

## FIR Filtre Tasarımı ve FPGA Ortamında Gerçeklenmesi

Erman ÖZPOLAT, Barış KARAKAYA, Arif GÜLTEN

Fırat Üniversitesi Mühendislik Fakültesi Elektrik-Elektronik Mühendisliği Bölümü, 23119 ELAZIĞ  
bkarakaya@firat.edu.tr

(Geliş/Received: 12.06.2017; Kabul/Accepted:24.07.2017)

### Özet

Bu çalışmada, bir alçak geçiren FIR filtrenin MATLAB ortamında tasarımı yapılmış ve tasarım FPGA (Alanda Programlanabilir Kapı Dizileri) ortamında gerçekleştirilmektedir. Tasarlanan filtre, çalışmanın ilk uygulamasında Verilog donanım tanımlama dili kullanılarak FPGA’da programlanmıştır. İkinci uygulamada ise aynı filtre tasarımı, MATLAB Simulink ile uyumlu çalışan XSG (Xilinx Sistem Üretici) platformunda gerçekleştirilmiştir. Her iki tasarım için de giriş olarak EEG sinyali uygulanmış ve filtre çıkışlarındaki sinyaller incelenmiştir. Uygulama sonuçları, FPGA’da kapladığı alan ve kaynak kullanımı bakımından karşılaştırılmıştır.

**Anahtar Kelimeler:** FPGA, MATLAB Simulink, Sayısal Filtreler, XSG.

## Design and Implementation of FIR Filter on FPGA Platform

### Abstract

In this study, design of a lowpass FIR filter is executed on MATLAB platform and the design is implemented on Field Programmable Gate Array (FPGA) platform. The designed filter was programmed on FPGA using Verilog HDL in the first application of study. In the second application, the same filter design was implemented on XSG (Xilinx System Generator) platform which runs compatibly with MATLAB Simulink. For both of the two applications, the EEG (Electroencephalography) signal was applied as input and output signals of filters are examined. Results of applications were compared with each other by means of footprint and resource utilizations on FPGA.

**Keywords:** FPGA, MATLAB Simulink, Digital Filters, XSG.

### 1. Giriş

Sayısal sinyal işleme teknikleri iletişim ve multimedya gibi birçok uygulamada yaygın olarak kullanılmaktadır. Filtreler sayısal sinyal işlemede önemli bir kullanım alanına sahiptir. Çoğu elektronik cihazlarda sayısal filtreler kullanılmaktadır. Bu yüzden sayısal filtreler, benzetimlerle gerçekleştirilmelerinin yanı sıra donanımsal olarak da gerçekleştirilebilirler. Son yıllarda, gerçek zamanlı ve yüksek başarımlı gerektiren işaret işleme uygulamalarının gerçekleştirilmesinde FPGA (Alan Programlanabilir Kapı Dizileri) kullanımının arttığı görülmektedir. Başlangıçta sadece sayısal tasarımların kontrol edilmesi için geliştirilen FPGA çipleri, teknolojiye yeni gelişmelere paralel olarak, yüksek dereceli paralel çalışabilme kabiliyetine sahip olmuştur [1,2]. FPGA ile gerçekleştirilen tasarımların güç tüketimlerinin düşük ve çalışma hızlarının oldukça yüksek olduğu bilinmektedir.

Ayrıca FPGA ile aritmetik işlemler kolayca gerçekleştirilmektedir[3]. Literatürde sayısal filtre tasarımı FPGA kullanılmasıyla ilgili birçok çalışma vardır. Yapılan bir çalışmada, bant geçiren filtre işlemini gerçekleştiren yeniden programlanabilir filtre bankası yazılımı geliştirilmiş ve tekrardan programlanabilir donanım olarak FPGA kullanılmıştır [4]. Cardarilli ve arkadaşlarının çalışmasında [5], çok oranlı filtre bankalarının FPGA ile tasarlanmasında üç çeşit algoritmanın kapsamlı karşılaştırması yapılmıştır. Elde edilen sonuçlara göre çok evreli yarım-bant filtre bankasının yüksek verimli olduğuna karar verilmiştir. Lee ve arkadaşlarının çalışmalarında [6], çeşitli basamak-seri DSP fonksiyonlarının FPGA ile gerçekleştirilmesinde, işlem gecikmeleri ve FPGA üzerinde kapladıkları alanlar üzerinde inceleme yapılmıştır. Gerçek zamanlı DSP fonksiyonları için FPGA’ların daha iyi performans gösterdiği görülmüştür. Anurag ve

arkadaşlarının yaptığı çalışmada [7], XSG kullanılarak yüksek geçiren bir filtre gerçekleştirilmiştir. Çalışma sonucunda FPGA'nın paralel işlem kapasitesi sayesinde, sayısal filtre gerçekleştirilmesindeki hızının büyük ölçüde arttığı gözlemlenmiştir. Başka bir çalışmada ise Emmanuel ve arkadaşları [8], XSG kullanarak yüksek geçiren, alçak geçiren ve bant durduran üç çeşit FIR filtreyi ayrı ayrı tasarlamış ve daha sonra hepsini aynı XSG düzleminde tasarlayarak FPGA'nın işlem kapasitesine bakılmıştır. Hepsinin aynı anda uygulanmasında bile FPGA'nın çok iyi performans gösterdiği gözlemlenmiştir.

