

**DEPARTMENT OF  
ELECTRONICS AND COMMUNICATION ENGINEERING**

**EC6511 – DIGITAL SIGNAL PROCESSING LABORATORY**

**V SEMESTER - R 2013**

**LABORATORY MANUAL**

Name : \_\_\_\_\_

Register No. : \_\_\_\_\_

Section : \_\_\_\_\_

**VISION**

is committed to provide highly disciplined, conscientious and enterprising professionals conforming to global standards through value based quality education and training.

**MISSION**

- To provide competent technical manpower capable of meeting requirements of the industry
- To contribute to the promotion of Academic Excellence in pursuit of Technical Education at different levels
- To train the students to sell his brawn and brain to the highest bidder but to never put a price tag on heart and soul

**DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING****VISION**

To impart professional education integrated with human values to the younger generation, so as to shape them as proficient and dedicated engineers, capable of providing comprehensive solutions to the challenges in deploying technology for the service of humanity

**MISSION**

- To educate the students with the state-of-art technologies to meet the growing challenges of the electronics industry
- To carry out research through continuous interaction with research institutes and industry, on advances in communication systems
- To provide the students with strong ground rules to facilitate them for systematic learning, innovation and ethical practices

## **PROGRAMME EDUCATIONAL OBJECTIVES (PEOs)**

### **1. Fundamentals**

To provide students with a solid foundation in Mathematics, Science and fundamentals of engineering, enabling them to apply, to find solutions for engineering problems and use this knowledge to acquire higher education

### **2. Core Competence**

To train the students in Electronics and Communication technologies so that they apply their knowledge and training to compare, and to analyze various engineering industrial problems to find solutions

### **3. Breadth**

To provide relevant training and experience to bridge the gap between theory and practice which enables them to find solutions for the real time problems in industry, and to design products

### **4. Professionalism**

To inculcate professional and effective communication skills, leadership qualities and team spirit in the students to make them multi-faceted personalities and develop their ability to relate engineering issues to broader social context

### **5. Lifelong Learning/Ethics**

To demonstrate and practice ethical and professional responsibilities in the industry and society in the large, through commitment and lifelong learning needed for successful professional career

## **PROGRAMME OUTCOME (POs)**

- a) To demonstrate and apply knowledge of Mathematics, Science and engineering fundamentals in Electronics and Communication Engineering field
- b) To design a component, a system or a process to meet the specific needs within the realistic constraints such as economics, environment, ethics, health, safety and manufacturability
- c) To demonstrate the competency to use software tools for computation, simulation and testing of electronics and communication engineering circuits
- d) To identify, formulate and solve electronic and communication engineering problems
- e) To demonstrate an ability to visualize and work on laboratory and multidisciplinary tasks
- f) To function as a member or a leader in multidisciplinary activities
- g) To communicate in verbal and written form with fellow engineers and society at large
- h) To understand the impact of Electronics and Communication Engineering in the society and demonstrate awareness of contemporary issues and commitment to give solutions exhibiting social responsibility
- i) To demonstrate professional & ethical responsibilities
- j) To exhibit confidence in self-education and ability for lifelong learning
- k) To participate and succeed in competitive exams

## **EC6511 – DIGITAL SIGNAL PROCESSING LABORATORY SYLLABUS**

### **COURSE OBJECTIVES**

- To implement linear and circular convolution
- To implement FIR and IIR filters
- To study the architecture of DSP processor
- To demonstrate finite word length effect

#### **LIST OF EXPERIMENTS:**

1. Generation of sequences (functional & random) & correlation
2. Linear and circular convolutions
3. Spectrum analysis using DFT
4. FIR filter design
5. IIR filter design
6. Multirate filters
7. Equalization

#### **DSP PROCESSOR BASED IMPLEMENTATION**

1. Study of architecture of digital signal processor
2. MAC operation using various addressing modes
3. Linear convolution
4. Circular convolution
5. FFT Implementation
6. Waveform generation
7. IIR and FIR implementation
8. Finite word length effect

### **COURSE OUTCOME**

1. Carry out simulation of DSP systems
2. Demonstrate their abilities towards DSP processor based implementation of DSP system
3. Analyze finite word length effect on DSP systems
4. Demonstrate the applications of FFT to DSP

**EC6511 – DIGITAL SIGNAL PROCESSING LABORATORY****CONTENTS**

Sl.No.	Name of the Experiment	Page No.
--------	------------------------	----------

**CYCLE 1 – EXPERIMENTS**

1	Generation of signals	6
2	Linear and Circular Convolution of two sequences	11
3	Spectrum analysis using DFT	15
4	Design of FIR filters	17
5	Design of IIR filters	21
6	Multirate filters	24
7	Equalization	26

**CYCLE 2 – EXPERIMENTS**

8	Study of DSP processor TMS320C50	28
9	MAC operations using various addressing modes	33
10	Implementation of Linear Convolution	36
11	Implementation of Circular Convolution	38
12	Calculation of FFT	40
13	Waveform Generation	43
14	Implementation of IIR and FIR filters	48
15	Analysis of Finite Word Length Effects in fixed point DSP systems	50

**ADDITIONAL EXPERIMENTS BEYOND THE SYLLABUS**

16	Decimation by Polyphase Decomposition	53
17	Sampling of signals	55



**Expt.No.1****GENERATION OF SIGNALS****Aim:**

To write a program to generate the following sequences and to verify using MATLAB

1. Sine wave
2. Cosine wave
3. Ramp signal
4. Exponential signal
5. Impulse signal
6. Step signal
7. Saw tooth signal
8. Rectangular signal
9. Random sequence
10. Auto correlation
11. Cross correlation

**Apparatus required:**

PC with MATLAB software

**Theory:**

A signal is defined as any physical quantity that varies with time, space or any other independent variable or variables. Mathematically, we describe a signal as a function of one or more independent variables.

A system may be defined as a physical device that performs an operation on a signal. It can also be regarded as a process that transforms one signal into another. The input to a system and the output from a system are signals.

**Algorithm:****Sine Wave Generation:**

1. Start
2. Get the frequency and range of time 't'.
3. Determine the values of the sine function.
4. Plot the graph for obtained values in continuous and discrete forms.

5. Label the x and y axes.

6. Stop

#### Cosine Wave Generation:

1. Start

2. Get the frequency and range of time 't'

3. Determine the values of the cosine function.

4. Plot the graph for obtained values in continuous and discrete forms.

5. Label the x and y axes.

6. Stop

#### Ramp Signal Generation:

1. Start

2. Get the amplitude and range of time 't'.

3. Assign the ramp signal output equal to input time.

4. Plot the graph for the obtained values in continuous and discrete forms.

5. Label the x and y axes.

6. Stop

#### Exponential Signal Generation:

1. Start

2. Get the amplitude and ranges of time 't'.

3. Determine the values of the exponential function.

4. Plot the graph for the obtained values in continuous and discrete forms.

5. Label the x and y axes.

6. Stop

#### Unit Impulse Signal Generation:

1. Start

2. Get the ranges of time 't'.

3. Determine the values of the unit impulse signal.

4. Plot the graph for the obtained values in discrete form.

5. Label the x and y axes.

6. Stop



### Step Signal Generation:

1. Start
2. Get the amplitude and ranges of time 't'.
3. Determine the values of the unit step signal.
4. Multiply the unit step signal with given amplitude.
5. Plot the graph for the resultant values in continuous and discrete forms.
6. Label the x and y axes.
7. Stop

### Saw tooth Signal Generation:

1. Start
2. Get the ranges of time 't'.
3. Determine the values of the saw tooth signal.
4. Plot the graph for the obtained values in discrete form.
5. Label the x and y axes.
6. Stop

