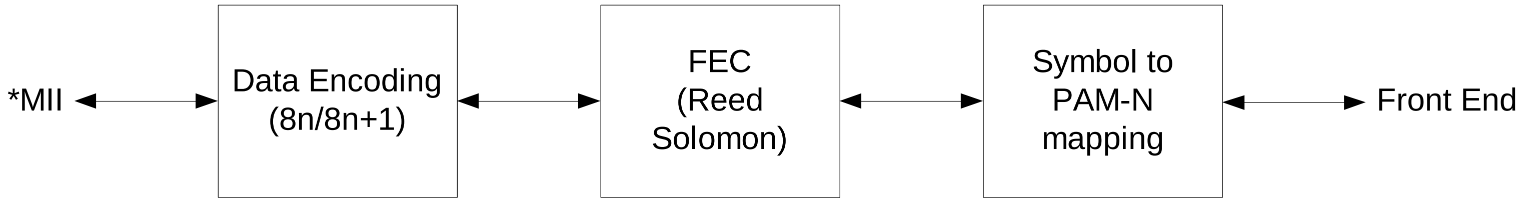
MGBASE-T1 PCS/PMA

William Lo, Axonne Inc.

July 9, 2018

General Approach to M-Gig PCS Definition

- Leverage 1000BASE-T1 architecture
- Elements in the architecture should be similar but should not be constrained to be identical



- Implementation friendly – as long as it meets performance

Data Encoding (8n/8n+1)

- 80/81 encoding not compatible for Mgig
  - See McClellan_3ch_01_0817.pdf page 5

1000BASE-T1

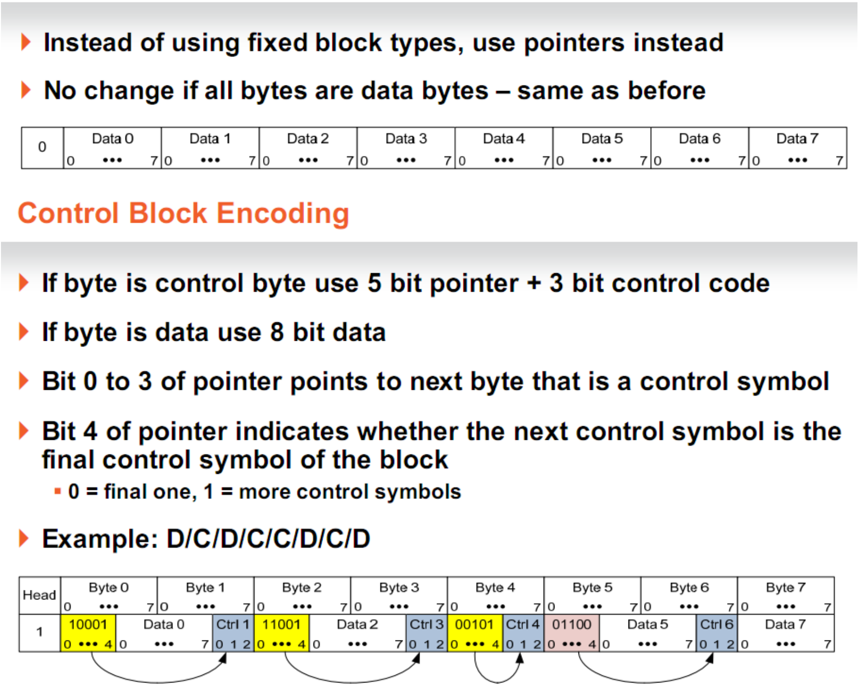
- GMII – 1 byte
- 80/81 Coding
- n=10
- 10 GMII bytes into 81 bits
- n not divisible by 4

MGBASE-T1

- XGMII – 4 bytes (Clause 46)
- 128/129 Coding
- n = 16
- 4 XGMII words into 129 bits
  - 4 x 4 bytes = 16 bytes
- Need to define XGMII control code mapping

