OAM for 100Mb/s Long Reach

16 September 2024

Goals

- Leverage as much of Clause 149 OAM text and state machines as possible
  - Preserve state machines but redefine the meaning of a few variables
- Introduce a framing layer for symbol serialization and deserialization
- Balance being efficient without complicating implementation

Clause 149 OAM Frame

- 10-bit symbol sent per PCS RS-Frame
- 10-bit symbol sent per EEE Refresh
- OAM RS(16, 14, GF(2^{10}))
  - Correction Optional
  - Useful only when symbols sent during EEE refresh
  - RS encoder size on the order of a 20-bit CRC generator
- D8 used for framing the 16 symbols
- D9 set to 0
- Parity computed on 10-bit symbols

	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
Symbol 0	Reserved	0	Reserved	Reserved	Reserved	Reserved	PingRx	PingTx	SNR<1>	SNR<0>
Symbol 1	Reserved	1	Valid	Toggle	Ack	TogAck	Message_Number<3:0>			
Symbol 2	Reserved	1	Message<0><7:0>							
Symbol 3	Reserved	1	Message<1><7:0>							
Symbol 4	Reserved	1	Message<2><7:0>							
Symbol 5	Reserved	1	Message<3><7:0>							
Symbol 6	Reserved	1	Message<4><7:0>							
Symbol 7	Reserved	1	Message<5><7:0>							
Symbol 8	Reserved	1	Message<6><7:0>							
Symbol 9	Reserved	1	Message<7><7:0>							
Symbol 10	Reserved	1	Message<8><7:0>							
Symbol 11	Reserved	1	Message<9><7:0>							
Symbol 12	Reserved	1	Message<10><7:0>							
Symbol 13	Reserved	1	Message<11><7:0>							
Symbol 14	RS(16,14) parity									
Symbol 15	RS(16,14) parity									

Figure 149–22—OAM frame

Issue for Baselined PCS

- Each PCS RS-Frame carries only 1 bit and not one 10-bit symbol
- Need to frame serialized bit stream of 160 bits and not just 16 10-bit symbols
- Reserved bits are not used, so be more efficient and just send 144 bits per OAM frame. (Don't transmit D9 and some D8)
- Clause 149 OAM frame re-annotated below

	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
Symbol 0	0	0	Res7	Res6	Res5	Res4	PingRx	PingTx	SNR<1>	SNR<0>
Symbol 1	0	1	Valid	Toggle	Ack	TogAck	MsgNum<3>	MsgNum<2>	MsgNum<1>	MsgNum<0>
Symbol 2	0	1	M<0><7>	Msg<0><6>	Msg<0><5>	Msg<0><4>	Msg<0><3>	Msg<0><2>	Msg<0><1>	Msg<0><0>
Symbol 3	0	1	Msg<1><7>	Msg<1><6>	Msg<1><5>	Msg<1><4>	Msg<1><3>	Msg<1><2>	Msg<1><1>	Msg<1><0>
Symbol 4	0	1	Msg<2><7>	Msg<2><6>	Msg<2><5>	Msg<2><4>	Msg<2><3>	Msg<2><2>	Msg<2><1>	Msg<2><0>
Symbol 5	0	1	Msg<3><7>	Msg<3><6>	Msg<3><5>	Msg<3><4>	Msg<3><3>	Msg<3><2>	Msg<3><1>	Msg<3><0>
Symbol 6	0	1	Msg<4><7>	Msg<4><6>	Msg<4><5>	Msg<4><4>	Msg<4><3>	Msg<4><2>	Msg<4><1>	Msg<4><0>
Symbol 7	0	1	Msg<5><7>	Msg<5><6>	Msg<5><5>	Msg<5><4>	Msg<5><3>	Msg<5><2>	Msg<5><1>	Msg<5><0>
Symbol 8	0	1	Msg<6><7>	Msg<6><6>	Msg<6><5>	Msg<6><4>	Msg<6><3>	Msg<6><2>	Msg<6><1>	Msg<6><0>
Symbol 9	0	1	Msg<7><7>	Msg<7><6>	Msg<7><5>	Msg<7><4>	Msg<7><3>	Msg<7><2>	Msg<7><1>	Msg<7><0>
Symbol 10	0	1	Msg<8><7>	Msg<8><6>	Msg<8><5>	Msg<8><4>	Msg<8><3>	Msg<8><2>	Msg<8><1>	Msg<8><0>
Symbol 11	0	1	Msg<9><7>	Msg<9><6>	Msg<9><5>	Msg<9><4>	Msg<9><3>	Msg<9><2>	Msg<9><1>	Msg<9><0>
Symbol 12	0	1	Msg<10><7>	Msg<10><6>	Msg<10><5>	Msg<10><4>	Msg<10><3>	Msg<10><2>	Msg<10><1>	Msg<10><0>
Symbol 13	0	1	Msg<11><7>	Msg<11><6>	Msg<11><5>	Msg<11><4>	Msg<11><3>	Msg<11><2>	Msg<11><1>	Msg<11><0>
Symbol 14	P<1><9>	P<1><8>	P<1><7>	P<1><6>	P<1><5>	P<1><4>	P<1><3>	P<1><2>	P<1><1>	P<1><0>
Symbol 15	P<0><9>	P<0><8>	P<0><7>	P<0><6>	P<0><5>	P<0><4>	P<0><3>	P<0><2>	P<0><1>	P<0><0>

