Marvell. Moving Forward Faster

8N/(8N+1) PCS encoding for GEPOF

GEPOF – Study Group- November 2014

William Lo, Marvell

Leverage PCS encoding from 802.3bp

- Asked by GEPOF study group members to share what was done in 1000BASE-T1 Task Force
- This presentation based on Lo_3bp_02_0314.pdf presented in Beijing and correction in Lo_3bp_01_0514.pdf
  - See presentation for motivations for using 8N/(8N+1) coding

1 bit overhead per 8N bits payload

- Maps GMII data onto 8N+1 bit blocks

- N can be 1 to 16 depending on need
- Flexible N allows 8N+1 block to fit with downstream FEC
- N GMII transfers are packed to 8N+1 bit block
- Control symbols can be anywhere in the N GMII transfers

- Will use N=8 as examples in this presentation

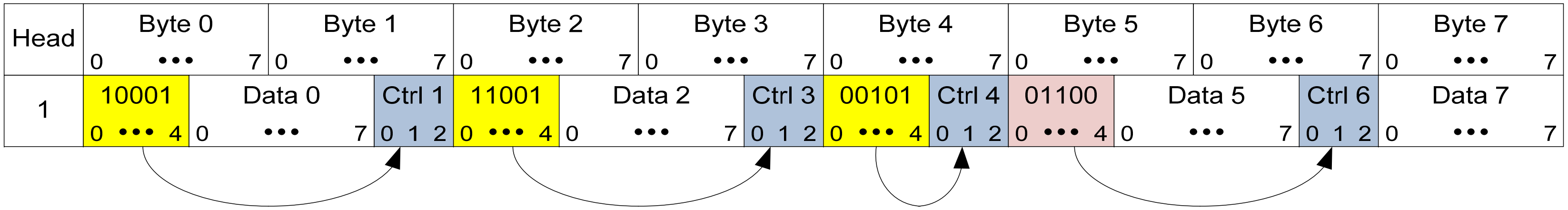
Data Block Encoding

- **All N GMII transfers are data bytes**
- **1 bit header set to 0. N data bytes are concatenated**

0	Data 0	Data 1	Data 2	Data 3	Data 4	Data 5	Data 6	Data 7
	0 ••• 7	0 ••• 7	0 ••• 7	0 ••• 7	0 ••• 7	0 ••• 7	0 ••• 7	0 ••• 7

Control Block Encoding

- If at least 1 GMII transfer is a control byte then 1 bit header is set to 1
- Control byte mapped as a split 5 bit pointer + 3 bit control code
- Bit 0 to 3 of pointer points to next byte that is a control symbol
- Bit 4 of pointer indicates whether the next control symbol is the final control symbol of the block
  - 0 = final one, 1 = more control symbols
- Example: D/C/D/C/C/D/C/D

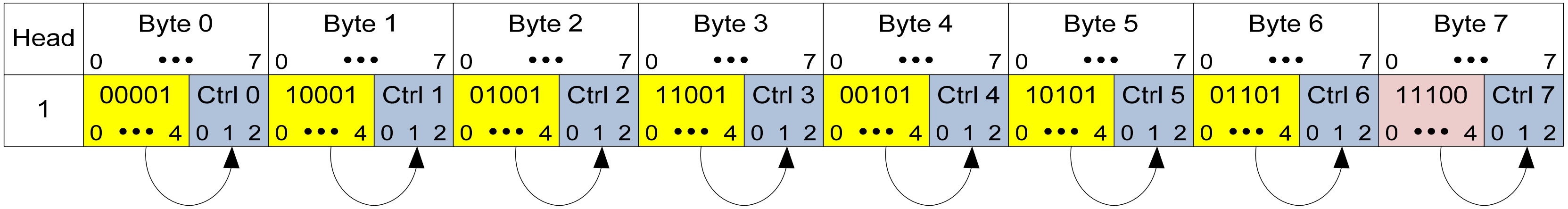


Head	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
	0 ••• 7	0 ••• 7	0 ••• 7	0 ••• 7	0 ••• 7	0 ••• 7	0 ••• 7	0 ••• 7

