Enhanced SMII Proposal

William Lo

November 15, 2023

Reduced Pin Count MII

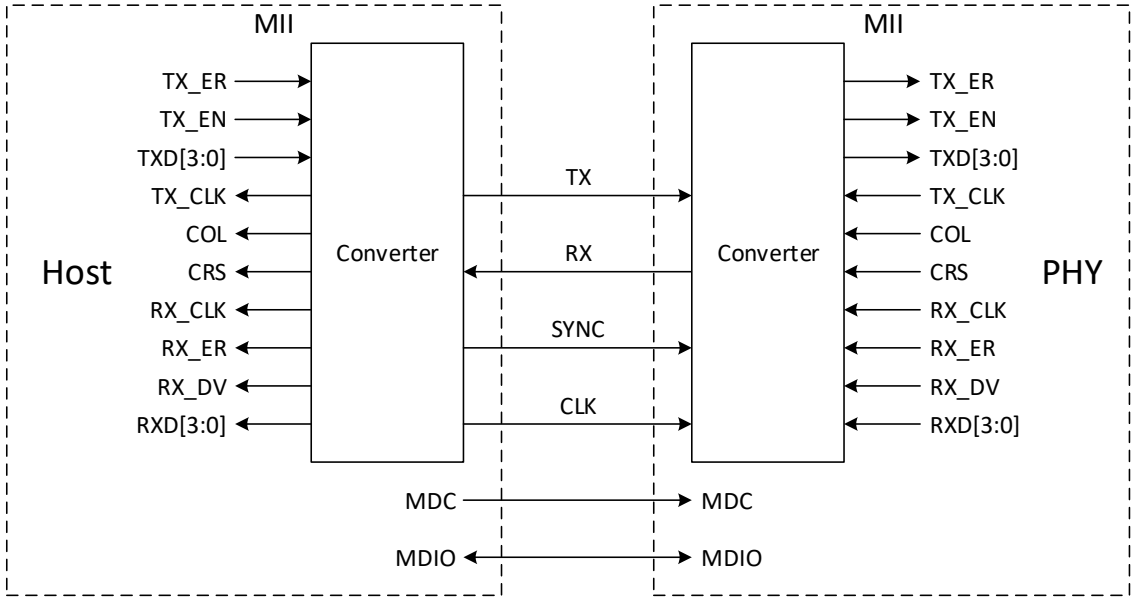
- Already proposed many years ago
- One implementation

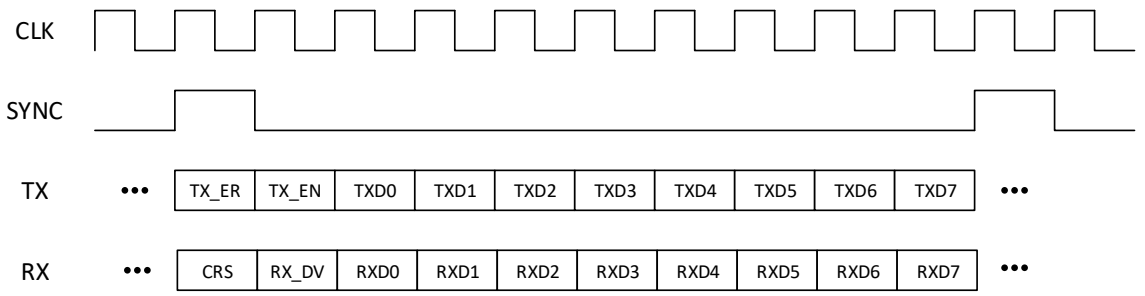
  https://opencores.org/ocsvn/smii/smii/trunk/doc/SMII.pdf.

