

**FPGA TABANLI IQ-MATH SAYI STANDARDINDA YSA AKTİVASYON
FONKSİYONLARININ TASARIMI VE GERÇEKLENMESİ**

YÜKSEK LİSANS TEZİ

Mehmet Şamil AKÇAY

Danışman

Doç. Dr. İsmail KOYUNCU

ELEKTRİK-ELEKTRONİK MÜHENDİSLİĞİ ANABİLİM DALI

Temmuz 2021

Bu tez çalışması 19.FEN.BİL.14 numaralı proje ile Afyon Kocatepe Üniversitesi
Bilimsel Araştırma Projeleri Koordinasyon Birimi tarafından desteklenmiştir.

AFYON KOCATEPE ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

YÜKSEK LİSANS TEZİ

**FPGA TABANLI IQ-MATH SAYI STANDARDINDA YSA
AKTİVASYON FONKSİYONLARININ TASARIMI VE
GERÇEKLENMESİ**

Mehmet Şamil AKÇAY

Danışman

Doç. Dr. İsmail KOYUNCU

ELEKTRİK-ELEKTRONİK MÜHENDİSLİĞİ
ANABİLİM DALI

Temmuz 2021

BİLİMSEL ETİK BİLDİRİM SAYFASI

TEZ ONAY SAYFASI

Mehmet Şamil AKÇAY tarafından hazırlanan "FPGA TABANLI IQ-MATH SAYI STANDARDINDA YSA AKTİVASYON FONKSİYONLARININ TASARIMI VE GERÇEKLENMESİ" adlı tez çalışması lisansüstü eğitim ve öğretim yönetmeliğinin ilgili maddeleri uyarınca 09/07/2021 tarihinde aşağıdaki jüri tarafından **oy birliği** ile Afyon Kocatepe Üniversitesi Fen Bilimleri Enstitüsü **Elektrik Elektronik Mühendisliği Anabilim Dalı'nda YÜKSEK LİSANS TEZİ** olarak kabul edilmiştir.

Danışman : Doç. Dr. İsmail KOYUNCU

Başkan : Prof. Dr. Ahmet ALTUNCU

Kütahya Dumlupınar Üniversitesi, Mühendislik Fakültesi

Üye : Prof. Dr. Yüksel OĞUZ

Afyon Kocatepe Üniversitesi, Teknoloji Fakültesi

Üye : Doç. Dr. İsmail KOYUNCU

Afyon Kocatepe Üniversitesi, Teknoloji Fakültesi

İmza

Afyon Kocatepe Üniversitesi
Fen Bilimleri Enstitüsü Yönetim Kurulu'nun
..... /..... /..... tarih ve
..... sayılı kararıyla onaylanmıştır.

.....
Prof. Dr. İbrahim EROL

Enstitü Müdürü

BİLİMSEL ETİK BİLDİRİM SAYFASI
Afyon Kocatepe Üniversitesi

Fen Bilimleri Enstitüsü, tez yazım kurallarına uygun olarak hazırladığım bu tez çalışmada;

- Tez içindeki bütün bilgi ve belgeleri akademik kurallar çerçevesinde elde ettiğimi,
- Görsel, işitsel ve yazılı tüm bilgi ve sonuçları bilimsel ahlak kurallarına uygun olarak sunduğumu,
- Başkalarının eserlerinden yararlanması durumunda ilgili eserlere bilimsel normlara uygun olarak atıfta bulunduğumu,
- Atıfta bulunduğum eserlerin tümünü kaynak olarak gösterdiğimi,
- Kullanılan verilerde herhangi bir tahrifat yapmadığımı,
- Ve bu tezin herhangi bir bölümünü bu üniversite veya başka bir üniversitede başka bir tez çalışması olarak sunmadığımı

beyan ederim.

09/07/2021


İmza

Mehmet Şamil AKÇAY

ÖZET

Yüksek Lisans Tezi

FPGA TABANLI IQ-MATH SAYI STANDARDINDA YSA AKTİVASYON FONKSİYONLARININ TASARIMI VE GERÇEKLENMESİ

Mehmet Şamil AKÇAY

Afyon Kocatepe Üniversitesi

Fen Bilimleri Enstitüsü

Elektrik-Elektronik Mühendisliği Anabilim Dalı

Danışman: Doç. Dr. İsmail KOYUNCU

Yapay Sinir Ağları (YSA), insan beyninin öğrenme ve karar verme gibi yetenekleri referans alınarak geliştirilmiş yapay zekanın bir alt alanıdır. İnsan beyninde mevcut olan öğrenme, sınıflandırma, tahmin etme ve karar verme gibi birçok özellik YSA için birer çalışma alanıdır. YSA, öğrenme ve adaptasyon yeteneği, paralel veri işleme ve doğrusal olmayan sistemlerde kullanılma gibi önemli özelliklere sahiptir. Bu özellikleri sayesinde, YSA, günümüzde sinyal ve görüntü işleme, kontrol, hata tespiti, sınıflandırma, optimizasyon, tahmin, karar verme, güvenli haberleşme, uzay teknolojisi, kaotik osilatör tasarımı ve tıp gibi birçok farklı alanda yaygın olarak kullanılmaktadır. YSA çalışmalarında, ağ çıkışını, gerekli sonlu aralıklarda sınırlamak için AF (Activation Function-Aktivasyon Fonksiyonları) kullanılmaktadır. YSA yapısında kullanılan AF, YSA çıkışını doğrudan etkilediği için AF seçimi büyük önem taşımaktadır.

Bu tez çalışmasında birinci aşamada, doğrusal AF olan TriBas (**Triangular Basis**), HardLim (**Hard-Limit**), HardLims (**Symmetric Hard-Limit**), PosLin (**Positive Linear**), SatLin (**Saturating Linear**) ve SatLins (**Symmetric Saturating Linear**) fonksiyonları VHDL (**Very High Speed Integrated Circuit Hardware Description Language-Çok Yüksek Hızlı Tümlleşik Devre Donanım Tanımlama Dili**) dilinde kodlanarak 32 bit (16I-16Q) IQ-Math sayı standardında tasarlanmıştır. Tasarımları test etmek için Xilinx ISE Design Suite 14.7 programı kullanılmıştır. Test işleminden elde edilen simülasyon test sonuçları sunulmuştur. Ardından, tüm doğrusal AF tasarımları Xilinx Kintex-7 ailesinin

XC7K70T-3FBG676 FPGA (Field Programmable Gate Array-Alanda Programlanabilir Kapı Dizileri) çipi için sentezlenmiştir. Sentez işleminden sonra Place & Route işlemi yapılarak FPGA çip istatistikleri sunulmuştur. Çip istatistikleri incelendiğinde, HardLim ve HardLims fonksiyonlarının en az çip kaynağı tükettiği görülmüştür. Ayrıca, doğrusal AF tasarımları içerisinde 1379.172 MHz ile en yüksek çalışma frekansına SatLin AF'nin sahip olduğu tespit edilmiştir.

Bu tez çalışmasının ikinci aşamasında, YSA uygulamalarında oldukça yaygın bir şekilde kullanılan ve doğrusal olmayan AF olan RadBas (**R**adial **B**asis), LogSig (**L**ogarithmic **S**igmoid) ve TanSig (**T**angent **S**igmoid) AF, 32 bit (16I-16Q) IQ-Math sayı standardında VHDL dili ile tasarlanmıştır. Doğrusal olmayan üç AF içerisinde bulunan e^x üstel fonksiyonu tasarımında CORDIC-LUT (**C**Oordinate **R**otation **D**igital **C**omputer-**L**ook-Up **T**able) yaklaşımı kullanılmıştır. Tasarımlar Xilinx ISE Design Suite 14.7 programında test edilerek simülasyon test sonuçları sunulmuştur.

Daha sonra doğrusal olmayan üç AF tasarımları Xilinx Kintex-7 XC7K70T-3FBG676 FPGA çipi için sentezlenmiştir. Ardından, Place & Route işlemi yapılarak FPGA çip istatistikleri sunulmuştur. Ayrıca, bu üç FPGA tabanlı AF tasarımlarının MSE (Mean Square Error-Ortalama Karesele Hata) ve RMSE (Root Mean Square Error-Ortalama Karesele Hatanın Karekökü) hata analizleri, nümerik tabanlı AF kullanılarak gerçekleştirilmiş ve elde edilen sonuçlar sunulmuştur. Yapılan bu doğrusal ve doğrusal olmayan AF tasarımları ile bir aktivasyon fonksiyonları kütüphanesi oluşturulmuştur.

Tez çalışmasının üçüncü aşamasında, FPGA tabanlı örnek bir üç girişli ve üç çıkışlı ileri beslemeli YSA tasarımı gerçekleştirilmiştir. YSA tasarımının gizli katmanında 8 nöron bulunmaktadır. Tasarımın gizli katmanında LogSig AF ve çıkış katmanında PureLin (**P**ure **L**inear) AF kullanılmıştır. Örnek YSA tasarımı 32 bit (16I-16Q) IQ-Math sayı standardında VHDL dilinde kodlanmıştır. Tasarım Xilinx ISE Design Suite 14.7 programında VHDL dilinde bir testbench oluşturularak test edilmiştir. Test işleminden elde edilen tasarıma ait simülasyon sonuçları için MSE ve RMSE hata analizleri yapılmış ve sonuçlar değerlendirilmiştir. Test işleminden sonra yapılan tasarım Kintex-7

XC7K70T-3FBG676 FPGA ipi iin sentezlenmiř ve Place & Route iřleminin ardından FPGA ip istatistikleri sunulmuřtur.

2021, xvii + 86 sayfa

Anahtar Kelimeler: YSA, FPGA, Aktivasyon Fonksiyonu, IQ-Math sayı standardı, VHDL, IEEE 754-1985 standardı.

ABSTRACT

M.Sc. Thesis

DESIGN AND IMPLEMENTATION OF FPGA-BASED ANN ACTIVATION FUNCTIONS IN IQ-MATH NUMBER STANDARD

Mehmet Şamil AKÇAY

Afyon Kocatepe University

Graduate School of Natural and Applied Sciences

Department of Electrical-Electronics Engineering

Supervisor: Assoc. Prof. İsmail KOYUNCU

Artificial Neural Networks (ANN) is a subfield of artificial intelligence developed by referencing the working structure of the human brain. Many features that exist in the human brain such as learning, classification, estimation, and decision making are the areas of study for ANN. ANN has important features such as learning and adaptation capability, parallel data processing, and the ability to be used in nonlinear systems. Due to these features, ANN has been widely used nowadays in many different fields such as image and sound processing, control, error detection, classification, optimization, prediction, decision making, secure communication, space technology, chaotic oscillator design, and medicine. Activation functions (AF) have been used in ANN studies to limit network output at the required finite intervals. Therefore, AF directly affects the result. For this reason, the choice of AF is very important.

In the first stage of this thesis, linear AF TriBas (Triangular Basis), HardLim (Hard-Limit), HardLims (Symmetric Hard-Limit), PosLin (Positive Linear), SatLin (Saturating Linear), and SatLins (Symmetric Saturating Linear) functions have been coded in VHDL (Very High Speed Integrated Circuit Hardware Description Language) and designed in 32 bit (16I-16Q) IQ-Math number standard. Xilinx ISE Design Suite 14.7 program was used to test the designs. Xilinx's ISE Design Suite 14.7 program was used to test the designs. The simulation test results obtained from the testing process are presented. Then, all linear AF designs were synthesized for the XC7K70T-3FBG676 FPGA (Field Programmable Gate Arrays) chip of the Xilinx Kintex-7 family. After the synthesis

process, the FPGA chip statistics are presented by performing the Place & Route process. When the chip statistics are examined, it is seen that the HardLim and HardLims functions consume the least chip resources. Besides, SatLin AF has been found to have the highest operating frequency as 1379.172 MHz among linear AF designs.

In the second phase of this thesis, RadBas (**R**adial **B**asis), LogSig (**L**ogarithmic **S**igmoid), and TanSig (**T**angent **S**igmoid) AF, which are non-linear AF and widely used in ANN applications, are coded with VHDL in 32-bit (16I-16Q) IQ Math number standard. The CORDIC-LUT (**C**Oordinate **R**otation **D**igital **C**omputer-**L**ook **U**p **T**able) approach was used in the design of the e^x exponential function in three non-linear AFs. The designs were tested in Xilinx ISE Design Suite 14.7 program and simulation test results were presented.

Later, three non-linear AF designs were synthesized for the Xilinx Kintex-7 XC7K70T-3FBG676 FPGA chip. Then, the FPGA chip statistics were presented by performing the Place & Route process. Besides, MSE (Mean Square Error) and RMSE (Root Mean Square Error) error analyses of these three FPGA-based AF designs were implemented using numerical-based AF and the acquired results are presented. With these linear and non-linear AF designs, a library of activation functions has been created.

In the third stage of the thesis, a sample of FPGA-based three-input and three-output feedforward ANN design has been implemented. There are 8 neurons in the hidden layer of the ANN design. LogSig AF is used in the hidden layer of the design and PureLin (Pure Linear) AF is used in the output layer. The sample ANN design has been coded in VHDL in 32 bit (16I-16Q) IQ-Math number standard. The design was tested by creating a testbench in VHDL in Xilinx ISE Design Suite 14.7 program. For the simulation results of the design obtained from the test process, MSE and RMSE error analyses were performed and the results were evaluated. After the test process, the design was synthesized for the Kintex-7 XC7K70T-3FBG676 FPGA chip, and FPGA chip statistics were presented after the Place & Route process.

2021, xvii + 86 pages

Keywords: ANN, FPGA, Activation Functions, IQ-Math fixed point, VHDL, IEEE 754-1985 standard.

TEŞEKKÜR

Tez çalışması boyunca maddi ve manevi her türlü desteğini esirgemeyen aynı zamanda tez amacının belirlenmesinde ve tez aşamalarında bilgi birikimini ve tecrübelerini benimle paylaşan çok saygı değer danışman hocam Sayın Doç. Dr. İsmail KOYUNCU'ya en içten samimi duygularıyla sonsuz teşekkürlerimi sunarım.

Tez çalışmalarım boyunca ihtiyaç duyduğum noktalarda yardımlarını esirgemeyen Sayın Dr. Öğretim Üyesi Murat ALÇIN ve Dr. Öğretim Üyesi Murat TUNA hocalarıma teşekkür etmeyi borç bilirim. Ayrıca yüksek lisans eğitimim boyunca her konuda öneri ve eleştirileriyle yardımlarını gördüğüm hocalarıma ve arkadaşlarıma en içten duygularıyla teşekkürlerimi sunarım.

Tez çalışmasının yapılmasında 19.FEN.BİL.14 numaralı proje ile maddi destek sağlayan Afyon Kocatepe Üniversitesi Bilimsel Araştırma Projeleri Koordinasyon birimine teşekkür ederim.

Tüm hayatım boyunca maddi ve manevi desteklerinden dolayı aileme teşekkür ederim.

Mehmet Şamil AKÇAY
Afyonkarahisar 2021

İÇİNDEKİLER DİZİNİ

| | Sayfa |
|---|-------|
| ÖZET | i |
| ABSTRACT | iv |
| TEŞEKKÜR | vii |
| İÇİNDEKİLER DİZİNİ..... | viii |
| SİMGELER ve KISALTMALAR DİZİNİ | x |
| ŞEKİLLER DİZİNİ | xii |
| ÇİZELGELER DİZİNİ..... | xiv |
| RESİMLER DİZİNİ | xv |
| 1. GİRİŞ..... | 1 |
| 2. LİTERATÜR BİLGİLERİ | 3 |
| 3. MATERYAL ve METOT | 9 |
| 3.1 Yapay Zeka | 9 |
| 3.1.1 Yapay Zekanın Tarihsel Gelişimi | 9 |
| 3.1.2 Yapay Zeka Teknikleri..... | 12 |
| 3.1.3 Uzman Sistemler | 12 |
| 3.1.4 Bulanık Mantık | 13 |
| 3.1.5 Doğal Dil İşleme | 17 |
| 3.1.6 Genetik Algoritmalar | 21 |
| 3.2 Yapay Sinir Ağları | 27 |
| 3.2.1 YSA Nöron Modeli..... | 27 |
| 3.2.2 Aktivasyon Fonksiyonu | 29 |
| 3.2.2.1 Doğrusal Aktivasyon Fonksiyonları..... | 29 |
| 3.2.2.2 Doğrusal Olmayan Aktivasyon Fonksiyonları | 32 |
| 3.2.3 YSA Yapıları..... | 34 |
| 3.2.3.1 Tek Katmanlı İleri Beslemeli YSA | 35 |
| 3.2.3.2 Çok Katmanlı İleri Beslemeli YSA..... | 35 |
| 3.2.3.3 Geri Beslemeli YSA..... | 36 |
| 3.3 Alanda Programlanabilir Kapı Dizileri..... | 37 |
| 3.4 Programlanabilir Mantık Cihazları | 38 |
| 3.4.1 Basit Programlanabilir Mantık Cihazları | 38 |
| 3.4.2 Karmaşık Programlanabilir Mantık Cihazları..... | 40 |
| 3.4.3 FPGA Çipleri | 41 |

| | |
|---|----|
| 3.4.3.1 Giriş ve Çıkış Blokları..... | 43 |
| 3.4.3.2 Programlanabilir Mantıksal Bloklar..... | 43 |
| 3.4.3.3 Ara Bağlantılar | 44 |
| 3.5 Kayan Noktalı Sayı Standardı..... | 44 |
| 3.6 Sabit Noktalı Sayı Standardı..... | 46 |
| 3.7 VHDL | 49 |
| 3.7.1 VHDL Dilinde Veri Nesneleri | 49 |
| 3.7.2 VHDL Dilinde Temel Tasarım Yapıları | 50 |
| 4. BULGULAR | 52 |
| 4.1 FPGA Tabanlı Tasarımlar..... | 52 |
| 4.1.1 Doğrusal AF'nin FPGA Tabanlı Tasarımları..... | 52 |
| 4.1.1.1 Doğrusal AF'nin Simülasyon Sonuçları..... | 57 |
| 4.1.2 Doğrusal Olmayan AF'nin FPGA Tabanlı Tasarımları | 59 |
| 4.1.2.1 Doğrusal Olmayan AF'nin Simülasyon Sonuçları..... | 65 |
| 4.1.2.2 Doğrusal Olmayan AF'nin Hata Analizleri..... | 66 |
| 4.1.3 FPGA Tabanlı YSA Tasarımı | 67 |
| 4.1.3.1 İleri Beslemeli YSA Tasarımının Simülasyon Sonuçları..... | 69 |
| 4.1.3.2 İleri Beslemeli YSA Tasarımının Hata Analizleri | 70 |
| 5. TARTIŞMA ve SONUÇ | 73 |
| 6. KAYNAKLAR..... | 75 |
| ÖZGEÇMİŞ..... | 86 |

SİMGELER ve KISALTMALAR DİZİNİ

Simgeler

| | |
|------------------|------------------------------------|
| a | Eğim parametresi |
| b_k | Eşik değeri |
| exp | Üs bitleri değeri |
| f | Kayan noktalı sayı standart sabiti |
| j | Sinaps |
| k | Nöron hücresi |
| sign | İşaret biti değeri |
| u_k | Toplam fonksiyonu çıkış değeri |
| y_k | Nöron çıkış sinyali |
| x_j | Sinap giriş sinyali |
| w_{kj} | Sinaptik ağırlık sinyali |
| $\varphi(\cdot)$ | Aktivasyon Fonksiyonu |

Kısaltmalar

| | |
|----------|--|
| AF | Aktivasyon Fonksiyonları (Activation Functions) |
| ANFIS | Adaptive Network Based Fuzzy Inference Systems (Uyarlamalı Ağ Tabanlı Bulanık Mantık Çıkarım Sistemi) |
| ANN | Artificial Neural Networks (Yapay Sinir Ağları) |
| ASIC | Application Specific Integrated Circuit (Uygulamaya Özel Tümüleşik Devre) |
| BbNNs | Block-based Neural Networks (Blok tabanlı Sinir Ağları) |
| BM | Bulanık Mantık (Fuzzy Logic) |
| BPN | Back Propagation Network (Geri Yayılım Ağı) |
| CCNs | Cellular Neural Networks (Hücresel Sinir Ağları) |
| CORDIC | Coordinate Rotation Digital Computer |
| CPLD | Complex Programmable Logic Device (Karmaşık Programlanabilir Mantık Cihazları) |
| CPU | Central Process Unit (Merkezi İşlem Birimi) |
| DDİ | Doğal Dil İşleme (Natural Language Processing) |
| DRAM | Dinamic RAM |
| DSP | Digital Signal Processor (Sayısal Sinyal İşlemcisi) |
| EEPLD | Electrically Erasable PLD (Elektriksel Silinebilen PLD) |
| EEPROM | Electrically Erasable PROM (Elektriksel Silinebilen PROM) |
| EPLD | Erasable PLD (Silinebilen PLD) |
| EPROM | Erasable PROM (Silinebilen PROM) |
| FPGA | Field Programmable Gate Array (Alanda Programlanabilir Kapı Dizisi) |
| GA | Genetik Algoritmalar (Genetic Algorithm) |
| GAL | General Array Logic (Genel Dizi Mantığı) |
| GPU | Graphics Processing Unit (Grafik İşlemci Birimi) |
| GWO | Gray Wolf Optimizer |
| IEEE | The Institute of Electrical and Electronical Engineers |
| IEEE-754 | IEEE Floating Point Number (Kayan Noktalı Sayı Formatı) |
| IP CORE | Intellectual Properties Core |

Kısaltmalar (Devam)

| | |
|---------|---|
| ISE | Integrated Software Environment (Tümleşik Yazılım Ortamı) |
| IQ-Math | Sabit-Noktalı Sayı Formatı |
| LogSig | Logarithmic Sigmoid |
| LUT | Look-Up Table (Değer Tablosu) |
| MLFF | Multi-Layer Feed Forward (Çok Katmanlı İleri Beslemeli) |
| MLPNN | Multilayer Perceptron Neural Network (Çok Katmanlı Algılayıcı Sinir Ağı) |
| MSE | Mean Square Error (Ortalama Karesel Hata) |
| PAL | Programmable Array Logic (Programlanabilir Dizi Mantıkları) |
| PEEL | Programmable Electrically-Erasable Logic (Programlanabilen Elektriksel Silinebilen Lojik) |
| PLD | Programmable Logic Device (Programlanabilen Mantık Cihazları) |
| PMB | Programlanabilen Mantıksal Bloklar |
| PROM | Programmable ROM (Programlanabilen Salt Okunabilen Bellek) |
| PSO | Parçacık Sürü Optimizasyonu |
| PUKS | Pehlivan-Uyaroglu Kaotik Sistemi |
| RadBas | Radial Basis |
| RAM | Random Access Memory (Rasgele Erişimli Bellek) |
| RMSE | Root Mean Square Error (Ortalama Karesel Hatanın Karekökü) |
| ROM | Read Only Memory (Salt Okunabilen Bellek) |
| SCPWL | Simplicial Canonical Piecewise Linear (Basit Kanonik Parçalı Doğrusal) |
| SPLD | Simple Programmable Logic Device (Basit Programlanabilir Mantık Cihazları) |
| SRAM | Static RAM |
| TanSig | Tanjant Sigmoid |
| TRNG | True Random Number Generator (Gerçek Rasgele Sayı Üretici) |
| US | Uzman Sistemler (Expert Systems) |
| VHDL | Very High Speed Integrated Circuit Hardware Description Language (Çok Yüksek Hızlı Tümleşik Devre Donanım Tanımlama Dili) |
| YZ | Yapay Zeka (Artificial Intelligence) |

ŞEKİLLER DİZİNİ

| | Sayfa |
|--|-------|
| Şekil 3.1 Uzman sistemlerin genel yapısı. | 13 |
| Şekil 3.2 Bulanık mantık sistemlerinin genel yapısı. | 15 |
| Şekil 3.3 Genetik algoritmalarda işlem adımları. | 23 |
| Şekil 3.4 Genetik algoritmalarda mutasyon işlemi. | 24 |
| Şekil 3.5 Biyolojik sinir hücresi. | 27 |
| Şekil 3.6 YSA nöron modeli. | 28 |
| Şekil 3.7 PosLin AF eğrisi. | 29 |
| Şekil 3.8 HardLim AF eğrisi. | 30 |
| Şekil 3.9 HardLims AF eğrisi. | 30 |
| Şekil 3.10 SatLin AF eğrisi. | 31 |
| Şekil 3.11 SatLins AF eğrisi. | 31 |
| Şekil 3.12 TriBas AF eğrisi. | 32 |
| Şekil 3.13 LogSig AF eğrisi. | 33 |
| Şekil 3.14 TanSig AF eğrisi. | 34 |
| Şekil 3.15 RadBas AF eğrisi. | 34 |
| Şekil 3.16 Tek-katmanlı ileri beslemeli YSA yapısı. | 35 |
| Şekil 3.17 Çok katmanlı ileri beslemeli YSA yapısı. | 36 |
| Şekil 3.18 Geri beslemeli YSA yapısı. | 37 |
| Şekil 3.19 SPLD yapısı. | 39 |
| Şekil 3.20 CPLD yapısı. | 40 |
| Şekil 3.21 FPGA çiplerinin genel mimarisi. | 42 |
| Şekil 3.22 PMB bloklarının iç yapısı. | 44 |
| Şekil 3.23 32 bit tek hassasiyetli IEEE 754-1985 sayı standardının gösterimi. | 45 |
| Şekil 3.24 50 sayısına ait 32 bitlik IEEE 754-1985 sayı formatı gösterimi. | 46 |
| Şekil 3.25 IQ-Math sabit noktalı sayı formatı gösterimi. | 47 |
| Şekil 3.26 50.3749 sayısının 16I-16Q sabit noktalı sayı formatında gösterimi. | 48 |
| Şekil 4.1 PosLin fonksiyonuna ait en üst seviye FPGA blok şeması. | 53 |
| Şekil 4.2 PosLin fonksiyonuna ait IQ-Math sayı tabanlı blok şeması. | 53 |
| Şekil 4.3 HardLim fonksiyonuna ait IQ-Math sayı tabanlı blok şeması. | 54 |
| Şekil 4.4 HardLims fonksiyonuna ait IQ-Math sayı tabanlı blok şeması. | 54 |
| Şekil 4.5 SatLin fonksiyonuna ait IQ-Math sayı tabanlı blok şeması. | 55 |
| Şekil 4.6 SatLins fonksiyonuna ait IQ-Math sayı tabanlı blok şeması. | 56 |

| | |
|---|----|
| Şekil 4.7 TriBas fonksiyonuna ait IQ-Math sayı tabanlı blok şeması..... | 57 |
| Şekil 4.8 TriBas fonksiyonuna ait Xilinx ISE simülasyon sonuçları..... | 57 |
| Şekil 4.9 HardLim fonksiyonuna ait Xilinx ISE simülasyon sonuçları..... | 58 |
| Şekil 4.10 HardLims fonksiyonuna ait Xilinx ISE simülasyon sonuçları..... | 58 |
| Şekil 4.11 PosLin fonksiyonuna ait Xilinx ISE simülasyon sonuçları..... | 58 |
| Şekil 4.12 SatLin fonksiyonuna ait Xilinx ISE simülasyon sonuçları..... | 58 |
| Şekil 4.13 SatLins fonksiyonuna ait Xilinx ISE simülasyon sonuçları..... | 58 |
| Şekil 4.14 Üstel fonksiyona ait IQ-Math sayı tabanlı blok şeması..... | 60 |
| Şekil 4.15 Üstel fonksiyona ait en üst seviye FPGA blok şeması..... | 61 |
| Şekil 4.16 Üstel fonksiyona ait ikinci seviye FPGA blok şeması..... | 61 |
| Şekil 4.17 RadBas fonksiyonuna ait IQ-Math tabanlı blok şeması..... | 62 |
| Şekil 4.18 RadBas fonksiyonuna ait ikinci seviye FPGA blok şeması..... | 62 |
| Şekil 4.19 LogSig fonksiyonuna ait IQ-Math tabanlı blok şeması..... | 63 |
| Şekil 4.20 LogSig fonksiyonuna ait ikinci seviye FPGA blok şeması..... | 64 |
| Şekil 4.21 TanSig fonksiyonuna ait IQ-Math tabanlı blok şeması..... | 64 |
| Şekil 4.22 TanSig fonksiyonuna ait ikinci seviye FPGA blok şeması..... | 65 |
| Şekil 4.23 RadBas fonksiyonuna ait Xilinx ISE simülasyon sonuçları..... | 65 |
| Şekil 4.24 LogSig fonksiyonuna ait Xilinx ISE simülasyon sonuçları..... | 65 |
| Şekil 4.25 TanSig fonksiyonuna ait Xilinx ISE simülasyon sonuçları..... | 66 |
| Şekil 4.26 İleri beslemeli YSA yapısına ait en üst seviye FPGA blok şeması..... | 68 |
| Şekil 4.27 İleri beslemeli YSA yapısına ait ikinci seviye FPGA blok şeması..... | 69 |
| Şekil 4.28 FFNN ileri beslemeli ağ yapısının Xilinx ISE simülasyon sonuçları..... | 70 |

ÇİZELGELER DİZİNİ

| | Sayfa |
|--|--------------|
| Çizelge 4.1 Doğrusal aktivasyon fonksiyonların FPGA çip istatistikleri. | 59 |
| Çizelge 4.2 Doğrusal olmayan aktivasyon fonksiyonların hata analizleri. | 67 |
| Çizelge 4.3 Doğrusal olmayan aktivasyon fonksiyonların FPGA çip istatistikleri..... | 67 |
| Çizelge 4.4 İleri beslemeli ağ yapısının hata analizleri..... | 70 |
| Çizelge 4.5 İleri beslemeli ağ yapısının FPGA çip istatistikleri. | 70 |
| Çizelge 4.6 Farklı sayı standardı ve FPGA platformu kullanılarak gerçekleştirilen dört girişli YSA nöron tasarımları ve özellikleri. | 71 |

RESİMLER DİZİNİ

| | Sayfa |
|---|--------------|
| Resim 3.1 Xilinx Kintex-7 FPGA kartı..... | 42 |
| Resim 3.2 Sayıcı devresine ait VHDL kodları..... | 51 |

1. GİRİŞ

Günümüzde ve öncesinde insan hayatını daha yaşanılabilir hale getirebilmek için birçok önemli çalışmalar yapılmıştır. Canlılar ve vücutlarında gözlemlenen mükemmel işleyiş incelenmiş ve bu işleyiş çalışmalarının birçoğuna ilham kaynağı olmuştur. Mükemmel şekilde çalışan yapısı ile insan beyninden de birçok alandaki çalışmalarda faydalanılmıştır. Yapay zeka alanlarından birisi olan Yapay Sinir Ağları (YSA) insan beyninin çalışma yapısı esas alınarak geliştirilmiştir. İnsan beyninde mevcut olan öğrenme, tahmin etme, karar verme ve sınıflandırma gibi birçok işlev YSA kullanılarak çeşitli alanlarda yapılabilmektedir. Bu nedenle bu ağlar mühendislik uygulama alanlarında yaygın bir şekilde kullanılmaktadır. Bu uygulama alanlarına, rasgele sayı üreticileri (Alçın vd. 2021), otomotiv sanayi (Ahmed vd. 2015), uzay sanayi (Wang vd. 2020), robotik (Du 2011), haberleşme (Kaur ve Sivia 2020) ve optik (Mello ve Ventura 2020) örnek verilebilir.

YSA üzerine yapılan çalışmalarda 1900'lü yıllarda önemli gelişmeler yaşanmıştır. 1943 yılında ilk YSA modeli Warren McCulloch ve Walter Pitts tarafından gerçekleştirilmiştir. İlgili modelde insan beyninde mevcut olan hesaplama yeteneği referans alınmış ve elektrik devreleri kullanılarak basit yapıda bir sinir ağı modellenmiştir. Bu sayede ilk YSA hücre yapısı oluşturularak bu yapı YSA için bir temel oluşturmuştur. 1949'da Hebb, öğrenilebilir ve adapte olabilir sinir ağı modelinin temelini oluşturan Hebb kuralını geliştirmiştir. Bu kuralda sinir ağındaki mevcut bağlantı sayısının değiştirilmesi ile ağın öğrenilebileceği savunulmuştur. 1950'li yıllarda YSA'nın hesaplama gücünü artırıcı nitelikte çalışmalar yapılmış ve bu çalışmalarda Hebb kuralı referans alınmıştır. 1957'de Frank Rosentblatt, Algılayıcıyı (Perceptron) tanıtması ile YSA üzerindeki gelişmeler hız kazanmıştır. 1959 yılında Bernard Widrow ve Marcian Hoff, ADALINE (ADaptive LInear NEuron-Uyarlanabilir Doğrusal Nöron) ile MADALINE (Many ADALINE) ağ modellerini tanıtmıştır. MADALINE, evrensel sorunlarda uygulanmış ilk sinir ağı niteliğindedir. Bu ağ modeli ilk kez telefon hatlarındaki yankıları yok etmek amacı ile kullanılmış ve halen de kullanımı devam etmektedir. İnsanlar bu başarılı çalışmaların ardından YSA üzerinde yüksek beklentilere girmekten kaçınamamışlardır. Bu yüksek beklentiler karşılanamayınca YSA üzerinde yapılan çalışmalarda bir duraklama

dönemine girilmiştir. Bu dönem 1981 yılında son bulmuştur. Bu yıldan itibaren günümüze kadar YSA kullanılarak yapılan çalışmalar devam etmiş ve günümüzde birçok çalışmada YSA sıklıkla kullanılmaktadır. YSA-tabanlı çalışmalar genel olarak yazılım ve donanım tabanlı olmak üzere iki gruba ayrılmaktadır (Koyuncu vd. 2019).

