

# FPGA Tabanlı Aritmi Sınıflandırıcı

## FPGA Based Arrhythmia Classifier

Ahmet Turan Özdemir<sup>1</sup>, Kenan Danışman<sup>1</sup>, Musa Hakan Asyali<sup>2</sup>

1. Elektrik-Elektronik Mühendisliği Bölümü  
aturan@erciyes.edu.tr, danismak@erciyes.edu.tr  
Erciyes Üniversitesi

2. Biyomedikal Mühendisliği Bölümü  
asyali@erciyes.edu.tr  
Erciyes Üniversitesi

### Özetçe

*EKG (Elektro KardiyoGram) kayıtlarının yazılım tabanlı uzman sistemler tarafından yorumlanması 1960'lı yıllara dayanmaktadır. Son 20 yıldır bu konuda birçok teknik üzerine çalışmalar yapılmıştır. Fakat güçlü tahminsel yeteneğine olan inanç sebebiyle, medikal tanı destek sistemi uygulamalarında kullanılan en popüler yöntem YSA (Yapay Sinir Ağları) olmuştur. Literatürde önerilen YSA modelleri, çok karmaşık yazılım tabanlı çözümlerdir ve bunlar gerçek zamanlı çalışmazlar. Bu tür yapıların donanım gerçeklemeleri ise ancak pahalı işlemciler üzerinde yapılabilmektedir. YSA donanım gerçeklemelerini taşınabilir ucuz cihazlar üzerinde oluşturmayı mümkün kılmak için, YSA girişine uygulanan EKG işaretinin özellik sayısı azaltılmalıdır. Bu sayede daha az sayıda bilgi ile daha basit bir mimari oluşturmak mümkün olacaktır. Bu çalışmada YSA'nın öğrenme hatası %5 gibi kabul edilebilir bir seviyede tutularak, TBA (Temel Bileşen Analizi) yöntemi kullanılarak EKG giriş işareti özellik sayısı önemli ölçüde azaltılmıştır. Bu sayede 8x2x1 boyutlu basit bir Matlab YSA modeli FPGA (Alan Programlanabilir Kapı Dizileri) donanımı üzerinde IEE 754 32 bit kayan noktalı nümerik sayı formatı kullanılarak gerçekleştirilmiştir.*

**Anahtar Sözcükler:** EKG, FPGA, YSA, TBA

### Abstract

*Processing of ECG (Electro CardioGram) records by software- based systems was started in the beginning of the 1960s. Many studies on different techniques about this topic have been made in the last 20 years. However, ANN (Artificial Neural Network) is the tool that is mostly used in medical diagnosis systems because of the belief in its powerful prediction characteristics. However, the suggested ANN architectures in literature are very complex software-based architectures. Consequently, these models with high computational complexity can only be run on expensive processors. To enable the implementation of ANN models on mobile and cheap devices, the features of ECG signal, which are applied to ANN inputs, should be reduced. This approach enables the implementation of a simple ANN architecture. In this study, the features of ECG signal are reduced dramatically using PCA (Principle Component Analysis), while keeping the error of the ANN learning rate at an acceptable level such as 5%. As a result, a simple Matlab ANN model, which consists of eight inputs, a hidden layer with two neurons and one output neuron, is implemented on an FPGA (Field Programmable Gate Arrays) by using IEE 754 32 bits floating-point numerical representation.*

**Keywords:** ECG, FPGA, ANN, PCA

### 1. Giriş

Kalp hastalıklarının teşhis ve tedavisinde başvurulan en önemli referans Elektro-KardiyoGram (EKG) işaretidir. EKG işaretlerinden aritmilerin belirlenmesi ve incelenmesi için hekimlere yardımcı birçok yazılım kullanılmaktadır. Bu yazılımların kullanıma sebebi çok uzun kayıtlarda gözden kaçan noktaların belirlenmesi, yorgunluk, stres ve telaş gibi insan kaynaklı hataların telafi edilebilmesidir. Bu yazılım tabanlı çözümler ile ilgili çalışmalar 1960'lı yılların başında başlamıştır.

