

# Gömülü Sistemler ve Uygulamaları Sempozyumu Bildiri Kitabı

---

**GÖMSİS 2010**

**İstanbul Teknik Üniversitesi  
4-5 Kasım 2010**

**GömSis 2010 - Gömülü Sistemler ve Uygulamaları Sempozyumu,  
İstanbul Teknik Üniversitesi, 4-5 Kasım 2010**

Telif hakları bildiri yazarlarına aittir.

# Kurullar

## Düzenleme Kurulu

Emre Harmancı, *İstanbul Teknik Üniversitesi* (Başkan)  
Müştak Erhan Yalçın, *İstanbul Teknik Üniversitesi*  
Osman Kaan Erol, *İstanbul Teknik Üniversitesi*  
Niyazi Saral, *Çizgi Tagem*  
Selda Kuruoğlu, *İstanbul Teknik Üniversitesi*  
Ramazan Yeniçeri, *İstanbul Teknik Üniversitesi*

## Program Kurulu

Emre Harmancı, *İstanbul Teknik Üniversitesi*  
Günhan Dünder, *Boğaziçi Üniversitesi*  
Arda Yurdakul, *Boğaziçi Üniversitesi*  
Müştak Erhan Yalçın, *İstanbul Teknik Üniversitesi*  
Feza Buzluca, *İstanbul Teknik Üniversitesi*  
Osman Kaan Erol, *İstanbul Teknik Üniversitesi*

## Bilim Kurulu

Eşref Adalı, *İstanbul Teknik Üniversitesi*  
İrfan Alan, *Ege Üniversitesi*  
Taner Arsan, *Kadir Has Üniversitesi*  
Atilla Ataman, *Yıldız Teknik Üniversitesi*  
Faik Başkaya, *Boğaziçi Üniversitesi*  
Feza Buzluca, *İstanbul Teknik Üniversitesi*  
Günhan Dünder, *Boğaziçi Üniversitesi*  
Oğuz Ergin, *TOBB Ekonomi ve Teknoloji Üniversitesi*  
Osman Kaan Erol, *İstanbul Teknik Üniversitesi*  
Sarp Ertürk, *Kocaeli Üniversitesi*  
Ece Olcay Güneş, *İstanbul Teknik Üniversitesi*  
Hasan Cengiz Güran, *Orta Doğu Teknik Üniversitesi*  
Emre Harmancı, *İstanbul Teknik Üniversitesi*  
İlker Hamzaoğlu, *Sabancı Üniversitesi*  
Bülent Örencik, *TÜBİTAK Bilişim Teknolojileri Enstitüsü*  
Ahmet Özkurt, *Dokuz Eylül Üniversitesi*  
Gökay Saldamlı, *Boğaziçi Üniversitesi*  
Coşkun Sönmez, *Yıldız Teknik Üniversitesi*  
Oğuz Tosun, *Boğaziçi Üniversitesi*  
Tuncay Uzun, *Yıldız Teknik Üniversitesi*  
Mürvet Kırıcı, *İstanbul Teknik Üniversitesi*  
Sıddıka Berna Örs Yalçın, *İstanbul Teknik Üniversitesi*  
Müştak Erhan Yalçın, *İstanbul Teknik Üniversitesi*  
Arda Yurdakul, *Boğaziçi Üniversitesi*



# Program ve İçindekiler

4 Kasım 2010

<b>Kayıt</b>	<b>9:00-10:15</b>
<b>Açılış</b>	<b>10:15-11:00</b>
<b>Davetli Konuşmalar</b>	<b>11:00-12:30</b>

Bir Ar-Ge katalizörü olarak Avrupa İşbirliği Projeleri . . . . .	1
<i>Serhat Özdemir</i>	

Araç içi ve Araçlar arası İletişimde Otomotiv’de Gömülü Sistemler . . . . .	3
<i>Reşat Yüzbaşıç ve Bülent Erbaş</i>	

<b>Öğle Yemeği</b>	<b>12:30-14:00</b>
--------------------	--------------------

## Sözlü Sunumlar

<b>1. Oturum</b>	<b>14:00-15:20</b>
------------------	--------------------

H.264 Çerçeve İçi Öngörü Donanımının Güç Kullanımını Azaltan Teknikler . . . . .	5
<i>Yusuf Adıbelli, Mustafa Parlak, İlker Hamzaoğlu</i>	

Çok Kipli Ayrık Hücreli Sinir Ağı Modeli . . . . .	11
<i>Mehmet Tükel, Arda Yurdakul</i>	

Fiziksel klonlanamayan fonksiyon devrelerinin iyileştirilmesi . . . . .	17
<i>Gürkan Kaya, Gökay Saldamlı, Müştak Erhan Yalçın</i>	

Ultrasonik Temizlik Sistemlerinde Rezonans Frekansı Tespiti İçin Yinelemeli Frekans Bandı Tarama Tekniği . . . . .	21
<i>Ulvi Güvenç, Mehmet Yakut, Ali Tangel</i>	

<b>Ara</b>	<b>15:20-15:40</b>
<b>2. Oturum</b>	<b>15:40-17:00</b>

Kablosuz Algılayıcı Ağları ile Yangın Tespit Sistemi . . . . .	25
<i>Çağdaş Döner, Gökhan Şimşek, K. Sinan Yıldırım, Aylın Kantarcı</i>	

Gömülü Çoklu İşlemcili Sistemlerde Yeniden Betimlenebilir Haberleşme Protokolleri . . . . .	29
<i>Salih Bayar, Arda Yurdakul</i>	

Küçük Ölçekli Gömülü Sistemler için Bir Yazılım Geliştirme Sistemi . . . . .	35
<i>İsmail Savaşkan, Feza Buzluca</i>	

Çoklu Referans Çerçevesi Kullanan Uyarlanabilir H.264 Hareket Tahmini Donanımları . . . . .	41
<i>Aydın Aysu, Onur Can Ulusel, İlker Hamzaoğlu</i>	

## 5 Kasım 2010

<b>Davetli Konuşmalar</b>	<b>9:30-11:00</b>
FPGA Her Beklentiyi Karşılatabilir mi? - Türkiye’de ASIC Üretimi . . . . .	47
<i>Aziz U. Çalışkan</i>	
Gömülü Sistemler ile Geleceğin Akıllı Cihaz Çözümleri . . . . .	49
<i>Uygur Doğan</i>	
<b>Ara</b>	<b>11:00-11:15</b>
<b>Davetli Konuşma</b>	<b>11:15-12:00</b>
Savunma Sanayisinde FPGA Kullanımı . . . . .	51
<i>Kutsal Anıl</i>	
<b>Sözlü Poster Sunumları</b>	<b>12:00-12:30</b>
<b>Öğle Yemeği</b>	<b>12:30-13:30</b>
<b>Poster Sunumları</b>	<b>13:30-15:00</b>
Küp Uydu için Yer İstasyonunda Yazılım Tanımlı Radyo Tasarımı . . . . .	53
<i>Seyyid Muhammed Dilek, Osman Ceylan, H. Bülent Yağcı</i>	
Gömülü Sistem ile Akıllı Gaz Algılayıcısı Geliştirme . . . . .	57
<i>Gürkan Yavuz, Halil İbrahim Çakır, Sevcan Aytekin, Ahmet Özmen</i>	
Balık Çiftlikleri için Uzaktan İzleme Sistemi Tasarımı . . . . .	61
<i>Gökhan Bölük, Orhan Baykan</i>	
Ulusal Akıllı Kart Tümdevresi ve UKiS İşletim Sistemi . . . . .	67
<i>Mustafa Başak, Koral Ataman, Ercan Doğan</i>	
CAN Protokol Tabanlı Bir Tren Simülatörünün Gerçeklenmesi . . . . .	75
<i>Fatih Yardım, Oğuzhan Urhan</i>	
Radyo Frekansıyla Konteyner Takip Sistemi . . . . .	81
<i>Muhammet Alican Güncan, Ali Yavuz Kahveci, Muhammet Erkoç, Arda Yurdakul</i>	
Bir Gömülü Sistem Uygulaması: Elektronik Pusula . . . . .	85
<i>Mustafa Selman Yıldırım</i>	
Telsiz Duyurga Ağları için Şifreleme ve Hata Tespiti İşlemlerinin Düşük Güç Tüketimiyle Gerçeklenmesi ve Ortak Analizi . . . . .	91
<i>Abdullah Usta, Arda Yurdakul</i>	
Gömülü Sistemlerde Medya Uygulamaları ve Performans . . . . .	97
<i>İsmail Uzun, Halid Ziya Yerebakan</i>	
<b>Panel</b>	<b>15:00-17:00</b>
<i>Türkiye’de Gömülü Sistemlerin Geleceği - Sanayi Öngörüsü</i>	

# Önsöz

Bu kitapta Gömülü Sistemler ve Uygulamaları Sempozyumu 2010 (GÖMSİS 2010)'da sunulacak akademik bildiri-lerin tam metinleri ile çağrılı konuşmacıların sunuşlarının özetleri yer almaktadır.

GÖMSİS 2010'u bu yıl İstanbul Teknik Üniversitesi, Elektrik - Elektronik Fakültesi düzenlemektedir. Boğaziçi Üniversitesi Bilgisayar ve Elektrik - Elektronik Bölümleri sempozyuma büyük akademik destek vermiştir. Sempozyumun maddi destekçisi Çizgi TAGEM'dir. Ayrıca IEEE Türkiye Şubesi ve Türkiye Bilimsel ve Teknolojik Araştırma Kurumu (TÜBİTAK) destekçilerimiz arasındadır.

GÖMSİS 2010, Gömülü Sistemler ve Uygulamaları konusunda iki yılda bir yapılması planlanan sempozyumlar dizisinin ikincisidir. GÖMSİS'lerin birincil amacı gömülü sistemler konusunda yeni filizlenen ARGE ve inovasyon çalışmalarını desteklemek ve yüreklendirmektir.

GÖMSİS 2010, üniversite - sanayi işbirliğini pekiştirecek bir köprü görevi yapma amacını da taşımaktadır. Sempozyum organizasyonu bilgi paylaşımını ve sonrasındaki yeni işbirliklerinin kurulmasını sağlamayı hedef edinmiştir.

Bu yıl sempozyuma toplam yirmi üç araştırma bildirisi başvurusu yapılmıştır. Bütün bildirimler, Program Kurulu tarafından Bilim Kurulunun içinden seçilen üç hakeme gönderilmiş, Program Kurulu hakem görüşlerini göz önüne alarak bu bildirimlerden sekiz tanesini sözlü sunuma kabul etmiş, dokuz tanesinin ise poster olarak sunulmasını uygun bulmuştur.

Sempozyumun sanayi ayağında bu yıl beş adet davetli konuşmacının yapacağı sunuşlar vardır. Bu sunuşlarda işlenecek konular şunlardır:

- Bir Ar-Ge katalizörü olarak Avrupa İşbirliği Projeleri,
- Araç içi ve Araçlar arası İletişimde Otomotiv'de Gömülü Sistemler,
- FPGA Her Beklentiyi Karşılabilir mi? - Türkiye'de ASIC Üretimi
- Gömülü Sistemler ile Geleceğin Akıllı Cihaz Çözümleri,
- Savunma Sanayisinde FPGA Kullanımı.

GÖMSİS 2010 Prof. Dr. Emre Harmancı'nın moderatörlüğünde "Türkiye'de Gömülü Sistemlerin Geleceği - Sanayi Öngörüsü" konulu bir panel ile son bulacaktır. Panelistlerin Bülent Candan (TÜBİTAK Bilişim Teknolojileri Enstitüsü), Adnan Sazlı (Altera), Burak İlhan (Texas Instruments), Kaan Dericioğlu (Ankara Patent), Kutsal Anıl (PAVO Tasarım Üretim Elektronik Ticaret A.Ş.), Uygur Doyuran (Intel) ve Serhat Özdemir'den (Aselsan A.Ş.) oluşması planlanmıştır.

GÖMSİS 2010'un düzenlenmesinde katkısı olan akademiden ve sanayiden tüm destekçilere en içten teşekkürlerimi sunarım.

Prof. Dr. A. Emre Harmancı,  
GÖMSİS 2010 Düzenleme Kurulu Başkanı  
Kasım 2010





## **Bir Ar-Ge Katalizörü Olarak Avrupa İşbirliği Projeleri**

*Serhat Özdemir*

Aselsan Mikroelektronik, GÜdüm ve Elektro-optik Grubu, Tasarım Lideri  
sozdemir@aselsan.com.tr

### **Özetçe**

Avrupa Topluluğu, özellikle ABD ve Japonya'daki teknolojik gelişmişlik düzeyini yakalamak için, bu aralar 7.si devam eden ve 8.si üzerinde çalışmaların başladığı Çerçeve Programları'nı oluşturmuş ve bunlara başka mekanizmalar da ekleyerek Ar-Ge destek şemsiyesini yıllar içinde genişletmiştir. Bunun sonucu olarak günümüzde şu ana Ar-Ge destek mekanizmaları mevcuttur: 7.Çerçeve Programı (Akademik Ar-Ge, Ana başlıkları: Kapasiteler, İnsanlar, İşbirlikleri, Fikirler), Eureka (Endüstriyel Ar-Ge, Bağlı Kümeler: ITEA2, Celtic, Euripides, Catrene, Eurogia, Acqueau, MF.IND, Eurostars), Teknoloji Platformları (Artemis, Eurob, vs.) ve diğerleri. Serhat Özdemir konuşmasında bu programlar hakkında genel bilgi verecek ve Aselsan'ın spesifik deneyimlerinden örnekler gösterecektir.



## Araç içi ve Araçlar arası İletişimde - Otomotiv’de Gömülü Sistemler

*Reşat Yüzbaşıç<sup>1</sup>, Bülent Erbaş<sup>2</sup>*

<sup>1</sup>Telemetri Sistemleri San. ve Tic. A.Ş. Genel Müdürü  
resat@telemetri.com

<sup>2</sup>KoçSistem Bilgi ve İletişim Hizmetleri A.Ş.  
erbasb@kocsistem.com.tr

### Özetçe

Otomotivde 1950’lerde yalnızca radyo ve birkaç basit elektrik donanımı dışında hiç elektronik donanım yok iken, günümüze gelindiğinde 10–20 bin adet elektrik-elektronik parça ve kendi aralarında ve dış dünya ile iletişim içinde 40’dan fazla gömülü sistem donanımı ile çok sayıda farklı yerel iletişim ağını içeren gömülü sistem dizgesinden oluşur hale gelmiştir. 1980 yılında toplam maliyete göre %1 oranındaki elektronik gömülü sistemler maliyeti 2005’te %20 oranına ulaşmış olup, 2015’te ise araç maliyetinin %40’ını aşması beklenmektedir. Giderek artan araç içi gömülü sistem (Electronic Control & Communication Units - ECUs) yoğunluğu iletişim, konfor, elektronik alış veriş ve eğlencenin yanı sıra giderek artan ölçüde araç ve trafik güvenliğini kapsamakta olup, gömülü sistemler için çok önemli bir uygulama alanına dönüşmektedir. KoçSistem koordinatörlüğünde, KoçSistem, TELEMETRİ A.Ş., Fiat-TOFAŞ ve Ford-OTOSAN tarafından ortaklaşa yürütülmekte olan “Güvenli trafik için araçlar arası haberleşme teknolojileri” adlı kısaca C2C veya V2V olarak anılan sistem geliştirme projesinde WAVE (Wireless Access in Vehicular Environments) platformu işlevlerini gerçekleştirecek gömülü sistem (Araçlar arası haberleşme sistemi - ARHASİS) tasarımı ve ürün geliştirme çalışmalarını kapsamaktadır. ARHASİS, karmaşık iletişim işlevlerinin yanı sıra, 500 km/h görelî hız için dinamik araç trafiğinde başta çarpışma olmak üzere öncelikli korunma ve uyarı işlevlerine yönelik hızlı karar verme ve uygulama işlevlerini de yerine getirecek yetkinlikte olacaktır.



## H.264 Çerçeve İçi Öngörü Donanımının Güç Kullanımını Azaltan Teknikler

Yusuf Adibelli<sup>1</sup>, Mustafa Parlak<sup>1</sup>, İlker Hamzaoğlu<sup>1</sup>

<sup>1</sup>Mühendislik ve Doğa Bilimleri Fakültesi  
Sabancı Üniversitesi, İstanbul

{yadibelli, mparlak, hamzaoglu}@sabanciuniv.edu

### Özetçe

H.264 çerçeve içi öngörü algoritmasının hesaplama karmaşıklığı yüksektir. Bu bildiride, piksellerin eşitliğine ve benzerliğine bakarak H.264 çerçeve içi öngörü algoritmasının işlem miktarını ve dolayısıyla H.264 çerçeve içi öngörü donanımının güç kullanımını azaltan teknikler önerdik. Önerilen piksellerin eşitliğine bakarak işlem miktarını azaltan (PEİMA) teknik, çerçeve içi öngörü denklemlerinde kullanılan piksellerin eşit olup olmadıklarına bakmaktadır. Eğer bu pikseller birbirine eşit ise denklemlerin hesaplanması basitleşmektedir. Önerilen piksellerin benzerliğine bakarak işlem miktarını azaltan (PBİMA) teknik ise çerçeve içi öngörü denklemlerinde kullanılan piksellerin benzer olup olmadıklarına bakmaktadır. Eğer bu pikseller benzer ise denklemlerin hesaplanması basitleşmektedir. PEİMA ve PBİMA teknikleri, çok az miktarda karşılaştırma yaparak, 4x4 çerçeve içi öngörü kiplerinin hesaplanması için gereken işlem miktarını sırasıyla %78 ve %89 oranında azaltmaktadır. PEİMA tekniği PSNR kaybına neden olmamaktadır. PBİMA tekniği ise bazı video çerçevelerinde PSNR'ı düşürmektedir. Bir H.264 4x4 çerçeve içi öngörü donanımı tasarladık ve Verilog HDL ile gerçekleştirdik. PEİMA ve PBİMA teknikleri bu donanımın güç tüketimini sırasıyla %13.7 ve %17.2 oranında azalttı.

### 1. Giriş

Video sıkıştırma günümüzde birçok uygulamada kullanılmaktadır. Son yıllarda, önceki video sıkıştırma standartlarından çok daha iyi sıkıştırma verimliliğine sahip yeni bir uluslararası video sıkıştırma standardı, H.264, geliştirildi. H.264 standardındaki sıkıştırma verimliliği birden fazla yeni kodlama tekniğinin kullanılmasından kaynaklanmaktadır. Bu tekniklerden bir tanesi H.264 standardının temel profilinde kullanılan çerçeve içi öngörü algoritmasıdır.

H.264 çerçeve içi öngörü algoritması bir Makroblok (MB) için uzamsal artıklığı kullanarak bir öngörü üretmektedir. H.264 çerçeve içi öngörü algoritması önceki video sıkıştırma standartlarında kullanılan çerçeve içi öngörü algoritmalarından daha iyi kodlama sonuçları vermektedir. Fakat, hesaplama karmaşıklığını önemli miktarda arttırmaktadır.

H.264 çerçeve içi öngörü algoritmasında yapılan işlem miktarını ve dolayısıyla H.264 çerçeve içi donanımının güç kullanımını piksellerin eşitliğine bakarak bit hızı ve PSNR kaybı olmadan önemli miktarda azaltan özgün bir tekniği [3]'te önerdik. Bu teknik bir çerçeve içi öngörü kipinin denklemlerinde kullanılan piksellerin eşit olup olmadığına

bakarak, bu kipin öngördüğü piksel değerlerini daha az işlem yaparak hesaplamayı sağlar. Eğer bir kipin denklemlerinde kullanılan bütün pikseller eşit ise, bu kipin öngördüğü piksel değerleri bu piksel değerine eşit olur. Bu nedenle, bu kipin öngörü denklemlerindeki işlemlerin yapılmasına gerek kalmaz.

