State	Condition for transition	Next state
	reset is active or FIFO is empty:	
IDLE	Rstb = 0 fifo_empty =1	IDLE
	Reset is inavtive and FIFO not empty :	
	Rstb = 1 && fifo_empty = 0	SOP_S0
SOP_S0	FIFO is empty : fifo_empty =1	SOP_S0
	FIFO is not empty and Start of Packet is High:	
	fifo_empty = 0 && sop=1 &&	
	- Current message length < 6	S1
	- Current message length > 8	S2
	 Current message length == 7 	S4
	- Current message length == 6	S6
S1	FIFO is empty : fifo_empty =1	S1
	always	S1_P
S1_P	FIFO is empty : fifo_empty =1	S1
	FIFO is not empty: fifo_empty =0 &&	
	 Current message length < 6 	S1
	 Current message length > 8 	S2
	- Current message length == 7	S4
	Current message length == 6	S6
S2	FIFO is empty : fifo_empty =1	S2
	FIFO is not empty: fifo_empty =0 &&	
	 Current message length < 6 	S1
	- Current message length > 8	S2
	- Current message length == 7	S4
	- Current message length == 0	S5
	- Current message length == 6	S6
S3	FIFO is empty : fifo empty =1	S3
	FIFO is not empty: fifo_empty =0 &&	
	 Current message length < 6 	S1
	- Current message length > 8	S2
	- Current message length == 7	S4
	- Current message length == 6	S6
S4	FIFO is empty : fifo empty =1	S4
	FIFO is not empty: fifo empty =0	S3
S5	FIFO is empty : fifo empty =1	S5
	FIFO is not empty: fifo empty =0 &&	
	- Current message length < 6	S1
	- Current message length > 8	S2
	- Current message length == 7	S4
	- Current message length == 6	S6
S6	FIFO is empty : fifo empty =1	S6
	FIFO is not empty: fifo empty =0	S2
	current message length = current message length	
	length of data in current input	
Outputs	Description	'

upd_mc upd_ml[1:0]

length of data in current input

Description

update message count

update message length

upd ml[1] -- message len[1],

upd_ml[0] -- message len[1],

upd_ml[0] -- message length

Current message length

Current message length -- length of data payload in

current input

start and end index of message count field, updated

only once

start and end index of message length field, updated

for every new message

start and end index of data payload field, updated

every clock

length of data payload in current input

FIFO read enable upd_len

mc_st, mc_end

ml_st, ml_end

p_st, p_end cur_len read_en

Outputs

State	Outputs mc_st = 7 mc_end = 6	Description message count field always present in data_bytes[7:6], this field is only valid when upd_mc =1	MC0 MC1		e count [
IDLE	upd_mc = 0 upd_m[1:0] = 0 ml_st = 0 oml_en = 0 upd_len = 0 p_st = 0 p_end = 0 cur_len = 0 read_en = 0 msg_end = 0	Clear all outputs	ML0 ML1 P	messag messag data pay	e count [: e length e length yload	[0]	D2	D2	D1	DO
S0	If (sop == 1): upd_mc = 1 upd_ml[1:0] = 2'b11 ml_st = 5 ml_end = 4 upd_len = 1 p_st = 3 p_end = 0 cur_len = 4 read_en = 1 msg_end = 0	Set outputs as below if sop is HIGH, else clear all Update message count update message length, both bytes For the very first data input, message length field always present in data_bytes[5:4] update message length as data payload will be available For the very first data input, message data field always present in data_bytes[3:0] For the very first data input, current data payload length is 4 since FIFO is not empty, start reading from it	MC0 mc_st	MC1 mc_end	ML0	ML1 ml_end	P p_st	P	P	P p_end
S1	upd_mc = 0 upd_m[1:0] = 2'b11 ml_st = 8 - cur_msg_len · 1 ml_end = 8 - cur_msg_len · 2 upd_len = 1 p_st = 7 p_end = 8 - cur_msg_len cur_len = 8 - (cur_msg_len + 2) read_en = 1 msg_end = 1	No message count present in the middle of the packer update message length as data payload will be available Update new message length field indices based on where the current message data payload ends Update current message data payload ends Current data payload starts from index 7 Next message data payload length Read next data input from FIFO end of current message	D7 P p_st FSM will	D6 P/ML p_end I only pro	P/ML	D4 P/ML or current	D3 P/ML message	D2 P/ML	D1 P/ML	D0 P
S1_P	upd_mc = 0 upd_m[[1:0] = 2'b00 ml_st = ml_st ml_end = ml_end upd_len = 1 p_st = ml_end - 1 p_end = 0 cur_len = cur_len read_en = 1 msg_end = 0	Current message was previous message when in S1 state, message length field was already extracted in S1 state Do not change ml_st and ml_end value to save power Current data payload is available, so update current message length Data payload always starts one byte immediately after message length Current message length was previously updated in S1 state		P/ML	P/ML ml_st M will prod	P/ML ml_end cess ML a		P/ML	P/ML sage	P p_end
S2	upd_mc = 0 upd_ml[1:0] = 2'b00 ml_st = ml_st ml_end = ml_end upd_len = 1 p_st = 7 p_end = 0 cur_len = 8 read_en = 1 msg_end = (cur_msg_len == 0)	This is an all data payload state If not more data payload , indicate end of message	P p_st	D6 P	D5 P	D4 P	D3 P	D2 P	D1 P	P p_end
S3	upd_mc = 0 upd_ml[1:0] = 2'b01 ml_st = ml_st ml_end = 7 upd_len = 1 p_st = 6 p_end = 0 cur_len = 7 read_en = 1 msg_end = 0	Partial message length state, only MessageLength[1] is available MessageLength[1] is present in data_bytes[7], MessageLength[0] will have been extracted in state S4 in previous clock	ML1 ml_end	P p_st	D5 P	D4 P	D3	D2 P	D1 P	P p_end
S4	upd_mc = 0 upd_ml[1:0] = 2'b10 ml_st = 0 ml_end = ml_end upd_len = 1 p_st = 7 p_end = 1 cur_len = 7 read_en = 1 msg_end = 1	Next message length field is present in data_bytes[0], only MessageLength[1] is available in this data input, MessageLength[0] will be available in next clock Current data payload is available, so update current message length end of current message	P p_st	P	P	P	P	P	P p_end	ML0 ml_st
S5	upd_mc = 0 upd_ml[1:0] = 2'b11 ml_ st = 7 ml_ end = 6 upd_len = 1 p_st = 5 p_end = 0 cur_len = 6 read_en = 1 msg_end = 0	Start of new message length and data payload Next message length field is present in data_bytes[7:6]	ML0 ml_st	D6 ML1 ml_end	D5 P p_st	D4 P	D3 P	D2 P	D1 P	P P_end
S6	upd_mc = 0 upd_ml[1:0] = 2'b11 ml_st = 1 ml_end = 0 upd_len = 1 p_st = 7 p_end = 2 cur_len = 6 read_en = 1 msg_end = 1	Next message length field is present in data_bytes[7:6] end of current message	P p_st	D6 P	D5 P	D4 P	D3 P	P p_end	ML0 ml_st	ML1 ml_end