Lab 8: Digital Modulation

SUMMER CHALLENGE COURSE

SMART LIGHTING

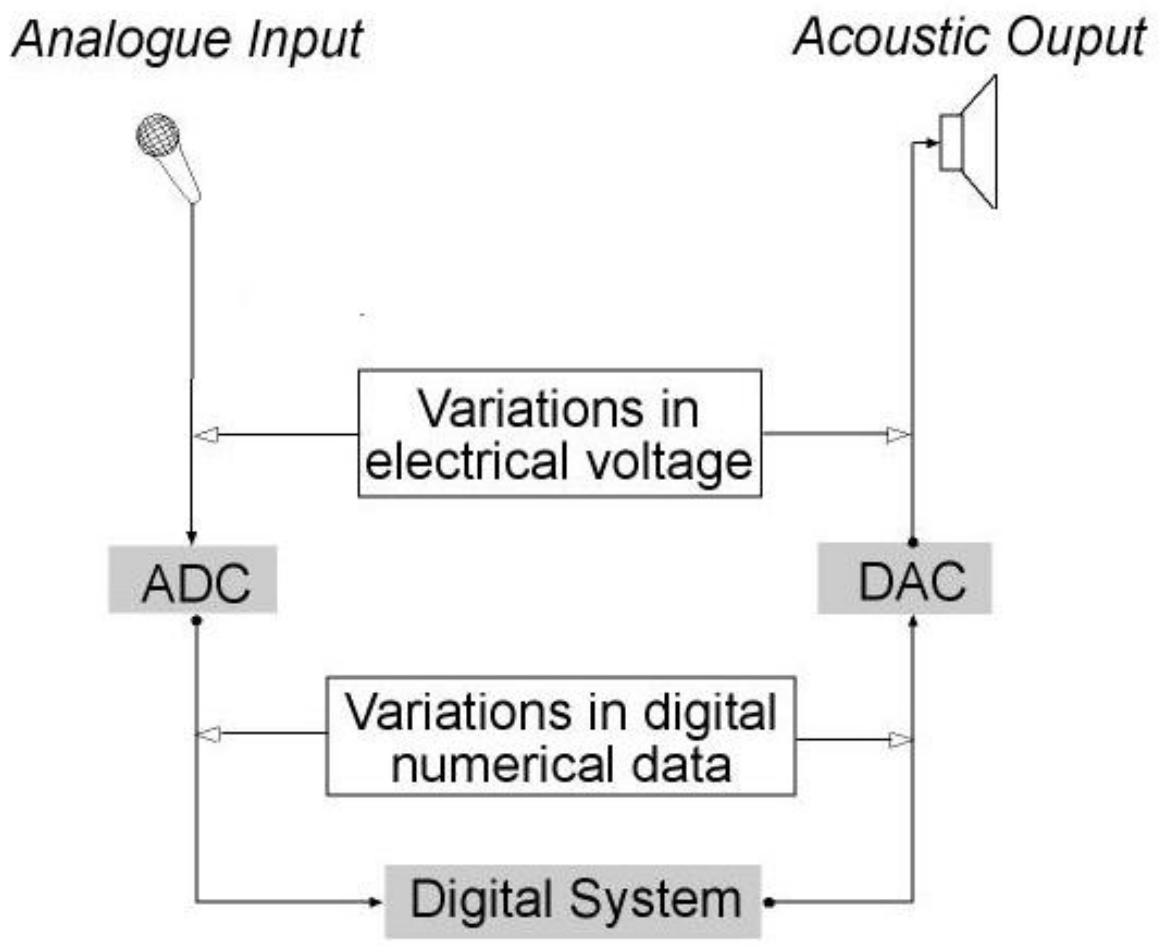
07/30/2013

Ozan Tuncer

otuncer@bu.edu

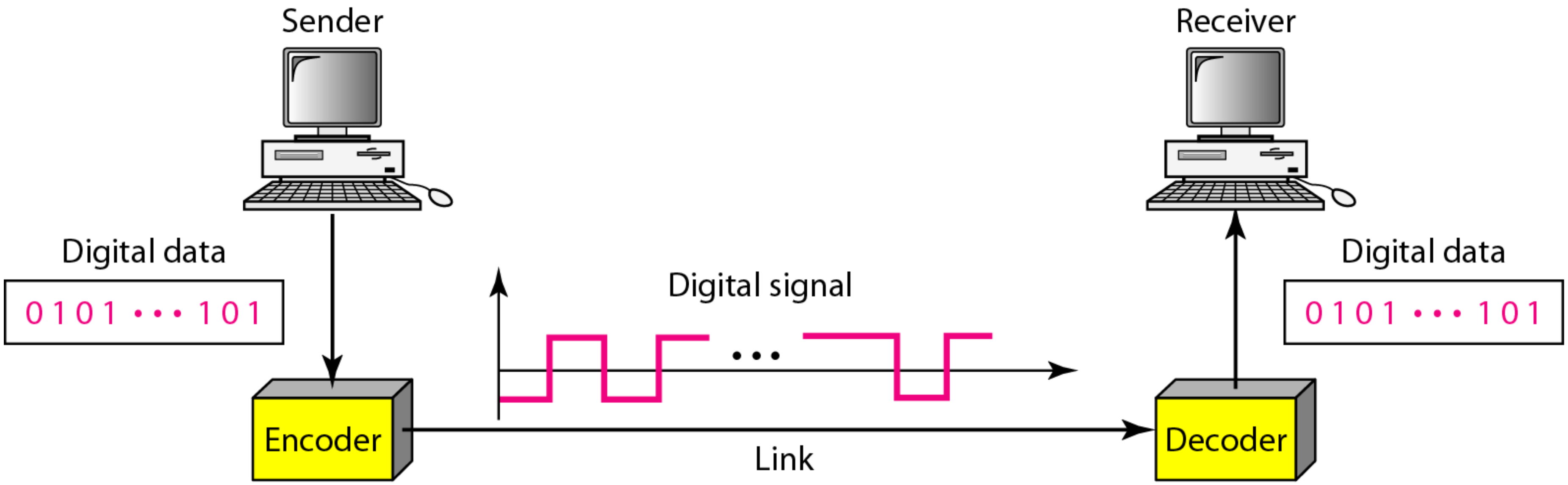
BOSTON UNIVERSITY

The Audio Signal Path

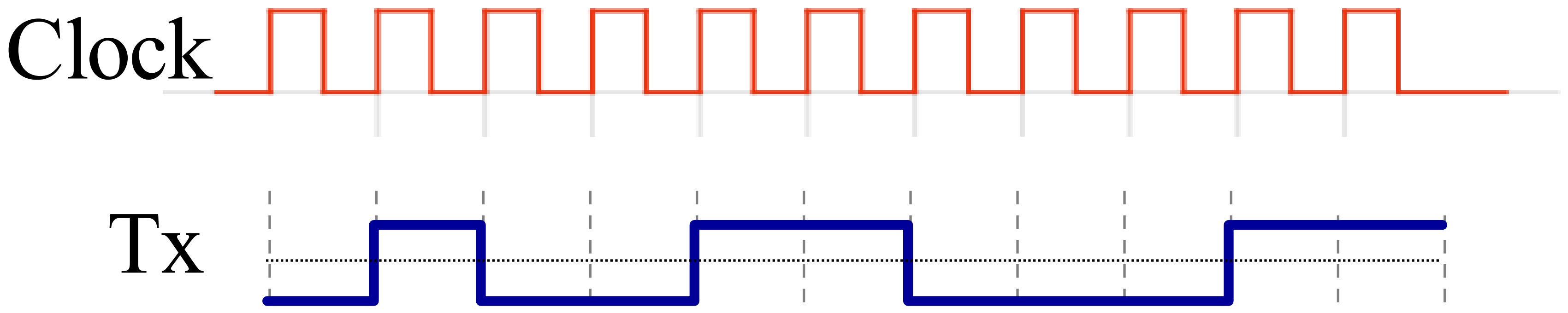

Analogue Input
Acoustic Ouput
Variations in electrical voltage
ADC
DAC
Variations in digital numerical data
Digital System

Line Encoding

- Converting a string of 1’s and 0’s (digital data) into a sequence of signals that denote the 1’s and 0’s.
- For example a high voltage level (+V) could represent a “1” and a low voltage level (0 or -V) could represent a “0”.