H.264 çerçeve içi öngörü algoritmasında yapılan işlem miktarını ve dolayısıyla H.264 çerçeve içi donanımının güç kullanımını piksellerin benzerliğine bakarak az bir PSNR kaybıyla daha fazla azaltan özgün bir tekniği ise [4]'te önerdik. Bu teknik bir çerçeve içi öngörü kipinin denklemlerinde kullanılan piksellerin benzer olup olmadığına bakarak, bu kipin öngördüğü piksel değerlerini daha az işlem yaparak hesaplamayı sağlar. Eğer bir kipin denklemlerinde kullanılan bütün pikseller benzer ise, bu kipin öngördüğü piksel değerlerinin benzer piksellerden bir tanesine eşit olduğu varsayılmaktadır. Bu nedenle, bu kipin öngörü denklemlerindeki işlemlerin yapılmasına gerek kalmamaktadır.

H.264 4x4 çerçeve içi öngörü kiplerinin bazı denklemleri aynıdır. Bu ortak denklemleri her kip için hesaplamaya gerek yoktur. Bu nedenle, bu bildiride, ortak öngörü denklemlerini sadece bir kere hesapladık ve sonuçları ilgili öngörü kiplerinde kullandık. Piksellerin eşitliğine bakarak işlem miktarını azaltan (PEİMA) tekniği ve piksellerin benzerliğine bakarak işlem miktarını azaltan (PBİMA) tekniği her öngörü denklemine ayrı ayrı uyguladık.

PEİMA tekniği bir öngörü denkleminde kullanılan piksellerin eşit olup olmadığına bakar. Eğer bu öngörü denkleminde kullanılan bütün pikseller eşit ise, bu denklemin öngördüğü piksel değeri bu piksel değerine eşit olur. Bu nedenle, bu öngörü denklemindeki işlemlerin yapılmasına gerek kalmaz. PEİMA tekniği H.264 çerçeve içi öngörü algoritmasında yapılan işlem miktarını ve dolayısıyla H.264 çerçeve içi donanımının güç kullanımını bit hızı ve PSNR kaybı olmadan önemli miktarda azaltmaktadır.

PBİMA tekniği bir öngörü denkleminde kullanılan piksellerin benzer olup olmadığına bakar. Eğer bu öngörü denkleminde kullanılan bütün pikseller benzer ise, bu denklemin öngördüğü piksel değerinin benzer piksellerden bir tanesine eşit olduğu varsayılmaktadır. Bu nedenle, bu öngörü denklemindeki işlemlerin yapılmasına gerek kalmamaktadır. PBİMA tekniği H.264 çerçeve içi öngörü algoritmasında yapılan işlem miktarını ve dolayısıyla H.264 çerçeve içi donanımının güç kullanımını az bir PSNR kaybıyla daha fazla azaltmaktadır.

H.264 JM 14.0 referans yazılımı [5] kullanılarak yapılan simülasyonlardan elde edilen sonuçlara göre, önerilen PEİMA tekniği çok az miktarda karşılaştırma işlemi yaparak 4x4 çerçeve içi öngörü kiplerinde yapılan işlem miktarını %78

oranında ([3]'te önerilen teknikten daha fazla) azaltılmaktadır. Önerilen PBİMA tekniği ise çok az miktarda karşılaştırma işlemi yaparak 4x4 çerçeve içi öngörü kiplerinde yapılan işlem miktarını %89 oranında ([4]'te önerilen teknikten daha fazla) azaltılmaktadır. Her iki teknikte bir 4x4 bloğun bütün öngörü kipleri için 12 karşılaştırma işlemi gerektirmektedir. Simülasyon sonuçları ayrıca önerilen PBİMA tekniğinin bazı video çerçeveleri için PSNR'ı çok az miktarda ([4]'te önerilen teknikten daha az) azalttığını göstermiştir.

Önerilen teknikleri içeren bir H.264 4x4 çerçeve içi öngörü donanımı tasarlandı. Bu donanım Verilog Donanım Tanımlama Dili (HDL) ile gerçekleştirildi ve Mentor Graphics ModelSim SE yazılımı ile simülasyonlar yapılarak doğrulandı. Verilog kodları Mentor Graphics Precision RTL yazılımı ile 2V8000ff1517 Xilinx Virtex II FPGA'ine sentezlendi ve Xilinx ISE 8.2i yazılımı kullanılarak aynı FPGA'ye 50 MHz hızında çalışacak şekilde yerleştirildi. Bu 4x4 çerçeve içi öngörü donanımının Xilinx Virtex II FPGA'inde 25 MHz'deki güç kullanımı Xilinx XPower yazılımı kullanılarak belirlendi. Önerilen PEİMA ve PBİMA teknikleri 4x4 çerçeve içi öngörü donanımının güç tüketimini sırasıyla %13.7 ve %17.2 oranında azalttı.

Literatürde, H.264 çerçeve içi öngörü algoritmasının hesaplama karmaşıklığını azaltmak için, bu bildiriye kullanılan tekniğe benzer veri tekrar kullanımı teknikleri kullanılmıştır [6, 7]. Ayrıca H.264 çerçeve içi öngörü algoritmasının hesaplama karmaşıklığını azaltmak için başka teknikler de kullanılmıştır [8, 9]. Fakat bu teknikler H.264 çerçeve içi öngörü algoritmasının hesaplama karmaşıklığını azaltmak için bütün çerçeve içi öngörü kiplerini kullanmak yerine seçilen bazı kipleri kullanmaktadırlar. Ayrıca bu tekniklerin önemli miktarda önhesaplama yapmak ve PSNR kaybına neden olmak gibi dezavantajları vardır.

Bu bildiriye önerilen teknikler ise bütün 4x4 çerçeve içi öngörü kiplerini kullanıyorlar ve bu teknikler literatürdeki hesaplama karmaşıklığını azaltan diğer tekniklerle beraber de kullanılabilirler [6, 7, 8, 9]. Literatürde H.264 çerçeve içi öngörü algoritmasını gerçekleyen donanım mimarileri önerilmiştir [10, 11, 12]. Fakat bu donanımların güç tüketimleri bildirilmemiş ve bu donanımlar bizim önerdiğimiz teknikleri kullanmamışlardır.

Bildirinin geri kalan kısmı şu şekilde organize edilmiştir. Bölüm 2'de H.264 çerçeve içi öngörü algoritması anlatılmıştır. Bölüm 3'te önerilen işlem sayısını azaltan teknikler anlatılmıştır. Önerilen H.264 4x4 çerçeve içi öngörü donanımı Bölüm 4'te anlatılmıştır. Donanımın güç tüketim sonuçları Bölüm 5'te verilmiştir. Bölüm 6'da sonuçlar sunulmuştur.

## 2. H.264 Çerçeve İçi Öngörü Algoritması

Çerçeve içi öngörü algoritması bir MB'nin piksel değerlerini komşu pikselleri kullanarak öngörür. Her luma MB için iki farklı öngörü hesaplanır. Bir MB'deki 16 4x4 bloğun herbiri için 9 farklı 4x4 çerçeve içi öngörü kipi kullanılarak öngörü hesaplanır. Birde 4 farklı 16x16 çerçeve içi öngörü kipi kullanılarak öngörü hesaplanır. Daha sonra kip seçimi algoritması bu çerçeve içi öngörülerden en iyisini seçer.

Şekil 1'de görüldüğü gibi her 4x4 çerçeve içi öngörü kipi A,...,M komşu piksellerini kullanarak 16 öngörülen piksel değeri hesaplar. Öngörülen piksel değeri komşu pikseller belli katsayılarla çarpılarak toplanıp belli miktarda sağa kaydırılarak hesaplanır.

Diagonal Down Left ve Diagonal Down Right 4x4 çerçeve içi öngörü kiplerinin öngörü denklemleri Şekil 2'de gösterilmiştir. [y, x] pikselin 4x4 bloğundaki yerini göstermektedir. [0, 0], [0, 3], [3, 0] ve [3, 3] bir 4x4 bloğun sırasıyla en üst sol, en üst sağ, en alt sol, ve en alt sağında bulunan pikselleri göstermektedir. pred[y, x] ise [y, x]'deki pikselin öngörülen değerini göstermektedir.

## 3. Önerilen Hesaplama Karmaşıklığını ve Güç Kullanımı Azaltan Teknikler

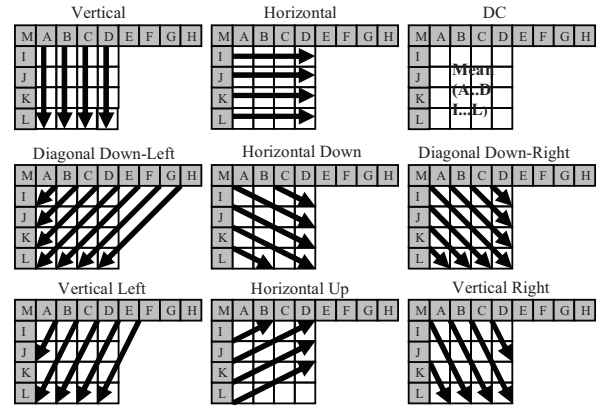
H.264 4x4 çerçeve içi öngörü kiplerinin bazı denklemleri aynıdır. Bu ortak denklemleri her kip için hesaplamaya gerek yoktur. Bu nedenle, bu bildiriye, ortak öngörü denklemlerini sadece bir kere hesapladık ve sonuçları ilgili öngörü kiplerinde kullandık.

Şekil 2'de görüldüğü gibi, denklem (1) Diagonal Down Left ve Diagonal Down Right öngörü kiplerinde ortaktır, ve Diagonal Down Right kipinin pred[0,2] ve pred[1,3] öngörü denklemleri aynıdır. Vertical Right kipinin pred[1,2] ve pred[3,3], Vertical Left kipinin pred[1,0] ve Horizontal Down kipinin pred[0,3] öngörü denklemleri de denklem (1) ile aynıdır.

$$(A + B) + (B + C) + 2 \gg 2 \quad (1)$$

H.264 4x4 çerçeve içi öngörü Diagonal Down-Left (DDL), Diagonal Down-Right (DDR), Vertical Right (VR), Vertical Left (VL), Horizontal Down (HD), Horizontal Up (HUP) kiplerinde toplam 96 (6x16) öngörü denklemi vardır. Bu denklemlerin 23 tanesi birbirinden farklıdır. Vertical ve Horizontal kipleri hesaplama gerektirmemektedir. Bu 23 denklem, bu denklemlerde kullanılan pikseller, bu denklemlerin kullanıldığı kip sayısı, bu denklemler tarafından öngörülen piksel sayısı ve bu denklemlerde yapılan Toplama ve Kaydırma işlemi sayıları Tablo 1'de gösterilmiştir.

H.264 4x4 çerçeve içi öngörü kipleri CIF (352x288) boyutundaki bir çerçeve için 884183 Toplama ve 529181 Kaydırma işlemi yapmaktadır. Eğer 23 öngörü denklemi sadece bir kere hesaplanırsa, 417997 Toplama ve 230839 Kaydırma işlemi yaparlar, ve bu Toplama ve Kaydırma işlemlerini %53 ve %56 azaltır.



Şekil 1: 4x4 Çerçeve İçi Öngörü Kipleri

$$\begin{aligned}
\text{pred}[0, 0] &= [(A + B) + (B + C) + 2] \gg 2 \\
\text{pred}[0, 1] = \text{pred}[1, 0] &= [(C + D) + (B + C) + 2] \gg 2 \\
\text{pred}[0, 2] = \text{pred}[1, 1] = \text{pred}[2, 0] &= [(C + D) + (D + E) + 2] \gg 2 \\
\text{pred}[0, 3] = \text{pred}[1, 2] = \text{pred}[2, 1] &= \\
&\quad \text{pred}[3, 0] = [(E + F) + (D + E) + 2] \gg 2 \\
\text{pred}[1, 3] = \text{pred}[2, 2] = \text{pred}[3, 1] &= [(E + F) + (F + G) + 2] \gg 2 \\
\text{pred}[2, 3] = \text{pred}[3, 2] &= [(G + H) + (F + G) + 2] \gg 2 \\
\text{pred}[3, 3] &= [(G + H) + (H + H) + 2] \gg 2
\end{aligned}$$

(a) 4x4 Diagonal Down-Left Kipi

$$\begin{aligned}
\text{pred}[0, 2] = \text{pred}[1, 3] &= [(A + B) + (B + C) + 2] \gg 2 \\
\text{pred}[0, 3] &= [(C + D) + (B + C) + 2] \gg 2 \\
\text{pred}[3, 0] &= [(J + K) + (K + L) + 2] \gg 2 \\
\text{pred}[2, 0] = \text{pred}[3, 1] &= [(J + K) + (I + J) + 2] \gg 2 \\
\text{pred}[1, 0] = \text{pred}[2, 1] = \text{pred}[3, 2] &= [(M + I) + (I + J) + 2] \gg 2 \\
\text{pred}[0, 0] = \text{pred}[1, 1] = \text{pred}[2, 2] &= \\
&\quad \text{pred}[3, 3] = [(M + I) + (M + A) + 2] \gg 2 \\
\text{pred}[0, 1] = \text{pred}[1, 2] = \text{pred}[2, 3] &= [(A + B) + (M + A) + 2] \gg 2
\end{aligned}$$

(b) 4x4 Diagonal Down-Right Kipi

Şekil 2: 4x4 Diagonal Down-Left ve Diagonal Down-Right Kiplerinin Öngörü Denklemleri

[3]'te önerilen piksellerin eşitliğine bakarak işlem miktarını azaltan (PEİMA) teknik, bir çerçeve içi öngörü kipinin denklemlerinde kullanılan bütün piksellerin eşit olup olmadığına bakar. Eğer bütün pikseller eşit ise, bu kipin öngördüğü piksel değerleri bu piksel değerine eşit olur. Bu nedenle, bu kipin öngörü denklemlerindeki işlemlerin yapılmasına gerek kalmaz. 4x4 çerçeve içi kiplerin öngörü denklemlerinde kullanılan pikseller Tablo 2'de gösterilmiştir. Örneğin, DDL kipinin öngörü denklemlerinde A, B, C, D, E, F, G, H pikselleri kullanılır. [0, 0]'daki pikselin öngörü denklemi (1)'de gösterilmiştir. Eğer A, B, C, D, E, F, G, H pikselleri birbirine eşitse, (1) denklemi basitleşerek (2)'deki gibi olur. Bu durum DDL kipinin Şekil 2'de gösterilen diğer öngörü denklemleri için de geçerlidir.

$$\text{pred}[y,x] = [4A+2] \gg 2 = A \quad (2)$$

[4]'te önerilen piksellerin benzerliğine bakarak işlem miktarını azaltan (PBİMA) teknik, bir çerçeve içi öngörü kipinin denklemlerinde kullanılan bütün piksellerin benzer olup olmadığına bakar. Eğer bütün pikseller benzer ise, bu kipin öngördüğü piksel değerlerinin benzer piksellerden bir tanesine eşit olduğu varsayılmaktadır. Bu nedenle, bu kipin öngörü denklemlerindeki işlemlerin yapılmasına gerek kalmamaktadır. Örneğin, eğer A, B, C, D, E, F, G, H pikselleri benzerse, DDL kipindeki (1) denkleminin basitleşerek (2)'deki gibi olduğu varsayılır.

Bu bildiride, PEİMA tekniğini her öngörü denklemine ayrı ayrı uyguladık. PEİMA tekniği bir öngörü denkleminde kullanılan piksellerin eşit olup olmadığına bakar. Eğer bu öngörü denkleminde kullanılan bütün pikseller eşit ise, bu denklemin öngördüğü piksel değeri bu piksel değerine eşit olur. Bu nedenle, bu öngörü denklemindeki işlemlerin yapılmasına gerek kalmaz. Örneğin, (1) denklemi DDL, DDR, VR, VL ve HD kiplerinde kullanılmaktadır. Eğer bu denklemde kullanılan A, B, C pikselleri eşitse, bu denklem basitleşerek (2)'deki gibi olur.

PBİMA tekniğini de her öngörü denklemine ayrı ayrı uyguladık. PBİMA tekniği bir öngörü denkleminde kullanılan piksellerin en önemsiz bitlerini belli bir miktarda (1, 2, 3 veya 4 bit) kırparak pikselleri karşılaştırıp benzer olup olmadıklarına bakar. Eğer bu öngörü denkleminde kullanılan bütün kırpılmış pikseller eşit ise, bu denklemin öngördüğü piksel değerinin benzer piksellerden bir tanesine eşit olduğu varsayılmaktadır. Bu nedenle, bu öngörü denklemindeki işlemlerin yapılmasına gerek kalmamaktadır. Örneğin, (1)

denklemi DDL, DDR, VR, VL ve HD kiplerinde kullanılmaktadır. Eğer bu denklemde kullanılan kırpılmış A, B, C pikselleri eşitse, bu denklemin basitleşerek (2)'deki gibi olduğu varsayılır.

Bir video çerçevesinde öngörü denklemlerindeki piksellerin eşit ve benzer olduğu 4x4 blokların miktarı çerçeveden çerçeveye değişir. Bu nedenle H.264 JM referans yazılımı 14.0 [5] kullanılarak CIF (352x288) boyutundaki Foreman, Akiyo ve Mother&Daughter çerçevelerinde 28, 35 ve 42 nicemleme parametresi (NP) değerlerinde öngörü denklemlerindeki piksellerin eşit ve benzer olduğu 4x4 blokların yüzdesi belirlendi ve Tablo 3'te gösterildi. Eşit pikseli denklemlere sahip 4x4 blokların oranı %10 ile %94 arasında, 4 bit kırpma (4bk) durumunda benzer pikseli denklemlere sahip 4x4 blokların oranı ise %50 ile %97 arasında değişmektedir.

Önerilen PEİMA ve PBİMA tekniklerinin CIF boyutundaki Akiyo, Foreman ve Mother&Daughter çerçeveleri için elde ettikleri işlem miktarındaki azalmalar Tablo 1 ve 3'teki bilgiler kullanılarak hesaplandı. CIF boyutunda bir çerçeve için 4x4 çerçeve içi öngörü, [3]'te önerilen PEİMA tekniği ile 4x4 çerçeve içi öngörü, veri tekrar kullanımlı 4x4 çerçeve içi öngörü, önerilen veri tekrar kullanımını ve PEİMA tekniği ile 4x4 çerçeve içi öngörü işlem miktarları Tablo 4'te gösterilmiştir. CIF boyutunda bir çerçeve için 4x4 çerçeve içi öngörü, [4]'te önerilen PBİMA tekniği (4bk) ile 4x4 çerçeve içi öngörü, veri tekrar kullanımlı 4x4 çerçeve içi öngörü, önerilen veri tekrar kullanımını ve PBİMA tekniği (4bk) ile 4x4 çerçeve içi öngörü işlem miktarları Tablo 5'te gösterilmiştir.

