Escape Decoder

Induction Training

Version 1.0

Block Owner

Si-Vision

Authors

Amira Khaled

This Page is left Blank Intentionally

1 Table of Contents

2 Revision History

Version	Date	Author(s)	Revision Notes	Owner Approval
1.0				

3 Overview

The Esc_Decoder module is responsible for detecting and decoding escape sequences transmitted on a C-PHY-based interface. It identifies specific command patterns (such as ULPS, Reset Trigger, and LPDT start), extracts data during low-power data transmission, and generates appropriate status and error signals based on the line state.

4 Operation and Description

4.1 Digital Interface

4.1.1 Parameters Names

Parameter Name	Default	Description
None	-	-

4.1.2 Ports Names

Port Name	Port Width	Port Type	Description
RxClkEsc	1	Input	Recovered low-power clock
RST	1	Input	Active-low reset
EscDecoderEn	1	Input	Escape Decoder Enable
A	1	Input	Input line state from C-PHY
B	1	Input	Input line state from C-PHY
C	1	Input	Input line state from C-PHY
RequestDetection	1	Input	Enable control sequence check
RxLpdtEsc	1	Output	Indicates LPDT ESC command detected
RxUlpsEsc	1	Output	Indicates ULPS ESC command detected
RxTriggerEsc	4	Output	Indicates Trigger ESC command detected
EscBit	1	Output	Decoded data bit during LPDT
ErrEsc	1	Output	Invalid ESC command error
ErrSyncEsc	1	Output	Incorrectly received bit count
ErrControl	1	Output	Control sequence line state violation
LpFsmStop	1	Output	Indicates LP FSM should stop on valid stop sequence

4.1.3 CDC Table

CDC signal	Source Domain	Destination Domain	Synchronization method
EscDecoderEn	RxClkFsm	RxClkEsc	2-stage Flip Flop synchronizer
RequestDetection	RxClkFsm	RxClkEsc	2-stage Flip Flop synchronizer

4.2 Functional Description

The Decoder uses a Finite State Machine (FSM) with the following states:

- idle: Waits for EscDecoderEn to become high.
- CommandRead: Shifts in 8 bits from input line A to form a command.
- LpdtData: Streams data bits during LPDT mode (low power data transmission).

When the system has RST = 0, The decoder will be in the idle state and all output signals will be grounded,

After the RST = 1, Escape Decoder starts to work with these possible conditions:

1- If EscDecoderEn = 0 and RequestDetection = 1, the Decoder will stay in the idle state and check the control sequence on the 3-line states (A, B and C) and detect if the 3 lines become 111 instead of 000 (bridge state after any control state on the lines) then ErrControl = 1 to indicate that there is violation on the control signals sequence.

2- If EscDecoderEn = 1 and RequestDetection = 0, the Decoder will go to the CommandRead state and start reading the first 8 bits on the lines (A, B and C) with the RXEscClk edge and store these 8 bits to compare them with the stored commands:

   a. If the command decoded = Low Power Data Transmission command then assert RxLpdtEsc to indicate that we will detect the following bits on the line states as a data and go to the LpdtData state.
      In the LpdtData state, EscBit will take the sampled value of A with each positive edge of the RxClkEsc and increament an internal bit counter.
      When stop condition is detected on the lines (A=1, B=1 and C=1):
      - LpFsmStop = 1 to indicate that the Escape mode has been ended
      - Check for the bit counter to check if the received data was multiple of 8 bits, if not then ErrSyncEsc = 1 to indicate that the received data is not correct.

   b. If the command decoded = Ultra Low Power command then assert RxUlpsEsc to indicate that we now in the ultra-low power mode.

   c. If the command decoded = Reset Trigger command then assert RxTriggerEsc to indicate that Reset trigger has been received

   d. If unknown command is received then assert ErrEsc to indicate that invalid command has been received.

   Note that: in any case if the EscDecoderEn = 0, decoder will go to the idle state waiting for the enable and then start reading the command from the beginning.

