Communication Course Project Report
2rd Year Computer Engineering

Manchester

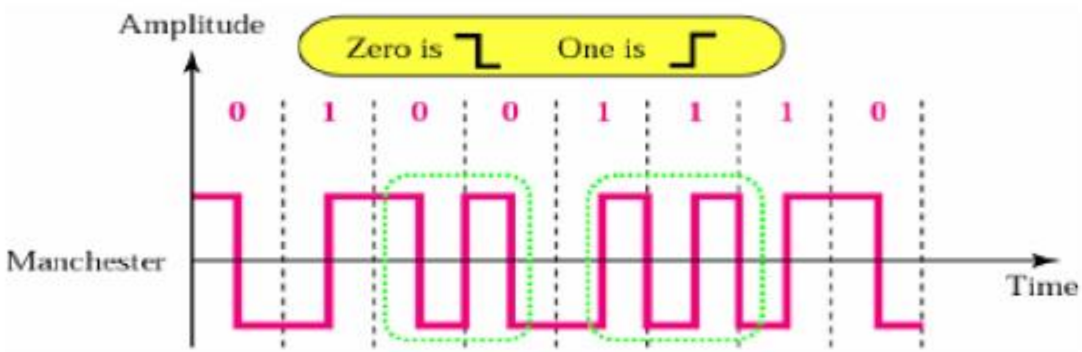
Team Members

SN	Student Name	Section Number
1	تسنيم عبد الرحمن حامد عبد الوهاب	1
2	ايه محمد القدري عامر حسين	1
3	امال محمد السعيد ابراهيم	1
4	منة الله أشرف حنفي	1
5	ياسمين احمد جوده	1

Table of Contents

1. Project Objective

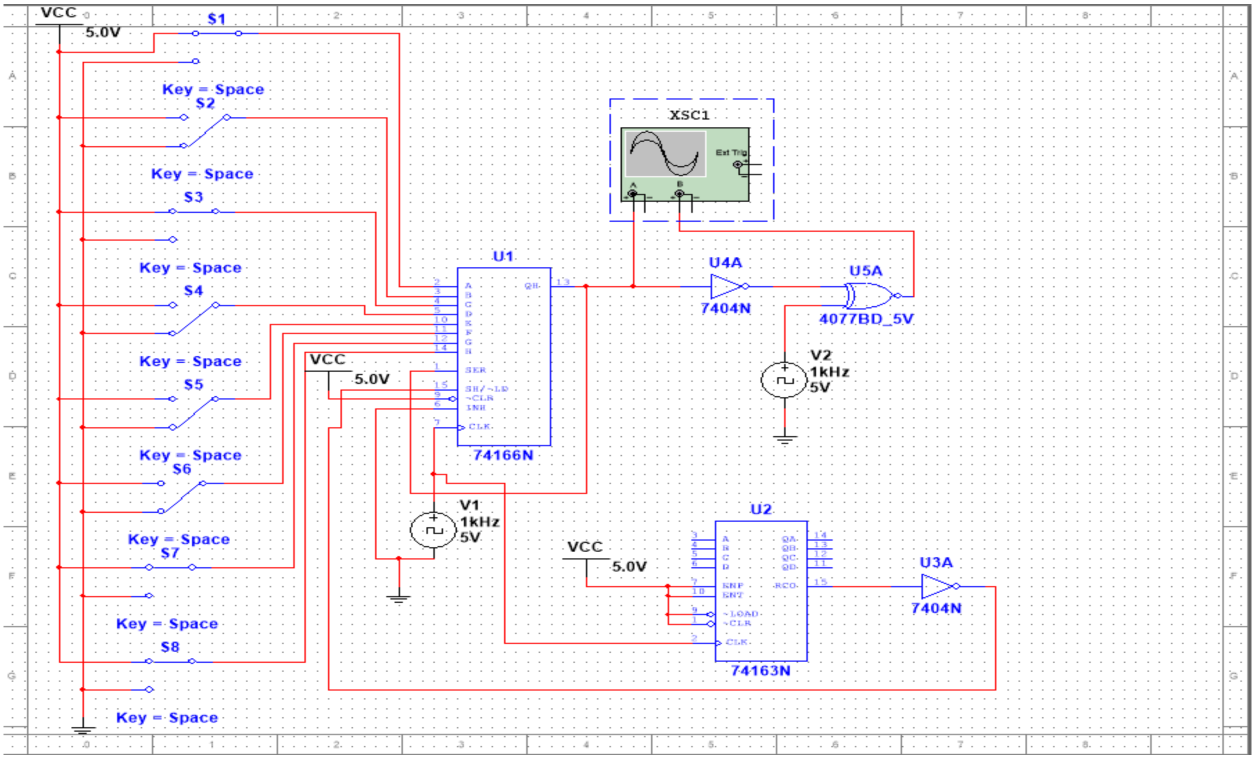
. We take the data (binary) and create NRZ code polar signal then convert it to Manchester code. By 2 methods multisim and MATLAB.



