CSE 311:
Data Communication

Instructor:

Dr. Md. Monirul Islam

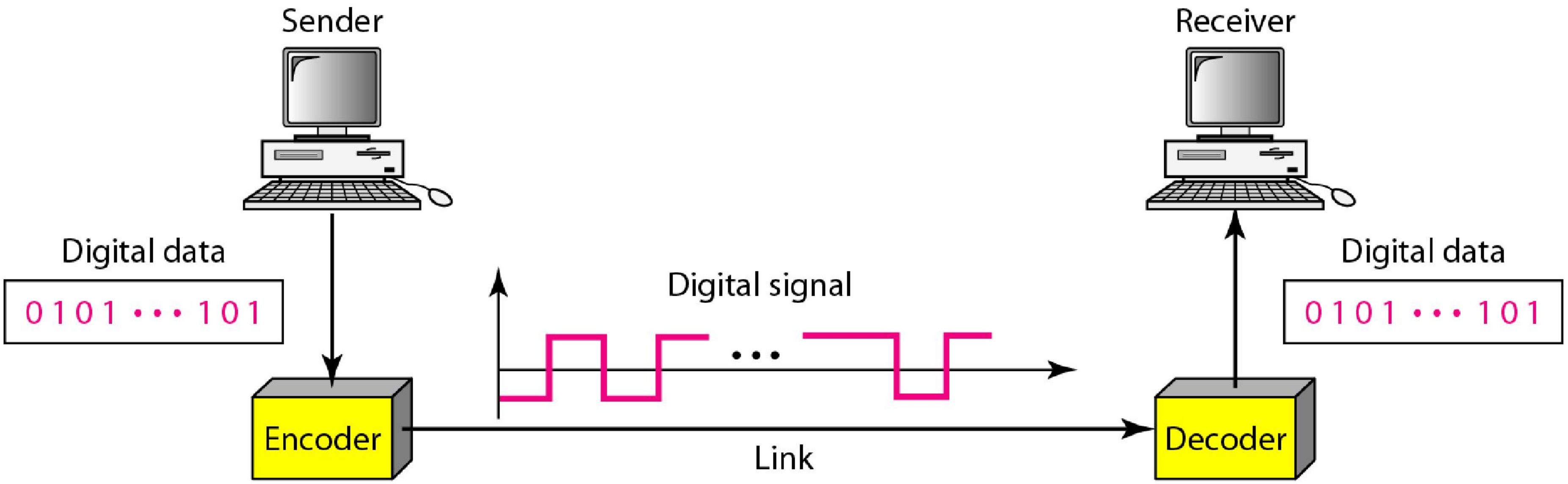
Digital-to-Digital Conversion: Line Coding, Block Coding and Scrambling

From

Data Communications and Networking, 5th Edition

By Behrouz A. Forouzan

Line Coding and Decoding



Line Coding

- converts *digital* data elements to *digital* signal elements

- Example:

  1 → +V

  0 → -V

Data Element and Signal Element

- data element
  - smallest entity to represent a piece of information
  - 0, 1, etc,
- signal element
  - shortest unit of a digital signal
  - Each represents one or more data element

Data Element and Signal Element

- data element
  - smallest entity to represent a piece of information
  - 0, 1, etc,
- signal element
  - shortest unit of a digital signal
  - Each represents one or more data element
- signal element is carrier, data element is carried

Data Element and Signal Element

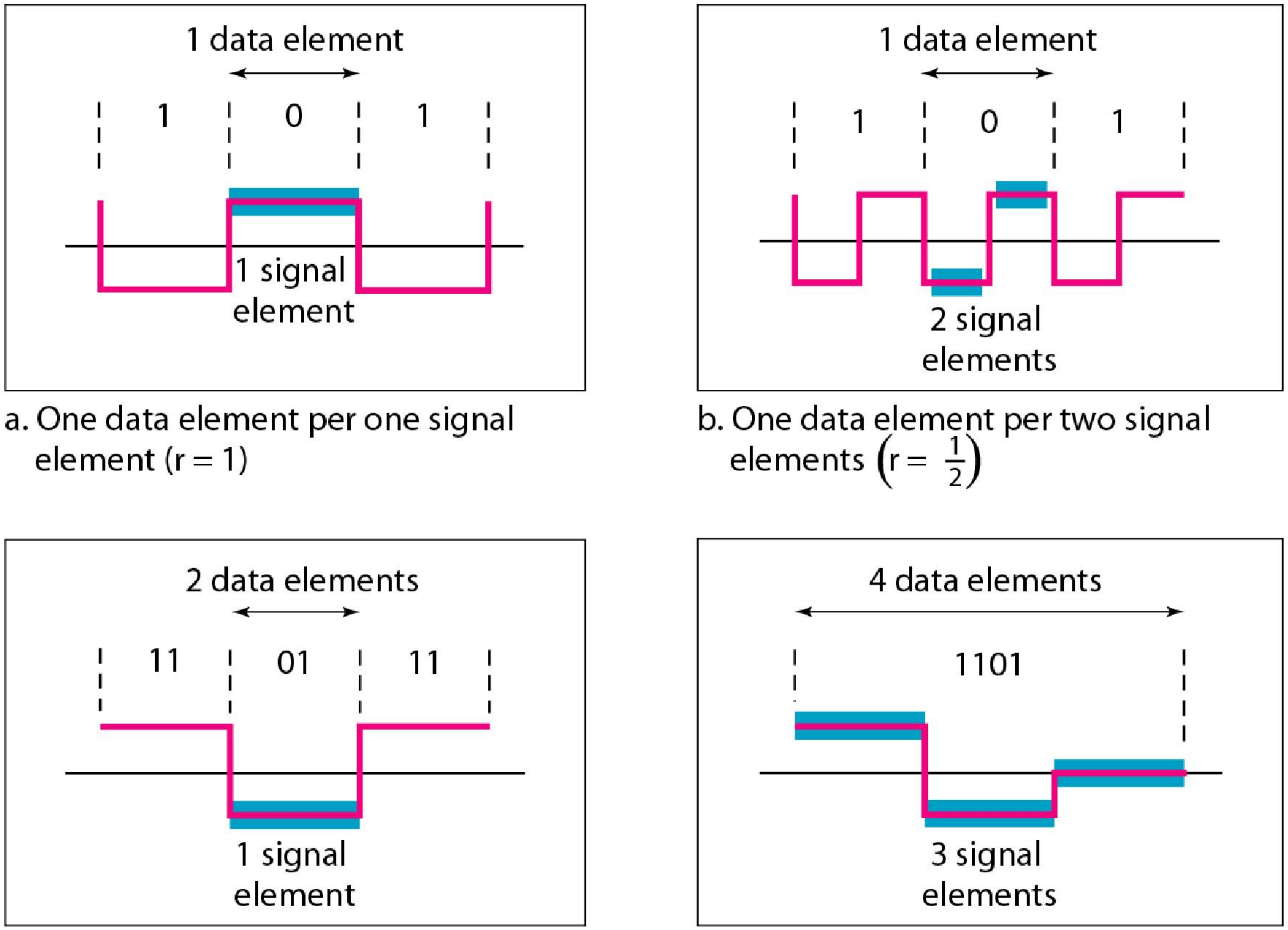
- data element
  - smallest entity to represent a piece of information
  - 0, 1, etc,

- signal element
  - shortest unit of a digital signal
  - Each represents one or more data element

- signal element is carrier, data element is carried

- *r* = No. of data element is carried by each signal element

Data Element and Signal Element



a. One data element per one signal element (r = 1)

b. One data element per two signal elements $\left(r = \frac{1}{2}\right)$

c. Two data elements per one signal element (r = 2)

d. Four data elements per three signal elements $\left(r = \frac{4}{3}\right)$

Data Rate and Signal Rate

- data rate, N
  - No. of data elements sent in 1 second
  - Other names: bit rate

- signal rate, S
  - No. of signal elements sent in 1 second
  - Other names: baud rate, pulse rate, modulation rate

- $r = N/S$ alternately, $S = N/r$

Data Rate and Signal Rate

- $r = N/S$ alternately, $S = N/r$
- Average Signal rate

$$S_{\text{ave}} = c \times N \times (1/r) \quad \text{baud}$$

Line Coding Guidelines

- *r* should be higher, $S = N/r$ should be lower
- Minimize *Baseline wandering*:
  - Running average of incoming signal power is *baseline*
  - Baseline is compared with incoming signal for decoding
  - Long runs of 0 or 1 drifts the baseline

Line Coding Guidelines

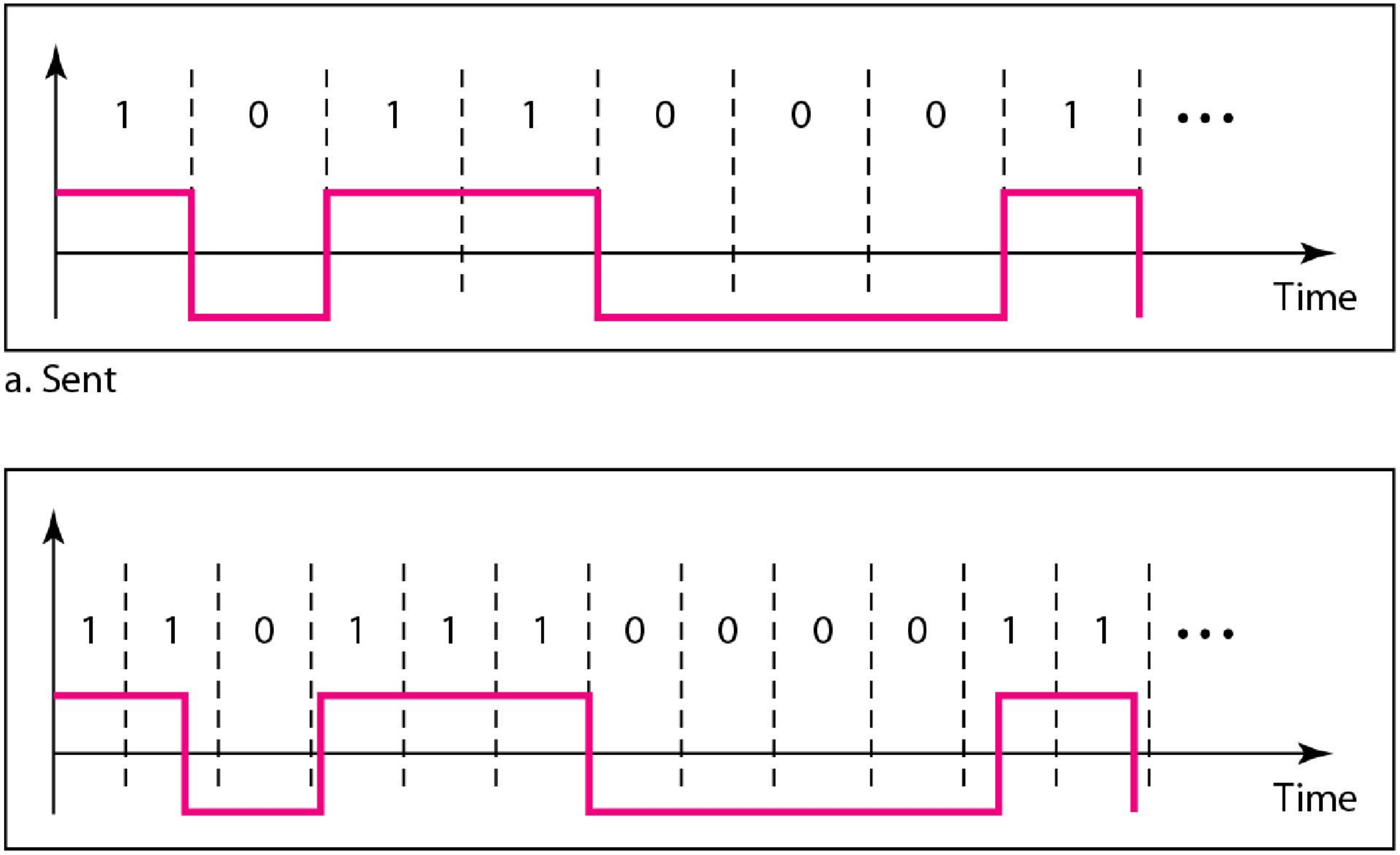
- Minimize *DC components*:
  - Constant voltage for long period
  - increaes low frequency components
  - bandpass channels do NOT support low frequencies