Son yirmi yıl içerisinde EKG işaretlerinin analizinde yeni birçok teknik önerilmiştir. Örneğin; Yapay Sinir Ağları (YSA) [1-3], Genetik Algoritma (GA) [4], Dalgacık Dönüşümü (DD) [5] ve çoğu lineer olmayan dönüşüm tabanlı sezgisel yöntemler. Fakat güçlü tahminsel yeteneğine olan inanç sebebiyle, medikal tanı destek sistemi uygulamalarında kullanılan en popüler yöntem YSA'dır [6]. Birçok araştırmacı, mevcut modellere göre YSA'nın hassasiyetini karşılaştırmalı olarak vermiş ve YSA'nın daha yüksek bir hassasiyet ile sonuçlar ürettiğini göstermişlerdir [7].

YSA aritmi sınıflandırmada kullanılan en yaygın yöntemlerin başında gelmesine rağmen, literatürde verilen örnekleri onlarca hatta yüzlerce giriş ve çok sayıda ara katmana sahip, işlemsel karmaşıklığı yüksek yapılarıdır. Bu tür yapılar ancak çevrimdışı olarak pahalı işlemciler üzerinde gerçekleştirilebilmektedir. Bu yapıların paralel çalışan mimariler olarak donanım üzerinde gerçekleştirilebilmesini mümkün kılmak için ilgili YSA mimarisinin boyutlarının küçültülmesi gerekmektedir. YSA boyutlarının küçülmesi için yapılacak en temel işlem, ilgili YSA modelindeki giriş sayısının azaltılmasıdır. Bu çalışmada, EKG işareti ham veri olarak alınıp, örnekleme işlemi sonunda elde edilen bilgiler daha az veri ile tanımlanmıştır. Bunun için de özellik çıkarım tekniklerine başvurulmuştur.

Vargas ve arkadaşları [8] TBA analizi yöntemi ile EKG işaretlerinden özellik çıkarımı yapmış ve 16 özellik ile ilgili EKG işaretini büyük bir doğrulukla ifade edebilmişlerdir. Önceki çalışmamızda TBA analizi yöntemi ile boyutları küçültülmüş YSA mimarileri oluşturulmuştur [9]. Ortaya konulan bu YSA modelleri ile %2-5 arasında kabul edilebilir bir hata ile sınıflandırma yapılmıştır. Yapılan bu çalışmada ise, boyutları azaltılmış bu yapılardan, sekiz giriş, iki nöronlu bir ara katman ve bir çıkış nöronundan oluşan bir YSA mimarisi FPGA üzerinde donanım olarak gerçekleştirilmiştir.

## 2. YSA Donanım Modellemesi

YSA insan beyindeki biyolojik yapının öğrenmesinden esinlenerek tasarlanmış bir hesaplama sistemidir. İnsan beyinde ortalama 100 milyar nöron ve 100 trilyon bağlantı bulunmaktadır. Bilgiler dağıtık olarak nöronların gövdelerinde bulunur ve bu dağıtılmış veriler sistem içerisinde paralel olarak işletilmektedir. Fakat YSA uygulamalarının büyük çoğunluğu yazılım tabanlı çözümlerdir ve sıralı işlem gören makineler üzerinde yapılmaktadır. Yazılım tabanlı YSA'lar da kullanıcı için ilgili mimarinin kaynak kullanımı veya hesaplamasal karmaşıklığı bir problem teşkil etmemektedir. İşlemler belli bir sıra dahilinde yapılır ve sonuçlar çevrimdışı olarak yorumlanır. Bu yapıları donanımlar üzerinde paralel olarak işletmek için 1980 ve 1990'lı yıllarda gerek akademi gerekse teknoloji çevreleri tarafından birçok çalışma yapılmış fakat bu çalışmalarda genel anlamda başarı sağlanamamıştır [10].