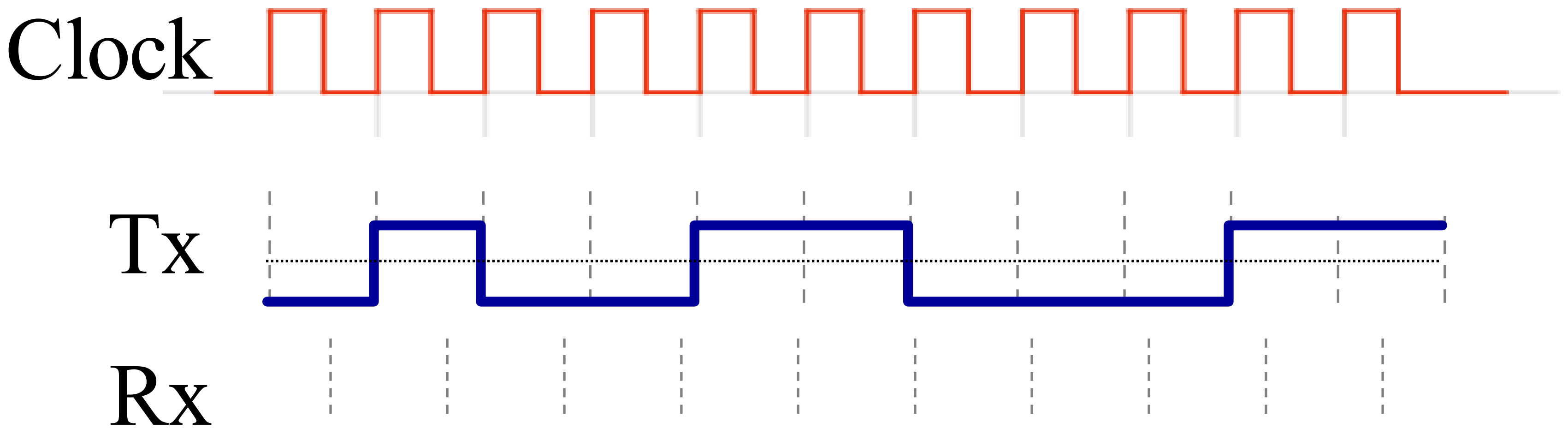


Nonreturn to Zero (NRZ) / ON-OFF Keying



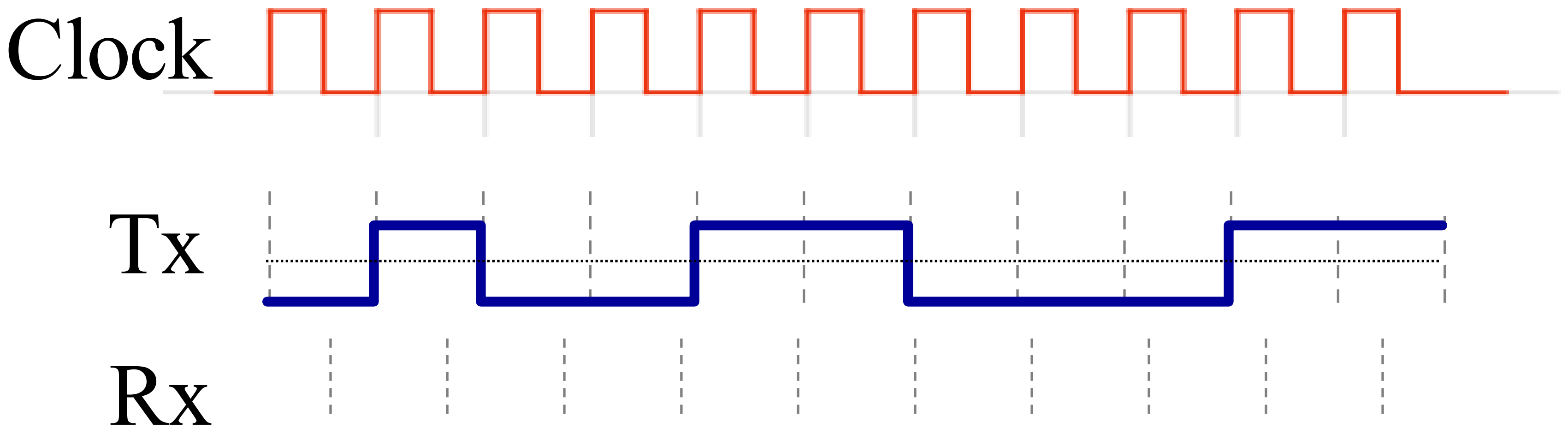
= power on (signal)
0 = power off (no signal)