Line Coding Guidelines

- Self synchronization
  - Sender’s and receiver’s clock must have same bit interval
  - Different intervals may misinterpret incoming signal

Line Coding Guidelines

Illustration of lack of synchronization



a. Sent

b. Received

Line Coding Guidelines

- Built-in Error Detection
  - Error occurs during transmission
  - Must be able to detect and/or correct them
  - Example: certain type of signal transition (change) is NOT part of the coding

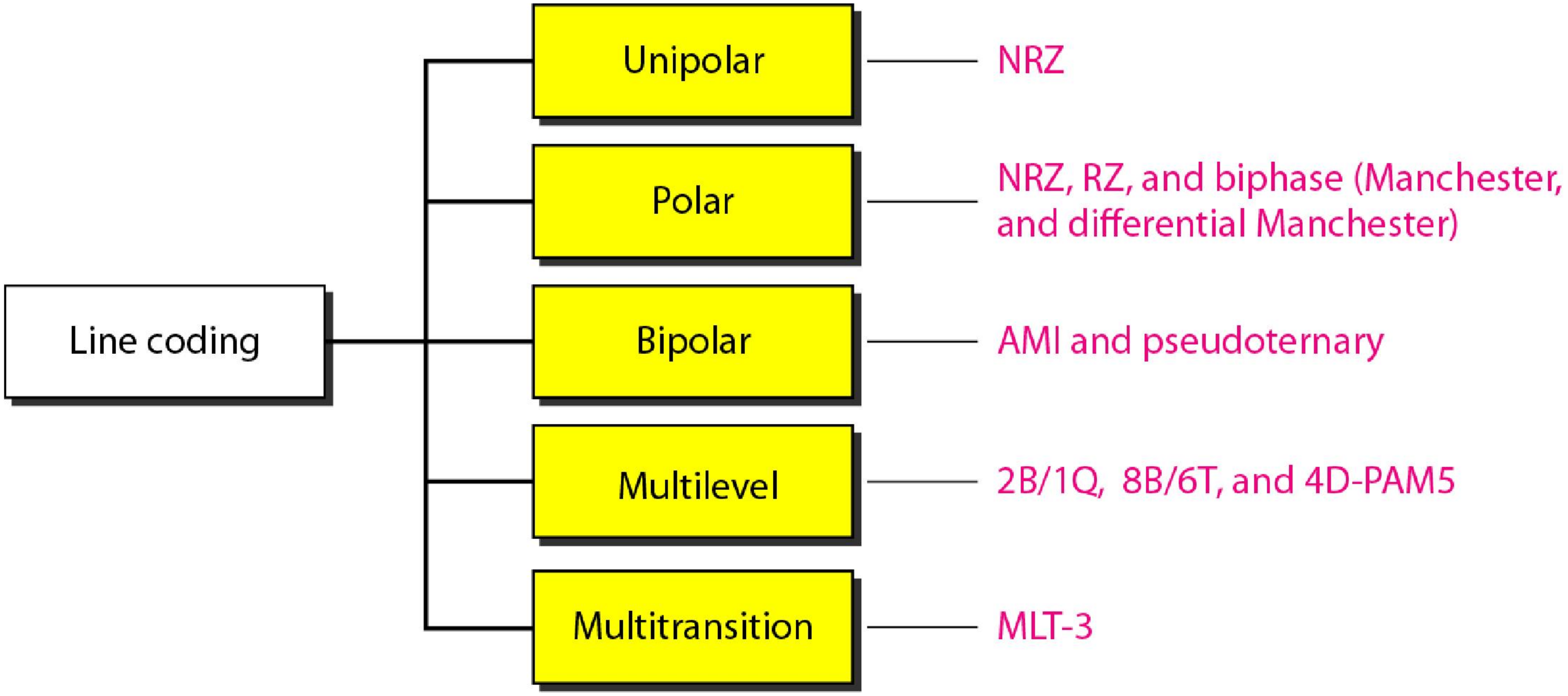
Line Coding Guidelines

- Higher Noise immunity
  - certain line coding prevents noises to be added to the signal
  - It is better than error detection

Line Coding Guidelines

- Complexity
  - Simpler is better
  - A coding with FOUR signal levels is more difficult to interpret than one that uses TWO signal levels

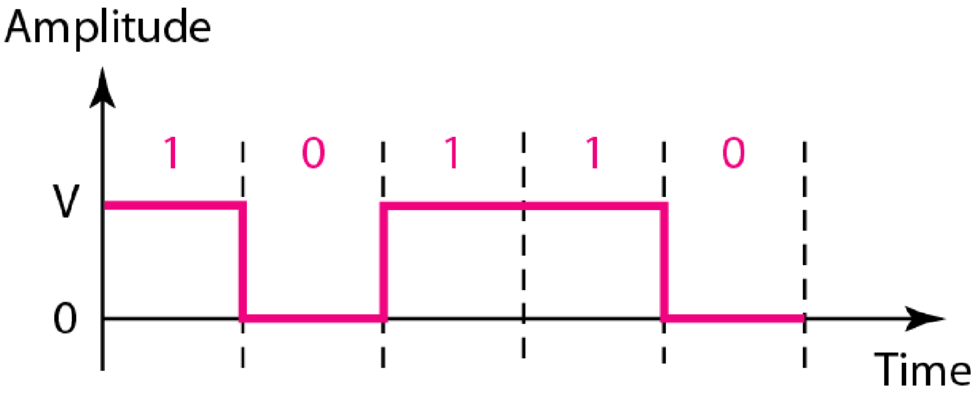
Line Coding Schemes


Line coding
Unipolar
Polar
Bipolar
Multilevel
Multitransition
NRZ
NRZ, RZ, and biphase (Manchester,
and differential Manchester)
AMI and pseudoternary
2B/1Q, 8B/6T, and 4D-PAM5
MLT-3

Unipolar

- All signal levels are on one side of the time axis - either above or below

- Example: Non Return to Zero (NRZ)
  - The signal level does not return to zero at middle of the bit

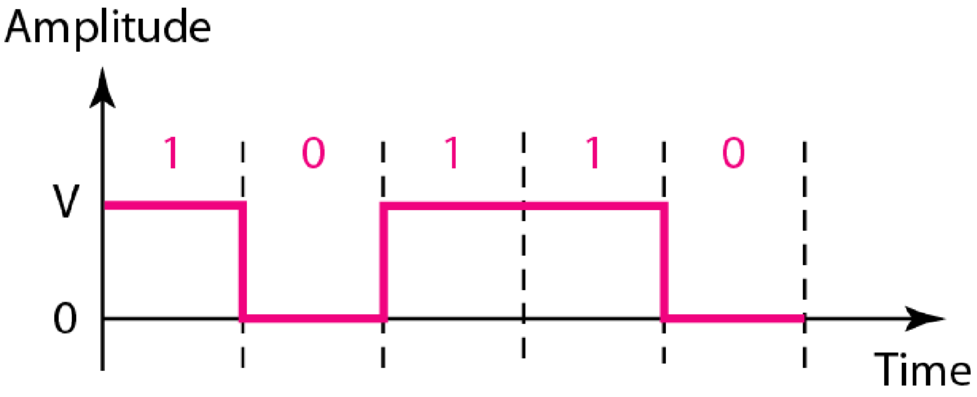


Unipolar: NRZ

- Pros
  - Simple

- Cons

  - Baseline wandering, DC components
  - No synchronization or error detection



$r = 1$

Polar

- signal levels are on BOTH sides of the time axis

- Polar NRZ
  - Uses two different voltage: +V and –V
  - Two versions: NRZ-Level (NRZ-L) and NRZ-Inversion (NRZ-I)

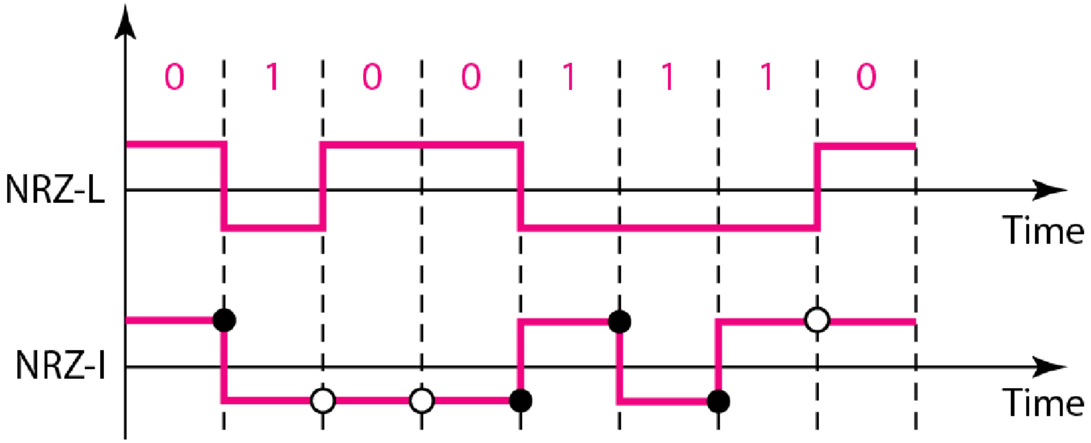
Polar: NRZ-L and NRZ-I

- Polar NRZ-L

+V: 0 and –V: 1

- Polar NRZ-I

Inversion: next bit is 1, No Inversion: next bit 0

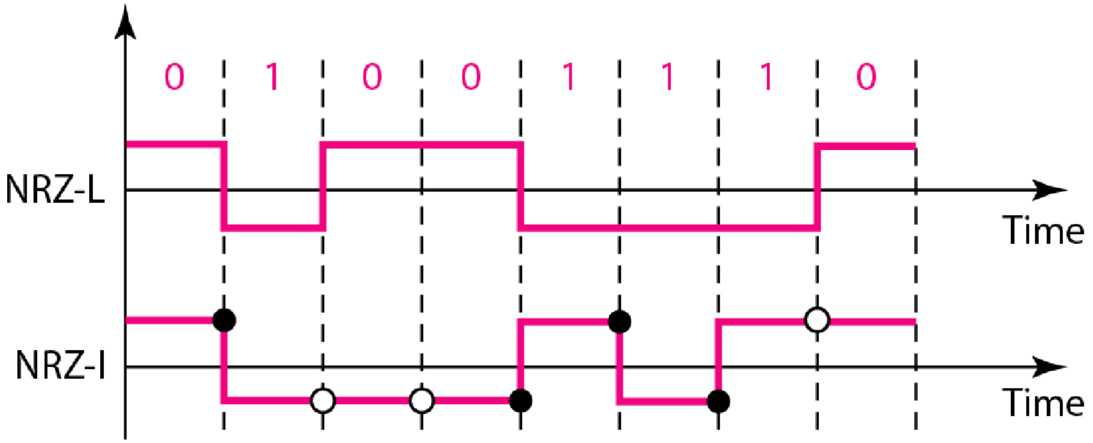


Polar: NRZ-L and NRZ-I

- Pros
  - Simple

- Cons

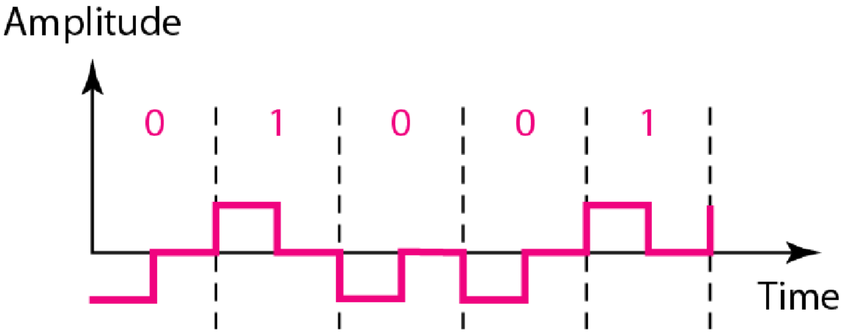
  - Baseline wandering, DC components, worse in NRZ-L
  - No synchronization or error detection