### Rectangular Signal Generation:

1. Start
2. Get the amplitude and ranges of time 't'.
3. Determine the values of the rectangular signal.
4. Multiply the unit step signal with given amplitude.
5. Plot the graph for the resultant values in continuous and discrete forms.
6. Label the x and y axes.
7. Stop

### Random Sequence:

1. Start
2. Get the length of input sequence.
3. Generate random signals of the specified length
4. Plot the graph for the resultant values in discrete form.
5. Label the x and y axes.
6. Stop

**Auto Correlation:**

1. Start
2. Get the length of input sequence.
3. Generate a rectangular signal of given length.
4. Calculate auto correlation of the rectangular signal.
5. Plot the graph for the resultant values in discrete forms.
6. Label the x and y axes.
7. Stop

**Cross Correlation:**

1. Start
2. Get the length of input sequence.
3. Generate sinusoidal and cosine waveforms of given length.
4. Calculate cross correlation of the two signals.
5. Plot the graph for the resultant values in discrete forms.
6. Label the x and y axes.
7. Stop

**Procedure:**

1. Click on the MATLAB Icon on the desktop.
2. Click on the 'FILE' Menu on menu bar.
3. Click on NEW M-File from the file Menu.
4. Save the file in directory.
5. Click on DEBUG from Menu bar and Click Run.
6. Open the command window\ Figure window for the output.

**Result:**

Thus the signal waveforms were analyzed for the given input.

**Outcome:**

After finishing this experiment the students are able to

1. Generate elementary signals/ waveforms.
2. Perform arithmetic operations on signals.

### Applications:

1. Aerospace and avionics
2. Defence applications
3. Wireless communication
4. Radar
5. GPS

### Viva-voce

1. Expand MATLAB
2. What is MATLAB? Where is MATLAB applicable?
3. What is MATLAB API?
4. What is M-file and MEX files in MATLAB?
5. List the operators of MATLAB.
6. Name the graphic system used in MATLAB.
7. What is disp() and input()?
8. What is the syntax to find the rank of the matrix?
9. What is the function of clear all and close all?
10. Define – Continuous time signal
11. Define – Discrete time signal
12. Define – Impulse function
13. Define – Unit step function
14. What is a ramp function?
15. Define – Delta function
16. What are the basic operations that can be performed over the signals?
17. List the types of discrete time signals.
18. List the types of continuous time signals.
19. What is an exponential function?
20. What are the types of program files that MATLAB allows to write?

## **Expt. No. 2      LINEAR AND CIRCULAR CONVOLUTION OF TWO SEQUENCES**

### **Aim:**

To write a program to perform the following convolutions and to verify using MATLAB

1. Linear convolution
2. Circular convolution
3. Linear convolution using circular convolution
4. Linear convolution using DFT
5. Circular convolution using DFT

### **Apparatus Required:**

PC with MATLAB software

### **Algorithm:**

#### **Linear Convolution:**

1. Start
2. Get the two input sequences.
3. Obtain the linear convolution of two sequences.
4. Calculate the lengths of input sequences.
5. Assign the length of resultant sequence(N) equal to length of first sequence (L) + length of second sequence (M) – 1 ( $N = L + M - 1$ ).
6. Plot the input sequences and output sequence in discrete form corresponding to their lengths.
7. Label the x and y axes.
8. Display the resultant sequence values.
9. Stop

#### **Circular Convolution:**

1. Start
2. Get the two input sequences.
3. Check the maximum length (N) of two sequences.
4. By zero padding, make both the two sequences with equal length (N).
5. Calculate circular convolution of equal length sequences.

6. Plot the input sequences and output sequence in discrete form corresponding to their lengths.
7. Label the x and y axes.
8. Display the resultant sequence values.
9. Stop

#### Linear Convolution using Circular Convolution:

1. Start
2. Get the two input sequences.
3. Calculate the lengths of input sequences.
4. Assign the length of resultant sequence(N) equal to length of first sequence(L)+ length of second sequence(M) – 1 ( $N = L + M - 1$ )
5. By zero padding, make both the two sequences with equal length (N).
6. Obtain the circular convolution of two sequences.
7. Plot the input sequences and output sequence in discrete form corresponding to their lengths.
8. Label the x and y axes.
9. Display the resultant sequence values.
10. Stop

#### Linear Convolution using DFT method:

1. Start.
2. Get the two input sequences.
3. Calculate the lengths of input sequences.
4. Assign the length of resultant sequence(N) equal to length of first sequence(L) +b length of second sequence(M) – 1 ( $N = L + M - 1$ )
5. By zero padding, make both the two sequences with equal length (N).
6. Calculate the N-point FFT of two sequences.
7. Multiply the two FFTs.
8. Calculate the N-point IFFT of the sequence.
9. Plot the input sequences and output sequences FFT and IFFT in discrete form corresponding to their lengths.
10. Label the x and y axes.
11. Display the resultant sequence values.
12. Stop.

### Circular Convolution using DFT method:

1. Start
2. Get the two input sequences.
3. Calculate the lengths of input sequences.
4. By zero padding, make both the two sequences with equal length (N).
5. Calculate the N-point FFT of two sequences.
6. Multiply the two FFTs.
7. Calculate the N-point IFFT of the sequence.
8. Plot the input sequences and output sequences FFT and IFFT in discrete form corresponding to their lengths.
9. Label the x and y axes.
10. Display the resultant sequence values.
11. Stop.

### Procedure:

1. Get two signals  $x(n)$  and  $h(n)$  in matrix form.
2. The convolute signal is denoted as  $y(n)$ .
3.  $y(n)$  is given by the formula  

$$y(n) = \sum x(k)h(n-k)$$
4.  $n = \text{length}(X)$  returns the size of the longest dimension of X. If X is a vector, this is the same as its length.
5.  $w = \text{conv}(u,v)$  convolves vectors u and v. Algebraically, convolution is the same operation as multiplying the polynomials whose coefficients are the elements of u and v. [subplot divides the current figure into rectangular panes that are numbered row wise]
6. Stop the program

### Result:

Thus the linear and circular convolution of the given two sequences was performed using MATLAB.

### Outcome:

After finishing this experiment the students are able to implement convolution of given sequences using linear, circular, linear using circular, linear convolution using DFT method, circular convolution using DFT method.



### Applications:

1. Finding the response of LTI systems
2. Spectral analysis
3. Study of system behavior
4. Frequency response of systems

**Viva-voce**

1. What is the difference between linear convolution and circular convolution?
2. List the methods of linear convolution.
3. List the steps to perform linear convolution using circular convolution.
4. Define – Convolution
5. What are the properties of convolution?
6. What is zero padding in convolution?
7. What is the difference between convolution & correlation?
8. What is half wave symmetry?
9. What is Dirichlet condition?
10. How to compute maximum length N for a circular convolution using DFT and IDFT.
11. What is sectioned convolution?
12. What is the advantage of sectioned convolution?
13. How to perform linear convolution using circular convolution?
14. What is the difference between linear shifting and circular shifting of signal?
15. What is the length of output sequence after circular convolution?
16. State the circular convolution property of DFT
17. What is circular convolution?
18. What is the length of output sequence after linear convolution?
19. List the properties of linear and circular convolution.
20. What is linear convolution?

**Expt.No.3****SPECTRUM ANALYSIS USING DFT****Aim:**

To write a program to analyze the spectrum of a signal using Discrete Fourier Transform (DFT) and Fast Fourier Transform (FFT) using MATLAB

**Apparatus Required:**

PC with MATLAB software

**Algorithm:**

1. Start
2. Get the input sequence.
3. Obtain the DFT of input sequence.
4. Find the magnitude and phase response of the DFT.
5. Plot the magnitude and phase in continuous and in discrete forms.
6. Stop

**Procedure:**

1. Enter the program in workspace.
2. Save and Run it in .m files.
3. Observe the output in command window or in figure window.

