Digital Transmission

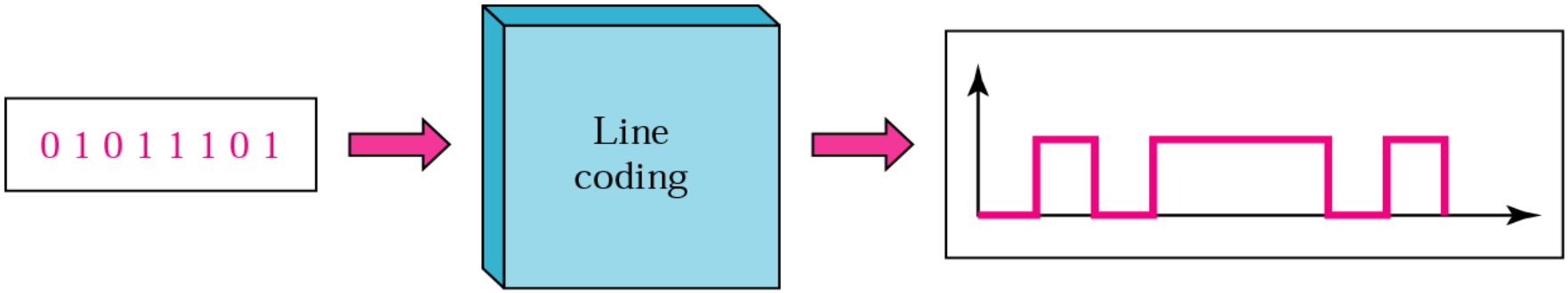
Line Coding

Some Characteristics

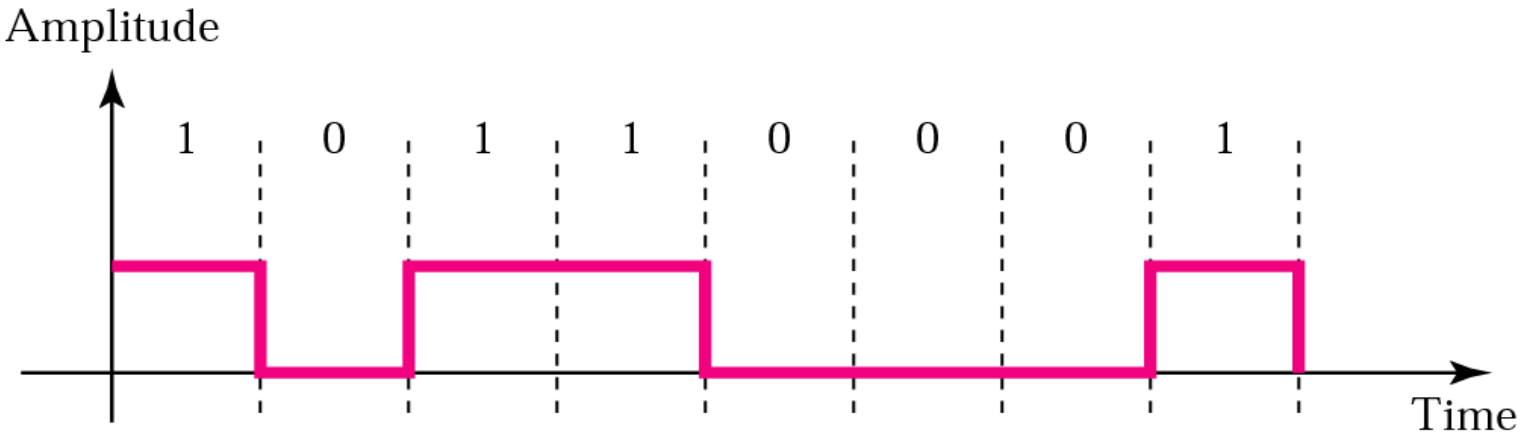
Line Coding Schemes

Some Other Schemes

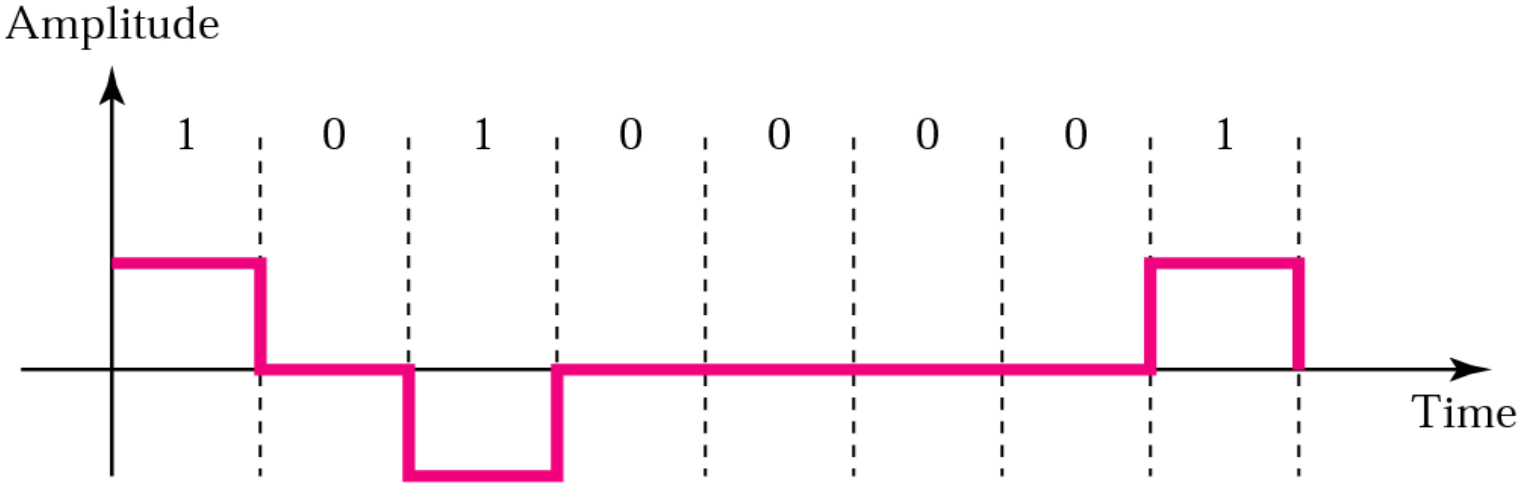
Line coding



Signal level versus data level

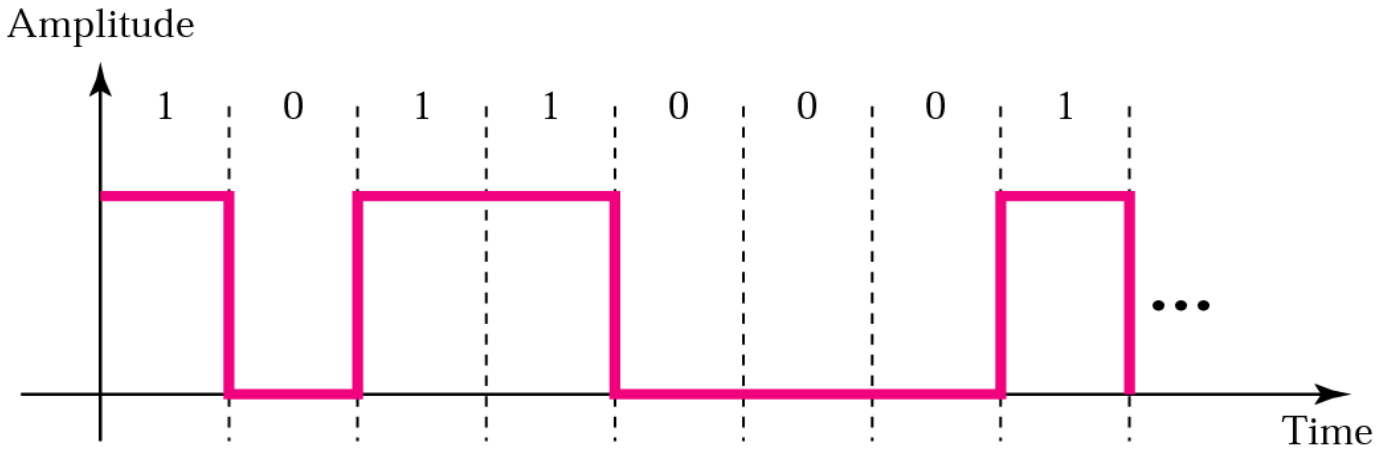


a. Two signal levels, two data levels

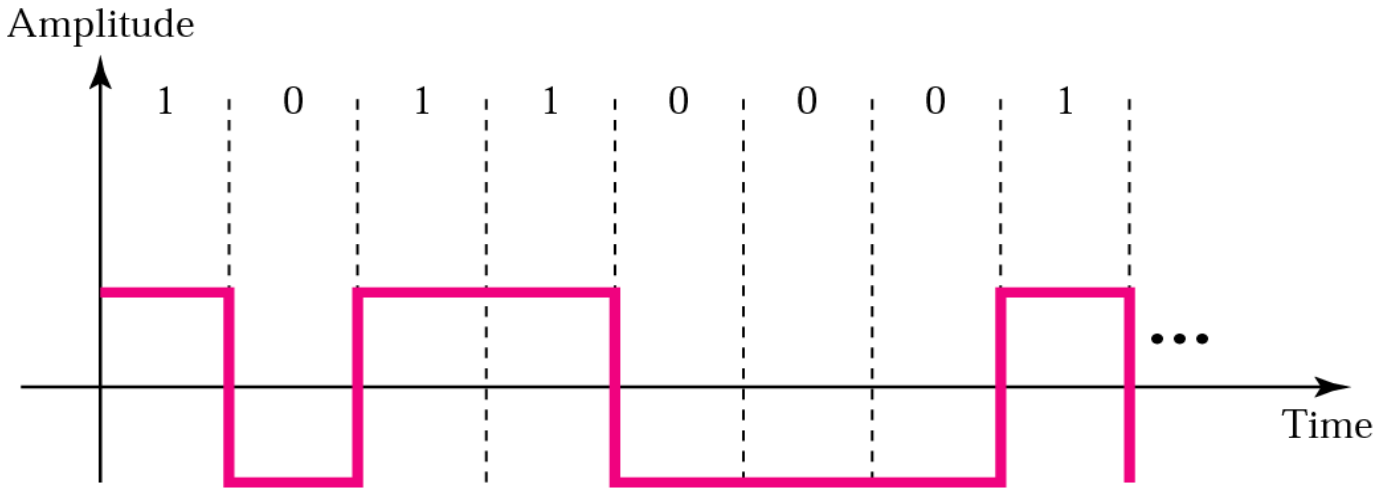


b. Three signal levels, three data levels

DC component



a. A signal with dc component



b. A signal without dc component

Pulse Rate versus Bit Rate

Bit Rate = Pulse Rate x Log_2 L

where L= number of data levels of signals

A signal has **two data levels** with a pulse duration of 1ms. We calculate the pulse rate and bit rate as follows:

