Constant Latency MII Encoding and Ordered Sets Onto 8N/(8N+1) encoder

15 July 2024

Agenda

- Propose MII encoding to for ordered sets
- Describe concept and rules before the 8N/(8N+1) encoder and after the 8N/(8N+1) decoder for constant latency
- Describe modification to 1000BASE-T1 8N/(8N+1) pointer and control codes to help with nibble constant latency

Supported Codes on MII

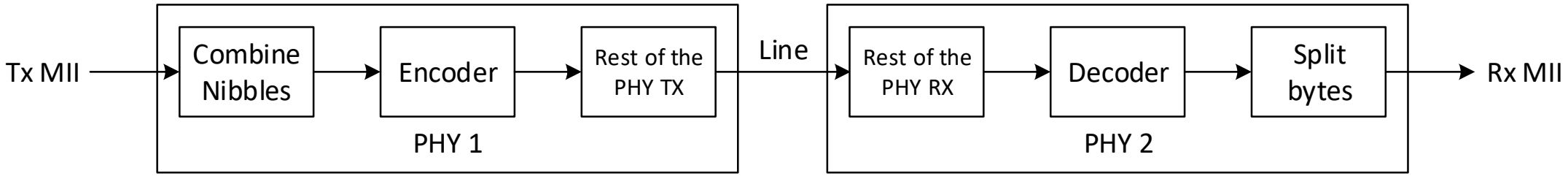
- Eight nibble sequence ordered set similar to the 4-byte sequence used in XGMII.
  Sequence, Sequence, Data1[3:0], Data1[7:4], Data2[3:0], Data2[7:4], Data3[3:0], Data3[7:4]
- Data1, Data2, Data3 mapping and meaning same as for XGMII lane1, lane2, lane3
- Half-Duplex not supported

TX_EN	TX_ER	TXD<3:0>	Indication
0	0	0000 to 1111	Normal inter-frame
0	1	0000	Reserved
0	1	0001	Assert LPI
0	1	0010	PLCA BEACON request (not supported)
0	1	0011	PLCA COMMIT request (not supported)
0	1	0100	Sequence (New)
0	1	0101 to 1111	Reserved
1	0	0000 to 1111	Normal data transmission
1	1	0000 to 1111	Transmit error propagation

RX_DV	RX_ER	RXD<3:0>	Indication
0	0	0000 to 1111	Normal inter-frame
0	1	0000	Normal inter-frame
0	1	0001	Assert LPI
0	1	0010	PLCA BEACON request (not supported)
0	1	0011	PLCA COMMIT request (not supported)
0	1	0100	Sequence (New)
0	1	0101 to 1101	Reserved
0	1	1110	False Carrier indication (not supported)
0	1	1111	Reserved
1	0	0000 to 1111	Normal data transmission
1	1	0000 to 1111	Transmit error propagation

