OS Creator IP Specification

1. Introduction

1.1. Basic Operation

This module is responsible for creating the required Ordered-Sets (OS) to complete the training process. It serves as an interface between the LTSSM and the Tx. The Creator is composed of 3 sub-modules, one for the creation of Training Sequence (TS) OS, the second one for the creation of other OS, and the third for creating the logical idle symbols.

1.2. Block I/O Signals

Figure 1 shows the I/O signals of the Creator module.

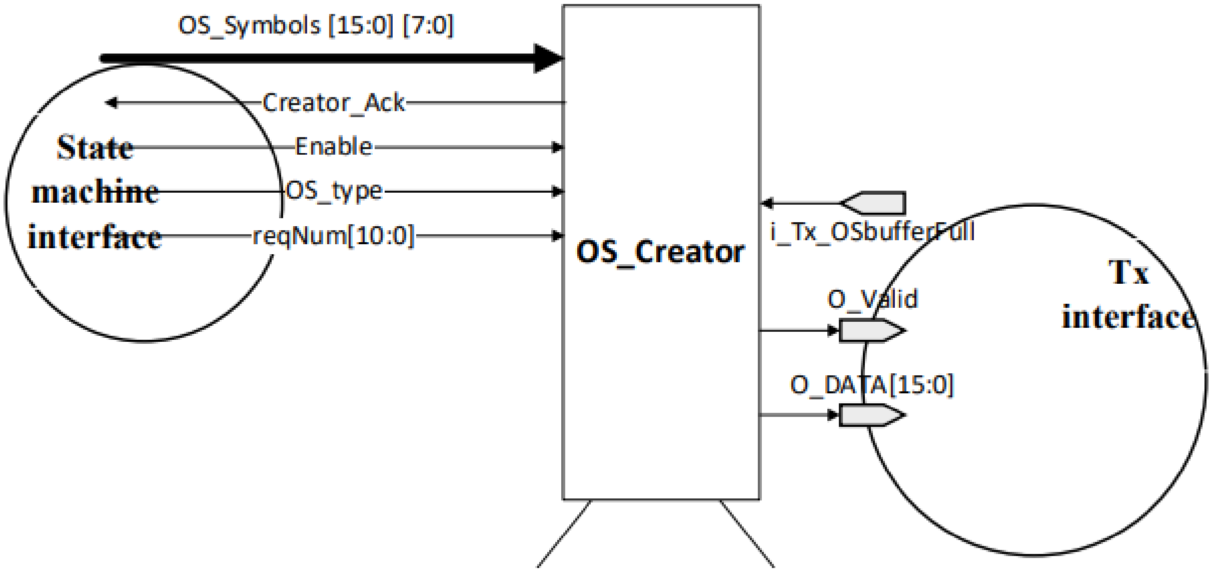

Figure 1: I/O signals of the Creator module.

1.3. Interaction with Other Modules

The Creator communicates with two modules: the LTSSM and Tx as shown in figure 2 below.

