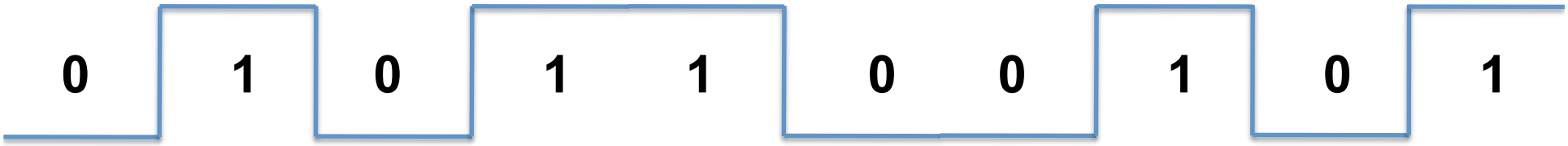
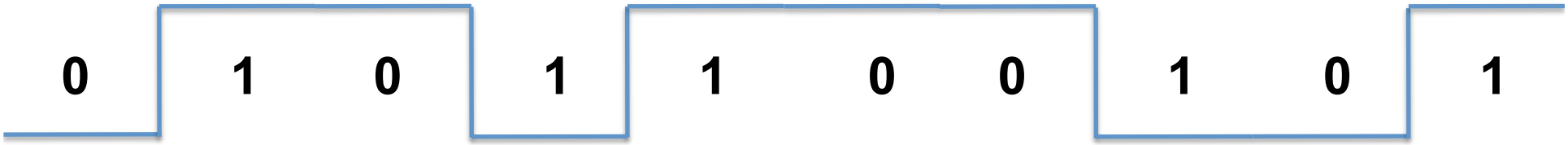
Line Coding Examples
where Baud=bit-rate

Non-Return-to-Zero (NRZ)

