Practical 4: Implementation and Analysis of Line Coding Schemes

* What is Line Coding?
* When sending of digital data has to be carried out, a line coding which is a pattern of different voltage levels used to represent the digital data in digital signal form.
* There are predefined rules for presenting each possible combination/bit in different voltage level.
* TYPES OF LINE CODING TECHNIQUE
* What are the problems encountered while transmitting digital signals?
* Baseline Wandering
* DC Component
* Self-Synchronization

1. Baseline Wandering

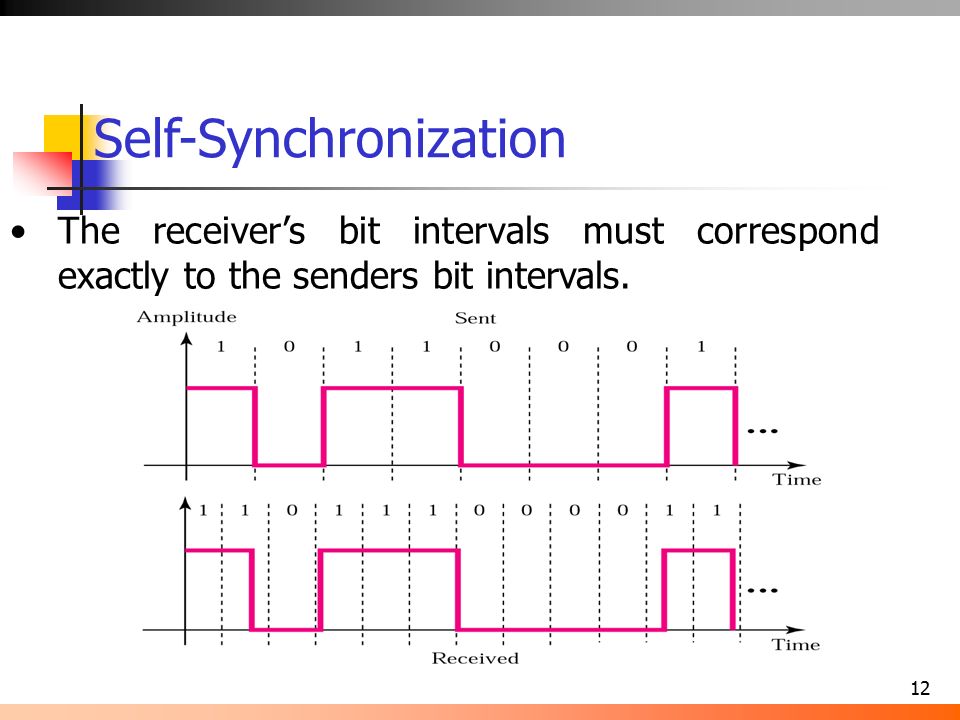
* The determination of signal to data is done at receiver side by measuring average power received by it.
* So if in our data there is long chains of 0s or 1s, it can create problem at receiver side to decode the transmitted data correctly.
* As if we have long number of 0s, the average power over distance would become nearly zero, so there will be problem of Baseline Wandering that receiver can’t decide the average power baseline for particular data. Same will be the case for large number of 1s.

1. DC Component

* DC Component is again the problem when there are large number of continuous 0s or 1s.
* Because of continuous voltage value, the signal will become partially DC rather than having up & down transitions in between.
* Due to which the frequency of said signal will decrease or say will be near to zero. So when there is channel which only accepts data at certain threshold frequency, our signal will be eliminated by such channels.
* Also some of electrical devices not let DC Voltage pass through them like capacitors. It eliminates the DC component passing through it & the data will be lost cause of that.

1. Self-synchronization

* When the timing clock of receiver and sender do not match even by few microseconds, this problem takes place.
* While transmitting continuous 0s or 1s, how receiver will come to know that after what time the current element’s interpretation should end and the next element should be decoded.

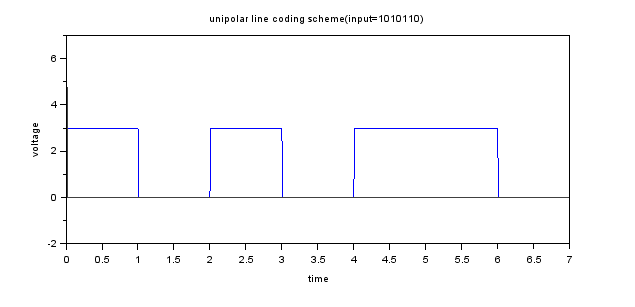


* UNIPOLAR

|  |  |
| --- | --- |
| Data element | Voltage level |
| 1 | Either +ve or –ve |
| 0 | 0 |

* In this line coding all the 3 problems will occur where there are continuous 0s or 1s.

