1. Introduction

1.1. Basic Operation

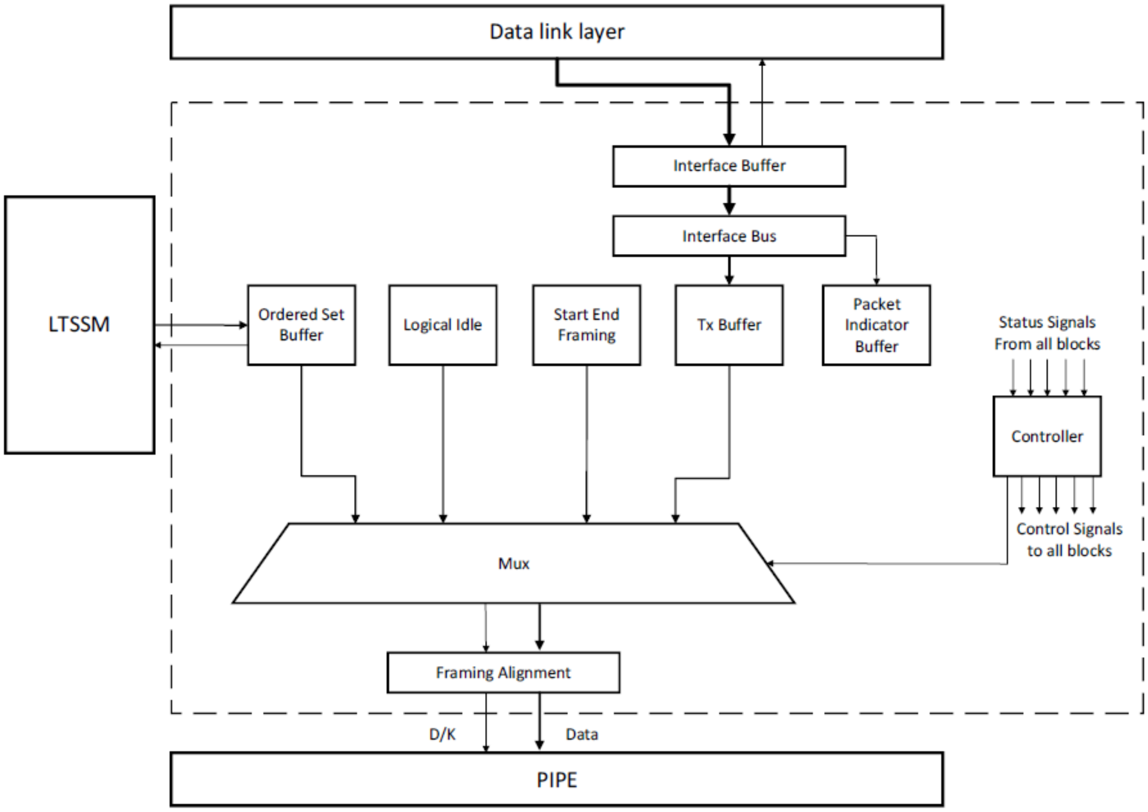
The group of modules (Start-End Framing, PCIe Mux, and Framing Alignment) will be treated as a bundled module.

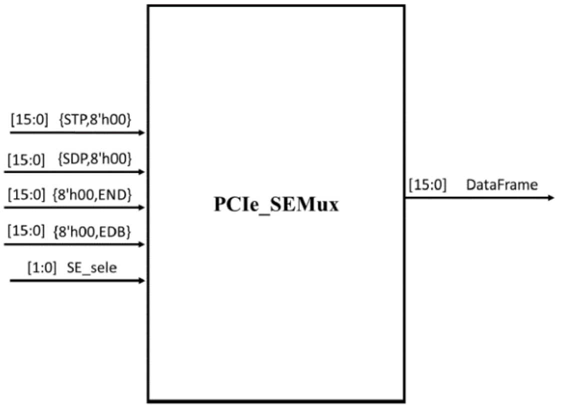
Start-End Framing is a simple Mux that is used to add 1 byte of characters to indicate the start of a TLP (STP), the start of DLLP (SDP), the end of a correct data (END), or the end of an incorrect data (EDB).