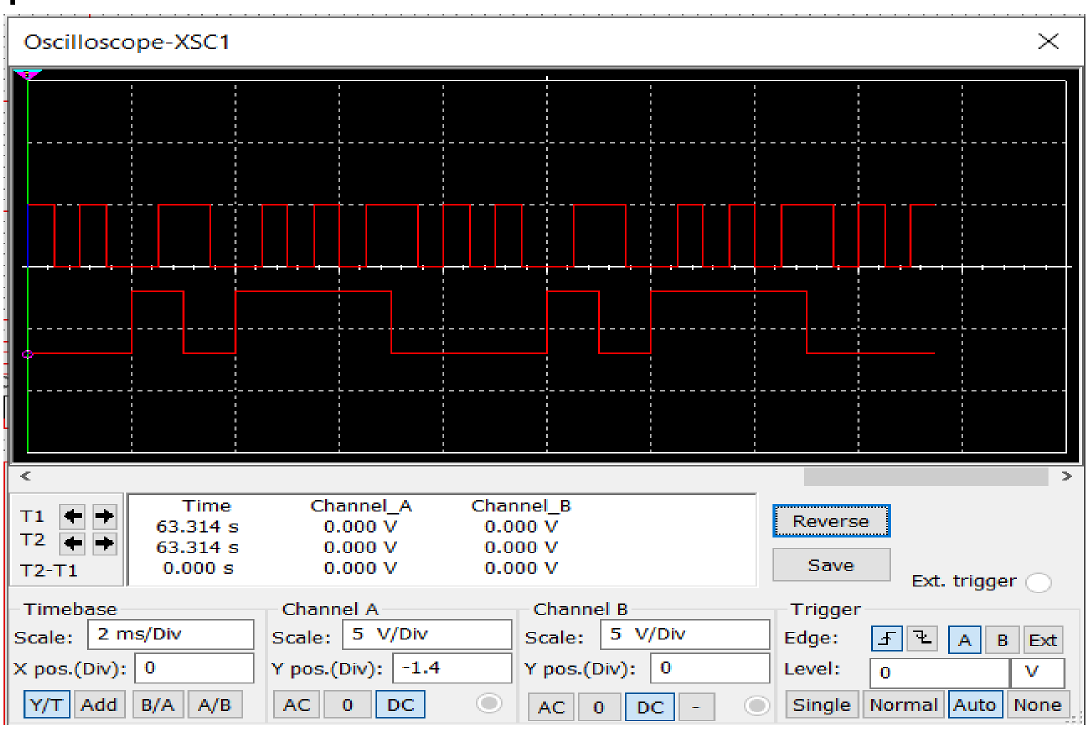
Manchester coding is one of the most common data coding methods used today. Similar to BiPhase, Manchester coding
provides a means of adding the data rate clock to the message to be used on the receiving end. Also Manchester provides
the added benefit of always yielding an average DC level of 50%. This has positive implications in the demodulator's circuit
design as well as managing transmitted RF spectrum after modulation. This means that in modulation types where the
power output is a function of the message such as AM, the average power is constant and independent of the data stream
being encoded. Manchester coding states that there will always be a transition of the message signal at the mid-point of the
data bit frame. What occurs at the bit edges depends on the state of the previous bit frame and does not always produce a
transition. A logical “1” is defined as a mid-point transition from low to high and a “0” is a mid-point transition from high to low
(see Figure 2-1). A more thorough look at methods to encode and decode data will be shown in detail in the next sections.

