MANCHESTER
1824
The University of Manchester

A (Very Hand-Wavy) Introduction to

PCI-Express

Jonathan Heathcote

Motivation

- **Six Week Project** Before PhD Starts:
  - SpiNNaker Ethernet I/O is **Slooooow**
  - How Do You Get Things In/Out of SpiNNaker, *Fast*?
  - Build a High-Speed Interface Via **Onboard FPGAs**

Motivation

- **Six Week Project** Before PhD Starts:
  - SpiNNaker Ethernet I/O is **Slooooow**
  - How Do You Get Things In/Out of SpiNNaker, *Fast*?
  - Build a High-Speed Interface Via **Onboard FPGAs**
- Obviously **Too Big** For Six Weeks
- Mainly Been Looking at **Guts of PCI-Express**

Outline

- PCI & PCI-Express **Background**
- The **Software** Perspective
- Bottom-up Through the **Protocol Layers**
  - Physical Layer
  - Data-Link Layer
  - Transaction Layer
- **SpiNNaker** & PCI-Express on **FPGA**

Peripheral Component Interconnect (PCI)

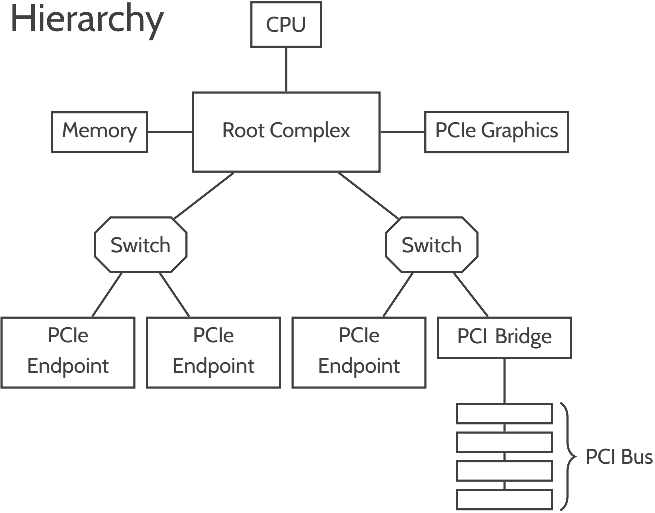
http://www.doubledecker-bus.com/models/

- Intel, 1993 (I was 2 years old)
- Widely Used, Even Today
- Extension of Processor Bus
- **Memory Mapped** Peripherals
- **Parallel** Signalling (32/64 bit)
- Multi-drop **Bus**
- Originally **1.1 Gb/s @** 33MHz
- Later **4.2 Gb/s @** 66MHz/64 bit

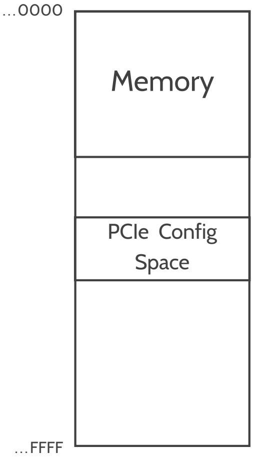
PCI EXPRESS® (PCIe)

- Intel, Dell, HP, IBM **2004** (I'm still only 13)
- Backward Compatible API
- Point-to-Point, Packet-Based
- One or More **High-Speed Serial 'Lanes'** (more later)
- Revision 1.0: **2.0 Gb/s/lane @** 2.5 GHz
  - **64 Gb/s**, 15×PCI Speed (32 lanes @ 2.5 GHz)
- Revision 3.0: **7.9 Gb/s/lane @** 8.0 GHz
  - State-of-the-art: **253.4 Gb/s** (32 lanes @ 8 GHz)

PCIe Hierarchy

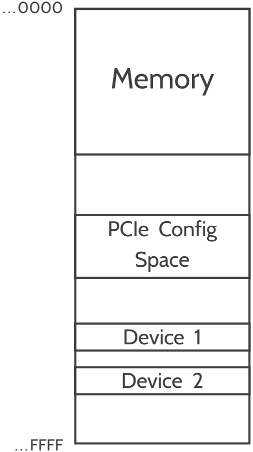


PCIe Software Interface



- PCIe Configuration Space
  - **Device Information**
  - ‘Base Address Registers (**BARs**) Control **Address Decoding**
  - BARs **Configured By OS**

PCIe Software Interface



...0000

Memory

PCIe Config Space

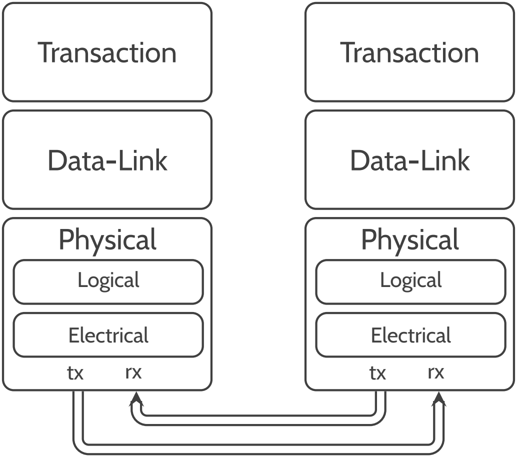
Device 1

Device 2

...FFFF

- PCIe Configuration Space
  - **Device Information**
  - ‘Base Address Registers (**BARs**) Control **Address Decoding**
  - BARs **Configured By OS**
- Device Spaces
  - **Device Specific** Contents
  - Driver Given Pointer

PCIe Protocol Layers



Transaction

Data-Link

Physical

Logical

Electrical

tx rx

Transaction

Data-Link

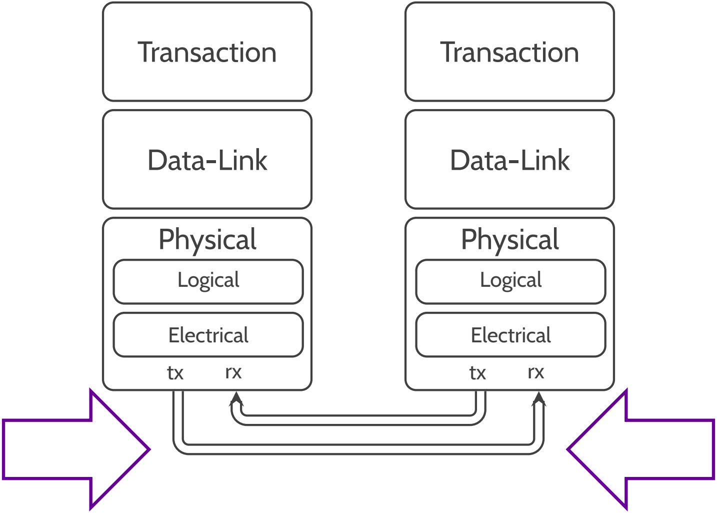
Physical

Logical

Electrical

tx rx

PCIe Protocol Layers



Transaction

Data-Link

Physical

Logical

Electrical

Transaction

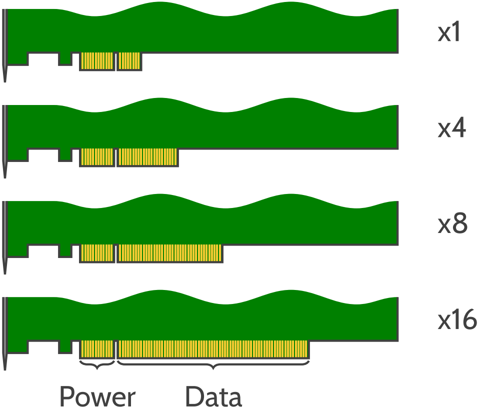
Data-Link

Physical

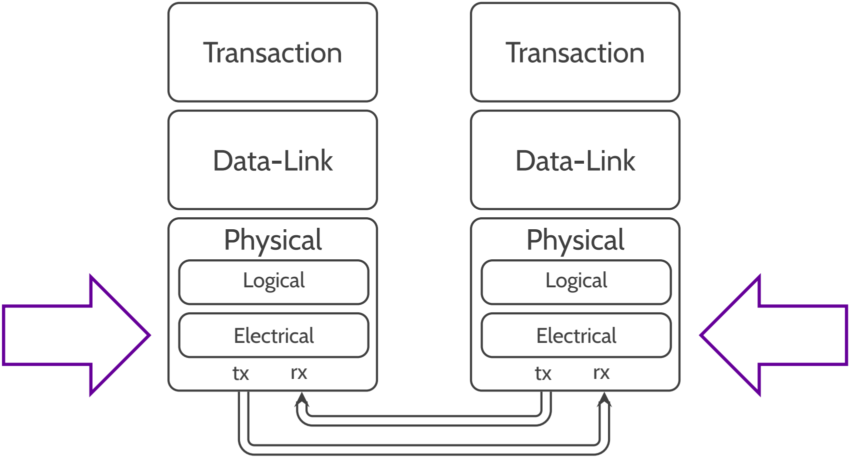
Logical

Electrical

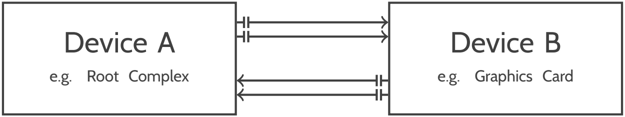
Mechanical Interface



PCIe Protocol Layers (Again)