Tablo 1: 4x4 Çerçeve İçi Öngörü Kiplerinin Denklemi

Piksel	Denklem	Kipler	Öng. Pikseller	Top. #	Kay. #
A,B,C	[(A + 2B + C) + 2] >> 2	5	7	3	2
B,C,D	[(B + 2C + D) + 2] >> 2	4	6	3	2
C,D,E	[(C + 2D + E) + 2] >> 2	2	5	3	2
D,E,F	[(D + 2E + F) + 2] >> 2	2	6	3	2
E,F,G	[(E + 2F + G) + 2] >> 2	2	4	3	2
F,G,H	[(F + 2G + H) + 2] >> 2	1	2	3	2
J,K,L	[(J + 2K + L) + 2] >> 2	3	4	3	2
I,J,K	[(I + 2J + K) + 2] >> 2	4	6	3	2
I,J,M	[(M + 2I + J) + 2] >> 2	3	6	3	2
A,I,M	[(A + 2M + I) + 2] >> 2	3	8	3	2
A,B	[(B + 2A + M) + 2] >> 2	3	6	3	2
A,B	[(A + B) + 1] >> 1	2	3	2	1
A,M	[(M + A) + 1] >> 1	1	2	2	1
B,C	[(B + C) + 1] >> 1	2	4	2	1
C,D	[(C + D) + 1] >> 1	2	3	2	1
D,E	[(D + E) + 1] >> 1	1	2	2	1
E,F	[(E + F) + 1] >> 1	1	1	2	1
G,H	[(G + 2H + H) + 2] >> 2	1	1	3	2
I,J	[(I + J) + 1] >> 1	2	3	2	1
I,M	[(M + I) + 1] >> 1	1	2	2	1
J,K	[(J + K) + 1] >> 1	2	4	2	1
K,L	[(K + L) + 1] >> 1	2	3	2	1
K,L	[(K + 2L + L) + 2] >> 2	1	2	3	2
L	[L]	1	6	0	0

Tablo 2: 4x4 Çerçeve İçi Kipleri ve Kullandıkları Pikseller

4x4 Çerçeve İçi Kipleri	Pikseller
Vertical	A, B, C, D
Horizontal	I, J, K, L
DC	A, B, C, D, I, J, K, L
Diagonal Down Left	A, B, C, D, E, F, G, H
Diagonal Down Right	A, B, C, D, I, J, K, L, M
Vertical Right	A, B, C, D, I, J, K, M
Horizontal Down	A, B, C, I, J, K, L, M
Vertical Left	A, B, C, D, E, F, G
Horizontal Up	I, J, K, L

Tablo 3: Öngörü Denklemlerindeki Piksellerin Eşit ve Benzer Olduğu 4x4 Blokların Yüzdesi

4x4 Öng. Denklemler	PEİMA									PBİMA (4bk)								
	Foreman			Akiyo			M&D			Foreman			Akiyo			M&D		
	NP	NP	NP	NP	NP	NP	NP	NP	NP	NP	NP	NP	NP	NP	NP	NP	NP	NP
28	35	42	28	35	42	28	35	42	28	35	42	28	35	42	28	35	42	
A,B,C	51.7	72.3	85.9	66.5	78.8	87.8	60.3	78.2	90.8	73.7	81.5	90.9	79.4	84.8	92.3	78.0	85.8	94.4
B,C,D	52.1	72.2	86.1	66.6	79.4	88.2	62.0	79.8	91.5	74.0	81.0	90.4	78.8	84.4	92.3	77.4	85.9	94.3
C,D,E	31.0	43.5	50.0	46.4	53.9	58.5	42.6	53.9	58.5	71.4	72.3	73.6	77.4	77.5	78.2	76.0	79.2	79.6
D,E,F	46.2	53.0	53.5	57.3	60.6	61.7	54.3	60.8	60.7	76.2	76.5	76.0	82.2	81.4	80.2	80.4	81.1	81.3
E,F,G	67.5	81.4	90.4	76.8	84.7	91.1	72.2	84.5	92.2	82.8	88.0	94.4	85.7	89.5	94.7	85.0	89.6	95.6
F,G,H	67.8	81.4	90.5	76.7	84.9	91.2	73.4	85.2	92.6	83.1	87.6	94.1	85.2	89.1	94.6	84.3	89.9	95.4
J,K,L	49.0	67.4	81.0	69.0	82.2	91.3	63.9	81.0	91.8	71.3	78.5	87.3	83.7	88.3	94.8	80.7	87.4	94.9
I,J,K	48.1	66.4	80.4	67.8	81.6	90.7	62.9	79.3	91.0	70.8	78.7	86.4	83.6	88.1	94.3	79.7	87.5	94.4
I,J,M	20.9	27.9	27.9	35.3	39.5	41.5	37.8	43.0	40.1	63.6	64.3	63.6	74.5	74.2	73.3	73.1	73.2	74.8
A,I,M	10.4	13.7	12.4	19.8	22.0	24.3	22.2	25.2	24.9	58.3	54.4	50.4	68.4	63.8	58.4	66.8	63.4	61.3
A,B,M	20.6	29.7	29.5	37.3	40.2	40.5	31.5	40.4	39.6	64.9	65.5	64.8	73.7	73.1	70.2	71.5	71.9	72.7
A,B	55.3	74.9	88.1	68.8	80.3	88.9	63.4	80.8	91.9	83.2	87.8	93.5	87.0	89.8	93.6	86.5	91.7	95.9
A,M	24.0	31.6	30.4	39.8	41.8	41.8	34.3	42.4	40.6	72.2	69.7	66.5	79.6	76.8	71.8	77.9	75.6	74.2
B,C	57.1	74.6	87.3	70.2	81.1	89.4	65.2	81.8	92.4	82.5	86.4	92.6	86.0	88.5	94.1	85.6	89.8	95.6
C,D	55.4	73.8	87.1	68.6	81.0	89.4	65.2	81.5	92.6	83.8	87.4	92.6	86.7	89.5	93.8	87.0	92.4	95.7
D,E	48.5	54.3	54.2	59.2	61.7	62.7	56.2	62.2	61.6	81.2	79.2	77.2	86.2	84.0	81.4	84.7	83.8	82.5
E,F	70.1	83.2	92.1	78.3	85.9	92.0	74.6	86.6	93.2	89.2	92.4	96.2	91.2	92.8	95.6	90.6	93.8	96.8
G,H	70.1	82.4	91.3	78.4	86.3	92.2	75.7	86.6	93.7	89.6	92.0	95.7	90.6	92.5	95.7	90.9	94.5	96.5
I,J	51.6	69.8	82.5	69.3	82.7	91.5	65.8	81.0	91.9	81.7	86.0	89.5	90.5	92.5	95.7	87.6	92.2	95.8
I,M	24.6	29.8	29.1	37.8	40.3	42.3	40.2	44.7	41.0	72.3	69.3	66.1	79.2	77.2	74.4	78.8	76.2	75.8
J,K	53.5	69.4	82.0	71.3	83.4	92.2	66.4	82.2	92.5	80.8	84.3	90.0	89.0	91.5	95.8	86.8	91.1	95.5
K,L	53.4	71.7	83.5	72.1	84.3	92.9	67.9	84.2	93.3	81.2	86.1	90.3	90.9	93.3	96.5	88.5	92.1	96.3

Tablo 4: CIF Boyutundaki bir Çerçeve için Yapılan Toplama ve Kaydırma İşlemleri

NP	4x4 Çerçeve İçi Öngörü		[3]'te Önerilen PEİMA Tekniği ile 4x4 Çerçeve İçi Öngörü		Veri Tekrar Kullanımlı 4x4 Çerçeve İçi Öngörü		Veri Tekrar Kullanımı ve PEİMA Tekniği ile 4x4 Çerçeve İçi Öngörü	
	Toplama	Kaydırma	Toplama	Kaydırma	Toplama	Kaydırma	Toplama	Kaydırma
28	884183	529181	652779	400120	417997	230839	186593	101778
35	884183	529181	602186	372580	417997	230839	136000	74238
42	884183	529181	571858	356330	417997	230839	105672	57988

Tablo 5: CIF Boyutundaki bir Çerçeve için Yapılan Toplama ve Kaydırma İşlemleri

NP	4x4 Çerçeve İçi Öngörü		[4]'te Önerilen PBİMA Tekniği (4bk) ile 4x4 Çerçeve İçi Öngörü		Veri Tekrar Kullanımlı 4x4 Çerçeve İçi Öngörü		Veri Tekrar Kullanımı ve PBİMA Tekniği (4bk) ile 4x4 Çerçeve İçi Öngörü	
	Toplama	Kaydırma	Toplama	Kaydırma	Toplama	Kaydırma	Toplama	Kaydırma
28	884183	529181	419857	250334	417997	230839	87142	46413
35	884183	529181	327753	195549	417997	230839	63142	33788
42	884183	529181	329238	196688	417997	230839	63669	34103

Tablo 6: Veri Tekrar Kullanımlı 4x4 Çerçeve İçi Öngörüsünde PEİMA ve PBİMA Tekniklerinin Etkileri

N P	PEİMA Tekniği				PBİMA Tekniği (4bk)				
	Top. Azalma		Kay. Azalma		Top. Azalma		Kay. Azalma		
	#	%	#	%	#	%	#	%	
FM	28	196164	46.9	109839	47.6	314058	75.1	176142	76.3
	35	256231	61.3	142592	61.8	330313	79.0	184022	79.7
	42	292404	70.0	162008	70.2	348193	83.3	193115	83.7
Akiyo	28	256640	61.4	142943	61.9	344143	82.3	191945	83.2
	35	294605	70.5	163398	70.8	353911	84.7	196438	85.1
	42	320555	76.7	177422	76.9	366512	87.7	202769	87.8
M&D	28	241409	57.8	134401	58.2	336906	80.6	188069	81.5
	35	295154	70.6	163814	71.0	354610	84.8	196753	85.2
	42	324017	77.5	179124	77.6	371468	88.9	205386	89.0

PEİMA ve PBİMA tekniklerinin veri tekrar kullanımı 4x4 çerçeve içi öngörüsünde yapılan işlemlerde yaptıkları azaltmalar Tablo 6'da gösterilmiştir. [3]'te önerilen PEİMA tekniği, ve bu bildiriye önerilen veri tekrar kullanımı ve PEİMA tekniği ile elde edilen işlem miktarındaki azalma Tablo 7'de gösterilmiştir. [4]'te önerilen PBİMA tekniği, ve

bu bildiriye önerilen veri tekrar kullanımı ve PBİMA tekniği ile elde edilen işlem miktarındaki azalma ise Tablo 8'de gösterilmiştir. İşlem miktarını veri tekrar kullanımı ile PEİMA tekniği %90, veri tekrar kullanımı ile PBİMA tekniği ise %95 oranında azaltılmaktadır. Bu bildiriye önerilen teknikler, işlem miktarında [3] ve [4]'te önerilen tekniklerden daha fazla azaltma yapmaktadırlar. Buna rağmen hem bu bildiriye önerilen iki teknik, hem de [3] ve [4]'te önerilen teknikler CIF boyutundaki bir çerçeve için sadece 74882 karşılaştırma işlemi yapmaktadırlar.

PBİMA tekniği az miktarda PSNR kaybıyla PEİMA tekniğine göre işlem miktarını daha fazla azaltılmaktadır. Bu nedenle H.264 JM 14.0 referans yazılımını kullanarak önerilen PBİMA tekniğinin 4x4 çerçeve içi öngörü algoritmasının PSNR'ına etkisini belirledik. CIF boyutundaki video çerçeveleri için orjinal 4x4 öngörü algoritması ile PBİMA tekniği kullanılan 4x4 öngörü algoritmasının ortalama PSNR karşılaştırması Tablo 9'da gösterilmiştir. Ortalama PSNR değerleri [13]'te anlatılan metod ile hesaplanmıştır.



Tablo 7: PEİMA Tekniğinin İşlem Sayısı Azaltması

	[3]'teki PEİMA Tekniği					Önerilen Veri Tekrar Kullanımı ve PEİMA Tekniği				
	Toplama Azalması		Kaydırma Azalması		NP	Toplama Azalması		Kaydırma Azalması		
	#	%	#	%		#	%	#	%	
FM	28	246939	27.9	146816	27.7	662350	74.9	408181	77.1	
	35	365863	41.4	216263	40.9	722417	81.7	440934	83.3	
	42	459269	51.9	269710	50.9	758590	85.8	460350	87.0	
Akiyo	28	386890	43.8	229707	43.4	722826	81.8	441285	83.4	
	35	461728	52.2	273099	51.6	760791	86.0	461740	87.3	
	42	521463	58.9	306887	57.9	786741	89.0	475764	89.9	
M&D	28	359883	40.7	214067	40.5	707595	80.0	432743	81.8	
	35	469840	53.1	278673	52.6	761340	86.1	462156	87.3	
	42	539033	60.9	317345	59.9	790203	89.4	477466	90.2	

Tablo 8: PBİMA Tekniğinin İşlem Sayısı Azaltması

	[4]'teki PBİMA Tekniği (4bk)					Önerilen Veri Tekrar Kullanımı ve PBİMA Tekniği (4bk)				
	Toplama Azalması		Kaydırma Azalması		NP	Toplama Azalması		Kaydırma Azalması		
	#	%	#	%		#	%	#	%	
FM	28	410869	46.6	246741	46.7	780244	88.2	474484	89.7	
	35	470764	53.3	282662	53.5	796499	90.0	482364	91.2	
	42	511344	57.9	307139	58.2	814379	92.1	491457	92.9	
Akiyo	28	519800	58.9	311536	59.0	810329	91.6	490287	92.7	
	35	562924	63.8	337540	63.9	820097	92.7	494780	93.5	
	42	586565	66.5	351819	66.6	832698	94.1	501111	94.7	
M&D	28	503723	57.1	301789	57.2	803092	90.8	486411	91.9	
	35	557736	63.2	334094	63.3	820796	92.8	495095	93.6	
	42	603375	68.4	361596	68.5	837654	94.7	503728	95.2	

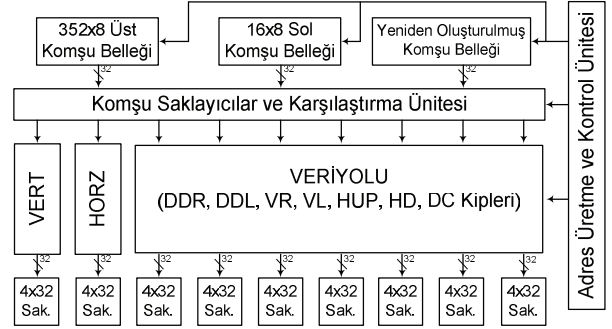
Tablo 9: PBİMA Tekniğinin PSNR'a Etkisi

Çerçeve	Orjinal (dB)	Önerilen PBİMA Tekniği				
		3 bit Kırpma (dB)	$\Delta$ PSNR (dB)	4 bit Kırpma (dB)	$\Delta$ PSNR (dB)	
FM	Y	35.28	35.27	-0.01	35.19	-0.09
AK	Y	37.25	37.21	-0.04	37.12	-0.13
M&D	Y	36.82	36.80	-0.02	36.72	-0.10

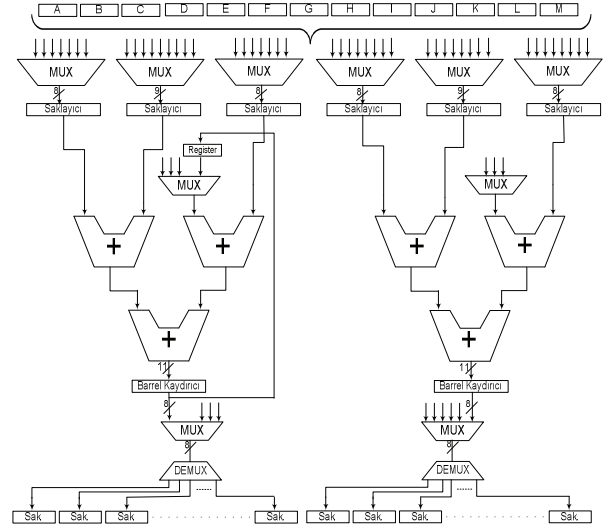
#### 4. Önerilen 4x4 Çerçeve İçi Öngörü Donanımı

Önerilen 4x4 çerçeve içi öngörü donanımının mimarisi Şekil 3'de gösterilmiştir. Bu mimari üç bellek (Üst Komşu Belleği, Sol Komşu Belleği ve Yeniden Oluşturulmuş Komşu Belleği), kontrol ünitesi ve adres üretici, komşu piksel saklayıcıları ve karşılaştırma ünitesi, 3 paralel veriyolu ve öngörü sonuçlarının yazıldığı saklayıcılardan oluşmaktadır. İlk iki veriyolu Vertical ve Horizontal kiplerinin öngörülerini hesaplar. Şekil 4'de gösterilen veriyolu ise DDL, DDR, VR, VL, HD ve HUP kiplerinde kullanılan 23 denklemin ve DC kipinde kullanılan denklemlerin öngörülerini hesaplar.

Üst komşu pikseller üst komşu belleğinde, sol komşu pikseller sol komşu belleğinde ve yeniden oluşturulmuş komşu pikseller yeniden oluşturulmuş komşu belleğinde saklanmaktadır. Yeni bir 4x4 blok geldiğinde gerekli komşu pikseller buldukları belleklerden ilgili komşu piksel saklayıcılarına yazılır ve komşu pikseller arasında karşılaştırmalar yapılır. Karşılaştırma sonuçlarına göre veri yolları her kipin öngörü değerlerini hesaplayıp ilgili 4x32 saklayıcılara yazılır.



Şekil 3: H.264 4x4 Çerçeve İçi Öngörü Donanımı



Şekil 4: DDL, DDR, VR, VL, HD, HUP ve DC Kiplerinin Denklemlerini Hesaplayan Veriyolu

Önerilen donanım Verilog HDL ile gerçekleştirildi ve Mentor Graphics ModelSim SE yazılımı ile simülasyonlar yapılarak doğrulandı. Verilog kodları Mentor Graphics Precision RTL yazılımı ile 2V8000ff1517 Xilinx Virtex II FPGA'ine sentezlendi ve Xilinx ISE Series 8.2i yazılımı kullanılarak aynı FPGA'ye 50 MHz hızında çalışacak şekilde yerleştirildi. Donanım bu FPGA'de 1070 LUT, 497 DFF ve 2 BlokRAM kullanılmaktadır.

#### 5. Güç Kullanım Analizi

4x4 çerçeve içi öngörü donanımının Xilinx Virtex II FPGA'inde 25 MHz'deki güç kullanımı Xilinx XPower yazılımı kullanılarak belirlendi. Bunun için 4x4 çerçeve içi öngörü donanımının ModelSim SE yazılımı ile Akiyo, Foreman, Mother&Daughter video çerçeveleri girdi olarak kullanılarak değişik nicemleme parametresi (NP) değerleri için 25 MHz saat frekansında zamanlama simülasyonları yapıldı. Bu zamanlama simülasyonları sonucunda elde edilen VCD dosyaları kullanılarak donanımın Xilinx Virtex II FPGA'inde harcadığı güç belirlendi ve Tablo 10, 11, 12'de gösterildi. Bu sonuçlar, PEİMA ve PBİMA tekniklerinin 4x4

çerçeve içi öngörü donanımının güç tüketimini sırasıyla %13.7 ve %17.2 oranında azalttıklarını göstermektedir.

## 6. Sonuç

Bu bildiriye, piksellerin eşitliğine ve benzerliğine bakarak H.264 çerçeve içi öngörü algoritmasının işlem miktarını azaltan PEİMA ve PBİMA tekniklerini önerdik. Bu iki teknik, çok az miktarda karşılaştırma yaparak 4x4 çerçeve içi öngörü kiplerinin hesaplanması için gereken işlem miktarını sırasıyla %78 ve %89 oranında azaltmaktadır. PEİMA ve PBİMA teknikleri bir H.264 4x4 çerçeve içi öngörü donanımının güç tüketimini sırasıyla %13.7 ve %17.2 oranında azalttı.