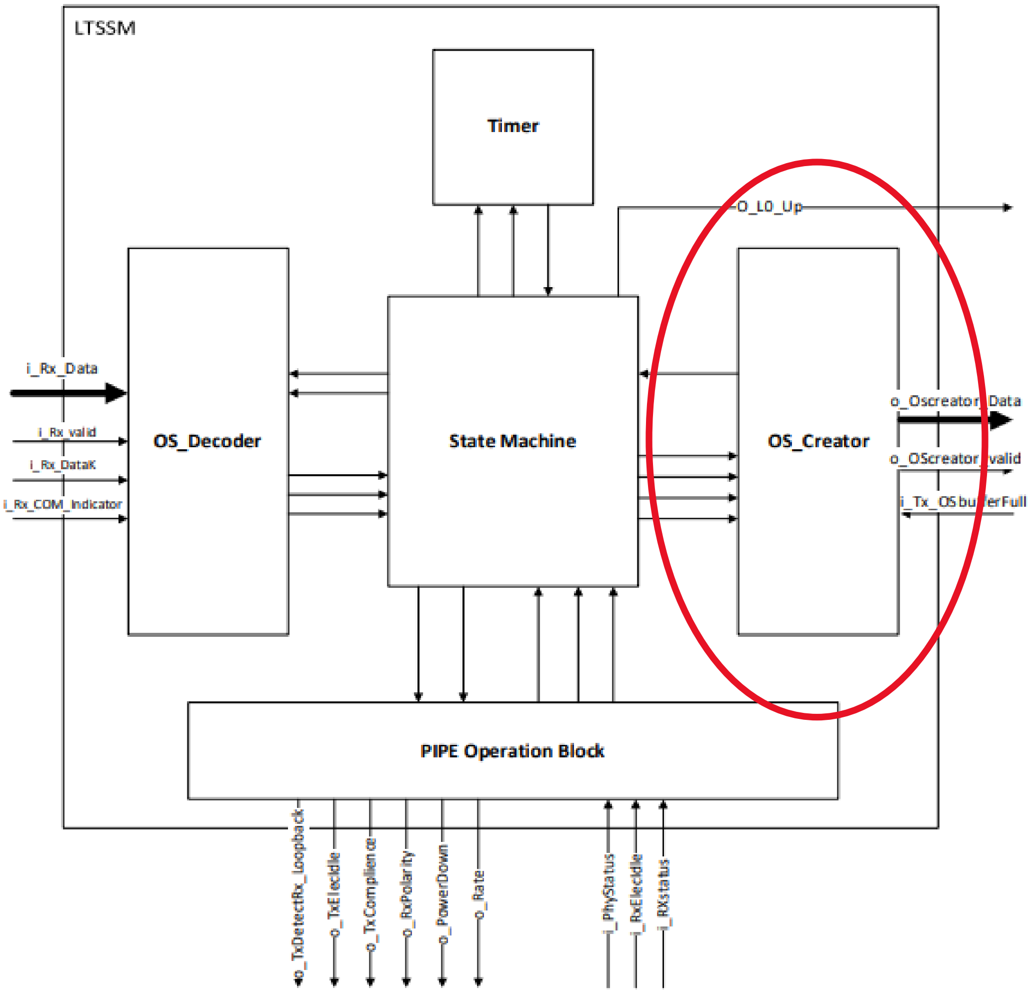

Figure 2: The Creator's location in the hierarchy.

2. I/O Signals Description

2.1. Inputs

Name	Width	Source	Description
i_clk	1 bit	Global	Clock signal for positive edge registers.
i_resetn	1 bit	Global	Synchronous low-level reset.
i_enable	1 bit	State Machine	Enable signal for the module. • 0: disable the module from capturing the input OS symbols. • 1: enable the module to capture the input OS symbols.
i_Tx_OSbufferFull	1 bit	Tx	Indicates if the transmitter buffer is full or not. If full, the LTSSM needs to stop the transmission of Ordered-Sets data. • 0: when Tx buffer is not full. • 1: when Tx buffer is full.
i_LTSSM_stateChange	1 bit	State Machine	Indicates that a state change has occurred. • 0: when no state change has occurred. • 1: when a state change has occurred.
i_reset_TS_count	1 bit	State Machine	Resets the value of the TS counter. • 0: do nothing. • 1: reset the counter.
i_OSreqNum	11 bits	State Machine	Number of repetitions of the OS.

Name	Width	Source	Description
i_OScreatorTypes	2 bits	State Machine	Chooses from the different creator OS types. • 00: Other OS. • 01: TS OS. • 10: IDLE OS. • else: no output.
i_OScreatorSymbol0	8 bits	State Machine	Value of byte 0 of the OS. Examples: PAD, COM, FTS, SKP, etc.
i_OScreatorSymbol1	8 bits	State Machine	Value of byte 1 of the OS. Examples: PAD, COM, FTS, SKP, etc.
i_OScreatorSymbol2	8 bits	State Machine	Value of byte 2 of the OS. Examples: PAD, COM, FTS, SKP, etc.
i_OScreatorSymbol3	8 bits	State Machine	Value of byte 3 of the OS. Examples: PAD, COM, FTS, SKP, etc.
i_OScreatorSymbol4	8 bits	State Machine	Value of byte 4 of the OS. Examples: PAD, COM, FTS, SKP, etc.
i_OScreatorSymbol5	8 bits	State Machine	Value of byte 5 of the OS. Examples: PAD, COM, FTS, SKP, etc.
i_OScreatorSymbol6	8 bits	State Machine	Value of byte 6 of the OS. Examples: PAD, COM, FTS, SKP, etc.
i_OScreatorSymbol7	8 bits	State Machine	Value of byte 7 of the OS. Examples: PAD, COM, FTS, SKP, etc.
i_OScreatorSymbol8	8 bits	State Machine	Value of byte 8 of the OS. Examples: PAD, COM, FTS, SKP, etc.
i_OScreatorSymbol9	8 bits	State Machine	Value of byte 9 of the OS. Examples: PAD, COM, FTS, SKP, etc.

