T.C.
YILDIZ TEKNİK ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

MEMRİSTİF SİNAPTİK DEVRELER İÇİN UYARLANABİLİR PARÇALI LİNEER YAKLAŞIM

Barış ŞAHİNTEKİN

YÜKSEK LİSANS TEZİ
Elektronik ve Haberleşme Mühendisliği Anabilim Dalı
Elektronik Programı

Danışman
Dr. Öğr. Üyesi Zehra Gülru ÇAM TAŞKIRAN

Ağustos, 2024

T.C.
YILDIZ TEKNİK ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

MEMRİSTİF SİNAPTİK DEVRELER İÇİN UYARLANABİLİR PARÇALI LİNEER YAKLAŞIM

Barış ŞAHİNTEKİN tarafından hazırlanan tez çalışması 01.08.2024 tarihinde aşağıdaki jüri tarafından Yıldız Teknik Üniversitesi Fen Bilimleri Enstitüsü Elektronik ve Haberleşme Mühendisliği Anabilim Dalı Elektronik Programı **YÜKSEK LİSANS TEZİ** olarak kabul edilmiştir.

Dr. Öğr. Üyesi Zehra Gülru ÇAM TAŞKIRAN
Yıldız Teknik Üniversitesi
Danışman

Jüri Üyeleri

Dr. Öğr. Üyesi Zehra Gülru ÇAM TAŞKIRAN, Danışman
Yıldız Teknik Üniversitesi _____

Doç. Dr. Umut Engin AYTEN, Üye
Yıldız Teknik Üniversitesi _____

Dr. Öğr. Üyesi Surhan BOZKURT, Üye
Doğuş Üniversitesi _____

Danışmanım Dr. Öğr. Üyesi Zehra Gülru ÇAM TAŞKIRAN sorumluluğunda tarafımca hazırlanan Memristif Sinaptik Devreler İçin Uyarlanabilir Parçalı Lineer Yaklaşım başlıklı çalışmada veri toplama ve veri kullanımında gerekli yasal izinleri aldığımı, diğer kaynaklardan aldığım bilgileri ana metin ve referanslarda eksiksiz gösterdiğimi, araştırma verilerine ve sonuçlarına ilişkin çarpıtma ve/veya sahtecilik yapmadığımı, çalışmam süresince bilimsel araştırma ve etik ilkelerine uygun davrandığımı beyan ederim. Beyanımın aksinin ispatı halinde her türlü yasal sonucu kabul ederim.

Barış ŞAHİNTEKİN

İmza

Annem'e

TEŞEKKÜR

Çalışmamda bana yol gösteren, yönlendiren, destek ve emeklerini esirgemeyen, öğrencisi olduğum için kendimi çok şanslı hissettiğim ve her zaman şanslı hissedeceğim tez danışmanım Dr. Öğr. Üyesi Zehra Gülru ÇAM TAŞKIRAN'a,

Akademik hayatım boyunca her konuda bana destek olan, karşılaştığım zorlukları çözebilmek için desteğini hiç esirgemeyen, her koşulda yanımda olan Nagihan AYDEMİR'e

Eğitim hayatımın başından itibaren her zaman yanımda olup bir an bile olsun desteklerini esirgemeyen kıymetli aileme,

Tez sürecim boyunca sabırla, hoşgörüyle ve iyi niyetle her zaman yanımda olan, her koşulda desteğini hissettiren değerli arkadaşlarıma

En içten dileklerimle teşekkür ederim.

Barış ŞAHİNTEKİN

İÇİNDEKİLER

SİMGE LİSTESİ vii

KISALTMA LİSTESİ viii

ŞEKİL LİSTESİ ix

TABLO LİSTESİ xi

ÖZET xii

ABSTRACT xiv

1 GİRİŞ 1

1.1 Literatür Özeti 1

1.2 Tezin Amacı 3

1.3 Hipotez 3

2 YÖNTEM ve ARAÇLAR 5

2.1 Memristör 6

2.2 Parçalı Lineer Karakteristik (PLK) 9

2.3 R_{ON} - R_{OFF} Osilasyonu (OSC) 9

2.4 Eğim Değişimi Hızlandırıcı (EDH) 10

2.5 FPGA 11

2.5.1 FPGA Mimarisi 12

2.5.2 ZYNQ Mimarisi 13

2.5.3 FPGA Tasarımı 13

3 SİMÜLASYON SONUÇLARI ve DENEYSEL SONUÇLAR 19

3.1 Sinüzoidal Sinyal Üretimi 20

3.2 Darbe Sinyali Üretimi 25

3.3 Analog Sinyal Okuma 26

4 KARŞILAŞTIRMA VE TARTIŞMA 30

4.1 Değer Güncelleme Yöntemleri Karşılaştırması 30

4.2 Büyük Ölçekli Prototip Tasarım Önerisi 35

5 SONUÇ **41**

KAYNAKÇA **43**

TEZDEN ÜRETİLMİŞ YAYINLAR **45**

SİMGE LİSTESİ

i	Akım
t	Darbe Süresi
R	Direnç
m	Eğim
q	Elektriksel Yük
f	Frekans
v	Gerilim
G	İletkenlik
R_{OFF}	Maksimum Memristans
ψ	Manyetik Akı
M	Memristans
M_E	Memristör Emülatörü
R_{ON}	Minimum Memristans

KISALTMA LİSTESİ

ADC	Analog to Digital Converter
AXI	Advanced eXtensible Interface
BRAM	Block Random Access Memory
DAC	Digital to Analog Converter
DC	Direct Current
DEMUX	Demultiplexer
FF	Flip-Flop
FPGA	Field Programmable Gate Array
GPIO	General Purpose Input/Output
ILA	Integrated Logic Analyzer
IP	Intellectual Property
LUT	Lookup Table
MSPS	Mega Sample per Second
MUX	Multiplexer
SPI	Serial Peripheral Interface
SS	Slave Select

ŞEKİL LİSTESİ

Şekil 2.1 Önerilen yöntemde kullanılan algoritmaya ait psödo kod 5
Şekil 2.2 Farklı memristör modellerine ait memristans-zaman grafikleri . . 7
Şekil 2.3 Farklı memristör modellerine ait akım-gerilim histerezis eğrileri . 8
Şekil 2.4 Parçalı lineer karakteristik algoritması ile memristans güncellemesi 9
Şekil 2.5 R_{ON} - R_{OFF} osilasyon problemi 10
Şekil 2.6 Temel FPGA mimarisi 12
Şekil 2.7 Zybo Z-7010 FPGA kartı ZYNQ mimarisi 14
Şekil 2.8 FPGA tasarımı blok şeması 15
Şekil 2.9 Kullanılan 14-DIP kılıfına sahip MCP4922 dijitalden analoga dönüştürücü entegresi 16
Şekil 2.10 Zybo Z-7010 geliştirme kartının üstten görüntüsü 17
Şekil 2.11 Tasarlanan sistem ait blok diyagramı 18
Şekil 3.1 Deneysel ölçümler için kurulmuş devre şeması 20
Şekil 3.2 MCP4922 DAC entegresi için yazma komutu zaman diyagramı . . 21
Şekil 3.3 Sinüzoidal sinyal üretimi için oluşturulmuş blok tasarımı 22
Şekil 3.4 ILA kullanılarak oluşturulmuş SPI simülasyonu 23
Şekil 3.5 MOSI pininin osiloskop ekranında görülen çıktısı 23
Şekil 3.6 SS pininin osiloskop ekranında görülen çıktısı 24
Şekil 3.7 DAC entegresinin çıkışlarının osiloskop ekranında görülen çıktısı 24
Şekil 3.8 Darbe sinyali için oluşturulmuş blok diyagram 26
Şekil 3.9 Darbe sinyali üretimi için ILA üzerindeki simülasyon sonuçları . 26
Şekil 3.10 Üretilen darbe sinyalinin osiloskop ekranındaki görüntüsü 27
Şekil 3.11 Analog gerilim okuma için oluşturulmuş blok diyagram 28
Şekil 3.12 Seri monitör ekranından görülen gerilim ve empedans verileri . . 28
Şekil 3.13 FPGA güç tüketimi tablosu 29
Şekil 3.14 FPGA kaynak tüketimi tablosu 29
Şekil 4.1 Sürekli ölçüm yöntemi için oluşturulmuş blok diyagram 33
Şekil 4.2 Sürekli ölçüm yöntemi için kaynak tüketim tablosu 34
Şekil 4.3 Sürekli ölçüm yöntemi için güç tüketim tablosu 35
Şekil 4.4 Altı adet memristif sinaptik devre hücresine sahip sinaptik devre için değer güncelleme şeması 36

Şekil 4.5 Altı adet memristif sinaptik devre hücresine sahip sinaptik devre için değer güncelleme algoritması 38
Şekil 4.6 FPGA geliştirme kartında 6 sinaptik hücreli prototip için oluşturulmuş blok tasarımı 39
Şekil 4.7 FPGA geliştirme kartında 6 sinaptik hücreli tasarımın güç tüketimi tablosu 39
Şekil 4.8 FPGA geliştirme kartında 6 sinaptik hücreli tasarımın kaynak tüketimi tablosu 40

TABLO LİSTESİ

Tablo 3.1 Maksimum %5 hata ile 100 farklı değer güncelleme sürecindeki toplam okuma-yazma döngüsü sayıları 20
Tablo 4.1 Literatürde yer alan farklı memristans değer güncelleme yöntemlerinin kıyaslanması 32

ÖZET

MEMRİSTİF SİNAPTİK DEVRELER İÇİN UYARLANABİLİR PARÇALI LİNEER YAKLAŞIM

Barış ŞAHİNTEKİN

Elektronik ve Haberleşme Mühendisliği Anabilim Dalı
Yüksek Lisans Tezi

Danışman: Dr. Öğr. Üyesi Zehra Gülru ÇAM TAŞKIRAN

Yapay zeka tabanlı sistemler ve sinaptik devreler son yıllarda bir çok araştırmacı için ilgi çekici bir konu haline gelmiştir. Yüksek işlem gücü gerektiren bu sistemlerin gelişmesiyle beraber yazılımsal olarak gerçekleştirilmesi hem zaman hem de işlem maliyeti açısından oldukça zorlayıcı olmaya başlamıştır. Bu zorlukların etkisiye bu sistemlerin donanımsal olarak gerçeklenmesi ihtiyacı ortaya çıkmıştır. Sinaptik devrelerin donanımsal olarak gerçeklenmesiyle beraber güç tüketimi ve işlem maliyetinde düşüş, işlem kapasitesinde ise bir artış beklenmektedir. Sinaptik devrelerde duyulan hafızalı eleman ihtiyacı için literatürde farklı çözümler mevcuttur. Memristör elemanı bu ihtiyaç için sıklıkla kullanılmakta olan bir pasif devre elemanıdır. Memristör elemanı ile gerçekleştirilen sinaptik devreler memristif sinaptik devreler olarak isimlendirilmektedir.

Bu çalışmada memristif sinaptik devreler için donanımsal bir yöntem önerilmiştir. Önerilen yöntemde yapay sinir ağının ağırlık parametrelerine denk gelen memristör elemanının memristans değeri kontrol edilerek ağ yapısının yeniden konfigüre edilebilmesi amaçlanmıştır. Önerilen yöntemde memristörün değerini kontrol etmek amacıyla donanımsal bir tasarım gerçekleştirilmiş ve bu tasarım üzerinden gerekli işlemleri yapacak algoritma önerilmiştir. Önerilen yöntemde memristör üzerine sinüzoidal sinyal uygulayarak memristans değerini ölçmek ve darbe sinyali uygulayarak memristansın değerini değiştirmek gibi alt bloklar mevcuttur. Önerilen yöntem memristör emülatörü devresi üzerinde test edilmiş ve elde edilen sonuçlar

paylaşılmıştır. Aynı zamanda algoritmik olarak farklı memristör modelleri ile sistem test edilmiş ve sonuçları paylaşılmıştır.

Anahtar Kelimeler: Sinaptik devreler, memristör programlama, parçalı lineer yaklaşım, FPGA uygulaması.

ABSTRACT

ADAPTIVE PIECEWISE LINEAR CHARACTERISTIC APPROACH FOR MEMRISTIVE SYNAPTIC CIRCUITS

Barış ŞAHİNTEKİN

Department of Electronics and Communication Engineering
Master of Science Thesis

Supervisor: Asst. Prof. Zehra Gülru ÇAM TAŞKIRAN

Artificial intelligence-based systems and synaptic circuits have become an interesting topic for many researchers in recent years. With the development of these systems, which require high processing power, software implementation has become quite challenging in terms of both time and computational cost. Due to these difficulties, the need for hardware implementation of these systems has emerged. With the hardware implementation of synaptic circuits, a decrease in power consumption and computational cost and an increase in computational capacity are expected. There are different solutions in the literature for the memory element requirement in synaptic circuits. Memristor element is a passive circuit element that is frequently used for this need. Synaptic circuits realized with the memristor element are called memristive synaptic circuits.

In this study, a hardware method is proposed for memristive synaptic circuits. In the proposed method, it is aimed to reconfigure the network structure by controlling the memristance value of the memristor element corresponding to the weight parameters of the artificial neural network. In the proposed method, a hardware design is realized to control the value of the memristor and an algorithm is proposed to perform the necessary operations through this design. In the proposed method, there are sub-blocks such as measuring the memristance value by applying a sinusoidal signal on the memristor and changing the memristance value by applying a pulse signal. The proposed method is tested on a memristor emulator circuit and the results are shared. At the same time, the system is algorithmically tested with

different memristor models and the results are shared.

Keywords: Synaptic circuits, memristor programming, piece-wise linear approach, FPGA implementation.

1
GİRİŞ

1.1 Literatür Özeti

Direnç, kapasitör ve indüktör gibi pasif devre elemanları elektronik devrelerde sıklıkla kullanılmaktadır. Bu elemanlar gerilim, akım, elektriksel yük ve manyetik akı gibi temel elektriksel büyüklükler arasındaki ilişki kurulmasını sağlamaktadır. Direnç elemanı ile gerilim ve akım arasındaki ilişki ifade edilmektedir. Kapasitör elemanı gerilim ve elektriksel yük arasında bir ilişki kurmaktadır. İndüktör elemanı da akım ve manyetik akı büyüklükleri arasında bir ilişki kurulmasını sağlamaktadır. Chua tarafından 1971 yılında tanımlanan memristör elemanı 4 temel büyüklükten ilişki kurulmamış tek ikili olan manyetik akı ve elektriksel yük arasındaki ilişkiyi açıklamaktadır [1]. Tanımlandığı 1971 yılında itibaren bu alandaki çalışmacıların odaklandığı memristör elemanı, 2008 yılında HP firmasının memristörün fiziksel üretimini gerçekleştirdiğini duyurmasının ardından araştırmacılar için daha ilgi çekici bir konu haline gelmiştir [2]. Elektriksel yük ve manyetik akı arasındaki ilişki grafiği doğrusal olmasına rağmen memristör elemanının akım-gerilim grafiği doğrusal değildir. Memristör ismi hafıza direnci anlamına gelen memory resistor kelimelerinin birleşiminden ortaya çıkmıştır. Memristör direnci (memristans), memristör elemanı üzerine uygulanan manyetik akıya bağlı olarak değişmektedir. Ardından üzerine bir gerilim veya akım uygulanmadığında bile mevcut memristör direnci değerini korumaktadır. Bu özelliği memristör elemanını uçucu olmayan kalıcı bir hafızaya sahip olduğunu göstermektedir.

Memristör elemanının uçucu olmayan hafızaya sahip olması ve pasif bir eleman olması düşük güç tüketimi gerektiren işlemsel uygulamalarda kullanımı konusunda odaklanılan bir alan haline getirmiştir. Memristör elemanı sinaptik devrelerin donanımsal gerçeklemesi gibi düşük güç tüketimi gereksinimleri olan bazı alanlardaki hafıza elemanları ihtiyacını karşılayabilmek açısından önemli bir konumda bulunmaktadır. Yapay sinir ağlarının donanımsal gerçeklemesi ve sinaps gerçeklemelerinde kullanılan çapraz çubuk bağlantı yapılarında matris-vektör çarpımının gerçekleştirilmesinde sağladığı kolaylık nedeniyle memristör elemanı

sıklıkla kullanılmaktadır [3–6]. Ayrıca literatürde memristör elemanının değerini kontrol edebilme yeteneği sağlayan uyarlanabilir analog devrelere dair örnekler de mevcuttur [7, 8]. Bu analog devre örneklerinde en önemli kriterlerden biri hedeflenen memristans değerine yüksek doğrulukla ve hızlı bir şekilde ulaşılması konusundaki performanstır. Bununla beraber her memristörün akım-gerilim karakteristiğinin doğrusal olmama seviyesindeki farklılıklar hedeflenen memristans değerine ulaşma konusunda karşılaşılabilecek olası zorlukların başında gelmektedir. Bu doğrusal olmayan karakteristik memristif sinaptik devrelerde çevrimiçi eğitim için büyük bir zorluk olarak ortaya çıkmaktadır, bununla beraber çevrimdışı eğitim süreçlerinde de performans kaybına sebep olmaktadır [9]. Buna ek olarak her memristörün maksimum ve minimum memristans değerlerindeki farklılık bir başka zorluk olarak görülmektedir. Ayrıca memristör elemanının sahip olduğu ideal olmama durumu göz önüne alındığında matematiksel model geliştirilmesi de oldukça zordur. Programlama performansında yüksek doğruluk elde etmenin bir yolu olarak görülen belirli bir sabit değerin altındaki tüm memristans değerlerini hafızaya kaydetmek ve kontrol amaçlı sürekli olarak memristans ölçümü yapma yöntemi zaman ve güç tüketimi açısından oldukça maliyetlidir [5][10][11][12][13].

Bu nedenle memristör elemanlarının hedeflenen memristans değerlerine ulaştırılması için alternatif yöntemler geliştirilmiştir. Geliştirilen analog programlama devrelerinden bazıları memristörle beraber bir direnç elemanı kullanarak geleneksel bir gerilim bölücü devre yapısı kullanmaktadır ve bu yöntem memristans değerinin yüksek doğrulukla güncellenmesi konusunda kısıtlamalara neden olmaktadır [14]. Farklı bir çalışmada programlama için daha karmaşık ve daha büyük bir donanım yapısı gerektiren köprü tekniği kullanılmaktadır [14, 15]. Ayrıca programlama için köprü tekniğinin kullanıldığı çalışmalarda donanım yapısının 4 memristör gerektirmesi maliyet açısından bir dezavantaj yaratmaktadır. Geleneksel gerilim bölücü devre yapısının kullanıldığı çalışmalarda seçilen memristör elemanına bağlı olarak belirlenmesi gereken parametreler bulunmaktadır. Bu çalışmalar memristör elemanının karakteristiğinin farklı DC (Direct Current, Doğru Akım) gerilim seviyelerinde eğimlerindeki değişimleri temel almaktadır [16, 17]. Literatürde programlama için gerekli olan okuma-yazma döngü sayısını azaltmak amacıyla bir çoklu satır programlama şemasının önerildiği çalışma mevcuttur [18]. Bununla beraber önerilen bu yöntemde programlamaya ihtiyaç duymayan memristör elemanlarının da okuma ve yazma işlemine tabi tutulması sebebiyle enerji tüketiminde bir artış görülmektedir.