Donanımsal YSA modelleri temel anlamda analog ve sayısal model olarak iki farklı yapıda incelenmiştir. Analog yapılarda aktivasyon fonksiyonları CMOS yarıiletken karakteristiği ile kolayca oluşturulabilmiştir. Fakat bu yapılarda hassasiyet, sıcaklıkla değişen sistem cevabı ve programlanamama gibi dezavantajlar vardır. Sayısal yapılar ASIC ve FPGA yongaları üzerinde oluşturulmaktadır. Bu yapılar ilk yıllarda sınırlı donanım kaynakları sebebi ile verimsiz olmalarına karşın, günümüz teknolojisi sayesinde daha etkili bir şekilde kullanılmaktadır. Sayısal donanımlarda en büyük problem aktivasyon fonksiyonlarının oluşturulmasıdır. FPGA, programlanabilme özelliği, paralel işlem ve ortaya konulacak yapının güçlü hesaplama yeteneği sayesinde YSA uygulamaları için çok uygun bir platformdur.

FPGA teknolojisindeki gelişmelere rağmen, yazılım tabanlı YSA mimarileri bu yongalar için hala çok büyük kaynak kullanan yapılardır. Bu yüzden ilgili YSA yapıları daha az kaynak kullanır şekilde yeniden tasarlanmalıdır. Böylelikle indirgenmiş boyutlu YSA yapıları FPGA yongaları üzerinde oluşturulabilir. YSA boyutu ve ürettiği hata oranı arasında bir denge bulunmaktadır. Fakat daha önceleri düşünülen, bir YSA mimarisinde boyut büyüdükçe üretilen hata oranı azalır inancı doğru değildir. 1990'larda CAM-Brain projesi ile YSA zekâsının, sadece YSA boyutu ile tanımlanamayacağı ortaya konulmuştur [11, 12]. Bugün YSA araştırmacıları tarafından ortaya konulan genel fikir, YSA başarımında ağ mimarisi ve iç bağlantıların, önemli birer parametre olduğudur. Bunun anlamı küçük boyutlu bir YSA mimarisi ile de kabul edilebilir bir hata ile sonuçların üretilebilmesidir.

## 3. Materyaller Ve Yöntemler

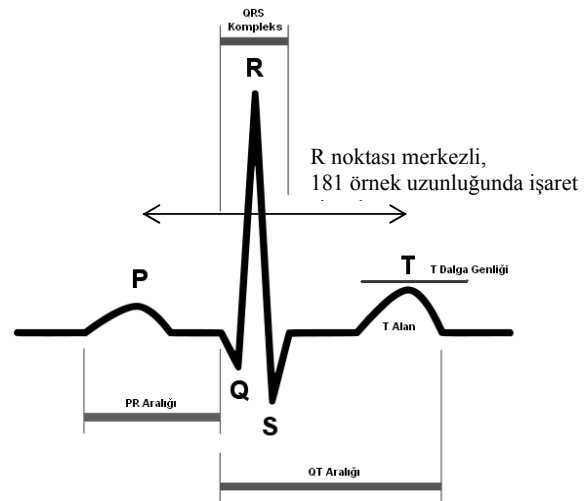
Bu çalışmada MIT BIH [13] veritabanından alınan kayıtlı EKG veri dosyaları kullanılmıştır. Bu dosyalarda 360 örnek/saniye hızında, 10 mV aralığında, 11 bit çözünürlükte sayısallaştırılmış, 30 dakika uzunluğunda, 2 kanal EKG verisi vardır. Bu veri tabanındaki dört veri dosyası (No: 205, 208, 210 ve 213) için üç sınıf veri kullanılmıştır. Bu sınıflar Normal (N), Fusion (F) ve Erken Karıncık Kasılması'dır (PVC, Premature Ventricular Contraction). İlgili aritmi tipleri .atr uzantılı başka bir dosyada tutulmaktadır. Bu dosyada aritmilerin tipi, QRS işaretinde R noktasının karşılık geldiği

yere uzman doktor tarafından işaretlenmiştir. Bu dosyada hem zaman bilgisi hem de aritmi sınıfının tipi kayıtlıdır. Matlab ortamında yapılan işlemlerde QRS tanıma işlemi yapılmamaktadır. Çünkü referans R noktaları .atr uzantılı dosyadan çekilmektedir. EKG işaretleri insandan alındığı zaman bu durumda ayrıca QRS tanıma işleminin de ayrıca bir ön-işlem olarak yapılması gerekecekti.

