

Resposta: por que um tiranossauro não consegue coçar as costas?

Interconexões

Paulo Ricardo Lisboa de Almeida

A CPU precisa se comunicar

Memória principal, armazenamento, teclado, monitor, placa de vídeo, ...

Barramento (bus)

Barramento (bus).

Caminho de comunicação entre dois ou mais dispositivos.

Compartilhado quando usado para mais de dois dispositivos.

Um sinal enviado é visto por todos dispositivos conectados.

Se dois dispositivos tentam enviar o sinal ao mesmo tempo, o resultado vai ser distorcido e incompreensível.

Necessária arbitragem.

Barramento (bus)

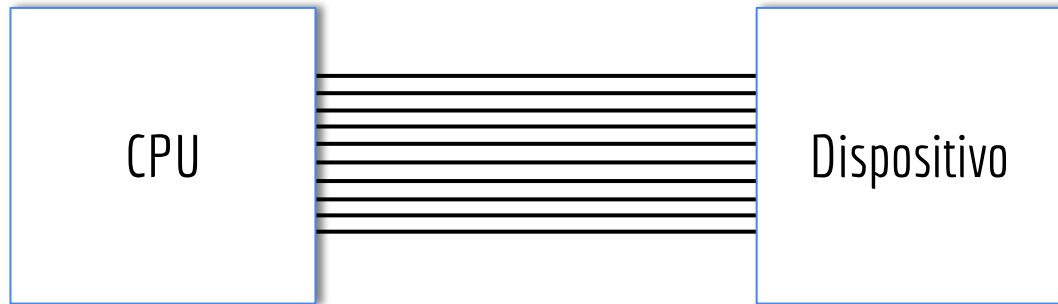
Um barramento pode ser visto como um conjunto de fios.

Cada fio pode transferir um bit por vez.

Adicionar mais fios aumenta a taxa de transmissão.

16, 32, 64, ... fios.

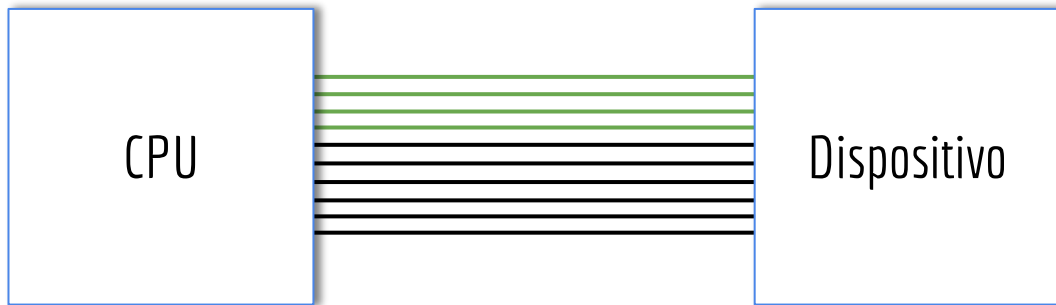
Barramento (bus)



Barramento (bus)

Três categorias básicas:

Linhas de endereço: qual o endereço (e.g., de memória) de onde o dado deve vir?



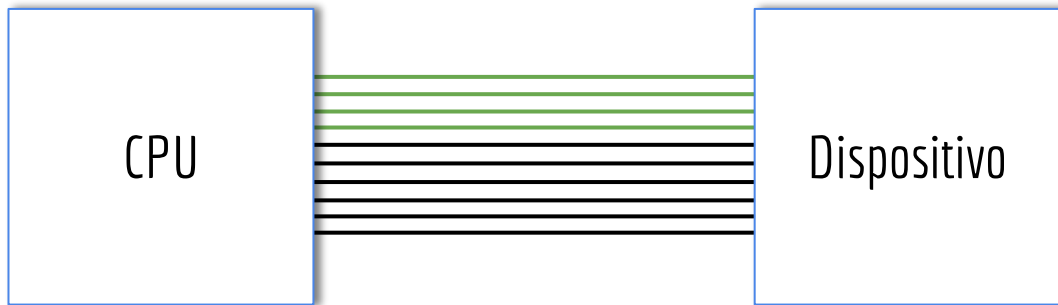
Barramento (bus)

Três categorias básicas:

Linhas de endereço: qual o endereço (e.g., de memória) de onde o dado deve vir?

Limitação: se temos poucas linhas de endereço, poucos endereços são possíveis (e.g., pouca memória mapeável).

Mitigação: enviar o endereço usando mais de um ciclo de clock.

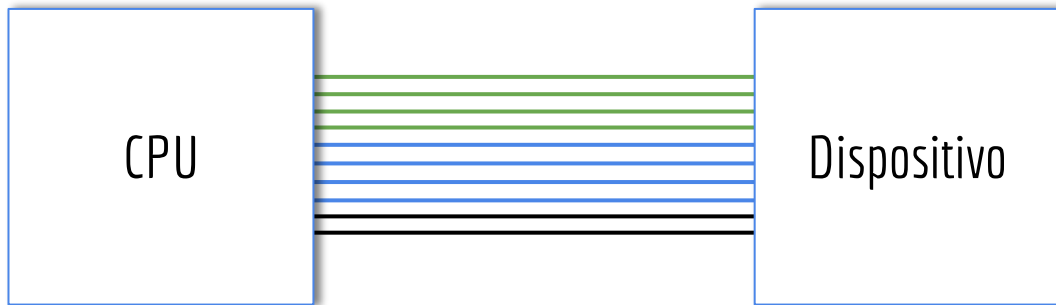


Barramento (bus)

Três categorias básicas:

Linhas de endereço: qual o endereço (e.g., de memória) de onde o dado deve vir?

Linhas de dados: Formam o barramento de dados, e efetivamente transferem a informação.



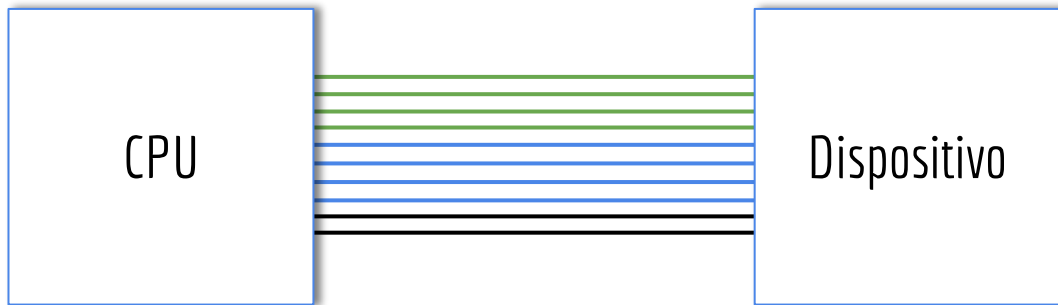
Barramento (bus)

Três categorias básicas:

Linhas de endereço: qual o endereço (e.g., de memória) de onde o dado deve vir?

Linhas de dados: Formam o barramento de dados, e efetivamente transferem a informação.

Linhas de controle: clock, memWrite, memRead, requisição de bus, concessão de bus, ...



