



FPGA TABANLI FIR FİLTRE TASARIMI

Erman ÖZPOLAT

**Yüksek Lisans Tezi
Elektrik-Elektronik Mühendisliği Anabilim Dalı
Danışman: Doç. Dr. Arif GÜLTEN
AĞUSTOS-2017**

T.C
FIRAT ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

FPGA TABANLI FIR FİLTRE TASARIMI

YÜKSEK LİSANS TEZİ

Erman ÖZPOLAT

(132113103)

Tezin Enstitüye Verildiği Tarih : 11 Temmuz 2017

Tezin Savunulduğu Tarih : 01 Ağustos 2017

Tez Danışmanı : Doç. Dr. Arif GÜLTEN (F.Ü)

Diğer Jüri Üyeleri : Yrd. Doç. Dr. Vedat ÇELİK (F.Ü)

Yrd. Doç. Dr. M. Eşref ALKIŞ (Muş A.Ü)

AĞUSTOS-2017

ÖNSÖZ

Bu tez çalışmasını yürüten ve tezin hazırlanma aşamasında bilgi birikimi, tecrübesini ve yardımlarını eksik etmeyen değerli danışman hocam Doç. Dr. Arif GÜLTEN'e teşekkürlerimi sunarım.

Ve tabi ki hayatımın bütün aşamalarında yanımda olan ve desteklerini sonuna kadar esirgemedi gösteren sevgili eşime ve aileme sonsuz şükranlarımı sunarım.

Erman ÖZPOLAT

Elazığ-2017

İÇİNDEKİLER

Sayfa No

ÖNSÖZ	II
İÇİNDEKİLER	III
ÖZET.....	V
ABSTRACT.....	VI
ŞEKİLLER LİSTESİ.....	VII
TABLOLAR LİSTESİ.....	IX
KISALTMALAR LİSTESİ.....	X
1. GİRİŞ.....	1
2. FİLTRELER.....	3
2.1. Filtrenin Tanımı	3
2.2. Filtre Türleri ve Temel Filtre Terimleri	4
2.2.1.Kararlılık	7
2.2.2. Nedensellik.....	8
2.2.3.Rasyonellik.....	9
2.2.4. Analog Filtre İle Sayısal Filtre Arasındaki Farklar	9
2.3. Sayısal Filtreler	10
2.3.1. FIR Filtre Yapısı	11
2.3.2. IIR Filtre Yapısı	12
2.3.3. FIR (Sonlu Darbe Yanıtı) ve IIR (Sonsuz Darbe Yanıtı) Karşılaştırması	14
3. FPGA(ALANDA PROGRAMLANABİLİR KAPI DİZİLERİ).....	15
3.1 Programlanabilir Mantık.....	16
3.1.1 Programlanabilir Ara Bağlantı	16
3.1.2. Programlanabilir I/O	16
3.1.3. Donanım Tanımlama Dili	17
3.1.4. Bir FPGA'nın seçimi	17
3.1.5. FPGA Uygulamaları	18
3.1.6. FPGA Teknolojisinin faydaları	19
3.2. Xilinx System Generator.....	21
3.2.1. Tasarım için kullanılan Xilinx System Generator Blokları	21
3.2.2 FIR Derleyici.....	22
3.2.3 Matlab Desteği	23
3.2.4 Sistem Kaynak Kestirici.....	24
3.2.5 Eş zamanlı Simülasyon	25
3.2.6 Sistem Entegrasyon Platformu.....	26
3.2.7 Xilinx Geçit Giriş Bloğu	27
3.2.8 Xilinx Geçit Çıkış Bloğu.....	27
3.2.9 Xilinx Kayıt Bloğu.....	28
3.2.10 Xilinx Dönüştürme Bloğu.....	28
3.2.11 Xilinx FDTTool Bloğu	28
3.2.12 Xilinx JTAG Eş Zamanlı Simülasyon Bloğu.....	29
4. UYGULAMALAR	30
4.1 FPGA Tabanlı Verilog Dili Kodlanarak Tasarlanan Filtre Uygulaması.....	30

4.2 Xilinx System Generator Kullanılarak FIR Filtre Tasarımı Uygulaması	39
5. SONUÇLAR	50
KAYNAKLAR	51
ÖZGEÇMİŞ.....	53



ÖZET

Bu tez çalışmasında FPGA (Alanda Programlanabilir Kapı Dizileri) kullanılarak bir sayısal filtre tasarımına yönelik bir yaklaşım sunulmuştur. İlk uygulamada FPGA’da bir yazılım dili olan Verilog kullanılarak sayısal bir filtre tasarlanmıştır. İkinci uygulamada ise aynı filtre Matlab Simulink ile uyumlu çalışan XSG (Xilinx System Generator) kullanılarak tasarlanmıştır. Daha sonra tasarlanan filtrelerin girişlerine EEG (Elektroensefalografi) sinyali uygulanmış ve sonuçlar incelenmiştir. Çalışma sonucunda FPGA’da tasarlanan sayısal filtreler FPGA yapısının hızı, kapladığı alanı, kaynak kullanımı ve kullandığı flip flop sayısı bakımından karşılaştırılmıştır. Yapılan bu çalışma sonucunda XSG kullanılarak tasarlanan filtrenin daha az kaynak kullandığı gözlemlenmiştir.

Anahtar Kelimeler : FPGA, Sayısal Filtreler, Xilinx System Generator, Matlab Simulink

ABSTRACT

FPGA BASED FIR FILTER DESIGN

An approach to designing a digital filter using FPGA (Field Programmable Gate Array) has been suggested in this dissertation study. The first application is that a digital filter has been designed, coding Verilog, a software language in FPGA. The second application is that using XSG (Xilinx System Generator) running compatibly with MATLAB Simulink. And then EEG (electroencephalography) signal has been applied to the systems input in the design filters and results were examined. The digital filters designed in FPGA following this study have been compared by FPGA speed, footprint, use of resource and number of flip flop. As a result of this study, it has been observed that the filter designed using XSG uses less resources.

Keywords: FPGA, Digital Filters, Xilinx System Generator, Matlab Simulink

ŞEKİLLER LİSTESİ

Sayfa No

Şekil.2.1 Genel Filtre Devresinin Görünümü.....	3
Şekil.2.2 Temel ideal filtre türlerinin genlik-frekans değişimleri, a) Alçak geçiren b) Yüksek geçiren c) Band geçiren d) Band durduran e) Tüm geçiren	5
Şekil 2.3 Alçak geçiren filtre cevabı, a) İdeal b) Yaklaşık.....	6
Şekil 2.4 Kararlı filtreye ait, a) Sıfır b) Kutup gösterimi	8
Şekil 2.5 a) Nedensel olmayan b) Nedensel olan sisteme ilişkin darbe cevabı.....	9
Şekil 2.6 İleri beslemeli FIR filtre yapısı	11
Şekil 2.7 FIR filtreye geri besleme katının ilave edilmesiyle oluşan IIR filtre yapısı.	12
Şekil 2.8 Direkt Form IIR filtre yapısı.	13
Şekil 3.1 FPGA'nın iç yapısı.....	15
Şekil 3.2 DSP yapı blokları	22
Şekil 3.3 FIR derleyici.....	23
Şekil 3.4 Mcode bloğu.....	24
Şekil 3.5 Kaynak kestirici.....	25
Şekil 3.6 Eşzamanlı simülasyon bloğu	26
Şekil 3.7 Sistem entegrasyon platformu	26
Şekil 3.8 Geçit giriş bloğu	27
Şekil 3.9 Geçit çıkış bloğu.....	28
Şekil 3.10 Kayıt bloğu	28
Şekil 3.11 Dönüştürme bloğu	28
Şekil 3.12 FDATool bloğu	29
Şekil 3.13 JTAG eşzamanlı simülasyon bloğu.....	29
Şekil 4.1 Oluşturulan alçak geçiren FIR filtre	31
Şekil 4.2 FPGA tabanlı FIR filtre işlemcisine ait sistem.....	31
Şekil 4.3 FIR filtre işlemcinin blok diyagramı	32
Şekil 4.4 Matlab yardımıyla dizayn edilen filtre ve katsayıları.....	33
Şekil 4.5 FIR filtrenin alçak geçiren cevabı	33
Şekil 4.6 Toplama ve Çarpma Devreleri	34
Şekil 4.7 Filtre edilmemiş EEG sinyali	36
Şekil 4.8 1 ile -1 arasına normalize edilmiş EEG sinyali	36
Şekil 4.9 FIR filtre İşlemcisi tarafından filtre edilmiş EEG sinyali	36
Şekil 4.10 Giriş EEG sinyalinin tek taraflı genlik spektrumu	37
Şekil 4.11 Filtrelenmiş EEG sinyalinin tek taraflı genlik spektrumu.....	37
Şekil 4.12 0-100 Hz aralığında giriş ve çıkış sinyalinin spektrumları.....	38
Şekil 4.13 FIR filtre İşlemcisi tarafından filtrelenmiş EEG sinyalinin birinci bölümü.....	38
Şekil 4.14 Filtrelenmiş EEG sinyalinin heksadesimal gösterimi	39
Şekil 4.15 Filtrelenmemiş EEG sinyalinin x[0]-x[9] girişlerinin heksadesimal gösterimi	39
Şekil 4.16 Filtre tasarımı için oluşturulan sistem	40
Şekil 4.17 XSG DSP blokları ile FIR filtre tasarımı	41
Şekil 4.18 Kullanılan 4. Dereceden alçak geçiren filtre	42

Şekil 4.19 Katsayı aktarımı	43
Şekil 4.20 Filtre tanıtımı.....	43
Şekil 4.21 Filtrenmemiş normalize edilmiş sinyal	44
Şekil 4.22 Filtre edilmiş sinyal	44
Şekil 4.23 System generator blok penceresi.....	45
Şekil 4.24 Tasarlanan hwcosim bloğu	46
Şekil 4.25 Simulink model Gösterimi	46
Şekil 4.26 Donanım eşzamanlı simülasyon bloğu kullanılarak oluşturulan filtre modeli.....	47
Şekil 4.27 Filtre işlemi sonrası çıkış sinyali	48
Şekil 4.28 Kullanılan kaynağı hesaplama aracı.....	48



TABLÖLAR LİSTESİ

	<u>Numara</u>
Tablo 3.1 Çeşitli Xilinx ailelerine ait kaynak özellikleri .	18
Tablo 4.1. Aritmetik devrelerin kaynak kullanımı	35
Tablo 4.2. Dizayn edilen FIR filtrenin kaynak kullanımı	35
Tablo 4.3. Tasarlanan FIR filtrenin kaynak kullanımı	49
Tablo 4.4. Kaynak Kullanımı Karşılaştırması	49



KISALTMALAR LİSTESİ

FPGA	: Alanda Progranabilir Kapı Dizileri (Field Programmable Gate Array)
ASIC	: Uygulamaya Özgü Tümeşik Devre (Aplication Specified Integrated Circuit)
DSP	: Sayısal Sinyal İşleme (Digital Signal Processing)
VHDL	: Çok Hızlı Donanım Tanımlama Dili (Very High Hardware Description Language)
HDL	: Hızlı Donanım Tanımlama Dili (Very High Hardware Description Language)
PROM	: Programlanabilir Sadece Okunur Bellek (Programmable Read Only Memory)
RAM	: Rastgele Erişilen Bellek (Random Access Memory)
ISE	: Tümeşik Yazılım Ortamı (Integrated Software Environment)
IP	: Akıllı Özellik (Intelligent Property)
XSG	: Xilinx Sistem Generator (Xilinx System Generator)
FIR	: Sonlu Darbe Cevaplı (Finite Impulse Response)
IIR	: Sonsuz Darbe Cevaplı (Infinite Impulse Response)

1. GİRİŞ

Sayısal sinyal işleme teknikleri iletişim ve multimedya gibi bir çok uygulamada yaygın olarak kullanılmaktadır. Filtreler sayısal sinyal işleme alanında önemli bir kullanım alanına sahiptir. FIR (Sonlu Darbe Cevaplı) filtre ve benzerleri sayısal sinyal işleme fonksiyonları analog benzerleri ile karşılaştırıldığında bir çok avantaja sahiptirler. Pek çok elektronik cihazlarda sayısal filtreler kullanılmaktadır. Bu yüzden sayısal filtreler, simülasyonlarla gerçekleştirilmelerinin yanı sıra donanımsal olarak da gerçekleştirilebilirler. Son yıllarda, gerçek zamanlı ve yüksek başarı gerektiren işaret işleme uygulamalarının gerçekleştirmesinde FPGA (Alan Programlanabilir Kapı Dizileri) kullanımının arttığı görülmektedir. Başlangıçta sadece sayısal tasarımların kontrol edilmesi için geliştirilen FPGA çipleri, FPGA teknolojisindeki yeni gelişmelere paralel olarak, yüksek dereceli paralel çalışabilme kabiliyetine sahip olmuştur [1,2]. FPGA ile gerçekleştirilen tasarımlarda güç tüketimlerinin düşük ve performanslarının yüksek olduğu bilinmektedir. FPGA ile yapılan tasarımlar oldukça esnek olabilmektedir ve FPGA'ların bir işlemi gerçekleştirme hızları oldukça yüksektir. Ayrıca FPGA'lar ile aritmetik işlemler kolayca gerçekleştirilebilirler [3]. Literatürde sayısal filtre tasarımında FPGA'ların kullanılmasıyla ilgili birçok çalışma vardır. Yapılan bir araştırmada band geçiren filtre işlemini gerçekleştiren yeniden programlanabilir filtre bankası yazılımı geliştirilmiş ve tekrardan programlanabilir donanım olarak FPGA kullanmışlardır [4]. Cardarilli G. C. ve arkadaşlarının çalışmasında [5], çok oranlı filtre bankalarının FGPA ile tasarlanmasında üç çeşit algoritmanın kapsamlı karşılaştırmasını yapılmıştır. Elde edilen sonuçlara göre çok evreli yarım-band filtre bankasının yüksek verimli olduğuna karar verilmiştir. Lee H ve arkadaşlarının çalışmalarında [6], çeşitli basamak-seri DSP fonksiyonlarının FPGA ile gerçekleştirilmesinde, işlem gecikmeleri ve FPGA üzerinde kapladıkları alanlar üzerinde inceleme yapılmıştır. Gerçek zaman DSP fonksiyonları için FPGA'ların daha iyi performans gösterdiği görülmüştür. Anurag A, ve arkadaşlarının yaptığı çalışmada [7], XSG kullanılarak yüksek geçiren bir filtre gerçekleştirilmiştir. Çalışma sonucunda FPGA'nın paralel işlem kapasitesi sayesinde, sayısal filtre gerçekleştirmesindeki hızının büyük ölçüde arttığı gözlemlenmiştir. Emmanuel S. Kolawole ve arkadaşlarının yaptığı başka bir çalışmada ise [8], XSG kullanılarak yüksek geçiren, alçak geçiren ve band durduran üç çeşit FIR filtre ayrı ayrı tasarlanmış ve daha sonra hepsi birlikte aynı XSG düzleminde

tasarlanarak 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 kullanılarak tasarlanmış ve kaynak kullanımları bakımından karşılaştırılmıştır.

Tezin ilk bölümünde, öncelikle tezin amacı hakkında bilgi verilmiştir. Ayrıca FPGA ve XSG ile gerçekleştirilen daha önceki çalışmalara değinilmiştir. İkinci bölümünde filtreler, filtre çeşitleri ve sayısal filtreler hakkında bilgiler verilmiştir. Üçüncü bölümünde ise FPGA yapılarına değinilmiştir. Dördüncü bölümde yapılan uygulamalar ve sonuçları verilmiştir. Son bölümde ise elde edilen sonuçlar karşılaştırılmıştır ve gelecekteki çalışmalar için önerilerde bulunulmuştur.

2. FİLTRELER

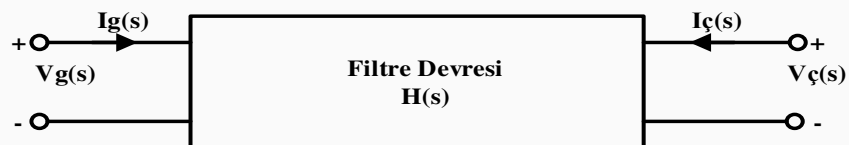
Filtreler günümüzde telefon, telgraf, radyo, radar, uydu ve haberleşme sistemleri, işaret işleme uygulamaları gibi birçok alanda kullanılmaktadır. Filtreler genellikle elektronik devrelerde, gürültüyü ve istenmeyen işaretleri süzmek, belirli frekansları birbirinden ayırmak, örnekleme işleminden önce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

Elektrik elektronik sistemlerde çok önemli elemanlar olan filtreler özellikle radyo, televizyon, ses, resim, ve data haberleşmesinde gerekli devrelerdir. Örneğin filtreler, telefon devreleri için olmazsa olmaz devre yapılarıdır, telefon devreleri filtresiz gerçekleştirilemezler. Ses ve hi-fi sistemlerde 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, rezonanstan kaçınmada veya rezonans oluşturmada, işaret biçimlendirmede, işaret zayıflatmada ve güç faktörü düzeltmede kullanılırlar [9].

Yapısal olarak filtreler analog ve sayısal veya aktif ve pasif filtreler olarak sınıflandırılabilir. Pasif R, L, C elemanları ile yapılan filtreler Pasif Filtre, transistör ve/veya op-ampların da kullanıldığı filtreler ise Aktif Filtre olarak adlandırılmaktadır.

2.1 Filtre Tanımı

İnsan zihni aynı anda sadece bir noktaya konsantre olabilmektedir. Çevremizde ne kadar çok uyarıcı olsa da kişi sadece bir noktaya dikkatini verebilmektedir. Elektronik sistemlerde çevrede ne kadar çok sinyal veya giriş bilgisi olsa da sistemler sadece bir tanesiyle işlem yapacak şekilde düzenlenmiştir. Bir işaretin frekans bileşenlerinin bir kısmını geçiren ve bir kısmını bastıran elektronik devrelere filtre denir. En genel anlamda 2 uçlu bir sistemde filtre devresinin gösterimi Şekil 2.1’de verilmiştir.



Şekil.2.1. Genel filtre devresinin görünümü

Şekil 2.1’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. Buna göre, Denklem 2.1 gerilim transfer fonksiyonu, Denklem 2.2 akım transfer fonksiyonu, Denklem 2.3 resistans transfer fonksiyonu ve Denklem 2.4 kondüktans transfer fonksiyonu olarak tanımlanır [9].

$$H(s) = \frac{V_{\phi}(s)}{V_g(s)} \quad (2.1)$$

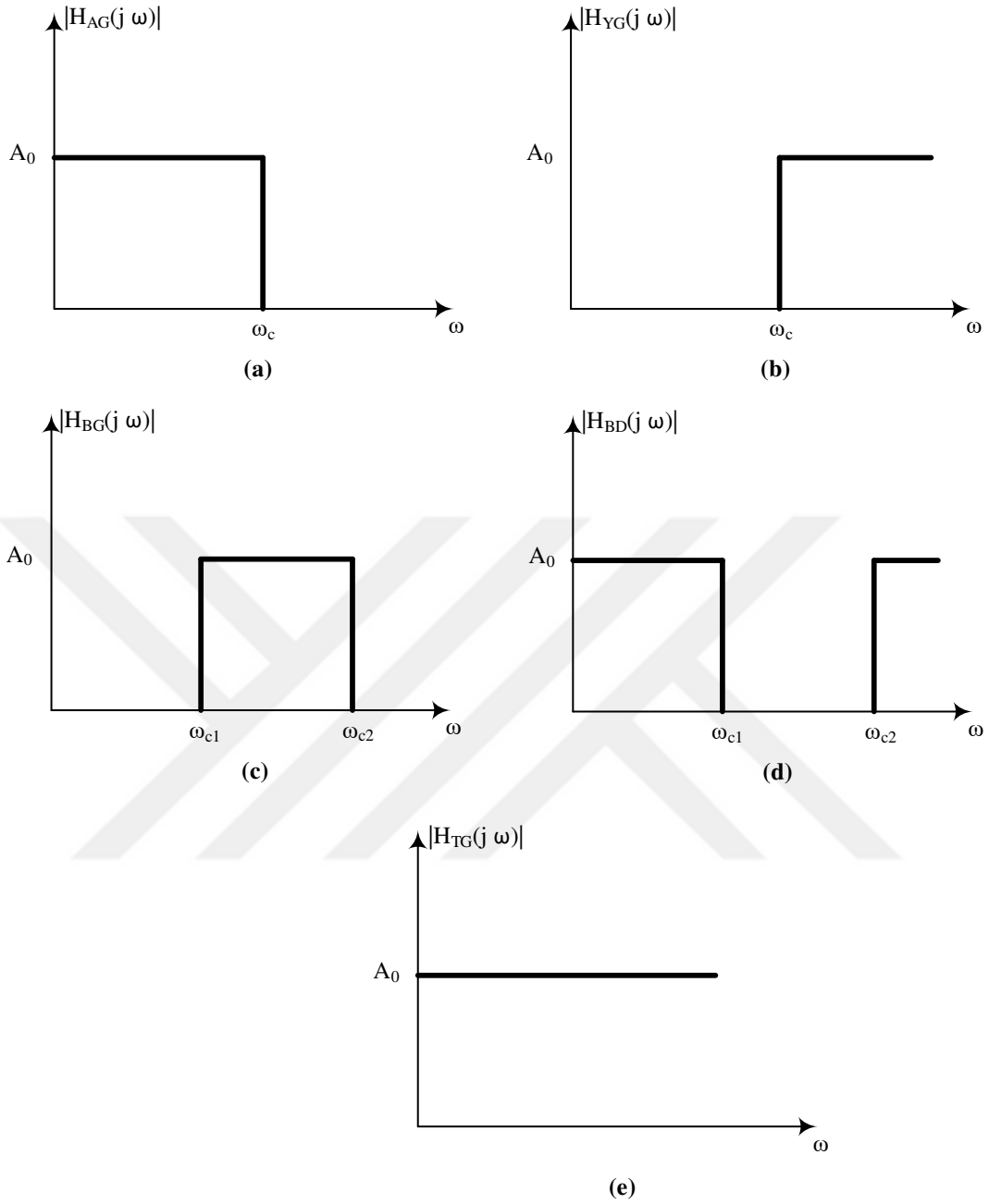
$$H(s) = \frac{I_{\phi}(s)}{I_g(s)} \quad (2.2)$$

$$H(s) = \frac{V_{\phi}(s)}{I_g(s)} \quad (2.3)$$

$$H(s) = \frac{I_{\phi}(s)}{V_g(s)} \quad (2.4)$$

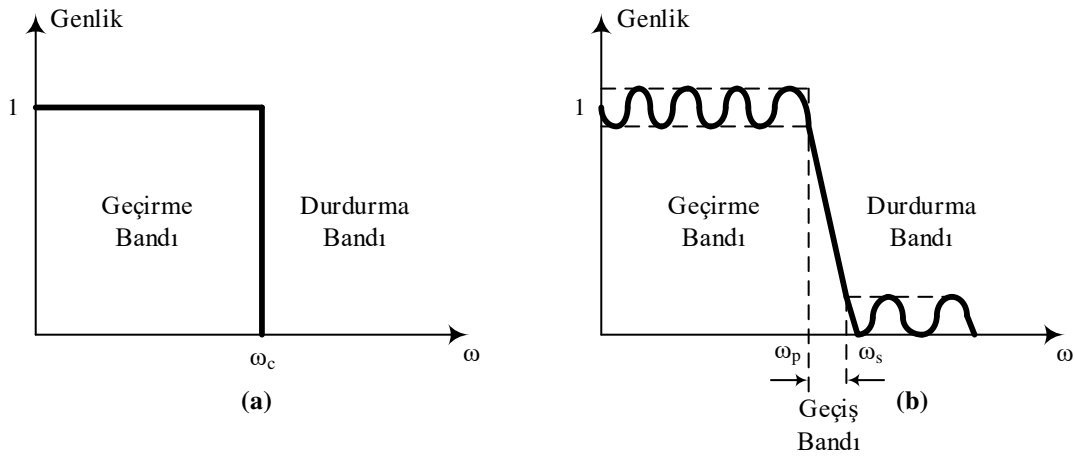
2.2. Filtre Türleri ve Temel Filtre Terimleri

Filtre devreleri, bastırdıkları ve geçirdikleri frekans bileşenlerinin bulunduğu bölgelere göre genel anlamda, alçak geçiren, yüksek geçiren, bant geçiren, bant durduran ve tüm geçiren filtreler olmak üzere beş gruba ayrılırlar. Bu beş filtre tipinin “ideal” genlik cevaplarına ait grafiksel gösterimler Şekil 2.2’de verilmiştir [10].



