



Implementation of Protocol Multiplexing in PCI Express

Yogesh Chaudhary
Lead Member Technical Staff
Mentor Graphics



Disclaimer

Presentation Disclaimer: All opinions, judgments, recommendations, etc. that are presented herein are the opinions of the presenter of the material and do not necessarily reflect the opinions of the PCI-SIG®.

Agenda

- Introduction
 - ✓ Protocol Multiplexing
- PMUX Packet
 - ✓ PMUX Packet header
 - ✓ PMUX Packet flow
- Protocol Multiplexing implementation
 - ✓ Transaction Layer
 - ✓ Data Link Layer
 - ✓ Physical Layer
- Configuration Space
- Conclusion

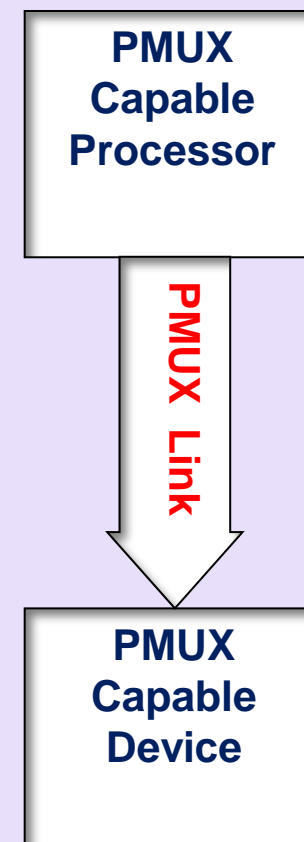
Introduction

■ Protocol Multiplexing (PMUX)

✓ Standard mechanism

- To transport non-PCI Express protocol across a PCI Express Link
- Supports the multiplexing of PMUX packets and TLPs on to single a PCI Express Link

Note : PMUX support is optional.



Introduction (cont.)

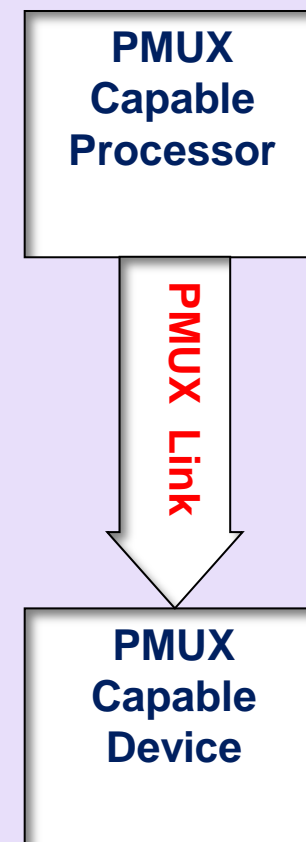
■ System topology

✓ PCI Express Link.

- Protocol Multiplexing is disabled.
- It works as normal PCI Express link with TLP flow.

✓ PMUX Link.

- Protocol Multiplexing is enabled.
- Both TLPs and PMUX packets are used in a coordinated fashion.
- PMUX packets may be used to support additional protocols efficiently.



Protocol Multiplexing Attributes

- Important attributes of Protocol Multiplexing:
 - ✓ PMUX support is optional normative.
 - ✓ PMUX has no impact on PCI Express components that do not support it.
 - ✓ PMUX has no impact on PCI Express TLPs and DLLPs, even when it is enabled.
 - ✓ A Link may be used for both TLPs and PMUX Packets at the same time.
 - ✓ PMUX is selectable on a per-Link basis. As it can be used on any collection of Links in a system.

Attributes (cont.)

- ✓ PMUX Packets also contain an LCRC. This is used to provide data resiliency in a similar fashion as PCI Express TLPs.
- ✓ A PMUX Link may support up to 4 simultaneously active PMUX Channels. Software configures the protocol used on each PMUX Channel.
- ✓ PMUX Packet received at ports that supports Protocol Multiplexing are ignored until Protocol Multiplexing is enabled by software.
- ✓ **Protocol Multiplexing is disabled by default and should be enabled by software.**

PMUX Packet

- PMUX Packet Header
 - ✓ PMUX Packet contains the following information

PMUX Channel ID [1:0]

PMUX Packet Metadata [11:0]

PMUX Packet Data DWORD 0 [31:0]

PMUX Packet Data DWORD 1 [31:0]



PMUX Packet Data DWORD n [31:0]