Bu çalışmada da alçak geçiren FIR filtre, hem FPGA Verilog dili kullanarak hem de FPGA ile uyumlu çalışan XSG platformu kullanılarak tasarlanmış ve kaynak kullanımları bakımından karşılaştırılmıştır.

Bu giriş ile birlikte, ikinci bölümde materyal ve metot yer almaktadır. Bu bölümde analog ve sayısal filtreler hakkında kısa bir bilgi verildikten sonra FIR filtre yapısı ve çalışması anlatılmaktadır. Ayrıca, FPGA ve XSG platformları tanıtılırken FPGA ve XSG ortamında yapılan tasarımlar ve bu tasarımlara ait sonuçlara yer verilmektedir. Üçüncü bölümde ise elde edilen sonuçlar değerlendirilmekte ve kaynaklar bölümü ile çalışma sonlandırılmaktadır.

## 2. Materyal ve Metot

### 2.1. Filtreler

Filtreler günümüzde telefon, telgraf, radyo, radar, uydu ve haberleşme sistemleri, işaret işleme uygulamaları gibi birçok alanda kullanılmaktadır. Filtreler genellikle elektronik devrelerde, gürültüyü ve istenmeyen işaretleri süzmek, belirli frekansları birbirinden ayırmak, örnekleme işleminden önce sinyalleri sınırlamak gibi çeşitli amaçlarla kullanılmaktadırlar [9].

Elektronik sistemlerde çok önemli elemanlar olan filtreler özellikle radyo, televizyon, ses, resim ve veri haberleşmesinde kullanılan vazgeçilmez devrelerdir. Ses, hi-fi sistemleri ve elektronik müzik uygulamalarında önemli işlevleri vardır. Ayrıca sismoloji, jeofizik, tıbbi elektronik, beyin dalgaları ve uzaktan ölçüm gibi birçok bilimsel araştırma konusunda da çok

önemli işlevleri mevcuttur. Bunun yanı sıra filtreler genellikle doğru bileşenlerin süzülmesinde, gürültü azaltmada, rezonans kaçınmada veya rezonans oluşturmada, işaret biçimlendirmede, işaret zayıflatmada ve güç faktörü düzeltmede kullanılırlar [9].

Yapısal olarak filtreler analog ve sayısal veya aktif ve pasif olarak sınıflandırılırlar. Direnç, indüktans ve kapasitans elemanlarının kullanıldığı filtreler Pasif Filtre, transistör ve/veya op-amplamların da kullanıldığı filtreler ise Aktif Filtre olarak adlandırılmaktadır. En genel anlamda 2 kapılı bir filtre devresinin gösterimi Şekil 1'de verilmektedir.

Şekil 1'de gösterilen genel filtre yapısında  $H(s)$ , filtrenin transfer fonksiyonu olarak adlandırılır ve giriş – çıkış işaretleri arasındaki ilişkiyi ifade eder. Transfer fonksiyonu, giriş ve çıkış işaretlerine göre isimlendirilir.



Şekil 1. Temel filtre devresi görünümü.

#### 2.1.1. FIR filtre

Fiziksel bir olay hakkında veri taşıyan ve o olay hakkında bize bilgiler veren, bir veya daha fazla sayıda değişken içeren fonksiyonlara işaret denir. İstenilen özelliklere sahip işaret üreten veya girişine uygulanan işarete göre çıkışında istenilen özelliklere sahip çıkış üreten yapılara ise sistem denir. İşaret işleme ise bilgisayar ya da özel olarak üretilmiş olan sayısal işaret işleme donanımları sayesinde, sistemleri meydana getiren yapıdır. İşaret işleme genel olarak tanımlanacak olursa, bir işaret dizisinin çeşitli işlemler yapıldıktan sonra istenilen hale getirilmesi işlemidir [10].

