

## Hücresel Yapay Sinir Ağı Hücresi Olarak Bir Relaksasyon Osilatörünün FPGA Üzerinde Donanım-Yazılım Ortak Tasarımı ve Gerçeklenmesi

Selman ERGÜNAY

*Elektronik ve Haberleşme  
Mühendisliği Bölümü  
Elektrik-Elektronik Fakültesi  
İstanbul Teknik Üniversitesi,  
Maslak, İSTANBUL  
selman.ergunay@itu.edu.tr*

Ramazan YENİÇERİ

*Elektronik ve Haberleşme  
Mühendisliği Bölümü  
Elektrik-Elektronik Fakültesi  
İstanbul Teknik Üniversitesi,  
Maslak, İSTANBUL  
ramazan.yeniceri@itu.edu.tr*

Müştak E. YALÇIN

*Elektronik ve Haberleşme  
Mühendisliği Bölümü  
Elektrik-Elektronik Fakültesi  
İstanbul Teknik Üniversitesi,  
Maslak, İSTANBUL  
mustak.yalcin@itu.edu.tr*

### Özet

*Bu bildiri, robot yönlendirme probleminde doğrusal olmayan uzay-zaman dalgaları ile çözüm üretmekte kullanılan bir hücresel doğrusal olmayan ağa ait hücrenin, FPGA üzerinde gerçekleştirilen yazılım-donanım ortak tasarımı ayrılaştırılmış bir gevşemeli osilatörün (relaksasyon osilatörünün) sayısal devre gerçekleştirilmesi iken yazılım parçası 32-bitlik mikroişlemci tarafından çalıştırılan ve bu devreyi kontrol eden C kodudur.*

### 1. Giriş

Yerel bağlantıya sahip ağ mimarilerinin tümdevre üzerinde gerçekleştirilebilirlikleri ile sinir hücresi temel alınarak kurulan doğrusal olmayan devrelerin yani yapay sinir ağlarının birleştirilmesi fikri ilk olarak 1988'de Chua ve Yang tarafından Hücresel Yapay Sinir Ağları (Cellular Neural Network) adı ile ortaya konmuştur [1]. Çalışmalar doğanın da bu keşfi desteklediğini göstermektedir. Canlılarda görsel bilginin ilk işlendiği yerin retina üzerindeki birden fazla katmandan oluşan hücresel mimarideki sinir ağı olduğu bilinmektedir [2]. Bu tür mimarideki sinir hücresi benzeri doğrusal olmayan devre ya da sistemlerin uzay-zaman dalgalarını üretebildiğinin gözlenmesi [3] ile bu tür ağların doğadakininden farklı uygulamaları üzerinde de araştırmalar sürdürülmüştür [4]. Bu araştırmalar ayrıca hücresel yapay sinir ağlarının analog ve sayısal devre gerçeklemeleri ile desteklenmiştir [5].

İki boyutlu ve üzerinde engeller bulunduran bir düzlemde bir robotun en kısa yol üzerinden hedefe yönlendirilmesi ile ilgili çalışmalar [6],[7], uzay-zaman dalgaları üretip yayabilen bir ağ modeli [8] kullanarak gerçekleştirilmiştir. Bu hücresel doğrusal olmayan ağ modeline ait öykünücüler (emülatörler) Sahada Programlanabilir Kapı Dizileri (FPGA) ile gerçekleştirilmiştir [9], [10].

Robot yönlendirmesi için geliştirilen algoritmanın gerçek zamanlı çalışabilmesi için ağı bilgisayar üzerinde benzetimi kullanılmıştır. Bilgisayar üzerinde robot yönlendirme sisteminin girişlerinin hazırlanması ve çıkışlarının kullanılabilmesi yazılım esnekliği sayesinde kolaylıkla sağlanmaktadır. Fakat bilgisayar üzerinde önerilen ağı benzetimi, geliştirilen ağ öykünücülerin başarımından daha uzun sürelerde yapılabilmektedir. Yazılım esnekliğine bu türlü çalışmalarda hep ihtiyaç duyulması sebebiyle çözüm olarak donanım ve yazılımın birlikte tasarımı yöntemleri kullanılır. Bu çalışmada da kullandığımız ağı oluşturan hücre öykünücülerinden biri donanım, onu denetleyip kullanacak yapı da yazılım olarak tasarlanmıştır.