1	10001	Data 0	Ctrl 1	11001	Data 2	Ctrl 3	00101	Ctrl 4	01100	Data 5	Ctrl 6	Data 7
	0 ••• 4	0 ••• 7	0 1 2	0 ••• 4	0 ••• 7	0 1 2	0 ••• 4	0 1 2	0 ••• 4	0 ••• 7	0 1 2	0 ••• 7

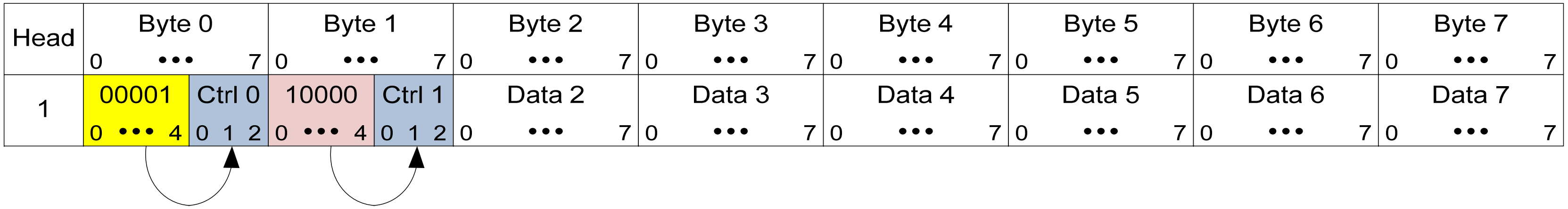
More Examples

All control codes



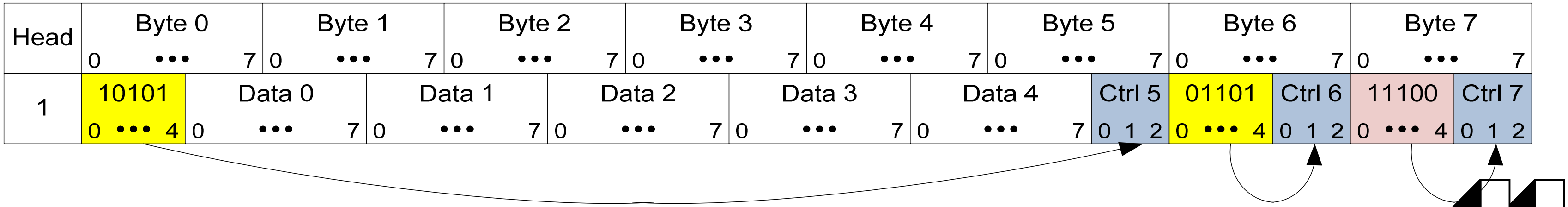
Head	Byte 0		Byte 1		Byte 2		Byte 3		Byte 4		Byte 5		Byte 6		Byte 7	
	0 ••• 7		0 ••• 7		0 ••• 7		0 ••• 7		0 ••• 7		0 ••• 7		0 ••• 7		0 ••• 7	
1	00001 0 ••• 4	Ctrl 0 0 1 2	10001 0 ••• 4	Ctrl 1 0 1 2	01001 0 ••• 4	Ctrl 2 0 1 2	11001 0 ••• 4	Ctrl 3 0 1 2	00101 0 ••• 4	Ctrl 4 0 1 2	10101 0 ••• 4	Ctrl 5 0 1 2	01101 0 ••• 4	Ctrl 6 0 1 2	11100 0 ••• 4	Ctrl 7 0 1 2