Yazılım tabanlı YSA çalışmalarında öğrenme, tahmin etme, karar verme ve sınıflandırma gibi birçok uygulama başarılı bir şekilde yapılabilmektedir. Ancak yoğun matematiksel hesaplamaların, hız ve paralelliğin büyük önem taşıdığı gerçek zamanlı YSA uygulamalarında yazılım tabanlı YSA çalışmaları performans açısından yetersiz kalmaktadır. Bu nedenle donanım tabanlı YSA çalışmalarında ASIC (Application Specific Integrated Circuit-Uygulamaya Özel Tümlleşik Devre), GPU (Graphic Processor Unit-Grafik İşlemci Ünitesi) ve FPGA (Field Programmable Gate Array-Alanda Programlanabilir Kapı Dizileri) gibi farklı platformlar kullanılmaktadır. Bu platformlardan birisi olan FPGA çipleri yüksek çalışma frekansları, hızlı ilk prototipleme, paralel çalışma, tekrar tekrar programlanabilme, düşük güç tüketimi ve esneklik gibi özellikleri sayesinde diğer platformlara kıyasla önemli avantajlara sahiptir. Bu avantajları ile FPGA platformları gerçek zamanlı YSA uygulamalarının ihtiyaç duyduğu hız ve performans gereksinimlerini karşılamaktadır (Akçay vd. 2020).

2. LİTERATÜR BİLGİLERİ

Yaşadığımız evren incelendiğinde birçok harika sistemin içinde bulunduğu gözlemlenmektedir. Bu harika sistemlerin incelemesi ile birçok önemli bilimsel gelişmeler kaydedilmiştir. İnsan beyni ve mükemmel şekilde çalışan yapısı uzun yıllar bilimsel çalışmalara konu olmuştur. İnsan beyninin muhteşem çalışma yapısının referans alınması ve matematiksel olarak modellenmesi ile yapay zekanın bir alt çalışma alanı olan YSA meydana gelmiştir. YSA, insan beyninde mevcut olan öğrenme, tahmin etme, karar verme ve sınıflandırma gibi birçok işlevi taklit edebilmektedir. Ayrıca bu ağların bazı üstünlükleri aşağıdaki gibi sıralanabilir:

- Doğrusal olmayan sistemlerde kullanılabilme
- Eksik bilgiye rağmen çözüm üretme
- Paralel ve dağıtılmış bilgi işleme
- Adaptasyon yeteneği
- Öğrenme yeteneği ile tanıtılmamış örneklerde çözüm üretebilme
- Yazılımsal ve donanımsal olarak gerçekleştirilebilme

Bu üstünlükleri ile YSA günümüzde sinyal ve görüntü işleme (Şahin ve Koyuncu 2012), kontrol (Fei ve Ding 2012), hata tespiti (Patnaik vd. 2007), tahmin, karar verme, sınıflandırma (Şahin ve Temür 2016), optimizasyon (Lin ve Tsai 2008), güvenli haberleşme (Tuna vd. 2019), uzay teknolojisi (Gankidi ve Thangavelautham 2017), kaotik osilatör tasarımı (Koyuncu ve Şeker 2019), tıp (Avcı vd. 2014) ve rasgele sayı üreteçleri (Koyuncu vd. 2019) gibi farklı alanlarda yaygınlıkla kullanılmaktadır. YSA üzerinde yapılan literatür çalışmalarında bu ağlar yazılım ve donanım tabanlı uygulanmaktadır. Donanım tabanlı YSA çalışmalarında ASIC, GPU, DSP, FPGA gibi farklı platformlar kullanılmaktadır.

ASIC platformunun kullanıldığı donanım tabanlı YSA çalışmaları diğer platformlara kıyasla daha yüksek frekans değerlerinde çalışabilmesine rağmen ilk tasarımda ve test sürecinde ciddi maliyet gerektirmesi, esnek yapıda olmaması gibi dezavantajlara sahiptir. Diğer bir platform olan DSP sistemleri karmaşık matematiksel ifadeleri gerçekleştirmeye

uygun donanım yapısına sahiptir. Fakat DSP'ler işlemleri ardışık şekilde yapmasından dolayı sistemin cevap süresi artmakta ve çalışma frekansında düşüş görülmektedir (Koyuncu 2014). GPU sistemleri ise hız performansı bakımından avantajlı olsa da fazla güç tüketimi ve prototipleme işlemlerinde kullanılamaması gibi dezavantajlara sahiptir (İnt.Kyn.3). FPGA çiplerinin yüksek çalışma frekansı, hızlı ilk prototipleme, paralel çalışma, tekrar tekrar programlanabilme ve düşük güç tüketimi gibi özellikleri ile diğer platformlara kıyasen önemli üstünlükleri bulunmaktadır (Taşdemir vd. 2020).

YSA'nın donanım tabanlı çalışmalarında FPGA ve birçok farklı platform kullanılmaktadır. Literatürdeki donanım tabanlı YSA çalışmalarına örnek olarak;

Literatür çalışmalarında Himavathi ve arkadaşları, FPGA üzerinde YSA uygulaması için yeni yöntem geliştirmişlerdir. Yöntemde yalnızca ağıın en büyük katmanı FPGA'lerde uygulanmış ve diğer katmanların tasarımında bu katman tekrar uygulanmıştır. Sonuç olarak bu yöntem sayesinde YSA uygulamasının FPGA üzerindeki donanım maliyeti verimli şekilde indirgenmiştir (Himavathi ve Muthuramalingam 2007).

Şahin vd., RadBas, LogSig ve TanSig transfer fonksiyonlarını kullanarak YSA nöronlarının FPGA çiplerinde 32 bit uzunluğunda IEEE 754-1985 kayan noktalı sayı tabanlı tasarım ve uygulamasını yapmışlardır. Sonuçta, 10 ya da daha fazla nöron bulunan YSA'nın FPGA platformunda kolaylıkla uygulanabileceğini ifade etmişlerdir (Şahin ve Koyuncu 2012).

Çavuşlu ve arkadaşları, PSO (Parçacık Sürü Optimizasyonu) algoritmasını kullanarak FPGA çiplerinde donanımsal olarak YSA eğitimi gerçekleştirmişlerdir. Tasarımı Altera firmasının EP2C35F672C6 FPGA çipine uygulamışlardır. Ağ yapısında giriş ve gizli katmanlarında 2 nöron, çıkış katmanında 1 nöron kullanılmıştır. Çalışmada en fazla 539 parçacığın kullanılabilmesi gösterilmiştir (Çavuşlu vd. 2010).

Adetiba vd., MLP-ANN (Multi-Layer Perceptron-Artificial Neural Networks-Çok Katmanlı Perceptron YSA) nöronunu FPGA platformunda donanımsal olarak uygulamışlardır. Nöronun test edilmesinde TanSig ve LogSig transfer fonksiyonları

kullanılmıştır. Sonuçta, ilgili nöronun donanım uygulamasında TanSig fonksiyonundan daha verimli sonuçlar alındığı gözlemlenmiştir (Adetiba vd. 2014).

Alçın ve arkadaşları, PUKS (Pehlivan-Uyaroglu Kaotik Sistemi) kaotik sisteminin modellenmesi için yeni bir YSA'nın FPGA üzerinde donanımsal uygulamasını yapmışlardır. Sonuçta, elde edilen sonuçlar incelenmiş ve FPGA platformlarında YSA tabanlı kaotik sistemlerin başarılı olarak modellenebildiği ifade edilmiştir (Alçın vd. 2016).

Hajduk çalışmasında yüksek hassasiyetli YSA transfer fonksiyonu uygulamasını 32 bit uzunluğunda kayan noktalı sayı standardında FPGA üzerinde gerçekleştirmiştir. Yapılan çalışmada hiperbolik tanjant ve sigmoid AF kullanılmıştır. İki fonksiyonda ortak bulunan üstel fonksiyonu için McLaurin serisi ile Padé polinomları yaklaşımları kullanılmıştır. Sonuçta, iki transfer fonksiyonu için ilgili yaklaşımlardan elde edilen maksimum mutlak hata değerleri sunulmuştur. Her iki yaklaşımdaki sonuçlarda sigmoid fonksiyonundan daha az hata alınmıştır (Hajduk 2017).

Nambiar vd., çalışmalarında geliştirilebilir BbNNs (Block-based Neural Networks-Blok tabanlı Sinir Ağları) ağlarının FPGA çiplerindeki donanım uygulamasını sunmuşlardır. Çalışmada düşük maliyete sahip sigmoid benzeri yeni bir transfer fonksiyonu kullanılmıştır. BbNNs ağları çalışmada SoC (System on Chip-Çip üzerinde Sistem) olarak tasarlanmıştır. Tasarım işlevsel olarak doğrulanıp çeşitli çalışma durumları için testi yapılmıştır. Çalışma gerçek zamanlı sınıflandırma uygulamaları için yeterli sistem performansına sahiptir. Sonuç olarak çalışmanın gömülü yazılıma kıyasla yaklaşık 410 kat daha hızlı çalıştığı sunulmuştur (Nambiar vd. 2014).

Çelikel çalışmasında, yüksek frekansa sahip sinyallerin kullanılması ile hesaplanmakta olan saft durum hatalarını azaltmaya yönelik önerdiği ölçüm yöntemini YSA tabanlı tasarlamıştır. Çalışmada kullanılan YSA, 3 girişe, gizli katmanında 10 nörona ve 2 çıkışa sahip yapıda olarak FPGA platformunda uygulanmıştır. Sonuç olarak, önerilen yöntem geleneksel yöntemlere kıyasla çok düşük hataya sahip olduğu gözlemlenmiştir (Çelikel 2019).

Al-Rikabi ve arkadaşları, sundukları çalışmada RadBas, LogSig, TanSig doğrusal olmayan AF ile SCPWL (Simplicial Canonical PieceWise Linear-Basit Kanonik Parçalı Doğrusal) genel modelini kullanmışlardır. Modelde optimize işlemi GWO (Gray Wolf Optimizer) algoritması ile yapılmıştır. Yapılan çalışma VHDL (Very High Speed Integrated Circuit Hardware Description Language-Çok Yüksek Hızlı Tümlüşik Devre Donanım Tanımlama Dili) dilinde Xilinx Arty A7 FPGA çipi üzerinde sentezlenerek MSE hata değerleri sunulmuştur (Al-Rikabi vd. 2020).

Yılmaz ve arkadaşları, jeotermal enerjili hidrojen üretim sistemi için ekonomik analiz üzerine bir çalışma yapmışlardır. Çalışmada MLFF (Multi-Layer Feed Forward-Çok Katmanlı İleri Beslemeli) YSA modeli kullanılmıştır. Kullanılan ağ 4 girişe, her katmanında 5 nöron bulunan 2 gizli katmana ve 6 çıkışa sahiptir. İlgili ağın gizli katman seviyesinde TanSig, RadBas ve LogSig AF kullanılarak MSE hata değerleri verilmiştir. Çalışmadaki YSA kullanılarak FPGA yongalarında gerçek zamanlı uygulamaların yapılabileceğini öngörmüşlerdir (Yılmaz vd. 2019).

Vaidyanathan ve arkadaşları, dört boyutlu iki diskli dinamik sistemin YSA tabanlı modellenmesini yapmışlardır. Dinamik sistemin çatallanma analizi yapılarak gerekli olan devre tasarımı yapılmıştır. Daha sonra sistem FPGA platformunda gerçekleştirilmiştir. Bu tasarımla yeni bir TRNG (True Random Number Generator-Gerçek Rastgele Sayı Üretici) tasarlanmıştır. Tasarımın Xilinx Virtex-6 FPGA yongasında sentezlenmesi ile 272 MHz çalışma frekansı elde edilmiştir. Xilinx ISE Design Tools tasarım aracının 14.2 versiyonunda sistem test edilmiş ve simülasyon test sonuçları verilmiştir (Vaidyanathan vd. 2020).

Selvathi ve Nayagam, çalışmalarında WBCD (Wisconsin Breast Cancer Database-Wisconsin Meme Kanseri Veritabanı) için iki farklı ağ yapısı tasarlamışlardır. Tasarladıkları BPN (Back Propagation Network-Geri Yayılım Ağı) ve MLPNN (Multi Layer Perceptron Neural Network-Çok Katmanlı Perceptron Sinir Ağı) ağlarını veri tabanının eğitilmesi ve test edilmesinde kullanmışlardır. Yapılan çalışmada kanser hastalarında iyi ya da kötü huylu kanserin belirlenmesi amacıyla YSA kullanılmıştır.

Sonuçta, Xilinx Virtex-5 FPGA platformunda YSA sistemi sentezlenerek % 90.83 miktarda sınıflandırma başarısı alınmıştır (Selvathi ve Nayagam 2016).

El-Madany ve arkadaşları, çalışmalarında YSA kullanarak güç sistemi kontrolörünü uzay araçlarında kullanılması amacıyla tasarlamışlardır. Tasarladıkları YSA, yük akımını ve hata sinyalinin temsil eden 2 girişe, 3 nöronlu gizli katmana ve bataryadaki şarj akımının değişimini temsil eden bir çıkışa sahiptir. YSA sistemi, FPGA yongalarında sentezlenmek amacıyla IEEE-754 floating point sayı formatında VHDL dili kullanılarak kodlanmıştır. Tasarım Xilinx firmasının Virtex-2, Virtex-5 ve Spartan-3 olarak üç farklı FPGA yongasında sentezlenmiştir. Sonuçta, Virtex-5 FPGA çipinin en az çip kaynak kullanımına sahip olduğu gözlemlenmiştir (El-Madany vd. 2012).

Moorgas ve Govender, nesne hareketlerinin algılanması, çıkarılması ve filtrelenmesi için DSP ve YSA'nın birlikte kullanıldığı hibrit bir sistem geliştirmişlerdir. Hareket algılamada DSP sistemi işlemleri YSA'ya kıyasla daha hızlı yapmıştır. Fakat hibrit sistem ile daha iyi görüntü kalitesi oluşturan diziler üretilerek daha iyi performans sağlandığı gözlemlenmiştir (Moorgas ve Govender 2013).

Yılmaz yaptığı çalışmada, diferansiyel gelişim algoritması (DGA) kullanarak bir YSA yapısı tasarlamıştır. Tasarım VHDL dilinde 16 bit IEEE 754-1985 kayan noktalı sayı formatı kullanılarak Xilinx Virtex-7 VC707 FPGA geliştirme kitinde gerçekleştirmiştir. Çalışmada, DGA algoritması CSFNN (Conic Section Function Neural Network-Konik Kesit Fonksiyonlu Sinir Ağı) ve MLPNN ağlarının eğitilmesi ile test edilmiştir (Yılmaz 2014).

Jiang ve arkadaşları, çalışmalarında ektopik ventriküler atımların sınıflandırılması amacı ile geliştirilebilir BbNN ağlarını kullanarak FPGA tabanlı bir sınıflandırıcı tasarlamışlardır. Tasarımda öznelik çıkarımı için Hermit açılımı kullanılmıştır. Sonuçta, yapılan FPGA tabanlı sınıflandırıcıdan % 98.1 doğruluk oranı elde edilmiştir (Jiang ve Kong 2007).

Gupta ve arkadaşları, robotik ve gömülü uygulamalarda yaygın kullanılan SNN (Spiking Neural Networks-Darbeli Sinir Ağları) ağları için FPGA çiplerini kullanarak bir model tasarlamışlardır. Tasarlanan SNN modeli basit yapıda ve verimli hesaplama yapmaktadır. Bu model Xilinx Virtex 6 FPGA yongasında çalışmak üzere 800 nöronlu ve 12.544 sinapslı bir ağı analiz edebilmektedir (Gupta vd. 2020).

Elrharras ve arkadaşları, frekans alanında yapılan temel bileşen analizinde ve YSA tabanlı spektrum algımda kullanmak üzere yeni bir teknik geliştirmişlerdir. Önerilen teknikte, temel bileşen analizi ile özneliklerin çıkarılması işleminde MLP algılayıcısı kullanılarak sınıflandırılma yapılmıştır. Çalışmada kullanılan ağ yapısının giriş katmanı ve gizli katmanında 10 nöron çıkış katmanında 1 nöron bulunmaktadır. Gizli katmanda LogSig AF, PWL (Piece Wise Linear-Parçalı Doğrusal) yaklaşımı ile kullanılmıştır. Çalışma kayan noktalı sayı formatında VHDL dili ile kodlanmıştır. Ayrıca bu çalışma Altera Quartus II programında sentezlenerek Altera DE2-70 geliştirme kiti için Place-Route işlemleri yapılmıştır. Bu işlemlerden elde edilen çip istatistikleri sunulmuştur (Elrharras vd. 2015).

3. MATERYAL ve METOT

3.1 Yapay Zeka

Zekâ, insanda mevcut olan düşünme, akıl yürütme, objektif gerçekleri algılama, yargılama ve sonuç çıkarma yeteneklerinin bütünü olarak tanımlanmaktadır (İnt. Kyn. 1). Yapay Zeka (YZ) kavramı insana mahsus bu yeteneklerin gerekli analizlerin yapılması ile makinelerde kullanılabilmesidir (Çömlekçi 2020). Diğer bir tanım ile YZ, insan zekâsına ait olan akıl yürütme, anlam çıkartma, geçmiş deneyimlerden öğrenme gibi birçok yetenekleri insan zekâsının çalışma yapısını model alarak bir bilgisayarda veya bilgisayar kontrolündeki bir robotta gerçekleştirilmesi olarak da tanımlanabilir (Batal 2016). İnsan beyninin çalışma yapısını referans alan diğer çalışma alanlarından farklı olarak YZ çalışmaları insan beyninin sadece zekâ birimi ile ilgilenmektedir (Çömlekçi 2020). YZ çalışmaları incelendiğinde aşağıdaki maddelerin amaçlandığı öngörülebilmektedir:

- İnsan beyninin işlevlerini anlamak için bilgisayar modelleri kullanmak ve insana mahsus olan öğrenme, sonuç çıkarma gibi bazı teknikleri araştırmak.
- İnsana mahsus olan yetenekleri bilgisayarlarda işleyebilmek ve tasarlanan bu sistemlerin kullanılabilmesi mümkün olacak formata getirmek.
- Bir alana mahsus bilgi toplaması yapılarak bilgi sistemi oluşturmak, YZ iş yardımcılarının, genel bilgi sistemlerinin ve zeki robot ekiplerinin geliştirilmesini sağlamak.
- YZ alanında araştırma yapacak yeni araştırmacılar yetiştirmek (Çömlekçi 2020, Balaban ve Kartal 2018).

3.1.1 Yapay Zekanın Tarihsel Gelişimi

“Yapay Zeka” terimini literatürde ilk kez 1956 yılında John McCarthy kullanmıştır (İnt.Kyn.2). Fakat bu yıldan önce de YZ alanı için temel niteliğinde birçok bilimsel gelişme meydana gelmiştir. YZ teriminin literatürde kullanılmasının öncesinde ve

sonrasındaki tarihsel sürece ait bazı önemli gelişmeler kronolojik sırası ile aşağıdaki gibidir:

- 1921 yılında Karel Capek'in yazdığı "Rossum's Universal Robots" isimli oyununun yayınlanması ile birlikte "Robot" kelimesi ilk kez İngilizcede kullanılmıştır.
- 1936-1937 yıllarında Alan Turing'in evrensel Turing makinesi bilim dünyasına sunulmuştur. Böylece bu makine ile modern bilgisayar kavramının temelleri atılmıştır.
- 1943 yılında Warren McCulloch ve Walter Pitts "A Logical Calculus of the Ideas Immanent in Nervous Activity" çalışmasını yayınlarak sinir ağlarının temelini oluşturmuşlardır. Çalışma bir sinir hücresinin yapay modelini sunması bakımından en önemli çalışmadır.
- 1950 yılında Turing "Computing Machinery and Intelligence" çalışması yayınlanmıştır. Çalışmada "Turing Testi" olarak bilinen üç oyuncudan oluşan bir oyun bulunmaktadır. Bu oyunda oyunculardan birisi sorduğu sorulara karşılık aldığı cevaplara göre karşısındaki oyuncunun bilgisayar ya da insan olduğunu anlamaya çalışmaktadır. Bu çalışmada bilgisayarda düşünme yeteneği olan bir program olduğu farzedilerek Turing'e göre uygun şartlar oluşturulduğunda bilgisayarların bu testi geçebileceği gösterilmiştir.
- 1956 yılında John McCarthy Dartmouth Konferansında "yapay zeka" kavramını bir konu başlığı olarak kullanması ile bu kavramı literatüre kazandırmıştır. Alan Newell, J.C. Shaw ve Herbert Simon tarafından geliştirilen ilk YZ programı olan "Logic Theorist" verilmiş ve böylece YZ'nin yeni bir çalışma alanı olarak tüm dünyada kabul görme süreci başlamıştır.
- 1957 yılında Newell, Shaw ve Simon, GPS (The General Problem Solver-Genel Problem Çözücü) sistemini tanıtmışlardır.
- 1952-1962 yıllarında Arthur Samuel dünya dama oyunu şampiyonu ile mücadele edebilecek ilk oyun oynayabilen programı geliştirmiştir.
- 1958 yılında McCarthy, LISP (Locator/ID Separation Protocol-Konum Belirleyici/Kimlik Ayırma Protokolü) programlama dilini bilim dünyasına duyurmuştur.

- 1958 yılında Rosenblatt, görüntü tanıma çalışmalarında kullanmak için Perceptron yapısını tanıtmıştır. Bu yapı yapay sinir ağlarının ilkel bir modeli olması bakımından önemlidir.
- 1962 yılında ilk endüstriyel robot firması olan Unimation kurulmuştur.
- 1969 yılında hareketlilik, algılama ve problem çözme yetkilerini birlikte yapan ilk robot projesi olan Shakey robotu Stanford Araştırma Enstitüsü Yapay Zeka Merkezi'nde geliştirilmiştir.
- 1980'li yılların ortalarında sinir ağlarının geriye yayılım algoritması ile kullanılması sayesinde sinir ağları geniş kullanım alanı bulmaya başlamıştır.
- 1988 yılında hücresel sinir ağı modeli ve teorisini ilk kez Chua ve Yang bilim dünyasına önermişlerdir. Yayınladıkları “Cellular neural networks: theory” başlıklı makale, CNNs (Cellular Neural Networks-Hücresel Sinir Ağları) ağ modeli ve teorisinin temeli niteliğindedir (Chua ve Yang, 1988). Bununla birlikte bu sinir ağları, yapıları ve paralel sinyal işleme yetenekleri ile gerçek zamanlı sinyal işleme çalışmalarında yaygın şekilde tercih edilmeye başlanmıştır.
- 1997 yılında Deep Blue satranç programı dünya satranç şampiyonu Garry Kasparov'u yenmiştir.
- 2000'li yıllarda birçok sayıda ve farklı çeşitlerde ticari amaçlı akıllı oyuncaklar üretilmiştir.
- 2005 yılında Stanford'ın geliştirdiği otonom araç olan Stanley DARPA Grand Challenge yarışını kazanmıştır.
- 2005 yılından günümüze kadar YZ alanındaki çalışmalar devam ederek gündelik hayatımızda yaygınlaşarak kullandığımız akıllı telefonlar, temizlik robotları, akıllı saatler gibi birçok cihazda YZ'ye rastlanmaktadır. Öte yandan endüstride askeri, tıp, biyomedikal, lojistik, üretim endüstrisi, otomotiv, haberleşme, akıllı bina teknolojisi, güvenlik ve kontrol sistemleri gibi birçok alanda YZ'nin kullanımı yaygınlaşmaktadır (Çömlekçi 2020).

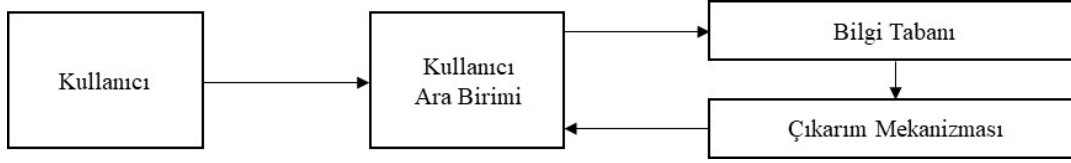
3.1.2 Yapay Zeka Teknikleri

1950-1960'lı yıllarda YZ'den faydalanılarak dama oynayan bir program geliştirilmiş 1969 yılında ise ilk kez YZ kullanılarak Shakey robotu üretilmiştir. 2005 senesinde ise Stanley otonom aracı üretilmiştir. YZ üzerine çalışmalar zaman içerisinde ilerledikçe farklı yapay zekâ teknikleri geliştirilmiştir. Günümüzde YZ çalışmaları bu tekniklerin kullanılması ile savunma sanayi, güvenlik ve kontrol sistemleri, otomotiv sanayi, tıp, görüntü işleme gibi birçok alana yayılmış durumdadır. YZ tekniklerinin birçok farklı alanlarda uygulanabilmesinde yüksek başarı sağlamaları ve değişebilen koşullara karşılık yeni çözümler üretebilmeleri önemli bir etkidir (Geçgel 2019). Literatürde birçok farklı YZ teknikleri kullanılmaktadır. Bazıları aşağıdaki gibi sıralanabilir:

- Uzman Sistemler
- Bulanık Sistemler
- Doğal Dil İşleme
- Genetik Algoritmalar
- Yapay Sinir Ağları

3.1.3 Uzman Sistemler

Uzman Sistemler (US), insanlarda mevcut olan değerlendirme, teşhis ve bu teşhislerden yorum çıkarma gibi yeteneklerin benzetimini yapabilen yapay zekâ tekniğidir (Allahverdi 2002, Nabiyevev 2005). US, bilgisayar ve iletişim teknolojilerinde mevcut olan hızlı ve tutarlı şekilde veri işleme, çok fazla veri depolama ve etkin iletişim işlemleri üretme gibi avantajları kullanmaktadır. Bu avantajlar ile çözüm odaklı, zeki bir sistem meydana getirebilmeyi amaçlamaktadır (Afrah 2020). US bilgi tabanı, kullanıcı ara birimi, çıkarım motoru veya mekanizması olmak üzere 3 temel bileşenden meydana gelmektedir (Giarratano ve Riley 1994). Şekil 3.1'de US'nin genel yapısı verilmiştir (İçen ve Günay 2014).



Şekil 3.1 Uzman sistemlerin genel yapısı.

Bilgi Tabanı ünitesi bilgi mühendisleri aracılığı ile oluşturulmaktadır. Bilgi mühendisleri birden fazla gerçek insan uzmanına ait bilgileri toplayarak ardından kurallara ve stratejilere dönüştürmektedir. Bilgi tabanı kurallar ve gerçekler bileşenlerinden oluşmaktadır. Bilgi tabanının aynı kalması göz önünde bulundurularak konu kapsamı genişletilebilir ve güncellemelere gerek duyulduğunda eklemeler yapılabilir (Kastal ve Köse 2009). Çıkarım Mekanizması ünitesinde, bilgi tabanında mevcut olan bilgi kullanılarak kullanıcı tarafından sorulan soruya karşılık uygun yanıtlar oluşturulmaktadır. Kullanıcı ara birimi ünitesi uzman sistemlerin yalnızca gerçekleşen duruma ait bilgiyi kullanıcıdan almasını sağlamaktadır. Kullanıcıdan alınan bilgi çıkarım mekanizması ünitesinde işlenir ve bilgi tabanında bulunan uzman bilgileri kullanıcı ara birimi vasıtası ile kullanıcıya aktarılır (İçen ve Günay 2014). Kullanıcı ara birimindeki bu iletişim doğal dil haberleşmesi, grafik haberleşmesi ve soru-cevap haberleşmesi olmak üzere üç farklı şekilde yapılabilmektedir (Baykal ve Beyan 2004). Günümüzde teşhis, sınıflandırma, danışmanlık, simülasyon, planlama ve programlama, veri analizi ve yorumu, kontrol ve eğitim, tahmin ve hata ayıklama gibi işlemlerin yapıldığı birçok uygulamada US kullanılmaktadır (Şahin 2008, Gül Gökay ve Çağatan 2002, Toktaş 2003).

3.1.4 Bulanık Mantık

Bulanık mantık (BM), insanlara ait deneyimleri, verileri kullanarak edinilen değerlerin belirli algoritmalar kullanılarak işlenip, oluşturulacak her kurala karşılık gelen ilgili matematiksel fonksiyonlar ile sonuç çıktılarının elde edilmesidir (Keskenler ve Keskenler 2017). Mantık kavramı eski ve modern dönemlere kadar uzanmakta iken BM kavramı 1900'lerde tanıtılmıştır. 1900'lerin başlarında Jan Lukasiewicz, Aristo'nun $[0, 1]$ den oluşan ikili değer mantığına üçüncü bir değeri daha ekleyerek kendi yöntemini $[0, 1, 2]$ şeklinde ifade etmiştir. Donald E. Knuth, Lukasiewicz'in geliştirdiği $[0, 1, 2]$ değerleri yerine kendi yöntemini $[-1, 0, 1]$ tamsayı değerleri ile tanıtmıştır. Fakat bu 3 değerli

yöntemler önemsenmeyerek geçerli olmamıştır (Dağdelen 1996, Kıyak 2003, Klir ve Yuan 1995, Şen 2012, Yıldırım 1998, Zadeh 1965).

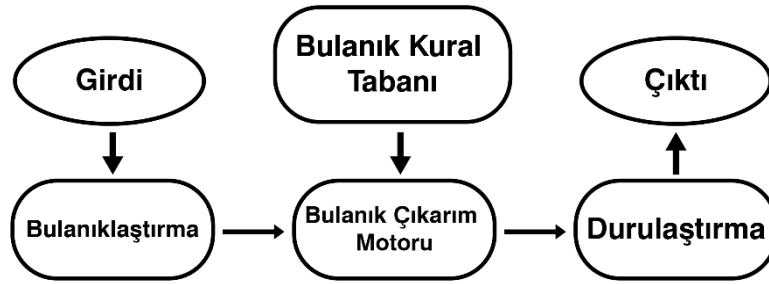
1965 yılında Prof Dr. Zadeh, Jan Lukasiewicz'in çok değerlikli mantık teorisini geliştirerek "Bulanık Kümeler-Fuzz Sets" çalışmasını yayınlayarak ilk kez bulanık mantık kavramını ortaya çıkarmıştır (Değerli 2008). BM, yukarıdaki diğer mantık yöntemlerinden farklı olarak $[0, 1]$ değerlerine ara değerleri de ekleyerek çok değerli sonuçlar üretmektedir. Sözel dilde kullanılan az, çok, biraz, orta, uzun, gibi ifadelere uygun sonuç büyüklükleri üretebilmektedir. Bu işlemde ara değerlerden (0.7, 0.84 gibi) faydalanılmaktadır. Zadeh'in sunduğu bu yöntem ile iki değerli üyeliğin çok değerliliğe taşınması sağlanmış ve genelleme yeteneği de eklenmiştir (Keskenler ve Keskenler 2017).

BM'de, bulanık küme kavramı öneminin vurgulanması gereken bir ifadedir. Bulanık küme temelinde üyelik derecelerinin sürekli olması mantığı bulunan bir nesne sınıfıdır. Bu kümede bulunan elemanların sahip olduğu üyelik dereceleri sıfır ile bir arasında değişmektedir (Zadeh 1965). Klasik küme keskin sınırlara sahiptir. Başka ifade ile var - yok ikilemini barındırır ya vardır ya da yoktur, bir ya da sıfırdır. Bulanık küme klasik kümeden farklı olarak nispi üyelik barındırır. Bulanık kümede "1" tam üyeliği, "0" hiç üye olmamayı, 1 ile 0 arasındaki sayılar nispi üyelikleri ifade etmek için kullanılır (Arslan 2019). Klasik ve bulanık mantık arasındaki farkı anlamakta Bart Kosko'nun yaptığı deney yardımcı niteliğindedir. Bart Kosko, konferans sırasında dinleyicilerden "erkek" bireylerin ellerini kaldırmasını ister ve erkekler ellerini kaldırır, bayanlar kaldırmaz. Böylece bayan olmayan bir dizi oluşmuş olur. Bu sefer "bayan" bireylerin ellerini kaldırmasını ister ve böylece erkek olmayan bir dizi oluşmuş olur. "Erkekler ve erkek olmayanlar" ya da "bayanlar ve bayan olmayanlar" kümesi iki değerli kümeleme değildir. Kosko bu sefer biraz daha zor bir soru sorar:

—Kaç kişi işinden memnun?