## 7. Teşekkür

TUBITAK'a 106E153 sayılı proje kapsamında bu çalışmayı desteklediği için teşekkür ederiz.

## 8. Kaynakça

- [1] T. Wiegand, G. J. Sullivan, G. Bjontegaard, and A. Luthra, "Overview of the H.264/AVC Video Coding Standard", *IEEE Trans. on CAS for Video Technology*, Cilt: 13, No: 7, Temmuz 2003.
- [2] Joint Video Team of ITU-T VCEG and ISO/IEC MPEG, Draft ITU-T Recommendation and Final Draft International Standard of Joint Video Specification, *ITU-T H.264 and ISO/IEC 14496-10 AVC*, Mayıs 2003.
- [3] M. Parlak, Y. Adibelli and I. Hamzaoglu, "A Novel Computational Complexity and Power Reduction Technique for H.264 Intra Prediction", *IEEE Trans. on Consumer Electronics*, Cilt: 54, No: 4, Kasım 2008.
- [4] Y. Adibelli, M. Parlak and I. Hamzaoglu, "Pixel Similarity Based Computation and Power Reduction Technique for H.264 Intra Prediction", *IEEE Trans. on Consumer Electronics*, Cilt: 56, No:2, Mayıs 2010.
- [5] Joint Video Team of ITU-T VCEG and ISO/IEC MPEG, Joint Model Reference Software, Version 14.0, <http://iphome.hhi.de/suehring/tml>
- [6] E. Sahin, I. Hamzaoglu, "An Efficient Hardware Architecture for H.264 Intra Prediction Algorithm", *IEEE DATE Conference*, Nisan 2007.
- [7] Y. Lai, T. Liu, Y. Li, C. Lee, "Design of An Intra Predictor with Data Reuse for High-Profile H.264 Applications", *IEEE ISCAS Conference*, Mayıs 2009.
- [8] F. Pan, X. Lin, S. Rahardja, K. Lim, Z. Li, S. Dajun Wu, "Fast mode decision algorithm for intraprediction in H.264/AVC video coding", *IEEE Trans. on CAS for Video Technology*, Cilt: 15, No: 7, Temmuz 2005.
- [9] I. Choi, J. Lee, B. Jeon, "Fast Coding Mode Selection With Rate-Distortion Optimization for MPEG-4 Part-10 AVC/H.264", *IEEE Trans. on CAS for Video Technology*, Cilt: 16, No: 12, Aralık 2006.
- [10] Y. Huang, B. Hsieh, T. Chen, and L. Chen, "Analysis, Fast Algorithm and VLSI Architecture Design for H.264/AVC Intra Frame Coder", *IEEE Trans. on CAS for Video Technology*, Mart 2005.
- [11] G. Jin, J. Jung, and H. Lee, "An Efficient Pipelined Architecture for H.264/AVC Intra Frame Processing", *IEEE ISCAS Conference*, Mayıs 2007.
- [12] I. Hamzaoglu, O. Tasdizen and E. Sahin, "An Efficient H.264 Intra Frame Coder System", *IEEE Trans. on Consumer Electronics*, Cilt: 54, No: 4, Kasım 2008.

- [13] G. Bjontegaard, "Calculation of average PSNR differences between RD-curves," *13th Video Coding Experts Group Meeting*, 2001.

Tablo 10: Güç Tüketimi (NP=28)

Çerçeve	Kategori	4x4 Öngörü Donanımının Güç Tüketimi (mW)					
		Orj.	PEİMA	PBİMA (1bk)	PBİMA (2bk)	PBİMA (3bk)	PBİMA (4bk)
FM	Saat	24	21	21.83	21.94	20.79	21
	Lojik	7.29	6.15	6.35	6.18	5.67	5.48
	Sinyal	23.90	22.66	23.00	23.14	20.80	20.02
	Toplam	55.19	49.81	51.18	51.26	47.26	46.50
	Oran		<b>9.76</b>	<b>7.27</b>	<b>7.12</b>	<b>14.37</b>	<b>15.75</b>
AK	Saat	24	21	22	22	21	21
	Lojik	7.06	5.62	5.96	5.74	5.37	5.22
	Sinyal	22.87	20.85	21.22	21.19	19.32	18.56
	Toplam	53.93	47.47	49.18	48.93	45.69	44.78
	Oran		<b>11.99</b>	<b>8.81</b>	<b>9.28</b>	<b>15.29</b>	<b>16.97</b>
M&D	Saat	24	21	22	22	21	21
	Lojik	7.06	5.71	6.01	5.84	5.5	5.27
	Sinyal	22.85	20.85	21.22	21.2	19.54	18.66
	Toplam	53.91	47.55	49.23	49.04	46.04	44.93
	Oran		<b>11.79</b>	<b>8.68</b>	<b>9.03</b>	<b>14.59</b>	<b>16.65</b>

Tablo 11: Güç Tüketimi (NP=35)

Çerçeve	Kategori	4x4 Öngörü Donanımının Güç Tüketimi (mW)					
		Orj.	PEİMA	PBİMA (1bk)	PBİMA (2bk)	PBİMA (3bk)	PBİMA (4bk)
FM	Saat	24	21	22	22	21	20.95
	Lojik	7.16	5.75	6.03	5.85	5.54	5.33
	Sinyal	23.44	21.68	22.04	22.13	20.29	19.52
	Toplam	54.60	48.43	50.07	49.98	46.83	45.80
	Oran		<b>11.30</b>	<b>8.30</b>	<b>8.48</b>	<b>14.23</b>	<b>16.13</b>
AK	Saat	24	21	22	22	21	21
	Lojik	6.98	5.49	5.79	5.57	5.3	5.18
	Sinyal	22.68	20.31	20.61	20.69	19.05	18.39
	Toplam	53.65	46.80	48.4	48.26	45.35	44.57
	Oran		<b>12.77</b>	<b>9.79</b>	<b>10.05</b>	<b>15.48</b>	<b>16.93</b>
M&D	Saat	24	21	22	22	21	21
	Lojik	6.89	5.42	5.71	5.53	5.31	5.21
	Sinyal	22.51	20.07	20.45	20.45	18.98	18.28
	Toplam	53.40	46.49	48.16	47.98	45.29	44.49
	Oran		<b>12.94</b>	<b>9.82</b>	<b>10.15</b>	<b>15.19</b>	<b>16.69</b>

Tablo 12: Güç Tüketimi (NP=42)

Çerçeve	Kategori	4x4 Öngörü Donanımının Güç Tüketimi (mW)					
		Orj.	PEİMA	PBİMA (1bk)	PBİMA (2bk)	PBİMA (3bk)	PBİMA (4bk)
FM	Saat	24	21	22	22	21	21
	Lojik	7.04	5.59	5.79	5.61	5.38	5.29
	Sinyal	23.11	21.05	21.34	21.32	19.67	19.13
	Toplam	54.15	47.64	49.130	48.93	46.05	45.42
	Oran		<b>12.02</b>	<b>9.27</b>	<b>9.64</b>	<b>14.96</b>	<b>16.12</b>
AK	Saat	24	21	22	22	21	21
	Lojik	6.90	5.34	5.61	5.41	5.18	5.11
	Sinyal	22.33	19.71	20	20.02	18.53	17.99
	Toplam	53.23	46.05	47.61	47.43	44.71	44.10
	Oran		<b>13.49</b>	<b>10.56</b>	<b>10.90</b>	<b>16.01</b>	<b>17.15</b>
M&D	Saat	24	21	22	22	21	21
	Lojik	6.83	5.22	5.55	5.38	5.2	5.11
	Sinyal	22.18	19.53	19.8	19.76	18.44	17.81
	Toplam	53.02	45.75	47.35	47.14	44.64	43.92
	Oran		<b>13.71</b>	<b>10.69</b>	<b>11.09</b>	<b>15.80</b>	<b>17.16</b>

## Çok Kipli Ayırık Hücresel Sinir Ağı Modeli

Mehmet Tükel<sup>1</sup> ve ArdaYurdakul<sup>2</sup>

<sup>1</sup>Elektronik ve Haberleşme Mühendisliği Bölümü  
İstanbul Teknik Üniversitesi, İstanbul.  
tukel@itu.edu.tr

<sup>2</sup>Bilgisayar Mühendisliği Bölümü  
Boğaziçi Üniversitesi, İstanbul.  
yurdakul@boun.edu.tr

### Özetçe

Bu çalışmada, gerçek zamanlı görüntü ve video işlemede kullanılan Hücresel Yapay Sinir Ağı'na (HYSA)<sup>1</sup> farklı bir yaklaşım getirilmiştir. Literatürde sayısal işaret işleme için önerilen, temelde kısmi programlanabilirlik ve kaynak paylaşımına dayanan çok kipli tasarım bakış açısıyla, HYSA'yı gerçekleyen yapı üzerinde farklı görüntü işleme algoritmalarını da gerçekleyebilecek bir model önerilmiştir. Modelde, çok kipli tasarımın getirdiği esneklik ve donanım paylaşma özelliği ile HYSA'nın yüksek hızda görüntü işleme için uygun yapısı bağdaştırılmıştır. Sağladığı programlanabilirlikle HYSA'ya farklı algoritmaları da çalıştırabilme özelliği sunan bu model, gerçek zamanlı gömülü sistemlere uygundur. Kırmık üzerinde ağ ile haberleşmeye<sup>2</sup> dayalı, uygulamaya özel çoklu işlemcili bir mimari üzerine kurulu olan bu modelde iki boyutlu HYSA'nın matrisel yapısı korunmuş ve HYSA'daki gibi haberleşmenin komşu işlemcilerle yapılması sağlanmıştır. İşlemcilerin eş zamanlı çalışmasına rağmen haberleşmenin sadece komşu hücrelerle yapılması sayesinde daha kısa zamanda veri alışverişi olmaktadır ve ana bellek üzerindeki yük azaltılmıştır. Bu çalışmada çok kipli HYSA üzerinde HYSA ve HYSA'dan oldukça farklı bir şekilde çalışan Öbek Eşleştirme<sup>3</sup> algoritmasının birlikte tasarımı sunulmuştur. Öbek Eşleştirme için literatürde bir ve iki boyutta kasımlı dizi<sup>4</sup> veya eş zamanlı çalışan işlemcilerle gerçekleştirilen çok çeşitli mimariler mevcuttur. HYSA ile topolojik olarak benzerliğinden faydalanmak amacıyla iki boyutlu bir yapı tercih edilmiştir. Sonuç olarak, bu çalışmadaki model ile aynı elemanların zaman paylaşımını kullanımı sağlayan ve farklı algoritmalar için programlanabilirlik özelliği sunan yeni bir yüksek hızlı görüntü ve video işleme sistemi sunulmuştur.

### 1. Giriş

Görüntü işleme, tıbbi ve askeri uygulamalar başta olmak üzere birçok mühendislik uygulamasında kullanılmaktadır.

Görüntü boyutu, bir imgeciğin<sup>5</sup> ifade edildiği bit sayısı ve/veya görüntü akış hızı arttıkça görüntüyü bilinen yöntemlerle işlemek hayli zaman almaktadır. Günümüzde yüksek çözünürlüklü ve yüksek hızlı videoları işlemek için değişik sistemler önerilmiştir. Bu sistemler temel olarak sayısal işaret işleme için özel kırmıklardan, mikroişlemcilerden, Sahada Programlanabilen Kapı Dizilerinden (SPKD)<sup>6</sup> veya analog kırmıklardan oluşabilmektedir. Özellikle mikroişlemci veya sayısal işaret işleme kırmıkları kullanılan çözümler her zaman istenilen ihtiyaçlara cevap verememektedir. Yüksek hızlarda görüntü işlemeyi gerektiren sistemler için görüntüyü analog olarak işleyen ve sadece zaman sabiti kadar gecikmenin ardından sonuç veren Hücresel Yapay Sinir Ağı (HYSA) uygun bir çözüm olmaktadır [1]. HYSA'yı merkezi işlem birimi olarak alan ve bu birimi yerel ve evrensel bellek birimleri, program belleği gibi çevre birimleriyle birlikte bir bilgisayara dönüştüren CNN-UM<sup>7</sup> [2] 1993 yılında Roska ve Chua tarafından önerilmiştir. Bu yapıda işlem zamanı görüntü boyutundan bağımsızdır. Ancak bu yapı bahsedildiği gibi analog işlemcilerin çalıştığı bir yapıdır. CNN-UM'yi - bilhassa çok hücre olduğu durumlarda - tek kırmık üzerinde gerçeklemek hayli zor olduğu için HYSA'yı gerçekleyen alternatif yollar üzerinde durulmuştur [3-12].

Analog tasarımın zorluğuna karşın dijital tasarımın gürlüğe olan bağımsızlığı ve eleman parametrelerine toleransı gibi avantajları göz önüne alınırsa, HYSA'yı gerçekleyen dijital yapılar hem görüntü işleme hızı açısından hem de tasarım kolaylığı açısından dikkate değerdir. Tasarlanan bir hücresel işlemci kolaylıkla paralel çalıştırılabilir. Bu hücresel işlemcilerin kontrolü birbirinin aynıdır. Ayrıca hücresel işlemcilerdeki blokların tasarımı olabildiğince değiştirilebilir. İşlemci bloklarının içindeki toplayıcı, çarpıcı ve karşılaştırıcı gibi yapılar sistemin veri çözünürlüğüne göre analog tasarımla kıyas edilemeyecek bir hızda yeniden tasarlanabilir. Tasarım kolaylığı ve tasarım esnekliği değerlendirilince HYSA'yı dijital olarak gerçeklemek önemli bir konu olarak karşımıza çıkmaktadır [13].

Bu bildiride sunulan SystemC ile modellenen HYSA'da çok kipli tasarım yöntemi kullanılmıştır [14]. Bu yöntem hız kaybı olmadan, devre alanında en az artmayla programlanabilirlik

<sup>1</sup> İng. Cellular Neural Network (CNN)

<sup>2</sup> İng. Network on Chip (NOC)

<sup>3</sup> İng. Block Matching

<sup>4</sup> İng. Systolic array

<sup>5</sup> İng. Pixel

<sup>6</sup> İng. Field Programmable Gate Array (FPGA)

<sup>7</sup> İng. CNN Universal Machine

sağlayan bir yöntemdir ve ileri sayısal işaret işleme devrelerinde oldukça yaygın kullanılmaktadır [15-17]. Bu tekniğin HYSA'yı diğer algoritmalarla birlikte çalıştıracak şekilde yeni bir görüntü ve video işleme sistemi oluşturmak üzere kullanılmak istenmesinin sebebi, görüntü ve video işlemede kullanılan tekniklerin HYSA'nın yapabildikleriyle sınırlı olmamasıdır. Literatürde çok çeşitli doğrusal ve doğrusal olmayan yöntem görüntü ve video işleme uygulamalarında kullanılmak üzere önerilmiştir. HYSA görüntü veya video üzerinde basamaklı filtreleme işlemleri yaparak istenilen algoritmaları gerçekleyebilmektedir. Bazı algoritmaları yapabilmek için birden fazla filtrenin değişik defalarda çalıştırılması gerekebilmektedir. Sonuç olarak, bu bildiride HYSA'nın görece uzun sürede yapabileceği veya hiç yapamayacağı görüntü işleme algoritmalarını gerçekleyen genel amaçlı bir görüntü işleme sisteminin sayısal işaret işlemede kullanılan çok kipli tasarım yöntemi kullanılarak yeniden tasarlanabileceği gösterilmek istenmiştir.

Bu bildiride anlatılan çalışmada üst seviyeli bir tasarım yöntemi uygulanmıştır. Burada, tasarımın detaylarından ziyade önerinin bir modeli gösterilmek istendiği için HDL'den<sup>1</sup> daha yüksek seviyeli bir modelleme dili olan SystemC kullanılmıştır. SystemC genel kabul gören bir sistem modelleme dilidir. C++ kütüphaneleri kullanılarak oluşturulan bu dil, donanım ve yazılım birlikte tasarımlarını yüksek seviyeli olarak modellemede kullanılır ve modelin benzetimini yapmak için bir çekirdeğe<sup>2</sup> sahiptir.

HYSA'nın yapamayacağı bir algoritma olarak Öbek Eşleştirme algoritması seçilmiştir. Öbek Eşleştirme video sıkıştırma ve hareket kestirimi gibi değişik konularda kullanılmaktadır. İşlemsel yoğunluğu nedeniyle genellikle donanım üzerinde gerçekleşir. Literatürde birçok gerçekleştirilmesi mevcuttur ancak bu çalışmada HYSA ile topolojik benzerlik gösteren mimariler üzerinde durulacaktır. HYSA'da ve Öbek eşleştirme algoritmasında iletişim sadece hemen yandaki birimlerle yapılmaktadır. Böylece aynı haberleşme ağı kullanılabilir. Ayrıca iki algoritmanın da ihtiyaç duyduğu temel aritmetik birimler aynıdır. Böylelikle çok kipli tasarım yönteminde uygulanan donanım paylaşımı ve programlanabilirlik hem veri alışverişi hem de aritmetik birimde kullanılmıştır. HYSA ve Öbek Eşleştirme algoritmalarını gerçekleyen yapıların ayrı ayrı gerçekleşmesi yerine çok kipli tasarım yöntemiyle paylaşımlı kullanılması sonucu işgal edilen devre alanı ve devrenin harcadığı güç açısından kazanç sağlanmaktadır. Bu kazanç sonucu devre çalışma hızında bir azalma olmamaktadır.

Bu bildirinin devam eden bölümleri şu şekilde düzenlenmiştir: İkinci bölümde, bu bildiriyle yakından ilgili çalışmalar kısaca anlatılmıştır. Üçüncü bölümde bu bildiride önerilen görüntü ve video işleme sisteminin modeli sunulmuştur. Dördüncü bölümde sonuçlar verilip yorumlarla bildiri tamamlanmıştır.

## 2. İlgili Çalışmalar

Sayısal HYSA için önerilen oldukça çeşitli yöntem literatürde mevcuttur. Bunların bir kısmı iki boyutta, bir kısmı da bir boyutta çalışan hücrel işlemcilerden oluşmaktadır. Bu çalışmada iki boyutlu HYSA modeli kullanıldığı için literatürdeki iki boyutta yerleştirilmiş hücrel işlemcilerden oluşan ve iki boyuttaki işlemcilerin de birbirleriyle veri

alışverişi yapabildiği bir mimariden bahsedilecektir. Burada ayrıca Ayrık Zamanlı HYSA'nın matematiksel modeli verilecektir.

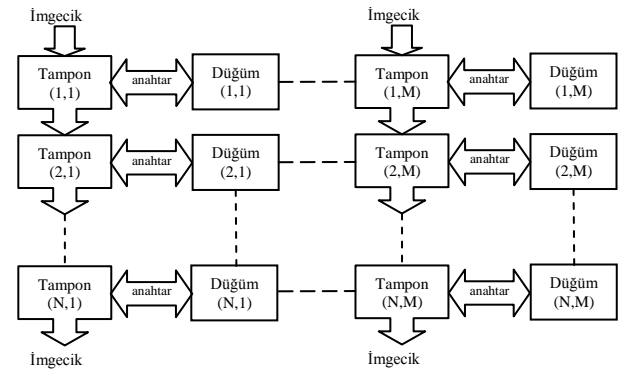
Çok kipli sayısal işaret işleminin tanıtılması için işaret akış çizgeleriyle gösterilen bazı basit uygulamalar gösterilecektir. Son olarak da, Öbek Eşleştiriminin matematiksel tanımı verilip bu çalışmayla ilgili olan bir mimariden bahsedilecektir.

