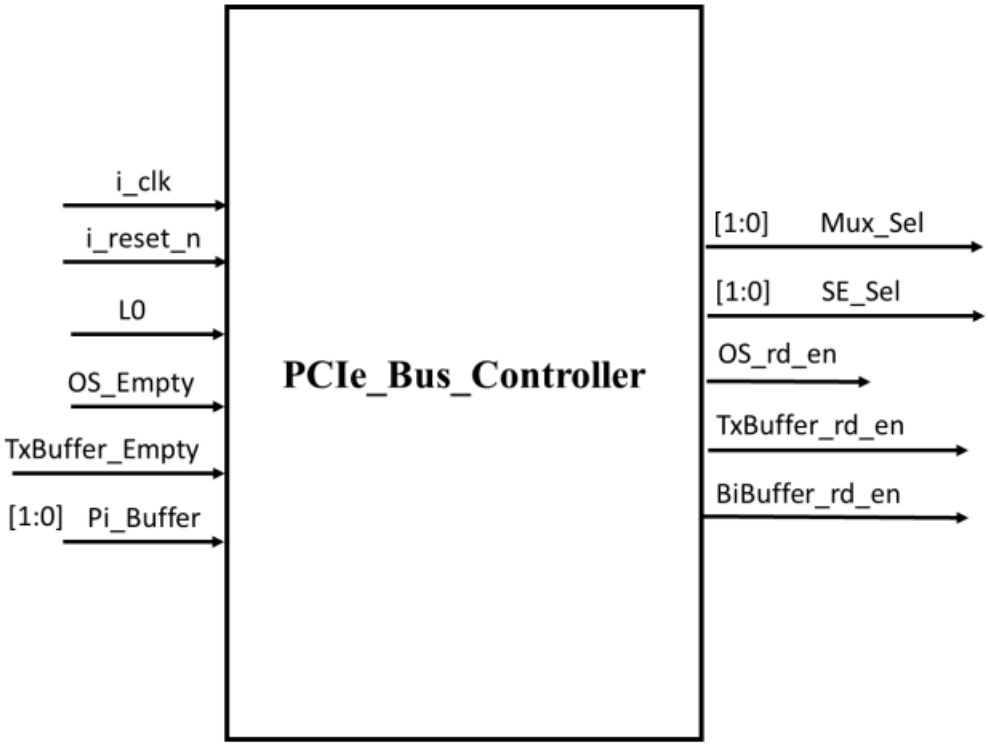
Transmitter Controller IP Specification

1. Introduction

1.1. Basic Operation

The Tx Controller is the brain of the Tx block as it controls all operations of the Tx block during link training and normal data transmission. Druring Link training, the Transmitter will be receiving data from the LTSSM and transmitting it to the PIPE interface, while during normal operation, the data will be coming from the higher Data-Link layer.

1.2. Block I/O Signals



1.3. Interaction with Other Modules

The controller receives status signals from the Tx buffer, Packet Indicator buffer, Ordered set buffer and LTSSM. And it sends control signals to the MUX, Tx buffer, Packet Indicator buffer, Ordered Set buffer and the Start – End framing. The output signals control the flow of data and determine who has priority to transmit to the PIPE interface depending on the current state.

2. I/O Signals Description

2.1. Inputs

Name	Width	Source	Description
i_clk	1 bit	**Global**	**Clock signal at positive edge registers**
i_reset_n	1 bit	**Global**	**Synchronous negative-edge reset**
L0	1 bit	**Input from the LTSSM**	If it equals 1, the normal operation mode is activated. If it equals 0, LTSSM is still in link training or initialization mode.
OS_Empty	1 bit	**Input from the Ordered set buffer**	If it equals 1, the ordered set buffer is empty. If it equals 0, the ordered set buffer is not empty.
TxBuffer_Empty	1 bit	**Input from the Tx buffer**	If it equals 1, the Tx buffer is empty. If it equals 0, the Tx buffer is not empty.
Pi_Buffer	2 bits	**Input from the packet indicator buffer**	The two bits data of the packet indicator: • 00 = invalid data. • 01 = current byte is the start of a transaction packet