The 144 Serialized Bits Transmitted

- Compact 160 bits to 144 bits
- Serialize only the lower 9 bits of each symbol for the first 12 symbols
- Serialize only the lower 8 bits for the next 2 symbols
- Serialize the two 10-bit parity symbols
- Transmit order shown below

	1st bit transmitted								9th bit transmitted
1st row transmitted	0	Res7	Res6	Res5	Res4	PingRx	PingTx	SNR<1>	SNR<0>
	1	Valid	Toggle	Ack	TogAck	MsgNum<3>	MsgNum<2>	MsgNum<1>	MsgNum<0>
	1	M<0><7>	Msg<0><6>	Msg<0><5>	Msg<0><4>	Msg<0><3>	Msg<0><2>	Msg<0><1>	Msg<0><0>
	1	Msg<1><7>	Msg<1><6>	Msg<1><5>	Msg<1><4>	Msg<1><3>	Msg<1><2>	Msg<1><1>	Msg<1><0>
	1	Msg<2><7>	Msg<2><6>	Msg<2><5>	Msg<2><4>	Msg<2><3>	Msg<2><2>	Msg<2><1>	Msg<2><0>
	1	Msg<3><7>	Msg<3><6>	Msg<3><5>	Msg<3><4>	Msg<3><3>	Msg<3><2>	Msg<3><1>	Msg<3><0>
	1	Msg<4><7>	Msg<4><6>	Msg<4><5>	Msg<4><4>	Msg<4><3>	Msg<4><2>	Msg<4><1>	Msg<4><0>
	1	Msg<5><7>	Msg<5><6>	Msg<5><5>	Msg<5><4>	Msg<5><3>	Msg<5><2>	Msg<5><1>	Msg<5><0>
	1	Msg<6><7>	Msg<6><6>	Msg<6><5>	Msg<6><4>	Msg<6><3>	Msg<6><2>	Msg<6><1>	Msg<6><0>
	1	Msg<7><7>	Msg<7><6>	Msg<7><5>	Msg<7><4>	Msg<7><3>	Msg<7><2>	Msg<7><1>	Msg<7><0>
	1	Msg<8><7>	Msg<8><6>	Msg<8><5>	Msg<8><4>	Msg<8><3>	Msg<8><2>	Msg<8><1>	Msg<8><0>
	1	Msg<9><7>	Msg<9><6>	Msg<9><5>	Msg<9><4>	Msg<9><3>	Msg<9><2>	Msg<9><1>	Msg<9><0>
	Msg<10><7>	Msg<10><6>	Msg<10><5>	Msg<10><4>	Msg<10><3>	Msg<10><2>	Msg<10><1>	Msg<10><0>	Msg<11><7>
	Msg<11><6>	Msg<11><5>	Msg<11><4>	Msg<11><3>	Msg<11><2>	Msg<11><1>	Msg<11><0>	P<1><9>	P<1><8>
	P<1><7>	P<1><6>	P<1><5>	P<1><4>	P<1><3>	P<1><2>	P<1><1>	P<1><0>	P<0><9>
16th row transmitter	P<0><8>	P<0><7>	P<0><6>	P<0><5>	P<0><4>	P<0><3>	P<0><2>	P<0><1>	P<0><0>