- Agenda
  - Quickly go over SMII
  - Discuss additional enhancements

Serial MII (SMII)

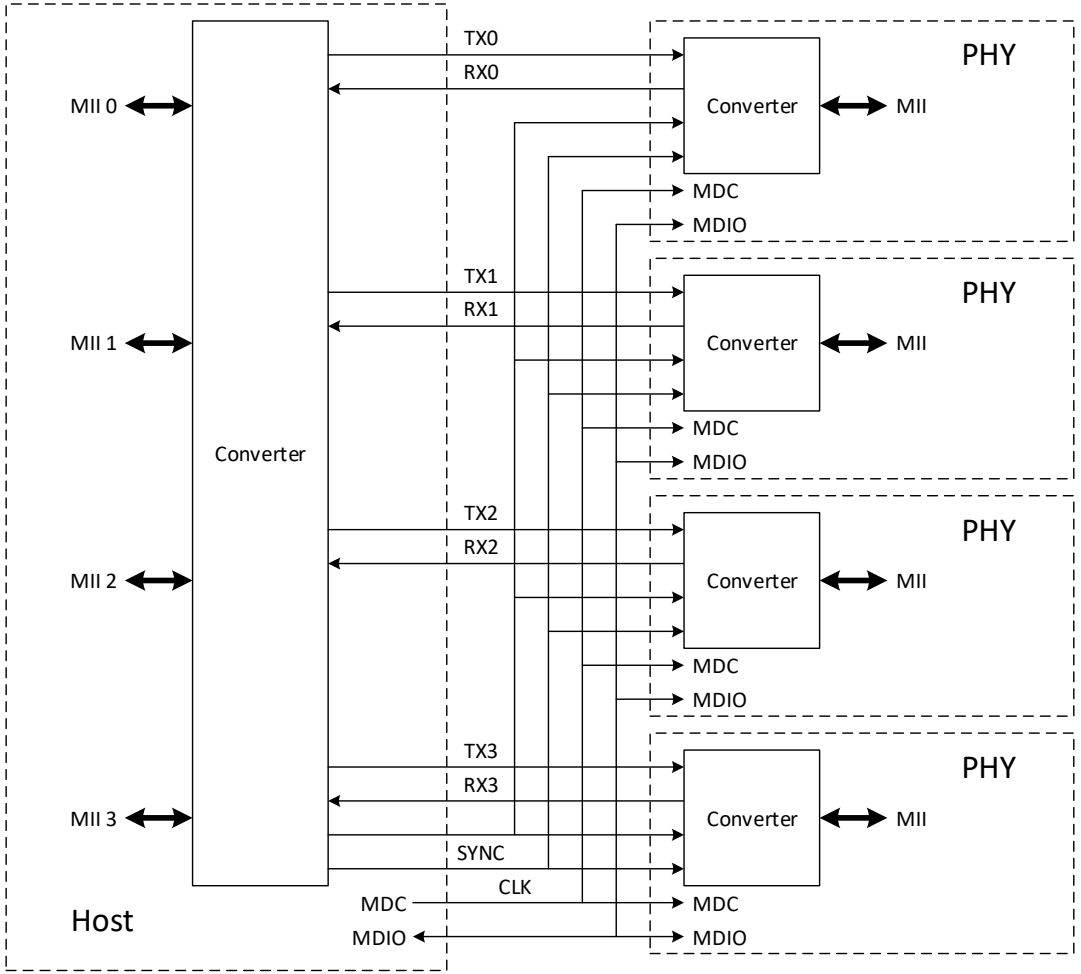
- 4-Pin Digital Interface
- Serializes two nibbles to 10 bits
- 10-bit data framed by SYNC pulse
- Common SYNC and CLK
- CLK nominally 125MHz for 100Mb/s and 12.5MHz for 10 Mb/s
- Supports half duplex
- MDC/MDIO remains separate