8n/8n+1 coding

- See Lo_3bp_02_0314.pdf for more details
- Pick n = 16 ➔ 128/129
- Good tradeoff between efficiency and latency vs 64/65 and 512/513



- Instead of using fixed block types, use pointers instead
- No change if all bytes are data bytes – same as before

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

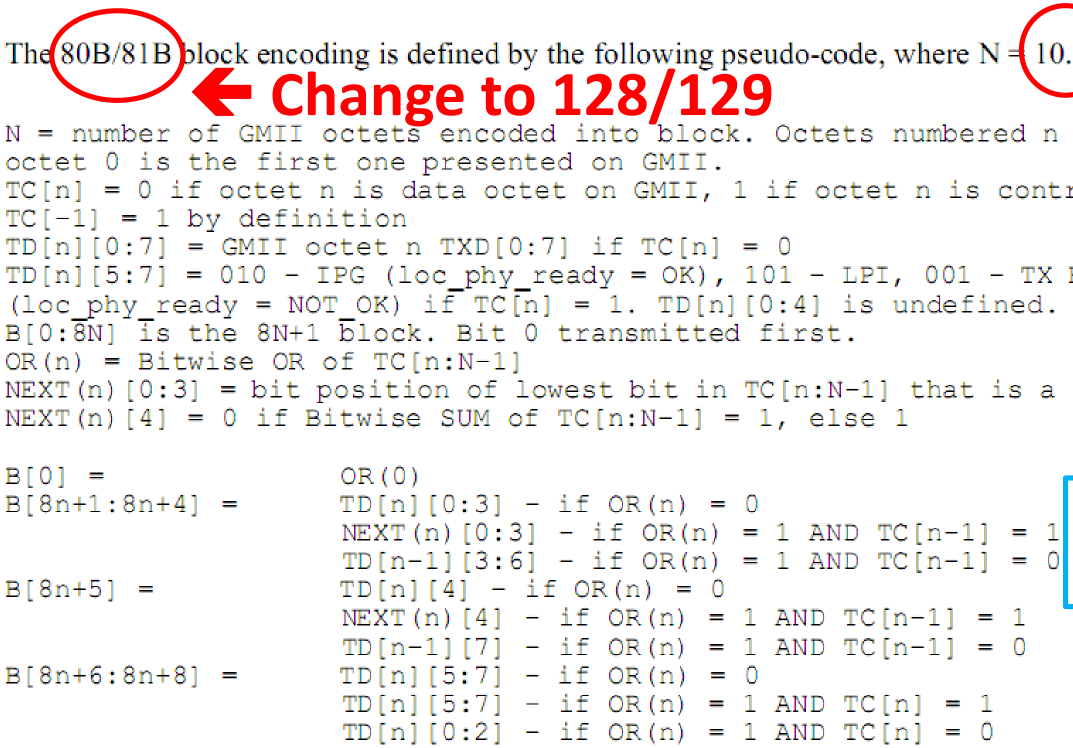
Control Block Encoding

- If byte is control byte use 5 bit pointer + 3 bit control code
- If byte is data use 8 bit data
- 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 0 ••• 7	Byte 1 0 ••• 7	Byte 2 0 ••• 7	Byte 3 0 ••• 7	Byte 4 0 ••• 7	Byte 5 0 ••• 7	Byte 6 0 ••• 7	Byte 7 0 ••• 7

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

Modification to 1000BASE-T1 Text for MGBASE-T1



The 80B/81B block encoding is defined by the following pseudo-code, where N = 10.