Eller yukarı kalkar-iner, birçok dirsek kırılmış durumda aşağı inerler. Fakat birkaçı ellerini iyice kaldırır ya da hiç kaldırmaz. Kişilerin birçoğu bu iki uç arasına dahil olmaktadır. İşte bu durum "işinden memnun olanlar"ın bulanık kümesini tanımlamaktadır. Ardından eller indirilerek bu sefer işinden memnun olmayan kişilerin

ellerini kaldırması istenir. Aynı ellerden birçoğu kalkar-iner, ortalarda bir yerde asılı kalır. İşte bu durum da "işinden memnun olmayanlar"ın bulanık kümesini, diğer bir deyiş ile birincinin tersini tanımlamaktadır. Bu iki deney incelendiğinde birinci deneyde erkek dizisi ile bayan dizisi kesişemez. Çünkü bir insan erkek ya da bayandır, ikisi birden olamaz. İkinci deneyde ise kişilerin çoğu işinden kısmen memnun kısmen memnun değildir; başka ifade ile hem memnun hem de değildir. Kişilerin çok azı işinden % 100 memnun veyahut % 100 memnun değildir. İşte burada işinden memnun olma durumu 1 ile, memnun olmama durumu 0 ile gösterildiğinde; işinden mutlak memnun olmaya daha yakın fakat az miktarda şikayeti olan bir kişi 0.83 ya da 1'e yakın herhangi bir ara değer ile ifade edilebilir. BM, bu çeşit kümeleri ve tanımları kullanarak işlem yapmaktadır. Klasik mantıkta, aynı durum işinden memnun olanlar ve olmayanlar olmak üzere kesin iki küme oluşturulması ile devam ettirilerek bitirilmektedir. Fakat bu kümeler gerçeği yansıtmamaktadır. Çünkü insanların iş memnuniyetini % 100 memnun, % 100 memnun değil olmak üzere iki kısımda sınırlamaktadır (Işıklı 2007). Bundan yola çıkarak klasik mantığın değişen, gelişen yaşamımızda BM'ye göre daha kısır sonuçlar ürettiği söylenebilmektedir. Şekil 3.2'de BM sistemlerinin genel yapısı verilmiştir (Katircioğlu vd. 2018).



Şekil 3.2 Bulanık mantık sistemlerinin genel yapısı.

Bulanık küme oluşturma işlemi bulanıklaştırma olarak tanımlanmaktadır. Girdiler ve oluşturulan kurallar ile birlikte elemanlar arasında bağlantı kurulması ve üyelik fonksiyonlarının da kullanılması ile bulanık kümeler oluşturularak bulanıklaştırma süreci tamamlanır (Işıklı 2010). Bulanıklaştırma sürecinde birçok üyelik fonksiyonu kullanılmaktadır. Literatürde Cauchy, Yamuk, Çan Eğrisi, Sinüsoid, Gaussian, Üçgen, S, Pi (π) ve Sigmoid üyelik fonksiyonları yaygın olarak kullanılmaktadır (Osman ve Uygunoğlu 2005, Akyazı vd. 2011, Özdemir ve Kalınkara 2020). Bulanık Kural Tabanı

ünitesi, veri tabanındaki girdiler ile çıktı değişkenleri arasındaki bağlantıyı oluşturan mantıksal, If-Then (Eğer-İse) türünde yazılabilmekte olan tüm kurallar bütünüdür. Bu kurallar Şekil 3.2’de görüldüğü gibi sadece girdiler ile çıktılar arasındaki tüm ara bağlantılar dikkate alınarak yazılmaktadır (Katircioğlu vd. 2018).

Bulanık Çıkarım Motoru ünitesinde girdi ve çıktı bulanık kümeleri arasındaki mevcut tüm parça ilişkiler bir arada toplanarak sistemin bir çıkışlı çalışması için gerekli işlemler yapılmaktadır. Bu üniteye bilginin modellenmesi için çeşitli yöntemlere başvurulmaktadır (Sivanandam vd. 2007). Çıkarım yöntemleri olarak bilinen bu metotlar Mamdani, Larsen, Tsukamoto ve Tagaki-Sugeno-Kang yöntemleridir (Akıllı vd. 2014). Mamdani bulanık çıkarım yöntemi bu metotlar içerisinde en çok kullanılanıdır. Bu metot ayrıca ilk bulanık mantıkla çalışan kontrol sistemlerine uygulanmıştır. Mamdani metodunda toplama işleminin ardından veriler bulanık kural tabanı birimine iletilir. Bu birimden çıkan çıktılar bulanık tabanlı olduğu için durulaştırma ünitesine iletilir. Durulaştırma ünitesinde bulanık tabanlı çıktı değerlerinden gerçek hayatta kullanılacak sayısal değerler elde edilir (Temurtaş 2007). Mamdani bulanık mantık metodunun sezgi yeteneğine dayalı olması, yaygın olarak kabul edilmesi ve insani terimlere yakın olması gibi üstünlükleri bulunmaktadır (Değerli 2008).

Sugeno bulanık çıkarım metodu Mamdani metoduna benzer şekildedir. İki metotta da girdiler ve bulanıklaştırma işlemleri aynı şekildedir. Farklı olarak Sugeno bulanık mantık metodunda üyelik fonksiyonlarının çıktıları sabit ya da doğrusaldır (Temurtaş 2007). Sugeno bulanık mantık metodunun etkin olarak sayısal sonuç çıkarma ve matematiksel analiz işlemlerinde kullanılması, optimizasyon ve uyarlama işlemlerinde başarılı olması gibi avantajları bulunmaktadır (Değerli 2008).

BM’nin literatüre kazandırılması ile birlikte birçok araştırma ve uygulama yapılmaya başlanılmıştır. Bu ilk çalışmalardan başarılı sonuçların elde edilmesi, sonraki yıllarda yapılan çalışmalar için bir temel oluşturmuştur. BM’nin tanıtılması ile 1900’lü yıllarda yapılan ilk uygulama çalışmalarından birkaçı aşağıdaki gibidir:

- 1975 tarihinde Assilian ve Mamdani bir buhar makinasına ait kontrol sistemine BM'yi uygulayarak ilk defa bir uygulamada BM'nin kullanılmasını sağlamışlardır. Kontrol sisteminde “Eğer türbine ait hızlanma ivmesi yükseliyorsa basınç çok düştüğünde buhar vanasını bir miktar aç” benzeri kurallar ile BM kullanılmıştır (Bih 2006).
- Japonya'da gelişmiş bir ulaşım aracı olan Sendai metrosu, yaklaşık 14 km mesafe boyunca 16 farklı istasyonda durmaktadır. Fakat bu metro öylesine fazla yumuşak hareket etmektedir ki, içi su dolu bir akvaryumun, suyu hiç dökülmeden taşınması mümkündür. Bu dizge BM tabanlı çalışmaktadır ve insanların yapmış oldukları dizgelere kıyas ile etki-tepki süresi üç kat daha kısadır. Buna ek olarak % 10 yakıt tasarrufu sağlamakta ve şimdiye kadar hiçbir tehlikeye sebebiyet vermemiştir (Işıklı 2007).
- 1988'de Tokyo Borsasında meydana gelen krizin sinyallerini 18 gün önceden haber veren sistem Yamaichi Securities firmasının geliştirdiği BM temelli bir akıllı sistemdir (Keskenler ve Keskenler 2017).

Günümüzde BM savunma sanayi, uzay, kontrol sistemleri ve havacılık sanayi gibi birçok farklı endüstri alanında kullanılmaktadır (Cirstea vd. 2002).

3.1.5 Doğal Dil İşleme

Doğal dil, insanların kendilerini anlatabilmeleri ve iletişim yapabilmeleri için gerekli olan bir vasıttır. Dil biliminin araştırma sahası insanların dili nasıl edindiği, ürettiği ve kavradığıdır. Bu sahada araştırma yapan bilim insanları dil bilimci olarak bilinmektedir. Dil bilimciler kural tabanlı yaklaşımlar ileri sürerek dilsel ifadeleri işlemeye çalışmışlardır. Fakat doğal dilin üretken bir şekilde kullanımı kuralları yetersiz kılabilir. Bu sebeple doğal dil ifadelerine uygulamak için istatistiksel yaklaşımlar ileri sürülmüştür. İstatistiksel yaklaşımlar aracılığı ile doğal dil için ortak prensipler oluşturulmaya çalışılmıştır (Babüroğlu 2019, Schütze ve Manning 1999).

Diğer taraftan YZ'nin kullanıldığı birçok uygulamada başarılı sonuçlar alınması ile insan-makine haberleşmesi gündeme gelmiştir. Bununla birlikte doğal dillerin

bilgisayarlara öğretilmesi gereği gündeme gelmiştir. Bu gelişmeler sonucunda dilbiliminin ve YZ'nin bir ortak alt dalı olan Doğal Dil İşleme (DDİ) doğmuştur.

Dilbilimi, dillerin genel ve özel niteliklerini, dillerin doğuşlarını ve zaman içerisindeki gelişmelerini ve dil olayları gibi konuları inceleyen bir bilim dalıdır. DDİ, yapay zekanın ve dilbiliminin ortak bir alt dalı olarak doğal dillerdeki metinlerin, ses dalgalarının algılanması ile gerekli yazılımlar kullanılarak çözümlenmesi ve bilgisayar ortamına aktarılmasını incelemektedir (Ay 2019).

DDİ, doğal dillerin sahip olduğu kurallı yapının çözümlenerek anlaşılması veya yeniden üretilmesini amaçlamaktadır. Bu çözümlenme ile insanoğluna, bilgi sağlama, soru-yanıt makineleri, konuşma sentezi, yazılı dokümanlar için otomatik çevrilme, komut anlama ve otomatik konuşma, konuşma üretme, otomatik metin özetleme gibi kolaylıklar sağlamaktadır. Örneğin; kelime işlem yazılımlarının tamamı birer imla düzeltme aracı taşımaktadır. Bu araçlar aslında yazılan metnin çözümlenerek dil kurallarının denetlendiği DDİ yazılımlarıdır (Tarcan ve Çakar 2008). DDİ çalışmalarında ulaşılmak istenen temel amaçlar şu şekilde sıralanabilir (Delibaş 2008):

- Doğal dillerin işlev ve yapılarını daha fazla anlamak
- Bilgisayarlar ile insanlar arasında doğal dilin arabirim olarak kullanılması ve bilgisayar ile insan arasındaki haberleşmenin kolaylaştırılmasını sağlamak
- Bilgisayarın kullanılması ile dil çevirisi yapmak

Bu temel amaçlar çerçevesinde DDİ üzerinde yapılan ilk çalışmalar kronolojik sıra ile aşağıdaki gibi sıralanabilir.

1954'te "The Georgetown Experiment" deneyi ile altı Rusça cümlelerin İngilizceye çevirebilmesi sağlanmıştır. Deney sahibi 3-5 yıl aralığında dil çevrimi probleminin çözüleceğini öngörse de bu öngörü gerçekleşmemiştir.

1966'da yayınlanan ALPAC (Automatic Language Processing Advisory Committee-Otomatik Dil İşleme Danışma Kurulu) raporunda yıllarca süren çalışmalara rağmen iyi

sonular alınamadığı ve makine evrimi iin verilen destek fonlamalarının dştüğü ifade edilmiştir.

1960’larda oldukça başarılı sayılabilecek Terry Winograd’in geliřtirdiğı “SHRDLU” DDİ sistemi tanıtılmıştır (Kaya 2018).

1970’te Woods tarafından dilin matematiksel modeli oluşturulmuřtur. Bu alıřmada doęal dil analizinin yapılması iin geniřletilmiş geiř aęları yaklařımı ileri sürülmüřtür. Bu yaklařım sayesinde sonlu sayıdaki bir küme aracılığı ile verilen herbir harfe göre bir sonrakinin hangi harf olabileceęi sıralanmıştır (Woods 1970).

1970’lerde canlıyı kavrayabilen birok program tanıtılmıştır;

- 1973’te MARGIE programı (Schank vd. 1973).
- 1976’da TaleSpin programı (Meehan 1976).
- 1977’de QUALM programı (Lehnert 1977).
- 1978’de SAM ve PAM programı (Cullingford 1977, Wilensky 1977).
- 1979’da Politics programı (Carbonell 1978).
- 1981’de Plot Units programı (Lehnert 1981) geliřtirilmiştir.

Bu süreç ierisinde insan konuřmasının simüle edilebildiğı Parry, Racter, Jabberwacky programları da geliřtirilmiştir (Kaya 2018).

1977’de Dempster ve arkadařları alıřmalarında eksik, kısaltılmış ve sansürlenmiş veri seti ierisinden doęru bilgi ekmek amacı ile bir algoritma geliřtirmişlerdir (Dempster vd. 1977).

1983’te Bahl ve arkadařları alıřmalarında konuřma tanıma işlemleri iin istatistiksel algoritmalar önermiştir. Bu algoritmalar aracılığı ile konuřma girdisi analiz edilip eksik verinin tamamlanmasına alıřılmıştır (Bahl vd. 1983).

1990 yılında Brown ve arkadaşları yaptıkları çalışmada istatistiksel yaklaşımların kullanılması ile oluşturulan model vasıtasıyla Fransızcadan İngilizceye çeviri yapılmasını amaçlamıştır (Brown vd. 1990).

1993'te Hindle ve Rooth çalışmalarında kişi ögesinin belirsiz olduğu edat cümlelerinde kişiyi tahmin etmek için bir algoritma geliştirmişlerdir (Hindle ve Rooth 1993).

2004 yılında Nivre ve Scholz tarafından İngilizce cümlelerin ögelerine ayrıldığı bir algoritma tasarlanmıştır. Bu algorithmada bulunan en büyük özellik hafıza tabanlı öğrenme temeline dayalı olmasıdır (Nivre ve Scholz 2004).

Önerilen algoritmalar ve yaklaşımlar DDİ üzerine yapılan çalışmaların gelişmesine katkı sağlamıştır. Bu bağlamda günümüzde dokümanların değerlendirilmesinde, uluslararası çalışan şirketlerin müşteri profili belirlemede, e-ticarette, savunma ve istihbaratta (güvenlik ve suçlu tespiti), makine çevirisi, e-sözlük, mobil telefonlarda konuşma algılama sistemleri, otomatik özet çıkarma ve görme engelliler için bilgisayar kullanımı gibi birçok uygulama alanında DDİ kullanılmaktadır (Tarcan ve Çakar 2008).

Literatürde de birçok farklı alanda DDİ üzerine çeşitli uygulamalar yapılmıştır. Swartz ve arkadaşları radyoloji alanında çalışma yapmıştır. Çalışmada hasta raporlarının incelenmesi ile pozitif ve negatif olarak yapılan bir sınıflandırma amaçlanmıştır. Bu çalışmada doğal dil işleme ile kullanıcının kişiselleştirilebildiği açık kaynak kodlu bir araç tasarlanmıştır. Bu aracı kullanarak yapılan sınıflandırmada % 95'in üzerinde bir doğruluk oranı tespit edilmiştir (Swartz vd. 2017).

Zhang ve arkadaşları şantiye kaza raporlarının incelenmesi ile kaza nedenlerini DDİ kullanarak tespit eden bir model tasarlamıştır. Bu modelde raporlarda bulunan kelimelerin analizi yapılarak kaza nedenlerini tespit etmek hedeflenmiştir. Algoritma kısmında yalın bayes, ensemble metodu, doğrusal regresyon, karar ağacı, destek vektör makineleri ve K- en yakın komşu olmak üzere 5 farklı algorithmadan yararlanılmıştır. 16.323 adet rapor incelenen modelde en çok doğru kaza nedeni ensemble metodu kullanılarak oluşturulan DDİ modeli tarafından saptanmıştır (Zhang vd. 2019).

Tanguy ve arkadaşları havayolu güvenlik raporlarında kullanılmak üzere bir DDİ modeli geliştirmiştir. Çalışmada farklı tiplerde ve dillerde yazılmış raporlar arasından acil güvenlik sorunu teşkil eden durumlar DDİ aracılığıyla tespit edilmek hedeflenmiştir. Tekrar edilen ya da bilinen durumların tespit edilmesinde başarı miktarı yüksek iken nadir karşılaşılması ile birlikte acil güvenlik sorunu meydana getiren durumlarda başarı miktarı oldukça düşük çıkmıştır (Tanguy vd. 2015).

Cheng ve arkadaşları robotik alanında bir çalışma yapmışlardır. Bu çalışmada DDİ kullanılarak verilen komutlar vasıtasıyla muhtemel belirsizliklerin önüne geçilmek hedeflenmiştir. Çalışmada robotik bir kola komut verme işlemi konuşarak yapılmakta ve robot bu komutları görev sıralamasına koymaktadır. Eğer yeni görev ile mevcut görev arasında uyumsuzluk meydana geliyor ise robot ara görevler üreterek geçiş yapmaktadır (Cheng vd. 2014).

Wächter ve arkadaşları yaptıkları çalışmada insansı bir robotun sesli komutları algılaması yoluyla verilen görevi yerine getirmesini hedeflemişlerdir. Bu çalışmada robot DDİ kullanılarak programlanmış ayrıca robota bir mutfak alanı öğretilmiştir. Verilen komutlar aracılığıyla robottan istenilen eylemlere dair planlamanın yapılması beklenilmiştir. Bu planlar kapsamında robot verilen komutu yerine getirmeye çalışmıştır. Başarısızlık durumunda ise yeni bir plan kullanımı ile robot tekrar başlatılmaktadır. Verilen tüm komutlar örnek bir senaryonun oluşturulması sayesinde robot tarafından yerine getirilmiştir (Wachter vd. 2018).

3.1.6 Genetik Algoritmalar

Kainatta her biri bir amaç için mükemmel olarak yaratılmış canlılarda kalıtımı sağlamak ile görevli kromozom olarak bilinen birimler mevcuttur. Bu birimlerde meydana gelen eşleştirme, çaprazlama ve mutasyon gibi işlemler sonucunda iyi olan kromozom hayatta kalmakta kötü olan ise ölmektedir. Bu biyolojik sürecin taklit edilmesi ile genetik algoritmalar (Genetic Algorithms-GA) geliştirilmiştir. Bu bağlamda GA aslında birer

arama ve optimizasyon algoritması niteliğindedir (Deb 2001). GA üzerine tarihsel süreçte meydana gelen ilk gelişmeler aşağıdaki gibi sıralanabilir:

- GA'dan ilk kez 1967 yılında Bagley yaptığı çalışmada söz etmiştir.
- Literatürde ilk kez 1975 yılında Bagley'in öğrencisi olan John Holland tarafından uygulanmaya başlanmıştır. Holland, GA'nın birleşimsel en iyileme problemlerine karşı etkili bir çözüm sağladığını ifade etmiştir.
- Holland'ın öğrencisi David Goldberg tez çalışmasında GA kullanmıştır. Yaptığı tez çalışmasında gaz boru hattının kontrol edilmesi ile ilgili yaşanan bir problemde GA kullanarak çözüm sağlamıştır (Çelebi 2007).
- 1985'te Davis, GA ilk defa atölye çizelgeleme problemlerinin çözümü için etkili şekilde kullanmıştır.
- 1987 yılında GA ilk kez Liepis iki makineli çizelgeleme işlemlerinde kullanmıştır.
- 1990'da Biegal ve Daven yaptıkları çalışmada GA bütünleşik imalat çevrimi içerisinde bir, iki ve daha fazla makineden oluşan sistemlere tatbik etmişlerdir.
- 1995 yılında Chen vd. literatürde kullanılan diğer sezgisel yaklaşımlara göre GA'ndan daha iyi sonuçlar alındığını ifade etmişlerdir.
- 1996'da Nakano atölye çizelgeleme problemleri için GA'nın ikili kod sistemindeki ifadesini kullanmıştır (Elmas 2018).

Genetik algoritmaların sahip olduğu avantajlar şöyle sıralanabilir:

- Türevsel ifadelerle ihtiyaç duymama.
- Birçok muhtemel çözümleri araştırma.
- Karmaşık işlemlerin kullanılmasına gereksinim duymama.
- Neredeyse her çeşit probleme uygulanabilirlik.
- Büyük boyuttaki arama uzaylarında bile hızlı sonuç verebilme (Zorlu ve Sunca 2017, Eroğlu ve Şişman 2020).

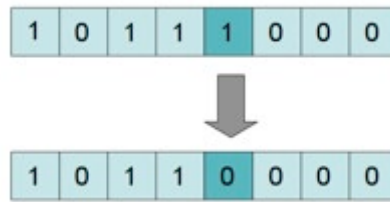


Şekil 3.3 Genetik algoritmalarda işlem adımları.

Şekil 3.3'te GA'da meydana gelen işlem sırası verilmiştir (Zorlu ve Sunca 2017). İlk olarak kullanılacak parametreler, uygunluk fonksiyonu ve durdurma kriteri tanımlanmaktadır. Ardından parametreler temsil edilip başlangıç popülasyonu oluşturulmaktadır. Burada bir problemdeki çözüm için kullanılan en küçük parametre genidir. Genler algoritma yapısında birçok defa bir bit uzunluğunda 0 ya da 1 değeri olarak kullanılmaktadır. Çözüme ulaşılmasında katkı yapan genler ayrıca bir araya gelip birlikte bireyi temsil eden kromozomları oluştururlar. Bu kromozomlar aynı zamanda bir çözüm kümesi niteliğindedir. GA yapısından ötürü tek doğruyu kullanarak çözüme ulaşmak yerine aynı zaman diliminde birçok çözüm kümesine yoğunlaşarak çözüme ulaşmaktadır. Burada popülasyon kavramı ortaya çıkmaktadır. Çünkü algoritma yapısında kullanılan çözüm kümelerinin bütünü bir popülasyon teşkil etmektedir (Işık vd. 2017). Oluşturulan

popülasyonun boyutu küçük olduğunda arama uzayı içerisinde sınırlı bölgede arama işlemi başladığı için optimum çözüme ulaşma süresinde gecikme olacaktır. Aksine büyük boyuttaki bir popülasyonda ise işlem fazlalığı olması ile birlikte algoritma hızında da yavaşlama görülecektir. Bundan dolayı popülasyonun boyutu dikkatli bir şekilde seçilmelidir. Popülasyonun oluşturulmasının ardından, popülasyonda mevcut her bir kromozoma karşılık gelen uygunluk değeri öngörülmektedir. Uygunluk değeri istenen çözüme ne kadar yaklaşıldığını gösteren bir kıstastır (Karaboğa 2017). Uygunluk değerlendirilmesi adımında probleme göre değişken olan uygunluk fonksiyonu devreye girmektedir. Burada uygunluk fonksiyonunun yardımıyla popülasyondaki kromozomlar uygunluk değerlerine göre ayırt edilmektedir. Uygunluk değeri yüksek olan kromozomlar hayatta kalır iken uygunluk değeri düşük olanlar ölmektedir. Seçme adımında üretilecek yeni nesil için popülasyona rulet tekerleği, turnuva ve elitizm gibi yöntemler ile seçme işlemi uygulanmaktadır. Çaprazlama işleminde, seçme işleminden geçen kromozomlardan rastgele iki kromozom seçilmektedir. Daha sonra tek nokta, çift nokta, çok nokta ve uniform çaprazlama yöntemleri ile çaprazlama yapılmaktadır (Eroğlu ve Şişman 2020).

Mutasyon işleminde, çaprazlama yapılmış kromozomlardaki genlerden herhangi birinde değişiklik yapılmaktadır. Şekil 3.4'te örnek bir mutasyon işlemi verilmiştir. Bu işlem sayesinde popülasyondaki çeşitlilik korunmakla birlikte erken yakınsama engellenmiş olmaktadır.



Şekil 3.4 Genetik algoritmalarda mutasyon işlemi.

Durdurma kriteri işleminde, mutasyona uğrayan kromozomlar istenen optimum çözüm değerine en yakın olması ya da algoritmanın maksimum iterasyon sayısına ulaşması gibi kıstaslar çerçevesinde değerlendirilmektedir. Probleme göre bu kıstaslardan birisi sağlandığında algoritma tamamlanmaktadır. Aksinde ise uygunluk değerlendirilmesi

işlem adımına dönülerek devamındaki adımlar tekrarlanmaktadır. Bu döngü kıstaslardan birisi sağlanana dek devam etmektedir.

Günümüzde GA sinyal işleme, görüntü ve ses tanıma, havacılık ve uzay, otomotiv mühendisliği, tıp, yapay zeka, işletme, robotik ve istatistik gibi birçok alanda yol bulma, çizelgeleme, en iyileme vb. çeşitli problemlerin çözümünde yaygın şekilde kullanılmaktadır (Gençal 2019, Çakır vd. 2018).

Literatürde Okur ve Atlas yaptıkları çalışmada Eskişehir Halk Ekmek A.Ş.'nin bünyesindeki dağıtım araçlarının, 51 farklı satış büfesine en kısa mesafeyi izleyerek ulaşması için gerekli rota hesaplamasına optimum çözümü aramış ve mevcut durum arasındaki karşılaştırmayı yapmışlardır. Çalışmada üretim tesisi ile satış büfeleri arasındaki mesafeler Google Maps harita servisi aracılığıyla sağlanmıştır. İlgili problemin çözümü için genetik algoritma 2000 kez çalıştırılarak yaklaşık olarak 25 sn'de sonuç alınmıştır. Mevcut durumda işletmenin 17 yıllık tecrübesi ile 3 araç için günlük toplam kat edilen mesafe 236.54 km olmaktadır. Genetik algoritma kullanılarak yapılan araç rotalama işleminde 3 araç için günlük toplam kat edilen mesafe 210.24 km olarak hesaplanmıştır. Bu değerler kıyaslandığında 26.3 km'lik bir iyileştirmenin olduğu gözlemlenmiştir (Okur ve Atlas 2020).

Toprak ve arkadaşları yaptıkları çalışmada çoklu-odaklı görüntü birleştirme işlemleri için piksel tabanlı yeni bir yöntem önermişlerdir. Önerilen yöntemde, kaynak görüntülerin sahip olduğu NDF (Noktasal Dağılım Fonksiyonları) fonksiyonlarına ait dağılım parametreleri GA kullanılarak araştırılmaktadır. Bu sayede optimum dağılım parametrelerine sahip NDF elde edilmektedir. Bu fonksiyonlar aracılığıyla kaynak görüntüler yapay şekilde bulanıklaştırılmaktadır. Sonunda üretilmiş olan yapay görüntüler ve kaynak görüntülerin kullanılması ile görüntülerdeki net pikseller tespit edilmektedir. Sonuç olarak net pikseller taşınarak birleşik görüntü oluşturulmaktadır. Çalışmada önerilen yöntem gerçek ve yapay görüntü setlerine uygulanarak denenmiş ve diğer klasik metotlar ile kıyası yapılmıştır. Sonuçta önerilen yöntemden sayısal ve görsel olmak üzere yapılan değerlendirmeler çerçevesinde klasik metotlara göre daha başarılı sonuçlar alındığı sunulmuştur (Aslantaş vd. 2013).

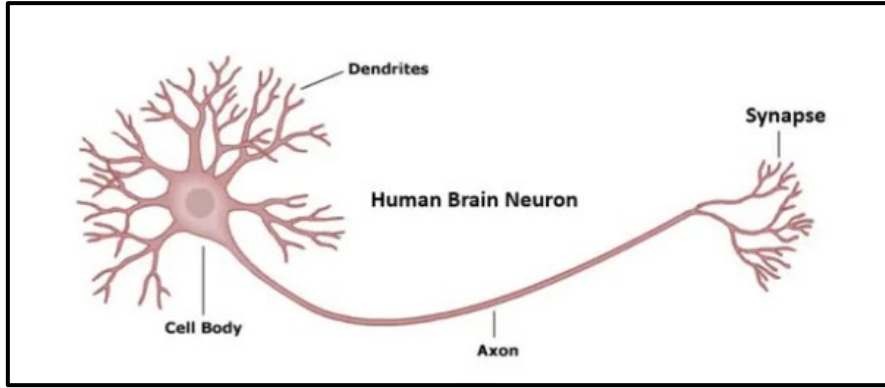
Haznedar ve arkadaşları çalışmalarında karaciğer mikrodizi gen ifadesi kanser verisini sınıflandırmak için GA'dan faydalanmışlardır. Çalışmada mikrodizi gen ifadesi verileri içerisinde genlerin sayıca çokluğu ve bu verilerin aralarında doğrusal olmayan bağlantıların bulunması gibi birçok problem bulunduğu ifade edilmiştir. Bundan dolayı geleneksel sınıflandırma algoritmaları kullanılarak yapılan sınıflandırmalardaki başarı miktarı yetersiz kaldığı belirtilmiştir. Bu nedenle çalışmada ilgili kanser verisini sınıflandırmak için ANFIS (Adaptive Network Based Fuzzy Inference Systems-Uyarlamalı Ağ Tabanlı Bulanık Mantık Çıkarım Sistemi) modeli kullanılmıştır. Bu model GA, geri yayılım, Hibrid algoritmaları ile eğitilip performans kıyası yapılmıştır. Sonuçta GA kullanımı ile yapılan ANFIS modeli eğitiminin diğer yaklaşımlara göre daha başarılı olduğu gözlemlenmiştir (Haznedar vd. 2017).

Halat ve Özkan yaptığı çalışmada muhtemel bir İstanbul depreminin sonrasında oluşabilecek hasarların havadan gözlemlenmesini hedeflemiştir. Böyle bir deprem durumunda ilk 24 saat zarfında İstanbul'dan kalkan bir İHA'nın (İnsansız Hava Aracı) nasıl bir rotada uçması gerektiği bir problem teşkil etmektedir. Bu problem için İstanbul'da İHA tarafından ziyaret edilecek 230 tane aday grid nokta belirlenmiştir. Aday noktaların her biri için nüfus yoğunluğu ile deprem risk ağırlığının birleşiminden oluşan ağırlık değerleri atanmıştır. Çalışmada İHA'nın en çok aday noktanın ağırlık değerlerini toplayacağı rotada menzil alması hedeflenmiştir. Bu bağlamda gerçek hayata uygun şekilde deprem sonrasında İHA'nın ayrıldığı havalimanından başlayarak yaptığı günlük ziyaret (ya da görüntü sayısı) durumlarının bulunduğu 15 farklı senaryo yapılmıştır. GA, SA (Simulated Annealing-Benzetilmiş Tavlama) ve IP (Integer Programming-Tam Sayılı Programlama) olmak üzere önerilen üç model ile senaryolar çözülmüştür. 15 senaryo içerisinde 2'si için optimum çözümleri IP modeli elde etmiştir. Diğer senaryolarda GA daha başarılı sonuçlar vererek uygun CPU sürelerinde sonuç üretmiştir (Halat ve Özkan 2021).

3.2 Yapay Sinir Ağları

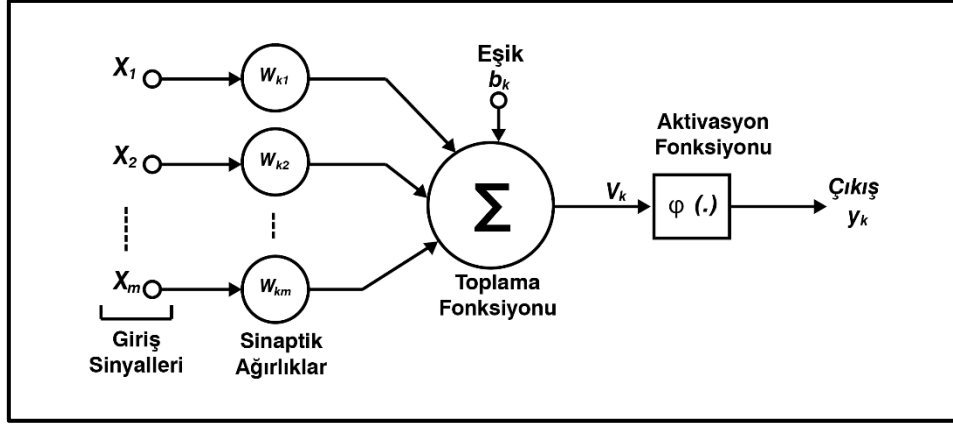
Bilimsel çalışmalarda, çeşitli yapılarıdaki canlılar ve harikulade işleyen sistemleri incelenmekte ve bilim dünyasına önemli kazanımlar sunulmaktadır. Canlılardaki en kompleks sistemlerden birisi de insan beynidir. İnsan beyni muhteşem yapısı ve işleyişi ile uzun yıllarca araştırılmış ve bilimsel çalışmalara konu olmuştur. Muazzam şekilde çalışan bu sistemdeki işleyiş incelenmiş ve matematiksel modellenmesi yapılmıştır. Böylece yapay zekânın bir alt çalışma alanı YSA (Yapay Sinir Ağları) meydana gelmiştir. İnsan beyninin yapabildiği öğrenme, sınıflandırma, tahmin etme ve karar verme gibi birçok yetki YSA için referans niteliğindedir. YSA bu yetki işlemlerini yapabilmek için elektrokimyasal yöntemli sinir hücreleri ile kıyaslandığında daha hızlı çalışabilmektedir (Koyuncu vd. 2019). Bu ağlar yapıca sinir ağları ile benzerlik göstermektedir.

3.2.1 YSA Nöron Modeli



Şekil 3.5 Biyolojik sinir hücresi.

Şekil 3.5'te biyolojik sinir ağındaki sinapslar, sinir hücresi, dendritler, hücre gövdesi ve akson YSA'da sırası ile ağırlıklar, işlem elemanı, toplam fonksiyonu, aktivasyon fonksiyonu ve eleman çıkışına karşılık gelmektedir. YSA yapısında her girdi kendi ağırlığı ile çarpılmakta ve ardından toplama fonksiyonunda eşik değeri de eklenerek toplama işlemine tabi tutulmaktadır. Buradan çıkan sonuç aktivasyon fonksiyonunda işlenerek çıkış bilgisi üretilmektedir. Bu çalışma yapısı ile bir YSA nöronu Şekil 3.6'da modellenmiştir (Alçın 2017).



