VIT

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

SCHOOL OF ELECTRONICS ENGINEERING

Winter Semester 2024-2025

BECE301P – Digital Signal Processing LAB

L47 +L48

FACULTY: SUDHAKAR M S

TASK-5

Obtain the output waveforms pertaining to the individual blocks that formulate an Adaptive Differential Pulse Code Modulation scheme given in Figure 1. This modulation scheme forms the essential block of the G.726 ITU transactions for audio compression.

DONE BY

KAUSHIK KUMAR PS

23BEC0142

Task 5:

Obtain the output waveforms pertaining to the individual blocks that formulate an Adaptive Differential Pulse Code Modulation scheme given in Figure 1. This modulation scheme forms the essential block of the G.726 ITU transactions for audio compression.

Adaptive Differential Pulse Code Modulation (ADPCM)

Block Diagram: -

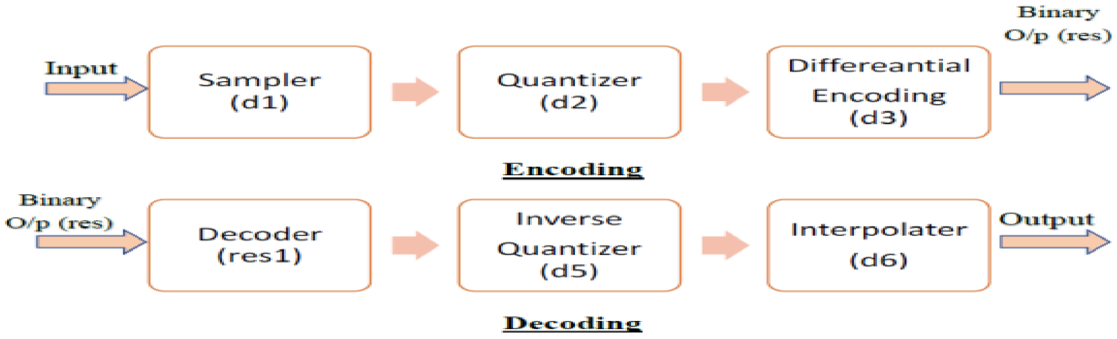
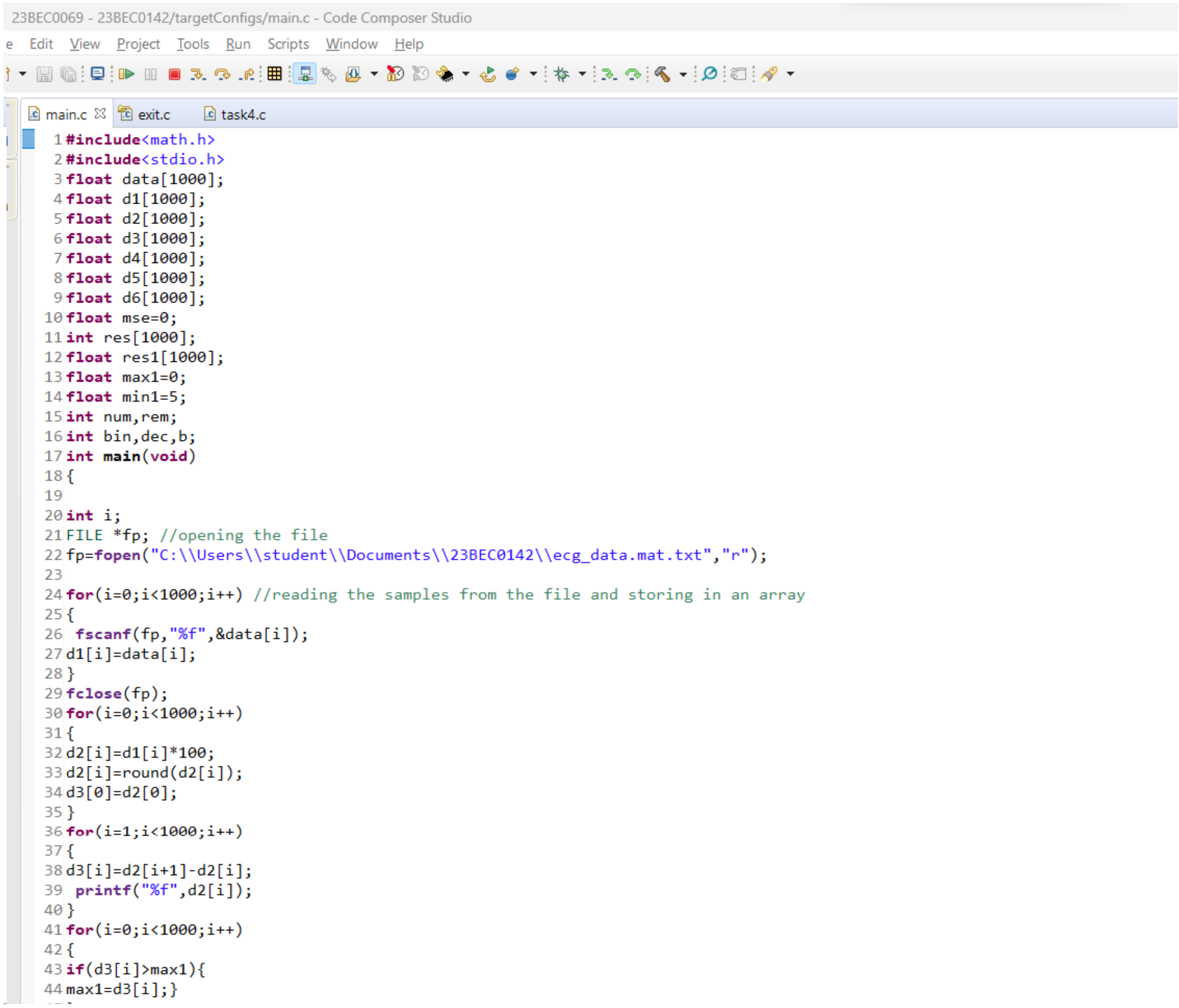
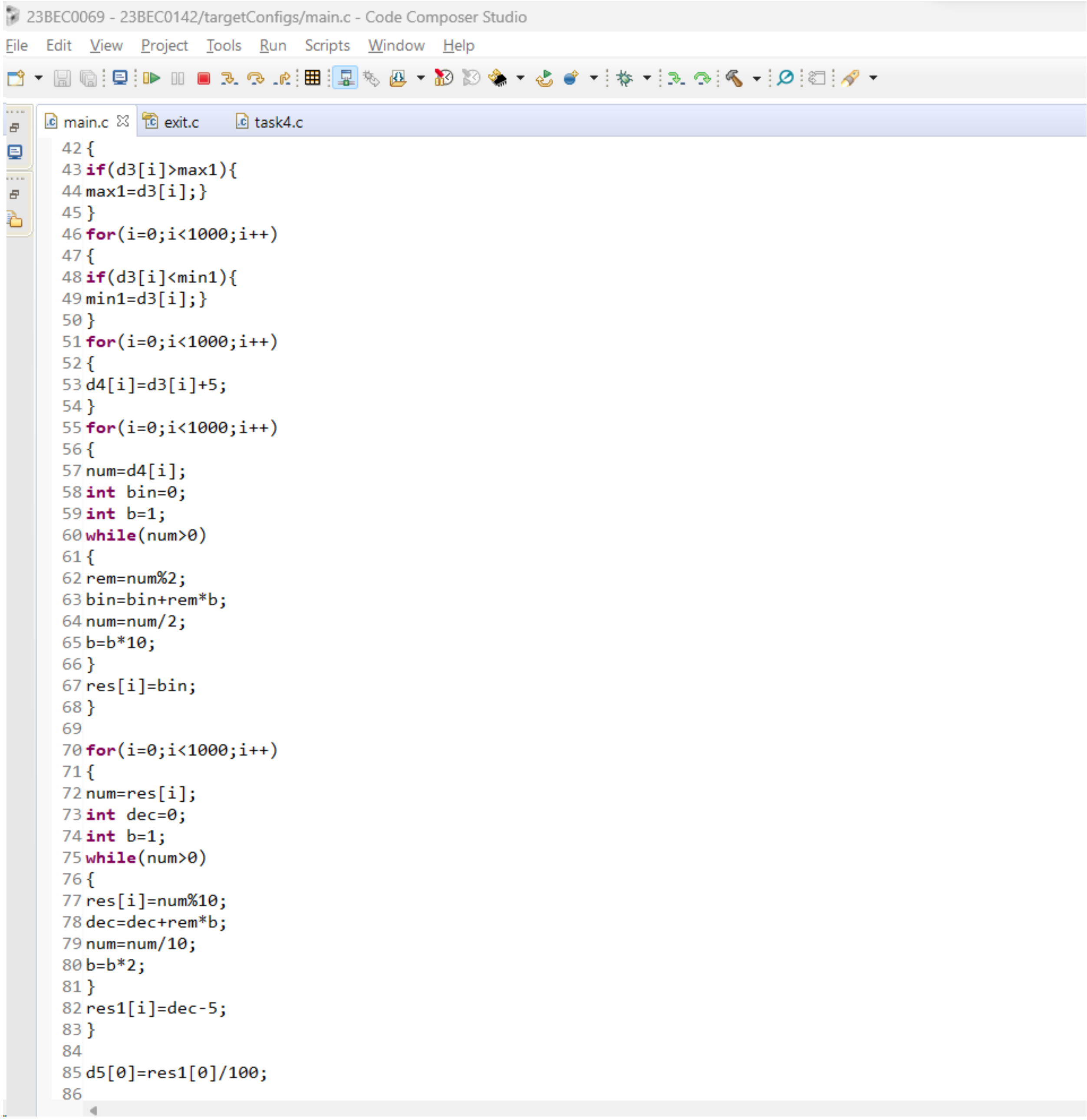