**Result:**

Thus the spectrum of a signal was analyzed using Discrete Fourier Transform (DFT).

**Outcome:**

After finishing this experiment the students are able to analyze the spectrum of the signal using Discrete Fourier Transform (DFT)

**Applications:**

1. Spectral analysis of biomedical signals, RF signals, speech and seismic signals
2. Implementation of filters using circuits
3. Correlation of sequences in RADAR

4. Digital communication systems for TV broadcasting

Viva-voce

1. What is the difference between DFT and IDFT?
2. Differentiate DIT from DIF algorithm.
3. What is decimation in frequency?
4. What is decimation in time?
5. How many multiplication terms are required to perform DFT by expressional method and FFT method?
6. How many stages are required for 8 point DFT?
7. Compute inverse DFT of  $X(k) = \{3, 4, 5, 6\}$ .
8. How to calculate output of DFT using MATLAB?
9. List the properties of FFT.
10. List the properties of DFT
11. What is the difference between FFT and IFFT?
12. What is "Bit reversal"?
13. List the advantages of FFT.
14. When the DFT  $X(k)$  of a sequence  $x(n)$  is imaginary?
15. What's the difference between FFT and DFT?
16. Define – Discrete Fourier series
17. When the DFT  $X(k)$  of a sequence  $x(n)$  is real?
18. What is the importance of FFT in OFDMA technology?
19. What is the advantage of Fourier transform in DSP?
20. What is twiddle factor?

**Expt.No.4****DESIGN OF FIR FILTERS****Aim:**

To write a program to design and obtain the magnitude and phase responses for the given FIR filter using MATLAB

**Apparatus Required:**

PC with MATLAB software

**Algorithm:****Low pass filter using rectangular window**

1. Get the pass band and stop band edge frequencies.
2. Get the sampling frequency
3. Get the order of the filter (N)
4. Find the window co-efficient using the MATLAB function boxcar (N)
5. Draw the magnitude and phase response.

**High pass filter using rectangular window**

1. Get the pass band and stop band edge frequencies.
2. Get the sampling frequency
3. Get the order of the filter (N)
4. Find the window co-efficient using the MATLAB function boxcar (N)
5. Draw the magnitude and phase response.

**Band pass filter using rectangular window**

1. Get the pass band and stop band edge frequencies.
2. Get the sampling frequency
3. Get the order of the filter (N)
4. Find the window co-efficient using the MATLAB function boxcar (N)
5. Draw the magnitude and phase response.

**Band stop filter using rectangular window**

1. Get the pass band and stop band edge frequencies.
2. Get the sampling frequency
3. Get the order of the filter (N)

4. Find the window co-efficient using the MATLAB function boxcar (N)
5. Draw the magnitude and phase response.

#### Low pass filter using hamming window

1. Get the pass band and stop band edge frequencies.
2. Get the sampling frequency
3. Get the order of the filter (N)
4. Find the window co-efficient using the MATLAB function hamming (N)
5. Draw the magnitude and phase response.

#### High pass filter using hamming window

1. Get the pass band and stop band edge frequencies.
2. Get the sampling frequency
3. Get the order of the filter (N)
4. Find the window co-efficient using the MATLAB function hamming (N)
5. Draw the magnitude and phase response.

#### Band pass filter using hamming window

1. Get the pass band and stop band edge frequencies.
2. Get the sampling frequency
3. Get the order of the filter (N)
4. Find the window co-efficient using the MATLAB function hamming (N)
5. Draw the magnitude and phase response.

#### Band stop filter using hamming window

1. Get the pass band and stop band edge frequencies.
2. Get the sampling frequency
3. Get the order of the filter (N)
4. Find the window co-efficient using the MATLAB function hamming (N)
5. Draw the magnitude and phase response.

#### Low pass filter using Hanning window

1. Get the pass band and stop band edge frequencies.
2. Get the sampling frequency
3. Get the order of the filter (N)
4. Find the window co-efficient using the MATLAB function Hanning (N)

5. Draw the magnitude and phase response.

#### High pass filter using Hanning window

1. Get the pass band and stop band edge frequencies.
2. Get the sampling frequency
3. Get the order of the filter (N)
4. Find the window co-efficient using the MATLAB function Hanning (N)
5. Draw the magnitude and phase response.

#### Band pass filter using Hanning window

1. Get the pass band and stop band edge frequencies.
2. Get the sampling frequency
3. Get the order of the filter (N)
4. Find the window co-efficient using the MATLAB function Hanning (N)
5. Draw the magnitude and phase response.

#### Band stop filter using Hanning window

1. Get the pass band and stop band edge frequencies.
2. Get the sampling frequency
3. Get the order of the filter (N)
4. Find the window co-efficient using the MATLAB function Hanning (N)
5. Draw the magnitude and phase response.

#### Procedure:

1. Enter the program in workspace.
2. Save and Run it in .m files.
3. Observe the output in figure window.

#### Result:

Thus the FIR filter was designed for the given specifications and its magnitude and phase responses were observed.

#### Outcome:

After finishing this experiment the students are able to:

1. Able to Implement Low pass, High pass and Band pass FIR filter for a given sequence
2. Calculate the filter coefficients.



### Applications:

1. Implementation of circuits for filtering operations
2. Study of frequency response of filters
3. Audio applications
4. Image processing and music processing

**Viva-voce**

1. What are the specifications required to design a filter?
2. List the design techniques for linear phase FIR filter design.
3. Distinguish between FIR and IIR filter.
4. What is window method? List the steps to design FIR filter using window method.
5. What is the MATLAB command for Hamming window?
6. What are the advantages of FIR Filters (compared to IIR filters)?
7. What are the disadvantages of FIR Filters (compared to IIR filters)?
8. What are the desirable characteristics of the window?
9. What are the difference types of filters based on impulse response?
10. What are the difference types of filters based on frequency response?
11. What are the desirable and undesirable features of FIR filters?
12. Write the expression for Hamming and Blackman windows?
13. What is the necessary and sufficient condition for linear phase characteristic in FIR filter?
14. Write the expression for Bartlett and Hamming windows?
15. What is the principle of designing FIR filter using frequency sampling method?
16. What is the need for employing window technique for FIR filter design?
17. What are the advantages of Kaiser window?
18. What is Gibbs phenomenon?
19. What are the specifications required to design filter?
20. Name the window function which has less peak amplitude of side lobe.

## Expt.No.5

## DESIGN OF IIR FILTERS

### Aim:

To write a program to design and obtain the response of the given IIR filter using MATLAB

### Apparatus Required:

PC with MATLAB software

### Algorithm:

A digital filter, the derivation of the digital filter transfer function requires three steps.

1. Map the desired digital filter specification into those for an equivalent analog filter.
2. Derive the analog transfer function for the analog prototype.

The various parameters in the figure are,

$\omega_p$  = Pass band frequency in radians

$\omega_s$  = Stop band frequency in radians

$\omega_c$  = 3-db cutoff frequency in radians

$\epsilon$  = Parameter specifying allowable pass band

$\lambda$  = Parameter specifying allowable stop band

### Design of IIR filters from analog filters:

There are several methods that can be used to design digital filters having an infinite duration unit sample response. The techniques described are all based on converting an analog filter into digital filter. If the conversion technique is to be effective, it should possess the following desirable properties.

The  $j\Omega$ -axes in the s-plane should map into the unit circle in the z-plane. Thus there will be a direct relationship between the two frequency variables in the two domains.

