LW-01. Digital Signals Line & Block Coding Schemes

1. Lab Target

Understand the coding schemes for digital signals. Get the experience of coding signals.

2. Lab Assignment

2.1. Create an individual variant of the original data.

a) Write your surname in the letters of the English alphabet. Must be at least 5 letters, if not enough, then add the required number of letters from the name.
For example, for Li Yuriy there will be LIYUR.

b) Replace the first 5 letters with their ordinal numbers in the alphabet, writing the numbers as two-digit decimal numbers (with leading zero).
For example, 12 09 25 21 18.

c) Translate each decimal two-digit number into a binary 5-bit number (with leading zero) and remove last bit.
For example, 01100 01001 11001 10101 1001~~0~~.

d) The resulting sequence of 24 bits is your version of the original sequence for encoding (the original data).
For example, original signal:* 011000100111001101011001*.

2.2. Create line coding schemes diagrams.

Create 10 diagrams illustrating various line, scrambling and block coding schemes for your original data sequence.

2.2.1. NRZ-I. Polar Coding None-Return-to-Zero Inverse.

2.2.2. NRZ-M. Polar coding None-Return-to-Zero Mark.

2.2.3. Biphase-I. Manchester coding by standard IEEE 802.3.

2.2.4. RZ-3. Three-level encoding with a Return-to-Zero.

2.2.5. AMI. Bipolar code Alternate Mark Inversion.

2.2.6. MLT-3. Multi Level Transmit – 3.

2.2.7. 2B/1Q. Four-level coding Two-Binary / One-Quaternary.

2.2.8. B8ZS - Bipolar with 8-Zeros Substitution (scrambling code).

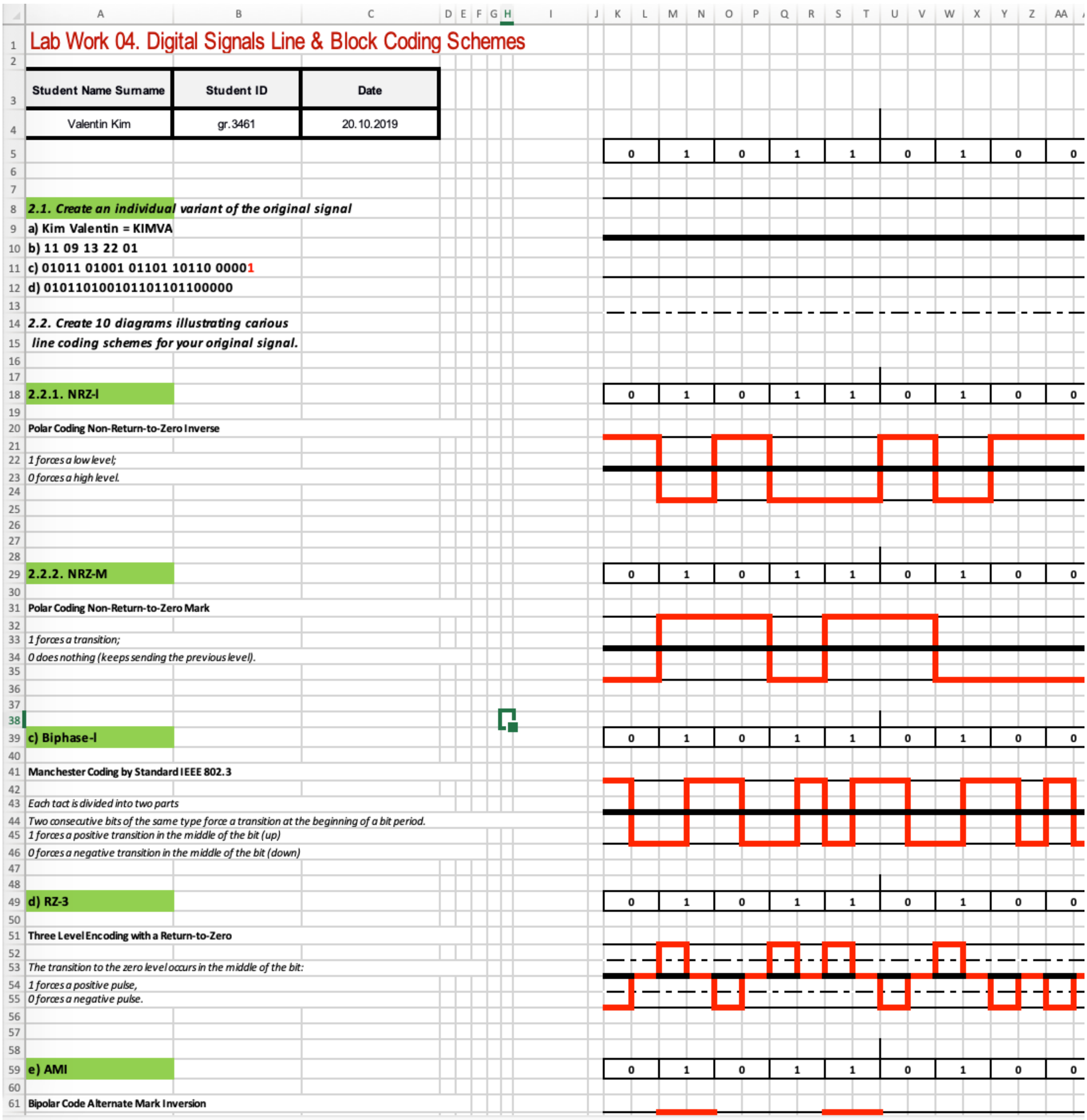
2.2.9. HDB3 - High-Density-Bipolar 3-Zeros (scrambling code).