Şekil.2.2. Temel ideal filtre türlerinin genlik-frekans değişimleri, a) Alçak geçiren b) Yüksek geçiren c) Band geçiren d) Band durduran e) Tüm geçiren

Bir filtrenin girişte uygulanan işaretin frekansını tam olarak geçirdiği bölgeye filtrenin geçirme bandı, işaretin frekansını iletmeyip durdurduğu bölgeye de filtrenin durdurma bandı denir. Filtrelerin ideal karakteristiklerinde gözlemlenen bu keskin geçiş bandı ve durdurma bandı özellikleri uygulamalarda farklılık göstermektedir. Şekil 2.3 incelendiğinde teoride (idealde) ve uygulamada genlik cevabı verilen alçak geçiren filtre üzerinde bu farklılık dikkat çekecektir [10].



Şekil 2.3. Alçak geçiren filtre cevabı, a) İdeal b) Yaklaşık.

Şekil 2.3’de de görüleceği üzere alçak geçiren yaklaşık filtre cevabı, ideal cevaba ek olarak, geçirme ve durdurma bandının yanında, geçiş bandı da içermektedir. Bu sebeple burada filtre terimleri olarak ω_p , geçirme bandı ile geçiş bandını ayıran kesim frekansı (passband frequency) ve ω_s , durdurma bandı ile geçiş bandını ayıran kesim frekansı (stopband frequency) terimleri ile karşılaşılır [10].

Şekil 2.2’de görüldüğü gibi ideal cevaplar göz önüne alındığında tüm filtrelerin uygulanan işaretlerin frekanslarını kendi filtre karakteristiklerine göre bazı bölgelerde geçirdikleri, bazı bölgelerde de durdurdukları görülmektedir. Filtrelerin bu karakteristiğini belirleyen fonksiyon ise filtre transfer fonksiyonudur.

Analog filtre tasarımında kesim frekansı, merkez frekans, kalite faktörü, güç tüketimi, harmonik bozulma ve dinamik çalışma aralığı gibi parametreler filtrenin performans ölçütlerini belirlemektedir [10].

Doğrusal filtrelerde filtre karakteristiğine bağlı olarak girişe uygulanan işaretin belirli frekans aralıkları için geçirilmesine izin verilirken belirli aralıklar içindeki frekans bileşenleri bastırılır. Fakat çıkış işaretinde hala istenilen frekans bileşenlerinin bastırılmaması filtre cevabında harmonik bozulmaya neden olur.

Toplam Harmonik Bozulma (THD) harmonik frekans bileşenlerinin gücünün temel frekans bileşenlerinin gücüne oranına eşittir. Dinamik çalışma sahası ya da aralığı ortalama maksimum işaretin, işlenebilir işarete oranıdır. Pratikte, bir THD değeri için işlenebilir maksimum işaret genliği ortalama maksimum işareti belirtirken, verilen bir İşaret Gürültü Oranı (SNR) için işlenebilir minimum işaret genliği minimum işaret olarak bilinir [10].

Elektronik devrelerin transfer fonksiyonları devreye uygulanan giriş işareti ve devreden elde edilen çıkış arasındaki ilişkiyi ifade etmektedir. Bir filtre devresinin transfer fonksiyonunu farklı şekillerde ifade etmek mümkündür. Transfer fonksiyonları iki polinomun oranı şeklinde ifade edilebilirler. Denklem 2.5’de genel anlamda Laplace domeninde transfer fonksiyonu gösterimi verilmiştir.

$$H(s) = \frac{b_m s^m + b_{m-1} s^{m-1} + b_{m-2} s^{m-2} + \dots + b_0}{s^n + a_{n-1} s^{n-1} + a_{n-2} s^{n-2} + \dots + a_0} \quad (2.5)$$

Denklem 2.5 incelendiğinde burada m pay polinomunun derecesini, n payda polinomunun derecesini, $\{b_i\}_{i=0}^m$ pay polinomunun katsayılarını ve $\{a_i\}_{i=0}^n$ de payda polinomunun katsayılarını ifade etmektedir.

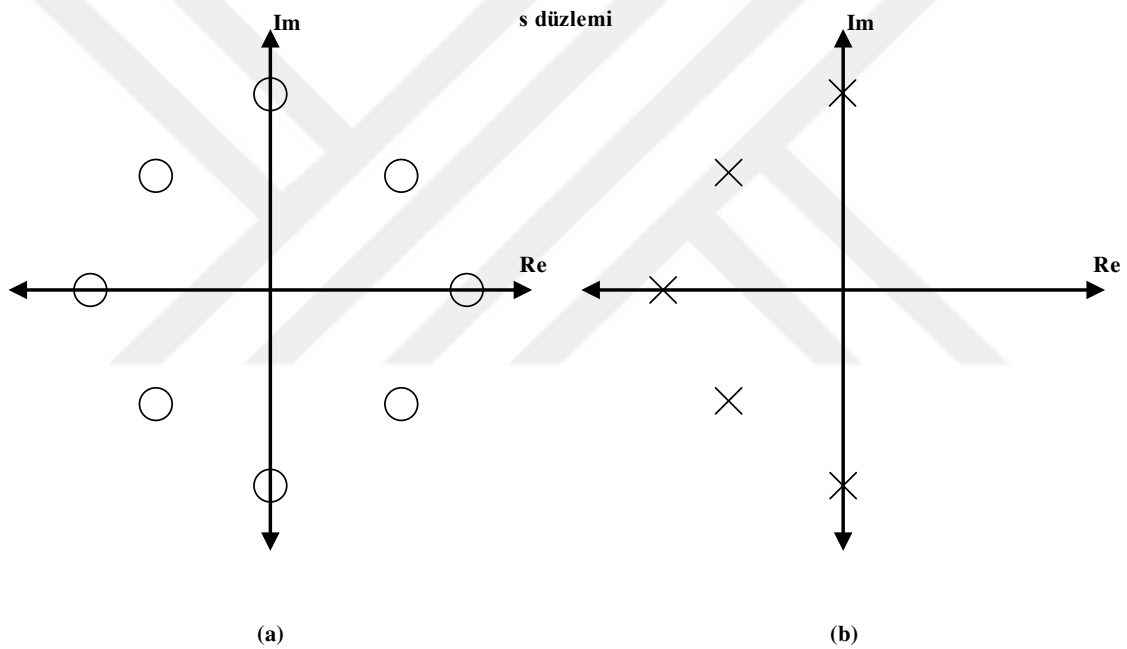
$$H(s) = \frac{b_m (s - z_1)(s - z_2)(s - z_3) \dots (s - z_m)}{(s - p_1)(s - p_2)(s - p_3) \dots (s - p_n)} \quad (2.6)$$

Denklem 2.6 transfer fonksiyonunun kutup sıfır gösterimi olarak adlandırılır. Bu gösterimde $z_1, z_2, z_3, \dots, z_m$ transfer fonksiyonunun sıfırlarını, $p_1, p_2, p_3, \dots, p_n$ ilgili fonksiyonun kutuplarını tanımlar. Filtre devrelerinin Denklem 2.5 veya 2.6’da verildiği gibi gerçekleştirilmesi için bazı temel koşulları sağlamaları gerekmektedir. Bu koşullar tasarlanan filtrenin analiz sonuçlarını etki eden koşullardır. Hatırlanacağı üzere filtrelerin gerçek ortamda ve teorideki davranışlarının birbirinden farklı olduğu daha önce ifade edilmişti. Bir filtrenin analiz sonuçları ideale ne kadar yakın çıkarsa filtre performansı da o kadar iyi demektir. Dolayısıyla filtre tasarımını etkileyen bu koşulların da filtre tasarımı yapılırken iyi analiz edilmesi gerekmektedir. Bunlar; kararlılık, nedensellik ve rasyonelliktir [11].

2.2.1 Kararlılık

Bir filtre devresinin kararlılık durumu incelenirken temelde sağlanması gereken iki koşul vardır. Bunlardan birincisi, filtre transfer fonksiyonunun payda polinomunun derecesinin pay polinomunun derecesinden büyük ya da eşit olmasıdır. Yani Denklem 2.5 ve Denklem 2.6’daki transfer fonksiyonları düşünüldüğünde, $m \leq n$ olmak zorundadır [12].

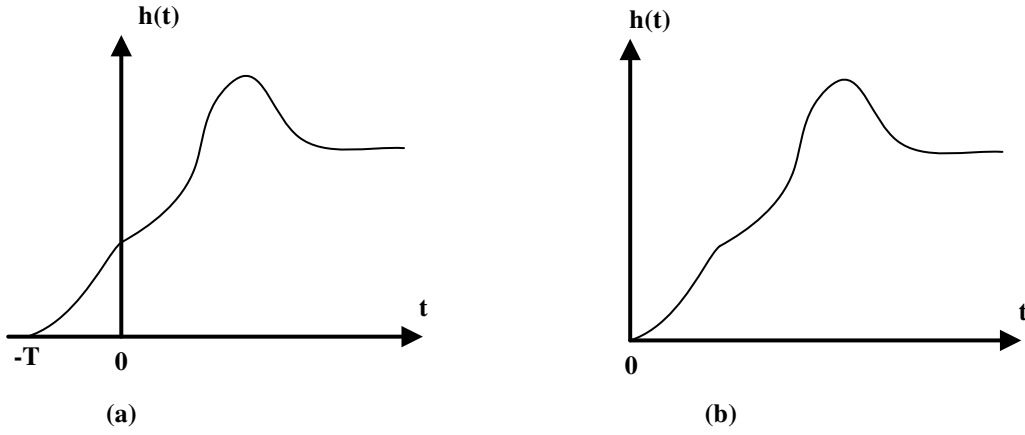
Bu kriteri sağlayan fonksiyon için ikinci kararlılık şartı incelenir. Bu şart için transfer fonksiyonunun kutup – sıfır grafiğine bakılır. Tasarlanan sistemin kararlı olabilmesi için sistemin tüm kutuplarının s-düzleminin sol tarafında yer alması gerekmektedir. Eğer kutupların tamamı s-düzleminin sol tarafında ise sistem kararlıdır, sağ tarafında kutup varsa kararsızdır ya da karmaşık eksen üzerinde katlı olmayan kutup veya kutuplar varsa sistem kritik kararlıdır. Filtre devrelerinin tasarımı yapılırken devrenin kararsız ya da kritik kararlı olması istenmez. Bu nedenle filtre tasarımı yapılırken bu kararlılık şartlarının sağlanmasına dikkat etmek gerekmektedir. Kararlılık şartlarını sağlayan bir filtrenin örnek kutup-sıfır dağılımı Şekil 2.4’de verilmiştir.



Şekil 2.4. Kararlı filtreye ait, a) Sıfır b) Kutup gösterimi

2.2.2 Nedensellik

Bir devrenin nedenselliğinin araştırılması için devrenin birim darbe cevabına bakılır. Eğer birim darbe cevabı $t < 0$ için, sıfıra eşitse sistem nedenseldir; değilse sistem nedensel değildir. Nedensel olan ve olmayan iki sisteme ilişkin darbe cevapları Şekil 2.5’de gösterilmektedir. Bu sistemlerden nedensel olan sistem gerçekleştirilebilirken, nedensel olmayan sistem gerçekleştirilemez.



Şekil 2.5. a) Nedensel olmayan b) Nedensel olan sisteme ilişkin darbe cevabı.

2.2.3 Rasyonellik

Bir devrenin gerçekleştirilebilmesi için bir diğer kriter de transfer fonksiyonunu Laplace domeninde rasyonel olmasıdır. Filtre transfer fonksiyonları genelde iki polinomun birbirine oranı şeklinde olduğundan rasyoneldir.

2.2.4. Analog Filtre İle Sayısal Filtrenin Karşılaştırılması

Analog filtreler ile sayısal filtrelerin birbirlerine göre avantaj ve dezavantajlarını sıralayacak olursak;

- Sayısal filtreler, analog filtrelerle mümkün olmayan karakteristiğe sahiptir yani gerçel olarak doğrusal faz cevaplıdır.
- Analog filtrelerin tersine sayısal filtreler ortam değişikliğinden etkilenmezler (örneğin sıcaklık değişimi).
- Sayısal filtrelerin frekans cevabı eğer programlanabilir bir işlemci kullanılarak gerçekleştirilirse (yani adaptif filtrelerde) frekans cevabı otomatik olarak ayarlanır.
- Sayısal filtrelerde, hem filtrelenmiş hem de filtrelenmemiş veriler daha sonra kullanılmak üzere saklanabilir.
- Sayısal filtrelerin bir diğer avantajı üretiminde VLSI teknolojisinde aşırı bir ilerlemenin olması ve çok küçük boyutlarda, düşük güç tüketiminde daha az ücretle temin edilebilmesidir.

- Sayısal filtreler çoğu biyomedikal uygulamalarda olduğu gibi düşük frekanslarla kullanılabilir. Ayrıca örnekleme frekansını değiştirmek suretiyle çok geniş frekans aralığında kullanılabilirler.

2.3 Sayısal Filtreler

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 yada 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 [13].

Sayısal işaret işlemenin amacı, bir işaretin frekans spektrumunu 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 impuls cevaplarına göre ikiye ayrılmaktadırlar. Bunlar FIR(sonlu darbe cevaplı) ve IIR(sonsuz darbe cevaplı) filtrelerdir [14].

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 2.7'deki gibi tanımlanır.

$$y(n) = \sum_{k=0}^{\infty} a^k \cdot x(n - k) \quad (2.7)$$

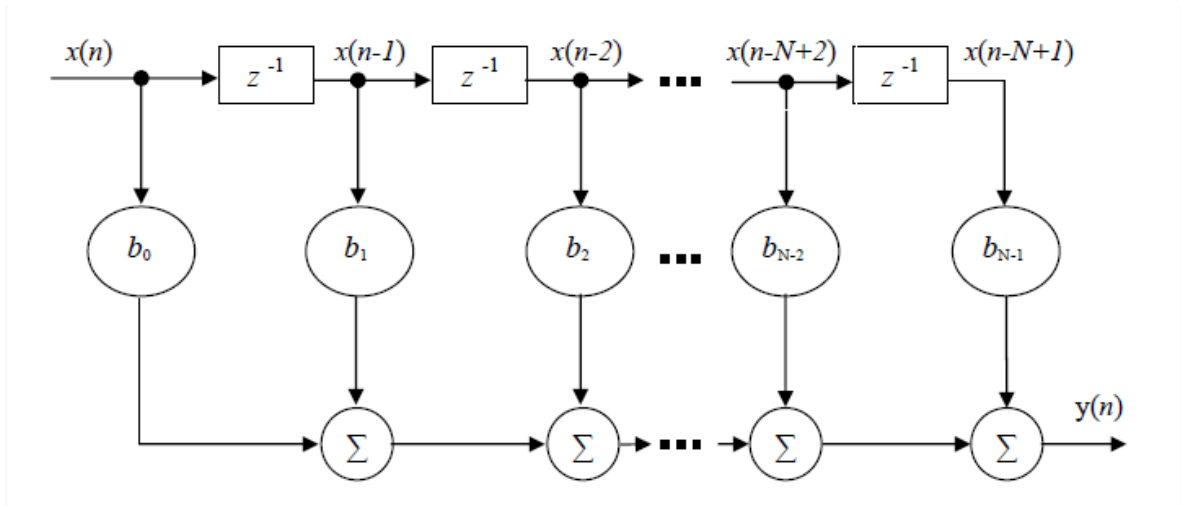
Bu denklem herhangi bir $x(n)$ girişi için $y(n)$ çıkışının hesaplanmasına izin vermesine rağmen, bu gösterim hesaplama açısından çok etkin değildir. Bazı durumlarda girişin mevcut ve geçmiş değerlerine ek olarak çıkışın da geçmiş değerler cinsinden ifade edilmesi mümkün olabilir. Örneğin önceki sistemin birinci dereceden ifadesi Denklem 2.8'deki gibi tanımlanabilir.

$$y(n) = a \cdot y(n - 1) \quad (2.8)$$

2.8’de verilen denklem doğrusal sabit katsayılı fark denklemi olarak bilinir.

2.3.1. FIR Filtre Yapısı

En yaygın olarak kullanılan FIR filtre yapısı Şekil 2.6 ile gösterilmiş olan ileri beslemeli filtre (Transversal Filter) yapısıdır [15]. İleri beslemeli filtre yapısı, dallı gecikme hatlı (Tapped-Delay line) filtre veya sonlu darbe cevaplı filtre olarak da adlandırılmaktadır.