The left-half plane of the s-plane should map into inside of the unit circle in the z-plane. Thus a stable analog filter will be converted to a stable digital filter.

### Algorithm:

#### Butterworth low pass filter

1. Get the pass band and stop band ripples.
2. Get the pass band and stop band edge frequencies.
3. Get the sampling frequency.
4. Calculate the order the filter
5. Find the filter coefficients.
6. Draw the magnitude and phase response.

**Butterworth high pass filter**

1. Get the pass band and stop band ripples.
2. Get the pass band and stop band edge frequencies.
3. Get the sampling frequency.
4. Calculate the order the filter
5. Find the filter coefficients.
6. Draw the magnitude and phase response.

**Butterworth band pass filter**

1. Get the pass band and stop band ripples.
2. Get the pass band and stop band edge frequencies.
3. Get the sampling frequency.
4. Calculate the order the filter
5. Find the filter coefficients.
6. Draw the magnitude and phase response.

**Butterworth band stop filter**

1. Get the pass band and stop band ripples
2. Get the pass band and stop band edge frequencies
3. Get the sampling frequencies
4. Calculate the order of the filter
5. Find the filter co-efficient
6. Draw the magnitude and phase response

**Procedure:**

1. Enter the program in workspace.
2. Save and Run it in .m files.
3. Observe the output in figure window.

**Result:**

Thus the IIR filter was designed for the given specifications and its magnitude and phase responses were observed.

**Outcome:**

After finishing this experiment the students are able to:

1. Implement IIR filter for given specifications

2. Plot the response of the same.

### Applications:

1. Implementation of filter circuits
2. Speech processing and video processing
3. Radio transmission
4. Filters in various industrial applications

Viva-voce

1. Define – IIR Filter
2. List the steps to design a digital filter using impulse invariant method.
3. What are the disadvantages of impulse invariant method?
4. Differentiate recursive & non-recursive systems.
5. List the parameters obtained from the Chebyshev filter specifications.
6. What is warping effect? Where we found this effect? How can we eliminate warping effect
7. What is the difference between type I & type II Chebyshev filters?
8. Define – Cut-off frequency
9. Compare direct form I and direct form II realization of IIR filter.
10. List the digital techniques used to convert analog filter to digital filter.
11. List the properties of Butterworth low pass filter.
12. What is the role of delay element in filter design?
13. List the properties of Chebyshev low pass filter.
14. What are the properties of bilinear transformations?
15. Distinguish between the frequency responses of Chebyshev type I filter for N odd and N even.
16. Write the expression of the order of the Butterworth filter.
17. Define – Prewarping
18. List the steps to design digital filter from analog filter.
19. How many number of additions, multiplications and memory locations are required to realize a digital system in IIR filter?
20. Write the expression of the order of the Chebyshev filter.

**Expt.No.6****MULTIRATE FILTERS****Aim**

To design multirate FIR filters using interpolation and decimation and to plot the impulse responses and the magnitude responses for all designs.

**Apparatus Required:**

PC with MATLAB software

**Procedure:**

1. Enter the program in workspace.
2. Save and Run it in .m files.
3. Observe the output in command window or in figure window.

**Result:**

Thus the multirate filters was designed using interpolation and decimation for the given input.

**Outcome:**

After finishing this experiment the students are able to design linear-phase FIR multirate filters and do interpolation and decimation of the signals

**Applications:**

1. Sampling rate conversion systems
2. Communication systems
3. Design of phase shifters
4. Antenna arrays



Viva-voce
-----------

1. What is the importance of decimation for a given signal/sequence?
2. What is aliasing? What is the condition to avoid aliasing for sampling?
3. Define – Decimation
4. Define – Interpolation
5. What is multirate signal processing?
6. Name the process of converting a signal from a given rate to a different rate
7. State the formula for poly phase interpolation using z-transform.
8. List the applications of multirate signal processing.
9. List the methods used in sampling for converting a digital signal
10. List the advantages of multirate processing.
11. What is up sampling?
12. List the applications of multirate DSP.
13. What is the need for antialiasing filter prior to down sampling?
14. List the applications of adaptive filters
15. What are the sections in Quadrature mirror filter bank?
16. What is down sampling?
17. What is the need for anti-imaging filter after up sampling a signal?
18. State the formula for poly phase decimation using z-transform.
19. What is anti-aliasing filter?
20. Define – Sampling theorem



**Expt.No.7****EQUALIZATION****Aim**

To write MATLAB program to design adaptive channel equalization using LMS algorithm

**Apparatus Required:**

PC with MATLAB software

**Procedure:**

1. Enter the program in workspace.
2. Save and Run it in .m files.
3. Observe the output in command window or in figure window.

**Result:**

Thus the adaptive channel equalization was designed using LMS algorithm.

**Outcome:**

After finishing this experiment the students are able to design adaptive channel equalization using LMS algorithm.

**Applications:**

1. Echo cancellation in telecommunication systems
2. Digital radio communication systems
3. Channel equalization for dispersive channels
4. Signal processing and control applications

Viva-voce

1. Define – Equalization
2. What is recursive algorithm?
3. List the steps in the process of Adaptive equalization.
4. What is decision directed mode?
5. What are the uses of adaptive filters?
6. Define – Rate of Convergence
7. What is the relation between analog and digital filter?
8. Name the algorithm used in Adaptive Equalization.
9. What is the impact of LMS algorithm?
10. What is the relation between analog and digital filter?
11. What is the factor for designing an equalizer?
12. What is non-recursive algorithm?
13. What is the purpose of receiver decision?
14. What is difference between recursive and non-recursive algorithm?
15. What is the maximum expected time delay spread in the channel?

## **Expt. No. 8      STUDY OF DSP PROCESSOR TMS320C50**

### **Aim:**

To study the architecture overview of DSP processor TMS320C50

### **Introduction:**

It is needless to say that in order to utilize the full feature of the DSP chip TMS320C50, the DSP engineer must have a complete knowledge of the DSP device. This chapter is an introduction to the hardware aspects of the TMS320C50. The important units of TMS320C50 are discussed.

### **The DSP Chip TMS320C50:**

The TMS320C50 is a 16-bit fixed point digital signal processor that combines the flexibility of a high speed controller with the numerical capability of an array processor, thereby offering an inexpensive alternative to multichip bit-slice processors. The highly paralleled architecture and efficient instruction set, provide speed and flexibility capable of executing 10 MIPS (Million Instructions Per Second). The TMS320C50 optimizes speed by implementing functions in hardware that other processors implement through microcode or software. This hardware intensive approach provides the design engineer with processing power previously unavailable on a single chip.

The TMS320C50 is the third generation digital signal processor in the TMS320 family. Its powerful instruction set, inherent flexibility, high-speed number-crunching capabilities, and innovative architecture have made this high-performance, cost-effective processor the ideal solution to many telecommunications, computer, commercial, industrial, and military applications.

