

ISTANBUL TECHNICAL UNIVERSITY
COMPUTER AND INFORMATICS FACULTY
COMPUTER ENGINEERING

BLG438E
DIGITAL SIGNAL PROCESSING LABORATORY

DENEY - 1

Merve Cansu KÖSEOĞLU
150120704

Mehmet AYSEVİNÇ
150110705

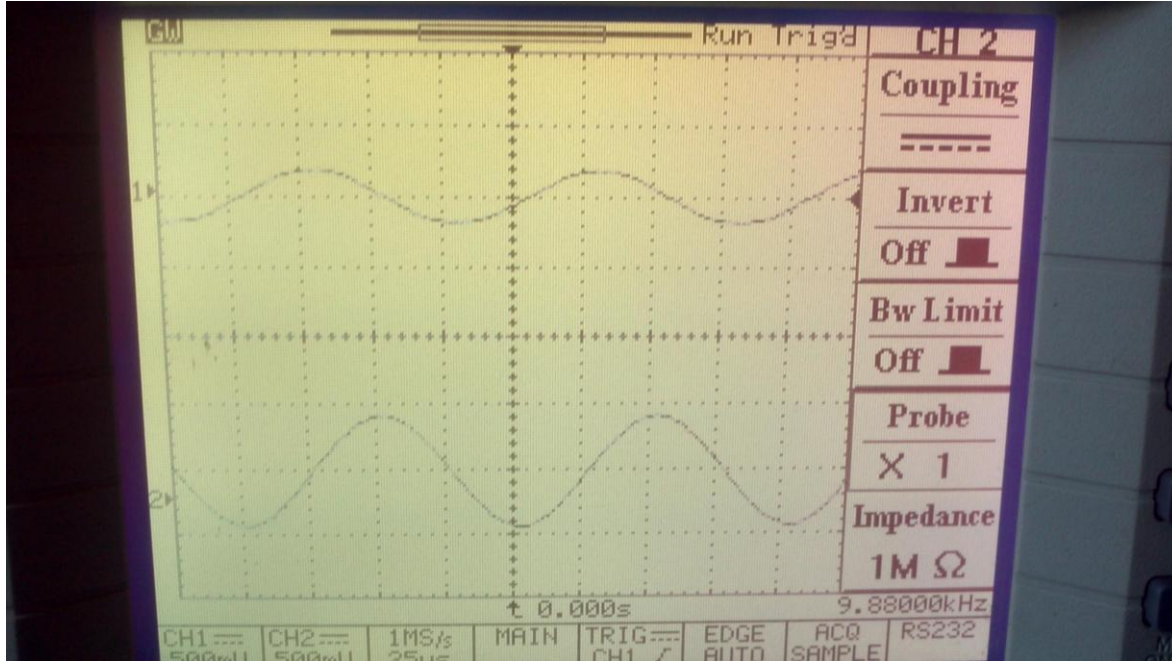
INSTRUCTOR: ASSOC. PROF. Berk ÜSTÜNDAĞ
RES. ASST. Hasan ÜNLÜ

Daha sonra “Process_data.c” kaynak dosyasındaki kodda sinüs giriş frekansını yarıya indirmek için yapılan değişiklik aşağıda verilmiştir.

2.sıradaki değişkeni kullanmamızın sebebi kiti o porta bağladığımızdan kaynaklanmaktadır. Diğer değişkenleri denediğimizde outputta herhangi bir değişim gözlenmediğinden 2.sıradaki değişkenin uyguladığımız port olduğunu anladık.

```
void Process_Data(void){  
    iChannel0LeftOut = iChannel0LeftIn;  
    iChannel0RightOut = (iChannel0RightIn)/2;  
    iChannel1LeftOut = iChannel1LeftIn;  
    iChannel1RightOut = iChannel1RightIn;  
}
```

Böylece sinüs çıkış işareti de yarıya indirilmiş oldu.