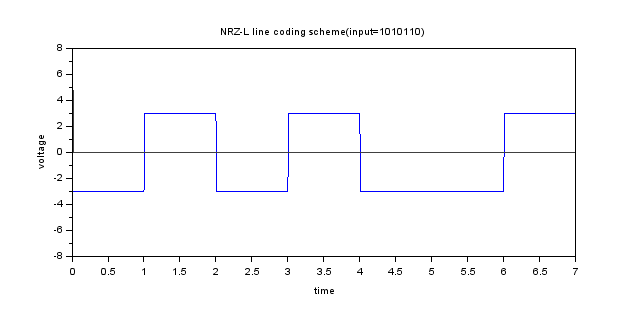
(NOTE: In all the below mentioned graphs the data bits presented are: 1010110)



* NRZ-L

|  |  |
| --- | --- |
| Data element | Voltage level |
| 1 | Negative voltage |
| 0 | Positive voltage |

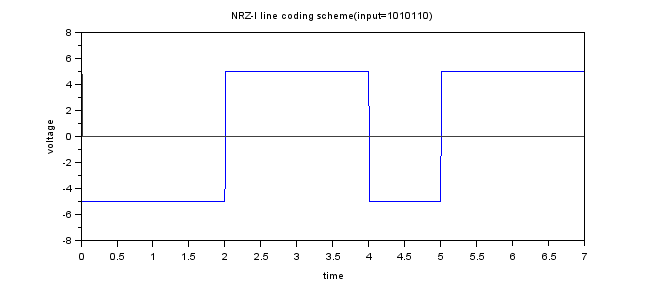
* Here also all the 3 problems will occur if there are continuous 0s or 1s.



* NRZ-I

|  |  |
| --- | --- |
| Data element | Voltage level |
| 1 | Inversion of last voltage level |
| 0 | No inversion |

* In this method, if there are continuous 1s there is no problem as for each next 1, our signal is getting opposite sign from last voltage level. But for 0s the problem still exists.



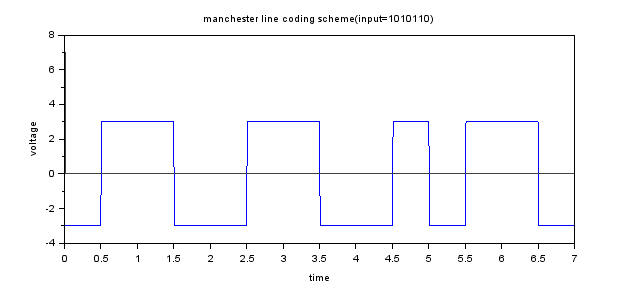
* Polar-RZ (Return Zero)

|  |  |
| --- | --- |
| Data element | Voltage level |
| 1 | Positive voltage + zero voltage |
| 0 | Negative voltage + zero voltage |

* In this method, the above stated problems are solved as there is no continuous voltage level for long period of time.
* But focus on signal elements used to create/encode a single data element. Two signal elements are used to send out one bit of data. So the bandwidth required for data transmission would be double as of above stated methods.
* Manchester

|  |  |
| --- | --- |
| Data element | Voltage level |
| 1 | Negative voltage + Positive voltage |
| 0 | Positive voltage + Negative voltage |

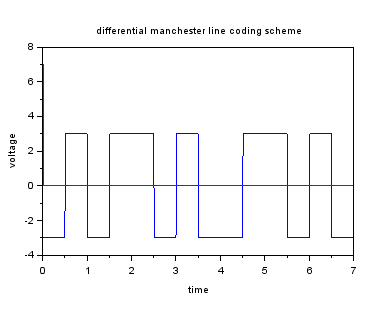
* Again the three problems are solved here but the required signal bits to send one data bit is two, which takes time at receiver side to decode & process further.
* Same way it is applied to Differential Manchester.



* Differential Manchester

|  |  |
| --- | --- |
| Data element | Voltage level |
| 1 | 1st half>Negative voltage + 2nd half>Positive voltage |
| 0 | Inversion to last representation |

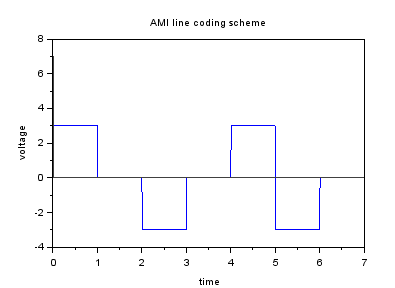
* Here, for one data bit to be send over channel, two signal elements are required which means the bandwidth required for sending data compared to other methods will be twice.



* AMI (Alternate Mark Inversion)

|  |  |
| --- | --- |
| Data element | Voltage level |
| 1 | Inversion to what last 1 was represented by |
| 0 | Positive voltage + Negative voltage |

* In AMI method, while transmitting out data if 1 is encountered; we’ll change the magnitude of signal element opposite to the magnitude with last 1 was represented.
* In this method, bandwidth requirement is not doubled so bandwidth is out of concern. But for long chains of 0s, the above mentioned problems still exits.



Conclusion

To conclude we can say that the decision to choose between different line coding techniques depends upon the user’s wants.

If user do have more bandwidth capacity than choosing Manchester or Differential Manchester may help but when bandwidth is limited and fast processing of data is required than we can go with RZ coding scheme.