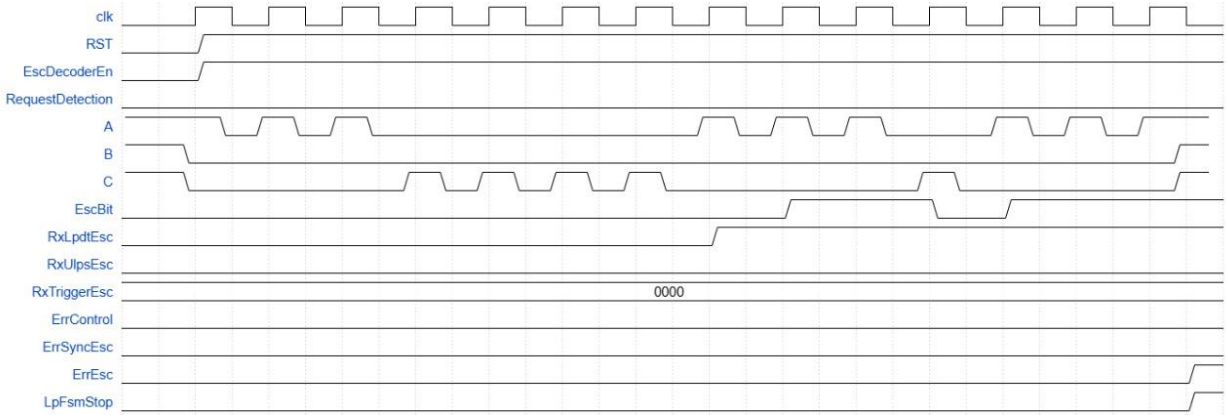
3- If EscDecoderEn = 0 and RequestDetection = 0, the Decoder will be in the idle state and will not detect the control sequence on the lines.

4- If EscDecoderEn = 1 and RequestDetection = 1, then the stop state that is needed to detect the end of the Escape mode will be detected as LpfsmStop and ErrCtrl at the same time so this case should be avoided from the main FSM of the Slave.

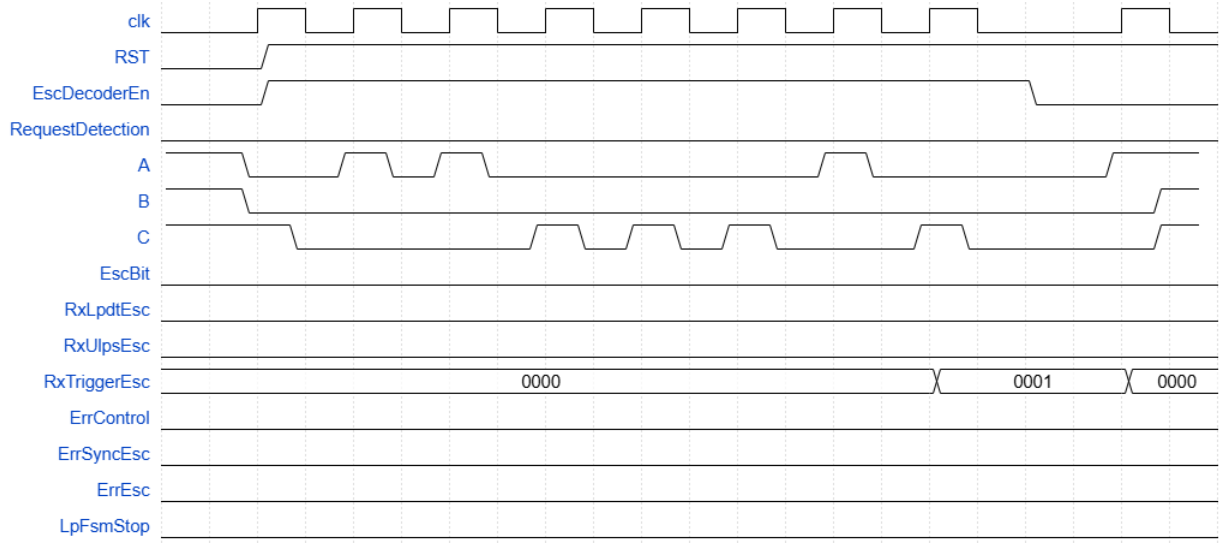
Note that: as the Clock used to sample the data is recovered from the same data (The clock edge and data change occur at the same time) then we need to delay the input clock to that block to avoid race condition.