Şekil 2.6. İleri beslemeli FIR filtre yapısı

Görüldüğü gibi 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 çıkışında $b_k x(n-k)$ çıkışı üretilecektir. Toplayıcılar ise, çarpıcı çıkışlarını birbiri ile toplayarak filtreye ait toplam çıkışı üretecektir. Yukarıda verilmiş olan FIR filtrenin giriş çıkış ilişkisi Denklem 2.9’daki gibi verilebilir,

$$y(n) = \sum_{k=0}^{N-1} b_k x(n-k) \quad (2.9)$$

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

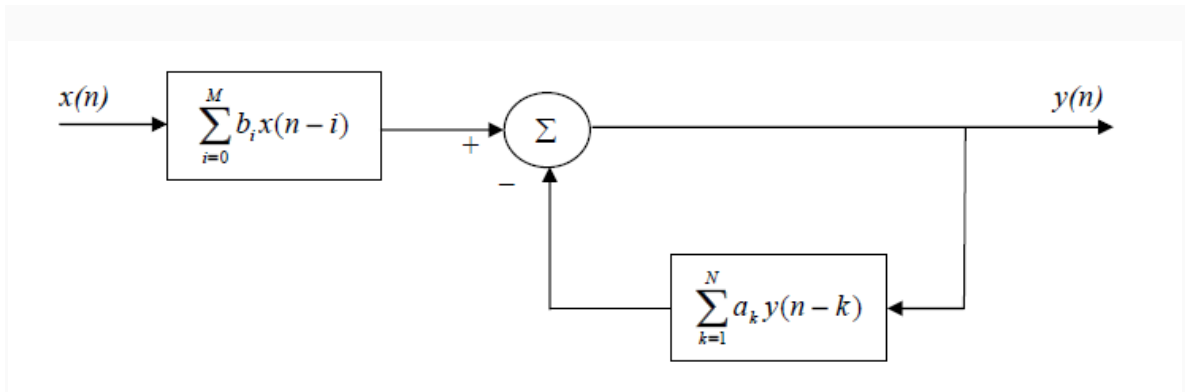
2.3.2. IIR Filtre Yapısı

Sonsuz darbe cevaplı filtreler ile sonlu darbe cevaplı filtreler arasındaki en temel fark, geri besleme katıdır. FIR filtrelere geri besleme katı ilave edildiğinde IIR filtreler elde edilirler. Bir FIR filtreye geri besleme katının ilave edilmesiyle oluşan IIR filtre yapısı Denklem 2.10 ile gösterilmiştir [16].

$$y(n) = \sum_{i=0}^M b_i x(n-i) - \sum_{k=1}^N a_k y(n-k) \quad (2.10)$$

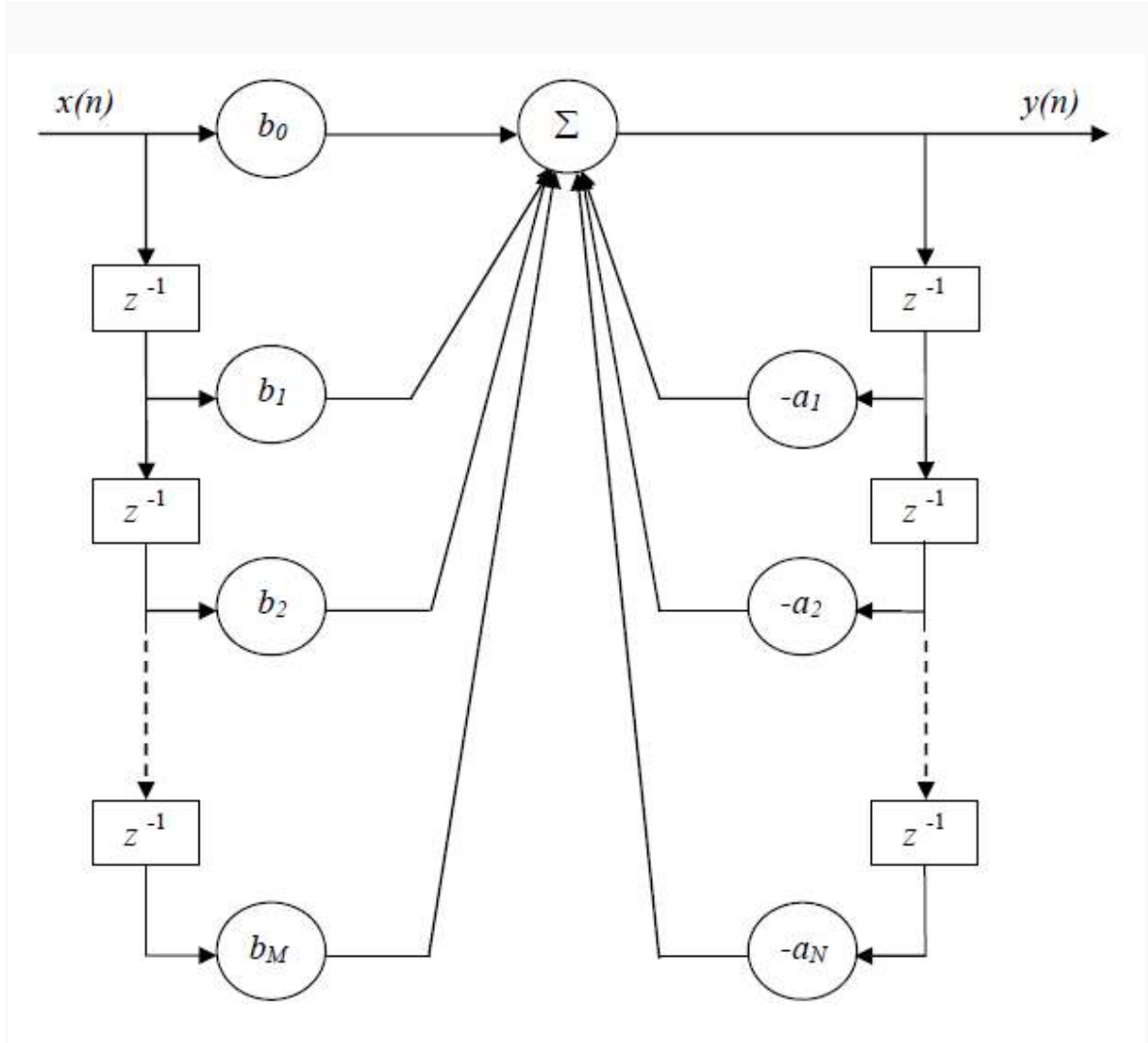
burada, a_k ve b_i filtre katsayılarını temsil etmektedir. $x(n)$ ve $y(n)$, sırasıyla, giriş ve çıkış işaretlerini, ($N \geq M$) ise filtrenin derecesini tanımlamaktadır.

IIR filtreler geri beslemeli yapıda oldukları için darbe cevapları sonsuz uzunluktadır. Geri beslemeli yapının getirdiği önemli bir problem ise, kararsızlıktır. Geri besleme katsayıları uygun bir şekilde seçilmediğinde, tasarlanan filtre kararsız olabilmektedir.



Şekil 2.7. FIR filtreye geri besleme katının ilave edilmesiyle oluşan IIR filtre yapısı

Literatürde en sık kullanılan IIR filtre yapılarından birisi olan Direkt Form IIR filtre yapısı Şekil 2.7 ile gösterilmiştir. Görüldüğü gibi, IIR filtre yapıları da birim geciktirme elemanı, çarpıcı ve toplayıcı olmak üzere üç temel elemandan oluşmaktadır. Şekil 2.8 ile verilmiş olan direkt form bir IIR filtre 2.10 fark denklemi ile tanımlanabilir,



Şekil 2.8. Direkt Form IIR filtre yapısı

Denklem 2.10.'da fark denklemleri ile tanımlanmış olan IIR filtrenin transfer fonksiyonu Denklem 2.11 gibi elde edilir,

$$H(z) = \frac{Y(z)}{X(z)} = \frac{\sum_{k=0}^M b_k z^{-k}}{1 + \sum_{k=1}^N a_k z^{-k}} \quad (2.11)$$

Yukarıda verilmiş olan denklemde pay polinomunun kökleri filtrenin sıfırları, payda polinomunun kökleri ise filtrenin kutupları olarak adlandırılır. Bir IIR filtrenin kararlı olabilmesi için tüm kutuplarının karmaşık düzlemde birim daire içerisinde yerleşmiş olması gerekir. Karmaşık düzlemde bir veya daha fazla kutbu birim dairenin dışında yerleşmiş olan IIR filtreler kararsızdırlar.

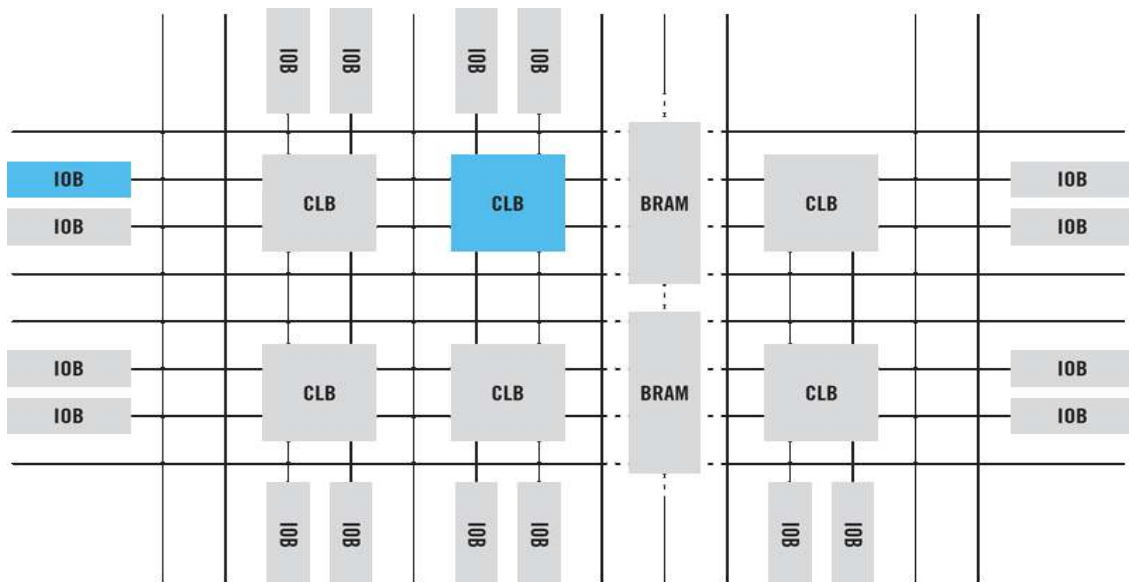
2.3.3. FIR (Sonlu Darbe Yanıtı) ve IIR (Sonsuz Darbe Yanıtı) Karşılaştırması

FIR ile IIR filtrelerinin karşılaştırılması yapılacak olursa özellikleri aşağıdaki gibi verilebilir.

- Teknik göstergelerde belirtilen aynı koşullar altında, IIR filtre çıkışının, girişe geri beslemesi vardır, böylece gereksinimleri FIR'den daha iyi şekilde karşılayabilmektedir. Depolama üniteleri, IIR'dekinden daha az olduğu gibi hesaplama sayısı da daha az olduğundan daha ekonomiktir.
- FIR filtresinin fazı, büyük ölçüde doğrusal iken IIR'ninki doğrusal değildir. IIR filtresinin seçiciliği ne kadar iyiye fazın değişimi o ölçüde doğrusal olmayan karakterdedir.
- FIR filtresi, özyinesiz bir yapıdadır ve sonlu kesinlik aritmetik hata oranı çok düşüktür. IIR filtresi, özyineli yapıda olup filtrenin kullanımı sırasında parazit salınımı meydana gelebilmektedir.
- Hızlı Fourier Dönüşümü (FFT), FIR filtresinde kullanılabilir iken IIR'de kullanılamamaktadır.

3. FPGA(ALANDA PROGRAMLANABİLİR KAPI DİZİLERİ)

Genellikle FPGA’lar sayısal mantığın, sistemlerde uygulanmasına yönelik bir alternatiftir. FPGA’lar herhangi bir sayısal tasarımı uygulamak üzere elektriksel olarak programlanabilen önceden üretilmiş 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 içermiştir. 64 CLB ve 58 giriş/çıkış içeren ilk FPGA’dan bu yana FPGA’lar karmaşıklık açısından son derece gelişmiştir. Günümüzün modern FPGA’sı, yaklaşık 330.000 mantık bloğu ile 1100 giriş ve çıkış içerebilmektedir. FPGA’nın temel mimarisi, üç ana bileşenden meydana gelmektedir: mantık fonksiyonlarını yerine getiren programlanabilir mantık blokları; bu fonksiyonları yerine getirmek üzere programlanabilir yönlendirme (ara bağlantı kurma) ve çip-dışı bağlantıları yapmak üzere I/O blokları. Genel bir FPGA mimarisinin resmi, Şekil 3.1’de gösterilmektedir [17].



Şekil 3.1. FPGA'nın iç yapısı.

3.1. Programlanabilir Mantık

Bir FPGA'daki programlanabilir mantık bloğunun amacı, sayısal sistemlerde kullanılan temel hesaplama ve depolama elemanlarını sağlamaktır. Temel mantık elemanı, belirli formlardaki programlanabilir bileşimli mantık, bir flip-flop devre ya da mandal ile alanı azaltmak ve maliyeti ertelemek üzere belirli hızlı taşıma mantığını içerir; bu, bütün bir işlemci olabilir. Çok sayıda modern FPGA, temel mantık bloğuna ilaveten, bazıları, özgül bellek blokları, çoğaltıcılar veya çoklayıcılar gibi sadece belirli fonksiyonlar için kullanılabilen heterojen bir farklı blok karışımı içermektedir. Şüphesiz, konfigürasyon belleği blok içindeki her bir elemana özgü fonksiyonu kontrol etmek üzere mantık bloğu boyunca kullanılmaktadır.

3.1.1. Programlanabilir Ara Bağlantı

FPGA'daki programlanabilir yönlendirme, kullanıcı tanımlı bir tasarımı tamamlamak üzere mantık blokları ve I/O blokları arasında bağlantılar sağlamaktadır. Programlanabilir yönlendirme, istenen bağlantıyı oluşturan çoklayıcılar, geçiş transistörleri ve üç durumlu tamponlardan oluşur. Genel olarak geçiş transistörleri ve çoklayıcılar, mantık elemanlarını bağlamak amacıyla bir mantık grubu bünyesinde kullanılırken üçü birlikte daha global yönlendirme yapıları için kullanılmaktadır. Ada bağlantı modeli ile hücresel, veriyolu tabanlı ve kodlu (ardışık düzenli) mimariler gibi çeşitli global yönlendirme yapıları FPGA'larda kullanılmıştır.

3.1.2. Programlanabilir I/O

Mantık blokları ile yönlendirme mimarilerini, çok çeşitli FPGA harici bileşenlerine arayüzlemek için gereken ortam veya vasıta, I/P pedleri veya programlanabilir I/O olarak adlandırılmaktadır. I/O pedi ve çevreleyen destekleyici mantık devre sistemi, bir I/O hücresini oluşturmaktadır. Bu hücreler, bir FPGA'nın önemli bir bileşenidir ve FPGA'ların alanını büyük ölçüde (%40) kullanmaktadır. Besleme gerilimi ve referans gerilim standartlarında büyük bir çeşitlilik bulunduğundan I/O programlanabilir bloklarının tasarımı zordur. I/O mimarisinin tasarımında en önemli kararlardan biri, desteklenecek

standartların seçimidir. Bu, dikkatli bir şekilde yapılan dengelemeleri içerir, çünkü I/O hücreleri, herhangi bir sayısal fonksiyonu uygulayabilen LUT'ların (taramalı tabloların) aksine genellikle tasarımcılar tarafından seçilen gerilim standartlarını uygulayabilir. Çok sayıda standardın desteklenmesi, I/O hücreleri için gereken silikon alanını büyük ölçüde artırabilir. Ayrıca, daha fazla sayıda standardı desteklemek amacıyla pin kapasitesi, daha fazla sayıda pin kullanılmasıyla artabilir ki bu durum, performansı sınırlandırabilir.

Temel FPGA mimarisi, daha özelleştirilmiş programlanabilir fonksiyonel blokların eklenmesiyle zamanla daha da gelişmiştir. Gömülü bellek (Blok RAM'ler), aritmetik mantık (ALU'lar), çoğaltıcılar, DSP-48 ve hatta gömülü mikroişlemciler gibi özel fonksiyon blokları, bir uygulama için bu kaynaklara sıkça ihtiyaç duyulduğundan ilave edilmiştir. Sonuç olarak çoğu FPGA, önceki FPGA'lara kıyasla heterojen bir kaynak karışımına sahiptir.

3.1.3. Donanım Tanımlama Dili

VHDL, Verilog, SystemC ve Handle-C gibi donanım tanımlama dilleri sıklıkla FPGA programlaması için kullanılmaktadır. VHDL ve Verilog endüstri standardı haline gelmiştir. Çok sayıda sağlayıcısı olan HDL'ler, simülasyon 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 kitaplıktır. Taban dili C++ olduğundan yazılım süreçleri, daha klasik bir HDL.2'dekine kıyasla daha kolay modellenebilmektedir. SystemC'ye yönelik sentez araçları ortaya çıkmakta olup VHDL ya da Verilog sentez ürünlerinin olgunluğuna yaklaşamamıştır. 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'ın yaptığı gibi süreçler arası iletişime yönelik bir iç yapı içerir.

3.1.4. Bir FPGA'nın seçimi

FPGA tabanlı sistem geliştirilirken bir gömülü sistem geliştiricisinin karşılaşılabileceği en kritik sorular şunlardır: belirli gömülü uygulama için uygun bir FPGA çipi nasıl seçilir? Tasarımına uymak üzere ne kadar büyüklükte bir cihaza ihtiyaç

duyacaktır? Gömülü sistem geliştiricisi tarafından, bir FPGA çipi seçilirken göz önünde bulundurulması gereken çok sayıda parametre ve özellik bulunmaktadır. Ancak gömülü sistem geliştiricisi tarafından, belirli bir uygulamaya yönelik olarak FPGA'ları karşılaştırmak ve seçmek üzere bir FPGA çipine ait özellikler incelenirken dikkate alınması gereken en önemli parametrelerden bazıları şunlardır: dilimler ya da mantık hücreleri gibi konfigüre edilebilir mantık blokları; çoğaltıcılar gibi sabit fonksiyon mantığı ile gömülü blok RAM gibi bellek kaynakları. Aşağıda verilen Tablo 3.1, çeşitli Xilinx aileleri bünyesindeki FPGA çiplerini karşılaştırmak için kullanılabilen kaynak özelliklerini göstermektedir.

Tablo 3.1. Çeşitli Xilinx ailelerine ait kaynak özellikleri

	Virtex- II 1000	Virtex- II 3000	Spartan- 3 1000	Spartan- 3 2000	Virtex- 5 LX30	Virtex- 5 LX50	Virtex- 5 LX85	Virtex- 5 LX110
Gates	1 million	3 million	1 million	2 million	-	-	-	-
Flip-Flops	10240	28672	15360	40960	19200	28800	51840	69120
LUTs	10240	28672	15360	40960	19200	28800	51840	69120
Multipliers	40	96	24	40	32	48	48	64
Block RAM (kb)	720	1728	432	720	1152	1728	3456	4608

Bir FPGA çipinin donatıldığı mantık geçidi sayısı, FPGA çipi büyüklüğü ve kapasitesini ASIC (Uygulamaya Özgü Tümlleşik Devre) teknolojisi ile karşılaştırmak için genellikle en fazla kullanılan ölçü olmuştur. Ancak, bu parametre, bir FPGA çipinin gerçekten donatıldığı özellikler ve bileşenleri tam olarak tarif etmez. İşte bu yüzden Xilinx, yeni Vertex-5 FPGA ailesi için eşdeğer sayıda mantık geçidinden bahsetmemiştir.

Xilinx'e göre FPGA cihazı kapasitesini tarif etmek üzere belirlenen üç ölçü bulunmaktadır: maksimum mantık geçitleri, maksimum bellek bitleri ve tipik geçit aralığı.

3.1.5. FPGA Uygulamaları

FPGA uygulamaları oldukça çeşitli ve geniş kapsamlıdır. Genel olarak bakacak olursak FPGA'nın ana uygulama alanları şunlardır: Sayısal Sinyal İşleme (DSP), video işleme, yazılım tanımlı telsiz, kontrol sistemleri mühendisliği, biyobilişim, uzay ve

savunma sistemleri, bilgisayarlı görme, konuşma tanıma ve işleme, tıbbi görüntüleme, bilgisayar donanımı emülasyonu, ASIC prototiplendirme, yeniden konfigüre edilebilir hesaplama ve radyo astronomi vb.

Genel olarak FPGA'lar, üretim hacminin az olduğu uygulamalarda ve gelişim kaynaklarında kullanılmaktadır ve bu düşük hacimli uygulamaya ait bir ASIC oluşturmak için ihtiyaç duyulan masraflar, yanına yaklaşılmayacak ölçüde yüksektir.

FPGA teknolojisinde elde edilen gelişme ile FPGA'ların uygulama alanları her geçen gün artmaktadır. FPGA'lar, bilhassa uygulamalar ya da FPGA mimarisi tarafından sunulan paralel işlemenin, yüksek performans sağlamak üzere kullanılabileceği algoritmaların uygulanması için uygundur. İç mimari ve mantık kaynakları tarafından sunulan yapısal paralellikten ötürü FPGA'lar düşük saat oranında bile yüksek çıktı verebilmektedir.

FPGA teknolojisinde elde edilen gelişme ve ileri gelişim araçlarının var olması ile FPGA uygulama alanları oldukça yüksek bir oranda artmakta ve bu teknolojinin ileride ASIC'in bile yerine geçmesi beklenmektedir [18].

3.1.6. FPGA Teknolojisinin Faydaları

Aşağıda FPGA teknolojisinin bize sağladığı 5 faydaya yer verilmiştir [19].

- **Performans:** Donanım paralelliği avantajından faydalanan FPGA'lar, ardışık yürütümü kırarak ve saat çevrimi başına daha fazlasını sağlayarak sayısal sinyal işlemcilerinin (DSP'ler) hesaplama gücünü geçmektedir. Bilinen bir analist ve karşılaştırma firması olan BDTI, nasıl FPGA'ların, bazı uygulamalarda her DSP çözümü doları için kat kat fazla işleme gücü sunabildiğini gösteren karşılaştırma çalışmaları yayınlamıştır. Donanım seviyesindeki kontrol girişleri ve çıkışları (I/O), uygulama gereksinimleriyle yakından uyumu sağlamak üzere daha hızlı yanıt süreleri ve özelleştirilmiş fonksiyonellik sağlamaktadır.
- **Pazara sürüm süresi:** FPGA teknolojisi, uzatılmış pazara sürüm süresi kaygıları karşısında esneklik ve hızlı prototiplendirme kabiliyetleri sunmaktadır. Özel ASIC tasarımına ait uzun fabrikasyon sürecini yaşamadan bir fikri ya da konsepti donanımda test edebilir ya da doğrulayabilirsiniz. Böylece haftalar yerine saatler içerisinde bir FPGA tasarımı üzerinde artımlı değişiklikler uygulayabilir ve bunları tekrarlayabilirsiniz. Kullanıcı tarafından programlanabilir bir FPGA çipine hal-i

hazırda bağılı olan farklı I/O tipleri ile ticari kullanıma hazır donanım da mevcuttur. Yüksek seviyeli yazılım araçlarının giderek artan kullanılabilirliği, soyutlama katmanları ile öğrenme eğrisini düşürmekte ve çoğunlukla ileri kontrol ile sinyal işlemeye yönelik değerli IP çekirdekleri (önceden oluşturulmuş fonksiyonlar) sunmaktadır.