### 2.1. Sayısal HYSA Mimarileri

Ayrık zamanlı HYSA'nın matematiksel tanımı şu şekildedir:

$$x(k+1) = \sum a_d y^d(k) + \sum b_d u^d(k) + i; d \in N, \quad (1)$$

Burada,  $x(k+1)$  bir sonraki durumu,  $y^d(k)$  komşuluk içindeki imgeciklerin şu anki çıkışlarını,  $(a_d, b_d)$  A ve B şablonlarının elamanlarını,  $u^d(k)$  giriş imgecik değerlerini,  $i$  eşik değerini temsil etmektedir [11]. Bu matematiksel tanımın iki boyutta yerleştirilen hücrel işlemcilerle (burada düğüm diye adlandırılmıştır) gerçekleştirilen bir örneği Şekil 1'de gösterilmiştir.



Şekil 1: İki boyutlu sayısal HYSA mimarisine bir örnek.

Şekil 1'de görüldüğü gibi, görüntü bir tampon yardımıyla saklanmakta ve diğer birimlere iletilmektedir. Burada ilk giren-ilk çıkar türü bir veri iletimi söz konusudur. Düğüm (i,j) diye adlandırılan birimler sayısal HYSA'da nokta çarpımı<sup>3</sup> gerçekleyen birimdir, toplama ve çarpmayı yapan aritmetik birim ve hücrel kontrolden oluşur. Anahtar diye adlandırılan birimler tamponla düğüm arasında veri aktarımı yapılacağı zaman açılıp kapanan birimlerdir. Böylece kırkık üzerinde ağ haberleşmesi yapılmaktadır. Anahtarların ve yönlendiricilerin yardımıyla veri akışının iki boyutta da olabileceği kaynak [11] de gösterilmiştir.

Tamponlar ilk giren ilk çıkar yöntemiyle veriyi almaktadır. Ağdaki tüm tampon dolunca tamponlarla düğümler veri alışverişi yapmaktadır. Düğümler kendilerindeki verilerle bir önceki durum/imgecik için nokta çarpım hesaplamasını yapıp beklemektedir. Tampon dolduğunda ise hesaplanmış veri ile yeni veriler takas edilir ve yeni iterasyon hesaplanmaya başlanır. Ayrıca ağdaki tampon yine aynı şekilde hesaplanmış verileri aşağı doğru ileterek ana belleğe yazılmasını sağlar. Burada önemli olan tamponun dolma zamanıdır. Tamponun dolma zamanı düğümlerin hesaplama zamanından büyük olduğu sürece ve takas sırasında bir bekleme olmadığı kabul edildiğinde hesaplamadan kaynaklanan bir bekleme söz konusu değildir. Düğümlerin çalışması 5 grup halinde yapılmaktadır. Her bir düğüm teorik olarak 2 saat çevriminde bütün komşu verilere sahip olabilir ancak 5'li gruplamayla birlikte bu süre 10 saat çevrimi olmaktadır [11].

<sup>1</sup> İng. Hardware Description Language

<sup>2</sup> İng. Kernel

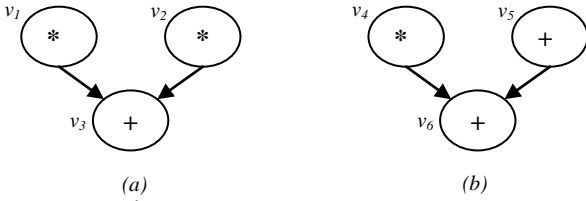
<sup>3</sup> İng. Dot product

## 2.2. Çok Kipli Sayısal İşaret İşleme

Çok kipli tasarımın temel amacı aynı tasarımın zaman paylaşımı olarak birden fazla işaret akış çizgesini gerçekleştirebilmesidir. Bu yapılırken de devre çalışma hızının aynı olması ve/veya devre alanının mümkün oldukça artırılmaması istenmektedir. Böylelikle programlanabilirlik ve donanım paylaşımı sağlanmış olmaktadır.

Üstte bahsedilenlerin yapılması için literatürde değişik yöntemler kullanılmaktadır. Bunlardan bazıları şunlardır: esnek aritmetik birimlerin kullanımıyla ortak işaret akış çizgeleri elde etmek [15], [16] ve elde edilen ortak işaret akış çizgelerinin bir çizelgeleme algoritması dahilinde aynı devre üzerinde çalıştırılması [17]. Temel olarak üstteki maddelerle ifade edilebilecek çok kipli mimarilerde çizge birleştirme, benzer çizge bulma için de çalışmalar yapılmaktadır. Ancak burada sadece bazı örneklerle yetinilecektir.

Çarpma ve toplama işlemini birlikte gerçekleştirebilen bir birimin ismi Esnek Aritmetik Birim (EAB) olsun, 1 çarpıcıdan devre alanı olarak daha büyük ve 1 çarpıcı + 1 toplayıcıdan daha küçük olduğu kabul edilsin. Şekil 4'teki iki işaret akış çizgesi incelendiğinde EAB kullanmanın ortak çizge oluşturma ve devre alanından kazanma gibi faydalarının olduğu görülecektir.

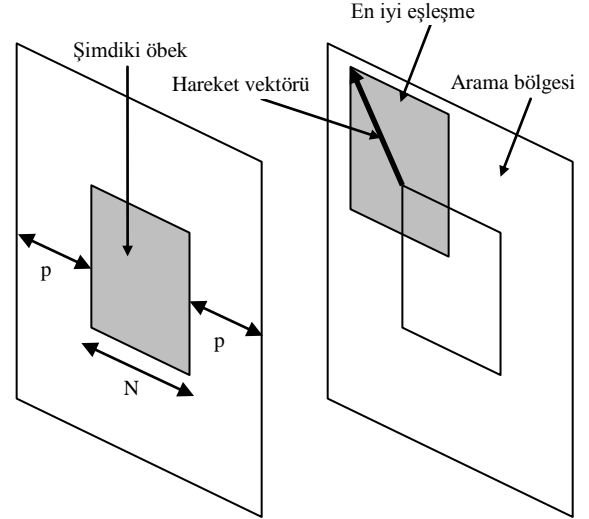


Şekil 2: İşaret akış çizgeleri, çizge (a) ve çizge (b)

Şekil 4'te görülen iki çizge ayrı ayrı gerçekleştirildiğinde çizge (a) için 2 çarpıcı ve 1 toplayıcı, çizge (b) için 1 çarpıcı ve 1 toplayıcı olmak üzere toplam 3 çarpıcı ve 2 toplayıcıya ihtiyaç vardır. Hâlbuki çizgeler aynı devre üzerinde gerçekleştirilebilirler.  $v_1$  ve  $v_4$  için bir tane çarpıcı,  $v_2$ ,  $v_3$ ,  $v_5$  ve  $v_6$  için de bir tane EAB kullanıldığı durumda toplam devre maliyeti 1 EAB ve 1 çarpıcı olmaktadır. Çizge (a) ve (b) nin aynı zamanlarda çalıştırıldığı ve bir çizgenin iki çevrimde gerçekleştirilebileceği kabul edilmektedir. EAB kullanılmasının bir kazancı da, çizgeler arasında geçiş olduğunda toplayıcı girişlerinin ( $v_3$ ,  $v_5$ ) çarpıcı veya toplayıcıdan seçilmek zorunda olmamasıdır. Böylelikle bir çoğullayıcıya da ihtiyaç kalmamaktadır.

## 2.3. Öbek Eşleştirme

Hareket tahmini video işlemede resimler (çerçeveler) arası kestirimsel kodlamada kullanılır ve hesaplama anlamında video işlemede en yoğun işlemlerden biridir. Hareket tahmininde birbirini takip eden iki resim analiz edilir ve imgeciklerin (bir öbek imgeciğin) hareket vektörü kestirilir. Bir andaki resim yerine onun hareket vektörleri iletilir. Hareket vektörlerinin baskılanmasıyla o andan önceki resimlerle iyi bir şekilde sıkıştırılma yapılabilmektedir. Basitliği sebebiyle Öbek Eşleştirme, hareket tahmini için en çok tercih edilen algoritmadır. Öbek eşleştirmede takip eden iki görüntü  $N \times N$  öbeklere ayrılır ve her bir öbek içindeki imgeciklerin aynı hareket vektörüne sahip olduğu varsayılır. Şekil 3'te Öbek Eşleştirme ve hareket vektörü hesaplaması görülmektedir.



Şekil 3: Öbek Eşleştirme ve hareket vektörü hesabı.

Video akışında bir önceki resimdeki araştırılan bölge, araştırma penceresi olarak adlandırılır,  $\pm p$  imgecik ile sınırlandırılır. Böylece araştırma penceresi  $(n+2p)^2$  tane imgecikten oluşmaktadır. En iyi eşleşmeyi bulmak için birçok değişik koşul uygulanabilir. Bunlardan bazıları karşı-ilişki fonksiyonu, ortalama karesel hata ve ortalama mutlak farktır. Ortalama mutlak fark basitliği sebebiyle birçok uygulamada tercih edilmiştir. Öbek eşleştirmenin detayları için kaynak [18]'ya bakılabilir.

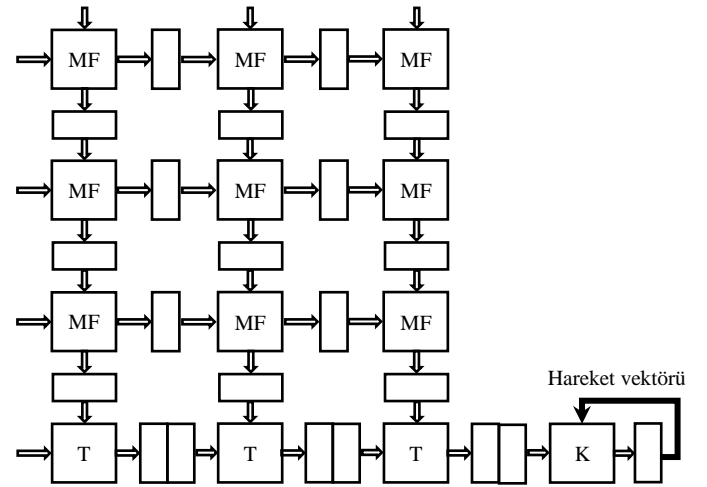
Blok eşleştirmenin matematiksel tanımı şu şekilde verilebilir:

$$s(m, n) = \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} |x(i, j) - y(i + m, j + n)|,$$

$$u = \min_{(m, n)} \{s(m, n)\},$$

$$v(m, n)|_u, -p \leq m, n \leq p.$$

Burada  $x(i, j)$  ve  $y(i, j)$  sırasıyla, o anki resimdeki referans öbeğin ve önceki resimdeki aday öbeğin imgecik değerleridir. İki boyutta işlemel birimlerin yerleştirildiği bir mimari Şekil 4'te gösterilmiştir.



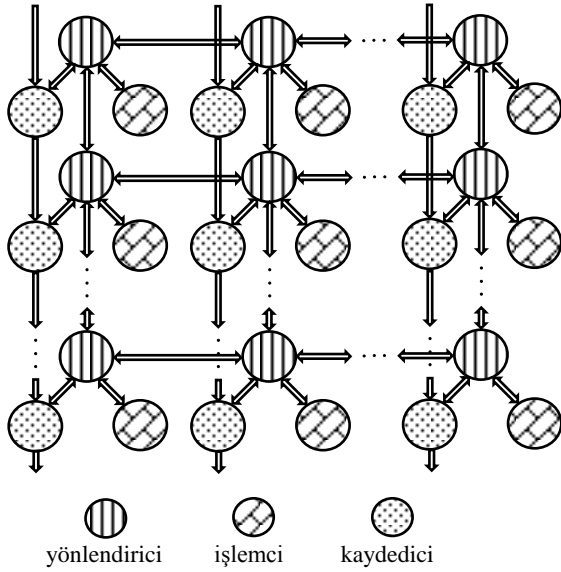
Şekil 4: Öbek Eşleştirme yapan bir mimari örneği

Şekil 4'teki mimaride MF: mutlak fark alan birim; T, toplayıcı birim; K ise karşılaştırıcıdır. İçi boş kutular kaydedicileri temsil etmektedir. Bu şekil için  $N=3$ ,  $p=2$ 'dir.

### 3. Çok Kipli Ayrık Zamanlı HYSA

Bu bölümde, bir kipte Sayısal HYSA bir kipte de Öbek Eşleştirme için çalışacak yapının modeli sunulacaktır. Önerilen model temel olarak üç ana modülden oluşmaktadır. Bu modüller, yönlendirici, işlemci ve kaydedici diye isimlendirilmiştir. Üçünden oluşan üst birime de düğüm denilecektir. Şekil 5'te görüleceği üzere düğümler düşey ve yatay olmak üzere iki eksende yerleştirilmiştir. Herbir düğümün bir yanındaki düğümle direkt bağlantısı ayrıca diğer düğümlerle de dolaylı bağlantısı vardır.

Kaydediciler sisteme veri (imgelik ve durum değerleri) taşınmasını sağlamaktadır. İşlenmemiş veriyi getirip, işlenmiş veriyi almaktadırlar. Bu sistemde haberleşme yönlendiriciler vasıtasıyla yapılmaktadır. Herbir yönlendirici girişine gelen paketteki adrese göre veriyi yönlendirmektedir. Böylece veriyi gönderenin adresi de birlikte göndermesi gerekmektedir. Paket adresine göre yönlendirilerek varacağı hedefe ulaşacaktır. Bu yönüyle yönlendiriciler içerik adreslemeli yönlendiricilere benzemektedir. İşlemci iki kipte çalışacaktır bunlardan birincisinde Sayısal HYSA algoritmasını ikinci kipte ise Öbek Eşleştirme algoritmasını uygulayacaktır. Her iki kipte de ortak veriyolu elemanları kullanılacağı için devre alanındaki artış sadece veriyolu elemanlarının girişlerindeki çoğullayıcıların artırdığı miktar kadar olacaktır.

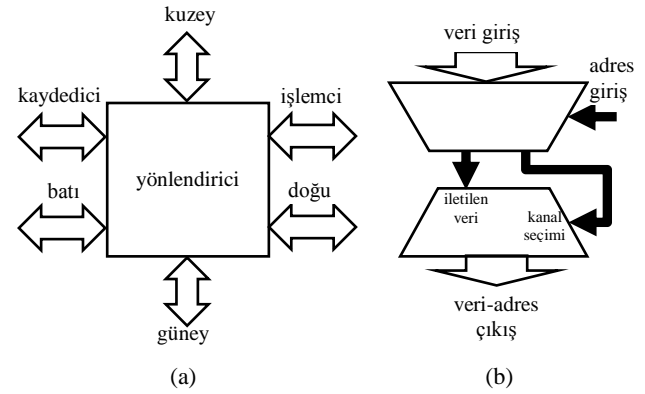


Şekil 5: Modeli verilen çok kipli HYSA'nın blok gösterimi.

#### 3.1. İçerik Adreslemeli Yönlendiricinin Yapısı

Bu çalışmada Şekil 5'te şematik gösterimi verilen sistemde kullanılan yönlendiriciler literatürde anlatılan içerik adreslemeli yönlendiricilerde kullanılan benzer bir iletim tekniğiyle çalışmaktadır. Temel olarak bir yönlendirici kendisine kuzey, güney, doğu, batı, kaydedici ve işlemciden gelen verileri, verilere dâhil olan adres değerleriyle

yorumlayıp verinin nereye gideceğine karar vermektedir. Tüm ağda aynı anda bir yönlendiriciye birden fazla kanaldan veri gelmediği varsayılmaktadır; ancak geçici anlarda oluşabilecek hatalara karşı dayanıklı çalışabilmesi için yönlendiriciye giren kanallar arasında bir öncelik tayin edilmiştir. Bu yönlendirici öncelik sırasına göre aynı anda iki kanal aktif olsa bile öncelikli olan kanalın verisi istenilen kanala aktarılacaktır. Yönlendiricilerin yapısının anlaşılması için daha düşük seviyeli bir çizim Şekil 6'da gösterilmiştir. Doğu, batı, güney, kuzey, kaydedici ve işlemci için birer kanal görülmektedir. Hangi kanalın verisinin nereye aktarılacağı kanalların adreslerinin aktif olup olmamasına bağlıdır. Öncelikli ve aktif olan kanalın verisi, adresinin gösterdiği kanala aktarılır ve ayrıca bir de çıkış adresi üretilir. Bu adres de verinin gönderildiği kanala çıkış adresi olarak gönderilir. Böylece bir verinin bir komşuluktan ileri düğümlere de gidebilmesi sağlanmış olmaktadır.



Şekil 6:a) Yönlendiricinin giriş çıkışları gösterilmektedir. Herbir kanalda (kuzey, güney, vb.) veri giriş ve çıkışı ayrıca adres giriş ve çıkışı vardır. b) Yönlendiricinin giriş ve çıkışları arasındaki ilişki gösterilmektedir. iletilen veri ve kanal seçimi içsel sinyallerdir.

Yönlendirici temel olarak bir çoğullayıcı ve bir çoğullama çözücüdür oluşmaktadır. Bir sonraki düğüme veya kaydedici ve işlemciye gidecek adres-veri çifti aktif ve öncelikli kanaldan çoğullayıcı ile seçilir, sonra çoğullama çözücü ile adres içeriğinin gösterdiği yere yönlendirilir.

#### 3.2. Kaydedicilerle Veri İletimi

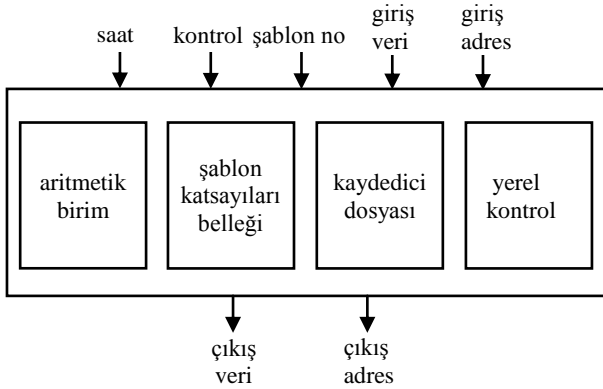
Şekil 5'te görülen kaydediciler sistem çalışırken veri haberleşmesinden etkilenmeden veri aktarımı (işlenmiş verileri ana belleğe, işlenmemiş verileri ana bellekten sisteme aktarımı) yapabilmek için tasarımı koyulmuştur. Bu kaydediciler ilk gelen ilk çıkar mantığında bir iletim yapmaktadır. Kaydedicilerin hepsi (sistemin tamponu) dolduğunda işlenmiş verilerle işlenmemiş veriler takas edilir ve kaydediciler verileri aşağı yönde ileterek işlenmiş verilerin ana belleğe iletimini sağlar.

#### 3.3. İşlemcinin Yapısı

İşlemci temel olarak çok kipli bir aritmetik birim, kaydedici dosyası<sup>1</sup>, şablon katsayıları belleği (HYSA için) ve yerel kontrolden oluşmaktadır. Burada devre alanı açısından en fazla yer kaplayabilecek alt birim, aritmetik birimdir. Bu yüzden aritmetik birimin çok kipli tasarım yöntemiyle

<sup>1</sup> İng. Register File

tasarlanması tercih edilmiştir. Bu yöntem, hızdan ödün vermeden devre alanından kazanç sağlamaktadır, aynı zamanda değişik kipleri gerçekleştirebilmek için programlanabilirlik sunmaktadır. Şekil 7’de işlemcinin alt birimleri ve giriş-çıkışları gösterilmiştir.



Şekil 7: İşlemcinin giriş-çıkışları ve alt birimleri.

