Yet Another OAM Proposal

William Lo, Axonne Inc.

October 17, 2018

Problem Statement

- Need a standard method to read failed states
- Need status even when micro is “out to lunch” – no communication between MAC/PHY interface
- http://www.ieee802.org/3/ch/public/adhoc/wienckowski_3ch_01_072618.pdf Slide 2

What does the problem statement mean

- A standard method to read failed states
  - Function of the failed state status is well defined
  - Register location to interrogate the status is well defined

- Need status even when micro is “out to lunch”
  - PHY can read link partner failed state status even if there is no CPU active at the link partner

Two Previously Proposed Access Mechanisms

- Continuously exchange one symbol every RS frame
  - http://www.ieee802.org/3/ch/public/sep18/wienckowski_3ch_01b_0918.pdf
- More generalized approach to read any register in the link partner
  - http://www.ieee802.org/3/ch/public/adhoc/langner_3chah_01_100218.pdf

Does The Access Mechanism Matter?

- No – both the IEEE register location and function specification must be defined regardless of access mechanism used.

Access Mechanism	Symbol Exchange	Generalized Register Read
How is fixed address location specified	Indirectly - Specify OAM bit position. OAM bit maps to IEEE register	Directly - IEEE register
Where to specify failed state definition	Either normative in the standard, or informative in an annex	Must be normative since IEEE register must be specified

What about when the CPU goes out to lunch?

- When CPU is running we can use layer 2 protocols such as LLDP to monitor link partner status. CPU can read any register it wants from the PHY
- When link partner CPU dies, is it really necessary to have general flexibility to read any register in the link partner?
- At the point of CPU failure just a few critical statuses are needed.

- What is the use case for layer 1 generalized register access?
  - It should not be a replacement for layer 2 access

Pros and Cons of the Two Access Mechanisms

Symbol exchange

- Pro:
  - Simple mechanism
  - No interference with IEEE register clearing
  - No security concerns about which registers are readable
  - Option to specify various statuses as normative or informative
- Con:
  - Only 4 bits available

Generalized Register Access

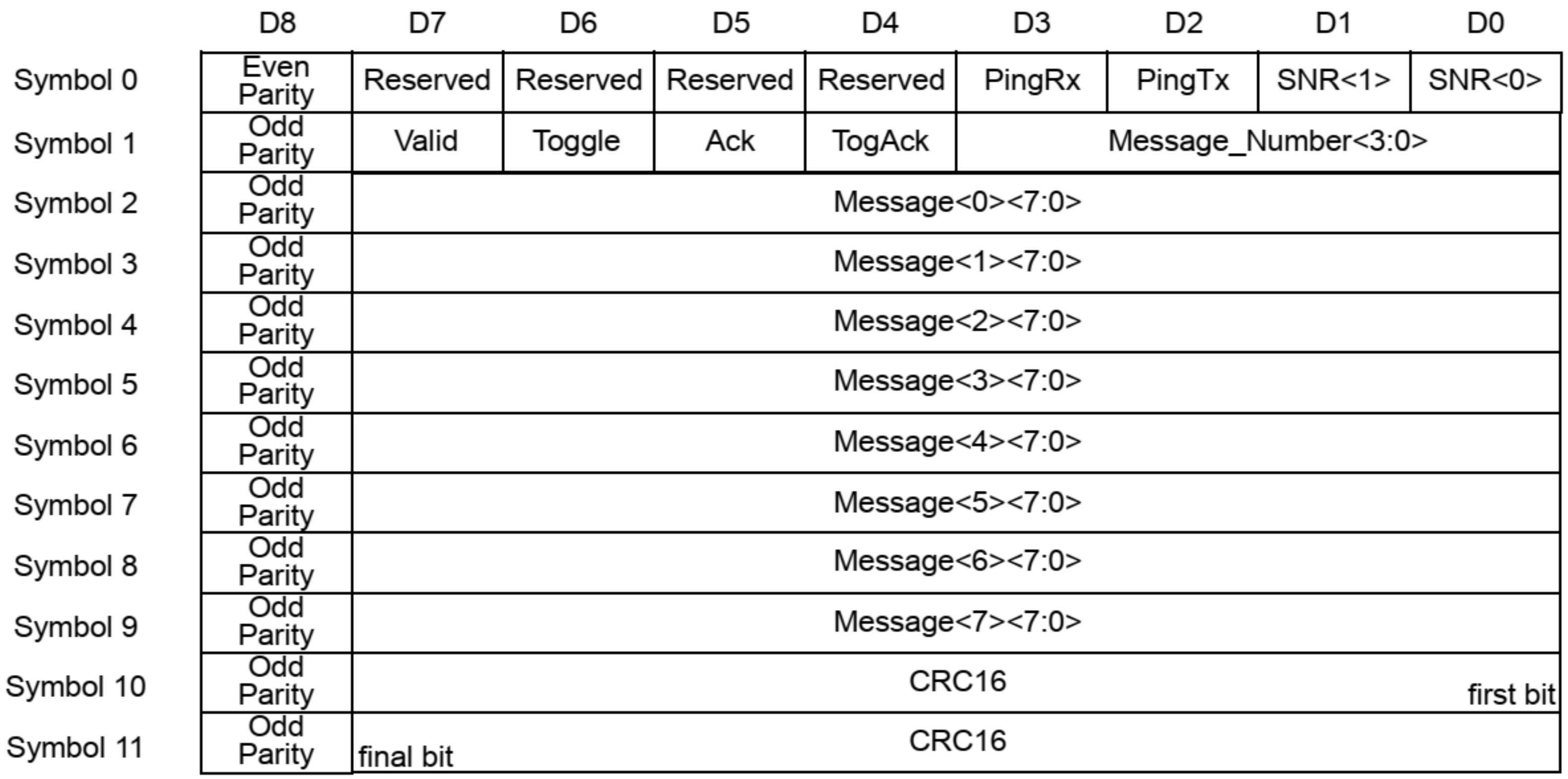
- Pro:
  - Flexible and powerful – any defined register is potentially available
- Con:
  - Complex mechanism
  - Interferes with IEEE register clearing
  - Need to control which set of registers are readable
  - Must specify registers as normative

