

---

# Coding schemes for 1394B

Alistair Coles

Presented to P1394B  
June 10/11 1997

# Note on Error Detection

- 1394 employs a 32 bit CRC to detect errors in data fields.

Speed	Max payload (bytes) - async. packet	CRC32 will detect...
S100	512	all combinations of up to <i>three</i> single error events .
S200	1024	
S400	2048	
S800	4096	
S1600	8192	
S3200	16384	all combinations of up to <i>two</i> single error events.

- Error detection can be hampered or improved by coding scheme.

# FDDI 4B5B code

- Code is not d.c. balanced.

e.g. consecutive data nibbles [0000] result in pattern:

[11110][11110]..-> NRZI -> ..0][10100][10100]..

e.g. proposed TX\_PARENT\_NOTIFY (L symbol) results in:

[00101][00101]..-> NRZI -> ..0][00110][00110]..

e.g. proposed no tpBias (QQ symbol) results in:

[00000][00000]..-> NRZI -> ..0][00000][00000]..

- Detection of all three single bit errors is not guaranteed.

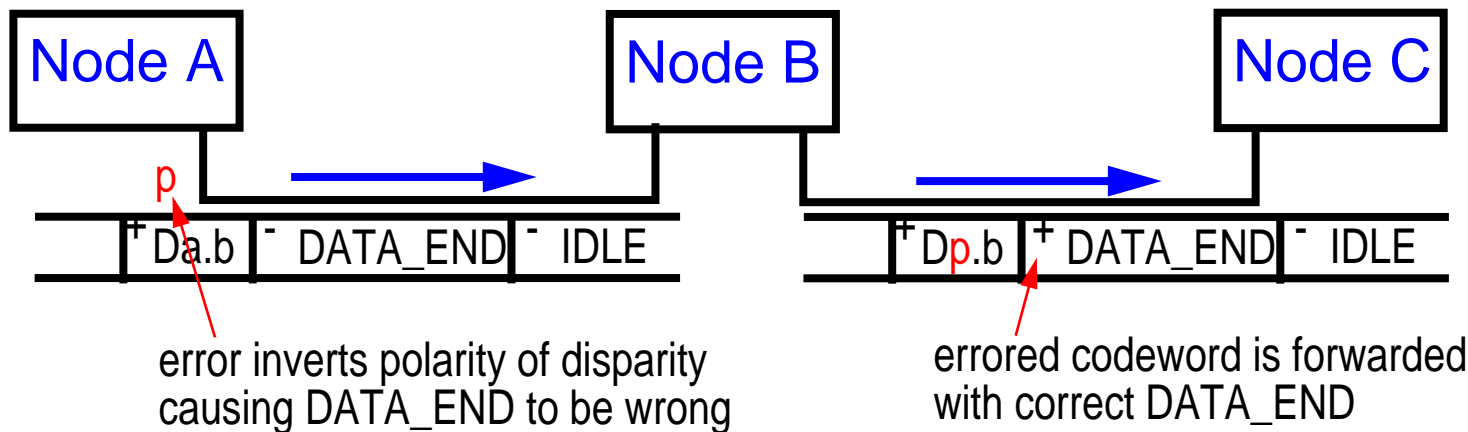
(“Error Characteristics of FDDI”, Raj Jain, IEEE Trans. Comms., 38, August 1990, p1244-1252)

- Control signals are repetitive resulting in discrete spectral lines e.g. proposed IDLE results in:

[11111][11111]..-> NRZI -> ..0][10101][01010]..

# IBM 8B10B code

- dc balanced: disparity at end of packet is indicated by DATA\_END packet.
- Any odd number of single error events during a packet will be detected since DATA\_END will appear incorrect.
- However, when the packet is re-coded and forwarded, this information is lost:



- Need a distinct DATA\_END\_ERROR signal to mark packets corrupted on a previous link: similar to FDDI frame status field and 802.12 Invalid Packet Delimiter.

---

# — IBM 8B10B code (cont.)

- Single error can change data codeword  $Dx.y$  to K28.5 (i.e. Hamming distance 1 between control and data):
  - possibility of false DATA\_END. This should be detected by checking Data Length field in packet header.
  - possibility of false “embedded” control signals.
- [K28.5,  $Dx.y$ ] structure for control signals introduces 16 bit latency.
- Repetitive [K28.5,  $Dx.y$ ] patterns for control signals result in discrete spectral lines (see Eric Deliot’s presentation).