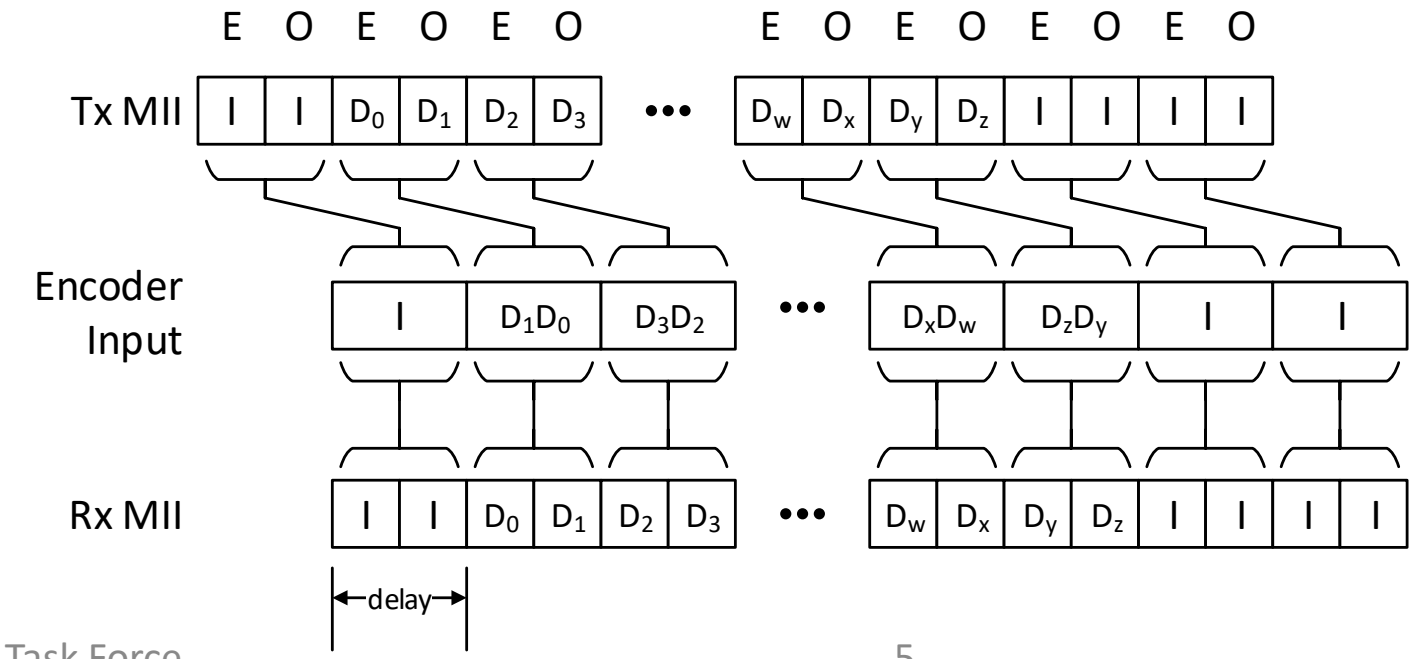
Diagram Reference

- E is even boundary relative to encoder, O is odd boundary
- Encoder input and decoder output are assumed to be the same. Assume constant latency between these 2 points.
- Diagram drawn with zero latency for any paths with constant latency.
- Focus is on how to combine the nibbles and split the bytes for constant latency between Tx MII and Rx MII



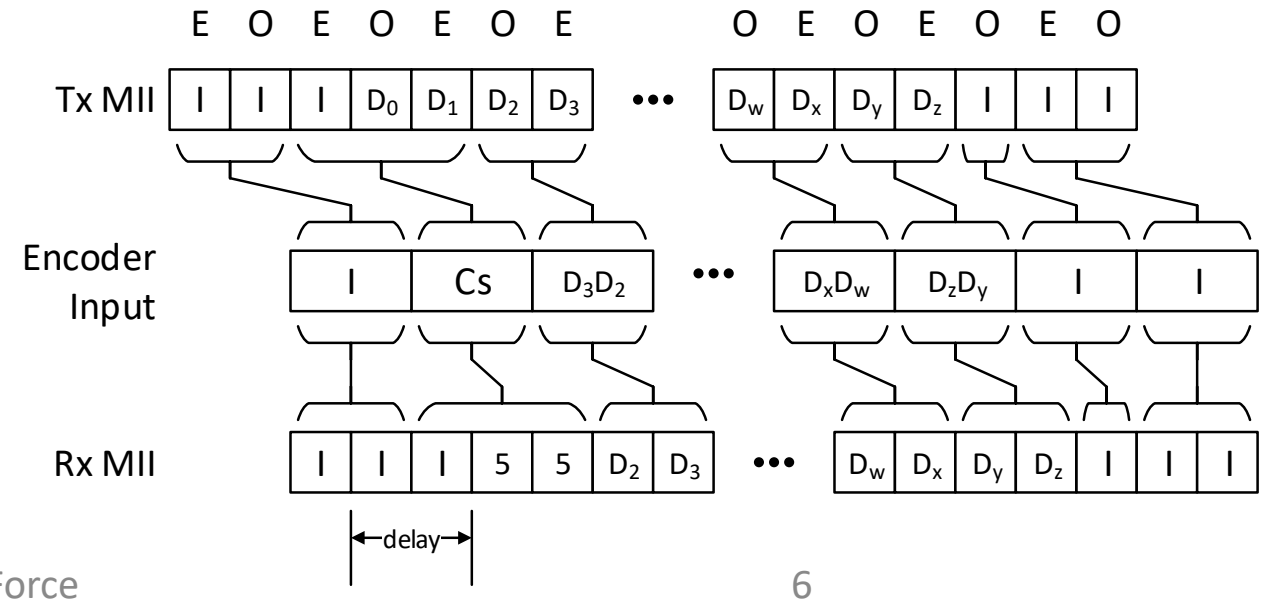
Case 1: start on even cycle, packet even number of nibbles