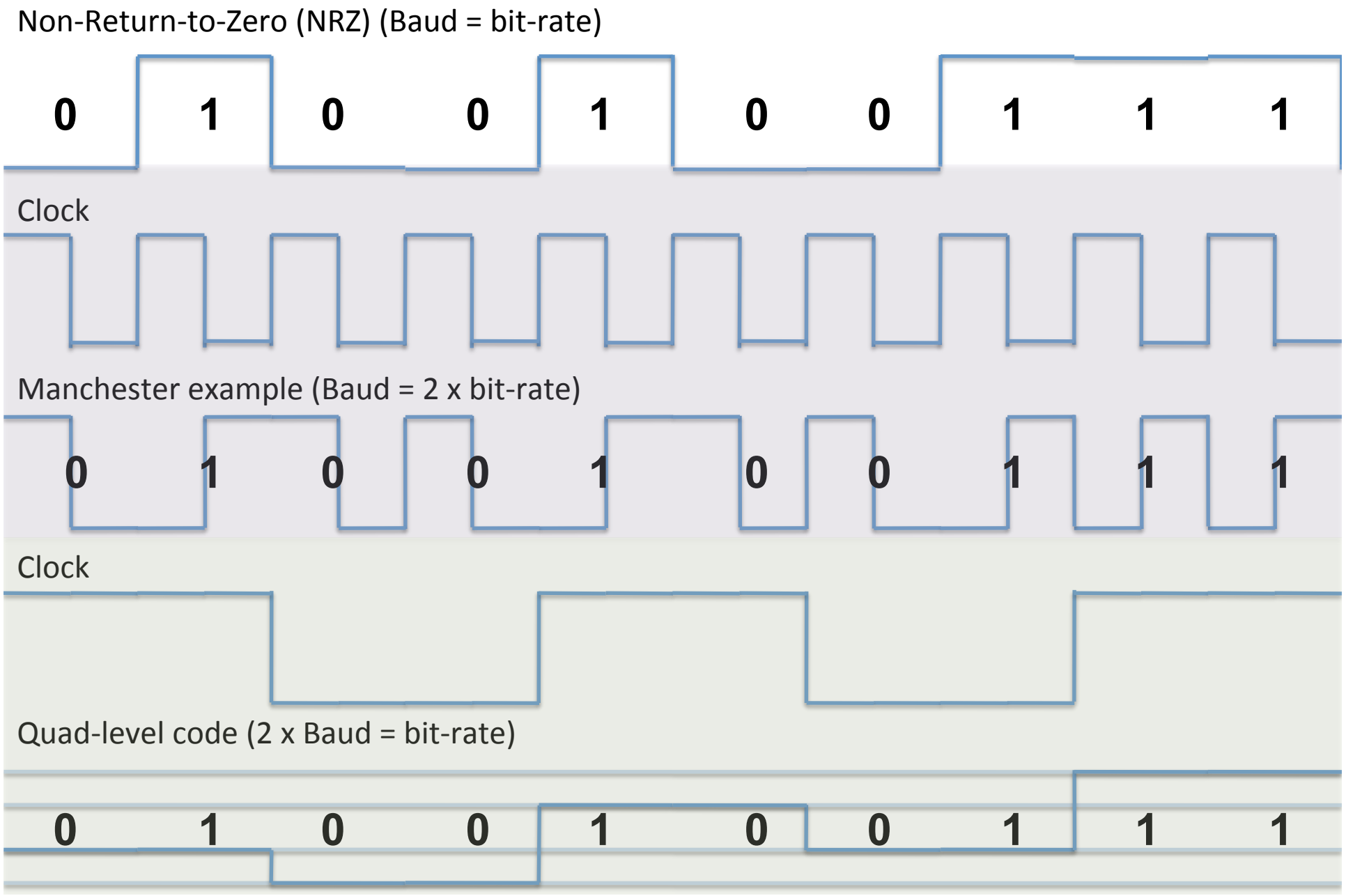


Non-Return-to-Zero-Mark (NRZM) 1 = transition 0 = no transition



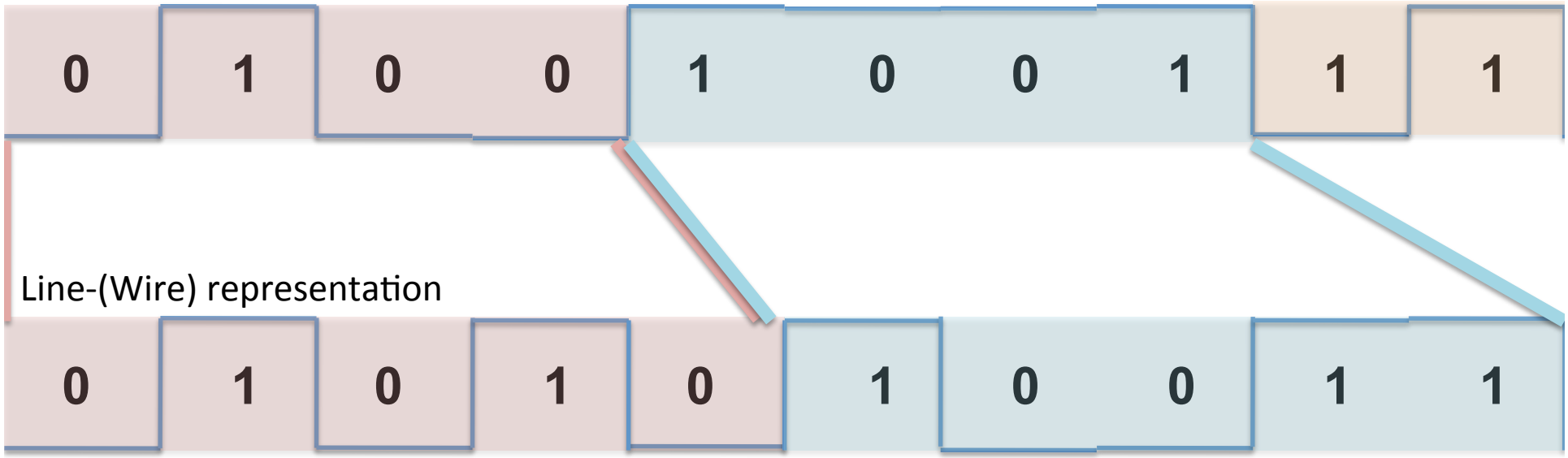
Non-Return-to-Zero Inverted (NRZI) (note transitions on the 1)

Line Coding Examples - II



Line Coding Examples - III

Data to send



Name	4b	5b	Description
0	0000	11110	hex data 0
1	0001	01001	hex data 1
2	0010	10100	hex data 2
3	0011	10101	hex data 3
4	0100	01010	hex data 4
5	0101	01011	hex data 5
6	0110	01110	hex data 6
7	0111	01111	hex data 7
8		1000	10010 hex data 8
9		1001	10011 hex data 9
A	1010	10110	hex data A
B	1011	10111	hex data B
C	1100	11010	hex data C
D	1101	11011	hex data D
E	1110	11100	hex data E
F	1111	11101	hex data F