Sayısal işaret işlemenin amacı, bir işaretin frekans spektrumu üzerinde belirli frekanslarda istenilen işlemleri yapması ve bu işlemlerden sonra istenilen özellikte sonuç alınmasını sağlamaktır. Yapılan bu sayısal işaret işleme yapısına sayısal filtre adı verilmektedir. Sayısal filtreler yazılım veya donanımla gerçekleştirilebilirler. Sayısal filtreler darbe

cevaplarına göre ikiye ayrılmaktadırlar. Bunlar FIR (sonlu darbe cevaplı) ve IIR (sonsuz darbe cevaplı) filtrelerdir [11].

Rasyonel bir sistem fonksiyonu ile doğrusal ötelemeyle değişmez bir sistem için  $x(n)$  girişi ve  $y(n)$  çıkışı doğrusal sabit katsayılı fark denklemiyle ilişkilendirilir. Örneğin  $h(n) = a^n$ .  $u(n)$  birim örnek cevabına sahip olan bir sistemin çıkışına ilişkin ifade Denklem 1'deki gibi tanımlanır.

$$y(n) = \sum_{k=0}^{\infty} a^k \cdot x(n - k) \quad (1)$$

Bu denklem herhangi bir  $x(n)$  girişi için  $y(n)$  çıkışının hesaplanmasına izin vermesine rağmen, bu gösterim hesaplama açısından çok etkin değildir. Bazı durumlarda girişin mevcut ve geçmiş değerlerine ek olarak çıkışın da geçmiş değerler cinsinden ifade edilmesi mümkün olabilir. Örneğin önceki sistemin birinci dereceden ifadesi Denklem 2'deki gibi tanımlanabilir.

$$y(n) = a \cdot y(n - 1) \quad (2)$$

FIR filtre yapısı birim geciktirme elemanı, çarpıcı ve toplayıcı olmak üzere üç temel elemandan oluşmaktadır. Birim geciktirme elemanlarının sayısı darbe cevabının sonlu süresini tanımlamaktadır.  $n-1$  olarak gösterilmiş olan birim geciktirme elemanların sayısı filtrenin derecesini belirlemektedir. Çarpıcı elemanlar ise bağlı oldukları giriş işaretini ilgili filtre katsayısı ile çarpırlar. Yani,  $k$ . geciktiricinin çıkışını temsil eden  $x(n-k)$  girişine karşılık çarpıcının çıkışı üretilecektir. Toplayıcılar ise, çarpıcı çıkışlarını birbiri ile toplayarak filtreye ait toplam çıkışı üretecektir. Yukarıda verilmiş olan FIR filtrenin giriş çıkış ilişkisi Denklem 3'deki gibi verilebilir;

$$y(n) = \sum_{k=0}^{N-1} b_k x(n - k) \quad (3)$$

burada  $x(n)$  giriş işaretini,  $y(n)$  ise filtre çıkışını temsil etmektedir.

## 2.2. FPGA (Alanda Programlanabilir Kapı Dizileri)

FPGA, sayısal mantığın sistemlerde uygulanmasına yönelik bir alternatiftir. FPGA'lar

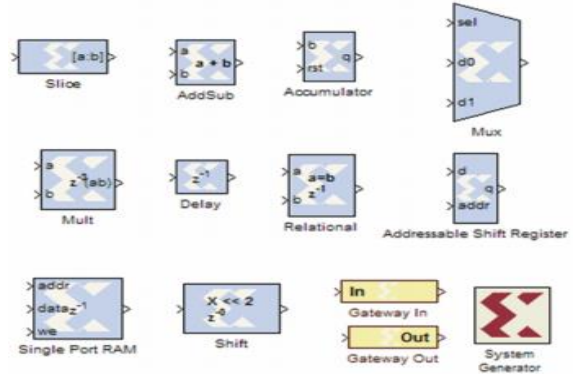
herhangi bir sayısal tasarımı uygulamak üzere elektriksel olarak programlanabilen silikon çiplerdir. İlk statik bellek tabanlı FPGA (genellikle SRAM-tabanlı FPGA olarak adlandırılır), 1967 yılında Wahlstrom tarafından tasarlanmıştır. Bu mimari, bir konfigürasyon bitleri dizisi kullanarak mantık ve ara bağlantı konfigürasyonuna olanak tanımıştır. Daha sonra modern çağımızın ilk ticari FPGA'sı, 1984 yılında Xilinx tarafından sunulmuştur. Bu FPGA, düşük sayıda bir klasik Konfigüre Edilebilir Mantık Blokları (CLB'ler) dizisi ile girişler/çıkışlar içermektedir. 64 CLB ve 58 giriş/çıkış içeren ilk FPGA'dan bu yana FPGA'lar karmaşıklık açısından son derece gelişmiştir. Günümüzün modern FPGA'ları, yaklaşık 330.000 mantık bloğu ile 1100 giriş ve çıkış içerebilmektedir. FPGA'nın temel mimarisi, üç ana bileşenden meydana gelmektedir. Bunlar; mantık fonksiyonlarını yerine getiren programlanabilir mantık blokları, bu fonksiyonları yerine getirmek üzere programlanabilir yönlendirme blokları (ara bağlantı kurma) ve çip-dışı bağlantıları yapmak üzere giriş/çıkış bloklarıdır.