Bu çalışmada EKG işaretlerinden aritmi tiplerini ayırt eden bir YSA mimarisi FPGA üzerinde donanımsal olarak oluşturulmuştur. Bu amaçla önce referans YSA modeli Matlab ortamında modellenmiştir. Matlab ortamında yapılan modellemede, kabul edilebilir bir hata ile çalışan (%2-%5), küçük boyutlu bir YSA mimarisi oluşturmak hedef alınmıştır.

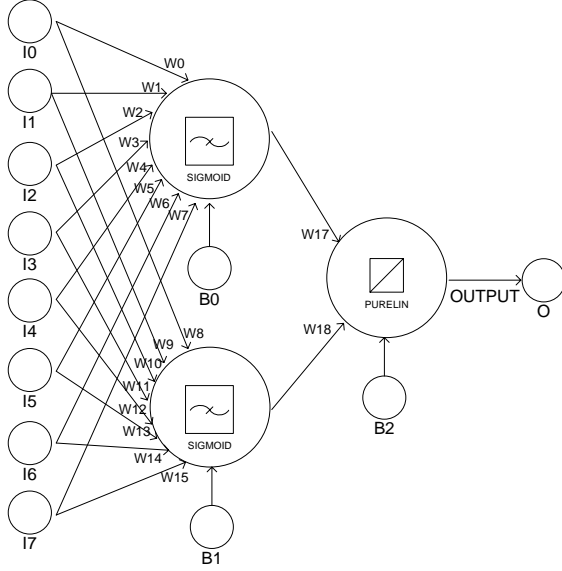
Literatürdeki bazı çalışmalarda [14] özellik çıkarma aynı zamanda boyut azaltma yöntemi olarak temel bileşen analizi (TBA) kullanılmıştır. TBA yönteminde verideki değişkenlik (varyans) öznitelik çıkarma metoduyla incelenir ve bu değişkenlik büyükten küçüğe doğru sıralanacak şekilde yeni bir eksen sisteminde ifade edilir. Böylece bu yeni eksen sisteminde değişkenliğin büyük olduğu bileşenler alınıp diğer bileşenler atılarak orijinal verideki bilgiyi daha az sayıda özellikle aktarmak mümkün olur [15].

EKG kayıtları QRS bileşke vurusu ile yorumlanmaktadır. QRS bileşke vurusu Şekil 3.1'de gösterilmektedir. Şekilden de görüldüğü üzere QRS işareti R tepesi ve bunun sağından ve solundan alınan 90 örnek ile toplam 181 örnek olarak ifade edilmiştir. 181 örnek ile alınmış her bir QRS vurusunun uygulanacağı ağın 181 girişi ile ifade edilmesi beklenmektedir. Fakat bu giriş sayısı ilgili ağ modellemesinin FPGA üzerinde oluşturulmasını imkansız hale getirir. 181 örnek uzunluğundaki QRS bilgisi, Vargas ve arkadaşları tarafından TBA yöntemi ile 16 işaret bileşeni ile ifade etmiştir [8]. TBA yöntemi kullanılarak yapılan indirgemedeki farklı boyutlu birçok YSA modeli oluşturulmuş ve bunların başarımları bir önceki çalışmamızda incelenmişti [9]. Şekil 3.2'de gösterilen %4.6 hata ile sonuç üretebilen, 8x2x1 boyutlu YSA modeli bu çalışma için referans model olarak seçilmiştir.