4.3 Timing Diagram

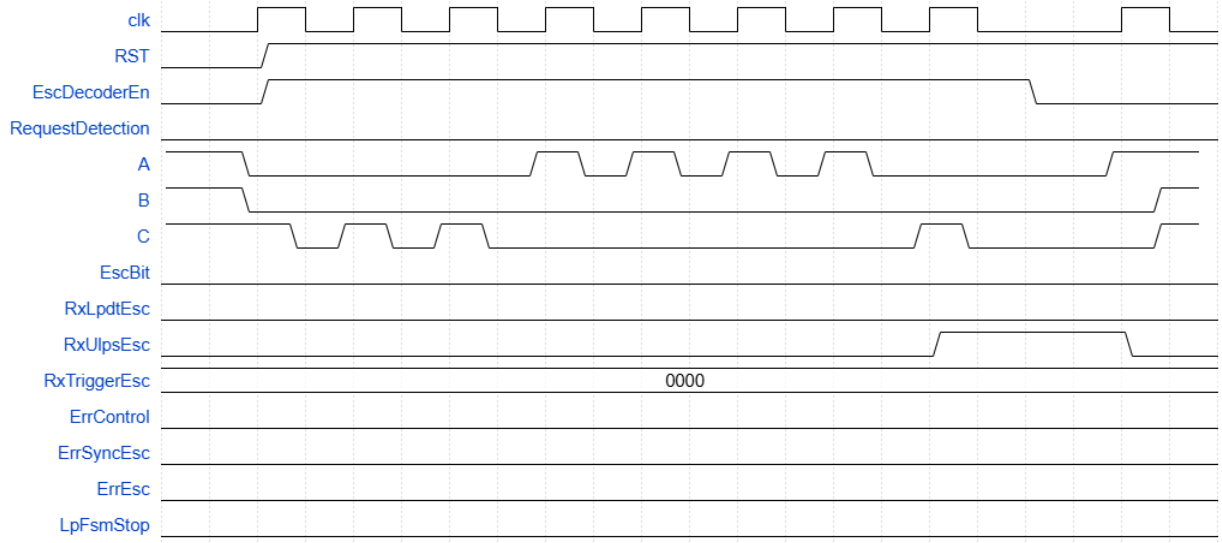
- Low Power Data Mode with wrong number of bits