Pulse Rate = $1/ 10^{-3} = 1000$ pulses/s

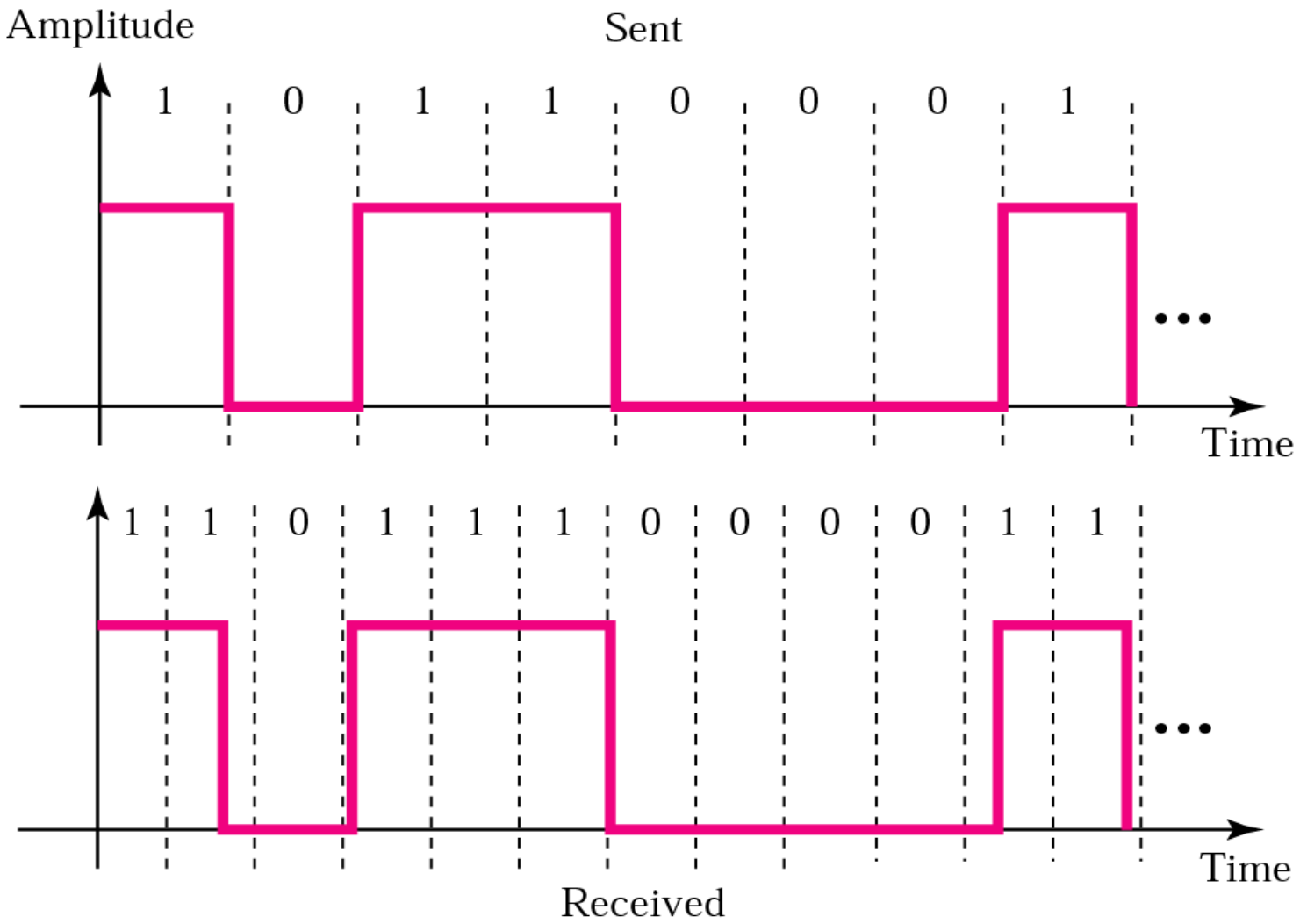
Bit Rate = Pulse Rate x $\log_2$ L = 1000 x $\log_2$ 2 = 1000 bps

A signal has **four data levels** with a pulse duration of 1ms. We calculate the pulse rate and bit rate as follows:

Pulse Rate = $1/ 10^{-3}$ = 1000 pulses/s

Bit Rate = PulseRate x $\log_2$ L = 1000 x $\log_2$ 4 = 2000 bps

Lack of synchronization



In a digital transmission, the receiver clock is 0.1 percent faster than the sender clock. How many extra bits per second does the receiver receive if the data rate is 1 Kbps? How many if the data rate is 100Mbps? 1 GBps ?