Şekil 3.1. Bir EKG kaydına ait normal bir QRS kompleks vurusu.

Şekil 3.2’de verilen YSA modeli IEEE 754 32 bit kayan noktalı nümerik sayı formatını kullanmaktadır. Ağın 8 adet giriş verisi I[0..7] TBA analizi sonucu üretilen her bir QRS kompleks vurusuna karşılık gelen 8 özelliştir. Ağırlık W[0..18] ve bias B[0..2] değerleri Matlab tarafından yapılan eğitim sonucu elde edilmiş değerlerdir ve bilgiler FPGA içerisindeki hafızada saklanmaktadır. İlgili ağı eğitimi LM (Levenberg Marquardt) algoritmasıyla [16] yapılmıştır. Ara katmanda yer alan nöronların aktivasyon fonksiyonları sigmoid çıkış katman nöronunun aktivasyon fonksiyonu ise purelin’dir



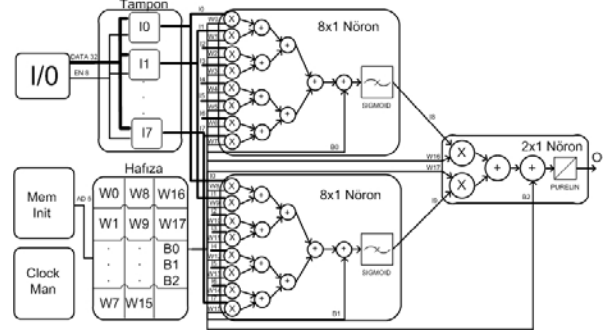
Şekil 3.2. 8x2x1 boyutlu ileri beslemeli YSA modeli

Bu çalışmada ortaya konulan YSA modeli 3 sınıf (F, PVC, ve N) veri için sırasıyla 1, 2 ve 3 çıkış değerlerini (ÇD) üretmeyi hedeflemektedir. ÇD sürekli değişkeninin F (1), PVC (2) veya N (3) olacak şekilde 3 sınıfa ayrılması Denklem 1’de verilen fonksiyona göre yapılmıştır.

$$\text{Sınıf kararı} = \begin{cases} F, & \text{ÇD} \leq 1.5 \\ V, & 1.5 < \text{ÇD} \leq 2.5 \\ N, & 2.5 < \text{ÇD} \end{cases} \quad (1)$$

Şekil 3.3’de 8x2x1 boyutlu yapının FPGA gerçekleştirilmesi blok diyagramı olarak verilmiştir. YSA içerisindeki bütün veriler 32 bit kayan noktalı nümerik formattadır. Clock Man bloğu bütün bir ağı paralel olarak işletilmesinden sorumludur. Şekildeki hafıza bloğu YSA’nın ağırlık ve bias değerlerinin saklandığı yerdir. FPGA yongası ilk enerjilendiğinde bu değerler YSA’nın ilgili birimlerine atanır. Bu değerler eğitim sonucunda elde edilen verilerdir. Hafızadaki verilerin ilgili birimlere atanmasını Mem Init bloğu kontrol etmektedir. I/O bloğu Matlab tarafından üretilmiş olan 8 özellikli TBA sonuçlarını FPGA girişine taşımaktadır. Bu veriler EN işareti ile YSA girişlerine taşınmaktadır. Tampon bloğu içerisindeki I[0..7] birimleri veri yolundaki bilgiyi ancak ilgili EN girişleri aktif olduğu zaman alırlar. Böylece veri yolundaki bilgi doğru girişlere taşınmış olur.

Şekil 3.3’deki YSA bloğunun ağırlık, bias değerleri atandıktan sonra, giriş verileri I/O bloğu tarafından üretilir. Bu giriş verileri 8x1 Nöron bloklarında 16 farklı ağırlık değeri ile çarpılır ve bu çarpımların toplamları ilgili bias değerleri ile toplanır. Sonuçlar sigmoid fonksiyonundan geçirilerek 2x1 Nöron bloğu girişine uygulanır. 2x1 Nöron bloğunun aktivasyon fonksiyonu purelin’dir.