Figure 1

CODE:

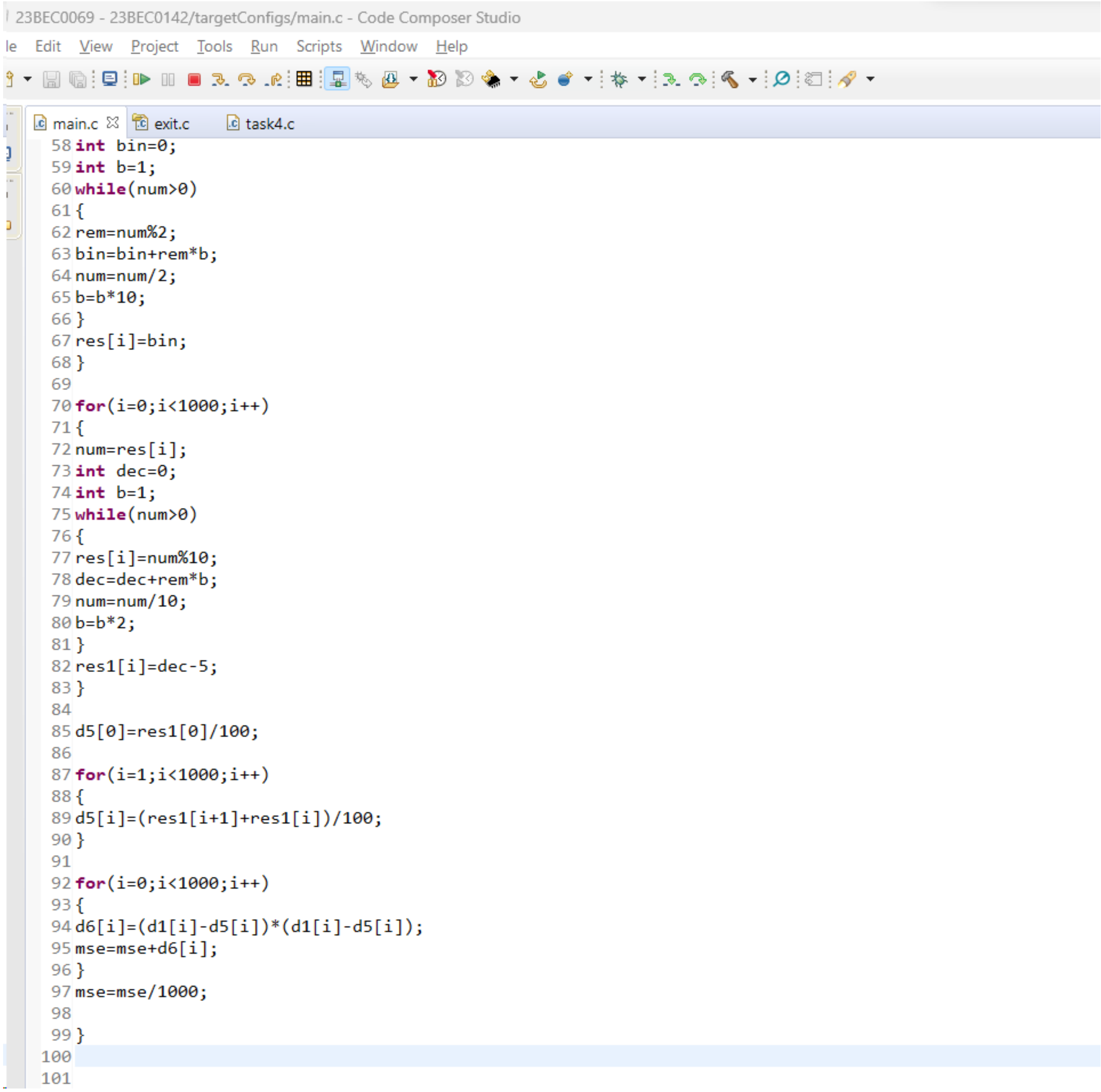

23BEC0069 - 23BEC0142/targetConfigs/main.c - Code Composer Studio
e Edit View Project Tools Run Scripts Window Help
main.c
exit.c
task4.c

```
#include<math.h>
#include<stdio.h>
float data[1000];
float d1[1000];
float d2[1000];
float d3[1000];
float d4[1000];
float d5[1000];
float d6[1000];
float mse=0;
int res[1000];
float res1[1000];
float max1=0;
float min1=5;
int num,rem;
int bin,dec,b;
int main(void)
{

int i;
FILE *fp; //opening the file
fp=fopen("C:\\Users\\student\\Documents\\23BEC0142\\ecg_data.mat.txt","r");

for(i=0;i<1000;i++) //reading the samples from the file and storing in an array
{
 fscanf(fp,"%f",&data[i]);
d1[i]=data[i];
}
fclose(fp);
for(i=0;i<1000;i++)
{
d2[i]=d1[i]*100;
d2[i]=round(d2[i]);
d3[0]=d2[0];
}
for(i=1;i<1000;i++)
{
d3[i]=d2[i+1]-d2[i];
 printf("%f",d2[i]);
}
for(i=0;i<1000;i++)
{
if(d3[i]>max1){
max1=d3[i];}
```



23BEC0069 - 23BEC0142/targetConfigs/main.c - Code Composer Studio
File Edit View Project Tools Run Scripts Window Help
main.c
exit.c
task4.c