Yet another proposed access mechanism

- 64 bits available to define various status
- 16 message types

- Use existing 1000BASE-T1 PCS-OAM mechanism
  - No interference with IEEE register clearing
  - No security concerns about which registers are readable
  - Option to specify various statuses as normative or informative

1000BASE-T1 OAM Frame



	D8	D7	D6	D5	D4	D3	D2	D1	D0
Symbol 0	Even Parity	Reserved	Reserved	Reserved	Reserved	PingRx	PingTx	SNR<1>	SNR<0>
Symbol 1	Odd Parity	Valid	Toggle	Ack	TogAck	Message_Number<3:0>			
Symbol 2	Odd Parity	Message<0><7:0>							
Symbol 3	Odd Parity	Message<1><7:0>							
Symbol 4	Odd Parity	Message<2><7:0>							
Symbol 5	Odd Parity	Message<3><7:0>							
Symbol 6	Odd Parity	Message<4><7:0>							
Symbol 7	Odd Parity	Message<5><7:0>							
Symbol 8	Odd Parity	Message<6><7:0>							
Symbol 9	Odd Parity	Message<7><7:0>							
Symbol 10	Odd Parity	CRC16							first bit
Symbol 11	Odd Parity	final bit CRC16							

Figure 97–15—OAM Frame

MGBASE-T1 OAM Frame

	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
Symbol 0	0	Even Parity	Message_Type<3:0>				Ping Rx	Ping Tx	SNR<1>	SNR<0>
Symbol 1	0	Odd Parity	Valid	Toggle	Ack	TogAck	Message_Number<3:0>			
Symbol 2	0	Odd Parity	Message<0><7:0>							
Symbol 3	0	Odd Parity	Message<1><7:0>							
Symbol 4	0	Odd Parity	Message<2><7:0>							
Symbol 5	0	Odd Parity	Message<3><7:0>							
Symbol 6	0	Odd Parity	Message<4><7:0>							
Symbol 7	0	Odd Parity	Message<5><7:0>							
Symbol 8	0	Odd Parity	Message<6><7:0>							
Symbol 9	0	Odd Parity	Message<7><7:0>							
Symbol 10	0	Odd Parity	CRC16							
Symbol 11	0	Odd Parity	CRC16							

- Type of message exchanged depends on the message type
- Define Message_Type 0000 and 0001. The rest are reserved.
- Message_Type 0000 – identical to 1000BASE-T1 PCS-OAM

Message_Type = 0001

	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
Symbol 0	0	Even Parity	0	0	0	1	Ping Rx	Ping Tx	SNR<1>	SNR<0>
Symbol 1	0	Odd Parity	Valid	Toggle	Ack	TogAck	CMD<1:0>		Reserved	Reserved
Symbol 2	0	Odd Parity	Message<0><7:0>							
Symbol 3	0	Odd Parity	Message<1><7:0>							
Symbol 4	0	Odd Parity	Message<2><7:0>							
Symbol 5	0	Odd Parity	Message<3><7:0>							
Symbol 6	0	Odd Parity	Message<4><7:0>							
Symbol 7	0	Odd Parity	Message<5><7:0>							
Symbol 8	0	Odd Parity	Message<6><7:0>							
Symbol 9	0	Odd Parity	Message<7><7:0>							
Symbol 10	0	Odd Parity	CRC16							
Symbol 11	0	Odd Parity	CRC16							

- CMD<1:0>
- 00 – Do nothing. Message<7:0><7:0> format undefined
- 01 – Request. PHY send its status in Messsage<7:0><7:0> and requests link partner status
- 10 – Response. PHY send its status in Messsage<7:0><7:0> in response to link partner's request
- 11 – Both. Respond to link partner's request and request new status from link partner

Summary

- Must define functionality and IEEE register location independent of choice of access mechanism
- All 3 proposals work without CPU intervention at the link partner
- Symbol exchange and current proposal have option to specify status as being normative or informative.

- Current proposal uses existing 1000BASE-T1 PCS-OAM mechanism and introduces Message_Type<3:0>
  - Message_Type = 0000 – Used defined message (Identical to 1000BASE-T1)
  - Message_Type = 0001 – Predefined format for automotive applications

THANK YOU