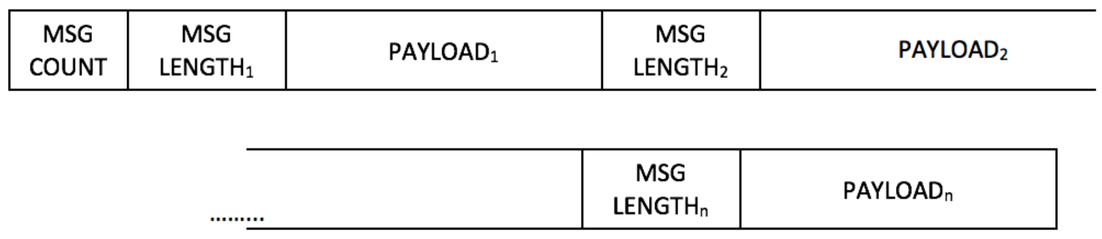
Message Extractor

The format of a single packet is given below. The incoming data stream will consist of multiple packets.

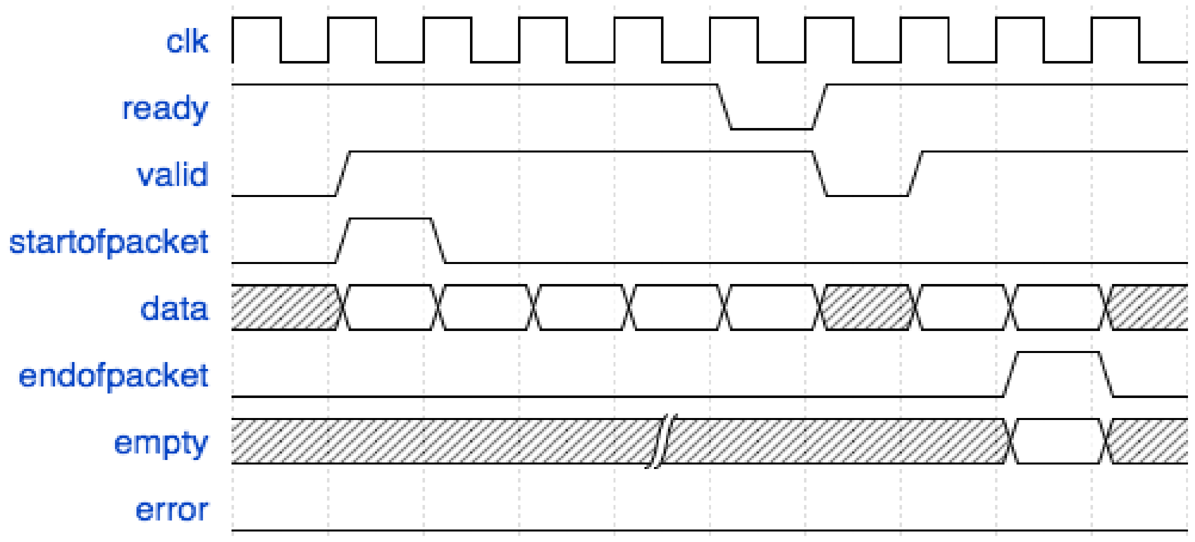


Field name	Length	Description
Message Count	2 bytes	Number of messages in the packet
Message Length	2 bytes	Length of the following message (excluding this field)
Payload	Variable	Message Payload data

The expected output of the block is the payload data of these messages.

Input setup

1. The input of the module is a 64-bit Avalon Streaming interface. The I/O signals are given below.



Signal Name	Direction	Width (bits)	Description
clk	Input	1	Clock
reset_n	Input	1	Active low reset
in_ready	Output	1	Indicates when the sink module (module being designed) is ready to accept data. Read Latency =1
in_valid	Input	1	High when in_data is valid, 0 otherwise
in_startofpacket	Input	1	High for the 1st clock cycle of the incoming packet, 0 otherwise
in_endofpacket	Input	1	High for the last clock cycle of the incoming packet, 0 otherwise
in_data	Input	64	Incoming packet data
in_empty	Input	3	Indicates the number of bytes that are empty during cycles that contain the end of a packet. Should only be qualified with incoming end of packet.

in_error	Input	1	A bit mask used to mark errors affecting the incoming data being transferred in the current cycle.