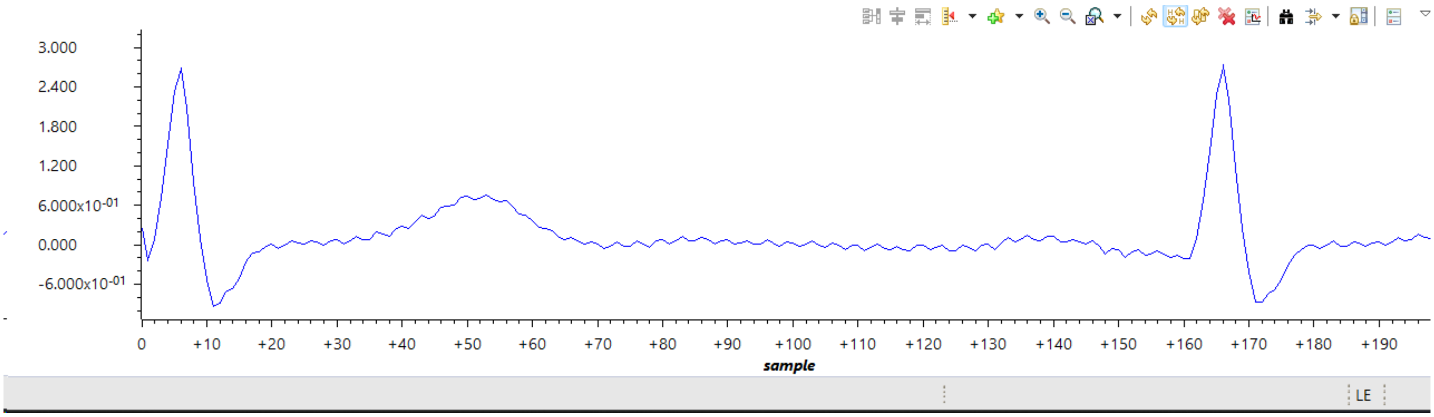
```
{
if(d3[i]>max1){
max1=d3[i];}
}
for(i=0;i<1000;i++)
{
if(d3[i]<min1){
min1=d3[i];}
}
for(i=0;i<1000;i++)
{
d4[i]=d3[i]+5;
}
for(i=0;i<1000;i++)
{
num=d4[i];
int bin=0;
int b=1;
while(num>0)
{
rem=num%2;
bin=bin+rem*b;
num=num/2;
b=b*10;
}
res[i]=bin;
}

for(i=0;i<1000;i++)
{
num=res[i];
int dec=0;
int b=1;
while(num>0)
{
res[i]=num%10;
dec=dec+rem*b;
num=num/10;
b=b*2;
}
res1[i]=dec-5;
}

d5[0]=res1[0]/100;

```


23BEC0069 - 23BEC0142/targetConfigs/main.c - Code Composer Studio
le Edit View Project Tools Run Scripts Window Help
main.c
exit.c
task4.c

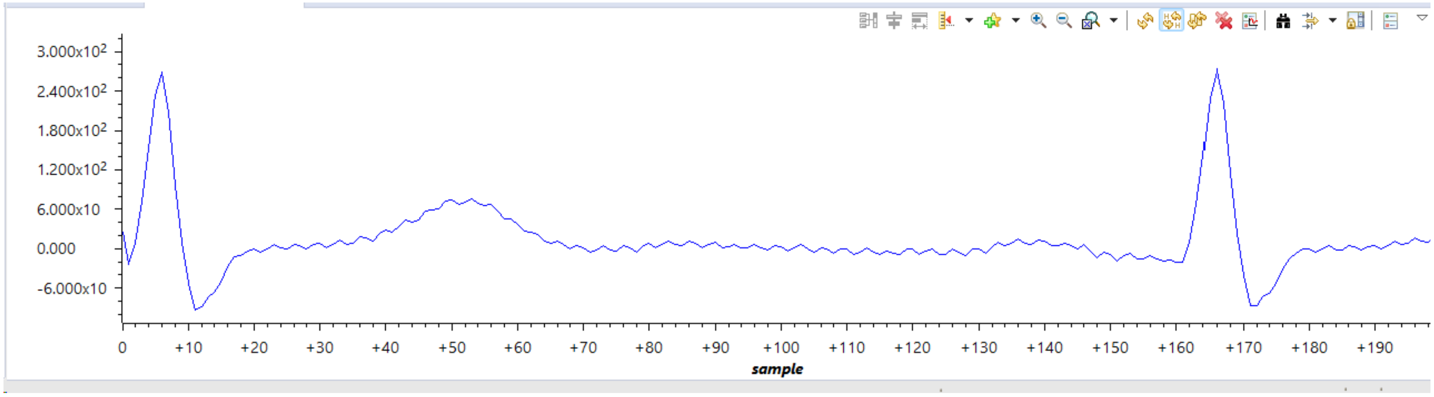
```
int bin=0;
int b=1;
while(num>0)
{
rem=num%2;
bin=bin+rem*b;
num=num/2;
b=b*10;
}
res[i]=bin;
}

for(i=0;i<1000;i++)
{
num=res[i];
int dec=0;
int b=1;
while(num>0)
{
res[i]=num%10;
dec=dec+rem*b;
num=num/10;
b=b*2;
}
res1[i]=dec-5;
}

d5[0]=res1[0]/100;

for(i=1;i<1000;i++)
{
d5[i]=(res1[i+1]+res1[i])/100;
}

for(i=0;i<1000;i++)
{
d6[i]=(d1[i]-d5[i])*(d1[i]-d5[i]);
mse=mse+d6[i];
}
mse=mse/1000;

}