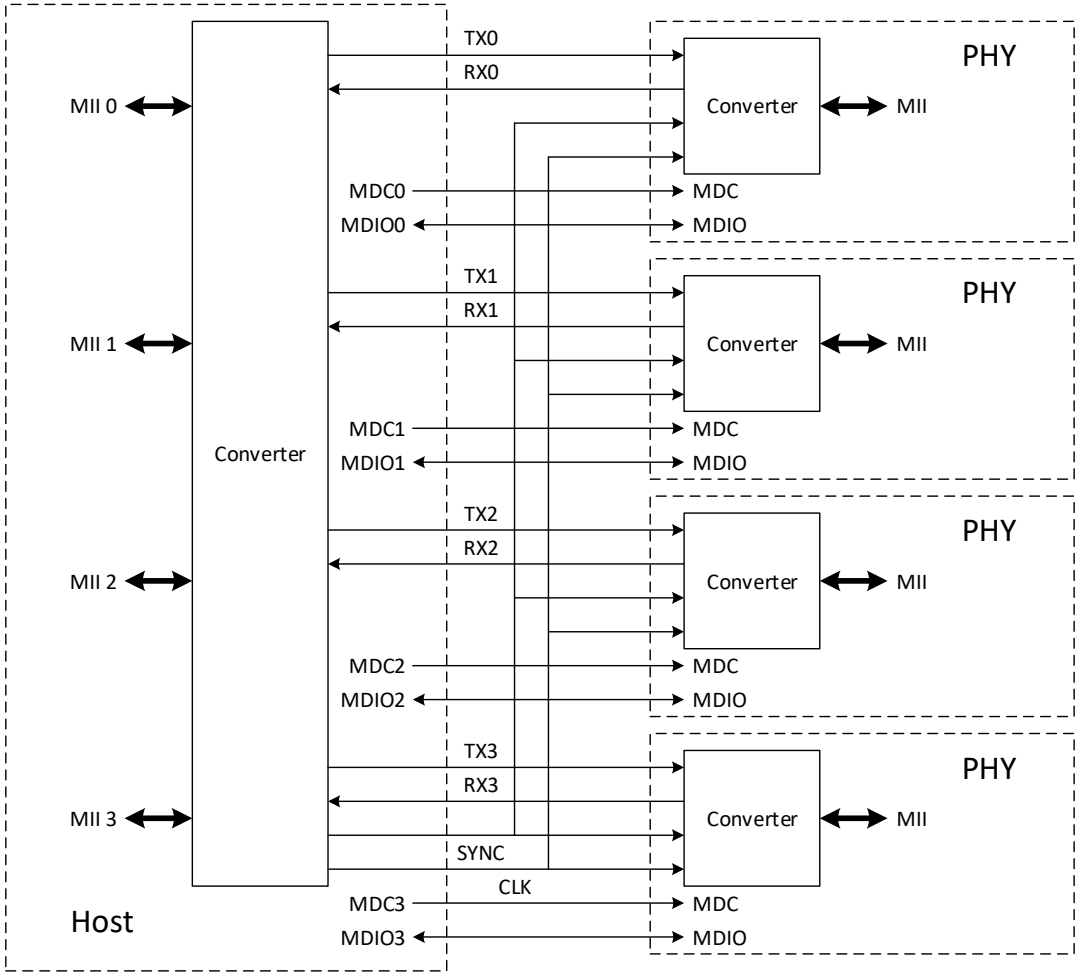




Multiport SMII

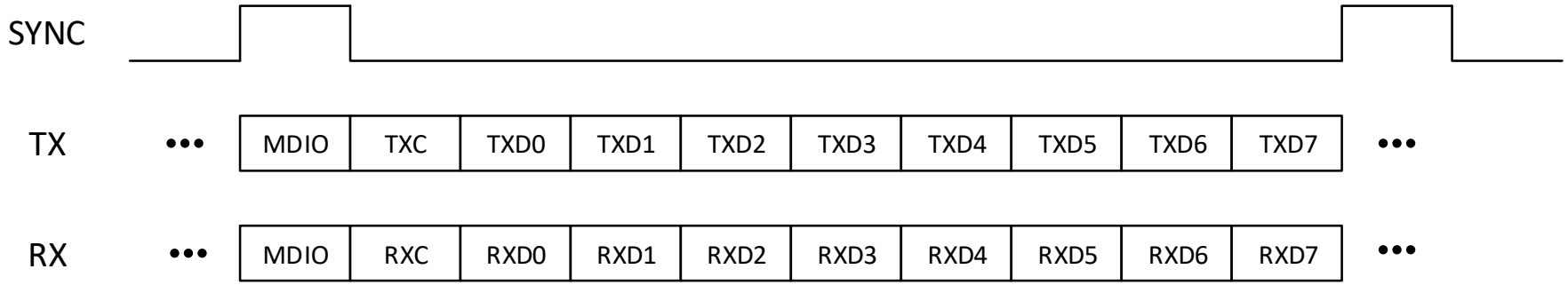
- 2N + 4 pins per port shared MDC/MDIO
- 4N + 2 pins per port independent MDC/MDIO