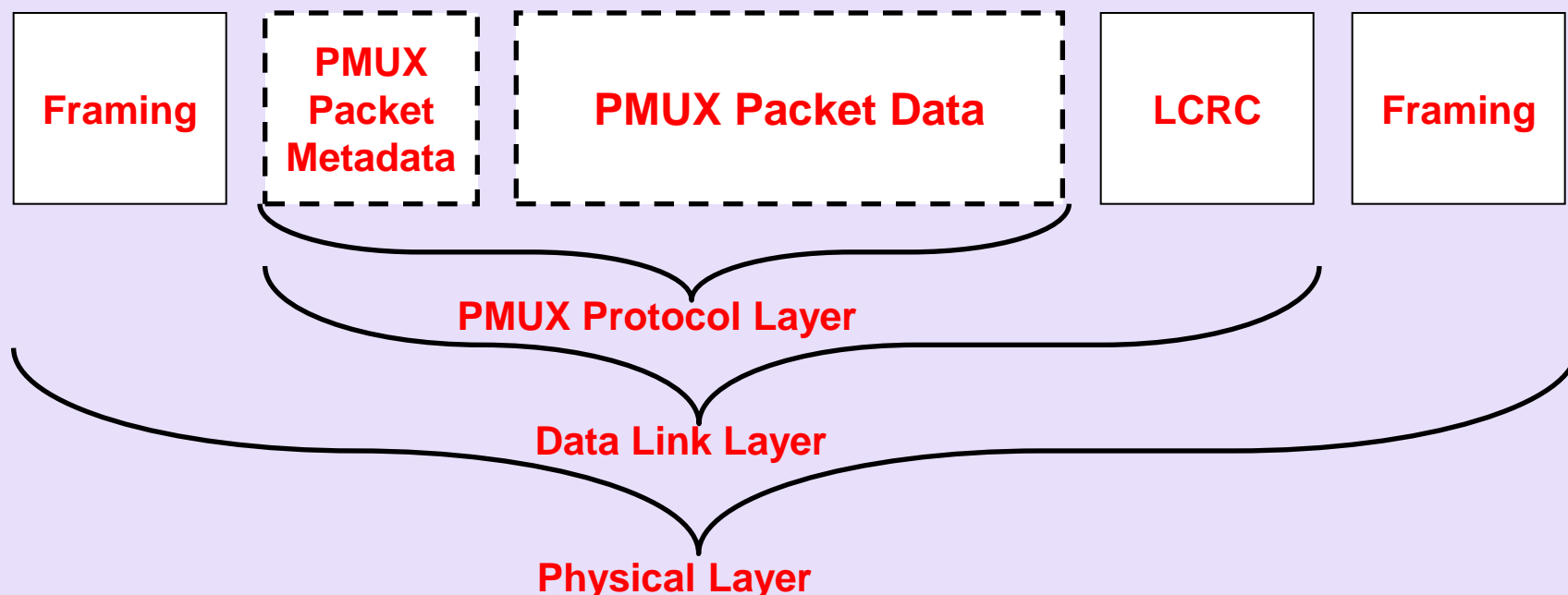
PMUX Packet (cont.)

- ✓ PMUX Channel ID
 - It's a 2-bit field that identifies which protocol is associated with a PMUX Packet.
- ✓ PMUX Packet Metadata
 - It's a 12-bit field that provides information about the PMUX Packet.
- ✓ PMUX Packet Data
 - It consists of between 0 to 125 DWORDs.
 - A PMUX Packet need not have any PMUX Packet Data may consist only PMUX Channel ID and PMUX Packet Metadata.

Note : Definition of PMUX Packet Metadata and PMUX Packet Data fields are protocol specific and is outside the scope of the specification.

