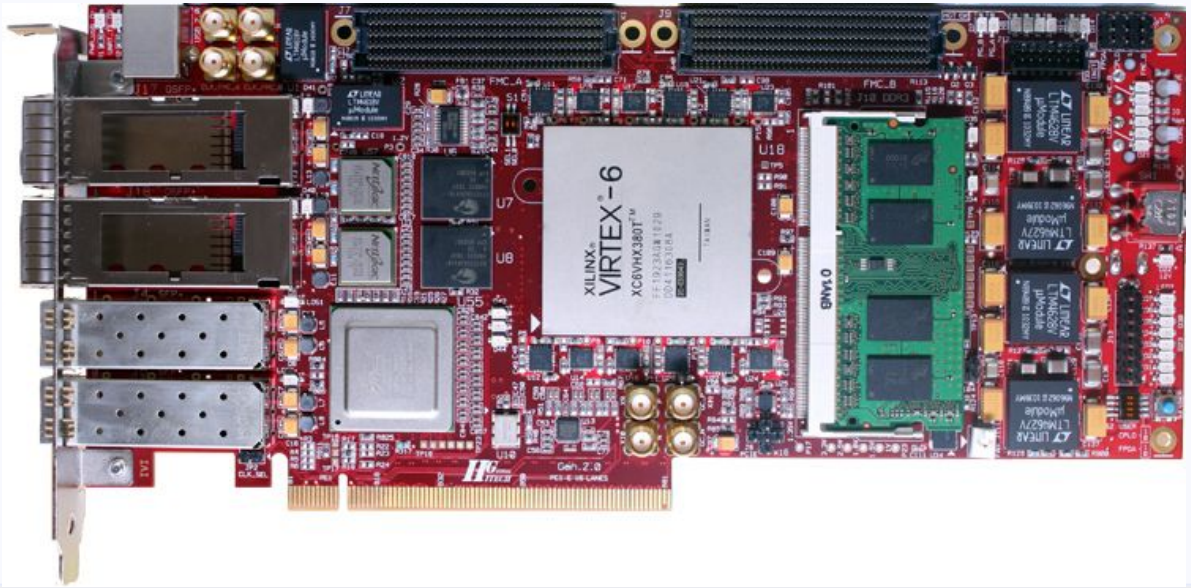
All together now…





But this is FPGAtors? Why should I care!

- Mid to high end FPGAs often have hard PCIe blocks that take care of the lower layers
  - Artix-7: Gen 2 x4, Kintex/Virtex UltraScale+: Gen 3/Gen 4, Versal and Agilex: Gen 5
- DMA IPs (e.g. AMD XDMA/QDMA) also exists for FPGAs to take advantage of PCIe
- Applications:
  - High-frequency trading (everybody loves that), latency sensitive applications
  - Network cards, video transcoding, compute offload, etc.
- Can also be used for prototyping ASICs that will have PCIe
- Some FPGAs even have the CEM goldfinger to be plugged directly into a computer





TIME FOR A DEMO!





FPGAtors

Questions?