			• 10 = current byte is the start of a data link packet • 11 = Valid data,but not the start of a packet

2.2. Outputs

Name	Width	Destination	Description
Mux_Sel	2 bits	**Output to the Multiplexer**	The selection line of the multiplexer. • 00 = select Tx_buffer data. • 01 = select SE_buffer data. • 10 = select logical idle signal • 11 = select OS_buffer data
SE_Sel	2 bits	**Output to the Start – End Framing**	The selection line of the multiplexer of the start – End Framing. • 00 = select STP signal (start of transaction packet). This is also the default value when not in use. • 01 = select SDP signal (start of data link packet). • 10 = select END signal. • 11 = select EDB signal (not used in this implementation).
OS_rd_en	1 bit	**Output to the Ordered set buffer**	If it equals 1, the ordered set buffer outs two bytes of its data.

TxBuffer_rd_en	1 bit	**Output to the Tx_buffer**	If it equals 1, the Tx buffer outs two bytes of its data.
BiBuffer_rd_en	1 bit	**Output to the Packet indicator buffer**	If it equals 1, the Packet indicator buffer outs two bytes of its data.

3. Detailed Operation

The function of the controller can be described in the following points:

• If L0 equals zero, the MUX can't take data from Tx buffer.

• If there is data on the ordered set buffer, the MUX should take this data until the ordered set buffer gets empty.

• If the ordered set buffer is empty and the Tx buffer is empty, the MUX should select the logical idle.

• If the ordered set buffer is empty and the Tx buffer is not empty, the MUX should select the Start – End framing with the correct choice then take the data from the Tx buffer.

• If a new data comes to the ordered set buffer while the MUX selects the Tx buffer, the MUX should still select the Tx buffer till the current packet is finished with END then select the ordered set buffer.

More Specifically, During normal transmission (L0 = 1)

- The data mux will choose the logica_idle signal if both the OS_buffer and the Tx_buffers are empty.

- If the OS_buffer is not empty, the data_mux will select its data and the OS_rd_en signal will be driven high.
- If the OS_buffer is empty but the Tx_buffer is not empty, the data_mux will choose between the SE_data and the Tx_data based on the signal for data packet type coming from the Pi_buffer.
    - If the Pi_buffer gives 00 (invalid data), the data_mux will choose the logical_idle signal.
    - If the PI_buffer gives 01 or 10 (STP or SDP), the SE_sel signal will select the corresponding output (00 or 01) and the data mux will choose the data from the SE_mux.
    - If the PI_buffer gives 11 (valid data but not start of packet), the Data_mux will select the data from the Tx_buffer and the TxBuffer_rd_en and BiBuffer_rd_en signals will be driven high.
- If the Tx_buffer gets empty, or the PIbuffer transitions from 11 (non-start signal) to 10 or 01 (start signals) while transmitting from the Tx_buffer, an end of packet signal must be transmitted regardless of the whether the OS_buffer is empty or not. The SE_Sel signal will be driven to 10 (END) and data_mux will be choose the data from the SE_mux.

During Link Training (L0 = 0):

During link training (L0 = 0), the controller will command the Data_mux to only take data from the OS_buffer. The Mux_Sel signal will choose the OS_data and the OS_rd_en signal will always be driven high whenever the OS_buffer is not empty. The data_mux will then take data from the OS_Buffer if it is not empty. If the OS_buffer is empty, the Logical Idle signal will be chosen by the Data_mux.

The only exception is when the Data_mux was currently taking data from the Tx_Buffer befor exiting L0. In that case, the controller will continue to choose data from the Tx_buffer until it gets empty. Then, an end packet character will be transmitted from the SE mux.

4. Configuration

4.1. Configuration Sequence

Only a reset sequence is needed to configure this module.