1.2 Tezin Amacı

Yapay zeka konusunda son yıllardaki gelişmeler ve bu alandaki büyüme göz önüne alındığında yapay zeka temelli sistemlerin donanımsal olarak gerçeklenmesi ihtiyacı oluşmuştur. Yapay sinir ağlarındaki parametrelerin bu gelişmeler ışığında artması bu sistemlerin performans gereksinimlerini giderek arttırmaktadır. Artan parametreler yüksek işlem gücü gerektirdiğinden bu sistemlerin donanımsal olarak gerçeklenmesi ihtiyacı doğmuştur. Bu tezin amacı bu yapay sinir ağlarının donanımsal gerçeklenmesi konusunda ağ parametrelerinin güncellenmesini sağlayan bir yöntem tasarlamaktır. Önerilen yöntemde memristör elemanının lineer olmama seviyesi, minimum ve maksimum memristans değerlerine ihtiyaç duymadan, herhangi bir memristans değerinden istenen memristans değerine en az sayıda okuma-yazma döngüsü yaparak yüksek hassiyetle ayarlanmasını sağlamak amaçlanmıştır. Bu doğrultuda yüksek doğruluk ve düşük güç tüketimi gereksinimlerini karşılayacak donanımsal bir yöntem geliştirilmiş ve geliştirme kartı üzerinde test edilmiştir. Aynı zamanda bu süreçlerin kontrolü için bir algoritma geliştirilmiş, sistemin düşük güç tüketimi gereksinimini sağlamak amacıyla yardımcı algoritmalar kullanılarak sistem optimize edilmiştir. Ayrıca önerilen yöntem farklı memristör modellerinin kullanılabilmesini sağlamak amacıyla uyarlanabilir şekilde tasarlanmıştır. Önerilen algoritma sinaptik devrelerin çevrimdışı veya döngüde donanım eğitimlerinde kullanımına, memristörlü uyarlanabilir kontrol sistemleri gerçeklemelerine uygundur.

1.3 Hipotez

Önerilen yöntemin donanımsal tasarımı FPGA (Field Programmable Gate Array, Alanda Programlanabilir Kapı Dizisi) geliştirme kartı üzerinde geliştirilmiştir. Ayrıca sistem yapısında gerekli olan okuma ve yazma fonksiyonları için dahili ve harici entegre devreler kullanılmıştır. Okuma işlemi sırasında dahili analogdan dijitale dönüştürücü entegresi kullanılmış, yüksek doğruluk ve yüksek hızda ölçümler gerçekleştirilmiştir. Yazma işlemi sırasında dahili ve harici bloklar kullanılmış, anahtarlama ve yüksek hız açısından tatmin edici performans gösterdiği görülmüştür. Bu özellikleri açısından çalışmada önerilen yöntemin düşük güç tüketimi sağlayacağı, yüksek hız ve doğrulukla çalışacağı düşünülmektedir. Ayrıca çalışmadaki yöntemin matematiksel modeller yerine donanımsal bir tasarıma sahip olması sistemin farklı memristör modellerine uyarlanabilirliğini sağlamaktadır. Prototip niteliğinde olan bu çalışmanın FPGA geliştirme kartı gibi paralel işlem yapabilme yeteneğine sahip bir platformda gerçekleştirilmesi, gerçek problemler için gerekli olan daha kompleks ve büyük memristif sinaptik devreler için ihtiyaç

duyulan paralel işlem kabiliyetinin kazandırılmasını sağlamıştır. FPGA geliştirme kartı üzerinde yapılan tasarım analiz edildiğinde geliştirme kartındaki kaynakların kullanımının düşük kaldığı görülmüştür.

2
YÖNTEM ve ARAÇLAR

Bu çalışmada memristif sinaptik devrelerde kullanılmak üzere uyarlanabilir parçalı lineer yaklaşım kullanan bir algoritma önerilmiştir. Ayrıca önerilen algoritmanın işlem süresini düşürmek ve dolayısıyla işlem maliyeti ve güç tüketimini minimize etmek adına yardımcı algoritmalar eklenmiştir. Önerilen algoritma FPGA üzerinde gerçeklenmiş ve memristör emülatörü kullanılarak önerilen yöntemin deneysel doğrulaması yapılmıştır.

Önerilen yöntemde memristans değer okuması için memristör emülatörü ve bir dirençten oluşan klasik bir gerilim bölücü devre şeması kullanılmıştır. Sistem R_{ON} and R_{OFF} değerleri ve memristans karakteristiği bilinmediğinin varsayıldığı bir memristör için hedef memristans değerinin girdi olarak alınması prensibiyle çalışır. İlk olarak klasik gerilim bölücü devre üzerinden memristans değerinin ölçümü gerçekleştirilir. Ardından memristör elemanının parçalı lineer karakteristiğe sahip olduğu varsayılarak kısa süreli darbe sinyali uygulanır. Bunun ardından ikinci defa memristans değeri ölçülür, ölçülen iki memristans değeri ve uygulanan darbe sinyalinin süresi kullanılarak memristör karakteristiğinin lineer varsayılan denklemi elde edilir. Şekil 2.1'de önerilen yöntemde kullanılan algoritmaya ait psödo kod verilmiştir.

Algoritma 1 Memristans Ayarlama Algoritması

procedure MEMRISTANS_AYARLAMA(M_{hedef},M_{hata})
  M_1 memristansını oku
  Δt süresi boyunca pulse sinyali uygula
  M_2 memristansını oku
  Eğimi hesapla: $m = \frac{M_2 - M_1}{\Delta t}$
  while $|M_{hedef} - M_2| > M_{hata}$ **do**
    Pulse süresini hesapla: $t_{pulse} = \frac{M_{hedef} - M_2}{m}$
    t_{pulse} süresi boyunca pulse sinyali uygula
    $M_1 \leftarrow M_2$
    M_{yeni} memristansını oku
    $M_2 \leftarrow M_{yeni}$
  end while
  return M_{yeni}
end procedure

Şekil 2.1 Önerilen yöntemde kullanılan algoritmaya ait psödo kod

2.1 Memristör

Memristör elemanı başlangıçta elektriksel yük $q = \int idt$ ve manyetik akı $\psi = \int vdt$ arasında özgün bir ilişki kuran tek kapılı bir devre elemanı olarak tanımlanmıştır [1]. Memristörün kurucu ilişkileri olarak isimlendirilen $\psi = \psi(q)$ ve $q = q(\psi)$ eşitlikleri memristör elemanının özel nitelikleridir. Memristör elemanı devrenin geri kalanına nasıl bağlandığından bağımsız olarak, anlık gerilim ve akım ilişkilerine bakılmaksızın tanımının geçerliliğini korur. Bu durum memristör elemanının yalnızca üzerinden geçen elektriksel yük q ve üzerindeki manyetik akı ψ değişkenlerine bağlı olan bir direnç elemanı gibi davranmasına sebep olmaktadır. Bu nedenle Ohm yasasının iki karşılıklı ikili formu memristör için geçerlidir. Bu bağıntıların tanımladığı memristans ve memdüktans, belirli bir zamandaki kurucu ilişkilerin hesaplanan eğimine denk gelmektedir [19]. Denklem 2.1'de memristör için geçerli Ohm yasasının ikili formu verilmiştir. Bu denklemlerde R_M memristör direncini ifade ederken G_M ise memristör iletkenliğini ifade etmektedir. Denklem 2.2'de ise bu eşitliklerdeki memristans ve memdüktans değerlerinin eşitlikleri verilmiştir.

$$v = R_M(q)i, i = G_M(\psi)v \tag{2.1}$$

$$R_M(q) = \frac{d\psi}{dq}, G_M(\psi) = \frac{dq}{d\psi} \tag{2.2}$$

Bu çalışmada önerilen yöntem, literatürde bulunan ve doğrulanmış bazı matematiksel memristör modelleri üzerinde test edilecektir [19]. Kullanılacak olan matematiksel memristör modelleri Denklem 2.3'te verilmiştir. Bu verilen memristör modellerinin memristans-zaman karakteristiğini elde etmek amacıyla akım değeri sabit olmamakla beraber 500 mV sabit gerilim uygulanmıştır. Bu işlem sonucu matematiksel memristör modellerinin memristans değerlerinin zamana bağlı değişimi Şekil 2.2'de verilmiştir. Elde edilen grafikler incelendiğinde eğrilerin lineer olmadığı görülmektedir. Bu durumda memristans değerinin güncellenmesi için lineer bir yaklaşım kullanılamayacağı görülmektedir. Aynı zamanda memristör modellerinin karakteristiklerine bakıldığında her bir matematiksel memristör modelinin lineer olmama seviyesinin farklı olduğu görülmüştür.

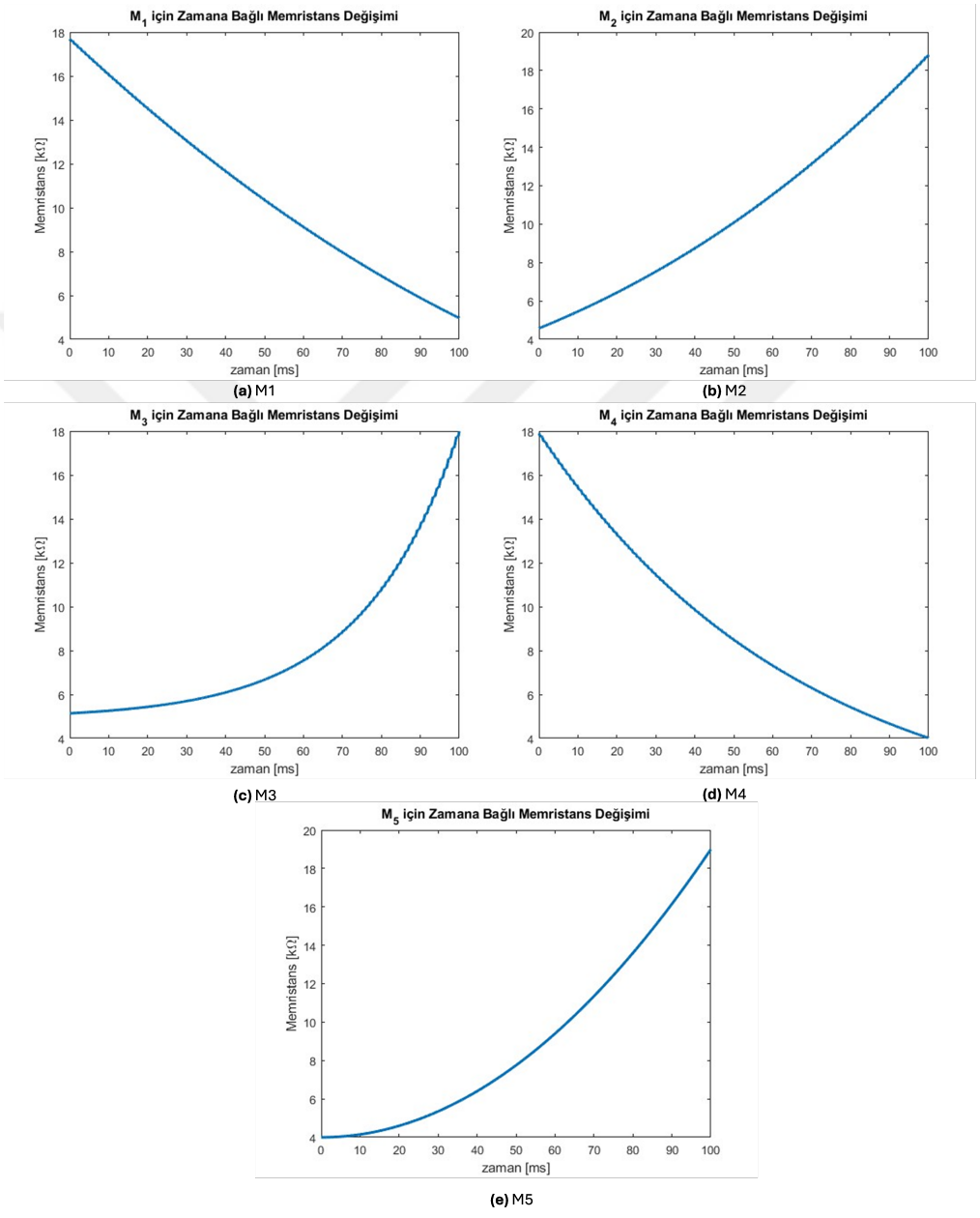
$$M_1(q) = (133 - 1.25 \times 10^6 \times q)^2 \tag{2.3a}$$

$$M_2(q) = (16.6 + 2 \times 10^5 \times q)^3 \tag{2.3b}$$

$$M_3(q) = 4900 + 240 \times e^{8\times10^4\times q} \tag{2.3c}$$

$$M_4(q) = \frac{18000}{\sqrt{0.01 + e^{6\times10^4\times q}}} \tag{2.3d}$$

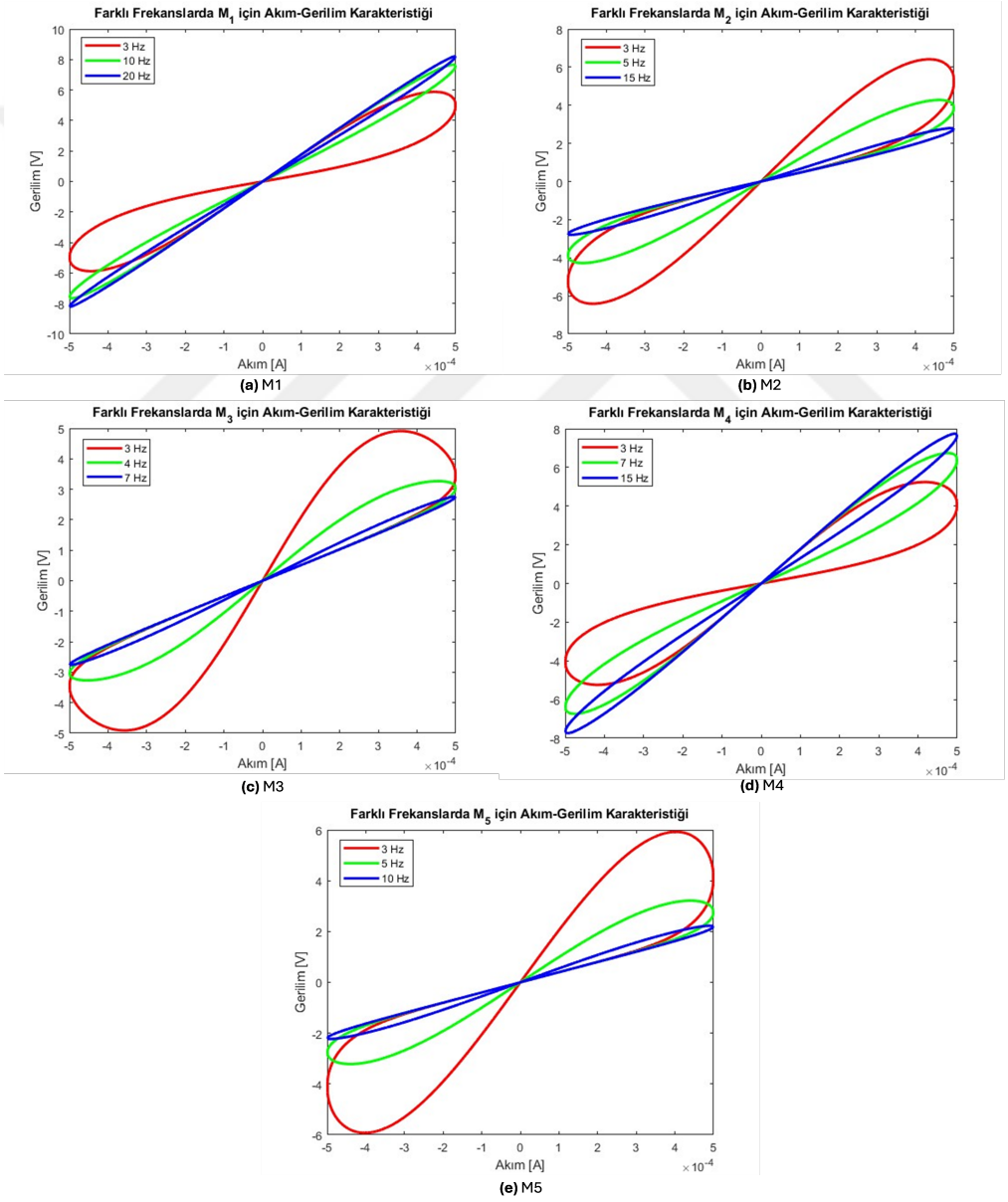
$$M_5(q) = 4000 + 6 \times 10^6 \times q^2 \tag{2.3e}$$



Şekil 2.2 Farklı memristör modellerine ait memristans-zaman grafikleri

Farklı memristörlere ait matematiksel modellerinin karakteristiği elde edildikten sonra bu modellerin gerçekten memristif davranış gösterip göstermediği incelenmiştir. Bunun için tüm memristör modelleri için akım-gerilim histerezis eğrisi kullanılmıştır. Tüm memristör modelleri için farklı olmak üzere 3 farklı

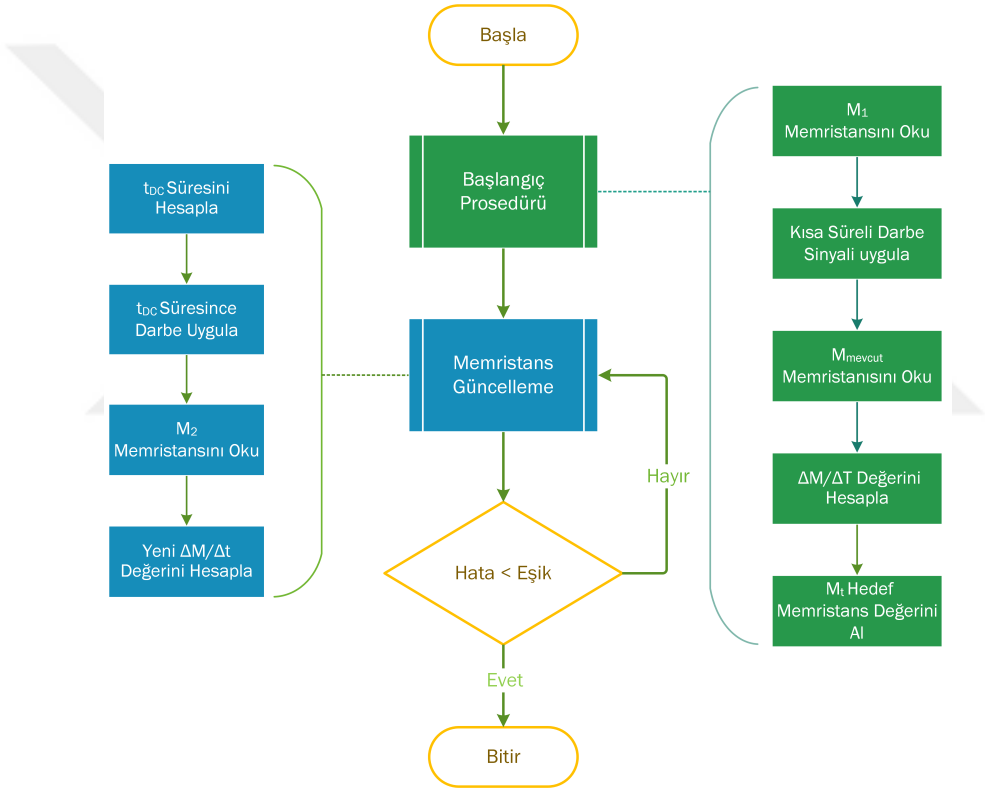
frekans seçilmiş ve o seçilen frekanslarda histerezis eğrisi çizdirilmiştir. Farklı memristör modelleri için farklı frekanslarda oluşturulmuş histerezis eğrileri Şekil 2.3'te verilmiştir. Akım-gerilim histerezis eğrileri memristörlerin memristif davrandığı frekans bölgelerinin tespiti için kullanılabilmektedir. Frekans değeri arttıkça histerezis eğrileri lineer bir eğri haline dönüşmektedir [20]. Bu çalışmada kullanılan memristörlere ait matematiksel modellerin histerezis eğrilerinden de görüleceği üzere frekans değeri düşük tutulduğunda lineerlikten uzak bir davranış gösteren matematiksel modeller frekans değeri yükseldikçe lineer olmaya oldukça yaklaşmaktadır. Tüm matematiksel modeller için geçerli olan bu durum seçilen matematiksel modellerin memristif davranış sergilediğini ispat etmiştir.



Şekil 2.3 Farklı memristör modellerine ait akım-gerilim histerezis eğrileri

2.2 Parçalı Lineer Karakteristik (PLK)

Parçalı lineer karakteristik algoritmasında ardışık ölçülen iki memristans değeri sonrası oluşan lineer denklem kullanılarak güncel memristans değerinden hedef memristans değerine ulaşılması için gereken darbe sinyalinin süresi ve işareti hesaplanır ve bu darbe sinyali memristör elemanı üzerine uygulanır. Ulaşılan son memristans değeri ve bir önceki memristans ölçümü değeri ile hedefe ulaşılması için uygulanmış darbe süresi kullanılarak yeni eğim değeri hesaplanır ve bu eğim değeriyle yeni lineer denklem elde edilir. Hedeflenen memristans değerine ulaşabilmek için her adımda bu işlem tekrarlanır ve adaptif olarak eğim değeri güncellenerek parçalı lineer bir eşitlik elde edilir. Parçalı lineer karakteristik algortimasına ait blok diyagramı Şekil 2.4'te verilmiştir.