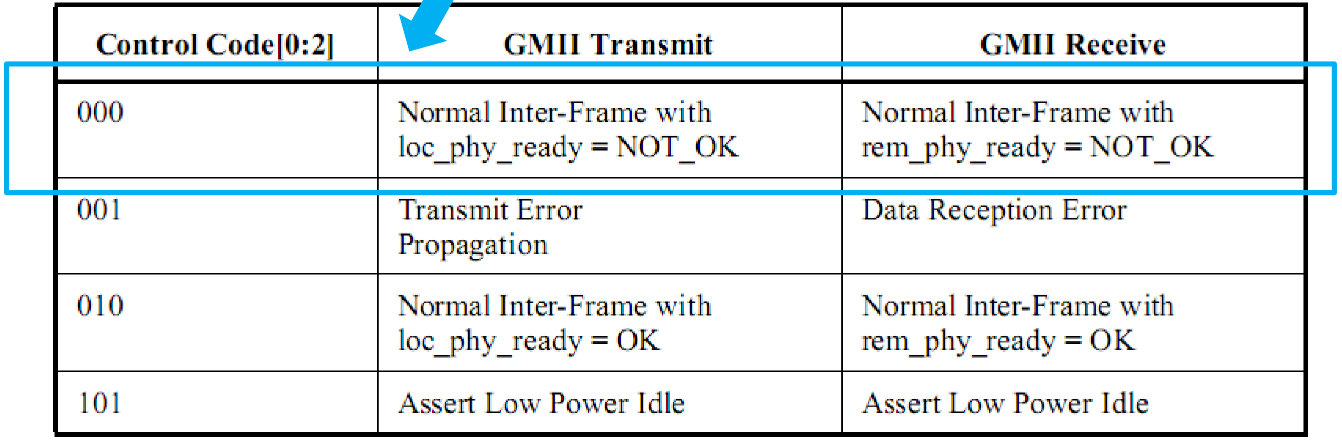
Change to 128/129

Change to 16

```
N = number of GMII octets encoded into block. Octets numbered n = 0,1,2,…,N-1.
octet 0 is the first one presented on GMII.
TC[n] = 0 if octet n is data octet on GMII, 1 if octet n is control octet on GMII
TC[-1] = 1 by definition
TD[n][0:7] = GMII octet n TXD[0:7] if TC[n] = 0
TD[n][5:7] = 010 - IPG (loc_phy_ready = OK), 101 - LPI, 001 - TX Error, 000 - IPG
(loc_phy_ready = NOT_OK) if TC[n] = 1. TD[n][0:4] is undefined.
B[0:8N] is the 8N+1 block. Bit 0 transmitted first.
OR(n) = Bitwise OR of TC[n:N-1]
NEXT(n)[0:3] = bit position of lowest bit in TC[n:N-1] that is a 1. Bit 3 is MSB.
NEXT(n)[4] = 0 if Bitwise SUM of TC[n: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
```

This internal message (not visible at the *GMII) is not defined in 64/65 or 512/513 blocks



Control Code[0:2]	GMII Transmit	GMII Receive
000	Normal Inter-Frame with loc_phy_ready = NOT_OK	Normal Inter-Frame with rem_phy_ready = NOT_OK
001	Transmit Error Propagation	Data Reception Error
010	Normal Inter-Frame with loc_phy_ready = OK	Normal Inter-Frame with rem_phy_ready = OK
101	Assert Low Power Idle	Assert Low Power Idle

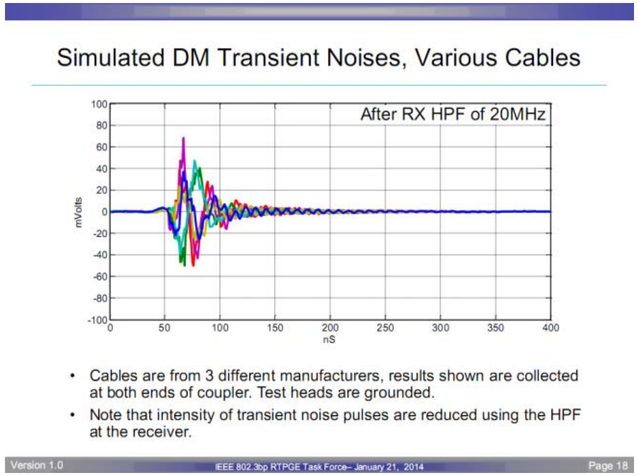
- **Define a control code for Sequence (011).**
- **Can also define Start and Terminate codes but not really needed as start of packet and end of packet can be inferred**

Reed Solomon Parameter Considerations

- Burst correction – most important
- Latency – important
- Fits integer number of 8n/8n+1 blocks plus OAM symbol – good for implementation

Burst Protection

- How many ns of burst protection is required?
  - More protection ➔ more latency, more hardware
- 1000BASE-T1 transient noise measurement and modeling:
  - Chini_Tazebay_3bp_01a_0114.pdf
  - Conclusion – at least 50ns burst protection.