Name	Width	Source	Description
i_OScreatorSymbol10	8 bits	State Machine	Value of byte 10 of the OS. Examples: PAD, COM, FTS, SKP, etc.
i_OScreatorSymbol11	8 bits	State Machine	Value of byte 11 of the OS. Examples: PAD, COM, FTS, SKP, etc.
i_OScreatorSymbol12	8 bits	State Machine	Value of byte 12 of the OS. Examples: PAD, COM, FTS, SKP, etc.
i_OScreatorSymbol13	8 bits	State Machine	Value of byte 13 of the OS. Examples: PAD, COM, FTS, SKP, etc.
i_OScreatorSymbol14	8 bits	State Machine	Value of byte 14 of the OS. Examples: PAD, COM, FTS, SKP, etc.
i_OScreatorSymbol15	8 bits	State Machine	Value of byte 15 of the OS. Examples: PAD, COM, FTS, SKP, etc.

2.2. Outputs

Name	Width	Destination	Description
o_OScreator_Data	16 bits	Tx	Value of the OS data.
o_OScreator_valid	1 bit	Tx	Indicates the validity of the OS data. • 0: invalid data. • 1: valid data.
o_OScreator_Ack	1 bit	State Machine	Indicates that the number of transmitted OS is equal to or more than the required number. • 0: required number reached. • 1: required number not reached.

3. Detailed Operation

As shown in the figure below, the Creator consists of four modules. Based on the values of *i_enable* and *i_OScreatorTypes*, the demultiplexer enables only one of the other three modules at an instance. The TS Creator generates TS1 and TS2 OS required for link training and initialization. The Other OS Creator generates other OS such as SKP, FTS, and IDLE. The IDLE Creator generates the logical idle (8'h00) symbols. Each of these three modules generates a number of sequences or symbols according to the value of the *i_OSreqNum* signal. 16 *i_OScreatorSymbol* input signals, each with a byte width, are used to generate each OS from symbols defined in the State Machine. All 16 symbols are inputs to the TS Creator while only the first 4 symbols are inputs to the Other OS Creator. The IDLE Creator does not need any of these symbols.