### **Key Features of TMS320C50:**

The key features of the Digital Signal Processor TMS320C50 are:

- 35-/50-ns single-cycle fixed-point instruction execution time (28.6/20 MIPS)
- Upward source-code compatible with all C1X and C2x devices
- RAM-based memory operation (C50)
- 9K x 16-bit single-cycle on-chip program/data RAM (C50)
- 2K x 16-bit single-cycle on-chip boot ROM (C50)
- 1056 x 16-bit dual-access on-chip data RAM
- 224K x 16-bit maximum addressable external memory space (64K program, 64K data, 64K I/O)

- 32-bit arithmetic logic unit (ALU), 32-bit accumulator (ACC), and 32-bit accumulator buffer (ACCB)
- 16-bit parallel logic unit (PLU)
- 16 x 16-bit parallel multiplier with a 32-bit product capability.
- Single-cycle multiply/accumulate instructions
- Eight auxiliary registers with a dedicated auxiliary register arithmetic unit for indirect addressing.
- Eleven context-switch registers (shadow registers) for storing strategic CPU controlled registers during an interrupt service routine
- Eight-level hardware stack
- 0- to 16-bit left and right data barrel-shifters and a 64-bit incremental data shifter
- Two indirectly addressed circular buffers for circular addressing
- Single-instruction repeat and block repeat operations for program code
- Block memory move instructions for better program/data management
- Full-duplex synchronous serial port for direct communication between the C5x and another serial device
- Time-division multiple-access (TDM) serial port
- Interval timer with period, control, and counter registers for software stop, start, and reset
- 64K parallel I/O ports, 16 of which are memory mapped
- Sixteen software programmable wait-state generators for program, data, and I/O memory spaces.

## Theory:

A detailed architectural block diagram of TMS320C50 is illustrated in Figure. The TMS320C50 utilizes a modified Harvard architecture for speed and flexibility. In a strict Harvard architecture, program and data memory are in two separate spaces, permitting a full overlap of instruction fetch and execution. The TMS320 family's modification of the Harvard architecture allows transfer between program and data spaces, thereby increasing the flexibility of the device. This modification permits coefficients stored in program memory to be read into the data RAM, eliminating the need for a separate coefficient ROM. It also makes available immediate instructions and subroutines based on computed values.

## 32-bit Accumulator:

The TMS320C50 contains a 32-bit ALU and accumulator for support of double-precision, two's complement arithmetic. The ALU is a general purpose arithmetic unit that operates on 16-bit words taken from the data RAM or derived from immediate instructions. In addition to the usual arithmetic instructions, the ALU can perform Boolean operations, providing the bit manipulation ability required of a high-speed controller. The accumulator stores the output from the ALU and is often an input to the ALU. Its word length is 32-bit. The accumulator is divided into a high order word (bits 31 through 16) and a low-order word (bits 15 through 0). Instructions are provided for storing and loading the high and lower order accumulator words to memory.

## 16 X 16-bit Parallel Multiplier:

The multiplier performs a 16 x 16-bit two's complement multiplication with a 32-bit result in a single instruction cycle. The multiplier consists of three units: the T-Register, P-Register, and multiplier array. The 16-bit T-Register temporarily stores the multiplicand and the P-Register stores the 32-bit product. Multiplier values either come from the data memory or are derived immediately from the MPY (multiply immediate) instruction word. The fast on-chip multiplier allows the device to perform fundamental operations such as convolution, correlation, and filtering. Two multiply/accumulate instructions in the instruction set fully utilize the computational bandwidth of the multiplier, allowing both operands to be processed simultaneously.

## Shifters:

A 16-bit scaling shifter is available at the accumulator input. This shifter produces a left shift of 0 to 16-bits on the input data to accumulator. TMS320C50 also contains a shifter at the accumulator output. This shifter provides a left shift of 0 to 7, on the data from either the ACCH or ACCL register. In addition



one shifter at the output of P- register, can shift the product by 1 or 4-bits left or 6-bits right, before transferring the product to accumulator.

### Interrupts and Subroutines:

The TMS320C50 has three external maskable user interrupts available for external devices that interrupt the processor. The TMS320C50 contains a eight-level hardware stack for saving the contents of the program counter during interrupts and subroutine calls. Instructions are available for saving the device's complete context. PUSH and POP instructions permit a level of nesting restricted only by the amount of available RAM.

### Serial Port:

A full-duplex on-chip serial port provides direct communication with serial devices such as codec's, serial A/D converters and other serial systems. The interface signals are compatible with codec's and many others serial devices with a minimum of external hardware.

### Input and Output:

The 16-bit parallel data bus can be utilized to perform I/O functions in two cycles. The I/O ports are addressed by the four LSBs on the address lines, allowing 16 input and 16 output ports. In addition, polling input for bit test and jump operations (BIO) and three interrupt pins (INT0 - INT2) have been incorporated for multitasking.

### Result:

Thus the architecture overview of DSP processor TMS320C50 was studied.

### Outcome:

After finishing this experiment the students are able to know the architecture of DSP processor TMS320C50.

### Applications:

1. Audio signal processing
2. Speech recognition
3. Biomedicine
4. Seismology





5. RADAR
6. Digital communication

Viva-voce

1. List the types of DSP processors.
2. List the features of TMS 320C6X processor.
3. Differentiate Von Neumann architecture and Harvard architecture.
4. Name the architecture used in DSP processor
5. How many instructions can be executed per cycle in TMS320C6X DSP processor?
6. Name the register that stores the result of multiplication in DSP processor.
7. How many instruction cycles are used for 16 x 16 multiplication in TMS320C6X DSP processor?
8. What are the applications for the TMS320 DSP's?
9. Name the software tool required to compile and run the DSP assembly program.
10. What is the functional unit adapted for transferring the data from the register in TMS 320 C6x processor?
11. What are the difference buses of TMS320C5X and their functions?
12. What are the different stages in pipelining?
13. What is pipeline depth of TMS320C50 and TMS320C54x?
14. What is the function of parallel logic unit?
15. How many functional units are in TMS 320C6X DSP processor?
16. What are the advantages and disadvantages of VLIW architecture?
17. What is pipeline depth?
18. List the on chip peripheral in C5x.
19. What is the logical instruction of c5x?
20. What are the factors that influence selection of DSP?



## **Expt.No.9      MAC OPERATIONS USING VARIOUS ADDRESSING MODES**

### **Aim:**

To study the various addressing modes of TMS320C50 processor using the following programs:

1. Addition of two data
2. Subtraction of two data
3. Multiplication of two data

### **Apparatus Required:**

1. Digital signal processor - TMS320C50 Trainer kit
2. PC with C50 debugger

### **Algorithm:**

#### **Addition of two data:**

1. Start
2. Initialize the memory mapped registers.
3. Load the data pointer with address 100H.
4. Load the first data in accumulator.
5. Add the second data with accumulator.
6. Store the result in memory.
7. Check the result in various addressing modes.
8. Stop

#### **Subtraction of two data:**

1. Start
2. Initialize the memory mapped registers.
3. Load the data pointer with address 100H.
4. Load the first data in accumulator.
5. Subtract the second data from the accumulator.
6. Store the result in memory.
7. Check the result in various addressing modes.
8. Stop

**Multiplication of two data:**

1. Start.
2. Initialize the memory mapped registers.
3. Load the data pointer with address 100H.
4. Load the first data in accumulator.
5. Multiply the second data with accumulator.
6. Store the result in memory.
7. Check the result in various addressing modes.
8. Stop

**Procedure:**

1. Enter the program in work space.
2. Save and compile the program to generate ASCII codes.
3. Download the ASCII codes to micro 50 trainer kit.
4. Execute the program.
5. Observe the output in memory.

**Result:**

Thus the various addressing modes of TMS320C50 are studied and the outputs are verified.

**Outcome:**

After finishing this experiment the students are able to perform arithmetic operations using various addressing modes in TMS320C50 Processor.

