



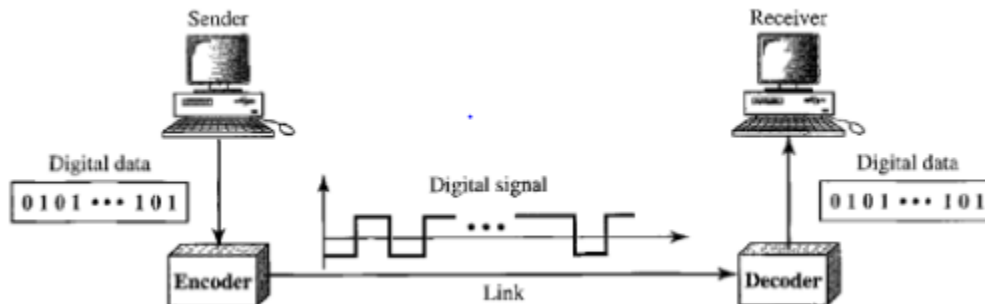
EAST WEST UNIVERSITY

Department of CSE Lab Report

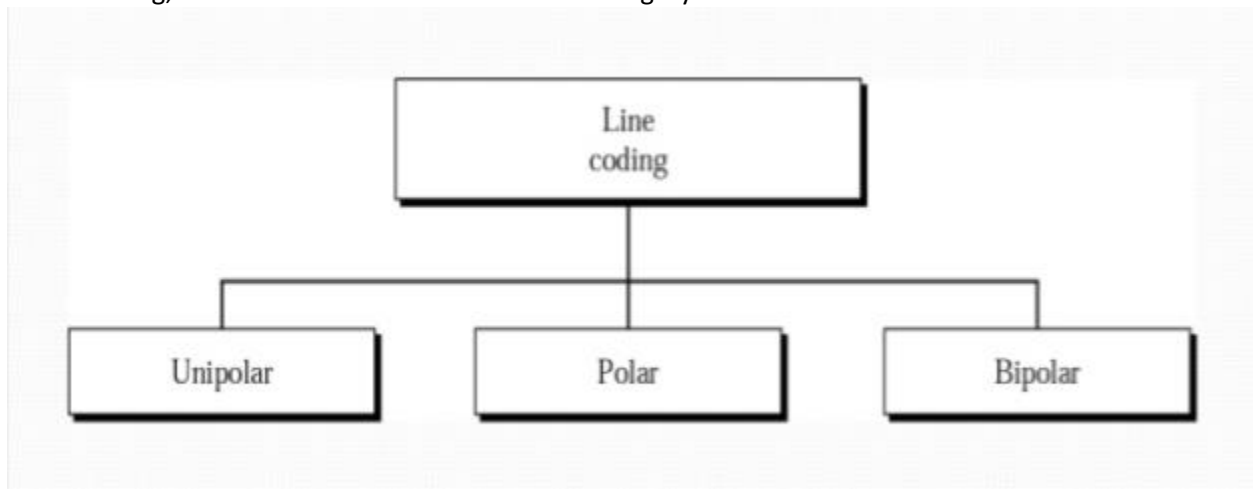
Course Code and Name: CSE350 – Data communication	
Lab:	
Name of Student: Intesar Islam Khan ID: 2019-1-60-043	Course Instructor information: Md. Mahir Ashhab Lecturer Department of Computer Science & Engineering
Section: 02	Date of Submission: 14/11/2022

Introduction:

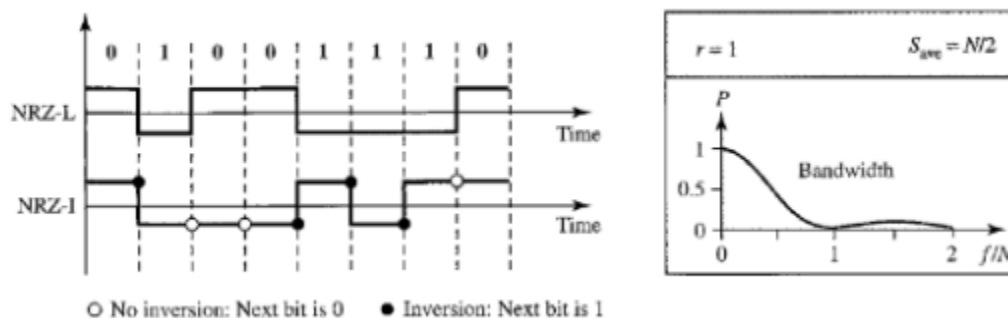
Line coding is the process of converting digital data to signal data. This data can be either digital or analog. The conversion involves three techniques: line coding, block coding, and scrambling.