Bu bildirinin ikinci bölümünde uzay-zaman dalgaları için kullanılan ağ ve her bir relaksasyon osilatörünün çalışması tanıtılmakta, üçüncü bölümünde tasarlanan sistem bileşenleri detaylı olarak açıklanmakta ve dördüncü bölümünde elde edilen sonuçlar ile bildiri sonlandırılmaktadır.

### 2. Gevşemeli Osilatör ve Bağlı Osilatörler Ağı

Yalçın'ın 2008'de önerdiği uzay-zaman dalgaları yayan hücresel doğrusal olmayan ağ modeli Denklem (1)'de verilmektedir.

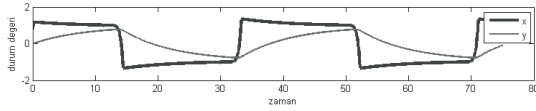
$$\begin{aligned} \dot{x}_{i,j} &= \alpha x_{i,j} + \beta y_{i,j} + g(x_{i,j}) + I_{i,j} + u_{i,j}, \\ \dot{y}_{i,j} &= \epsilon x_{i,j} + \sigma y_{i,j}. \end{aligned} \quad (1)$$

Hücre modelinde belirtilen  $g(\cdot)$  fonksiyonu doğrusal olmayan terimdir ve Denklem (2)'de belirtildiği gibi parça parça doğrusal bir ifadeye karşılık gelmektedir.

$$g(x_{i,j}) = \begin{cases} \mu \cdot (x_{i,j} - \lambda) & \text{if } x_{i,j} > \lambda; \\ 0 & \text{if } |x_{i,j}| \leq \lambda; \\ \mu \cdot (x_{i,j} + \lambda) & \text{if } x_{i,j} < -\lambda; \end{cases} \quad (2)$$

Ağ üzerindeki esasen gevşemeli osilatör davranışı sergileyen, ikinci mertebeden doğrusal

olmayan her bir dinamik hücrenin zaman içerisinde durum değişkenlerinin değişimi Şekil 1'de gösterildiği gibidir.

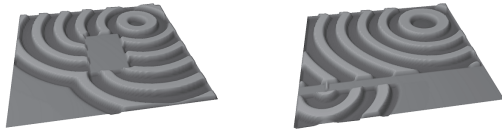


Şekil 1. Bir hücrenin salınımı.

Hücrelerin iki boyutlu ızgara düzeninde yerleştirilmesi ve dört ana komşuluk üzerinden birbirlerine bağlanması, Denklem (1)'deki x durum denkleminde Denklem (3)'te belirtilen I bağlantı teriminin eklenmesi ile oluşturulur.

$$I_{i,j} = a_{i,j+1}x_{i,j+1} + a_{i-1,j}x_{i-1,j} + a_{i,j-1}x_{i,j-1} + a_{i+1,j}x_{i+1,j} \quad (3)$$

Bu şekilde relaksasyon osilatörlerinden oluşturulan büyük boyutlu ağlarda uygun parametreler ve uygun başlangıç koşulları altında Şekil 2'de gösterildiği gibi otodalgalar (autowave) üretilip yayılabilmektedir.



Şekil 2. Otodalgaların ağ üzerinde yayılımı.

[7] numaralı çalışmada, uzay-zaman dalgalarının robot yönlendirmesine uygulanması açıklanmıştır. Yapılan bu çalışmada giriş işaretinin işlenmesi, ağ benzetimi ile bilgi üretimi ve üretilen bilgi ile robotun kumanda edilmesi için yazılım kullanılmıştır.

Giriş bölümünde de belirtildiği gibi bu çalışmada gevşemeli osilatör öykünümü gerçekleştiren bir özel donanım ve onun yazılım ile kontrolü gerçekleştirilmiş, bu sistemin tüm bileşenleri Bölüm 3'te anlatılmıştır.

### 3. Sistem

Sistem, Şekil 3'te görüldüğü gibi dört bloktan oluşmaktadır. Mikroişlemci bloğu, osilatör bloğunun denetimini bellekteki program uyarınca yapmakta ve kullanıcı ile haberleşmesini seri haberleşme kanalı üzerinden sağlamaktadır.