- **Maliyet:** Özel ASIC tasarımının tekrarlanmayacak mühendislik masrafı, FPGA tabanlı donanım çözümlerinkini oldukça geçmektedir. ASIC'lere yapılan büyük başlangıç yatırımının, her yıl binlerce çip gönderen orijinal malzeme üreticileri için gerekçelendirilmesi kolaydır. Ancak çok sayıda son kullanıcı, geliştirilmekte olan onlarca-yüzlerce sisteme yönelik olarak özel donanım fonksiyonelliğine ihtiyaç duymaktadır. Programlanabilir silikonun tam niteliği, hiçbir fabrikasyon maliyetinizin ya da kurulum için uzun teslim sürenizin olmayacağı anlamına gelmektedir. Sistem gereksinimleri çoğunlukla zaman geçtikçe değiştiğinden FPGA tasarımlarına artımlı değişiklikler yapma maliyeti, bir ASIC'i yeniden döndürmeye ilişkin büyük masrafla karşılaştırıldığında göz ardı edilebilir.
- **Güvenilirlik:** Yazılım araçları, programlama ortamı sağlasa da FPGA devre sistemi, program yürütmenin tam olarak “zor” uygulamasıdır. İşlemci tabanlı sistemler çoğunlukla birden fazla işlemci arasında görevleri planlamaya ve kaynakları paylaşmaya yardımcı olmak üzere birkaç soyutlama katmanı içermektedir. Sürücü katmanı, donanım kaynaklarını kontrol ederken OS (işletim sistemi), bellek ve işlemci bant genişliğini yönetmektedir. Herhangi bir işlemci çekirdeği için bir seferde sadece bir talimat yürütülebilir ve işlemci tabanlı sistemler için devamlı birbirinin önüne geçmeye çalışan zaman açısından kritik görevler söz konusudur. OS'leri kullanmayan FPGA'lar tam olarak paralel yürütme ve her göreve ayrılan belirleyici donanımla güvenilirlik kaygılarını en aza indirmektedir.
- **Uzun vadeli bakım:** Daha önce de bahsedildiği gibi FPGA çipleri yerinde yükseltilebilir bir yapıya sahiptir ve ASIC'in yeniden tasarımı ile ilişkili süre ve masrafı gerektirmez. Örneğin sayısal iletişim protokolleri zamanla değişebilen özelliklere sahiptir ve ASIC tabanlı arayüzler, bakım ve ileriye dönük uyumluluk açısından zorluklara yol açabilmektedir. Yeniden konfigüre edilebilir FPGA çipleri, ihtiyaç duyulabilecek gelecek modifikasyonlara ayak uydurabilmektedir. Bir ürün

veya sistem olgunlaştıkça donanımı yeniden tasarlamak ya da devre kartı düzenini değiştirmek için zaman harcamadan fonksiyonel gelişmeleri uygulayabilirsiniz.

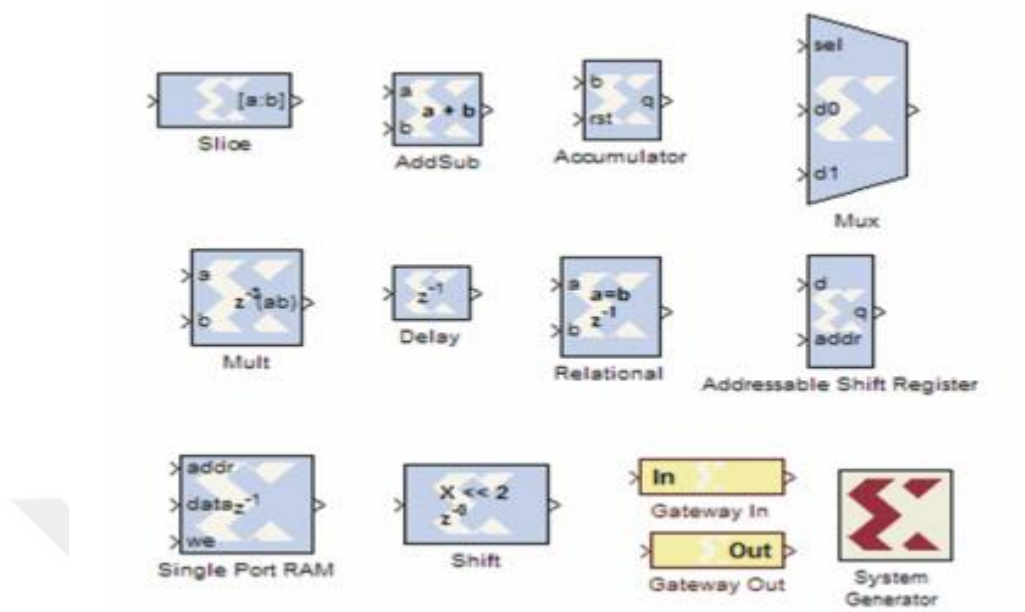
3.2 Xilinx System Generator

XSG çeşitli Xilinx FPGA'lara uygulanabilecek birkaç donanım işlemine yönelik bir dizi Simulink blokları (modelleri) sağlamaktadır. Bu bloklar, Simulink ortamını kullanarak donanım sisteminin fonksiyonelliğini simüle etmek üzere kullanılabilmektedir. Çoğu DSP (Sayısal Sinyal İşleme) uygulamasının niteliği, veri gösterimine yönelik kayan/gezer nokta formatı gerektirmektedir. Simulink gibi yüksek seviyeli modelleme yazılımını çalıştıran çeşitli bilgisayar sistemlerinde bunu uygulamak kolay olsa da donanım yazılımında bu, kayan/gezer nokta aritmetiğini 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 nokta formatı kullanmaktadır. System Generator, simülasyon ortamının (bizim durumumuzda bu Simulink'tir) yazılım tarafından ve donanım tarafından (System Generator blokları) sağlanan verileri dönüştürmek üzere bazı bloklar sağlamaktadır. Bu, XSG kullanılan tasarım sürecinde anlaşılması gereken önemli bir kavramdır [20].

System Generator, FPGA tasarımına yönelik olarak MathWorks model-tabanlı Simulink tasarım ortamının kullanılmasına olanak tanıyan Xilinx'ye 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 aşağı doğru tüm uygulama adımları, bir FPGA programlama dosyası oluşturmak üzere otomatik olarak gerçekleştirilmektedir [21].

3.2.1. Tasarım için kullanılan Xilinx System Generator Blokları

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 [21]. Şekil 3.2'de genel DSP yapı blokları verilmiştir.

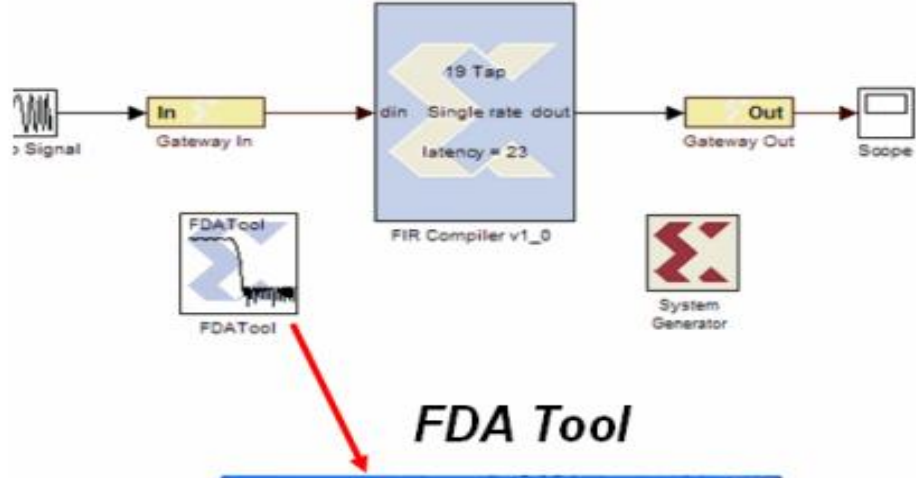


Şekil 3.2. DSP yapı blokları

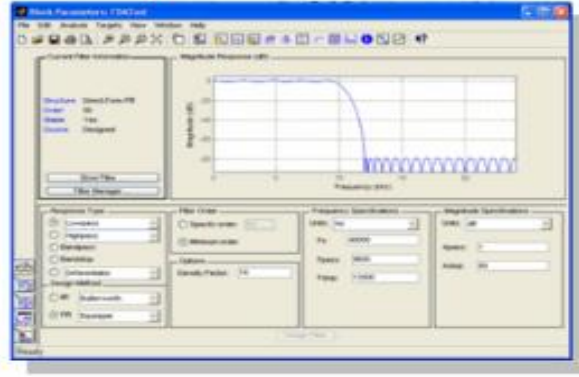
3.2.2. FIR Derleyici

System Generator, 500 Mhz'yi geçen bir değerde çalışabilen oldukça optimize uygulamalar oluşturmak üzere Virtex®-4 ve Virtex-5 cihazlarındaki özel DSP48 donanım kaynaklarına yönelik bir FIR Derleyici blok içermektedir. Konfigürasyon seçenekleri, direkt, çok-fazlı örnek seyreltme, çok-fazlı ara değerlendirme ve yüksek hızda örnekleme uygulamalarının oluşturulmasına olanak tanımaktadır. MathWorks FDatool veya fir2 gibi standart MATLAB fonksiyonları, Xilinx FIR Derleyici için katsayıları yaratmak üzere kullanılabilir [21]. Şekil 3.3'de FIR Derleyici bloğu gösterilmiştir.

FIR Compiler



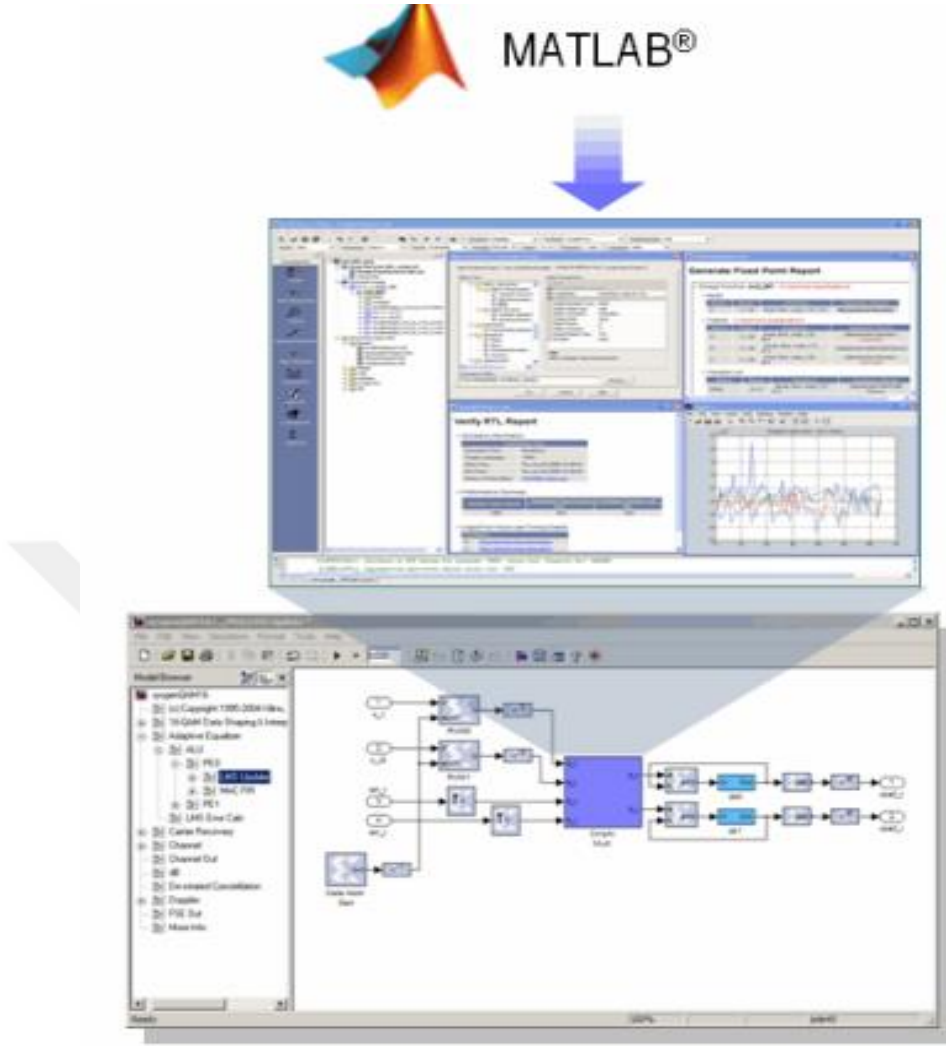
FDA Tool



Şekil 3.3. FIR derleyici

3.2.3. Matlab Desteği

Algoritmik MATLAB modelleri, AccelDSP™ aracılığıyla System Generator'a dahil edilebilmektedir. AccelDSP, kayan/gezer nokta MATLAB'i bir girdi olarak alan ve System Generator ile kullanılmak üzere tamamen planlanmış bir sabit nokta modeli oluşturan güçlü bir algoritmik sentez içermektedir. Özellikler, kayan/gezer noktadan sabit noktaya dönüştürmeyi kapsamaktadır. System Generator'de ayrıca basit kontrol işlemlerinin modellenmesi ve uygulaması için algoritmik olmayan MATLAB'in kullanımına olanak tanıyan bir MCode blok da yer almaktadır [21]. Şekil 3.4'de Mcode bloğu verilmiştir.

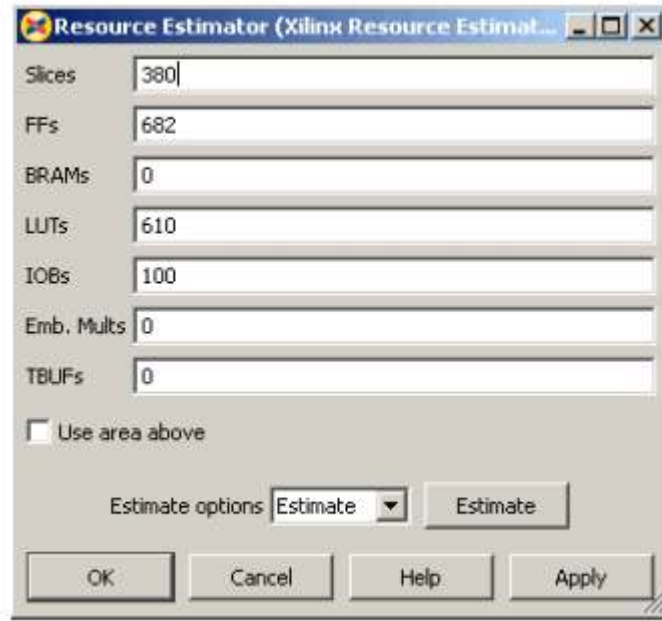


Şekil 3.4. Mcode bloğu

3.2.4. Sistem Kaynak Kestirici

Xilinx Kaynak Kestirici, bir Sistem Generator altsistemi ya da modelini uygulamak için gereken FPGA kaynaklarına ilişkin hızlı tahminler sağlamaktadır. Bu tahminler, Xilinx bloklarına yönelik olarak bloğa özel kestiricilerin çalıştırılması ve taramalı tablolar (LUT'lar), flip-floplar, blok bellekleri (BRAM), 18x18 çoğaltıcılar, üç durumlu tamponlar ve I/O'lara ait toplu tahminler elde etmek üzere bu değerlerin toplanması yoluyla hesaplanmaktadır. FPGA kaynakları gerektiren her bir Xilinx bloğu, kendi kaynak gereksinimlerini içeren bir vektör depolayan maske parametresine sahiptir. Kaynak Kestirici bloğu, bu vektörleri doldurma (yani parametreler ya da veri türleri değiştirildikten sonra) veya vektörlerde saklanan önceden hesaplanmış değerleri toplama gibi temel

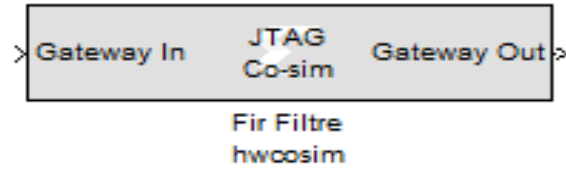
fonksiyonları çalıştırabilmektedir. Her bloğun kaynak kestirimine yönelik olarak, kestirici fonksiyonunu atlayıp bunun yerine vektörde saklanan tahminleri kullanan bir “FPGA alanını belirlemeye” yönelik bir işaret kutucuğu vardır. Bir kestirici blok, bir modelin herhangi bir alt sistemine yerleştirilebilir. Başka bir kestirici blok, bir kestiricinin altındaki hiyerarşide bulunduğu bloklar aşağıda belirtildiği gibi etkileşimde bulunmaktadırlar [24]. Şekil 3.5’de kaynak kestirici penceresi verilmiştir.



Şekil 3.5. Kaynak kestirici

3.2.5. Eşzamanlı Simülasyon

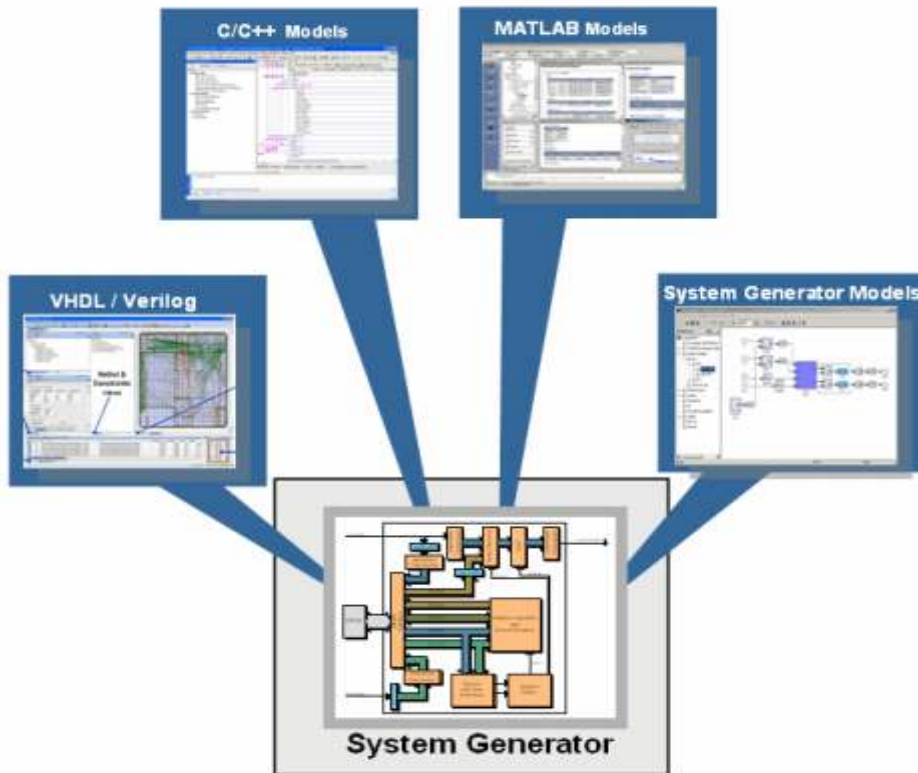
System Generator, donanım eş zamanlı simülasyon aracılığıyla hızlandırılmış simülasyon sağlamaktadır. System Generator otomatik olarak, 20’nin üzerinde desteklenen donanım platformundan biri üzerinde çalışacak Xilinx DSP blok setinde yakalanan bir tasarıma ilişkin olarak bir donanım simülasyon simgesi oluşturacaktır. Bu donanım, 1000x’e kadar bir simülasyon performansı artışı sağlamak üzere Simulink sisteminin geri kalan kısmı ile eş zamanlı olarak simüle edecektir [21]. Şekil 3.6 eş zamanlı simülasyon bloğunu göstermektedir.



Şekil 3.6. Eşzamanlı simülasyon bloğu

3.2.6. Sistem Entegrasyon Platformu

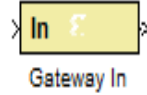
Sistem Generator, DSP FPGA'ların tasarımı ilişkin olarak, bir DSP sisteminin RTL, Simulink, MATLAB ve C/C++ bileşenlerinin tek bir simülasyon ve uygulama ortamında bir araya gelmesine izin veren bir sistem entegrasyon platformu sağlamaktadır. System Generator, RTL'nin, Simulink'e aktarılmasına ve ModelSim veya Xilinx® ISE® Simülatör ile birlikte simüle edilmesine olanak tanıyan bir kara kutu bloğu desteklemektedir. System Generator ayrıca C/C++ programlarını çalıştıran bir MicroBlaze™ gömülü işlemcinin dahil edilmesini de desteklemektedir [21]. Şekil 3.7'de sistem entegrasyon platformu verilmiştir.