```

WAVEFORMS:

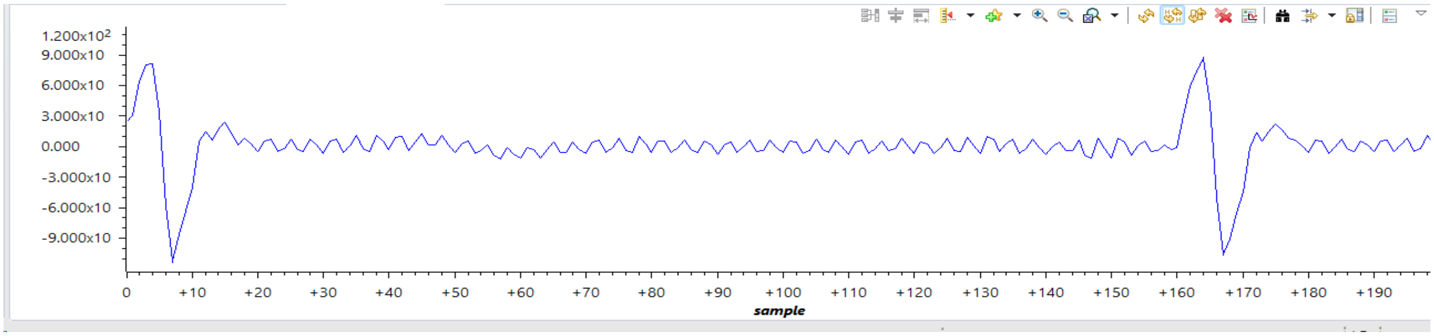
d1



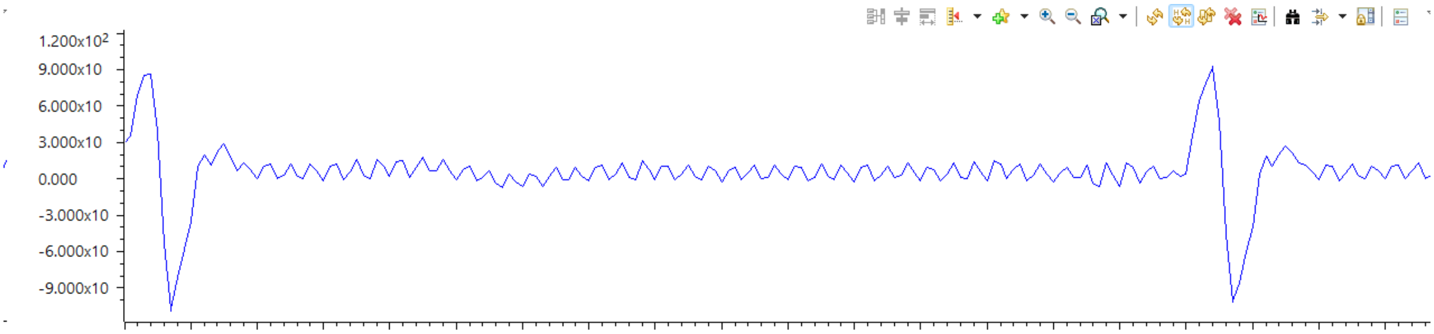
d2



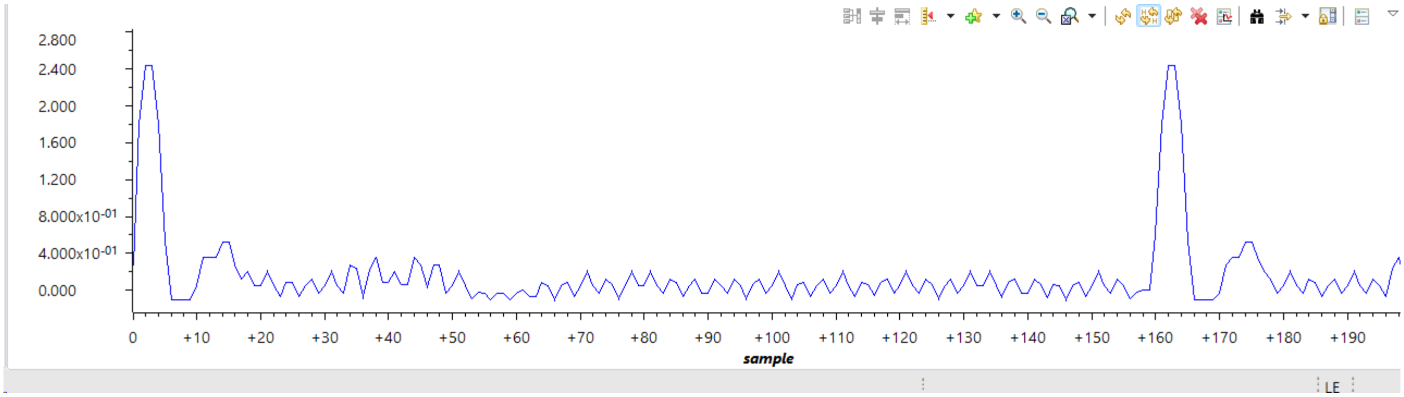
d3