Tasarımda, mikroişlemci olarak Xilinx tarafından geliştirilen 32-bitlik ve RISC mimarisine sahip Microblaze işlemcisi kullanılmıştır. Bu işlemci hem FPGA içerisindeki blok RAM'leri hem de harici belleği desteklemektedir. Kontrol edilecek blok RAM boyutu sistem oluşturulurken belirlenir ve kullanılan FPGA'da en fazla 64 KB olabilmektedir. Bu durum yalnızca blok RAM'den

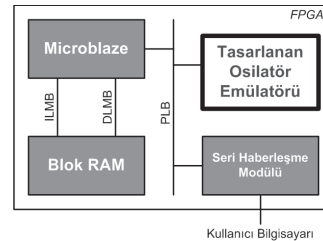
çalıştırılacak yazılım için bir kısıtlama olarak görülmektedir. Büyük yazılımlar için çözüm harici DDR bellekleri kullanmaktır [11].

Microblaze işlemcisi, Xilinx'in gömülü sistem geliştirme ortamı (EDK) kullanılarak uygulamada ihtiyaç duyulan genel çevre birimleri ve uygulamaya özel olarak tasarlanan birimler ile donatılabilmektedir. EDK [12] mikroişlemci tabanlı gömülü sistemlerin tasarımını kolaylaştıran bir ortamdır. Donanım tasarımı ve bağlantıları için XPS ile yazılım geliştirme için SDK programlarını, ayrıca C/C++ kodlarını derlemek için ise çapraz derleyiciyi ihtiva eden bu sistem, hem Xilinx tarafından tasarlanmış donanımları hem de uygulamaya özel hazırlanmış donanımları sisteme eklemeye imkân sağlamaktadır.

Bu sistemde seri porta ve tasarlanan osilatör öykünücüyü erişmek için Xilinx'in 128 bit genişlikli PLB (Processor Local Bus) veri yolu kullanılmıştır. Bu yola veri aktarımı ve bu yoldan veri alımı Microblaze işlemcisinin PLB denetim birimi tarafından kontrol edilmektedir. PLB'ye bağlanan seri port, Microblaze için hem standart giriş, hem de standart çıkış olacak şekilde yapılandırılmıştır. Böylece seri haberleşme programı kullanılarak elde edilen verilerin ana bilgisayar üzerinde görüntülenmesi sağlanmıştır.

Microblaze işlemcisi, blok RAM'e erişim program ve veri bölümleri için ayrı ayrı LMB (Local Memory Bus) veri yolları kullanır. Adı geçen yollar Xilinx tarafından tasarlanmış DLMB (Data side LMB) ve ILMB (Instruction side LMB) denetleyicileri tarafından kontrol edilen 32 bitlik yollardır. Tasarımda Microblaze'in reset vektörü blok RAM'in başlangıcını göstermektedir. Böylece blok RAM'e koyulacak yazılım, sistem başladığında Microblaze tarafından ILMB ve DLMB yolları üzerinden başlatılacaktır.

Tasarlanan sistemin çalışması, yazılımın parametreleri ve girişleri öykünücü donanımda yer alan kaydedicilere PLB üzerinden yazıp başlama işareti göndererek, üretilen çıkışı yine PLB yoluyla seri porta göndermesine dayalıdır.

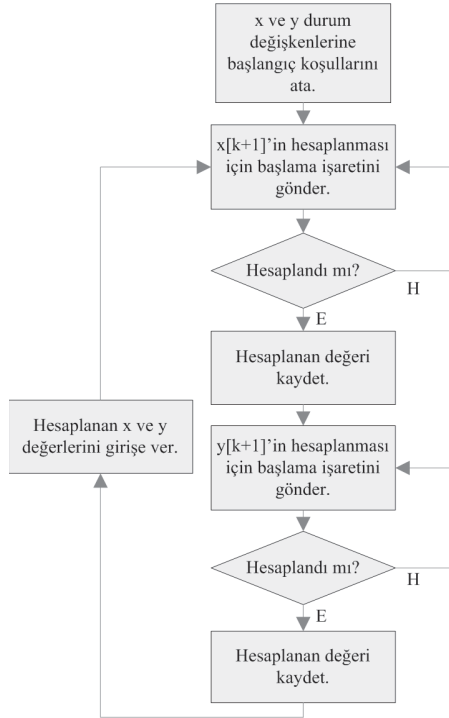


Şekil 3. Sistem blok diyagramı.