$r = 1$

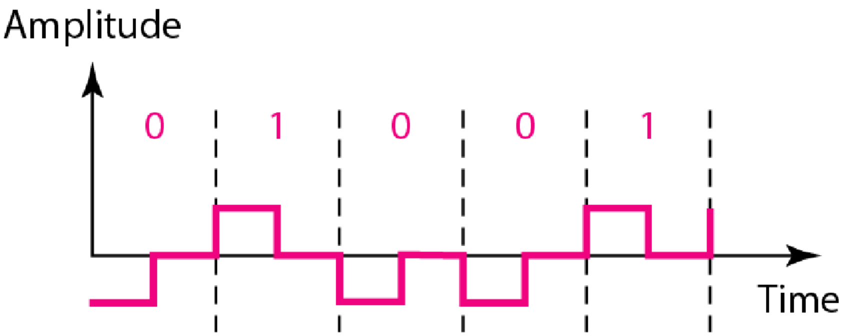
Polar: Return to Zero (RZ)

- 3 signal levels: +V, -V, 0
- Signal transition in the middle of the bit
  - bit 0: -V to 0 bit 1: +V to 0



Polar: Return to Zero (RZ)

- Pros
  - NO Baseline wandering, NO DC components
  - Self synchronization: transition synchronizes both sender and receiver
- Cons
  - More complex: 3 voltage levels to identify
  - No error detection



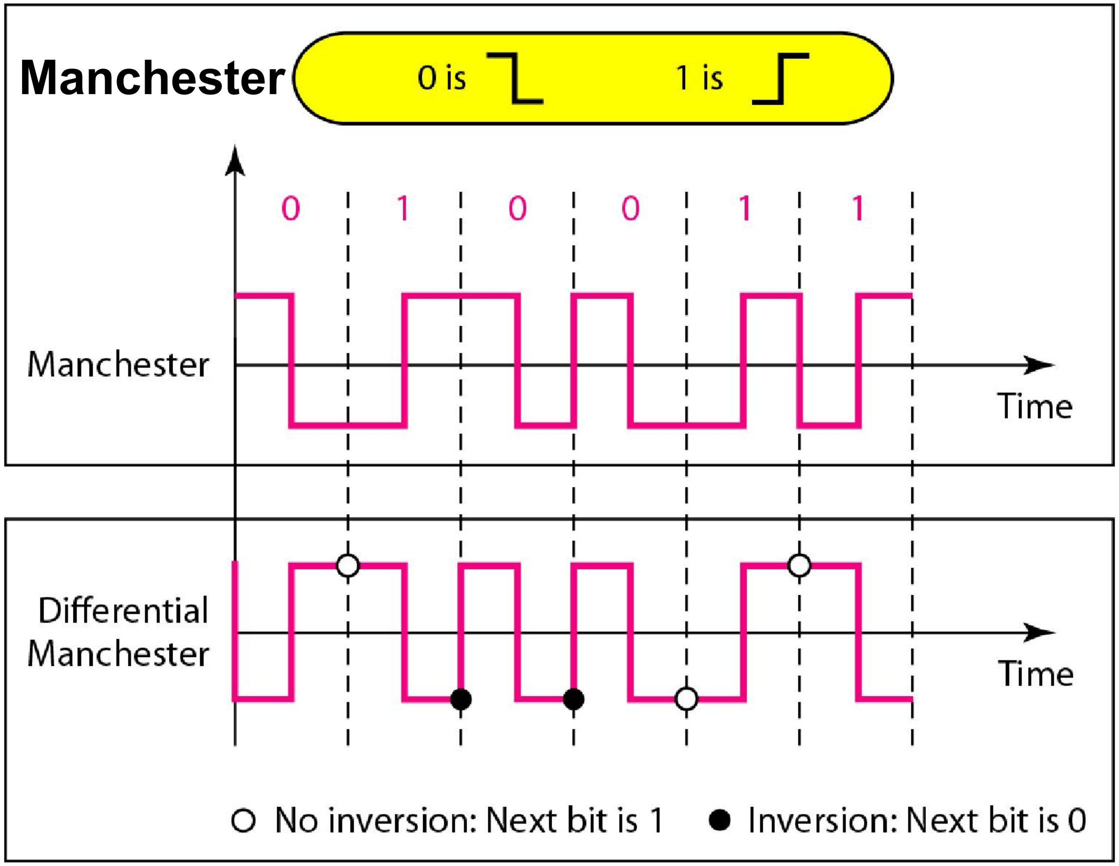
$r = ½$

Polar: Biphase-Manchester and Differential Manchester

- 2 signal levels: +V, -V
- Signal transition *always* in the middle of the bit, *similar to RZ*
- Manchester
  - Combines NRZ-L and RZ
- Differential Manchester
  - Combines NRZ-I and RZ

Polar: Biphase-Manchester and Differential Manchester

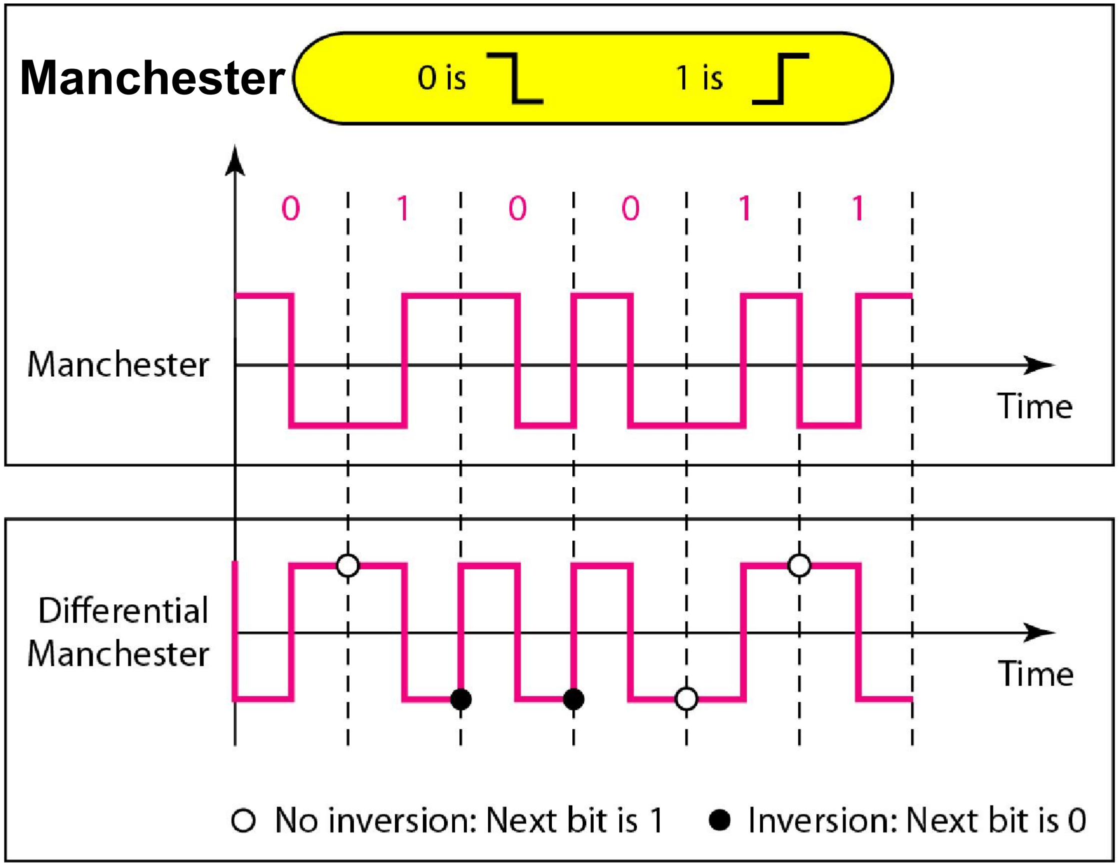
- Manchester
  - Level determines bits
- Differential Manchester
  - Inversion/no inversion determines bits



Polar: Biphase-Manchester and Differential Manchester

- Pros
  - 2 signal elements
  - NO Baseline wandering, NO DC components
  - Self synchronization

- Cons
  - ***r* = 1/2**
  - No error detection

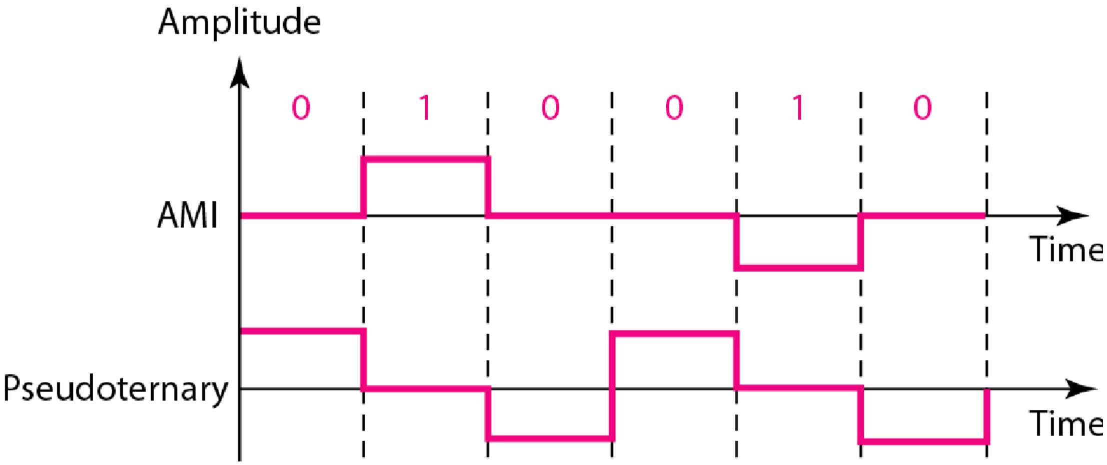


Bipolar

- 3 signal levels: 0, +V, -V
- 2 variations: alternate mark inversion (AMI) and Pseudoternary
- AMI
  - bit 0: signal level 0
  - bit 1: successively alternates between +V and -V
- Pseudoternary
  - bit 1: signal level 0
  - bit 0: successively alternates between +V and -V

Bipolar

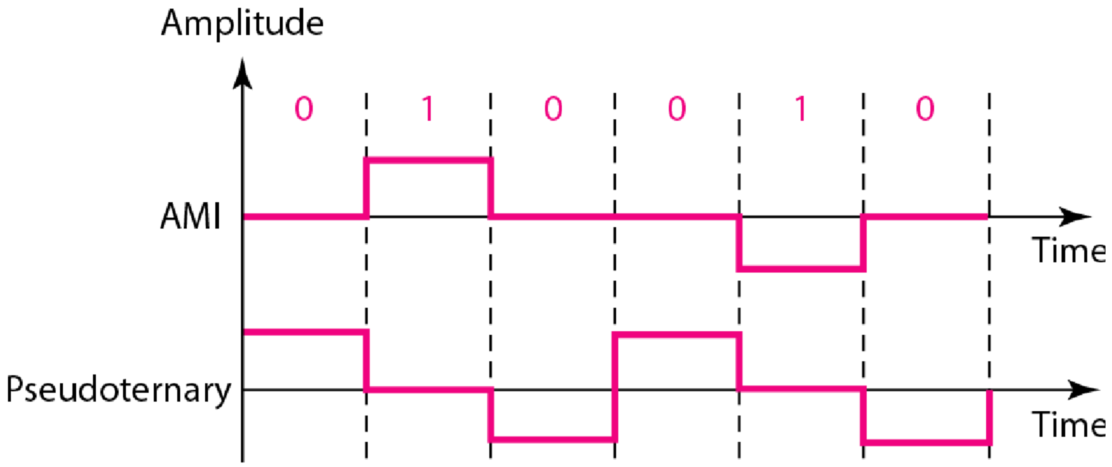
- AMI
  - bit 0: signal level 0
  - bit 1: successively alternates between +V and -V
- Pseudoternary
  - bit 1: signal level 0
  - bit 0: successively alternates between +V and -V