Şekil 3.6 YSA nörön modeli.

Şekil 3.6’ da bir k nörönüne bağlanmış olan j sinapsının giriş sinyali x_j şeklinde ifade edilmektedir. Burada j değeri $[1, m]$ tam sayı aralığında değişmektedir. Bu sinyal w_{kj} sinaptik ağırlık sinyali ile çarpılmaktadır. Sinaptik ağırlık sinyali w_{kj} ifadesinde ilk alt indis ilgili nörönü, ikinci alt indis ağırlığın belirttiği ilgili sinapsa ait giriş ucunu temsil etmektedir. Bu ağırlık sinyali biyolojik sinapsa kıyasen negatif ile beraber pozitif değerler de içermektedir. Çarpım sonucu oluşan sinyallere b_k eşik değerinin de eklenmesi ile toplama fonksiyonunda toplama işlemi uygulanmaktadır. Böylece net girdi oluşturulmaktadır. Eşik değeri pozitif değerli olduğunda net girdiyi yükseltme negatif değerli olduğunda ise düşürme niteliğine sahiptir. Eşik değeri istenilmezse kullanılmayabilmektedir. Net girdinin $\varphi(.)$ ile ifade edilen aktivasyon fonksiyonunda işlenmesi ile y_k nörön çıkış sinyali elde edilmektedir. Bir k nörönüne ait matematiksel eşitlik denklem (3.1) kullanılarak Denklem (3.2)’deki gibi elde edilmiştir (Alçın 2017).

$$u_k = \sum_{j=1}^m w_{kj} \cdot x_j \quad (3.1)$$

$$y_k = \varphi(u_k + b_k) \quad (3.2)$$

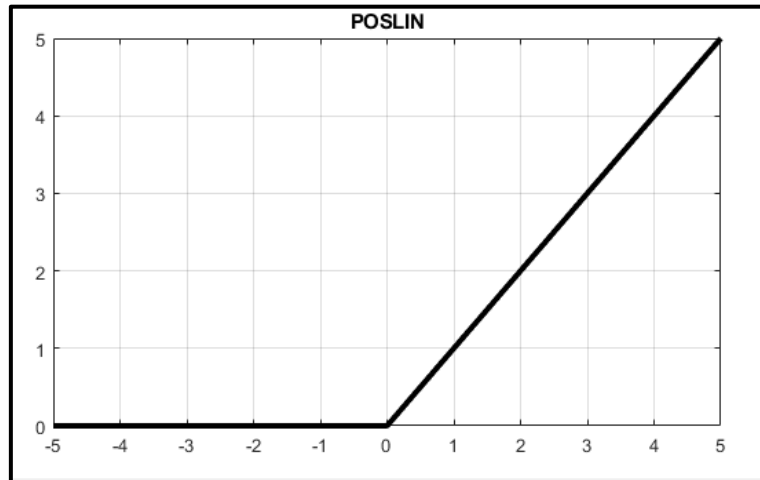
3.2.2 Aktivasyon Fonksiyonu

Aktivasyon fonksiyonu, tasarımda çözülmesi planlanan problemin ilgili özelliklerinin karşılanması amacı ile kullanılmaktadır. Nöron kullanılan AF'ye bağlı olarak problemi çözmeye uğraşmaktadır. Bu bağlamda bir nöronun beklenen gerçek çıkış kullanılan AF'ye göre değişmektedir (Alçın 2017). Bu yüzden problemin çözümünde kullanılan AF'nin seçilmesi büyük önem teşkil etmektedir. Seçilen AF vasıtasıyla çıkış sinyali gerekli sonlu aralıklarda sınırlandırılmaktadır. YSA sistemlerinde AF'ler doğrusal (linear) ya da doğrusal olmayan (nonlinear) formlarda kullanılmaktadır (Akçay vd. 2020).

3.2.2.1 Doğrusal Aktivasyon Fonksiyonları

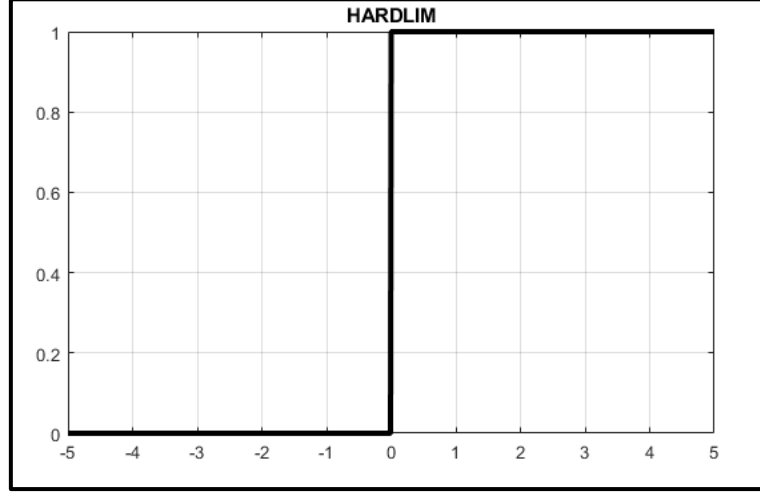
Karmaşık olmayan problemlerin çözümünde doğrusal AF'ler kullanılmaktadır. Literatürdeki YSA çalışmalarında çeşitli doğrusal AF'ler kullanılmıştır. Bu tez çalışmasında AF kütüphanesinin tasarlanması için PosLin, HardLim, HardLims, SatLin, SatLins, TriBas doğrusal fonksiyonları kullanılmıştır. Aşağıda ilgili fonksiyonların verilen sırayla denklem ve sonuç eğrileri verilmiştir. Denklemlerde kullanılan n değişkeni bir rasyonel sayı olmak üzere fonksiyonların giriş değerini ifade etmektedir.

$$\text{PosLin} = \begin{cases} n, & \text{eğer } n \geq 0 \\ 0, & \text{eğer } n \leq 0 \end{cases} \quad (3.3)$$



Şekil 3.7 PosLin AF eğrisi.

$$\text{HardLim} = \begin{cases} 1, & \text{eğer } n \geq 0 \\ 0, & \text{diğer durumlarda} \end{cases} \quad (3.4)$$



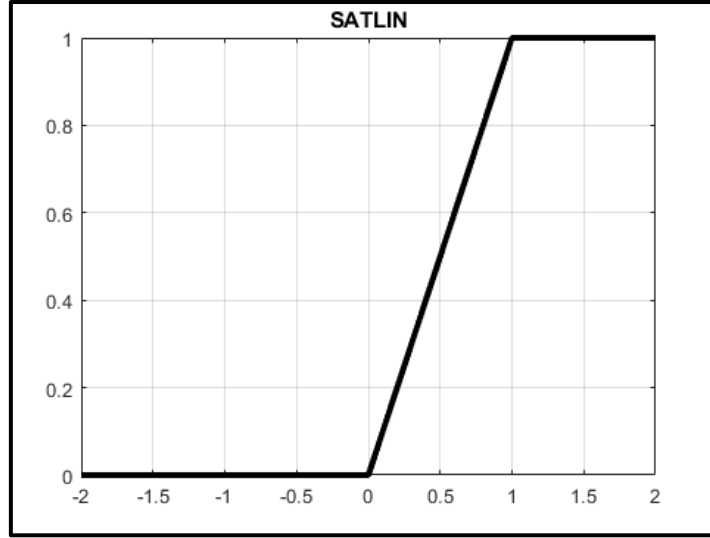
Şekil 3.8 HardLim AF eğrisi.

$$\text{HardLims} = \begin{cases} 1, & \text{eğer } n \geq 0 \\ -1, & \text{diğer durumlarda} \end{cases} \quad (3.5)$$



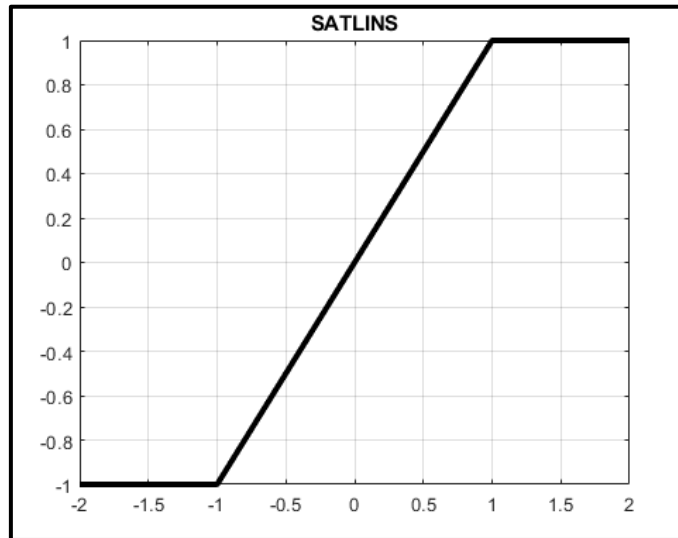
Şekil 3.9 HardLims AF eğrisi.

$$\text{SatLin} = \begin{cases} 0, & \text{eğer } n \leq 0 \\ n, & \text{eğer } 0 \leq n \leq 1 \\ 1, & \text{eğer } n \geq 1 \end{cases} \quad (3.6)$$



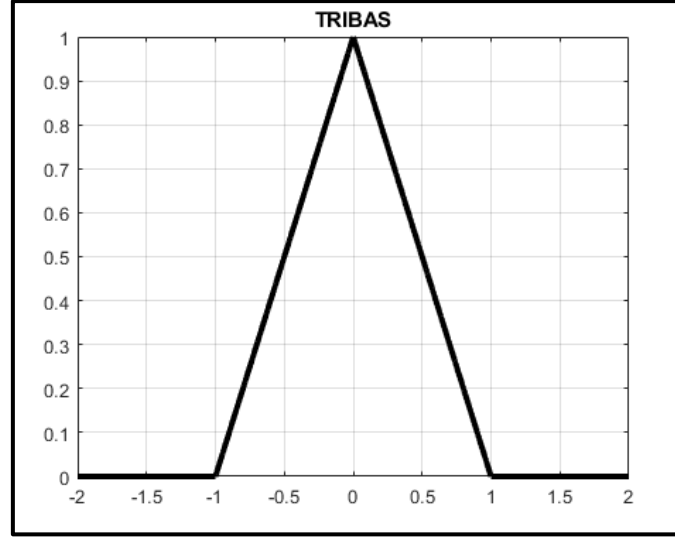
Şekil 3.10 SatLin AF eğrisi.

$$\text{SatLins} = \begin{cases} -1, & \text{eğer } n \leq -1 \\ n, & \text{eğer } -1 \leq n \leq 1 \\ 1, & \text{eğer } n \geq 1 \end{cases} \quad (3.7)$$



Şekil 3.11 SatLins AF eğrisi.

$$\text{TriBas} = \begin{cases} 1 - |n|, & \text{eğer } -1 \leq n \leq 1 \\ 0, & \text{diğer durumlarda} \end{cases} \quad (3.8)$$



Şekil 3.12 TriBas AF eğrisi.

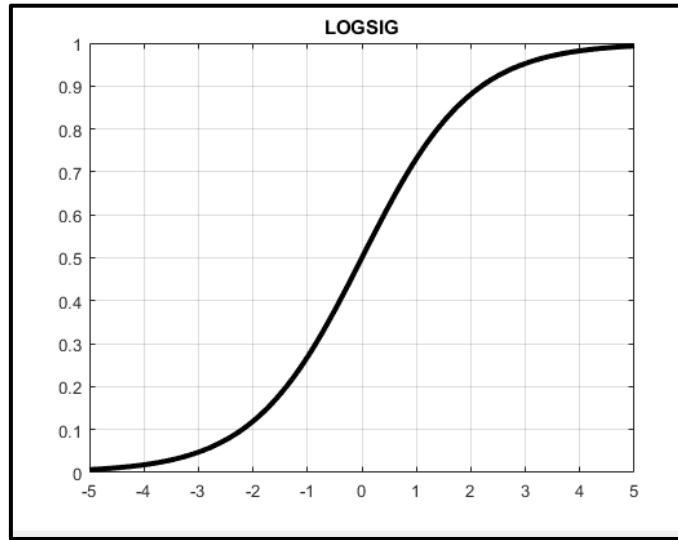
3.2.2.2 Doğrusal Olmayan Aktivasyon Fonksiyonları

Karmaşık problem çözümlerinde doğrusal AF'nin kullanımı yeterli olmamaktadır. Bu problem çözümlerinde doğrusal olmayan AF yaygınlıkla kullanılarak başarılı sonuçlar alınmaktadır. Literatürdeki YSA çalışmalarında birçok farklı doğrusal olmayan AF kullanılmaktadır. Bunların içerisinde en çok sigmoid fonksiyonları kullanılmaktadır. Sigmoid fonksiyonlarının önemli bir niteliği türevinin alınabilir olmasıdır. Bu niteliği sayesinde karmaşık problem çözümlerinde sıklıkla tercih edilmektedir. Denklem (3.9)'da verilen sigmoid fonksiyonuna ait denklemde, $\varphi(v)$ aktivasyon fonksiyonu ve a eğim parametresi olup değişken değerlidir. Bu fonksiyonda girişe uygulanan $-\infty$ ile $+\infty$ arasındaki bir değere karşılık $(0, 1)$ aralığında sonuç üretilmektedir (Haykin 2005, Demuth vd. 2014).

$$\varphi(v) = \frac{1}{1 + \exp(-av)} \quad (3.9)$$

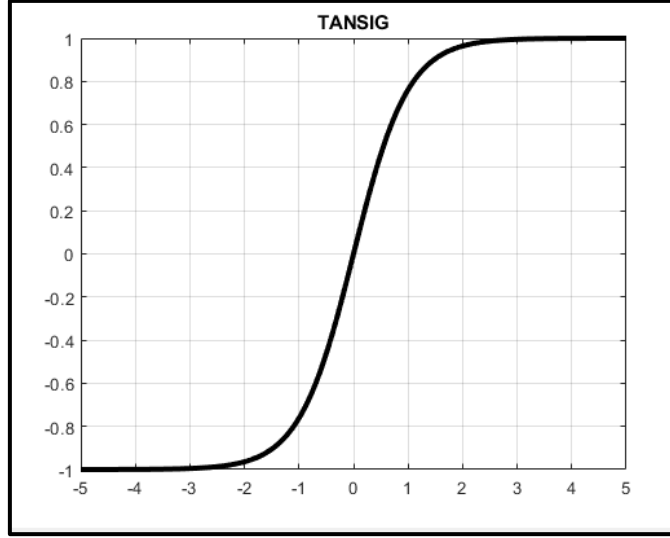
Bu tez çalışmasında AF kütüphanesi oluşturulurken sigmoid AF'nin iki farklı formu olan LogSig ve TanSig fonksiyonları kullanılmıştır. Ayrıca bu çalışmada karmaşık problem çözümlerinde kullanılan bir diğer doğrusal olmayan RadBas fonksiyonu da kullanılmıştır. İlgili doğrusal olmayan AF'lere ait denklemler ve eğriler aşağıda sırası ile verilmiştir. Denklemlerde kullanılan x değişkeni bir rasyonel sayı olmak üzere fonksiyonların giriş değerini ifade etmektedir.

$$\text{LogSig} = \frac{1}{1 + e^{-x}} \quad (3.10)$$



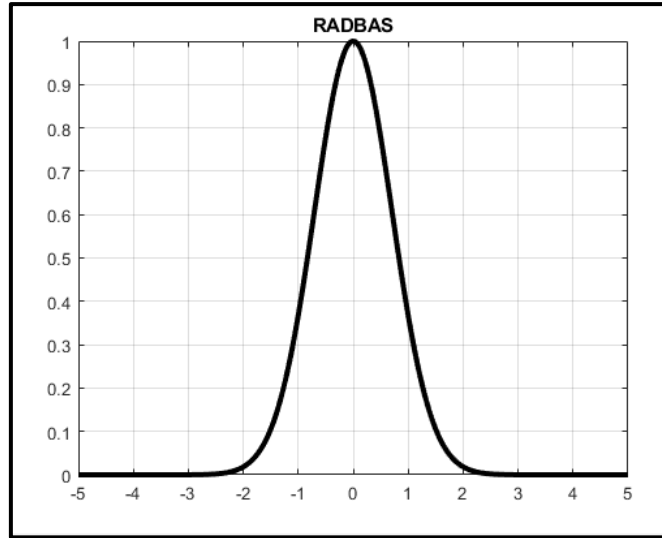
Şekil 3.13 LogSig AF eğrisi.

$$\text{TanSig} = \frac{2}{1 + e^{-2x}} - 1 \quad (3.11)$$



Şekil 3.14 TanSig AF eğrisi.

$$\text{RadBas} = e^{-(x^2)} \quad (3.12)$$



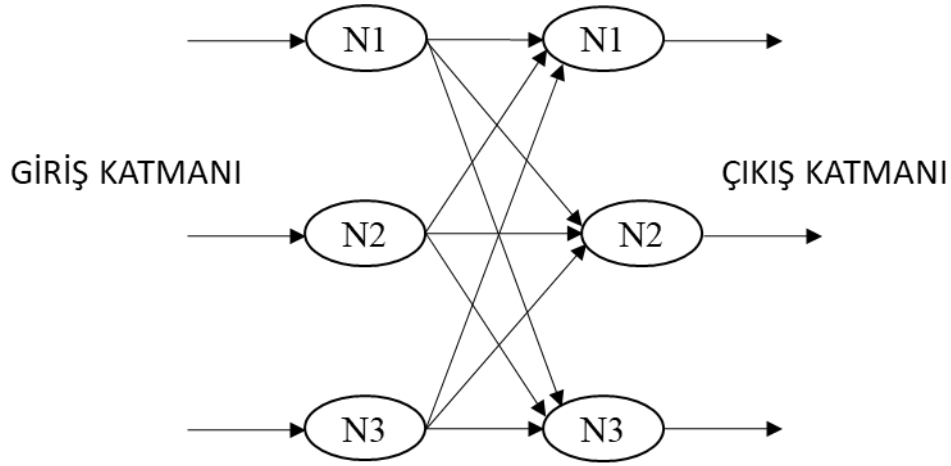
Şekil 3.15 RadBas AF eğrisi.

3.2.3 YSA Yapıları

Bir YSA'da ağıın eğitilmesinde gerekli öğrenme algoritması ve nöronların yapılandırılma şekli arasında çok yakın bir ilişki mevcuttur. Bu yüzden YSA tasarımlarında kullanılan öğrenme algoritmalarının yapılaşdırılması kolaylık sağlamaktadır. YSA yapıları genel olarak üç temel sınıfta incelenebilmektedir (Alçın 2017).

3.2.3.1 Tek Katmanlı İleri Beslemeli YSA

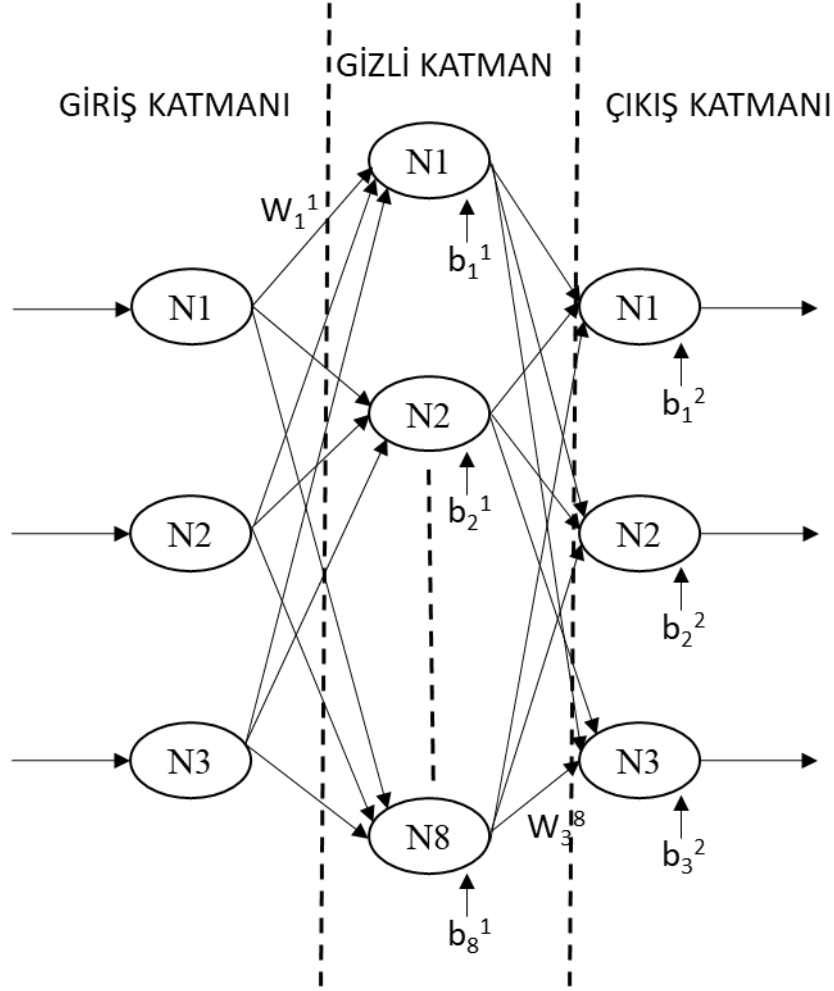
Katmanlı YSA tasarımlarında nöronlar katmanlara dağıtılmıştır. En sade ifade ile katmanlı bir YSA’da, nöronlar çıkış katmanına doğru yansıtılmakta bir giriş ve bir çıkış katmanına dağıtılmaktadır. Şekil 3.16’da giriş ve çıkış katmanlarında üçer nöron bulunan ileri beslemeli ağ yapısı gösterilmektedir (Alçın 2017).



Şekil 3.16 Tek-katmanlı ileri beslemeli YSA yapısı.

3.2.3.2 Çok Katmanlı İleri Beslemeli YSA

Çok katmanlı ileri beslemeli YSA yapısı tek katmanlı yapıdan gizli katman ya da katmanların olması bakımından bariz şekilde ayrılmaktadır. Bu YSA yapısında giriş katmanından gelen dış dünya girdileri gizli katman ya da katmanlarda işlenerek çıkış katmanına iletilmektedir. Çıkış katmanında verilerin işlenmesi sonucunda net çıktı elde edilmektedir. Gizli katman ve gizli nöron sayıları belirlenen probleme göre farklılık göstermektedir. Nöron sayısının artırılması hesaplama karmaşıklığı ve sonuç üretme süresinin uzaması gibi dezavantajları meydana getirmektedir. Fakat YSA tasarımının genelleme özelliği kazanması bakımından önemli bir unsurdur (Ersoy ve Karal 2012). Bu bağlamda problem için ideal yapının belirlenmesinde gizli katman ve nöron sayılarında değişiklik yapılarak aralarından en uygunu seçilmektedir. Bu tez çalışmasında Şekil 3.17’de verilen giriş ve çıkış katmanında 3 nöron, gizli katmanda ise 8 adet nörona sahip çok katmanlı ileri beslemeli YSA yapısı kullanılmıştır.



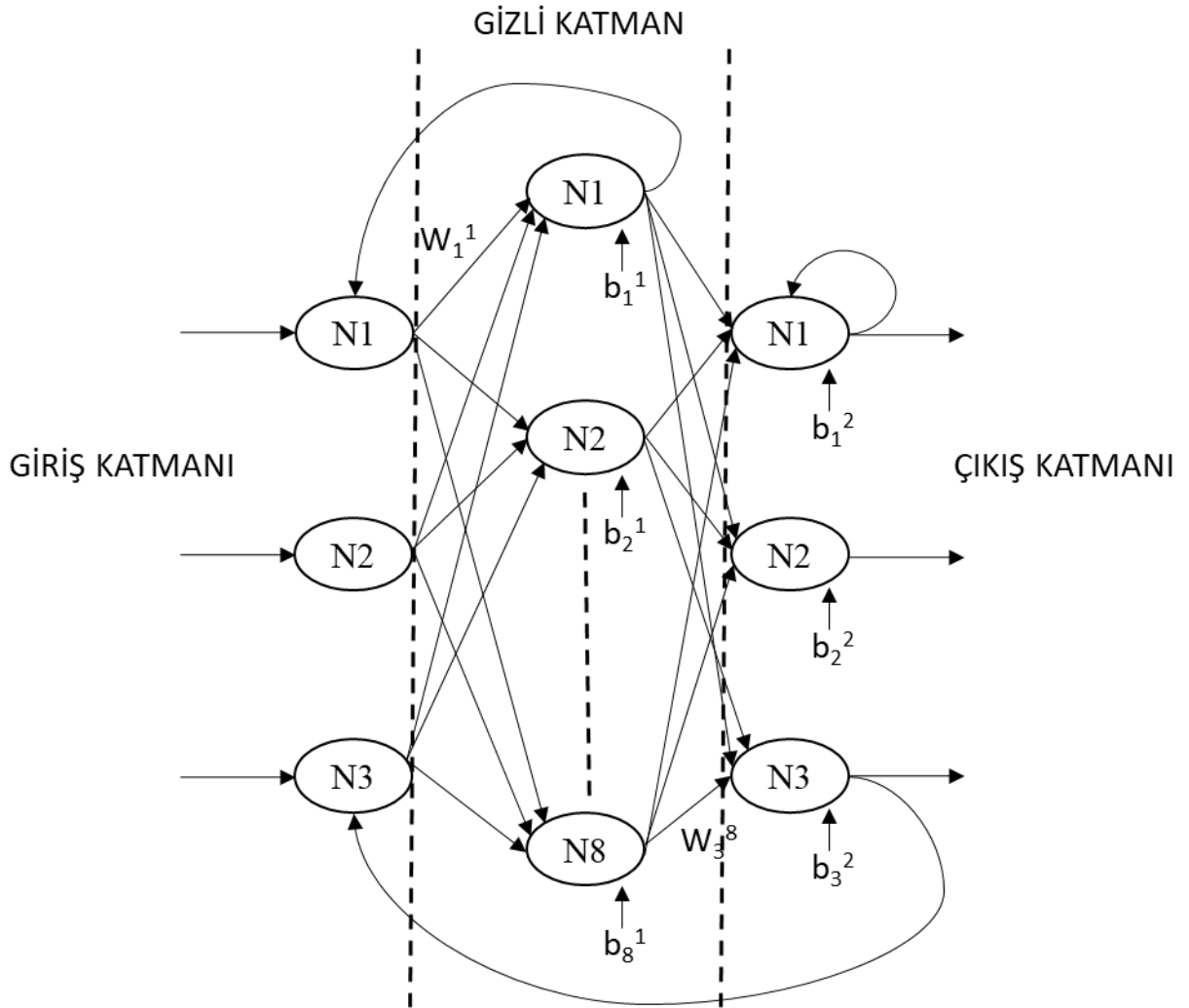
Şekil 3.17 Çok katmanlı ileri beslemeli YSA yapısı.

Şekil 3.17’de N1, N2, ,N8 ilgili katmandaki nöronları, W_1^1, \dots, W_3^8 ilgili katmandaki ağırlıkları ve b_1^1, \dots, b_3^2 ilgili katmandaki eşik değerlerini ifade etmektedir. Ağırlık ifadelerinde üst indis ilgili nöronu, alt indis ise nöronun kaçınıcı ağırlığı olduğunu temsil etmektedir. Eşik değer ifadelerinde üst indis ilgili katmanı alt indis ise kaçınıcı nörona ait olduğunu temsil etmektedir. YSA yapısında gizli katmandaki nöronlarda LogSig, çıkış katmanında bulunan nöronlarda ise PureLin aktivasyon fonksiyonu kullanılmıştır.

3.2.3.3 Geri Beslemeli YSA

Geri beslemeli YSA yapısı, ileri beslemeli YSA yapısından geri besleme döngü ya da döngülerini bulundurması bakımından ayrılmaktadır. Bu YSA yapılarında nöron çıkışları

ağ içerisinde girdi olarak verilerek ağa geri dönmesi sağlanmaktadır. Geri beslemeli ağlarda ileri (çıkış katmanına) doğru bir iletimin yanı sıra nöronların kendileri ya da kendilerinden önceki katmanda bulunan nöronlarla da geri yönde bağlantısı bulunmaktadır (Alçın 2017). Şekil 3.18’de geri beslemeli YSA yapısı gösterilmiştir.



Şekil 3.18 Geri beslemeli YSA yapısı.

3.3 Alanda Programlanabilir Kapı Dizileri

FPGA çipleri tasarımcının belirlediği mantıksal fonksiyonu donanımsal olarak gerçekleştirmek için üzerinde tekrar tekrar programlama işlemi yapılabilen sayısal entegre devrelerdir. Genel ifade ile FPGA’lar giriş-çıkış pinleri, mantık blokları ve ara bağlantılardan oluşmaktadır. Bu üç bileşen tasarımda ihtiyaç duyulan fonksiyona bağlı

olarak programlanmaktadır. FPGA platformları tekrar tekrar programlanma, paralel işlem yeteneği ve yüksek çalışma frekanslarından dolayı günümüzde artmakta olan bir popülerliğe sahiptir. Programlanabilir mantık cihazları FPGA'ların gelişim sürecinde bir temel niteliğindedir (Şeker 2019).

3.4 Programlanabilir Mantık Cihazları

PLD (Programmable Logic Device-Programlanabilir Mantık Cihazları) cihazları temelinde mantık kapıları (AND, OR, NOT) ve flip-floplardan oluşmaktadır. Bu cihazlar üretim sürecinde bir işlev ya da göreve tanımlı olarak tasarlanmazlar. Bu nedenle elektronik devre tasarımlarında kullanıldığında kesinlikle belirlenen görev ya da işleve göre yapılandırılmaya ihtiyaç duyulmaktadır (Şeker 2019).

PLD cihazları üç bölümde incelenebilmektedir.

- SPLD (Simple PLD-Basit PLD'ler)
- CPLD (Complex PLD-Karmaşık PLD'ler)
- FPGA (Field Programmable Gate Array-Alanda Programlanabilir Kapı Dizileri)

3.4.1 Basit Programlanabilir Mantık Cihazları

SPLD az sayıda mantık hücrelerine sahip PLD çeşitidir. Genel olarak bir SPLD'de 4-22 arasında programlanabilen hücre bulunmaktadır. SPLD cihazlarını başlıca üç kısımda incelemek mümkündür (Çetin 2014).

- PROM (Programmable ROM-Programlanabilen ROM)

ROM (Read Only Memory-Salt Okunur Bellek) cihazları ilk programlanabilir salt okunur belleklerdir. Programlanabilen ROM cihazları PROM ismi ile tanımlanmaktadır. PROM cihazlarının ROM ile ortak özellikleri olsa da üretim aşamasında programlanma zorunluluğu olmaması ile ROM'lardan ayrılmaktadır. Bunun yanı sıra PROM bellekleri bir defa programlanabilmekte ve ardından defalarca okuma yapılabilir. PROM'ların iki çeşiti olan EPROM (Erasable PROM-Silinebilen PROM) ve EEPROM

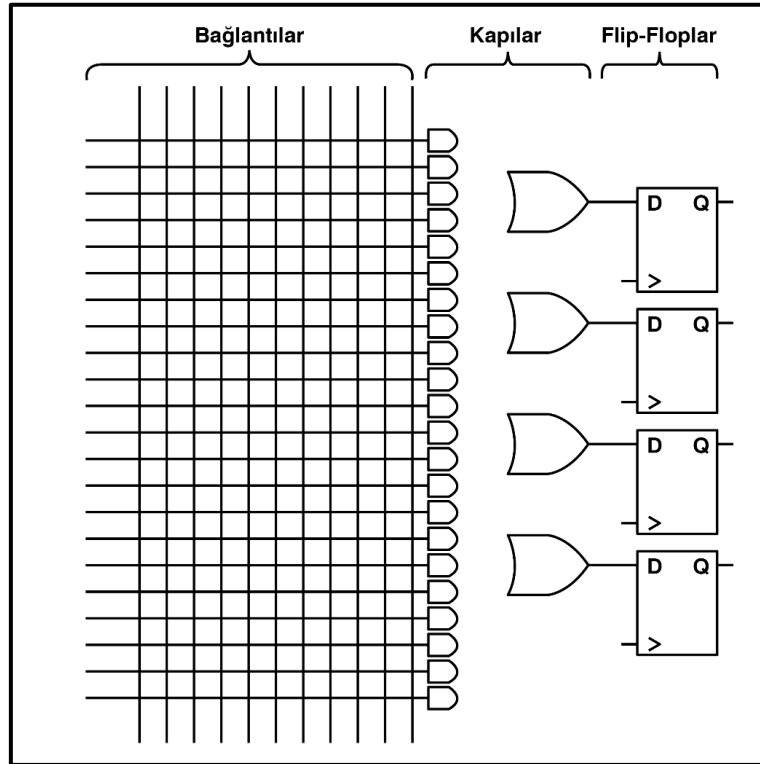
(Electrically Erasable PROM-Elektriksel Silinebilir PROM) ortak özelliklere sahiptir. Aralarındaki fark ise EPROM cihazları elektrik ile programlanıp ultraviyole ışık ile silinmekte iken EEPROM elektrik ile yazılıp-silinmektedir.

- PAL (Programmable Array Logic-Programlanabilen Lojik Diziler)

PAL yapısal olarak programlanabilir AND kapı dizileri ve sabit çıkışlı OR kapılarından meydana gelmektedir (Akpolat 2015).