#### 3.1. Gevşemeli Osilatör Öykünücü

2009'daki çalışmamızda [10] geliştirilmiş olan doğrusal olmayan işlem birimi (NPE), Microblaze'in PLB veri yolu üzerinden haberleşebileceği biçimde yeniden düzenlenmiş

ve gevşemeli osilatör öykünücüye dönüştürülmüştür. Öykünücü içerisinde, Denklem (1)'i ileri Euler metodu ile ayrıklaştırılmış olarak çözen NPE'nin tüm parametre ve veri girişlerini süren 13 adet 32 bitlik tutucu mevcuttur ve Microblaze bu tutuculara değer atayabilmektedir. Öykünücünün durum denklemlerinin ikisini de hesaplaması yaklaşık 180 saat çevrimi zaman almaktadır. Sistemi, FPGA kartı üzerindeki 20ns periyotlu saat işareti sürmektedir. Öykünücünün tutucularına parametreler başlangıçta yazılım tarafından yüklenmektedir. Öykünüm boyunca, ileri Euler ayrıklaştırması sebebiyle her yineleme (iterasyon) sonucu yeniden öykünücüye giriş olarak verilmekte ve adım adım Denklem (1)'deki diferansiyel denklemler çözülmektedir. Bütün bu işlemler tamamen yazılım kontrolünde yürütülmektedir.



Şekil 4. Osilatör denetim yazılımının algoritmik akışı.

### 3.2. Osilatör Denetim Yazılımı

Şekil 4'te Microblaze üzerinde koşan yazılımın algoritmik akış diyagramı görülmektedir. Microblaze NPE birimi için başlangıç durum değerlerini bir kere yükler ve sonsuz döngü içerisinde akış diyagramının işlenmesini sağlar. Her yineleme sonucunda, anlık hesaplanan x ve y durumları belleğe kaydedilir. Bellek miktarının sınırlı olması sebebiyle, durum değerlerinin kaydedilmesi belirli koşullara bağlanabilir.

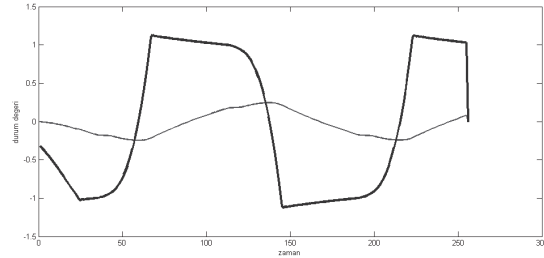
Ayrıca, istendiği takdirde osilasyonun frekansını ve fazını değiştirmek için yazılım ile

öykünücü üzerinde tutulan durum denklemleri parametrelerine müdahale edilebilir.

Daha ileri bir Microblaze yazılımı ile tek bir öykünücü donanımın zaman paylaşımı kullanımı sağlanarak birden fazla gevşemeli osilatörün birlikte öykünümü gerçekleştirilebilir. Öyleki, blok RAM'lerin tamamının kullanılması ile yaklaşık beş bin osilatörün birlikte öykünümü mümkündür. Bu miktarda hücrenin kullanılması 70 x 70 boyutlu bir ağı kurulması anlamına gelmektedir.

## 4. Sonuçlar

Şekil 5'te, tasarlanan ve Xilinx Spartan3E 1600E FPGA geliştirme kartında gerçekleştirilen sistemin zamanda ürettiği cevap görülmektedir. Bilgisayar benzetimi sonucu olan Şekil 1'deki grafikten farklı olmasının sebebi öykünücüde kullanılan parametrelerin farklı olmasıdır.



Şekil 5. FPGA üzerinde yazılım ve donanım gerçekleştirilen sistemin ürettiği işaretler.