Finding the OAM Frame Boundary

- Take every 9th bit and search for 0111_1111_1111_xxxx pattern.
- Once pattern match check the parity. If good then assume locked
- Otherwise offset 1 bit and try searching for the pattern again
- Once locked – deserialize and expand to present 10-bit symbols

	1st bit transmitted								9th bit transmitted
1st row transmitted	0	Res7	Res6	Res5	Res4	PingRx	PingTx	SNR<1>	SNR<0>
	1	Valid	Toggle	Ack	TogAck	MsgNum<3>	MsgNum<2>	MsgNum<1>	MsgNum<0>
	1	M<0><7>	Msg<0><6>	Msg<0><5>	Msg<0><4>	Msg<0><3>	Msg<0><2>	Msg<0><1>	Msg<0><0>
	1	Msg<1><7>	Msg<1><6>	Msg<1><5>	Msg<1><4>	Msg<1><3>	Msg<1><2>	Msg<1><1>	Msg<1><0>
	1	Msg<2><7>	Msg<2><6>	Msg<2><5>	Msg<2><4>	Msg<2><3>	Msg<2><2>	Msg<2><1>	Msg<2><0>
	1	Msg<3><7>	Msg<3><6>	Msg<3><5>	Msg<3><4>	Msg<3><3>	Msg<3><2>	Msg<3><1>	Msg<3><0>
	1	Msg<4><7>	Msg<4><6>	Msg<4><5>	Msg<4><4>	Msg<4><3>	Msg<4><2>	Msg<4><1>	Msg<4><0>
	1	Msg<5><7>	Msg<5><6>	Msg<5><5>	Msg<5><4>	Msg<5><3>	Msg<5><2>	Msg<5><1>	Msg<5><0>
	1	Msg<6><7>	Msg<6><6>	Msg<6><5>	Msg<6><4>	Msg<6><3>	Msg<6><2>	Msg<6><1>	Msg<6><0>
	1	Msg<7><7>	Msg<7><6>	Msg<7><5>	Msg<7><4>	Msg<7><3>	Msg<7><2>	Msg<7><1>	Msg<7><0>
	1	Msg<8><7>	Msg<8><6>	Msg<8><5>	Msg<8><4>	Msg<8><3>	Msg<8><2>	Msg<8><1>	Msg<8><0>
	1	Msg<9><7>	Msg<9><6>	Msg<9><5>	Msg<9><4>	Msg<9><3>	Msg<9><2>	Msg<9><1>	Msg<9><0>
	Msg<10><7>	Msg<10><6>	Msg<10><5>	Msg<10><4>	Msg<10><3>	Msg<10><2>	Msg<10><1>	Msg<10><0>	Msg<11><7>
	Msg<11><6>	Msg<11><5>	Msg<11><4>	Msg<11><3>	Msg<11><2>	Msg<11><1>	Msg<11><0>	P<1><9>	P<1><8>
	P<1><7>	P<1><6>	P<1><5>	P<1><4>	P<1><3>	P<1><2>	P<1><1>	P<1><0>	P<0><9>
16th row transmitter	P<0><8>	P<0><7>	P<0><6>	P<0><5>	P<0><4>	P<0><3>	P<0><2>	P<0><1>	P<0><0>