- GAL (General Array Logic-Genel Dizi Mantığı)

GAL cihazı temel niteliklerde PAL ile benzerlik gösterse de en bariz iki farkı; GAL cihazının tekrar programlanabilir olması ve programlanabilir çıkış devreleri bulundurmasıdır (Şeker 2019). Şekil 3.19’ da SPLD yapısı gösterilmiştir (Çetin 2014).

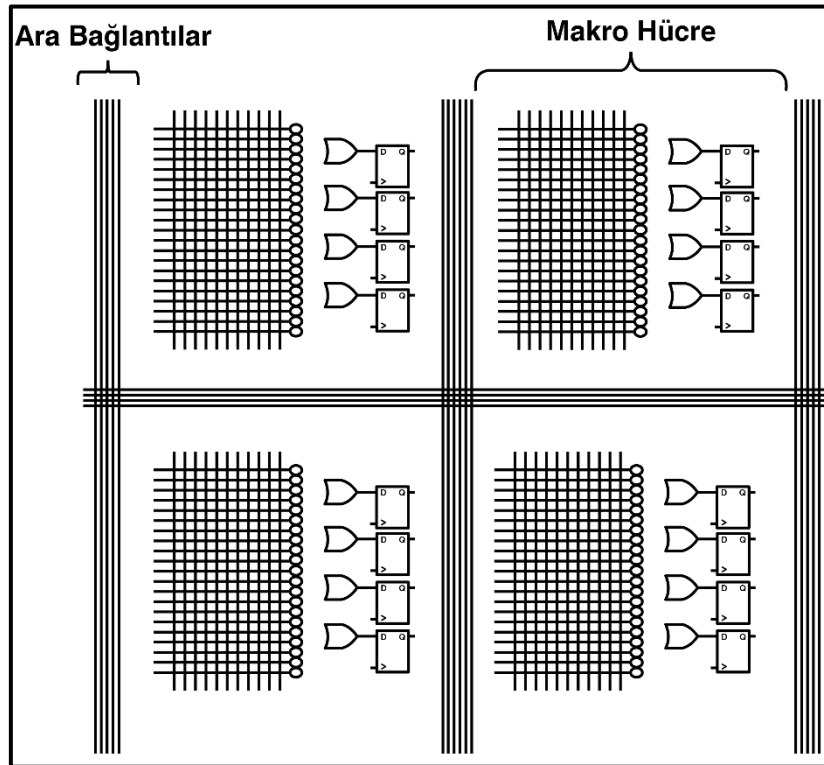


Şekil 3.19 SPLD yapısı.

3.4.2 Karmaşık Programlanabilir Mantık Cihazları

SPLD cihazları karmaşık devre tasarımlarında hem yetersiz kalmakta hem de tasarım güçlüğü bulunmaktadır. Bunlar göz önüne alınarak SPLD blokları birleştirilerek CPLD (Complex Programmable Logic Device-Karmaşık Programlanabilir Mantık Cihazları) yapısı oluşturulmuştur. Ayrıca bu blok yapıları aralarında haberleşme yapabilmektedir. CPLD cihazları dört sınıfta incelenebilmektedir. CPLD yapısı Şekil 3.20’de gösterilmiştir (Çetin 2014, Gürsoy 2016).

- EPLD (Silinebilen Programlanabilen Lojik Cihaz-Erasable PLD)
- PEEL (Programlanabilen Elektriksel Silinebilen Lojik-Programmable Electrically Erasable Logic)
- EEPLD (Elektriksel Silinebilen Programlanabilir Lojik Cihaz-Electrically EPLD)
- Çoklu Dizi Matrisi-Multi Array Matrix



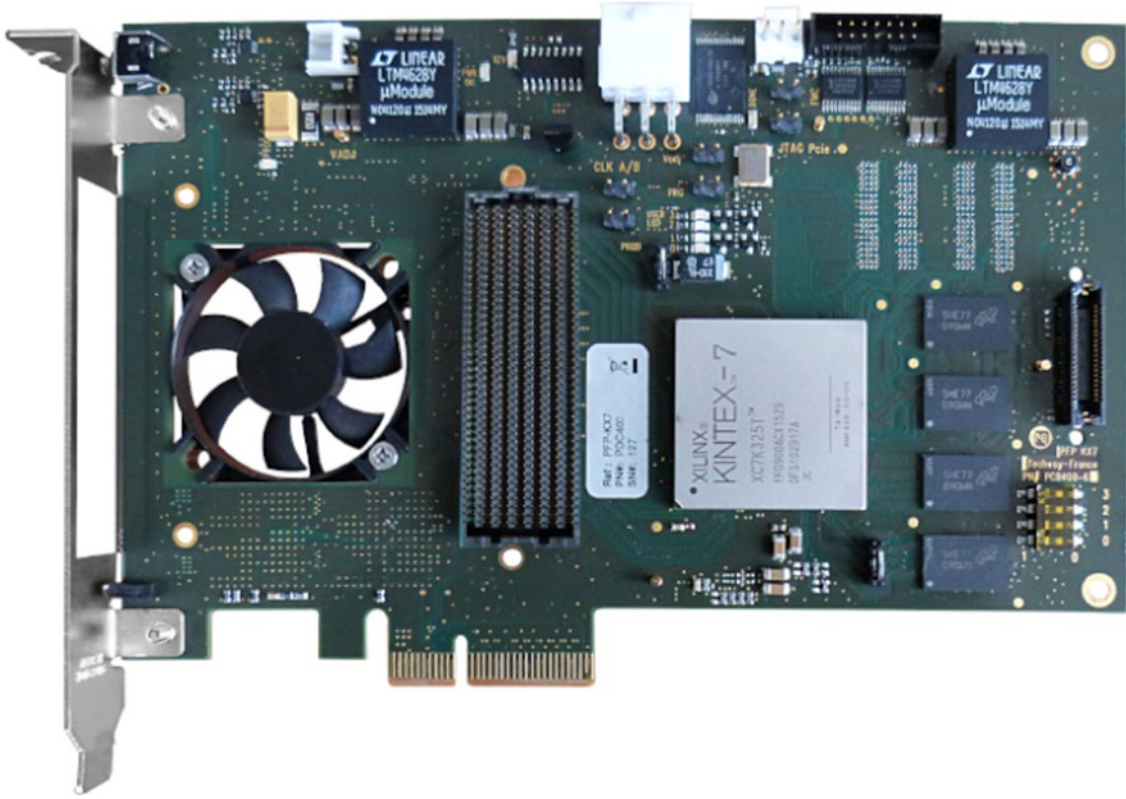
Şekil 3.20 CPLD yapısı.

3.4.3 FPGA ipleri

FPGA ipleri alanda programlanabilir kapı dizileri olarak tanımlanmaktadır. Alanda programlanabilir ifadesinin kullanılmasının nedeni üretim aşamasından sonra programlanabilmesidir. Bu ipler giriş-ıkış blokları, programlanabilen mantıksal bloklar ve mantıksal bloklar arasında bulunan ara bağlantılardan oluşan entegre devrelerdir. Standart entegrelerin içerisinde transistörler arasında bulunan ara bağlantılar sabittir. FPGA’de ise ara bağlantılar programlanabilmektedir. Tasarımcının ihtiyacına göre programlanabilir ara bağlantılar ile mantıksal bloklarda yapılacak işlev belirlenmektedir (etin 2014).

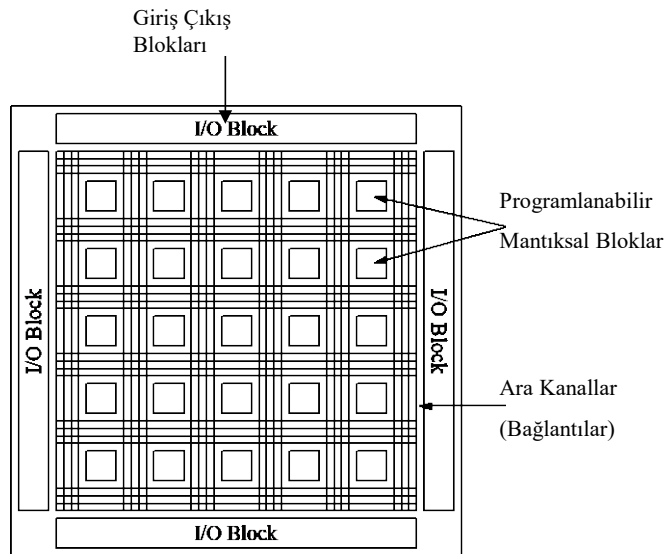
FPGA platformlarının hızlı ilk üretim, tekrar tekrar programlanma, paralel işlem yeteneđi, yüksek alışma frekansları ve düşük güç tüketimi gibi üstünlükleri bulunmaktadır. Bu üstünlükleri ile günümüzde sinyal ve görüntü işleme Pauštaitis ve Dosinas (2009), kaotik osilatör tasarımı Koyuncu vd. (2018), algoritma hızlandırma Şahin (2010), modelleme Tuntaş (2015), yapay sinir ađları Koyuncu vd. (2017), robotik Bargsten ve Fernandez (2020), güvenli iletişim Savran (2017), tıp Abdullah ve Younis (2019), hidrojen üretim sistemleri Yılmaz vd. (2019), modülasyon Mohammed ve Abdullah (2020) ve rasgele sayı üretimi Alın vd. (2019) gibi birçok alışma alanda sıklıkla kullanılmaktadır.

FPGA yongaları üreten bazı firmalara Xilinx, Altera, Actel, Lattice Semiconductor, QuickLogic örnek olarak verilebilir. Bu firmalar içinde Xilinx en köklü ve yaygın tercih edilen firmalardandır. FPGA’ler tasarımcının ihtiyacına göre VHDL, Verilog, Handel-C, System C dillerinde kodlanmaktadır (Koyuncu vd. 2019). Resim 3.1’de Xilinx firmasının Kintex-7 ailesine ait bir FPGA kartı verilmiştir.



Resim 3.1 Xilinx Kintex-7 FPGA kartı.

FPGA'ler giriş ve çıkış blokları, PMB (Programlanabilir Mantıksal Bloklar) ve ara bağlantılar olarak başlıca üç kısımdan oluşmaktadır. Şekil 3.21'de FPGA çiplerinin genel mimarisi gösterilmiştir (Şeker 2019).



Şekil 3.21 FPGA çiplerinin genel mimarisi.

Şekil 3.21 incelendiğinde bir FPGA, ara bağlantılar, PMB blokları ve bu yapıyı kuşatan giriş-çıkış bloklarından oluşmaktadır. FPGA platformları CPLD'lere kıyasen çok daha fazla flip-flop içermektedir. Bu sayede daha karmaşık ve özel sıralı devreler FPGA çipleri ile inşa edilebilmektedir. İki cihazın temel farklarından birisi, ara bağlantıların saklanma şeklindedir. CPLD cihazlarının uçuculuğu bulunmamaktadır. Diğer bir ifade ile güç bağlantısı kesildiğinde program kaybolmaz. Devamlı tazelenmesi gereken DRAM (Dinamic Random Access Memory-Dinamik Rasgele Erişimli Bellek) belleklerinin aksine SRAM (Static Random Access Memory-Durağan Rasgele Erişimli Bellek) belleklerinde güç bağlantısı aktif oldukça bellekteki içerik korunmaktadır. Güç bağlantısı kesildiğinde SRAM'lerin uçuculuk özelliği gereği program kaybolmaktadır. Birçok FPGA çipi SRAM kullandığı için uçuculuk özelliği bulunmaktadır. Fakat SRAM'ler ile FPGA çiplerinde alandan tasarruf sağlanmakta ve çip maliyetinde düşüş olmaktadır. Çünkü FPGA çipleri programlanabilen bağlantılar kullanmaktadır. Ayrıca yeniden programlamaya ihtiyaç duyulmadığı durumlarda avantajlı olarak kullanılabilen uçucu olmayan FPGA platformları da bulunmaktadır (Lortoğlu 2019).

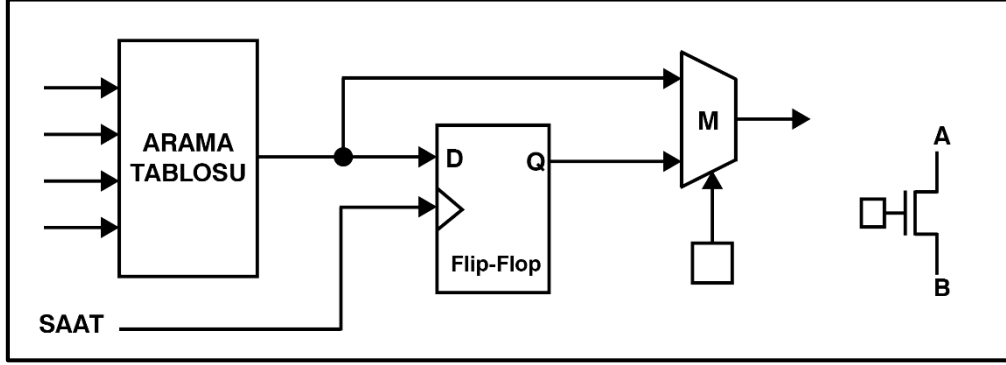
3.4.3.1 Giriş ve Çıkış Blokları

Giriş-Çıkış blokları FPGA çiplerinin programlanabilen pinleridir. Dış dünya ile FPGA platformu arasındaki veri iletişimi bu pinler aracılığı ile yapılmaktadır. Bu pinler programlanabilir olduklarından dolayı tasarımcının ihtiyaç duyduğu durumlara göre değişken olarak giriş, çıkış ya da iki yönlü (giriş-çıkış) olarak programlanmaktadır (Şeker 2019).

3.4.3.2 Programlanabilir Mantıksal Bloklar

PMB (Programlanabilir Mantıksal Bloklar) blokları Flip-Flop'lar, LUT (Look-Up Table) ve Carry Logic'ten meydana gelmektedir. Genellikle bir FPGA platformunun içerisinde PMB blokları ve on binlerce Flip-Flop bulunabilmektedir. LUT sayısına bağlı olarak PMB bloklarının hafıza kapasitesi belirlenmektedir. PMB bloklarında mevcut olan esneklik ve simetri yapının sayesinde uygulamalar kolayca FPGA platformlarına

yerleştirilebilmektedir. Şekil 3.22’de PMB bloklarının içyapısı gösterilmiştir (Lortoğlu 2019).



Şekil 3.22 PMB bloklarının iç yapısı.

3.4.3.3 Ara Bağlantılar

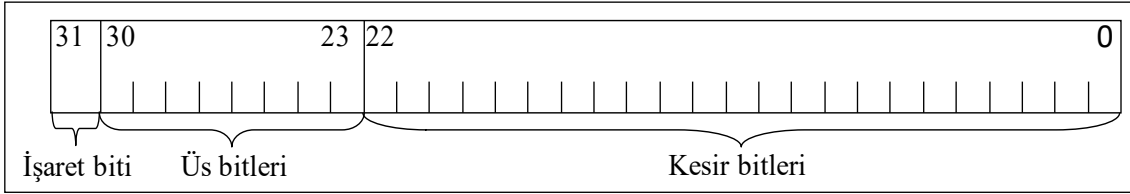
Ara bağlantılar PMB bloklarının hem kendi aralarında hem de giriş-çıkış blokları ile aralarındaki bağlantıların oluşturulmasında kullanılmaktadır. Bu bağlantı kanalları programlanabilir olduklarından dolayı esnek bir yapıdadırlar (Lortoğlu 2019).

3.5 Kayan Noktalı Sayı Standardı

Günlük yaşamda kullanılmakta olan sayılar eksi sonsuz ile artı sonsuz aralığını kapsayan çok büyük tanım aralığında ifade edilebilmektedir. Bu tanım aralığındaki tüm sayı değerlerini ifade edebilmek donanım tabanlı uygulamalarda veya benzetim çalışmalarında mümkün olmamaktadır. Bu nedenle donanım platformlarının kapasitesine göre sonsuza giden değerlerin yerine yaklaşık değerlerin ifade edildiği farklı sayı standartları geliştirilmiştir. Kayan noktalı ve sabit noktalı sayı standartları bu farklı sayı standartlarına örnek verilebilir.

Kayan noktalı sayı standardının dinamiklik ve hassas çözüm üretme gibi avantajları bulunmaktadır. Diğer yandan bu sayı standardı çok fazla donanım kaynağı tüketmektedir. Bu sayı standardının her platformla uyumlu ortak gösterime sahip olması amacı ile IEEE 754-1985 standardı sunulmuştur (Kahan, 1996). Kayan noktalı sayı standardı donanımsal

uygulamalarda birçok farklı formatları ile kullanılmaktadır. Bunlardan bazıları; yarım duyarlı (16 bit), tek duyarlı (32 bit) ve çift duyarlı (64 bit) formatlarıdır. 32 bit tek hassasiyete sahip IEEE 754-1985 kayan noktalı sayı standardının gösterimi Şekil 3.23'te verilmiştir (Şeker 2019).



Şekil 3.23 32 bit tek hassasiyetli IEEE 754-1985 sayı standardının gösterimi.

Şekil 3.23 incelendiğinde 32 bitlik bir sayının 31'inci biti işaret biti olarak ifade edilmektedir. İşaret biti 0 (sıfır) değerinde ise sayı pozitif olmakta, işaret biti 1 değerinde ise sayı negatif olmaktadır. 8 bitlik üs bitlerinde sayının üs değeri tutulmaktadır. 23 bitlik kesir bitlerinde ise sayının kesirli bölümü tutulmaktadır (Şeker 2019). Bir sayının gerçek değerinin kayan noktalı sayı standardına dönüştürülmesinde kullanılan eşitlik denklem (3.13)'te verilmiştir.

$$\text{sayı} = (-1)^{\text{sign}} (1.f) * 2^{(\text{exp}-127)} \quad (3.13)$$

Denklemden f sabiti 0 ile 1 arasında değişmektedir. exp-127 değeri, ilgili sayıdan küçük ya da eşit olan ve 2'nin kuvveti şeklinde yazılabilen en büyük sayıya eşittir. Kayan noktalı sayı standardına dönüşüm işlemini daha iyi kavramak amacı ile aşağıda 50 sayısının 32 bit IEEE 754-1985 kayan noktalı sayı standardına dönüşümü yapılmıştır (Şeker 2019).

$50 > 0$ olduğundan işaret bitini temsil eden sign=0 değerini almaktadır.

$2^5 \leq 50 < 2^6$ olduğundan exp-127=5 değerini almaktadır.

exp=127+5=132 → üs bit değerinin onluk sayı tabanındaki karşılığı bulunmaktadır.

$132 = (10000100)_2$ → üs bit değeri ikilik sayı tabanına çevrilmektedir.

$50 = (-1)^0 \cdot (1.f) \cdot 2^5$ → ilgili sayının kesirli kısmı onluk sayı tabanında hesaplanmaktadır.

$50/32 = 1.f = 1.5625$

f=0.5625

$$0.5625 \times 2 = 1.125 \rightarrow 1$$

$$0.125 \times 2 = 0.25 \rightarrow 0$$

$$0.25 \times 2 = 0.50 \rightarrow 0$$

$$0.50 \times 2 = 1.00 \rightarrow 1$$

$$0 \times 2 = 0 \rightarrow 0$$

⋮ ⋮

Bu yapılan işlemlerin sonucunda 50 sayısına ait 32 bit IEEE 754-1985 kayan noktalı sayı standardı gösterimi Şekil 3.24'te verilmiştir.

| İşaret Biti | Üs Bitleri | Kesir Bitleri |
|-------------|------------|--------------------------|
| 0 | 10000100 | 100100000000000000000000 |

Şekil 3.24 50 sayısına ait 32 bitlik IEEE 754-1985 sayı formatı gösterimi.

İşlem adımlarında, f değeri 2 ile çarpılarak bu sonucun tam sayı kısmı alınmaktadır. Kesirli kısım tekrar 2 ile çarpılarak bu sonucun tam sayı kısmı alınmaktadır. Bu işlem 23 kez tekrarlanır ya da kesirli kısmın sıfır olmasına kadar devam edilmektedir. Çarpma işleminden sıfır sonucu alındığında tüm değerler en ağırlıklı bitten itibaren yazılmaya başlanmaktadır. Elde edilen sonucun yerine yazılmasından sonra boşta kalan bitlere sıfır yazılmaktadır (Çavuşlu vd. 2011).

3.6 Sabit Noktalı Sayı Standardı

IQ-Math sabit noktalı sayı standardını işaret biti (S), tam sayı kısmı (integer-I) ve kesirli kısım (fractional-Q) oluşturmaktadır. Kayan noktalı sayılardan farklı olarak bu sayı standardında tam sayı ve kesirli sayı kısımları tasarımcının ihtiyacı doğrultusunda ayarlanabilmektedir. Sabit noktalı sayı standardı Şekil 3.25'te gösterilmiştir. Bu şekildeki S işaret bitini temsil etmekte ve 0 değerini aldığı anda ilgili sayı pozitif, 1 değerinde ise negatif olmaktadır. İlgili sayının tam sayı kısmındaki bitleri hesaplamak için sayının tam kısmı ikilik sayı tabanına çevrilmektedir (Erick 2007, Özkan vd. 2011).

| İşaret Biti (S) | Tam Sayı Kısmı (I) | Kesirli Kısım (Q) |
|-----------------|-----------------------------|-----------------------------|
| 0 / 1 | $I_n \dots I_3 I_2 I_1 I_0$ | $Q_n \dots Q_3 Q_2 Q_1 Q_0$ |

Şekil 3.25 IQ-Math sabit noktalı sayı formatı gösterimi.

İlgili sayının kesirli kısmının hesaplanması için iki çeşit metot kullanılmaktadır. Birinci yöntem; ilgili sayının kesirli kısmı tasarımcının ayarladığı bit büyüklüğüne kadar iki ile çarpılmaktadır. Sonuç 1 sayısından küçük olduğunda ilgili bit 0 değerini, küçük değil ya da eşit ise ilgili bit 1 değerini almaktadır. Ardından 1 sayısı ile çıkarılarak işlem devam ettirilmektedir (Şeker 2019). Bu yöntem kullanılarak 50.3749 sayısının 16I-16Q sabit noktalı sayı formatı karşılığının hesaplanması aşağıda verilmiştir.

$50.3749 > 0$ olduğundan işaret bitini temsil eden $S=0$ değerini almaktadır.

Tam sayı kısmı=50

Kesirli kısım=0.3749

Tam sayı kısmının ikilik sayı tabanına dönüştürülmesi;

$50=(000000000110010)_2$ olarak bulunmaktadır.

Kesirli kısım olan 0.3749 sayısının ikilik sayı tabanına dönüştürülmesi;

- 1-) $0.3749 * 2 = 0.7498 \rightarrow 0$
- 2-) $0.7498 * 2 = 1.4996 \rightarrow 1 \rightarrow 1.4996 - 1 = 0.4996$
- 3-) $0.4996 * 2 = 0.9992 \rightarrow 0$
- 4-) $0.9992 * 2 = 1.9984 \rightarrow 1 \rightarrow 1.9984 - 1 = 0.9984$
- 5-) $0.9984 * 2 = 1.9968 \rightarrow 1 \rightarrow 1.9968 - 1 = 0.9968$
- 6-) $0.9968 * 2 = 1.9936 \rightarrow 1 \rightarrow 1.9936 - 1 = 0.9936$
- 7-) $0.9936 * 2 = 1.9872 \rightarrow 1 \rightarrow 1.9872 - 1 = 0.9872$
- 8-) $0.9872 * 2 = 1.9744 \rightarrow 1 \rightarrow 1.9744 - 1 = 0.9744$
- 9-) $0.9744 * 2 = 1.9488 \rightarrow 1 \rightarrow 1.9488 - 1 = 0.9488$
- 10-) $0.9488 * 2 = 1.8976 \rightarrow 1 \rightarrow 1.8976 - 1 = 0.8976$
- 11-) $0.8976 * 2 = 1.7952 \rightarrow 1 \rightarrow 1.7952 - 1 = 0.7952$
- 12-) $0.7952 * 2 = 1.5904 \rightarrow 1 \rightarrow 1.5904 - 1 = 0.5904$

- 13-) $0.5904 * 2 = 1.1808 \rightarrow 1 \rightarrow 1.1808 - 1 = 0.1808$
 14-) $0.1808 * 2 = 0.3616 \rightarrow 0$
 15-) $0.3616 * 2 = 0.7232 \rightarrow 0$
 16-) $0.7232 * 2 = 1.4464 \rightarrow 1$ olarak hesaplanmaktadır.

Birinci yöntem kullanılarak 50.3749 sayısının 16I-16Q sabit noktalı sayı formatında gösterimi Şekil 3.26'da verilmiştir.

| İşaret Biti (S) | Tam Sayı Kısmı (I) | Kesirli Kısm (Q) |
|-----------------|--------------------|------------------|
| 0 | 0000000000110010 | 0101111111111001 |

Şekil 3.26 50.3749 sayısının 16I-16Q sabit noktalı sayı formatında gösterimi

İkinci yöntemde ise birinci yöntemde olduğu gibi ilgili sayının tam sayı kısmı ikilik sayı tabanına çevrilmektedir. Kesirli kısmın hesaplanmasında $Q * 2^n$ işlemi yapılmaktadır. İşlemdaki n sayısı ile kesirli kısmın bit sayısı, Q ile kesirli kısmın onluk sayı tabanındaki karşılığı temsil edilmektedir. $Q * 2^n$ işlem sonucunun ikilik sayı tabanına dönüştürülmesi ile ilgili sayının sabit noktalı sayı formatına dönüştürülmesi tamamlanmaktadır (Şeker 2019). Aşağıda dönüşüm işleminin adımları sırası ile yapılarak açıklanmıştır.

$50.3749 > 0$ olduğundan işaret bitini temsil eden $S=0$ değerini almaktadır.

Tam sayı kısmı=50

Kesirli kısım=0.3749

Tam sayı kısmının ikilik sayı tabanına dönüştürülmesi;

$50 = (0000000000110010)_2$ olarak bulunmaktadır.

Kesirli kısım ise;

$0.3749 * 2^{16} = 24569.4464$ olarak bulunmaktadır.

Hesaplanan bu değer tam sayı kısmı alınarak ikilik sayı tabanına dönüştürüldüğünde sabit noktalı sayı standardında kesirli kısmın karşılığı $24.569 = (0101111111111001)_2$

olarak hesaplanmaktadır. Sonuçta bu ikinci yöntem ile 50.3749 sayısının 32 bit (16I-16Q) sabit noktalı sayı standardındaki karşılığı 0000000000110010010111111111001 olarak bulunmaktadır.

3.7 VHDL

VHDL (Very High Speed Integrated Circuit **H**ardware **D**escription **L**anguage-Çok Yüksek Hızlı Tümlşik Devre Donanım Tanımlama Dili) kısaltmasını İngilizce ismindeki baş harflerden almaktadır. Bu kodlama dili FPGA platformlarında sayısal devrelerin tasarımı ve test edilmesi amacı ile kullanılmaktadır. 1981 yılında Amerika Savunma Bakanlığı VHDL dilini geliştirmiştir. Zaman içerisinde VHDL dili üzerinde geliştirmeler yapılarak günümüze kadar gelmiştir. 1986 yılında bu kodlama dilinin tüm geliştirilme hakları IEEE organizasyonuna verilmiştir. IEEE organizasyonu VHDL kodlama dili için IEEE-1987 standardını oluşturmuş ve bu standardın geliştirilmesi ile IEEE 1076-2008 standardı oluşturulmuştur. Bu versiyon, `std_logic_arith`, `std_logic`, `numeric_bit`, `std_logic_vector` gibi birçok veri tipini ve alt program paketini içerisinde bulundurmaktadır. FPGA çiplerinde işlenecek olan kodun sentezlenmesi ve kodun simülasyonunun gerçekleştirilmesi amacı ile VHDL dili bu çiplerde sıklıkla kullanılmaktadır. VHDL dilinden farklı olarak Verilog, Handle-C gibi farklı donanım tanımlama dilleri de kullanılmaktadır (Çetin 2014, Savran 2017).

3.7.1 VHDL Dilinde Veri Nesneleri

VHDL kodlama dilinde kullanılan verilerin anlamlı olarak saklanması ve tutulmasında nesne yapıları kullanılmaktadır. Bu nesne yapılarına sabitler, sinyaller, dosyalar ve değişkenler örnek olarak verilebilmektedir.

Sinyal, devrede bulunan ara bağlantıları temsil eden ve güncel değerleri tutan nesne türüdür. Sinyaller kodlama yapılırken mimari kısımdaki tanım bölümü içerisinde tanımlanmaktadır. Bir sinyal tanımlaması, `SIGNAL sinyal_adi: sinyal_türü := ilk_değer;` formatı kullanılarak yapılmaktadır.

Sabit, tasarımcı tarafından verilen değerin tasarım boyunca değişmediği nesne türüdür. Bu nesne türü kodlama yapılırken mimari kısımdaki tanım bölümü içerisinde bir değer atanarak tanımlanmakta ve atanan değer sabit kalmaktadır. Bir sabit tanımlaması, `COSTANT nesne_adi : türü := sabit_değer; formatı` kullanılarak yapılmaktadır.

Değişken, tasarım sırasında geçici değerlerin saklandığı ve ihtiyaç duyulduğunda bu değerlerin değiştirilmesinin mümkün olduğu nesne türüdür. Bir değişken tanımlaması, `VARIABLE değişken_adi : türü := ilk_değer; formatı` kullanılarak yapılmaktadır (Çetin 2014, Akpolat 2015).

Dosya veri nesnesi, tasarım içerisinde VHDL kodlama dilinde bir veri üzerinde yazma ya da okuma işleminin yapılmasını sağlamaktadır (Kösten ve Çavuşlu 2015).

3.7.2 VHDL Dilinde Temel Tasarım Yapıları

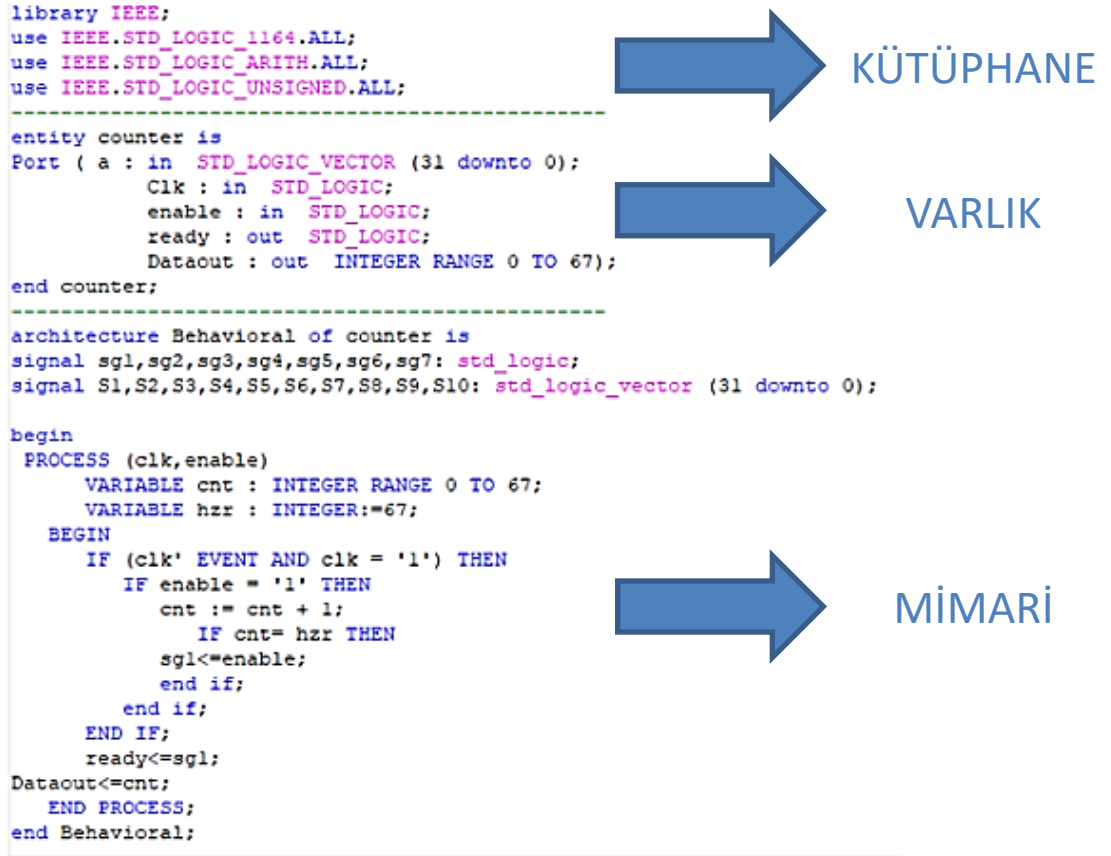
VHDL kodlama dili entity (varlık), library (kütüphane) ve architecture (mimari) olarak başlıca üç yapıdan meydana gelmektedir.

Kütüphane yapısında, tasarımda kullanılması planlanan hazır standart paket kütüphanelerin tanımları yapılmaktadır. Ayrıca burada tasarımcının geliştirdiği kütüphaneler de tanımlanabilmektedir.