2.2.10. 4B/5B. Block redundancy code.

3. REPORT

3.1. REPORT BLANK.

For graphical representation of diagrams, it is recommended to use a spreadsheet grid MS Excel (look Report Blank). All 10 lab assignment to one spreadsheet.



This example is not full!

3.2. GRADUATION.

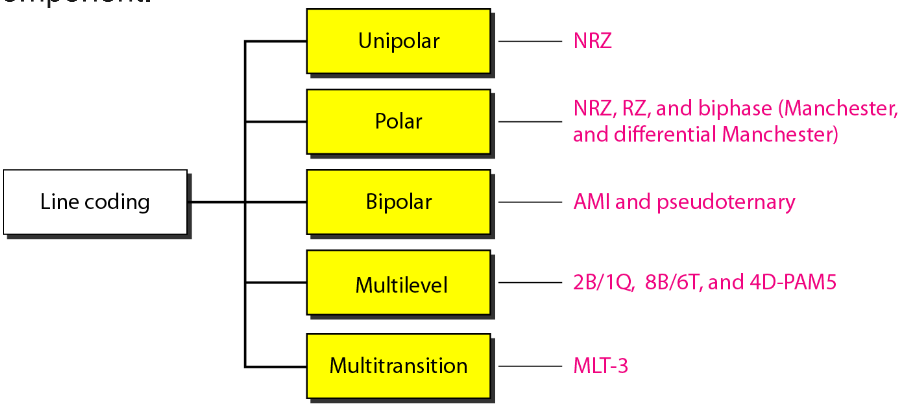
Grade on 10 points: correctly formed initial sequence and correctly made of all 10 variants of coding schemes.