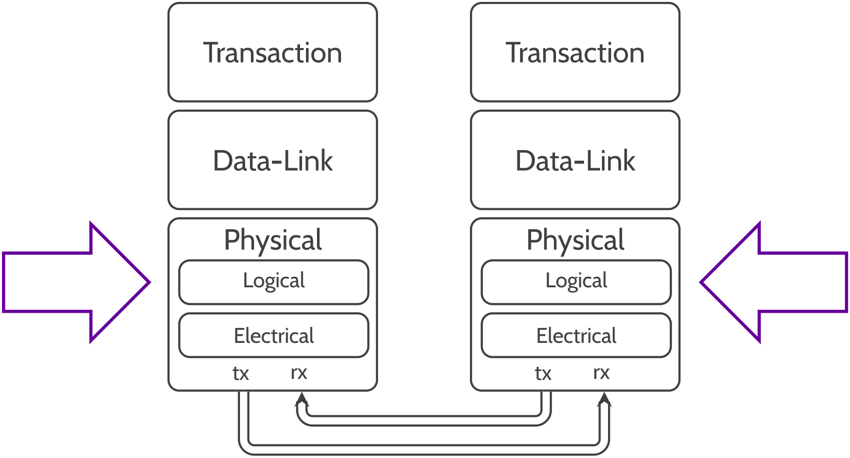


Electrical Interface



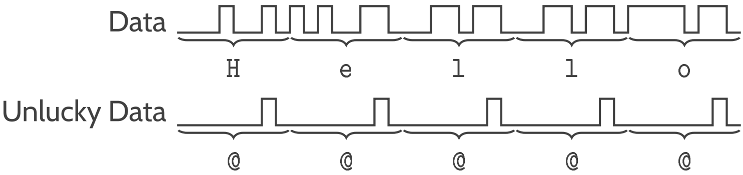
- **Transmit** & **Receive** Differential Pairs (Per Lane)
- **Peer-to-Peer** Arrangement
- **AC Coupled** via capacitor in transmitter

PCIe Protocol Layers (Yet Again)



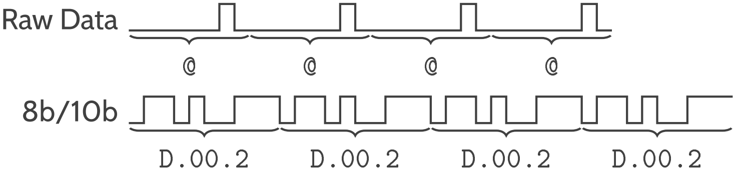
MANCHESTER 1824
The University of Manchester

Clock Recovery



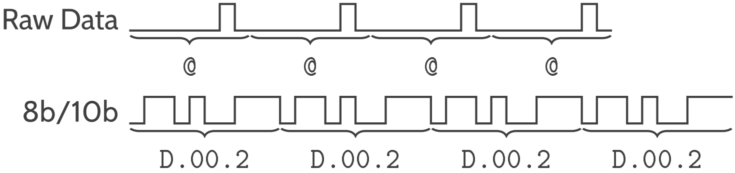
- No Clock Signal
- Must **Extract Clock** From Data
- Requires **Frequent Transitions**

8b/10b Coding (PCIe $\leq$ 2.1)



- Encodes **8-bit** bytes into **10-bit** symbols
- Transitions at least every 5 bits (for clock recovery)
- Bonus: Maintains **DC Balance** (due to AC coupling)
- Bonus: 8 Special **'K' Symbols** (out-of-band)

8b/10b Coding (PCIe $\leq$ 2.1)



- Encodes **8-bit** bytes into **10-bit** symbols
- Transitions at least every 5 bits (for clock recovery)
- Bonus: Maintains **DC Balance** (due to AC coupling)
- Bonus: 8 **Special ‘K’ Symbols** (out-of-band)

↳ Cereal Signalling...

A Quick Aside…

- 128b/130b encoding introduced in PCIe v3.0
- **Different mechanism** (scrambling, not lookup table)
- **Reduces overhead** from 20% to 1.54%
- Not covered in this presentation

Aligning 8b/10b Symbols

```
...011101000101010011101101110101100
010111001101001111000011010011001010
101001110110110101010110000110001110
100111000011010011001101110001011110
100110000110000101001110100011000100
110011001101110011010110101000100100
010101011011100111001010010011001101
110000110100111010001010100111011011
101011000101110011010011110000110100
110010101010011101101101010101100001
100011101001110000110100110011011...
```

→

```
...0xfd0xdc?0xfd@0xf6J?Ux
0xf20xf6P??d0xfd0xf9P0xbf
?0xf5?0xa2P0xf60xfd0xdc?0
xfd@0xf6J?Ux0xf20xf6P??d0
xfd0xf9P0xbf?0xf5?0xa2P0x
f60xfd0xdc?0xfd@0xf6J?Ux0
xf20xf6P??d0xfd0xf9P0xbf?
0xf5?0xa2P0xf60xfd0xdc?0x...
```

- Where does one code end and another begin?

Aligning 8b/10b Symbols

```
...011101000101010011101101110101100
010111001101001111000011010011001010
101001110110110101010110000110001110
100111000011010011001101110001011110
100110000110000101010100111100111010
001100010011001100110111001101011010
100010010001010101101110011100101001
001100110111000011010011101000101010
011101101110101100010111001101001111
000011010011001010101001110110110101
01011000011001010011110011101 0011...
```

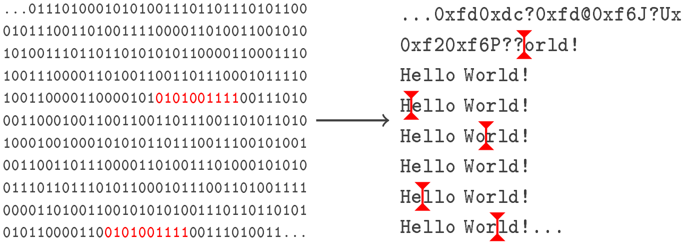
→

```
...0xfd0xdc?0xfd@0xf6J?Ux
0xf20xf6P??d>0xfd0xf9P0xb
f?0xf5?0xa2P0xf60xfd0xdc?
0xfd@0xf6J?Ux>0xf20xf6P??
d0xfd0xf9P0xbf?0xf5?0xa2P
0xf60xfd0xdc?0xfd>@0xf6J?
Ux0xf20xf6P??d0xfd0xf9P0x
bf?0xf5?0xa2>P0xf60xfd0xd...
```

- Where does one code end and another begin?
- **‘Comma’** ‘K’ Symbol inserted periodically

MANCHESTER
1824
The University of Manchester

Aligning 8b/10b Symbols



```
...011101000101010011101101110101100
010111001101001111000011010011001010
101001110110110101010110000110001110
100111000011010011001101110001011110
100110000110000101010100111100111010
001100010011001100110111001101011010
100010010001010101101110011100101001
001100110111000011010011101000101010
011101101110101100010111001101001111
000011010011001010101001110110110101
01011000011001010011110011101001 1...
```

```
...0xfd0xdc?0xfd@0xf6J?Ux
0xf20xf6P??orld!
Hello World!
Hello World!
Hello World!
Hello World!
Hello World!
Hello World!...
```

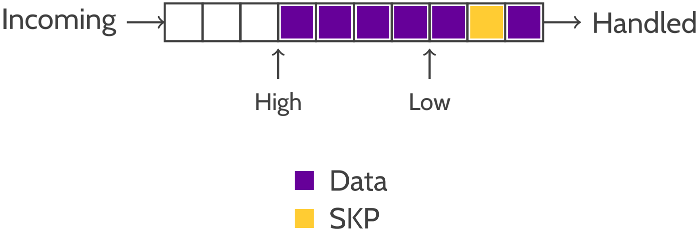
- Where does one code end and another begin?
- **‘Comma’** ‘K’ Symbol inserted periodically
- When a comma is found, align bits

Clock Rate Disparity Tolerance

- Reference Clock 600 PPM
- Off-by-one every 1666 bits
- At 8 GHz, **0.5MB extra sent/received per second**
- Solved by inserting extra **'SKP Ordered Sets'**
- (A Sequence of 'K' Symbols)

