
INDIAN INSTITUTE OF INFORMATION TECHNOLOGY,
DESIGN AND MANUFACTURING,
KANCHEEPURAM

# DSP SYSTEM DESIGN PRACTICE LAB 3 REPORT

**K.Nithesh**
**ESD19I008**

## AIM:

To Perform Sampling on MicroDSP6748 DSP Board.

## Apparatus Required:

Code Composer Studio, TMS320C6748 DSP board, XDS100v3 Emulator.

## Theory:

Sampling in signal processing is the process of converting a continuous signal into a discrete signal by taking regular intervals of the continuous signal. This process is essential because it enables signal processing algorithms to be applied to the signal, and it also simplifies the signal's representation and processing.

The Nyquist-Shannon sampling theorem states that a continuous signal can be reconstructed perfectly from its discrete samples if the sampling rate is at least twice the highest frequency of

the continuous signal. This theorem is essential because it ensures that the sampling process does not lose any critical information that may be essential for processing the signal.

There are various types of sampling methods in signal processing, such as uniform and non-uniform sampling, undersampling, oversampling, and random sampling. Each of these methods has its advantages and disadvantages and is chosen based on the specific needs of the signal processing task.

Overall, sampling is a crucial process in signal processing as it enables continuous signals to be processed and analyzed with digital signal processing algorithms.

## Procedure:

1. Connect the MicroDSP6748 DSP Board to the computer using a USB cable.
2. Open Code Composer Studio (CCS) software on the computer.
3. Create a new project in CCS for the MicroDSP6748 DSP Board.
4. Write the code for sampling in C language.
5. Connect the input signal source to the MicroDSP6748 DSP Board. This can be a microphone or any other audio source.
6. Configure the sampling rate and sampling depth in the code. These parameters will depend on the specific requirements of the project.
7. Compile and build the code.
8. Download the code to the MicroDSP6748 DSP Board.
9. Run the code on the MicroDSP6748 DSP Board.
10. Monitor the output of the sampling process on a computer screen or through other means, depending on the specific project requirements.
11. Analyze the sampled data and use it for further processing or analysis, depending on the specific project requirements

## Code:

Write a C code to interface the Audio Codec AIC3106 to sample the input channel and perform under-sampling, critical sampling and oversampling operations by varying the sampling frequency. Make a note of the sampling frequency values at which the above conditions first occur.

```
#include "L138_LCDK_aic3106_init.h"

interrupt void interrupt4(void)  // interrupt service
routine
{
  uint32_t sample;

  sample = input_sample(); // read L + R samples from ADC
  output_sample(sample);   // write L + R samples to DAC
  return;
}

int main(void)
{

L138_initialise_intr(FS_48000_HZ,ADC_GAIN_0DB,DAC_ATTEN_0DB,
LCDK_LINE_INPUT);
  while(1);
}
```

## Inference:

- For Sampling frequency 48KHz I can hear the audio clearly with good sound  So I consider it as the perfect sampling frequency .
- For Sampling frequency 16KHz I can hear the low audio with a lot of distortions.So I can do this as Under Sampling.

Write a C code to interface the Audio Codec AIC3106 to sample the input channel and perform under-sampling, critical sampling and over-sample. Reconstruct the signal using the interrupt method with MicroDSP6748 DSP Board, make a note of the sampling frequency and listen to the output with the below wires in your earphone:

```
#include "L138_LCDK_aic3106_init.h"

interrupt void interrupt4(void)  // interrupt service
routine
{
  uint32_t sample;

  sample = input_sample(); // read L + R samples from
ADC
  output_sample(sample);   // write L + R samples to
DAC
  return;
}

int main(void)
{

L138_initialise_intr(FS_48000_HZ,ADC_GAIN_0DB,DAC_ATTEN
_0DB,LCDK_LINE_INPUT);
  while(1);
}
```

```
(i) Left
 output_left_sample(sample);
(ii) Right
 output_right_sample(sample);
```

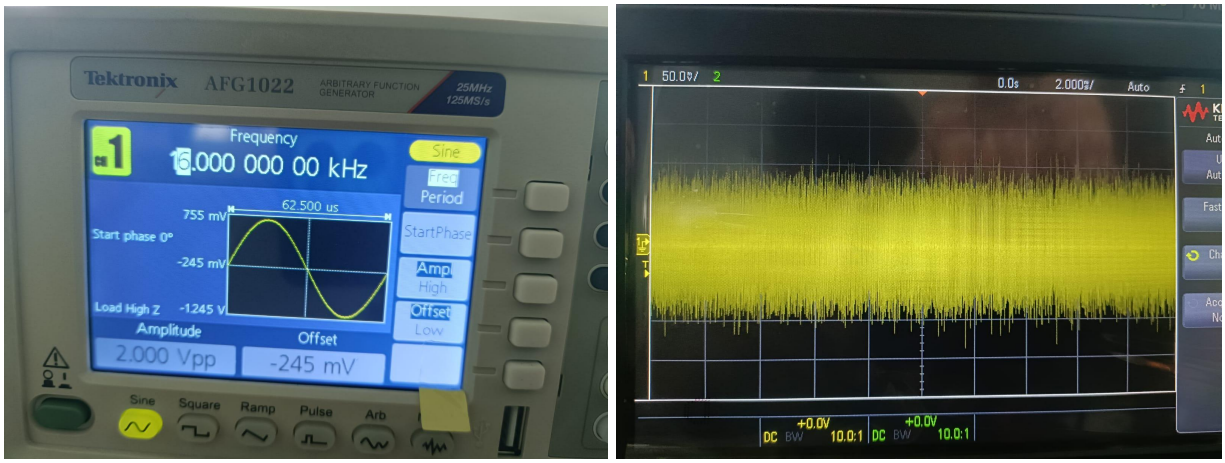
## Inference:

- In Case 1 audio is audible from the left part of the earphone and the right part has no audio output.
- In Case 2 audio is audible from the right part of the earphone and the left part has no audio output.

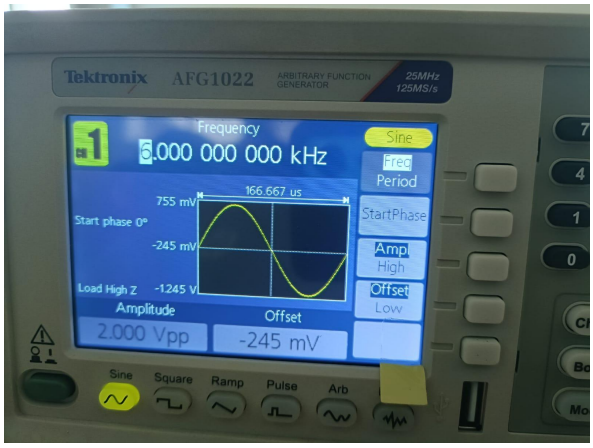
Perform sampling and reconstruction of signal using Function Generator and observe the
results in Digital Signal Oscillator by keeping the sampling frequency constant. Now vary
the input frequency and verify the types of sampling.

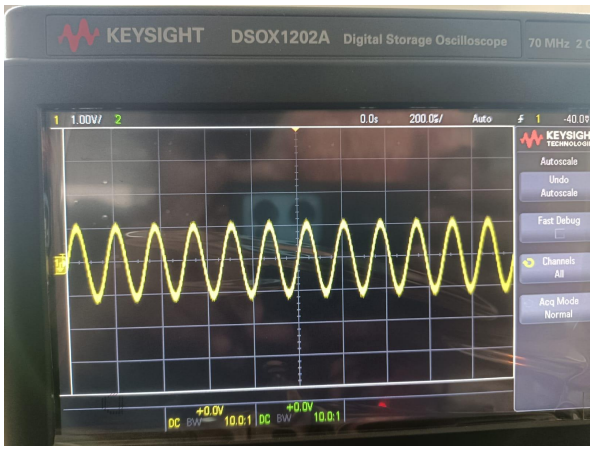
**Outputs:**

**Sampling frequency 12KHz is constant and input frequency is varied.**

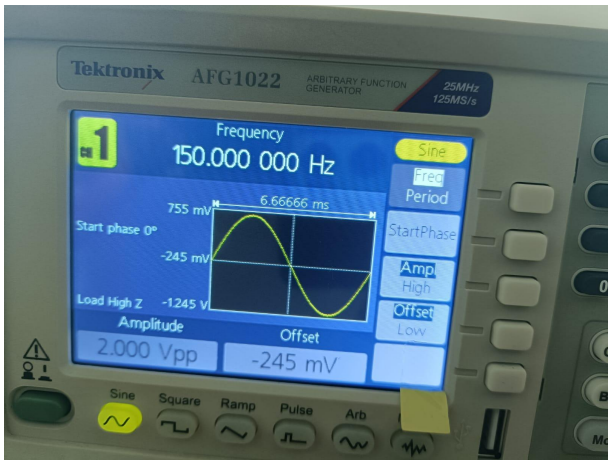


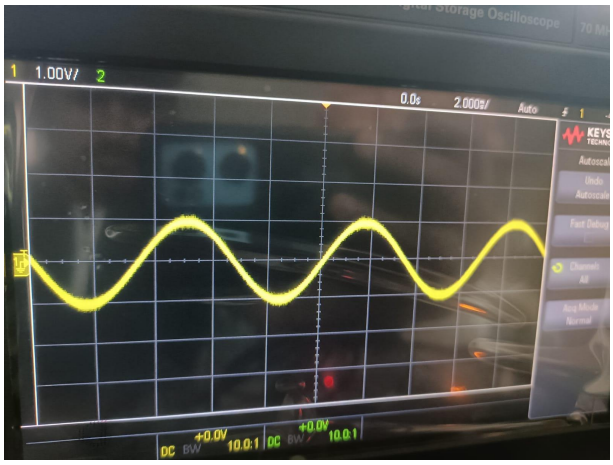

**Over Sampling For input Frequency 16Khz**




**Perfect Sampling For input Frequency 6 Khz**




**Under Sampling For input Frequency 150Mhz**

**Result:**

- Performed Sampling On MicroDSP6748 DSP Board.
- Observed UnderSampling and oversampling Using a digital oscillator.