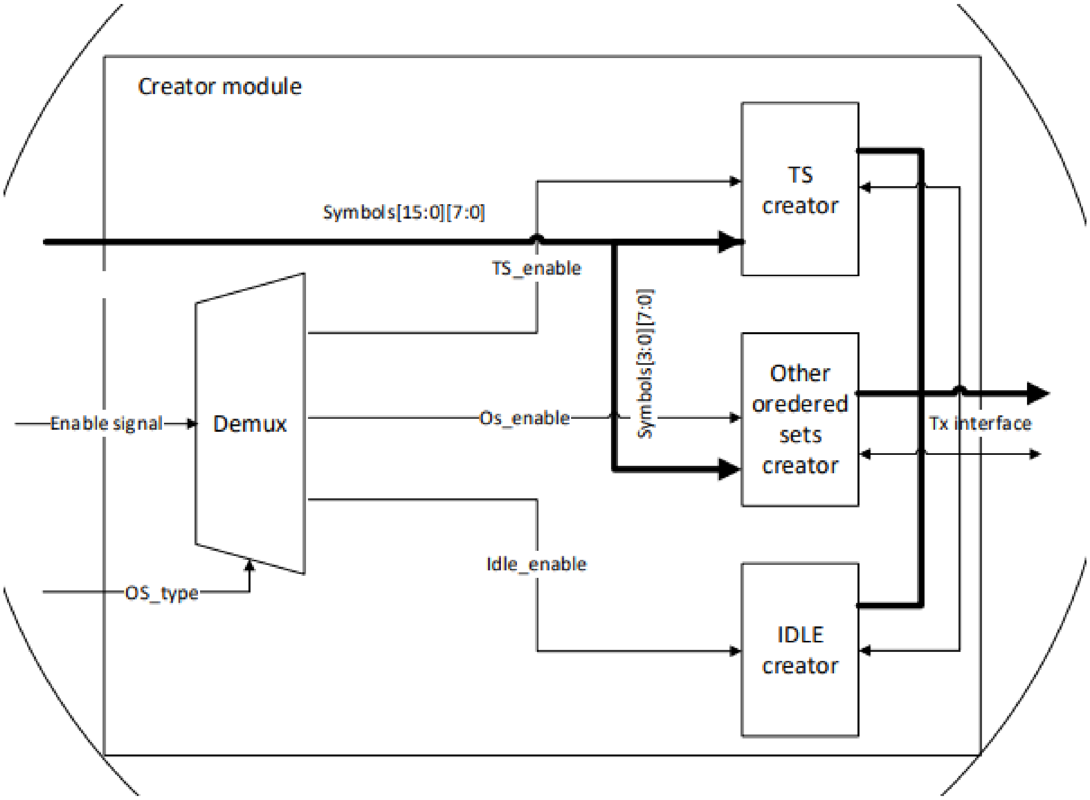

Figure 3: The Creator's internal modules.

4. Configuration

4.1. Configuration Sequence

The Creator operates normally directly after a reset sequence.

4.2. Latency

The Creator is sequential with a latency of 1 clock cycle.

4.3. Input Format and Constraints

The input symbols are defined as shown in the table below:

Symbol Name	Symbol Value
LIDLE	8'h00
COM	8'hBC
Link	8'h01
Lane	8'h01
PAD	8'hF7
FTS	8'h3C
NFTS	8'h00 - 8'hFF
SKP	8'h1C
DR_ID	8'h02
TS1_ID	8'h4A
TS2_ID	8'h45
IDLE	8'h7C

4.4. Driving Mechanism

All inputs should be driven simultaneously and will be captured at the positive edge of the clock. Inputs must not change until the complete OS and its repetitions have been sent.

5. Sequences and Waveforms

This section describes the possible sequences that can be input to the creator and the outputs of each sequence. Waveforms are also added when needed and possible. Note that if an input signal is not mentioned in the inputs table, then it is a don't care. The vertically downward direction in a table represents the progression of the value of the signal at a time interval of one clock cycle. The "-" symbol in a table indicates that the value of the signal does not change.

5.1. Reset Sequence (RST)

This sequence represents the reset procedure for the module.

Inputs:

i_resetn
1'b0

Outputs:

o_OScreator_Data	o_OScreator_valid	o_OScreator_Ack
{LIDLE, LIDLE)	1'b0	1'b0

Waveform(s):

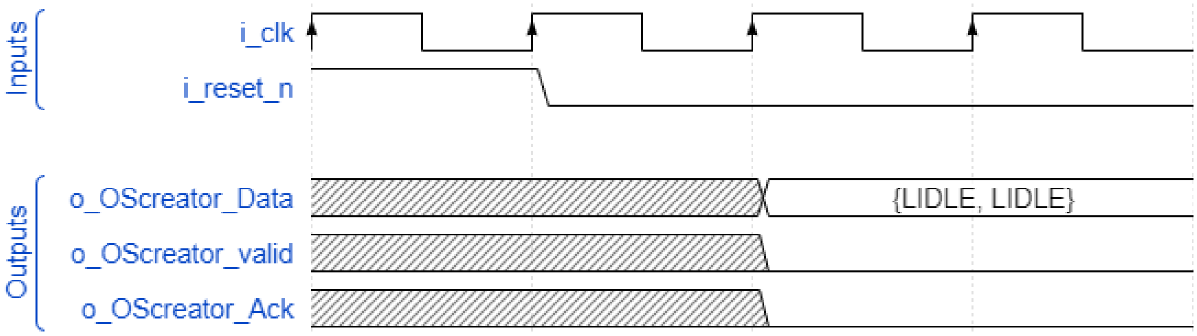

Figure 4: Reset Sequence.

5.2. Single TS1 Sequence (STS1)

This sequence represents sending a single TS1 OS.