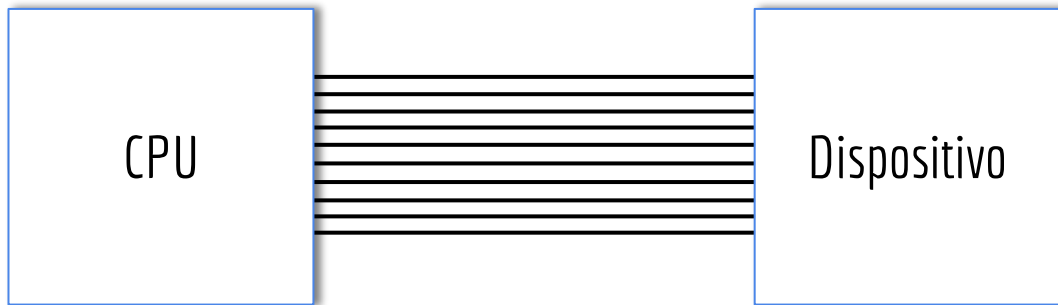
Barramento (bus)

Podemos ainda compartilhar as linhas.

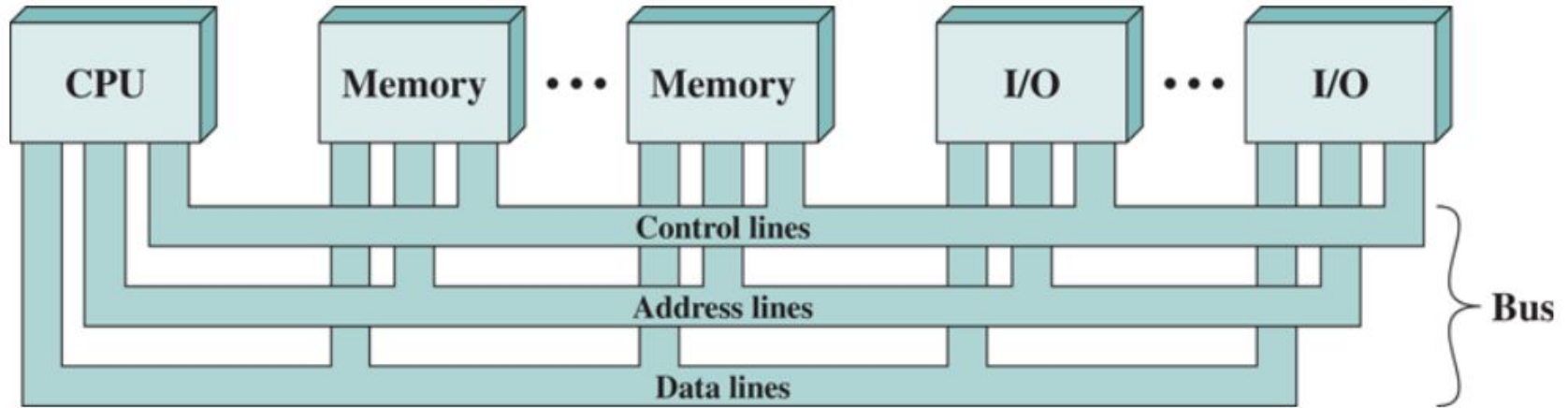
Por exemplo, compartilhar endereço e dados.

No primeiro ciclo, é enviado o endereço.

No próximo ciclo, é enviado o dado.



Barramento (bus)



Stallings, W. (2016).

Barramento (bus)

Barramentos podem ser...

De acordo com o sincronismo:

- Síncronos (por clock).

- Assíncronos (handshake).

De acordo com a arbitragem:

- Arbitragem centralizada.

 - Uma central (e.g., CPU) define quem vai usar o barramento, e quando.

- Arbitragem distribuída.

 - Vários árbitros definem o uso do barramento em conjunto.

Problemas

Barramentos caíram em desuso.

São caros e complicados.

Problemas

Barramentos caíram em desuso.

São caros e complicados.

- Muitas linhas em paralelo podem gerar interferências umas nas outras;

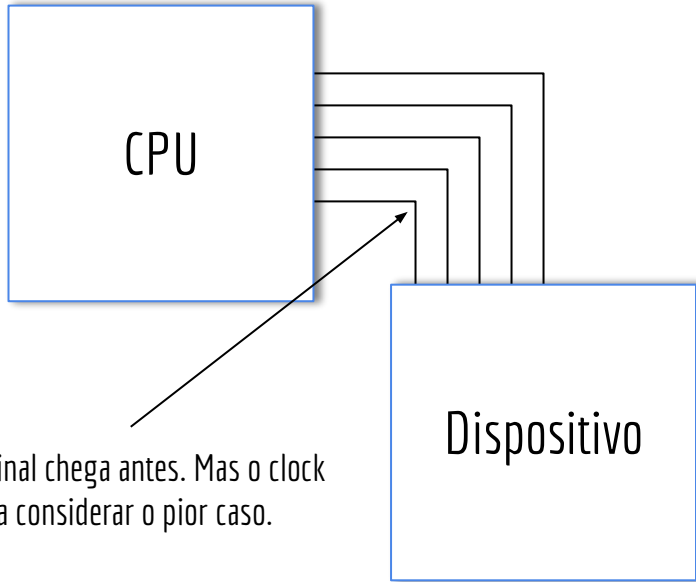
- Enviar sinais de clock em linhas longas é um problema.

 - Leia sobre clock skew.

- Restrições elétricas.

- Muitos dispositivos rápidos competindo pelo uso do barramento geram overheads e gargalos.

Problemas



Esse sinal chega antes. Mas o clock precisa considerar o pior caso.

Onde são usados

É comum o uso de barramentos na confecção de circuitos com microcontroladores.

Alguns barramentos dedicados também são comuns, como os barramentos que ligam diretamente a memória principal (DRAM) à CPU na maioria das placas-mãe.



Table 5 — 240-Pin DDR3 SDRAM Pin Assignment (Part 1 of 2)

Pin #	Front Side	Pin #	Back Side	Pin #	Front Side	Pin #	Back Side	Pin #	Front Side	Pin #	Back Side	Pin #	Front Side	Pin #	Back Side
1	VREFDQ	121	VSS	31	DQ25	151	VSS	61	A2	181	A1	91	DQ41	211	VSS
2	VSS	122	DQ4	32	VSS	152	DM3, DQS12_t, TDQS12_t	62	VDD	182	VDD	92	VSS	212	DM5, DQS14_t, TDQS14_t
3	DQ0	123	DQ5	33	DQS3_c	153	NC, DQS12_c, TDQS12_c	63	NC, CK1_t	183	VDD	93	DQS5_c	213	NC, DQS14_c, TDQS14_c
4	DQ1	124	VSS	34	DQS3_t	154	VSS	64	NC, CK1_c	184	CK0_t	94	DQS5_t	214	VSS
5	VSS	125	DM0, DQS9_t, TDQS9_t	35	VSS	155	DQ30	65	VDD	185	CK0_c	95	VSS	215	DQ46
6	DQS0_c	126	NC, DQS9_c, TDQS9_c	36	DQ26	156	DQ31	66	VDD	186	VDD	96	DQ42	216	DQ47
7	DQS0_t	127	VSS	37	DQ27	157	VSS	67	VREFCA	187	EVENT_n, NC	97	DQ43	217	VSS
8	VSS	128	DQ6	38	VSS	158	CB4, NC	68	PAR_IN, NC	188	A0	98	VSS	218	DQ52
9	DQ2	129	DQ7	39	CB0, NC	159	CB5, NC	69	VDD	189	VDD	99	DQ48	219	DQ53
10	DQ3	130	VSS	40	CB1, NC	160	VSS	70	A10/AP	190	BA1	100	DQ49	220	VSS