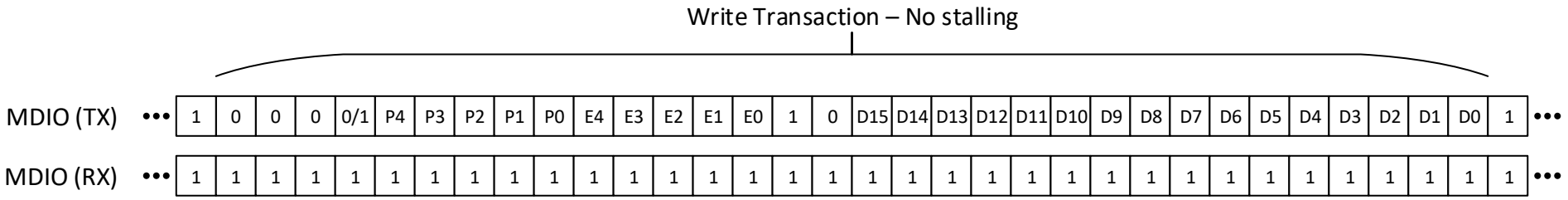
Enhancement – Embedded MDIO

- Re-Encode 2 nibbles to 9 bits
- Carry one MDC worth of MDIO information (1 bit) when SYNC is high
  - Allows parallel per port access to registers without MDC/MDIO pins
  - Much faster than MDC/MDIO – 80ns per bit (12.5MHz)
  - No change to basic MDIO protocol – 32-bit transactions, sync up transactions using preambles, preamble suppression.



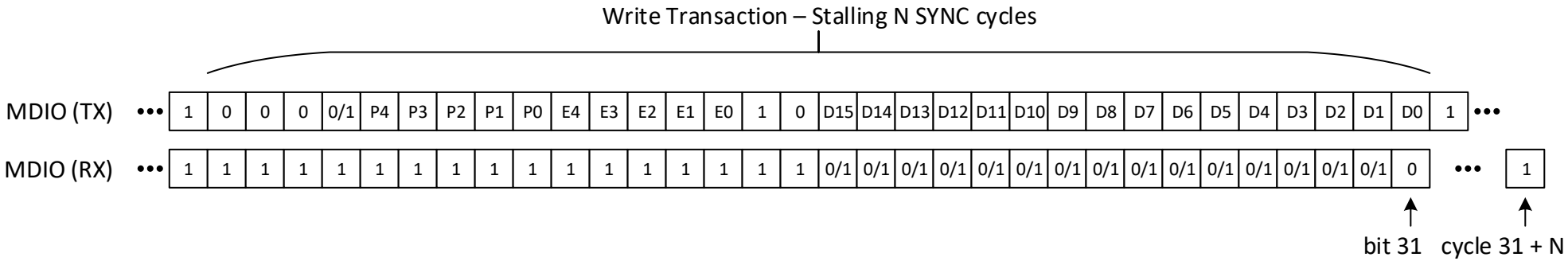
Write Operation

- MDIO bit set to 1 when inactive – just like the MDIO bus being pulled high when device Hi-Z
- Write protocol exactly the same as on a physical MDIO bus.
- MDIO bit on RX always set to 1 if no stalling
- Clause 45 example below