Şekil 3.7. Sistem entegrasyon Platformu

3.2.7. Xilinx Geçit Giriş Bloğu

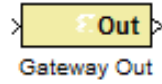
Xilinx Geçit Giriş Blokları (Xilinx Gateway In blocks), Simulink tasarımınızın Xilinx kısmına yönelik girişlerdir. Bu bloklar, Simulink tam sayı, çift ve sabit nokta verileri türlerini, Sistem Generator sabit nokta türüne dönüştürmektedir. Her bir blok, Sistem Generator tarafından üretilen HDL tasarımında üst seviye bir giriş portu tanımlamaktadır. Geçit Giriş Blokları, bir çift nokta türünü Sistem Generator sabit nokta türüne dönüştürürken, seçilen taşıma ve niceleme opsiyonlarını kullanmaktadır. Taşıma için opsiyonlar, en büyük pozitif/negatif değere doldurmak, kaydırmak (yani bitleri, en önemli gösterilebilir bitin soluna atmak) veya bir taşımayı, simülasyon sırasındaki bir Simulink hatası olarak işaretlemektir. Niceleme için opsiyonlar, en yakın gösterilebilir değere (ya da iki tane eşit uzaklıkta en yakın gösterilebilir değer var ise sıfırdan en uzak olan değere) yuvarlamak veya kırpmaktır (yani bitleri, en az önemli olan gösterilebilir bitin sağına atmak). Taşıma ve nicelemenin, donanımda meydana gelmediğini; donanım fazına geçmeden önce blok yazılımının kendisi içerisinde olduğunu fark etmek önemlidir [24]. Şekil 3.8’de geçit giriş bloğu verilmiştir.



Şekil 3.8. Geçit giriş bloğu

3.2.8. Xilinx Geçit Çıkış Bloğu

Xilinx Geçit Çıkış Blokları (Xilinx Gateway Out blocks), Simulink tasarımınızın Xilinx kısmından çıkışlardır. Bu blok, System Generator sabit nokta veri türünü, Simulink çift türüne dönüştürmektedir. Bu konfigürasyona göre Geçit Çıkış Blokları ya System Generator tarafından üretilen HDL tasarımında üst seviye bir çıkış portu tanımlayabilmekte ya da sadece donanım gösteriminden kesilecek bir test noktası olarak kullanılabilir [24]. Şekil 3.9’da geçit çıkış bloğu verilmiştir.



Şekil 3.9. Geçit çıkış bloğu

3.2.9. Xilinx Kayıt Bloğu

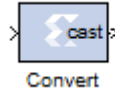
Xilinx Kayıt Bloğu (Xilinx Register Block), bir örnek dönemi kadar gecikme süresi olan D tipi bir flip-flop kayıt modellemektedir [24]. Şekil 3.10’da kayıt bloğu verilmiştir.



Şekil 3.10. Kayıt bloğu

3.2.10. Xilinx Dönüştürme Bloğu

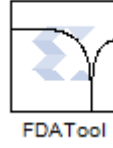
Xilinx Dönüştürme Bloğu (Xilinx Convert Block), her bir giriş örneğini, istenen bir aritmetik türde sayıya dönüştürmektedir. Örneğin, bir sayı, işaretli (ikiye tümleyen) veya işaretsiz bir değere dönüştürülebilir [24]. Şekil 3.11’de dönüştürme bloğu verilmiştir.



Şekil 3.11. Dönüştürme bloğu

3.2.11. Xilinx FDATool Bloğu

Xilinx FDATool bloğu, MATLAB Sinyal İşleme Araç Kutusunun bir parçası olarak bulunan FDATool yazılımına bir arayüz sağlamaktadır. Blok, Sinyal İşleme Araç Kutusu yüklü değilse doğru şekilde işlev göstermez ve dolayısıyla kullanılmamalıdır. Bu blok, bir FDATool objesi tanımlamaya ve bunu, bir System Generator modelinin bir parçası olarak saklamaya yönelik bir yol sağlamaktadır. FDATool, grafik kullanıcı arayüzü olan sayısal filtreleri tanımlamaya ilişkin güçlü bir araç temin etmektedir [24]. Şekil 3.12’de FDATool bloğu verilmiştir.



Şekil 3.12. FDATool bloğu

3.2.12. Xilinx JTAG Eş Zamanlı Simülasyon Bloğu

Xilinx JTAG Eş Zamanlı Simülasyon bloğu (Xilinx JTAG Co-Simulation block), JTAG ve bir Paralel Kablo IV ya da Platform USB kullanılarak donanım eş zamanlı simülasyon yapmanıza olanak tanımaktadır. JTAG donanımı eş zamanlı simülasyon arayüzü, simülasyon döngüsü kabiliyeti içerisindeki System Generator donanımını, diğer sayısız FPGA platformlarına genişletmek üzere JTAG'ın aynı anda birçok yerde bulunma özelliğinden faydalanmaktadır. Eş zamanlı simülasyon bloğunun port arayüzü değişiklik göstermektedir. JTAG donanımı eş zamanlı simülasyonu için bir model uygulandığında yeni bir kitaplık oluşturulur ki bu kitaplık, orijinal modelden geçit isimleri (ya da alt sistem üst seviyede değilse port isimleri) ile eşleşen portları olan bir özel JTAG eş zamanlı simülasyon bloğu içerir. Eş zamanlı simülasyon bloğu, bir Simulink simülasyonu sırasında FPGA donanım platformu ile etkileşimde bulunur. Bloğun giriş portlarına yazılan simülasyon verileri, blok aracılığıyla donanıma geçirilir. Buna karşılık, eş zamanlı simülasyon bloğunun çıkış portlarından veriler okunduğunda blok, donanımdan doğru değerleri okur ve bunların Simulink'te yorumlanabilmesi için çıkış portlarına iletir. Ayrıca, blok otomatik olarak platformu açar, konfigüre eder, yürütür ve kapatır [24]. Şekil 3.13'de JTAG Eş zamanlı simülasyon bloğu verilmiştir.



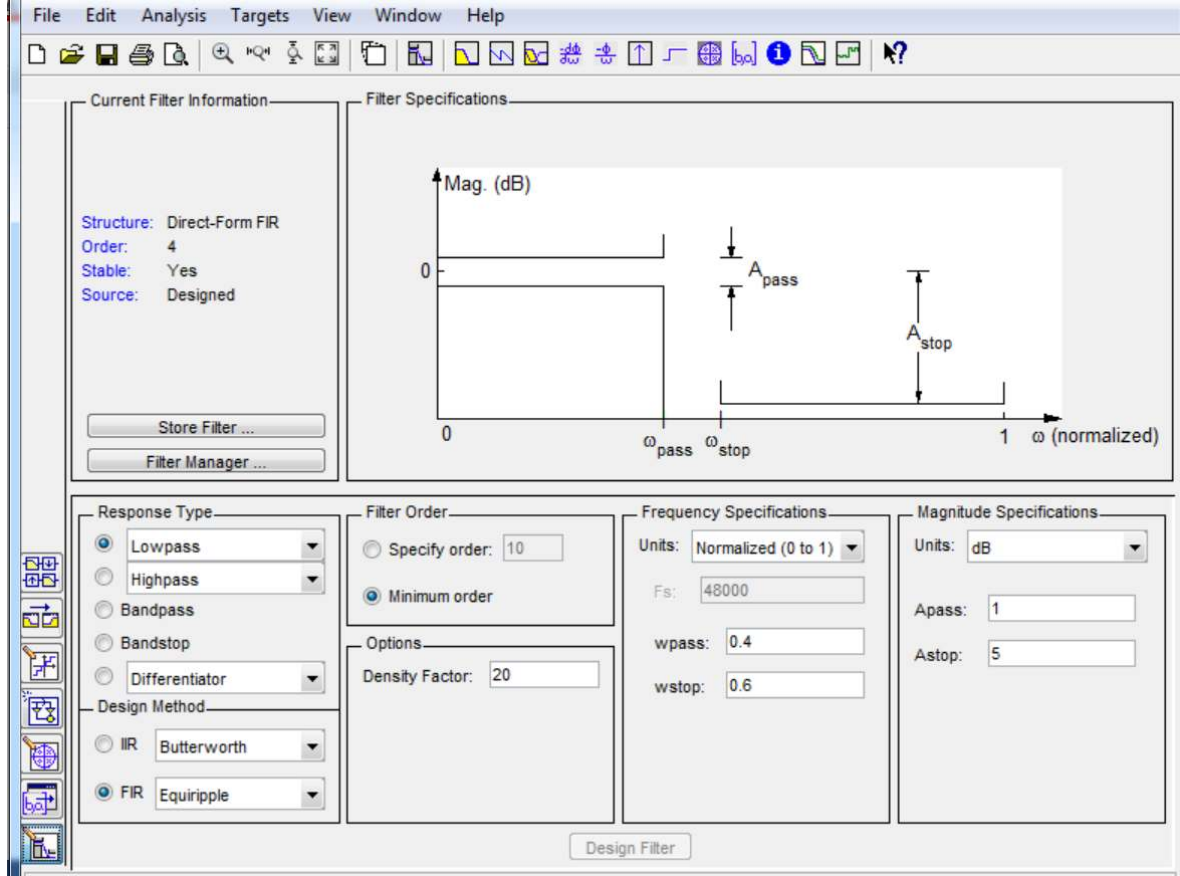
Şekil 3.13. JTAG eş zamanlı simülasyon bloğu

4.UYGULAMALAR

Bu bölümde yapılan uygulamalar gerçekleştirilmiş ve sonuçlar sunulmuştur. Uygulamalar iki bölümden oluşmaktadır. Her iki uygulamada da filtre olarak alçak geçiren 4.dereceden FIR filtre kullanılmıştır. Verilog dili kullanılarak tasarlanan sistem, Xilinx FPGA ISE editörü ve kodlamanın gerçekleştirildiği bir bilgisayardan oluşmaktadır. EEG sinyalinin filtrelemek için kullanılan FIR filtrenin katsayıları Matlab’da FDATool (filtre dizayn aracı) kullanılarak oluşturulmuş ve daha sonra FPGA’daki gömülü sisteme kodlanmıştır. XSG FPGA’da verimli sayısal sinyal işleme algoritmaları tasarımında kullanılır. XSG kullanılarak tasarlanan filtre ilk uygulamada tasarlanan filtrenin aynısıdır. Gerçekleştirilen filtre daha sonra Digilent Basys 2 kitinde yer alan Xilinx Spartan 3E-1600FG320 FPGA çipinde sentezlenmiştir.

4.1. FPGA Tabanlı Verilog Dili Kodlanarak Tasarlanan Filtre Uygulaması

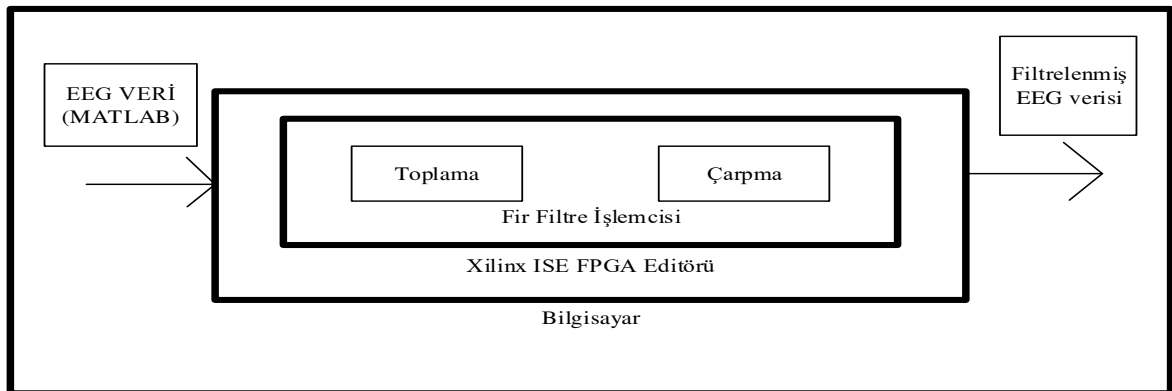
Son zamanlarda, hastanelerde ve biyomedikal cihaz tasarımlarında, hastalığı tanımlamak için bir çok yeni teknik geliştirilmiştir. Daha hızlı ve daha doğru sonuç veren bu teknikler çok büyük önem taşımaktadırlar. Bu çalışmada, bir EEG sinyalinin FPGA platformunda FIR filtre tasarlanarak işlenmesi gerçekleştirilmiştir. Tasarım Xilinx ISE FPGA editöründe Verilog yazılım dili kullanarak programlanmıştır. Filtrelenmek için kullanılan EEG sinyali [22] The International Federation of Clinical Neurophysiology tarafından kaydedilmiştir. Filtre tasarımı öncesinde filtre katsayılarını belirlemek için Matlab platformunda fdatool komutu kullanılarak 4. Dereceden bir Alçak geçiren FIR filtre tasarlanmıştır. Bahsedilen filtrenin FDATool arayüzü Şekil 4.1’de gösterilmiştir.



Şekil 4.1. Oluşturulan alçak geçiren FIR filtre

Tasarlanan filtrenin bazı parametreleri ω_{pass} :0.4, ω_{stop} :0.6, A_{pass} :1, A_{stop} :5 dB 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 4.2’de görülmektedir.



Şekil 4.2. FPGA tabanlı FIR filtre işlemcisine ait sistem

FIR Filtre işlemcisi toplama ve çarpma aritmetik devrelerinden oluşmaktadır. -1 ile 1 arasında normalize edilmiş EEG sinyali sabit nokta sayı gösterim formatına dönüştürülmüştür. Tasarımdaki herhangi bir sayının uzunluğu 16 bit seçilmiştir. Sayılar -1 ile 1 arasında kuantize edildiğinden dolayı, Qm.n sayı gösterme biçimi Q3.13 olarak düzenlenmiştir, burada m tamsayı kısmını ve n kesirli sayı kısmını temsil etmektedir.

Tasarımda tüm sayılar işaretli sayı olarak kullanıldığından, işaret biti için 1 bit ayarlanmıştır. Dolayısıyla, sayı gösterim biçimi son olarak Q2.13 16 bitlik işaretli sabit sayı, gösterim biçimi olarak MATLAB programında düzenlenmiştir. Bu durumda sayıların hassasiyeti $0,122 \times 10^{-4}$ olmuştur.

Bu çalışmada programlanmış model Şekil 4.3’de gösterilmektedir.

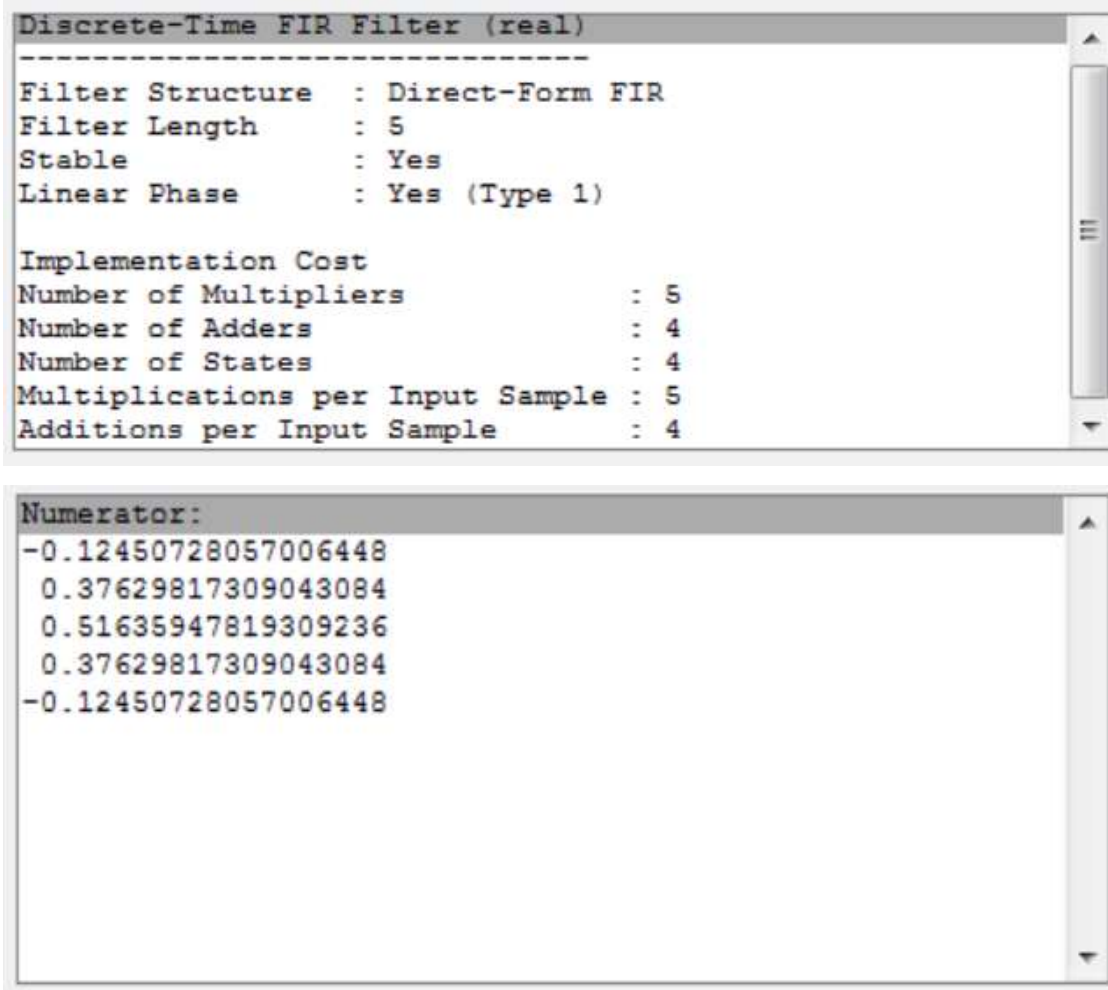


Şekil 4.3. FIR filtre işlemcinin blok diyagramı

Şekilde gösterildiği gibi, FIR filtre modeli katsayılarıyla birlikte bir dizi girişi ve bir dizi çıkışından oluşmaktadır. FIR filtre eşitliği Denklem 4.1’de gösterilmiştir.

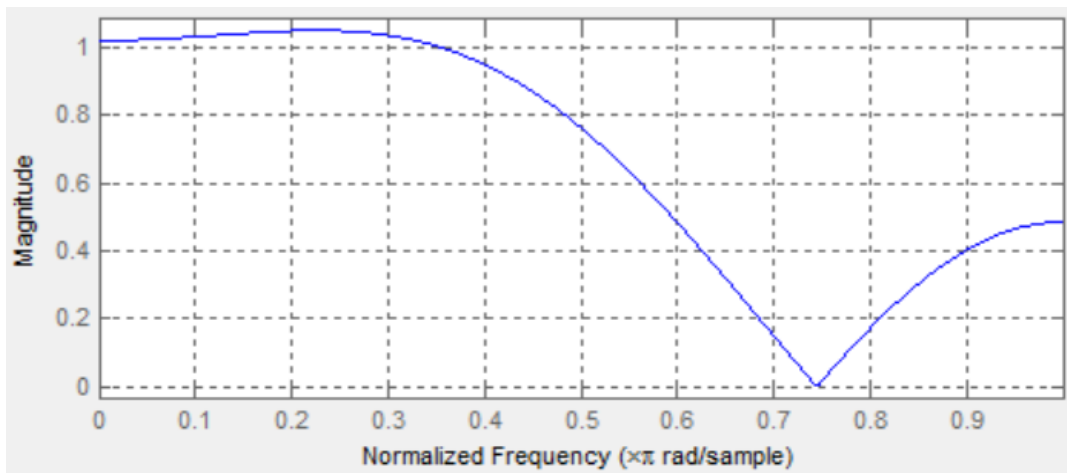
$$y(n) = \sum_{k=0}^4 a_k \cdot x(n - k) = a_0 \cdot x(n) + a_1 \cdot x(n - 1) + a_2 \cdot x(n - 2) + a_3 \cdot x(n - 3) + a_4 \cdot x(n - 4) \quad (4.1)$$

MATLAB’ta oluşturulan FIR filtre tasarımı ve katsayıları Şekil 4.4’de görülmektedir.



Şekil 4.4. Matlab yardımıyla tasarlanan filtre ve katsayıları

Matlab ortamında tasarlanan alçak geçiren FIR filtrenin genlik cevabı Şekil 4.5’de gösterildiği gibidir.



Şekil 4.5. Alçak geçiren FIR filtrenin genlik cevabı

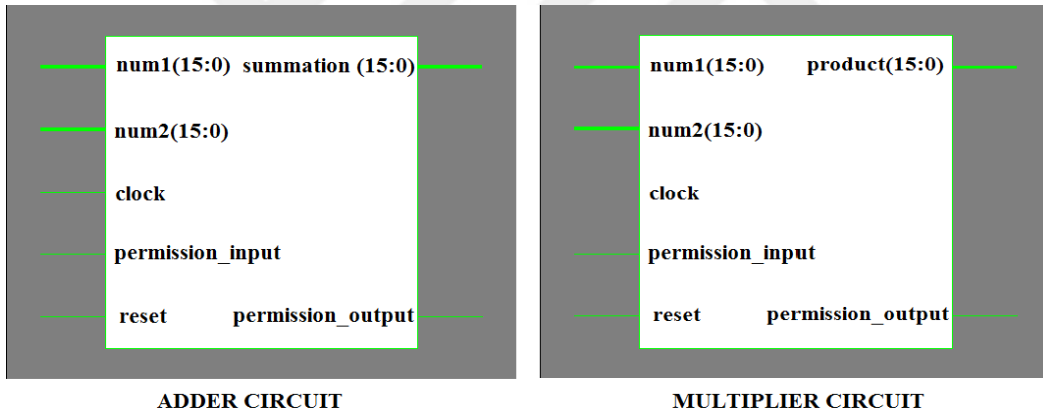
Filtre derecesi 4 olduğundan dolayı Filtrenin 5 katsayısı bulunmaktadır. Katsayılar tam değer olacak şekilde

$$\alpha_0 = -0.125, \alpha_1 = 0.375, \alpha_2 = 0.515, \alpha_3 = 0.375, \alpha_4 = -0.125$$

seçilmiştir.