At 1 Kbps:

1000 bits sent ➔1001 bits received➔1 extra bps

At 100Mbps:

100,000,000 bits sent ➔100,0100,000 bits received➔100,000 extra bps

At 1 Gbps:

1000,000,000 bits sent ➔1,001,000,000 bits received➔1,000,000 extra bps

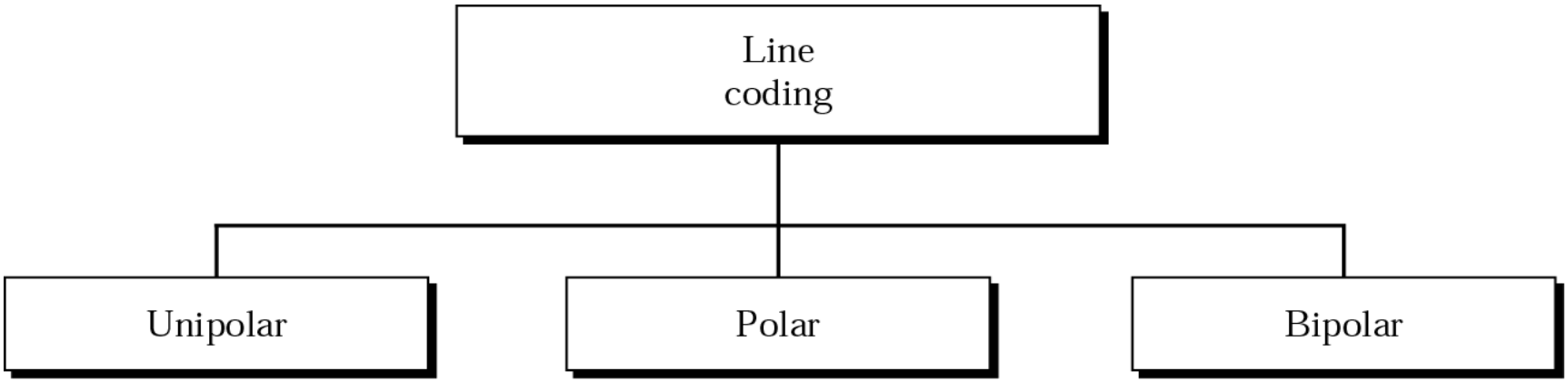
Self Synchronization

A self synchronization digital signal includes timing information in the data being transmitted.

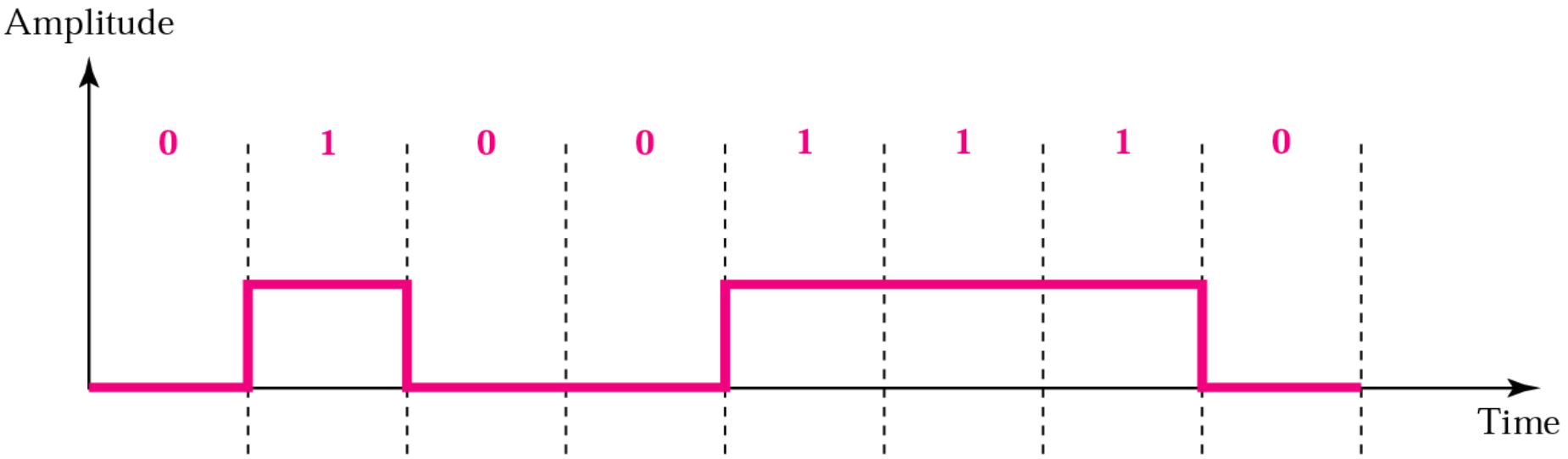
This can be achieved if there are transitions in the signal that alert the receiver to the

- beginning,
- middle or
- end of the pulse.

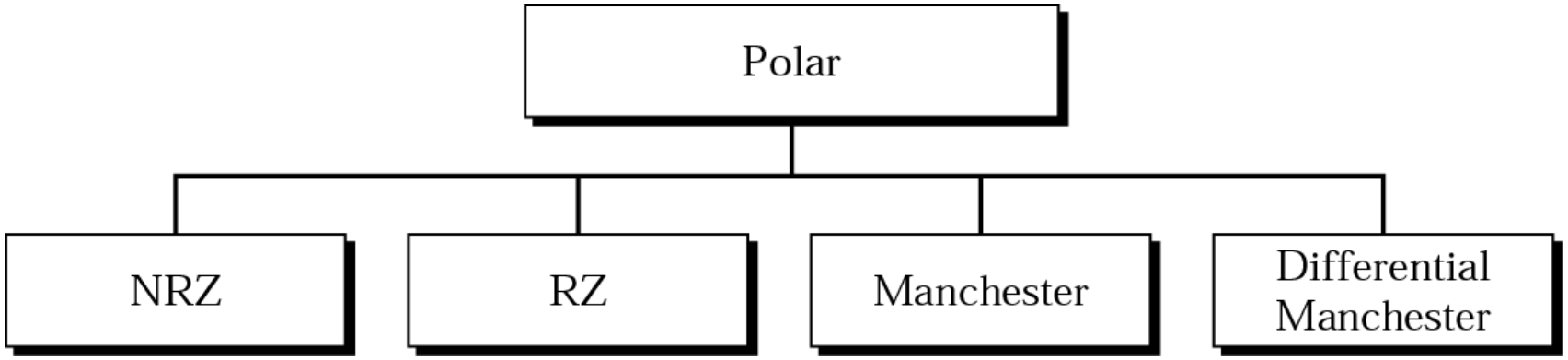
Line coding schemes



Unipolar encoding uses only one voltage level.