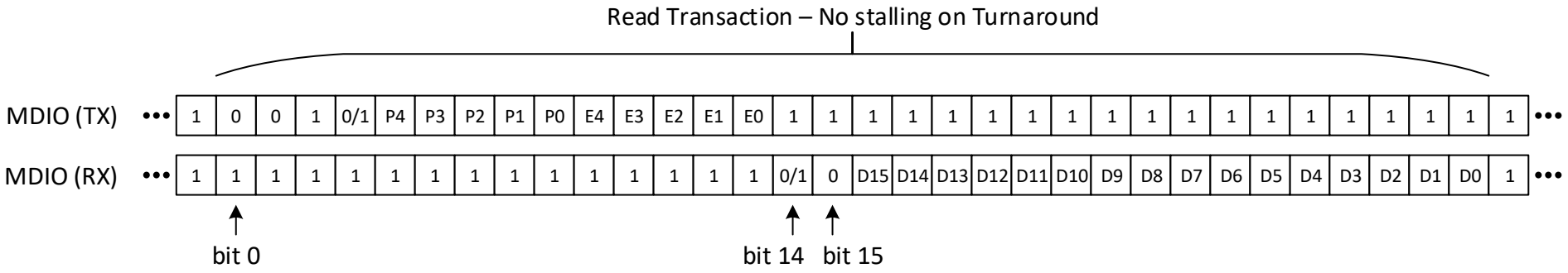
Write Operation with stalling

- In case PHY cannot process write transaction fast enough set bit 31 of the transaction to 0 on the RX to indicate to STA not to start another transaction
- Set bit to 1 once ready for another transaction
- Can optionally set other bits to 0 for earlier indication of a stall



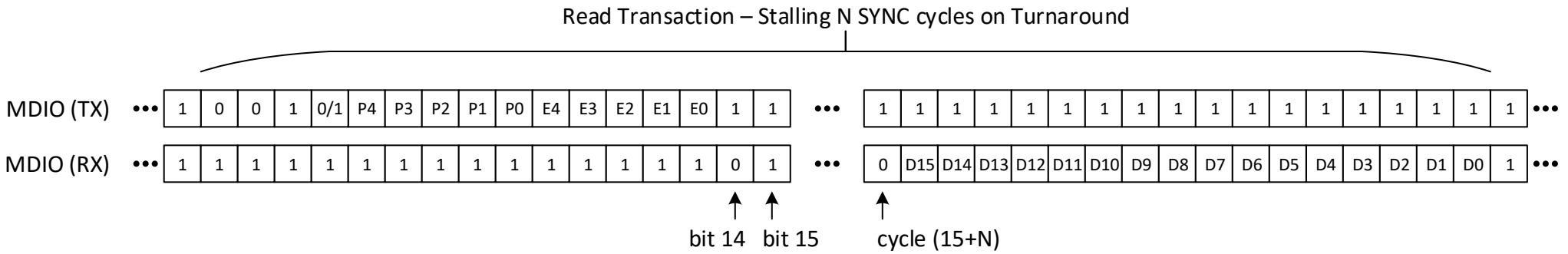
Read Operation

- Read protocol exactly the same as on a physical MDIO bus except handling during turn around.
- Bit 15 set to 0 on RX if data is available starting in bit 16
- Bit 14 can be 0 or 1 for easier implementation (next slide)