Varlık yapısının, içerisinde yapılan tanımlamalar sayesinde sistem dış dünya ile iletişim sağlamaktadır. Diğer bir ifade ile tasarımcının sistem için belirlediği giriş ve çıkış portlarının VHDL dilinde kodlandığı kısımdır. Her bir tasarımda tek bir varlık tanımlaması bulunmakta ve bu tanımlama kütüphane tanımlamasının ardından yapılmaktadır.

Mimari yapısı, varlık kısmındaki giriş ve çıkış portlarının aralarındaki ilişkinin belirtildiği kısımdır. Böylece tasarımın iç yapısı ve davranışı belirlenmektedir. Mimari yapı iki kısımdan meydana gelmektedir. Birinci kısımda, sinyal, sabit gibi veri nesnelere tanımlanmaktadır. İkinci kısımda ise sinyal ve mimari bileşenlerin atamaları ile process

gibi yapılar kullanılarak eş zamanlı işlemler yapılmaktadır (Kösten ve Çavuşlu 2015) (Karataş ve Sarıtaş 2013). Yukarıda bahsedilen kütüphane, varlık ve mimari yapılarının daha iyi anlaşılması için sayıcı devresine ait VHDL kodları Şekil 3.29’da verilmiştir.



Resim 3.2 Sayıcı devresine ait VHDL kodları.

4. BULGULAR

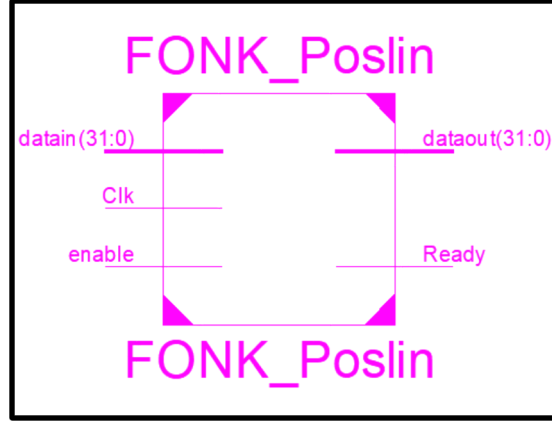
4.1 FPGA Tabanlı Tasarımlar

Bu tez çalışmasında FPGA çiplerinde 32 bit (16I-16Q) IQ-Math sayı standardına uygun olarak doğrusal ve doğrusal olmayan AF VHDL dilinde kodlanmıştır. Böylece bir AF kütüphanesi oluşturulmuş ve YSA tasarımı için gerekli olan AF bu kütüphaneden entegre edilmiştir. Ardından ileri beslemeli örnek bir YSA tasarımı yapılmıştır.

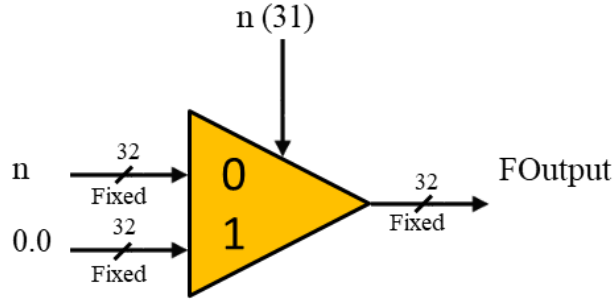
4.1.1 Doğrusal AF'nin FPGA Tabanlı Tasarımları

Bu tez çalışmasında, FPGA çipleri için TriBas, HardLim, HardLims, PosLin, SatLin, SatLins doğrusal AF VHDL dili kullanılarak 32 bit (16I-16Q) IQ-Math sayı standardında kodlanmıştır. Tasarımları test etmek için Xilinx ISE Design Suite 14.7 programı kullanılmıştır. Daha sonra Xilinx firmasının Kintex-7 XC7K70T-3FBG676 FPGA çipinde tüm tasarımlar sentezlenmiş ve Place & Route işlemi yapılarak elde edilen çip istatistikleri sunulmuştur. PosLin fonksiyonuna ait matematiksel eşitlik denklem (4.1)'de verilmiştir. Şekil 4.1'de PosLin fonksiyonuna ait FPGA en üst seviye blok şeması gösterilmiştir. Bu şemada 32 bitlik datain giriş sinyali n girdi sinyalini temsil etmektedir. Clk ve enable sinyalleri 1 bit uzunluğunda olup tasarımda bulunan alt ünitelerin eş zamanlı olarak çalışması için kullanılmıştır. Tasarımdan üretilen sonuçlar 32 bitlik dataout sinyalinden alınmaktadır. Ready sinyali 1 bit uzunluğunda tasarımda kontrol amaçlı kullanılmıştır. Bu sinyal dataout çıkış sinyalinde sonuçlar üretilmeye başlanmasına kadar lojik 0, sonuçlar üretilirken lojik 1 üretmektedir. Bu yapısı ile en üst seviye FPGA blok şeması diğer 5 farklı doğrusal AF için de aynıdır. PosLin fonksiyonun 32 bit IQ-Math sayı standardı ile tasarımına ait blok şeması Şekil 4.2'de gösterilmiştir. Blok şemasında n bir rasyonel sayı olmak üzere 32 bit n giriş sinyalinin sıfıra eşit ve büyük olması durumunda n giriş sinyali, sıfırdan küçük olduğu durumlarda ise çıkış sinyali olarak 32 bit uzunluğunda 0 sayısı alınmaktadır. Bu koşulları sağlamak için çoğullayıcı (multiplexer) ünitesi kullanılmıştır. Bu üniteye n girdi sinyalinin işaret biti seçici sinyal olarak kullanılmasının sebebi işaret bitinin lojik 0 olması durumunda pozitif, lojik 1 olmasında ise negatif sayıyı ifade etmesidir.

$$\text{PosLin} = \begin{cases} n, & \text{eğer } n \geq 0 \\ 0, & \text{eğer } n \leq 0 \end{cases} \quad (4.1)$$



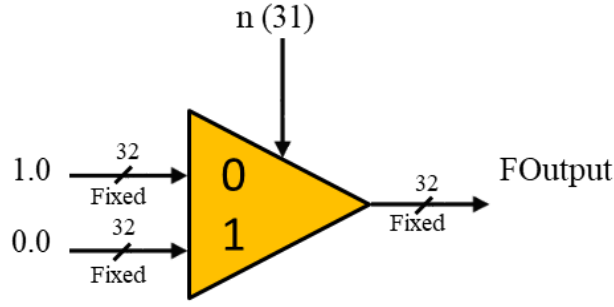
Şekil 4.1 PosLin fonksiyonuna ait en üst seviye FPGA blok şeması.



Şekil 4.2 PosLin fonksiyonuna ait IQ-Math sayı tabanlı blok şeması.

HardLim fonksiyonuna ait matematiksel eşitlik denklem (4.2)'de verilmiştir. HardLim fonksiyonun 32 bit IQ-Math sayı standardı kullanılarak tasarlanmasına ait blok şeması Şekil 4.3'te gösterilmiştir. Bu şemada n girdi sinyalinin sıfıra eşit ve büyük olması durumunda 32 bit uzunluğunda 1 sayısı, diğer durumlarda çıkışta 32 bit uzunluğunda 0 sayısı üretilmektedir.

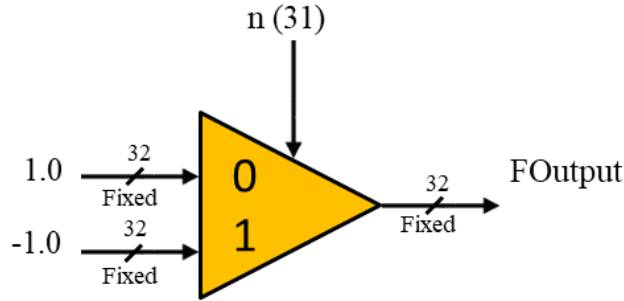
$$\text{HardLim} = \begin{cases} 1, & \text{eğer } n \geq 0 \\ 0, & \text{diğer durumlarda} \end{cases} \quad (4.2)$$



Şekil 4.3 HardLim fonksiyonuna ait IQ-Math sayı tabanlı blok şeması.

Denklem (4.3)'te HardLims fonksiyonuna ait matematiksel eşitlik verilmiştir. Şekil 4.4'te HardLims fonksiyonun 32 bit IQ-Math sayı standardına uygun olarak tasarlanmasına ait blok şeması görülmektedir. Bu şemada n girdi sinyali sıfıra eşit ve büyük olduğu sürece çıkıştan 32 bit uzunluğunda 1 sayısı diğer durumlarda ise -1 sayısı üretilmektedir.

$$\text{HardLims} = \begin{cases} 1, & \text{eğer } n \geq 0 \\ -1, & \text{diğer durumlarda} \end{cases} \quad (4.3)$$

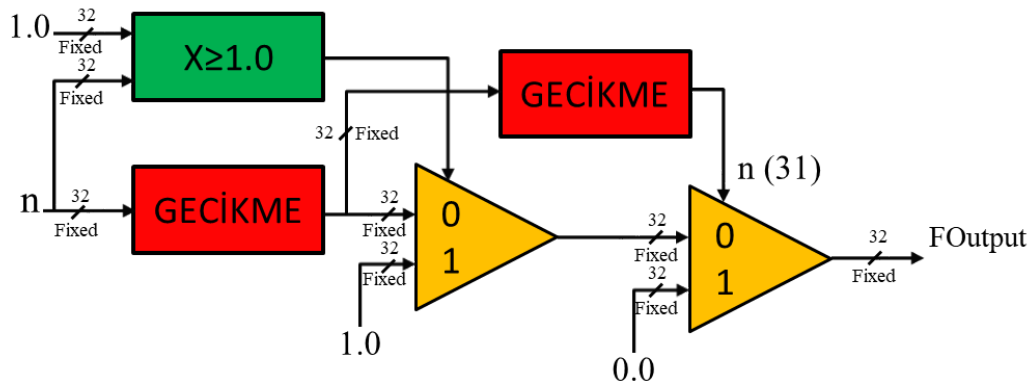


Şekil 4.4 HardLims fonksiyonuna ait IQ-Math sayı tabanlı blok şeması.

Denklem (4.4)'te SatLin fonksiyonuna ait matematiksel eşitlik verilmiştir. SatLin fonksiyonuna ait 32 bit IQ-Math sayı tabanlı blok şeması Şekil 4.5'te gösterilmiştir. Bu şemada girdi sinyalinin 1'den büyük veya eşit olma durumu karşılaştırıcı ünitesinde değerlendirilmiştir. Bu şemada, FPGA çiplerinin paralel çalışma yeteneklerinden faydalanılarak tasarımın paralel çalışmasını sağlamak için 2 adet gecikme ünitesi kullanılmıştır. Birinci gecikme ünitesinde giriş sinyali karşılaştırıcı ünitesindeki işlem süresi kadar geciktirilmiştir. Böylece karşılaştırıcı ünitesinin çıkış sinyali ile n girdi sinyali eş zamanlı olarak çoğullayıcı ünitesine giriş yapılmaları sağlanmıştır. İkinci

gecikme ünitesinde ise geciktirilmiş n girdi sinyali çoğullayıcı ünitesindeki işlem süresi kadar geciktirilmiştir. Böylece ikinci çoğullayıcı ünitesine n girdi sinyalinin işaret biti ile birinci çoğullayıcı ünitesinin çıkış sinyali eş zamanlı olarak giriş yapmışlardır. Tasarımda n girdi sinyalinin sıfır ve sıfırdan küçük olduğu durumlarda 0 sayısı, $[0, 1]$ aralığında olması durumunda n girdi sinyali, 1 ve 1 den büyük olduğu durumlarda 1 sayısı üretilmektedir.

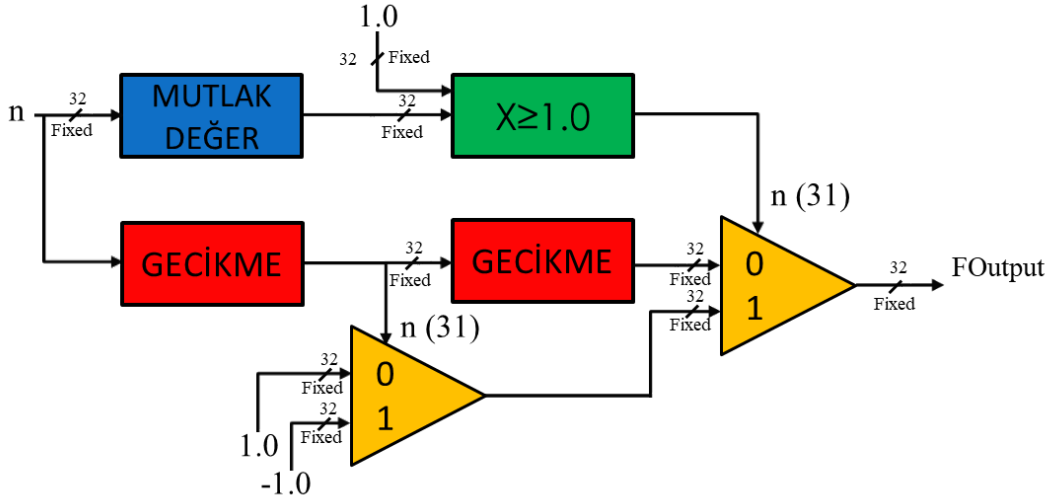
$$\text{SatLin} = \begin{cases} 0, & \text{eğer } n \leq 0 \\ n, & \text{eğer } 0 \leq n \leq 1 \\ 1, & \text{eğer } n \geq 1 \end{cases} \quad (4.4)$$



Şekil 4.5 SatLin fonksiyonuna ait IQ-Math sayı tabanlı blok şeması.

SatLins fonksiyonuna ait matematiksel eşitlik denklem (4.5)'te verilmiştir. SatLins fonksiyonuna ait 32 bit IQ-Math sayı tabanlı blok şeması Şekil 4.6'da görülmektedir. Bu şemada birinci gecikme ünitesinde n girdi sinyali mutlak değer ünitesindeki işlem süresi kadar geciktirilmiştir. Bu sinyalin işaret biti çoğullayıcı ünitesi için seçici sinyal olarak değerlendirilmiştir. Geciktirilmiş n girdi sinyali ikinci gecikme ünitesinde karşılaştırıcı ünitesindeki işlem süresi kadar geciktirilmiştir. Böylece yapılan gecikme işlemleri ile karşılaştırıcı ünitesinin çıkış sinyali ile n girdi sinyali eş zamanlı olarak ikinci çoğullayıcı ünitesine giriş yapmıştır.

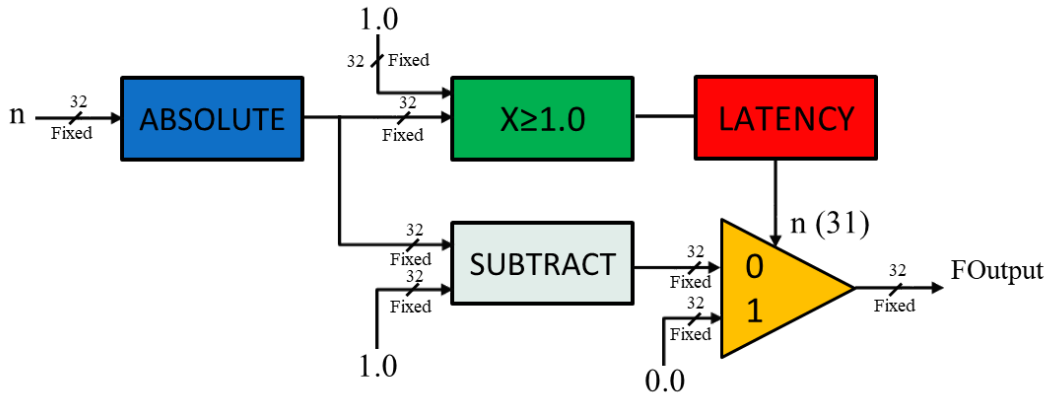
$$\text{SatLins} = \begin{cases} -1, & \text{eğer } n \leq -1 \\ n, & \text{eğer } -1 \leq n \leq 1 \\ 1, & \text{eğer } n \geq 1 \end{cases} \quad (4.5)$$



Şekil 4.6 SatLins fonksiyonuna ait IQ-Math sayı tabanlı blok şeması.

TriBas fonksiyonuna ait matematiksel eşitlik denklem (4.6)'da verilmiştir. TriBas fonksiyonuna ait IQ-Math sayı tabanlı blok şeması Şekil 4.7'de verilmiştir. Bu şemada çıkarıcı ünitesinde yapılan işlem 5 saat darbesi sonucunda tamamlanmaktadır. Bundan dolayı karşılaştırıcı ünitesinden çıkan sinyal 4 saat darbesi geciktirilmiştir. Bunun sayesinde çıkarıcı ve karşılaştırıcı ünitelerine ait çıkış sinyalleri eş zamanlı olarak çoğullayıcı ünitesine iletilmiştir. Tasarımda n girdi sinyali $[-1, 1]$ aralığında olması durumunda çıkıştan 1 sayısından n sinyalinin mutlak değeri çıkarılması ile elde edilen sonuç diğer durumlarda ise çıkıştan 32 bit uzunluğunda 0 sayısı alınmıştır.

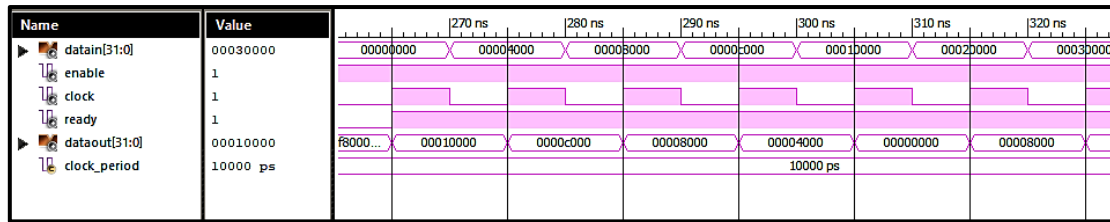
$$\text{TriBas} = \begin{cases} 1 - |n|, & \text{eğer } -1 \leq n \leq 1 \\ 0, & \text{diğer durumlarda} \end{cases} \quad (4.6)$$



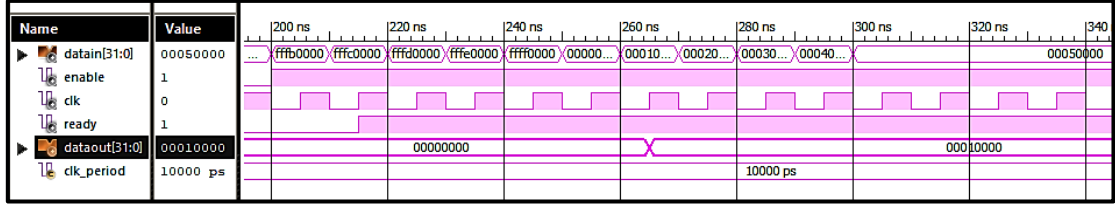
Şekil 4.7 TriBas fonksiyonuna ait IQ-Math sayı tabanlı blok şeması.

4.1.1.1 Doğrusal AF'nin Simülasyon Sonuçları

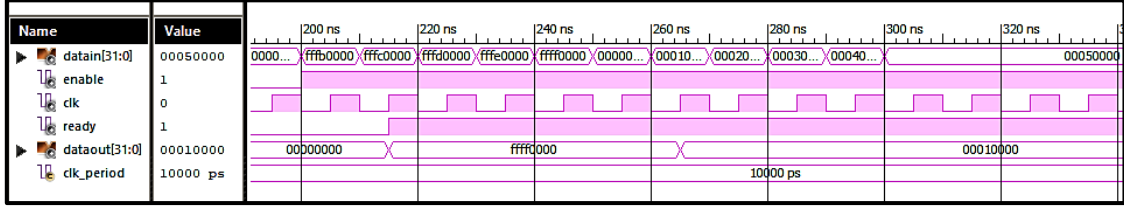
FPGA çipleri kullanılarak yapılan tasarımların test edilmesi için çeşitli programlar kullanılmaktadır. Bu tez çalışmasında, Xilinx ISE Design Suite 14.7 programı kullanılarak doğrusal AF tasarımları $[-5, 5]$ aralığında test edilerek simülasyon gerçekleştirilmiştir. Sırası ile TriBas, HardLim, HardLims, PosLin, SatLin, SatLins fonksiyonlarının Xilinx ISE Design Suite 14.7 programındaki simülasyon sonuçları gösterilmiştir (Şekil 4.8, 4.9, 4.10, 4.11, 4.12, 4.13). Simülasyonda 1 saat darbesi 10 ns olmak üzere PosLin, HardLim ve HardLims 1 saat darbesi, SatLin ve SatLins 3 saat darbesi ve TriBas AF tasarımında 7 saat darbesi kadar süre geçtikten sonra sonuçlar üretilmeye başlanmaktadır. Sonuçlar daha anlaşılabilir olması için hexadecimal (on altılık) sayı tabanında gösterilmiştir.



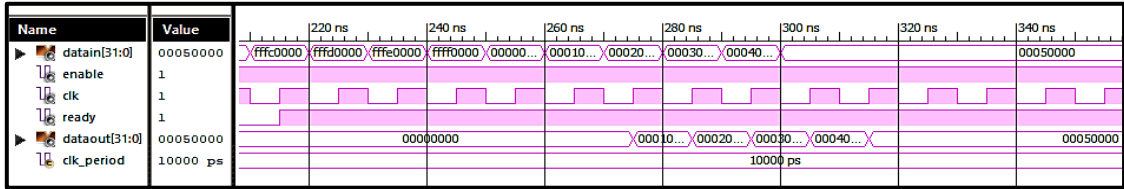
Şekil 4.8 TriBas fonksiyonuna ait Xilinx ISE simülasyon sonuçları.



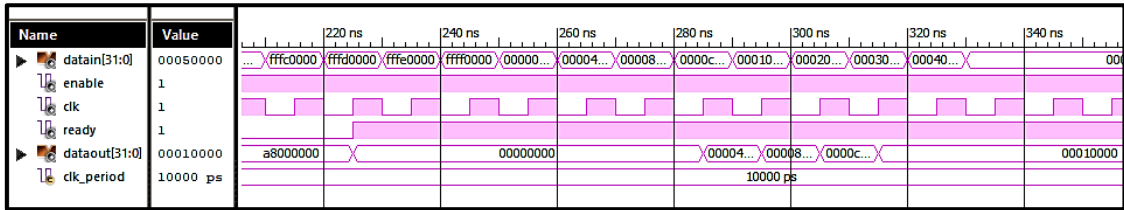
Şekil 4.9 HardLim fonksiyonuna ait Xilinx ISE simülasyon sonuçları.



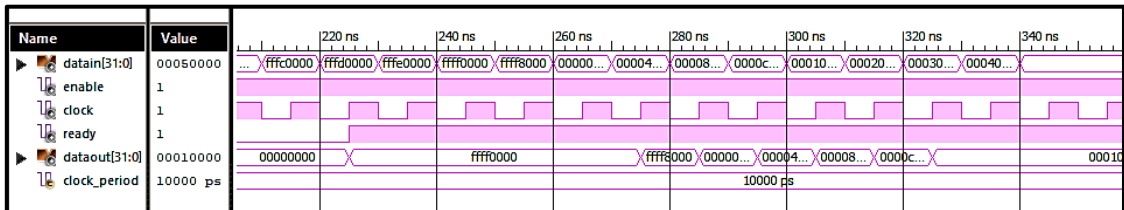
Şekil 4.10 HardLims fonksiyonuna ait Xilinx ISE simülasyon sonuçları.



Şekil 4.11 PosLin fonksiyonuna ait Xilinx ISE simülasyon sonuçları.



Şekil 4.12 SatLin fonksiyonuna ait Xilinx ISE simülasyon sonuçları.



Şekil 4.13 SatLins fonksiyonuna ait Xilinx ISE simülasyon sonuçları.

Doğrusal AF tasarımları Xilinx Kintex-7 ailesinin XC7K70T-3FBG676 FPGA çipi için sentezlenmiş ve Place & Route işleminin ardından elde edilen çip istatistikleri Çizelge 4.1’de sunulmuştur. Çizelge incelendiğinde HardLim ve HardLims fonksiyonları en az

ip kaynađı kullanımına sahiptir. alıřma frekansları incelendiđinde ise SatLin fonksiyonu en yksek deđere sahiptir.

izelge 4.1 Dođrusal aktivasyon fonksiyonların FPGA ip istatistikleri.

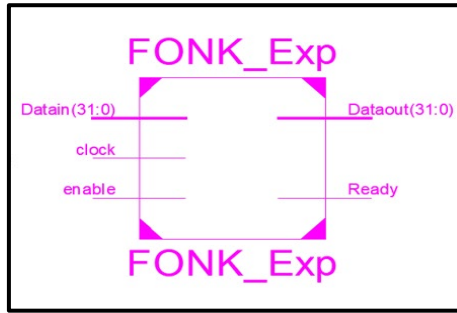
| FPGA ip İstatistikleri | Poslin | Hardlim | Hardlims | Satlin | Satlins | TriBas |
|--------------------------------|---------------|----------------|-----------------|---------------|----------------|---------------|
| Mak. alıřma Frekans (MHz) | 956.023 | 956.023 | 956.023 | 1379.172 | 825.355 | 822.504 |
| Slice Registers Sayısı | 63 | 3 | 3 | 102 | 86 | 151 |
| Slice LUTs Sayısı | 95 | 2 | 2 | 167 | 98 | 180 |
| IOBs Sayısı | 67 | 36 | 36 | 67 | 67 | 67 |
| BUFG/BUFGCTLs Sayısı | 1 | 1 | 1 | 1 | 1 | 1 |

4.1.2 Dođrusal Olmayan AF'nin FPGA Tabanlı Tasarımları

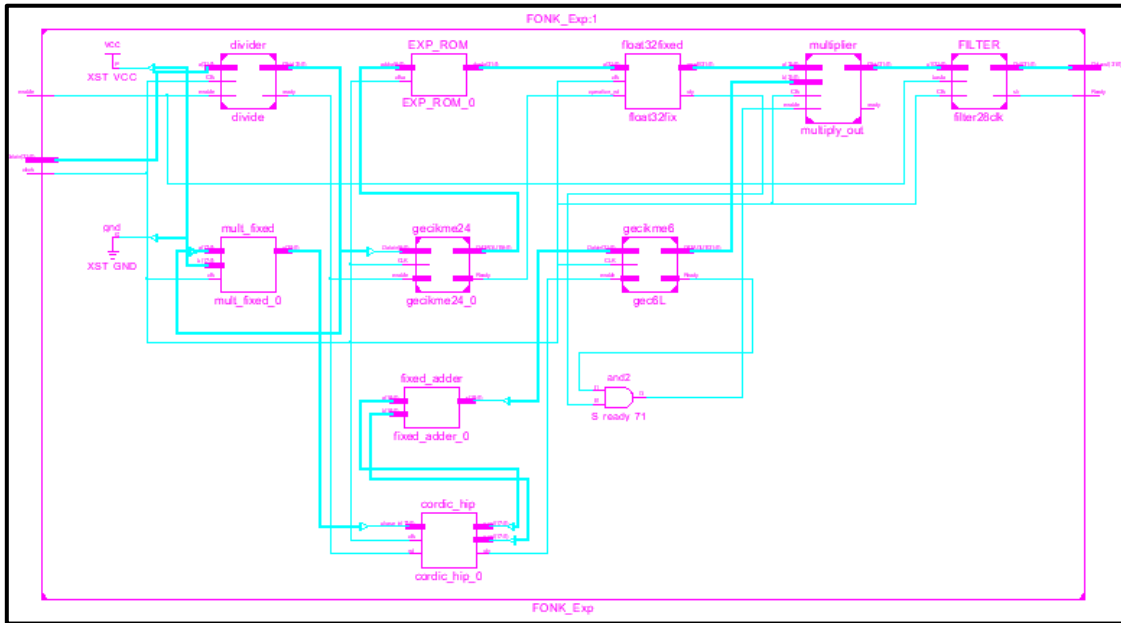
Bu tez alıřmasında FPGA platformunda dođrusal olmayan RadBas, LogSig, TanSig AF 32 bit (16I-16Q) IQ-Math sayı tabanlı olarak VHDL dilinde kodlanmıřtır. Bu AF tasarlanırken ilk nce bu fonksiyonların matematiksel eřitliklerinde ortak olan e^x stel fonksiyonun tasarımı yapılmıřtır. Literatrde e^x stel fonksiyonun FPGA platformunda tasarlanması iin Taylor serisi, CORDIC(COordinate Rotation DIgital Computer), Elliot Sigmoid ve LUT(Look Up Table) gibi. eřitli yaklařımlar kullanılmıřtır (Koyuncu ve řahin 2011). Bu alıřmada stel fonksiyon ve dođrusal olmayan AF'yi FPGA iplerinde uygulamak iin 32 bit (16I-16Q) IQ-Math sayı tabanlı CORDIC – LUT birleřimi bir yaklařım kullanılmıřtır. Denklem (4.7)'de CORDIC tabanlı tasarıma ait matematiksel eřitlik verilmiřtir. řekil 4.14'te stel fonksiyonun 32 bit (16I-16Q) IQ-Math sayı tabanlı blok řeması verilmiřtir. Tasarımda kullanılan LUT, dnřtrc, arpıcı, CORDIC ve toplayıcı niteleri Xilinx ISE Design Tools ile geliřtirilen IP CORE Generator tarafından tasarlanmıřtır.

$$\exp(v) = \sinh(v) + \cosh(v) \quad (4.7)$$

değeri CORDIC ünitesine uygulanarak bu değere göre çıkıştan $\sinh(w)$ ve $\cosh(w)$ değerleri eş zamanlı olarak elde edilmektedir. Ardından toplayıcı ünitesinde $\sinh(w)$ ve $\cosh(w)$ değerleri toplanarak w değerine göre e^w değeri IQ-Math sabit noktalı sayı formatında hesaplanmaktadır. Hesaplanan e^w değeri gecikme ünitesinde 6 saat darbesi geciktirilir. Bu sayede e^w değeri ile $e^{t.0,75}$ değeri eş zamanlı olarak çarpıcı ünitesinde işlenmektedir. Böylece çarpma ünitesinden e^x değeri elde edilmiş olmaktadır (Alçın 2017). Üstel fonksiyona ait en üst seviye FPGA blok şeması 4.15'te, ikinci seviye FPGA blok şeması 4.16'da verilmiştir. Tasarımda filtre ünitesi istenmeyen sinyalleri filtrelemek için kullanılmıştır. En üst seviye blok şeması doğrusal olmayan AF'de de benzer şekildedir.



Şekil 4.15 Üstel fonksiyona ait en üst seviye FPGA blok şeması.



Şekil 4.16 Üstel fonksiyona ait ikinci seviye FPGA blok şeması.

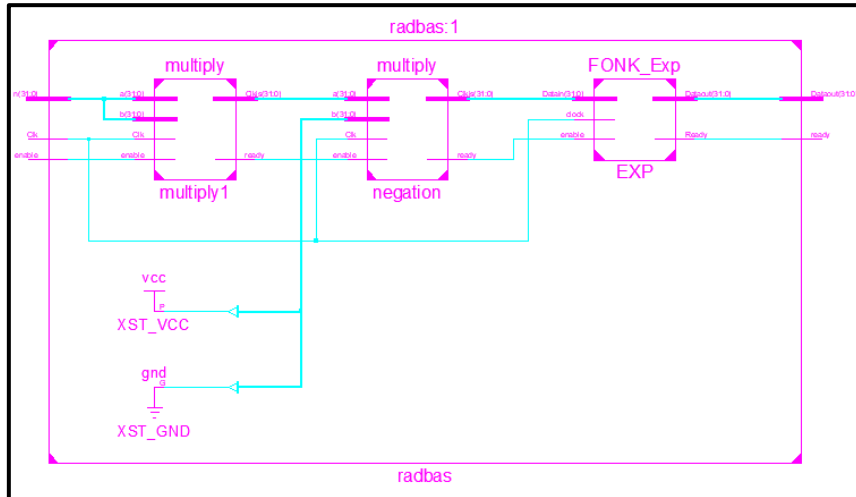
Üstel fonksiyon 3 farklı doğrusal olmayan AF’de de ortak olduğu için üstel fonksiyon ünitesinden bu fonksiyonların tasarımında da faydalanılmıştır. İlk olarak RadBas AF 32 bit (16I-16Q) IQ-Math sabit noktalı sayı standardına uygun olarak VHDL dilinde FPGA çiplerinde tasarlanmıştır. Denklem (4.8)’de RadBas fonksiyonuna ait matematiksel eşitlik verilmiştir.

$$\text{RadBas} = e^{-(n^2)} \quad (4.8)$$

RadBas fonksiyonuna ait IQ-Math tabanlı blok şeması Şekil 4.17’de verilmiştir. Blok şemasında n giriş değerinin karesini alma işlemi çarpıcı ünitesinde yapılarak çıkış değeri negatifleyici ünitesine iletilmektedir. Ardından negatifleyici ünitesinde negatifleme işlemi yapılır ve bu ünitiden çıkan değer e^n ünitesinde işlenerek istenen sonuç üretilmektedir. RadBas fonksiyonuna ait ikinci seviye FPGA blok şeması Şekil 4.18’de verilmiştir.



Şekil 4.17 RadBas fonksiyonuna ait IQ-Math tabanlı blok şeması.