4.2. Latency

All outputs take only one clock cycle to come out.

4.3. Input Format and Constraints

No Specific constraints are needed for the inputs..

5. Example Sequences and Waveforms

5.1. Reset sequence:

Description:

This sequence resets the state of the controller.

Inputs:

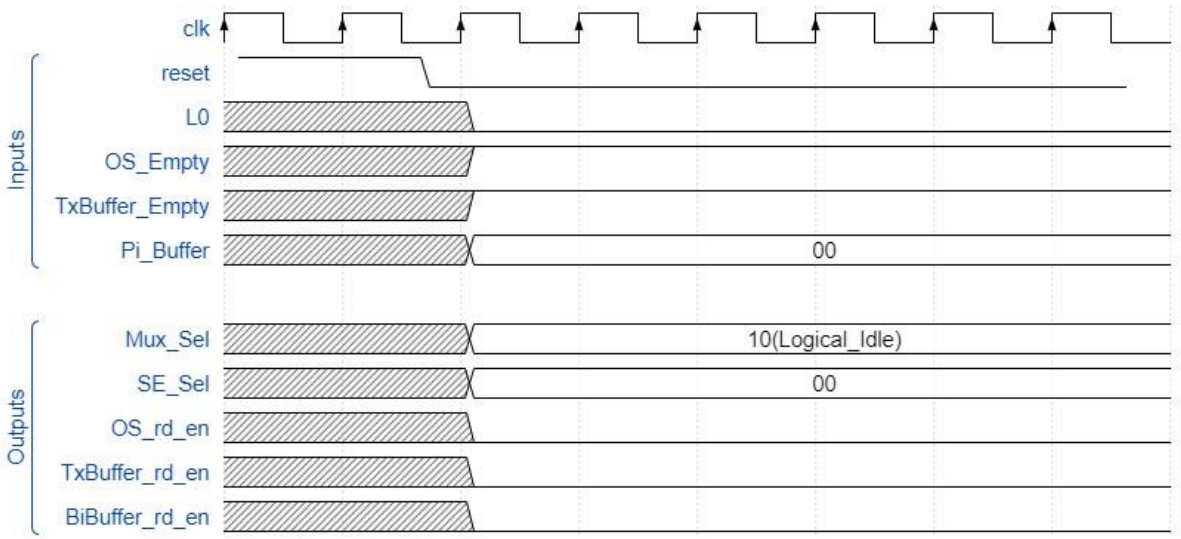
Input signal:	i_reset_n	L0	OS_Empty	TxBuffer_Empty	Pi_Buffer
value	0	0	1	1	00

Outputs:

Output signal:	Mux_Sel	SE_Sel	OS_rd_en	TxBuffer_rd_en	BiBuffer_rd_en

value	10 (logical idle)	00	0	0	0

Waveform(s):



5.2. Normal-Operation-Priority sequence:

Description:

This sequence tests which buffer (Tx_buffer or OS_buffer) takes priority during transmission.

- If the data mux is currently selecting data from the OS_buffer while new data comes into the Tx_buffer (TX_empty turns 0), then the data mux will keep selecting from the OS_buffer until it gets empty.
- If the data mux is currently selecting data from the Tx_buffer while new data comes into the OS_buffer (OS_empty turns 0), then the data mux will keep selecting from the Tx_buffer until it gets empty. And end of packet signal must also be transmitted from the SE_mux before starting to select the OS_buffer.
- If the mux is not selecting either the TX_buffer or the OS_buffer, but both of them receive dataat the same time, then the OS_buffer takes priority and will be selected.

