802.3ch Link Partner Register Access

Paul Langner

George Zimmerman

Saied Benyamin

Problem

- Construct a standard HW access mechanism that utilizes the existing OAM to allow reliable reading of link-partner MDIO registers. Provide in-field debug help.
- Framework:
  - The OAM provides a 4-bit message number field and 8 octets of message data as follows. I have only shown the payloa section of the OAM

D7	D6	D5	D4	D3	D2	D1	D0
Reserved	Reserved	Reserved	Reserved	PingRx	PingTx	SNR<1>	SNR<0>
Valid	Toggle	Ack	TogAck	Message_Number<3:0>			
Message<0><7:0>							
Message<1><7:0>							
Message<2><7:0>							
Message<3><7:0>							
Message<4><7:0>							
Message<5><7:0>							
Message<6><7:0>							
Message<7><7:0>							

Design Constraints

- Need an interrupt bit to reflect the state of the remote interrupt pin
- Need to be able to send MMD, Address, and Command
- Need a Transaction Number to distinguish read operations
- Command list should include:
  - Read (Read one word)
  - Read response
  - A path to future addition of a secure authenticated write function
- Need to be non-disruptive to link partner operation (i.e. reading latching bits does not clear them, reading self clearing counters does not clear them, etc)
- Allow link partner to refuse the read operation

Details of following tables

- INT = 1 indicates the link-partner has an interrupt, a reserved bit is repurposed to permanently carry this indicator not just during MDIO transactions
- Message Number =0001 is used for remote MDIO access
- Transaction ID is a 4 bit hardware counter that tracks requests and responses
  - Each PHY has its own rolling 4-bit counter and matches read transaction IDs against read-response transaction IDs
- Message<0><3:0> is used for command
  - 0 = Invalid Command
  - 1 = Read
  - 2 = Read response
  - 3 = Reserved for MDIO future expansion of authenticated write
  - 0, 4-15 remain open for future use
- MMD/Address are for the register to read
- Data is the returned data read by the link partner

Read command

- Read command done via CMD = 0001
- Transaction ID, MMD and Address are stuffed as below

D7	D6	D5	D4	D3	D2	D1	D0
Reserved	Reserved	Reserved	INT	PingRx	PingTx	SNR<1>	SNR<0>
Valid	Toggle	Ack	TogAck	Message Number = 'b0001			
Reserved	Reserved	Reserved	Reserved	CMD<3:0> = 'b0001			
Transaction ID<3:0>				MMD<3:0>			
Address<7:0>							
Address<15:8>							
reserved							
reserved							
reserved							
reserved							

Read Response

- Read Response done via CMD = 0010, continues as long as the link partner is sending the read command
  - Read Status=0 successful read
  - Read Status=1 Read failed (non-existent address or other reason)
  - Read Status =2 Read Refused

D7	D6	D5	D4	D3	D2	D1	D0
Reserved	Reserved	Reserved	INT	PingRx	PingTx	SNR<1>	SNR<0>
Valid	Toggle	Ack	TogAck	Message Number = 'b0001			
Reserved	Reserved	Reserved	Reserved	CMD<3:0> = 'b0010			
Transaction ID<3:0>				ReadStatus<3:0>			
Data<7:0>							
Data<15:8>							
reserved							
reserved							
reserved							
reserved							

Timing

- Example read times:

1. Data Mode: Read request round-trip takes:
   - 320ns x 12 = 4.3us transmit + 0.5us decode + 3.2 us read register* + 320ns x 12 = 4.3us response = ~12uS
2. EEE Mode: Read request round-trip takes:
   - 40uS x 12 x = 480uS transmit + 0.5us decode + 3.2 us read register + 480us
   - Response = ~1ms

* Typical MDIO transaction time at 10 MHz = ~3.2us

Example Usage

Read Example:

1. Write OAM Message<2:1> with desired Address to read
2. Write OAM Message<0><3:0> with the desired MMD
3. Write OAM Message<0><7:4> with desired transaction ID
4. Write Message<0><3:0> CMD=0001
5. Write OAM Message Valid to a ‘1’
6. Poll link Partner Message Valid until set to ‘1’
7. Check link partner Message<1><3:0> for status to see if read was successful
8. Check that we have a transaction ID match, data in Message<2:1>

Interrupt Example:

1. INT bit in link partner register is constantly upate with link partner’s interrupt status

* Or connect this bit into the local interrupt tree, so polling is not required

Thank you.

AQUANTIA®

ACCELERATING CONNECTIVITY