Bipolar

- Pros
  - $r = 1$

- Cons
  - 3 signal elements
  - NO synchronization
  - NO error detection



Multilevel Schemes

- Target: to increase *r* or *bit rate*
- Individual bit is NOT coded
- rather a sequence of *m* bits is coded by a sequence of *n* signal elements
- Assuming, No. of signal levels is *L*
  - Possible data patterns: 2^m
  - Possible signal patterns: L^n

$$2^m \textit{ data patterns} \quad \rightarrow \quad L^n \textit{ signal patterns}$$

Multilevel Schemes

2^m *data patterns* → L^n *signal patterns*

- $2^m > L^n$: mapping is NOT possible

- $2^m = L^n$: exact mapping is possible

- $2^m < L^n$:
  - Redundant signal patterns
  - Very flexible mapping is possible
  - Better noise immunity and error detection

Multilevel Schemes

- Coding symbol: short representation is *mBnL*
- *L* is replace by character code
  - $L = 2$: *B* meaning Binary
  - $L = 3$: *T* meaning Ternary
  - $L = 4$: *Q* meaning Quaternary

2*B*1*Q* Scheme

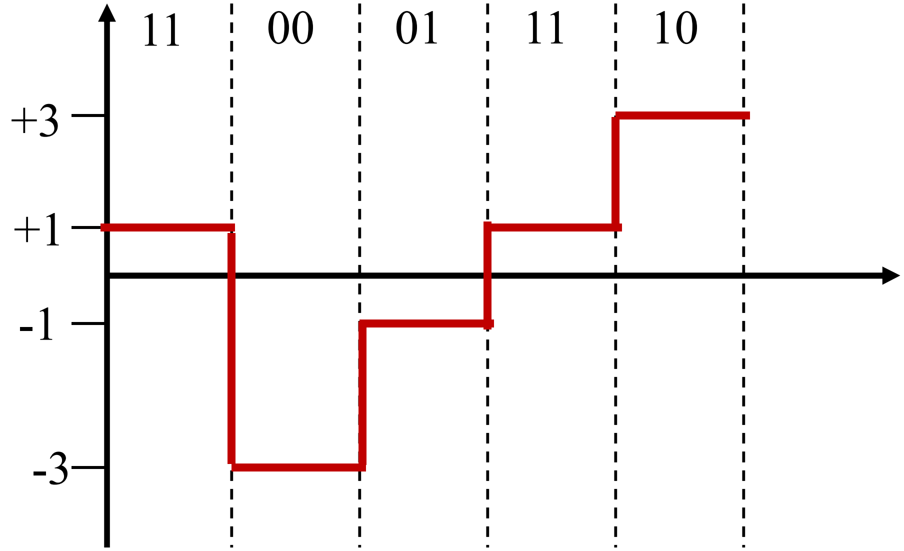
Rules

00: -3

01: -1

10: +3

11: +1

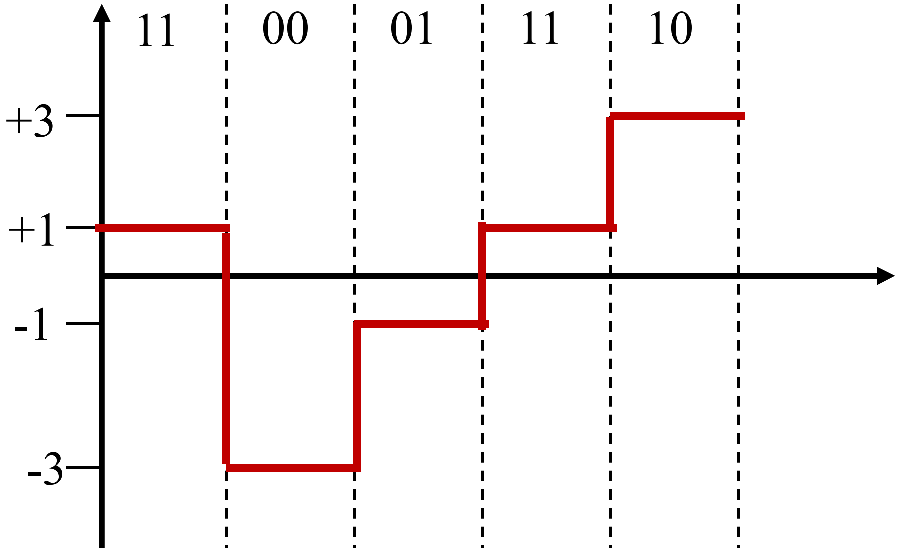


Multilevel Schemes

- Pros
  - ***r* = 2**
  - Used in DSL to provide high speed connection to the Internet

- Cons
  - 4 signal elements,
  - Baseline wandering, DC components are possible
  - NO redundancy, NO error detection

2*B*1*Q* Scheme
Rules
00: -3
01: -1
10: +3
11: +1



Multilevel Schemes: *8B6T*

- 2^8 = 256 different data patterns
- 3^6 = 729 different signal patterns of 3 levels, +V, -V and 0
- No. of redundant signal patterns 729-256 = 473
- Flexible pattern mapping
- Synchronization and error detection by 473 redundant signal patterns

Multilevel Schemes: *8B6T*

- All mapped signal patterns have weight +1 or 0 dc values

Mapping Table

Data patterns	Signal patterns	weight
00010001	– 0 – 0 + +	0
01010011	– + – + + 0	+1
01010000	+ – – + 0 +	+1

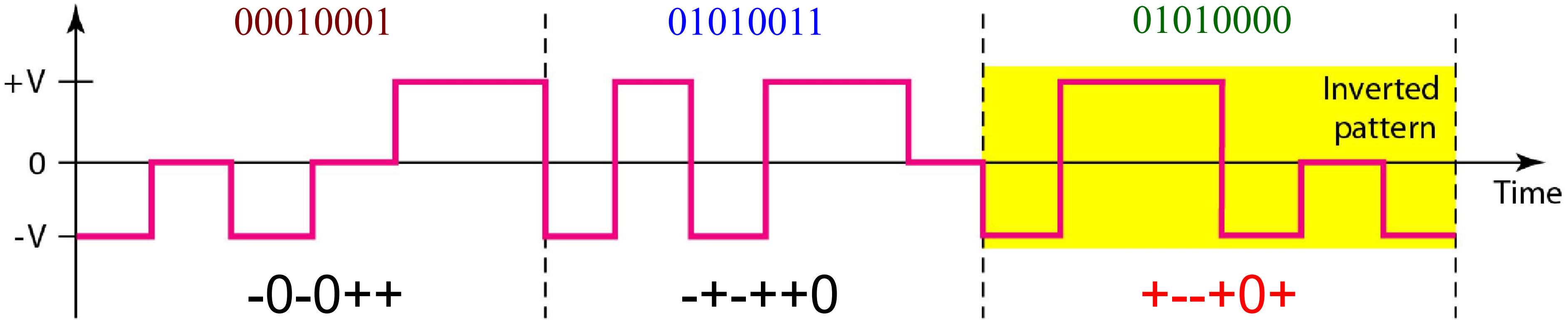
- NO signal patterns have weight -1

- DC balance is maintained by inverting polarity of signal pattern

if necessary + – – + 0 + is replaced by - + + - 0 - (WEIGHT: -1)

Multilevel Schemes: *8B6T*

Let, we have to send data sequence
000100010101001101010000



Weight: 0

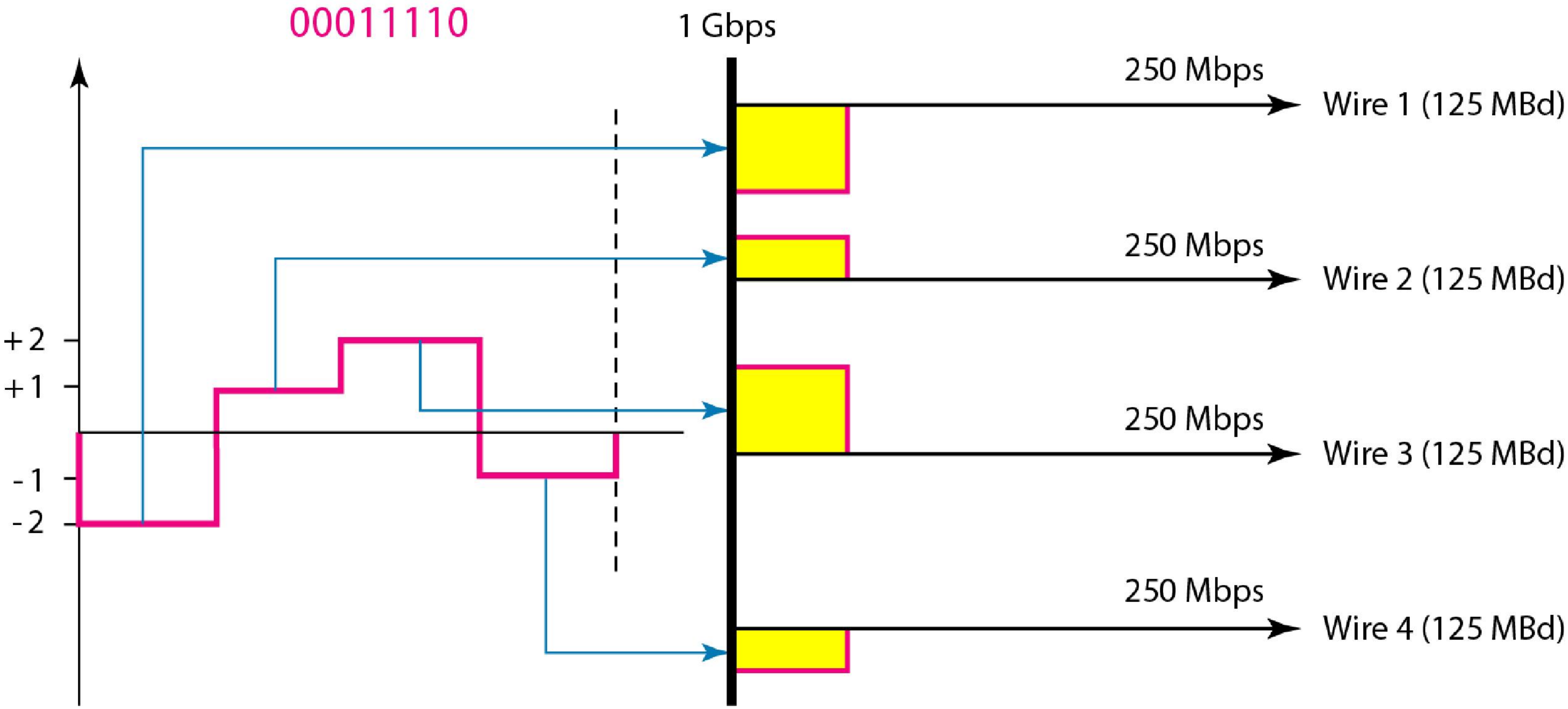
Weight: +1

Weight was +1
Now it is -1
to ensure DC balance

Multilevel Schemes: *4D-PAM5*

- 4 *D*imensional: 4 signal elements, each sent through separate wire
- *P*ulse *A*mplitude *M*odulation 5: 5 different levels -2, -1, 0, +1, +2
- Used signal levels: -2, -1, +1, +2
- Total signal patterns: $4^4 = 256$, which support 256 binary data patterns
- 4*D-PAM*5 is equivalent to *linear* 8*B*4*Q*