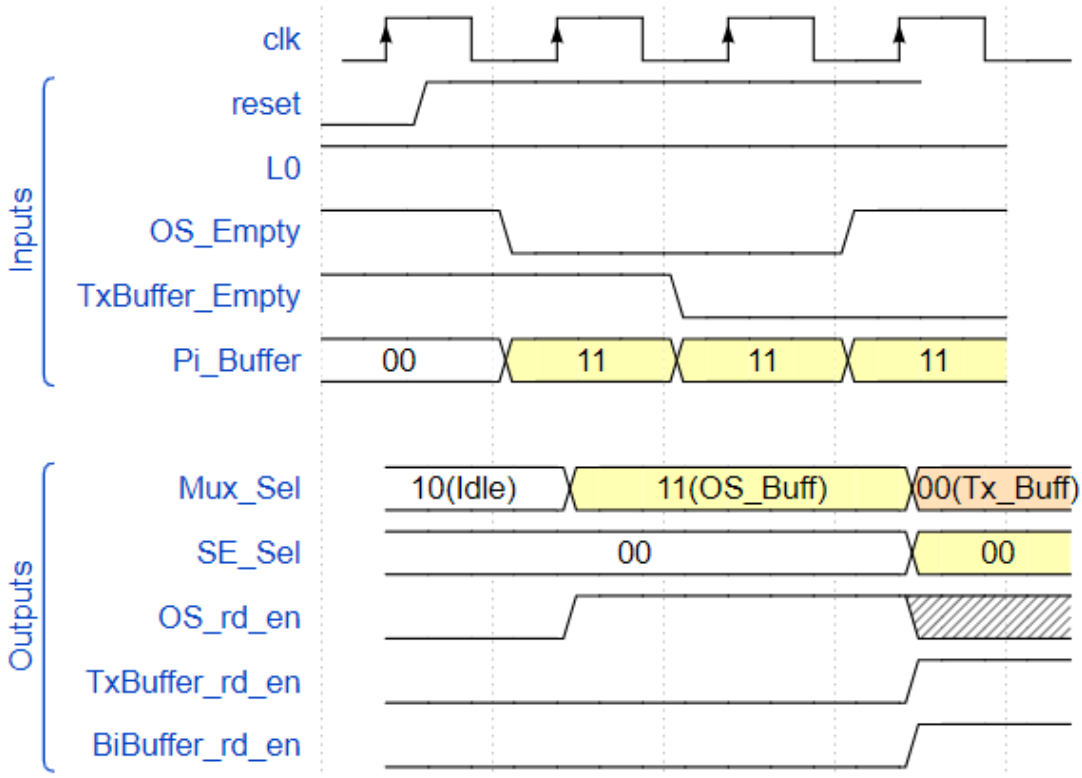
Inputs:

Input signal:	i_reset_n	L0	OS_Empty	TxBuffer_Empty	Pi_Buffer
Value 1:	1	1	0	1	11
Value 2:	1	1	0	0	11
Value 3:	1	1	1	0	11

Outputs:

Output signal:	Mux_Sel	SE_Sel	OS_rd_en	TxBuffer_rd_en	BiBuffer_rd_en
Value 1:	11 (OS_buffer)	00	1	0	0
Value 2:	11 (OS_buffer)	00	1	0	0
Value 3:	00 (Tx_buffer)	00	0	1	1

Waveform(s):



5.3. SE_mux sequence 1:

Description:

The SE_mux output is transmitted through the main data_mux to signal the start or end of a data packet.

- STP and SDP characters (start-transaction-packet and start-datalink-packet) are generated (SE_sel = 00 or 01 respectively) and selected by the main Data_mux (Mux_Sel = 01) when:
    - 1) The OS_buffer is empty.
    - AND 2) The Tx_buffer is not empty.
    - AND 3) The PI_Buffer input shows that the current packet has started (PIBuffer = 01 or 10 respectively).