Start of packet on byte 2



Head	Byte 0		Byte 1		Byte 2		Byte 3		Byte 4		Byte 5		Byte 6		Byte 7	
	0 ••• 7		0 ••• 7		0 ••• 7		0 ••• 7		0 ••• 7		0 ••• 7		0 ••• 7		0 ••• 7	
1	00001 0 ••• 4	Ctrl 0 0 1 2	10000 0 ••• 4	Ctrl 1 0 1 2	Data 2 0 ••• 7		Data 3 0 ••• 7		Data 4 0 ••• 7		Data 5 0 ••• 7		Data 6 0 ••• 7		Data 7 0 ••• 7	

End of packet on byte 4



Head	Byte 0		Byte 1		Byte 2		Byte 3		Byte 4		Byte 5		Byte 6		Byte 7	
	0 ••• 7		0 ••• 7		0 ••• 7		0 ••• 7		0 ••• 7		0 ••• 7		0 ••• 7		0 ••• 7	
1	10101 0 ••• 4	Data 0 0 ••• 7		Data 1 0 ••• 7		Data 2 0 ••• 7		Data 3 0 ••• 7		Data 4 0 ••• 7		Ctrl 5 0 1 2	01101 0 ••• 4	Ctrl 6 0 1 2	11100 0 ••• 4	Ctrl 7 0 1 2

GMII Control Code Mapping

- **3 bit control code**

Control Code[0:2]	GMII Transmit	GMII Receive
001	Transmit Error Propagation	Data Reception Error
010	Normal Inter-Frame	Normal Inter-Frame
101	Assert Low Power Idle	Assert Low Power Idle
else	Reserved	Reserved

Error handling

- Encoding sensitive to bit errors. Need some form of FEC after encoding
- If FEC cannot correct errors then FEC marks all 8N/(8N+1) blocks as errors. If not done then:
- Unobservable errors
  - Header bit corrupted to 0
  - Data byte corrupted
  - Pointer corrupted to another valid pointer
  - Control code corrupted to another valid control code
  - Hopefully CRC on MAC layer will catch
- Observable errors
  - Pointer pointing to byte that does not exist
  - Pointer not pointing to later byte
  - Invalid control code
  - 8N/(8N+1) block marked as error when observable error detected

Possible Extensions

- Can extend concept to N = 1 to 32 if we use 6 bit pointers and 2 bit control code for a max of 256/257

Formal Encoder Definition (5 bit pointer)

- Define:
  - N = number of GMII bytes encoded into block
  - Bytes numbered n = 0, 1, 2, …, N-1. Byte 0 is the first one presented on GMII.
  - TC[n] = 0 if byte n is data byte on GMII, 1 if byte n is control byte on GMII
  - TC[-1] = 1 by definition
  - TD[n][0:7] = GMII byte n TXD[0:7] if TC[n] = 0
  - TD[n][5:7] = 010 – IPG, 101 – LPI, 001 – TX Error if TC[n] = 1. TD[n][0:4] is undefined.
  - B[0:8N] is the 8N+1 block. Bit 0 transmitted first.
  - OR(p) = Bitwise OR of TC[p:N-1]
  - NEXT(p)[0:3] = bit position of lowest bit in TC[p:N-1] that is a 1. Bit 3 is MSB.
  - NEXT(p)[4] = 0 if Bitwise SUM of TC[p:N-1] = 1, else 1

- B[0] = OR(0)
- B[8n+1:8n+4] =
  - TD[n][0:3] – if OR(n) = 0
  - NEXT(n)[0:3] – if OR(n) = 1 AND TC[n-1] = 1
  - TD[n-1][3:6] – if OR(n) = 1 AND TC[n-1] = 0

- B[8n+5] =
  - TD[n][4] – if OR(n) = 0
  - NEXT(n)[4] – if OR(n) = 1 AND TC[n-1] = 1
  - TD[n-1][7] – if OR(n) = 1 AND TC[n-1] = 0
- B[8n+6:8n+8] =
  - TD[n][5:7] – if OR(n) = 0
  - TD[n][5:7] – if OR(n) = 1 AND TC[n] = 1
  - TD[n][0:2] – if OR(n) = 1 AND TC[n] = 0

THANK YOU