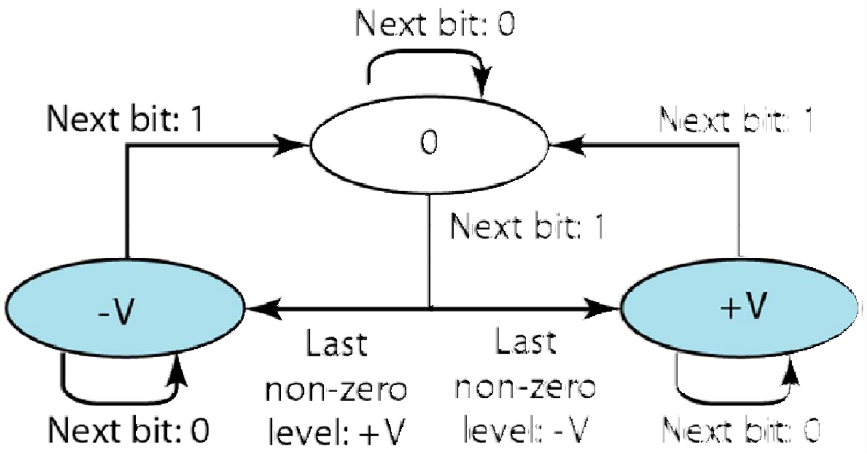
Multilevel Schemes: *4D-PAM5*



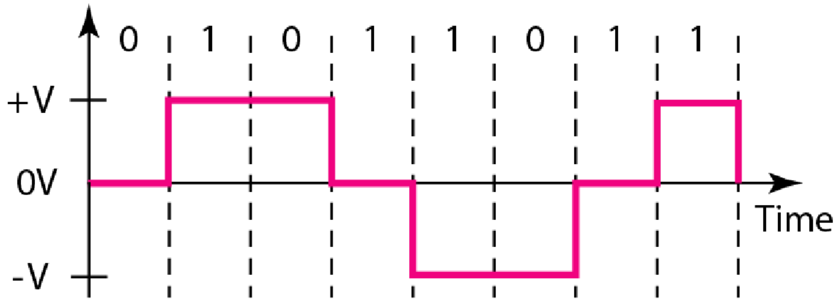
- 1 Gbps Gigabit LAN through 4 copper wires of 125 Mbaud each
- Self synchronization, NO DC components

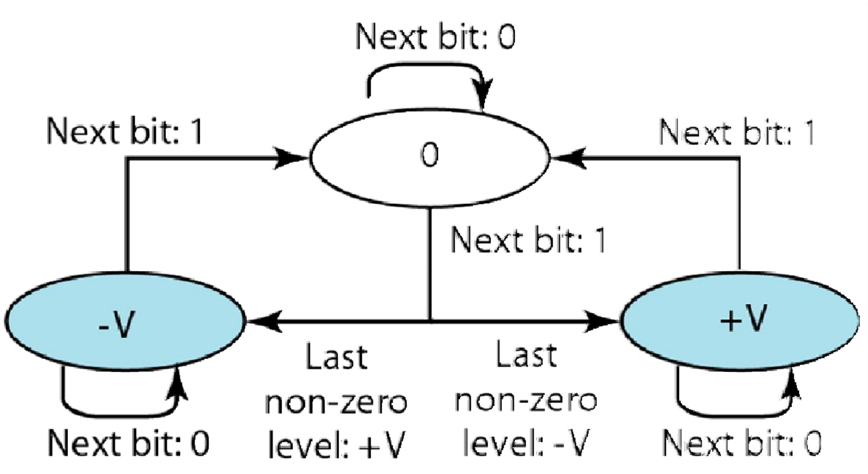
Multitransition: MLT-3

- Similar to differential coding like NRZ-I and differential Manchester
- Uses more than 2 signal levels, MLT-3 uses 3 levels: +V, 0, -V
- *Complex* transition rules
  - If next bit = 0, no transition
  - If next bit = 1 and current level = ±V, transition to level 0
  - If next bit = 1 and current level = 0, transition to opposite of last nonzero level



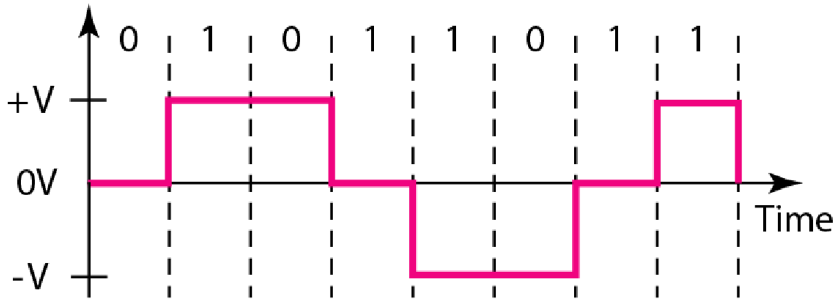
Multitransition: MLT-3

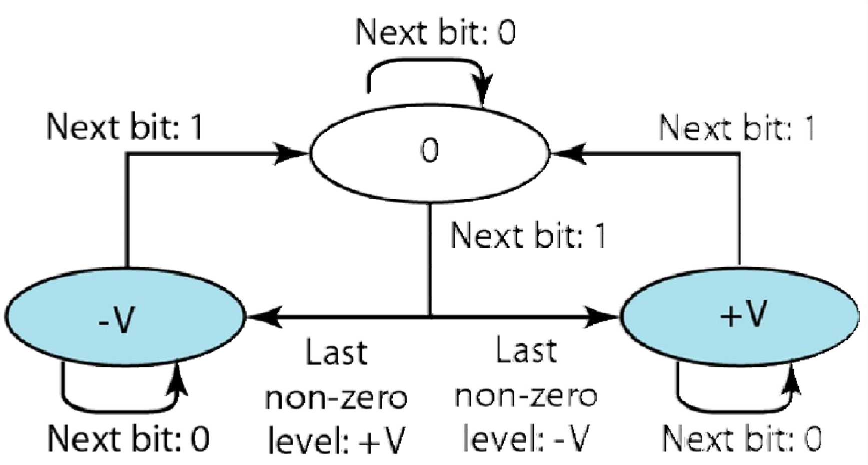


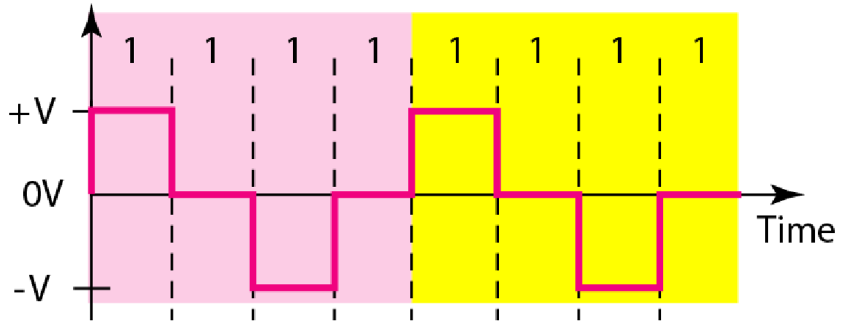


- *Complex* transition rules
- *r* = 1
- No synchronization for long sequence bit 0’s