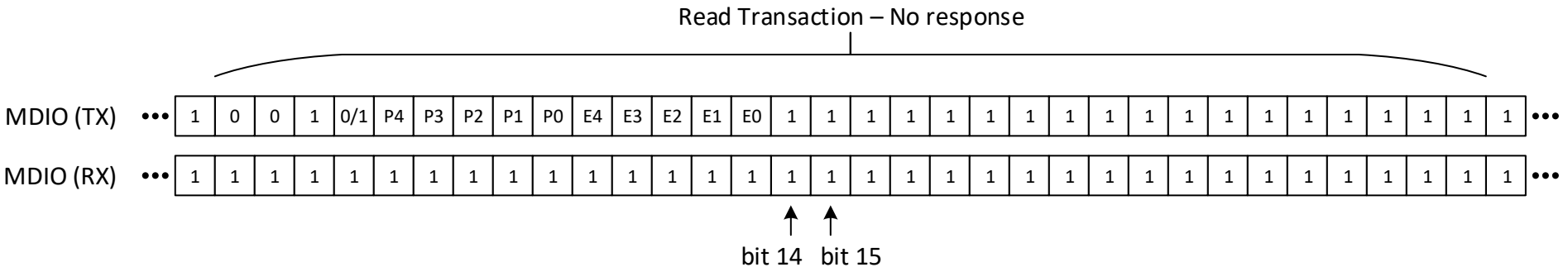
Read Operation with stalling

- PHY on physical MDC/MDIO has no options for clock stretching during turnaround
- Bit 14 set to 0 on RX if data is not going to be available starting in bit 16
- Set MDIO bit to 1 starting in bit 15 and hold to 1 until read data is ready. Set next bit to 0 and then continue returning read data.



No response to Read Operation

- Bits 14 and 15 never set to 0 indicating no response



9-bit coding

- Align to XGMII capabilities
  - Full Duplex Only (We can add codes for half duplex if needed)
  - Support Sequence Ordered Set

MDIO	TXC/RXC	TXD/RXD[7:0]	Description
X	0	Data[7:0]	Data
X	1	0xFF	Idle
X	1	0x0F	Low Power Idle
X	1	0x01	Symbol Error
X	1	0xAA	Sequence
X	1	0xA5	Abort Sequence
X	1	0x54	Invalid (Sync)
X	1	0x55	Invalid (Sync)
X	1	0x52	Invalid (Sync)
X	1	0x56	Invalid (Sync)
X	1	0x4A	Invalid (Sync)
X	1	0x5A	Invalid (Sync)
X	1	0x2A	Invalid (Sync)
X	1	0x6A	Invalid (Sync)
X	1	else	Reserved

Symbol 0	Symbol 1	Symbol 2	Symbol 3	Description
Sequence	0x00	0x00	0x00	Reserved
Sequence	0x00	0x00	0x01	Local Fault
Sequence	0x00	0x00	0x02	Remote Fault
Sequence	0x00	0x00	0x03	Link Interruption
Sequence	0x55	0x55	0x55	Sync SMII
Sequence	other combinations			Reserved

Code value chosen to minimize toggling

- Idle – Assuming MDIO is not active
  - TXC = 1, TXD[7:0] = 0xFF - no toggling on TX.
- Low power idle
  - TXC = 1, TXD [7:0] = 0x0F – One low to high, one high to low per SYNC cycle
- Symbol error
  - TXC = 1, TXD [7:0] = 0x01 – One low to high, one high to low per SYNC cycle
- Sequence symbol
  - TXC = 1, TXD[7:0] = 0xAA – reason discussed later

Sequence Ordered Set

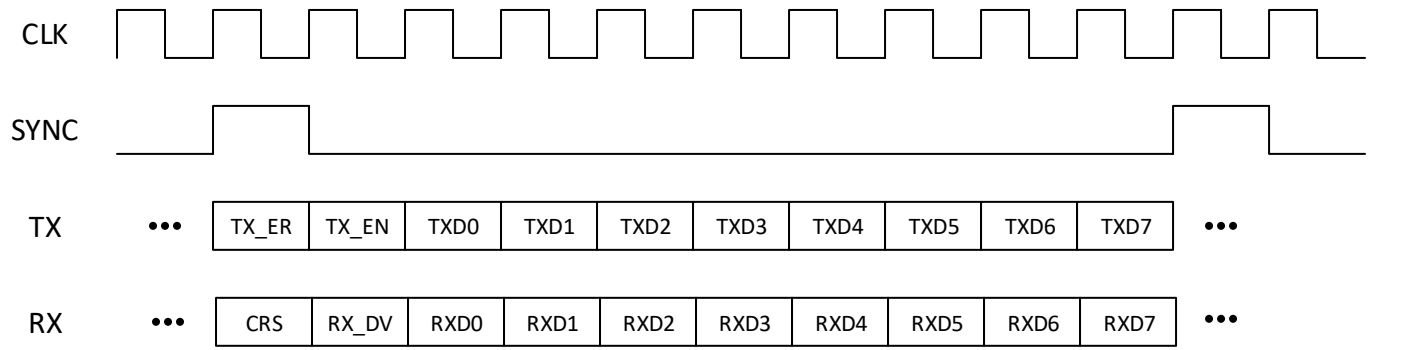
- In XGMII, ordered sets are a control symbol plus 3 data symbols aligned to the XGMII 4-lane boundary
- No concept of 4-lane boundary in MII