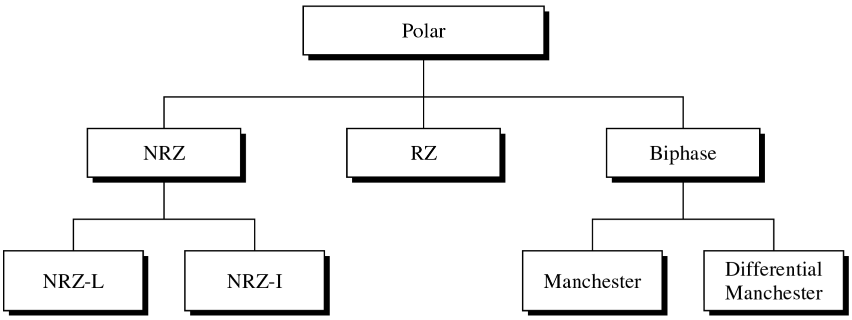
4. LAB GUIDELINES

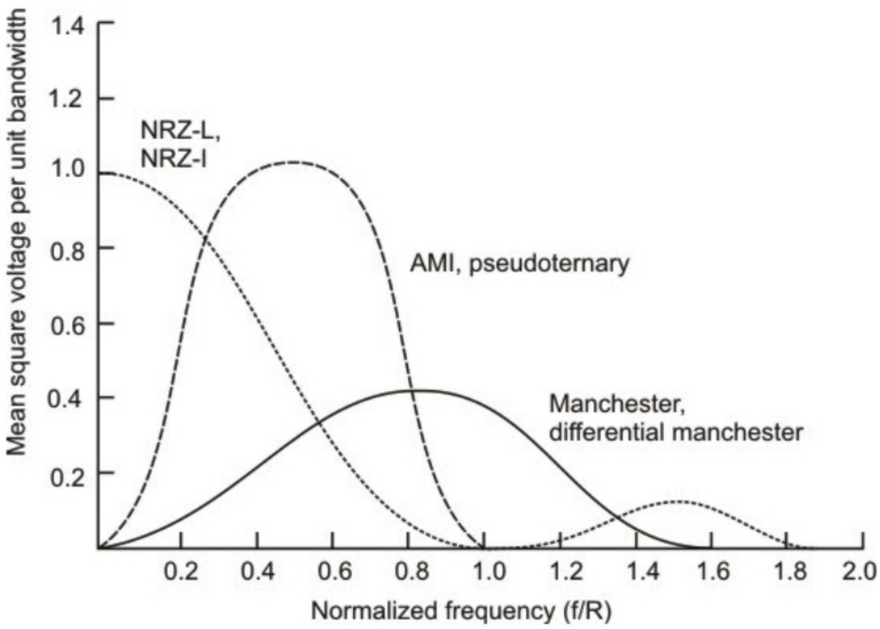
4.1. Theoretical basis.

Each line code has advantages and disadvantages.
The particular line code used is chosen to meet one or more of the following criteria:

- Minimize transmission hardware;
- Facilitate synchronization;
- Ease error detection and correction;
- Minimize spectral content;
- Eliminate a DC component.