2. Multisim

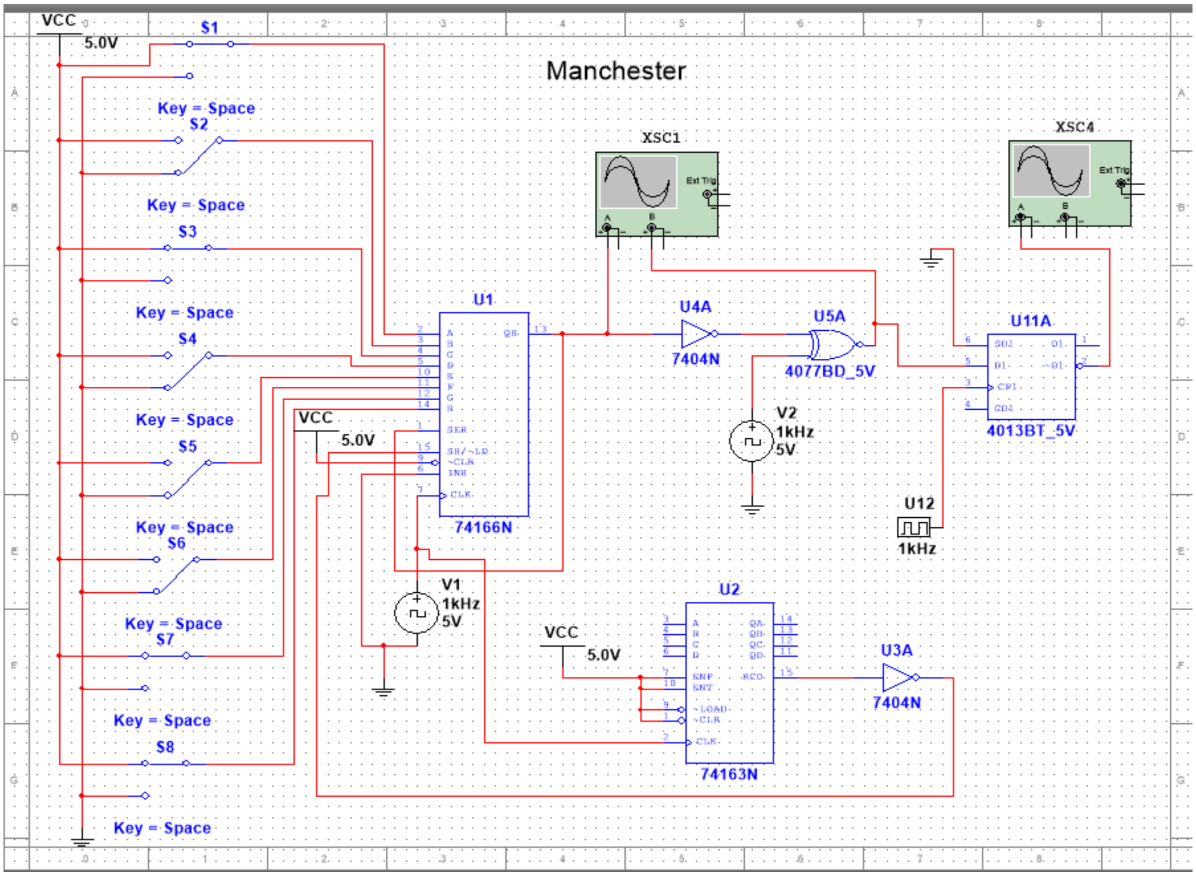
a. Encoding