Hızlı değişim gösteren eğri x durum değişkenine, yavaş değişen eğri y durum değişkenine aittir. Bu osilatörlerin bir araya getirilmesi ile oluşan ağ üzerinde, komşu hücrelerin birbirine olan etkisi ile osilasyonlar arasında faz farkı meydana gelmektedir. Bu faz farkı neticesinde ağın oluşturduğu iki boyutlu uzayda yayılan dalgalar gözlemlenmektedir.

Yazılım ve donanımın birlikte tasarımı ile uzay-zaman dalgalarını üretecek bu osilatör ağı, bellek miktarının izin verdiği ölçüde, istenen boyuta büyütülebilir. Ayrıca yazılım ile ağ dinamiği, öykünücü donanım dinamiği sabit kalmak üzere istendiği gibi değiştirilebilir. Bu özellik zaman içerisinde değişen sistem yaratmak, dolayısı ile uyarlanabilir algoritmaları da gerçeklemek için kullanılabilir.

Benzer çalışmalar, kullanılan kart üzerindeki DDR bellek tümdevresi ve daha yüksek merteben dinamik sistemlerin öykünümünü sağlayacak donanımlar ile de tasarlanabilir. Bu tür tasarımlarla, bilgisayar ortamında benzetimi uzun zaman alan dinamik sistemler incelenebilir.

## 5. Teşekkür

Bu çalışma, TÜBİTAK 105E103 projesi kapsamında desteklenmiştir.

## 6. Kaynaklar

- [1] L. Chua and L. Yang, "Cellular neural networks: theory," *Circuits and Systems, IEEE Transactions on*, vol. 35, no. 10, pp. 1257–1272, Oct 1988.
- [2] T. Roska and A. Rodriguez-Vazquez, "Toward visual microprocessors," *Proceedings of the IEEE* 90, pp. 1244–1257, Jul 2002.
- [3] M. E. Yalcin and J. A. K. Suykens, "Spatiotemporal pattern formation on the ACE16k CNN chip," *International Journal of Bifurcation and Chaos* 16, pp. 1537-1546, May 2006.
- [4] A. Adamatzky, P. Arena, A. Basile, R. Carmona-Galan, B. Costello, L. Fortuna, M. Frasca, and A. Rodriguez-Vazquez, "Reaction-diffusion navigation robot control: from chemical to VLSI analogic processors," *Circuits and Systems I: Regular Papers, IEEE Transactions on* 51, pp. 926–938, May 2004.
- [5] T. Roska and L. Chua, "The CNN universal machine - an analogic array computer," *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing*, vol. 40, no. 3, pp. 163–173, Mar 1993.
- [6] R. Yeniceri and M. Yalcin, "Path planning on cellular nonlinear network using active wave computing technique," *Proceedings of SPIE*, vol. 7365, 736508, May 2009.
- [7] V. Kilic, R. Yeniceri and M. Yalcin, "A New Active Wave Computing Based Real Time Mobile Robot Navigation Algorithm for Dynamic Environment," *12th International Workshop on Cellular Nanoscale Networks and their Applications, CNNA 2010*, Feb 2010.
- [8] M. Yalcin, "A simple programmable autowave generator network for wave computing applications," *IEEE Transactions on Circuits and Systems II-Express Briefs*, vol. 55, no. 11, pp. 1173–1177, Nov 2008.
- [9] R. Yeniceri and M. Yalcin, "An implementation of 2D locally coupled relaxation oscillators on an FPGA for real-time autowave generation," *11th International Workshop on Cellular Neural Networks and their Applications, CNNA 2008*, pp. 29–33, Jul 2008.
- [10] R. Yeniceri and M. Yalcin, "An emulated digital wave computer core implementation," *Proceeding of the 19th European Conference on Circuit Theory and Design, (ECCTD'09)*, Aug 2009.
- [11] Xilinx LogiCORE Local Memory Bus (LMB) kılavuzu, [http://www.xilinx.com/support/documentation/ip\\_documentation/lmb.pdf](http://www.xilinx.com/support/documentation/ip_documentation/lmb.pdf) den, 16.03.10'da alınan.
- [12] Xilinx EDK9.2i tasarım aracı dokümantasyonu, [http://www.xilinx.com/support/documentation/dt\\_edk\\_edk9-2.htm](http://www.xilinx.com/support/documentation/dt_edk_edk9-2.htm) den, 16.03.10'da alınan.