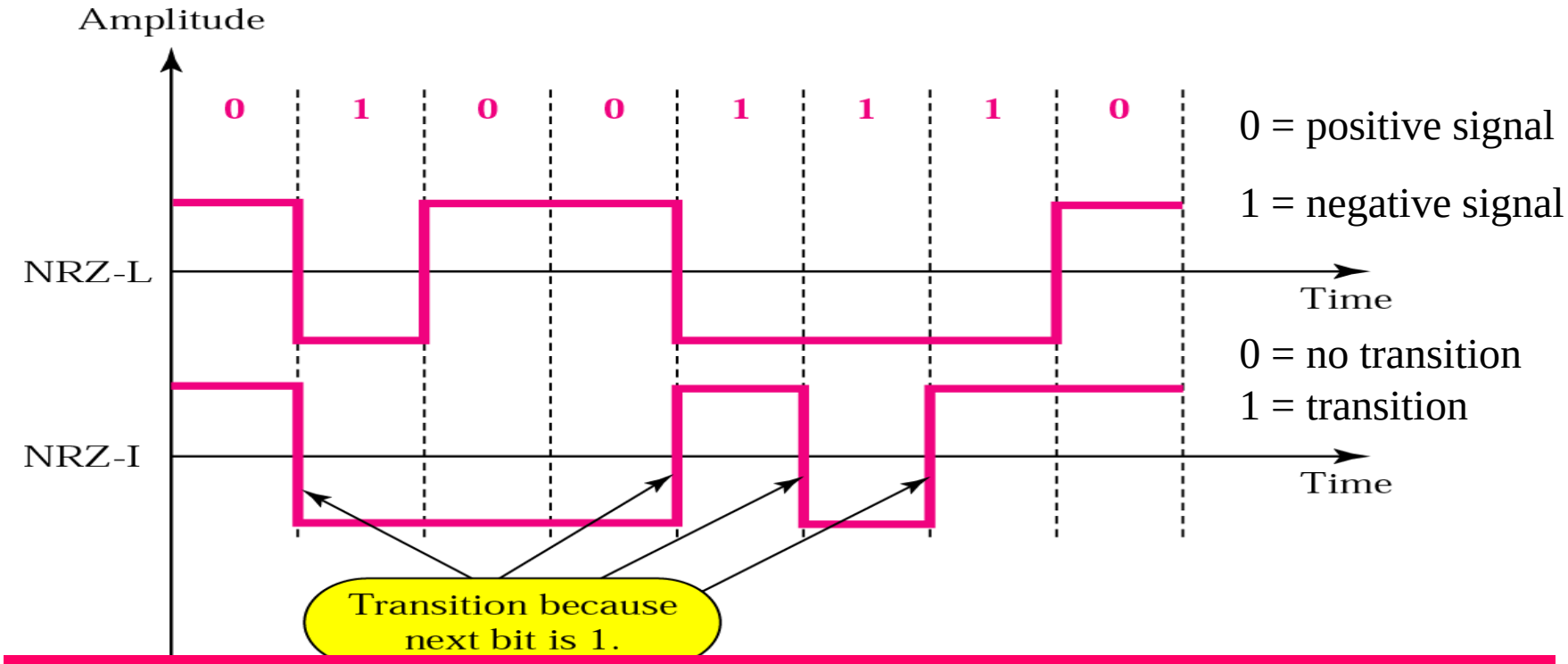


Polar encoding uses two voltage levels (positive and negative).



NRZ-L (Non Return to zero- Level) and NRZ-I (Non Return to zero –Invert encoding

In NRZ-L the level of the signal is dependent upon the state of the bit.



In NRZ-I the signal is inverted if a 1 is encountered.

If the original data has a string of consecutive 0 the receiver can lose its place and cannot do the "galvanic separation"

Nonreturn to Zero-Level (NRZ-L)

- Two different voltages for 0 and 1 bits
- Voltage constant during bit interval
  - no transition I.e. no return to zero voltage
- e.g. Absence of voltage for zero, constant positive voltage for one
- More often, negative voltage for one value and positive for the other
- This is NRZ-L

Nonreturn to Zero Inverted

- *Nonreturn to zero inverted on ones*
- *Constant voltage pulse for duration of bit*
- *Data encoded as presence or absence of signal transition at beginning of bit time*
- *Transition (low to high or high to low) denotes a binary 1*
- *No transition denotes binary 0*

NRZ pros and cons

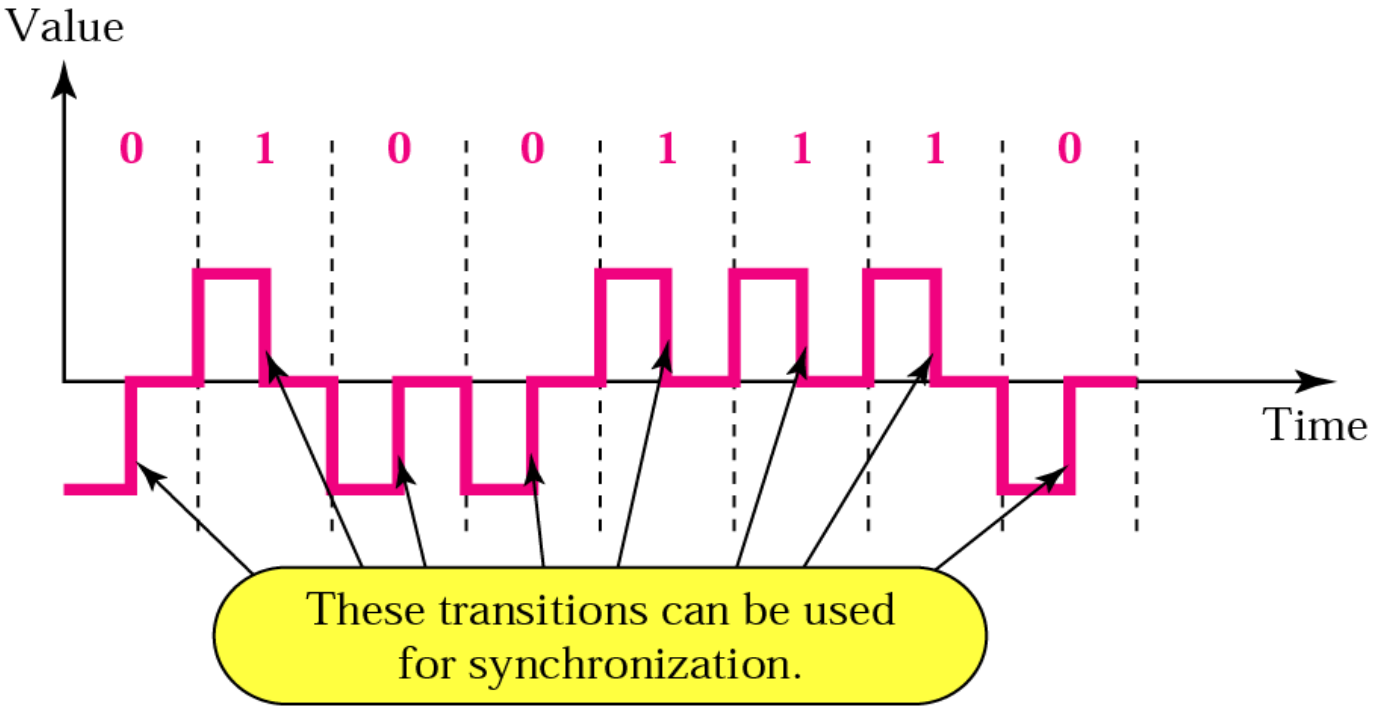
- *Pros*
  - *Easy to engineer*
  - *Make good use of bandwidth*
- *Cons*
  - *dc component*
  - *Lack of synchronization capability*
- *Used for magnetic recording*
- *Not often used for signal transmission*

RZ (Return to zero) encoding

RZ uses three values :positive; negative and zero.
The signal changes during each bit.
Halfway each bit interval the signal return to zero

1 = transition from positive to zero

0 = transition from negative to zero



A good encoded digital signal must contain a provision for synchronization.

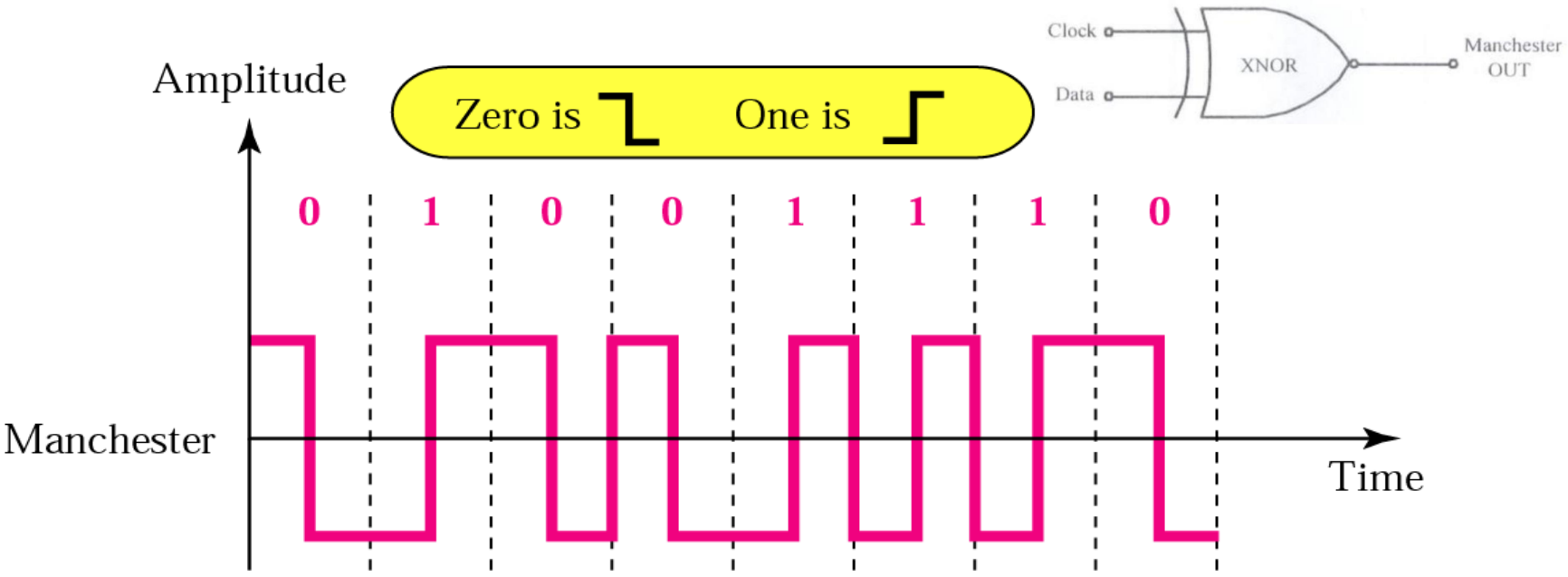
- *Manchester encoding*
- *Differential Manchester encoding*
- *Bipolar AMI (Alternate mark inversion) encoding*
- *2B1Q Two binary one quaternary)*
- *MLT-3 signal Multi-line transmission, three level*