Nonreturn to Zero (NRZ) / ON-OFF Keying



= power on (signal)
0 = power off (no signal)

Nonreturn to Zero (NRZ) / ON-OFF Keying

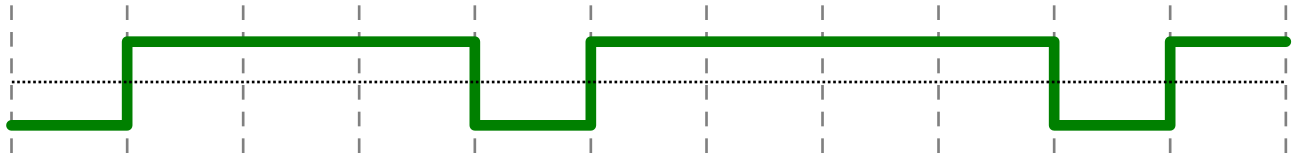


= power on (signal)
0 = power off (no signal)

Problem(s)

- lack of “clock” recovery during long string of or bits
- “baseline wander” during long string of or bits

Nonreturn to Zero Inverted (NRZI)



= change of signal level (on-off or off-on)

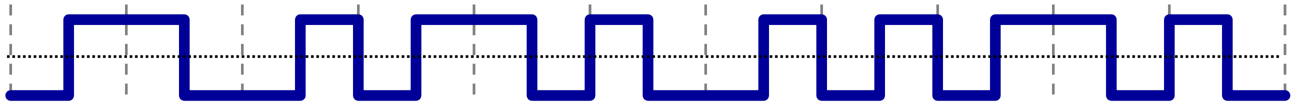
0 = no change of signal level

- NRZI is an example of **differential encoding**
- Fixes clocking problem for long string of bits

Problem(s)

- Lack of clock recovery during long string of bits

Manchester Encoding



Always transition in middle of bit period:
    0 = low-to-high transition
    1 = high-to-low transition

- Good clock recovery

- How to implement this?

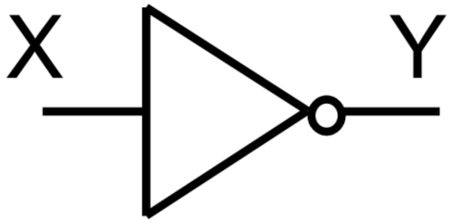
- **Boolean Functions (Logic Functions):** are function that return truth values; variables can only be 1 (true) or 0 (false).

NOT function: Y = ~X

- **Boolean Functions (Logic Functions):** are function that return truth values; variables can only be 1 (true) or 0 (false).

  NOT function: $Y = \sim X$

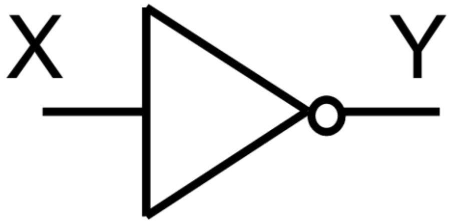
- **A Logic Gate:** is a physical device implementing a Boolean function.



- **Boolean Functions (Logic Functions):** are function that return truth values; variables can only be 1 (true) or 0 (false).

NOT function: Y = ~X

- **A Logic Gate:** is a physical device implementing a Boolean function.

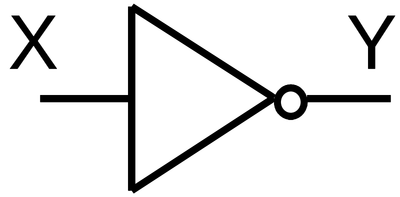


- **A Truth Table:** shows how a logic circuit's output responds to inputs.