PMUX Packet Flow

- PMUX Packet flow changes from PCI Express TLP flow are:



PMUX Packet at TL

- PMUX Packets use a protocol specific PMUX Protocol Layer instead of the PCI Express Transaction Layer.
 - ✓ PMUX does not have any effect on PCI Express Transaction Layer or vice-versa
 - No effect of TLP Prefix on PMUX Packet
 - Locked transaction have no effect on any protocol specific forwarding of PMUX Packet
 - VCs doesn't get effected by PMUX Packets
 - Nullified PMUX Packets has no effect on PCI Express and nullified TLPs doesn't affect Protocol Multiplexing

PMUX Packet at DLL

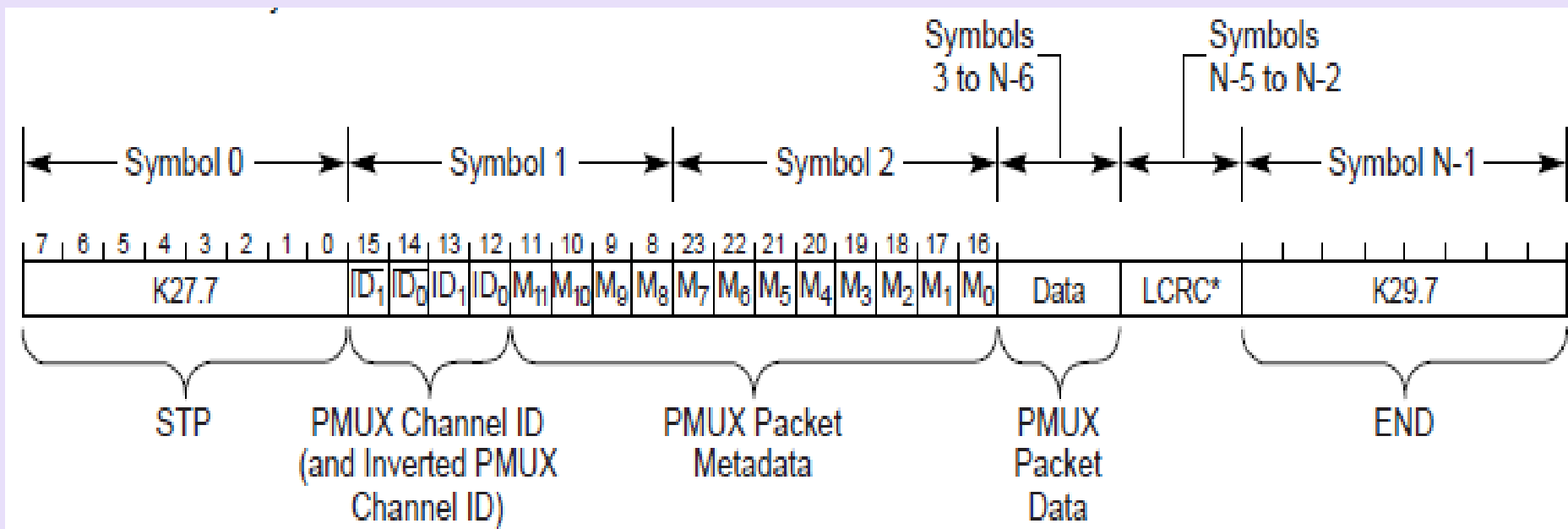
- PMUX Packet use a simplified Data Link Layer.
 - ✓ With DL_Down all PMUX Channels on a Link are disabled
 - ✓ No effect on Protocol Multiplexing for any DLLP lost
 - ✓ Packet integrity portion is unchanged
 - ✓ LCRC computation uses a different seed value
 - ✓ PMUX Packet LCRC error has no effect on PCI Express
 - ✓ TLP Sequence Number
 - PMUX Packet do not contain a TLP sequence number
 - 12-bit field is repurposed as PMUX Packet Metadata and is available for multiplexed protocol specific use.

PMUX Packet at PL

- The Physical Layer is slightly modified to provide a mechanism to identify PMUX Packets.
 - ✓ All PMUX Channels on the link are disabled by
 - FLR and PERST#
 - ✓ All PMUX Channels on the link are disabled upon entering LTSSM states
 - Disabled, Loopback, Hot Reset and Detect
 - ✓ Recovery LTSSM state doesn't have any effect on Protocol Multiplexing.
 - PMUX Packets are notified as TLPs to hold its transmission till link transition back to L0 from Recovery.