Simulated DM Transient Noises, Various Cables

- Cables are from 3 different manufacturers, results shown are collected at both ends of coupler. Test heads are grounded.
- Note that intensity of transient noise pulses are reduced using the HPF at the receiver.

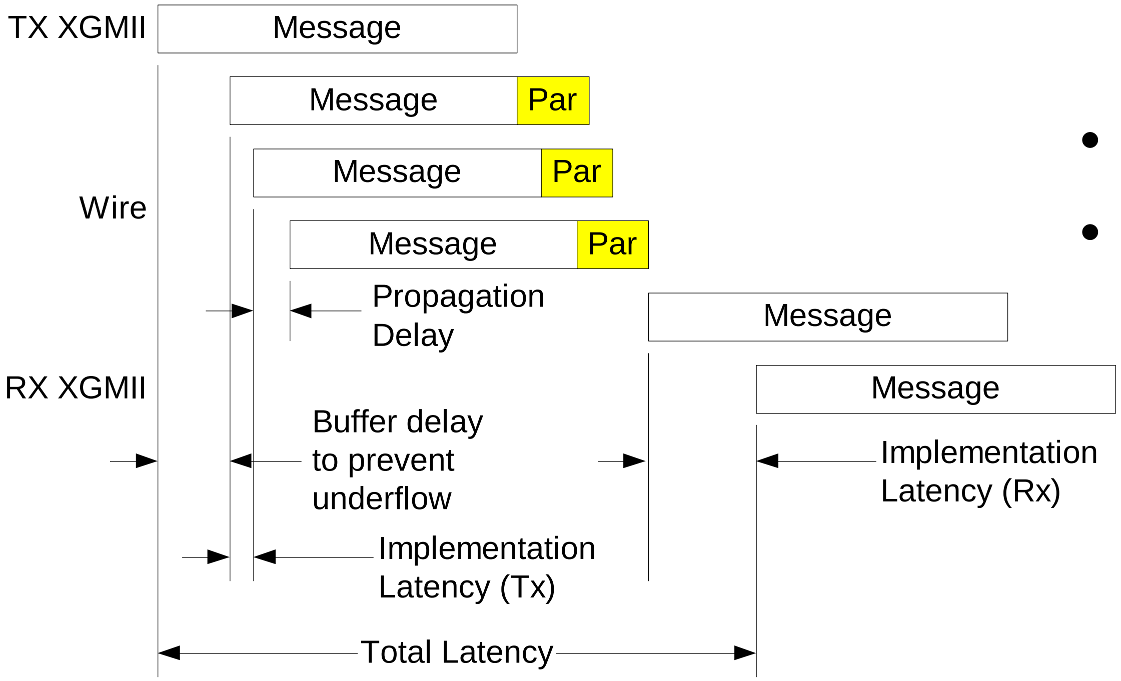
Version 1.0 IEEE 802.3bp RTPGE Task Force– January 21, 2014 Page 18

- Need similar study for MGBASE-T1
  - Is it still 50ns as assumed in tu_3ch_01b_0518.pdf page 12?