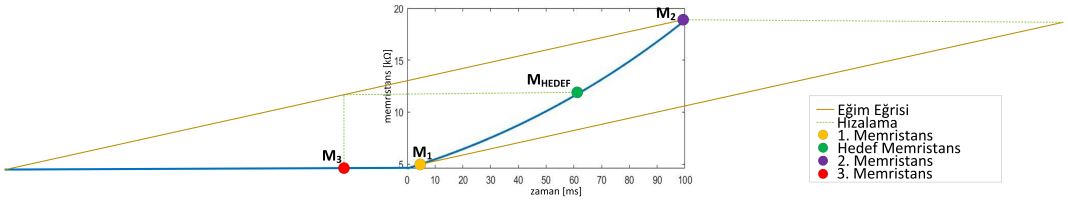


Şekil 2.4 Parçalı lineer karakteristik algoritması ile memristans güncellemesi

2.3 R_{ON} - R_{OFF} Osilasyonu (OSC)

Parçalı lineer yaklaşım algoritmasının kullanılması sırasında ilk seçilen t_{darbe} süresinin olması gerekenden daha büyük seçilmesi, memristans değerinin maksimum ve minimum memristans değerleri olan R_{ON} ve R_{OFF} değerleri arasında osilasyon yapmasına sebep olabilmektedir. Lineer olmayan memristans değişim karakteristiklerinde R_{ON} ve R_{OFF} değerlerine yakın olan bölgelerde eğim farkının oldukça fazla olduğu görülmektedir. Eğimin düşük olduğu bölgelerde hesaplanan

t_{darbe} süresi olması gereken süreden önemli ölçüde fazla olarak hesaplanmaktadır ve bu durumda darbe uygulanan süre boyunca memristans değeri limit değerine ulaşmakta ve limit değerinde sabit kalmaktadır. Ardından tam tersi yönde uzun süreli bir darbe uygulanarak memristans karakteristiğindeki diğer limit değerine ulaşılmaktadır. Bahsedilen probleme ait görsel Şekil 2.5'te gösterilmiştir.



Şekil 2.5 R_{ON} - R_{OFF} osilasyon problemi

Başlangıç prosedüründen sonra mevcut durumdaki memristans değeri M_1 olarak gösterilmiştir. Başlangıç eğimi hesaplanırken komşu iki memristans değeri kullanıldığından eğim değeri düşüktür. Bu sebeple hedef memristans değeri olan M_{HEDEF} değerine ulaşılması için gereken darbe sinyalinin süresini ifade eden t_{darbe} değeri olması gerekenden çok daha uzun bir değer olarak hesaplanır. Darbe sinyali uygulanan süre boyunca memristans değeri R_{OFF} değerine ulaşır. Güncellenmeden önce hesaplanan eğim değeri $\frac{M_2-M_1}{t}$ başlangıç eğimine oldukça yakın hesaplanır. Hesaplanan bu yeni eğim değeriyle M_{HEDEF} değerine ulaşılması için gereken darbe sinyal süresi t_{darbe} boyunca uygulanan darbe sinyali memristans değerini R_{ON} değerine ulaştırır. Bir sonraki adımda ise R_{ON} - R_{OFF} osilasyonu sona erer. Bununla birlikte memristör karakteristiğinin ideal olmama seviyesi yükseldikçe R_{ON} - R_{OFF} osilasyonunu bitirmek için daha fazla adım gerekmektedir. Bu durumdan kaçınmak için mevcut memristans değeri hafızadan silinmek üzere olan bir önceki memristans değeriyle karşılaştırılır. Bu iki memristans değerinin eşit olduğu durumda osilasyondan kaçınabilmek için hesaplanan t_{darbe} süresi belirlenen bir k katsayısı kadar azaltılır. Karşılaştırma yapıldıktan sonra eski memristans değerinin hafızada tutulmasına gerek yoktur. Bu karşılaştırma süreci sayesinde istenilen herhangi bir memristans değerinden başlanılarak hedeflenen herhangi bir memristans değerine ulaşılması sağlanmış olur. Aynı zamanda hedefe ulaşılması için tekrarlanan döngü sayısında bir düşüşe sebep olarak güç tüketimini ve işlem maliyetini de düşürmeye yardımcı olur.

2.4 Eğim Değişimi Hızlandırıcı (EDH)

Memristör karakteristiğine bağlı olarak darbe sinyali uygulanan süre boyunca memristans değerinde, minimum R_{ON} değerinden maksimum R_{OFF} değerine veya maksimum R_{OFF} değerinden minimum R_{ON} değerine bir değişim gözlenebilir.

Bu değişimler süresince mevcut memristans değerinden hedef memristans değerine ulaşmaya çalışırken oluşan hatanın hedef memristansa göre yönü bir hızlandırıcı olarak kullanılabilir. Darbe sinyali uygulanan t_{darbe} süresinden önceki hatanın yönü ile uygulandıktan sonraki hatanın yönü yaklaşık olarak eşit olduğu durumda, uygulanan t_{darbe} süresi yetersizdir ve bu sürenin arttırılabilir olduğunu göstermektedir. Bu durum ardışık olarak belirlenen iki eğim değerinin arasındaki farkın daha büyük olması gerektiği sonucunu doğurmaktadır. Başka bir deyişle eğer bu hataların yönü aynı ise ortaya çıkacak olan eğim değişimi bu hatalarla orantılı olarak arttırılabilir. Bir sonraki eğim değerinin belirlenmesi için kullanılan eşitlik Denklem 2.4'te verilmiştir. Denklemde yer alan m_l bir sonraki eğim değerini ifade etmektedir. Denklemde yer alan M_t hedef memristansı, M_l mevcut memristansı, M_{l-1} bir değer güncelleme süreci önceki memristans değerini, M_{l-2} ise iki değer güncelleme süreci önceki memristans değerini göstermektedir. Denklemdeki t_l bir önceki değer güncelleme sürecinde kullanılan darbe sinyali süresini ifade ederken, t_{l-1} ise iki önceki değer güncelleme sürecinde kullanılan darbe sinyali süresini ifade etmektedir.

$$m_l = \frac{\frac{M_l - M_{l-1}}{t_l}(M_t - M_l) + \frac{M_{l-1} - M_{l-2}}{t_{l-1}}(M_t - M_{l-1})}{(M_t - M_l) + (M_t - M_{l-1})} \tag{2.4}$$

Benzer bir şekilde darbe sinyali uygulanan t_{darbe} süresinden önceki hatanın yönü ile uygulandıktan sonraki hatanın yönü farklı olduğu durumda, uygulanan t_{darbe} süresinin olması gerekenden uzun olduğu ve bu sürenin azaltılabilir olduğunu göstermektedir. Bu yöntemle hedef memristansa ulaşılması için gerekli döngü sayısı büyük ölçüde azaltılabilir ve programlama süresinde bir düşüş sağlanabilir. Buna bağlı olarak enerji tüketiminde de bir düşüş gözlemlenebilir.

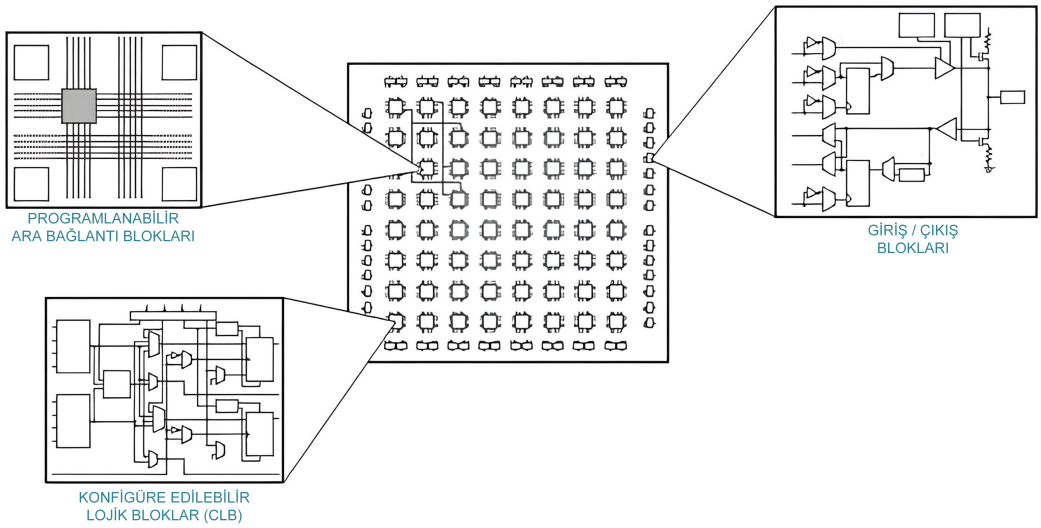
2.5 FPGA

FPGA içerdiği lojik elemanlar ve bu elemanlar arasında yeniden konfigüre edilebilir bağlantılar sayesinde esnek bir tasarım imkanı sunabilen bir geliştirme kartı yapısıdır. Kullanıcılar donanım tanımlama programlama dillerini kullanarak bu geliştirme kartı içerisindeki bağlantıları ve elemanları konfigüre edebilir. Mikroişlemci tabanlı sistemlerin aksine paralel işlem yapabilmeleri, düşük güç tüketimleri ve yüksek frekanslarda işlem yapabilme yetenekleri sayesinde sinyal işleme, kontrol sistemleri, yüksek hızlı haberleşme sistemleri ve yapay sinir ağlarının donanımsal gerçeklemeleri gibi konularda avantaj sağlayabilmektedirler. Bunun yanı sıra tasarım sürelerinin uzun olması ve yüksek karmaşıklık içermesi kullanıcı açısından dezavantajlı bir durumdur.

FPGA geliştirme kartının yüksek frekanslarda çalışabilmesi, paralel işlem yapabilme yetisine sahip olması ve düşük güç tüketimi gibi avantajları nedeniyle bu çalışmada önerilen algoritmanın donanımsal olarak gerçeklenmesinde kullanılmasına karar verilmiştir. Sinaptik devrelerin karmaşıklık seviyesiyle beraber artan eleman sayısı ihtiyacı düşünüldüğünde paralel işlem yapabilme yeteneği FPGA geliştirme kartını öne çıkaran unsurlardan biri olmuştur. Ayrıca önerilen algoritmanın donanımsal gerçeklemesinde güç tüketimini doğrudan etkileyecek olan adım sayısı parametresi konusunda da FPGA geliştirme kartının yüksek frekanslarda çalışmasının pozitif etki sağlayacağı düşünülmektedir. Önerilen algoritmada memristans değerini değiştirmek için uygulanması gereken darbe sinyalinin süresi hesaplanırken ortaya çıkabilecek çok kısa süreli sinyalleri üretebilme konusunda FPGA geliştirme kartı mikroişlemci tabanlı geliştirme kartlarının önünde yer almaktadır.

2.5.1 FPGA Mimarisi

FPGA mimarisi lojik kaynaklar, bağlantı matrisi ve giriş/çıkış birimleri olmak üzere 3 temel başlık altında toplanabilir. Lojik kaynaklar; konfigüre edilebilir lojik blokları ve hafıza birimlerini içermektedir. Bağlantı matrisi olarak isimlendirilen bloklar donanım tanımlama dilleriyle programlanarak bloklar arası bağlantıları sağlayan bloklardır. Giriş/çıkış birimleri ise geliştirme kartının gerçek dünya ile iletişime geçmesini sağlayan tüm bağlantı noktalarına verilen genel isimdir. FPGA temel mimarisi Şekil 2.6'da gösterilmiştir.



Şekil 2.6 Temel FPGA mimarisi

- **Lojik Kaynaklar:** Lojik kaynaklar temel olarak LUT (Lookup Table, Bakma Tablosu) elemanları ve FF (Flip Flop) elemanları gibi lojik işlem bloklarını

ve BRAM (Block Random Access Memory, Blok Rastgele Erişimli Bellek) gibi hafıza yapılarını içermektedir. Bu kaynaklar her FPGA kartında farklı sayılarda ve özelliklerde bulunmaktadır ve konfigürasyona bağlı olarak bu elemanlar kullanılabilmektedir.

- **Bağlantı Matrisi:** Bağlantı matrisi FPGA içerisindeki blokların donanım tanımlama dili aracılığıyla kodlanarak birbirlerine bağlanabilmesine olanak sağlayan blokların ismidir.

- **Giriş/Çıkış Birimleri:** Giriş/çıkış birimleri FPGA tümdevresinin dış dünya ile bağlantı kurabilmesinin sağlayan blokların genel ismidir. SPI (Serial Peripheral Interface, Seri Çevresel Arayüz) haberleşme protokolü, ADC (Analog to Digital Converter, Analogdan Dijitale Dönüştürücü) bağlantı noktası, ethernet portu gibi evrensel ve önceden tanımlanmış arayüzlerin yanı sıra, kullanım amacına göre konfigüre edilebilen genel amaçlı bağlantı arayüzlerini de içermektedir.

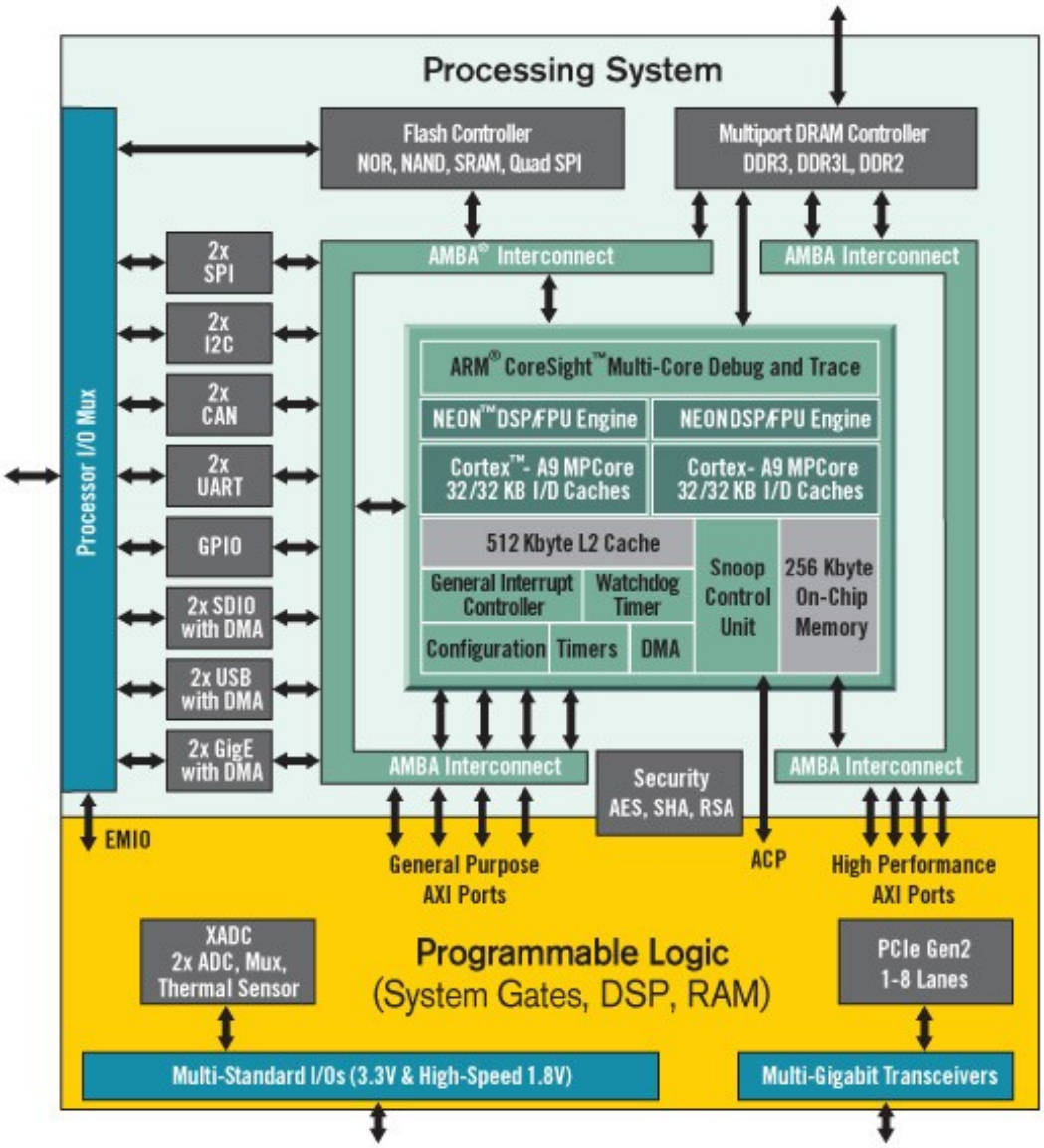
2.5.2 ZYNQ Mimarisi

Bazı FPGA geliştirme kartları programlanabilir lojik (PL) bloklarından oluşan mimari yapısına ek olarak ardışıl işlemler için avantaj sağlayan geleneksel mikroişlemci tabanlı bir işletim sistemi (PS) mimari yapısı kullanır. Xilinx şirketinin geliştirme kartlarında kullandığı bu mimari ZYNQ olarak isimlendirilir.

Arm tabanlı mikroişlemcilerin kullanıldığı bu yapılar kullanılarak donanım tanımlama dilleri yerine C programlama dili ile FPGA konfigüre edilebilir. Programlanabilir lojik mimarisinin işletim sistemi mimarisinde kullanılabilmesi için IP (Intellectual Property) blokları kullanılabilmektedir. Bu sayede programlanabilir lojik mimarisi içerisinde bulunan bazı bloklar ve arayüzler geliştirme kartı içindeki haberleşme sistemleri sayesinde işletim sistemi mimarisi üzerinden kontrol edilebilmektedir. Zyno Z-7010 FPGA kartı içerisinde bulunan ZYNQ mimarisi Şekil 2.7'de gösterilmiştir.

2.5.3 FPGA Tasarımı

Önerilen algoritmanın donanımsal gerçeklemesinde FPGA geliştirme kartı kullanılmıştır. Yapılan tasarımda memristans ölçümü ve memristans değer güncellemesi için iki ana blok bulunmaktadır. Memristans ölçüm bloğunda amaç değeri güncellenecek olan memristöre ait memristans değerini ölçebilmek, yine aynı şekilde değer güncellemesi denemesi yapıldıktan sonra tekrar memristörün memristans değerini ölçerek sistemi doğrulamaktır. Bunu yapabilmek için



Şekil 2.7 Zybo Z-7010 FPGA kartı ZYNQ mimarisi

memristör karakteristiğine bağlı olarak frekans ve genlik değeri değişebilen bir sinüzoidal dalga üretebilmesi gerekmektedir. Üretilen sinüzoidal dalga memristans değerinin korumasını sağlarken bir direnç ve bir memristör elemanıyla oluşturulan klasik gerilim bölücü devresi yardımıyla memristör üzerine düşen gerilimi kullanarak memristans değerininin elde edilmesine olanak sağlamaktadır. Memristans değerinin ölçülmesinde kullanılacak ikinci aşama olan memristans üzerine düşen gerilimin ölçülmesi kısmında bir direnç ve bir memristörün seri bağlanmasıyla oluşturulan gerilim bölücü devreye sinüzoidal sinyal uygulanırken iki elemanın üzerine düşen toplam gerilim ve memristans elemanı üzerine düşen gerilim ölçülür. Daha sonra bu ölçümler kullanılarak gerilim bölücü devre eşitliği kullanılarak memristans değeri elde edilmiş olur. Memristans değerinin hesaplanmasında ilişkin eşitlik Denklem 2.5'te verilmiştir.

$$M = \frac{V_{memristor} * R_{referans}}{(V_{toplam} - V_{memristor})} \tag{2.5}$$

FPGA geliştirme kartında hazırlanan tasarıma ait blok şeması Şekil 2.8'de gösterilmiştir. Blok şemasında da görüldüğü üzere iki ana bloğun altında farklı görevler için kullanılan toplam 3 alt blok bulunmaktadır. Bu bloklar darbe sinyal üretimi, sinüzoidal sinyal üretimi ve analog sinyal okuma bloklarıdır.