X	Y
0	1
1	0

NOT Gate -- Inverter

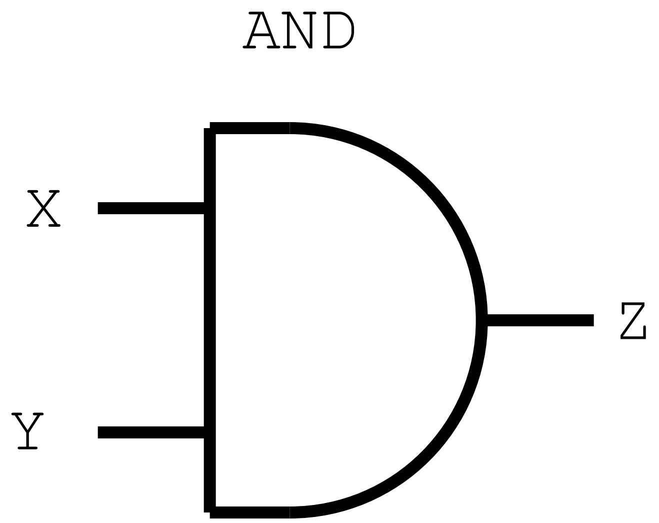
NOT



Y = ~X

X	Y
0	1
1	0

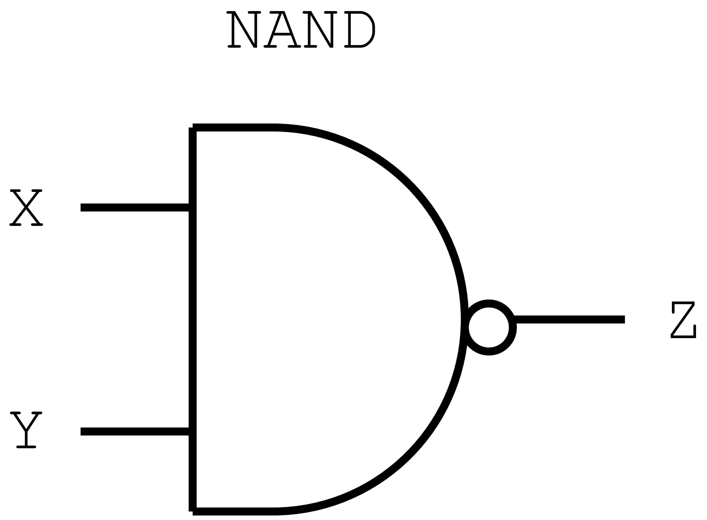
AND Gate



Z = X & Y

X	Y	Z
0	0	0
0	1	0
1	0	0
1	1	1

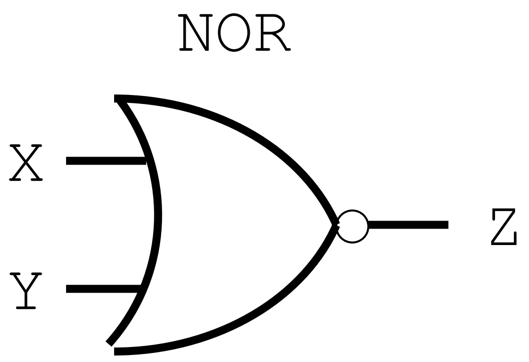
NAND Gate



X	Y	Z
0	0	1
0	1	1
1	0	1
1	1	0

```
Z = ~(X & Y)
nand(Z,X,Y)
```

NOR Gate

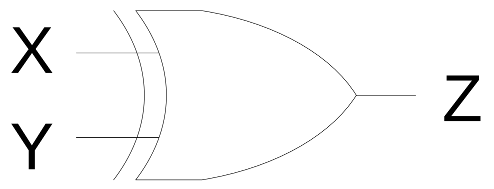


```
Z = ~(X | Y)
nor(Z,X,Y)
```

X	Y	Z
0	0	1
0	1	0
1	0	0
1	1	0

Exclusive-OR Gate

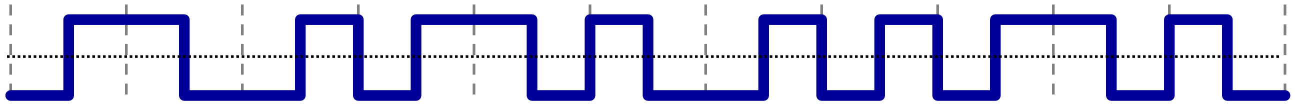
XOR



```
Z = X ^ Y
xor(Z,X,Y)
```

X	Y	Z
0	0	0
0	1	1
1	0	1
1	1	0

Manchester Encoding



Always transition in middle of bit period:

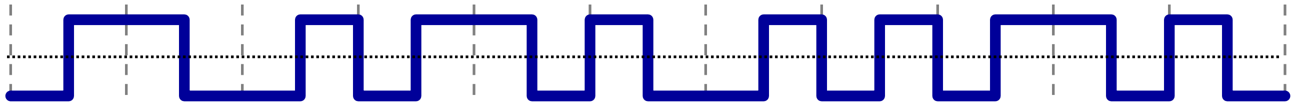
0 = low-to-high transition

1 = high-to-low transition

- Good clock recovery

- How to implement this?