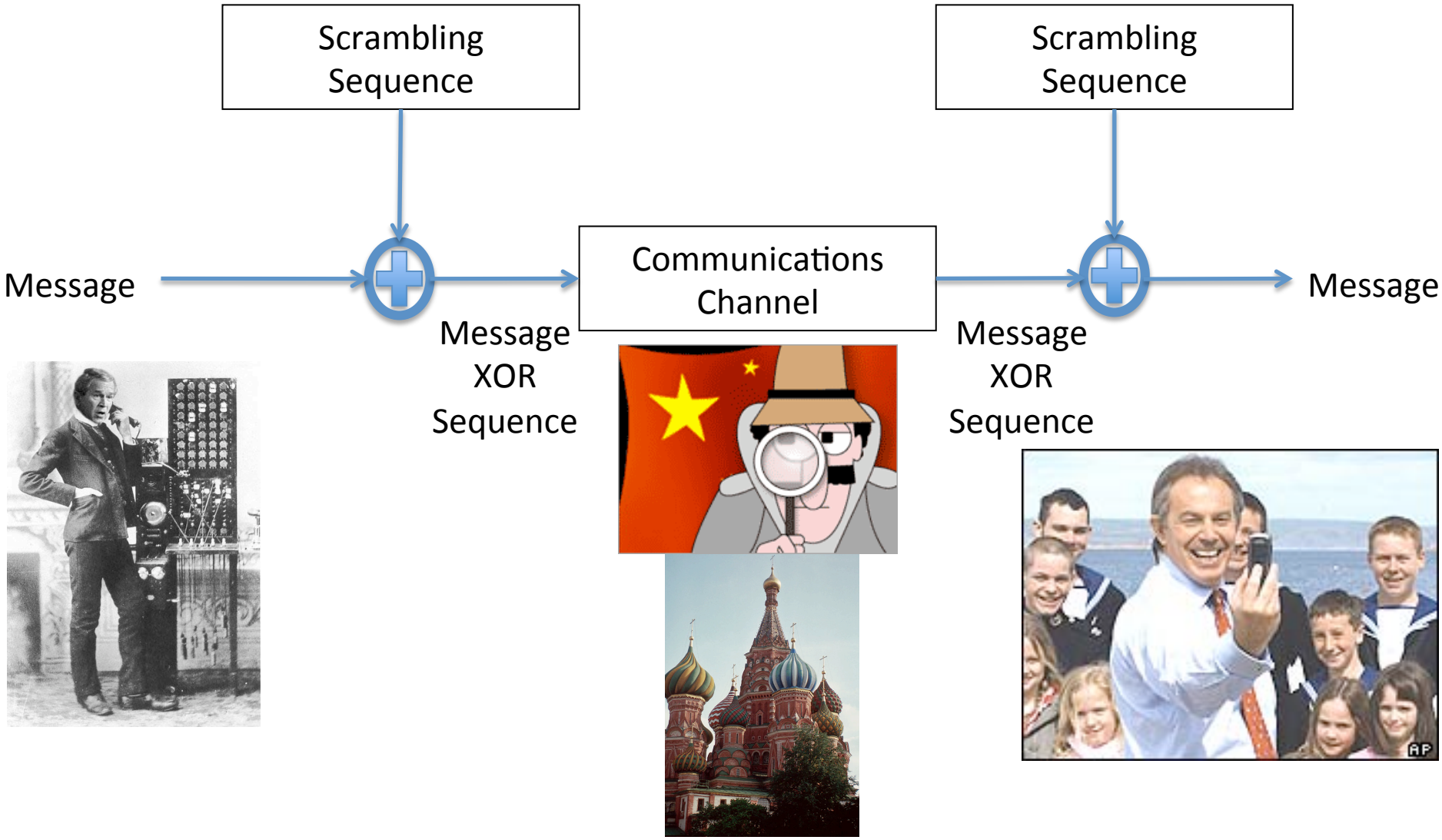
Name	4b	5b	Description
Q	-NONE-	00000	Quiet
I	-NONE-	11111	Idle
J	-NONE-	11000	SSD #1
K	-NONE-	10001	SSD #2
T	-NONE-	01101	ESD #1
R	-NONE-	00111	ESD #2
H	-NONE-	00100	Halt

Block coding transfers data with a fixed overhead: 20% less information per Baud in the case of 4B/5B

