2 a) a. Unipolar NRZ

clear all;

close all;

clc;

%bits = input('Enter bit sequence: ');

bits = [1 0 1 1 0 1 0 0];

bitrate = 1;

n = 1000;

T = length(bits)/bitrate;

N = n\*length(bits);

dt = T/N;

t = 0:dt:T-dt;

%Encoding

x = zeros(1,length(t));

for i=1:length(bits)

if bits(i)==1

x((i-1)\*n+1:i\*(n)) = 1;

else x((i-1)\*n+1:i\*(n)) = 0;

end

end

subplot(2, 1, 1);

plot(t, x, 'Linewidth', 3);

axis([0,length(bits),-2,2]);

title('Encoded signal (Unipolar-NRZ)');

xlabel('Time','fontweight','bold','fontsize',12);

ylabel('Amplitude','fontweight','bold','fontsize',12);

disp('NRZ-L Encoding:');

grid on;

disp(x);

%decoding

counter = 0;

for i = 1:length(t)

if t(i)>counter

counter = counter + 1;

if x(i)>0

result(counter) = x(i);

else result(counter) = 0;

end

end

end

t1 = 1 : bitrate : length(bits);

subplot(2, 1, 2);

stem(t1, result, 'Linewidth', 3);

axis([1,length(bits),-2,2]);

title('Decocoded Digital Data(Unipolar-NRZ)');

xlabel('Time','fontweight','bold','fontsize',12);

ylabel('Amplitude','fontweight','bold','fontsize',12);

grid on;

disp('NRZ-L Decoding:');

disp(result);

b. Polar NRZ-L NRZ-I,

clear all;

close all;

clc;

%bits = input('Enter bit sequence: ');

bits = [0 1 0 0 1 1 1 0];

bitrate = 1;

n = 1000;

T = length(bits)/bitrate;

N = n\*length(bits);

dt = T/N;

t = 0:dt:T-dt;

%Encoding

x = zeros(1,length(t));

for i=1:length(bits)

if bits(i) == 1

x((i-1)\*n+1:i\*(n)) = -1;

else x((i-1)\*n+1:i\*(n)) = 1;

end

end

subplot(2, 1, 1);

plot(t, x, 'Linewidth', 3);

axis([0,length(bits),-2,2]);

title('Encoded signal (POLAR NRZ-L)');

xlabel('Time','fontweight','bold','fontsize',12);

ylabel('Amplitude','fontweight','bold','fontsize',12);

disp('NRZ-L Encoding:');

grid on;

disp(x);

%decoding

counter = 0;

for i = 1 : length(t)

if t(i)> counter

counter = counter + 1;

if x(i) < 0

result(counter) = 1;

else

result(counter) = 0;

end

end

end

t1 = 1 : bitrate : length(bits);

subplot(2, 1, 2);

stem(t1, result, 'Linewidth', 3);

axis([1,length(bits),-2,2]);

title('Decocoded Digital Data (POLAR NRZ-L)');

xlabel('Time','fontweight','bold','fontsize',12);

ylabel('Amplitude','fontweight','bold','fontsize',12);

grid on;

disp('NRZ-L Decoding:');

disp(result);

b NRZ I:

clear all;

close all;

clc;

%bits = input('Enter bit sequence: ');

bits = [0 1 0 0 1 0 1 0];

bitrate = 1;

n = 1000;

T = length(bits)/bitrate;

N = n\*length(bits);

dt = T/N;

t = 0:dt:T;

% NRZ-I Encoding

x = zeros(1, length(t));

prev\_bit = 1; % Initialize previous bit with 1

for i = 1:length(bits)

if bits(i) == 1

x((i-1)\*n+1:i\*n) = -prev\_bit;

prev\_bit = -prev\_bit; % Invert the polarity for consecutive 1s

else

x((i-1)\*n+1:i\*n) = prev\_bit;

end

end

% Plotting the encoded signal

plot(t, x, 'Linewidth', 3);

axis([0, length(bits), -2, 2]);

title('POLAR NRZ-I', 'fontweight', 'bold', 'fontsize', 20);

xlabel('Time', 'fontweight', 'bold', 'fontsize', 20);

ylabel('Amplitude', 'fontweight', 'bold', 'fontsize', 20);

disp('NRZ-I Encoding:');

grid on;

disp(x);