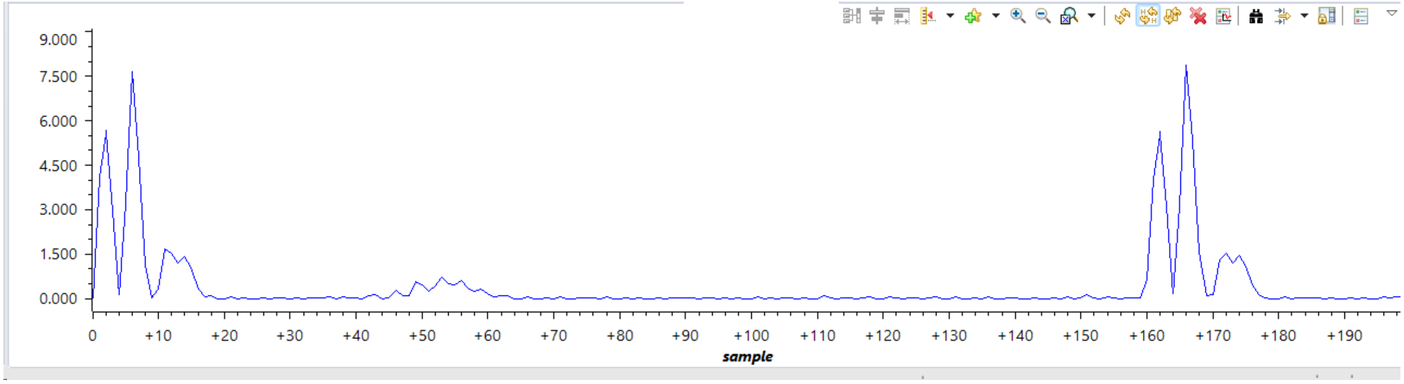
d4



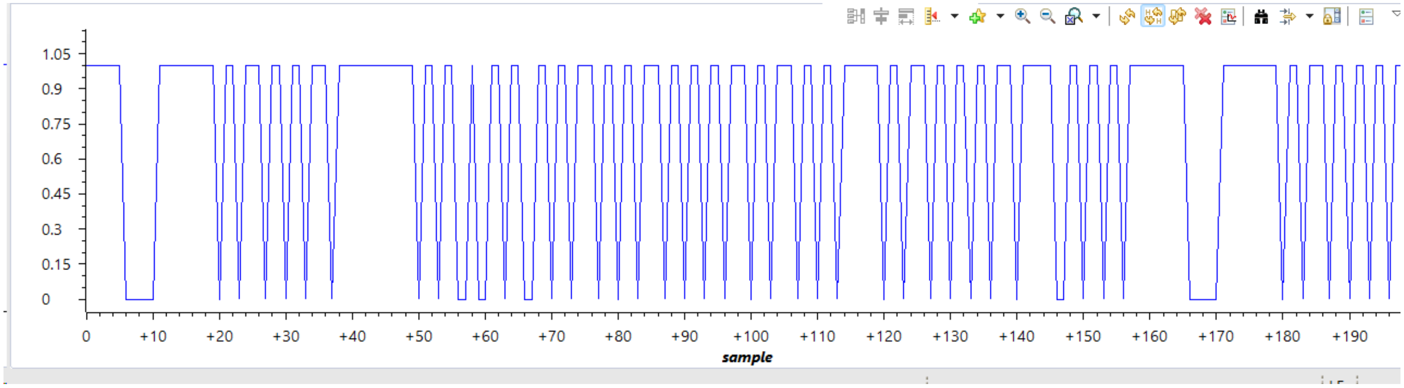
d5



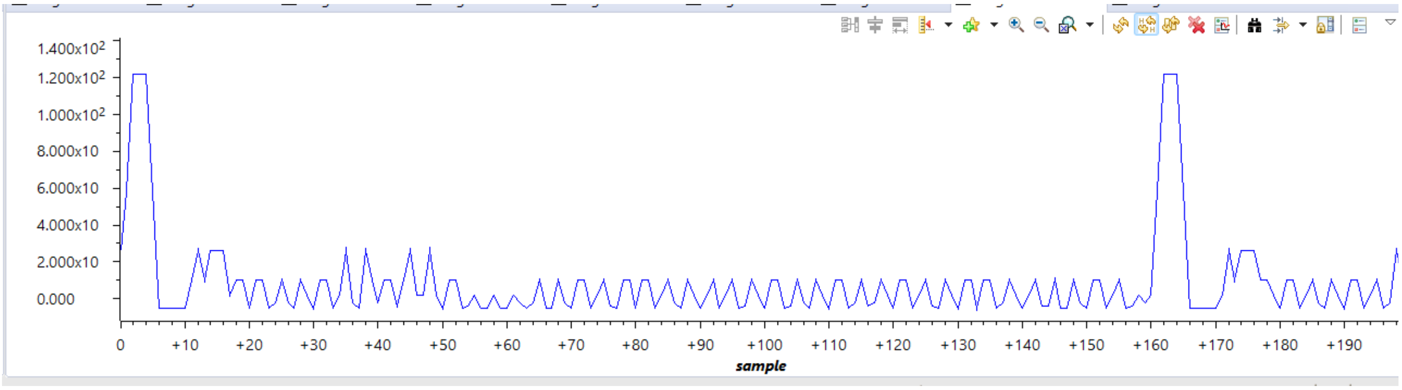
d6



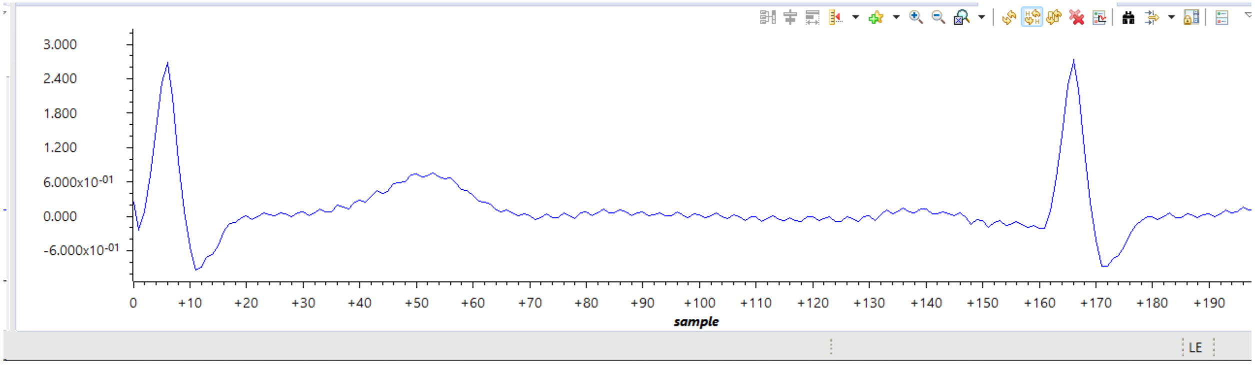
res



res1



Data


3.000
2.400
1.800
1.200
6.000x10^{-01}
0.000
-6.000x10^{-01}
0
+10
+20
+30
+40
+50
+60
+70
+80
+90
+100
+110
+120
+130
+140
+150
+160
+170
+180
+190
sample
LE