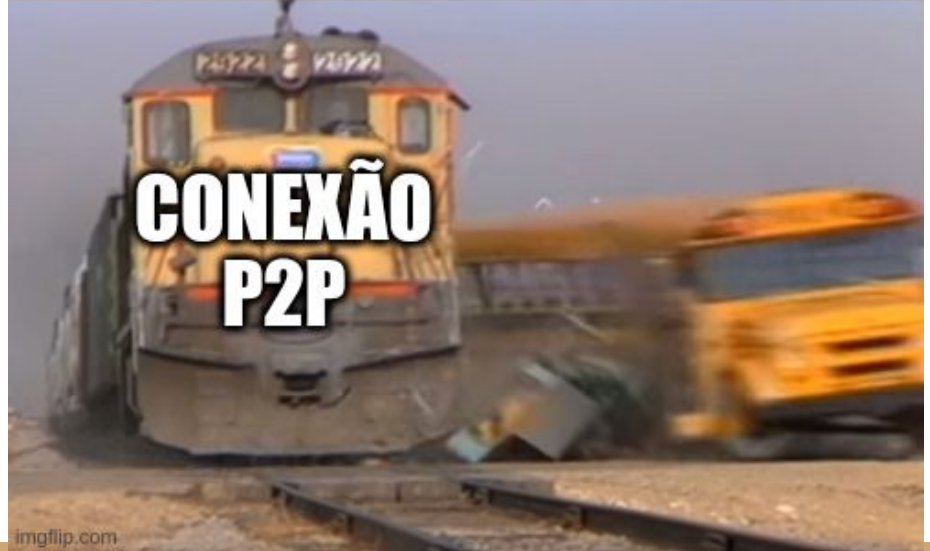
...

Ponto a Ponto

Boa parte das interconexões entre os componentes atualmente são **ponto-a-ponto**.

Algumas características comumente encontradas:

- Múltiplas conexões diretas (*pairwise*).
- Protocolos multicamadas ao invés de sinais de controle (como em uma rede TCP/IP).
- Transferência de dados em pacotes.



PCI Express

PCI Express - PCIe.

Padrão atual para comunicação com dispositivos (e.g., GPU, Placa de Som, Placa Ethernet, ...).

Padrão PCI foi criado originalmente em 1990 pela Intel.

Baseado em barramento.

PCIe atual é baseado em conexão **ponto a ponto**.



ASUS ROG STRIX X570-E

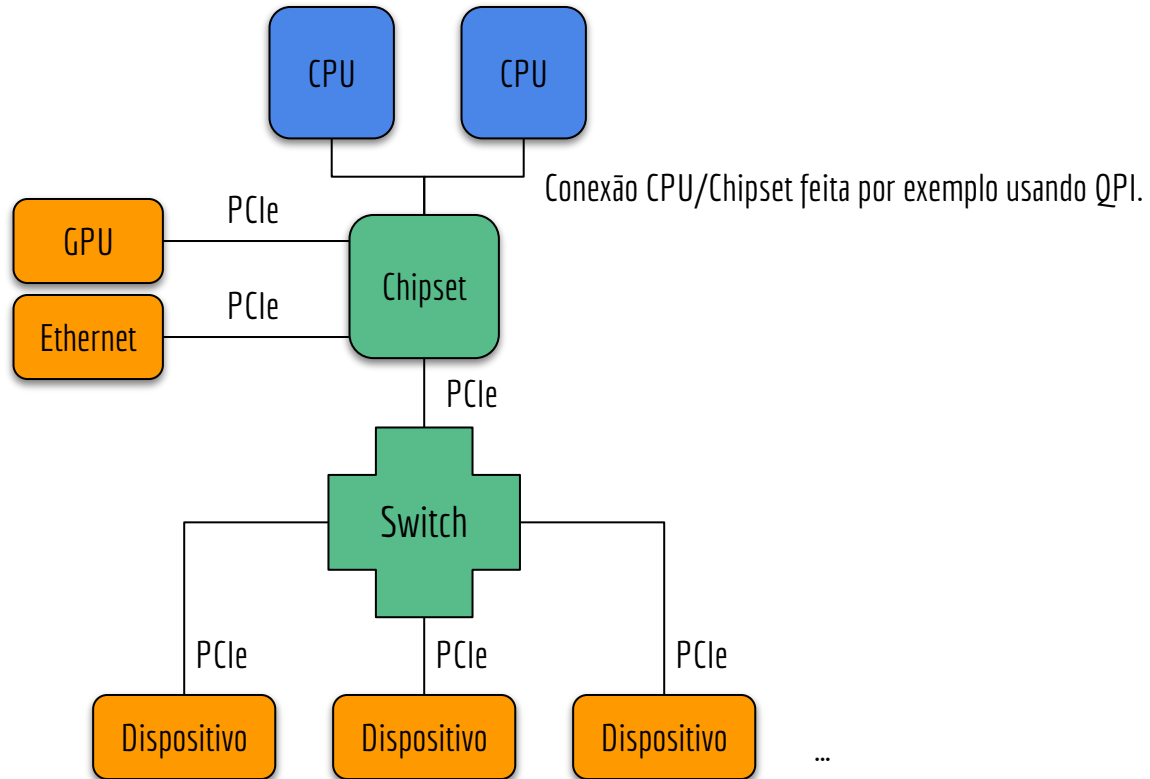
Versões

Existem várias versões do padrão.

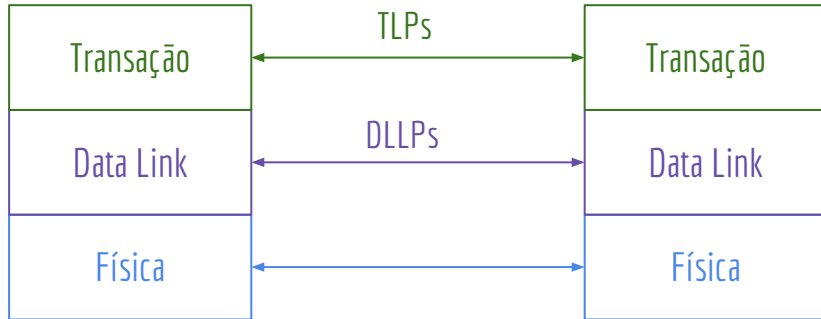
Os demais slides se referem ao padrão PCIe 3.0, que ainda é usado em boa parte dos computadores.

Versões posteriores como a PCIe 4.0 mudam detalhes com frequência máxima de operação.

Esquema Típico



Camadas do Protocolo PCIe



Camada Física

A camada física é composta por **lanes** (faixas).

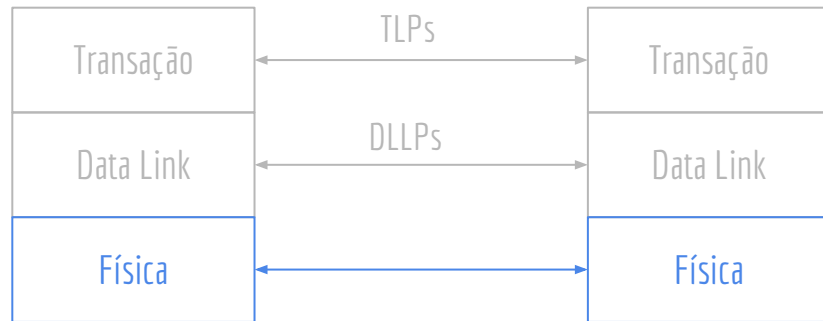
Cada lane é bidirecional.

Envio e recepção simultânea.