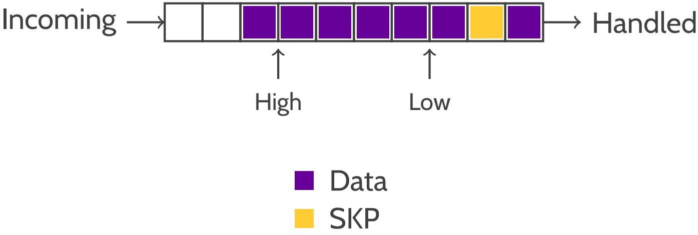
SKP Example: Sender Faster

- Fills faster than empties
- Eventually falls above ‘high’ mark
- SKP Sequence **skipped to empty buffer faster**



SKP Example: Sender Faster

- Fills faster than empties
- Eventually falls above ‘high’ mark
- SKP Sequence **skipped to empty buffer faster**



Incoming → → Handled

High

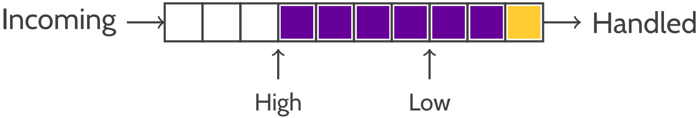
Low

🟪 Data

🟨 SKP

SKP Example: Sender Faster

- Fills faster than empties
- Eventually falls above ‘high’ mark
- SKP Sequence **skipped to empty buffer faster**

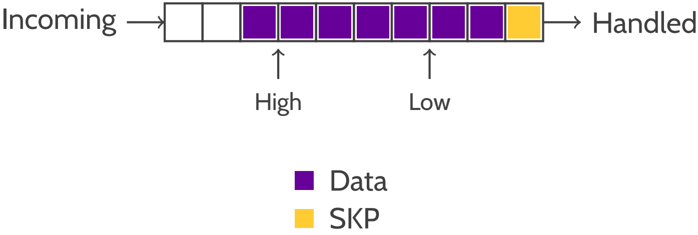


🟪 Data

🟨 SKP

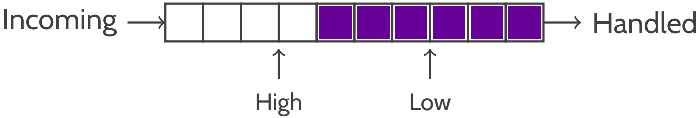
SKP Example: Sender Faster

- Fills faster than empties
- Eventually falls above ‘high’ mark
- SKP Sequence **skipped to empty buffer faster**



SKP Example: Sender Faster

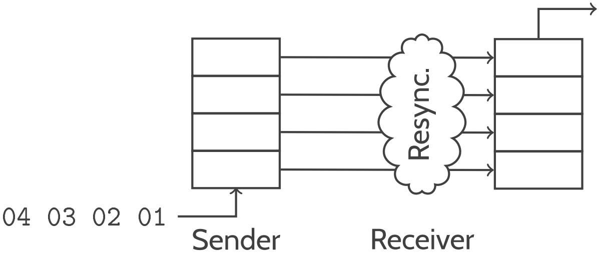
- Fills faster than empties
- Eventually falls above ‘high’ mark
- SKP Sequence **skipped to empty buffer faster**



🟪 Data

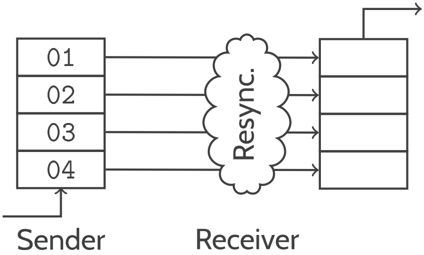
🟨 SKP

PCIe Lanes



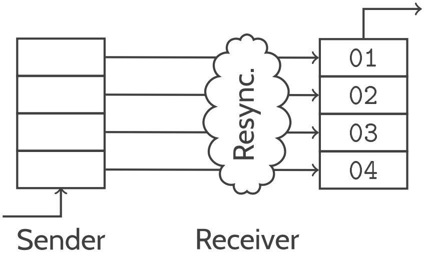
- Allows Some Speed Up by **Adding Hardware**
- *N* Serial Signals
- *N* **Consecutive Bytes** Sent in Parallel
- **Resynchronised** at Receiver (More Later)

PCIe Lanes



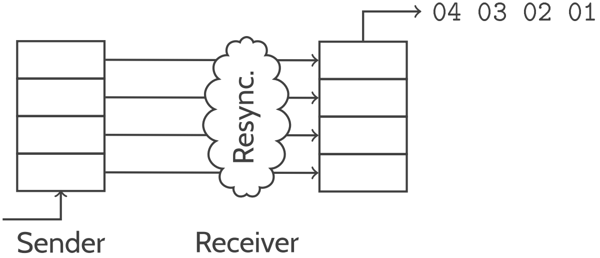
- Allows Some Speed Up by **Adding Hardware**
- *N* Serial Signals
- *N* **Consecutive Bytes** Sent in Parallel
- **Resynchronised** at Receiver (More Later)

PCIe Lanes



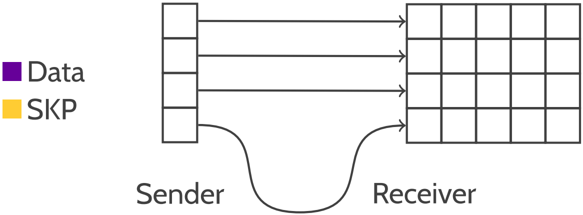
- Allows Some Speed Up by **Adding Hardware**
- *N* Serial Signals
- *N* **Consecutive Bytes** Sent in Parallel
- **Resynchronised** at Receiver (More Later)

PCIe Lanes



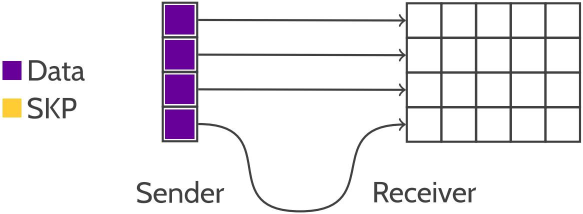
- Allows Some Speed Up by **Adding Hardware**
- *N* Serial Signals
- *N* **Consecutive Bytes** Sent in Parallel
- **Resynchronised** at Receiver (More Later)

Lane-to-Lane De-skew



- Lanes have **varying skew**
- **Buffer** each lane separately
- SKP sequences **sent simultaneously** on all lanes
- **De-skew** on such packets

Lane-to-Lane De-skew



🟪 Data

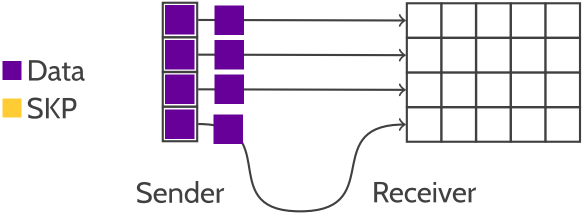
🟨 SKP

Sender

Receiver

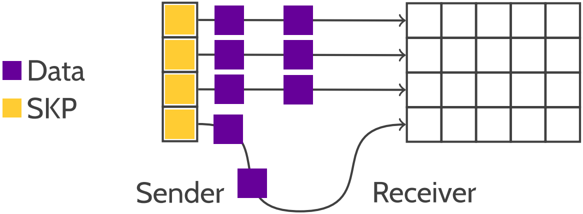
- Lanes have **varying skew**
- **Buffer** each lane separately
- SKP sequences **sent simultaneously** on all lanes
- **De-skew** on such packets

Lane-to-Lane De-skew



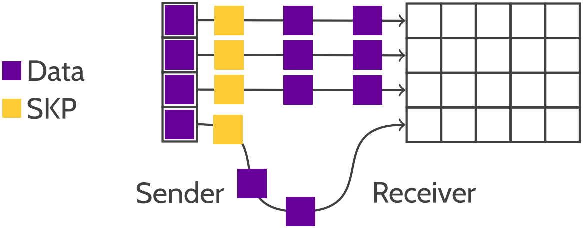
- Lanes have **varying skew**
- **Buffer** each lane separately
- SKP sequences **sent simultaneously** on all lanes
- **De-skew** on such packets

Lane-to-Lane De-skew



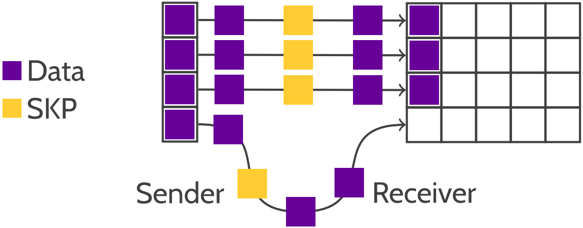
- Lanes have **varying skew**
- **Buffer** each lane separately
- SKP sequences **sent simultaneously** on all lanes
- **De-skew** on such packets