**Applications:**

1. Implementation of digital filters for speech and music processing
2. Correlation of signals in RADAR
3. Computing the output of LTI systems for various input sequences
4. Study of system behaviour



Viva-voce

1. List the addressing modes supported by C5X processors.
2. List the on-chip peripherals in C5X.
3. What are the features of TMS320C50 processor?
4. What are the different stages in pipelining?
5. What is parallel processing?
6. What is pipelining?
7. What are the classification digital signal processors?
8. What is MAC?
9. What is barrel shifter? List its advantages in DSP processors
10. What is digital signal processor?
11. What is the expansion of MAC in TMS320C50?
12. What is the purpose of shadow registers in TMS320C50 processor?
13. Differentiate between RISC and CISC architectures.
14. What is the general purpose of I/O pins?
15. List the different quantization methods



## **Expt. No. 10 IMPLEMENTATION OF LINEAR CONVOLUTION**

### **Aim:**

To write a program to find the Linear Convolution of two sequences using TMS320C50 processor

### **Apparatus Required:**

1. System with TMS 320C50 debugger software
2. TMS 320C50 Kit.
3. RS232 cable

### **Algorithm:**

#### **Linear Convolution**

1. Start
2. Initialize the memory mapped registers.
3. Load the data pointer with address 100H and temporary register with length (time period).
4. Load the input sequence into one auxiliary register.
5. Load the impulse sequence into another auxiliary register.
6. Perform linear convolution.
7. Store the result in output memory location.
8. Stop

### **Procedure:**

1. Enter the program in work space.
2. Save and compile the program to generate ASCII codes.
3. Download the ASCII codes to micro 50 trainer kit.
4. Execute the program.
5. Observe the output in memory.

### **Result:**

Thus the linear convolution of two sequences is performed using TMS320C50 processor.

### **Outcome:**

After finishing this experiment the students are able to implement convolution of given sequences using linear using TMS320C50 Processor.

### Applications:

1. Implementation of FIR filters
2. Audio signal processing
3. Digital communication systems

**Viva-voce**

1. List the methods of linear convolution.
2. What is the difference between linear convolution and circular convolution?
3. List the steps to perform linear convolution using circular convolution.
4. Define – Convolution
5. What are the properties of convolution?
6. What is zero padding in convolution?
7. What is the difference between convolution & correlation?
8. What is half wave symmetry?
9. What is Dirichlet condition?
10. How to compute maximum length N for a circular convolution using DFT and IDFT.
11. What is sectioned convolution?
12. What is the advantage of sectioned convolution?
13. How to perform linear convolution using circular convolution?
14. What is the difference between linear shifting and circular shifting of signal?
15. What is the length of output sequence after circular convolution?
16. State the circular convolution property of DFT
17. What is circular convolution?
18. What is the length of output sequence after linear convolution?
19. List the properties of linear and circular convolution.
20. What is linear convolution?



## **Expt.No.11                      IMPLEMENTATION OF CIRCULAR CONVOLUTION**

### **Aim:**

To write a program to find the Circular Convolution of two sequences using TMS320C50 processor

### **Apparatus Required:**

1. System with TMS 320C50 debugger software
2. TMS 320C50 Kit.
3. RS232 cable

### **Algorithm:**

#### **Circular Convolution**

1. Start
2. Initialize the memory mapped registers.
3. Load the data pointer with address 100H and temporary register with length (time period).
4. Load the input sequences into one auxiliary register.
5. Perform circular convolution.
6. Store the result in output memory location.
7. Stop

### **Procedure:**

1. Enter the program in work space.
2. Save and compile the program to generate ASCII codes.
3. Download the ASCII codes to micro 50 trainer kit
4. Execute the program.
5. Observe the output in memory.

### **Result:**

Thus the circular convolution of two sequences is performed using TMS320C50 Processor.

### **Outcome:**

After finishing this experiment the students are able to implement convolution of given sequences using circular using TMS320C50 Processor.

### Applications:

1. Image processing
2. Radio therapy planning treatment systems
3. Computing response of LTI systems in electrical engineering
4. Fluorescence spectroscopy

**Viva-voce**

1. What is the difference between linear convolution and circular convolution?
2. List the methods of linear convolution.
3. List the steps to perform linear convolution using circular convolution.
4. Define – Convolution
5. What are the properties of convolution?
6. What is zero padding in convolution?
7. What is the difference between convolution & correlation?
8. What is half wave symmetry?
9. What is Dirichlet condition?
10. How to compute maximum length  $N$  for a circular convolution using DFT and IDFT.
11. What is sectioned convolution?
12. What is the advantage of sectioned convolution?
13. How to perform linear convolution using circular convolution?
14. What is the difference between linear shifting and circular shifting of signal?
15. What is the length of output sequence after circular convolution?
16. State the circular convolution property of DFT
17. What is circular convolution?
18. What is the length of output sequence after linear convolution?
19. List the properties of linear and circular convolution.
20. What is linear convolution?

**Expt.No.12****CALCULATION OF FFT**

Aim:

To write a program to calculate the FFT of the input sequence using TMS320C50 Processor

Apparatus Required:

1. Digital signal processor - TMS320C50 Trainer kit
2. CRO
3. PC with C50 debugger.

Theory:

The fast fourier transform (FFT) is an algorithm used to compute the DFT with minimum number of calculations. It makes use of the symmetry and periodicity properties of twiddle factor  $W_N^k$  to effectively reduce the DFT computation time. It is based on the fundamental principal of decomposing the computation of DFT of a sequence of length 'N' into successively smaller discrete Fourier transforms.

The FFT algorithm provides speed-increase factors, when compared with direct computation of the DFT. The direct evaluation of DFT requires  $N^2$  complex multiplications and  $N(N-1)$  complex additions. Thus for reasonably large values of N (in the order of 1000) direct evaluation of the DFT requires an inordinate amount of computation. By using FFT algorithms the number of computations can be reduced. For example, for an N-point DFT, the number of complex multiplications required using FFT is  $N/2 \log_2 N$ . If  $N=16$ , the number of complex multiplications required for direct evaluation of DFT is 256, whereas using FFT only 32 multiplications are required.

Two Types of FFT:

**Decimation in time (DIT) Algorithm:**

Decimation in-time algorithm is used to calculate the DFT of a N-point sequence. The idea is to break the N-point sequence into two sequences, the DFTs of which can be combined to give the DFT of the original N-point sequence. Initially the N-point sequence is divided into two  $N/2$  -point sequences  $x_e(n)$  and  $x_o(n)$ , which have the even and odd members of  $x(n)$  respectively. The  $N/2$  -point DFTs of these two sequences are evaluated and combined to give the N-point DFT. Similarly the  $N/2$  -point DFTs can be expressed as a combination of  $N/4$  -point DFTs. This process is continued until we left with 2-point DFT. This algorithm is called decimation-in-time because the sequence  $x(n)$  is often splitted into smaller subsequences.

### Decimation in frequency (DIF) algorithm:

It is a popular form of the FFT algorithm. In this the output sequence  $x(k)$  is divided into smaller and smaller subsequences, that is why the name decimation in frequency. Initially the input sequence  $x(n)$  is divided into two sequences  $x_1(n)$  and  $x_2(n)$  consisting of the first  $N/2$  – samples of  $x(n)$  and the last  $N/2$  – samples of  $x(n)$  respectively. Then we find the  $N/2$  – point sequences  $f(n)$  and  $g(n)$  as

$$f(n) = x_1(n) + x_2(n) \text{ and } g(n) = [x_1(n) - x_2(n)] W_N^N$$

The  $N/2$  – point DFTs of the above two sequences give even numbered and odd numbered output samples respectively. The above procedure can now be iterated to express each  $N/2$  – point DFT as a combination of two  $N/4$  – point DFTs. This process is continued until we left with 2-point DFT.