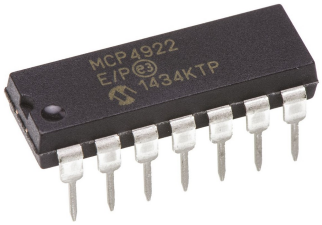


Şekil 2.8 FPGA tasarımı blok şeması

- **Darbe Sinyal Üretimi Alt Bloğu:** Darbe sinyal üretimi bloğunda memristans değerinin güncellenmesi için gerekli olan darbe sinyali üretilir. Bu darbe sinyali üretilirken FPGA geliştirme kartının genel amaçlı giriş/çıkış blokları kullanılmıştır. Bu çıkışlar kart üretiminde tanımlanmış belirli standartlara göre sinyal üretebilmektedir. Bu çalışmada kullanılan çıkış pini 3.3 volt genliğe sahip sinyal üretebilmektedir. Ayrıca ZYNQ mimarisinin kullanıldığı bu projede, FPGA geliştirme kartının programlanabilir lojik kısmında bulunan bu çıkış pinini kontrol etmek için AXI (Advanced eXtensible Interface, Gelişmiş Genişletilebilir Arayüz) haberleşmesi kullanılmaktadır. Böylece FPGA geliştirme kartının sağladığı yüksek frekanslı saat sinyali ile çıkışlar kontrol edilebilmektedir. Yüksek frekansın sağladığı yüksek anahtarlama hızı güç tüketimini düşürmeyi amaçlayan bu çalışma için oldukça önemli bir özelliktir.

- **Sinüzoidal Sinyal Üretimi Alt Bloğu:** Sinüzoidal sinyal üretim bloğunda memristans değerinin ölçümü sırasında uygulanması gereken sinüzoidal sinyal üretilir. Bu sinyalin memristör karakteristiğine bağlı olarak farklı genlik ve frekanslarda olması gerekmektedir. FPGA geliştirme kartı sayısal bir sistem olduğu için sinüzoidal sinyaller gibi analog sinyalleri üretme konusunda harici entegre devrelere ihtiyaç duymaktadır. Bu çalışmada kullanılacak olan sinyalin frekans ve genlik gereksinimleri düşünüldüğünde Microchip Technology şirketi tarafından üretilmiş 12 bit dijital giriş çözünürlüğüne sahip MCP4922 model numaralı DAC (Digital to Analog

Converter, Dijitalden Analoga Dönüştürücü) entegre devresi kullanılmıştır. Çift kanallı çıkış imkanı sağlayan MCP4922 entegre devresi analog çıkış gerilimi için harici bir referans girişine sahiptir. Bu sayede entegre devre esnek tasarımlara olanak sağlayabilmektedir. Kullanılan dijitalden analoga dönüştürücü entegresi Şekil 2.9'da verilmiştir.

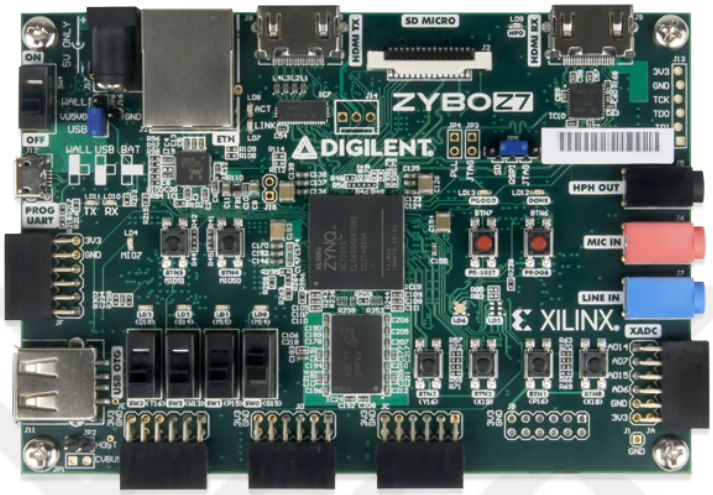


Şekil 2.9 Kullanılan 14-DIP kılıfına sahip MCP4922 dijitalden analoga dönüştürücü entegresi

MCP4922 dijitalden analoga dönüştürücü entegre devresi kullanılan sayısal sistemle haberleşmek için SPI haberleşme porotokolünü kullanmaktadır. Kullanılan entegre devre 12 bit giriş çözünürlüğünün yanı sıra 4 bitlik bir kontrol dizisi kullanmaktadır. Toplamda 16 bitlik paketler halinde iletişim sağlayan entegre devrenin ilk biti entegre devre içindeki hangi kanalın kullanılacağının seçilmesini sağlar. İkinci bitinde gönderilen verinin entegre devre içerisinde bulunan geçici bellekler aracılığıyla tamponlanıp tamponlanmayacağına karar verir. Üçüncü bitinde x1 ve x2 olmak üzere çıkış kazancı ayarlanabilmektedir. Dördüncü bitinde ise sistemin yazılımsal olarak sıfırlanması için bir bitlik bir kontrol biti mevcuttur. 16 bitten oluşan bir haberleşme paketinin geriye kalan 12 bitlik kısmı ise veri için ayrılmıştır [21].

- **Analog Sinyal Okuma Alt Bloğu:** Analog gerilim okuma bloğunda memristör değeri hesabı için gerekli olan gerilim ölçümleri gerilim bölücü devre üzerinden okunmaktadır. Dijital bir sistem olan FPGA geliştirme kartında bu değerlerin kullanılabilmesi için analogdan dijitale dönüştürücü entegre devresi gerekmektedir. Bu çalışmada kullanılan geliştirme kartı olan Zybo Z-7010 FPGA geliştirme kartında dahili olarak bir analogdan dijitale dönüştürücü entegresi bulunmaktadır. Harici bir başka entegre devre yerine dahili sistem kullanılması bu çalışmanın düşük güç tüketimi hedefine katkı sağlayan bir unsur olmuştur. 4 kanallı analogdan dijitale dönüştürücü entegre devresi bulunan FPGA geliştirme kartında bu kanalların girişinde kart üretim standartları gereği maksimum 3.3 volt analog sinyal ölçülebilmektedir. Bu çalışma kapsamında kullanılacak genlik değerleri hesaba katıldığında çalışmanın bu kısıtlamalardan etkilenmediği görülmüştür.

Daha önce de belirtildiği üzere bu çalışmanın FPGA tasarımı kısmında yapılan tasarımlar Xilinx tarafından üzertlmiş olan Zybo Z-7010 FPGA geliştirme kartı üzerinde gerçekleştirilecektir. Programlanabilir lojik kısmının yanı sıra ZYNQ mimarisine sahip olması ve giriş/çıkış arayüzleri hesaba katıldığında bu çalışmadaki isterleri sağlayacağı kanaatine varılmıştır. Zybo Z7010 geliştirme kartının üstten görünüşü Şekil 2.10'da verilmiştir.

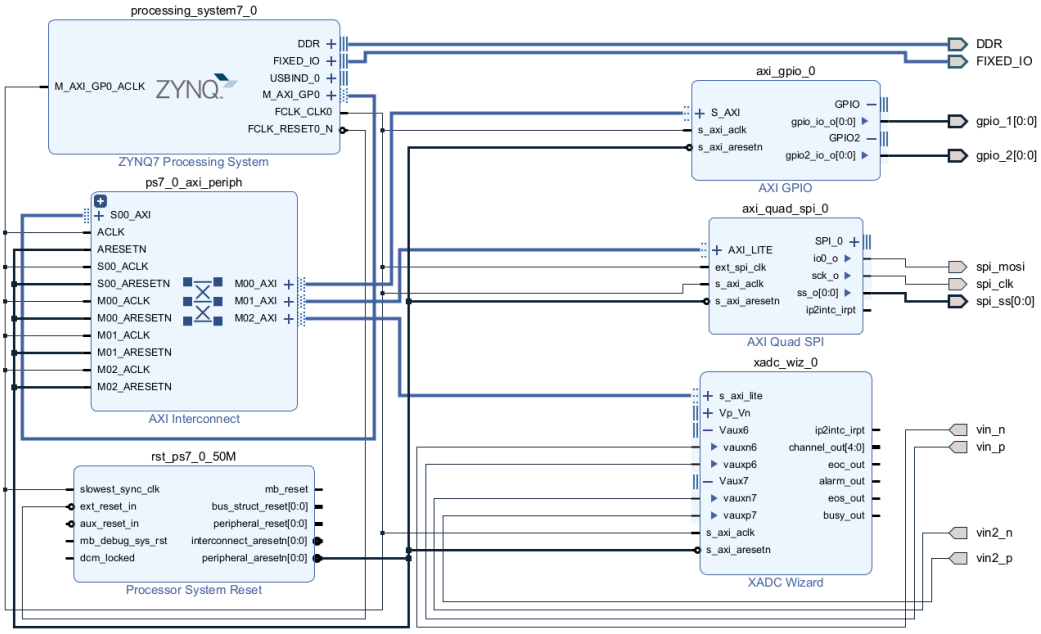


Şekil 2.10 Zybo Z-7010 geliştirme kartının üstten görüntüsü

Zybo Z-7010 geliştirme kartı kullanılarak oluşturulan FPGA tasarımı genel olarak ZYNQ mimarisi üzerinde çalışacak şekilde oluşturulmuştur. Yapılan tasarımda programlanabilir lojik kısımda bulunan giriş/çıkış arayüzleri ve blokları işletim sistemine aktarmak amacıyla IP bloklar kullanılmıştır. Bu sayede paralel çalışabilme yeteneğinden faydalanılırken aynı zamanda sıralı işlemlerde avantaj sağlayan işletim sistemi mimarisinin avantajları da kullanılmıştır. Bu IP blokların işletim sistemine aktarılması amacıyla bir blok tasarım oluşturulmuş, oluşturulan bu blok tasarım donanımsal olarak işletim sistemi tarafına aktarılmıştır. Xilinx şirketi tarafından Xilinx geliştirme kartlarının programlanması için oluşturulmuş Vivado programı üzerinde hazırlanmış sistemin blok diyagramı Şekil 2.11'de verilmiştir.

Tasarlanan sistem işletim sistemi olarak ZYNQ mimarisini kullanmaktadır. AXI Interconnect IP bloğu ZYNQ işletim sisteminin diğer IP bloklarla ve dolayısıyla programlanabilir lojik mimarisiyle haberleşmesine olanak sağlamaktadır. Processor Reset System IP bloğu ise sistemin senkron sıfırlama fonksiyonunu sağlamak amacıyla kullanılmıştır ve tüm blokları senkron şekilde sıfırlayabilme yeteneğine sahiptir.

Darbe sinyali üretimi için kullanılan AXI GPIO (General Purpose Input Output, Genel Amaçlı Giriş Çıkış) IP bloğu genel amaçlı giriş/çıkış pinlerinin kontrolü için



Şekil 2.11 Tasarlanan sistem ait blok diyagramı

kullanılmaktadır ve işletim sistemi ile AXI protokolü üzerinden haberleşmektedir. Sinüzoidal sinyal üretimi için kullanılan AXI QUAD SPI IP bloğu SPI haberleşmesi için gerekli konfigürasyonların yapılmasına olanak sağlamaktadır. Aynı zamanda bu IP blok ile 4 kanallı iletişim kurulabilmektedir. Bu çalışmada tek kanal kullanılmıştır. Analog gerilim değeri okuma için kullanılan XADC Wizard IP bloğu ise dahili analogdan dijitale dönüştürücü entegre devresinin kontrolü için eklenmiştir.

3
SİMÜLASYON SONUÇLARI ve DENEYSEL SONUÇLAR

Bu bölümde tasarlanan algoritmanın memristans güncelleme adım sayısına olan etkisi incelenerek sonuçlar paylaşılacaktır. Aynı zamanda daha önceki kısımlarda anlatılan FPGA tasarımına dair simülasyonlar ve algoritmanın uygulamaya geçirildiği sisteme dair sonuçlar paylaşılacaktır. Yapılan tasarıma ait alt blokların tamamı ayrı olarak test edilmiş olup, çıktılar da ayrı ayrı paylaşılacaktır. Test edilen ve simüle edilen alt bloklar sinüzoidal sinyal üretimi bloğu, darbe sinyali üretimi bloğu ve analog sinyal okuma bloğu olmak üzere 3 ayrı başlıkta incelenecektir.

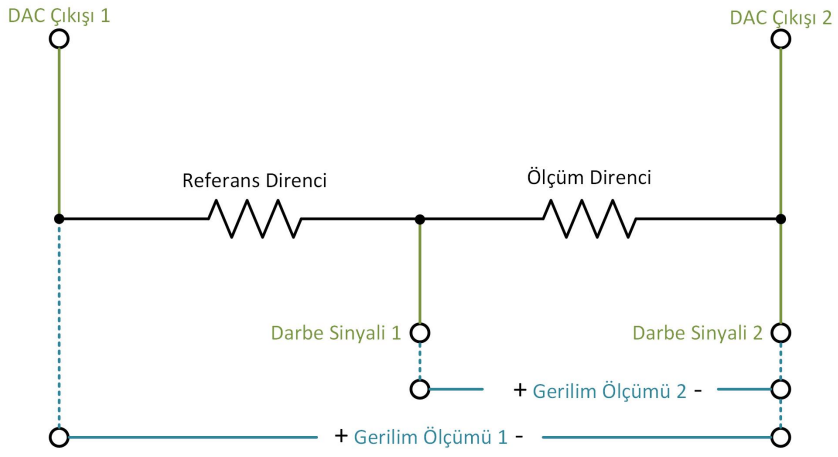
Algoritmanın adım sayısına etkisi test etmek amacıyla her bir memristör modeli için belirlenen bir başlangıç memristans değerinden 100 farklı memristans değerine ulaşabilmesi için gereken adım sayıları hesaplanmıştır. Aynı zamanda bir memrsitör emülatörü için de bu testler gerçekleştirilmiştir. Bu değer güncelleme sürecinde hata eşik değeri %5 seçilmiştir. Tablo 3.1'de tasarlanan algoritmanın farklı memristör modelleri ve memristör emülatörü için 100 farklı değer güncelleme sürecinin toplam adım sayıları paylaşılmıştır. Tabloda yer alan M_1, M_2, M_3, M_4 ve M_5 matematiksel memristör modellerini ifade ederken M_E ise memristör emülatörünü ifade etmektedir. Tabloda paylaşılan adım sayıları sonuçları 3 algoritmanın farklı kombinasyonlarla bir araya getirilerek elde edilmiş sonuçları göstermektedir. İlk koşulda yalnızca Parçalı Lineer Karakteristik (PLK) algoritması kullanılarak elde edilen sonuçlar tablonun ikinci satırında yer almaktadır. İkinci koşulda Parçalı Lineer Karakteristik (PLK) algoritmasına ek olarak R_{ON}-R_{OFF} Osilayonu (OSC) algoritması eklenerek elde edilen sonuçlar tablonun üçüncü satırında verilmiştir. Üçüncü ve son kombinasyonda ise bu iki algoritmaya ek olarak Eğim Değişimi Hızlandırıcı (EDH) algoritması eklenerek adım sayıları hesaplanmış ve tablonun dördüncü satırında paylaşılmıştır.

Şekil 3.1'de deneysel ölçümler için kurulmuş devre şeması verilmiştir. Bu devre şeması temel olarak bir referans direnci ve bir memristörü temsil eden ölçüm

Tablo 3.1 Maksimum %5 hata ile 100 farklı değer güncelleme sürecindeki toplam okuma-yazma döngüsü sayıları

Kullanılan Algoritma	M_1	M_2	M_3	M_4	M_5	M_E
PLK	167	247	770	203	829	194
PLK+OSC	167	247	520	203	587	194
PLK+OSC+EDH	159	247	463	192	577	191

direncinden oluşmaktadır. Bu gerilim bölücü devre yapısı üzerinde sinüzoidal sinyal oluşturabilmek adına devre yapısının iki ucuna iki farklı DAC entegre çıkışı bağlanmıştır. Buradaki amaç negatif sinyal üretemeyen DAC entegresinin çıkışları arasındaki faz farkının yardımıyla farksal sinyal oluşturmak ve bu farksal sinyal sayesinde memristör elemanını sembolize eden ölçüm direnci üzerinde sinüzoidal sinyal oluşturmaktır. Gerilim ölçümü yapılan iki farklı noktadan ilkinde referans direnci ve ölçüm direnci üzerinde oluşan toplam gerilim ölçülmektedir. Ölçüm yapılan ikinci noktada ise amaç sadece ölçüm direnci üzerinde oluşan gerilimi elde etmektir. Daha sonra ölçülen bu iki gerilim değeri ve referans direnci değerini kullanarak ölçüm direncinin değeri elde edilmektedir. Darbe sinyali uygulanan iki nokta mevcuttur. Bu iki nokta memristör elemanını sembolize eden ölçüm direncinin iki ucuna bağlanmıştır. Buradaki amaç ise iki darbe sinyali çıkışını ayrı ayrı kontrol ederek ölçüm direnci üzerinde hem pozitif hem negatif darbe sinyali üretebilmektir.

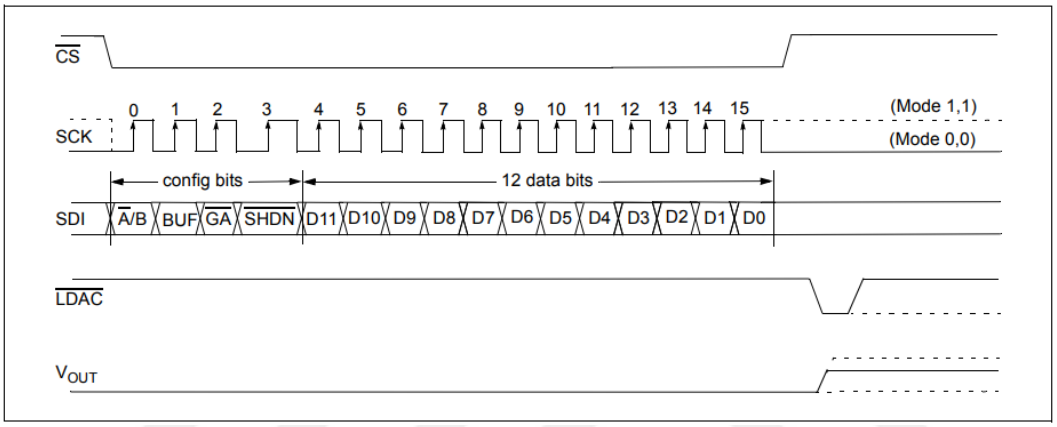


Şekil 3.1 Deneysel ölçümler için kurulmuş devre şeması

3.1 Sinüzoidal Sinyal Üretimi

FPGA tasarımı bölümünde genel bilgilerin verildiği sinüzoidal sinyal üretim bloğu, sayısal bir sistem olan FPGA geliştirme kartı kullanılarak analog sinyal üretilmesi işlevini yerine getirmektedir. Kullanılan geliştirme kartı olan Zybo Z-7010 kartında

dahili olarak bir dijitalden analoga dönüştürücü entegresi bulunmadığı için bu fonksiyonu yerine getirebilmek adına harici bir dijitalden analoga dönüştürücü entegre devresi kullanılmıştır. FPGA geliştirme kartı üzerinde tasarım yapılırken Microchip Technology firması tarafından üretilmiş MCP4922 DAC entegresinin isterleri dikkate alınmıştır. Aynı zamanda bu entegre seçilirken harici bir güç devresi veya dönüştürücü devresi kullanılmaması için gerilim seviyeleri ve haberleşme protokolü uygunluğu düşünülerek seçim yapılmıştır. Kullanılan harici DAC entegresi dijital sinyal girişindeki verileri SPI haberleşme protokolü üzerinden kabul etmektedir. Şekil 3.2'de kullanılan DAC entegresinin bir paket veri için zaman diyagramı verilmiştir [21].