Lane-to-Lane De-skew



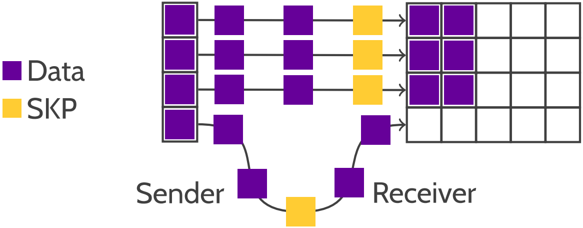
- Lanes have **varying skew**
- **Buffer** each lane separately
- SKP sequences **sent simultaneously** on all lanes
- **De-skew** on such packets

Lane-to-Lane De-skew



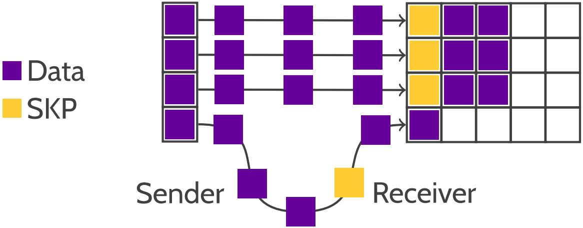
- Lanes have **varying skew**
- **Buffer** each lane separately
- SKP sequences **sent simultaneously** on all lanes
- **De-skew** on such packets

Lane-to-Lane De-skew



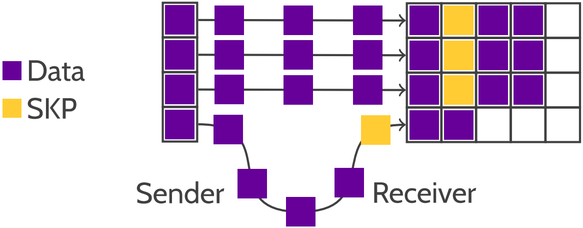
- Lanes have **varying skew**
- **Buffer** each lane separately
- SKP sequences **sent simultaneously** on all lanes
- **De-skew** on such packets

Lane-to-Lane De-skew



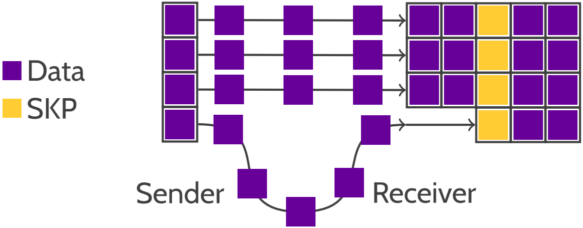
- Lanes have **varying skew**
- **Buffer** each lane separately
- SKP sequences **sent simultaneously** on all lanes
- **De-skew** on such packets

Lane-to-Lane De-skew



- Lanes have **varying skew**
- **Buffer** each lane separately
- SKP sequences **sent simultaneously** on all lanes
- **De-skew** on such packets

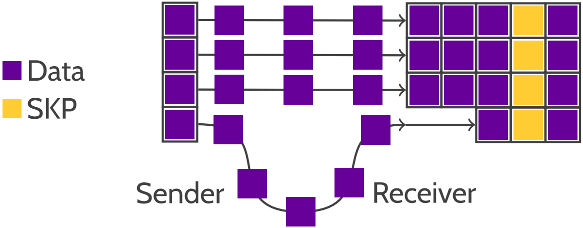
Lane-to-Lane De-skew



- Lanes have **varying skew**
- **Buffer** each lane separately
- SKP sequences **sent simultaneously** on all lanes
- **De-skew** on such packets

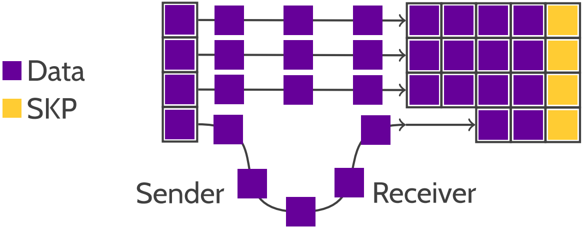
MANCHESTER
1824
The University of Manchester

Lane-to-Lane De-skew



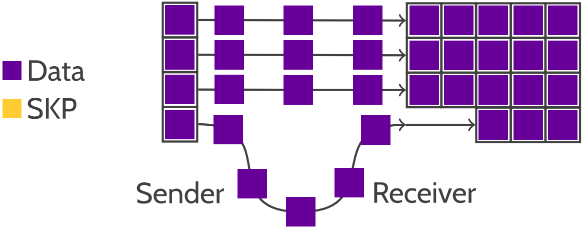
- Lanes have **varying skew**
- **Buffer** each lane separately
- SKP sequences **sent simultaneously** on all lanes
- **De-skew** on such packets

Lane-to-Lane De-skew



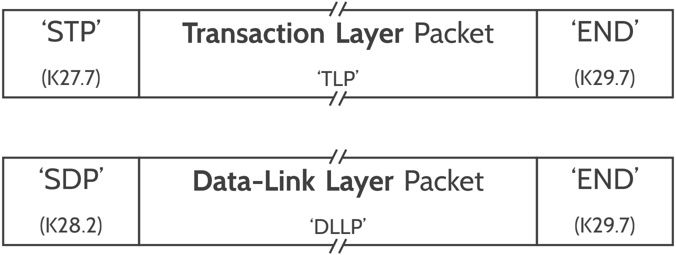
- Lanes have **varying skew**
- **Buffer** each lane separately
- SKP sequences **sent simultaneously** on all lanes
- **De-skew** on such packets

Lane-to-Lane De-skew



- Lanes have **varying skew**
- **Buffer** each lane separately
- SKP sequences **sent simultaneously** on all lanes
- **De-skew** on such packets

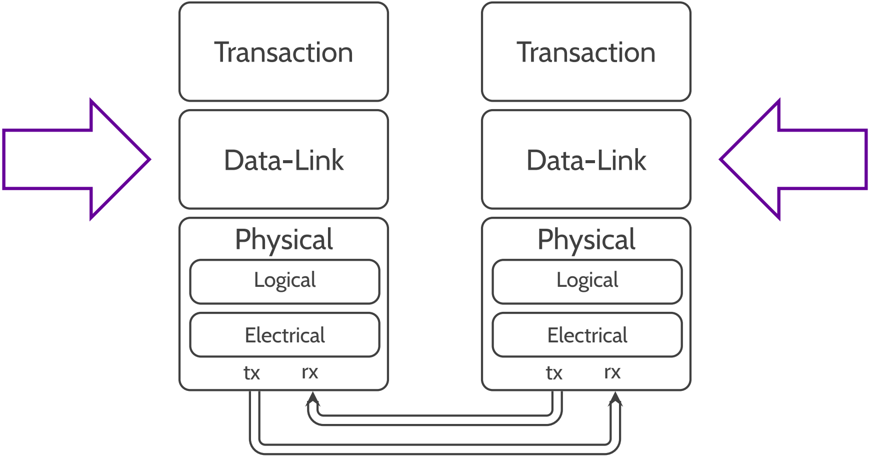
Packet Framing



'STP' (K27.7)	**Transaction Layer** Packet 'TLP'	'END' (K29.7)
'SDP' (K28.2)	**Data-Link Layer** Packet 'DLLP'	'END' (K29.7)

- 'K' Symbol **Marks Start/End** of Higher-Level Packets
- Distinguishes Target **Protocol Layer**

PCIe Protocol Layers (Yet Again, Again)

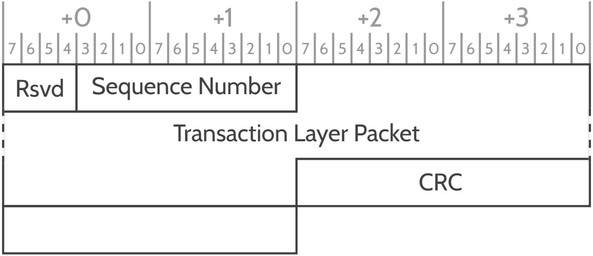


Data-Link Layer

- Reliable Delivery of **Transaction Layer Packets** (TLPs)
  - Uses: Memory Transactions, Interrupt Signalling
  - Error Detection
  - **Retransmission**
- Out-of-band **Data-Link Layer** Packets (DLLPs)
  - Uses: ACK/NACK, Power Management
  - Error Detection
  - **No Retransmission**
  - Uses Must **Tolerate Non-Delivery**

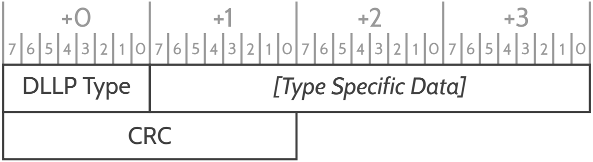
MANCHESTER 1824
The University of Manchester

Transaction Layer Packet (TLP) Framing



- **Variable** Length
- Seq. number allows many TLPs to be 'ACKed' at once
- CRC for error detection: send 'NACK'

Data-Link Layer Packets (DLLPs)