Clause 149 tx_boundary Variable Redefinition

- tx_boundary is asserted once every 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 8, 8, 10, 10 PCS RS-Frames.
  - In other words, get a new 10-bit symbol when it is needed.
- Behavior during EEE refresh to be determined

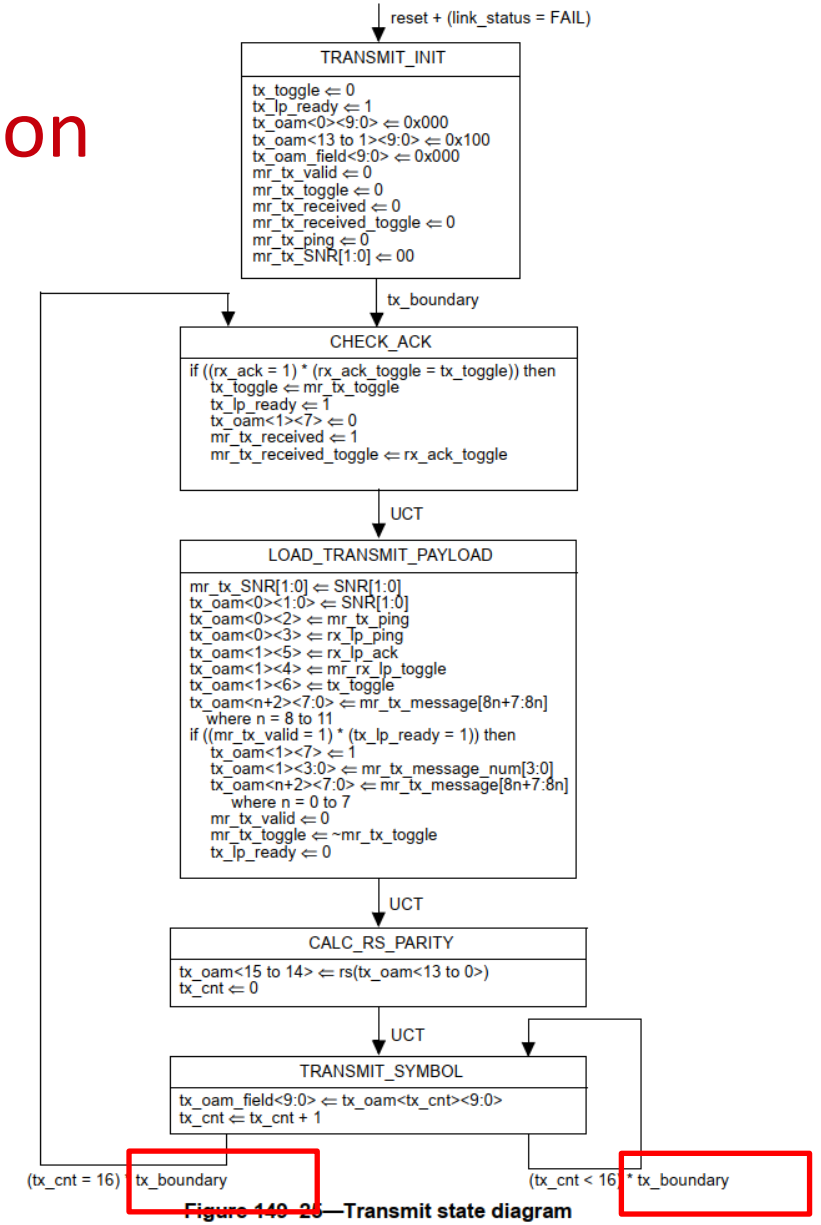

Figure 149–26—Transmit state diagram

Clause 149 rx_boundary Variable Redefinition

- rx_boundary is asserted once every 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 8, 8, 10, 10 PCS RS-Frames once the frame alignment is known
  - In other words, output a new 10-bit symbol when it is ready.
- Behavior during EEE refresh to be determined