VHDL, Verilog, SystemC ve Handel-C gibi donanım tanımlama dilleri sıklıkla FPGA programlaması için kullanılmaktadır. VHDL ve Verilog endüstri standardı haline gelmiştir. Çok sayıda sağlayıcısı olan Donanım Tanımlama Dilleri (HDL), benzetim ve sentez araçları sunmaktadır. Tanımlama davranışı ile RTL ve yapısal seviyeleri bu dillerde alternatifli olarak kullanılabilir. SystemC, sistem seviyesi davranışını modellemek için kullanılan C++ tabanlı bir kitaplardır. Taban dili C++ olduğundan yazılım süreçleri daha klasik bir HDL.2'dekine kıyasla daha kolay modellenmektedir. SystemC'ye yönelik sentez araçları ortaya çıkmakta olup VHDL ya da Verilog sentez ürünlerinin olgunluğuna yaklaşmamıştır. Handel-C de VHDL ya da Verilog'a nazaran yeni bir üründür. Handel-C, İletişim Ardışık Süreci (CSP) modelini takip etmektedir. Handel-C, tasarımcının bir süreç içerisindeki paralel işlem bloklarını açıkça tarif etmesini gerektirmektedir. Bu, SystemC 2.0'm yaptığı gibi süreçler arası iletişime yönelik bir iç yapı içerir.

### 2.2.1. XSG (Xilinx Sistem Üretici)

XSG, çeşitli Xilinx FPGA'lara uygulanabilecek birkaç donanım işleme yönelik bir dizi Simulink blokları sağlamaktadır. Bu bloklar, Simulink ortamını kullanarak donanım sisteminin fonksiyonelliğini ve benzetimini yapmak üzere kullanılmaktadır. Çoğu DSP (Sayısal Sinyal İşleme) uygulamasının niteliği, veri gösterimine yönelik kayan noktalı sayı formatı gerektirmektedir. Simulink gibi yüksek seviyeli modelleme yazılımını çalıştıran çeşitli bilgisayar sistemlerinde bunu uygulamak kolay olsa da donanım yazılımında bu işlem, kayan noktalı sayı aritmetiğinin uygulama karmaşıklığından ötürü çok daha zordur. Sistem tasarımına daha sınırlayıcı kısıtların uygulandığı taşınabilir DSP sistemler ile bu zorluklar artmaktadır. Bu nedenlerden ötürü XSG, sistemdeki bütün sayısal değerleri göstermek için sabit noktalı sayı formatı kullanılmaktadır. Sistem üretici, benzetim ortamının yazılım tarafından ve donanım tarafından sağlanan verileri dönüştürmek üzere bazı bloklar sağlamaktadır [12].

Sistem üretici, FPGA tasarımına yönelik olarak MathWorks model-tabanlı Simulink tasarım ortamının kullanılmasına olanak tanıyan Xilinx'e ait bir DSP tasarım aracıdır. Tasarımlar, Xilinx'e özel bir blok seti kullanılarak DSP dostu Simulink modelleme ortamında yakalanmaktadır. Sentez, yerleştirme ve yönlendirme gibi FPGA'nın ileri seviye tüm uygulama adımları, bir FPGA programlama dosyası oluşturmak üzere otomatik olarak gerçekleştirilmektedir [13]. 90'dan fazla DSP yapı bloğu, Simulink'e yönelik olarak Xilinx DSP blok setinde sağlanmaktadır. Bu bloklar; toplayıcılar, çoğaltıcılar ve kaydediciler gibi genel DSP yapı bloklarını içermektedir. Gönderim yönünde hata düzeltim blokları, FFT'ler, filtreler ve bellekler gibi kompleks DSP yapı blokları dizisi de yer almaktadır. Bu bloklar, seçilen cihaz için optimize sonuçlar elde etmek üzere Xilinx IP çekirdek üreticilerini güçlendirmektedir [13]. Şekil 2'de genel DSP yapı blokları verilmiştir.



Şekil 2. Bazı DSP yapı blokları.