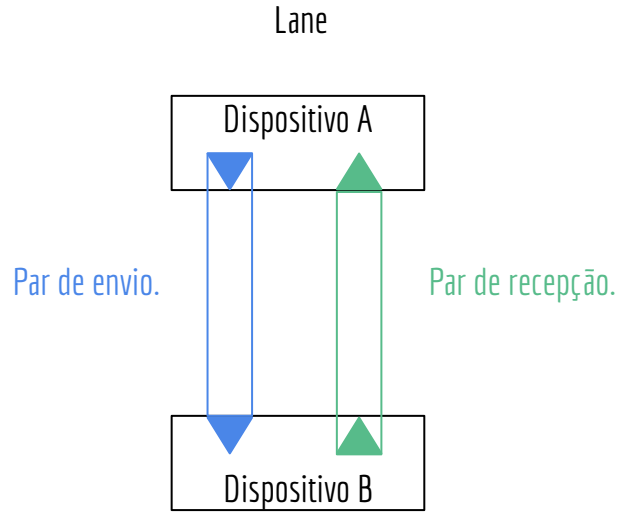
Sinalização Diferencial utilizando pares de fios.

A direção da corrente diz se o valor binário é 0 ou 1.

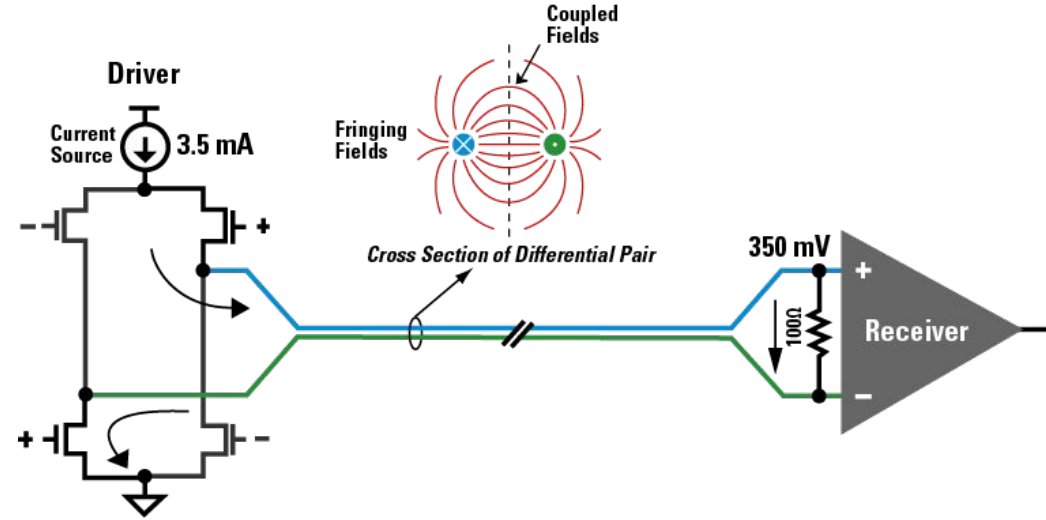
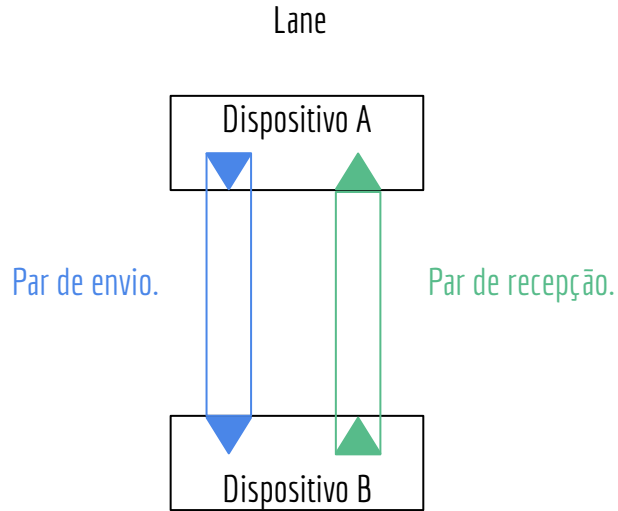
Corrente igual e em direções opostas nos pares de fios reduz ruídos eletromagnéticos.



Lane



Lane



en.wikipedia.org/wiki/File:Basic_LVDS_circuit_operation.png

Sincronização

Cada lane transmite uma stream de 128 bits (16 Bytes) por vez , mais 2 bits de controle, totalizando 130-bits.

Os bits são carregados em um buffer, e enviados um por vez (não são 130 fios paralelos).

Existe um clock apenas para algumas operações básicas de sinalização.

O clock não é usado para sincronizar a transmissão.

Através da sinalização diferencial, o receptor precisa **inferir automaticamente** o clock do segmento de 130 bits.

Sincronização

Se o pacote possuir uma sequência longa de 0's ou 1's, fica difícil inferir as transições e qual deve ser o período de clock.

No início da transmissão é inserido um cabeçalho de controle de 2 bits.

- 10 para bloco de dados.

- 01 para informação de bloco de camada de link (veremos a seguir).

Um circuito de scrambling altera a stream para aumentar a probabilidade de transições entre 0's e 1's no sinal.

- O receptor desfaz essa operação para obter o sinal original.

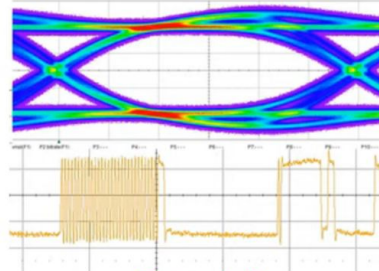
Sincronização

O sinal ainda é alterado de forma que sua forma de onda se encaixe em uma de 11 formas de onda padrão.

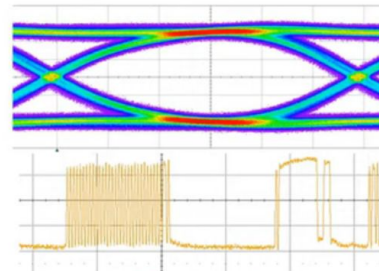
Minimizar interferências entre múltiplos lanes.

Veja esse vídeo da Texas Instruments.