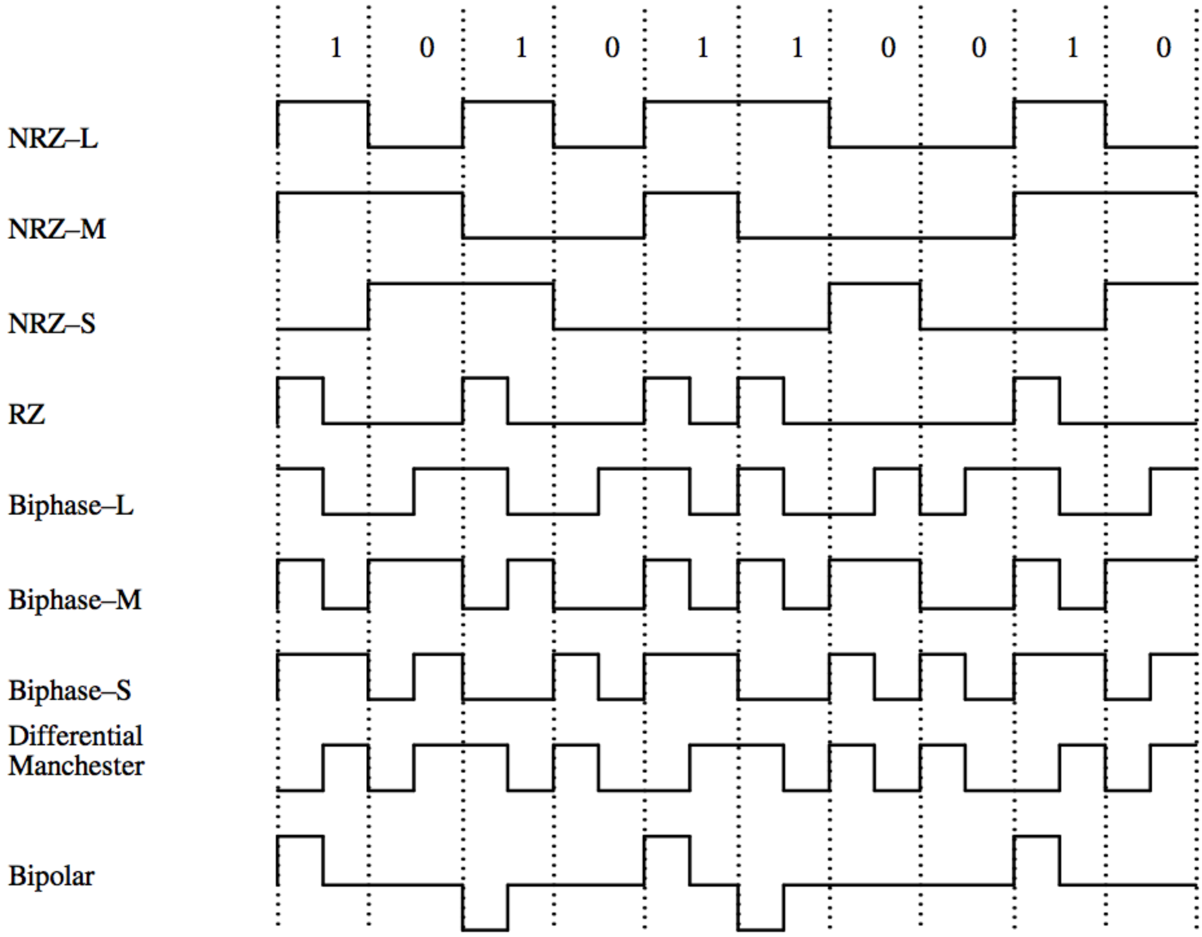


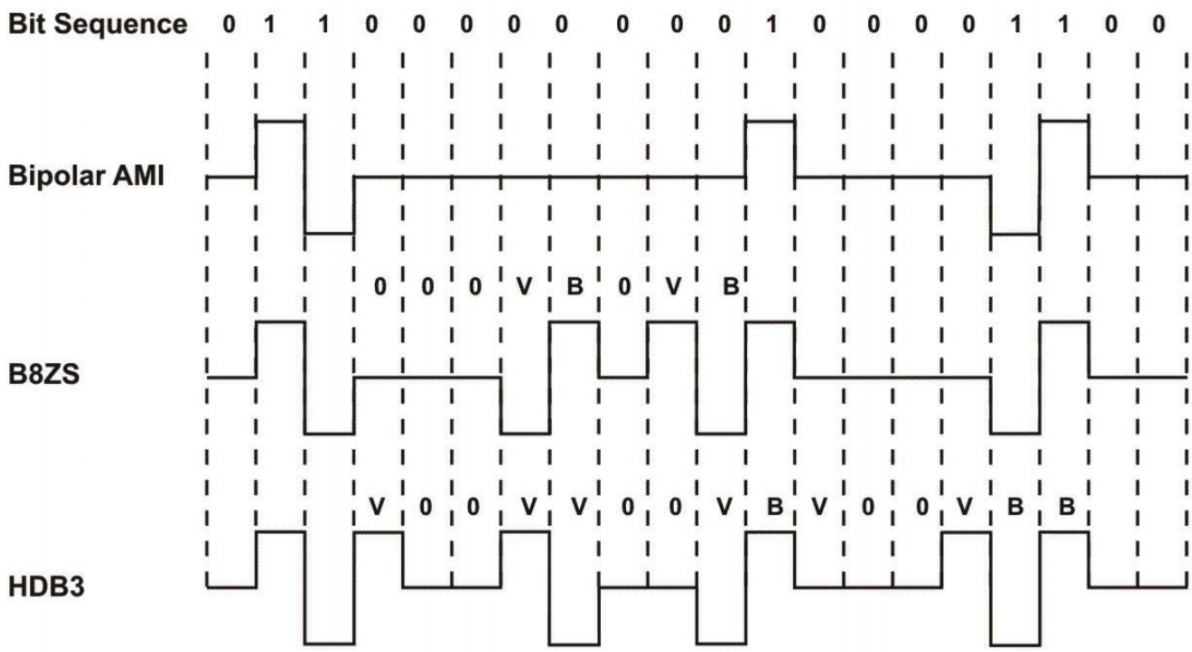


Frequency spectrum of different encoding schemes

Some of the more common binary line codes include:
Read more https://en.wikipedia.org/wiki/Line_code .