Şekil 3.3. 8x2x1 boyutlu ileri beslemeli YSA modeli

Bu çalışmada Altera Quartus II 8.1 sentez editörü kullanılmıştır. YSA modeli Cyclone III EP3C120F780C8 FPGA yongası üzerinde gerçekleştirilmiştir. Yapının donanım kaynak kullanımları Tablo 3.1’de verilmiştir. YSA içerisinde en büyük kaynak kullanımına sebep olan yapı sigmoid fonksiyonudur.

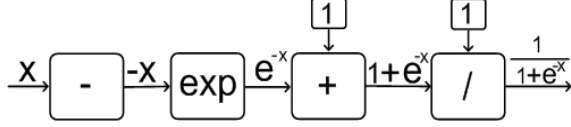
Tablo 3.1. 8x2x1 boyutlu YSA modelinin kaynak kullanımı

Toplam Lojik Eleman	23104/119088	%19
Toplam Kombinasyonel Fonksiyon	22259/119088	%19
Toplam Flip Flop	10816/119088	%9
Toplam Hafıza	12946/3981312	%1
9 Bit Gömülü Çarpıcı	220/576	%38

FPGA üzerinde YSA oluşturmada en zor kısım aktivasyon fonksiyonunun oluşturulmasıdır [17]. Basamak, rampa ve sigmoid fonksiyonları kaynak kullanımı bakımından sırası ile en az olanı basamak, sonra rampa ve en çok kaynak kullanan sigmoid fonksiyonudur [18]. Rampa ve eşik fonksiyonlarının sonuçları oldukça keskindir. Sigmoid fonksiyonunun çıkışı yumuşak geçişlidir ve biyolojik nöron modeline en yakın olan cevabı üretir [19]. Sigmoid fonksiyonunun gerçekleştirilmesi için literatürde birçok yöntem kullanılır. Örneğin Parçalı Lineer (PL) yaklaşım, Taylor serisi açılımı ve LUT yöntemleri gibi. Sonuçlar belirli hata oranları ile üretilmiştir [sig, imp, ]. [20] çalışmasında Taylor serisi açılımı kullanılmıştır ve sigmoid fonksiyonunun hata değeri diğer yaklaşımlara göre düşük olmasına karşın, çok sayıda çarpıcı eleman kullandığı için çok fazla silikon alanı işgal etmektedir. [21] çalışmasında parçalı lineer yaklaşımı ile sigmoid fonksiyonu gerçekleştirilmiştir. Bu gerçekleştirilmede silikon alan kullanımı oldukça düşük olmasına karşın sigmoid fonksiyonunun doğruluğu önemli ölçüde azalmıştır.

Bu çalışmada kullanılan sigmoid fonksiyon gerçekleştirilmesi literatürdeki benzerlerinden daha fazla silikon alanı kullanmasına rağmen, gerçek sigmoid fonksiyonu cevabını hatasız olarak üretmektedir. Sigmoid fonksiyonunun gerçekleştirilmesinde Altera Quartus II Mega Function

kütüphanesinin expfpt işlem bloğu kullanılmıştır. Bu blok 32 bit kayan noktalı nümerik formatta  $\exp(x)$  işlemini gerçekleştirmektedir. Sigmoid fonksiyonu çalışma blok diyagramı Şekil 3.4'de verilmiştir. exp bloğu sayesinde lineer olmayan aktivasyon fonksiyonunun matematiksel tanımı tam olarak yapılabilmektedir. Bu sayede literatürdeki uygulamalardan daha iyi sonuçlar üretilmiştir.