### 2.3. FIR filtre uygulamaları

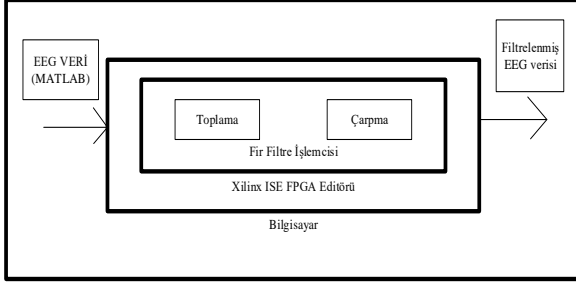
Bu bölümde MATLAB ortamında yapılan uygulamalar sunulmaktadır. Çalışmada iki uygulama bulunmaktadır. Her iki uygulamada da filtre olarak alçak geçiren 4. dereceden FIR filtre tasarımı kullanılmaktadır. Verilog dili kullanılarak yapılan uygulama, Xilinx FPGA ISE editörü ve kodlamanın gerçekleştirildiği bir bilgisayardan oluşmaktadır. EEG sinyalini filtrelemek için kullanılan FIR filtrenin katsayıları MATLAB FDATool (filtre tasarım aracı) kullanılarak oluşturulmuş, daha sonra FPGA'daki gömülü sisteme kodlanmıştır. XSG, FPGA'da verimli sayısal sinyal işleme algoritmaları tasarımında kullanılır. XSG kullanılarak yapılan uygulamadaki filtrenin parametre ve özellikleri, ilk uygulamada kullanılan filtrenin aynıdır. Gerçekleştirilen filtre daha sonra Digilent Basys 2 kitinde yer alan Xilinx Spartan 3E-1600FG320 FPGA çipinde sentezlenmiştir.

#### 2.3.1. Verilog HDL kodlama ile yapılan uygulama

Bu çalışmada, bir EEG sinyali FPGA platformunda Verilog HDL ile programlanan FIR filtre tasarımı ile işlenmiştir. Filtrelemek için kullanılan EEG sinyali [14], The International Federation of Clinical Neurophysiology tarafından kaydedilmiştir.

MATLAB ortamında tasarlanan filtrenin bazı parametreleri  $\omega_{pass} = 0.4$ ,  $\omega_{stop} = 0.6$ ,  $A_{pass} = 1$ ,  $A_{stop} = 5$  dB olacak şekilde alınmıştır. Elde edilen bu katsayılar EEG sinyalini filtrelemek için FPGA da oluşturulan FIR filtre işlemcisine programlanmıştır. FPGA tabanlı FIR filtre

işlemcisi olarak adlandırılan tasarım Şekil 3’de görülmektedir.



Şekil 3. FPGA tabanlı FIR filtre işlemcisine ait sistem

FIR Filtre işlemcisi, toplama ve çarpma aritmetik devrelerinden oluşmaktadır.  $-1$  ile  $1$  arasındaki değerlere normalize edilmiş EEG sinyali sabit noktalı sayı gösterim formatına dönüştürülmüştür. Kullanılan tüm sayıların uzunluğu  $16$  bit seçilmiştir. Sayılar  $-1$  ile  $1$  arasına normalize edildiğinden dolayı,  $Qm.n$  sayı gösterme biçimi  $Q3.13$  olarak düzenlenmiş olup, burada  $m$  ondalıklı sayının tamsayı kısmını ve  $n$  ondalıklı kısmını temsil etmektedir.

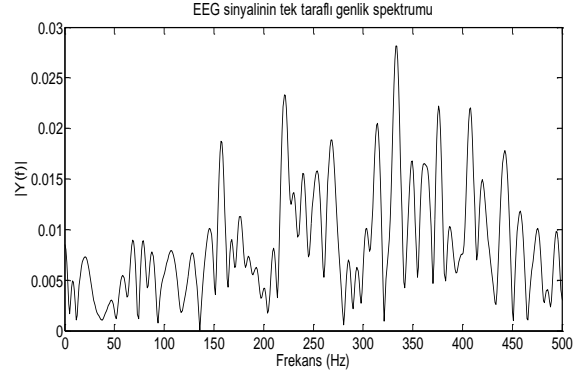
Kullanılan tüm sayılar işaretli sayı olduğundan, işaret biti için  $1$  bit ayrılmıştır. Dolayısıyla, sayı gösterim biçimi son olarak  $Q2.13$   $16$  bitlik işaretli sabit noktalı sayı gösterim biçimi olarak MATLAB programında düzenlenmiştir. Bu durumda sayıların hassasiyeti  $0,122 \times 10^{-4}$  olarak elde edilmiştir.

Filtre derecesi  $4$  olarak seçildiğinden, filtrenin  $5$  adet katsayısı bulunmaktadır. Bu katsayılar;  $a_0 = -0.125$ ,  $a_1 = 0.375$ ,  $a_2 = 0.515$ ,  $a_3 = 0.375$  ve  $a_4 = -0.125$  olarak elde edilmiştir.