Multitransition: MLT-3



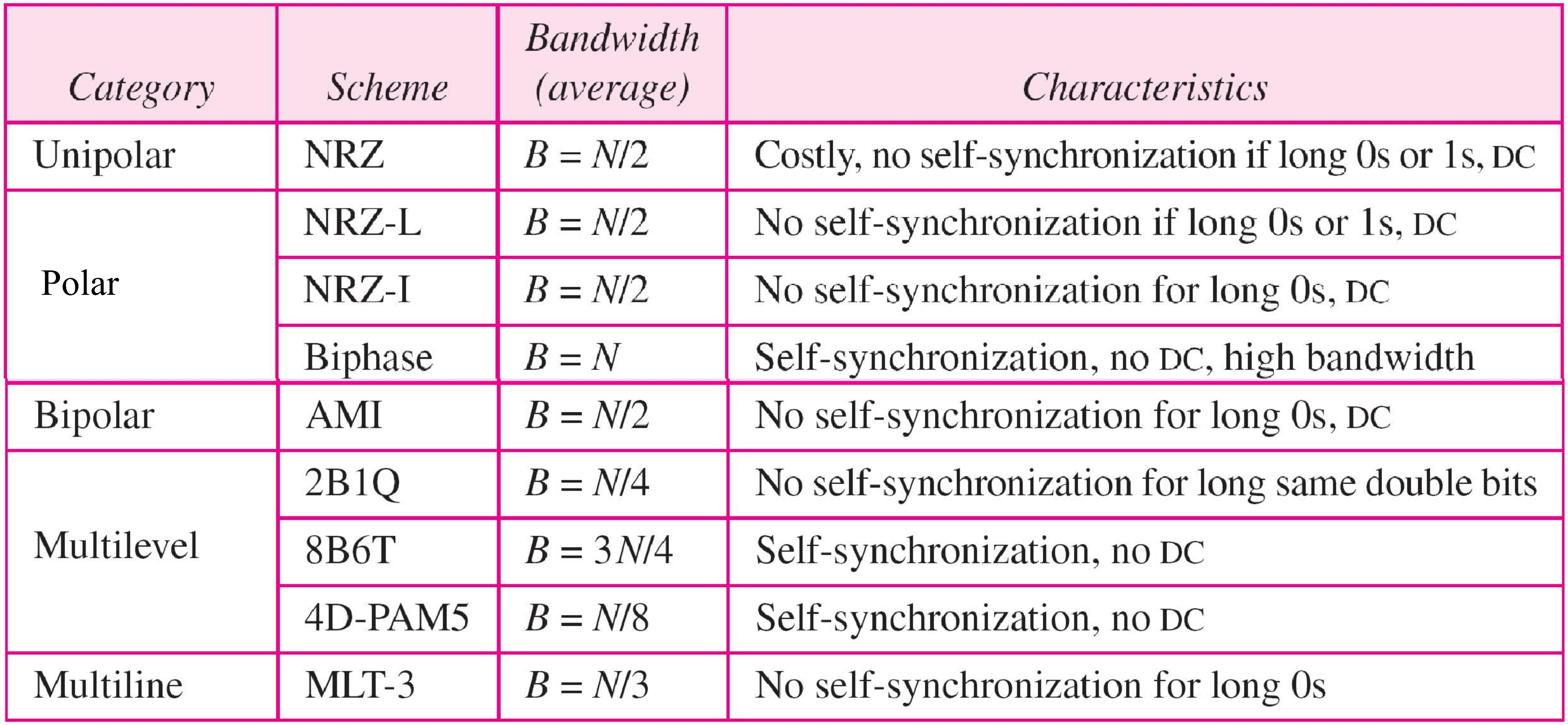




Nonperiodic to periodic signal
with a period of 4× bit duration

Repeating pattern: +V0-V0

Line Coding summary

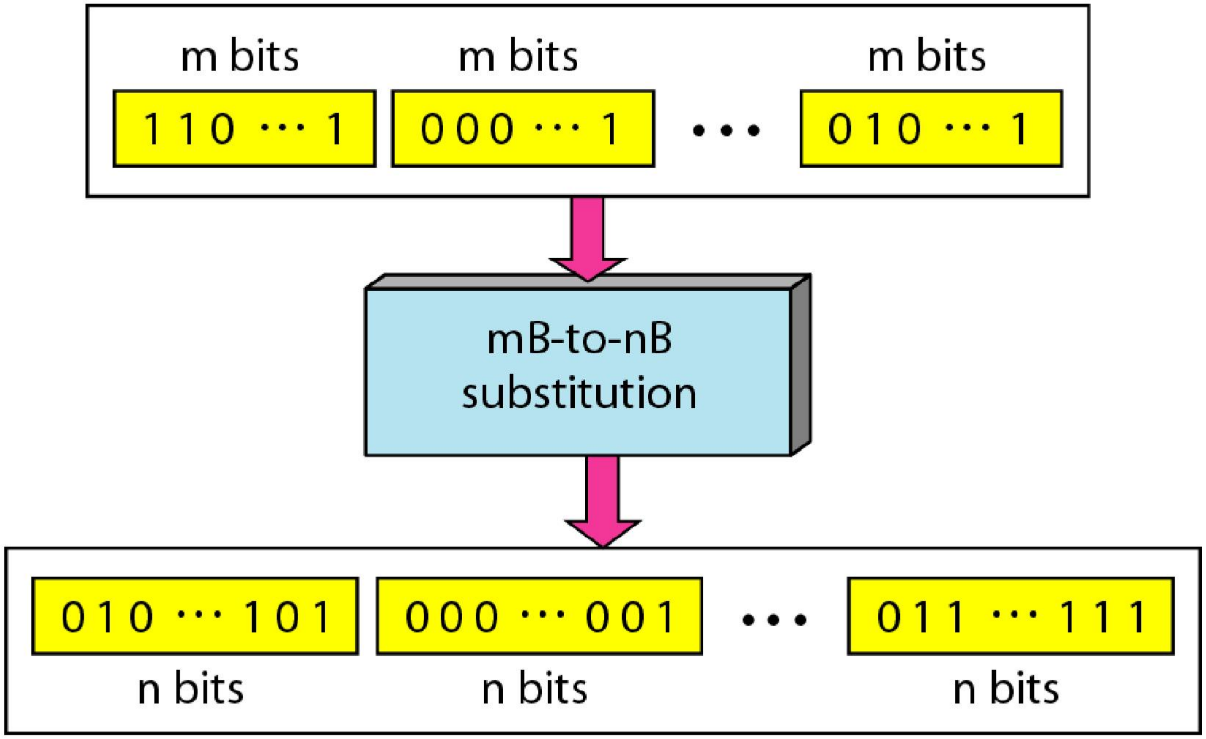


Category	*Scheme*	*Bandwidth (average)*	*Characteristics*
Unipolar	NRZ	$B = N/2$	Costly, no self-synchronization if long 0s or 1s, DC
Polar	NRZ-L	$B = N/2$	No self-synchronization if long 0s or 1s, DC
	NRZ-I	$B = N/2$	No self-synchronization for long 0s, DC
	Biphase	$B = N$	Self-synchronization, no DC, high bandwidth
Bipolar	AMI	$B = N/2$	No self-synchronization for long 0s, DC
Multilevel	2B1Q	$B = N/4$	No self-synchronization for long same double bits
	8B6T	$B = 3N/4$	Self-synchronization, no DC
	4D-PAM5	$B = N/8$	Self-synchronization, no DC
Multiline	MLT-3	$B = N/3$	No self-synchronization for long 0s

Block Coding: *mB/nB*

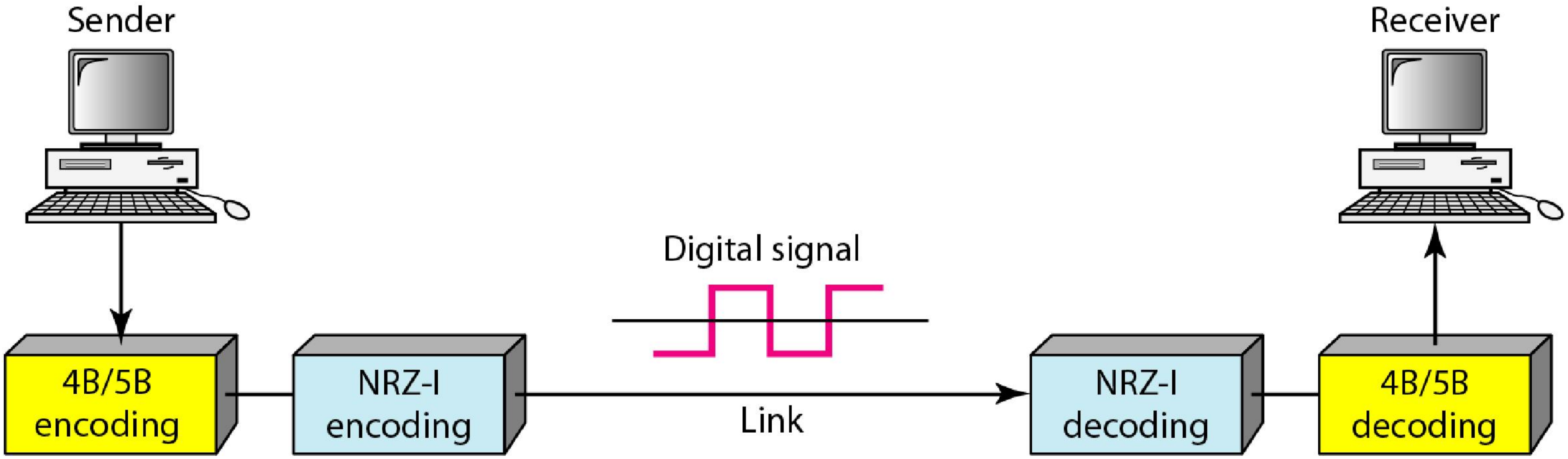
- replaces *m* bits by *n* bits, where $m < n$
- increaes redundancy and synchronization
- Done before line coding using 3 steps
  - division
  - substitution
  - combination

Division of a stream into m-bit groups



Combining n-bit groups into a stream

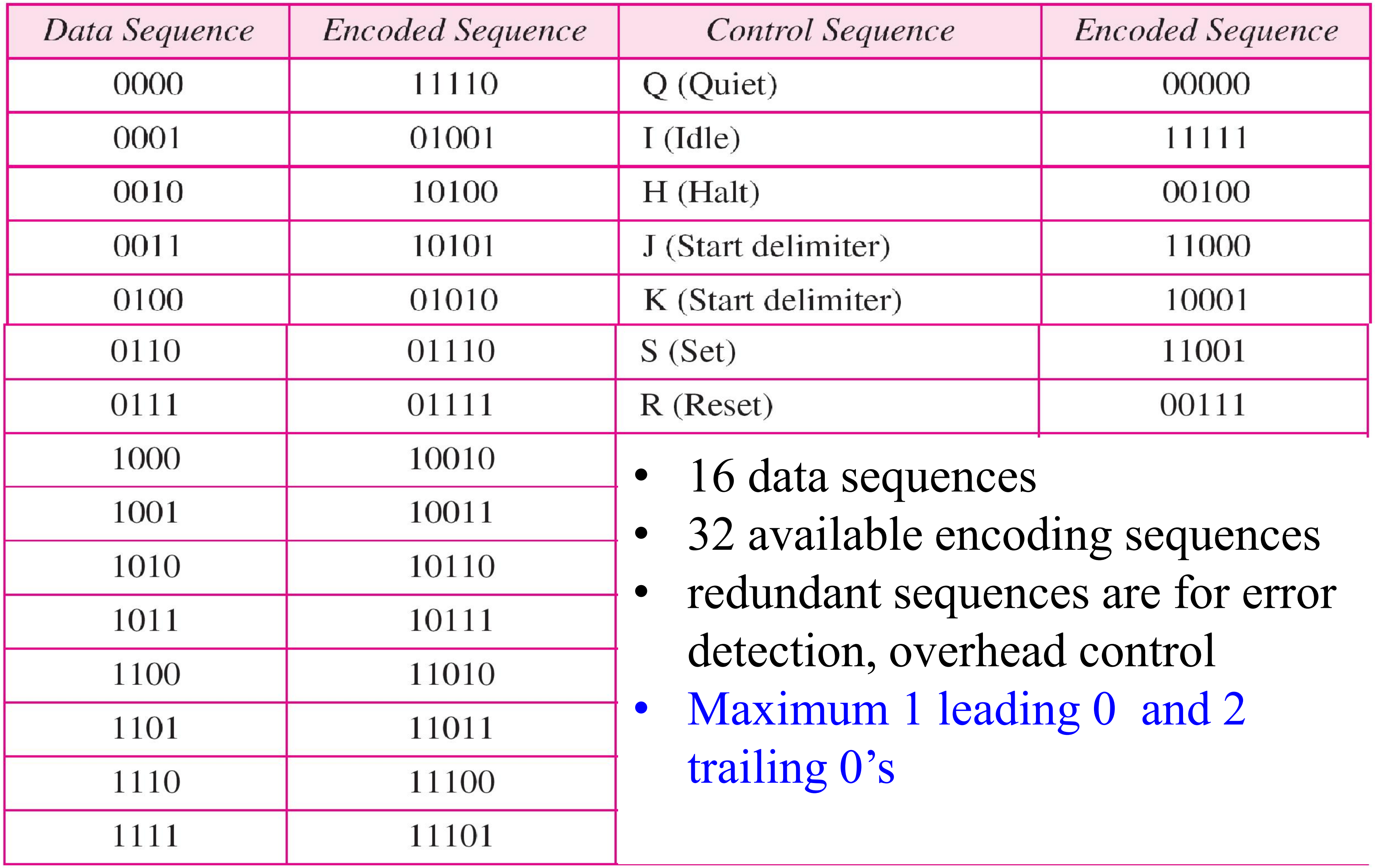
Block Coding: *4B/5B*



- Done in combination with NRZ-I which has sync problem for long sequence of 0’s

- 4*B*/5*B* ensures no more than 3 consecutive 0’s

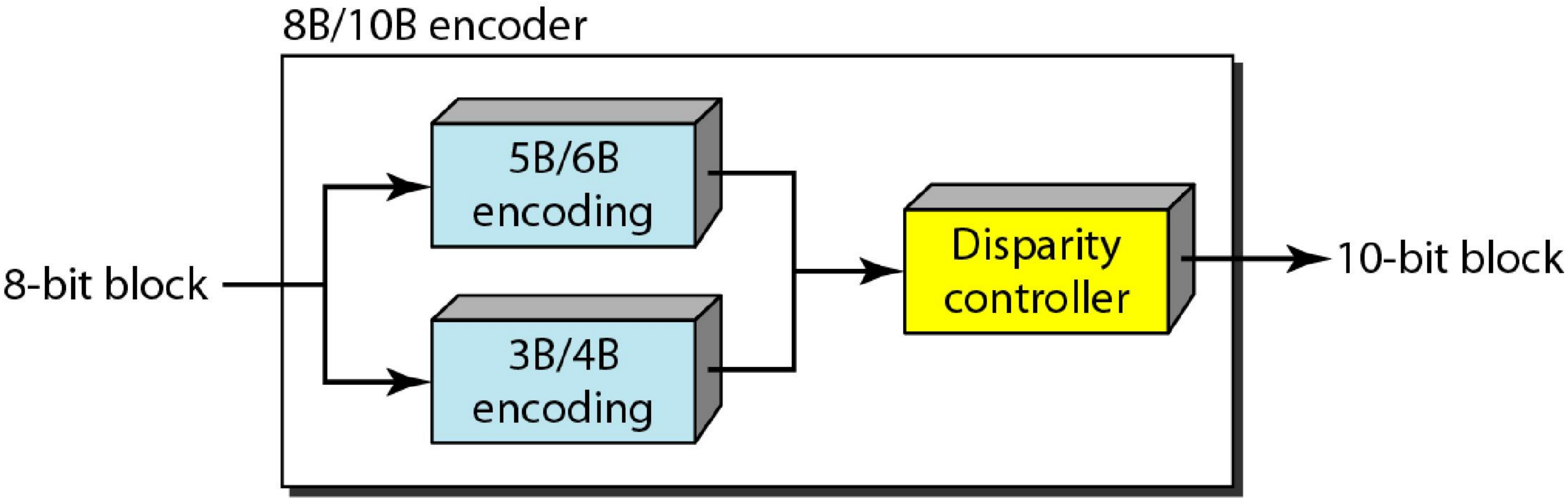
Block Coding: *4B/5B*



Data Sequence	*Encoded Sequence*	*Control Sequence*	*Encoded Sequence*
0000	11110	Q (Quiet)	00000
0001	01001	I (Idle)	11111
0010	10100	H (Halt)	00100
0011	10101	J (Start delimiter)	11000
0100	01010	K (Start delimiter)	10001
0110	01110	S (Set)	11001
0111	01111	R (Reset)	00111
1000	10010		
1001	10011		
1010	10110		
1011	10111		
1100	11010		
1101	11011		
1110	11100		
1111	11101		

- 16 data sequences
- 32 available encoding sequences
- redundant sequences are for error detection, overhead control
- Maximum 1 leading 0 and 2 trailing 0's

Block Coding: *8B/10B*



- More redundancy and better error detection than 4*B*/5*B*

- Actually done by combination of 5*B*/6*B* and 3*B*/4*B* to simplify mapping table