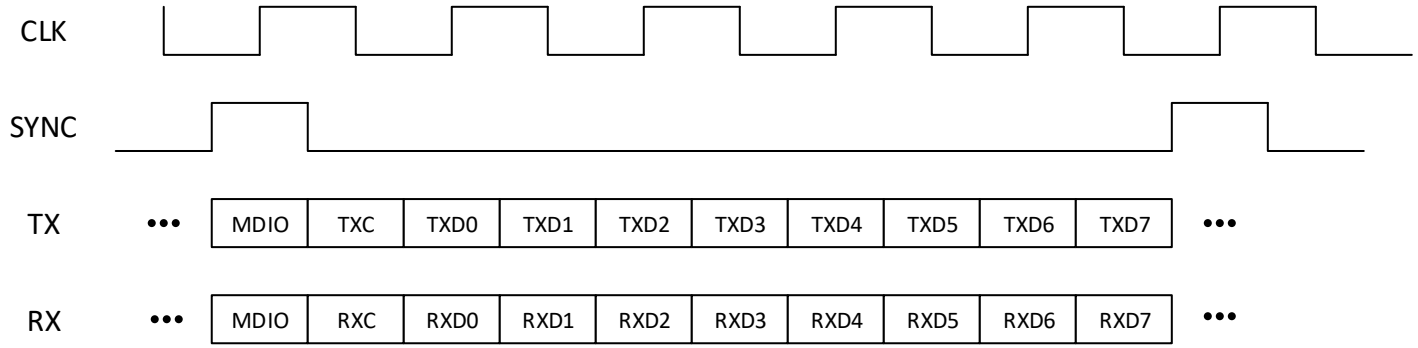
- Ordered set can start at any byte boundary
  - Once sequence symbol seen, the next 3 bytes are part of the sequence ordered set
- Can terminate the 4 symbol sequence early with Abort Sequence symbol. i.e. a packet is pending and we don't want to delay it.
  - Ignore the terminated sequence ordered set
- Alternative possibility – once sequence ordered set starts it must be completed.

Enhancement – Half Speed Clock

- Reduce CLK toggle rate
- Also helps with synchronization (next slide)

- We can discuss reasonable setup/hold times





Enhancement – Eliminate SYNC pin

- When combined with half rate clock, the SYNC boundary can be found using the Sync SMII sequence ordered set pattern.
- TXC = 1, TXD[7:0] = 1x1x1x1x pattern cannot be generated by skewing any bitstream with TXC = 0, TXD[7:0] = xxxxxxxx on the falling edges of CLK
- Hence the only combinations of control symbols with other symbols can create a pattern that looks like the Sync SMII sequence ordered set.
- Chart below in gray shows the control symbols to exclude from use (invalid Sync) to prevent false alignments