% For data bits

ax1 = gca;

ax2 = axes('Position', get(ax1, 'Position'), 'Color', 'none');

set(ax2, 'XAxisLocation', 'top');

set(ax2, 'XLim', get(ax1, 'XLim'));

set(ax2, 'YLim', [-2, 2]);

set(ax2, 'XTick', [bitrate/2:bitrate:T]);

set(ax2, 'YTick', []);

set(ax2, 'XTickLabel', bits, 'fontweight', 'bold', 'fontsize', 18);

% Decoding

counter = 0;

decoded\_bits = zeros(1, length(bits));

prev\_polarity = 1; % Initialize previous polarity with 1

for i = 1:length(t)

if t(i) > counter

counter = counter + 1;

if x(i) == prev\_polarity

decoded\_bits(counter) = 0;

else

decoded\_bits(counter) = 1;

prev\_polarity = -prev\_polarity; % Invert the polarity for consecutive 1s

end

end

end

disp('NRZ-I Decoding:');

disp(decoded\_bits);

c Polar RZ:

clear all;

close all;

clc;

%bits = input('prompt');

bits = [0 1 0 0 1 0 1 1];

bitrate = 1;

n = 1000;

T = length(bits)/bitrate;

N = n\*length(bits);

dt = T/N;

t = 0:dt:T;

x = zeros(1,length(t));

%Encoding

for i=1:length(bits)

if bits(i)==1

x((i-1)\*n+1:(i-1)\*n+n/2) = 1;

else x((i-1)\*n+1:(i-1)\*n+n/2) = -1;

end

end

subplot(2, 1, 1);

plot(t, x, 'Linewidth', 3);

axis([0,length(bits),-2,2]);

title('Encoded signal (POLAR-RZ)');

xlabel('Time','fontweight','bold','fontsize',12);

ylabel('Amplitude','fontweight','bold','fontsize',12);

disp('NRZ-I Encoding:');

grid on;

disp(x);

%Decding

counter = 0;

for i = 1 : length(t)

if t(i)> counter

counter = counter + 1;

if x(i) < 0

result(counter) = 0;

else

result(counter) = 1;

end

end

end

t1 = 1 : bitrate : length(bits);

subplot(2, 1, 2);

stem(t1, result, 'Linewidth', 3);

axis([0,length(bits),-2,2]);

title('Decocoded Digital Data (POLAR-RZ)');

xlabel('Time','fontweight','bold','fontsize',12);

ylabel('Amplitude','fontweight','bold','fontsize',12);

grid on;

disp('NRZ-L Decoding:');

disp(result);

Manchester:

clear all;

close all;

clc;

%bits = input('prompt');

bits = [0 1 0 0 1 1 0 1];

bitrate = 1;

n = 1000;

T = length(bits)/bitrate;

N = n\*length(bits);

dt = T/N;

t = 0:dt:T-dt;

x = zeros(1,length(t));

%encoding

for i=1:length(bits)

if bits(i)==1

x((i-1)\*n+1:(i-1)\*n+n/2) = -1;

x((i-1)\*n+n/2:i\*n) = 1;

else

x((i-1)\*n+1:(i-1)\*n+n/2) = 1;

x((i-1)\*n+n/2:i\*n) = -1;

end

end

subplot(2, 1, 1);

plot(t, x, 'Linewidth', 3);

axis([0, length(bits), -2, 2]);

title('Encoded signal (Polar-Biphase: Manchester)');

xlabel('Time','fontweight','bold','fontsize',12);

ylabel('Amplitude','fontweight','bold','fontsize',12);

grid on;

%Decoding

counter = 0;

for i = 1:length(t)

if t(i)>counter

counter = counter + 1;

if x(i)>0

result(counter) = 0;

else

result(counter) = 1;

end

end

end

subplot(2, 1, 2);

t1 = 0 : bitrate : length(bits) - 1;

stem(t1, result, 'linewidth', 3);

axis([0, length(bits)-1, -2, 2]);

title('Decoded Digital Data (Polar-Biphase: Manchester)');

xlabel('Time','fontweight','bold','fontsize',12);

ylabel('Amplitude','fontweight','bold','fontsize',12);

grid on;

disp('Manchester Decoding:');

disp(result);

3 different mainchester

clear all;

close all;

clc;