+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

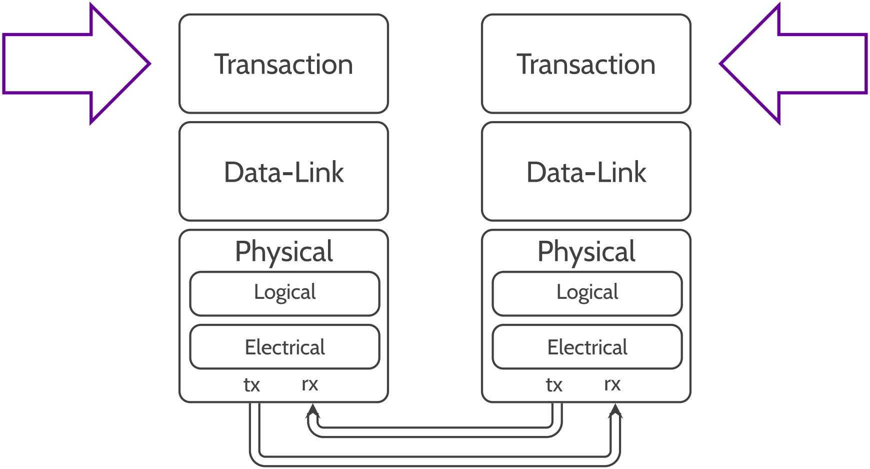
DLLP Type

[Type Specific Data]

CRC

- **Fixed** (Short) Length
- Smaller CRC (As Less Data)

PCIe Protocol Layers (Yet Again, Again, One Last Time)



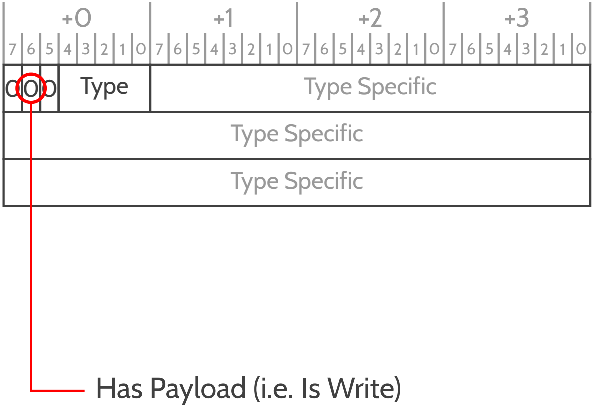
PCIe Transaction Layer

- Usually Exposed by FPGAs
- Transaction Types
  - **Memory** Read/Write
  - **I/O** Read/Write (Deprecated)
  - **Configuration** Read/Write
  - **Message** ('Interrupts')
- **Split Transactions** (Requests & Completions)
- No Completion for **Posted Requests** (e.g. writes)

Transaction Layer Packets

+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
0	0	0	Type					Type Specific																							
Type Specific																															
Type Specific																															

Transaction Layer Packets



+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

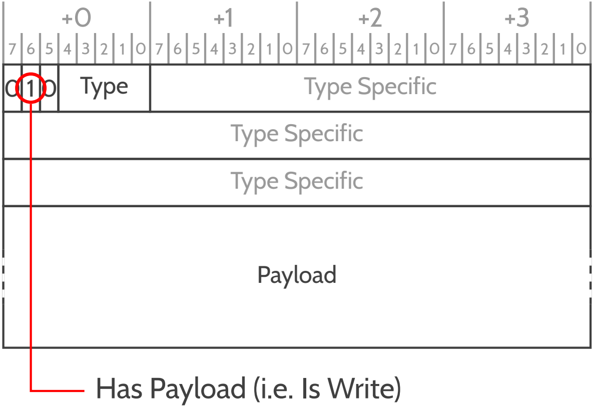
0 0 0 Type Type Specific

Type Specific

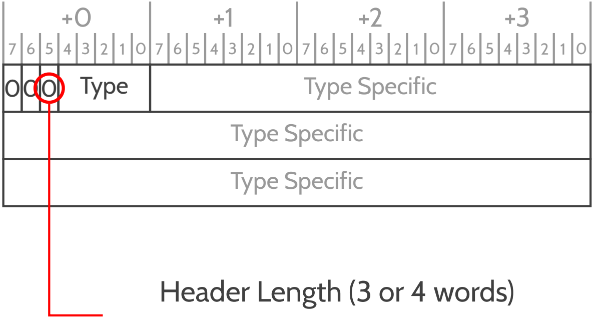
Type Specific

Has Payload (i.e. Is Write)

Transaction Layer Packets



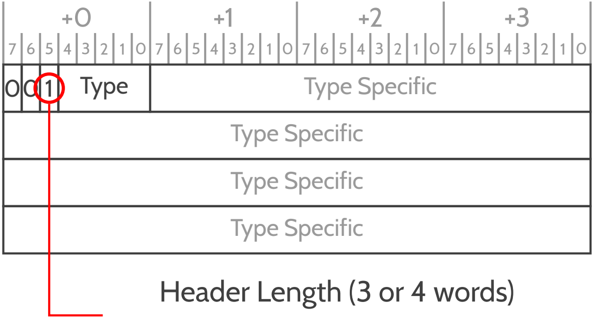
Transaction Layer Packets



Header Length (3 or 4 words)

Usually Specifies Either **32 or 64 bit Address**

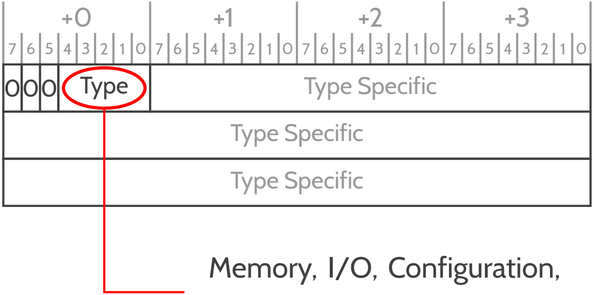
Transaction Layer Packets



Header Length (3 or 4 words)

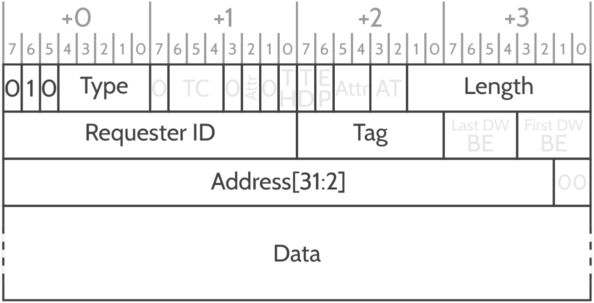
Usually Specifies Either **32 or 64 bit Address**

Transaction Layer Packets



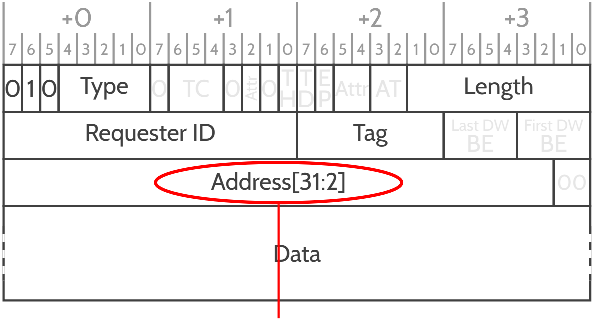
Memory, I/O, Configuration,
Message, Completion

Memory & I/O **Request** Packets



<table>
<tr><td colspan="8">+0</td><td colspan="8">+1</td><td colspan="8">+2</td><td colspan="8">+3</td></tr>
<tr><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td>0</td><td>1</td><td>0</td><td colspan="5">Type</td><td>0</td><td colspan="3">TC</td><td>0</td><td>Attr</td><td>0</td><td>TH</td><td>TD</td><td>EP</td><td colspan="2">Attr</td><td colspan="2">AT</td><td colspan="10">Length</td></tr>
<tr><td colspan="16">Requester ID</td><td colspan="8">Tag</td><td colspan="4">Last DW BE</td><td colspan="4">First DW BE</td></tr>
<tr><td colspan="30">Address[31:2]</td><td colspan="2">00</td></tr>
<tr><td colspan="32">Data</td></tr>
</table>