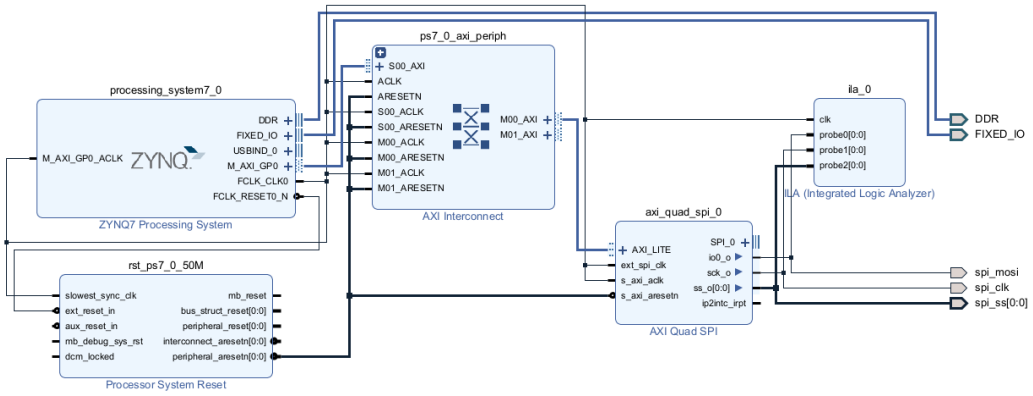
Şekil 3.2 MCP4922 DAC entegresi için yazma komutu zaman diyagramı

Yapılan tasarımda sinüzoidal sinyal üretimi için 12 bitlik verilerden oluşan bir bakma tablosu oluşturulmuş ve bu tablodaki her bir değer 4 bitlik konfigürasyon bitleriyle birleştirilerek 16 bitlik veri paket halinde gönderilecek şekilde ayarlanmıştır. Bakma tablosuna yazılan veriler bir periyot boyunca sinyalin alacağı değerleri oluşturmaktadır. Bakma tablosundaki değer sayısı üretilecek olan sinüzoidal sinyalin frekansını belirlemektedir. SPI haberleşmesinin frekans değeri bir bitlik veri aktarımı için belirlenmiş frekans değerini vermektedir. Frekans değeri ayarlanırken bu kısıtlar hesaba katılmış ve bir periyot boyunca gönderilecek bit sayısı hesaba katılarak üretilen sinyalin frekans değeri elde edilmiştir. Denklem 3.1'de sinüzoidal sinyalin frekansının eşitliği verilmiştir.

$$f_{sin} = \frac{f_{SPI}}{16 * \ boyut(LUT)} \tag{3.1}$$

Yapılan tasarımı test etmek amacıyla Vivado programı üzerinden blok tasarımı oluşturularak işletim sisteminde programlanmak üzere Vitis platformuna aktarılmıştır. Simülasyon sonuçlarını görüntüleyebilmek adına ILA (Integrated Logic Analyzer, Entegre Lojik Analizör) modülü de tasarıma eklenmiştir. ILA

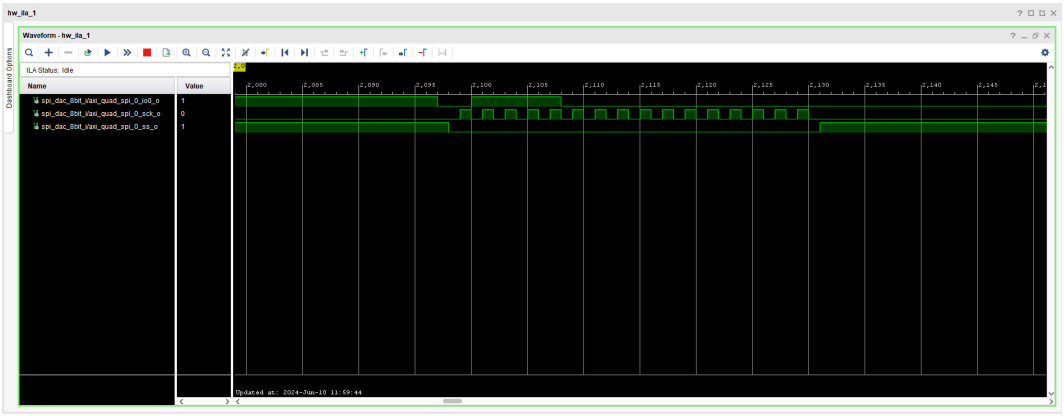
Xilinx şirketi tarafından tasarlanmış, FPGA geliştirme kartlarında bulunan entegre lojik analizördür. Bu kısımdaki kullanılan SPI gibi haberleşme protokollerinin testi için önemli bir bileşendir. Şekil 3.3'te ILA kullanılarak hazırlanan simülasyon blok tasarımı verilmiştir.



Şekil 3.3 Sinüzoidal sinyal üretimi için oluşturulmuş blok tasarımı

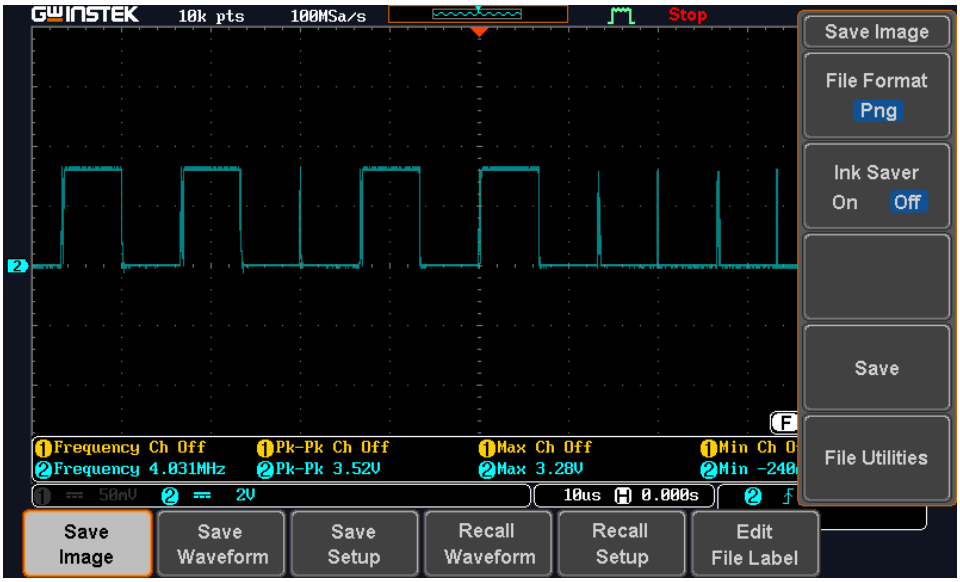
Blok diyagramda yer alan DDR ve FIXED_IO çıkışları FPGA geliştirme kartının dahili bağlantılarını sağlamak amacıyla Vivado programı tarafından otomatik olarak oluşturulmuştur. SPI haberleşme arayüzünde kullanılmak üzere oluşturulmuş olan spi_mosi, spi_clk ve spi_ss çıkışları ise geliştirme kartı üzerinde bulunan çıkışlara aktarılmıştır. Bu aktarım yapılırken pinlerin gerilim seviyeleri gibi parametrelerini belirleyen standartlar göz önüne alınmıştır. Bu çıkışlardan MOSI (Master Output Slave Input, Yönetici Çıkışı Bağımlı Girişi) FPGA geliştirme kartından DAC entegre devresine veri gönderilen kanalı temsil eder. Diğer bir çıkış olan CLK (Clock, Saat) ise SPI haberleşme protokolü için gerekli olan saat sinyalini temsil eder. Son çıkış olan SS (Slave Select, Bağımlı Seçimi) ise SPI ile iletişim kurulan DAC entegresinin aktivasyonunda kullanılır.

Şekil 3.4'te ILA kullanılarak yapılan gerçek zamanlı simülasyon sonuçları paylaşılmıştır. Paylaşılan simülasyon görselinde bakma tablosunda sinüzoidal sinyalin bir değeri için veri aktarımı diyagramı gösterilmiştir. Gerçek zamanlı olarak kontrol edildiğinde bakma tablosundaki tüm değerlerin sırasıyla çıkışa aktarıldığı teyit edilmiştir. Şekilde de görüldüğü gibi veri aktarımı başlamadan önce SS pini lojik '1' seviyesindedir, yani deaktive durumdadır. Transfer başlayacağı zaman SS pini lojik '0' seviyesine getirilir. SS pini lojik '1' seviyesindeyken saat sinyali çıkışa aktarılmaz, böylece güç tüketiminde bir düşüş sağlanmış olur. SS pini lojik '0' seviyesine geldikten sonra saat sinyali çıkışa aktarılır, bununla beraber saat darbesiyle senkronize bir şekilde gönderilecek veriler bit bit çıkışa aktarılır. 16 bitten oluşan veri paketinin son biti gönderildikten sonra SS pini yeniden lojik '1' seviyesine getirilir ve bir sonraki veri paketi aktarımına kadar sabit kalır.



Şekil 3.4 ILA kullanılarak oluşturulmuş SPI simülasyonu

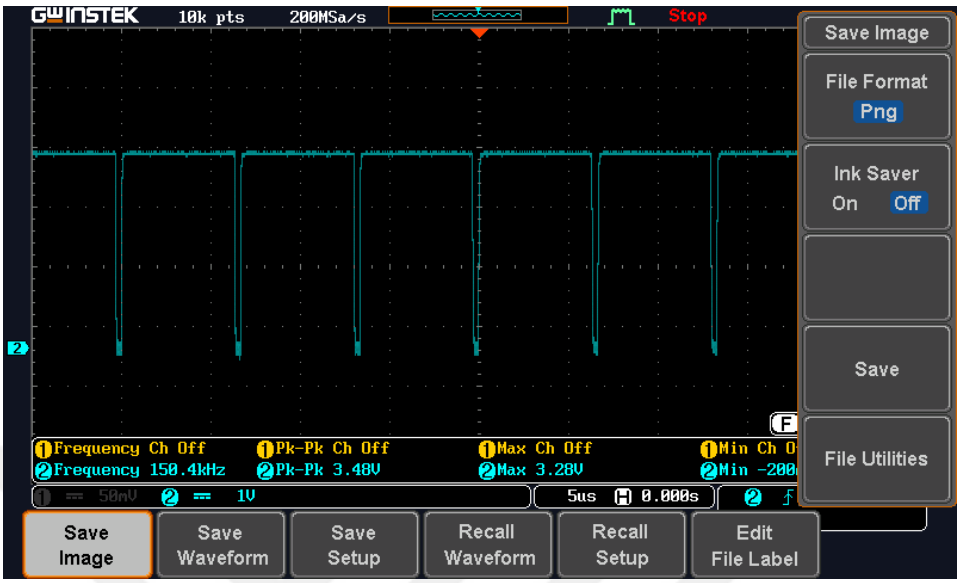
Önerilen tasarım laboratuvar ortamında test edilmiş ve deneysel sonuçlar osiloskop görüntüsü olarak kaydedilmiştir. Öncelikle gönderilen veriler kontrol edilmiş ve FPGA geliştirme kartından DAC entegre devresine giden sinyaller ölçülmüştür. Osiloskop ekranında görülen veriler Şekil 3.5'te paylaşılmıştır. Şekilde de görüleceği üzere 4.03 MHz SPI frekansıyla gönderilen verilerin genliği FPGA geliştirme kartı üreticisi tarafından belirlenen LVCMOS33 standardına uygun olarak 3.3 Volt olarak ölçülmüştür. Her bir periyot boyunca farklı sinyallerin geldiği işaretin 0b101001010000 sayısını gönderdiği görülmüştür. Verilen bakma tablosunda yer alan bu dizi sistemin doğruluğunu ispatlamıştır. Daha uzun süreli veriler de incelendiğinde bir sorun olmadığı gözlenmiştir.



Şekil 3.5 MOSI pininin osiloskop ekranında görülen çıktısı

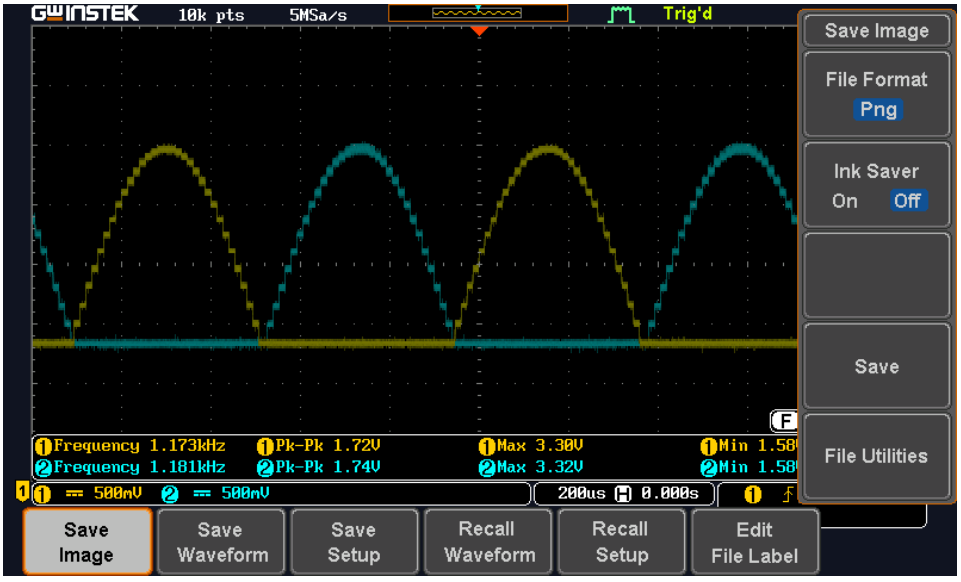
Ayrıca SS pininin çıkış işareti de gözlenmiş ve görülen veriler Şekil 3.6'da paylaşılmıştır. SS pininin bağlı olduğu çıkış da yine LVCMOS33 standardına sahip oldğuu için sinyalin 0-3.3 Volt arasında değiştiği görülmektedir. İşaretten

de görüldüğü üzere veri gönderilen kısa bir süre için SS pini lojik'0' seviyesine getirilerek veriler çıkışa aktarılmıştır. Ardından veri işleme kısmında güç tüketimini düşürmek için bir süreliğine SS pini lojik '1' seviyesine geri getirilmiştir.



Şekil 3.6 SS pininin osiloskop ekranında görülen çıktısı

Ayrıca DAC entegre devresinin bu giriş işaretine karşılık oluşturduğu çıkış sinyali gözlenmiş ve osiloskop görüntüsü Şekil 3.7'de verilmiştir. Sayısal bir sistem olan FPGA geliştirme kartı negatif genlikli gerilim üretmediği için iki farklı sinyal üretilmiş olup, bu sinyaller devre yapısı üzerinde bağlantıların yardımıyla sinüzoidal sinyale dönüştürülecektir.



Şekil 3.7 DAC entegresinin çıkışlarının osiloskop ekranında görülen çıktısı

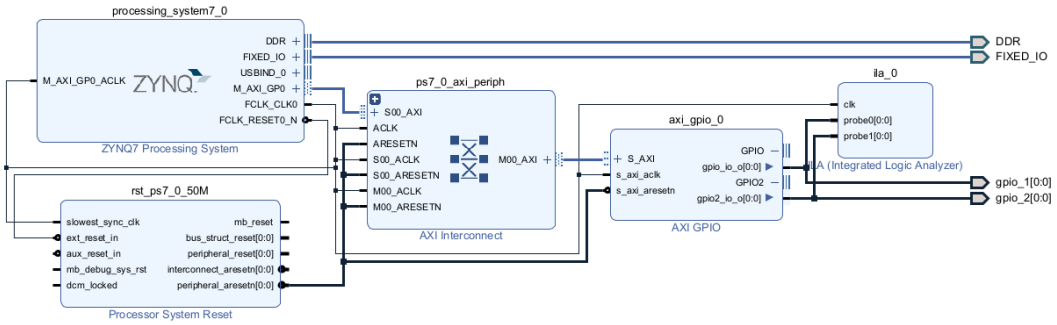
3.2 Darbe Sinyali Üretimi

Daha önceki kısımlarda genel bilgilerin verildiği darbe sinyali üretimi bloğu, değer güncelleme için gerekli olan darbe sinyalinin üretilmesi işleminin gerçekleştirildiği bloktur. Bu blokta üretilen darbe sinyalinin süresi algoritma üzerinde yapılan hesaplamalar neticesinde belirlenir. Memristör elemanının karakteristiğe bağlı olarak uygulanması gereken darbe sinyalinin işareti değişebilmektedir. Bu farklı işaretlerdeki sinyali üretebilmek amacıyla iki kanallı bir darbe sinyali üretilmiş ve bu iki kanal memristör elemanının iki ucuna bağlanarak gerektiği durumlarda negatif gerektiği durumlarda da pozitif darbe sinyali uygulanması sağlanmış olur. Değer güncelleme algroitmasındaki denklemlere bağlı olarak hesaplanan darbe sinyalinin süresi darbe sinyali üreteci bloğunun giriş parametresi olarak alınır ve o süre boyunca darbe sinyali belirlenen sinyal işaretinde uygulanır. İşlem maliyetini azaltmak ve güç tüketimini azaltmak adına hesaplanan kısa süreleri uyguyabilecek bir sistem tasarlamak kritik bir durumdur. Hesaplanan darbe süresi sistem tarafından uygulanabilecek minimum darbe süresinden daha kısa olduğu durumda sistem önce minimum süre miktarını ve uygulanması gereken sinyal süresi boyunca belirtilen sinyal doğrultusunda üretir, ardından ilk darbe sinyalinin tersi yönünde minimum darbe süresi miktarı kadar yeniden darbe uygular. Bu durum işlem süresi ve güç tüketimi açısından 1 adım fazla işlem yapılması durumuna yol açar. Bu durumların önüne geçebilmek adına seçilen FPGA geliştirme kartı yüksek saat frekansı ve güvenilir anahtarlama performansı sağlamaktadır. Yapılan testlerde sistemin 12 nanosaniye anahtarlama frekansına sahip olduğu görülmüştür.

Genel amaçlı giriş/çıkış pinlerinden çıkışların alındığı darbe sinyali üretim bloğunda, çıkışların bulunduğu portlar programlanabilir lojik mimarisi bölgesinde yer almaktadır. Bu çıkışların işletim sistemi mimarisi tarafından kontrol edilebilmesi için AXI_GPIO IP bloğu kullanılmıştır. AXI haberleşme protokolü ile geliştirme kartı içerisinde gerçekleşen haberleşme sayesinde işletim sisteminde yazılan kontrol algoritması programlanabilir lojik tarafında yer alan çıkış portunun anahtarlaması gerçekleştirilmiştir.

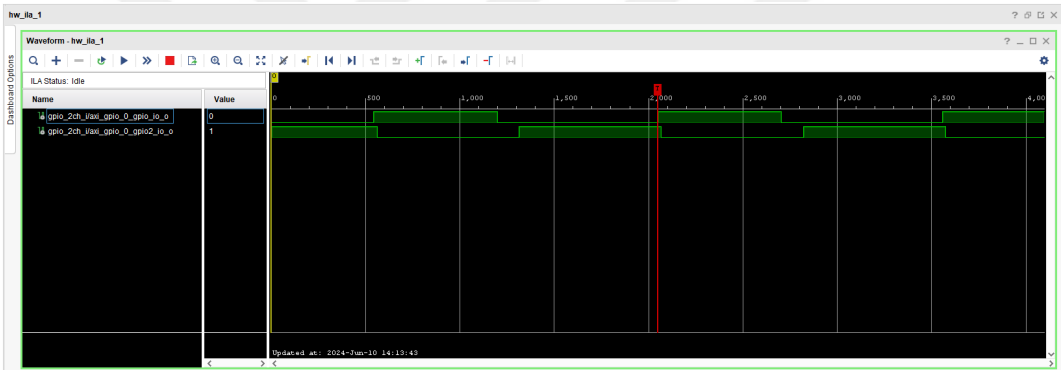
Yapılan tasarımı test etmek amacıyla Vivado programı üzerinde bir blok tasarımı yapılmış ve bu tasarım donanım olarak Vitis programına aktarılarak işletim sistemi mimarisi üzerinden kontrol edilmiştir. Darbe sinyallerini test etmek için de yine ILA IP bloğu kullanılmıştır. Şekil 3.8'de yapılan tasarıma ait blok diyagramı verilmiştir. Blok diyagramda yer alan DDR ve FIXED_IO çıkışları sistem tarafından konfigürasyon için otomatik olarak oluşturulmuş bağlantı çıkışlarıdır.