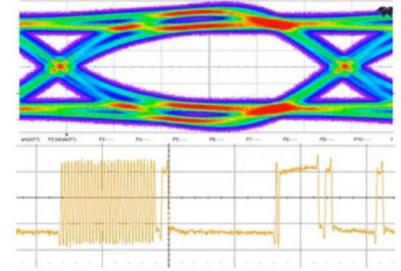
<https://youtu.be/L9qXjmJdQXY>



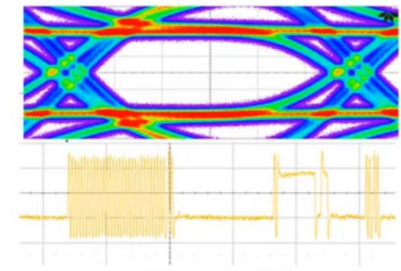
Preset 0



Preset 2

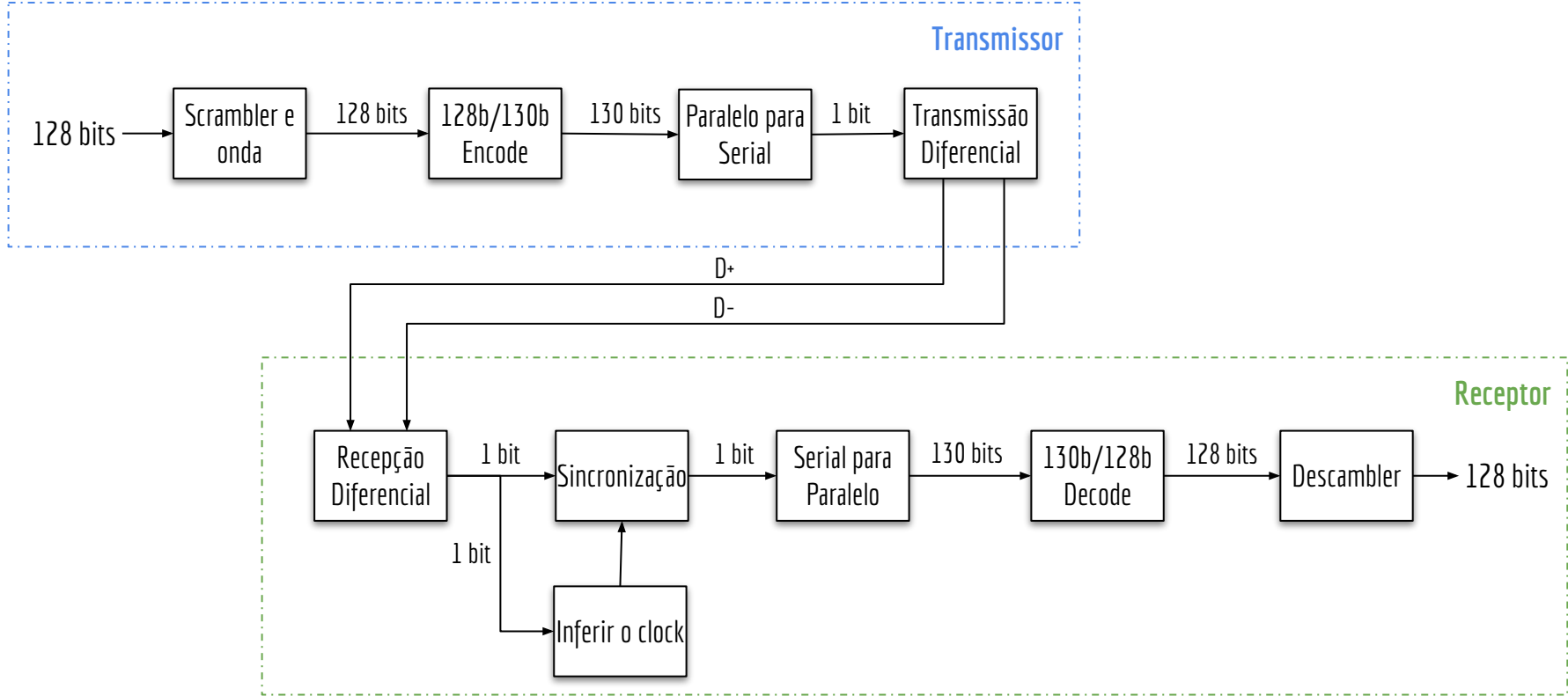


Preset 7



Preset 10

Esquema para um lane



Múltiplos Lanes

Um **lane** então é um esquema de transmissão serial.

Um bit por vez.

Podemos aumentar a velocidade transmitindo os bits mais rapidamente, ou adicionando mais lanes.

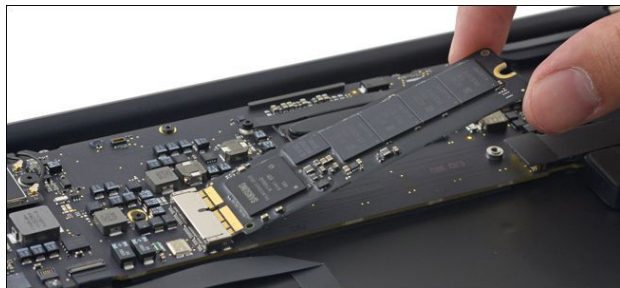
Múltiplos Lanes

O número de Lanes geralmente é de 1, 4, 8 ou 16.

Fabricantes chamam de PCIe x1, PCIe x2, ... PCIe x16.



m.2 PCIe



PCIe x1

PCIe x16



Múltiplos Lanes

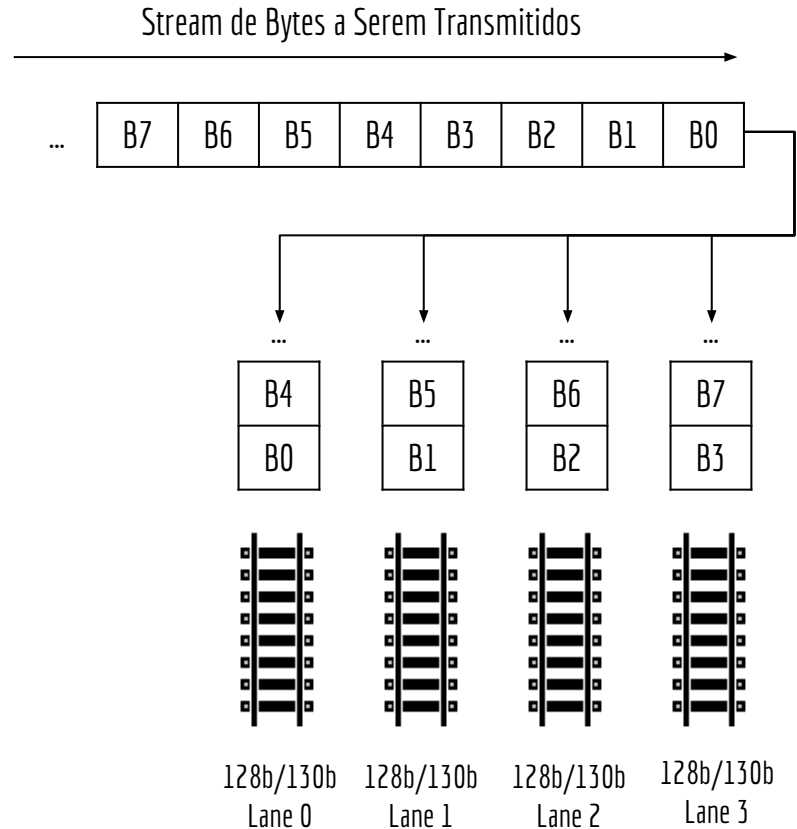
Usando múltiplos Lanes, a *stream* (fluxo) de Bytes é distribuída usando Round-Robin nos Lanes.

1^o Byte no 1^o Lane, 2^o Byte no 2^o Lane, ...

Quando acabam os Lanes, volta para o primeiro.

Quando um Lane junta 16 Bytes, os dados são despachados.

Exemplo para 4 Lanes.



Taxas de Transferência

Versão	Ano da Especificação	Encoding	Taxa de transferência.				
			x1	x2	x4	x8	x16
1.0	2003	8b/10b	0,25 GB/s	0,5 GB/s	1GB/s	2GB/s	4GB/s
2.0	2007	8b/10b	0,5 GB/s	1 GB/s	2 GB/s	4 GB/s	8 GB/s
3.0	2010	128b/130b	0,98 GB/s	1,97 GB/s	3,94 GB/s	7,88 GB/s	15,76 GB/s
4.0	2017	128b/130b	1,97 GB/s	3,94 GB/s	7,88 GB/s	15,75 GB/s	31,5 GB/s
5.0	2019	128b/130b	3,94 GB/s	7,88 GB/s	75,75 GB/s	31,5 GB/s	63,01 GB/s

Camada de Transação

A camada de transação (TL) recebe as requisições de envio do software que está acima dela, e transforma a requisição em pacotes.

Transaction Layer Packets - **TLPs**.

Muitas das transações são na forma requisição e resposta:

Pacote de requisição é enviado por um dispositivo fonte.

Dispositivo aguarda resposta - pacote de conclusão.