EEG sinyalinin filtrelenmesi sırasında veri, giriş olan  $x(n)$  için  $14$  değer ve çıkış olan  $y(n)$  için  $10$  değer olarak işlemciye transfer edilmiştir. Kullanılan EEG sinyali  $160$  veriye sahiptir. Filtre tasarımı, her çalıştırmada  $10$  giriş verisi işleyebildiğinden dolayı, toplamda  $16$  kez çalıştırılmıştır.

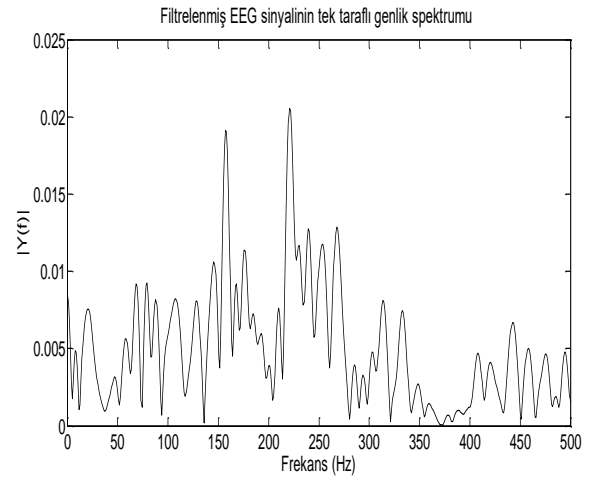
Toplama ve Çarpma Devreleri, Xilinx ISE FPGA editöründe programlanmış olup  $2$  saat çevrimi süresince cevap vermektedir [15]. Girişe verilen EEG sinyalinin işlenmesi için gereken toplam saat çevrimi sayısı  $194$ 'tür. Bu yüzden filtreleme işleminin tamamlanması  $1940$  ns sürmüştür. FIR Filtre İşlemci modelinde kullanılan EEG verisi MATLAB'tan işlemciye transfer edilmektedir. Filtreleme işlemi

sonrasında yeni giriş verisi tekrar filtreleme işlemi için işlemciye uygulanmaktadır. Gerçekleme sonucunda bütün filtrelenmiş veriler çizdirilmek üzere MATLAB' a transfer edilmektedir. Örnekleme frekansı ( $f_s$ )  $1000$  Hz alınarak orjinal EEG sinyalinin tek taraflı genlik spektrumu Şekil 4’de verilmiştir.

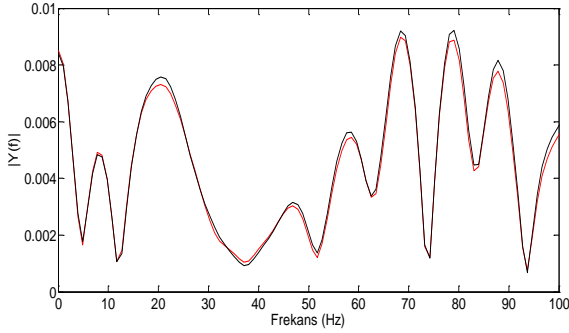


Şekil 4. Giriş EEG sinyalinin tek taraflı genlik spektrumu

Şekil 5’de filtrelenmiş EEG sinyalinin tek taraflı genlik spektrumu verilmiş olup yüksek frekans değerlerinde genliğin bastırıldığı görülmektedir. Düşük frekans değerlerinde ise tasarladığımız alçak geçiren  $4$ . dereceden FIR Filtre özelliğinden dolayı giriş sinyaline oldukça yakın bir sinyal elde edilmiş olup Şekil 6’da siyah sinyal; giriş sinyalini, kırmızı sinyal ise çıkış sinyalini gösterecek şekilde  $0-100$  Hz frekans aralığında çizdirilerek neredeyse aynı genlikte oldukları ve filtreleme işleminin başarıyla gerçekleştirildiği gözlemlenmiştir.



Şekil 5. Filtrelenmiş EEG sinyalinin tek taraflı genlik spektrumu



Şekil 6. 0-100 Hz aralığında giriş ve çıkış sinyalinin spektrumları

### 2.3.2. XSG blokları ile yapılan uygulama

Bu uygulamada, EEG sinyali XSG kullanılarak önce MATLAB Simulink ortamında tasarlanan FIR Filtre ile filtre edilmiş ve daha sonra FPGA üzerinde gerçekleştirilmesi incelenmiştir.