Programlama dili olarak FPGA editöründe Verilog seçilmiştir. 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. Bu yüzden filtreleme tasarımı, her çalıştırmada 10 giriş verisi işleyebildiğinden toplamda 16 kez çalıştırılmıştır.

Toplama ve Çarpma Devreleri Xilinx ISE FPGA editöründe programlanmıştır ve 2 saat çevrimi süresince cevap vermektedirler. Şekil 4.6 bu aritmetik devrelerin girişlerini ve çıkışlarını göstermektedir [23].



Şekil 4.6. Toplama ve Çarpma Devreleri

Bu aritmetik devreler *permission_input* sinyallerle çalışırlar ve kombinasyonel olarak toplama ve çarpma operasyonlarını gerçekleştirirler. Toplama veya çarpma operasyonundan sonra, aritmetik devreler, devrelerin yeni hesaplamaya hazır olup olmadığını belirlemek için kontrol sinyalleri gönderirler. Toplama ve çarpma devreleri, sabit sayı aritmetiği İşlemci uygulamalarında kullanılmak için özellikle dizayn edilmişlerdir [23].

İşlemcide dizayn edilmiş aritmetik devrelerin kaynak kullanımları ve gecikmeleri Tablo 4.1 de görünmektedir. Tablo 4.2 de ise tasarlanan FIR filtre işlemcisinin kaynak kullanımları görülmektedir.

Tablo 4.1. Aritmetik devrelerin kaynak kullanımı

Aritmetik Devreler	Kullanılan Dilimler	Kullanılan FFs	Kullanılan LUTs
Toplama	9	1	18
Çarpma	35	20	61

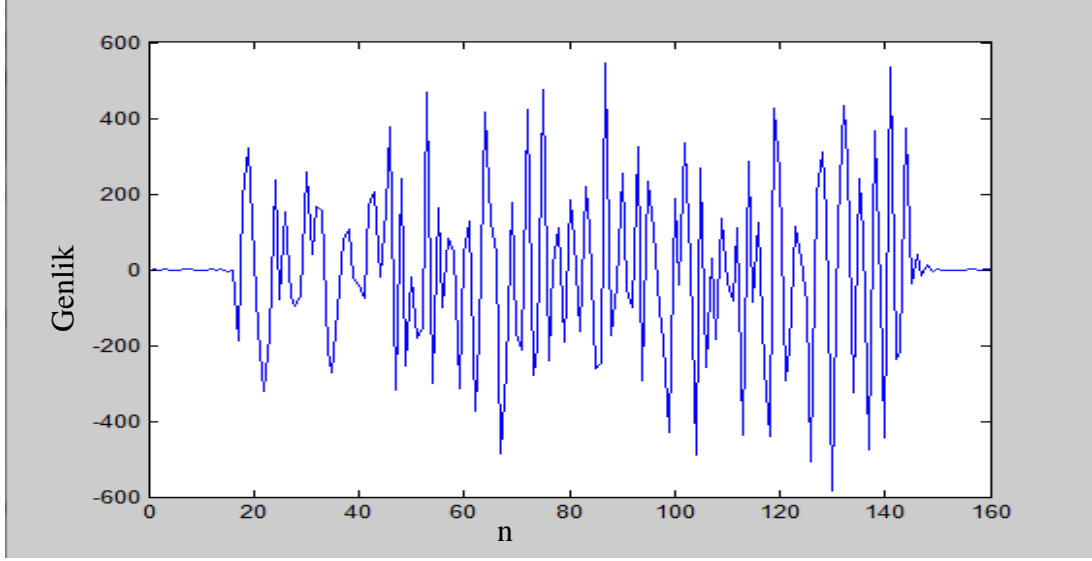
Tablo 4.2. Dizayn edilen FIR filtrenin kaynak kullanımı

Aritmetik Devreler	Kullanılan Dilimler	Kullanılan FFs	Kullanılan LUTs
FIR Filtre	12145	14523	21785

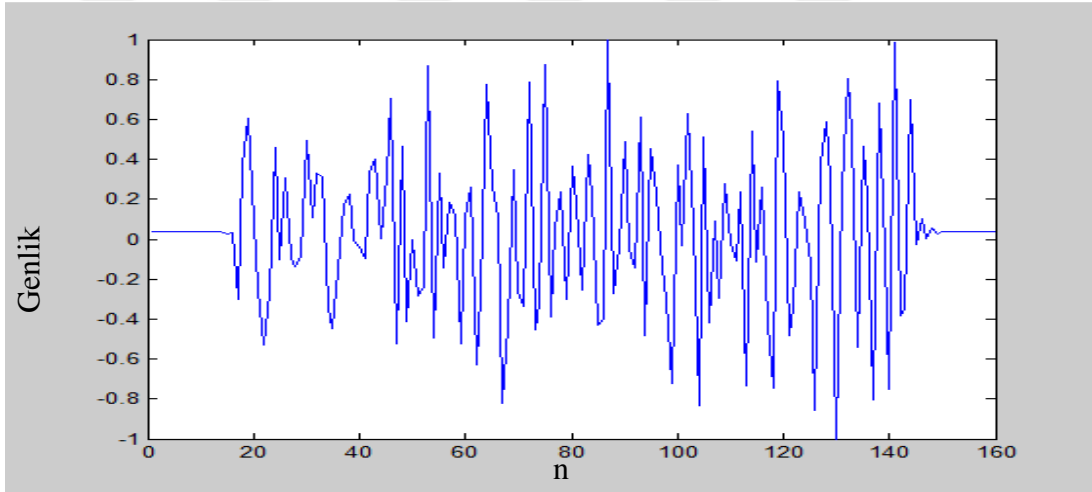
FIR Filtre İşlemci modeli Xilinx ISE FPGA Editöründe programlanmıştır ve Isim Simulator ile simülasyonu gerçekleştirilmiştir. Simülasyonun saat çevrimi 100 Mhz olarak ayarlanmıştır. Sinyal işleme tasarımını gerçekleştirmesi için toplam saat çevrimi sayısı 194'tür. Bu yüzden filtreleme işleminin tamamlanması 1940 ns sürmüştür.

Tasarlanan FIR Filtre İşlemci modelinde, kullanılan EEG verisi MATLAB'tan işlemciye bölüm bölüm transfer edilmiştir. Her bölüm 14 $x(n)$ girişine sahiptir. Bu veri işlemciye transfer edilmiş ve daha sonra işlemci tarafından filtre edilmiştir. Filtreleme işlemi sonrasında yeni giriş verisi tekrar filtreleme işlemi için işlemciye uygulanmıştır. Gerçekleme sonucunda bütün filtrelenmiş veriler yazdırılmak için Matlab'a transfer edilmiştir.

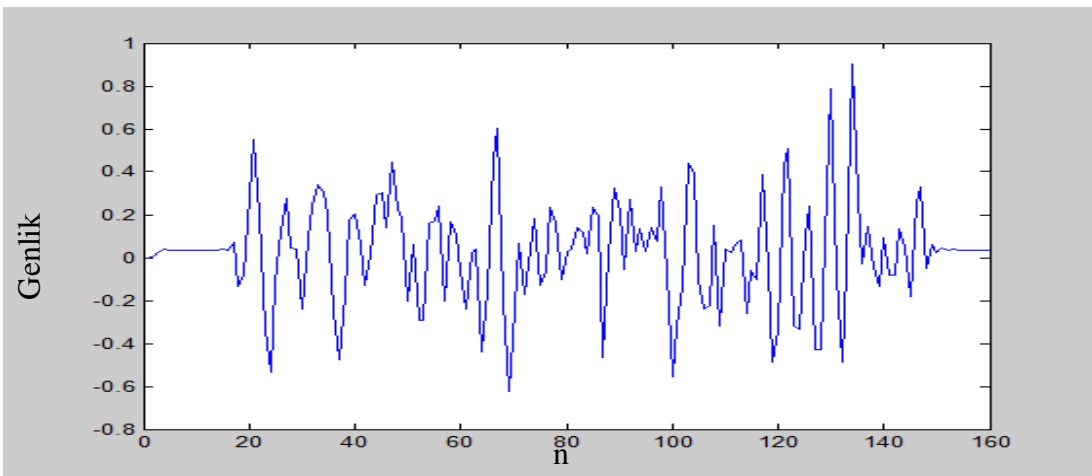
Şekil 4.7 filtre edilmemiş EEG sinyalini, Şekil 4.8 ise Matlab'da normalize edilmiş EEG sinyalini göstermektedir. Şekil 4.9'da tasarladığımız FIR Filtre işlemcisi tarafında filtrelenmiş EEG sinyali verilmiştir.



Şekil 4.7. Filtre edilmemiş EEG sinyali

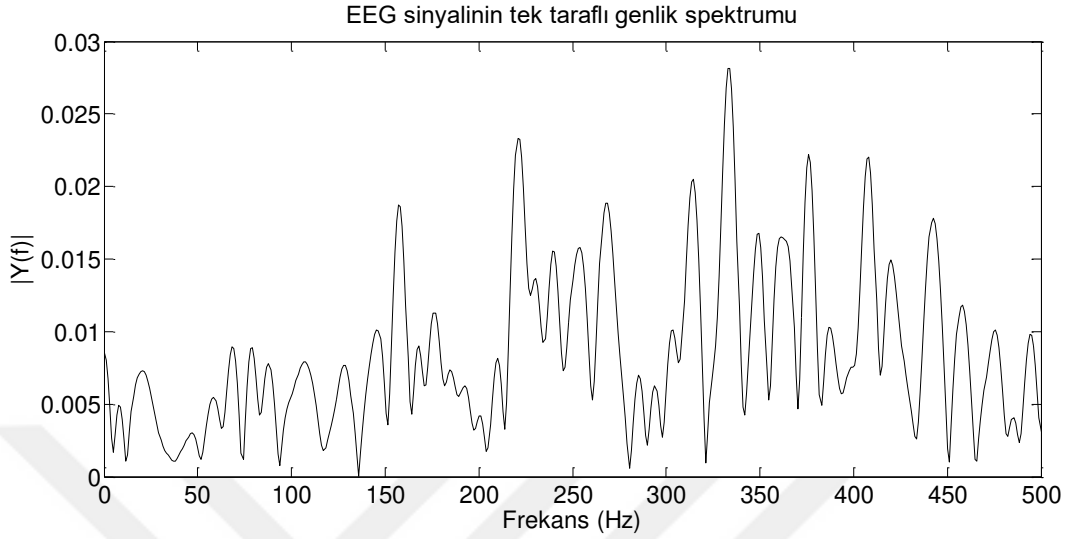


Şekil 4.8. 1 ile -1 arasına normalize edilmiş EEG sinyali



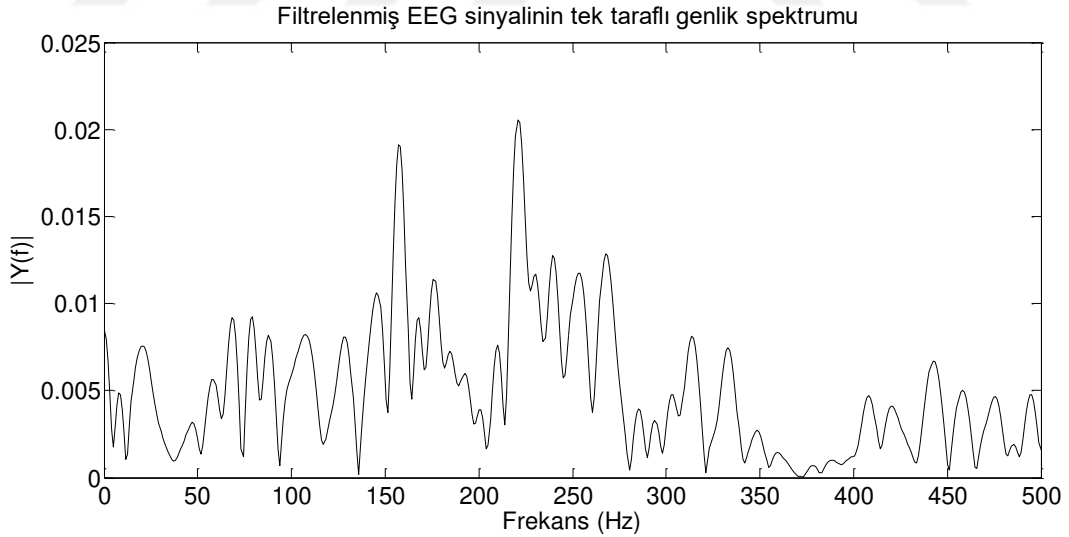
Şekil 4.9. FIR filtre İşlemcisi tarafından filtre edilmiş EEG sinyali

F_s örnekleme frekansı 1000 Hz alınarak orijinal EEG sinyalinin tek taraflı genlik spektrumu Şekil 4.10'da verilmiştir.



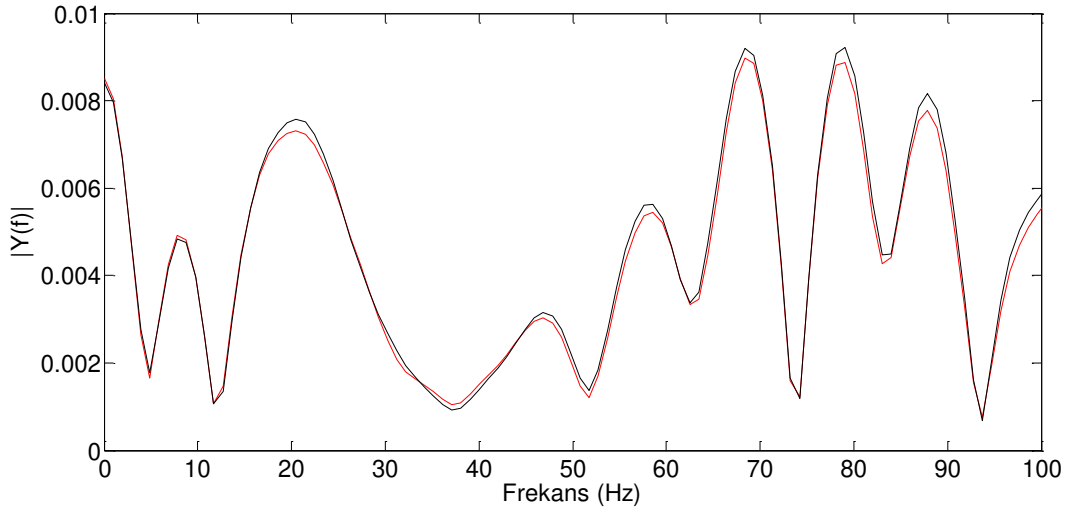
Şekil 4.10. Giriş EEG sinyalinin tek taraflı genlik spektrumu

Şekil 4.11'de filtrelenmiş EEG sinyalinin tek taraflı genlik spektrumu verilmiş olup yüksek frekans değerlerinde genliğin bastırıldığı görülmektedir.



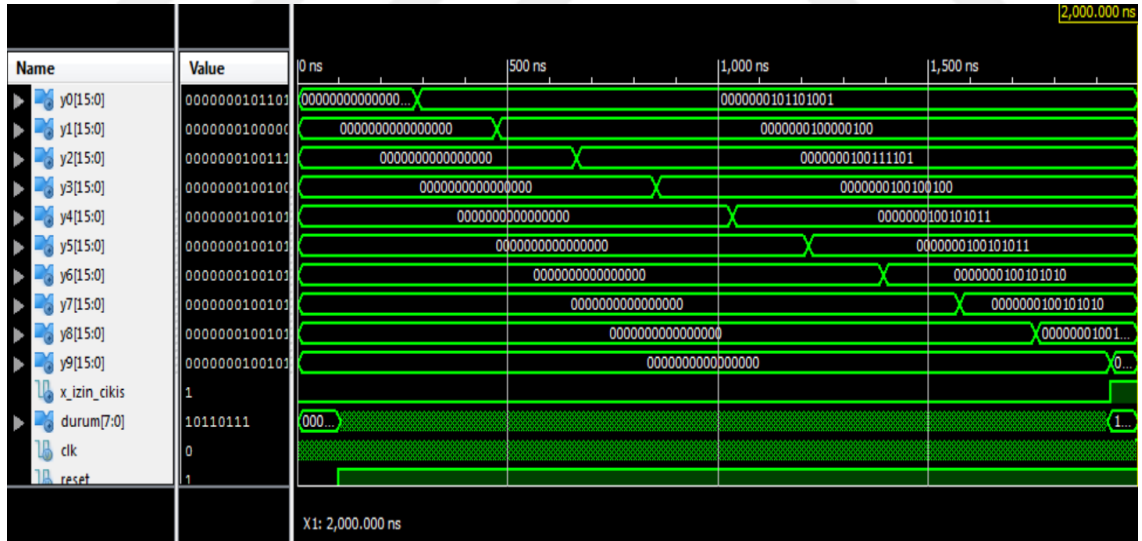
Şekil 4.11. Filtrelenmiş EEG sinyalinin tek taraflı genlik spektrumu

Düşük frekans değerlerinde ise tasarladığımız alçak geçiren 4. Derece FIR Filtre özelliğinden dolayı giriş sinyaline oldukça yakın bir sinyal elde edilmiş olup Şekil 4.12'de siyah giriş sinyalini kırmızı ise çıkış sinyalini gösterecek şekilde 0-100 Hz frekans aralığında çizdirilerek neredeyse aynı genlikte oldukları gözlemlenmiş ve filtre işlemi başarıyla gerçekleştirilmiştir.

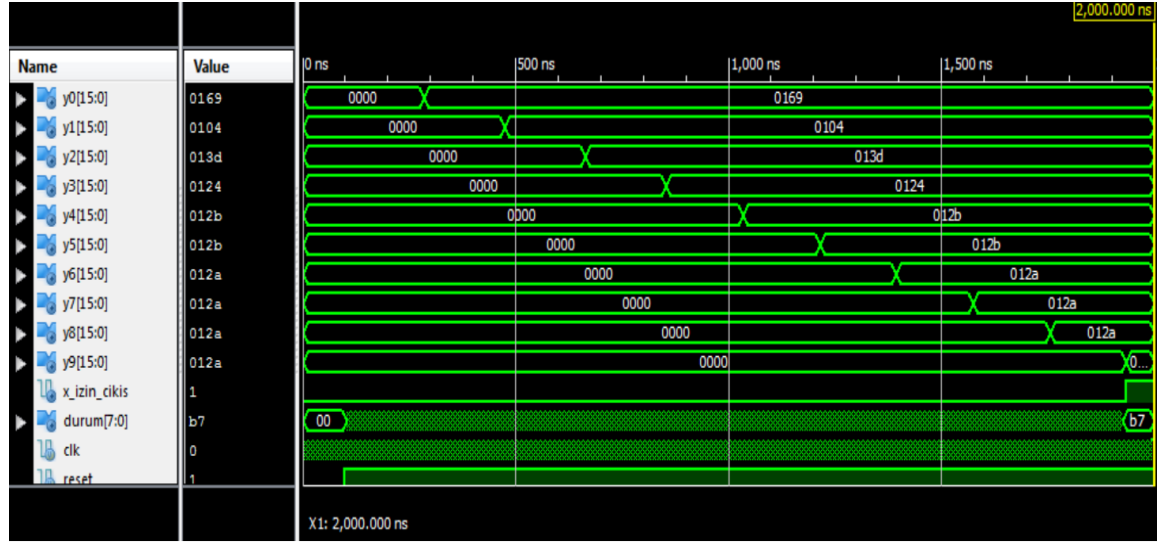


Şekil 4.12. 0-100 Hz aralığında giriş ve çıkış sinyalinin spektrumları

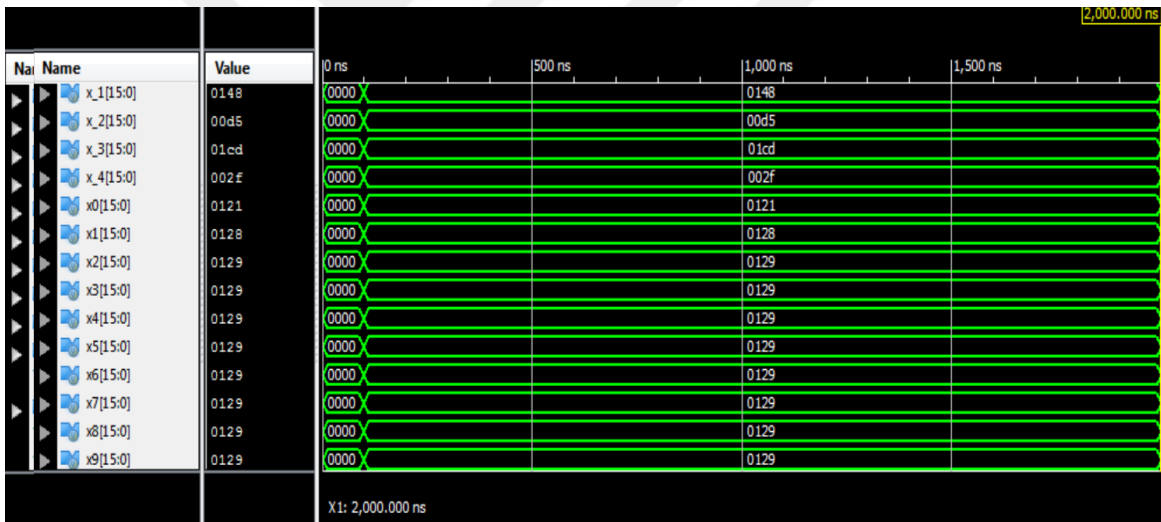
Şekil 4.13’de ise FIR Filtre işlemcisi tarafında filtrelenmiş EEG sinyalinin FPGA’daki ilk bölümünü, Şekil 4.14’de ise FPGA’da bu verinin heksadesimal sayı formatında gösterimi verilmiştir. Şekil 4.15’de ise filtrelenmemiş EEG verisinin x[0] ile x[9] arasındaki giriş heksadesimal sayı formatında gösterilmiştir.



