INDIAN INSTITUTE OF INFORMATION TECHNOLOGY,
DESIGN AND MANUFACTURING,
KANCHEEPURAM

DSP System Design Practice LAB 2 Report

K.Nithesh
ESD19i008

Aim:

To implement Convolution and Correlation on the MicroDSP6748 DSP Board.

Apparatus Required:

Code Composer Studio, TMS320C6748 DSP board, XDS100v3 Emulator.

Procedure for Convolution:

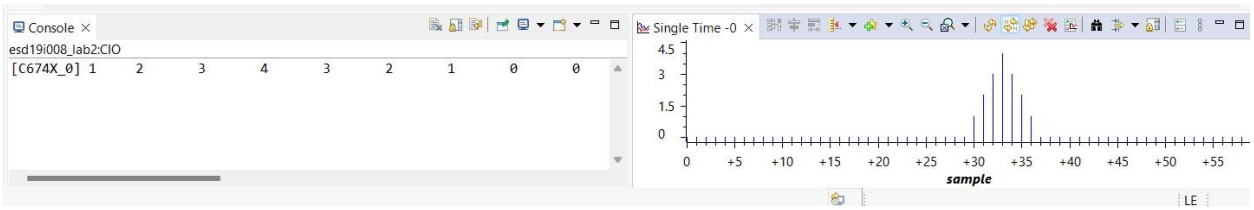
Convolution is a mathematical operation that takes two signals and produces a third signal that represents the overlap between them. Convolution is widely used in various fields such as digital signal processing, image processing, and machine learning.

The following steps can be followed to implement Convolution on the MicroDSP6748 DSP Board:

1. Load the input signals: The first step is to load the input signals into the DSP board's memory. The input signals can be stored in arrays.
2. Pad the input signals: If the input signals are not of the same length, they need to be padded with zeros to make them the same length. This is important for the convolution operation.
3. Flip the filter coefficients: The filter coefficients need to be flipped for the convolution operation to be performed correctly. This can be done by reversing the order of the coefficients.
4. Perform the convolution operation: The convolution operation can be performed using a loop that iterates through each sample of the input signals. For each sample, the corresponding coefficients are multiplied and the result is accumulated.
5. Store the output signal: After the convolution operation is completed, the output signal can be stored in an array or written to a file for further processing.

Linear Convolution:

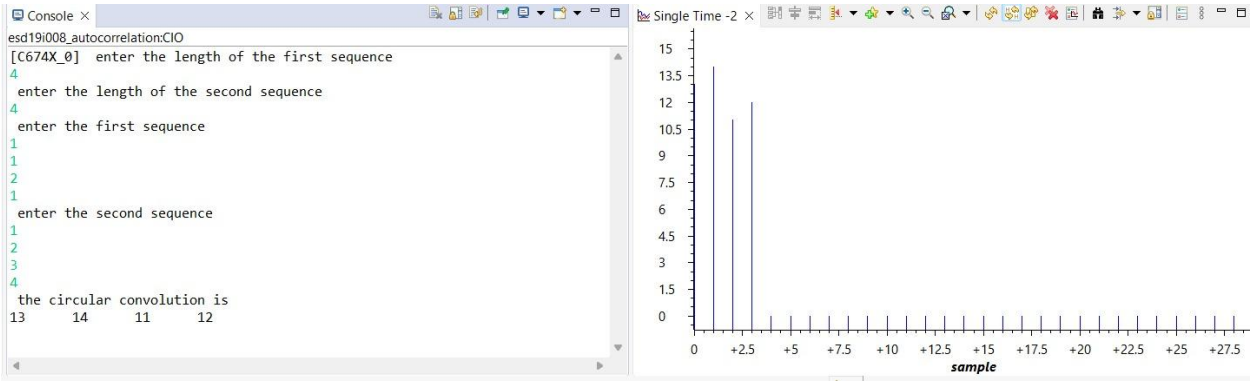
```
#include<stdio.h>
int y[30];
int i,j;
int x[14]={1,1,1,1,0,0,0,0,0,0,0,0,0,0};
int k[14]={1,1,1,1,0,0,0,0,0,0,0,0,0,0};
void main()
{

    for(i=0;i<=27;i++)
    {
     y[i]=0;
        for(j=0;j<=13;j++)
        {
            if((i-j+1)>0 && (i-j)<14)
            y[i]=y[i]+x[j]*k[i-j];

        }
            printf("%d\t",y[i]);
    }
}
```



Circular Convolution:

```
#include<stdio.h>

int m,n,x[30],h[30],y[30],i,j, k,x2[30],a[30];

void main()
{
            printf(" enter the length of the first sequence\n");
            scanf("%d",&m);
            printf(" enter the length of the second sequence\n");
            scanf("%d",&n);

            printf(" enter the first sequence\n");
            for(i=0;i<m;i++)
                        scanf("%d",&x[i]);

            printf(" enter the second sequence\n");
            for(j=0;j<n;j++)
                        scanf("%d",&h[j]);

            if(m-n!=0)        /If length of both sequences are not equal/
            {
                        if(m>n)                          /* Pad the smaller
sequence with zero*/
                        {
                        for(i=n;i<m;i++)
                                    h[i]=0;
                                    n=m;
                        }
                        for(i=m;i<n;i++)
                                    x[i]=0;
                                    m=n;
            }

            y[0]=0;
            a[0]=h[0];
```

```
            for(j=1;j<n;j++)            /folding h(n) to h(-n)/
                        a[j]=h[n-j];

            /Circular convolution/
            for(i=0;i<n;i++)
                        y[0]+=x[i]*a[i];

            for(k=1;k<n;k++)
            {
                        y[k]=0;
                        /circular shift/

                        for(j=1;j<n;j++)
                                    x2[j]=a[j-1];
                                    x2[0]=a[n-1];
                        for(i=0;i<n;i++)
                        {
                                    a[i]=x2[i];
                                    y[k]+=x[i]*x2[i];
                        }
            }

            /displaying the result/
            printf(" the circular convolution is\n");
            for(i=0;i<n;i++)
                        printf("%d \t",y[i]);
}
```