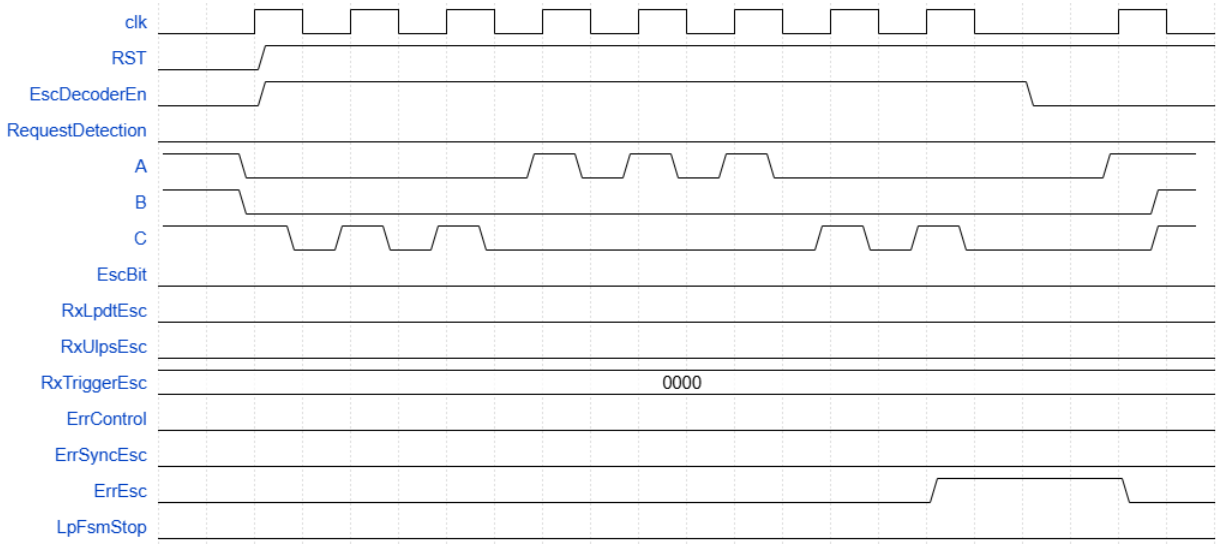
- Reset Trigger



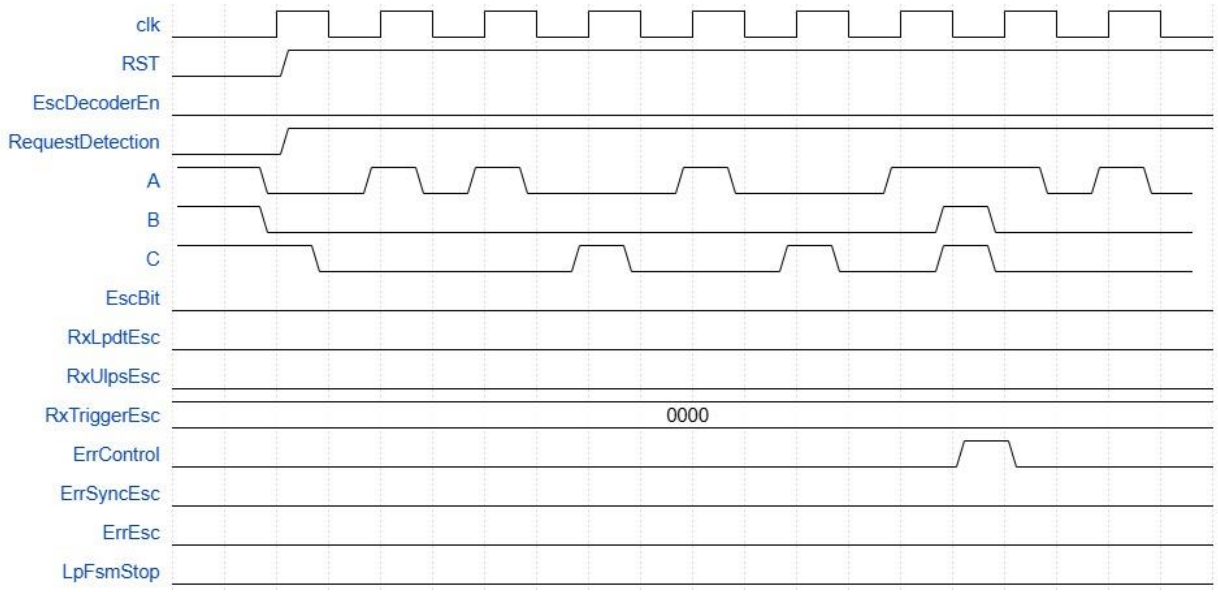
- Ultra Low Power mode



- Invalid command



- Control sequence detection



4.4 Verification Requirements

- Low Power Data Mode:

  Sends the low power data command and verifies correct detection and data reception.

- Ultra Low Power Mode:

  Sends the ultra-low power command and verifies proper detection of ULPS mode.

- Reset Trigger:

  Sends reset trigger command and checks for trigger detection output.

- Invalid Command:

  Sends an invalid command and expects the ErrEsc flag to be asserted.

- Low Power Data with Wrong Bit Count:

  Sends low power data command followed by incorrect data bit count and verifies error detection.

- Control Sequence Error:

  Simulates wrong control sequences and expects the ErrControl flag assertion.