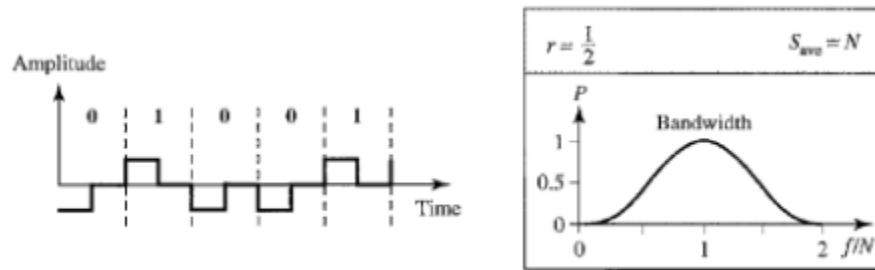
In line coding, there are several schemes in each category.



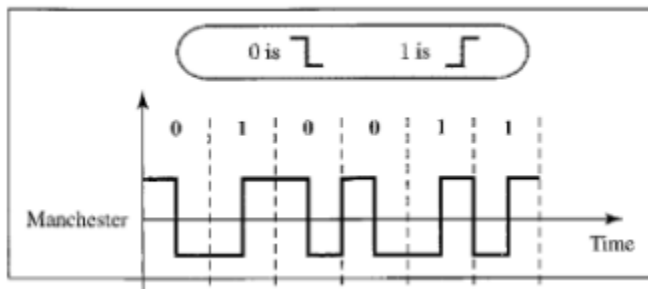
Non-Return to Zero(NRZ): In polar NRZ encodings, there are two levels of voltage amplitude and two versions of polar NRZ: NRZ-L and NRZ-I. For NRZ-L On the sender side if the binary value is zero then the signal will be + and if the binary value is one then the signal will be minus. For NRZ-I if the binary value is 0, a signal will be the same as previous signal, and if one it will invert.



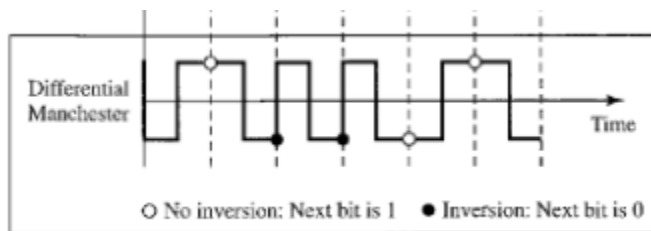
Return to Zero (RZ): In sender side if binary value is zero we have to print minus zero in signal and when binary value is one that time print +0.



Manchester: In the sender side if the binary value is 0 print '+-' and if the value is 1 print '-+'.



Differential Manchester: In the sender side if binary value is 1, it will be continuing and 0 then print invert.



Lab Task:

Creating a sender receiver environment to show how OSI model works and print every 125-character segment from receiver side

Implementation:

I created 3 separated c++ files 1. main.cpp, 2. sender.cpp, 3. seceiver.cpp and also 4 text files 1. input.txt, 2. temp.txt, 3. output.txt, 4. **signal.txt**

sender.cpp: to create the OSI environment I create 7 more function which works as layers.

sender (function): reads text file or takes user input and send the string to application layer.

```

240 void sender()
241 {
242     string txt;
243     printf("\nSender : ");
244     getline(cin,txt);
245     /*
246     ifstream input("input.txt");
247     while(getline(input,txt))
248     {
249         //cout << str << '\n';
250     }
251     input.close()
252     */
253     int txtlen = txt.size();
254     //cout << txtlen << '\n';
255     int i=0;
256     while(i!=(txtlen/125)+1)
257     {
258         string str = txt.substr(125*i,125);
259         //cout << str << '\n';
260         applicationLayer(str);
261         i++;
262     }
263     //nrzl();
264     //nrzl();
265     //man();
266     //man();
267     diffman();
268 }