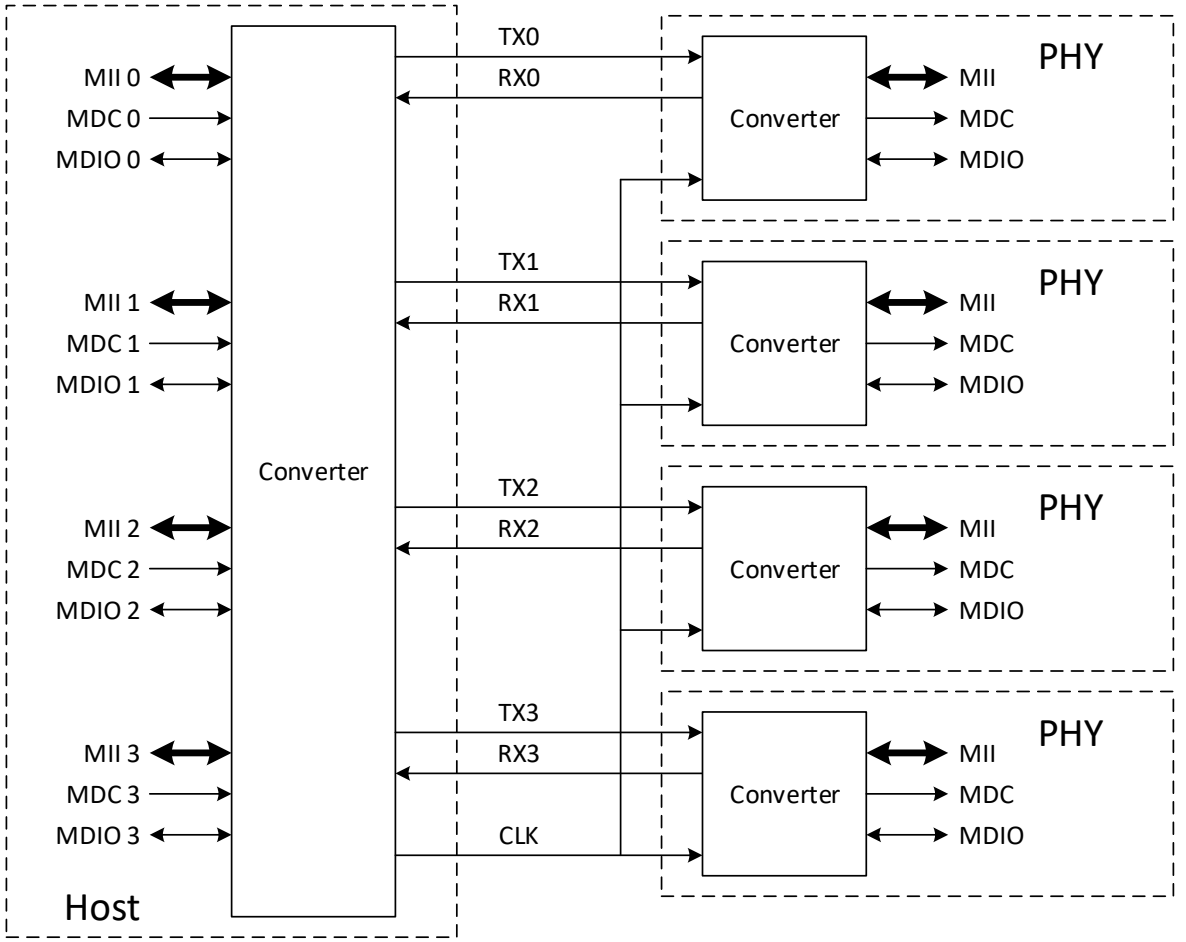
MDIO	TXC	TXD0	TXD1	TXD2	TXD3	TXD4	TXD5	TXD6	TXD7	MDIO	TXC	TXD0	TXD1	TXD2	TXD3	TXD4	TXD5	TXD6	TXD7	MDIO	TXC	TXD0	TXD1	TXD2	TXD3	TXD4	TXD5	Symbol causing issues
x	1	0	1	0	1	0	1	0	1	x	0	1	0	1	0	1	0	1	0									OK
		x	1	0	1	0	1	0	1	0	1	x	0	1	0	1	0	1	0	1	0							0x54, 0x55
				x	1	0	1	0	1	0	1	0	1	x	0	1	0	1	0	1	0	1	0					0x52, 0x56
						x	1	0	1	0	1	0	1	0	1	x	0	1	0	1	0	1	0	1	0			0x4A, 0x5A
								x	1	0	1	0	1	0	1	0	1	x	0	1	0	1	0	1	0	1	0	0x2A, 0x6A

Eliminate SYNC pin

- SMII sync sequence ordered sets are needed only on the TX. The RX should align to the TX boundary.
- Send this ordered set at the beginning of startup, or periodically.
- The PHY should view this ordered set as idle or low power idle and not forward this to the network.

Summary

- If all enhancements implemented then 2N + 1 pins for N ports
- Embedded MDIO
- Sequence Ordered Set
- Half rate clock
- No Sync pin



THANK YOU