Inputs:

i_resetn	i_enable	i_Tx_OSbufferFull	i_LTSSM_stateChange
1′b1	1′b1	1′b0	1′b0
i_reset_TS_count	i_OSreqNum	i_OScreatorTypes	i_OScreatorSymbol0
1′b0	11′d1	2′b01	COM
i_OScreatorSymbol1	i_OScreatorSymbol2	i_OScreatorSymbol3	i_OScreatorSymbol4
PAD or Link	PAD or Lane	NFTS	DR_ID
i_OScreatorSymbol5	i_OScreatorSymbol6 … i_OScreatorSymbol15	-	-
LIDLE	TS1_ID	-	-

Outputs:

o_OScreator_Data	o_OScreator_valid	o_OScreator_Ack
{COM, PAD} or {COM, Link}	1’b1	1’b0
{PAD, NFTS} or {Lane, NFTS}	1’b1	1’b0
{DR_ID, LIDLE}	1’b1	1’b0
{TS1_ID, TS1_ID}	1’b1	1’b0
{TS1_ID, TS1_ID}	1’b1	1’b0
{TS1_ID, TS1_ID}	1’b1	1’b0
{TS1_ID, TS1_ID}	1’b1	1’b0
{TS1_ID, TS1_ID}	1’b1	1’b0
{LIDLE, LIDLE}	1’b0	1’b1

Waveform(s):

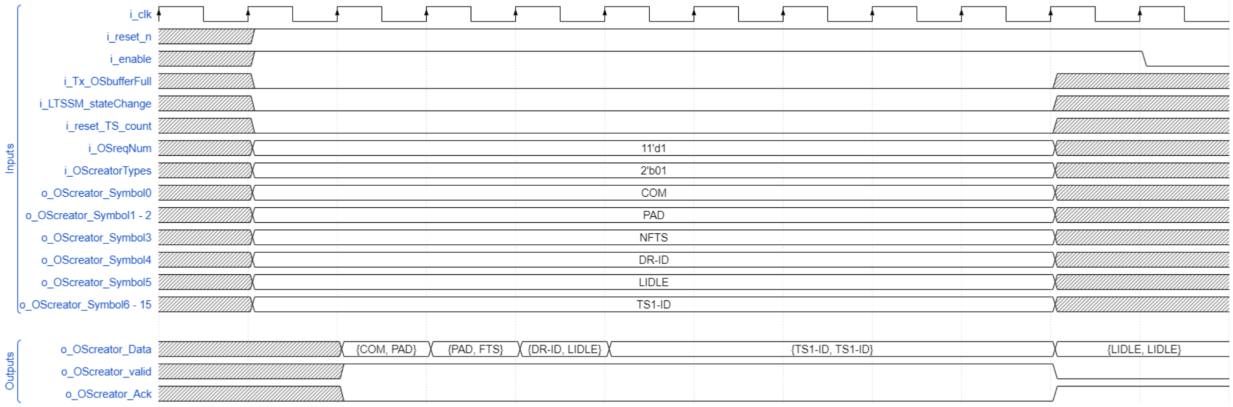

Figure 5: Single TS1 OS Sequence.

5.3. Multiple TS1 Sequence (MTS1)

This sequence is an example of multiple repetitions of TS1 OS. In this specific example, two TS1 OS are sent. Higher repetitions can follow the same pattern.

Inputs:

i_resetn	i_enable	i_Tx_OSbufferFull	i_LTSSM_stateChange
1'b1	1'b1	1'b0	1'b0
i_reset_TS_count	i_OSreqNum	i_OScreatorTypes	i_OScreatorSymbol0
1'b0	11'd2	2'b01	COM
i_OScreatorSymbol1	i_OScreatorSymbol2	i_OScreatorSymbol3	i_OScreatorSymbol4
PAD or Link	PAD or Lane	NFTS	DR_ID
i_OScreatorSymbol5	i_OScreatorSymbol6 … i_OScreatorSymbol15	-	-
LIDLE	TS1_ID	-	-

Outputs:

o_OScreator_Data	o_OScreator_valid	o_OScreator_Ack
{COM, PAD} or {COM, Link}	1’b1	1’b0
{PAD, NFTS} or {Lane, NFTS}	1’b1	1’b0
{DR_ID, LIDLE}	1’b1	1’b0
{TS1_ID, TS1_ID}	1’b1	1’b0
{TS1_ID, TS1_ID}	1’b1	1’b0
{TS1_ID, TS1_ID}	1’b1	1’b0
{TS1_ID, TS1_ID}	1’b1	1’b0
{TS1_ID, TS1_ID}	1’b1	1’b0
{COM, PAD} or {COM, Link}	1’b1	1’b0
{PAD, FTS} or {Lane, FTS}	1’b1	1’b0
{DR_ID, LIDLE}	1’b1	1’b0
{TS1_ID, TS1_ID}	1’b1	1’b0
{TS1_ID, TS1_ID}	1’b1	1’b0
{TS1_ID, TS1_ID}	1’b1	1’b0
{TS1_ID, TS1_ID}	1’b1	1’b0
{TS1_ID, TS1_ID}	1’b1	1’b0
{LIDLE, LIDLE}	1’b0	1’b1

5.4. Single TS2 Sequence (STS2)

This sequence represents sending a single TS2 OS. This is identical to STS1 except for the TS1_ID symbol being replaced by TS2_ID symbol.