Shem	Comments
NRZ–L	Non return to zero level - is the standard positive logic signal format used in digital circuits. 1 forces a high level 0 forces a low level
NRZ–I	Non return to zero level inverse. This is the inverse variant NRZ-L. 1 forces a low level 0 forces a high level
NRZ–M	Non return to zero mark 1 forces a transition 0 does nothing (keeps sending the previous level)
NRZ–S	Non return to zero space 1 does nothing (keeps sending the previous level) 0 forces a transition
RZ	Return to zero 1 goes high for half the bit period and returns to low 0 stays low for the entire period
Biphase–L	Manchester as per G.E. Thomas. Two consecutive bits of the same type force a transition at the beginning of a bit period. 1 forces a negative transition in the middle of the bit 0 forces a positive transition in the middle of the bit
Biphase-I	Manchester (inverse) IEEE 802.3. Two consecutive bits of the same type force a transition at the beginning of a bit period. 1 forces a positive transition in the middle of the bit 0 forces a negative transition in the middle of the bit
Biphase–M	Variant of Differential Manchester. There is always a transition halfway between the conditioned transitions. 1 forces a transition 0 keeps level constant
Biphase–S	Differential Manchester used in Token Ring. There is always a transition halfway between the conditioned transitions. 1 keeps level constant 0 forces a transition
Bipolar	The positive and negative pulses alternate. 1 forces a positive or negative pulse for half the bit period 0 keeps a zero level during bit period
RZ-3	Return to zero 3 Level 1 forces a positive pulse for half the bit period 0 forces a negative pulse for half the bit period
AMI	AMI - Alternate Mark Inversion. The positive and negative pulses alternate. 1 forces a positive or negative pulse for the bit period 0 keeps a zero level during bit period
MLT-3	MLT-3 - Multi Level Transmit 3 1 forces a transition to next level 0 does nothing (keeps sending the previous level)
2B1Q	2B1Q - two-binary, one-quaternary. Transmits a pair of bits in one clock interval. Each possible pair of signals has its own level of four possible levels of potential: 00 this is -U; 01 this is -U/2; 11 this is +U/2; 10 this is +U.



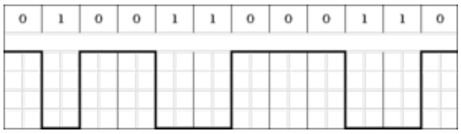


V= Bipolar Violation, B = Valid Bipolar Signal

4.2. RECOMMENDATIONS FOR CREATING DIAGRAMS OF LINEAR CODES SCHEMES.

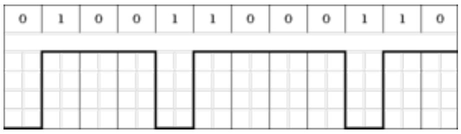
4.2.1. NRZ-I. Polar coding None-Return-to-Zero Inverse.

1 forces a low level
0 forces a high level



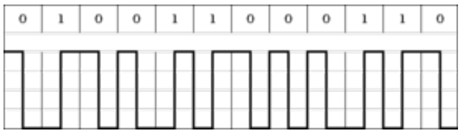
4.2.2. NRZ-M. Polar coding None-Return-to-Zero Mark.

1 forces a transition
0 does nothing (keeps sending the previous level)



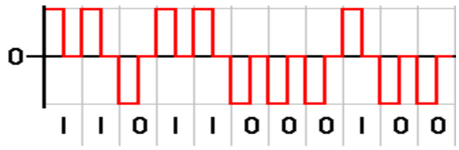
4.2.3. Biphase-I. Manchester coding by standard IEEE 802.3.

Each tact is divided into two parts.
Two or more consecutive bits of the same type force a transition at the beginning of a bit period.
1 forces a positive transition in the middle of the bit (up)
0 forces a negative transition in the middle of the bit (down)



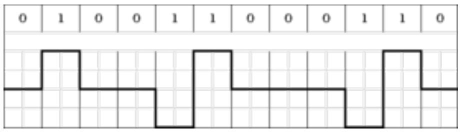
4.2.4. RZ-3. Three-level coding with a Return-to-Zero.

The transition to the zero level occurs in the middle of the bit:
1 forces a positive pulse,
0 forces a negative pulse.



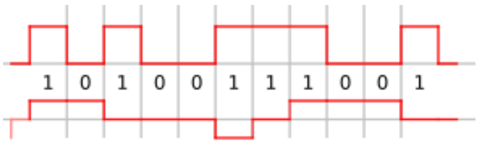
4.2.5. AMI. Bipolar code Alternate Mark Inversion.

The AMI code uses the following bit representations:
1 are represented alternately by the values -U or + U (B),
0 is represented by zero voltage (0 V).



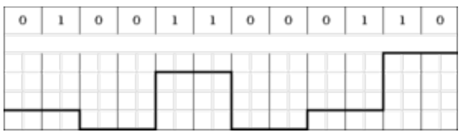
4.2.6. MLT-3. Multi Level Transmit – 3.