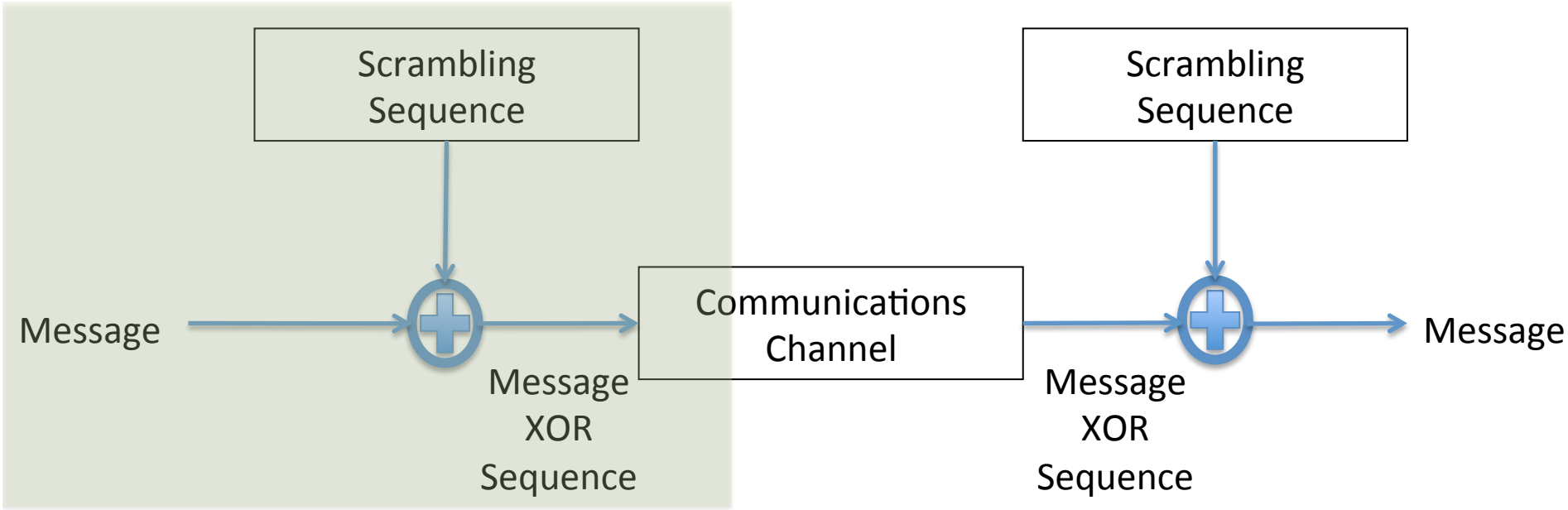
So to send data at 100Mbps; the line rate (the Baud rate) must be 125Mbps.

1Gbps uses an 8b/10b codec; encoding entire bytes at a time but with 25% overhead