---

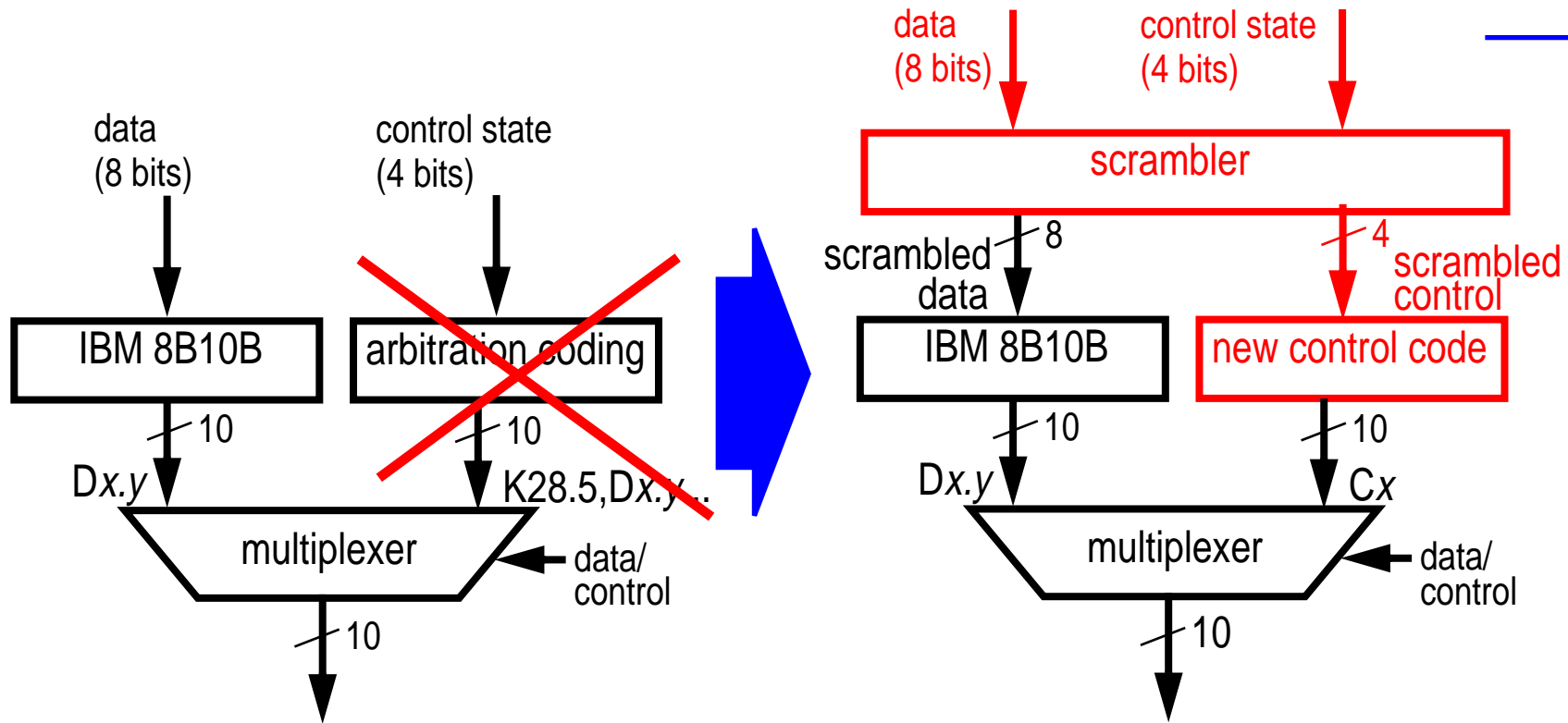
# — HP 8B10B code

- Control represented by single 10 bit codewords, which are Hamming distance 2 from data codewords.
- Compatible with use of scrambler -> good spectral properties.
- Spectral shaping similar to 3 level MLT3 code.
- Training/synchronization procedures described at last meeting.
- More complex implementation.
- New!