%bits = input('prompt');

bits = [1 0 1 1 0 1 0 0];

bitrate = 1;

n = 1000;

T = length(bits)/bitrate;

N = n\*length(bits);

dt = T/N;

t = 0:dt:T-dt;

x = zeros(1, length(t));

lastbit = 1;

for i=1:length(bits)

if bits(i)==0

x((i-1)\*n+1:(i-1)\*n+n/2) = -lastbit;

x((i-1)\*n+n/2:i\*n) = lastbit;

else

x((i-1)\*n+1:(i-1)\*n+n/2) = lastbit;

x((i-1)\*n+n/2:i\*n) = -lastbit;

lastbit = -lastbit;

end

end

subplot(2,1,1);

plot(t, x, 'Linewidth', 3);

axis([0, length(bits), -2, 2]);

title('Encoded signal (Polar-Biphase: Differential Manchester)');

xlabel('Time','fontweight','bold','fontsize',12);

ylabel('Amplitude','fontweight','bold','fontsize',12);

grid on;

%decoding

counter = 0;

lastbit = 1;

for i = 1:length(t)

if t(i)>counter

counter = counter + 1;

if x(i)==lastbit

result(counter) = 1;

lastbit = -lastbit;

else result(counter) = 0;

end

end

end

t1 = 0 : bitrate : length(bits) - 1;

subplot(2,1,2);

stem(t1, result, 'linewidth', 3);

axis([0, length(bits) - 1, -2, 2]);

title('Decoded Digital Data (Polar-Biphase: Differential Manchester)');

xlabel('Time','fontweight','bold','fontsize',12);

ylabel('Amplitude','fontweight','bold','fontsize',12);

disp('Differential Manchester Decoding:');

disp(result);

D) bipolar ami

clear all;

close all;

clc;

%bits = input('prompt');

bits = [0 1 0 0 1 0 1 1];

bitrate = 1;

n = 1000;

T = length(bits)/bitrate;

N = n\*length(bits);

dt = T/N;

t = 0:dt:T-dt;

x = zeros(1,length(t));

lastbit = -1;

for i=1:length(bits)

if bits(i)==1

x((i-1)\*n+1:i\*n) = -lastbit;

lastbit = -lastbit;

end

end

subplot(2,1,1);

plot(t, x, 'Linewidth', 3);

axis([0, length(bits), -2, 2]);

title('Encoded signal (Bipolar: AMI)');

xlabel('Time','fontweight','bold','fontsize',12);

ylabel('Amplitude','fontweight','bold','fontsize',12);

grid on;

%decoding

counter = 0;

lastbit = 1;

for i = 1:length(t)

if t(i)>counter

counter = counter + 1;

if x(i)== 0

result(counter) = 0;

else

result(counter) = 1;

end

end

end

t1 = 0 : bitrate : length(bits) - 1;

subplot(2,1,2);

stem(t1, result, 'Linewidth', 3);

axis([0, length(bits) - 1, -2, 2]);

title('Decoded Digiatal Data (Bipolar: AMI)');

xlabel('Time','fontweight','bold','fontsize',12);

ylabel('Amplitude','fontweight','bold','fontsize',12);

grid on;

disp('AMI Decoding:');

disp(result);

E MLT 3

clc;

clear all;

close all;

data = [0 1 0 1 1 0 1 1];

point = 100;

flag = [1 0 -1 0];

status = 0;

index = 1;

encoded\_signal = zeros(1,length(data)\*point);

for i=1:length(data)

if data(i) == 1

status = flag(index)

encoded\_signal((i-1)\*point+1:i\*point) = status;

index = mod(index,4)+1;

else

encoded\_signal((i-1)\*point+1:i\*point) = status;

end

end

size = 0:1/point:length(data)-1/point;

plot(size,encoded\_signal, 'Linewidth', 3);

axis([0, length(data), -2, 2]);

title('MLT-3 Encoding', 'fontweight', 'bold', 'fontsize', 18);

ylim([-2,2]);

disp('MLT-3 Encoding:');

grid on;

disp(encoded\_signal);

%Decoded\_Signal

%status same -> 0

%status change -> 1

status = 0;

decoded\_signal = zeros(1,length(data));

for i=1:length(data)

if encoded\_signal(i\*point)==status

decoded\_signal(i) = 0;

else

decoded\_signal(i) = 1;

status = encoded\_signal(i\*point);

end

end

disp('MLT-3 Decoding:');

disp(decoded\_signal);