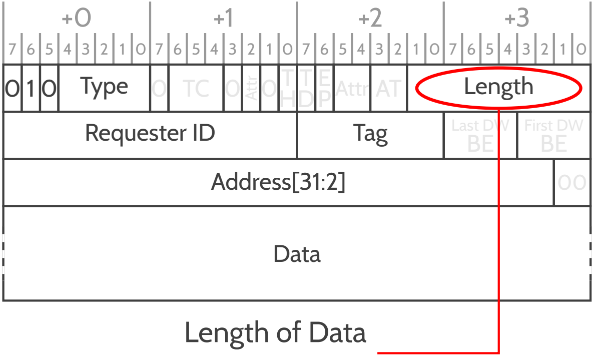
Memory & I/O **Request** Packets



Word-Aligned Address

Determines Target

Memory & I/O **Request** Packets



+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

0 1 0 Type 0 TC 0 Attr 0 TH TD EP Attr AT Length

Requester ID Tag Last DW BE First DW BE

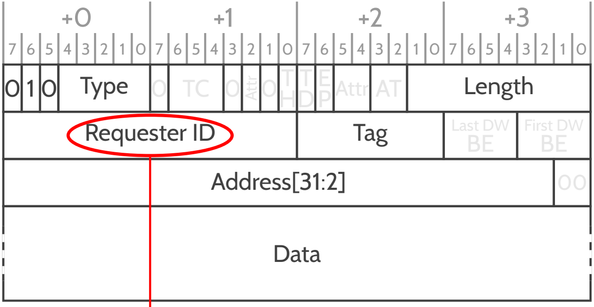
Address[31:2] 00

Data

Length of Data

1 – 1024 Words

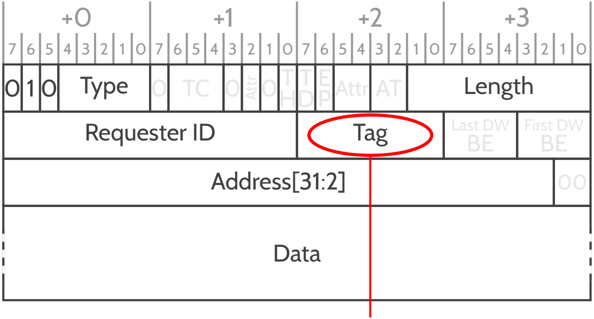
Memory & I/O **Request** Packets



Determines Completion Target

Derived From Physical Slot

Memory & I/O **Request** Packets



+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

0 1 0 Type 0 TC 0 Attr 0 TH TD EP Attr AT Length

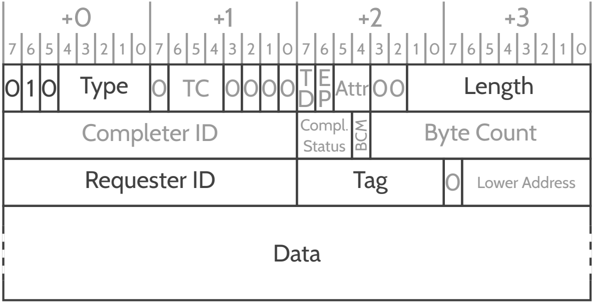
Requester ID Tag Last DW BE First DW BE

Address[31:2] 00

Data

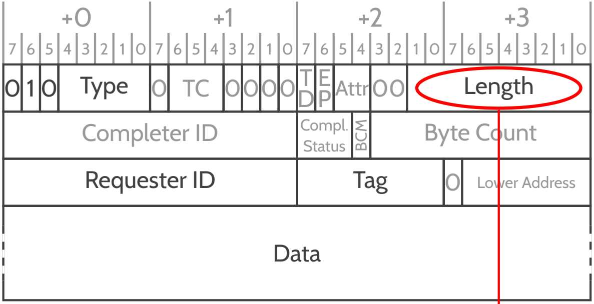
Identifies Outstanding Transactions

Memory & I/O **Completion** Packets



+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
0	1	0	Type					0	TC			0	0	0	0	TD	EP	Attr		0	0	Length									
Completer ID																Compl. Status			BCM	Byte Count											
Requester ID																Tag								0	Lower Address						
Data																															

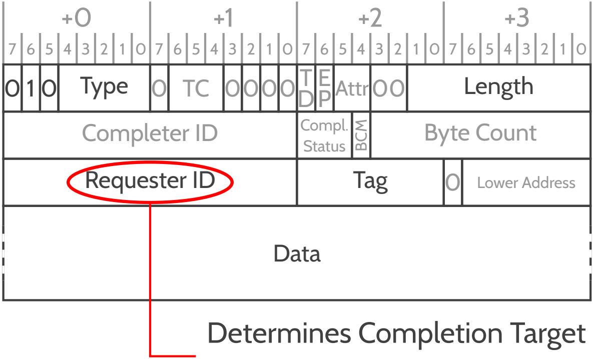
Memory & I/O **Completion** Packets



Length of Data

1 – 1024 Words

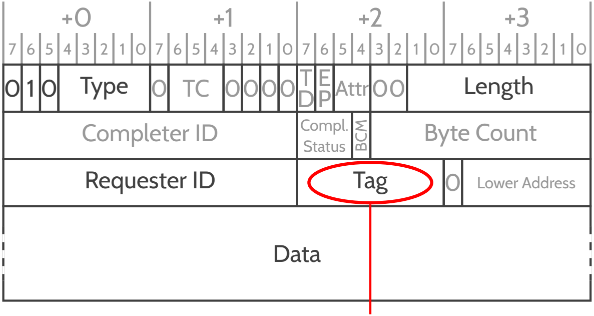
Memory & I/O **Completion** Packets



Determines Completion Target

Derived From Physical Slot

Memory & I/O **Completion** Packets



+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

0 1 0 Type 0 TC 0 0 0 0 TD EP Attr 0 0 Length

Completer ID Compl. Status BCM Byte Count

Requester ID Tag 0 Lower Address

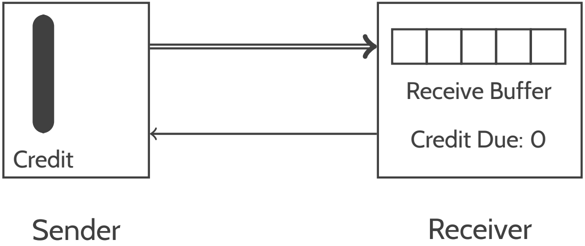
Data

Identifies Outstanding Transactions

Packet Routing Methods

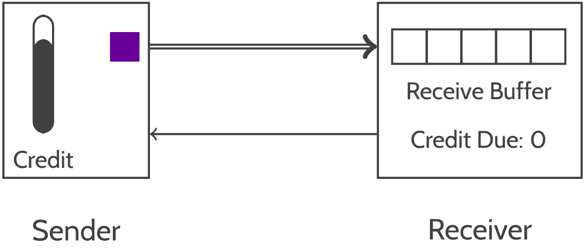
- **Address** Based
  - Based on **BARs**
  - Used **By Memory** & **I/O Requests**
- **ID** Based
  - Based on Physical Location
  - Used By **Configuration Requests**
  - Used By **All Completions**
- **Implicit**
  - e.g. Send to Root Complex
  - **Message Requests**

Credit-Based Flow Control



- **Sender** Has A **Limited Supply** Of ‘Credit’
- Sending Data **Uses Credit**
- **Receiver Returns Credit** When Buffer Space Is Freed

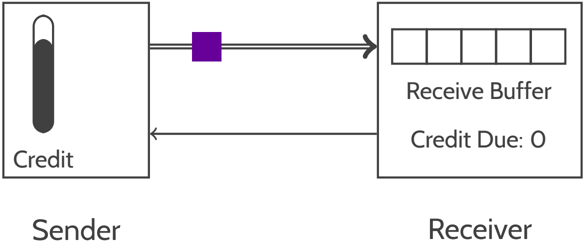
Credit-Based Flow Control



- **Sender** Has A **Limited Supply** Of 'Credit'
- Sending Data **Uses Credit**
- **Receiver Returns Credit** When Buffer Space Is Freed

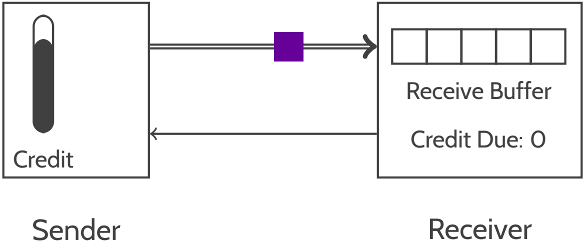
MANCHESTER
1824
The University of Manchester

Credit-Based Flow Control



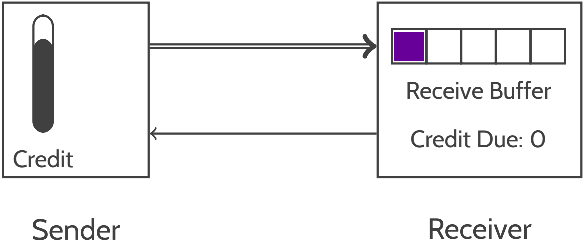
- **Sender** Has A **Limited Supply** Of 'Credit'
- Sending Data **Uses Credit**
- **Receiver Returns Credit** When Buffer Space Is Freed

Credit-Based Flow Control



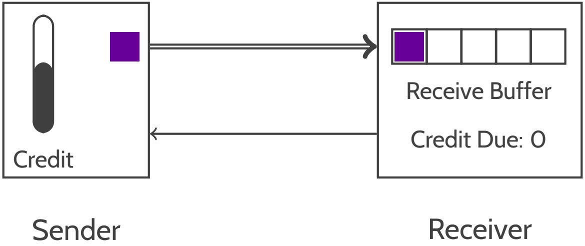
- **Sender** Has A **Limited Supply** Of ‘Credit’
- Sending Data **Uses Credit**
- **Receiver Returns Credit** When Buffer Space Is Freed