# — Comparison of coding schemes (1)

Code	Pros	Cons
FDDI 4B5B	Simple, low latency.	Error detection poor. Unconstrained disparity. No scrambler.
IBM 8B10B	Well known, simple. Good disparity control.	Large latency for control. No scrambler. Poor error detection with embedded control signals.
HP 8B10B	Good spectrum. Scrambler. Good error detection.	New. More complex.

# Modified IBM 8B10B scheme



- Replace 20 bit K28.5, Dx.y arbitration encoding with new set of sixteen 10-bit control codewords which are Hamming distance 2 from IBM data codeword set.
- Add a scrambler.

# New control codeword set

scrambled control state input			codeword
hex	binary	label	
0	0000	C0	0 0 0 0 0 1 1 1 1 1
1	0001	C1	0 0 0 0 1 0 1 1 1 1
2	0010	C2	0 0 0 0 1 1 1 1 1 0
3	0011	C3	0 0 0 1 0 0 1 1 1 1
4	0100	C4	0 0 1 0 0 0 1 1 1 1
5	0101	C5	1 1 0 0 0 0 0 1 1 1
6	0110	C6	0 1 0 0 0 0 1 1 1 1
7	0111	C7	1 0 0 0 0 0 1 1 1 1
8	1000	C8	0 1 1 1 1 1 0 0 0 0
9	1001	C9	1 0 1 1 1 1 0 0 0 0
a	1010	C10	0 0 1 1 1 1 1 0 0 0
b	1011	C11	1 1 0 1 1 1 0 0 0 0
c	1100	C12	1 1 1 0 1 1 0 0 0 0
d	1101	C13	1 1 1 1 0 0 0 0 0 1
e	1110	C14	1 1 1 1 0 1 0 0 0 0
f	1111	C15	1 1 1 1 1 0 0 0 0 0

All zero disparity.

Max. run length is 10 (cf. 5 with IBM data codewords)

Hamming distance 6 between subsets of codewords with two MSBs of input inverted

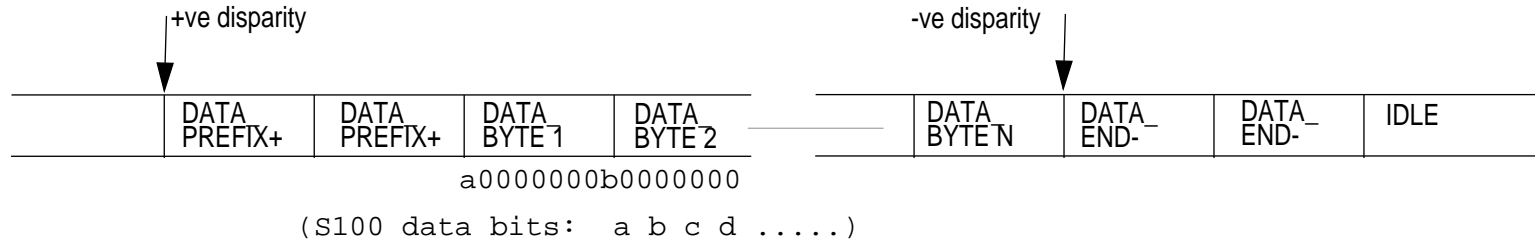
# Control State Mapping

1394 state	Binary control state
IDLE	0000
REQUEST / GRANT	0001
PARENT_NOTIFY	0010
CHILD_NOTIFY	0011
SPEED_SIGNAL (rds +ve)	0100
DATA_PREFIX+ (rds -ve)	0101
DATA_END- (rds -ve)	0110
DATA_END+E (Errored packet, rds +ve)	0111

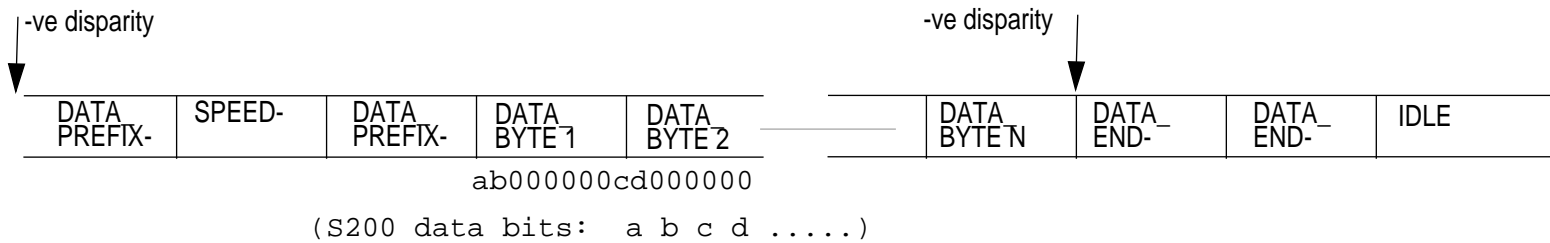
1394 state	Binary control state
SPEED- (rds -ve)	1000
DATA_PREFIX+ (rds +ve)	1001
DATA_END+ (rds +ve)	1010
DATA_END-E (Errored packet, rds -ve)	1011
spare	1100
spare	1101
spare	1110
RESET	1111