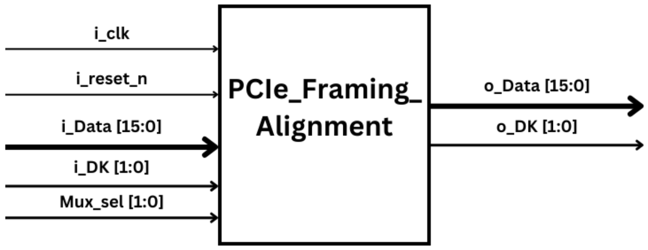
PCIe Mux is also a simple mux that is used to choose between two bytes data from four different modules. These modules are the Tx buffer, the Start–End Framing, the logical idle and the ordered set buffer.

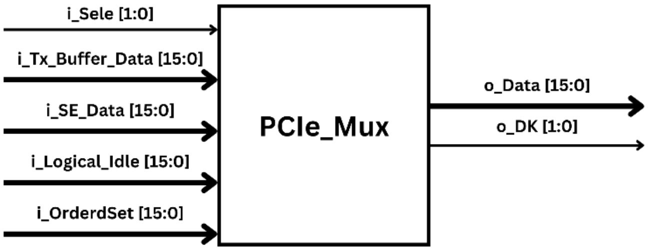
Framing Alignment is used only when the Data Mux chooses the Start-End Framing, and its main job is to remove the 2 bytes added from the Start-End Framing.

1.2. Block I/O Signals









1.3. Interaction with Other Modules

Start-End Framing takes the SE_Sele from the controller and supplies DataFrame to the PCIe_Mux.

PCIe Mux takes the i_Sele from the controllers and supplies its output to the Framing Alignment module.

Framing Alignment takes the Mux_Sel from the controller and supplies its output to the PIPE.

2. I/O Signals Description

2.1. Inputs

Name	Width	Source	Description
{STP, 8'h00}	16 bits	16’hFB00	Start of TLP
{SDP, 8'h00}	16 bits	16’h5C00	Start of DLLP
{8'h00, END}	16 bits	16’h00FD	End of the correct TLP or DLLP
{8'h00, EDB}	16 bits	16’h00FE	End of the TLP or DLLP with an error
SE_sele	2 bits	Controller	The selection line of the SEMux. 00 → STP 01 → SDP 10 → END 11 → EDB
i_Tx_Buffer_Data	16 bits	TX Buffer	The two bytes data of the packet.
i_SE_Data	16 bits	Start-End Framing	The two bytes of the Start – End characters.
i_Logical_Idle	16 bits	Logicle Idle	The two bytes of the logical Idle characters.
i_OrderdSet	16 bits	OS Buffer	The two bytes of the ordered set packets.

i_Sele	2 bits	Controller	The selection line of PCIe Mux & Framing Alignment. 00 → Tx Buffer Data 01 → Start-End Framing Data 10 → Logical Idle 11 → OS Buffer Data

2.2. Outputs

Name	Width	Destination	Description
o_Data	16 bits	PIPE	Two Bytes data after Framing Alignment
o_DK	2 bits	PIPE	One bit for each byte. If the bit equals 0, it means the corresponding byte is data. If the bit equals 1, it means the corresponding byte is control.

3. Detailed Operation

3.1. Start-End Framing

For the Start-End Framing, it is just a simple mux that takes its select signal from the controller and chooses which DataFrame to use as input to the PCIe Mux.

The selection line of the Start-End Framing Mux is as follows.

- 00 → STP (Start of TLP)
- 01 → SDP (Start of DLLP)
- 10 → END (End of the correct TLP or DLLP)
- 11 → EDB (End of the TLP or DLLP with an error) (not deployed)

These characters have constant values which are implemented in the top module of the transmitter.