Şekil 3.9'da yapılan tasarıma ait simülasyon çıktıları verilmiştir. Simülasyonda iki kanalın anahtarlama hızları ve performansları test edilmiş olup, kontrol



Şekil 3.8 Darbe sinyali için oluşturulmuş blok diyagram

parametrelerine yüksek doğrulukla yanıt verdiği gözlemlenmiştir. Gereken kısa sürelerde hızlı anahtarlama yapan sistemde pozitif darbe sinyali üretmek için 1. kanal aktive edilirken 2. kanal deaktive edilerek gerilimi sıfırlanmıştır. Aynı şekilde negatif darbe sinyali için ise 1. kanal deaktive edilerek gerilimi sıfırlanırken 2. kanal aktive edilerek memristör elemanı üzerinde negatif gerilim oluşması sağlanmıştır.

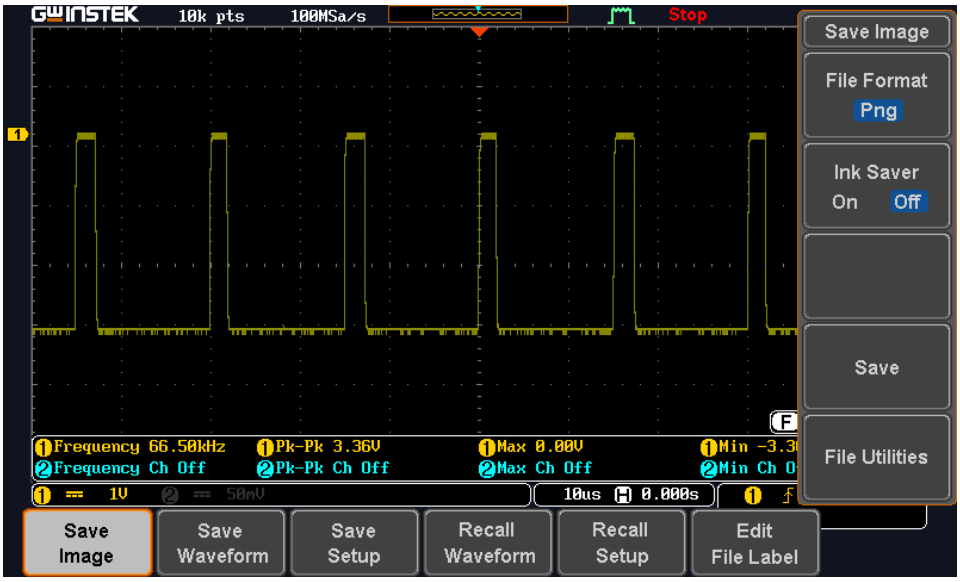


Şekil 3.9 Darbe sinyali üretimi için ILA üzerindeki simülasyon sonuçları

Önerilen yöntemde kullanılacak olan darbe sinyali laboratuvar ortamında test edilmiş ve elde edilen grafik Şekil 3.10'da paylaşılmıştır. Tasarım test edilirken anahtarlama süresi test edilmiş ve 2 μs gibi bir sürede sorunsuz anahtarlama yapıldığı görülmüştür.

3.3 Analog Sinyal Okuma

Önerilen yöntemde memristans değerinin elde edilmesi için yapılan gerilim ölçümleri analog değer okuma bloğunda gerçekleştirilmiştir. Bu ölçümlerin sayısal bir sistem olan FPGA geliştirme kartında kullanılabilmesi için dijital değerlere dönüştürülmesi gerekmektedir. Bu işlem için Zybo Z-7010 geliştirme kartında dahili olarak bulunan analogdan dijitale dönüştürücü entegresi kullanılmıştır. Bu dahili ADC entegre devresinde geliştirme kartı içerisindeki sıcaklık, geliştirme kartındaki gerilim seviyeleri gibi dahili ölçümler alınabildiği gibi aynı zamanda

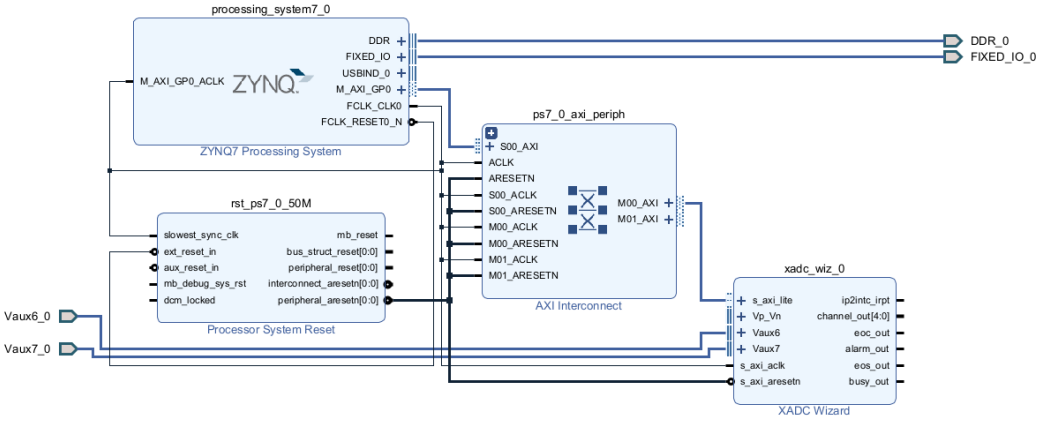


Şekil 3.10 Üretilen darbe sinyalinin osiloskop ekranındaki görüntüsü

harici olarak analog ölçüm yapabilen 4 kanal mevcuttur. Harici ölçüm kanallarında farksal ölçüm yapabilme olanağı sağlamaktadır. Aynı zamanda bu ADC entegresi 12-bit veri ölçüm çözünürlüğüne sahiptir ve 1 MSPS (Mega Sample per Second, Saniyede Başına Milyon Örnek) veri hızına sahiptir [22].

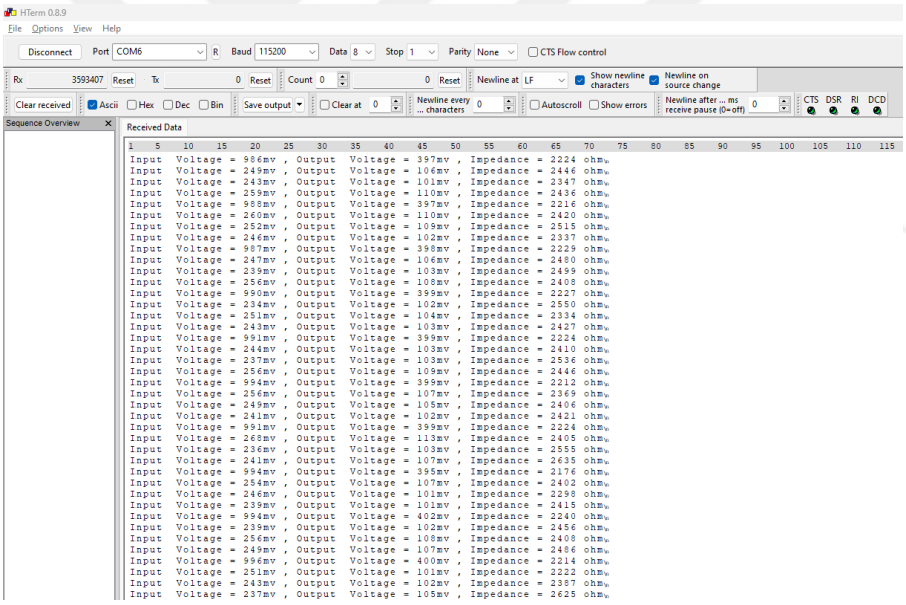
Önerilen yöntemde ölçüm için yapılan tasarıma göre FPGA geliştirme kartında bulunan XADC dahili ADC entegresinden alınan çıktıları kullanabilmek adına XADC Wizard IP bloğu kullanılmış, böylece alınan verilerin işletim sistemi mimarisine AXI haberleşme protokolü üzerinden aktarımı sağlanmıştır. Analog sinyal okuma bloğunu test etmek amacıyla bir blok diyagram hazırlanmış ardından bu tasarlanan blok diyagram kullanılarak sinüzoidal bir sinyal ölçülmüştür. Şekil 3.11'de tasarlanan blok diyagram verilmiştir. Verilen blok diyagramda yer alan Vaux6 ve Vaux7 girişleri harici gerilim ölçüm girişlerini temsil etmektir. Yukarıda da belirtildiği gibi memristans değerinin elde edilebilmesi için iki noktadan gerilim ölçülmesi gerekmektedir. Bu iki giriş kullanılarak yapılan ölçümler memristans hesabı için kullanılmıştır.

Analog sinyal okuma bloğunu test edebilmek adına laboratuvar ortamında iki adet dirençle klasik bir gerilim bölücü devre yapısı kurulmuştur. Bu devrede referans direnci olarak 3.3 kΩ değerinde bir direnç kullanılmıştır. Ayrıca ölçüm yapılacak gerilim bölücü devreye 1 Volt genlikli bir sinüzoidal sinyal uygulanmış, uygulanan sinyal nedeniyle oluşacak gerilimler ölçülerek yukarıda da belirtilen eşitlik kullanılarak memristör elemanının empedans değeri elde edilmiştir. Memristör elemanını simüle etmesi açısından bu ölçümlerde 2.2 kΩ değerinde bir direnç kullanılmıştır.



Şekil 3.11 Analog gerilim okuma için oluşturulmuş blok diyagram

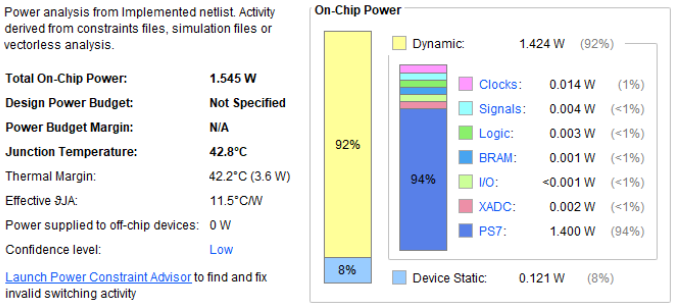
Şekil 3.12'de hterm seri monitör programında ölçülen gerilim değerleri ve bu gerilimler kullanılarak hesaplanmış empedans değeri görülmektedir. Ölçüm sonuçlarına bakıldığında farklı genlik durumlarında ölçüm sonuçlarının tutarlı olduğu gözlenmiştir.



Şekil 3.12 Seri monitör ekranından görülen gerilim ve empedans verileri

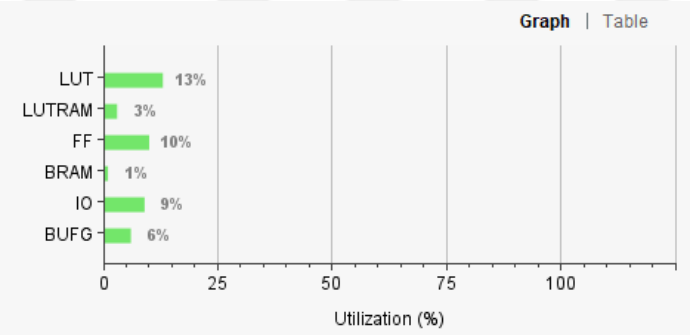
Bu çalışmada yapılan tasarımın güç tüketimi analizinde tasarımın toplam güç tüketimine ulaşmak için FPGA geliştirme kartının güç tüketimi analizi verileri ve harici DAC entegresinin maksimum güç tüketimi verileri hesaba katılmıştır. Şekil 3.13'te FPGA geliştirme kartı üzerindeki toplam güç tüketimine ait detaylı veriler paylaşılmıştır. Verilerden de görüleceği üzere güç tüketiminin büyük bir kısmı işletim sistemi mimarisine ait olup güç tüketiminin optimize edildiği blokların güç tüketimi oldukça düşük kalmıştır. DAC entegresinin maksimum besleme akımı 50 mA olup besleme gerilimi olan 3.3 V değeri de hesaba katıldığında maksimum 0.165 W olarak hesaplanmıştır [21]. Toplam güç tüketimi ise 1.71 W olarak

hesaplanmıştır.



Şekil 3.13 FPGA güç tüketimi tablosu

Son olarak bu prototip çalışma FPGA geliştirme kartı kaynak tüketimi açısından analiz edilmiştir. Tasarım yapılan Vivado yazılımı üzerinden prototip çalışmada kullanılan FPGA kaynakları grafiği elde edilmiş ve Şekil 3.14'te paylaşılmıştır.



Şekil 3.14 FPGA kaynak tüketimi tablosu

4
KARŞILAŞTIRMA VE TARTIŞMA

Bu bölümde memristif sinaptik devrelerde memristans değer güncellemesi için kullanılan yöntemler incelenerek önerilen yöntemle literatürde bulunan alternatif yöntemler avantajları ve dezavantajları paylaşılarak kıyaslanacaktır. Bu kıyaslama yapılırken güç tüketimi, maliyet, tasarımın esnekliği, değer güncellemesinin doğruluğu ve hızı gibi parametreler dikkate alınacaktır.

Ayrıca bu çalışmada önerilen ve prototip niteliğindeki tasarımın baha büyük ölçekli prototipleri gerçekleştirilirken oluşabilecek sorunlara çözüm önerilmiş ve teorik olarak bir yöntem önerilecektir.

4.1 Değer Güncelleme Yöntemleri Karşılaştırması

Bu kısımda literatürde yer alan ve memristör durum güncelleme amacıyla kullanılan farklı analog kontrol yöntemleri ayrıntılı olarak incelenecek ve sonuçları paylaşılacaktır. Literatür özeti kısmında paylaşılan 3 farklı yöntem güç ve kaynak tüketimi, memristör durum güncelleme hızı ve doğruluğu, maliyet ve tasarım esnekliği gibi kriterler göz önüne alınarak kıyaslanacak ve detaylı sonuçları paylaşılacaktır.

Literatürde memristör durum güncelleme amacıyla kullanılan bir yöntemde belirli bir DC değerin altındaki tüm memristans değerleri hafızaya kaydedilir ve kontrol amaçlı düzenli memristans ölçümü yapılmaktadır [5][10][11][12][13]. Önerilen bu yöntemde sürekli olarak memristans ölçümü yapmanın bir avantajı olarak herhangi bir durumdan herhangi bir hedef duruma ulaşılması için gerekli darbe sinyali süresi ve darbe sinyali yönü yüksek doğrulukla elde edilebilmektedir. Darbe sinyaline dair parametrelerde ulaşılan bu yüksek doğruluk memristans değer güncelleme sürecinin de benzer şekilde yüksek doğrulukla ve bunun bir getirisi olarak yüksek hızla gerçekleştirilebilmektedir. Ayrıca literatürdeki bu yöntemde bir sinaptik devrede bir memristör elemanı bulunması maliyet açısından da avantajlı bir durum yaratmaktadır. Ancak bu avantajların yanı sıra sürekli olarak memristans

ölçümü yapılması güç tüketimi üzerine olumsuz bir etki göstermektedir. Benzer şekilde ölçüm yapılan tüm değerlerin hafızada tutulması kaynak tüketimi açısından bir dezavantaj yaratmaktadır. Bu önerilen çalışmadaki en büyük dezavantaj ise hafızada tutulan memristans değerlerinin kullanılan memristör karakteristiğine özel değerler olmasıdır. Devre yapısı içerisinde yer alan memristör elemanlarından biri kullanılamaz hale geldiğinde aynı parametreleri kullanabilmek için parametrelerin yeniden belirlenmesi ve kaydedilmesi gerekir veya kullanılamaz duruma gelen memristörün aynı karakteristiğe sahip başka bir memristör ile değiştirilmesi gerekir. Bu durum literatürdeki bu yöntemin tasarım esnekliği açısından dezavantajlı bir duruma sahip olduğunu göstermektedir.

Literatürde yer alan bir diğer yöntemde ise memristans değer güncelleme süreci için geleneksel bir gerilim bölücü devre yapısı kullanmaktadır [14]. Kullanılan bu devre yapısı bir direnç ve bu dirence seri bağlanmış bir memristör elemanından oluşmaktadır. Bu devre yapısına yüksek frekanslı sinüzoidal sinyal uygulanarak memristör elemanının değeri korunurken memristans değeri elde edilmektedir. Memristans değerini elde etmek için seri bağlanan bu iki elemanın üzerindeki toplam gerilim ve memristör elemanının üzerine düşen gerilim ölçülür ve bu iki gerilim değeriyle referans direnç elemanının değeri kullanılır. Bu devre yapısında kullanılan memristör elemanının minimum ve maksimum memristans değerleri, güncel memristans değeri ve memristans karakteristiği bilgilerine ihtiyaç duyulmamaktadır. Bu sebeplerden dolayı güç tüketimi ve kaynak tüketimi diğer yöntemlere kıyasla düşük olmaktadır. Aynı zamanda bir sinaps hücresi için tek bir memristör elemanı ve bir direnç elemanı kullanılması bu yöntemin maliyetini düşük tutmaktadır. Aynı zamanda memristör modelinden bağımsız bir şekilde anlık verileri kullanarak memristans değer güncellemesi yapabilmesi bu yöntemin tasarım esnekliği açısından avantajlı olduğunu göstermektedir. Bu avantajlarının yanı sıra doğrusal olmayan karakteristiğe sahip elemanların anlık değerlerini kullanılarak hesaplanan darbe süresinde oluşabilecek hata diğer yöntemlere kıyasla daha yüksek olabilmektedir. Bu durum da memristans değer güncellemesinin yüksek hız ve yüksek doğrulukla gerçekleştirilebilmesine engel olmaktadır.

Literatürde memristör durum güncelleme amacıyla önerilen bir diğer yöntemde ise köprü bağlantı şeması kullanılmıştır [15]. Köprü bağlantı şemasının kullanılmasındaki temel amaç ise doğrusal olmayan bir karakteristik sergileyen memristör elemanının köprü bağlantı şeması kullanılarak doğrusal bir karakteristik sergilemesini sağlamaktır. Tasarlanan bağlantı şeması neticesinde 4 memristör elemanı kullanılarak elde edilen devrenin uçları arasında doğrusal karakteristik sergileyen bir memristör elemanı elde edilmiş olur. Aynı zamanda memristans değer güncellemesi için kullanılan devre şemasında aktif elemanlar kullanılmıştır.

Literatürdeki bu yöntem memristör karakteristiğinin doğrusal bir şekilde yeniden elde edilmesi sayesinde değer güncelleme sürecinde yüksek doğruluk sağlayabilmektedir. Bunun nedeni ise doğrusal bir karakteristik elde edildikten sonra uygulanacak darbe sinyalinin süresi ile memristans değişiminin doğru orantılı bir şekilde değişmesidir. Yine benzer şekilde uygulanacak darbe sinyali süresinin yüksek doğrulukla belirlenmesinin bir sonucu olarak değer güncelleme hızı da yüksek olmaktadır. Ayrıca önerilen bu yöntemde memristans karakteristiğinin hafızada tutulması gereksinimi olmadığı için kaynak tüketimi hafızada değerlerinin tutulduğu yönteme göre düşük olmaktadır. Güç tüketimi açısından incelendiğinde ise sürekli memristans ölçümü yapmaması, sürekli ölçüm yapılan yönteme göre nispeten daha düşük güç tüketimi yapmasını sağlamaktadır. Sinaptik devrenin tasarımı incelendiğinde bir sinaps hücresi için 4 memristör elemanı gerektirmesi ve aktif elemanlar kullanılarak oluşturulmuş bir devre şemasına sahip olması maliyet açısından bir dezavantaj yaratmaktadır. Aynı zamanda köprü bağlantı şeması aracılığıyla elde edilen doğrusal karakteristik memristör elemanının karakteristiğine bağlı olduğundan memristör elemanı değiştiğinde sonuçta elde edilen karakteristik de değişebilmektedir. Bu durum da tasarım esnekliği açısından dezavantajlı bir duruma sebep olmaktadır.