Sistemde kullanılacak EEG sinyali MATLAB çalışma ortamından alınarak Xilinx Geçit Giriş bloğuna uygulanmıştır. Daha sonra Xilinx Geçit Giriş bloğundan alınan sinyal önce kayıt bloğuna sonra ise FIR Derleyici bloğuna uygulanmıştır. FIR Derleyici bloğunda kullanılan filtre, FDA Tool bloğu tarafından tasarlanmıştır.

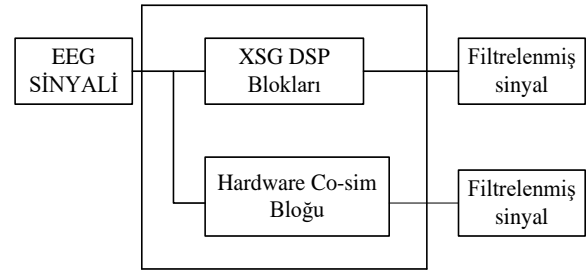
Filtre tasarlandıktan sonra filtre katsayıları, FIR Derleyici bloğuna tanımlanmak üzere MATLAB çalışma ortamına aktarılmıştır. Filtre katsayılarının aktarımından sonra FIR Derleyici 5.0 bloğuna girilerek kullanacağımız filtrenin katsayıları FIR Derleyici bloğuna tanıtılmıştır. FIR Derleyici 5.0 bloğundan alınan sinyal tekrar kayıt bloğuna uygulanıp daha sonra Xilinx Geçit Çıkış bloğuna uygulanmıştır. Xilinx Geçit Çıkış bloğundan alınan sinyal MATLAB çalışma ortamına aktarılıp tekrardan çizdirilmiştir.

MATLAB Simulink ortamında tasarlanan sistemin FPGA'da gerçekleştirilmesini yapabilmek için FPGA kartını temsil eden "JTAG Co-sim" bloğu XSG tarafından oluşturulur. Bu bloğu oluşturmak için önce Sistem Üretici bloğuna girilir. Sistem Üretici bloğuna girildikten sonra açılan pencerede "Compilation" bölümünden Bitstream seçeneği seçilir. Sonrasında altında bulunan "Part" bölümünden kullanılan FPGA kartı modeli seçilir. Bu çalışmada Spartan-3E XC3S1600E-4FG320 kullanılmıştır. Daha sonra programlamada gerekli olan ".bit" uzantılı dosya

ve verilog yazılımı oluşturulur. Oluşturulan JTAG Co-sim bloğu Şekil 7'de gösterilmiştir. Bu blok, Xilinx Geçit Giriş bloğu ve Xilinx Geçit Çıkış blokları dahil arasındaki blokların gerçekleştirdiği işlemleri FPGA kartında yapan tek bir bloktur. Oluşturulan model ise Şekil 8'de gösterilmiştir.

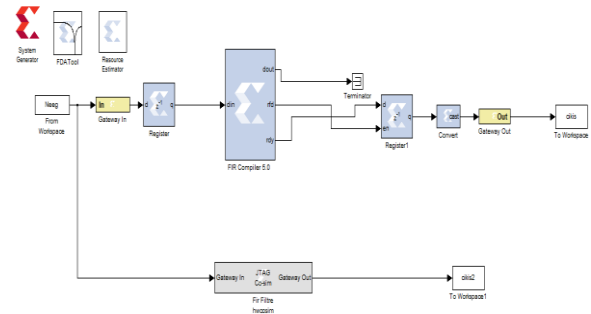


Şekil 7. Oluşturulan JTAG Co-sim bloğu



Şekil 8. Oluşturulan model gösterimi

Donanım eş zamanlı benzetim bloğu kullanılarak tasarlanan Simulink tabanlı 4. derecen FIR filtre modeli Şekil 9'da gösterilmektedir. Model çalıştırıldığında ".bit" uzantılı dosya USB portu üzerinden FPGA kartına gönderilir. Program çalıştırdıktan sonra giriş sinyalimiz olan EEG sinyali XSG DSP bloklarından ve aynı anda donanım eş zamanlı benzetim bloğundan geçerek FPGA kartı üzerinde filtrelenen sinyal filtrelenmiş şekilde çizdirilmiştir. Şekil 6'daki sonucun aynısı elde edilmiştir. Bu işlem seri ve sayısal olarak gerçekleştirilmiştir. FIR filtre uygulamalarına ait kaynak tüketimi Tablo 1'de verilmektedir.