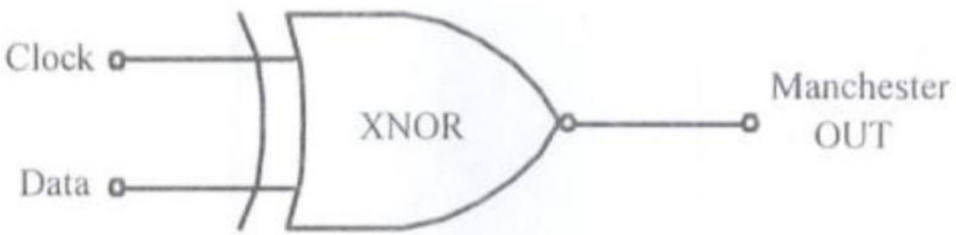
Biphase

- Manchester
  - Transition in middle of each bit period
  - Transition serves as clock and data
  - Low to high represents one
  - High to low represents zero
  - Used by IEEE 802.3
- Differential Manchester
  - Midbit transition is clocking only
  - Transition at start of a bit period represents zero
  - No transition at start of a bit period represents one
  - Note: this is a differential encoding scheme
  - Used by IEEE 802.5

Manchester encoding

In Manchester encoding, the transition at the middle of the bit is used for both synchronization and bit representation.





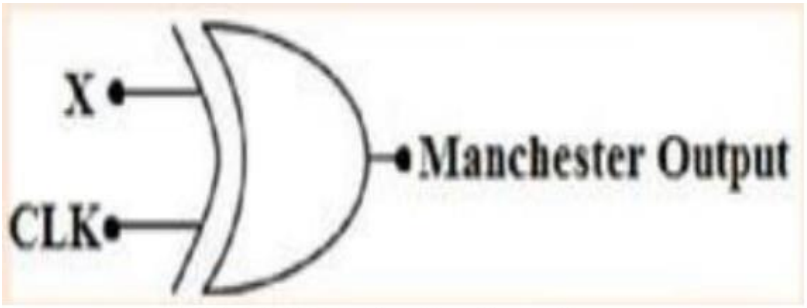
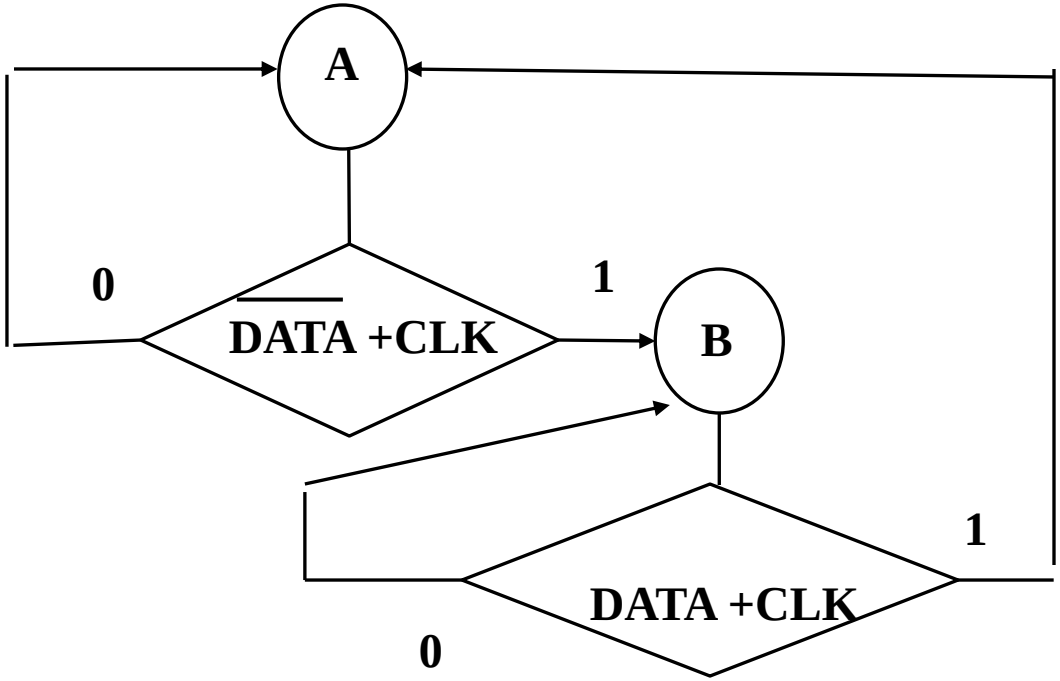


Fig.1.Manchester Encoder

$$Manchester\ Encoder = X \oplus CLK \quad (1)$$



Manchester encoding.