5.5. Multiple TS2 Sequence (MTS2)

This sequence is an example of multiple repetitions of TS2 OS. This is identical to MTS1 except for the TS1_ID symbol being replaced by TS2_ID symbol.

5.6. Single SKP OS Sequence (SSKP)

This sequence represents sending a single SKP OS.

Inputs:

i_resetn	i_enable	i_Tx_OSbufferFull	i_LTSSM_stateChange
1'b1	1'b1	1'b0	1'b0
i_reset_TS_count	i_OSreqNum	i_OScreatorTypes	i_OScreatorSymbol0
1'b0	11'd1	2'b00	COM
i_OScreatorSymbol1	i_OScreatorSymbol2	i_OScreatorSymbol3	
SKP	SKP	SKP	

Outputs:

o_OScreator_Data	o_OScreator_valid	o_OScreator_Ack
{COM, SKP}	1'b1	1'b0
{SKP, SKP}	1'b1	1'b0
{LIDLE, LIDLE}	1'b0	1'b1

Waveform(s):

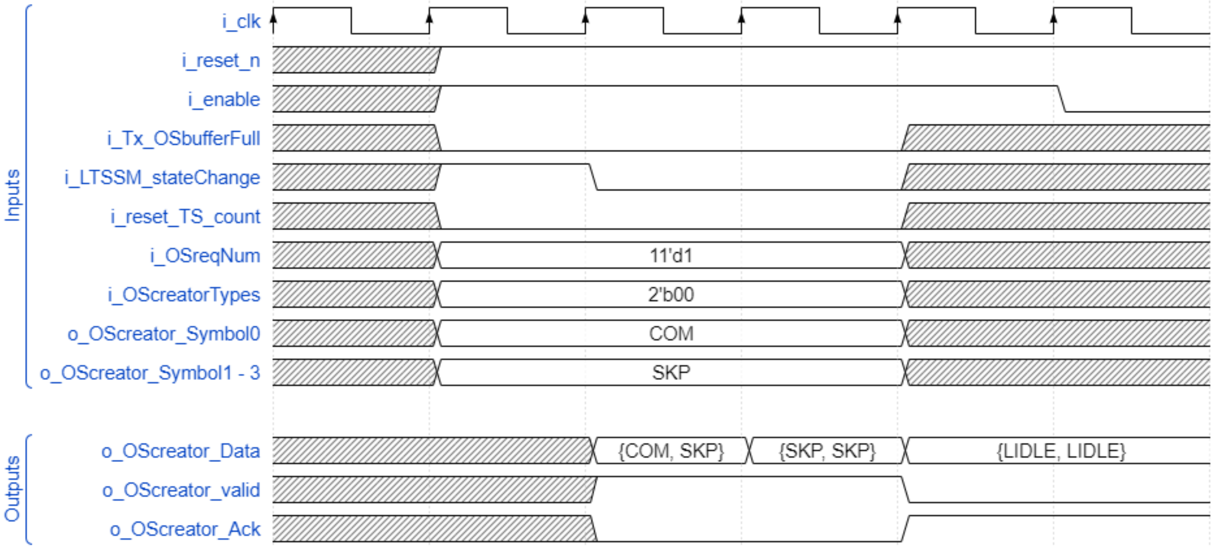

Figure 6: Single SKP OS Sequence.

5.7. Multiple SKP OS Sequence (MSKP)

This sequence is an example of multiple repetitions of SKP OS. In this specific example, two SKP OS are sent. Higher repetitions can follow the same pattern.