### Algorithm:

1. Start
2. Initialize the memory mapped registers and the sampling rate.
3. Load the data pointer with address 100H
4. Decompose an  $N$  point time domain signal into  $N$  time domain signals each consists of a single point.
5. Calculate the  $N$  frequency spectra corresponding to the  $N$  time domain signals.
6. Synthesize the  $N$  frequency spectra into single frequency spectrum.
7. Stop

### Procedure:

1. Enter the program in work space.
2. Save and compile the program to generate ASCII codes.
3. Download the ASCII codes to micro 50 trainer kit.
4. Execute the program.
5. Observe the output in memory.

### Result:

Thus the program was executed to calculate FFT of the input sequence and its output was verified.

### Outcome:

After finishing this experiment the students are able to calculate FFT of given two input sequences using TMS320C50 Processor.

### Applications:

1. Spectral analysis of signals in radio transmission
2. Study of frequency response of systems
3. Correlation using RADAR
4. Modeling of real time systems

**Viva-voce**

1. Define – Twiddle factor
2. How many number of multiplications needed in the calculation of FFT algorithm for 64-point sequence?
3. What is the sequence length of radix-2 FFT algorithm?
4. How many numbers of multiplication and addition are required in FFT algorithm?
5. What is the speed improvement factor in calculating 64-point using direct computation and FFT algorithm?
6. If the sequence  $x_1[n]$  and  $x_2[n]$  has  $N_1$  and  $N_2$  samples. What will be the convolved signal  $x_3[n]$ ?
7. Define Fourier series?
8. What are the properties of Continuous-Time Fourier Series?
9. What is the sufficient condition for the existence of F.T?
10. Define the Fourier transform of a signal?
11. What is the difference b/w Fourier Transform and Fourier series?
12. FFT is in complex domain how to use it in real life signals optimally?
13. What is the difference between FFT and IFFT?
14. Explain using convolution the effects of taking an FFT of a sample with no windowing
15. What is the need of FFT?
16. What's the difference between FFT and DFT?
17. Why do we need Fourier transform in DSP?
18. What is the importance of FFT in OFDMA technology?
19. In STB DVB how many point FFT is currently using?
20. What is "bit reversal"?
21. How does the FFT work?



**Expt.No.13****WAVEFORM GENERATION****Aim:**

To generate the following waveforms using TMS320C50 Processor:

1. Square wave generation
2. Saw tooth wave generation
3. Triangular wave generation

**Apparatus Required:**

1. Digital signal processor - TMS320C50 Trainer kit
2. CRO
3. PC with C50 software.

**Theory:**

The addressing modes in TMS320C50 are

1. Immediate addressing
2. Indirect addressing
3. Register addressing
4. Memory mapped register addressing
5. Direct addressing
6. Circular addressing modes.

**Immediate addressing modes:**

The immediate addressing mode is used to handle constant data. It allows the program to operate an actual value the data can be either a 16 bit constant or constant length T, 9 or 13 depending on the length of data the addressing. In long addressing mode the data contained in a portion of a bit in a single word instruction. At the assembly code level the developer uses a prefix to specify immediate addressing as en: LD #80HA; The instruction load an immediate value 80H into the accumulator.

ADD #K, Store immediate

ADD #LK, Long immediate

Long immediate – may # k for multiplication

add # LK [for shift]

OR #LK [for shift]



XOR #LK [for shift]

### Indirect addressing mode:

The addressing modes use the auxiliary register to hold the address of operand in memory. In indirect addressing any location in the 64-k word data memory space can be accessed using a 16-bit address contained in accumulator. There are 7 types of indirect addressing modes.

1. Auto increment
2. auto decrement port inducing by adding the content AR0
3. port inducing by subtracting the content AR0
4. single indirect addressing with no increment
5. Single indirect addressing with no decrements
6. Bit reversed addressing

EX: ADD {find} {shift} [ARM] SUB {shift} [ARM].

### Register addressing:

The register addressing mode uses operands in CPU registered either explicitly such as with a direct reference to a specific register or simplicity with instruction that reference contain register that if it is an BLPD- block move from the program to data memory with source address in BMAR.

### Memory mapped register addressing:

Memory mapped register addressing is used to access efficiently the cpu on chip peripheral register. It operates like the direct addressing except that the upper bit of the address that is accessed is assumed to be 0.05. This allows the addressing and operand can be represented using a single 16-bit word.

LAMM – load accumulator with memory mapped register.

SAMM – store accumulator in memory mapped register.

### Direct addressing mode:

Direct addressing mode allowed the CPU to operate as an operand by specifying an offer from a base address defined in data pointer. In their mode the address of the operand is obtained by concatenating the 7 bit data memory address with the bit of the data page pointer 16- bit data memory address is placed or an internal direct data memory address.

### Circular addressing mode:

Circular addressing is the CSX addressing mode many algorithm. Such as convolution and the FIR filtering can use circular. Buffer in memory to event data to be processed.

CBSR-1 – circular buffer 1 start register

CBSR-2 – circular buffer 2 start register

CBER-1 - circular buffer 1 end register.

CBER-2 - circular buffer 2 end register.

The 8 bit CBER enables A disables circular buffer operation to define circular first use load the start A and addressing into the corresponding buffer register.

**Algorithm:**

**Square wave generation:**

1. Start
2. Initialize the memory mapped registers.
3. Load the data pointer with address 100H and temporary register with length (time period).
4. Load the accumulator with zero.
5. OUT the accumulator value through DAC port for the given length.
6. Complement the accumulator.
7. OUT the accumulator value through DAC port for the given length.
8. Jump unconditionally to step 4.
9. Stop

**Saw tooth Wave Generation:**

1. Start
2. Initialize the memory mapped registers.
3. Load the data pointer with address 100H and temporary register with length (time period).
4. Load the accumulator with zero.
5. Add one with the accumulator.
6. OUT the accumulator value through DAC port.
7. Decrement the length. If it is not equal to zero repeat the steps from 5.
8. Jump unconditionally to step 4.
9. Stop

**Triangular Wave generation:**

1. Start
2. Initialize the memory mapped registers.
3. Load the data pointer with address 100H and temporary register with length.
4. Load the accumulator with zero.

5. Add one with the accumulator.
6. OUT the accumulator value through DAC port.
7. Decrement the length. If it is not equal to zero repeat the steps from 5.
8. Load the length in temporary register.
9. Subtract amplitude from accumulator.
10. OUT the accumulator value through DAC port.
11. Decrement the length. If it is not equal to zero repeat the steps from 9.
12. Jump unconditionally to step 4.
13. Stop

**Procedure:**

1. Enter the program in work space.
2. Save and compile the program to generate ASCII codes.
3. Download the ASCII codes to micro 50 trainer kit.
4. Execute the program.
5. Observe the output in memory/CRO.

**Result:**

Thus the waveforms were generated using TMS320C50 Processor.

**Outcome:**

After finishing this experiment the students are able to generate different waveforms using TMS320C50 Processor.

**Applications:**

1. Communication systems
2. Control systems
3. Study of system behavior
4. Designing, testing and troubleshooting signal processing devices
5. Microwave communication

Viva-voce

1. What is MATLAB API?
2. List the operators of MATLAB.
3. Name the graphic system used in MATLAB.
4. What is the syntax to find the rank of the matrix?
5. Define – Continuous time signal
6. Define – Discrete time signal
7. Define – Impulse function
8. Define – Unit step function
9. What is a ramp function?
10. Define – Delta function

## **Expt.No.14                      IMPLEMENTATION OF IIR AND FIR FILTERS**

### **Aim:**

To implement the IIR and FIR filters and to obtain the frequency and log magnitude responses using TMS320C50 Processor.