Tablo 4.1'de literatürde yer alan yöntemlerin güç tüketimi, kaynak tüketimi, güncelleme hızı, güncelleme doğruluğu, maliyet ve tasarım esnekliği açısından kıyaslaması paylaşılmıştır. Kıyaslanan yöntemlerden Sürekli Ölçüm Yöntemi (SÖY), Köprü Tekniği (KT), Gerilim Bölücü Devre Yapısı (GBDY) yöntemlerine ait değerlendirmeler tablonun ikinci, üçüncü ve dördüncü sütunlarında paylaşılmıştır. Tablodaki değerler yöntemlerin birbirlerine göre kıyaslamasını içermektedir.

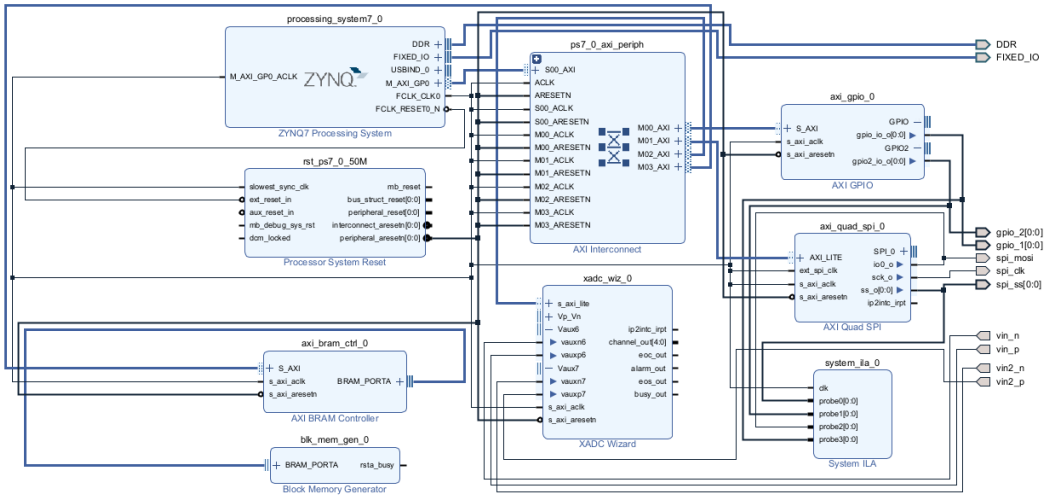
Tablo 4.1 Literatürde yer alan farklı memristans değer güncelleme yöntemlerinin kıyaslanması

Kriter	**SÖY [5][10][11][12][13]**	**KT [15]**	**GBDY [14]**
Güç Tüketimi	Yüksek	Orta	Düşük
Kaynak Tüketimi	Yüksek	Düşük	Düşük
Güncelleme Hızı	Yüksek	Yüksek	Düşük
Güncelleme Doğruluğu	Yüksek	Yüksek	Düşük
Maliyet	Düşük	Yüksek	Düşük
Tasarım Esnekliği	Düşük	Düşük	Yüksek

Tablodaki veriler incelenerek gerilim bölücü devre yapısının bu çalışmada kullanılmasına karar verilmiştir. Memristif sinaptik devrelerde önemli bir kriter olan güç tüketiminin düşük olması bu seçimde etkili olmuştur. Benzer şekilde

düşük kaynak tüketimi de büyük ölçekli prototipler için önemli bir kriter olarak görülmektedir. Güncelleme hızı ve güncelleme doğruluğunda diğer yöntemlerden daha düşük performans sergilemesine rağmen yüksek tasarım esnekliği ve düşük maliyet bu seçimin yapılmasındaki diğer önemli kriterler olmuştur.

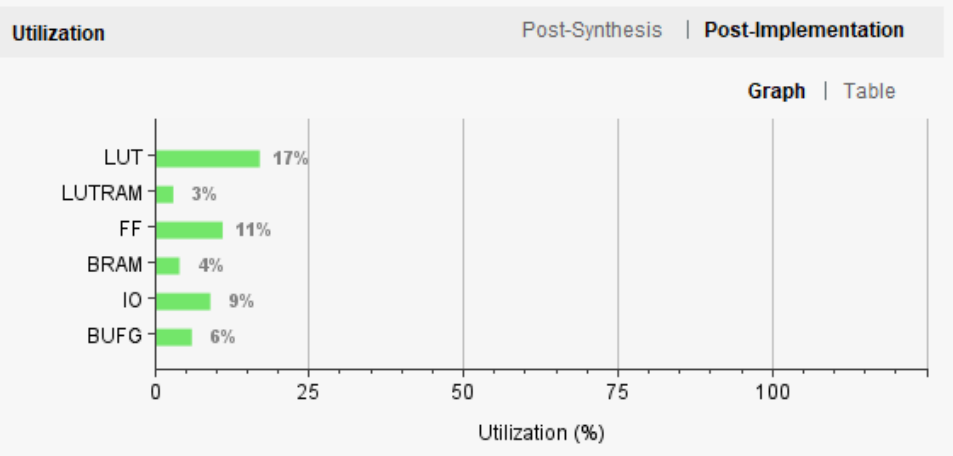
Bu kısımda ise önerilen yöntem olan gerilim bölücü devre yapısı ile literatürde yer alan bir diğer yöntem olan sürekli ölçüm yöntemi donanım tasarımı açısından kıyaslanacaktır. Bu kıyaslamalar gerçekleştirilirken FPGA geliştirme kartı üzerindeki teorik tasarımları gerçekleştirilerek kaynak ve güç tüketimi açısından karşılaştırılacaktır. Önerilen yönteme ait tasarımlar ve sonuçlar önceki bölümlerde detaylıca açıklanmıştır. Sürekli ölçüm yöntemi için bu yapılan tasarıma ek olarak FPGA geliştirme kartı üzerinde bir hafıza bloğu oluşturularak kıyaslanacak veriler bu hafıza bloklarına yazılmıştır. Sürekli ölçüm yöntemine ait FPGA tasarımının blok diyagramı Şekil 4.1'de verilmiştir.



Şekil 4.1 Sürekli ölçüm yöntemi için oluşturulmuş blok diyagram

Önerilen yönteme ait tasarımdan farklı olarak hafıza blokları eklenen bu blok tasarımda Arm tabanlı ZYNQ işlemcisiyle kontrol edilebilmesi için hafıza kontrol bloğu da eklenmiştir. BRAM kullanılarak oluşturulan bu hafıza yapısı konfigüre edilirken gerekli memristans karakteristik verilerini saklayacak şekilde tasarlanmıştır. Ayrıca bu memristans karakteristiği hafıza elemanına kaydedilirken karşılaştırmanın eşit koşullarda gerçekleştirilebilmesi için kabul edilebilir maksimum hata miktarları eşit olacak şekilde ayarlanmıştır. Karşılaştırmalar için önceki kısımlarda eşitliği ve karakteristiği verilen M_4 elemanı kullanılmıştır. Kabul edilebilir hata miktarı maksimum $\%5$ olacak şekilde ayarlandığında memristöre ait zamana bağlı memristans değişimi değerlerinde büyük bir azalma görülmüştür. Minimum meristans değeri $4k\Omega$ ve maksimum memristans değeri $18k\Omega$ arasında 100Ω aralıklarla değerler alınmış ve toplamda

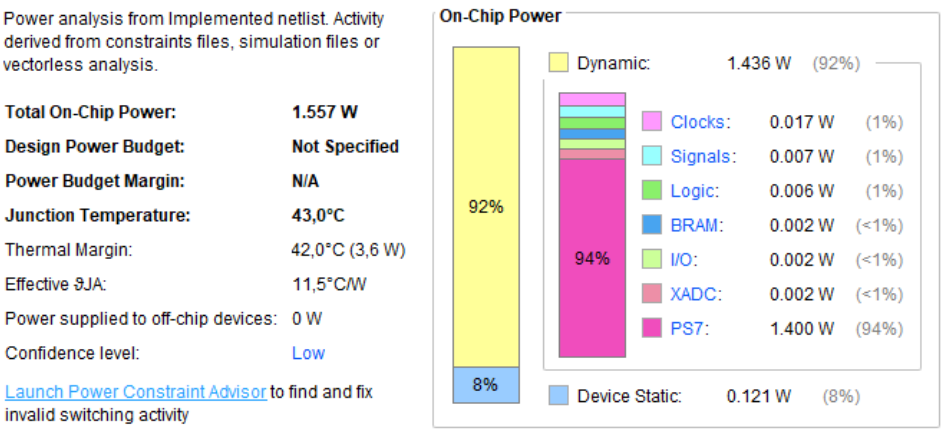
140 darbe sinyali süresi ve 140 memristans değeri elde edilmiştir. Bu elde edilen verilerin taşmadan sığabileceği kadar hafıza alanı oluşturularak tasarım tamamlanmıştır. Buradaki amaç ise, kaynak tüketimi kıyasında eşit koşullar oluşturmaktır. Bu tasarım sonucunda kaynak tüketimi incelenmiş ve elde edilen kaynak tüketim tablosu Şekil 4.2'de verilmiştir.



Şekil 4.2 Sürekli ölçüm yöntemi için kaynak tüketim tablosu

Kaynak tüketiminden gördüğümüz üzere sürekli ölçüm yönteminde oluşturulan hafıza elemanı bakma tablosu (LUT) ve blok rastgele erişilebilir bellek (BRAM) kaynaklarının tüketiminde büyük bir artışa sebep olmuştur. Önerilen yöntem olan gerilim bölücü devre yapısında LUT kaynakları geliştirme kartındaki mevcut kaynakların %13 oranında kullanılmasını gerektirirken, sürekli ölçüm yöntemi kullanılan LUT kaynaklarında %30 artışa sebep olarak geliştirme kartındaki mevcut kaynakların %17 oranında kullanılmasını gerektirmektedir. Aynı şekilde BRAM kaynaklarında da bir artış görülmektedir. Önerilen yöntemde %1 oranında kullanılan BRAM elemanı sürekli ölçüm yönteminde %4 oranında kullanılmıştır. Bu durum özellikle çok sayıda memristif sinaptik hücre bulundurabilecek büyük ölçekli prototip çalışmalarında bir kısıtlamaya sebep olacağı düşünülmektedir. Elde edilen sonuçlar kaynak tüketimi açısından önerilen yöntemin, sürekli ölçüm yöntemine göre daha başarılı olduğunu göstermektedir. Şekil 4.3'te sürekli ölçüm yöntemi için hazırlanan tasarıma ait güç tüketim tablosu verilmiştir.

Güç sonuçları analiz edildiğinde önerilen yöntem olan gerilim bölücü devre yapısında FPGA geliştirme kartında yapılan tasarımın güç tüketimi 1.545W olarak görülmüştü. Sürekli ölçüm yöntemi için yapılan tasarım sonucu güç tüketimi ise 1.557W olarak görülmüştür. Ancak güç tüketimi kıyaslaması yapılırken işletim sisteminin güç tüketimi de hesaba katıldığı için aradaki fark düşük görülmektedir. Kaynak tüketimindeki artışa bağlı olarak gerçekleşen güç tüketimini elde edebilmek için işletim sisteminin güç tüketimini çıkardığımızda önerilen

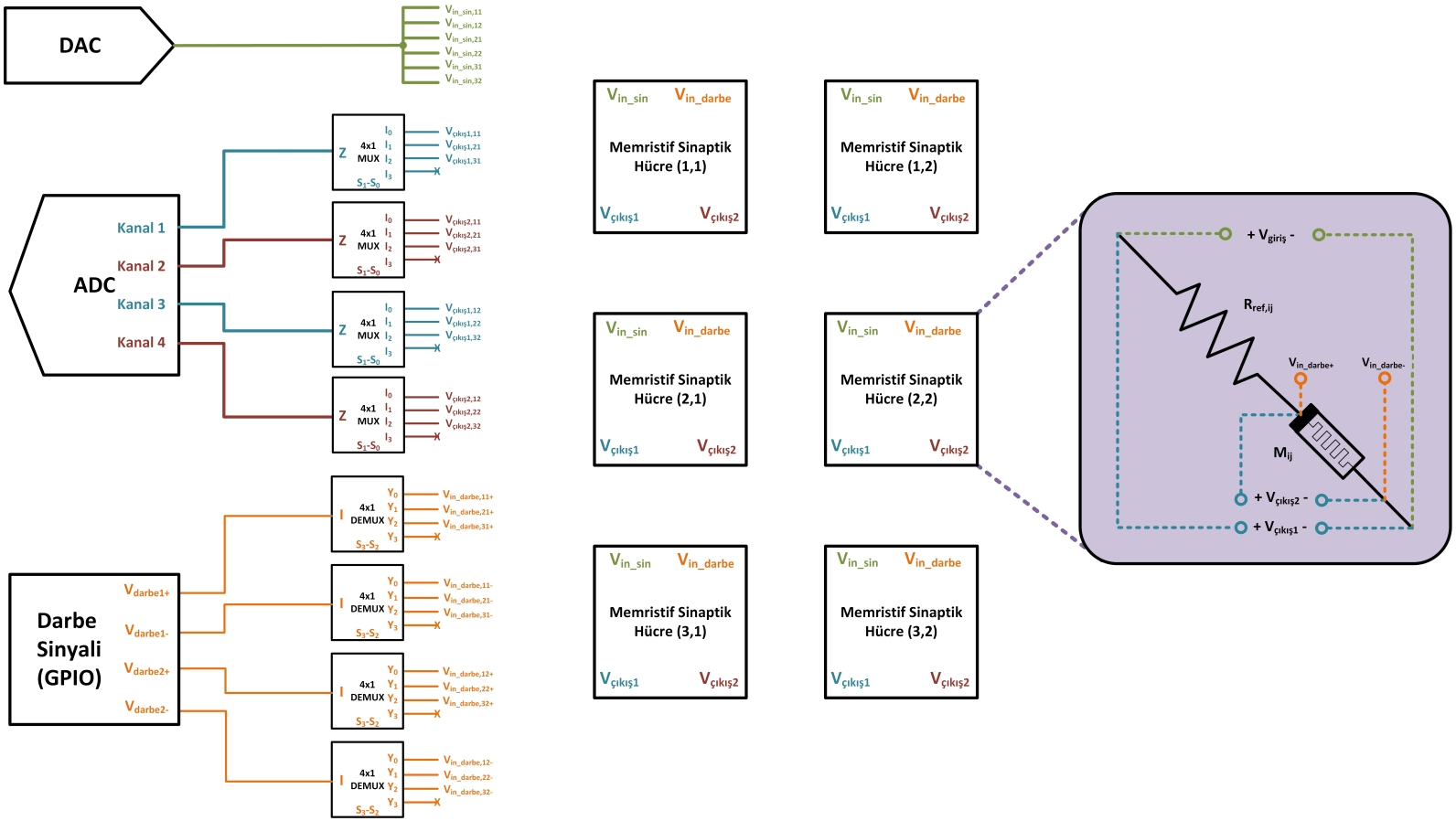


Şekil 4.3 Sürekli ölçüm yöntemi için güç tüketim tablosu

yöntemin lojik kaynaklarının güç tüketiminin 0.145W olduğu, sürekli ölçüm yönteminin güç tüketiminin ise 0.157W olduğu görülmektedir. Bu durum da lojik kaynaklardan kaynaklı güç tüketiminde yaklaşık olarak %8 oranında bir artış olduğunu göstermektedir. Sonuçlar ve tablolar incelendiğinde önerilen yöntem olan gerilim bölücü devre yapısı yönteminin güç tüketimi açısından daha avantajlı olduğunu ispatlar niteliktedir.

4.2 Büyük Ölçekli Prototip Tasarım Önerisi

Bu kısımda, bu çalışmada önerilen ve prototip niteliğindeki tasarımın büyük ölçekli prototipleri gerçekleştirilirken oluşabilecek tasarım kısıtları incelenecek ve büyük ölçekli prototip için teorik bir tasarım önerilecektir. Bu çalışmada önerilen sistem 6 sinaps hücresi için gerekli olan tasarımı içermektedir. Son yıllardaki çalışmalarla beraber ağ yapıları ve dolayısıyla sinaptik devrelerin boyutları büyüyen sistemler düşünüldüğünde büyük ölçekli bir sinaptik devrede memristör değer güncellemesi için farklı çözümlere ihtiyaç duyulmaktadır. Önerilen yöntemde memristans değer okuması için kullanılan devre yapısı sebebiyle memristans değerini elde etmek için iki farklı analog gerilim sinyalinin ölçülmesi gerekmektedir. Analog sinyallerin kullanılabilmesi için analogdan dijitale dönüştürücü entegresine ihtiyaç duyulmaktadır. Bu entegre devre sayısının kullanılacak sistemlerde sınırlı sayıda olması aynı anda değer güncellemesi yapılabilecek memristör sayısını kısıtlamaktadır. Bu kısıtlamalar nedeniyle sinaptik bir ağda yer alan sinaps hücrelerini sembolize eden memristör elemanlarının değerleri belirli bir düzende ve sırayla güncellenmelidir. Şekil 4.4'te $2x3$ boyutlarında toplam 6 sinaptik hücreden oluşan bir sinaptik devre ve bu devrenin değer güncelleme sürecinde kullanılacak devre şeması verilmiştir.



Şekil 4.4 Altı adet memristif sinaptik devre hücresine sahip sinaptik devre için değer güncelleme şeması

Değer güncelleme şeması 4 kanallı bir analogdan dijitale dönüştürücü entegre devresi, bu 4 kanalın anahtarlanarak kullanılması için gerekli olan 4x1 MUX elemanlarını içermektedir. Aynı zamanda çift kanallı olarak kullanılan dijitalden analoga dönüştürücü entegre devresinden 1 adet kullanılmıştır. Aynı anda 2 memristans değeri güncelleme yeteneğine sahip olan sistemde güncellenecek her bir memristör elemanı için aynı sinüzoidal sinyal kullanılmaktadır. Benzer şekilde sistem tarafından üretilecek darbe sinyallerinin alınacağı çıkışlar da kısıtlı olduğu için aynı anda 2 memristörün değerini kontrol edecek 4 darbe sinyali oluşturulmuştur. Bu sinyallerin de analog gerilim okuma süreciyle benzer bir sırada gerçekleştirilmesi gerekmektedir. Böylece 3 satırdan oluşan sistemde yer alan her satırdaki memristör elemanlarına sırasıyla darbe sinyali uygulanacaktır.

Değer güncellemesi yapılacak kontrolcü sistemde aynı anda 4 adet analogdan dijitale dönüştürücü entegre devresi kullanılacak şekilde tasarlanmıştır. FPGA geliştirme kartındaki giriş çıkış bağlantı noktaları kısıtlı olduğu için harici MUX (Multiplexer, Çoklayıcı) ve DEMUX (Demultiplexer, Çoğullama Çözücü) elemanları kullanılarak sırasıyla değer güncellemesi yapılması planlanmıştır. Tasarımda yer alan memristif sinaptik hücrelerin iç yapısı da verilerek bu hücrelere giren ve çıkan sinyaller gösterilmiştir. Ardından elde edilen sinyaller FPGA geliştirme kartı aracılığıyla anahtarlanabilen harici $4x1$ MUX elemanlarına verilerek anahtarlanacak şekilde tasarlanmıştır. Buradaki amaç, kısıtlı sayıdaki analogdan dijitale dönüştürücü entegresine sırasıyla aktararak tüm memristans değerlerini mümkün olduğu kadar hızlı güncellemektir. Her bir hücrede değeri okunması gereken iki gerilim değeri olduğu için her bir hücredeki sinyallerin anahtarlanması amacıyla iki adet MUX elemanı kullanılmıştır. Benzer şekilde değer güncelleme için gerekli darbe sinyalleri üretilerek $1x4$ DEMUX elemanları aracılığıyla değer güncelleme sürecinin de sırayla gerçekleştirilmesi sağlanmıştır. Böyle kısıtlı kaynaklara sahip kontrolcü kartı üzerinde sıralı işlemler yapılarak büyük ölçekli tasarımların da gerçekleştirilebilmesine olanak sağlanmış olur. Şekil 4.5'te değer güncelleme sürecinin akışını açıklayan psödo kod yer almaktadır.