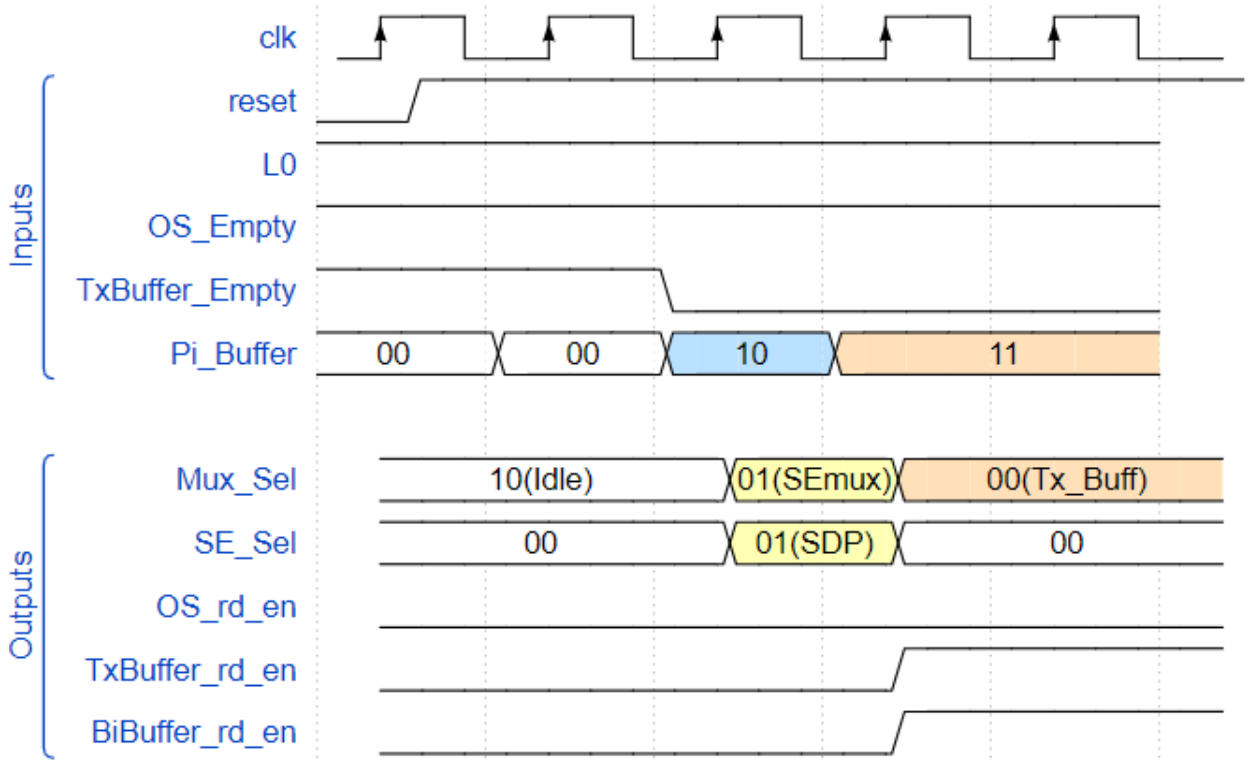
Inputs:

Input signal:	i_reset_n	L0	OS_Empty	TxBuffer_Empty	Pi_Buffer
Value 1:	1	1	1	1	00
Value 2:	1	1	1	0	10
Value 3:	1	1	1	0	11

Outputs:

Output signal:	Mux_Sel	SE_Sel	OS_rd_en	TxBuffer_rd_en	BiBuffer_rd_en
Value 1:	10 (Idle)	00	0	0	0
Value 2:	01 (SE_mux)	01	0	0	0
Value 3:	00 (Tx_buffer)	00	0	1	1

Waveform(s):



5.4. SE_mux sequence 2:

Description:

- An END character is generated (SE_sel = 10) and selected by the main Data_mux (Mux_Sel = 01) when:
    - 1) A new start of packet signal comes from from the PI_buffer (PIBuffer = 01 or 10).
    - OR 2) The Tx_buffer becomes empty.