- Combine nibbles in even followed by odd cycles into a byte presented to the encoder.
- Split byte to nibbles



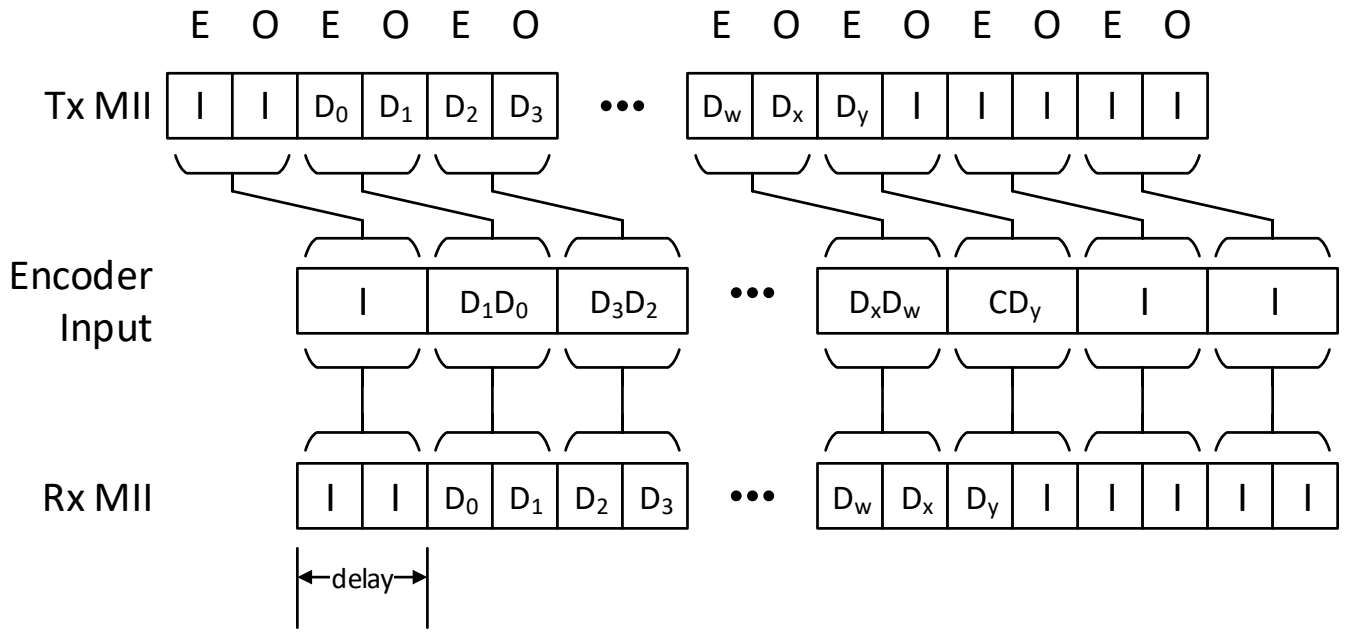
Case 2: start on odd cycle, packet even number of nibbles

- Combine idle nibbles plus first 2 nibbles of preamble to Cs symbol
- Form remaining bytes by combining nibbles in odd followed by even cycles
- First nibble after end of packet expanded to 1 byte of idle
- At decoder, convert Cs to idle, 0x5, 0x5
- Split remaining byte to nibbles
- Convert first idle after end of packet to a single idle