Search for Best Line Coding: Scrambling

- Characteristics of a good line coding
  - No increase in b/w for self sync
  - NO DC
- Biphase scheme
  - High bandwidth $(r < 1)$
- NRZ + block coding
  - DC still exists
- Bipolar AMI
  - NO sync for long sequence of 0's
  - Other characteristics are OK

Search for best line coding: Scrambling

- Bipolar AMI
  - NO sync for long sequence of 0’s
  - Other characteristics are OK

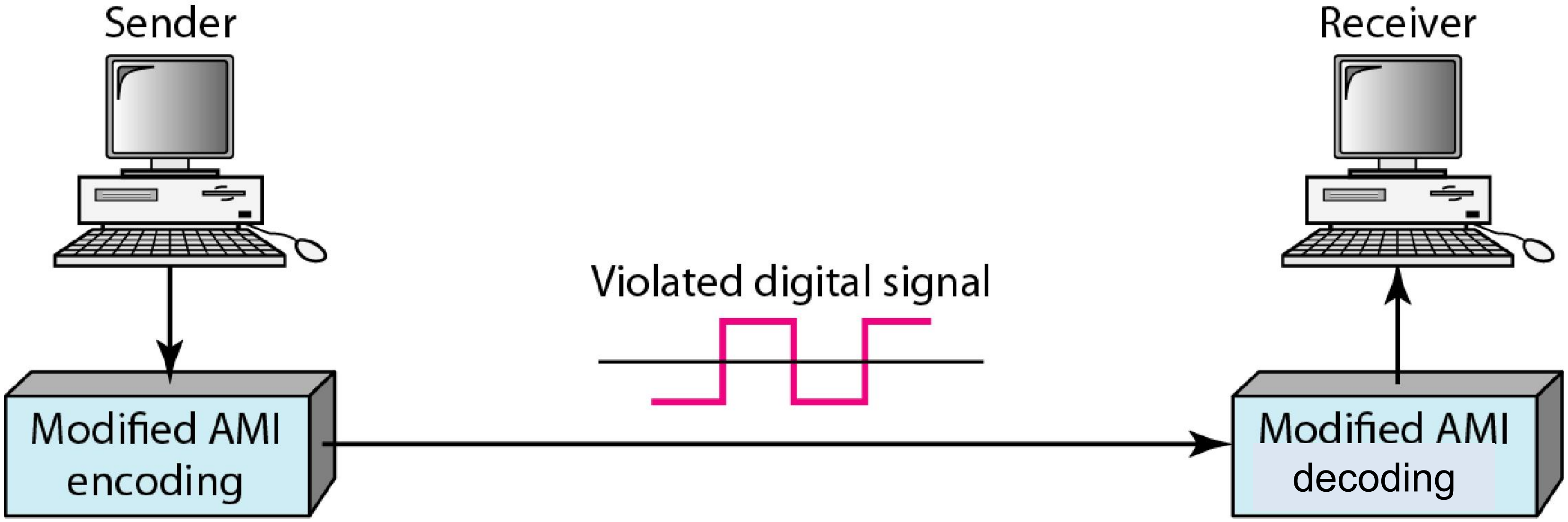
- Scrambling uses AMI but inserts nonzero pulses in long sequence of 0’s

Search for best line coding: Scrambling

- Bipolar AMI
  - NO sync for long sequence of 0's
  - Other characteristics are OK

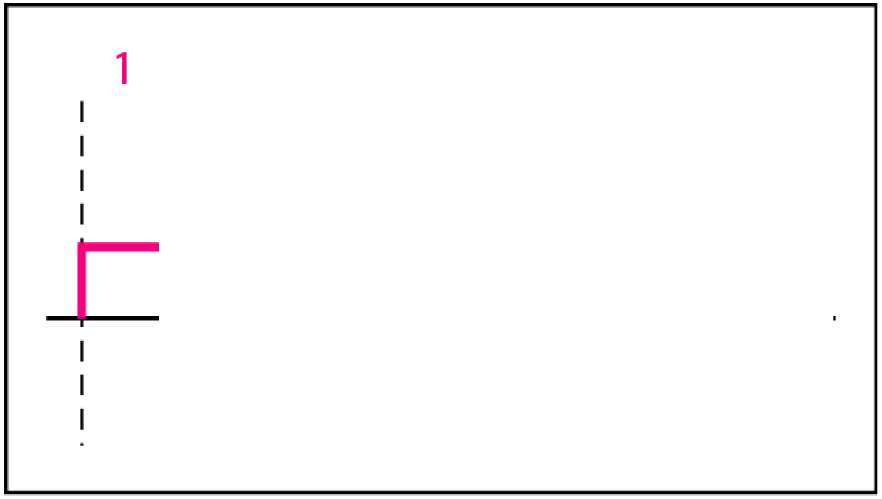
- Scrambling uses AMI but inserts nonzero pulses in long sequence of 0's

- Thus AMI's rules are violated on the fly



Scrambling: *B8ZS*

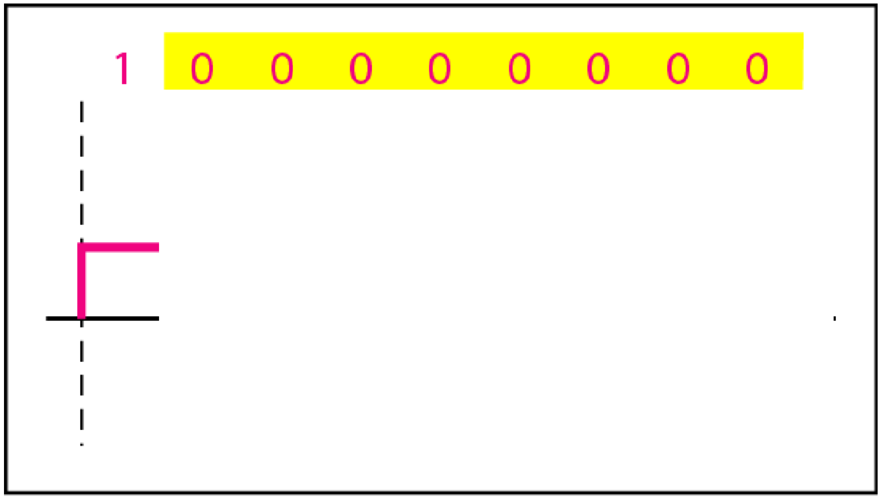
- Bipolar 8 Zeros (*B8ZS*)
  - Replaces 8 zeros by 0 0 0 V B 0 V B
  - V means Violation to AMI rules
  - B means normal polarity according to AMI rules



a. Previous level is positive.

Scrambling: *B8ZS*

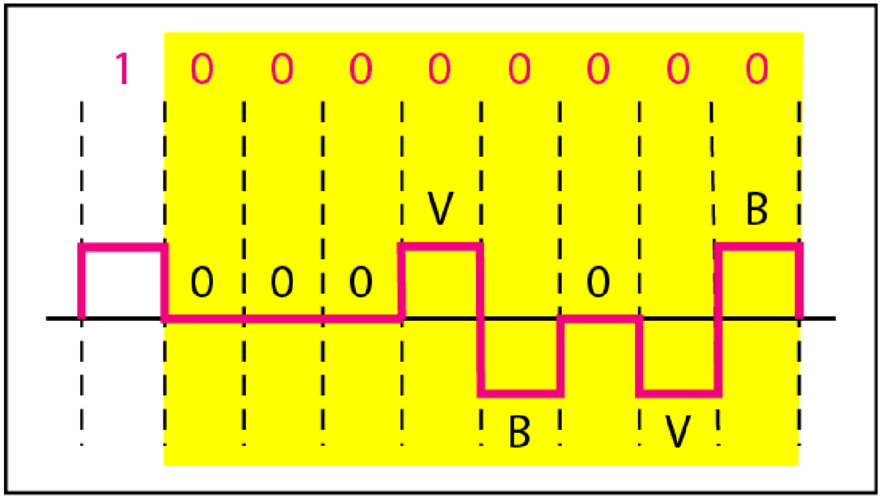
- Bipolar 8 Zeros (*B8ZS*)
  - Replaces 8 zeros by 0 0 0 V B 0 V B
  - V means Violation to AMI rules
  - B means normal polarity according to AMI rules



a. Previous level is positive.

Scrambling: *B8ZS*

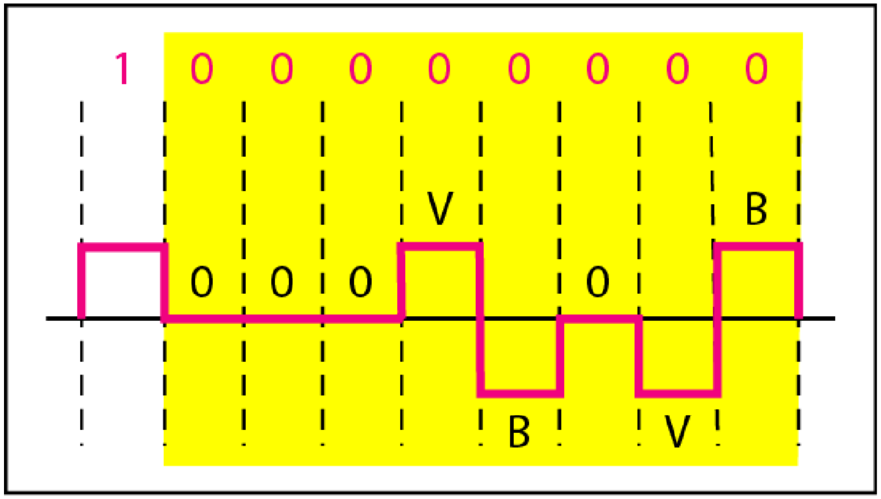
- Bipolar 8 Zeros (*B8ZS*)
  - Replaces 8 zeros by 0 0 0 V B 0 V B
  - V means Violation to AMI rules
  - B means normal polarity according to AMI rules



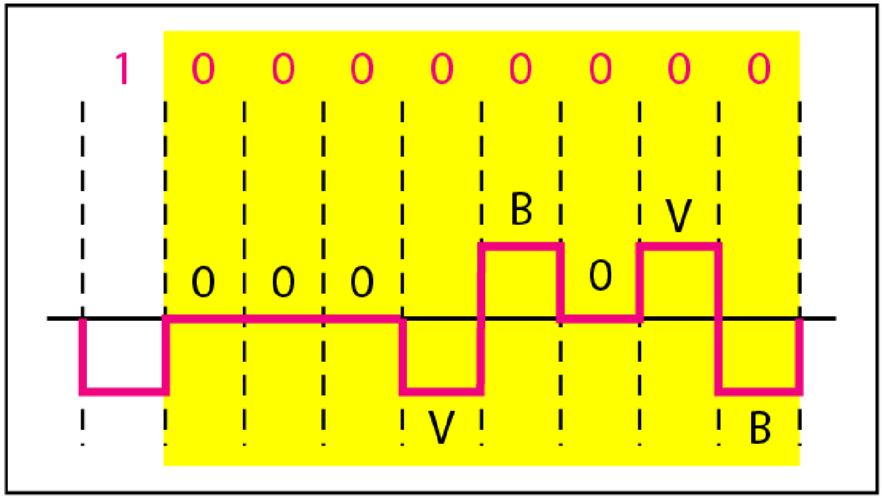
a. Previous level is positive.

Scrambling: ***B8ZS***

- Bipolar 8 Zeros (*B8ZS*)
  - Replaces 8 zeros by 0 0 0 V B 0 V B
  - V means Violation to AMI rules
  - B means normal polarity according to AMI rules



a. Previous level is positive.



b. Previous level is negative.