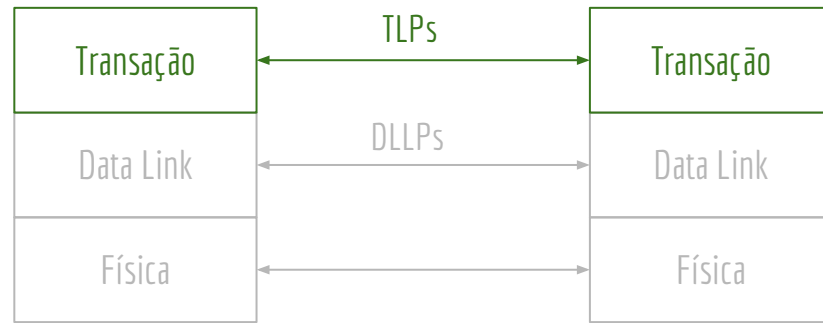
A resposta é enviada pelo dispositivo requisitado **somente quando ele está pronto para fazer a entrega.**

Enquanto isso, outros dispositivos podem usar o canal de comunicação.

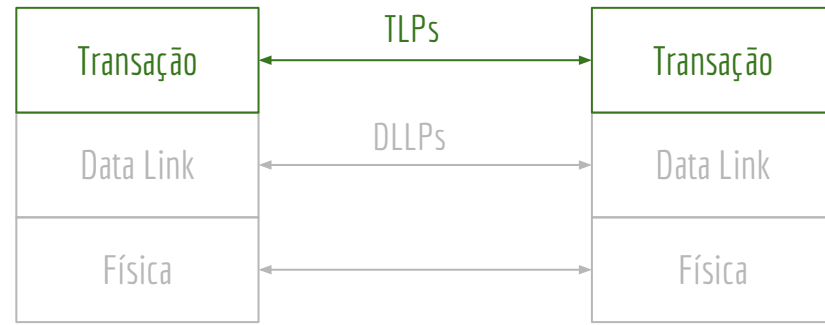
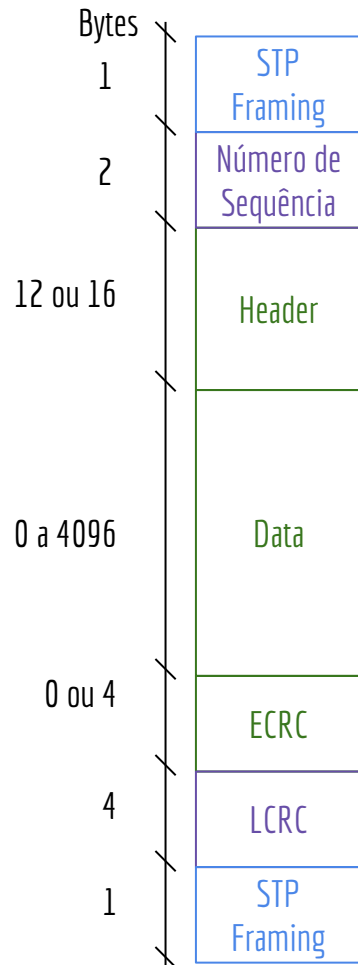
Todo pacote tem um id único.

Pacote de conclusão se refere a esse id, para saber qual requisição foi concluída.

Transações também podem ser do tipo post - envio sem a necessidade de resposta.

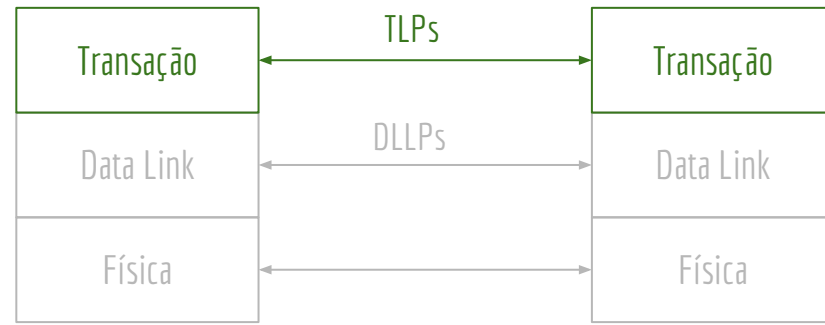
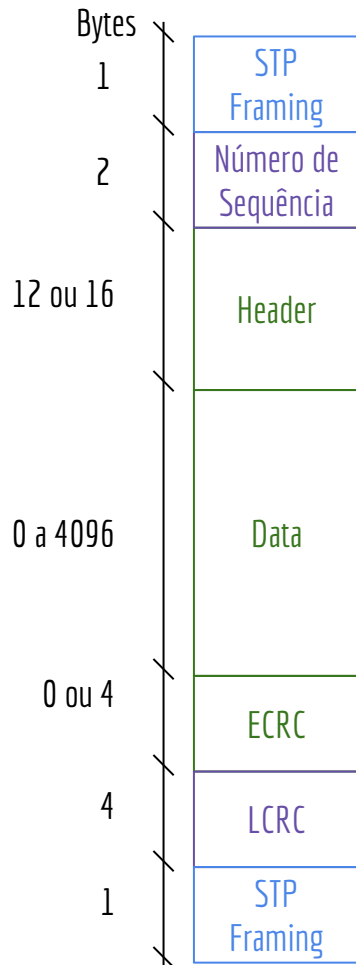


Pacote



Quando é solicitado um envio de informação (e.g., por um software) para dispositivos PCIe, a camada de transação cria um **TLP**.

Pacote

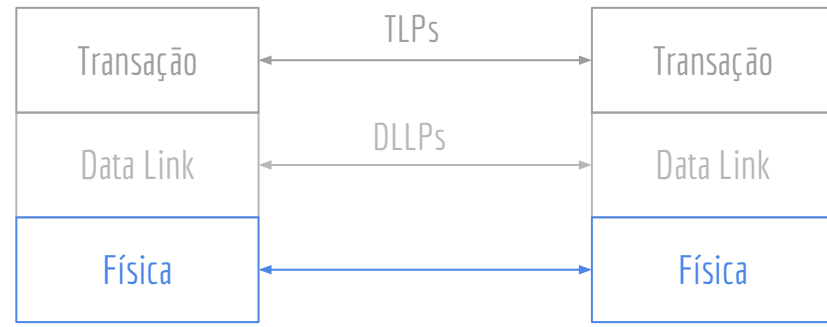
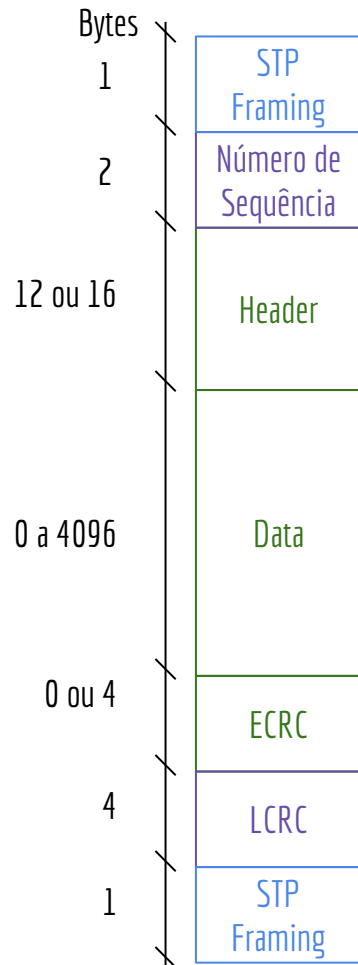


Header: Descreve o tipo de pacote. Contém informações necessárias pelo destinatário, como informações de roteamento.