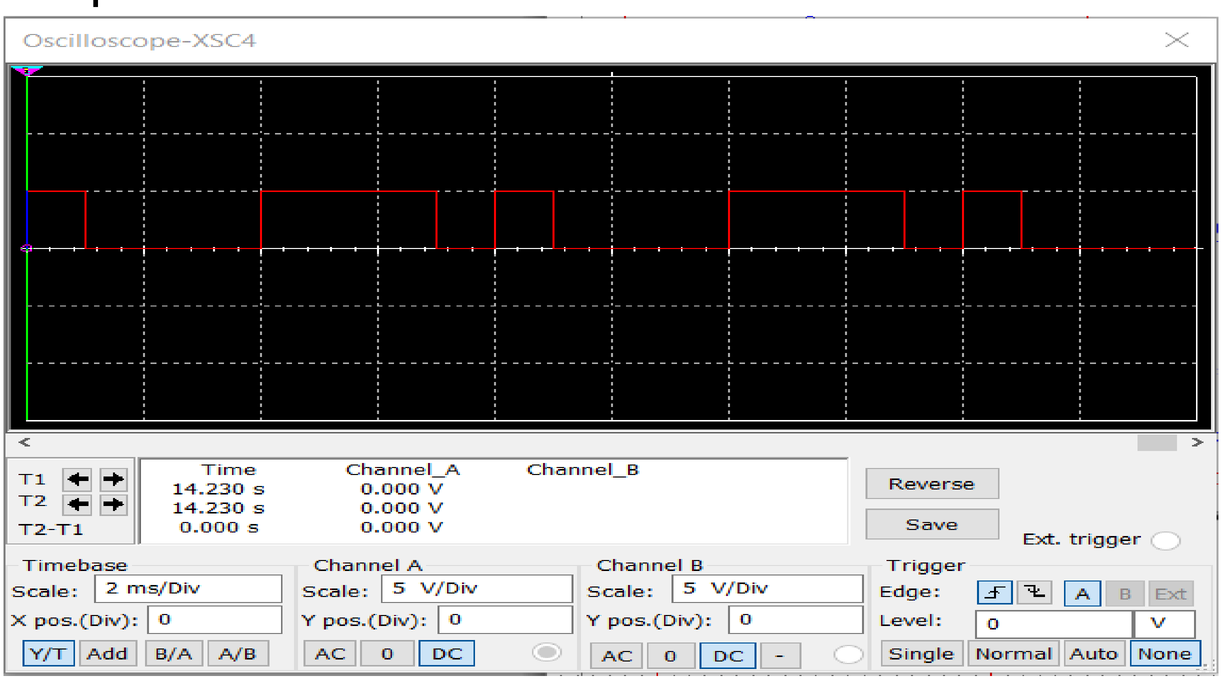
Output:



b. Decoding



Output:



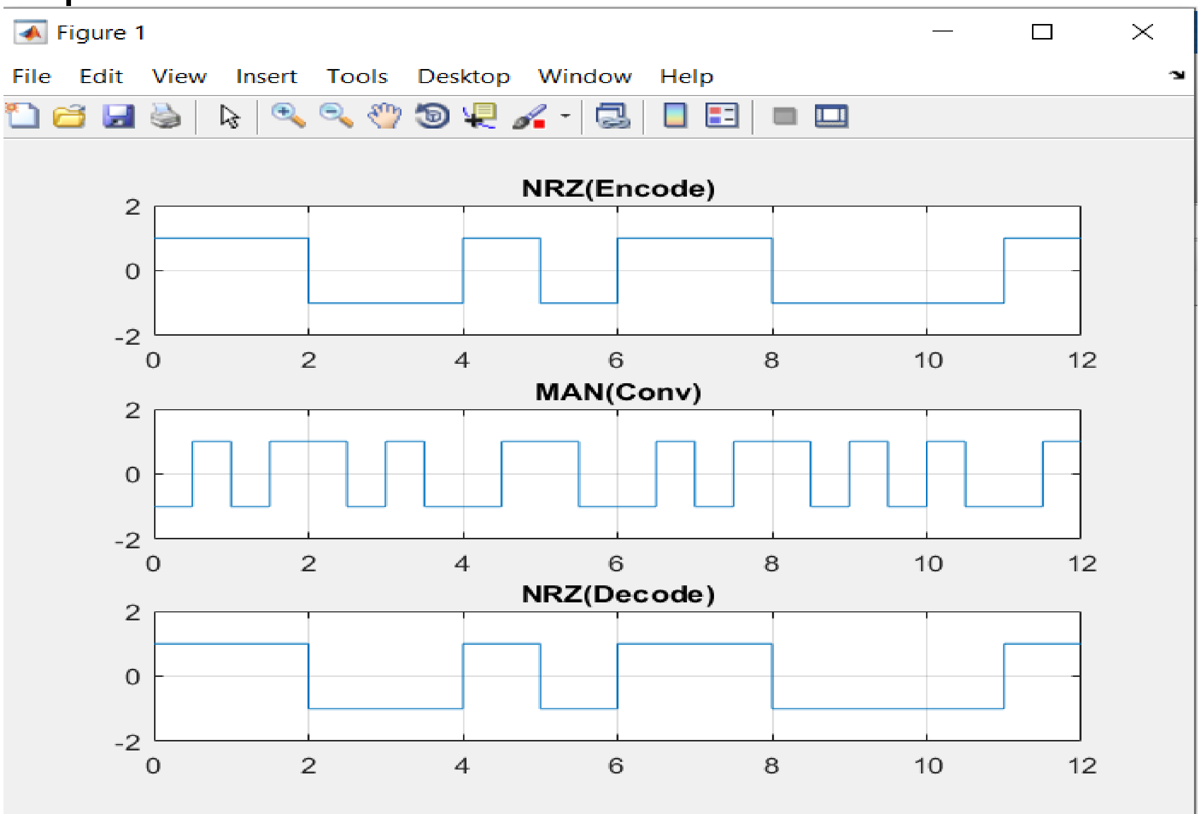
3. MATLAB

a. Code

```
m = input('enter stream of bits');
n = length(m);
x = [];
y = [];

u = [];
v = [];

r = [];
t = [];

res = [];
fprintf('Decoded stream :[');
for i=1:n
    %NRZ
    x=[x i-1 i];
    if(m(i)==1)
        y=[y 1 1 ];
    else
        y=[y -1 -1];
    end
    %MAN
      u=[u i-1 i-1+0.5 i-1+0.5 i];
    if(y(i*2)==-1)
        v=[v 1 1 -1 -1];
    else
        v=[v -1 -1 1 1];
    end
  %decoding
     r=[r i-1  i];
    if(v(i*4)==-1)

            t=[t -1 -1];
            fprintf(' %d',0);
    else
         t=[t 1 1];
          fprintf(' %d',1);

    end
```

```
end
fprintf(' ] \n');
subplot(3, 1, 1);
plot(x, y);
axis([0, n, -2, 2]);
grid on;
title('NRZ(Encode)');

% Plot second graph (a, b)
subplot(3, 1, 2);
plot(u, v);
axis([0, n, -2, 2]);
grid on;
title('MAN(Conv)');

subplot(3, 1, 3);
plot(r,t);
axis([0, n, -2, 2]);
grid on;
title('NRZ(Decode)');
```

Output:




```
Command Window
>> manchester
enter stream of bits [1 1 0 0 1 0 1 1 0 0 0 1]
Decoded stream :[ 1 1 0 0 1 0 1 1 0 0 0 1 ]
fx >>
```

4. References

1] William Stallings, “Data and Computer Communications”,
7th edition, Prentice Hall, 2004, pp. 864, ISBN 0-13-1006819.
[2]http://www.microcontrolador.com.br/datasheets/CD4014.pd
f , accessed on June 4, 2010.
[3]http://pdf1.alldatasheet.com/datasheet pdf/view/17691/PHILIPS/
HEF4020B.html, accessed on June 4,
2010.
[4] .Ibrahim A Khorwat, Nabil Naas, “ A new hardware Implementation of
Manchester Line encoder”, 69t