Their values are as follows:

- STP = 8'hFB
- SDP = 8'h5C
- END = 8'hFD
- EDB = 8'hFE

3.2. PCIe Mux

For the PCIe Mux, it chooses which data to pass based on the Select signal coming from the controller.

The selection line of the PCIe Mux is as follows:

- 00 → Tx Buffer Data
- 01 → Start-End Framing Data
- 10 → Logical Idle
- 11 → OS Buffer Data

It also generates a control signal as an output as follows:

- Tx Buffer is data signal (o_DK = 2’b00)
- Start-End Framing is control signal (o_DK = 2’b11)
- Logicle Idle is data signal (o_DK = 2’b00)
- OS Buffer is data signal unless it has one of the following (COM, PAD, SKP, FTS, IDL, EIE) signal which will be treated as a control byte. These signals have the following values:
    - PAD=8'hF7
    - COM=8'hBC
    - SKP=8'h1C
    - FTS=8'h3C
    - IDL=8'h7C
    - EIE=8'hFC

        - Ex: data = 16’hF7AB (PAD signal), therefore o_DK = 2’b10.

3.3. Framing Alignment

Framing Alignment block is used to remove the overhead bytes that are added in Start–End framing module only if the PCIe Mux chose the SE framing, otherwise, it just passes the data and control signals exactly as it got them from the PCIe Mux.

4. Configuration

4.1. Configuration Sequence

To configure both the PCIe Mux and the Framing Alignment, you have to give initial values to the: Tx Buffer, SE Mux, OS Buffer.

4.2. Latency

SE Framing and PCIe Mux are combinational blocks while the Framing alignment block takes one clock cycle.

4.3. Input Format and Constraints

SE Mux inputs are constant values and cannot be changed as follows:

- 16’hSTP00
- 16’hSDP00
- 16’h00END
- 16’h00EDB

PCIe Mux has the following inputs:

1. Tx Buffer (no constraints)
2. SE Mux (can only take 4 values as mentioned above)
3. Logical Idle (can only take 16’h0000)
4. OS Buffer (no constraints)
5. Select signal (no constraints)

Framing Alignment module doesn’t have any constraints.

5. Sequences and Waveforms

For the sequences and reference model, we will ignore the SE Mux for convenience, also it is a simple mux so no need to verify it.

5.1. Sequence 1: Tx Buffer Data

Inputs:

i_Tx_Buffer_Data → Randomized.

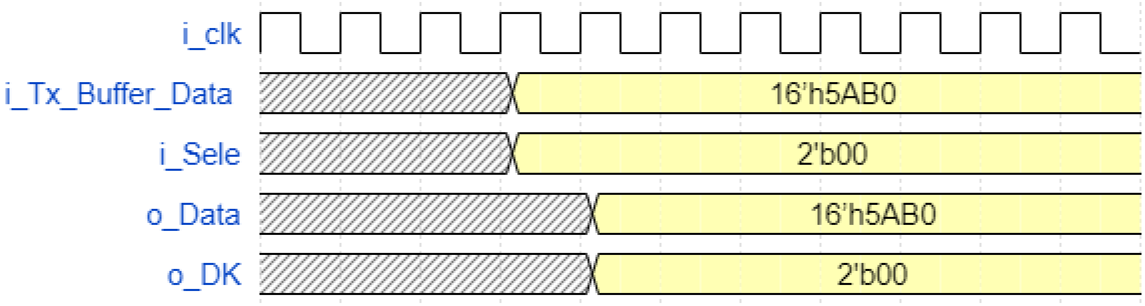
i_Sele → 2’b00

Outputs:

o_Data → i_Tx_Buffer_Data

o_DK → 2’b00

Waveform(s):



5.2. Sequence 2: Start-End Framing Data

Inputs:

i_SE_Data → Randomized within the 4 values it can be.