Psödo kodu verilen algoritmada öncelikle tüm memristif sinaptik hücrelere sinüzoidal sinyal uygulanır. Uygulanan bu sinüzoidal sinyal memristör elemanının yüksek frekanslarda doğrusal davranış sergilemesi nedeniyle işlem yapılmayacak olan memristör elemanlarının değerlerinde bir değişime sebebiyet vermeyecektir. Ardından MUX elemanlarının seçim uçlarına (S1-S0) lojik değerler verilerek hangi memristif sinaptik hücrelerden ölçüm alınacağı belirlenir. MUX elemanının eçim uçları '00' olduğu durumda (1,1) ve (1,2) hücrelerinin çıkışları analogdan dijitale dönüştürücü entegresinin giriş kanallarına aktarılmış olur. Ardından DEMUX elemanlarının seçim uçlarına (S3-S2) değerler verilerek hangi memristör

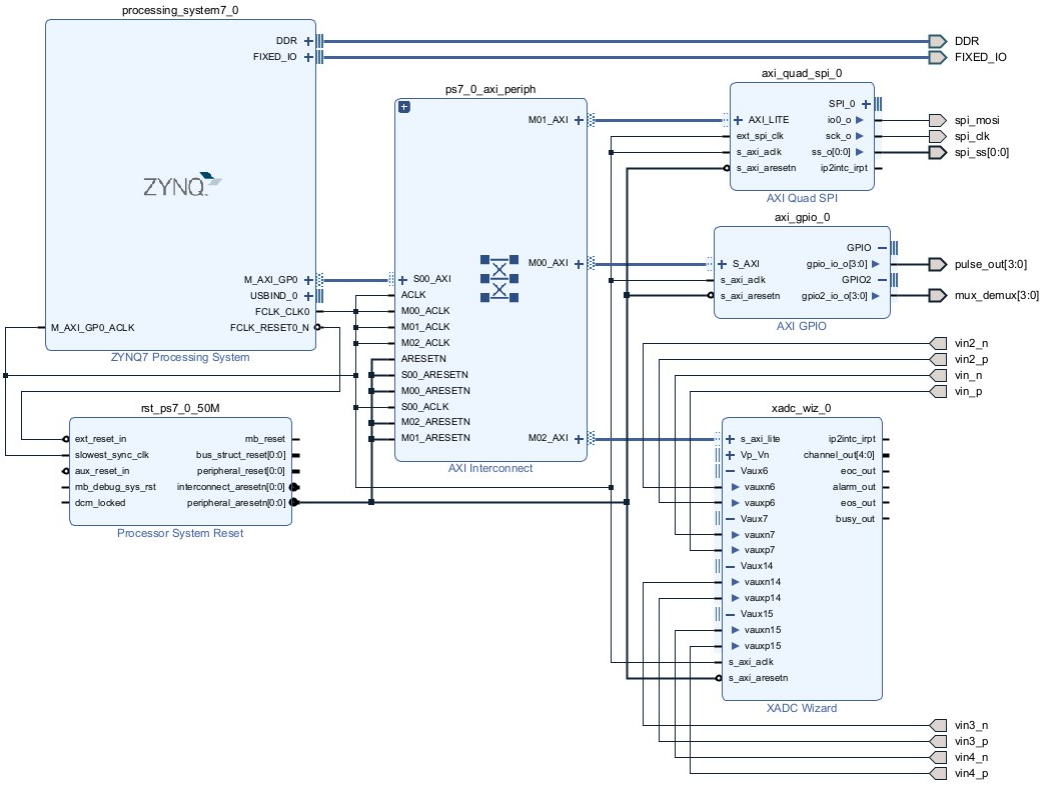
Algoritma 2 Dokuz Memristif Sinaptik Hücreli Tasarım Değer Güncelleme Algoritması

```
1: procedure SIRALI DEĞER GÜNCELLEME
2:     DAC aracılığıyla tüm hücrelere sinüzoidal sinyal uygula
3:     S1-S0 '00' yapılarak (1,1) ve (1,2) hücrelerinin çıkışlarını ADC üzerinden ölç
4:     S3-S2 '00' yapılarak (1,1) ve (1,2) hücrelerinin değerini güncelle
5:     Güncelleme tamamlanana kadar bekle
6:     S1-S0 '01' yapılarak (2,1) ve (2,2) hücrelerinin çıkışlarını ADC üzerinden ölç
7:     S3-S2 '01' yapılarak (2,1) ve (2,2) hücrelerinin değerini güncelle
8:     Güncelleme tamamlanana kadar bekle
9:     S1-S0 '10' yapılarak (3,1) ve (3,2) hücrelerinin çıkışlarını ADC üzerinden ölç
10:    S3-S2 '10' yapılarak (3,1) ve (3,2) hücrelerinin değerini güncelle
11:    Güncelleme tamamlanana kadar bekle
12: end procedure
```

Şekil 4.5 Altı adet memristif sinaptik devre hücresine sahip sinaptik devre için değer güncelleme algoritması

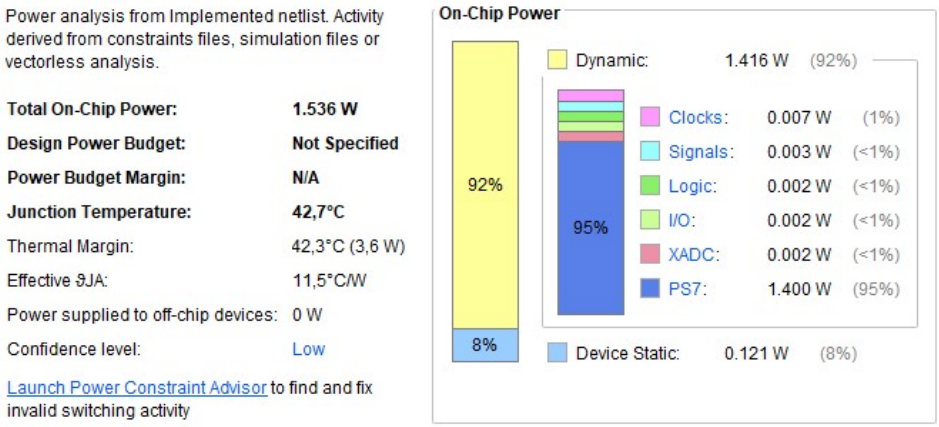
elemanlarının değerinin güncelleneceği seçilir. DEMUX elemanının seçim uçları '00' olduğu durumda (1,1) ve (1,2) sinaptik hücrelerindeki memristör elemanlarının değerleri güncellenir ve bu süreç tamamlandıktan sonra MUX elemanının seçim uçları bir sonraki satıra geçecek şekilde ayarlanır. MUX ve DEMUX elemanlarının seçim uçları '01' olduğu durumda ölçüm alınacak hücreler ve değeri güncellenecek hücreler (2,1) ve (2,2) olmaktadır. (3,1) ve (3,2) hücreleri için ise MUX ve DEMUX elemanlarının seçim uçları '10' olarak ayarlanır. Ardından bu tasarım için sıralı değer güncelleme süreci tamamlanır.

Yapılan bu teorik prototip çalışmanın güç ve kaynak tüketimini incelemek için FPGA geliştirme kartı üzerinde bir tasarım gerçekleştirilmiştir. Yapılan tasarımda analog sinyal üretimi için 3 adet sinyal üretilmiştir. Üretilen bu sinyaller dijitalden analoga dönüştürücü elemanı aracılığıyla sinüzoidal sinyale dönüştürülerek tüm hücrelere paralele olarak verilecektir. Aynı anda 2 memristörün değerini güncelleyebilmek adına çift kanallı 2 darbe sinyali üretilecek ve bu sinyallerin harici DEMUX elemanıyla kontrol edilebilmesi için 2 bitlik seçim ucu oluşturulmuştur. Benzer şekilde aynı anda 2 hücreden ikişer gerilim ölçüleceği için dahili analogdan dijitale dönüştürücü elemanı 4 kanallı olacak şekilde konfigüre edilmiştir. Sırasıyla yapılacak bu ölçümlerin kontrolü için kullanılacak harici MUX elemanlarının seçim uçları da yine 2 bit olacak şekilde üretilmiştir. FPGA geliştirme kartı üzerinde gerçekleştirilen blok tasarım Şekil 4.6'da verilmiştir. Blok tasarımda yer alan spi_mosi, spi_clk ve spi_ss sinyalleri sinüzoidal sinyal üretimi için kullanılacaktır. Farksal analog girişlerini vin_n, vin_p, vin2_n, vin2_p, vin3_n, vin3_p, vin4_n ve vin4_p sinyalleri göstermektedir. Üretilen çift kanallı 2 darbe sinyali pulse_out sinyaliyle gösterilmiş olup mux_demux sinyalleri ise seçim sinyallerini ifade etmektedir.



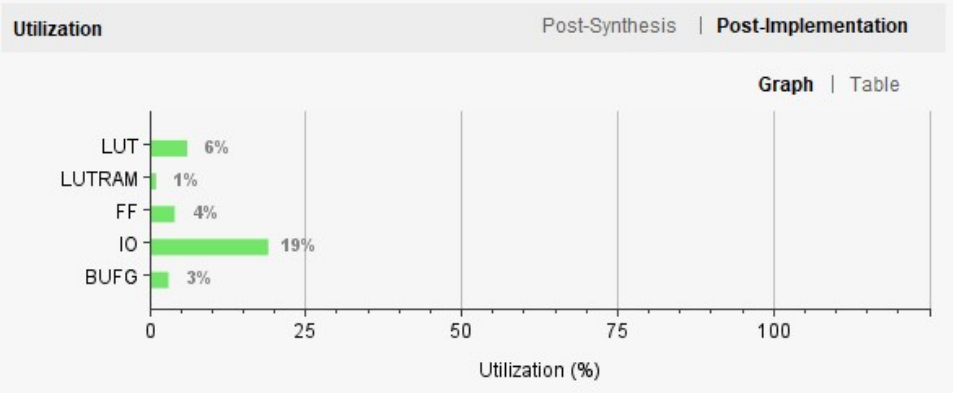
Şekil 4.6 FPGA geliştirme kartında 6 sinaptik hücreli prototip için oluşturulmuş blok tasarımı

Ardından bu blok tasarımda kullanılan sinyaller kullanılan FPGA geliştirme kartı olan Zybo Z-7010 üzerindeki ilgili bağlantı noktalarına atanarak güç tüketimi analizleri gerçekleştirilmiştir. Şekil 4.7'de 6 sinaptik hücreden oluşan prototip çalışmanın FPGA geliştirme kartı üzerindeki güç tüketimi tablosu paylaşılmıştır. Güç tüketimi tablosunda görüldüğü üzere tek hücreli prototipin güç tüketimine benzer değer elde edilmiştir. Her iki tasarımda da görüleceği üzere güç tüketimindeki en büyük pay işlemci yapısına aittir.



Şekil 4.7 FPGA geliştirme kartında 6 sinaptik hücreli tasarımın güç tüketimi tablosu

Son olarak yapılan bu tasarımın FPGA geliştirme kartı üzerindeki kaynak tüketimi incelenmiş olup elde edilen sonuçlar Şekil 4.8'de verilmiştir.



Şekil 4.8 FPGA geliştirme kartında 6 sinaptik hücreli tasarımın kaynak tüketimi tablosu

Elde edilen sonuçlar incelendiğinde büyük ölçekli prototip tasarımda en büyük fark giriş/çıkış kaynaklarının tüketiminde olduğu görülmüştür. Kontrol edilecek memristif sinaptik hücre sayısı arttıkça kontrol için gerekli sinyal sayısı da artacağı için büyük ölçekli prototiplerde en büyük kısıtın bu giriş çıkış sinyalleri olacağı düşünülmektedir.

Bu çalışmada kullanılan FPGA geliştirme kartının kısıtlarına uygun olarak gerçekleştirilen bu 6 memristif sinaptik hücreye sahip tasarım incelendiğinde güç ve kaynak tüketimi açısından tatmin edici sonuçlar elde edilmiştir. Memristif sinaptik hücre sayısının belirlenmesindeki en büyük kısıt kullanılan FPGA geliştirme kartında bulunan giriş/çıkış sinyalleri ve bunların alınacağı portların sayısı olmuştur. Dahili olarak 4 analogdan dijitale dönüştürücü entegresine sahip olan Zybo Z-7010 FPGA geliştirme kartından aynı anda maksimum 2 memristörün değeri okunabilmektedir. Daha büyük memristif sinaptik devreler için harici ADC entegresi ihtiyacı doğmaktadır.

5
SONUÇ

Önerilen yöntemde yer alan algoritmayı test etmek amacıyla 5 farklı memristör elemanı matematiksel modelleri kullanılmış ve elemanların memristans değişim karakteristikleri sabit 500 μA akım ile elde edilmiştir. Elde edilen bu karakteristikler simülasyon ve test sürecinde kullanılmamıştır. Memristör elemanlarına ait R_{ON} ve R_{OFF} değerleri gibi parametreler de yine bu süreçlerde kullanılmamış olup sistem tamamen yaptığı ölçümlere ve anlık duruma göre karar verecek şekilde kurgulanmıştır. Simülasyon testlerinde bu elemanlara ait matematiksel modeller rastgele belirlenen bir başlangıç durumunda algoritmaya verilmiştir. Performans kriteri, bu başlangıç durumundan her birinin belirlenen hedef memristans değerine ulaşması için geçen değer güncelleme döngüsü sayısı olarak belirlenmiştir. İlk zaman aralığı 0.5 ms olarak uygulanmıştır. OSC yönteminde kullanılan katsayı 6 olarak alınmıştır. Hedef memristans değerleri 5 $k\Omega$'dan 17 $k\Omega$'a kadar eşit aralıklarla, toplam 100 değer olacak şekilde belirlenmiştir. Her bir memristöre ait matematiksel model için seçilen 100 başlangıç memristans değerinden belirlenen 100 hedef memristans değerine ulaşmak için yapılması gerek okuma-yazma döngüsü sayısı hesaplanmıştır. Bu döngü sayılarının toplamı her bir matematiksel modelin başarı kriteri olarak seçilmiştir. Sadece parçalı lineer karakteristik (PLK) kullanılarak elde edilen okuma-yazma döngü sayıları önerilen optimizasyon yöntemleri ile azaltılmıştır. PLK yöntemine ek olarak OSC algoritması eklendiğinde her bir model için değişen ve maksimum %32.47 okuma-yazma adım sayısı kazancı sağladığı görülmüştür. Bu iki yönteme ek olarak EDH algoritması da eklendiğinde memristör modeline göre değişen ve maksimum %39.88 okuma-yazma adım sayısı kazancı sağladığı görülmüştür.

Ayrıca önerilen yöntemin donanımsal olarak gerçeklenmesi kısmında FPGA geliştirme kartı kullanılarak bir prototip oluşturulmuş, oluşturulan prototip bloklar test edilerek sonuçları paylaşılmıştır. Simülasyon ve deneysel sonuçlar incelendiğinde görüleceği gibi memristans değer güncelleme süresince gerekli olan sinyaller kısıtlara uygun şekilde üretilebilmiş, ölçülmesi gereken gerilim değerleri

doğrulukla ve hızlı bir şekilde ölçülebilmiştir. Güç tüketimini düşük tutmayı hedefleyen bu çalışmada tasarlanan blokların güç yönetimi yapılarak optimum güç tüketimi sağlanmıştır. Aynı zamanda kullanılan geliştirme kartındaki kaynak tüketimi incelendiğinde elde edilen sonuçlar gelecekteki büyük ölçekli prototip çalışmaları için umut vericidir.

KAYNAKÇA

[1] L. Chua, "Memristor-the missing circuit element," *IEEE Transactions on circuit theory*, vol. 18, no. 5, pp. 507–519, 1971.

[2] R. S. Williams, "How we found the missing memristor," *IEEE spectrum*, vol. 45, no. 12, pp. 28–35, 2008.

[3] P. Yao *et al.*, "Fully hardware-implemented memristor convolutional neural network," *Nature*, vol. 577, no. 7792, pp. 641–646, 2020.

[4] C. Li *et al.*, "Efficient and self-adaptive in-situ learning in multilayer memristor neural networks," *Nature communications*, vol. 9, no. 1, p. 2385, 2018.

[5] Y. Zhang, X. Wang, E. G. Friedman, "Memristor-based circuit design for multilayer neural networks," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 65, no. 2, pp. 677–686, 2017.

[6] X. Liu, Z. Zeng, D. C. Wunsch II, "Memristor-based lstm network with in situ training and its applications," *Neural Networks*, vol. 131, pp. 300–311, 2020.

[7] N. Wainstein, T. Tsabari, Y. Goldin, E. Yalon, S. Kvatinsky, "A dual-band cmos low-noise amplifier using memristor-based tunable inductors," in *2019 IEEE Computer Society Annual Symposium on VLSI (ISVLSI)*, IEEE, 2019, pp. 290–295.

[8] I. Marković, M. Potrebić Ivaniš, D. Tošić, "The dynamic tunability of memristor-based active filters," *Micromachines*, vol. 14, no. 11, p. 2064, 2023.

[9] S. Yu, "Neuro-inspired computing with emerging nonvolatile memorys," *Proceedings of the IEEE*, vol. 106, no. 2, pp. 260–285, 2018.

[10] M. Prezioso, F. Merrikh-Bayat, B. D. Hoskins, G. C. Adam, K. K. Likharev, D. B. Strukov, "Training and operation of an integrated neuromorphic network based on metal-oxide memristors," *Nature*, vol. 521, no. 7550, pp. 61–64, 2015.

[11] F. Alibart, E. Zamanidoost, D. B. Strukov, "Pattern classification by memristive crossbar circuits using ex situ and in situ training," *Nature communications*, vol. 4, no. 1, p. 2072, 2013.

[12] D. Soudry, D. Di Castro, A. Gal, A. Kolodny, S. Kvatinsky, "Memristor-based multilayer neural networks with online gradient descent training," *IEEE transactions on neural networks and learning systems*, vol. 26, no. 10, pp. 2408–2421, 2015.

[13] L. Yang, Z. Zeng, X. Shi, "A memristor-based neural network circuit with synchronous weight adjustment," *Neurocomputing*, vol. 363, pp. 114–124, 2019.

[14] S. Tu, J. Li, Y. Ren, Q. Jiang, S. Xiong, "A novel programming circuit for memristors," *Microelectronic Engineering*, vol. 280, May 2023.

[15] M. Tarkhan, M. Maymandi-Nejad, S. Haghzad Klidbary, S. Bagheri Shouraki, "A bridge technique for memristor state programming," *International Journal of Electronics*, vol. 107, no. 6, pp. 1015–1030, 2020.

[16] J. Gomez, I. Vourkas, A. Abusleme, G. C. Sirakoulis, A. Rubio, "Voltage divider for self-limited analog state programing of memristors," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 67, no. 4, pp. 620–624, 2019.

[17] I. Vourkas, J. Gómez, Á. Abusleme, N. Vasileiadis, G. C. Sirakoulis, A. Rubio, "Exploring the voltage divider approach for accurate memristor state tuning," in *2017 IEEE 8th Latin American Symposium on Circuits & Systems (LASCAS)*, IEEE, 2017, pp. 1–4.

[18] G. L. Zhang *et al.*, "An efficient programming framework for memristor-based neuromorphic computing," in *2021 Design, Automation & Test in Europe Conference & Exhibition (DATE)*, IEEE, 2021, pp. 1068–1073.

[19] Z. Biolek, D. Biolek, V. Biolkova, "Differential equations of ideal memristors," *Radioengineering*, vol. 24, no. 2, pp. 369–377, 2015.

[20] S. P. Adhikari, M. P. Sah, H. Kim, L. O. Chua, "Three fingerprints of memristor," *Handbook of Memristor Networks*, pp. 165–196, 2019.

[21] *12-bit dual output dac with spi*, MCP4922, Rev. A, Microchip Technology Inc., Apr. 2010.

[22] *7 series fpgas and zynq-7000 soc xadc dual 12-bit 1 msps analog-to-digital converter*, UG480, v1.11, Advanced Micro Devices, Inc., Jun. 2022.

TEZDEN ÜRETİLMİŞ YAYINLAR

Konferans Bildirisi

1. B. Şahintekin, Z.G. Çam Taşkıran, "Memristif Sinaptik Devreler için Uyarlanabilir Parçalı Lineer Karakteristik Yaklaşımı", Grad Colloquium'24 Artificial Intelligence, 42, 2024