Şekil 3.4. Sigmoid fonksiyonu donanım blok diyagramı

YSA mimarisinin oluşturulmasında hassasiyet ve kaynak kullanımı arasında bir orantı vardır. Hassasiyet arttıkça FPGA üzerinde kullanılan alan da artmaktadır. Tek ve çift hassasiyetli kayan noktalı formatlar nümerik hassasiyeti artırmaktadır fakat bu yapılar FPGA içerisindeki kaynak kullanımını da artırmaktadır. Bunun yanında sabit noktalı nümerik sayı formatı daha az kaynak kullanmasına rağmen kuantalama hataları öğrenmeyi zorlaştıracak ve hatalı sonuçlar üretilmesine sebep olacaktır. [22] çalışmasına göre 32 bit kayan noktalı nümerik format yerine 16 bit sabit noktalı nümerik tanımlama kullanılmış ve 12 kat daha hızlı ve 13 kat daha küçük bir alan kullanılmıştır [23]. FPGA içerisinde sabit noktalı nümerik yapının kullanılması daha fazla sayıda nöron oluşturulmasına müsaade etmektedir fakat kayan noktalı uygulamada ağ hassasiyeti başarıyı daha yüksek olmaktadır. Kayan noktalı nümerik format, sunmuş olduğu yüksek hassasiyet sebebi ile daha büyük alan kullanımına rağmen YSA yapılarında kullanılmaktadır [24].

#### 4. Sonuç Ve Tartışma

Bu çalışmada ortaya konulan FPGA tabanlı YSA modelinin gerçekleştirimi 32 bit kayan noktalı nümerik tanımlama ile yapılmıştır. Bu format sistemin nümerik hassasiyetini arttırmaktadır. Sistemin nümerik hassasiyetini korumak amacı ile sigmoid aktivasyon fonksiyonu da parçalı lineer veya LUT yöntemlerine başvurulmadan gerçekleştirilmiştir. Bu çalışmada sunulan aktivasyon fonksiyonu literatürdeki çalışmalarda sunulmuş olan bütün sigmoid fonksiyonlarından daha büyük alan kaplamasına karşın en iyi hassasiyetle sonuç üretmektedir. Bütün ağıın ürettiği sonuçlar Matlab Simulink ortamında tasarlanmış olan model ile karşılaştırılarak test edilmiştir. Sonuçlar sistemin %0 hata ile çalıştığını göstermiştir. Yapılan bu çalışmada elde edilen sonuçlar, acil ve yoğun bakım ünitelerinde rahatlıkla kullanılabilir, hızlı, düşük enerji tüketimli, ucuz ve taşınabilir özellikte cihaz tasarımına bir örnek teşkil etmektedir.

#### 5. Kaynakça

[1] Hu, Y. H.; at all, "Applications of Artificial Neural Networks for ECG Signal Detection And Classification," J. of Electrocardiology, Vol. 26, Page: 66-73, 1993  
[2] Strintzis, M.G.; Stalidis, G.; Magnisalis, X.; Maglaveras, N., "Use of Neural Networks for Electrocardiogram (ECG) Feature