MLT-3 similar to the NRZ-M code, but has three signal levels.
1 corresponds to a transition from one signal level to another, and the signal level change takes place sequentially taking into account the previous transition;
0 the signal does not change.



4.2.7. 2B/1Q. Four-level coding Two-Binary / One-Quaternary.

Transmits a pair of bits in one clock interval. Each possible pair of signals has its own level of four possible levels of potential:
00 this is -U;
01 this is -U/2;
11 this is +U/2;
10 this is +U.



4.2.8. B8ZS - Bipolar with 8-Zeros Substitution (scrambling code).

Note for You. If your original sequence does not have eight zero's in a row, then replace the bits from 10 to 17 with 0's and perform B8ZS encoding. For example, this

011000100~~11100110~~10110010

replace to it

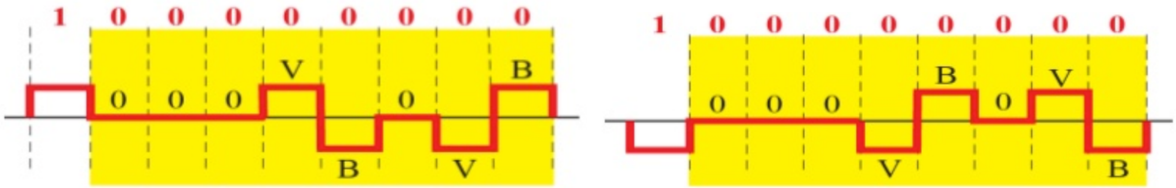
011000100**00000000**10110010

B8ZS used in North American T1. The B8ZS code is designed so that its DC component is zero for any sequence of binary digits. Read more https://en.wikipedia.org/wiki/Modified_AMI_code

B8ZS – this is modified AMI code, replaces each string of 8 consecutive zero's with the special pattern "000VB0VB" depending on the polarity of the preceding mark, where V=BipolarViolation, B=ValidBipolarSignal

A sequence of eight 0's is replaced by the following encoding:

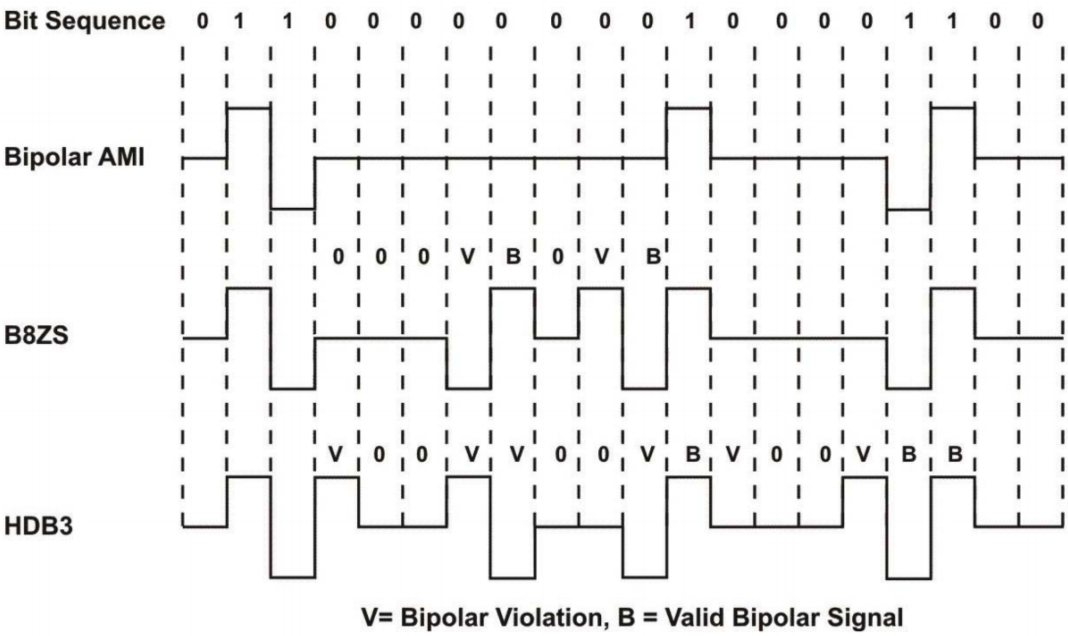
a) 000 + - 0 - +, if the previous pulse was positive (+).
b) 000 - + 0 + -, if the previous pulse was negative (-).