Şekil 4.13. FIR filtre İşlemcisi tarafından filtrelenmiş EEG sinyalinin birinci bölümü



Şekil 4.14. Filtrelenmiş EEG sinyalinin heksadesimal gösterimi

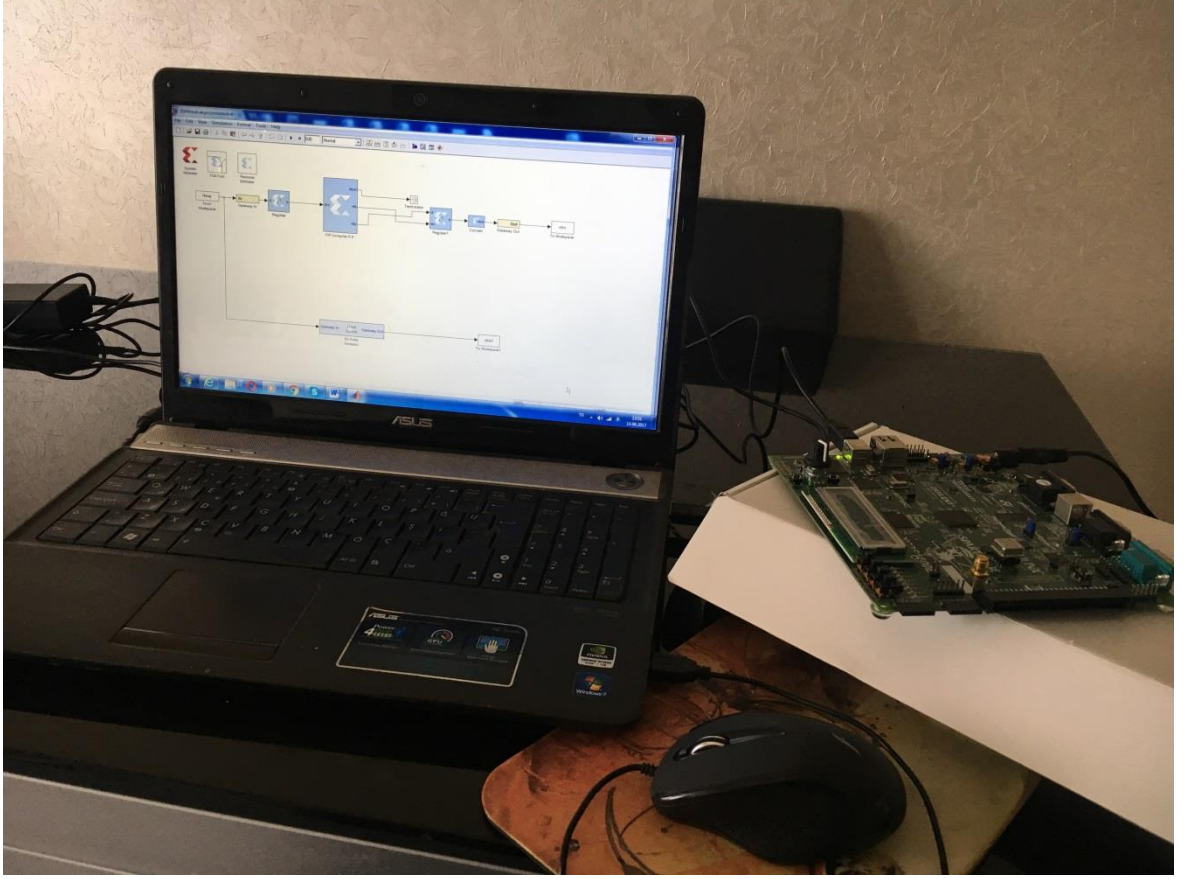


Şekil 4.15. Filtrelenmemiş EEG sinyalinin x[0]-x[9] girişlerinin heksadesimal gösterimi

4.2 Xilinx Sistem Generator Kullanılarak FIR Filtre Tasarımı Uygulaması

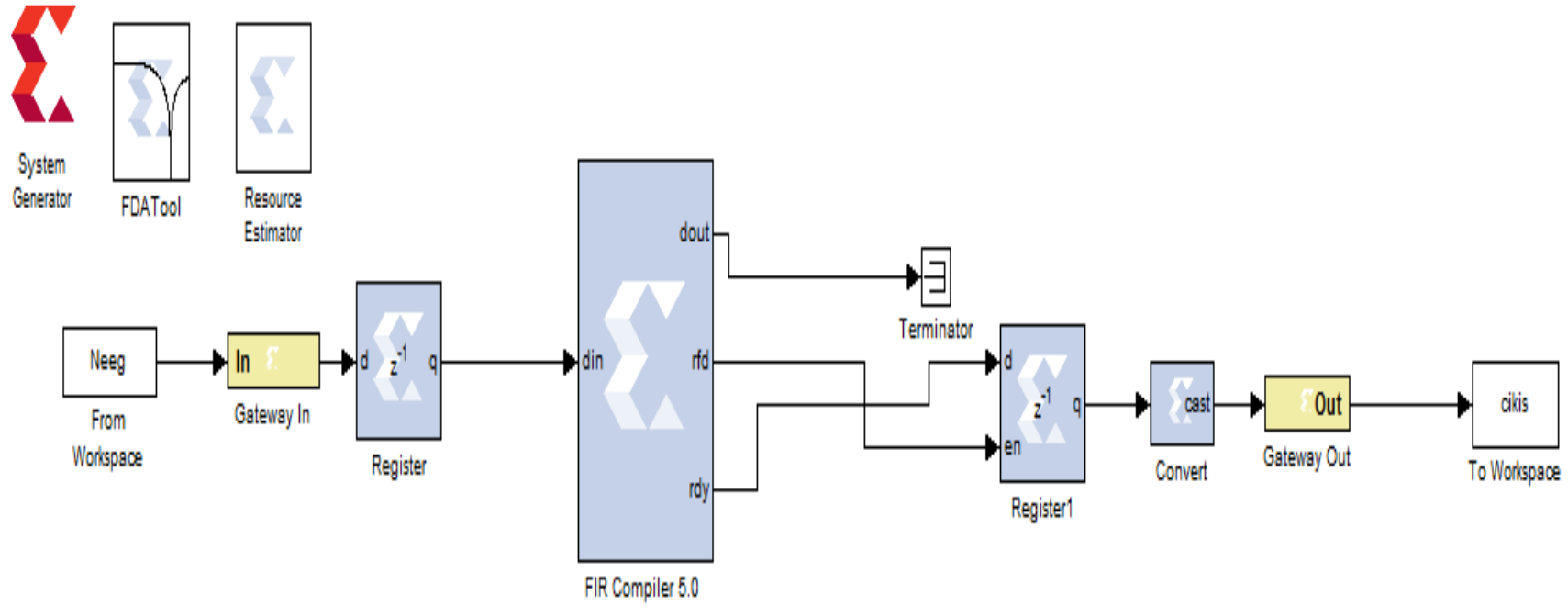
Bu uygulamada, bioelektriksel bir işaret olan EEG sinyali, XSG kullanılarak önce MATLAB Simulink ortamında tasarlanan FIR Filtre ile filtre edilmiş ve daha sonra FPGA üzerinde gerçekleştirilmesi incelenmiştir. Çalışma için tasarlanan sistemde kullanılan filtre, 4.1 uygulamasındaki filtre ile aynıdır. Filtreyi tasarlarken FDATool bloğu kullanılmıştır. Sisteme uygulanan EEG sinyali yine ilk uygulamada kullanılan EEG sinyalinin aynıdır. Aynı anda giriş sinyali olan EEG sinyali hem Simulink üzerinden hem de FPGA üzerinde

FIR filtreye uygulanarak filtre işlemi gerçekleştirilebilir. Filtrelenmiş sinyal hem FPGA üzerinden gerçekleştirilen filtre çıkışından hemde Simulink üzerinden simule edilen filtre çıkışında görülebilir. Filtre tasarımı için oluşturulan sistem Şekil 4.16’da görülmektedir.



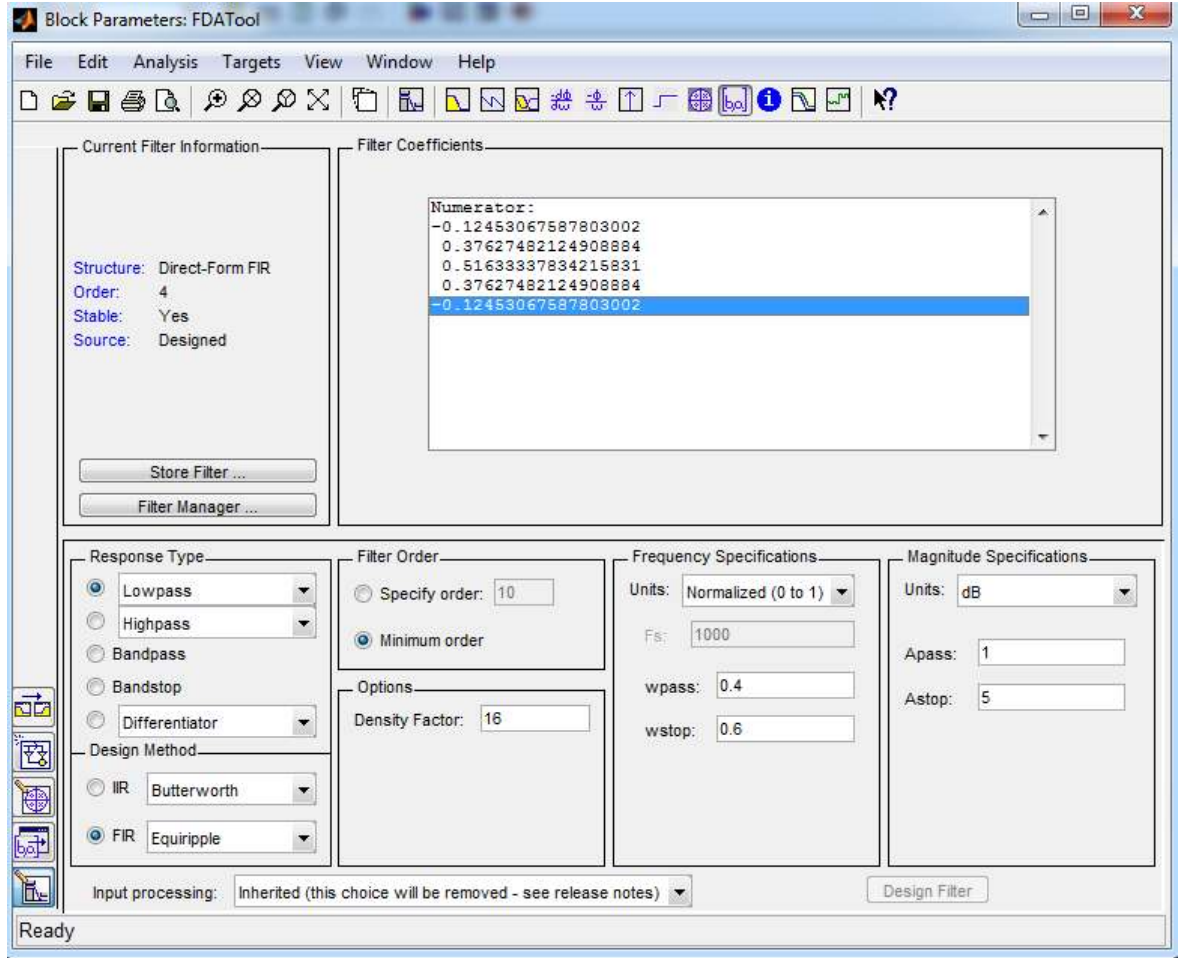
Şekil 4.16. Filtre tasarımı için oluşturulan sistem

Simulink ortamında gerçekleştirilen filtre tasarımı Şekil 4.17’de verilmiştir.



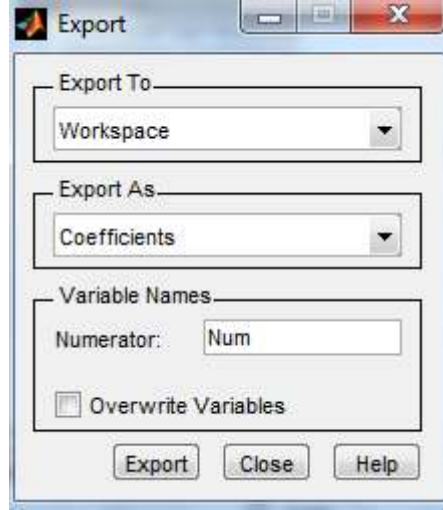
Şekil 4.17. XSG DSP blokları ile FIR filtre tasarımı

Sistemde kullanılacak EEG sinyali Matlab çalışma ortamından alınarak Xilinx Geçit Giriş bloğuna uygulanmıştır. Daha sonra Xilinx Geçit Giriş bloğundan alınan sinyal önce kayıt bloğuna daha sonra ise FIR Derleyici bloğuna uygulanmıştır. FIR Derleyici bloğunda kullanılan filtre FDATool bloğu tarafından tasarlanmıştır. Tasarlanan filtre Şekil 4.18’de görülmektedir.



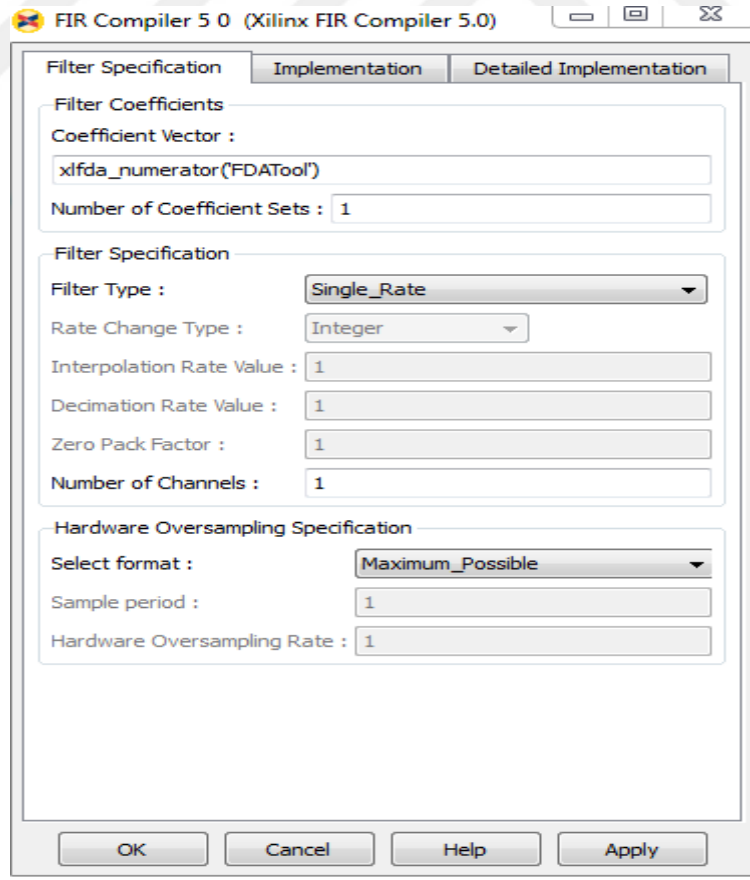
Şekil 4.18. Kullanılan 4. Dereceden alçak geçiren filtre

Filtre tasarlandıktan sonra filtre katsayıları FIR Derleyici bloğuna tanımlanmak için Matlab çalışma ortamına aktarılmıştır. Bu aktarım Şekil 4.19’da görülmektedir.



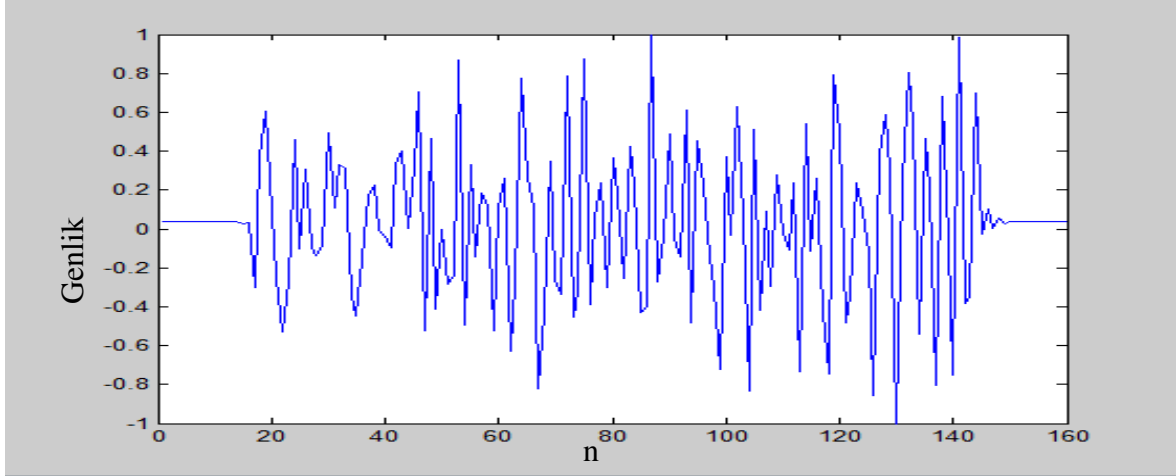
Şekil 4.19. Katsayı aktarımı

Filtre katsayılarının aktarımından sonra FIR Derleyici 5.0 bloğuna girilerek kullanacağımız filtrenin katsayıları FIR Derleyici bloğuna tanıtılmıştır. Şekil 4.20’de tanıtım gösterilmektedir.

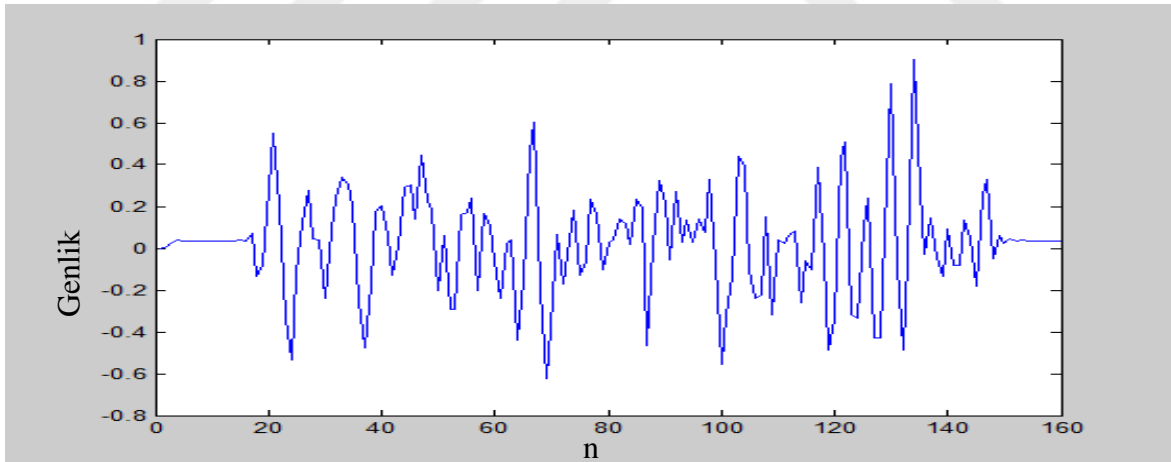


Şekil 4.20. Filtre tanıtımı

FIR Derleyici 5.0 bloğun'dan alınan sinyal tekrar kayıt bloğuna uygulanıp daha sonra Xilinx Geçit Çıkış bloğuna uygulanmıştır. Xilinx Geçit Çıkış bloğundan alınan sinyal Matlab çalışma ortamına atılıp tekrardan çizdirilmiştir. Filtre girişine uygulanan sinyal Şekil 4.21'de ve filtre çıkışında elde edilen sinyal Şekil 4.22'de görülmektedir.