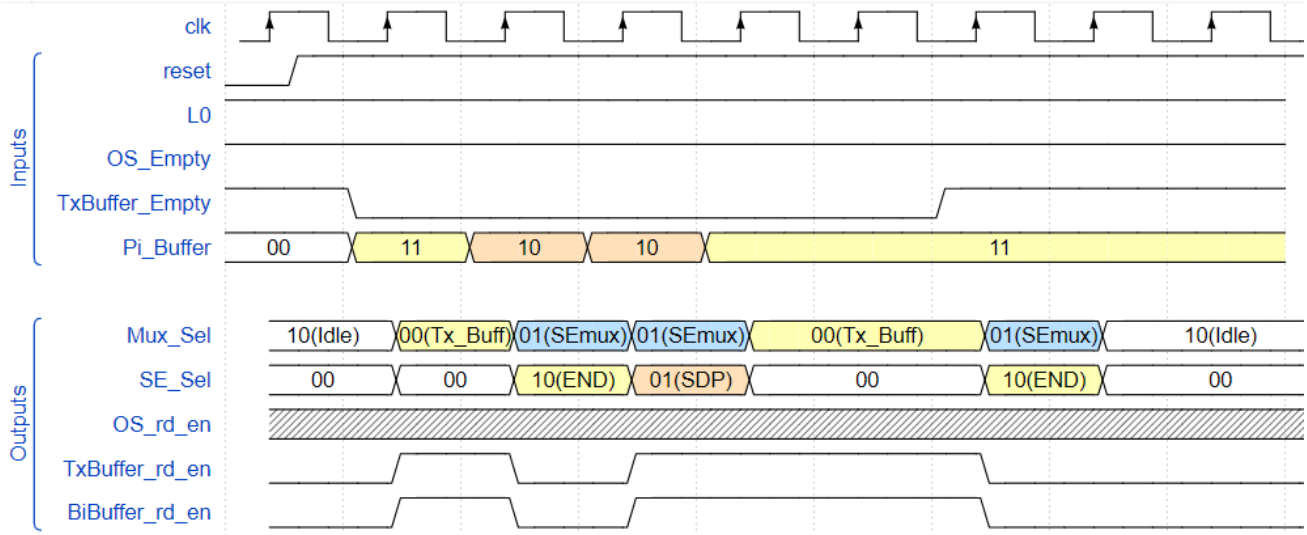
Inputs:

Input signal:	i_reset_n	L0	OS_Empty	TxBuffer_Empty	Pi_Buffer
Value 1:	1	1	1	0	11
Value 2:	1	1	1	0	10
Value 3:	1	1	1	0	10
Value 4:	1	1	1	0	11
Value 5:	1	1	1	0	11
Value 6:	1	1	1	1	11
Value 7:	1	1	1	1	11
Value 8:	1	1	1	1	11

Outputs:

Output signal:	Mux_Sel	SE_Sel	OS_rd_en	TxBuffer_rd_en	BiBuffer_rd_en
Value 1:	00 (Tx_buffer)	00	X	1	1
Value 2:	01 (SE_mux)	10(END)	X	0	0
Value 3:	01 (SE_mux)	01(SDP)	X	1	1
Value 4:	00 (Tx_buffer)	00	X	1	1
Value 5:	00 (Tx_buffer)	00	X	1	1
Value 6:	01 (SE_mux)	10(END)	X	0	0
Value 7:	10 (Idle)	00	X	0	0
Value 8:	10 (Idle)	00	X	0	0

Waveform(s):



5.5. L0 exit sequence:

Description:

If L0 is exited while a packet is being transmitted from the Tx_buffer. The OS_buffer will wait until the Tx_buffer becomes empty.

An END of packet signal will also be transmitted from the SE_mux once the Tx_buffer gets empty before taking data from the OS_buffer.