B8ZS

clear all;

close all;

clc;

%bits = input('prompt');

bits = [1 0 0 0 0 0 0 0 0 1];

bitrate = 1;

n = 1000;

T = length(bits)/bitrate;

N = n\*length(bits);

dt = T/N;

t = 0:dt:T-dt;

x = zeros(1,length(t));

counter = 0;

lastbit = 1;

%encoding

for i=1:length(bits)

if bits(i)==0

counter = counter + 1;

if counter==8

x((i-1-7)\*n+1:(i-7)\*n) = 0;

x((i-1-6)\*n+1:(i-6)\*n) = 0;

x((i-1-5)\*n+1:(i-5)\*n) = 0;

x((i-1-4)\*n+1:(i-4)\*n) = lastbit;

x((i-1-3)\*n+1:(i-3)\*n) = -lastbit;

lastbit = -lastbit;

x((i-1-2)\*n+1:(i-2)\*n) = 0;

x((i-1-1)\*n+1:(i-1)\*n) = lastbit;

x((i-1)\*n+1:i\*n) = -lastbit;

lastbit = -lastbit;

counter = 0;

end

else

counter = 0;

x((i-1)\*n+1:i\*n) = -lastbit;

lastbit = -lastbit;

end

end

subplot(2,1,1);

plot(t, x, 'Linewidth', 3);

axis([0, length(bits), -2, 2]);

grid on;

%decoding

counter = 0;

lastbit = 1;

for i = 1:length(t)

if t(i)>counter

counter = counter + 1;

if x(i)==lastbit

result(counter:counter+4) = 0;

counter = counter + 4;

else

if(x(i)==0)

result(counter) = 0;

else

result(counter) = 1;

lastbit = -lastbit;

end

end

end

end

t1 = 0 : bitrate : length(bits) - 1;

subplot(2,1,2);

stem(t1, result, 'Linewidth', 3);

axis([0, length(bits)-1, -2, 2]);

grid on;

disp('B8ZS Decoding:');

disp(result);

HDB3

clear all;

close all;

clc;

%bits = input('prompt');

bits = [0 1 1 0 0 0 0 1 0 0 0 0 1];

bitrate = 1;

n = 1000;

T = length(bits)/bitrate;

N = n\*length(bits);

dt = T/N;

t = 0:dt:T-dt;

x = zeros(1,length(t));

counter = 0;

lastbit = 1;

pulse = 0;

%encoding

for i=1:length(bits)

if bits(i)==0

counter = counter + 1;

if counter==4

if(mod(pulse, 2)==0)

x((i-1-3)\*n+1:(i-3)\*n) = -lastbit;

lastbit = -lastbit;

x((i-1-2)\*n+1:(i-2)\*n) = 0;

x((i-1-1)\*n+1:(i-1)\*n) = 0;

x((i-1)\*n+1:i\*n) = lastbit;

counter = 0;

pulse = 0;

else

x((i-1-3)\*n+1:(i-3)\*n) = 0;

x((i-1-2)\*n+1:(i-2)\*n) = 0;

x((i-1-1)\*n+1:(i-1)\*n) = 0;

x((i-1)\*n+1:i\*n) = lastbit;

counter = 0;

pulse = 0;

end

end

else

counter = 0;

x((i-1)\*n+1:i\*n) = -lastbit;

lastbit = -lastbit;

pulse = pulse + 1;

end

end

subplot(2,1,1);

plot(t, x, 'Linewidth', 3);

axis([0, length(bits)-1, -2, 2]);

grid on;

counter = 0;

lastbit = 1;

for i = 1:length(t)

if t(i)>counter

counter = counter + 1;

if x(i)==lastbit

result(counter-3:counter) = 0;

else

if(x(i)==0)

result(counter) = 0;

else

result(counter) = 1;

lastbit = -lastbit;

end

end

end

end

t1 = 0 : bitrate : length(bits)-1;

subplot(2,1,2);

stem(t1, result, 'Linewidth', 3);

axis([0, length(bits)-1, -2, 2]);

grid on;

disp('HDB3 Decoding:');

disp(result);