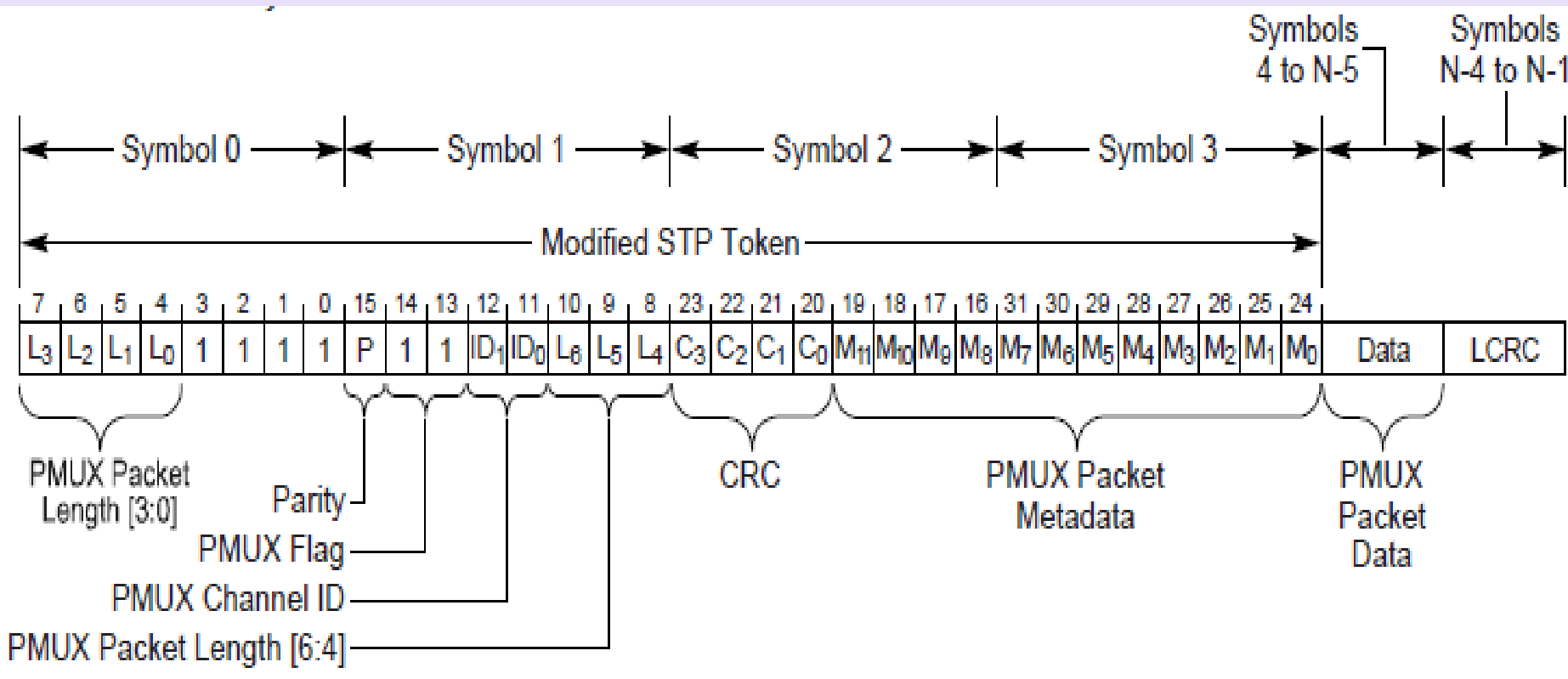
PMUX Packet Layout

- For 8b/10b Encoding



PMUX Packet Layout (cont.)

- For 128b/130b Encoding

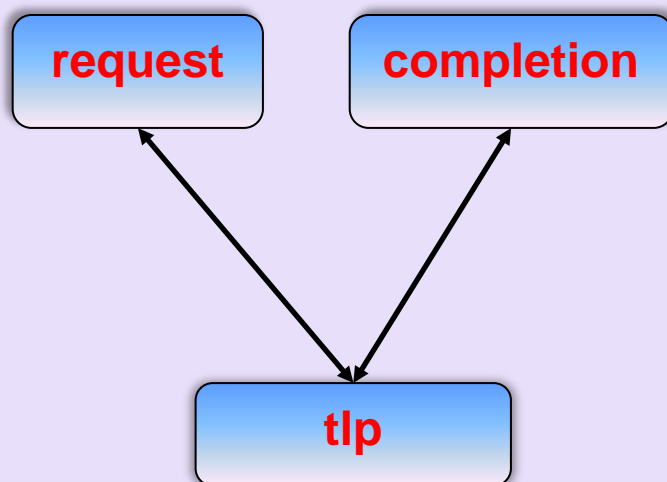


Implementation

- Focus on changes while implementing Protocol Multiplexing without rejecting the earlier architecture, so as to facilitate code reuse.
 - ✓ Transaction Layer
 - ✓ Data Link Layer
 - ✓ Physical Layer
 - ✓ PMUX Packet class