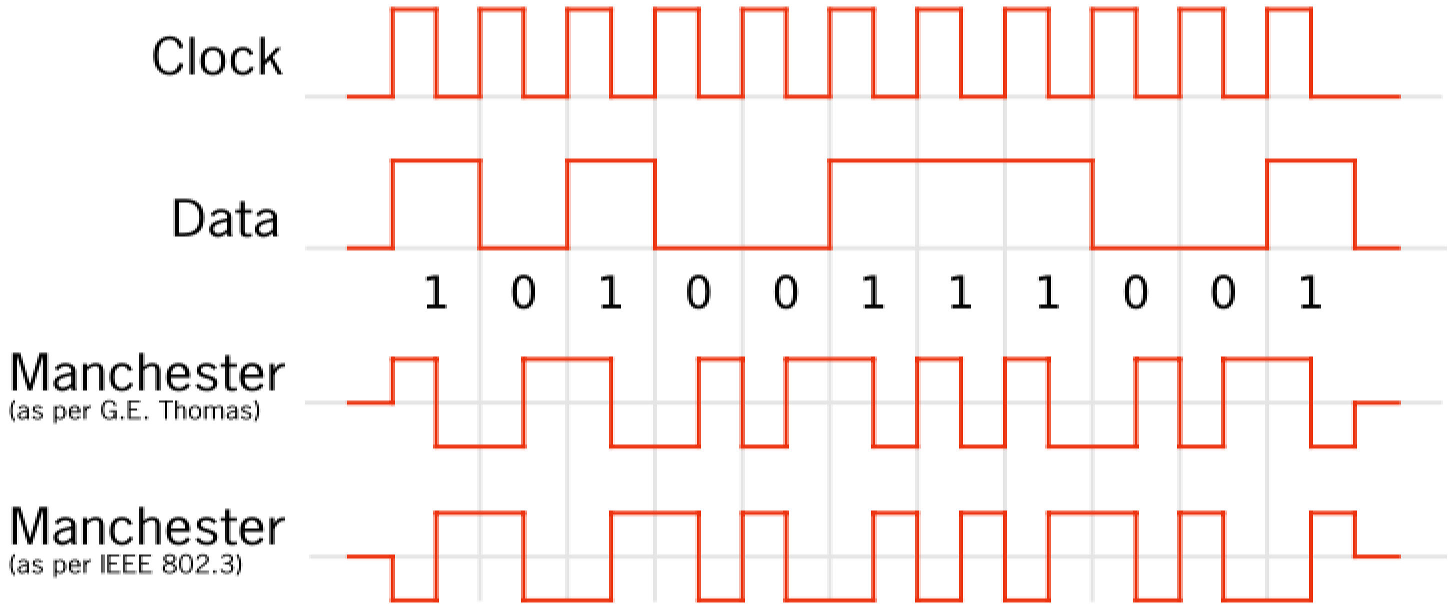
Manchester Encoding



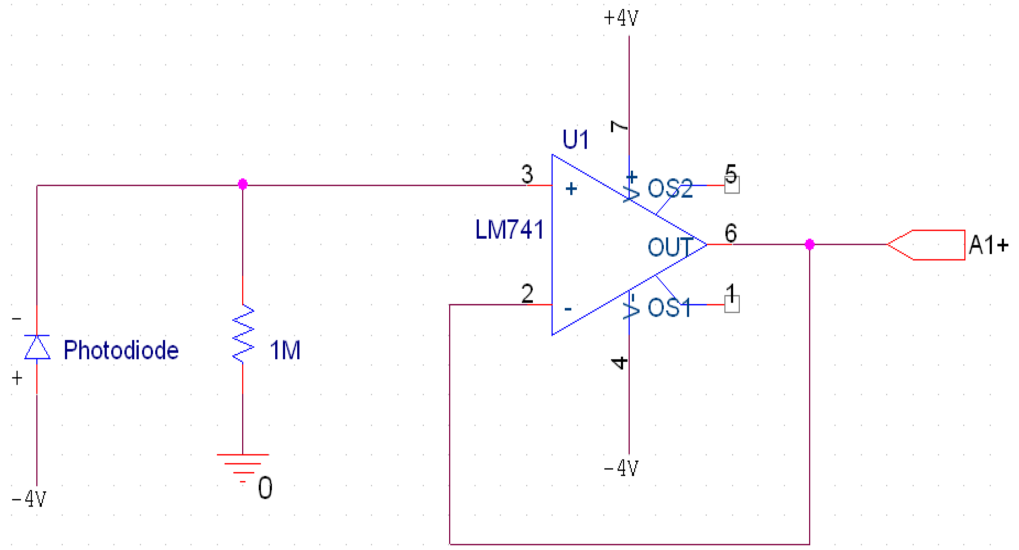
Always transition in middle of bit period:

0 = low-to-high transition

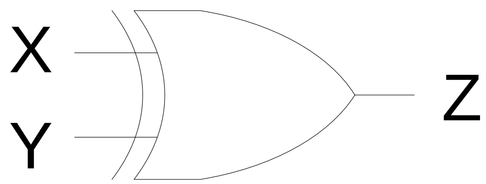
1 = high-to-low transition



Tx/Rx