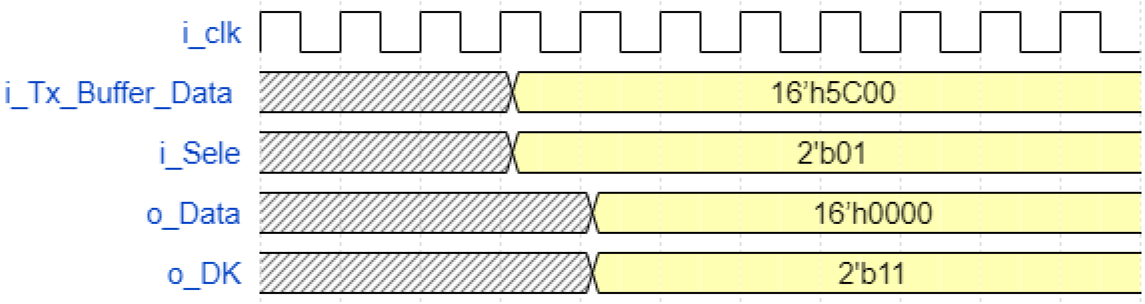
i_Sele → 2’b01

Outputs:

o_Data → 16’h0000

o_DK → 2’b11

Waveform(s):



5.3. Sequence 3: Logical Idle

Inputs:

i_Logical_Idle→ 16'h0000.

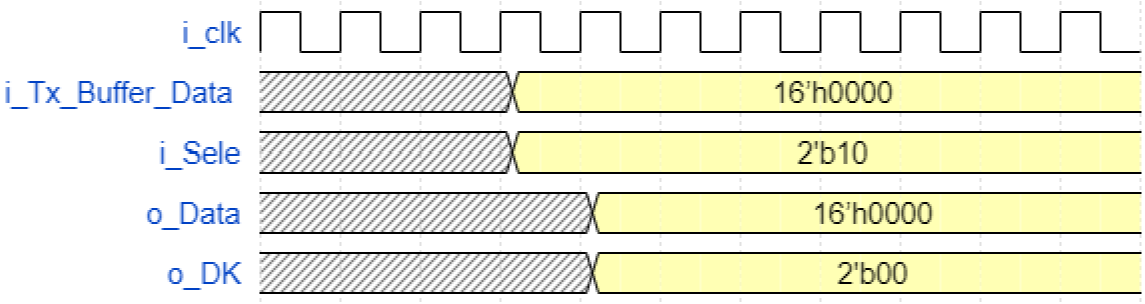
i_Sele → 2'b10

Outputs:

o_Data → 16'h0000

o_DK → 2'b00

Waveform(s):



5.4. Sequence 4: OS Buffer data

Inputs:

i_OrderdSet→ 16′ h3C00. (Most significant byte is FTS symbol)

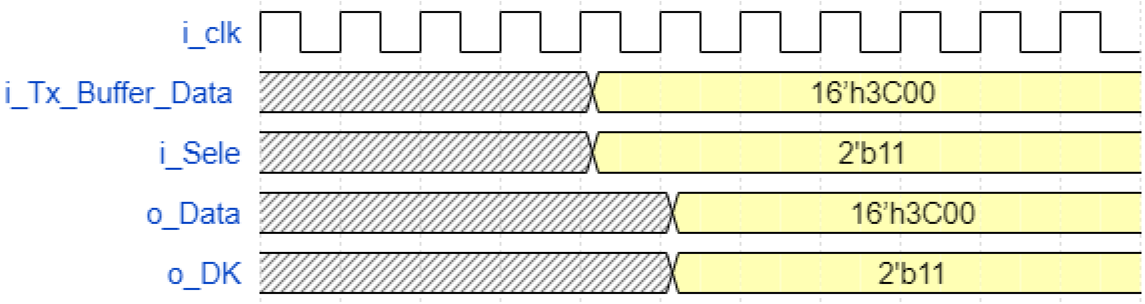
i_Sele → 2′b11

Outputs:

o_Data → 16′h3C00

o_DK → 2′b10

Waveform(s):



There are many sequences that can be generated using the above sequence by changing the i_OrderedSet to random values or values with control symbols which are shown in section 3.2.