### **Apparatus Required:**

1. Digital signal processor - TMS320C50 Trainer kit
2. CRO
3. A PC with C50 software.

### **Algorithm:**

1. Start
2. Initialize the memory mapped registers.
3. Load the data pointer with address 100H and temporary register with length.
4. Select the filter approximation type.
5. Select the filter type and order.
6. Select the Sampling rate and cut-off frequencies.
7. Generate the filter co-efficient.
8. Find magnitude and phase angle values.
9. View the output magnitude response and phase angle response in output figures.
10. Stop

### **Procedure:**

1. Enter the program in work space.
2. Save and compile the program to generate ASCII codes.
3. Download the ASCII codes to micro 50 trainer kit.
4. Execute the program.
5. Observe the output in CRO.

### **Result:**

Thus the IIR and FIR filters were implemented using TMS320C50 Processor.

### **Outcome:**

After finishing this experiment the students are able to:

1. Implement Low pass, High pass and Band pass FIR and IIR filter for a given sequence
2. Calculate the filter coefficients.

### Applications:

1. Telecommunication systems
2. Control systems
3. Audio signal processing
4. Computer graphics
5. Music synthesis

Viva-voce

1. Define – IIR Filter
2. List the steps to design a digital filter using impulse invariant method.
3. What are the disadvantages of impulse invariant method?
4. Differentiate recursive & non-recursive systems.
5. What is warping effect? Where we found this effect? How can we eliminate warping effect
6. What is the difference between type I & type II Chebyshev filters?
7. Define – Cut-off frequency
8. What is the role of delay element in filter design?
9. Write the expression of the order of the Butterworth filter?
10. Write the expression of the order of the Chebyshev filter?
11. What are the specifications required to design a filter?
12. List the design techniques for linear phase FIR filter design.
13. Distinguish between FIR and IIR filter.
14. What is window method? List the steps to design FIR filter using window method.
15. What is the MATLAB command for Hamming window?
16. What are the advantages of FIR Filters (compared to IIR filters)?
17. What are the disadvantages of FIR Filters (compared to IIR filters)?
18. What are the desirable characteristics of the window?
19. What are the specifications required to design filter?
20. Name the window function which has less peak amplitude of side lobe



## **Expt. No. 15 ANALYSIS OF FINITE WORD LENGTH EFFECT IN FIXED POINT DSP SYSTEMS**

### **Aim:**

To study the functions of finite word length effect in fixed point DSP systems

### **Apparatus Required:**

1. Digital signal processor - TMS320C50 Trainer kit
2. CRO
3. A PC with C50 software

### **Procedure:**

1. Find the ADC Noise Gain.
2. Determine the coefficient quantization error analysis
3. Plot the frequency response with quantization noise for desired word length
4. To compute and plot the frequency response with quantized coefficients
5. 'Frequency Response for Desired Word length
6. Analysis should be performed for Product Round off noise, Scaling.

### **Result:**

Thus the function of finite word length effect in fixed point DSP processor is verified.

### **Outcome:**

After finishing this experiment the students are able to analyze and obtain the coefficient error and quantization noise for desired word length.

### **Applications:**

1. DSP integrated circuits
2. Filters built using DS processors
3. Digital signal processors
4. RADAR

Viva-voce

1. Define – Dead band
2. What is overflow oscillation?
3. What is meant by quantization step size?
4. List the advantages of floating point representation.
5. Give the different quantization errors in digital filters.
6. What is input quantization error?
7. Define – Product Quantization Error
8. Define – Truncation
9. Define – Rounding
10. Define – Overflow Oscillations

**ADDITIONAL EXPERIMENTS BEYOND THE SYLLABUS**

## **Expt.No.16      DECIMATION BY POLYPHASE DECOMPOSITION**

### **Aim:**

To write a program to compute Convolution and m-fold decimation by polyphase decomposition

### **Apparatus Required:**

PC with MATLAB Software package

### **Theory:**

Decimation is the process of reducing the sampling frequency of a signal to a lower sampling frequency that differs from the original frequency by an integer value. Decimation also is known as down-sampling. The low pass filtering associated with decimation removes high-frequency content from the signal to accommodate the new sampling frequency.

Decimation is useful in applications in which the Nyquist frequency of a signal is much higher than the highest frequency of the signal. Decimation filters help you remove the excess bandwidth and reduce the sampling frequency of the signal. Decimation filters also help you reduce the computational resources required for processing and storing the signal. During the analog-to-digital (A/D) conversion process, decimation filters also can reduce the variance of quantization noise in a signal and maintain the signal power, thus improving the signal-to-noise ratio (SNR).

This filter contains a low pass FIR filter  $H(z)$ . This low pass FIR filter is an anti-aliasing filter followed by an M-fold decimator. The decimator passes every Mth sample and discards the other samples. After this operation, the decimation filter changes the sampling frequency  $f_s$  of the input signal  $x(n)$  to a new sampling frequency  $f_s/M$ . The decimation filter then returns an output signal  $y(n)$  with the new sampling frequency.

### **Algorithm:**

1. Get the input sequence.
2. Get the filter coefficients and also the decimation factor.
3. Find the response by using convolution.
4. Plot the graph.

**Result:**

Thus the program to compute Convolution and m-fold decimation by Polyphase decomposition has been computed successfully.

**Outcome:**

After finishing this experiment the students are able to compute Convolution and m-fold decimation using poly phase decomposition.

**Applications:**

1. Image processing
2. Speech processing
3. Video processing

**Viva-voce**

1. What are the classification digital signal processors?
2. What are the factors that influence selection of DSPs?
3. Define – Sampling
4. What are the applications of sampling?
5. Define – Twiddle Factor
6. How many number of multiplications needed in the calculation of FFT algorithm for 64-point sequence?
7. What is the sequence length of radix-2 FFT algorithm?
8. What is meant by quantization step size?
9. What are the two kinds of limit cycle behavior in DSP?
10. Give any practical application of FFT in daily life?

**Expt.No.17****SAMPLING OF SIGNALS****Aim:**

To write a program to convert analog signals into digital signals using TMS320C50 debugger

**Apparatus Required:**

1. System with TMS 320C50 debugger software
2. TMS 320C50 Kit.
3. CRO
4. Function Generator

**Theory:****Sampling Theorem:**

The theorem shows that a band limited analog signal that has been sampled can be perfectly reconstructed from an infinite sequence of samples if the sampling rate exceeds  $2B$  samples per second, where  $B$  is the highest frequency in the original signal.

**Aliasing:**

Aliasing is a phenomenon where the high frequency components of the sampled signal interfere with each other because of inadequate sampling  $\omega_s < 2\omega_m$ . Aliasing leads to distortion in recovered signal. This is the reason why sampling frequency should be at least twice the bandwidth of the signal.

**Algorithm:**

1. Initialize data pointer with 100H data
2. Give the analog signal as input
3. Introduce the time delay as per required.
4. Observe the discrete signal as output
5. Plot the graph.

**Procedure:**

1. Open MATLAB
2. Open new M-file and type the program
3. Save in current directory
4. Compile and Run the program



5. For the output see command window\ Figure window

### Result:

Thus the sampling was observed for the given analog signal using TMS320C50 processor.

### Outcome:

After finishing this experiment the students are able to sample the analog signals using TMS320C50 processor.

### Applications:

1. ADCs used in audio signal processing
2. Telephones
3. CD and DVD players
4. HDTV, SDTV
5. Camcorder

### Viva-voce

1. What are the properties that are maintained same in the transformation of analog to digital filter?
2. What are the requirements for an analog filter to be stable and causal?
3. What are the requirements for a digital filter to be stable and causal?
4. Define ripples in a filter
5. How a digital IIR filter is designed?
6. Mention any two techniques for digitizing the transfer function of an analog filter.
7. Define – Impulse Function
8. Define – Unit Step Function
9. What is a ramp function?
10. What is an exponential function?