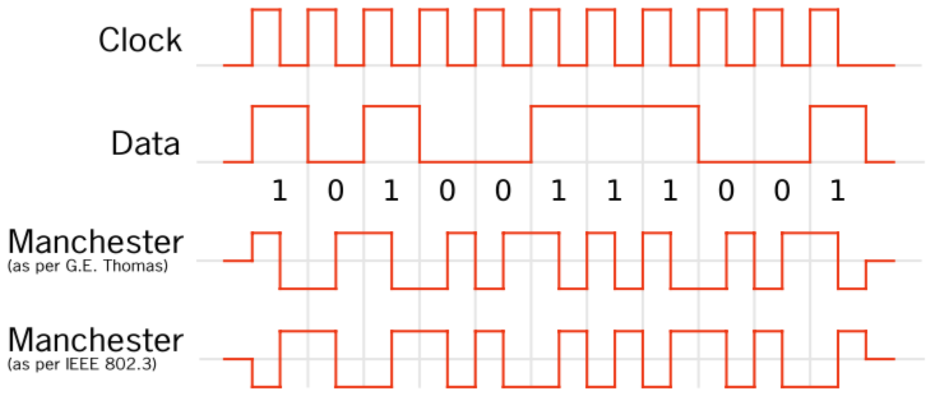


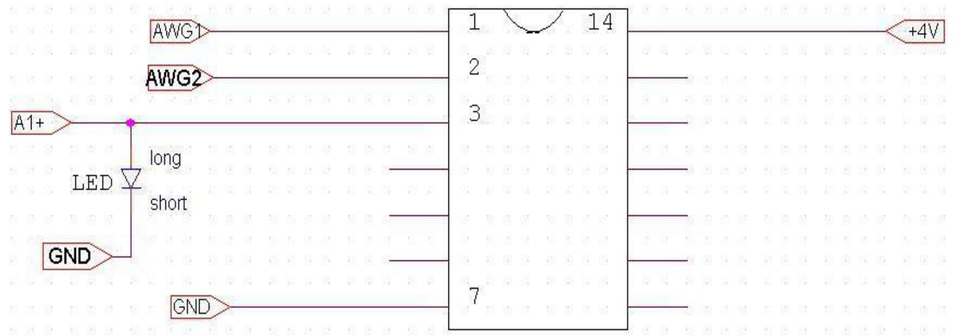
XOR



X	Y	Z
0	0	0
0	1	1
1	0	1
1	1	0

```
Z = X ^ Y
xor(Z,X,Y)
```


Clock
Data
Manchester
(as per G.E. Thomas)
Manchester
(as per IEEE 802.3)


AWG1
AWG2
A1+
LED
long
short
GND
GND
1
2
3
7
14
+4V