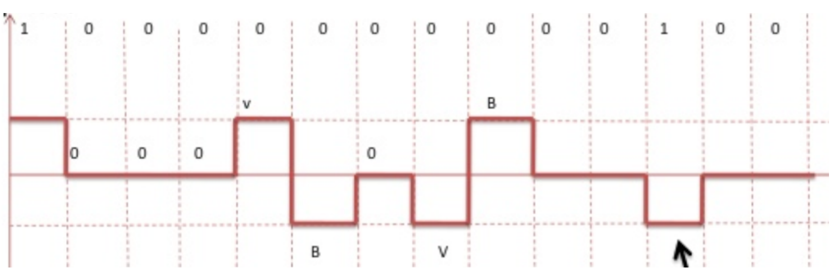
a. Previous level is positive. b. Previous level is negative.

As a result, at 8 clock cycles, the receiver observes 2 violations (V..V) - it is considered unlikely that the noise on the transmission line made this. Therefore, the receiver considers such violations to be encoded with 8 consecutive zeros and, after reception, replaces them with the original 8 zeros.

Example 1:



Example 2:



4.2.9. HDB3 - High-Density-Bipolar 3-Zeros (scrambling code).

Note for You. If in your original sequence there were no four zeros in a row 2 times, then replace bits from 5 to 8 and from 14 to 17 with 0's and perform HDB3 encoding. For example, this

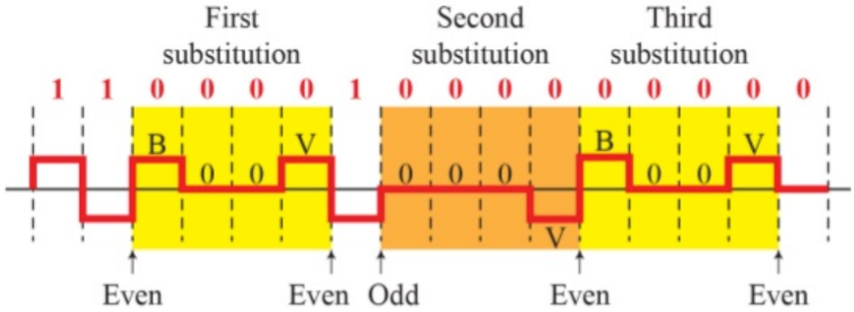
0110~~0010~~01110~~0110~~1011001

replace to it

0110~~0000~~01110~~0000~~1011001

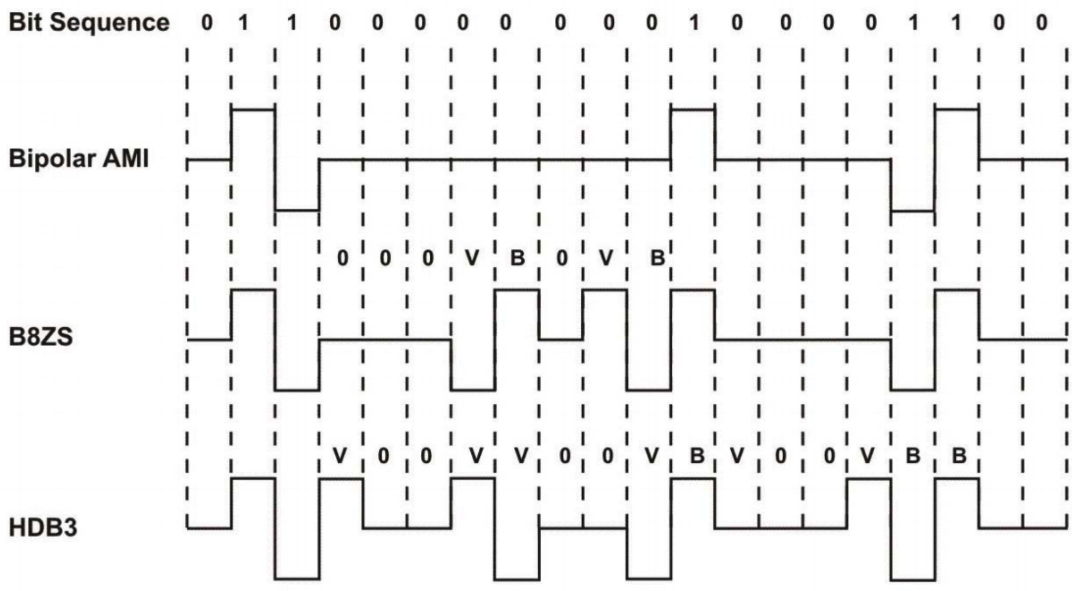
HDB3 used in European E-carrier 1,2,3. The HDB3 code is designed so that its DC component is zero for any sequence of binary digits. Read more https://en.wikipedia.org/wiki/Modified_AMI_code