İşlemci diğer tüm blokların kullandığı saatle eş zamanlı çalışmaktadır. Kontrol girişi olarak hangi kipte çalıştığına dair bilgileri almaktadır. Böylece aritmetik birim ilgili şekilde programlanarak yapacağı işi belirlemektedir. Sistem sayısal HYSYA algoritmasını gerçeklerken hangi şablonu kullanacağını bilmelidir. Bu bilgi tüm sisteme genel bir sinyalle gönderilmektedir. İstenilirse değişik düğümlere değişik şablon numaraları gönderilerek resimlerin değişik bölgelerinde değişik süzgeçler kullanılabilir. İşlemci veri giriş sinyaliyle hesaplamada kullanacağı veriyi almaktadır. Bu verinin ne olduğu işlemcinin o anki çalıştığı kipe göre değişir. Giriş veri sinyali, sayısal HYSYA kipinde görüntü girişlerini veya durumları alabilir, Öbek Eşleştirme kipinde de kısmi toplamları veya imgecik değerlerini almaktadır. İşlemcinin çalışmasında adres giriş-çıkış sinyallerinin önemi oldukça fazladır. Verinin geldiğine-çıktığına dair el sıkışma sinyallerini üretmesi adres sinyalleri ile olmaktadır. Öyle ki, işlemci sonuçları kaydediciye yazmak istediğinde yönlendiriciye çıkış verisinin yanısıra kaydedici için uygun adresi de göndermektedir.

İşlemci içindeki şablon katsayıları belleği sayısal HYSYA için gerekli şablonları saklayan bellektir. *SystemC* ile iki boyutlu bir dizi olacak şekilde gerçekleştirilmiştir. Devre gerçekleştirilmesi kalıcı bellek şeklinde yapılabilir.

Kaydedici dosyasına kaydedici dizisi denmemesinin sebebi bu ünitenin bir tampon olmadığını anlatmak içindir. Bu birimdeki kaydedicilere ayrı ayrı erişim mümkündür. Aritmetik işlemler için geçici bellek olarak veya işlemcinin girişlerine gelen verileri geçici olarak saklamak için kullanılmaktadır. Buradaki kaydediciler klasik bir işlemcinin kaydedicileriyle benzer görev görmektedir.

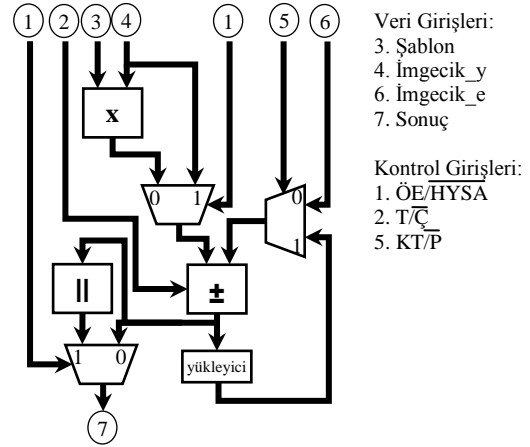
İşlemci içindeki yerel kontrol işlemciye gelen kontrol ve giriş adres sinyalleri gibi kontrol amaçlı sinyalleri işleyerek/çözerek aritmetik birim için daha düşük seviyeli kontrol sinyallerini üretmektedir. Aritmetik birimin detayları Şekil 8’de gösterilmiştir.

Aritmetik birimde bir toplayıcı/çıkarıcı, bir çarpıcı, bir mutlak değer devresi, bir kaydedici ve üç tane çoğullayıcı vardır. Çoğullayıcılar değişik kiplerde çarpıcı ve toplayıcının giriş-çıkışlarını değişik yerlere bağlayabilmek için konulmuştur. Böylece çarpıcı ve toplayıcı gibi çok devre alanı kaplayan

birimlerin farklı kiplerde yeniden kullanılması mümkün olmaktadır.

Aritmetik birimde kontrol girişi sayılabilecek üç tane sinyal vardır. Bu sinyaller iki değer alabilmektedir. Toplam farklı durumların sayısı 8 olmasına karşın bütün kombinasyonlar geçerli olmadığı için aritmetik birim 8’den daha az sayıda durum için koşullanmaktadır.

Şekil 8’de gösterilen, Sayısal HYSYA ve Öbek Eşleştirme algoritmalarını çok kipli yöntemle gerçeklemek için önerilen aritmetik birim bu işi yapabilecek çözümlerden biridir; ancak daha farklı aritmetik birimler tasarlamak da mümkündür.



Şekil 8: Aritmetik birimin kaydedici iletim seviyesinde tasarımı, giriş-çıkış uçları ve aritmetik birimi oluşturan alt birimler.

Aritmetik birimin giriş ve çıkışları kısaca şunlardır:

1. ÖE/HYSYA: Öbek Eşleştirme mi yoksa HYSYA algoritmasının mı çalıştığını gösterir. Öbek Eşleştirme seçili olduğunda HYSYA’yı ilgilendiren kontrol girişlerindeki değişimler anlamsızdır.
2. T/Ç: Toplama veya çıkarma için seçme sinyalidir. Öbek Eşleştirmede hem toplama hem çıkarma, HYSYA’da yalnızca toplama vardır. HYSYA’da bu sinyal çıkarma için koşullanmış olamaz.
3. Şablon: Şablon katsayıları HYSYA algoritması sırasında imgeciklerle çarpımda kullanılır.
4. İmgecik\_y: HYSYA için imgecik girişi, Öbek Eşleştirme için yeni imgecik girişidir.
5. KT/P: HYSYA algoritmasında kısmi toplamın mı, ilk sabit değer mi toplanacağını belirler. Öbek Eşleştirme için imgecik değerini alacak şekilde koşullandırılmalıdır.
6. İmgecik\_e: Öbek Eşleştirme için eski (referans) imgecikleri iletir, HYSYA için iterasyonun ilk adımında sabit değeri iletir.
7. Sonuç: Her iki algoritmanın çalışması durumunda da sonuçları verir.

#### 4. Sonuçlar

Bu çalışmada gerçek zamanlı görüntü işlemede kullanılan sayısal HYSYA devrelerini daha programlanabilir hale getirmek ve HYSYA ile gerçekleştirilemeyen algoritmaları da aynı devre üzerinde çalıştırabilmek için yöntemler önerilmiştir. İki boyutta yerleştirilen yönlendiricilerle içerik

adreslemeli haberleşme tekniği kullanılarak hesaplama zamanında veri iletimi sağlanmıştır. Böylece komşu düğümlerin değerlerine duyulan ihtiyaç, ana geçici belleğe erişilmeksizin giderilebilmektedir. İşlenmiş ve işlenecek verilerin iletimi için yönlendiricilere de bağlı olan bir tampon mekanizması gerçekleştirilmiştir. Tampon, hesaplama devam ederken veri alınmasını ve verilmesini sağlamaktadır.

Aritmetik birimde kullanılan çok kipli tasarım yöntemiyle veri yolu birleşenlerinin kullanımı daha verimli hale getirilmiştir. Farklı kipler için programlanabilirlik kazandırılmıştır. Bu yöntem çalışma hızını düşürmeden, farklı algoritmalar gerçekleştirilerek kullanılan devre alanını azaltma amacıyla burada kullanılmıştır.

Önerilen modelde HYSA ve Öbek Eşleştirme için tasarım sunulmasına karşın, gerek modeldeki haberleşme yöntemi gerekse veri yolu bileşenlerinin kullanımıyla ilgili çok kipli tasarım yöntemi oldukça geneldir/genişdir. İki boyutta kırkık üzerinde haberleşmeli işlemci birimleriyle gerçekleştirilecek farklı görüntü işleme algoritmaları da bu bakış açısıyla bu çalışmadaki model temel alınarak gerçekleştirilebilir.

## 5. Kaynakça

- [1] L.O. Chua ve L. Yang, "Cellular Neural Networks: Theory," *IEEE Transactions on Circuits and Systems*, Cilt: 35, s: 1257-1272, 1988.
- [2] T. Roska ve L. O. Chua, "The CNN Universal Machine: An Analogic Array Computer," *IEEE Trans. Circuits and Systems-II*, Cilt: 40, s: 163-173, 1993.
- [3] P. Keresztes, A. Zarándy, T. Roska, P. Szolgay, P. Hídvégi, P. Jonás ve A. Katona, "An emulated digital CNN implementation," *Int. J. VLSI Signal Processing*, Cilt: 23, s: 291-303, 1999.
- [4] K. A. Wen, J.Y. Su ve C. Y. Lu, "VLSI design of digital Cellular Neural Networks for image processing," *J. of Visual Communication and Image Representation*, Cilt: 5, No: 2, s: 117-126, 1994.
- [5] T. Ikenaga ve T. Ogura, "Discrete-time Cellular Neural Networks using highly-parallel 2D Cellular Automata CAM2," *Proc. of Int. Symp. on Nonlinear Theory and its Applications*, s: 221-224, 1996.
- [6] M. D. Doan, M. Glesner, R. Chakrabaty, M. Heidenreich ve S. Cheung, "Realisation of digital Cellular Neural Network for image processing," 3rd. IEEE International Workshop on Cellular Neural Networks and Their Applications Proceedings (CNNA'94), s: 85-90, 1994.
- [7] Z. Nagy ve P. Szolgay, "Configurable multi-layer CNN-UM emulator on FPGA," *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications*, Cilt: 50, s: 774-778, 2003.
- [8] Z. Voroshazi, Z. Nagy, A. Kiss ve P. Szolgay, "Implementation of embedded emulated-digital CNN-UM global analogic programming unit on FPGA and its application," *International Journal of Circuit Theory and Applications*, Cilt: 36, s: 589-603, 2008.
- [9] K. Kayaer ve V. Tavşanoğlu, "A new approach to emulate CNN on FPGAs for real time video processing," 10th. IEEE International Workshop on Cellular Neural Networks and Their Applications Proceedings (CNNA2008), s: 23-28, 2008.
- [10] S. Malki ve L. Spaanenburg, "CNN image processing on a Xilinx Virtex-II 6000," *Proc. European Conference on Circuit Theory and Design*, s: 261 - 264, 2003.
- [11] S. Malki ve L. Spaanenburg, "A DT-CNN Data-Flow Implementation," 10th. IEEE International Workshop on Cellular Neural Networks and Their Applications Proceedings (CNNA2008), s: 17-22, 2008.
- [12] M. Tükel ve M. E. Yalcin, "A New Architecture for Cellular Neural Network on Reconfigurable Hardware withan AdvanceMemory Allocation Method" 11th. IEEE International Workshop on Cellular Nanoscale Networks and Their Applications Proceedings (CNNA2010), California, USA, 2010.
- [13] M. Tükel, "VHDL ile Hücresel Yapay Sinir Ağı Gerçeklemesi," yüksek lisans tezi, Fen Bilimleri Enstitüsü, İstanbul Teknik Üniversitesi, İstanbul, Türkiye, 2009.
- [14] www.systemc.org
- [15] S. Chiricescu, M. Schuette, R. Ginton ve H. Schmit, "Morphable multipliers," *Proceedings of the International Conference on Field Programmable Logic and Applications*, s: 647-656, 2002.
- [16] V. V. Kumar ve J. Lach, "Highly Flexible Multimode Digital Signal Processing Systems Using Adaptable Components and Controllers" *EURASIP Journal on Applied Signal Processing*, Cilt: 2006, s. 1-9, 2006.
- [17] C. Chavet, C. Andriamisaina, P. Coussy, E. Casseau, E. Juin, P. Urard ve E. Martin, "A Design Flow Dedicated to Multi-mode Architectures for DSP Applications," *IEEE International Conference on Computer Aided Design, (ICCAD)*, s: 604-611, 2007.
- [18] K. K. Parhi, *VLSI Digital Signal Processing Systems*, John Wiley & Sons, Inc., 1999.



# Fiziksel klonlanamayan fonksiyon devrelerinin iyileştirilmesi

Gürkan Kaya<sup>1</sup>, Gökay Saldamlı<sup>2</sup>, M. Erhan Yalçın<sup>1</sup>

<sup>1</sup>Elektronik ve Haberleşme Mühendisliği Bölümü  
İstanbul Teknik Üniversitesi, İstanbul.  
{kayagu,mustak.yalcin}@itu.edu.tr

<sup>2</sup>Yönetim Bilişim Sistemleri Bölümü  
Boğaziçi Üniversitesi, İstanbul.  
{gokay.saldamli}@boun.edu.tr

## Özetçe

Fiziksel Klonlanamayan Fonksiyonlar (physically unclonable functions, PUF) çip üretimi sürecinde malzemeye dayalı olarak çiplerde oluşan küçük farklılıkları bilerek bir araya getiren devrelerden oluşur. Yakın bir zaman önce araştırmacılar PUF kullanarak ürettikleri, çiplere-özel nicelikler (ya da imzalar) sayesinde aynı kalıptan basılan yongaların dahi birbirinden ayırt edilebileceğini gösterdiler. Bu ilerleme, çip üreticilerinin uzun bir zamandır çözüm aradıkları yonga kimliğinin doğrulaması, fikri mülkiyetin (intellectual property, IP) korunması ve güvenli anahtar saklaması gibi problemlerin çözümlerinde umut verici olarak kabul edilmektedir. Sıcaklık ve voltaj dalgalanmaları PUF çıktılarını kötü bir biçimde etkilemekte olup, güvenilir nicelikler üretme işlemini zorlaştırmakta ve sistemin karmaşıklığının artmasına sebep olmaktadır. Bu sorunların irdelendiği bu çalışmada, ilk olarak Spartan 3E FPGA üzerine bir halka osilatör (ring oscillator, RO) tabanlı PUF devresi gerçekleştirilmiş ve devrenin dış etkenlere karşı direnci yeni olarak önerilen bir etiketleme yöntemi ile artırılmıştır. Önerilen yöntem PUF tutarlılığı artırırken devre karmaşıklığını değiştirmemektedir.

## 1. Giriş

Geçmişte, şirketler kendilerine ait olan devre tasarımlarının fikri mülkiyetlerini basitçe kasalarında saklayarak koruyabilecek durumdaydı. Günümüzde ise, devre tasarımında yaygınlaşan dış kaynak kullanımı ve harici şirketlerden hizmet alınması gibi yeni eğilimler, silikon işleme ve tümleşik devre geliştirmeyi daha erişilebilir hale getirdi. Böylece, yonga kimliğinin doğrulaması ve fikri hakların korunması, endüstride gayretle çözüm aranılan gerçek problemler haline geldi.

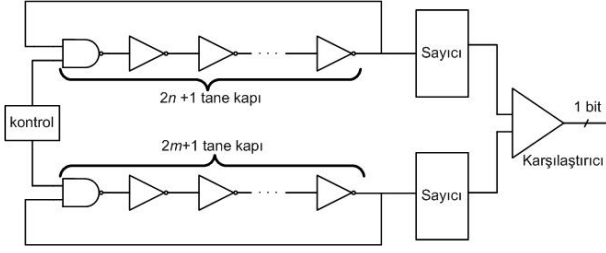
Bu çözümlenmesi zor problemlere getirilen – dikkate alınabilecek– çözümlerin çoğu, sistem tasarımına fazladan külfet yükleyecek karmaşık kriptografik tasarımlar ve yöntemler içermektedir. Dahası, eğer hedeflenen çözüm alanı RFID sistemi gibi kısıtlı bir yapıdaysa, bu ek sıkıntılar daha da katlanmakta ve en etkili çözümler bile kullanışsız hale gelmektedir. Bu nedenle tasarımcılar, muhtemel güvenlik açıklarını da beraberinde getirebilen uygulamaya yönelik özel

yöntemler kullanmaya yönelmektedirler. Halbu ki; yakın zamanda önerilen fiziksel olarak klonlanamaz fonksiyonlar, PUF, sağlam ve değiştirilmez bir yapıya sahip olmaları sebebiyle güvenlik çevrelerinin dikkatini çekmiş ve PUF kullanan bir çok yöntem yukarıda bahsedilen problemlere çözüm olarak önerilmiştir [1, 2, 3, 4, 5, 6].

Cihaza özel bazı kimlik bilgileri oluşturmak için lojik bileşenlerin uyumsuzluklarını kullanmak, yonga üretimi sürecinde malzemeye dayalı olarak çiplerde oluşan küçük farklılıkların kullanılması yolu ile yongaya özel imza oluşturulmasına indirgenebilir. Bu lojik bileşenlerin seçiminde bir kısıtlama olmadığından, değişik lojiklere bağlı farklı PUF devreleri oluşturmak mümkündür. PUF bloklarının gelecekte üretilecek olan hemen hemen bütün yongalarda bulunması oldukça olasıdır ve doğal olarak da şaşmayan PUF devreleri tasarlamak ve bu devreleri sınıflandırmak doğru bir araştırma yönüdür.

Literatürde bulunan SRAM , kelebek, hakem ve halka osilatör tabanlı PUF devreleri en fazla dikkate alınması gerekenler PUF örnekleridir. SRAM PUF [7], yonga çalışmaya başladığında SRAM hücrelerini oluşturan transistörlerin zamanlama farklarını kullanarak imzalar oluşturur, fakat [8]'de bildirildiği gibi SRAM tabanlı PUF'lar, bazı FPGA tiplerinde çalışma başlangıcında genel sıfırlama yapıldığı için FPGA için kullanılamamaktadır. Öte yandan, kelebek PUF [9], bağlaşımlı lojik bileşenlerin uyumsuzluğundan imza oluşturmak için çapraz bağlanmış devre elemanları veya latch kullanır. Hakemli PUF [10], yolların gecikme uyumsuzluklarından belirleyici özellikler oluşturmak için, iki eş ayarlanabilir gecikme yolu kullanır; gecikme yollarının ayarlanması ise hakem adı verilen kontrol devresi tarafından yapılır. Son olarak RO tabanlı PUF [11], RO çıkışının zamanlama farklılıklarını kullanır ve PUF çıktılarını bu ölçümlerle oluşturur. Morozov ve diğerleri [12]'de, karmaşık yönlendirmenin kelebek PUF ve hakemli PUF için uygun simetriye izin vermeyeceği fakat RO tabanlı bir PUF gerçekleştirilmesinin de mümkün olduğunu göstermişlerdir.

Bu çalışmayla amacımız RO tabanlı PUF devrelerinin güvenilirliğini incelemek ve geliştirmektir. Şekil 1, RO tabanlı bir PUF göstermektedir. Bu nispeten basit PUF yapısı, çevresel şartlardan kolaylıkla etkilenebilmekte ve bu beklenmedik etk-



Şekil 1: Halka osilatör tabanlı PUF.

ilerin sonucunda PUF çıktısı güvenilir olabilmektedir. Bu durum, PUF doğrulama aşamasında çıkan ve kesinlikle giderilmesi gereken sorunlar olarak karşımıza çıkmaktadır.

Bu makalenin geri kalan bölümleri şu şekilde özetlenebilir. Bir sonraki bölümde, istenmeyen çevresel etkilerin önlenmesi hakkında daha önce yapılmış çalışmalar ele alındıktan sonra Bölüm 3'te sıcaklık ve voltaj farklılıkları gibi çevresel etkilerin en aza indirgenmesi için önerdiğimiz yöntemin tasarım ve gerçekleştirme detayları açıklanmaktadır. RO çiftlerinin etiketlemesiyle elde edilen iyileştirmeleri gösteren ölçümler, Bölüm 4'te detaylandırılmıştır. Son olarak da Bölüm 5'te elde edilen sonuçla özetlenmiştir.