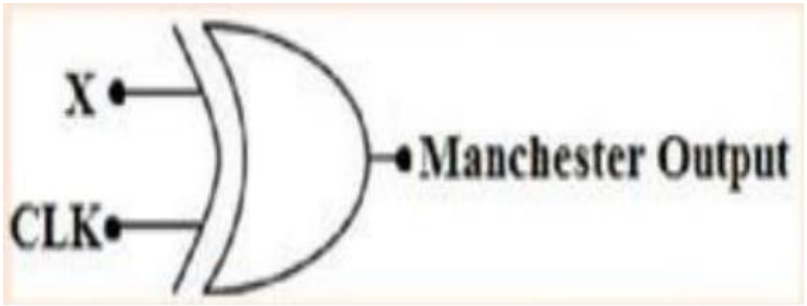


Fig.1.Manchester Encoder

$$Manchester\ Encoder = X \oplus CLK \quad (1)$$

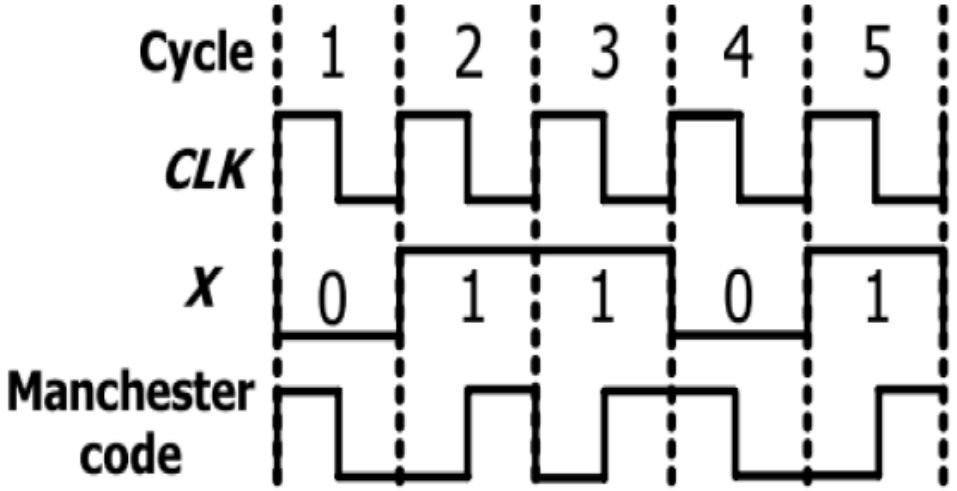


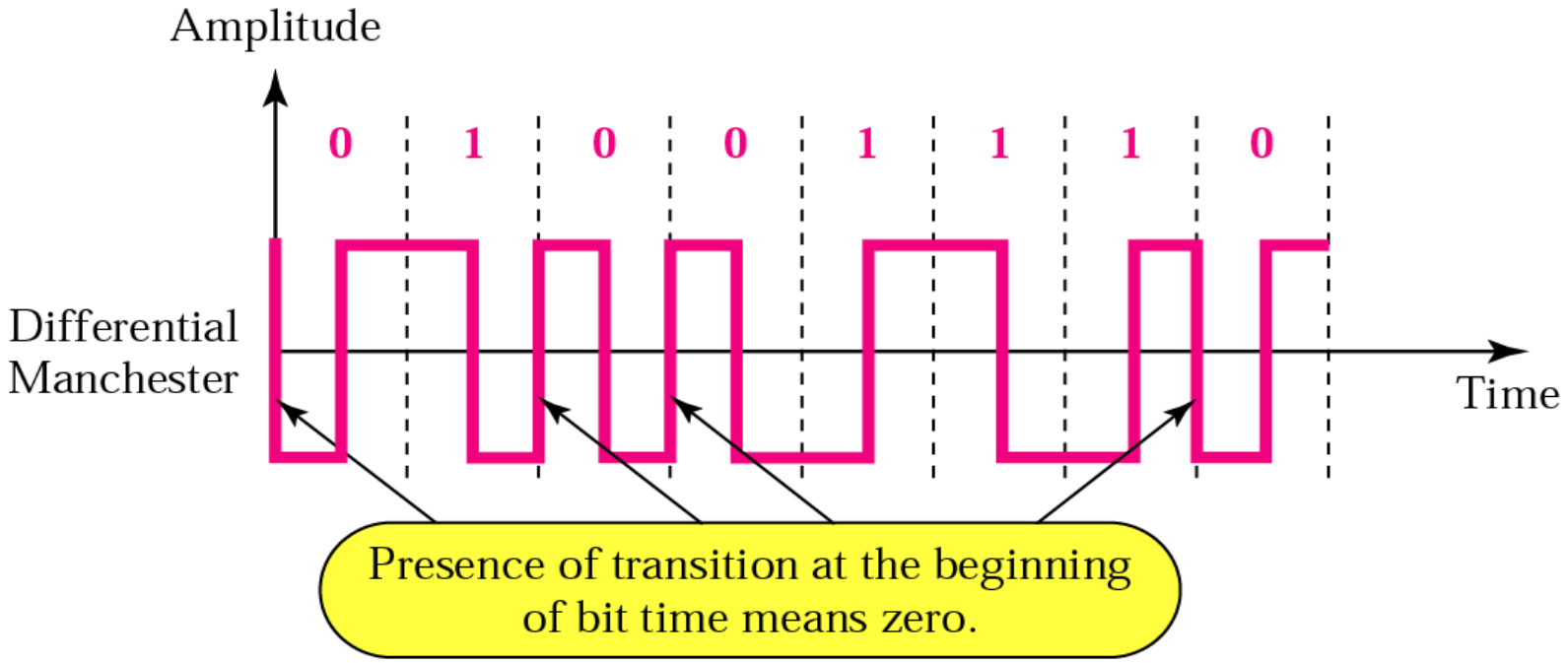
Fig-2: Example for Manchester encoding

Table II: Operation of Manchester Encoding

Input Data	Clock Signal	Manchester Code
0	1	1
	0	0
1	1	0
	0	1
1	1	0
	0	1
0	1	1
	0	0
1	1	0
	0	1

Differential Manchester encoding

In differential Manchester encoding, the transition at the middle of the bit is used only for synchronization. The bit representation is defined by the inversion "bit 0" or noninversion "bit 1"at the beginning of the bit.

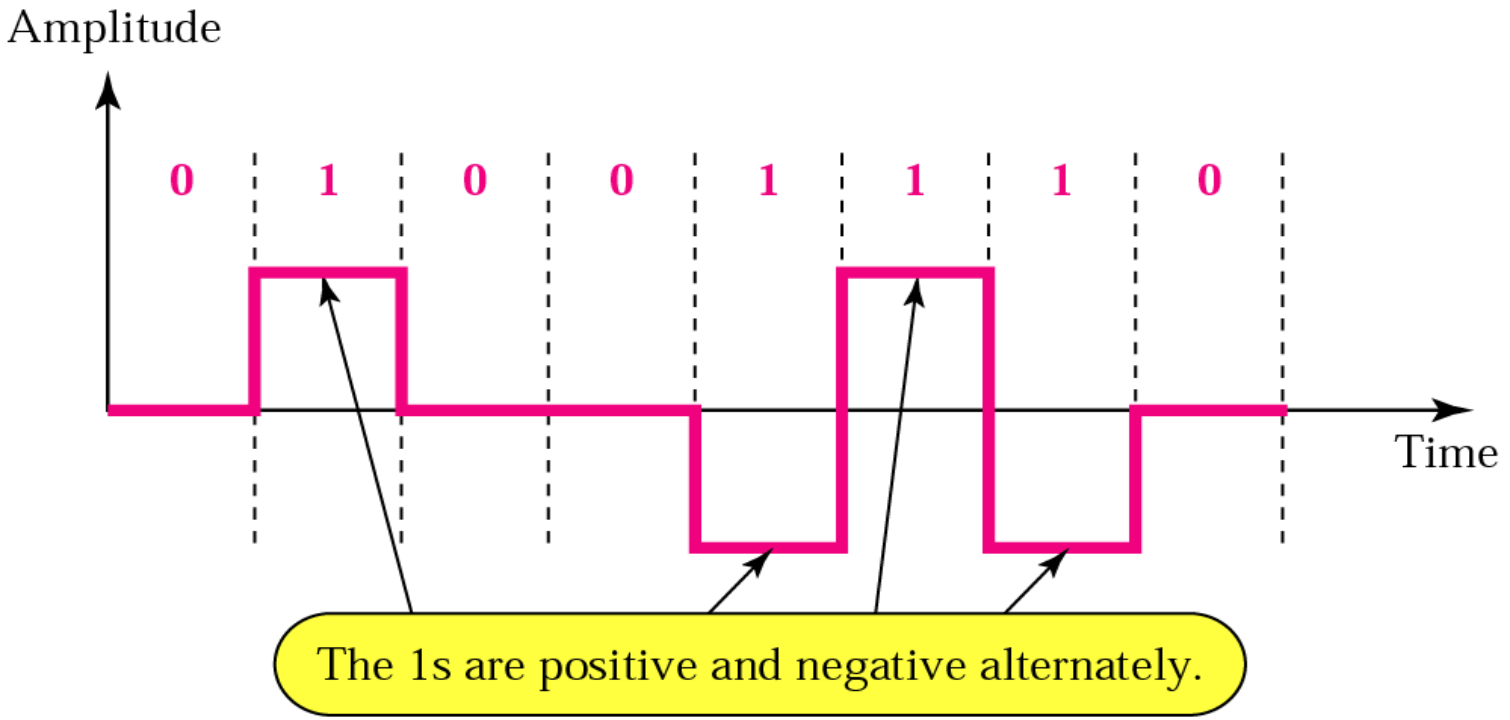


Biphase Pros and Cons

- Cons
  - At least one transition per bit time and possibly two
  - Maximum modulation rate is twice NRZ
  - Requires more bandwidth
- Pros
  - Synchronization on mid bit transition (self clocking)
  - No dc component
  - Error detection
    - Absence of expected transition