Credit-Based Flow Control



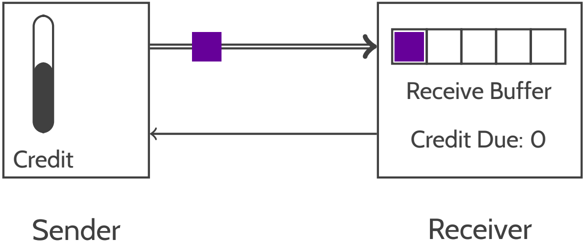
- **Sender** Has A **Limited Supply** Of 'Credit'
- Sending Data **Uses Credit**
- **Receiver Returns Credit** When Buffer Space Is Freed

Credit-Based Flow Control



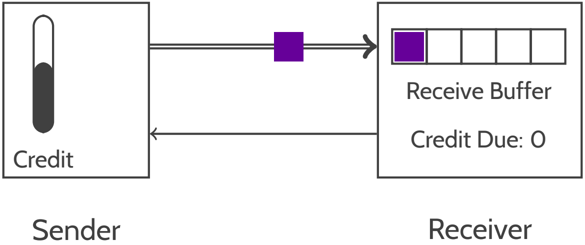
- **Sender** Has A **Limited Supply** Of ‘Credit’
- Sending Data **Uses Credit**
- **Receiver Returns Credit** When Buffer Space Is Freed

Credit-Based Flow Control



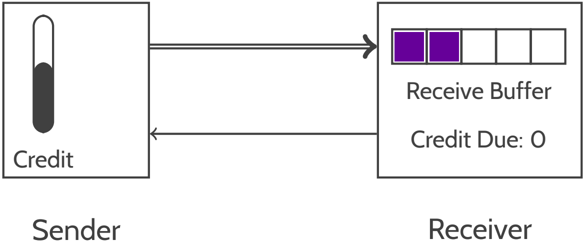
- **Sender** Has A **Limited Supply** Of ‘Credit’
- Sending Data **Uses Credit**
- **Receiver Returns Credit** When Buffer Space Is Freed

Credit-Based Flow Control



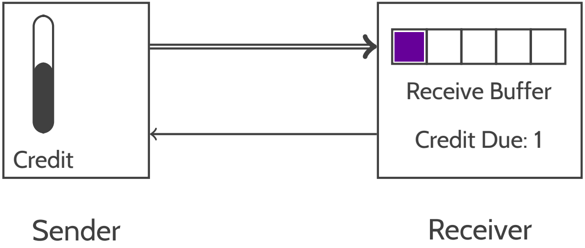
- **Sender** Has A **Limited Supply** Of ‘Credit’
- Sending Data **Uses Credit**
- **Receiver Returns Credit** When Buffer Space Is Freed

Credit-Based Flow Control



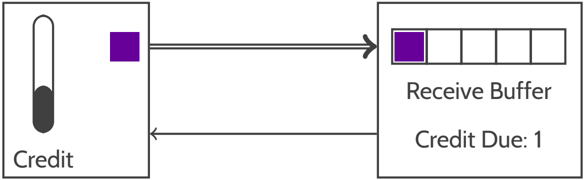
- **Sender** Has A **Limited Supply** Of ‘Credit’
- Sending Data **Uses Credit**
- **Receiver Returns Credit** When Buffer Space Is Freed

Credit-Based Flow Control



- **Sender** Has A **Limited Supply** Of 'Credit'
- Sending Data **Uses Credit**
- **Receiver Returns Credit** When Buffer Space Is Freed

Credit-Based Flow Control

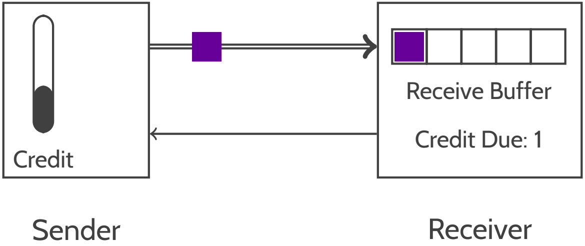


Sender Receiver

- **Sender** Has A **Limited Supply** Of ‘Credit’
- Sending Data **Uses Credit**
- **Receiver Returns Credit** When Buffer Space Is Freed

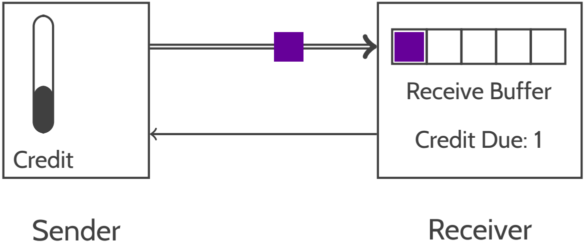
MANCHESTER 1824
The University of Manchester

Credit-Based Flow Control



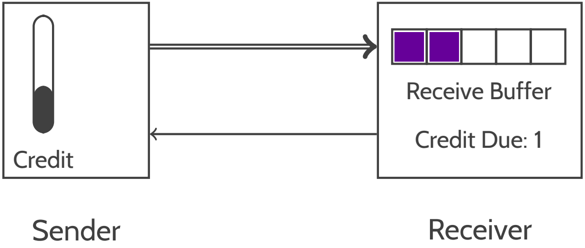
- **Sender** Has A **Limited Supply** Of ‘Credit’
- Sending Data **Uses Credit**
- **Receiver Returns Credit** When Buffer Space Is Freed

Credit-Based Flow Control



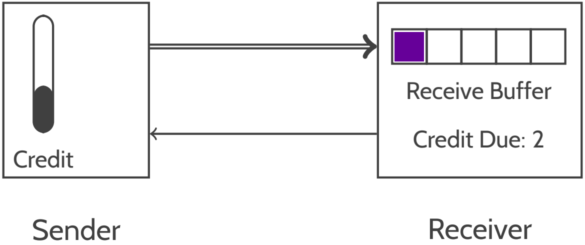
- **Sender** Has A **Limited Supply** Of ‘Credit’
- Sending Data **Uses Credit**
- **Receiver Returns Credit** When Buffer Space Is Freed

Credit-Based Flow Control



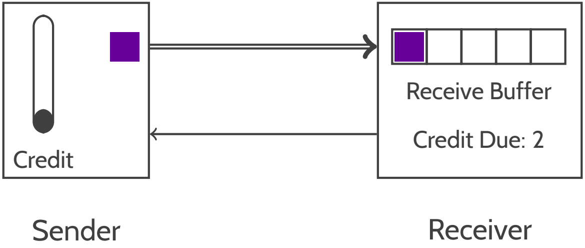
- **Sender** Has A **Limited Supply** Of ‘Credit’
- Sending Data **Uses Credit**
- **Receiver Returns Credit** When Buffer Space Is Freed

Credit-Based Flow Control



- **Sender** Has A **Limited Supply** Of 'Credit'
- Sending Data **Uses Credit**
- **Receiver Returns Credit** When Buffer Space Is Freed

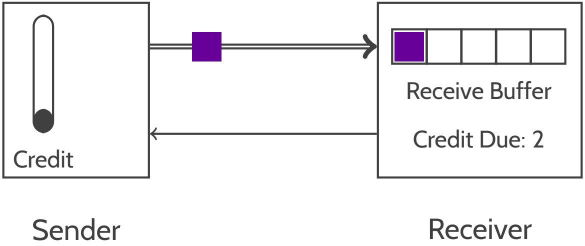
Credit-Based Flow Control



- **Sender** Has A **Limited Supply** Of ‘Credit’
- Sending Data **Uses Credit**
- **Receiver Returns Credit** When Buffer Space Is Freed

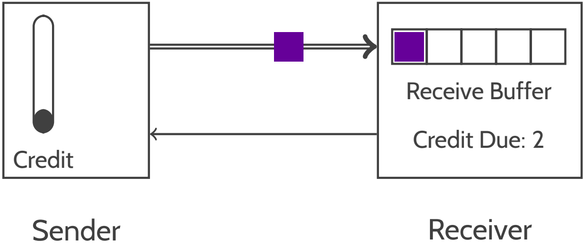
MANCHESTER
1824
The University of Manchester

Credit-Based Flow Control



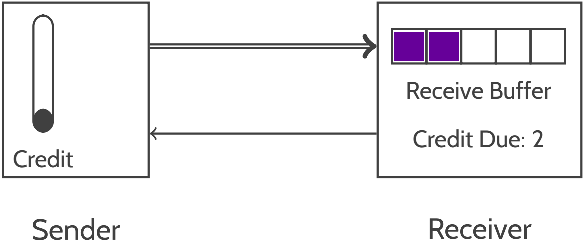
- **Sender** Has A **Limited Supply** Of ‘Credit’
- Sending Data **Uses Credit**
- **Receiver Returns Credit** When Buffer Space Is Freed