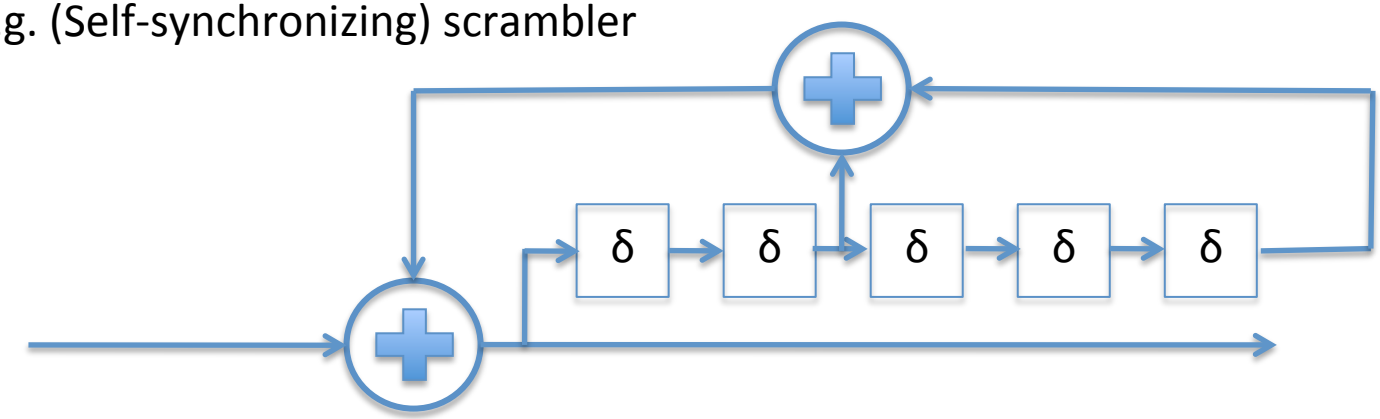
Line Coding Examples - IV



Line Coding Examples - IV

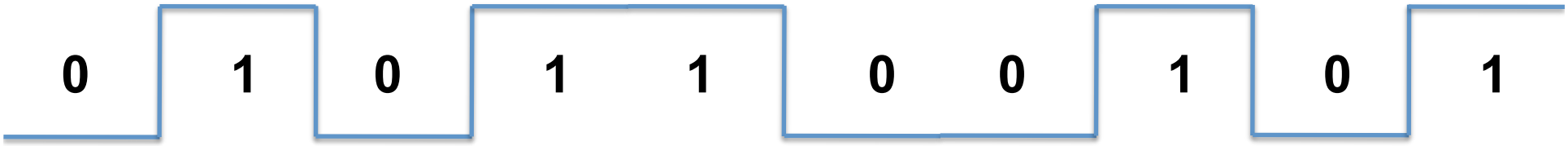


e.g. (Self-synchronizing) scrambler

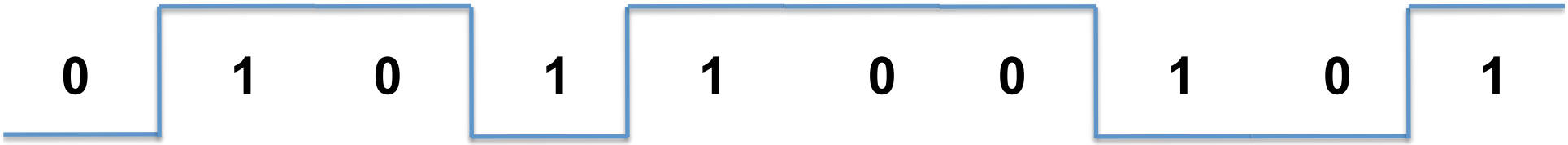


Line Coding Examples
where Baud=bit-rate

Non-Return-to-Zero (NRZ)