Reed Solomon Tradeoffs

- Burst Protection
- Latency
- Complexity – bits/symbol and # parity symbols
- Line rate – i.e. Coding overhead

Intrinsic Latency Definition

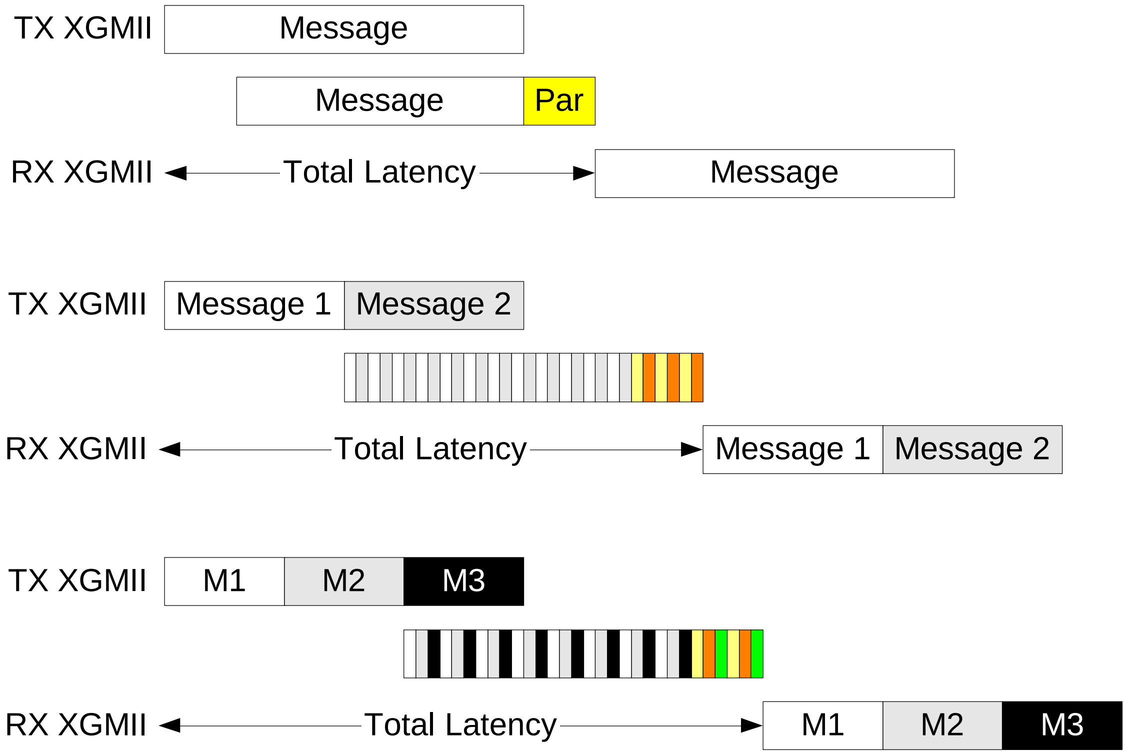


- Intrinsic Latency – Latency that is a consequence of the chosen FEC structure, impossible to eliminate
- Implementation Latency
  - Real circuits cannot run at infinite speeds
  - Variable among implementations
- Propagation Delay
- Implementation latency and propagation delay assumed to be zero for purposes of following discussion

* Implementation latency is distributed both inside and outside the FEC. It is shown here as lumped delay for clarity.

Interleaving Increases Intrinsic Latency on Tx side

- Implementation latency and propagation delay not included (assume 0)



Some Numbers

- Assume at least 120ns protection (i.e. protect error propagation from 50ns noise hit)