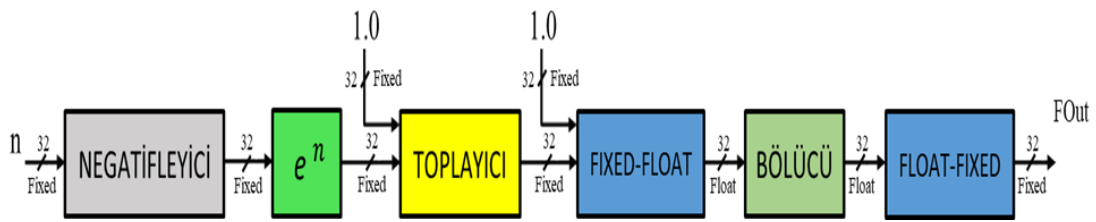


Şekil 4.18 RadBas fonksiyonuna ait ikinci seviye FPGA blok şeması.

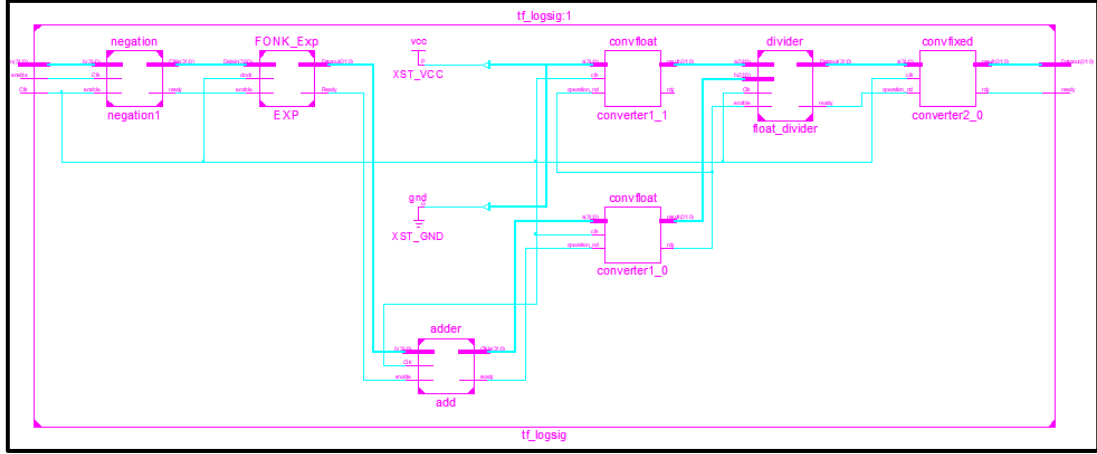
RadBas AF tasarımından sonra LogSig fonksiyonu 32 bit (16I-16Q) IQ-Math sabit noktalı sayı standardına uygun olarak VHDL dilinde FPGA çiplerinde tasarlanmıştır. Denklem (4.9)'da LogSig fonksiyonuna ait matematiksel eşitlik verilmiştir.

$$\text{LogSig} = \frac{1}{1 + e^{-n}} \quad (4.9)$$

Şekil 4.19'da LogSig fonksiyonuna ait 32 bit (16I-16Q) IQ-Math tabanlı blok şeması verilmiştir. Blok şemasında dönüştürücü ve bölücü üniteleri Xilinx Design Tools tasarım araçları tarafından geliştirilen IP CORE Generator kullanılarak tasarlanmıştır. Şemada n girdi sinyali negatifleyici ünitesinde negatifleme işlemine tabi tutulup e^n ünitesine iletilmektedir. e^n ünitesinden çıkan çıkış sinyali toplayıcı ünitesinde 1 sayısı ile toplanır. Ardından dönüştürücü ünitesinde 1 sayısı ve toplayıcı ünitesinin çıkış sinyali IQ-Math sabit noktalı sayı standardından kayan noktalı sayı standardına dönüştürülür. Bölücü ünitesinde bölme işlemi yapılarak sonuç çıkış sinyali aracılığı ile dönüştürücü ünitesine iletilir. Ardından dönüştürücü ünitesinde kayan noktalı sayı standardı IQ-Math sabit noktalı sayı standardına dönüştürülür ve tasarım tamamlanır. Bu tasarımda kayan noktalı sayı standardı ile çalışan bölücü ünitesi kullanıldığı için dönüştürücü ünitelerine ihtiyaç duyulmuştur. LogSig fonksiyonuna ait ikinci seviye FPGA blok şeması Şekil 4.20'de verilmiştir.



Şekil 4.19 LogSig fonksiyonuna ait IQ-Math tabanlı blok şeması.

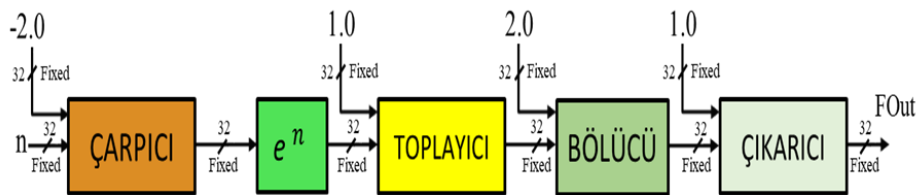


Şekil 4.20 LogSig fonksiyonuna ait ikinci seviye FPGA blok şeması.

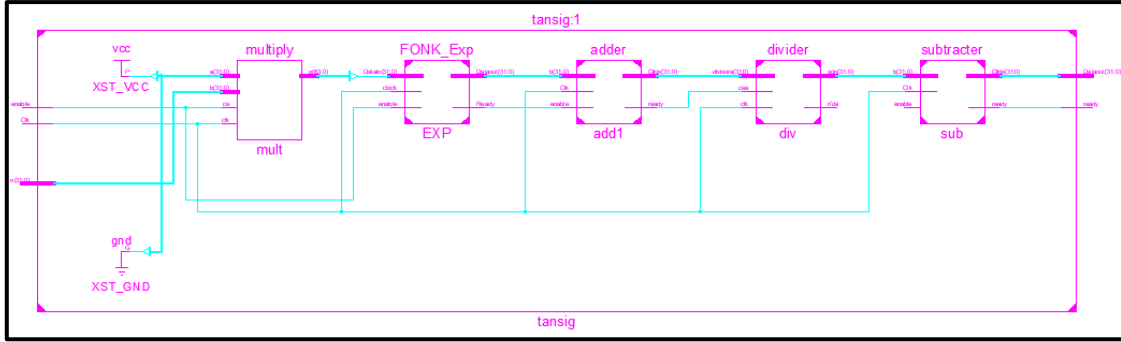
Tansig AF 32 bit (16I-16Q) IQ-Math sabit noktalı sayı standardına uygun olarak VHDL dilinde FPGA çiplerinde tasarlanmıştır. Denklem (4.10)'da TanSig fonksiyonuna ait matematiksel eşitlik verilmiştir.

$$\text{TanSig} = \frac{2}{1 + e^{-2n}} - 1 \quad (4.4)$$

Şekil 4.21'de TanSig fonksiyonuna ait IQ-Math tabanlı blok şeması verilmiştir. Blok şemasında bölücü ve çarpıcı üniteleri Xilinx Design Tools tasarım araçları ile geliştirilen IP CORE Generator kullanılarak tasarlanmıştır. Çarpıcı ünitesinde -2 sayısı ve n girdi bilgisine çarpma işlemi uygulanır ve sonuç sinyali e^n ünitesine iletilir. Toplayıcı ünitesinde e^n ünitesinden çıkan sinyal ile 1 sayısına toplama işlemi uygulanır ve sonuç sinyali bölücü ünitesine iletilir. Bölücü ünitesinde bölme işlemi yapılır ve sonuç sinyali çıkarıcı ünitesine iletilir. Çıkarıcı ünitesinde, bölücü ünitesinin çıkış sinyalinden 1 sayısı çıkarılarak tasarım tamamlanır. Şekil 4.22'de TanSig fonksiyonuna ait ikinci seviye FPGA blok şeması verilmiştir.



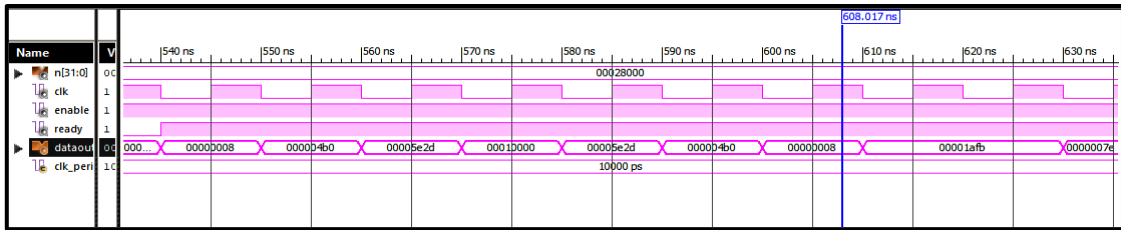
Şekil 4.21 TanSig fonksiyonuna ait IQ-Math tabanlı blok şeması.



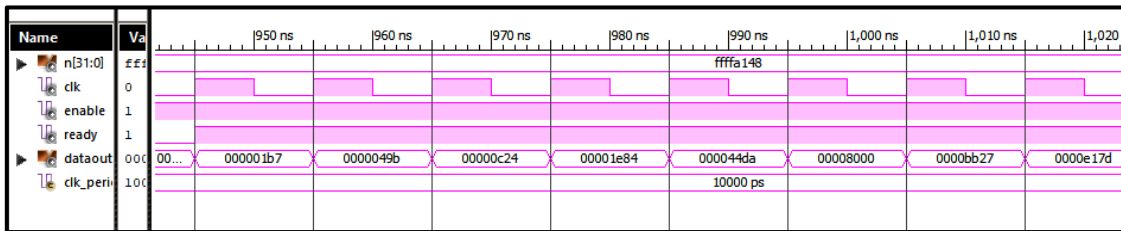
Şekil 4.22 TanSig fonksiyonuna ait ikinci seviye FPGA blok şeması.

4.1.2.1 Doğrusal Olmayan AF'nin Simülasyon Sonuçları

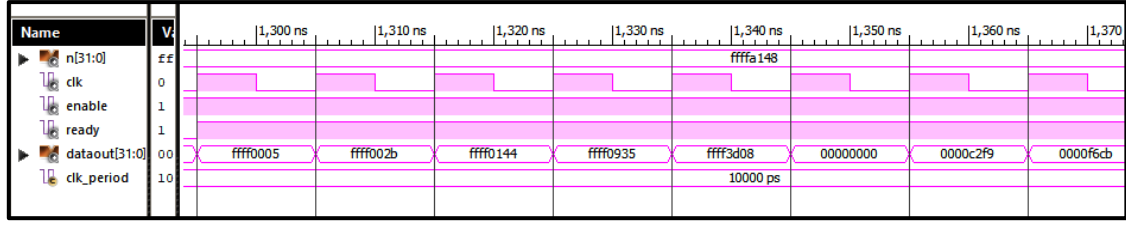
Bu tez çalışmasında RadBas, LogSig ve TanSig doğrusal olmayan AF tasarımları FPGA platformunda IQ-Math sabit noktalı sayı standardında 32 bit (16I-16Q) olarak VHDL dilinde kodlanmıştır. Tasarımları test etmek için Xilinx ISE Design Suite 14.7 programında bir testbench ünitesi oluşturulmuş ve bu ünitenin çalıştırılması ile simülasyon sonuçları elde edilmiştir. RadBas, LogSig ve TanSig fonksiyonlarının Xilinx ISE Design Suite 14.7 programındaki simülasyon sonuçları sırası ile verilmiştir (Şekil 4.23, 4.24, 4.25). Sonuçlar daha anlaşılabilir olması için hexadecimal (on altılık) sayı tabanında gösterilmiştir.



Şekil 4.23 RadBas fonksiyonuna ait Xilinx ISE simülasyon sonuçları.



Şekil 4.24 LogSig fonksiyonuna ait Xilinx ISE simülasyon sonuçları.



Şekil 4.25 TanSig fonksiyonuna ait Xilinx ISE simülasyon sonuçları.

Simülasyon sonuçlarında 1 saat darbesi 10 ns olmak üzere RadBas 33 saat darbesi, LogSig 74 saat darbesi ve TanSig AF tasarımında 109 saat darbesi kadar süre geçtikten sonra sonuçlar üretilmeye başlanmaktadır.

4.1.2.2 Doğrusal Olmayan AF'nin Hata Analizleri

Tasarımların test edilmesinden sonra MSE (Mean Square Error) ortalama karesel hata ve RMSE (Root Mean Square Error) ortalama karesel hatanın karekökü hata analizleri yapılmıştır. Denklem (4.11)'de MSE hata analizine ait matematiksel eşitlik verilmiştir.

$$MSE = \frac{1}{n} \sum_{i=1}^n (\hat{x}_i - x_i)^2 \quad (4.5)$$

Verilen matematiksel eşitlikte \hat{x}_i gerçek değeri, x_i tahmini değer veya ölçülen değeri, n ise hata analizinde kullanılan örnek sayısını temsil etmektedir. RMSE hata analizine ait matematiksel eşitlik denklem (4.12)'de verilmiştir (Şeker 2019).

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (\hat{x}_i - x_i)^2} \quad (4.6)$$

Çizelge 4.2'de doğrusal olmayan AF'nin MSE ve RMSE hata analizleri verilmiştir. Çizelge incelendiğinde doğrusal olmayan üç farklı AF tasarımlarından başarılı sonuçlar alınmıştır.

Çizelge 4.2 Doğrusal olmayan aktivasyon fonksiyonların hata analizleri.

| Aktivasyon Fonksiyonları | MSE | RMSE |
|---------------------------------|------------|-------------|
| RadBas | 3.41E-10 | 1.84E-05 |
| LogSig | 8.64E-11 | 9.30E-06 |
| TanSig | 4.25E-03 | 6.52E-02 |

Doğrusal olmayan üç farklı AF tasarımları simülasyon ve hata analizlerinin ardından Xilinx Kintex-7 ailesinin XC7K70T-3FBG676 FPGA çipi için sentezlenerek çip istatistikleri Çizelge 4.3’te sunulmuştur. Çizelge incelendiğinde RadBas AF diğer 2 farklı doğrusal olmayan AF’ye kıyasla daha az çip kaynağı tüketmiştir.

Çizelge 4.3 Doğrusal olmayan aktivasyon fonksiyonların FPGA çip istatistikleri.

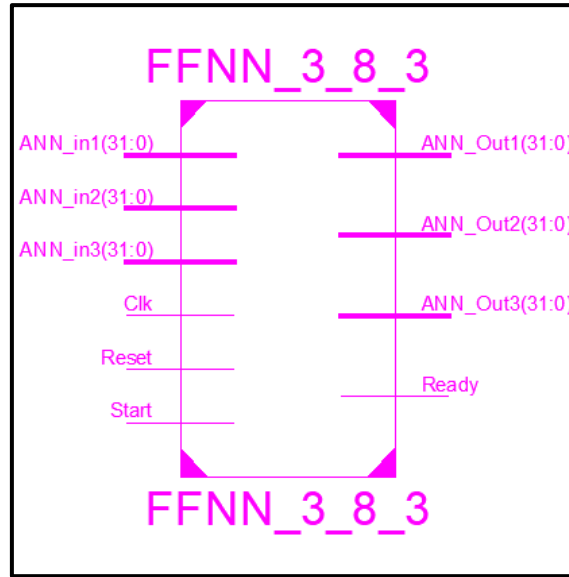
| FPGA Çip İstatistikleri | RadBas | LogSig | TanSig |
|--------------------------------|-------------------------|-------------------------|--------------------------|
| Mak. Çalışma Frekans (MHz) | 168.990 | 168.990 | 168.990 |
| Slice Registers Sayısı | 1.646 / 82.000 (% 2) | 3.696 / 82.000 (% 4) | 8.699 / 82.000 (% 10) |
| Slice LUTs Sayısı | 1.658 / 41.000 (% 4) | 3.079 / 41.000 (% 7) | 4.287 / 41.000 (% 10) |
| IOBs Sayısı | 67 / 300 (% 22) | 67 / 300 (% 22) | 67 / 300 (% 22) |
| BUFG/BUFG CTRLs Sayısı | 1 / 32 (% 3) | 1 / 32 (% 3) | 1 / 32 (% 3) |
| DSP48E1s Sayısı | 15 / 240 (% 6) | 7 / 240 (% 2) | 11 / 240 (% 5) |

Tasarımları biten doğrusal ve doğrusal olmayan AF ile bu tezde amaçlanan IQ-Math sabit noktalı sayı tabanlı aktivasyon fonksiyonları kütüphanesi oluşturulmuştur. Ardından örnek bir YSA tasarımı yapılmıştır.

4.1.3 FPGA Tabanlı YSA Tasarımı

Bu tezde tasarlanan AF kütüphanesinden faydalanılarak örnek bir YSA tasarımı yapılmıştır. YSA tasarımı FPGA çiplerinde uygulanmak üzere IQ-Math sabit noktalı sayı standardına uygun 32 bit (16I-16Q) olarak VHDL kodlama dilinde kodlanmıştır. YSA tasarımında MLFF (Multi Layer Feed-Forward -Çok Katmanlı İleri Beslemeli) ağ yapısı kullanılmıştır.

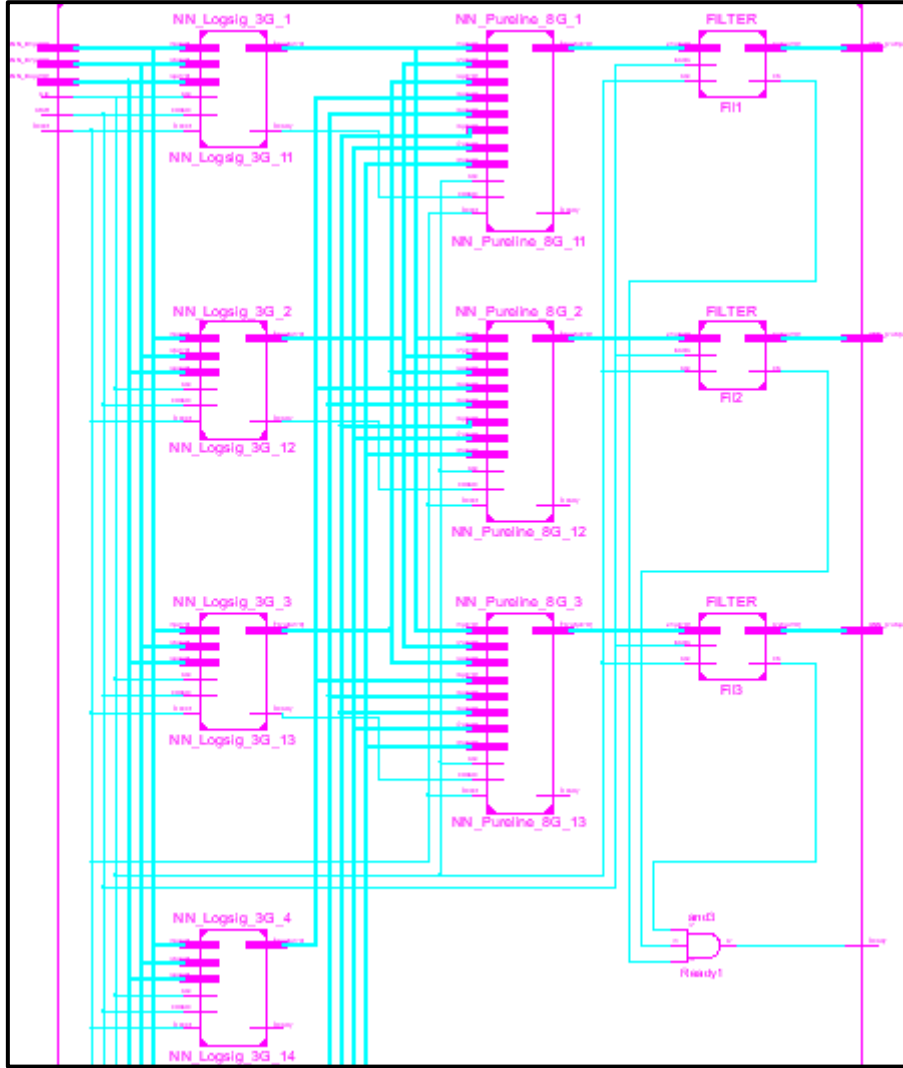
İleri beslemeli YSA yapısına ait en üst seviye FPGA blok şeması Şekil 4.27’de verilmiştir. Bu şemada YSA girdi bilgilerini ifade eden 32 bitlik ANN_in1, ANN_in2 ve ANN_in3 giriş sinyalleri bulunmaktadır. Clk, start ve reset sinyalleri 1 bit uzunluğunda olup tasarımda bulunan alt ünitelerin eş zamanlı olarak çalışması için kullanılmıştır. Tasarımdan üretilen sonuçlar 32 bitlik ANN_out1, ANN_out2 ve ANN_out3 sinyallerinden alınmaktadır. Ready sinyali 1 bit uzunluğunda tasarımda kontrol amaçlı kullanılmıştır. Bu sinyal tasarımda sonuçlar üretilmeye başlanmasına kadar lojik 0, sonuçlar üretilirken lojik 1 üretmektedir.



Şekil 4.26 İleri beslemeli YSA yapısına ait en üst seviye FPGA blok şeması.

Şekil 4.27’de ileri beslemeli YSA yapısının ikinci seviye FPGA blok şeması verilmiştir. Tasarımda bulunan her bir NN_Logsig üniteleri gizli katmandaki nöronları tasarlamak için kullanılmıştır. Bu ünitelerde giriş katmanından gelen girdi sinyalleri kendi ağırlıkları ile çarpılmakta ve bu çarpımlara ilgili nörona ait eşik değeri de eklenerek toplama işlemi yapılır. Toplam sonucu LogSig aktivasyon fonksiyonunda işlenerek ilgili nörona ait çıkış

sinyalleri üretilmektedir. Bu çıkış sinyalleri çıkış katmanındaki NN_Pureline ünitelerine iletilerek NN_Logsig ünitelerindeki işlemler tekrar edilir fakat toplam sonucu PureLin aktivasyon fonksiyonunda işlenmektedir. Böylece çıkış katmanındaki nöronların tasarımı da yapılmış olmaktadır. Son olarak gereksiz sinyalleri yok etmek için filter üniteleri kullanılarak tasarım tamamlanmaktadır.

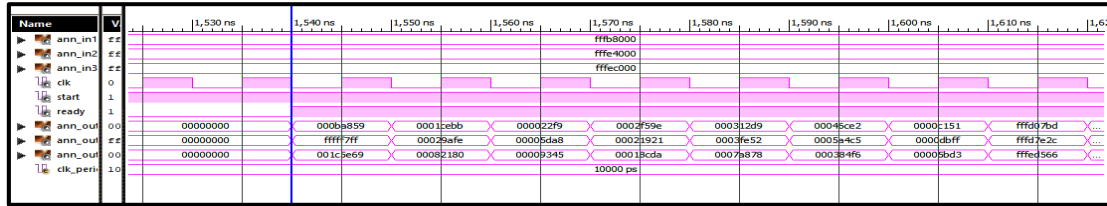


Şekil 4.27 İleri beslemeli YSA yapısına ait ikinci seviye FPGA blok şeması.

4.1.3.1 İleri Beslemeli YSA Tasarımının Simülasyon Sonuçları

Bu tez çalışmasında ileri beslemeli YSA tasarımları FPGA platformunda IQ-Math sabit noktalı sayı standardına uygun 32 bit (16I-16Q) olarak VHDL dilinde kodlanmıştır. Tasarımları test etmek için Xilinx ISE Design Suite 14.7 programında bir testbench

ünitesi oluşturulmuş ve bu ünitenin çalıştırılması ile simülasyon sonuçları elde edilmiştir. Şekil 4.29’da FFNN ileri beslemeli YSA tasarımının simülasyon sonuçları verilmiştir. Simülasyonda 1 saat darbesi 10 ns olmak üzere 134 saat darbesi sonunda sonuçlar üretilmektedir. Sonuçlar daha anlaşılabilir olması için hexadecimal (on altılık) sayı tabanında gösterilmiştir.



Şekil 4.28 FFNN ileri beslemeli ağ yapısının Xilinx ISE simülasyon sonuçları.

4.1.3.2 İleri Beslemeli YSA Tasarımının Hata Analizleri

İleri beslemeli YSA tasarımlarının test edilmesinden sonra MSE ve RMSE hata analizleri yapılmıştır. Çizelge 4.4’te ileri beslemeli YSA tasarımına ait MSE ve RMSE hata analiz değerleri verilmiştir.

Çizelge 4.4 İleri beslemeli ağ yapısının hata analizleri.

| YSA Tasarımı | MSE | RMSE |
|--------------|----------|----------|
| FFNN | 8.86E-06 | 2.98E-03 |

Simülasyon ve hata analizlerinin ardından YSA tasarımı Xilinx Kintex-7 XC7K70T-3FBG676 FPGA çipi için sentezlenmiş ve Place & Route işlemi yapılarak elde edilen çip istatistikleri Çizelge 4.5’te sunulmuştur.

Çizelge 4.5 İleri beslemeli ağ yapısının FPGA çip istatistikleri.

| FPGA Çip İstatistikleri | FFNN |
|----------------------------|---------|
| Mak. Çalışma Frekans (MHz) | 168.990 |

Çizelge 4.5 (Devam) İleri beslemeli ağ yapısının FPGA çip istatistikleri.

| | |
|------------------------|---------------------------|
| Slice Registers Sayısı | 31.926/ 82.000 (% 38) |
| Slice LUTs Sayısı | 26.878 / 41.000 (% 65) |
| IOBs Sayısı | 195 / 300 (% 65) |
| BUFG/BUFGCT RLs Sayısı | 1 / 32 (% 3) |
| DSP48E1s Sayısı | 188 / 240 (% 78) |

Bu tez çalışmasında yapılan tüm AF tasarımları ve YSA yapısından faydalanılarak doğrusal ve doğrusal olmayan AF ile oluşturulan dört girişli YSA nöronları tasarlanmıştır. Tüm bu nöron tasarımları literatürdeki bir çalışmada yapılan dört girişli YSA nöron tasarımları ile kıyaslanarak Çizelge 4.6’da verilmiştir. Çizelgede iki çalışma FPGA çip kullanımları, sayı standardı, kullanılan platform ile platformun nöron kapasitesi ve çalışma frekansı bakımından kıyaslanmıştır. Platformun nöron kapasitesi dilim kullanımı (slices utilization) parametresine göre belirlenmiştir. Sunulan tez çalışması genel olarak FPGA çip kullanımı ve çalışma frekansı bakımından daha avantajlıdır. Fakat doğrusal olmayan AF nöronlarının çalışma frekansları ilgili çalışmadan daha düşük olmaktadır.

Çizelge 4.6 Farklı sayı standardı ve FPGA platformu kullanılarak gerçekleştirilen dört girişli YSA nöron tasarımları ve özellikleri.

| Yapılan çalışma | Kullanılan AF | Sayı standardı | Kullanılan platform | Slice Regs. sayısı (%) | Slice LUTs sayısı (%) | Slices Utilization (%) | Platformun nöron kapasitesi | Çalışma frekansı (MHz) |
|-----------------------|---------------|--------------------------------|---------------------------|------------------------|-----------------------|------------------------|-----------------------------|------------------------|
| Koyuncu vd. (2017) | PosLin | 32 Bit IEEE-754 Floating-point | Xilinx Virtex-6 XC6VLX75T | 5 | 9 | 12 | 8 | 576.89 |
| Sunulan Tez Çalışması | PosLin | 32 bit (16I-16Q) IQ-Math | Xilinx Kintex-7 XC7K70T | 0 | 1 | 3 | 33 | 802.38 |
| Koyuncu vd. (2017) | HardLim | 32 Bit IEEE-754 Floating-point | Xilinx Virtex-6 XC6VLX75T | 5 | 9 | 11 | 9 | 576.89 |
| Sunulan Tez Çalışması | HardLim | 32 bit (16I-16Q) IQ-Math | Xilinx Kintex-7 XC7K70T | 0 | 1 | 3 | 33 | 802.38 |

Çizelge 4.6 (Devam) Farklı sayı standardı ve FPGA platformu kullanılarak gerçekleştirilen dört girişli YSA nöron tasarımları ve özellikleri.

| Yapılan çalışma | Kullanılan AF | Sayı standardı | Kullanılan platform | Slice Regs. sayısı (%) | Slice LUTs sayısı (%) | Slices Utilization (%) | Platformun nöron kapasitesi | Çalışma frekansı (MHz) |
|-----------------------|---------------|--------------------------------|---------------------------|------------------------|-----------------------|------------------------|-----------------------------|------------------------|
| Koyuncu vd. (2017) | HardLims | 32 Bit IEEE-754 Floating-point | Xilinx Virtex-6 XC6VLX75T | 5 | 9 | 11 | 9 | 576.89 |
| Sunulan Tez Çalışması | HardLims | 32 bit (16I-16Q) IQ-Math | Xilinx Kintex-7 XC7K70T | 0 | 1 | 3 | 33 | 802.38 |
| Koyuncu vd. (2017) | SatLin | 32 Bit IEEE-754 Floating-point | Xilinx Virtex-6 XC6VLX75T | 5 | 9 | 12 | 8 | 576.89 |
| Sunulan Tez Çalışması | SatLin | 32 bit (16I-16Q) IQ-Math | Xilinx Kintex-7 XC7K70T | 0 | 1 | 3 | 33 | 645.95 |
| Koyuncu vd. (2017) | SatLins | 32 Bit IEEE-754 Floating-point | Xilinx Virtex-6 XC6VLX75T | 5 | 9 | 12 | 8 | 576.89 |
| Sunulan Tez Çalışması | SatLins | 32 bit (16I-16Q) IQ-Math | Xilinx Kintex-7 XC7K70T | 0 | 1 | 3 | 33 | 569.22 |
| Koyuncu vd. (2017) | TriBas | 32 Bit IEEE-754 Floating-point | Xilinx Virtex-6 XC6VLX75T | 6 | 10 | 13 | 7 | 576.89 |
| Sunulan Tez Çalışması | TriBas | 32 bit (16I-16Q) IQ-Math | Xilinx Kintex-7 XC7K70T | 0 | 1 | 3 | 33 | 569.22 |
| Koyuncu vd. (2017) | RadBas | 32 Bit IEEE-754 Floating-point | Xilinx Virtex-6 XC6VLX75T | 9 | 17 | 21 | 7 | 304.53 |
| Sunulan Tez Çalışması | RadBas | 32 bit (16I-16Q) IQ-Math | Xilinx Kintex-7 XC7K70T | 2 | 5 | 9 | 11 | 168.99 |
| Koyuncu vd. (2017) | LogSig | 32 Bit IEEE-754 Floating-point | Xilinx Virtex-6 XC6VLX75T | 10 | 18 | 22 | 4 | 304.53 |
| Sunulan Tez Çalışması | LogSig | 32 bit (16I-16Q) IQ-Math | Xilinx Kintex-7 XC7K70T | 5 | 9 | 17 | 5 | 168.99 |
| Koyuncu vd. (2017) | TanSig | 32 Bit IEEE-754 Floating-point | Xilinx Virtex-6 XC6VLX75T | 10 | 18 | 23 | 4 | 304.53 |
| Sunulan Tez Çalışması | TanSig | 32 bit (16I-16Q) IQ-Math | Xilinx Kintex-7 XC7K70T | 11 | 12 | 22 | 4 | 168.99 |

5. TARTIŞMA ve SONUÇ

YSA insan beyninin öğrenme ve karar verme gibi yetenekleri referans alınarak geliştirilmiş yapay zekânın bir alt alanıdır. İnsan beyninde mevcut olan öğrenme, sınıflandırma, tahmin etme, karar verme gibi birçok özellik YSA için birer çalışma alanıdır. YSA öğrenme ve adaptasyon yeteneği, paralel veri işleme, doğrusal olmayan sistemlerde kullanılma gibi önemli özelliklere sahiptir. Bu özellikleri sayesinde günümüzde sinyal ve görüntü işleme, kontrol, hata tespiti, sınıflandırma, optimizasyon, tahmin, karar verme, güvenli haberleşme, uzay teknolojisi, kaotik osilatör tasarımı ve tıp gibi birçok farklı alanda YSA yaygın olarak tercih edilmektedir. Bu çalışma alanlarında YSA yazılım ve donanım tabanlı olarak kullanılmaktadır. Donanım tabanlı YSA çalışmalarında FPGA çipleri geniş kullanım alanına sahiptir.

Bu tez çalışmasının, ilk aşamasında doğrusal AF olan TriBas, HardLim, HardLims, PosLin, SatLin ve SatLins fonksiyonları VHDL dilinde kodlanarak 32 bit (16I-16Q) IQ-Math sayı standardında tasarlanmıştır. Ardından, tüm doğrusal AF tasarımları Xilinx ISE Design Suite 14.7 programı kullanılarak test edilmiş ve elde edilen simülasyon sonuçları verilmiştir. Test işleminin ardından tasarımlar Xilinx Kintex-7 ailesinin XC7K70T-3FBG676 FPGA çipi için sentezlenmiş ve Place & Route işleminden sonra elde edilen FPGA çip istatistikleri sunulmuştur. Çip istatistikleri incelendiğinde, HardLim ve HardLims fonksiyonlarının en az çip kaynağı tükettiği görülmüştür. Ayrıca, doğrusal AF tasarımları içerisinde 1379.172 MHz ile en yüksek çalışma frekansına SatLin AF'nin sahip olduğu tespit edilmiştir.