Data: contém os dados (payload) que precisam ser transferidos. Até 4096 Bytes de dados por pacote. Existem TLPs sem campos de dados

ECRC: Opcional. End-to-end CRC. Usado pela camada de transação para verificar erros no **header** e **data**.

Pacote



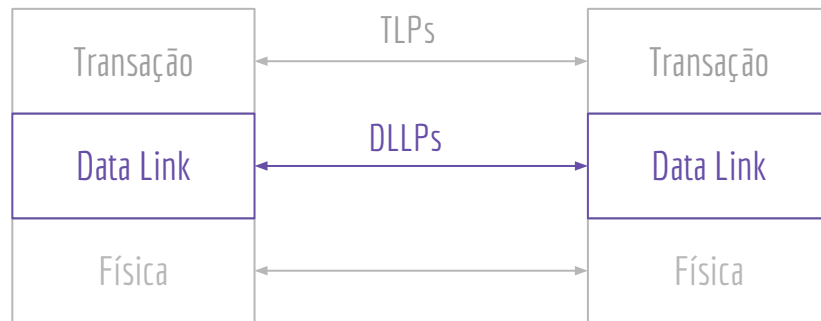
STP Framing: adicionado pela camada física. Adicionado para dar tempo para o receptor sincronizar.

Camada de Enlace de Dados

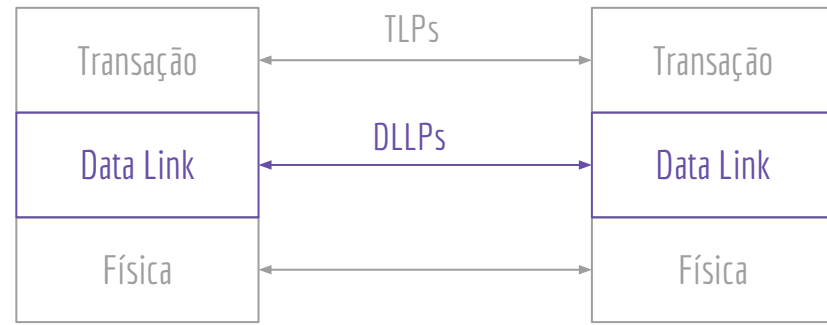
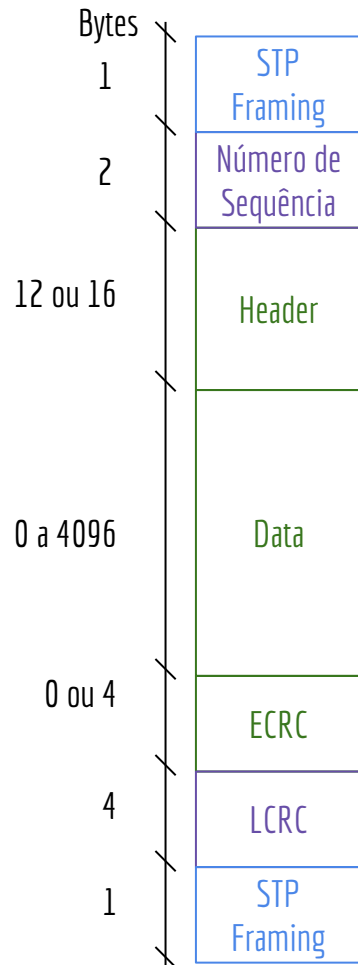
Data Link Layer - DLL.

Garantir a confiabilidade da transmissão.

Adiciona dois campos aos TLPs, criando um DLLP - Data Link Layer Packet.

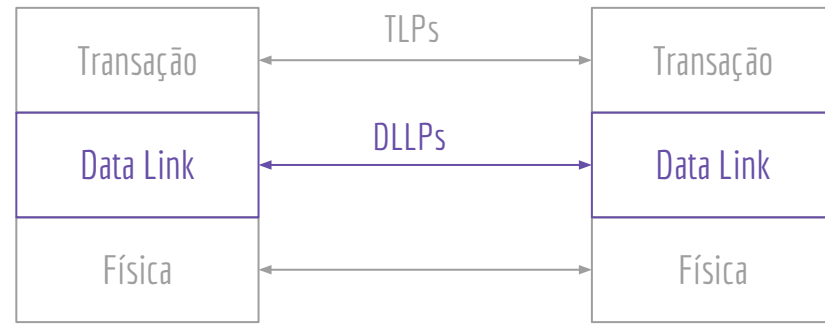
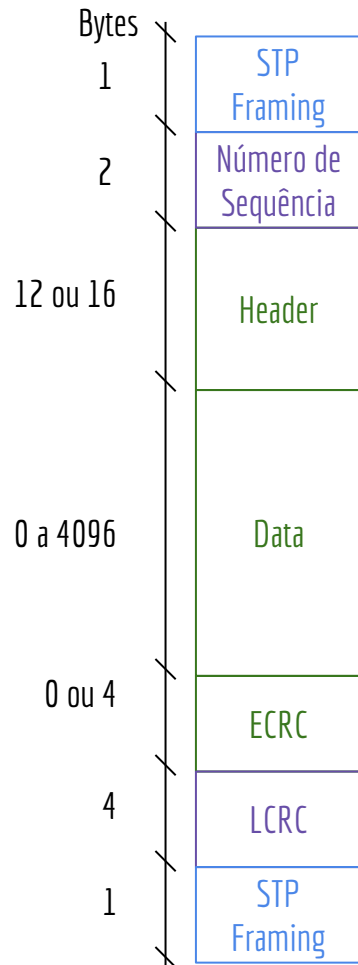


Pacote



Número de sequência de 16-bits: identificador do pacote.

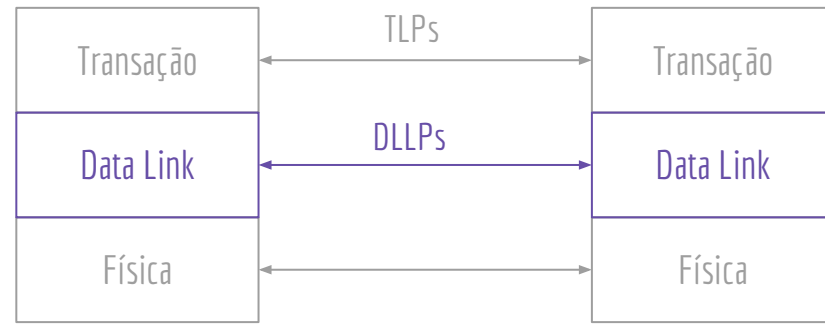
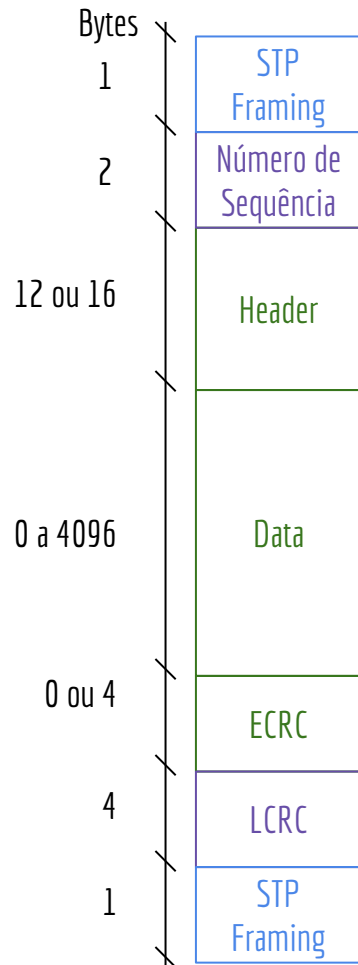
Pacote



LCRC: Código de detecção de erros da camada de Link.