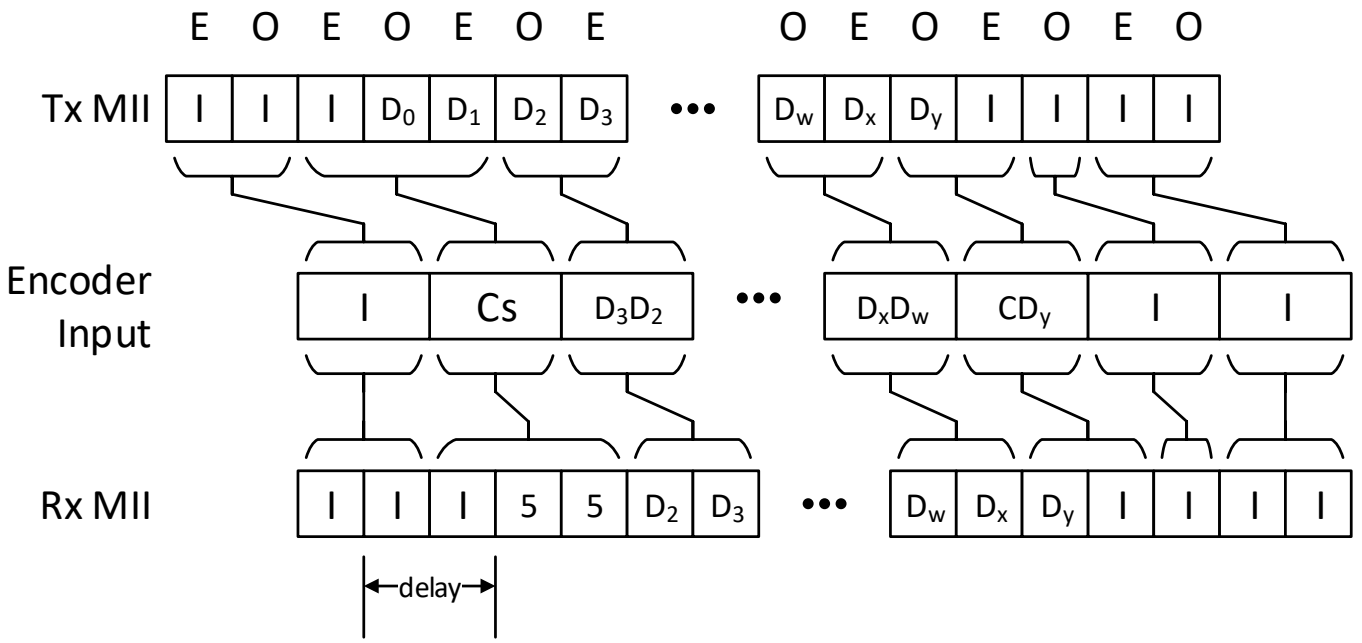
Case 3: start on even cycle, packet odd number of nibbles

- Similar to case 1 except dribble nibble and idle converted to one of 16 control symbols
- Split byte to nibbles with CD code split to data, idle



Case 4: start on odd cycle, packet odd number of nibbles

- Combination of case 2 at start of packet, case 3 at end of packet, and case 2 on idle handling



Ordered Set Handling – Even / Odd boundaries

- Ordered set always even number of nibbles
- Ordered set typically follow each other without idles in between
- Ordered set end is followed by idles before any packet starts

- Treat a stream of ordered set as if it is one big long packet
  - Case1 or Case2 handling
  - Use Co instead of Cs control code

Rule Summary – TX MII to Encoder Input

TX MII	**Data presented at the input of the encoder.**
Start of packet at even nibble	Group even nibble + odd nibbles to byte
Start of packet at odd nibble	Group first 3 nibbles - idle, data, data to Cs symbol aferwards group odd nibble + even nibble to byte
End of packet grouping is data + idle	Group data + idle to one of the 16 CD symbols
End of a packet that started on odd nibble	Convert single idle nibble to idle symbol

Rule Summary – Byte Grouping

- Usually, Control1 and Control2 being different does not happen
- May happen if transition in and out of LPI

First nibble	Second nibble	Byte
data1	data2	{data2, data1}
control	control	Both control are the same control code, then convert to equivalent control symbol. ie. idle, low power idle, sequence, error
x	error	Error symbol
error	x	Error symbol
data	idle	One of the 16 CD symbols
control1	control2	If control1 not equal to control2 and control2 is not an error symbol then encode as if both are control1