```

NRZ-L:

```

119 void nrzl()
120 {
121     int i=0;
122     string txt;
123     ifstream temp("temp.txt");
124     while(getline(temp,txt))
125     {
126         //cout << str << '\n';
127     }
128     temp.close();
129     //cout << txt;
130     ofstream signal;
131     signal.open("signal.txt");
132     while(i!=txt.size())
133     {
134         if(txt[i]=='0')
135         {
136             signal << '+';
137         }
138         else
139             signal << '-';
140         i++;
141     }
142     signal.close();

```

NRZ-I:

```

145 void nrzI()
146 {
147     int i=0;
148     string txt;
149     ifstream temp("temp.txt");
150     while(getline(temp,txt))
151     {
152         //cout << str << '\n';
153     }
154     temp.close();
155     //cout << txt;
156     ofstream signal;
157     signal.open("signal.txt");
158     char state = '+', antiState = '-', t;
159     while(i!=txt.size())
160     {
161         if(txt[i]=='0')
162         {
163             signal << state;
164         }
165         else
166         {
167             t = state;
168             state = antiState;
169             antiState = t;
170             signal << state;
171         }
172         i++;
173     }
174     signal.close();
175 }
176

```

RZ:

```

87 void rz()
88 {
89     int i=0;
90     string txt, str="";
91     ifstream temp("temp.txt");
92     while(getline(temp,txt))
93     {
94         //cout << str << '\n';
95     }
96     temp.close();
97     //cout << txt;
98
99     while(i!=txt.size())
100     {
101         if(txt[i]=='0')
102         {
103             str = str + "-"+ "0";
104         }
105         else
106         {
107             str = str + "+"+ "0";
108         }
109         i++;
110     }
111     //cout << "string: " << str;
112     ofstream signal;
113     signal.open("signal.txt");
114     signal << str;
115     signal.close();
116 }
117

```

Manchester:

```
9 void man()
10 {
11     int i=0;
12     string txt, str="";
13     ifstream temp("temp.txt");
14     while(getline(temp, txt))
15     {
16         //cout << str << '\n';
17     }
18     temp.close();
19     //cout << txt;
20
21     while(i!=txt.size())
22     {
23         if(txt[i]=='0')
24         {
25             str = str + "+-";
26         }
27         else
28         {
29             str = str + "-+";
30         }
31
32         i++;
33     }
34     //cout << "string: " << str;
35     ofstream signal;
36     signal.open("signal.txt");
37     signal << str;
38     signal.close();
39 }
```

Differential Manchester:

```
41 void diffman()
42 {
43     int i=0;
44     string txt, str="", state = "+-", antiState="-+", t;
45     ifstream temp("temp.txt");
46     while(getline(temp, txt))
47     {
48         //cout << str << '\n';
49     }
50     temp.close();
51     //cout << txt;
52     if(txt[i]=='0')
53     {
54         str = str+antiState;
55         t=antiState;
56         antiState=state;
57         state=t;
58     }
59     else
60     {
61         str=str+state;
62     }
63     i++;
64 }
```

```

64     while(i!=txt.size())
65     {
66         if(txt[i]=='1')
67         {
68             str = str+antiState;
69             t=antiState;
70             antiState=state;
71             state=t;
72         }
73         else
74         {
75             str=str+state;
76         }
77
78         i++;
79     }
80     cout <<"string: "<<str;
81     ofstream signal;
82     signal.open("signal.txt");
83     signal<<str;
84     signal.close();
85 }

```

receiver.cpp: to create the OSI environment I create 7 more function which works as layers. **receiver (function):** reads text file (temp.txt) and convert the binary string to char string and send the string to physical layer.

```

276 void receiver()
277 {
278     string signal;
279     ifstream temp("signal.txt");
280     while(getline(temp,signal))
281     {
282         //cout << str << '\n';
283     }
284     temp.close();
285     //cout << signal;
286
287     // signal Types
288     //RnrzL(signal);
289     //RnrzI(signal);
290     //Rrz(signal);
291     //Rmrz(signal);
292     Rdiffman(signal);
293
294     //cout << txtlen << '\n';
295     /*int i=0;
296     //cout << str << '\n';
297     while(i!=(txtlen/1200)+1)
298     {
299         string str = txt.substr(1200*i,1200);
300         //cout << str << '\n';
301         setStringtoASCII(str);
302         i++;
303     }*/
304     //cout << "XX" << '\n';
305 }