## 2. Literatür taraması

RO tabanlı PUF kullanarak güvenilir imza üretme işlemi birçok akademik çalışmada işlenmiştir. PUF devresi çevre koşullarındaki farklılıklara göre değişebileceğinden, eskime, sıcaklık, gerilim dalgalanmaları gibi etkilerden kurtulmak için PUF devrelerinin daha karmaşıklaştırılması gerekmektedir.

Suh ve Devadas [11], PUF çıktısı üretmek için  $k$  tane RO çifti çıktısından bir tanesini kullanma metodunu önermişlerdir. Seçilecek RO çifti çıktısının diğerlerine nazaran en yüksek frekans farkına sahip olması gerektiğini vurgulayan araştırmacılar, bu işlemin farklı sıcaklıklarda bit değişiminin engellenmesine yardım edeceğini bildirmişlerdir. Her ne kadar çevresel farklılıklar için bit değişim olasılığı 0.48 olarak verilmiş olsa da, metod  $m$  bitlik PUF çıktısı için  $2km$  halka osilatörüne ihtiyaç duyar.

Maiti ve Schaumont [13], ilintili işlem farklılıklarının PUF çıktılarının çakışma olasılığını arttırabileceğini ileri sürmüşlerdir. İlintiyi engellemek için karşılaştırılan halka osilatörlerini yakın tutmaya çalışmışlardır. Ayrıca, verimliliği artırma amaçlı küçük alan kaplayan RO tabanlı PUF devreleri önermiş ve farklı PUF devrelerini değişik konfigürasyonlarda kullanarak güvenilir PUF çıktıları elde etmek için çalışmışlardır. Çalışmalarının sonuçlarına bakıldığında, 128 bitlik bir PUF çıktısı için gerilim ve sıcaklık farklılıklarının 5 bitin altında bir bozulma yaratacağı görülmektedir. Bu çalışma her RO çifti için 3 bitlik ayar bilgisinin kaydedilmesine ihtiyaç duymaktadır.

Yu ve arkadaşları [14], RO tabanlı bir PUF kullanarak cihaz seri numarası üretimi için bir ortalama alma metodu önermişlerdir. Uyuşmazlık problemlerini azaltmak için [7]'de tanımlanmış olan yaygın kitle merkez düzeneğini kullanmışlardır. Çalışmaları bir bitlik bir PUF çıktısı için

4 halka osilatörüne ihtiyaç duyduğundan, tasarımları bu çalışmadaki önerilen tasarımdan yaklaşık olarak 2 kez daha büyüktür.

Yin ve Qu [15], RO çiftlerini sınıflandırmak için frekans ve gerilim arasında doğrusal bir ilişki kullanmaktadır ve birlikte bazılarını PUF çıktı üretimi için tercih etmektedir. Ancak çalışmaları karmaşık bir son işlem gerektirmektedir.

Bu çalışmada, çevresel farklılıkların etkisini azaltmak için bir sonraki bölümde anlatılacak olan etiketleme yöntemi önerilmiştir.

## 3. Tasarım

Bu çalışmada seçili RO çiftlerinin frekans farklılıklarını kullanan bir PUF devresi tasarlanmıştır. PUF devresinin çıkış bitleri RO çiftlerinden üretildiği için,  $m$  bitlik bir PUF çıktısı üretmek için  $2m$  tane halka osilatörüne ihtiyaç vardır. Test devresi ve etiketleme tekniği bir sonraki iki alt bölümde açıklanmaktadır.

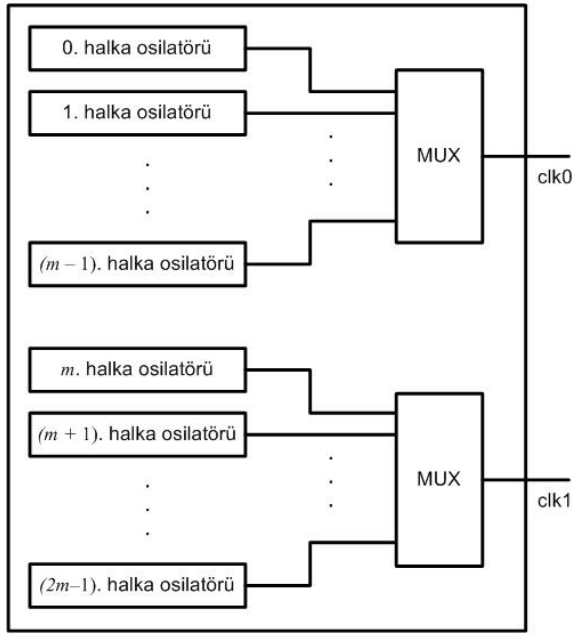
### 3.1. Test Devresi

Tasarımımızda halka osilatörleri ilk olarak iki eş alt gruba ayrılmıştır. Birinci grup referans halka osilatörlerinden oluşurken diğer grup, referans grupla olan frekans farklılıklarını ölçen ikincil halka osilatörleri olarak atanmışlardır. Frekans farklılıklarından üretilen her PUF çıkış biti için bir referans osilatörü bir de ikincil halka osilatörü görevlendirilmiştir. Frekans farklılıklarını ölçmek için ise biri referans osilatöre, diğeri de ikincil osilatöre bağlanmış olan iki sayaç devresi kullanılmıştır. Her iki sayaç da bir durum makinesi vasıtasıyla birlikte saymaya başlarlar; fakat referans sayaç önceden belirlenmiş sabit bir değere ulaştığında dururken, ikincil sayaç referans sayaçın durduğu andaki son saydığı değeri tutar.

En üst seviyedeki durum makinesi halka osilatörlerin etkin sinyallerini kontrol ettiği gibi RO dizisinin çıkışındaki çoğullayıcıyı da kontrol eder. Diğer taraftan, ikinci durum makinesi, sayaçların etkin sinyallerini kontrol eder ve gerektiğinde iki sayacı da aynı anda başlatır. Ne zaman ki; ilk sayaç önceden belirlenmiş sabit değere ulaşırsa o zaman her iki sayacı da durdurur ve ikinci RO sayacının son değeri, PUF çıkışının kararı için çıkışa verilir.

Şekil 2'ten de anlaşılacağı gibi RO dizisinde  $2m$  tane halka osilatörün olmasına karşın her bir bit PUF çıktısının üretiminde biri referans diğeri de ikincil olmak üzere yalnızca iki tanesi kullanılmaktadır.

Gerçeklemede bulunan bütün halka osilatörlerinin aynı karakterlere sahip olduğundan emin olmak için, RO tasarımları FPGA'nin tek bir konfigüre edilebilir lojik bloğunun içine uygulanmıştır. Uygulamamızda halka osilatörleri 7 gecikme elemanı içermekte ve bu yüzden tek bir RO donanım değeri 7 LUT olarak alınır. Çiftler halinde gerçekleştirilmiş olan RO'lar beraber seçildi ve etkinleştirildi. Ayrıca [13]'de tanımlandığı üzere mekansal korelasyondan kaçınmak için birbirine yakın tutuldu. Örneğin, 0 ve  $m$  halka osilatörleri birbirine komşu olarak yer aldı.



Şekil 2: RO dizisi.

### 3.2. Etiketleme tekniği

Klasik yöntemde halka osilatörlerine bağlı olan sayaçların çıkışları karşılaştırılarak karşılaştırıcı devrenin çıkışı PUF çıktısı olarak atanır. Aslında bu işlem referans olarak alınan sayacın sahip olduğu değerden diğer sayacın değerinin çıkartılmasına ve sonucun işaret bitinin de çıkış olarak dışarıya verilmesine eş değerdir. Ancak, çıkarma işleminin sonucunun mutlak değeri ya da aradaki frekans farkının yeterince büyük olmaması durumu bize, çıkış olarak verilen bitin ölçümden ölçüme değişebildiğini gösterir.

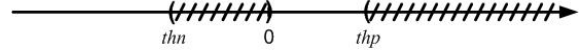
Bu çalışmada RO çiftlerini sadece sonucun işaretine göre değil, aynı zamanda sonucun değerine göre de işaretleyen bir etiketleme tekniği tanımlanmış ve gerçekleştirilmiştir. Yaptığımız deneylere göre RO çiftlerini etiketlemek PUF çıktısının ne olacağına doğru bir şekilde karar verilmesine yardımcı olmaktadır.

### 3.3. Etiketleme İşlemi

Referans ve diğer halka osilatörlerden gelen sayaç değerlerinin farkı bu iki RO arasındaki frekans farkıyla orantılıdır. Eğer aradaki frekans farkı belirlemeci ise bu sonucun sıfırdan büyük olmasının olasılığı 0.5 olmalıdır. Ancak pozitif değerler için bir eşik değeri  $thp$  (çıkarma işleminin sonucunun bu eşik değerinden büyük olma olasılığının 0.25 olduğu ve dolayısıyla sonucun 0 ile  $thp$  arasında olma olasılığının 0.25 olduğu eşik değeri) bulunabilir. Benzer bir şekilde negatif değerler için de bir eşik değeri,  $thn$ , bulunabilir. RO çiftleri PUF çıkışının oluşturulması esnasında '0' ve '1' ile etiketlenir. RO çiftleri, çıkarmanın sonucu  $thp$ 'den daha büyükse ya da  $thn$  ile 0 arasındaysa '0' ile, değilse '1' ile etiketlenir.

Çıkarma işleminin sonucunun dağılımının bir Gauss dağılımı olduğunu varsayarsak,  $thp$  ve  $thn$ , pozitif ve negatif

eşik değerleri olmak üzere Şekil 3'de ring osilatörlerinin nasıl etiketlendiği gösterilmektedir. Daha önce belirtildiği gibi taralı alanlardaki değerleri alıyorsa '0' ile, değilse '1' ile etiketlenir. Eğer çıkış olarak verilen bit, çıkarma işleminin sonucunun işaret biti olarak belirleniyorsa, işaretleme işleminin PUF çıktısı hakkında herhangi bir bilgi vermeyeceği söylenebilir.



Şekil 3: RO çiftlerinin etiketlenmesi.

#### 3.3.1. PUF çıktısının doğrulanması

RO çiftlerinin etiketleri PUF çıktısı hakkında bir bilgi vermemesine rağmen PUF doğrulanması sırasında çevresel etkiler dolayısıyla oluşan bit değeri değişimlerini başarıyla düzeltebilmektedir. Etiketleme yönteminde PUF çıktılarının tekrar oluşturulması şu şekilde gerçekleşir.

- RO çiftlerine bağlı olan sayaçların çıkarma sonucunun hesaplanması.
- Hesaplanan sonucun etiketi asıl etikete denk ise, hesaplanmış sonuç doğru bölgededir, değilse doğru bölgede değildir. Bunun için hesaplanmış sonucun en yakın asıl etiketli bölgeye taşınması.
- PUF çıktısının yeni atanmış bölgeye göre düzeltilmesi. Yeni bölge sıfırdan küçükse çıktı bit değeri '1', değilse '0' olarak alınır.

## 4. Deney sonucu

Uyguladığımız deneylerde 3 farklı Spartan 3E FPGA kartı üzerinde 128 bit PUF oluşturmak için 128 RO çifti kullanılmıştır. Ölçümler bilgisayardan gönderilen bir komutla başlar ve durum makinesi bir referans ve bir önceki bölümde açıklanan ikincil sayaç kullanarak halka osilatörler arasındaki frekans farkını ölçmeye başlar. Durum makinesi ikincil sayaç değerlerini toplamak için tüm halka osilatörleri bir kere taradıktan sonra, bu değerler bilgisayara gönderilir. Ancak, referans sayaç her zaman için önceden belirlenen 0x8000 değerine kadar saydığı için, referans değerini bilgisayara göndermeye gerek yoktur.

Deney kabaca üç bölümden oluşmaktadır. İlk bölümde oda sıcaklığında tüm cihazların değişik PUF üretimleri gözlemlenir. İkinci bölümde ise etiketleme tekniği için  $thp$  ve  $thn$  değerleri hesaplanır. Son olarak, sıcaklığın PUF üretimi üzerindeki etkisini görmek ve etiketleme tekniğindeki gelişimi derecelendirmek için sabit bir sıcaklıkta bir sıcaklık aralığı kullanılarak çevresel ısı 10°C den 65°C ye kadar arttırılır. Deneylerde, 25°C de aynı donanım ayarlarına sahip farklı cihazların değişik PUF değerleri ürettiği görülmüştür. Kullanılan üç cihazın PUF çıkışları Tablo 1 de gösterilmiştir. Ayrıca ölçülen PUF çıkışları arasında hesaplanan ortalama Hamming uzaklığı ise beklenen değere çok yakın bir değer olan, örneğin PUF çıkışının bit uzunluğunun yarısı, 63 olarak hesaplanmıştır.

Deneylerimizde,  $thp$  ve  $thn$  değerlerinin hesaplanması için gereken değişik cihazlar üzerindeki 1024 tane RO çiftinden

Kart numarası	PUF çıkışı
1	6E6BDE74CB8206BBCB00ADDDFF2E8A77
2	AD980885213C7DFFC80F4831F073BB12
3	91EB09330B99B99B0991496DF709A0FE

Table 1: PUF Çıktıları

alınan çıkartma sonuçlarının sıfırdan büyük olma ihtimalinin 0.5381 olduğu bulunmuştur. Bu sonuç bir miktar pozitif tarafa kaymasına rağmen yaklaşık olarak bizim beklediğimiz sonuçtur. Toparladığımız bu değerlerden, test kurulumunda *thp* değeri 164 ve *thn* değeri -150 olarak hesaplanmıştır.

Daha önce bahsedildiği gibi, ısı odası testleri kurulumunun çevresel sıcaklığını kontrol etmek için kullanılır. Tablo 1’de gösterilen değerleri ürettiğimiz PUF üretim fazına göre, tüm FPGA cihazları için RO çiftlerinin etiketleri tanımlanmıştır. Sırası ile 10,25,40,55 ve 65 derece olarak tanımladığımız her bir sıcaklık noktası için, 1000 adet ölçüm yapılmış ve herhangi bir bit eğer; orijinal PUF çıkışı ile ölçülen bir PUF çıkışı doğrulanmaz ise güvenilir veya sağlıklı olarak işaretlenmiştir. Bu ölçümlerin sonucunda bütün sıcaklık değerlerinde etiketlenmiş düzeneklerde herhangi bir sağlıksızlık görülmezken, etiketlenmemiş düzeneklerde tanımlanan sıcaklıklarda sırası ile 8, 8, 8, 14 ve 13 bit sağlıksız olarak tespit edilmiştir.

Deneyler, işaretleme tekniğinin kullanımının, sıkıntı yaratan sıcaklık etkilerine karşı PUF çıktısının güvenilirliğini arttırdığını göstermektedir. Değişik voltajlarındaki işaretleme tekniğinin analizini ise gelecekteki çalışmalara bırakıyoruz.

## 5. Sonuç

Bu çalışmada, Spartan 3E FPGA yongaları üzerinde RO tabanlı PUF devreleri gerçekleştirilmiştir. Aynı bit dizisinin farklı FPGA kartlarında farklı PUF çıktıları oluşturduğu gözlemlenmiştir. Ancak, çevresel değişiklikler nedeniyle PUF çıktısının bazı bitleri istenmeyen bir şekilde değişebildiği ve güvenilir olmayacağı PUF devrelerine sebep verebileceği tespit edilmiştir.

Bu etkilerin üstesinden gelebilmek için bir etiketleme tekniği önerilmiş ve gerçekleştirilen deneylerle önerilen işaretleme tekniğinin tasarımın karmaşıklığını arttırmaksızın PUF çıktısının kalitesini arttırdığı gözlemlenmiştir. Ayrıca, bu teknik ile PUF çıktısı oluşturulduktan sonra gereken karmaşık yapıdaki hata düzeltme çalışmalarının gerçekleştirilmesi de elimine edilmiştir.

## 6. Teşekkür

Katkılarından dolayı Serdar Aktaş, Ahmet Arış, Tuba Ayhan, Murat Cihan, İmran Ergüler ve Çağdaş Şahin’e teşekkür ederiz.

## 7. Kaynakça

[1] L. Bolotnyy and G. Robins, “Physically Unclonable Function-Based Security and Privacy in RFID Systems,” in *Proceedings of the Fifth IEEE International Conference on Pervasive Computing and Communications (PERCOM)*, 2007, pp. 211–220.

[2] B. Gassend, “Physical random functions,” Master’s thesis, Massachusetts Institute of Technology, 2003.

[3] B. Gassend, D. Clarke, M. van Dijk, and S. Devadas, “Silicon physical random functions,” in *Proceedings of Computer Communications Security Conf.*, 2002, pp. 148–160.

[4] D. Lim, “Extracting secret keys from integrated circuits,” Master’s thesis, Massachusetts Institute of Technology, 2004.

[5] D. Lim, J. W. Lee, B. Gassend, G. Suh, M. van Dijk, and S. Devadas, “Extracting Secret Keys from Integrated Circuits,” *IEEE Transactions on VLSI Systems*, vol. 13, no. 10, 2005.

[6] R. Pappu, B. Recht, J. Taylor, and N. Gershen-Feld, “Physical one-way functions,” *Science*, vol. 297, pp. 2026–2030, 2002.

[7] Y. Su, J. Holleman, and B. Otis, “A digital 1.6 pj/bit chip identification circuit using process variations,” *Solid-State Circuits, IEEE Journal of*, vol. 43, no. 1, pp. 69–77, jan. 2008.

[8] R. Maes, P. Tuyls, and I. Verbauwhede, “Intrinsic pufs from flip-flops on reconfigurable devices,” in *3rd Benelux Workshop on Information and System Security (WISec 2008)*, 2008.

[9] S. Kumar, J. Guajardo, R. Maes, G.-J. Schrijen, and P. Tuyls, “Extended abstract: The butterfly PUF protecting IP on every FPGA,” in *Hardware-Oriented Security and Trust, 2008. HOST 2008. IEEE International Workshop on*, 9-9 2008, pp. 67–70.

[10] D. Lim, J. Lee, B. Gassend, G. Suh, M. van Dijk, and S. Devadas, “Extracting secret keys from integrated circuits,” *Very Large Scale Integration (VLSI) Systems, IEEE Transactions on*, vol. 13, no. 10, pp. 1200–1205, oct. 2005.

[11] G. Suh and S. Devadas, “Physical unclonable functions for device authentication and secret key generation,” in *Design Automation Conference, 2007. DAC ’07. 44th ACM/IEEE*, 4-8 2007, pp. 9–14.

[12] S. Morozov, A. Maiti, and P. Schaumont, “A comparative analysis of delay based puf implementations on fpga,” *Cryptology ePrint Archive*, Report 2009/629, 2009, <http://eprint.iacr.org/>.

[13] A. Maiti and P. Schaumont, “Improving the quality of a physical unclonable function using configurable ring oscillators,” in *Field Programmable Logic and Applications, 2009. FPL 2009. International Conference on*, aug. 2009, pp. 703–707.

[14] H. Yu, P. Leong, H. Hinkelmann, L. Moller, M. Glesner, and P. Zipf, “Towards a unique fpga-based identification circuit using process variations,” in *Field Programmable Logic and Applications, 2009. FPL 2009. International Conference on*, aug. 2009, pp. 397–402.

[15] G. Qu and C.-E. Yin, “Temperature-aware cooperative ring oscillator puf,” in *Hardware-Oriented Security and Trust, 2009. HOST ’09. IEEE International Workshop on*, 27-27 2009, pp. 36–42.