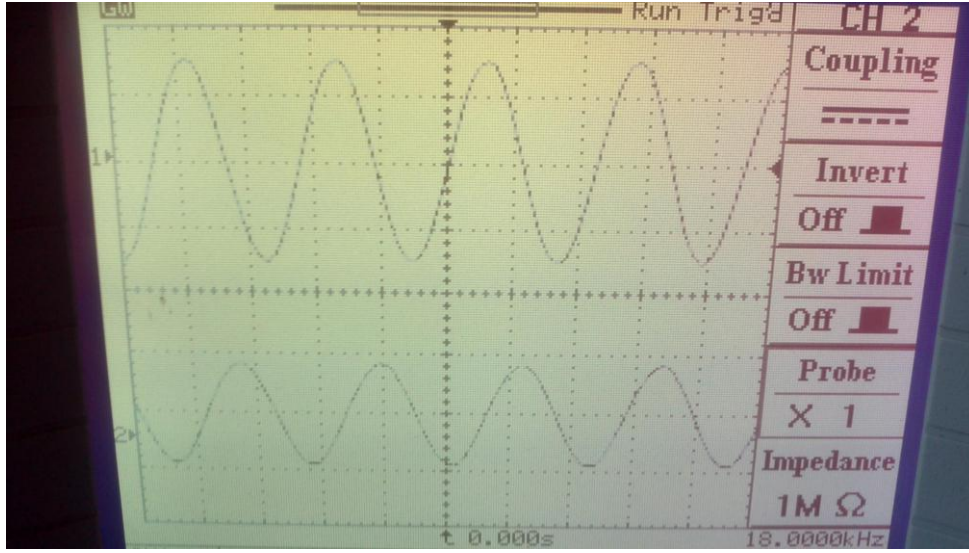


Alt kısımdaki görüntü input, üstteki ise outputtur.

Sinüs çıkış işaretini 2 katına çıkarmak için kodda yapılan ayarlama aşağıda verilmiştir.

```
void Process_Data(void){  
    iChannel0LeftOut = iChannel0LeftIn;  
    iChannel0RightOut = 2*iChannel0RightIn;  
    iChannel1LeftOut = iChannel1LeftIn;  
    iChannel1RightOut = iChannel1RightIn; }
```

Osiloskopta görülen değişimler gözlemlenmiştir.



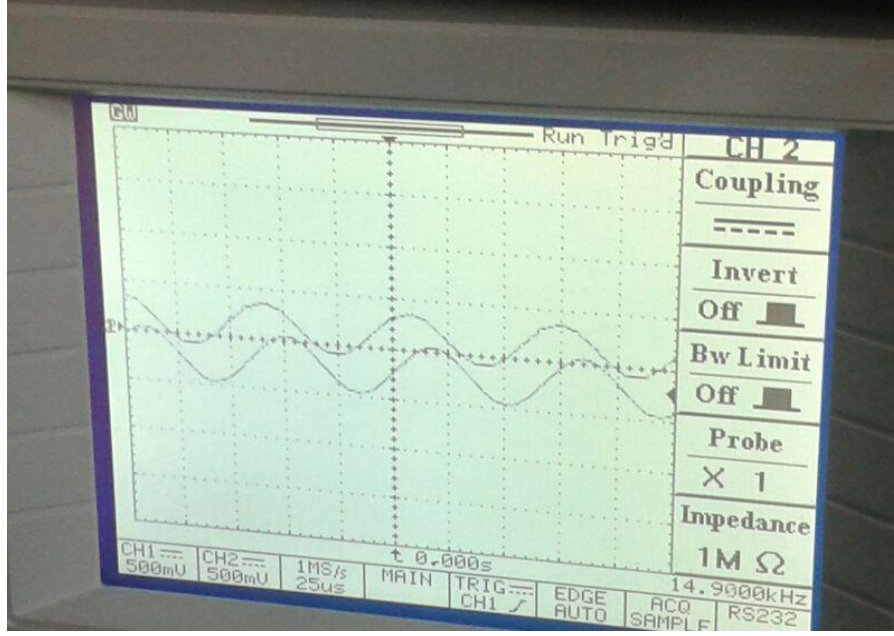
Alt kısımdaki görüntü input, üstteki ise outputtur.

Deney 1.2

Bu deneyin amacı sinyale DC bileşen eklemektir. Deneyde sinyale offset değeri ekleyerek öteleme yapmaktayız. Pek çok kit çalışmadığından karşılaşılan sorunlar nedeniyle deney sadece tek bir masada yapılabildiği görülmüştür. Deney sonuçları ve sinyal görüntüleri benzerlik göstermektedir.