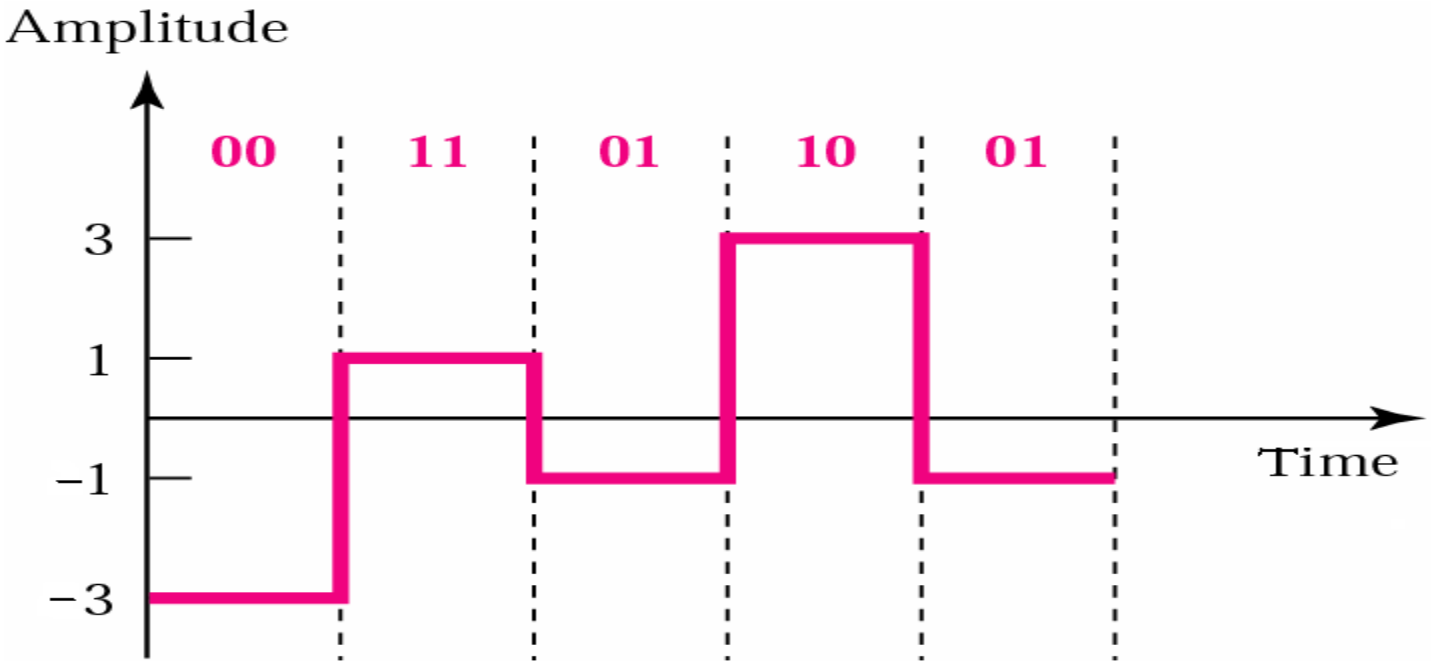
Bipolar AMI (Alternate mark inversion) encoding

In bipolar encoding, we use three levels: positive, zero and negative.
A neutral voltage represents "bit 0"
"Bit 1" is represented by alternating positive and negative voltages.



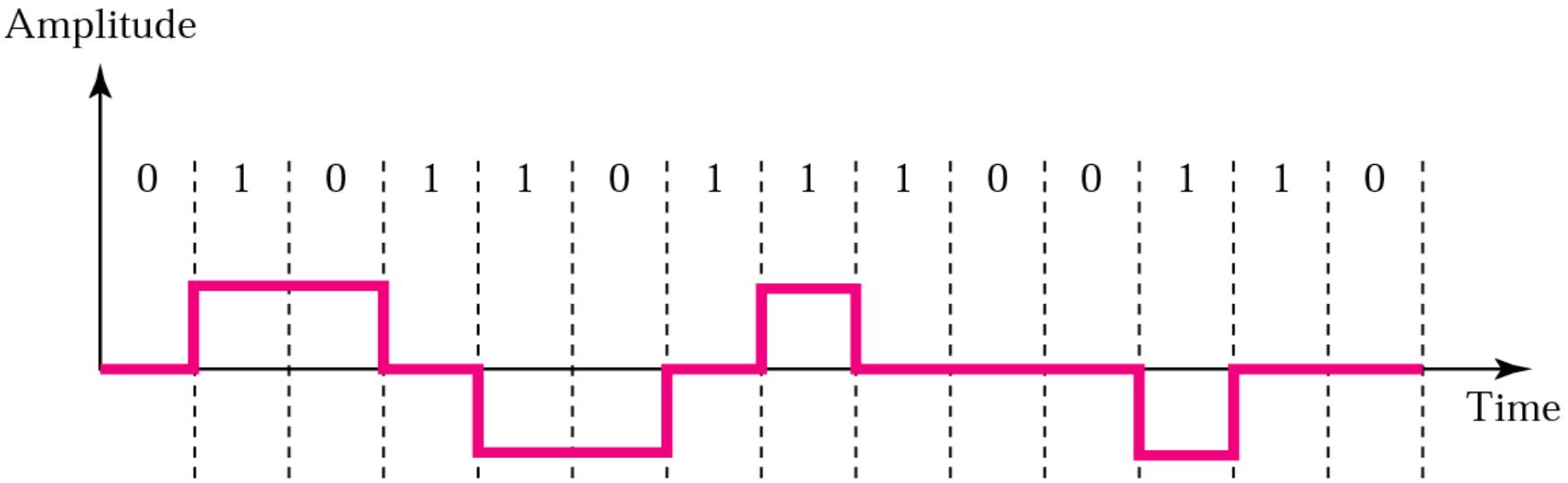
2B1Q (Two binary one quaternary)

Uses four voltage levels. Each pulse can then represent 2 bits, making each pulse more efficient.



MLT-3 signal Multi-line transmission, three level

It is similar to NRZ-I but it uses three levels of signals (+1, 0 and -1)



There is no transition at the beginning of 0 bit,
The signal transitions from one level to the next at the beginning of a 1 bit