m	n	k	inter-leave	Parity n-k	Coding Overhead	#128/129 blocks	OAM/ spare	total burst protection (ns)	TX + RX intrinsic latency (ns)	protection (ns)	RS frame duration (ns)
9	432	388	7	44	12.5%	27	1	123.2	4492.8	17.6	345.6
10	1008	904	3	104	12.5%	70	1	138.7	4480.0	46.2	896.0
10	720	646	4	74	12.5%	50	1	131.6	4480.0	32.9	640.0
11	1296	1162	2	134	12.5%	99	1	131.0	3801.6	65.5	1267.2
11	864	776	3	88	12.5%	66	2	129.1	4224.0	43.0	844.8
12	2304	2066	1	238	12.5%	192	2	126.9	2711.5	126.9	2457.6
12	1152	1034	2	118	12.5%	96	2	125.9	3686.4	62.9	1228.8
12	768	690	3	78	12.5%	64	2	124.8	4096.0	41.6	819.2
9	480	388	4	92	25.0%	27	1	132.5	2419.2	33.1	345.6
10	800	646	2	154	25.0%	50	1	123.2	1920.0	61.6	640.0
10	640	518	3	122	25.0%	40	2	146.4	2560.0	48.8	512.0
10	480	388	4	92	25.0%	30	1	147.2	2688.0	36.8	384.0
11	1440	1162	1	278	25.0%	99	1	122.3	1511.8	122.3	1267.2
11	800	646	2	154	25.0%	55	1	135.5	2112.0	67.8	704.0
11	480	388	3	92	25.0%	33	1	121.4	2112.0	40.5	422.4
12	1440	1162	1	278	25.0%	108	1	133.4	1649.3	133.4	1382.4
12	800	646	2	154	25.0%	60	1	147.8	2304.0	73.9	768.0
12	480	389	3	91	25.0%	36	2	131.0	2304.0	43.7	460.8

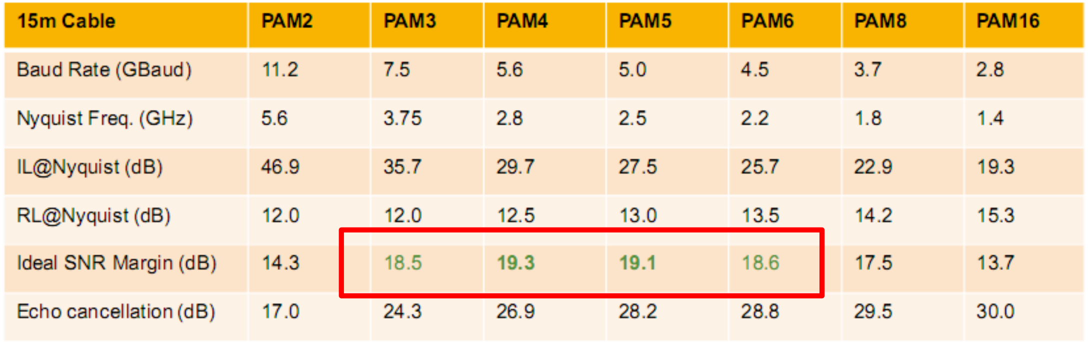
Some observations

- Can we tolerate at least 4.5us latency in 10G?
  - This is longer than 1000BASE-T1 (3.95us) (including implementation latency spec limit is 7.168us)
  - 10GBASE-T latency limit is 2.56us for everything (excluding propagation delay).
  - If yes then m=9 or 10 are good candidates with interleaving
- Lowest latency at 12.5% overhead is 2.71us (2304, 2076) 12-bit no interleave
- If we accept 25% overhead instead of 12.5% then latency can be cut by roughly half for same burst protection
  - Higher baud rate tradeoff against more symbol protection in FEC

Some observations

- RS(432, 388) 9-bit interleaved is an interesting case
  - Functionally a shortened RS(450, 406) used in 1000BASE-T1
  - Both have 44 parity symbols
  - Same FEC implementation except one processes 450 symbols and the other 432.
- Burst noise duration in ns probably similar for 2.5G, 5G, and 10G speeds
  - Can have different interleaving for different speeds i.e. none, 2x, 4x