- Hamming distance 6 between +ve and -ve rds (disparity) versions of control states.
- Hamming distance 6 between SPEED\_SIGNAL and DATA\_PREFIX.

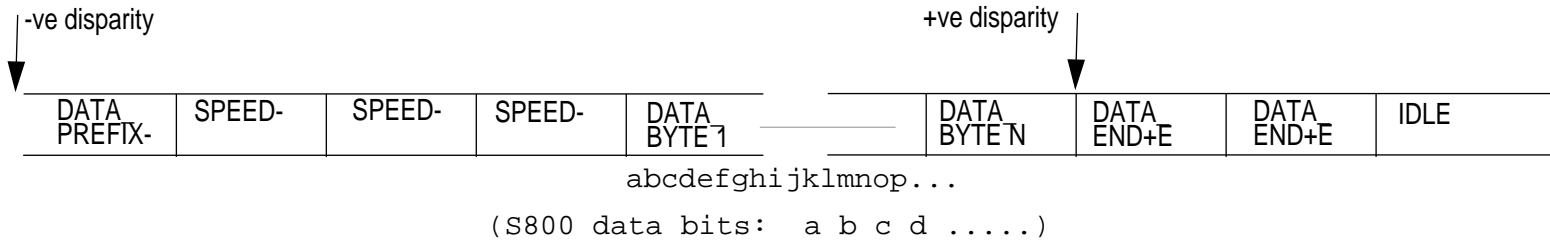
# Example packet structures



**S100 packet transported on S800 1394B link**

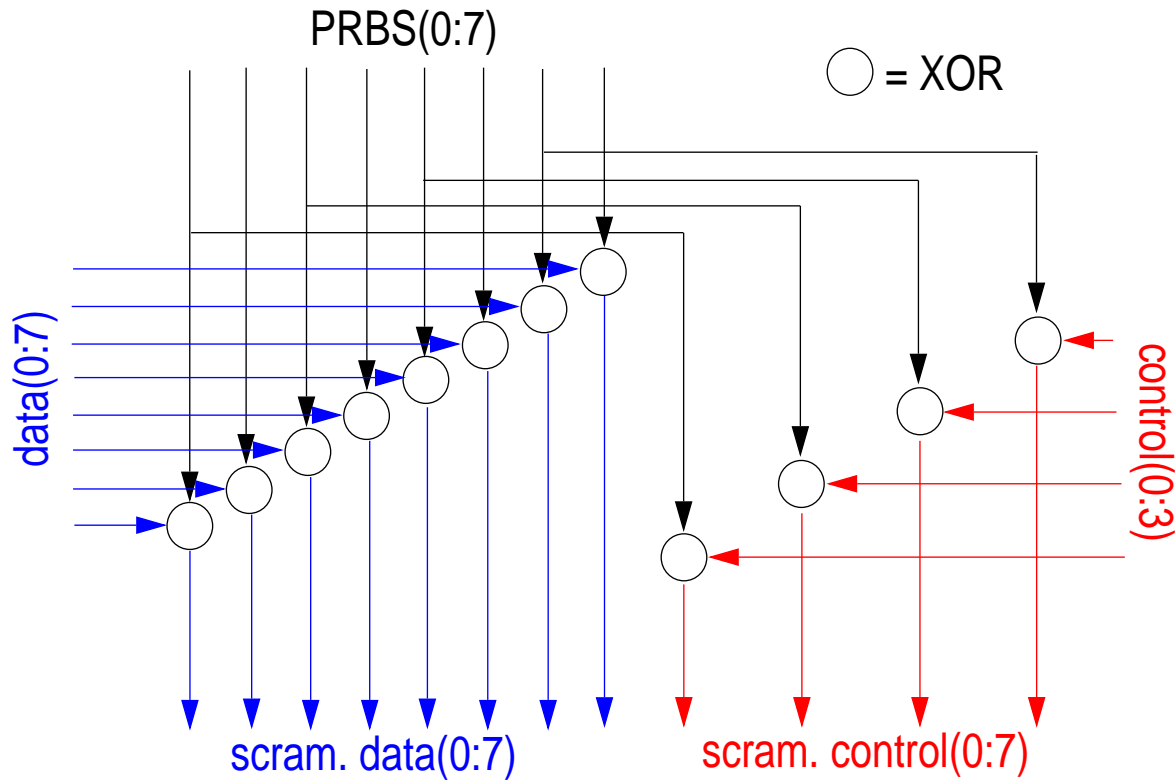


**S200 packet transported on S800 1394B link**



**Errored S800 packet forwarded on S800 1394B link**

# Scrambler operation



- Use alternate bits of PRBS output to scramble control.

# Example of coding

1394-1995 state	REQUEST	DATA_PREFIX				data....	
1394-1995 data						01110101	00110111
rds	+	+	+	+	+	+	+
control	REQUEST	DAT_PREF+	SPEED+	SPEED+	SPEED+		
control state	0 0 0 1	0 1 0 1	0 1 0 0	0 1 0 0	0 1 0 0		
scrambler	00000100	00000010	10000001	00010000	10101010	01000000	01101000
scram cntrl	0 0 0 1	0 1 0 0	1 1 0 0	0 1 0 0	1 0 1 1		
scram data						00110101	01011111
codeword	C1	C6	C12	C8	C11	D21.1	D31.2
coded output	0000101111	0100001111	1110110000	0111110000	1101110000	1010101001	0101000101
1394-1995 state	....data	DATA_END			IDLE		
1394-1995 data	....	00111001					
rds	....	-	-	-	-	-	-
control	....		DATA_END-	DATA_END-	DATA_END-	IDLE	
control state	....		0 1 1 0	0 1 1 0	0 1 1 0	0 0 0 0	0 0 0 0
scrambler	....	00111001	00011011	10101110	10100010	10000101	00010010
scram cntrl	....		0 1 0 1	1 0 0 1	1 0 1 1	1 0 0 0	0 0 0 1
scram data	....	00000000					
codeword	....	D0.0	C5	C9	C11	C8	C1
coded output	....	1001110100	1100000111	1011110000	1101110000	0111110000	0000101111