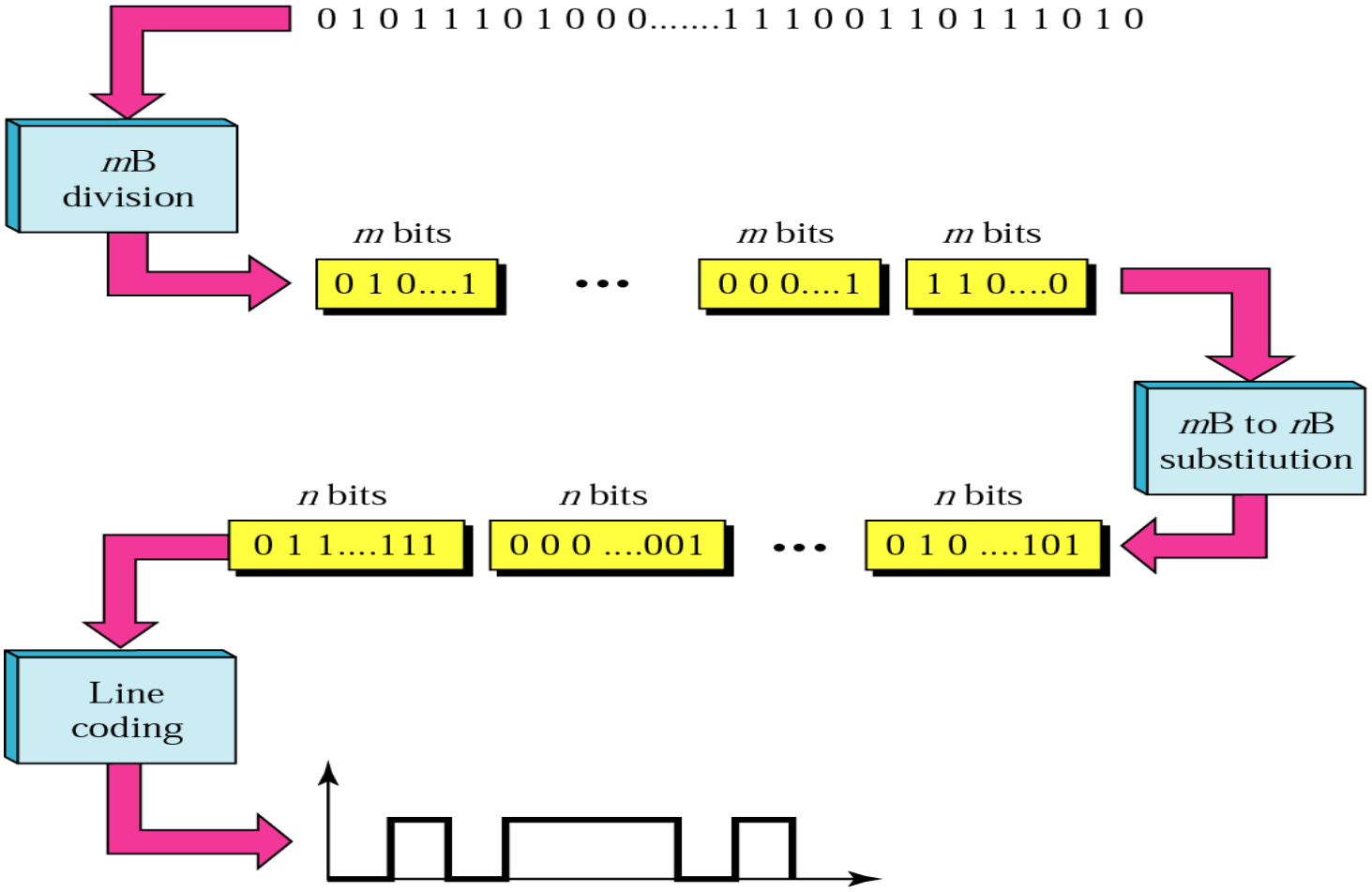
Block Coding

Steps in Transformation

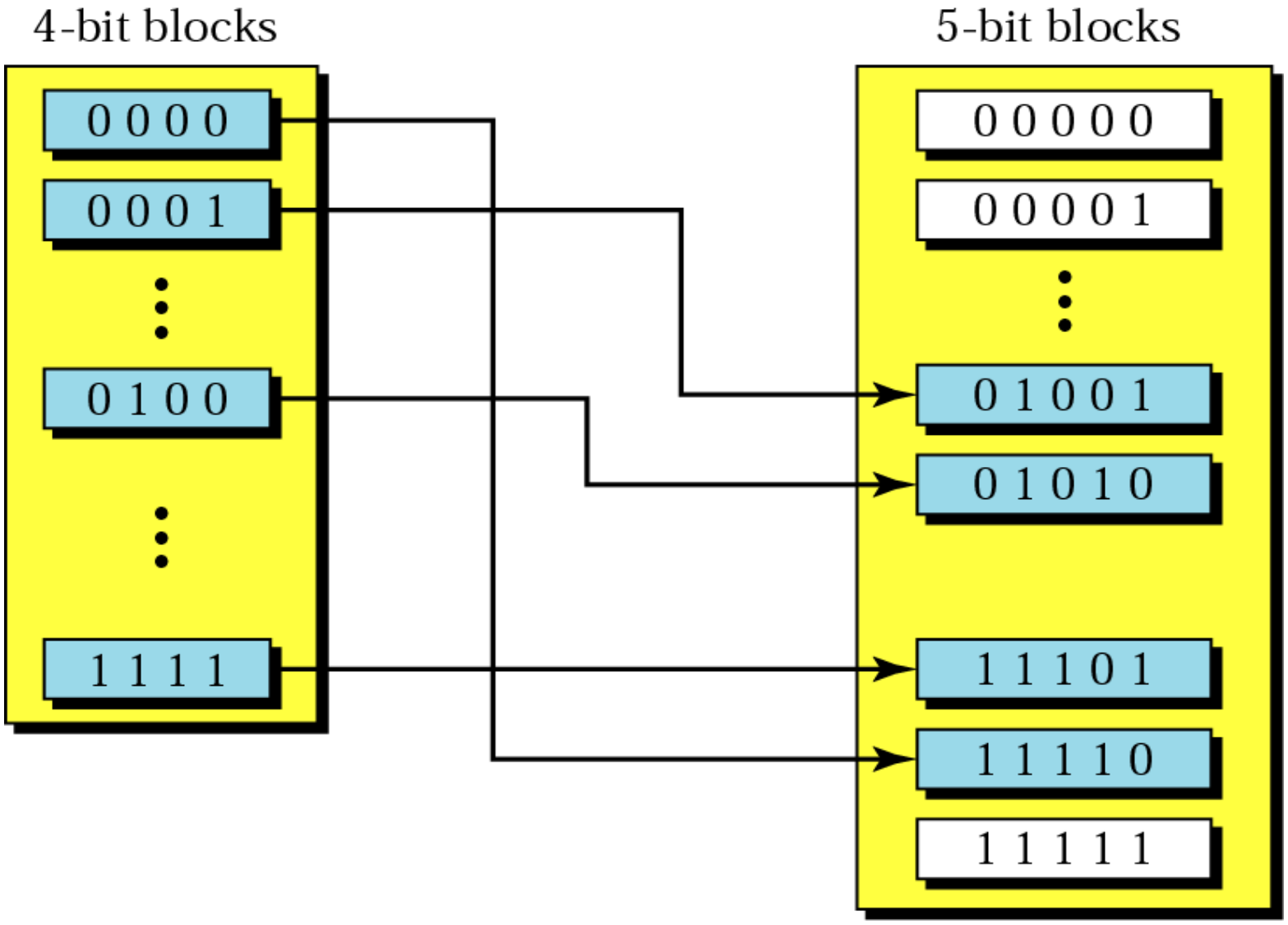
Some Common Block Codes

Block coding

We need some kind of redundancy to ensure synchronization.
We need to include other redundant bits to detect errors.
Block coding can achieve these two goals.



Substitution in block coding



4B/5B encoding

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

5 bit code

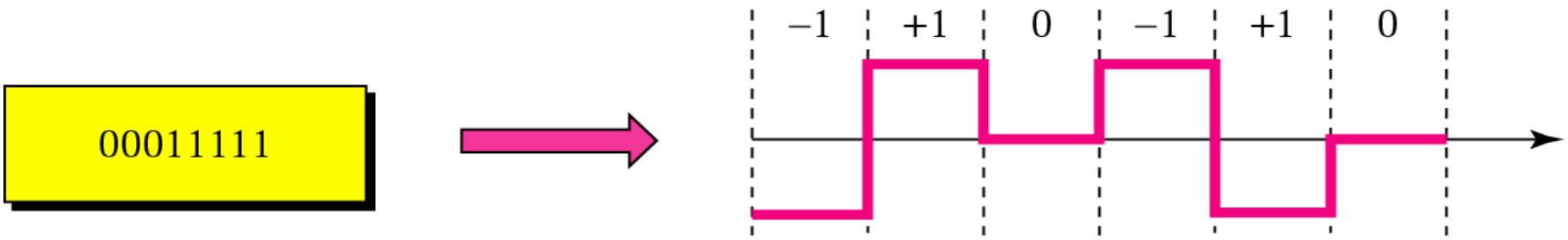
⇒No more than three consecutive 0s on sequence of data

⇒Some codes are used for synchronization and error detection

4B/5B encoding (Continued)

Data	Code
Q (Quiet)	**00000**
I (Idle)	**11111**
H (Halt)	**00100**
J (start delimiter)	**11000**
K (start delimiter)	**10001**
T (end delimiter)	**01101**
S (Set)	**11001**
R (Reset)	**00111**

Example of 8B/6T encoding



8B/6T encoding is design to substitute an 8-bit group with six symbol code
Each symbol is ternary, having one of three signal levels (+1,0,-1)
An 8-bit code can represent 256 possibilities (2^8)
A six symbol ternary signal can represent 729 possibilities (3^6)