O LCRC é obrigatório, e checado em toda vez que o pacote passa por algum nodo (o ECRC que é gerado na camada de transação é verificado somente quando chega ao destino).

Pacote



LCRC: Quando a verificação é feita:

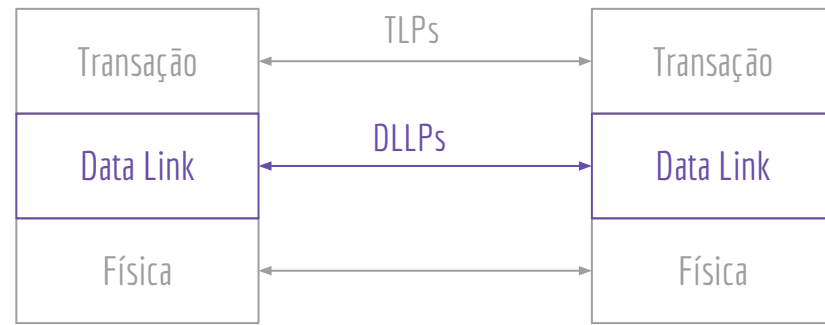
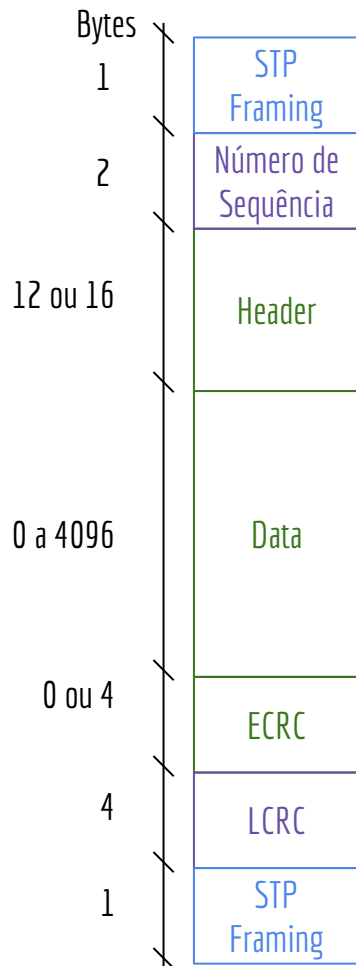
Se não existem erros.

A parte referente ao TLP é enviado para a camada de transação.

Se esse era o destino, a TL processa.

Senão, a TL determina a rota até o destino e repassa para a DLL novamente.

Pacote



LCRC: Quando a verificação é feita:

Se não existem erros.

A parte referente ao TLP é enviado para a camada de transação.

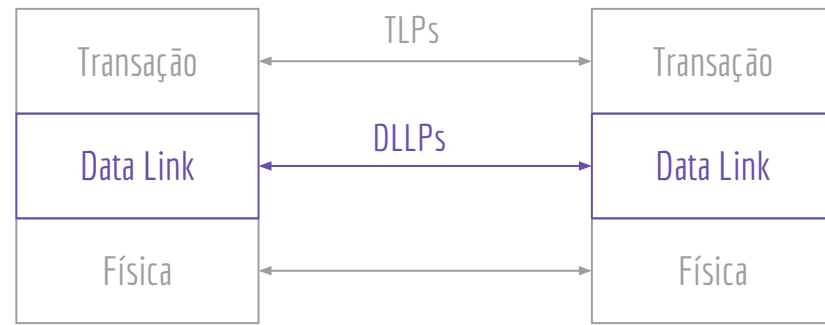
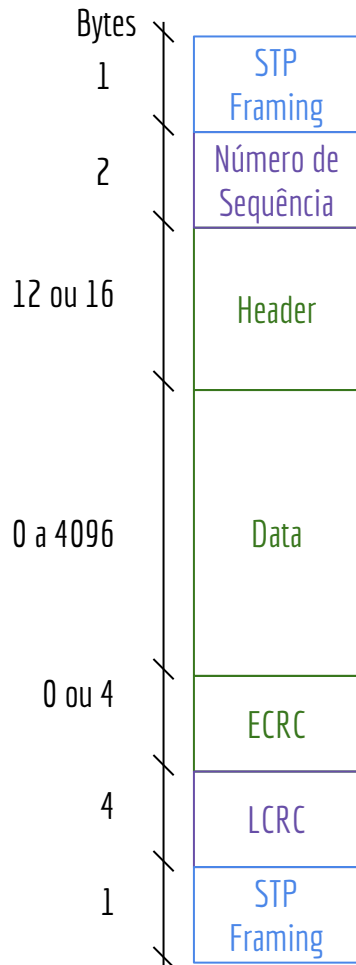
Se esse era o destino, a TL processa.

Senão, a TL determina a rota até o destino e repassa para a DLL novamente.

Se existem erros.

A DLL envia um pacote do tipo NAK (not acknowledged) para responsável pelo pacote, e o TLP é eliminado.

Pacote



LCRC: Quando a verificação é feita:

Se não existem erros.

A parte referente ao TLP é enviado para a camada de transação.

Se esse era o destino, a TL processa.

Senão, a TL determina a rota até o destino e repassa para a DLL novamente.

Se existem erros.

A DLL envia um pacote do tipo NAK (not acknowledged) para responsável pelo pacote, e o TLP é eliminado.

A camada DLL mantém uma cópia dos TLPs enviados.

Quando um NAK é recebido, a cópia é reenviada.

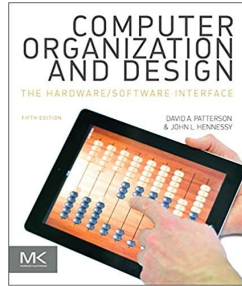
Quando um ACK é recebido (acknowledged), a cópia é removida do buffer.

Exercício

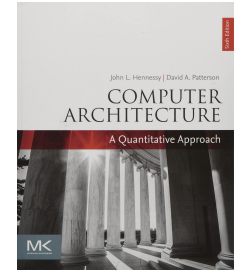
1. Considere que o dispositivo A precisa mandar uma mensagem de 8200 Bytes para o dispositivo B através de uma conexão PCIe. Quantos pacotes TLPs serão necessários? Quantos Bytes serão realmente transferidos no total?

Referências

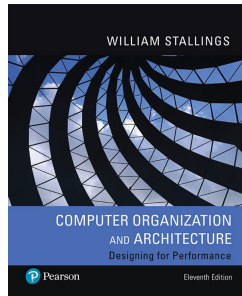
Patterson, Hennessy.
Arquitetura e Organização de
Computadores: A interface
hardware/software. 2014.



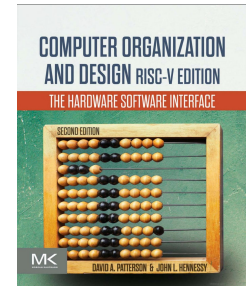
Hennessy, Patterson.
Arquitetura de Computadores:
uma abordagem quantitativa.
2019.



Stallings, W. Organização
de Arquitetura de
Computadores. 11a Ed.
2019.



Patterson, Hennessy.
Computer Organization and
Design RISC-V Edition: The
Hardware Software
Interface. 2020.



Licença

Esta obra está licenciada com uma Licença [Creative Commons Atribuição 4.0 Internacional](https://creativecommons.org/licenses/by/4.0/).