Şekil 9. Donanım eş zamanlı benzetim bloğu kullanılarak oluşturulan Filtre modeli

**Tablo 1.** Uygulamaların donanım üzerindeki kaynak tüketimi.

Donanımsal kaynaklar	Kullanım Sayısı	
	Kodlama ile yapılan tasarım	XSG ortamındaki tasarım
Lojik Dilim	12145	2231
Flip-Flop	14523	2612
Doğruluk Tablosu (LUT)	21785	3364

Tablo 1'deki veriler incelendiğinde XSG kullanılarak yapılan 4. dereceden alçak geçiren FIR filtrenin kaynak kullanımının daha az olduğu görülmüştür. Bu nedenle 4. dereceden alçak geçiren FIR filtrenin XSG kullanılarak gerçekleştirilmesinin, performans açısından daha iyi olduğu sonucuna varılmıştır.

### 3. Sonuçlar

Bu çalışmada MATLAB Simulink ile uyumlu çalışan XSG ve FPGA yazılım dili olan Verilog kullanılarak 4. dereceden alçak geçiren FIR filtre gerçekleştirilmiştir. Filtreleme işlemi için giriş sinyali olarak EEG sinyali kullanılmıştır. Bu çalışmanın sonucunda FPGA'nın hızından ve paralel işlem yeteneğinden dolayı tasarlanan sayısal filtrelerin filtreleme işlemini başarılı bir şekilde gerçekleştirdiği gözlemlenmiştir. Bu çalışmada yapılan filtre tasarımları görüntü işleme, sinyal işleme, haberleşme sistemleri gibi uygulamalar için kişiselleştirilebilen gerçek zamanlı bir sayısal filtre uygulamak için büyük kolaylıklar sağlar.

Kaynak kullanımları bakımından karşılaştıracak olursak XSG kullanılarak tasarlanan filtrenin daha az kaynak kullandığı görülmüştür.

### 4. Kaynaklar

1. DeHon A. (2000), The density advantage of configurable computing, *Computer*, **33** (4): 41-49.
2. Qasim S.M., Abbasi S.A., Almashary B. (2009), An overview of advanced FPGA architectures for

optimized hardware realization of computation intensive algorithms, In *Multimedia, Signal Processing and Communication Technologies, IMPACT'09*, 300-303.

3. Bıçakçı S., Çetinkaya M.B., Karaboğa N., (2005), Sayısal FIR Süzgeç İçin Vhdl Kod Geliştirilmesi, *Elektrik-Elektronik Bilgisayar Mühendisliği 11. Ulusal Kongresi ve Fuarı, İSTANBUL*, 332-334.

4. Louzao J, Paz S, Tejera D, Bellora G, Langwagen G., (2003), Architectural design of a programmable cell for the implementation of a filter bank on FPGA, *Microelectronics Reliability*.

5. Re M., Cardarilli G.C., Re A.D., Lojacona R., (2000), FPGA Implementation of a Demux Based on a Multirate Filter Bank, *ISCAS 2000 – IEEE International Symposium on Circuits and Systems*.

6. Lee H, Sobelman G.E., (1999), Performance evaluation and optimal design for FPGA- based digital DSP functions, *Computers and Electrical Engineering*.

7. Anurag A., Astha S., Tushar N., (2013), FIR Filter Designing using Xilinx System Generator, *International Journal of Computer Applications* 68 (11).

8. Emmanuel S., Kolawole H., Warsame A., Cofie P., Fuller J., Tolliver C., Pamela O., (2015), Design and Implementation of Low-Pass, High-Pass and Band-Pass Finite Impulse Response (FIR) Filters Using FPGA Circuits and Systems, **6**, 30-48.

9. Murat D., (2013), Video frekans uygulamaları için a-sınıfı logaritmik ortam epileptik filtre tasarımı, Yüksek Lisans Tezi, Pamukkale Üniversitesi Fen Bilimleri Enstitüsü.

10. Batık Z., (2011), Sayısal Filtre Tasarım Yöntemleri Ve Performans Analizleri, Yüksek Lisans Tezi, Sakarya Üniversitesi Fen Bilimleri Enstitüsü.

11. Ertürk S., Sayısal işaret işleme, Birsan Yayınevi, İstanbul, 293.

12. Areibi S., (2016), Tutorial - Using Xilinx System Generator 14.6 for Co-Simulation on Digilent NEXYS3 (Spartan-6) Board.

13. Xilinx, [https://www.xilinx.com/support/documentation/sw\\_manuals/xilinx11/sysgen\\_gs.pdf](https://www.xilinx.com/support/documentation/sw_manuals/xilinx11/sysgen_gs.pdf)

14. Clin N., (2006), American Clinical Neurophysiology Society. Guideline 8: guidelines for recording clinical EEG on digital media. **23(2)**:122–124.

15. Karakaya B., Yeniceri R., Yalcın M.E., (2015), Wave computer core using fixed-point arithmetic, 2015 IEEE International Symposium on Circuits and Systems (ISCAS), 1514-1517.