Transaction Layer

- No Changes required
- Protocol Multiplexing has its own PMUX Protocol Layer

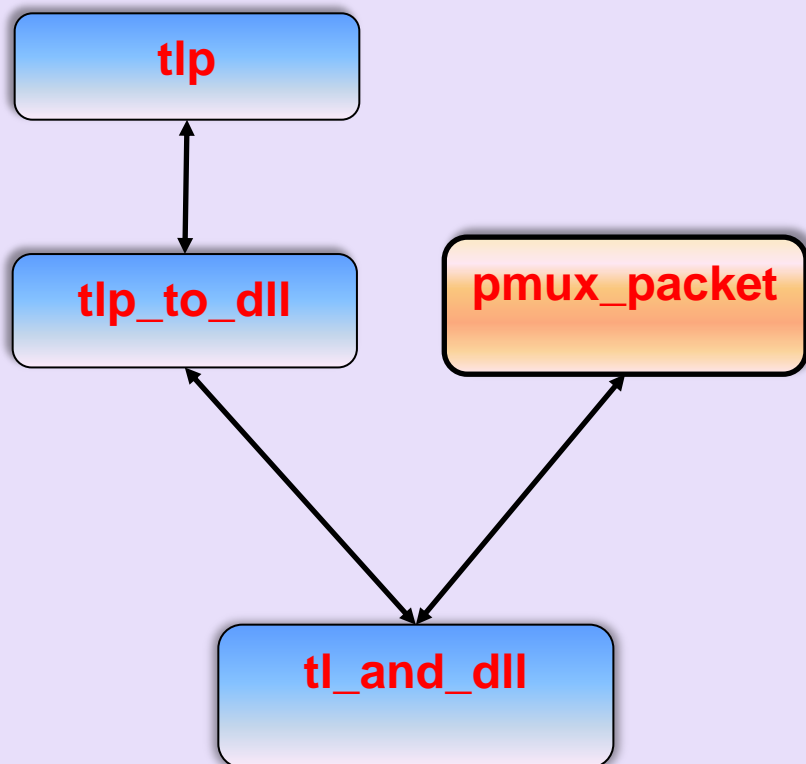


✓ PCI Express flow:

- request and completion are two top level call of transaction layer.
- tlp is the lowest with complete transaction as the black box. This black box serves as the input to Data Link Layer.

Data Link Layer

- Black box tlp call the tlp_to_dll then tl_and_dll
- A new pmux_packet from PMUX Protocol Layer mapped to tl_and_dll



✓ PMUX Packet flow:

- pmux_packet call is made from same level of the PCI Express as tlp_to_dll call.
- pmux_packet message has pmux_data, pmux_metadata and pmux_channel_id fields.
- tl_and_dll add the lcrc to the packets depending upon tlp or pmux packet.

Data Link Layer

- Once the PMUX Packet is available then the following logic needs to be bypassed. No entry should be made for transmitted/received PMUX Packets.
 - ✓ Update flow control mechanism
 - ✓ Retry Management
 - ✓ Sequence number increment
 - ✓ ACK/NAK handshaking

- PMUX LCRC algorithm is identical to TLP LCRC except
 - ✓ Seed value is FB3E E248h (TLP uses FFFF FFFFh)

TLP and PMUX at DLL

✓ TLP Packet :

- tlp_to_dll takes dynamic data array from tlp as black box
- tlp_to_dll attaches 12-bit sequence number to tlp
- tlp_to_dll add 4-bit reserved field

✓ PMUX Packet :

- pmux_packet has dynamic pmux_data array which corresponds to tlp black box
- pmux_packet has specific pmux_metadata like sequence number
- pmux_packet has 2-bit pmux_channel_id which corresponds to the lower 2-bits of reserved field in tlp

✓ Implementation Note:

- In PCIe 1.x/PCIe 2.x higher 2-bits of reserved field comes from the inverted pmux_channel_id bits
- In PCIe 3.0 its fixed to 2'b11