HDB3 – this is modified AMI code, replaces each string of 4 consecutive zero's with the two special patterns "B00V" or "000V", where V=BipolarViolation, B=ValidBipolarSignal



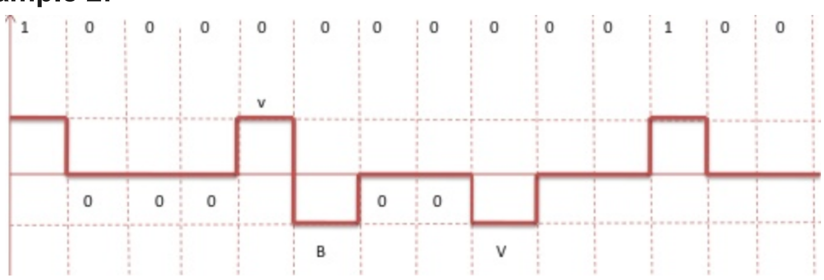
Rules for generating HDB3 code:
1. Every four zeros are replaced by four signals in which there is one signal V, which violates the polarity.
2. To suppress the DC component, the polarity of the V signal alternates with successive replacements.
3. Two samples of 4-bit codes are used for replacement:
    a) if source code contained odd number of 1's before the replacement, then sequence 000V is used;
    b) if the number of 1's was even, the sequence B00V is used.

Example 1:



V= Bipolar Violation, B = Valid Bipolar Signal

Example 2:



4.2.10. 4B/5B. Block Redundancy code 4 bit to 5 bit.

Read more https://en.wikipedia.org/wiki/4B5B

Redundant codes are based on dividing the original sequence of bits into chunks, which are called symbols. Then, each source symbol is replaced with a new one, which has more bits than the original.

For example, a 4B/5B block code (used in FDDI and Fast Ethernet technologies) replaces the original 4-bit characters with 5-bit characters.
Since the resulting symbols contain redundant bits, the total number of bit combinations in them is greater than in the original ones. So, in the 4B/5B code, the resulting characters can contain 2ˆ5 = 32 code words, while the source characters can only contain 2ˆ4 = 16, so the number of redundant code words is 32-16 = 16.

Therefore, in the resulting code, one can select 16 such combinations that do not contain a large number of zeros, and consider the rest as prohibited codes (code violation).

The correspondence of the source and resulting codes 4B/5B is presented below

Data Codes

4B Code	5B Symbol
0000	11110
0001	01001
0010	10100
0011	10101
0100	01010
0101	01011
0110	01110
0111	01111
1000	10010
1001	10011
1010	10110
1011	10111
1100	11010
1101	11011
1110	11100
1111	11101

Control and Invalid Codes

4B Code	5B Symbol
idle	11111
start of stream	11000
start of stream	10001
end of stream	01101
end of stream	00111
transmit error	00111
invalid	00000
invalid	00001
invalid	00010
invalid	00011
invalid	00100
invalid	00101
invalid	00110
invalid	01000
invalid	10000
invalid	11001

In addition to eliminating the constant component and imparting self-synchronization properties to the code, redundant codes allow the receiver to recognize distorted bits. If the receiver receives a forbidden code, it means that a signal distortion has occurred on the line.

And also, redundant codes allow you to transmit control commands.
The 4B / 5B code is then transmitted over the line using physical encoding according to one of the potential encoding methods, sensitive only to long sequences of zeros, for example, AMI.

5-bit 4B / 5V code symbols ensure that no combination of more than three consecutive zeros can occur on any line.

Example.

4B	1	0	1	1		0	1	0	0		1	0	1	1		0	1	1	0		1	1	0	0	
5B	1	0	1	1	1	0	1	0	1	0	1	0	1	1	1	0	1	1	1	0	1	1	0	1	0