Entity: tx_fsm_pt

- **File**: tx_fsm_pt.vhd

Diagram

Ports

Port name	Direction	Type	Description
clk	in	std_logic	100 MHz Clock
resetn	in	std_logic	Active low reset
tx_active	out	std_logic	Tell the PHY when a TX transaction starts
byte_valid	out	std_logic	Tell the PISO SR when a byte is valid on the data bus
byte_ready	in	std_logic	PISO SR is ready to receive a new byte
packet_ready	out	std_logic	Packet is ready to be received from AXI-S
packet_valid	in	std_logic	Packet is valid in RAM and ready to be transmitted
addr	out	std_logic_vector(10 downto 0)	RAM Data Address
data	in	std_logic_vector(7 downto 0)	RAM Data

Signals

Name	Type	Description
state	state_t	FSM State Variables
next_state	state_t	FSM State Variables
mem_state	mem_state_t	Memory Read FSM State Variables
mem_next_state	mem_state_t	Memory Read FSM State Variables
r_mem_read_req	std_logic	Memory read request
r_mem_read_valid	std_logic	Memory read is valid this cycle
r_packet_length	unsigned(10 downto 0)	Packet Length Counter
cnt_addr	unsigned(10 downto 0)	Address Counter
cnt_IFG	unsigned(10 downto 0)	IFG Counter
f_first_byte	std_logic	FSM Flag First Byte Output

Enums

state_t

Packet Transmission FSM states

Name	Description
IDLE	Awaiting new transmission
LOAD_LENGTH_UPPER	Load the upper part of packet length
LOAD_LENGTH_LOWER	Load the lower part of packet length
TX_LOAD	Load the next address's data into the shift register
TX_FIRST	Load the first byte (set tx active high)
TX_WAIT	Wait for this byte to finish shifting out
TX_LAST	Load the last byte (set tx active low at completion)
IFG	Inter-frame-gap between transmission and idle state.

mem_state_t

Memory Read FSM states

Name	Description
LADDR	Loading address.
VALID	Data is valid this cycle.

Processes

- unnamed: (clk, resetn)

  Description
  Sequential Logic Update & State Register Process

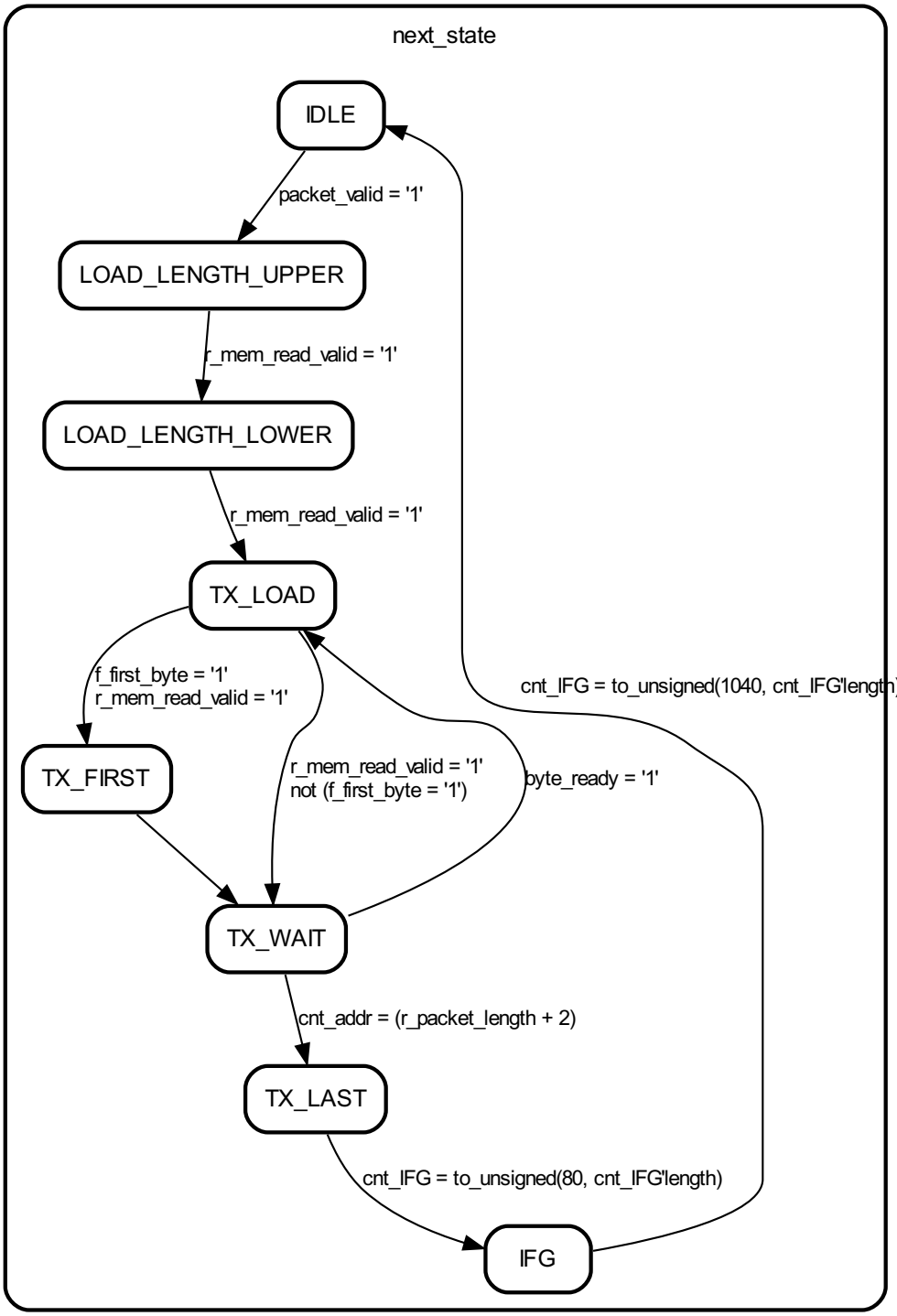
- p_main_fsm: (state, packet_valid, byte_ready, r_mem_read_valid, f_first_byte, cnt_addr, r_packet_length, cnt_IFG)

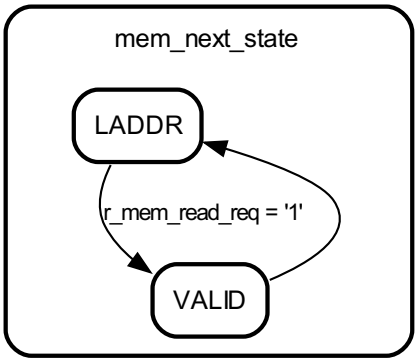
  Description
  Combinatorial logic for FSM state updates

- p_ram_fsm: (mem_state, r_mem_read_req)

State machines

- Combinatorial logic for FSM state updates


next_state
IDLE
packet_valid = '1'
LOAD_LENGTH_UPPER
r_mem_read_valid = '1'
LOAD_LENGTH_LOWER
r_mem_read_valid = '1'
TX_LOAD
f_first_byte = '1'
r_mem_read_valid = '1'
cnt_IFG = to_unsigned(1040, cnt_IFG'length)
TX_FIRST
r_mem_read_valid = '1'
not (f_first_byte = '1')
byte_ready = '1'
TX_WAIT
cnt_addr = (r_packet_length + 2)
TX_LAST
cnt_IFG = to_unsigned(80, cnt_IFG'length)
IFG


mem_next_state
LADDR
r_mem_read_req = '1'
VALID