Credit-Based Flow Control



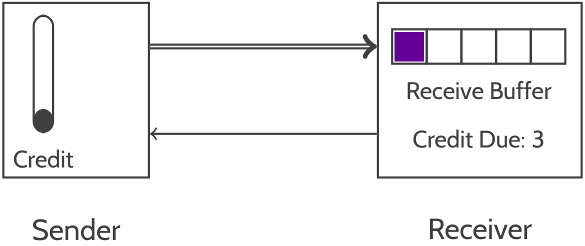
- **Sender** Has A **Limited Supply** Of ‘Credit’
- Sending Data **Uses Credit**
- **Receiver Returns Credit** When Buffer Space Is Freed

Credit-Based Flow Control



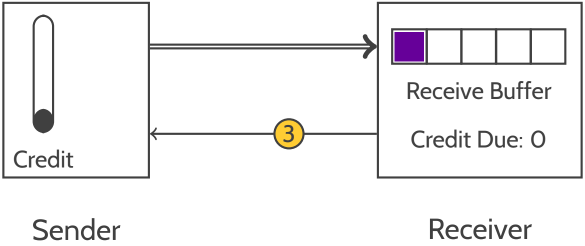
- **Sender** Has A **Limited Supply** Of ‘Credit’
- Sending Data **Uses Credit**
- **Receiver Returns Credit** When Buffer Space Is Freed

Credit-Based Flow Control



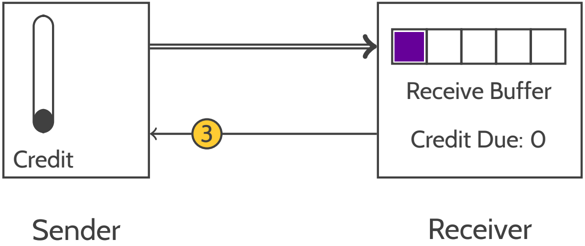
- **Sender** Has A **Limited Supply** Of ‘Credit’
- Sending Data **Uses Credit**
- **Receiver Returns Credit** When Buffer Space Is Freed

Credit-Based Flow Control



- **Sender** Has A **Limited Supply** Of ‘Credit’
- Sending Data **Uses Credit**
- **Receiver Returns Credit** When Buffer Space Is Freed

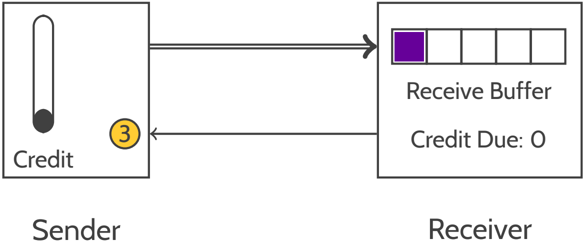
Credit-Based Flow Control



- **Sender** Has A **Limited Supply** Of ‘Credit’
- Sending Data **Uses Credit**
- **Receiver Returns Credit** When Buffer Space Is Freed

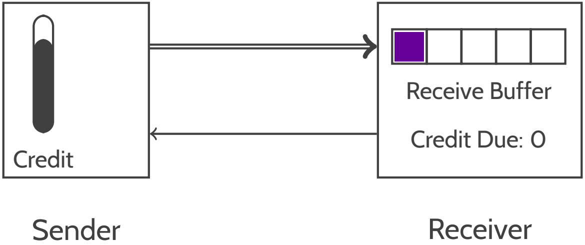
MANCHESTER 1824
The University of Manchester

Credit-Based Flow Control



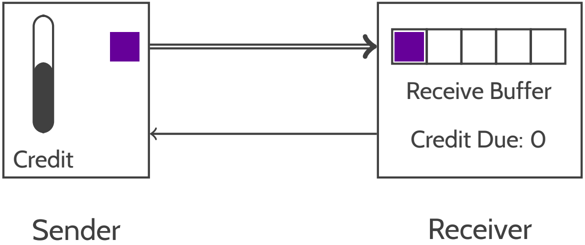
- **Sender** Has A **Limited Supply** Of ‘Credit’
- Sending Data **Uses Credit**
- **Receiver Returns Credit** When Buffer Space Is Freed

Credit-Based Flow Control



- **Sender** Has A **Limited Supply** Of ‘Credit’
- Sending Data **Uses Credit**
- **Receiver Returns Credit** When Buffer Space Is Freed

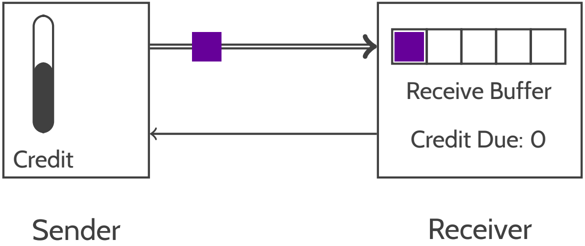
Credit-Based Flow Control



- **Sender** Has A **Limited Supply** Of ‘Credit’
- Sending Data **Uses Credit**
- **Receiver Returns Credit** When Buffer Space Is Freed

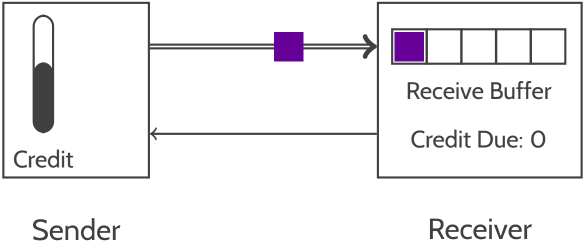
MANCHESTER
1824
The University of Manchester

Credit-Based Flow Control



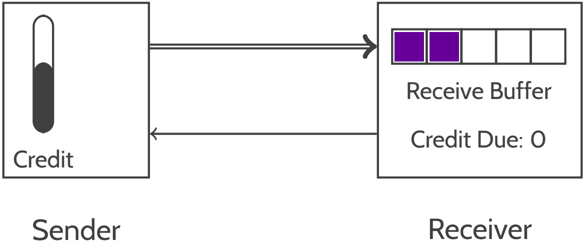
- **Sender** Has A **Limited Supply** Of ‘Credit’
- Sending Data **Uses Credit**
- **Receiver Returns Credit** When Buffer Space Is Freed

Credit-Based Flow Control



- **Sender** Has A **Limited Supply** Of ‘Credit’
- Sending Data **Uses Credit**
- **Receiver Returns Credit** When Buffer Space Is Freed

Credit-Based Flow Control

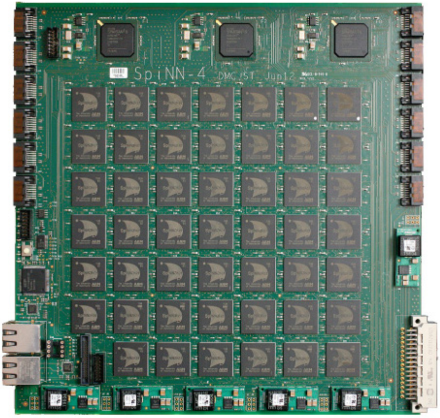


- **Sender** Has A **Limited Supply** Of ‘Credit’
- Sending Data **Uses Credit**
- **Receiver Returns Credit** When Buffer Space Is Freed

Protocol Layers Recap

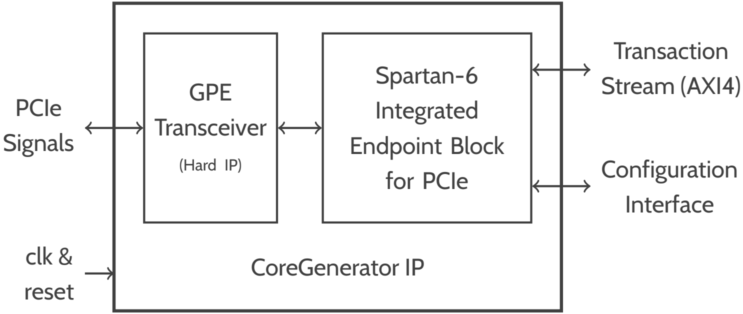
- Physical Layer
  - Send/Receive **Raw Data**
- Data-Link Layer
  - Send/Receive Some Data **Reliably**
- Transaction Layer
  - Send/Receive **High-Level** Requests

PCI-Express on SpiNNaker



- I/O Via Ethernet is **Slow**
- Three **FPGAs** per Board
- Some **Spare S-ATA** Cable Connectors
- Use S-ATA Cable to PCIe Card Adapter
- Implement **PCIe on FPGA**

Xilinx Spartan-6 PCIe IP Core



- PCIe v1.1 x1 (**2 Gb/s**)
- Comes 'Free' With FPGA

Summary

- SpiNNaker Needs a **Fast Host Interface**
- PCI-Express **Widely Used** In PCs
- Bottom-up Through the **Protocol Layers**
  - Physical Layer
  - Data-Link Layer
  - Transaction Layer
- SpiNNaker & FPGAs

Any Questions?