Extraction, Recognition and Classification" Neural Netw. World, Vol. 3, No. 4, Page: 313-327, 1992.  
[3] Vijaya, G.; Kumar, V.; Verma, H.K., "ANN-Based QRS-Complex Analysis of ECG," J. Med. Eng. Technol., Vol. 22, No. 4, Page: 160-167, 1998.  
[4] Poli, R.; Cagnoni, S.; Valli, G., "Genetic Design Of Optimum Linear And Nonlinear QRS Detectors," IEEE Trans. Biomed. Eng., Vol. 42, Page: 1137-1141, 1995  
[5] C. Li, C. Zheng, ve C. Tai, "Detection of ECG Characteristic Points Using Wavelet Transforms," IEEE Trans. Biomed. Eng., Vol. 42, Page: 21-28, 1995.  
[6] D. West, V. West, Improving Diagnostic Accuracy Using A Hierarchical Neural Network To Model Decision Subtasks, Int. J. Med. Inform., Vol. 57 (1), Page: 41-55, 2000.  
[7] Y. Hayashi, R. Setiono, Combining Neural Network Predictions For Medical Diagnosis, Com. Bio. Med. Vol 32, 237-246, 2002.  
[8] Vargas, F.; at all, "Electrocardiogram Pattern Recognition by Means of MLP Network and PCA", Neural Networks, Proceedings. VII Brazilian Symposium, Page: 200-205, Nov. 2002.  
[9] Asyalı M. H.; Özdemir, A. T.; Aksebzeci, B. H.; Danişman K.; "Aritmi Sınıflandırmada Kullanılan Yapay Sinir Ağlarının Karmaşıklık Azaltılması", BIYOMUT 2008, ODTÜ, Ankara, Türkiye, Mayıs 2008  
[10] Tisan, A.; Oniga, S.; Gavrinca, C.; "Hardware implementation of a MLP network with on-chip learning", Proceedings of the 5th WSEAS Int. Conf. on Communications & Computers, Romania, October 16-17, Page: 162-167, 2006.  
[11] H. de Garis, An Artificial Brain: ATP's Cam-Brain Project Aims to Building/Evolve an Artificial Brain with a Million Neural Net Modules Inside a Trillion Cell Cellular Automata Machine, New Generation Computing Journal, Vol. 12, No. 2, July 1994.  
[12] N. Nawa, M. Korin; H.de Garis, "ATP's CAM-Brain Project: The Evolution of Large-Scale Recurrent Neural Network Modules," in 1998 International Conference on Parallel and Distributed Processing Techniques and Application, 1998.  
[13] MIT-BIH ECG Database Home Page, <http://ecg.mit.edu/>  
[14] Jain, A.K.; Jianchang Mao; Mohiuddin, K.M.; "Artificial neural networks: a tutorial", IEEE Computer Society, Volume 29, Issue 3, March 1996 Page: 31 - 44.  
[15] Özdamar K., "Paket Programlar ile İstatistiksel Veri Analizi", Kaan Kitapevi, 2004.  
[16] Donald Marquardt, "An Algorithm for Least-Squares Estimation of Nonlinear Parameters". SIAM Journal on Applied Mathematics 11: 431-441, 1963.  
[17] Niculita, L.; George Carutasu, "Governance Intelligence for Romanian R&D Funding", Proceedings of the 5th WSEAS Int. Conf. on Data Networks, Communications & Computers, Athens, Greece, October, Page: 182-187, 2006  
[18] Banuelos, M.; Hernandez, J.; Thierry, S., "Implementation of a Neuron Model Using FPGAs", Journal of Applied Research and Technology, Vol 1, 3, Page: 248-255, 2003  
[19] Xi Chen, Gaofeng Wang, Wei Zhou, Sheng Chang, Shilei Sun, "Efficient Sigmoid Function for Neural Networks Based FPGA Design", ICIC 2006, Kunming, China, August 16-19, 2006  
[20] Ouali, J.; Saucier, G., "Fast Generation of Neuron-ASICs.", Proc. Int. Neural Networks Conf, Page: 563-567, 1990  
[21] Murtagh, P., Implementation Issues of Sigmoid Function and Its Derivative for VLSI Digital Neural Networks. IEE Proceedings on Computers and Digital Techniques, 207-213, 1992  
[22] Moussa, M.; Areibi, S.; Nichols, K., "On The Arithmetic Precision For Implementing Back-Propagation Networks On FPGA: A Case Study, FPGA Implementations of Neural Networks", October, Page: 37-61, 2006  
[23] Earl, D.D., Development of An FPGA-Based Hardware Evaluation System for use with GA-Designed Artificial Neural Networks", PhD Thesis, The University of Tennessee, US, 2004.  
[24] Nichols, K.; Moussa, M.; Areibi, S., "Feasibility Of Floating-Point Arithmetic In FPGA Based Artificial Neural Networks", Lecture Notes in Computer Science, Aug, Page: 1084-86, 2004