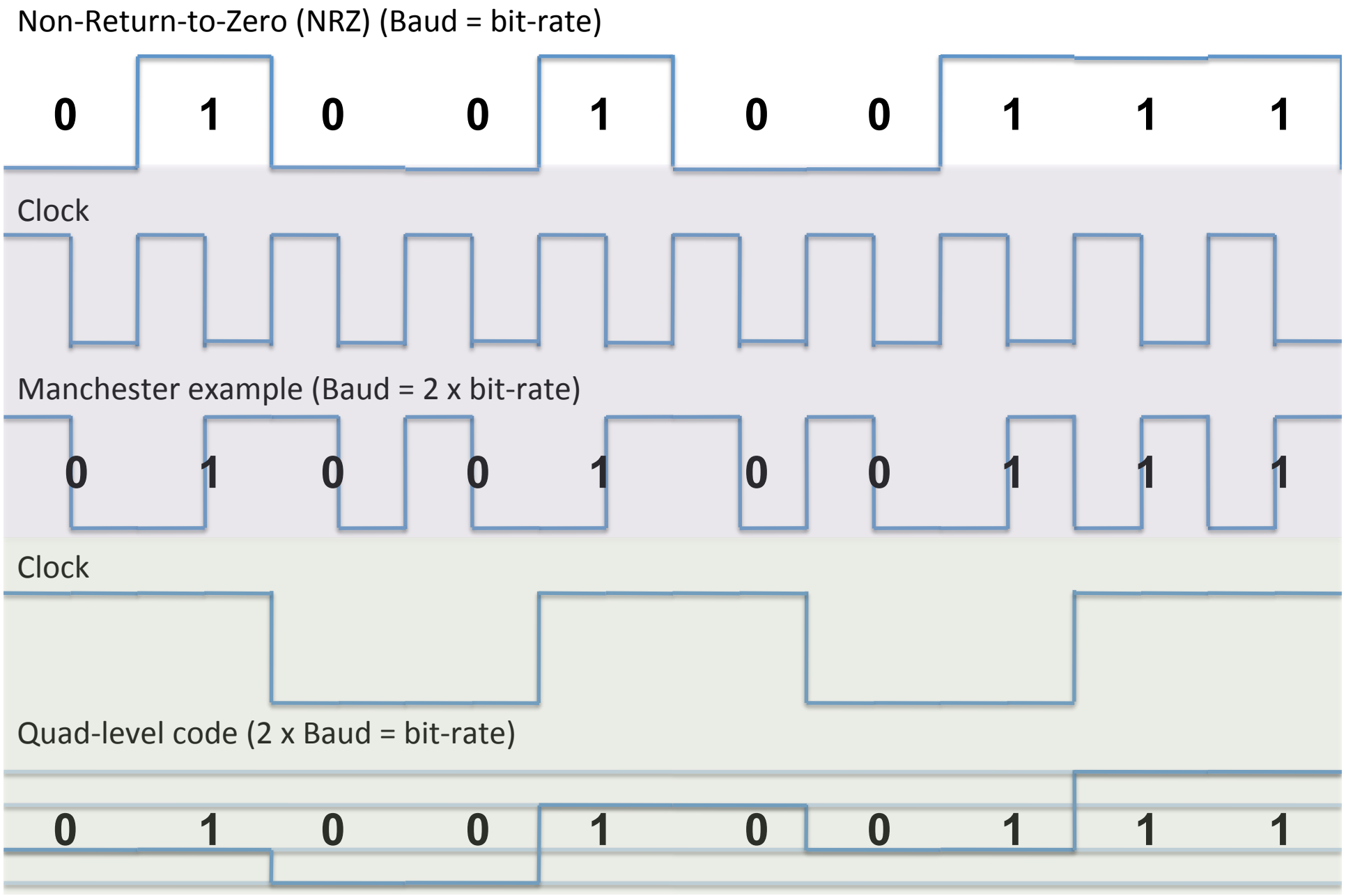


Non-Return-to-Zero-Mark (NRZM) 1 = transition 0 = no transition



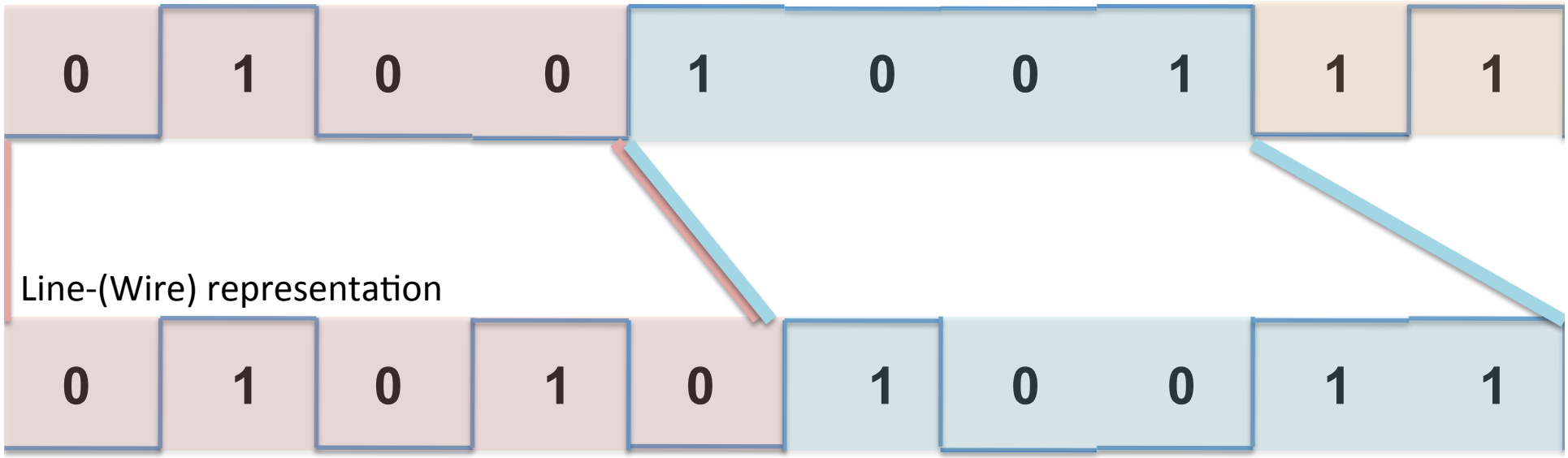
Non-Return-to-Zero Inverted (NRZI) (note transitions on the 1)

Line Coding Examples - II



Line Coding Examples - III

Data to send



Name	4b	5b	Description
0	0000	11110	hex data 0
1	0001	01001	hex data 1
2	0010	10100	hex data 2
3	0011	10101	hex data 3
4	0100	01010	hex data 4
5	0101	01011	hex data 5
6	0110	01110	hex data 6
7	0111	01111	hex data 7
8		1000	10010 hex data 8
9		1001	10011 hex data 9
A	1010	10110	hex data A
B	1011	10111	hex data B
C	1100	11010	hex data C
D	1101	11011	hex data D
E	1110	11100	hex data E
F	1111	11101	hex data F