Physical Layer

tl_and_dll

If (PCIe 1.x/PCIe 2.x) and
 $(\sim \text{rsvd_bits}[3:2] == \text{rsvd_bits}[1:0])$ then
 \Rightarrow It's a PMUX Packet

else if (PCIe 3.0) and
 $(\sim \text{rsvd_bits}[3:2] == 2'b11)$ then
 \Rightarrow It's a PMUX Packet

else
 \Rightarrow It's a PCI Express TLP

symbols

wires

- ✓ In physical layer PMUX Packet can be decoded by reserved bits.
- ✓ PMUX Protocol checking for length field can be done
- ✓ For PCIe 3.0 the length field changes in STP Token as:
 - Length reduced to 7-bits
 - Remaining 2-bits used by PMUX Channel ID
 - And 2-bits used for PMUX Flag (2'b11)
 - Length = {length[6:0], PMUX channel ID[1:0], PMUX flag}

PMUX Packet class

- Below mentioned pmux_packet class can be used to initiate PMUX Packets.

```
class pcie_device_end_pmux_packet extends mvc_sequence_item;
```

```
rand bit [1:0] pmux_channel_id;
```

PMUX Channel ID identifies which protocol is associated with the PMUX Packet

```
rand bit [11:0] pmux_metadata;
```

PMUX Packet Metadata provides information about the PMUX Packet

```
rand int unsigned pmux_data[ ];
```

PMUX Packet Data consists of between 0 to 125 DWORDS of Data for the PMUX Packet. Both inclusive.

```
rand bit pmux_nullified;
```

PMUX Packet Nullified for nullifying the PMUX Packet.

```
constraint pmux_data_constraint
{pmux_data.size() <= 125;}
```

```
endclass
```

SPI4 over PMUX

- SPI4 Packet types
 - ✓ Training Control and Data.
 - ✓ Payload Control and Data Burst.
- Mapping SPI4 Packets with PMUX
 - ✓ Training and Payload types can be differentiated by different values in PMUX Metadata.
 - ✓ Control and Data field values can be mapped to dynamic PMUX Packet Data field.

Note: Higher PMUX Protocol layer can handle the implementation specific logic of sending/receiving of these packets as per requirement.

SPI4 Packet mapping

- One to one fields mapping of SPI4 packets with PMUX packet

PMUX Channel ID => Channel mapped to SPI4

PMUX Packet Metadata => Type of packet

(Training Control, Data,
Payload Control or Data).

PMUX Packet Data => Dynamic width varies depending
upon the actual data contained
in the type of packet.

**Advantage: SPI4 packets can be transmit/receive at a high
speed of PCI Express.**

Ethernet over PMUX

- Ethernet Packet types
 - ✓ Local Fault and Remote Fault.
 - ✓ Control Frame and Data Frame.
- Mapping Ethernet Packets with PMUX
 - ✓ Packet types can be differentiated by different values in PMUX Metadata.
 - ✓ Fault, Control and Data field values can be mapped to dynamic PMUX Packet Data field.

Note: Higher PMUX Protocol layer can handle the implementation specific logic of sending/receiving of these packets as per requirement.

Ethernet Packet mapping

- One to one fields mapping of Ethernet packets with PMUX packet

PMUX Channel ID => Channel mapped to Ethernet

PMUX Packet Metadata => Type of packet

(Local Fault, Remote Fault,
Control or Data Frame).

PMUX Packet Data => Dynamic width varies depending
upon the actual data contained
in the type of packet.

**Advantage: Ethernet packets can be transmit/receive at the
speed of PCI Express.**

Multiple Protocol over PMUX

- Multiple protocols can be supported via single PCI Express Link over PMUX .
 - ✓ With the PMUX Channel ID different protocols can be mapped like
 - Channel ID 00 => SPI4
 - Channel ID 01 => Ethernet and so on.
 - ✓ PMUX Protocol layer can map and decode depending upon the PMUX Channel ID mapped to the corresponding Protocol higher layer.

Configuration Space

- Protocol Multiplexing is controlled by the registers in Configuration space of PCI Express component.
 - ✓ Protocol Multiplexing is disabled by default.
 - ✓ Protocol Multiplexing support is indicated by the presence of the PMUX Extended Capability.

PMUX Control

- Rules apply to components that support Protocol Multiplexing:
 - ✓ PMUX Packets received in a PMUX Channel that is not enabled are silently ignored.
 - ✓ PMUX Packet may not be transmitted unless the associated PMUX Channel is enabled.
 - ✓ Each PMUX Channel must be explicitly enabled by software at the end of each associated Link.
 - ✓ PMUX Channel also require additional, protocol specific initialization in configuration space before PMUX Packets may be transmitted.