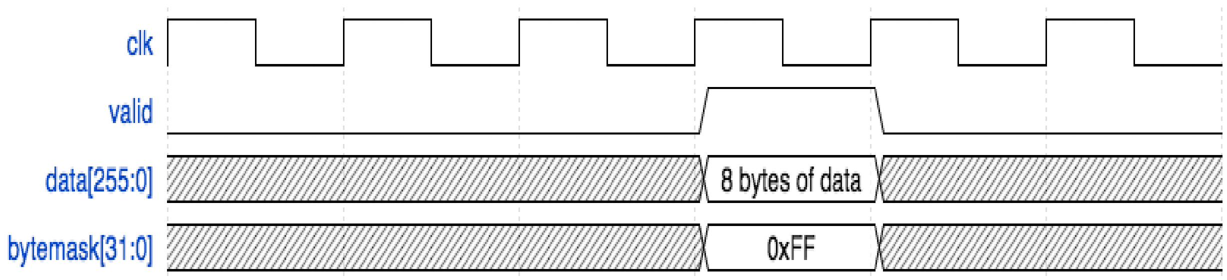
2. Assume that the minimum message length for any message is **8 bytes** and the maximum is **32 bytes**. The total size of each packet can be a maximum of **1,500 bytes**.
3. Assume **in_error is always 1'b0**.

Output setup

1. The output signals of the module are given below.

Signal Name	**Direction**	**Width (bits)**	**Description**
clk	Input	1	Clock
reset_n	Input	1	Active low reset
out_valid	Output	1	High when out_data is valid, 0 otherwise
out_data	Output	256	Outgoing message payload
out_bytemask	Output	32	Indicates the number of bytes valid in the payload.

For example, if the message length of a message reads 8 bytes, the expected output would be the 8 bytes of the payload in out_data bus with an out_bytemask of 32'hFF qualified by an out_valid.



<u>Example Packet</u>

Sample Input

in_data [63:0] (hex)	in_startof-packet	in_endof-packet	in_valid	in_empty	in_error
6262626108000800	1	0	1	X	0
68670c0063626262	0	0	1	X	0
6868686868686868	0	0	1	X	0
7070706f0a006968	0	0	1	X	0
0f00717070707070	0	0	1	X	0
7a7a7a7a7a7a7a79	0	0	1	X	0
007b7a7a7a7a7a7a	0	0	1	X	0
4d4d4d4d4d4d4c0e	0	0	1	X	0
004e4d4d4d4d4d4d	0	0	1	X	0
3838383838383711	0	0	1	X	0
3838383838383838	0	0	1	X	0
313131300b003938	0	0	1	X	0
0032313131313131	0	0	1	X	0
5a5a5a5a5a5a5909	0	0	1	X	0
XXXXXXXXXXXX5b5a	0	1	1	6	0

Note: in_valid can be de-asserted at any time after data starts streaming in.

Sample Output

out_data (hex)	out_bytemask (binary)	out_valid
6362626262626261	32’b00000000_00000000_00000000_11111111	1
696868686868686868686867	32’b00000000_00000000_00001111_11111111	1
7170707070707070706f	32’b00000000_00000000_00000011_11111111	1
7b7a7a7a7a7a7a7a7a7a7a7a7a7a79	32’b00000000_00000000_01111111_11111111	1
4e4d4d4d4d4d4d4d4d4d4d4d4d4c	32’b00000000_00000000_00111111_11111111	1
3938383838383838383838383838383837	32’b00000000_00000001_11111111_11111111	1
3231313131313131313130	32’b00000000_00000000_00000111_11111111	1
5b5a5a5a5a5a5a5a59	32’b00000000_00000000_00000001_11111111	1