Rule Summary – Decoder Output to RX MII

Output of decoder	RX MII
Cs	Output 3 nibbles - idle, 0x55, 0x55
Co	Output 3 nibbles - idle, O, O
CD	Output 2 nibbles - data, idle
First idle after end of packet and packet started with Cs or Co	Output 1 nibble - idle
First idle after end of packet and packet did not start with Cs or Co	Output 2 nibble - idle, idle
Data	Output 2 nibbles - data, data
Control	Output 2 nibbles - Control, Control

8N/(8N+1) Modifications

- 1000BASE-T1 used
  - Bits 0:3 – Pointer to next control code
  - Bit 4 – 1 = more control codes after the next control code, 0 = no more control codes after the next control code
  - Bits 5:7 control code

- Modification
  - Bits 0:2 – Pointer to next control code
  - Bit 3:4 – Mode indicator (see next slide)
  - Bits 5:7 control code

Control Codes

- Mode = 00 – No more control codes
- Mode = 01 – More control codes
- Mode = 1x – Dribble nibble
  - More codes implied since following idles are control words
- No distinction between
  - Normal Inter Frame *_PHY_READY = OK
  - Normal Inter Frame *_PHY_READY = NOT_OK
- If NOT_OK then send Sequence ordered set = remote fault

3	4	5	6	7		
Mode		**Control Code**			**Symbol**	**Definition**
0	0	0	0	0	O	No more control codes, Sequence
0	0	0	0	1	E	No more control codes, Transmit Error Propagation
0	0	0	1	0	I	No more control codes, Normal Inter-Frame
0	0	0	1	1	Cs	No more control codes, Start of Frame with leading idle
0	0	1	0	0	Co	No more control codes, Sequence with leading idle
0	0	1	0	1	L	No more control codes, Assert Low Power Idle
0	0	1	1	0		No more control codes, Reserved
0	0	1	1	1		No more control codes, Reserved
0	1	0	0	0	O	More control codes, Sequence
0	1	0	0	1	E	More control codes, Transmit Error Propagation
0	1	0	1	0	I	More control codes, Normal Inter-Frame
0	1	0	1	1	Cs	More control codes, Start of Frame with leading idle
0	1	1	0	0	Co	More control codes, Sequence with leading idle
0	1	1	0	1	L	More control codes, Assert Low Power Idle
0	1	1	1	0		More control codes, Reserved
0	1	1	1	1		More control codes, Reserved
1	0	0	0	0	CD0	Dribble Nibble, Data = 0x0
1	0	0	0	1	CD8	Dribble Nibble, Data = 0x8
1	0	0	1	0	CD6	Dribble Nibble, Data = 0x6
1	0	0	1	1	CDC	Dribble Nibble, Data = 0xC
1	0	1	0	0	CD2	Dribble Nibble, Data = 0x2
1	0	1	0	1	CDA	Dribble Nibble, Data = 0xA
1	0	1	1	0	CD6	Dribble Nibble, Data = 0x6
1	0	1	1	1	CDE	Dribble Nibble, Data = 0xE
1	1	0	0	0	CD1	Dribble Nibble, Data = 0x1
1	1	0	0	1	CD9	Dribble Nibble, Data = 0x9
1	1	0	1	0	CD5	Dribble Nibble, Data = 0x5
1	1	0	1	1	CDD	Dribble Nibble, Data = 0xD
1	1	1	0	0	CD3	Dribble Nibble, Data = 0x3
1	1	1	0	1	CDB	Dribble Nibble, Data = 0xB
1	1	1	1	0	CD7	Dribble Nibble, Data = 0x7
1	1	1	1	1	CDF	Dribble Nibble, Data = 0xF

Control code correspondence to Nibbles

Control Code	Nibble Equivalent
Sequence	O, O
Transmit Error Propagation	E, E
Normal Inter-Frame	Idle, Idle or Idle
Start of Frame with leading idle	Idle, 0x5, 0x5
Sequence with leading idle	Idle, O, O
Assert Low Power Idle	L, L

THANK YOU