Inputs:

i_resetn	i_enable	i_Tx_OSbufferFull	i_LTSSM_stateChange
1'b1	1'b1	1'b0	1'b1
i_reset_TS_count	i_OSreqNum	i_OScreatorTypes	i_OScreatorSymbol0
1'b0	11'd2	2'b00	COM
i_OScreatorSymbol1	i_OScreatorSymbol2	i_OScreatorSymbol3	-
SKP	SKP	SKP	-

Outputs:

o_OScreator_Data	o_OScreator_valid	o_OScreator_Ack
{COM, SKP}	1’b1	1’b0
{SKP, SKP}	1’b1	1’b0
{COM, SKP}	1’b1	1’b0
{SKP, SKP}	1’b1	1’b0
{LIDLE, LIDLE}	1’b0	1’b1

5.8. Single FTS OS Sequence (SFTS)

This sequence represents sending a single FTS OS. This is identical to SSKP except for the SKP symbol being replaced by FTS symbol.

5.9. Multiple FTS OS Sequence (MFTS)

This sequence is an example of multiple repetitions of FTS OS. This is identical to MSKP except for the SKP symbol being replaced by FTS symbol.

5.10. Single LIDLE OS Sequence (SLIDLE)

This sequence represents sending a single LIDLE OS.

Inputs:

i_resetn	i_enable	i_Tx_OSbufferFull	i_LTSSM_stateChange
1’b1	1’b1	1’b0	1’b0
i_reset_TS_count	i_OSreqNum	i_OScreatorTypes	-
1’b0	11’d1	2’b10	-

Outputs:

o_OScreator_Data	o_OScreator_valid	o_OScreator_Ack
{LIDLE, LIDLE}	1’b1	1’b0
{LIDLE, LIDLE}	1’b0	1’b1

Waveform(s):

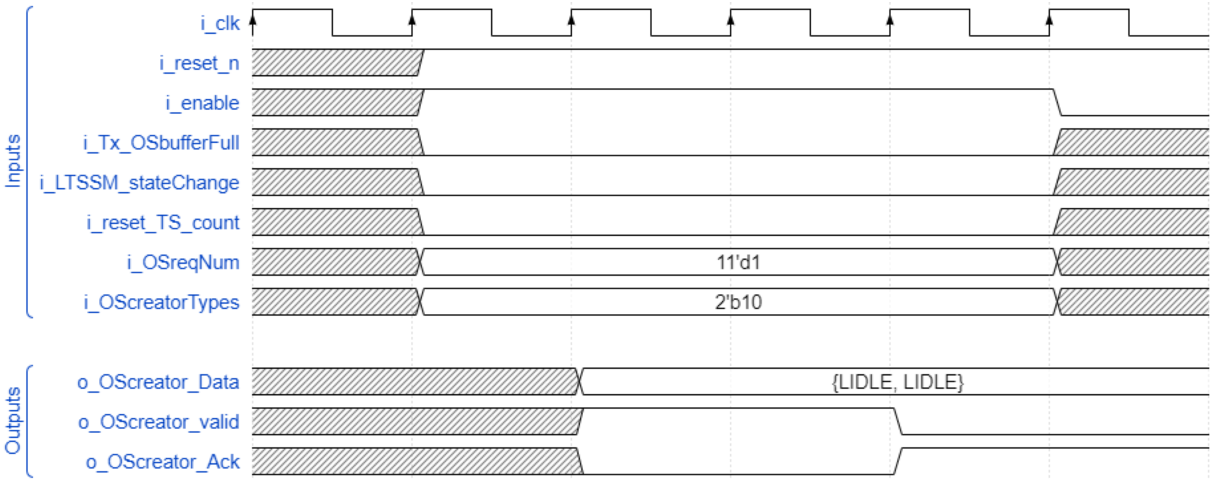

Figure 7: Single LIDLE OS Sequence.

5.11. Multiple LIDLE OS Sequence (MLIDLE)

This sequence is an example of multiple repetitions of LIDLE OS. In this specific example, two LIDLE OS are sent. Higher repetitions can follow the same pattern.

Inputs:

i_resetn	i_enable	i_Tx_OSbufferFull	i_LTSSM_stateChange
1’b1	1’b1	1’b0	1’b1
i_reset_TS_count	i_OSreqNum	i_OScreatorTypes	-
1’b0	11’d2	2’b10	-

Outputs:

o_OScreator_Data	o_OScreator_valid	o_OScreator_Ack
{LIDLE, LIDLE}	1’b1	1’b0
{LIDLE, LIDLE}	1’b1	1’b0
{LIDLE, LIDLE}	1’b0	1’b1