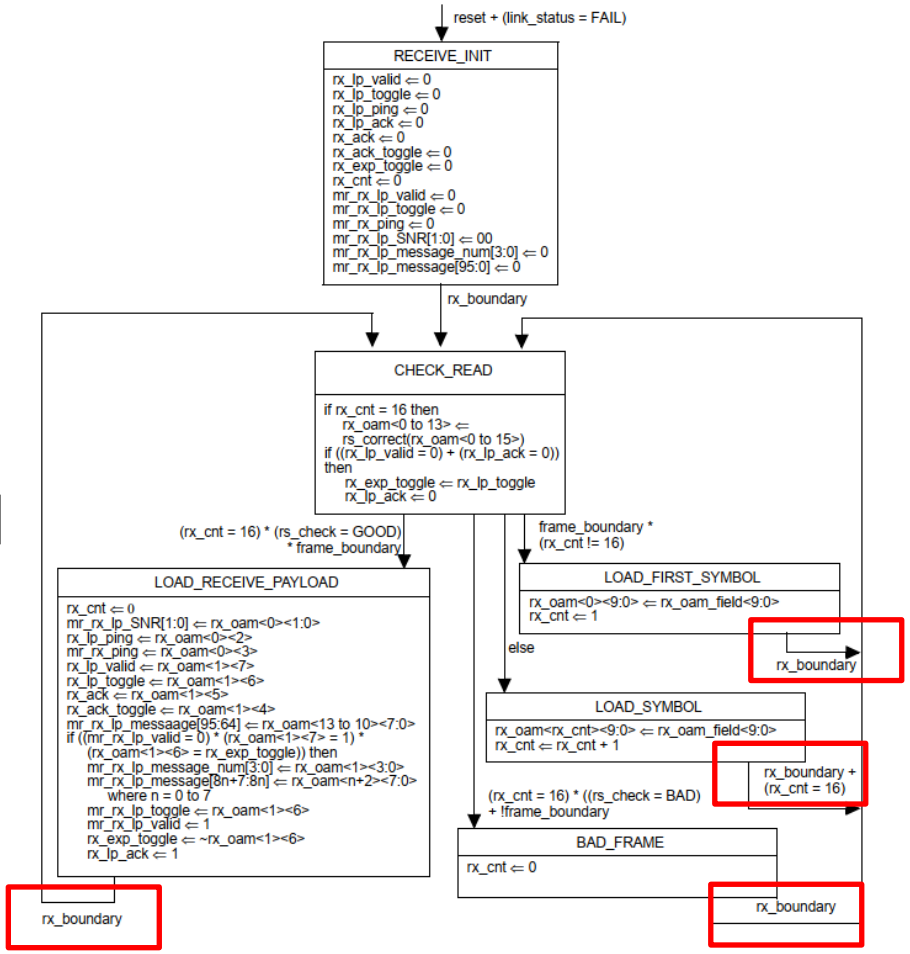

Figure 149–24—Receive state diagram

OAM During Low Power Idle

- Not limited to sending one bit during each EEE refresh
- Do we want to take advantage of the OAM RS-FEC correction?
  - Send aligned 10-bit symbol during refresh

Alternative Solution 1

- Be less efficient and do not compact the 160 bits and serialize as is
- tx_boundary is asserted once every 10 PCS RS-Frames
- rx_boundary is asserted once every 10 PCS RS-Frames once the frame alignment is known

Alternative Solution 2

- Be more efficient and compact to 132 bits
  - Serialize only the lower 8 bits for the first 14 symbols
  - Serialize the two 10-bit parity symbols
- Find alignment by testing for good parity
  - Is aliasing possible (incorrect alignment but good parity)??
- tx_boundary is asserted once 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 10, 10 PCS RS-Frames
- rx_boundary is asserted once 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 10, 10 PCS RS-Frames once the frame alignment is known

THANK YOU