# — Start-up procedure

- For start-up described by Colin Whitby-Strevens at May meeting, need S1, S2, and S3 patterns.
- Propose:

$$S1 = [K28.5 , D10.2] = [0011111010][0101010101]$$

$$S2 = [K28.5 , D21.5] = [0011111010][1010101010]$$

$$S3 = [K28.5 , K28.5] = [0011111010][1100000101]$$

(other possibilities exist)

- Synchronize scrambler on first occurrence of a Cx codeword.
- “Blind” codeword and scrambler synchronization is also possible i.e. receiver can deduce codeword and scrambler synchronization from transmitted IDLE.

# — Comparison of coding schemes (2)

Code	Pros	Cons
FDDI 4B5B	Simple, low latency.	Error detection poor. Unconstrained disparity. No scrambler.
IBM 8B10B	Well known, simple. Good disparity control.	Large latency for control. No scrambler. Poor error detection with embedded control signals.
Modified IBM 8B10B	Well known, simple. Good disparity control. Good spectrum. Scrambler. Good error detection. Shorter control patterns.	Some new parts. Max. run length = 10.
HP 8B10B	Good spectrum. Scrambler. Good error detection.	New. More complex.

# — Comparison of coding schemes (3)

	<b>FDDI 4B5B</b>	<b>IBM 8B10B</b>	<b>Modified IBM 8B10B</b>	<b>HP 8B10B</b>
max runlength	8	5	10	17
disparity	unbounded	+/-3	+/-6	+13/-11
three single error events detected in a packet	no	yes	yes	yes
Hamming distance control <-> data	1	1	2	2
length of control patterns	4/8	16	8	8
good control spectrum	no	no	yes	yes
complexity	low	low	low	medium
comma pattern	yes	yes	yes	yes