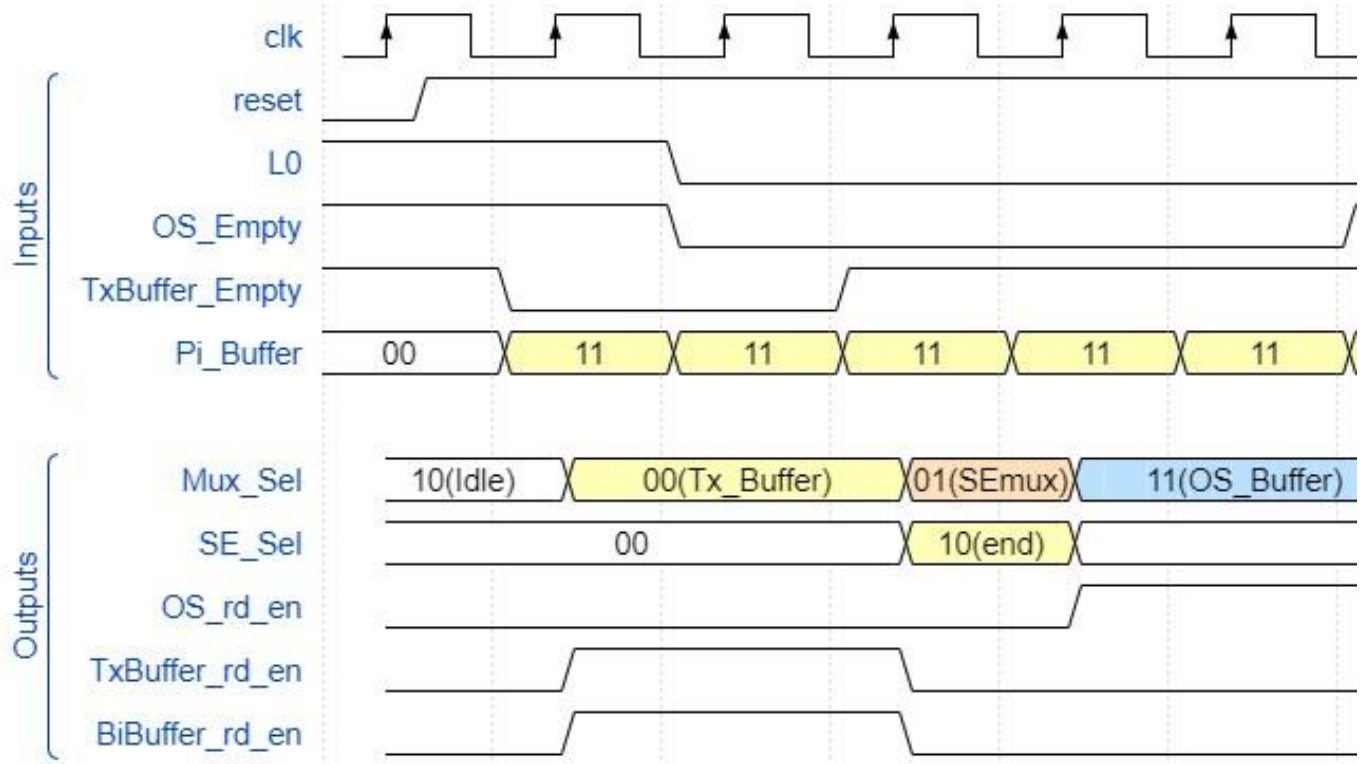
Inputs:

Input signal:	i_reset_n	L0	OS_Empty	TxBuffer_Empty	Pi_Buffer
Value 1:	1	1	1	0	11
Value 2:	1	0	0	0	11
Value 3:	1	0	0	1	11
Value 4:	1	0	0	1	11

Outputs:

Output signal:	Mux_Sel	SE_Sel	OS_rd_en	TxBuffer_rd_en	BiBuffer_rd_en
Value 1:	00 (Tx_buffer)	00	0	1	1
Value 2:	00 (Tx_buffer)	00	0	1	1
Value 3:	01 (SE_mux)	10(END)	0	0	0
Value 3:	11 (OS_buffer)	00	1	0	0

Waveform(s):



5.6. Link-Training-Priority sequence:

Description:

The Tx_buffer will never start transmitting data when not in L0.