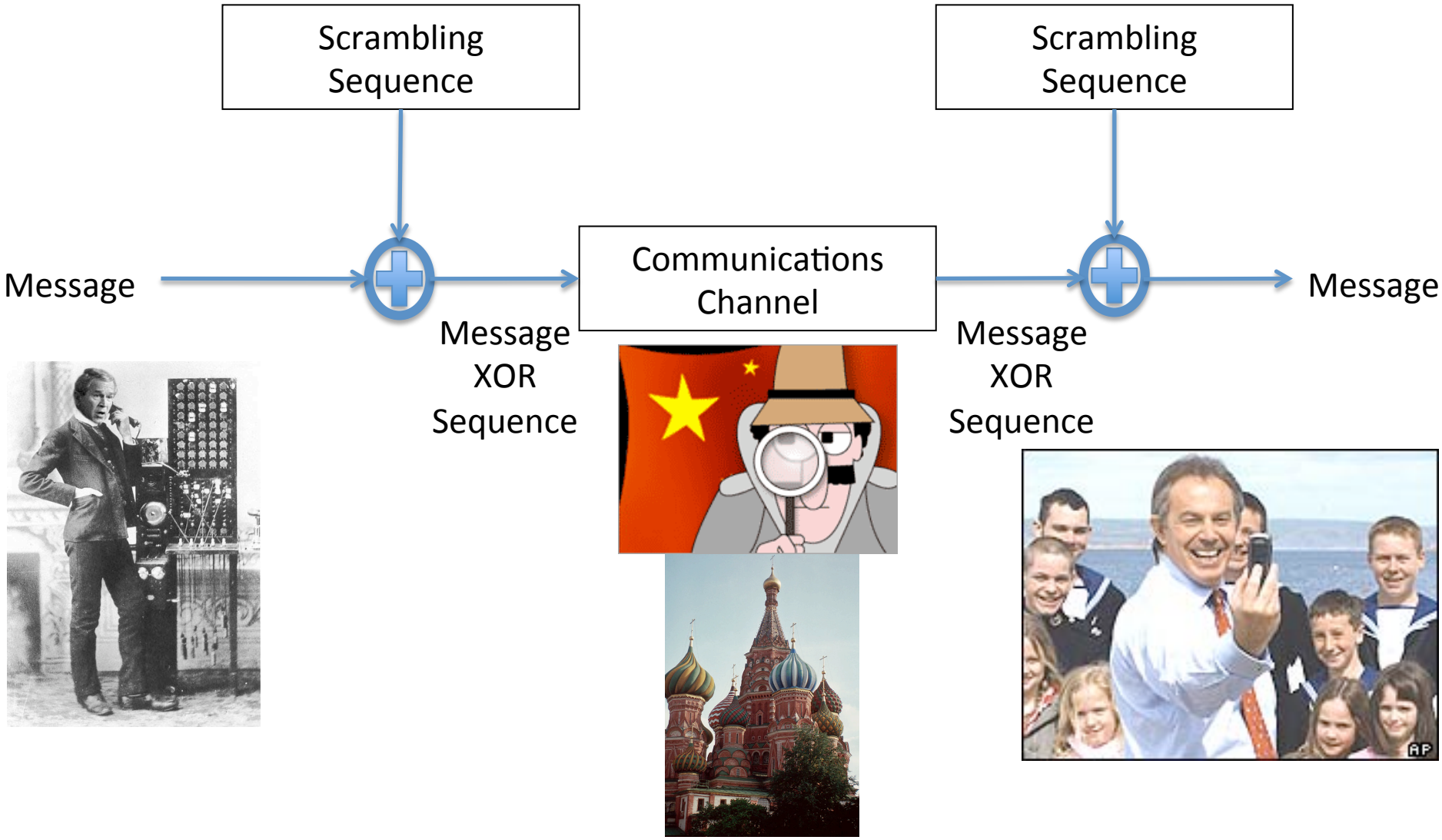
Name	4b	5b	Description
Q	-NONE-	00000	Quiet
I	-NONE-	11111	Idle
J	-NONE-	11000	SSD #1
K	-NONE-	10001	SSD #2
T	-NONE-	01101	ESD #1
R	-NONE-	00111	ESD #2
H	-NONE-	00100	Halt

Block coding transfers data with a fixed overhead: 20% less information per Baud in the case of 4B/5B