```

NRZ-L:

```

106 void RnrzL(string signal)
107 {
108     string txt="";
109     int i=0;
110     int siglen = signal.size();
111     //cout << siglen;
112     while(i!= siglen)
113     {
114         if(signal[i]=='-')
115         {
116             txt = txt+'1';
117         }
118         else
119             txt = txt+'0';
120         i++;
121     }
122     //cout << txt;
123     i=0;
124     int txtlen = txt.size();
125     //cout << str << '\n';
126     while(i!=(txtlen/1200)+1)
127     {
128         string str = txt.substr(1200*i,1200);
129         //cout << str << '\n';
130         setStringtoASCII(str);
131         i++;
132     }
133 }

```

NRZ-I:

```

135 void RnrzI(string signal)
136 {
137     char state='0',antiState = '1',t;
138     string txt="";
139     int i=0;
140     int siglen = signal.size();
141     //cout << siglen;
142     if(signal[i]=='+')
143         txt=txt+'0';
144     else
145         txt = txt + '1';
146     i++;
147     while(i!= siglen)
148     {
149         if((signal[i]=='-' && signal[i-1] == '-') || (signal[i] == '+' && signal[i-1] == '+'))
150             txt = txt +'0';
151         else
152             txt = txt + '1';
153         i++;
154     }
155     cout << txt;
156     cout << txt.size();
157     i=0;
158     int txtlen = txt.size();
159     //cout << str << '\n';
160     while(i!=(txtlen/1200)+1)
161     {
162         string str = txt.substr(1200*i,1200);
163         //cout << str << '\n';
164         setStringtoASCII(str);
165         i++;
166     }
167 }

```


RZ:

```
242 void Rrz(string signal)
243 {
244     string txt="";
245     int i=0;
246     int siglen = signal.size();
247     //cout << siglen;
248     while(i!= siglen/2)
249     {
250         if(signal[i*2]=='+')
251         {
252             txt = txt+'1';
253         }
254         else
255         {
256             txt= txt+'0';
257         }
258         i++;
259     }
260     cout << txt;
261
262     i=0;
263     int txtlen = txt.size();
264     //cout << txt << '\n';
265     while(i!=(txtlen/1200)+1)
266     {
267         string str = txt.substr(1200*i,1200);
268         //cout << str << '\n';
269         setStringtoASCII(str);
270         i++;
271     }
272 }
273 }
```

Manchester:

```
209 void Rmar(string signal)
210 {
211     string txt="";
212     int i=0;
213     int siglen = signal.size();
214     //cout << siglen;
215     while(i!= siglen/2)
216     {
217         if(signal[i*2]=='+')
218         {
219             txt = txt+'0';
220         }
221         else
222         {
223             txt= txt+'1';
224         }
225         i++;
226     }
227     //cout << txt;
228
229     i=0;
230     int txtlen = txt.size();
231     //cout << txt << '\n';
232     while(i!=(txtlen/1200)+1)
233     {
234         string str = txt.substr(1200*i,1200);
235         //cout << str << '\n';
236         setStringtoASCII(str);
237         i++;
238     }
239 }
240 }
```

Differential Manchester:

```

170 void Rdiffman(string signal)
171 {
172     string txt="";
173     int i=0;
174     int siglen = signal.size();
175     //cout << siglen;
176     if(signal[i]=='-')
177     {
178         txt = txt+'0';
179     }
180     else
181         txt= txt+'1';
182     i++;
183     while(i!= siglen/2)
184     {
185         if((signal[(i-1)*2]=='-' && signal[i*2]=='+') || (signal[(i-1)*2]=='+' && signal[i*2]=='-'))
186         {
187             txt = txt+'1';
188         }
189         else
190         {
191             txt= txt+'0';
192         }
193         i++;
194     }
195     //cout << txt;
196     i=0;
197     int txtlen = txt.size();
198     //cout << str << '\n';
199     while(i!=(txtlen/1200)+1)
200     {
201         string str = txt.substr(1200*i,1200);
202         //cout << str << '\n';
203         setStringtoASCII(str);
204         i++;
205     }
206 }

```

Outputs: Sample output for a single function.(RZ)

```
Sender : this is a text
Receiver: this is a text

Process returned 0 (0x0)   execution time : 5.108 s
Press any key to continue.
```

temp.txt:

temp.txt - Notepad

File Edit Format View Help

010100000100100000110101010000100010000101101010010000111000101101010010000101010000101101010000101001001011010100100001010000001011010101000011010000110100001101

output.txt:

[illegible]

Discussion:

While I was creating this application, I did not face any problem, mainly because I remember the pseudocode for every function and applied as it is.