The Data_mux only chooses The OS_buffer if it is not empty, otherwise it transmits the Logical_Idle Signal

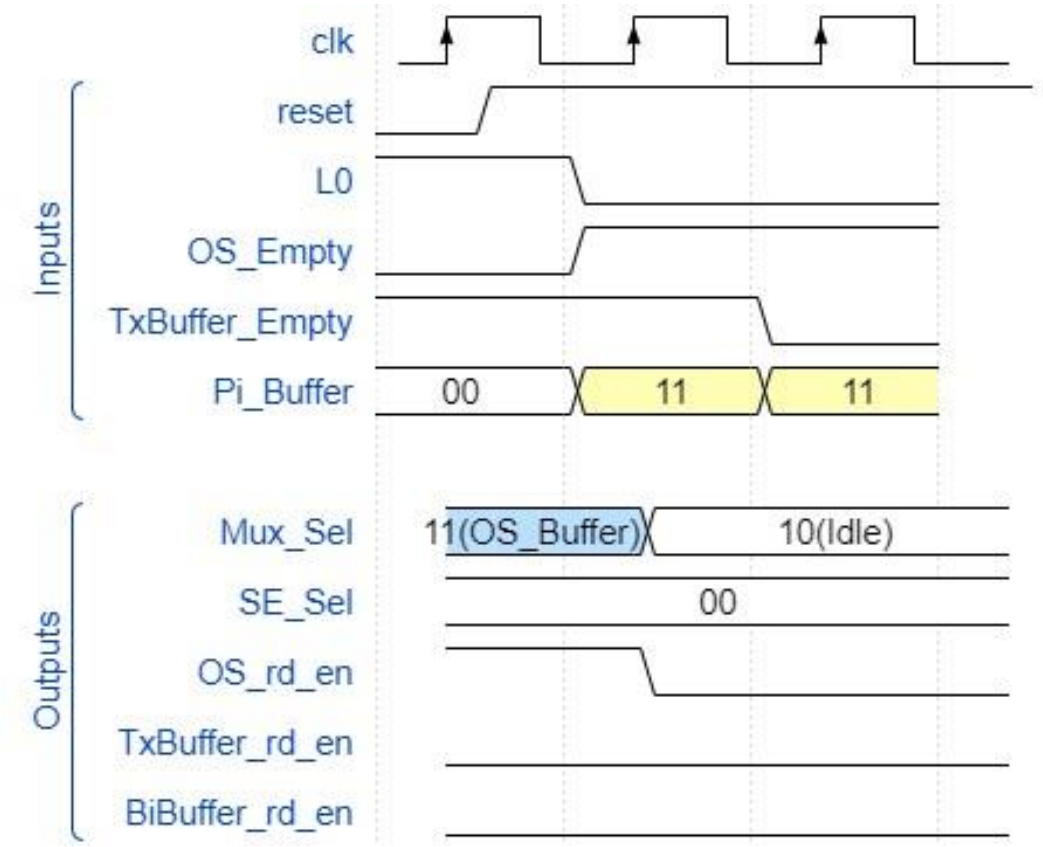
Inputs:

Input signal:	i_reset_n	L0	OS_Empty	TxBuffer_Empty	Pi_Buffer
Value 1:	1	0	0	1	11
Value 2:	1	0	1	1	11
Value 3:	1	0	1	0	11

Outputs:

Output signal:	Mux_Sel	SE_Sel	OS_rd_en	TxBuffer_rd_en	BiBuffer_rd_en
Value 1:	11 (OS_buffer)	00	1	0	0
Value 2:	10 (Idle)	00	0	0	0
Value 3:	10 (Idle)	00	0	0	0

Waveform(s):


clk
reset
L0
OS_Empty
TxBuffer_Empty
Pi_Buffer
Inputs
00
11
11
Mux_Sel
SE_Sel
OS_rd_en
TxBuffer_rd_en
BiBuffer_rd_en
Outputs
11(OS_Buffer)
10(Idle)
00