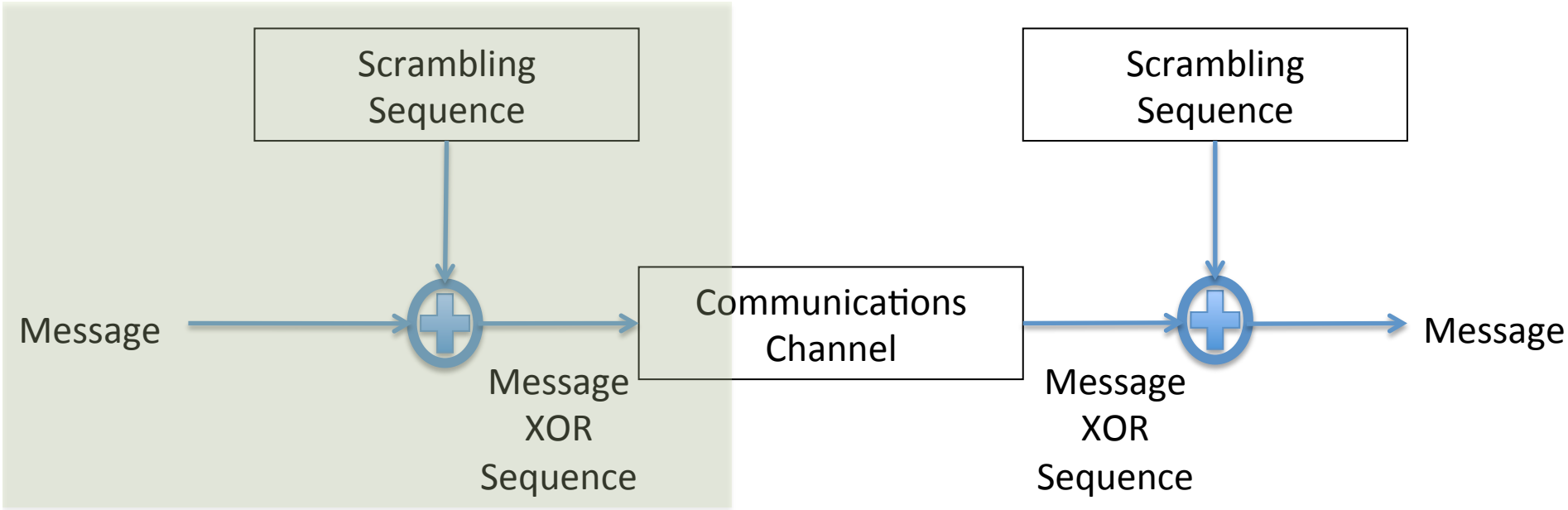
So to send data at 100Mbps; the line rate (the Baud rate) must be 125Mbps.

1Gbps uses an 8b/10b codec; encoding entire bytes at a time but with 25% overhead

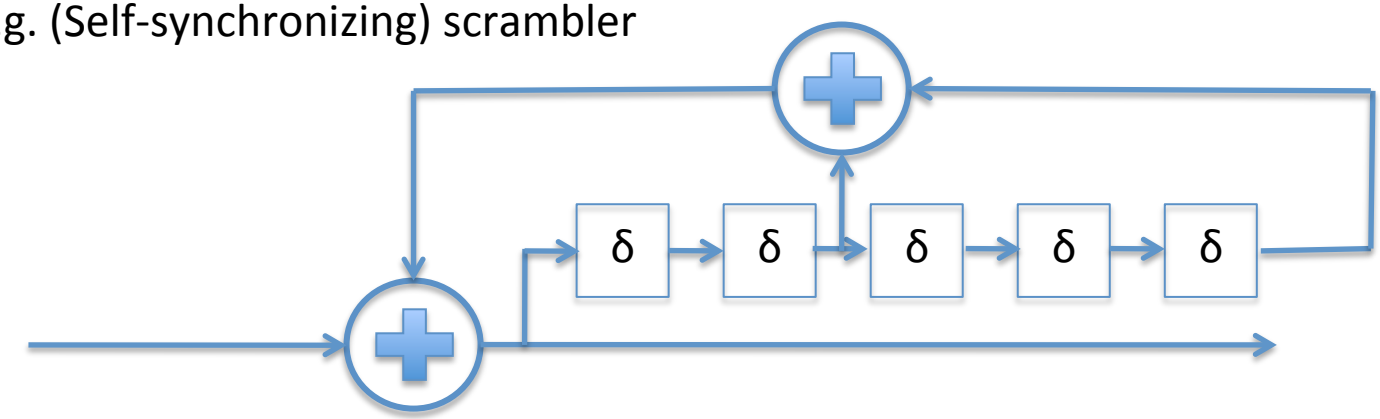
Line Coding Examples - IV



Line Coding Examples - IV



e.g. (Self-synchronizing) scrambler



Line Coding Examples – V
(Hybrid)

…1001111011010100010001011001110100010100101101010010011101011101 00…

…1001111011010100010100010110011101000101001011010100100111010111 0100…

Inserted bits marking “start of frame/block/sequence”

Scramble / Transmit / Unscramble

…0100010110011101000101001011010100100111010111010010010111011101 11000…

Identify (and remove) “start of frame/block/sequence”
This gives you the Byte-delineations for *free*

64b/66b combines a scrambler and a framer. The start of frame is a pair of bits 01 or 10: 01 means “this frame is data” 10 means “this frame contains data and control” – control could be configuration information, length of encoded data or simply “this line is idle” (no data at all)