PMUX Extended Capability

31	16 15	0	Byte Offset
PMUX Extended Capability Header			00h
PMUX Capability			04h
PMUX Control			08h
PMUX Status			0Ch
PMUX Protocol Array [1]			10h
PMUX Protocol Array [2]			14h
⋮			
PMUX Protocol Array [62]			104h
PMUX Protocol Array [63]			108h

- This capability contains a list (number) of protocols supported by the Link (the PMUX Protocol Array).
- Capability must be present in both Upstream and Downstream Ports of a Link in order for Protocol Multiplexing to be successfully enabled.

PMUX Control Register

31	30	29	24 23 22 21				16 15 14 13				8 7 6 5				0								
RsvdP		PMUX Channel 3 Assignment				RsvdP		PMUX Channel 2 Assignment				RsvdP		PMUX Channel 1 Assignment				RsvdP		PMUX Channel 0 Assignment			

■ PMUX Channel n Assignment

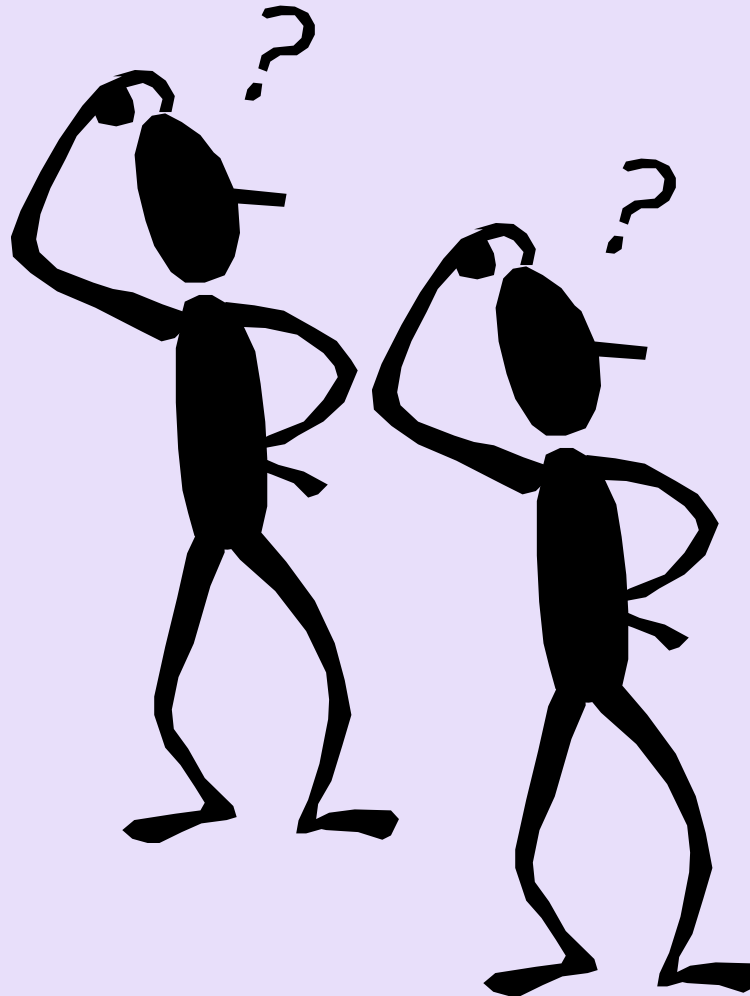
- ✓ Indicates the protocol assigned to PMUX Channel n .
 - If 0 (zero) no protocol is assigned
 - If non-zero, it is the index in the PMUX Protocol Array of the protocol assigned to the PMUX Channel n .
 - If PMUX Protocol Array size is 0 (zero), this entire field must be hardwired to 0 (zero).

Note: where n is the PMUX Channel ID – 0, 1, 2 & 3.

Conclusion

- When PMUX is enabled, then PCI Express TLPs can't use 4-bit reserved field of Data Link Layer.
- Serial protocols like SPI4 and Ethernet will have the great advantage of Protocol Multiplexing.
 - ✓ Protocol having extensive handshaking needs to implement more logic in PMUX Protocol layer.
 - ✓ Where as the protocol having less handshaking can be easily mapped to PMUX.

Questions



Thank you for attending the
PCI-SIG Developers Conference 2011.

For more information please go to
www.pcisig.com