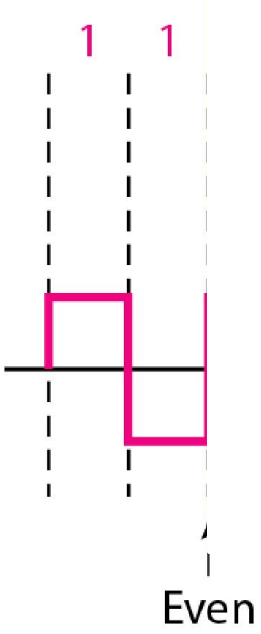
Scrambling: *HDB3*

- High Definition Bipolar 3-Zero (*HDB*3)
- Replaces 4 zeros by
  - 0 0 0 V if No. of nonzero pulses after last substitution is ODD
  - B 0 0 V if No. of nonzero pulses after last substitution is EVEN

Scrambling: *HDB3*

- High Definition Bipolar 3-Zero (*HDB*3)

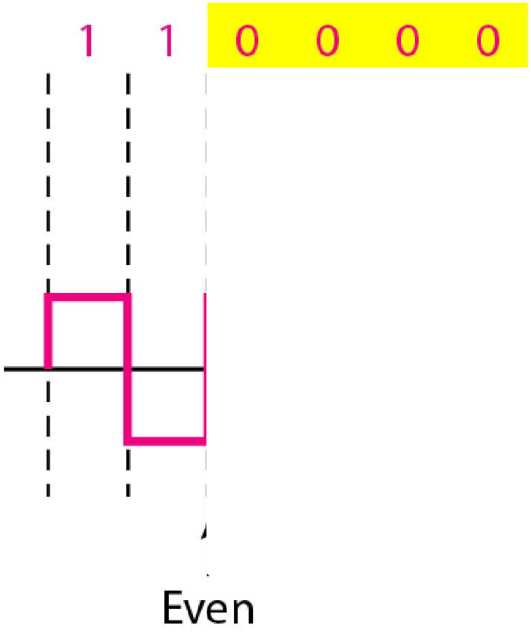
- Replaces 4 zeros by
  - 0 0 0 V if No. of nonzero pulses after last substitution is ODD
  - B 0 0 V if No. of nonzero pulses after last substitution is EVEN



Scrambling: *HDB3*

- High Definition Bipolar 3-Zero (*HDB*3)

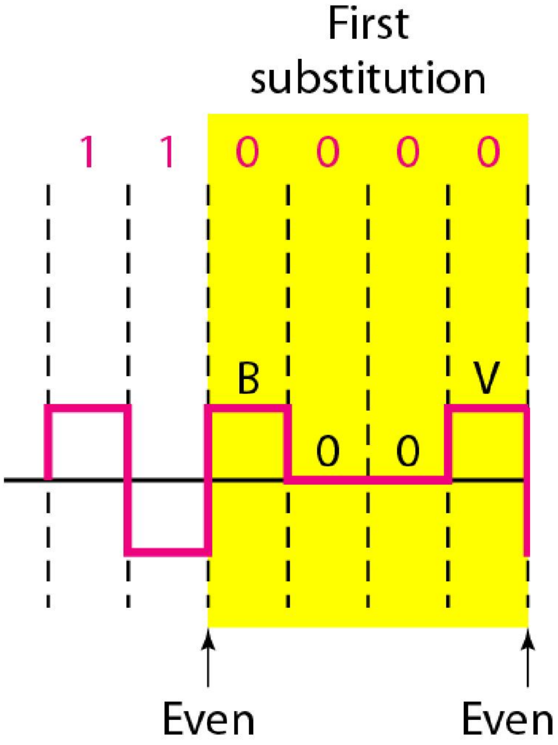
- Replaces 4 zeros by
  - 0 0 0 V if No. of nonzero pulses after last substitution is ODD
  - B 0 0 V if No. of nonzero pulses after last substitution is EVEN



Scrambling: *HDB3*

- High Definition Bipolar 3-Zero (*HDB*3)

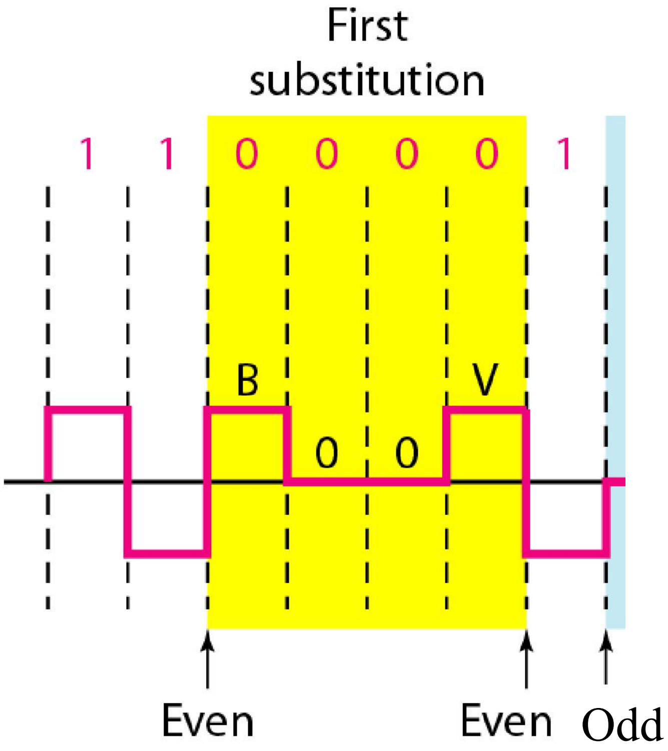
- Replaces 4 zeros by
  - 0 0 0 V if No. of nonzero pulses after last substitution is ODD
  - B 0 0 V if No. of nonzero pulses after last substitution is EVEN



Scrambling: *HDB3*

- High Definition Bipolar 3-Zero (*HDB*3)

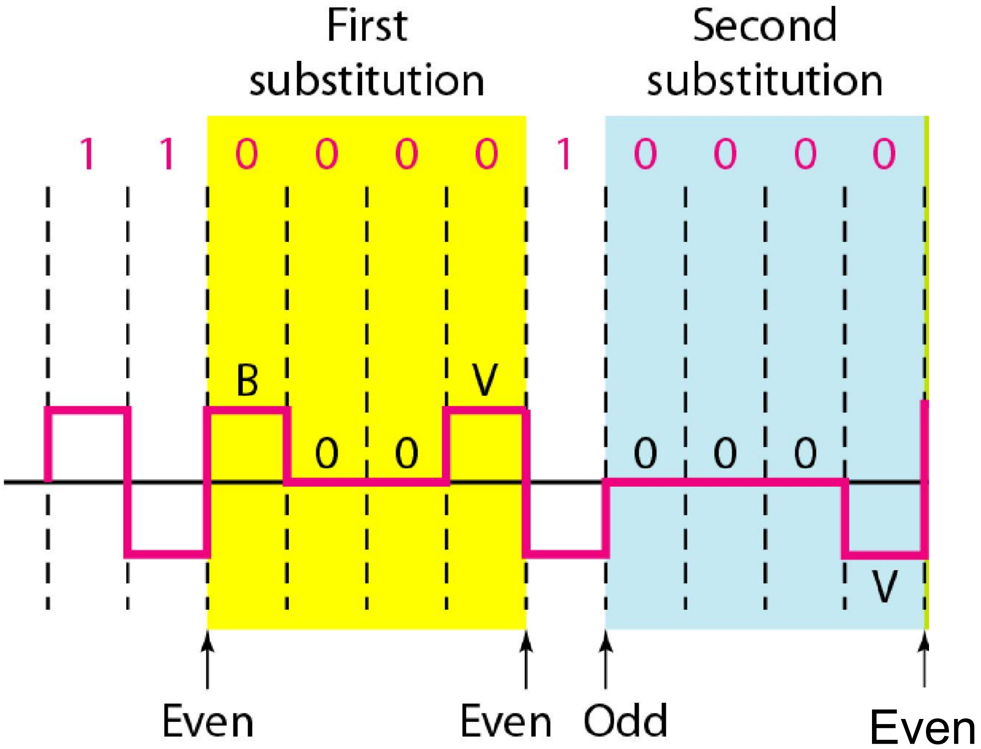
- Replaces 4 zeros by
  - 0 0 0 V if No. of nonzero pulses after last substitution is ODD
  - B 0 0 V if No. of nonzero pulses after last substitution is EVEN



Scrambling: *HDB3*

- High Definition Bipolar 3-Zero (*HDB*3)

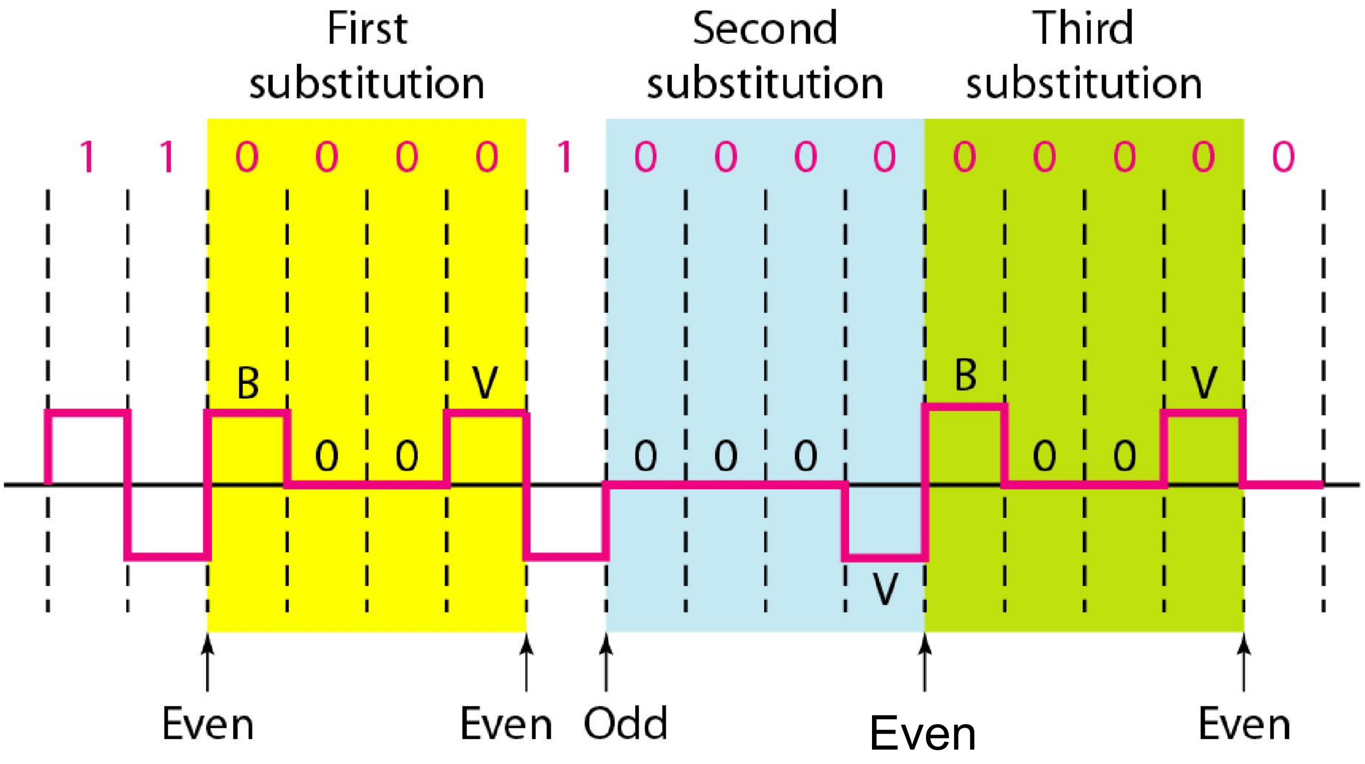
- Replaces 4 zeros by
  - 0 0 0 V if No. of nonzero pulses after last substitution is ODD
  - B 0 0 V if No. of nonzero pulses after last substitution is EVEN



Scrambling: *HDB3*

- High Definition Bipolar 3-Zero (*HDB*3)

- Replaces 4 zeros by
  - 0 0 0 V if No. of nonzero pulses after last substitution is ODD
  - B 0 0 V if No. of nonzero pulses after last substitution is EVEN


First substitution
Second substitution
Third substitution
1
1
0
0
0
0
1
0
0
0
0
0
0
0
0
0
B
0
0
V
0
0
0
V
B
0
0
V
Even
Even
Odd
Even
Even