Bu işlemi yapmak için 0-16 bit aralığında kuantalama yapıldı. 16 bitlik bir değer sinyale eklenerek sinyalde kayma oluşturuldu. Kod aşağıdaki gibi değiştirilmiştir.

```
void Process_Data(void){  
    iChannel0LeftOut = iChannel0LeftIn;  
    iChannel0RightOut = iChannel0RightIn + 0x1ffffff;  
    iChannel1LeftOut = iChannel1LeftIn;  
    iChannel1RightOut = iChannel1RightIn;  
}
```



Shannon Örnekleme Teoremi

Shanon örneklemesine göre bir sinyal, maksimum frekanslı bileşenin frekansının 2 katından daha büyük bir frekansla örneklenmelidir. Bu durumu aşağıdaki gösterimle ifade edebiliriz.

$$f_s > 2 f_{s_{\max}}$$

Sinyallerdeki bozulmaların pek çok sebebi bulunmaktadır. Bunlara gürültü, analog-sayısal işaretleme arasındaki geçiş sırasında veri kaybı ve iletim ortamında yayılan sinyalin yüksek frekans bileşenlerinin alçak frekans bileşenlerine göre daha fazla zayıflaması olarak örnek verilebilir.

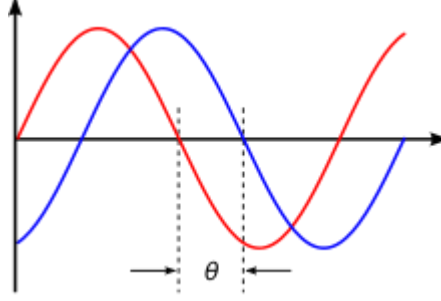
Faz Gecikmesi

Faz gecikmesi aşağıdaki eşitlikle gösterilmektedir:

$$T_{FAZ}(\omega) = -\frac{\angle H(\omega)}{\omega}$$

Gerçek hayatta $t_0 = -\angle H(\omega) / \omega$ büyüklüklerinin sabit olma şartı her zaman sağlanamaz. Bu sebeple faz ve genlik açısından sinyale ait sinüzoidal bileşenlerde bozukluklar ortaya çıkar. Bu bozulmaların sebepleri teknik donanımdan yada çevresel etmenlerden dolayı olabilir. Bu

bozulmalar, sinyalin olması gereken zamandan daha sonra oluşmasına sebep olur ve dalga boyunda kayma yaratır. Faz gecikmesi genellikle alakalı değildir. Genellikle alakalı olan grup gecikmesi olarak adlandırılır. Faz gecikmesi zaman gecikmesi ile aynı kabul edilebilir, birimi zamandır. Aşağıda faz farkı olan iki sinüzoidal dalga gösterilmiştir, kırmızı olan maviye göre kadar faz gecikmesine sahiptir denilebilir.



Kuantalama Hatası

Kuantalama, büyük kümedeki giriş değerlerini, küçük küme içine eşleme sürecidir. Bir başka deyişle örneklenmiş işaret değerlerinin önceden belirlenmiş sınırlı sayıdaki seviyeden en yakın olana yuvarlanmasıdır. Kuantalama yuvarlamalarda meydana gelecek olan hatalarda kuantalama hatasını vermektedir.

Kuantalama hatası sonucunda bilgi kaybı meydana gelmektedir. Kuantalanmış örnek işaret, mesaj işaretinin yaklaşık bir değeri olduğundan bir bozulma söz konusudur. Bu bozulma kuantlama gürültüsünü meydana getirmektedir. Bu gürültüyü azaltmak için kuantalama dilim sayısı artırılabilir. Bunun karşılık bir örneği belirlemek için kullanılması gerekli bit sayısı da artacaktır.

Deney için Kesme Kullanımı

Fonksiyon üreticisinden gelen veriler bir frame boyu kadar olduktan sonra kesme uygulanıyor. Bunun nedeni yeni gelen verilerin sisteme alınmasının önlenmesidir. Bu sayede bilgi karmaşıklığı ve zaman kaybı önlenir. Eğer kesme olmazsa kit yeni gelen verileri almaya çalışacak böylece zaman kaybı meydana gelecektir. ADSP-BF533 EZ-KIT kartı daha sonra kesme sonrası aldığı verileri işlemek için Process_Data() fonksiyonunu çağıracaktır. Daha sonra işlenen çıktı verileri uygun portların bufferlarına yüklenecektir.