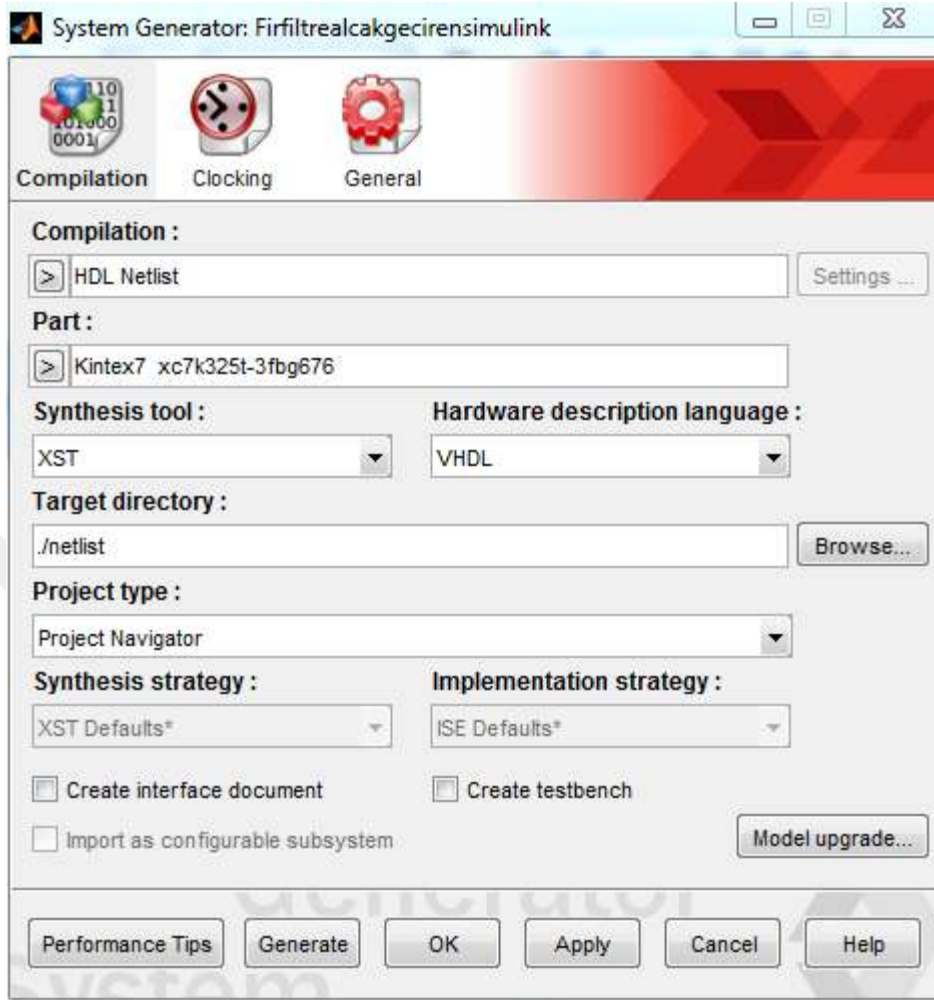


Şekil 4.21. Filtrenmemiş normalize edilmiş sinyal



Şekil 4.22. Filtre edilmiş sinyal

Sinyallerden görüldüğü üzere tasarlanan filtre simülasyonu çalışmaktadır. Daha sonra Matlab Simulink ortamında tasarlanan bu sistemin FPGA'da gerçekleştirmesini yapabilmek için FPGA kartını temsil eden "JTAG Co-sim" bloğu System Generator tarafından oluşturulur. Bu bloğu oluşturmak için önce System Generator bloğuna girilir. System Generator bloğunun bilgi ekranı Şekil 4.23'de verilmiştir.



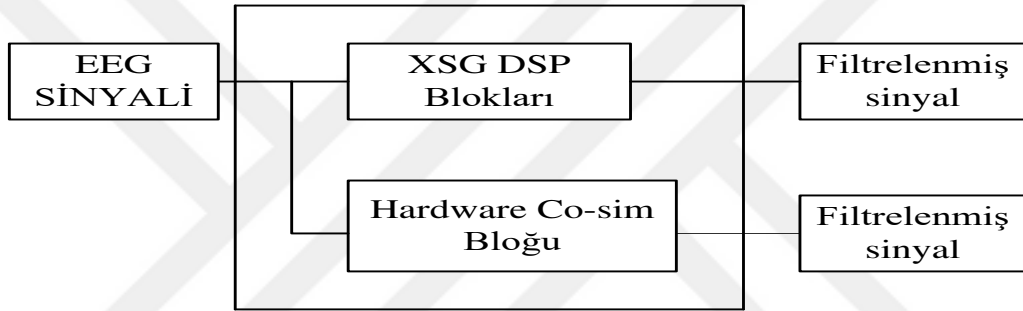
Şekil 4.23. Sistem generator blok penceresi

Sistem Generator 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.) Yazılım dili olarak Verilog seçilmiştir. Daha sonra programlamada gerekli olan .bit uzantılı dosya ve verilog yazılımı oluşturulur. Bu dosya oluşturduktan sonra FPGA kartını temsil eden “JTAG Co-sim” bloğu oluşturulur. Bu bloğu oluşturmak için “Compilation” bölümünden donanım eş zamanlı simülasyon seçeneği içerisinde “New Compilation Target” seçilir. Daha Sonra kullanacak FPGA sisteme tanıtıldıktan sonra “JTAG Co-sim” bloğu tasarlanır. Tasarlanan blok Şekil 4.24’de verilmiştir.



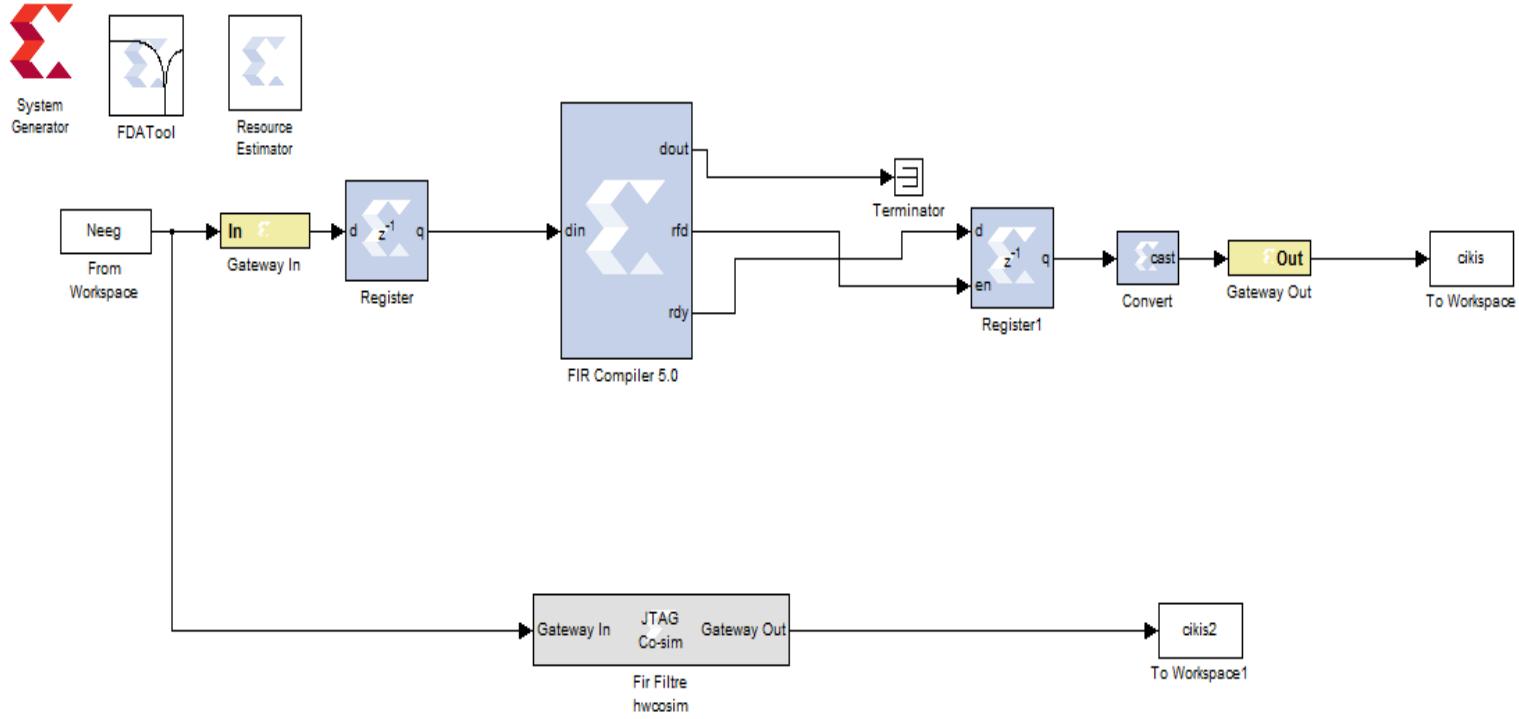
Şekil 4.24. Tasarlanan hwcossim bloğu

Tasarlanan JTAG Co-sim bloğu Şekil 4.17’deki Xilinx Geçit Giriş bloğu ve Xilinx Geçit Çıkış blokları dahil arasındaki blokların gerçekleştirdiği işlemleri FPGA kartında yapan tek bir bloktur. Yapılan model Şekil 4.25’de gösterilmiştir.



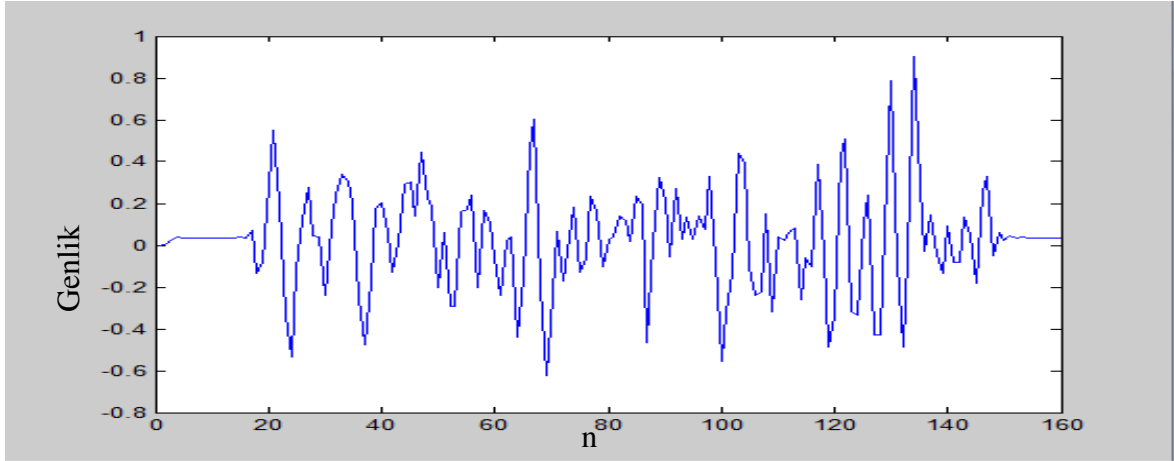
Şekil 4.25. Simulink model gösterimi

Donanım eş zamanlı simülasyon bloğu kullanılarak Simulink tabanlı 4.derecen FIR filtre modeli Şekil 4.26’da gösterilmektedir.



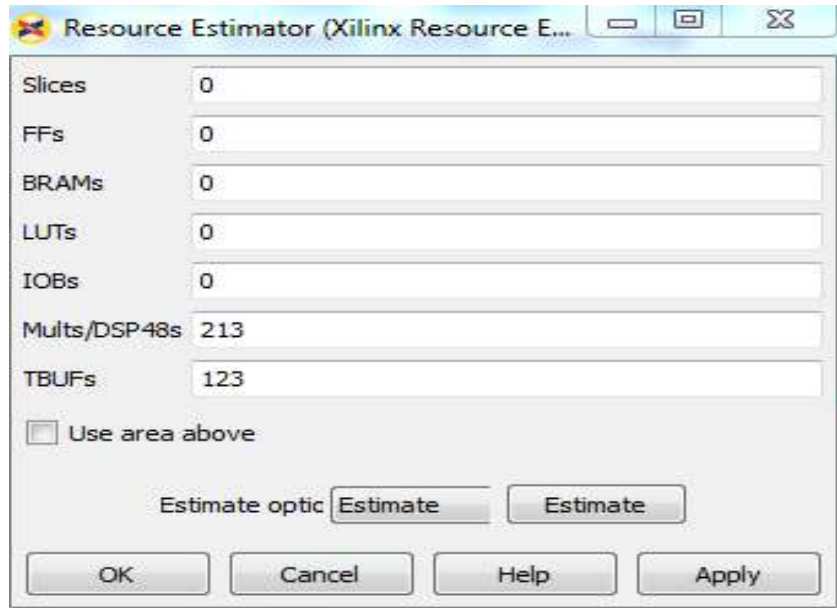
Şekil 4.26. Donanım eşzamanlı simülasyon bloğu kullanılarak oluşturulan Filtre modeli

Şekil 4.26’da gösterilen filtre modeli çalıştırıldığında .bit uzantılı dosya USB portu üzerinden FPGA kartına gönderilir. FPGA’nın programı çalıştırmasından sonra giriş sinyelimiz olan EEG sinyali XSG DSP bloklarından ve aynı anda donanım eş zamanlı simülasyon bloğundan geçerek FPGA kartı üzerinde filtrelenen sinyal filtrelenmiş şekilde çizdirilir. Bu işlem seri ve sayısal olarak gerçekleştirilir. Filtrelenmiş sinyal Şekil 4.27’de verilmiştir.



Şekil 4.27. Filtre işlemi sonrası çıkış sinyali

Kaynak kestirimi bloğu bilgi ekranı Şekil 4.28’de görülmektedir.



Şekil 4.28. Kullanılan kaynağı hesaplama aracı

Bu bilgi ekranından estimate butonu kullanılarak tasarlanan sistemin FPGA kaynak kullanımı bulunabilir. XSG kullanılarak tasarlanan FIR filtrenin FPGA kaynak kullanımı Tablo 4.3 de verilmiştir.

Tablo 4.3. Tasarlanan FIR filtrenin kaynak kullanımı

FIR Filtre	Kullanılan Dilimler	Kullanılan FFs	Kullanılan LUTs
Kaynak Kullanımı	2231	2612	3364

Verilog dili kullanılarak ve XSG kullanılarak tasarlanan FIR Filtrenin kaynak kullanımları Tablo 4.4 de verilmiştir.

Tablo 4.4. Kaynak Kullanımı Karşılaştırması

Kaynak Kullanımı			
	XSG FIR Filtre	Verilog FIR Filtre	Kullanılabilir
Kullanılan Dilimler	2231	12145	14,752
Kullanılan FFs	2612	14523	29,504
Kullanılan LUTs	3364	21785	29,504

Yukarıda ki tablodaki veriler incelendiğinde XSG kullanılarak tasarlanan 4. Dereceden alçak geçiren FIR Filtrenin kaynak kullanımlarının daha az olduğu görülmüştür. Bu nedenle 4. Dereceden alçak geçiren FIR filtrenin XSG kullanılarak gerçekleştirilmesi Verilog diliyle tasarımına göre kaynak kullanımı açısından karşılaştırıldığında daha iyi sonuç verdiği söylenebilir.

5.SONUÇLAR

Bu tez çalışmasında Matlab Simulink ile uyumlu çalışan XSG ve FPGA yazılım dili olan Verilog kullanılarak 4.dereceden alçak geçiren FIR Filtre tasarlanmıştır. Bu çalışmanın sonucunda FPGA'nın hızından ve paralel işlem yeteneğinden dolayı tasarlanan sayısal filtrelerin filtreme işlemini gerçekleştirdiği gözlemlenmiştir. Giriş sinyali olarak kullanılan EEG sinyali tasarlanan sistemlerde başarılı bir şekilde filtrelenmiştir. Bu tez çalışmasında yapılan filtre tasarımları görüntü işleme, sinyal işleme, haberleşme sistemleri gibi vb. 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. Hızlı tasarımından ve az kaynak kullanımından dolayı DSP uygulamaları için XSG tercih edilebilir.

İleriki çalışmalarda tasarlanan filtrelerin dereceleri değiştirilip FPGA kaynak kullanımları bakımından karşılaştırılabilirler. Aynı anda birden fazla başka derecelere sahip FIR filtreler XSG kullanılarak tasarlanıp girişlerine farklı sinyaller uygulanıp sonuçları incelenebilirler. Ayrıca tezde tasarlanan filtrelerinin girişine uygulanan EEG sinyalinden başka bir biyoelektriksel sinyal uygulanıp sonuçlarına bakılabilir.

KAYNAKLAR

- [1] **Dehon, A.**, 2000. The Density Advantage of Reconfigurable Computing, IEEE Computer, 33, 41-49.
- [2] **Qasim, S.M., Abbasi S.A., and Almashary, B.**, 2009. An Overview of Advanced FPGA Architectures for Optimized Hardware Realization of Computation Intensive Algorithms, IMPACT'09, 300-303
- [3] **Bıçakcı S., Çetinkaya M.B., Karaboğa N.**, "Sayısal FIR Süzgeç İçin Vhdl Kod Geliştirilmesi", Elektrik-Elektronik Bilgisayar Mühendisliği 11. Ulusal Kongresi ve Fuarı, İSTANBUL, TÜRKİYE, 22-25 Eylül 2005, ss.332-334
- [4] **Louzao J, Paz S, Tejera D, Bellora G, Langwagen G**; Architectural design of a programmable cell for the implementation of a filter bank on FPGA. Microelectronics Reliability; September 2003
- [5] **R.e M, Cardarilli G. C, Re A. D, Lojacona R**; FPGA Implementation of a Demux Based on a Multirate Filter Bank. ISCAS 2000 – IEEE International Symposium on Circuits and Systems; May 28-31, 2000, Geneva, Switzerland.
- [6] **Lee H, Sobelman G. E**; Performance evaluation and optimal design for FPGA- based digit-serial DSP functions. Computers and Electrical Engineering; January 1999.
- [7] **Anurag A, Astha S, Tushar N**, FIR Filter Designing using Xilinx System Generator International Journal of Computer Applications (0975 – 8887) Volume 68–No.11, April 2013
- [8] **Emmanuel S. Kolawole, Warsame H. Ali, Penrose Cofie, John Fuller, C. Tolliver, Pamela O**, Design and Implementation of Low-Pass, High-Pass and Band-Pass Finite Impulse Response (FIR) Filters Using FPGA Circuits and Systems, 2015, 6, 30-48 Published Online February 2015 in SciRes.
- [9] **D. Murat.**, Video frekans uygulamaları için a-sınıfı logaritmik ortam epileptik filtre tasarımı (Danışman: Doç.Dr. Erhan Yüce) Pamukkale Üniversitesi Fen bilimleri Enstitüsü yüksek lisans tezi,2013
- [10] **Fox R.M., Hyung – Jong Ko, Eisenstadt W.R.**, 2004: Differential log-domain filters with high-gain common-mode feedback. IEEE Transactions On Circuits And Systems—I: Regular Papers, vol. 51, no.2,254-263.
- [11] **Deliyannis, T., Sun, Y. and Fidler, J. K.**, 1999: Continuous-Time Active Filter Design, (1nd Ed.) CRC Press LLC, ISBN- 0-8493-2573-0.

- [12] **Sedra, A. S., Smith, K. C.**, 1998: Microelectronics Circuits, (4nd Ed.) Oxford University Press, ISBN 0-19-511663-1, 1237 p.
- [13] **Batık Z.**, Sayısal Filtre Tasarım Yöntemleri Ve Performans Analizleri, Yüksek Lisans Tezi, Sakarya Üniversitesi Fen Bilimleri Enstitüsü, (2011).
- [14] **Ertürk S.**, Sayısal işaret isleme, Birsan Yayınevi, İstanbul, 293s.
- [15] **Haykin, S.**, Adaptive Filter Theory, Prentice Hall, USA, 2002.
- [16] **NG, S.C.**, et al., The Genetic Search Approach :A New Learning Algorithm for Adaptive IIR Filtering, IEEE Signal Processing Magazine, 13 (6), 38-46, 1996.
- [17] **FPGA**, <http://coep.vlab.co.in/?sub=29&brch=88&sim=228&cnt=1>
- [18] **FPGA**, <https://www.fpgarelated.com/showarticle/17.php>
- [19] **FPGA**, <http://www.ni.com/white-paper/6984/en/>
- [20] **Areibi S.**, 2016: Tutorial - Using Xilinx System Generator 14.6 for Co-Simulation on Digilent NEXYS3 (Spartan-6) Board
- [21] **Xilinx**, https://www.xilinx.com/support/documentation/sw_manuals/xilinx11/sysgen_gs.pdf
- [22] **J Clin N.**, American Clinical Neurophysiology Society. Guideline 8: guidelines for recording clinical EEG on digital media. 23(2):122–4; 2006.
- [23] **Karakaya, B., Yeniceri, R., Yalcın M.E.** 2015. Wave computer core using fixed-point arithmetic. 2015 IEEE International Symposium on Circuits and Systems (ISCAS). pp. 1514-1517.
- [24] **Xilinx**, https://www.xilinx.com/support/documentation/sw_manuals/xilinx2015_3/ug897-vivado-sysgen-user.pdf

ÖZGEÇMİŞ

Erman ÖZPOLAT, 1989 yılında Elazığ'da doğdu. Lise öğrenimini Elazığ Özel Harput Lisesinde (Doğa Koleji) tamamladı. 2007 yılında Fırat Üniversitesi Elektrik-Elektronik Mühendisliği'ni kazandı ve 2011 yılında mezun oldu. Mezun olduktan sonra 1 yıl yurtdışı tecrübesi yaşadıktan sonra Türkiye'ye dönüp Elazığ'da özel bir firmada Elektrik Elektronik Mühendisi olarak çalıştıktan sonra Muş Alparslan Üniversitesi Elektrik Elektronik Mühendisliği Anabilim Dalında araştırma görevlisi olarak göreve başladı. Daha sonra Fırat Üniversitesi Fen Bilimleri Enstitüsü Elektrik-Elektronik Mühendisliği Anabilim Dalı'nda yüksek lisans öğrenimine başladı. Halen, Fırat Üniversitesi Mühendislik Fakültesi Elektrik-Elektronik Mühendisliği Bölümü'nde araştırma görevlisi olarak görev yapmaktadır.