Procedure for Correlation:

Correlation is a mathematical operation that measures the similarity between two signals. Correlation is widely used in various fields such as digital signal processing, image processing, and machine learning.

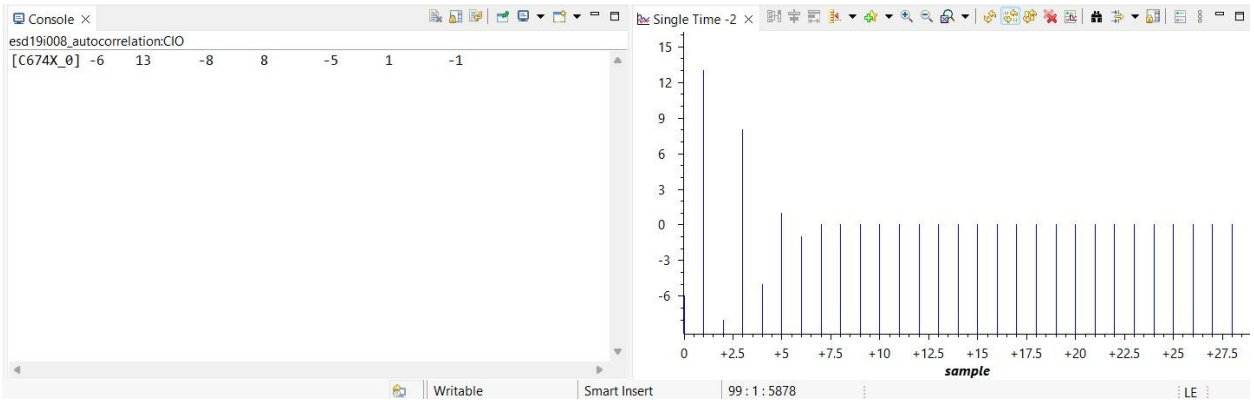
The following steps can be followed to implement Correlation on the MicroDSP6748 DSP Board:

1. Load the input signals: The first step is to load the input signals into the DSP board's memory. The input signals can be stored in arrays.
2. Zero-pad the input signals: If the input signals are not of the same length, they need to be padded with zeros to make them the same length. This is important for the correlation operation.
3. Perform the correlation operation: The correlation operation can be performed using a loop that iterates through each sample of the input signals. For each sample, the corresponding samples from the two signals are multiplied and the result is accumulated.
4. Store the output signal: After the correlation operation is completed, the output signal can be stored in an array or written to a file for further processing.

Cross correlation:

```
// Online C compiler to run C program online
#include <stdio.h>
int y[30];
int x[15]={-3,2,-1,1};
int h[15]={2,-3,0,-1};//given h[n]= {-1,0,-3,2} take reverse of h[n]

int main() {
    // Write C code here

    int n,a,b,k;
    a=3;
    b=3;

    for (n=0;n<(a+b-1);n++)
    {
        y[n]=0;
```

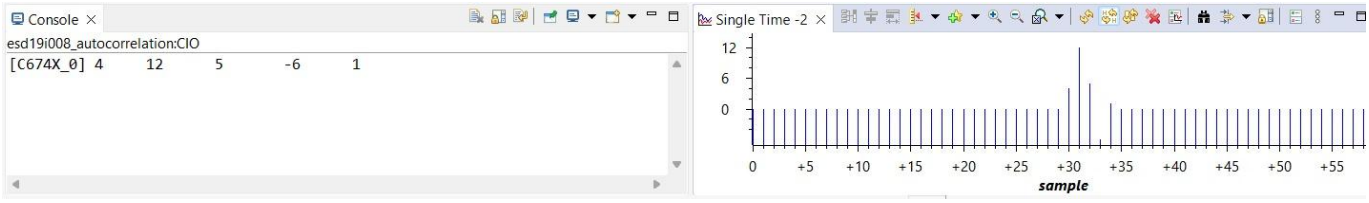
```
        for (k=0;k<3;k++)
        {
           if (n-k+1>0 && n-k<b)
              y[n]=y[n]+x[k]*h[n-k];
        }
        printf("%d\t",y[n]);
    }
}
```



Autocorrelation:

```
// Online C compiler to run C program online
#include <stdio.h>
int y[30];
int x[15]={2,3,-1};
int h[15]={2,3,-1};

int main() {
    // Write C code here

    int n,a,b,k;
    a=3;
    b=3;

    for (n=0;n<(a+b-1);n++)
    {
        y[n]=0;

        for (k=0;k<3;k++)
        {
           if (n-k+1>0 && n-k<b)
              y[n]=y[n]+x[k]*h[n-k];
```

```
        }
        printf("%d\t",y[n]);
    }
}
```



Results:

Implemented Convolution and Correlation on MicroDSP6748 DSP Board.