Tez çalışmasının ikinci aşamasına, YSA uygulamalarında oldukça yaygın bir şekilde kullanılan ve doğrusal olmayan AF olan RadBas, LogSig ve TanSig AF, 32 bit (16I-16Q) IQ-Math sayı standardında VHDL dili ile tasarlanmıştır. Doğrusal olmayan üç AF içerisinde bulunan e^x üstel fonksiyonu tasarımında CORDIC-LUT yaklaşımı kullanılmıştır. Tasarımlar Xilinx ISE Design Suite 14.7 programında test edilmiş ve simülasyon sonuçları verilmiştir. Test işleminin ardından yapılan tasarımlar Xilinx Kintex-7 XC7K70T-3FBG676 FPGA çipi için sentezlenmiş ve Place & Route işleminden sonra elde edilen FPGA çip istatistikleri sunulmuştur. Ayrıca, bu üç FPGA tabanlı AF

tasarımlarının MSE ve RMSE hata analizleri nümerik tabanlı AF kullanılarak gerçekleştirilmiş ve elde edilen sonuçlar sunulmuştur. Yapılan bu 6 farklı doğrusal ve 3 farklı doğrusal olmayan AF tasarımları ile bir aktivasyon fonksiyonları kütüphanesi oluşturulmuştur.

Tez çalışmasının üçüncü aşamasında, FPGA tabanlı örnek bir üç girişli ve üç çıkışlı ileri beslemeli YSA tasarımı gerçekleştirilmiştir. YSA tasarımının gizli katmanında 8 nöron bulunmaktadır. Tasarımın gizli katmanında LogSig AF ve çıkış katmanında PureLin AF kullanılmıştır. Örnek YSA tasarımı 32 bit (16I-16Q) IQ-Math sayı standardında VHDL dilinde kodlanmıştır. Tasarım Xilinx ISE Design Suite 14.7 programında VHDL dilinde bir testbench oluşturularak test edilmiştir. Test işleminden elde edilen tasarıma ait simülasyon sonuçları için MSE ve RMSE hata analizleri yapılmış ve sonuçlar değerlendirilmiştir. Test işleminden sonra yapılan tasarım Kintex-7 XC7K70T-3FBG676 FPGA çipi için sentezlenmiş ve Place & Route işleminin ardından FPGA çip istatistikleri sunulmuştur.

Ayrıca bu tez çalışmasında yapılan tüm AF tasarımları ve YSA yapısından faydalanılarak doğrusal ve doğrusal olmayan AF ile oluşturulan dört girişli YSA nöronları tasarlanmıştır. Bu nöronlar literatürdeki bir çalışmada yapılan dört girişli YSA nöron tasarımları ile FPGA çip kullanımları, sayı standardı, kullanılan platform ile platformun nöron kapasitesi ve çalışma frekansı bakımından karşılaştırılmıştır. Platformun nöron kapasitesi dilim kullanımı (slices utilization) parametresine göre belirlenmiştir. Sonuçta sunulan tez çalışması ilgili literatür çalışmasına kıyas ile genel olarak FPGA çip kullanımı ve çalışma frekansı bakımından daha avantajlıdır.

Gelecekteki çalışmalarda, bu tez çalışmasında 32 bit IQ-Math sayı standardı ile VHDL dilinde tasarlanan doğrusal ve doğrusal olmayan aktivasyon fonksiyonları kullanılarak sinyal ve görüntü işleme, güvenli haberleşme, gerçek ve sözde rasgele sayı üretici tasarımları ve savunma sanayi gibi birçok alandaki gerçek zamanlı YSA uygulamaları gerçekleştirilebilir. Ayrıca tasarımlarda kullanılan 32 bit sabit noktalı sayı standardı yerine daha düşük bit değerine sahip sayı standartları kullanılarak FPGA çipleri içerisinde daha büyük ağ yapılarının gerçekleştirilmesi sağlanabilir.

6. KAYNAKLAR

- Abdullah H T, Younis B M, 2019, FPGA Based Bone Fracture Detector, IOP Conference Series: Materials Science and Engineering, 745, Article number 012052.
- Adetiba E, Ibikunle F, Daramola S, and Olajide A, 2014, Implementation of Efficient Multilayer Perceptron ANN Neurons on Field Programmable Gate Array Chip, International Journal of Engineering and Technology IJET-IJENS, 14, 151-159.
- Afrah İ A, 2020, Kişisel Sağlık Desteği için Yapay Zeka Tabanlı Mobil Uzman Sistem Uygulaması Geliştirilmesi, Süleyman Demirel Üniversitesi, Fen Bilimleri Enstitüsü, Yüksek Lisans Tezi, Isparta.
- Ahmed R, Sayed M E, Gadsden S A, Tjong J, Habibi, S, 2015, Automotive Internal-Combustion-Engine Fault Detection and Classification Using Artificial Neural Network Techniques, IEEE Transactions on Vehicular Technology, 64, 21-33.
- Akçay M S, Koyuncu I, Alçın, M, and Tuna M, 2020, IQ-Math Tabanlı RadBas Aktivasyon Fonksiyonunun FPGA Üzerinde Gerçeklenmesi, International Asian Congress on Contemporary Sciences-IV, 599-607.
- Akıllı A, Atıl H, ve Kesenkaş H, 2014, Çiğ süt kalite değerlendirmesinde bulanık mantık yaklaşımı, Kafkas Üniversitesi Veteriner Fakültesi Dergisi, 20, 223-229.
- Akpolat A, 2015, FPGA tabanlı nesne algılama, Fırat Üniversitesi, Fen Bilimleri Enstitüsü, Yüksek Lisans Tezi, Elazığ.
- Akyazı Ö, Usta M A, ve Akpınar, A S, 2011, Kapalı ortam sıcaklık ve nem denetiminin farklı bulanık üyelik fonksiyonları kullanılarak gerçekleştirilmesi, 6th International Advanced Technologies Symposium, 158-162.
- Alçın M, 2017, FPGA üzerinde YSA tabanlı gerçek rasgele sayı üreticinin tasarımı ve gerçekleşmesi, Sakarya Üniversitesi, Fen Bilimleri Enstitüsü, Doktora Tezi, Sakarya.
- Alçın M, Koyuncu I, Tuna M, Varan M, and Pehlivan I, 2019, A novel high speed Artificial Neural Network-based chaotic True Random Number Generator on Field Programmable Gate Array, International Journal of Circuit Theory and Applications, 47, 365-378.
- Alçın M, Pehlivan I, and Koyuncu I, 2016, Hardware design and implementation of a novel ANN-based chaotic generator in FPGA, Optik, 127, 5500-5505.

- Alçın M, Tuna M, Erdoğan P, and Koyuncu İ, 2021, FPGA-based Dual Core TRNG Design Using Ring and Runge-Kutta-Butcher based on Chaotic Oscillator, Chaos Theory and Applications, 3, 20-28.
- Allahverdi N, 2002, Uzman Sistemler: Bir Yapay Zeka Uygulaması, Atlas Akademik Yayıncılık, 248s, İstanbul.
- Al-Rikabi H, Al-Ja'afari M, Ali A, and Abdulwahed S, 2020, Generic Model Implementation of Deep Neural Network Activation Functions Using GWO Optimized SCPWL Model on FPGA, Microprocessors and Microsystems, 77, Article number 103141.
- Arslan M, 2019, Öğretmen performanslarının bulanık mantık yöntemi ile değerlendirilmesi, Van Yüzüncü Yıl Üniversitesi, Eğitim Bilimleri Enstitüsü, Yüksek Lisans Tezi, Van.
- Aslantaş V, Toprak A N, Kurban R, ve Bendeş E, 2013, Çoklu-Odaklı Görüntülerin Genetik Algoritma Kullanılarak Birleştirilmesi, Sigma Mühendislik ve Fen Bilimleri Dergisi, 5, 25-37.
- Avcı D, Leblebicioğlu M K, Poyraz M, ve Dogantekin E, 2014, A new method based on adaptive discrete wavelet entropy energy and neural network classifier (ADWEENN) for recognition of urine cells from microscopic images independent of rotation and scaling, Journal of Medical Systems, 38, 1-9.
- Ay H, 2019, Üretim Sürecinin Doğal Dil İşleme ile Modellenmesi, Eskişehir Teknik Üniversitesi, Lisansüstü Eğitim Enstitüsü, Yüksek Lisans Tezi, Eskişehir.
- Babüroğlu B, 2019, İstatistiksel doğal dil işlemede derin öğrenme yöntemleri kullanılarak çevrimiçi Türkçe akademik derlem çözümlenmesi, Kahramanmaraş Sütçü İmam Üniversitesi, Fen Bilimleri Enstitüsü, Yüksek Lisans Tezi, Kahramanmaraş.
- Bahl L, Jelinek F, and Mercer R, 1983, A maximum likelihood approach to continuous speech recognition, IEEE Transactions On Pattern Analysis And Machine Intelligence, 2, 179-190.
- Balaban E M, Kartal E, 2018, Veri Madenciliği ve Makine Öğrenmesi Temel Algoritmaları ve R Dili ile Uygulamaları, Çağlayan Kitabevi, 319s, İstanbul.
- Bargsten V, de Gea Fernández J, 2020, Distributed computation and control of robot motion dynamics on FPGAs, International Journal of SN Applied Sciences, 2, Article number 1239.

- Batal M S, 2016, Yapay Zeka Uygulamaları ve Yapay Zekanın Geleceği, Uluslararası Sunhill Üniversitesi, Yüksek Lisans Tezi.
- Baykal N, Beyan T, 2004, Bulanık Mantık Uzman Sistemler ve Denetleyiciler, Bıçaklar Kitapevi, 509s, Ankara.
- Bih J, 2006, Paradigm shift - an introduction to fuzzy logic, Potentials IEEE, 25, 6-21.
- Brown P, Cocke J, Pietra S, Pietra V, Jelinek F, Lafferty J, vd., 1990, A statistical approach to machine translation, Computational Linguistic, 2, 79-85.
- Carbonell J G, 1978, POLITICS: Automated Ideological Reasoning, Cognitive science, 2, 27-51.
- Çelikel R, 2019, ANN based angle tracking technique for shaft resolver, Measurement, 148, Article number 106910.
- Cheng Y, Jia Y, Fang R, She L, Xi N, Chai J, 2014, Modelling and analysis of natural language controlled robotic systems, IFAC Proceedings Volumes, 47, 11767-11772.
- Chua L O, Yang L, 1988, Cellular neural networks: Theory, IEEE Transactions on Circuits and Systems, 35, 1273-1290.
- Cirstea M, Dinu A, Khor J, McCormick M, 2002, Neural and Fuzzy Logic Control of Drives and Power Systems, Newnes, 399p.
- Cullingford R E, 1977, Organizing World Knowledge for Story Understanding by Computer, Yale University, Ph.D. Thesis, New Haven.
- Çakır A, Küçüksille E U, Altıntaş V, 2018, Baskı Devre Yerleşim Optimizasyonu için Genetik Algoritma, Journal of Technical Sciences, 8, 5-10.
- Çavuşlu M A, Karakuzu C, Şahin S, 2010, Parçacık Sürü Optimizasyonu Algoritması ile Yapay Sinir Ağı Eğitiminin FPGA Üzerinde Donanımsal Gerçeklenmesi, Politeknik Dergisi, 13, 83-92.
- Çavuşlu M, Karakuzu C, Şahin S, Yakut M, 2011, Neural network training based on FPGA with floating point number format and it's performance, Neural Computing and Applications, 20, 195-202.
- Çelebi M, 2007, Genetik Algoritma İle Yağlı Bir Trafonun Maliyet Optimizasyonu, C.B.Ü. Fen Bilimleri Dergisi, 3, 41-48.
- Çetin O, 2014, Yapay sinir ağlarının uyarlanabilir donanımsal yapılarda gerçekleştirilmesi. Sakarya Üniversitesi, Fen Bilimleri Enstitüsü, Doktora Tezi, Sakarya.

- Çömlekçi O, 2020, Endüstriyel Otomasyon Sistemlerinde Yapay Zeka, Bursa Uludağ Üniversitesi, Fen Bilimleri Enstitüsü, Yüksek Lisans Tezi, Bursa.
- Dağdelen U, 1996, Bulanık Mantık ile Adım Motor Kontrolü, Erciyes Üniversitesi, Fen Bilimleri Enstitüsü, Yüksek Lisans Tezi, Kayseri.
- Değerli A, 2008, Maksimum Doğrultucu Moment Kolu Analizinin Bulanık Mantık ve Sinirsel Bulanık Mantık Kullanılarak Yapılması, *Electronic Letters on Science & Engineering*, 4, 1-10.
- Delibaş A, 2008, Doğal Dil İşleme ile Türkçe Yazım Hatalarının Denetlenmesi, İstanbul Teknik Üniversitesi, Fen Bilimleri Enstitüsü, Yüksek Lisans Tezi, İstanbul.
- Dempster A, Laird N, Rubin D, 1977, Maximum likelihood from incomplete data via the EM algorithm, *Journal of the Royal Statistical Society*, 1, 1-38.
- Demuth H B, Beale M H, Jess D O, Hagan M T, 2014, *Neural network design*, Martin Hagan, 800p.
- Du X K, 2011, The New Elman ANN Application in Accuracy Improvement of Robot Navigation and Obstacle Avoidance Technology, *Advanced Materials Research*, 383-390, 1447-1451.
- El-Madany H, Fahmy F, El-Rahman N, Dorrah H, 2012, Spacecraft Neural Network Control System Design Using FPGA, *The Online Journal on Electronics and Electrical Engineering*, 4, 495-500.
- Elmas Ç, 2018, *Yapay Zeka Uygulamaları*, Seçkin Yayıncılık, 424s, Ankara.
- Elrharras A, Moukhlis S, Saadane R, Wahbi M, Hamdoun A, 2015, FPGA-Based Fully Parallel PCA-ANN for Spectrum Sensing, *Computer and Information Science*, 8, 108-118.
- Erick L O, 2007, *Fixed-point representation & fractional math.*, Oberstar Consulting, Madison, United States.
- Eroğlu H, Şişman Y, 2020, Arazi Toplulaştırması Dağıtım İşleminde Tek Amaçlı Genetik Algoritmanın Kullanılması, *Journal of Geomatics*, 5, 91-99.
- Ersoy E, Karal Ö, 2012, Yapay sinir ağları ve insan beyni, *Journal of the Human and Social Science Researches*, 1, 188-205.
- Fei J, Ding H, 2012, Adaptive sliding mode control of dynamic system using RBF neural network, *Nonlinear Dynamics*, 70, 1563-1573.

- Gankidi P, Thangavelautham J, 2017, FPGA architecture for deep learning and its application to planetary robotics, 2017 IEEE Aerospace Conference, 1-9.
- Geçgel S, 2019, Güvenli Telsiz İletişim Sağlanmasına Yönelik Yapay Zeka Tabanlı Sınıflandırma Metotları, İstanbul Teknik Üniversitesi, Fen Bilimleri Enstitüsü, Yüksek Lisans Tezi, İstanbul.
- Gençal M C, 2019, Genetik Algoritmaların Performansını Geliştirme Üzerine Bir Çalışma, Çukurova Üniversitesi, Fen Bilimleri Enstitüsü, Doktora Tezi, Adana.
- Giarratano J, Riley G, 1994, Expert Systems: Principles and Programming, PWS Publishing, 608p.
- Gupta S, Vyas A, Trivedi G, 2020, FPGA Implementation of Simplified Spiking Neural Networks, 27th IEEE International Conference on Electronics, Circuits and Systems, 1-4.
- Gül Gökay E, Çağatan T, 2002, Genetik algoritmalar ve uygulama alanları, Uludağ Üniversitesi İktisadi ve İdari Bilimler Fak. Dergisi, 21, 129-152.
- Gürsoy H, 2016, FPGA tabanlı otomatik kontrol sistemleri geliştirme, Hacettepe Üniversitesi, Fen Bilimleri Enstitüsü, Yüksek Lisans Tezi, Ankara.
- Hajduk Z, 2017, High accuracy FPGA activation function implementation of neural networks, Neurocomputing, 247, 59-61.
- Halat M, Özkan Ö, 2021, The optimization of UAV routing problem with a genetic algorithm to observe the damages of possible Istanbul earthquake, Pamukkale University Journal of Engineering Sciences, 27, 187-198.
- Haykin S, 2005, Neural Network A comprehensive foundation, Macmillan Coll. Div., 696p.
- Haznedar B, Arslan M T, Kalınlı A, 2017, Karaciğer mikrodizi kanser verisinin sınıflandırılması için genetik algoritma kullanarak ANFIS'in eğitilmesi, Sakarya Üniversitesi Fen Bilimleri Enstitüsü Dergisi, 21, 54-62.
- Himavathi S, Muthuramalingam D A, 2007, Feedforward neural network implementation in FPGA using layer multiplexing for effective resource utilization, IEEE Trans. Neural Netw., 18, 880-888.
- Hindle D, Rooth M, 1993, Structural ambiguity and lexical relations, Computational Linguistics, 1, 103-120.
- İçen D, Günay S, 2014, Uzman Sistemler ve İstatistik, İstatistik&Aktüerya, 7, 37-45.

- Işık A H, Bilen M, Yiğit T, 2017, İnteraktif ve Web Tabanlı Genetik Algoritma Eğitim Yazılımı, Süleyman Demirel Üniversitesi Fen Bilimleri Enstitüsü Dergisi, 21, 928-934.
- Işıklı Ş, 2007, Bulanık Mantık ve Bulanık Teknolojiler, Araştırma Ankara Üniversitesi Dil ve Tarih-Coğrafya Fakültesi Felsefe Bölümü Dergisi, 19, 1-19.
- Işıklı Ş, 2010, Lotfi A. Zadeh'nin hayat hikâyesi ve bulanık paradigmanın üç temel unsuru. Kutadgubilig: Felsefe-Bilim Araştırmaları Dergisi, 17, 89-101.
- Jiang W, Kong S G, 2007, Block-Based Neural Networks for Personalized ECG Signal Classification, IEEE Transactions on Neural Networks, 18, 1750-1761.
- K Deb, 2001, Multi-objective optimisation using evolutionary algorithms, Chichester, UK: Wiley.
- Kahan W, 1996, IEEE standard 754 for binary floating-point arithmetic, Lecture Notes on the Status of IEEE, 754: 94720-1776.
- Karaboğa D, 2017, Yapay Zeka Optimizasyon Algoritmaları, Nobel Akademi Yayıncılık, 246s, Ankara.
- Karataş S, Sarıtaş E, 2013, Her yönüyle FPGA ve VHDL, Palme Yayıncılık, 411s, Ankara.
- Kastal A, Köse A O, 2009, Yapay Zeka Uzman Sistemler, XIV.Türkiye'de İnternet Konferansı, İstanbul Bilgi Üniversitesi.
- Katircioğlu F, Kelek M M, Şen M, Koyuncu I, Oğuz Y, 2018, FPGA-Based Design of Gaussian Membership Function for Real-Time Fuzzy Logic Applications, V. International Multidisciplinary Congress Of Eurasia, 33-39.
- Kaur M, Sivia J S, 2020, ANN and FA Based Design of Hybrid Fractal Antenna for ISM Band Applications. Progress In Electromagnetics Research C, 98, 127-140.
- Kaya S, 2018, Doğal Dil İşleme Teknikleriyle Yazar-Kitap Tanıma, İstanbul Aydın Üniversitesi, Fen Bilimleri Enstitüsü, Yüksek Lisans Tezi, İstanbul.
- Keskenler M F, Keskenler E F, 2017, Bulanık Mantığın Tarihi Gelişimi, Takvim-i Vekayi, 5, 1-10.
- Kıyak E, 2003, Bulanık Mantık Yöntemiyle Uçuş Kontrol Uygulamaları, Anadolu Üniversitesi, Fen Bilimleri Enstitüsü, Yüksek Lisans Tezi, Eskişehir.
- Klir G J, Yuan B, 1995, Fuzzy Sets and Fuzzy Logic: Theory and Applications, New Jersey: Prentice Hall PTR.

- Koyuncu İ, 2014, Kriptolojik Uygulamalar için FPGA tabanlı Yeni Kaotik Osilatörlerin ve Gerçek Rasgele Sayı Üreteçlerinin Tasarımı ve Gerçeklenmesi, Sakarya Üniversitesi, Fen Bilimleri Enstitüsü, Doktora Tezi, Sakarya.
- Koyuncu I, Sahin I. 2011, A CORDIC based ex Calculator Unit Design for FPGA Chips, e-Journal of New World Sciences Academy, 6, 1565-1572.
- Koyuncu I, Seker H, 2019, Implementation of Dormand-Prince based chaotic oscillator designs in different IQ-Math number standards on FPGA, Sakarya University Journal of Science, 23, 859-868.
- Koyuncu I, Akcay M S, Tuna M, Alçın M, 2019, Implementation of IQ-Math-based Linear Activation Functions on FPGA, 1st International Congress of Multidisciplinary Studies and Research, 114-124.
- Koyuncu I, Erdogmus P, Tuna M, Alçın M, 2019, FPGA üzerinde YSA-RİNG tabanlı yeni bir gerçek rasgele sayı üretici tasarımı, I. International Science and Innovation Congress, 463-468.
- Koyuncu I, Sahin I, Gloster C, Saritekin N K, 2017, A neuron library for rapid realization of artificial neural networks on FPGA: A case study of rössler chaotic system, Journal of Circuits, Systems, and Computers, 26, Article number 1750015.
- Koyuncu İ, Şeker H, Tuna M, Alçın M, 2018, Dormand-Prince tabanlı kaotik osilatör tasarımının FPGA üzerinde gerçekleşmesi, EurasianSciEnTech 2018, 1059-1065.
- Koyuncu I, Yılmaz C, Alçın M, Tuna M, 2019, Artificial Neural Network Optimization of a Geothermal Energy Assisted Hydrogen Generation Economy, 4th International Conference on Engineering Technology and Applied Sciences, 360-364.
- Kösten M, Çavuşlu M, 2015, VHDL ile Sayısal Tasarım ve FPGA Uygulamaları, Kodlab Yayıncılık, 328s, İstanbul.
- Lehnert W, 1977, Process of question answering, Yale University, Ph.D. Thesis.
- Lehnert W G, 1981, Plot Units and Narrative Summarization, Cognitive Science, 5, 293-331.
- Lin C J, Tsai H M, 2008, FPGA implementation of a wavelet neural network with particle swarm optimization learning, Mathematical and Computer Modelling, 47, 982-996.

- Lortođlu M, 2019, FPGA tabanlı yapay sinir ađı kullanılarak buđday turlerinin sınıflandırılması, KTO Karatay Üniversitesi, Fen Bilimleri Enstitüsü, Yüksek Lisans Tezi, Konya.
- Meehan J R, 1976, The Metanovel: Writing Stories by Computer, Yale University, Ph.D. Thesis.
- Mello M M, Ventura L, 2020, Method for estimating luminous transmittance using an RGB sensor and ANN, *Optical Engineering*, 59, Article number 124109.
- Mohammed R K, Abdullah H A, 2020, Implementation of digital and analog modulation systems using FPGA, *Analog Modul Indonesian Journal of Electrical Engineering and Computer Science*, 18, 485-493.
- Moorgas K, Govender P, 2013, Hybrid motion detection system using DSP and ANN ensembles, *Proceedings of the International MultiConference of Engineers and Computer Scientists*, 484-490.
- Nabiyev V, 2005, *Yapay Zeka*, Seçkin Yayıncılık, 764s, Ankara.
- Nambiar V, Khalil-Hani M, Sahnoun R, Marsono M, 2014, Hardware implementation of evolvable block-based neural networks utilizing a cost efficient sigmoid-like activation function, *Neurocomputing*, 140, 228-241.
- Nivre J, Scholz M, 2004, Deterministic dependency parsing of English text, *Proceedings of the 20th international conference on Computational Linguistics*, 64-70.
- Okur E, Atlas M, 2020, Araç Rotalama Probleminin Genetik Algoritma ile Çözümü, *Anadolu Üniversitesi Sosyal Bilimler Dergisi*, 20, 227-254.
- Osman Ü, Uygunođlu T, 2005, Seyitömer uçucu külünün betonun basınç dayanımına etkisi üzerine bulanık mantık yaklaşımı, *Yapı Teknolojileri Elektronik Dergisi*, 1, 13-20.
- Özdemir O, Kalıncara Y, 2020, Bulanık Mantık: 2000-2020 Yılları Arası Tez ve Makale Çalışmalarına Yönelik Bir İçerik Analizi, *Acta Infologica*, 4, 155-174.
- Özkan İ A, Sarıtaş İ, ve Herdem S, 2011, Manyetik Filtreler için FPGA Tabanlı Bulanık Kontrolör Tasarımı, *Selçuk Teknik Dergisi*, 10, 271-284.
- Patnaik A, Choudhury B, Pradhan P, Mishra R K, Christodoulou C, 2007, An ANN Application for Fault Finding in Antenna Arrays, *IEEE Transactions on Antennas and Propagation*, 55, 775-777.

- Paukštaitis V, Dosinas A, 2009, Pulsed Neural Networks for Image Processing, *Elektronika Ir Elektrotechnika*, 95,15-20.
- Sahin I, 2010, A 32-bit floating-point module design for 3D graphic transformations SRE, 5, 3070-3081.
- Sahin I, Koyuncu I, 2012, Design and Implementation of Neural Networks Neurons with RadBas, LogSig, and TanSig Activation Functions on FPGA, *Elektronika Ir Elektrotechnika*, 120, 51-54.
- Savran İ, 2017, Donanım Tanımlama Dili VHDL ve FPGA Uygulamaları, Papatya Yayıncılık Eğitim, 320s, İstanbul.
- Schank R C, Goldman N, Riager C J, Rissbeck C, 1973, Margie memory, analysis, response generation, and inference on English, IJCAI'73: Proceedings of the 3rd international joint conference on Artificial intelligence, 255-261.
- Schütze H, Manning C D, 1999, Foundation of Statistical Natural Language, MIT Press, 620p.
- Şeker H İ, 2019, IQ-Math sayı standartlarında FPGA-tabanlı kaotik osilatörün tasarımı ve gerçekleştirilmesi, Afyon Kocatepe Üniversitesi, Fen Bilimleri Enstitüsü, Yüksek Lisans Tezi, Afyon.
- Selvathi D, Nayagam R D, 2016, FPGA implementation of on-chip ANN for breast cancer diagnosis, *Intelligent Decision Technologies*, 10, 341-352.
- Sivanandam S N, Sumathi S, Deepa S N, 2007, Introduction to fuzzy logic using MATLAB, Springer-Verlag Berlin Heidelberg, 430p.
- Swartz J, Koziatek C, Theobald J, Smith S, Iturrate E, 2017, Creation of a simple natural language processing tool to support an imaging utilization quality dashboard, *International Journal of Medical Informatics*, 101, 93-99.
- Şahin İ, 2008, Uzman Sistem Kullanarak İki Boyutlu İzdüşümlerden Katı Modeller Oluşturma, Gazi Üniversitesi, Fen Bilimleri Enstitüsü, Doktora Tezi, Ankara.
- Şahin İ, Temür G, 2016, Yapay sinir ağlarının otomatik olarak FPGA çipine uygulanması için otomatik denetleyici tasarım aracı, *SDU International Journal of Technological Science*, 8, 34-52.
- Şen Z, 2012, Fuzzy Philosophy of Science, *Journal of Higher Education and Science*, 2, 20-24.

- Tanguy L, Tulechki N, Urieli A, Hermann E, Raynal C, 2015, Natural language processing for aviation safety reports: From classification to interactive analysis, *Computers in Industry*, 78, 80-95.
- Tarcan A, Çakar F, 2008, Bilgisayarlı Dil Tanımlamada Dilbilimsel Yaklaşımlar ve Bir Yazılım Denemesi, *Elektronik Sosyal Bilimler*, 7, 64-70.
- Tasdemir M F, Koyuncu I, Cosgun E, Katircioğlu F, 2020, Real-Time Fast Corner Detection Algorithm Based Image Processing Application on FPGA, *International Asian Congress on Contemporary Sciences-III*, 1-6.
- Temurtaş F, 2007, Bulanık Sistemler, *Sinirsel Bulanık Sistemler Ders Notları*, Sakarya Üniversitesi.
- Toktaş İ, 2003, Mekanik sistemler için bir kavramsal tasarım modelinin geliştirilmesinde yapay sinir ağlarının kullanılması, Gazi Üniversitesi, Fen Bilimleri Enstitüsü, Doktora Tezi, Ankara.
- Tuna M, Alçın M, Koyuncu I, Fidan C M, Pehlivan I, 2019, High speed FPGA-based chaotic oscillator design, *Microprocessors and Microsystems*, 66, 72-80.
- Tuntas R, 2015, The Modeling and Hardware Implementation of Semiconductor Circuit Elements by Using ANN and FPGA, *Acta Physica Polonica Series a*, 128(2B), 78-81.
- Vaidyanathan S, Pehlivan I, Leutcho D S, Jacques K, Alçın M, Tuna M, vd., 2020, A novel ANN-based four-dimensional two-disk hyperchaotic dynamical system, bifurcation analysis, circuit realisation and FPGA-based TRNG implementation, *International Journal of Computer Applications in Technology*, 62, 20–35.
- Wachter M, Ovchinnikova E, Wittenbeck V, Kaiser P, Szedmak S, Mustafa W, vd., 2018, Integrating multi-purpose natural language understanding, robot's memory, and symbolic planning for task execution in humanoid robots, *Robotics and Autonomous Systems*, 99, 148-165.
- Wang G, Liu K, Sun Z, Li Y, 2020, Attitude-Orbit Cooperative Control for Small-Scale Spacecraft with Chemical Propulsion: Adaptive Sliding Mode Control Based on Neural Network, *Journal of Aerospace Engineering*, 33, Artical number 04020080.
- Wilensky R, 1977, PAM: a program that infers intentions, *IJCAI'77: Proceedings of the 5th international joint conference on Artificial intelligence*, 15.

- Woods W, 1970, Transition network grammars for natural language analysis, Communications of the ACM, 13, 587-634.
- Yıldırım M, 1998, Bulanık Mantık Yapay Sinir Ağı ile Doğrusal Olmayan Sistem Modelleme, Kocaeli Üniversitesi, Fen Bilimleri Enstitüsü, Yüksek Lisans Tezi, Kocaeli.
- Yılmaz A, 2014, FPGA Üzerinde Diferansiyel Gelişim Algoritması İle Yapay Sinir Ağı Eğitimi, Yıldız Teknik Üniversitesi, Fen Bilimleri Enstitüsü, Yüksek Lisans Tezi, İstanbul.
- Yılmaz C, Koyuncu I, Alçın M, Tuna M, 2019, Artificial Neural Networks based thermodynamic and economic analysis of a hydrogen production system assisted by geothermal energy on Field Programmable Gate Array, International Journal of Hydrogen Energy, 44, 17443-17459.
- Zadeh L A, 1965, Information and control, Fuzzy sets, 8, 338-353.
- Zadeh L A, 1965, Fuzzy Algorithms. Information and Control, 12, 94-102.
- Zhang F, Fleyeh H, Wang X, Lu M, 2019, Construction site accident analysis using text mining and natural language processing techniques. Automation in Construction, 99, 238-248.
- Zorlu H, Sunca Ş, 2017, Genetik Algoritma Kullanılarak Ağırlıklandırılmış Myriad Filtrelerin Optimizasyonu. International Journal of Multidisciplinary Studies and Innovative Technologies, 1, 9-14.

İnternet Kaynakları

- 1- <https://sozluk.gov.tr>, 07.04.2021
- 2- <https://www.wsj.com/articles/SB10001424052970203911804576653530510986612>, 08.04.2021
- 3- <https://hardwarebee.com/fpga-vs-gpu-choose>, 12.04.2021

ÖZGEÇMİŞ

Adı Soyadı : Mehmet Şamil AKÇAY
Doğum Yeri ve Tarihi : SAMSUN/13.02.1995
Yabancı Dili : İngilizce
İletişim (E-posta) : mehmentsamilakcay55@gmail.com

Eğitim Durumu (Kurum ve Yıl)

Lise : Samsun Şehit İlhan Hamlı Mesleki Ve Teknik Anadolu Lisesi, (2009-2013)
Lisans : Yozgat Bozok Üniversitesi, Elektrik-Elektronik Mühendisliği Bölümü, (2013-2018)
Yüksek Lisans : Afyon Kocatepe Üniversitesi, Fen Bilimleri Enstitüsü, Elektrik Elektronik Mühendisliği Anabilim Dalı, (2018 –2021)

Projeler:

1- YSA Uygulamaları için FPGA Üzerinde IQ-Math-tabanlı Aktivasyon Fonksiyonları Kütüphanesinin Gerçeklenmesi, Proje No: 19.FEN.BİL.14, Afyon Kocatepe Üniversitesi BAP Projesi, Araştırmacı, 2019.

Akademik Çalışmalar:

1- Koyuncu I, Akcay M S, Tuna M, Alçın M, 2019, Implementation of IQ-Math-based Linear Activation Functions on FPGA, 1st International Congress of Multidisciplinary Studies and Research, 114-124.

2- Akcay M S, Koyuncu I, Alçın, M, and Tuna M, 2020, IQ-Math Tabanlı RadBas Aktivasyon Fonksiyonunun FPGA Üzerinde Gerçeklenmesi, International Asian Congress on Contemporary Sciences-IV, 599-607.