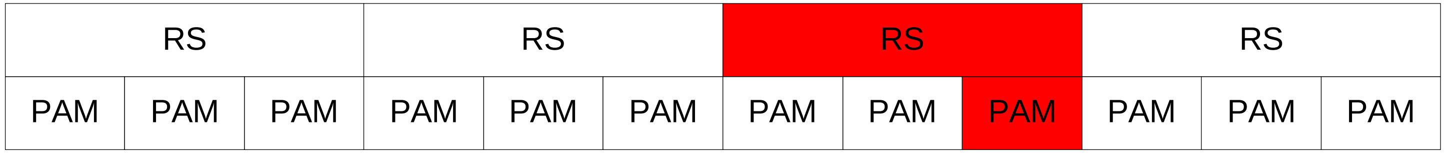
PAM Selection



15m Cable	PAM2	PAM3	PAM4	PAM5	PAM6	PAM8	PAM16
Baud Rate (GBaud)	11.2	7.5	5.6	5.0	4.5	3.7	2.8
Nyquist Freq. (GHz)	5.6	3.75	2.8	2.5	2.2	1.8	1.4
IL@Nyquist (dB)	46.9	35.7	29.7	27.5	25.7	22.9	19.3
RL@Nyquist (dB)	12.0	12.0	12.5	13.0	13.5	14.2	15.3
Ideal SNR Margin (dB)	14.3	18.5	19.3	19.1	18.6	17.5	13.7
Echo cancellation (dB)	17.0	24.3	26.9	28.2	28.8	29.5	30.0

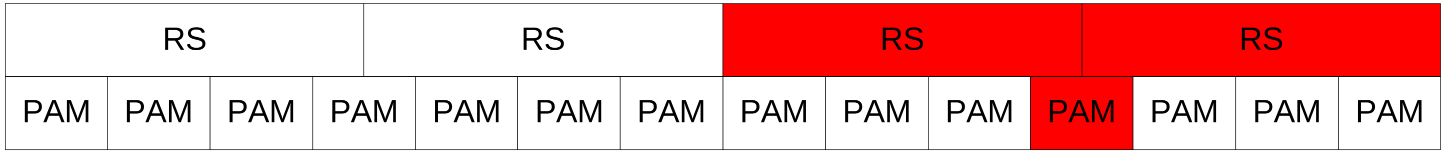
Pandey_3ch_01c_0518.pdf

- PAM 3, 4, 5, 6 roughly the same performance
- All else being equal pick the lower baud rate
  - slower circuits, fewer filter taps
- Ideally integer number of PAM symbols per RS symbol
  - Corruption of PAM symbol only corrupts one RS Symbol



RS			RS			RS			RS		
PAM	PAM	PAM	PAM	PAM	PAM	PAM	PAM	PAM	PAM	PAM	PAM

  - Non-integer may corrupt more than one



<table>
<tr><td colspan="7">RS</td><td colspan="7">RS</td><td colspan="7">RS</td><td colspan="7">RS</td></tr>
<tr><td colspan="2">PAM</td><td colspan="2">PAM</td><td colspan="2">PAM</td><td colspan="2">PAM</td><td colspan="2">PAM</td><td colspan="2">PAM</td><td colspan="2">PAM</td><td colspan="2">PAM</td><td colspan="2">PAM</td><td colspan="2">PAM</td><td colspan="2">PAM</td><td colspan="2">PAM</td><td colspan="2">PAM</td><td colspan="2">PAM</td></tr>
</table>

- RS(n, k) can only correct up to (n-k)/2 symbols

Matching PAM level with RS Selection

- If m=9 chosen then PAM3 or PAM5 matches best
- If m = 10 chosen then PAM4 or PAM6 matches best

		Line	baud / RS symbol	
	bits/baud	Gbaud/s	m=9	m=10
PAM 3	1.50	7.500	6.00	6.67
PAM 4	2.00	5.625	4.50	5.00
PAM 5	2.25	5.000	4.00	4.44
PAM 6	2.50	4.500	3.60	4.00

- Pick the slower speed
  - m = 9 then PAM5
  - m = 10 then PAM6

Recommendations and Actions

- Adopt 8n/8n+1 encoding where n=16 (128/129)
- Determine max burst to protect
  - Don’t over specify this
  - Hardware gets more expensive to implement as n-k gets larger
  - Latency increases with longer RS frame or more interleaving
- How much latency can the system tolerate?
  - Rough estimate add 30% to 50% to the latency numbers on page 11 to account for FEC implementation latency and other latencies outside the FEC.
- Narrow down PAM selection to be either PAM 5 or PAM 6

THANK YOU