# Ultrasonik Temizlik Sistemlerinde Rezonans Frekansı Tespiti İçin Yinelemeli Frekans Bandı Tarama Tekniği

Ulvi Güvenç<sup>1</sup>, Mehmet Yakut<sup>1</sup>, Ali Tangel<sup>1</sup>

<sup>1</sup>Elektronik ve Haberleşme Mühendisliği Bölümü  
Kocaeli Üniversitesi, Kocaeli.  
{ ulvi.guvencl,myakut,atangel}@kocaeli.edu.tr

## Özetçe

Bu makale, ultrasonik temizlik makinelerinin çalışma anı rezonans frekansının tespiti için geliştirilmiş olan yeni bir frekans bandı tarama metodunu açıklamaktadır. Sonuç olarak, bu çalışma ile ultrasonik temizlik sistemlerinde kararsız yük yapısı ve fiziksel koşullar nedeniyle rezonans frekansında gözlenen değişken davranış, yeni geliştirilen periyodik tarama süreci ile sürekli gözlenebilir hale getirilmiştir. Bu sayede sistem sürekli optimize edilerek yük temizleme verimsizliği minimize edilmiştir. Diğer yandan geliştirilen algoritma, önceki metodlara göre getirdiği özgün iki aşamalı tarama tekniği sayesinde daha hızlı, güncellenebilir bir tarama süreci ve daha az lojik kapasite kullanımı nedeniyle yüksek başarıma sahip bir yaklaşım olarak önerilmektedir. Önerilen algoritma Xilinx Spartan3 Fpga donanımı üzerinde gerçekleştirilmiştir.

## 1. Giriş

Ultrason, insan kulağının duyabileceğinin ötesinde frekanslarda yayılan ses dalgaları olarak tanımlanmaktadır[1]. Ultrasonu temizlik amacıyla kullanmanın altında yatan mantık ise şudur: sıvı içerisinde oluşturulan yüksek frekansta(20kHz–400kHz) milyonlarca vakum enerjisi içeren mikroskobik hava kabarcığının kirli yüzeylere çarparak patlaması sayesinde yağ, kireç, pas ve istenmeyen dokuların süratle yüzeyden uzaklaştırılması işlemidir [1], [2], [3], [4]. Kavitasyon sırasında hava kabarcıklarının ne kadar büyük olacağı, ultrasonik dönüştürücülerin(transducer) girişine uygulanan sinyalin genlik ve frekans bilgisine bağlıdır. Bir ultrasonik temizlik kazanının temizleme gücü kabarcıkların ne kadar şiddetli patladığıyla doğrudan ilişkilidir[4]. Bu nedenlerle, ultrasonik temizlik sistemlerinde frekans, güç ve gazdan arındırma(degassing) zamanı kontrolleri büyük önem taşır.

Ultrasonik temizlik sistemleri yıllardır alışlagelmiş mikrodenetleyicili yaklaşımlarla gerçekleştirilmekte idi. Bu konuda geliştirilen FPGA tabanlı yeni yaklaşım, mikrodenetleyici teknolojisine göre pekçok yeniliği beraberinde getirmiştir. Bu yeniliklerden bazıları: yüksek çalışma frekanslarına çıkılabilmesi sayesinde rezonans frekans taramasının çok daha küçük adım aralıklarıyla gerçekleştirilebilmesi, eş zamanlı olarak birden fazla ultrasonik temizlik kazanının sürülebilmesi ve bağımsız, modüler bir kontrol modülünün geliştirilebilmesidir.

Ultrasonik teknolojisinin temizlik alanında kullanılmasında karşılaşılan en büyük problem kararsız yük yapısı ve bunun etkileridir. Yük yapısındaki bu kararsızlık çalışma anı rezonans frekansının değişken bir davranışa sahip

olmasının önemli bir nedenidir. Ancak bunun dışında başka parametreler de rezonans frekansının bu davranışına etki etmektedir. Bir ultrasonik temizlik sisteminin doğal rezonans frekansı; kullanılan ultrasonik dönüştürücü, temizlik kazanı, temizlik sıvısı, temizlik materyali ve de temizlenecek nesnenin kütle ve geometri gibi fiziksel özelliklerinin bütün bir fonksiyonudur. Bu parametrelere pek çoğu temizlik sisteminin yapılandırılmasının ardından sabit hale gelse de, temizlik sıvısı ve temizlenecek nesnenin özelliklerinde meydana gelen değişimler sebebiyle bütün sistemin rezonans frekansının kararlılığının yeniden gözden geçirilmesi gereklidir.

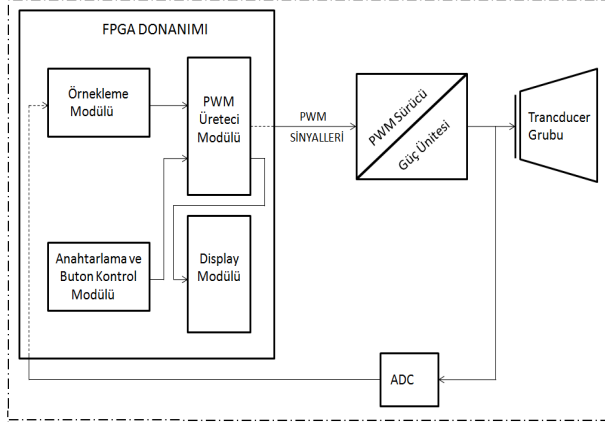
Paralel rezonans devresi kullanıldığında ultrasonik dönüştürücü için güç sürücü devresinin çıkış akımı rezonans frekansında maksimum olur. Rezonans frekansının belirlenmesinde bundan faydalanılır [4], [5]. Ultrasonik güç devrelerinde besleme gerilimi genellikle şebeke geriliminin tam dalga doğrultulmuş halidir. Bu gerilim genellikle içinde güç mosfetleri kullanılan yarım köprü DC/AC tipi dönüştürücü devreleri yardımıyla sinüsoidal sinyal olarak ultrasonik dönüştürücü bloğuna aktarılır [6], [7]. Burada elde edilen AC gerilim 1000Vpp genliğe sahip genlik modüleli bir işaretir.

Rezonans frekansının tespiti ve rezonans frekansında çalışma ile verimli bir temizlik başarılabilir. Diğer yandan, temizlik performansı yük koşulları (uzunluk, boyut ve geometrik şekil) ve temizlik sıvısının özelliklerindeki(sıvının yüzey gerilimi ve sıcaklığı gibi) değişikliklere bağlı olarak azalabilmektedir. Temizlik kazanlarının özellikle yüksek ultrasonik güç seviyelerinde sabit PWM frekansları ile sürülmesi, temizlenecek olan nesnelerin yüzeyinde kavitasyon erozyonu olarak bilinen yüzey aşınmalarına neden olabilir. Bu nedenle bu sistemlerde güç kontrolü önemlidir. Bunun yanında sabit frekanstaki çalışma durumunda komşu dönüştürücülerin yerel maksimum noktaları bu olaya neden olabilir. Sürücü çıkış frekansının önceden belirlenmiş rezonans frekansı etrafında dar bir bantta taranması, temizlik sıvısı içindeki yerel maksimum bölgelerini azaltır. Bu nedenle de bu sistemlerde frekans kontrolü önemlidir. Günümüz modern ultrasonik temizleme sistemlerinde mikrodenetleyicilerin kullanımı, gerekli olan frekans ve güç kontrolünün elde edilebilmesi için tercih sebebidir. Literatürde PWM sinyallerinin üretimi için güç elektroniği devrelerinde FPGA teknolojisinin kullanımı örneklerine rastlanabilir [8], [9]. Ancak, gerçek uygulamaya yönelik FPGA tabanlı ultrasonik temizlik makinesi tasarımları literatürde yer almamaktadır.

## 2. Yinelemeli Frekans Bandı Tarama Tekniđi

Yinelemeli frekans bandı tarama tekniđi, literatürdeki çalışmalara göre Fpga teknolojisi kullanımıyla yeni bir yaklaşım sunmaktadır. Bu yaklaşıma göre, rezonans frekansının tespiti ve rezonansa çalışma süreci periyodik bir yapıdadır. Diğer yandan, daha hızlı bir tarama süreci (iki aşamalı tarama tekniđi) ile yüksek doğrulukta rezonans frekans değeri tespit edilebilmektedir.

Fpga tabanlı ultrasonik temizlik sisteminin ve geri besleme devresinin çalışmasına ilişkin blok diyagram Şekil 1'de verilmiştir. Görüldüğü üzere sistem, üzerinde PWM üretici, display, örnekleme ve anahtarlama/buton kontrol modüllerinin gerçekleştirildiği bir FPGA donanımı, PWM sürücü/güç ünitesi, dönüştürücü grubu ve ADC'den oluşmaktadır. FPGA donanımı sayısal veriler üzerinde işlem yapabildiğinden PWM sürücü/güç ünitesi çıkışının sayısal hale dönüştürülmesi gerekmektedir ve bunun için bir ADC modülünü kullanmak zorunludur. Burada, kullanılan ADC'nin kontrol işlemleri FPGA yongası üzerinde geliştirilmiş olan ayrı bir modül ile gerçekleştirilmektedir.



Şekil 1: Sistem blok diyagramı

Yinelemeli frekans bandı tarama tekniđine göre, tasarlanan geri besleme ünitesi belirli zaman aralıklarında çalışarak sistem çıkışlarından ilgili işaret örnekleri alır. Çalışma prensibi ise şöyledir; taranacak frekans bandı içerisinde bir başlangıç ve bitiş frekans noktaları belirlenir. Belirlenen başlangıç frekans değerinden itibaren belirli bir frekans adım aralığıyla tüm frekans noktaları taranır. Tarama hassasiyetinin ne olacağı kullanıcıdan gelen talebe göre belirlenir ve VHDL kodu içerisinde ilgili değişkenin tuttuğu değer değiştirilerek güncellenebilir. Burada ulaşılabilecek en düşük frekans adım aralığı kullanılan FPGA donanımının teknolojisine doğrudan bağlıdır. FPGA donanımları üzerinde bulunan yerel osilatör frekansları tasarımcıyı sınırlandırmaktadır. Bu çalışma sırasında Xilinx Spartan3 donanımı kullanılmıştır. Bu donanım 50MHz'lik bir yerel osilatöre sahiptir. Ayrıca sistem içerisindeki PWM üretici modülünün çalışma frekansı da belirleyici olmaktadır. Yüksek çalışma frekanslarından düşük frekanslara doğru gidildikçe tarama adım aralığı değeri küçülmektedir. Şöyle ki, 48KHz'de çalışan bir PWM üretici modülü için adım aralığı yaklaşık olarak 46Hz iken çalışma frekansı 35KHz'e çekildiğinde tarama adım aralığı değeri yaklaşık olarak 24Hz olmaktadır. Bu değişken tarama hassasiyetinin nedeni kullanılan frekans bölme algoritmasıdır.

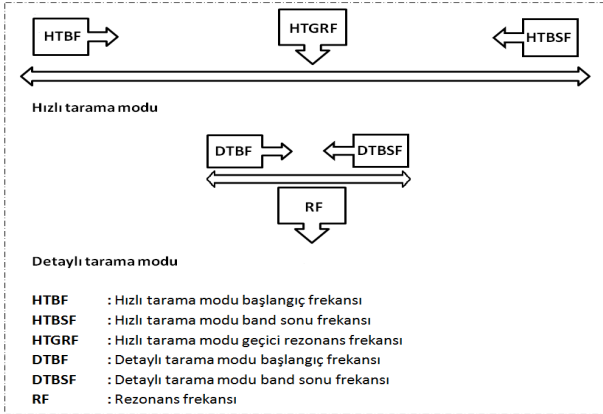
Kullanıcıdan gelen talep doğrultusunda bu değişim hassasiyeti ölçeklendirilebilmektedir.

Tarama sürecinde bir frekans noktasında sistem çalışırken bu frekans değeri için tek bir örnek alınması sağlıklı olmayacaktır. Bundan dolayı her bir frekans değeri için birden fazla sayıda (burada ikiyüzlü tane) örnek alınır ve bu örnek değerleri toplanarak kod içerisinde tanımlı olan bir ara değışkene ara toplam olarak atılır. O anki çalışma frekansından gelen bu ara toplam önceki tarama sonuçlarından elde edilen en büyük toplam ile karşılaştırılır. Eğer o anki çalışma frekansından elde edilen örnekler toplamı mevcut en büyük toplam değerinden büyük ise o anki çalışma frekansı rezonans frekansı olarak belirlenir ve o anki çalışma frekansında elde edilen örnekler toplamı da yeni en büyük toplam değeri olarak belirlenir. Sonrasında, mevcut çalışma frekansı olarak bir sonraki frekans tarama değeri belirlenerek aynı işlemler tekrarlanır. Böylece belirlenen bandda kullanıcıdan gelen talebe ya da sistem gereksinimlerine göre belirlenen adım aralıklarıyla tüm frekans değerleri taranır. Sonuç olarak en büyük örnek toplamını veren frekans değeri sistemin mevcut koşullar altındaki yeni rezonans frekansı olarak belirlenir. Belirlenen bu rezonans frekansı, rezonansa çalışma sürecinde belirli bir süre (kullanıcının isteđine bağlı olarak) kullanılmak üzere PWM üretici modülüne yeni çalışma frekansı olarak gönderilir.

Bir frekans değeri için ikiyüzlü örnek değeri alınarak bunların toplamının işleme tabi tutulmasının nedeni ise şöyle açıklanabilir. Örneklenen işarete çeşitli nedenlerle meydana gelebilecek salınımlar sonucunda o an alınan örnek değeri gerçeklikten uzak olabilir. Tarama sürecinde işaretin tepe değerleri ile ilgilenildiğinden, herhangi bir anda alınan örnek o anda gerçekleşen bir salınım nedeniyle aslında gerçekçi olmamasına rağmen tepe değeri olarak gözlenebilir. Bunu engellemenin tek yolu birden fazla örnek alıp daha sonra bu örnek değerlerinin ortalamasının hesaplanmasıdır. Sonuç itibarıyla ortalama örnek değerlerinin karşılaştırması, tek bir örneğe dayanarak yapılan değerlendirmeden çok daha sağlıklı olacaktır. Bu makalenin konusu olan çalışmada geliştirilen algoritmada ise örnek ortalama değerleri yerine örnek toplamları üzerinden karşılaştırma yapılması tercih edilmiştir. Ortalama hesabından kaçınılmasının sebebi ise kullanılan FPGA donanımının lojik kapasite kullanımını ve işlem yükünü azaltmaktır. Tasarım sürecinde VHDL donanım programlama dili kullanılmıştır. VHDL'de çarpma ve bölme işlemleri konusunda bazı sınırlamalar mevcuttur ve bu sınırlamalar sebebi ile bölme işleminin gerçekleştirilebilmesi için noktalı sayılarda işlemlere girilmesi gerekmektedir. Bilindiğı üzere, sayısal devre tasarımında her yeni lojik kapı ya da eleman kendi içinde bir gecikme etkisi getirmektedir.

Geliştirilen bu yeni tarama tekniđine iki aşamalı tarama isminin verilmesinin sebebi ise iki farklı tarama sürecinin kullanılmasıdır. Bunlar hızlı tarama ve ardından gelen hassas taramadır. İki aşamalı tarama tekniđine dair hızlı ve hassas tarama modları Şekil 2'de gösterilmektedir. Her iki tarama sürecinin de temel mantığı örnek toplamlarının karşılaştırılması esasına dayanmaktadır. Bu iki tarama süreci arasındaki tek fark tarama frekans adım aralığıdır. Tarama sürecinde ilk olarak hızlı tarama modu çalışır. Burada frekans bandı, belirlenen başlangıç ve bitiş frekans değerleri arasında geniş adım aralıklarıyla taranır ve sonuç olarak bant içerisindeki bir frekans değeri aday rezonans frekans noktası olarak tespit edilir. Daha sonra ikincil tarama süreci olan hassas taramaya geçilir. Bu süreçte, hızlı tarama sürecinde

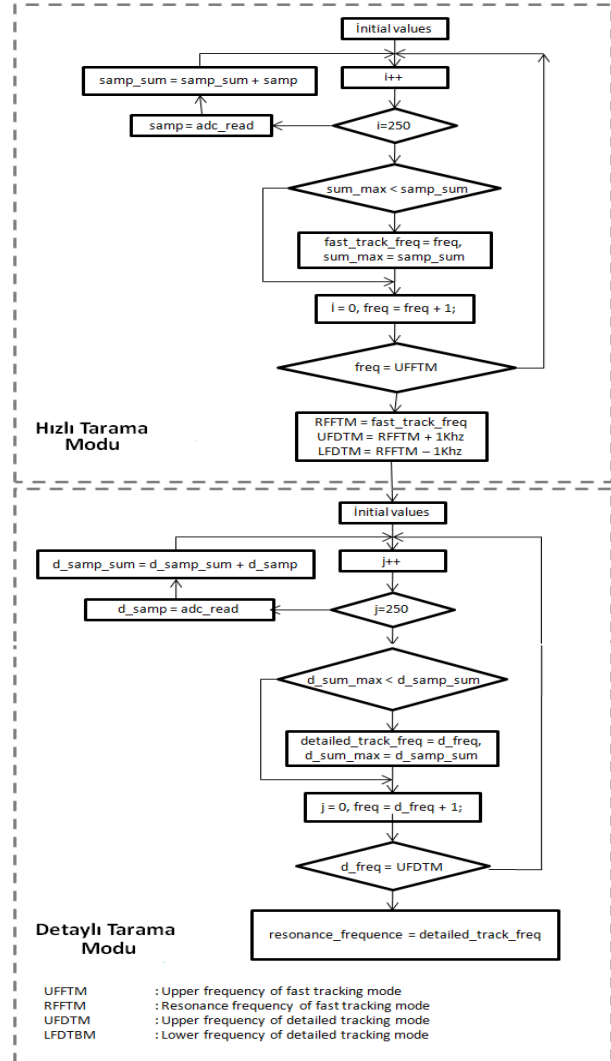
belirlenen aday rezonans frekansı noktasının belirli bir değer kadar altında ve üstünde yeni sınır frekans noktaları tespit edilerek (örnek olarak 35-45 KHz aralığı taranırken hızlı tarama süreci sonucunda 40 KHz noktası belirlendiyse, detaylı tarama bandı olarak 39-41 KHz aralığı belirlenerek detaylı tarama sürecine geçilir) yeni oluşturulan bu ikincil frekans bandında bu kez çok daha küçük frekans adım aralığı ile detaylı bir tarama yapılır. Detaylı rezonans frekansı tarama süreci sonucunda tespit edilen nokta ise sistem rezonans frekans değeri olarak kabul görerek ultrasonik temizlik sistemine yeni çalışma frekansı olarak gönderilir ve sistem rezonans frekansında çalışma sürecine geçer.



Şekil 2: İki aşamalı tarama tekniği

### 3. VHDL Kodu Akış Diyagramı

Ultrasonik temizlik sistemlerinde değişebilir yük yapısı ve çeşitli fiziksel nedenlerle gözlenen değişken rezonans frekansı davranışının periyodik olarak tespit edilebilmesi amacıyla geliştirilen algoritma Şekil 3’de verilmiştir. Görüldüğü üzere, ikili tarama algoritması iki temel tarama sürecinden oluşur. Öncelikle olarak hızlı tarama modu çalışır. İlk olarak ilgili başlangıç değerleri belirlenir. Tarama sürecinde herhangi bir frekans değeri için birden fazla (burada 250 örnek) örnek değeri alınacağından algoritmanın başında ilgili parametre bir artırılıp gerekli örnek sayısına ulaşıp ulaşılmadığı kontrol edilir. Sonrasında bütün örnek değerleri toplanarak bir genel toplam değişkeninde saklanır. Daha sonra, o anki frekans değeri için elde edilen örnek toplam değeri o ana kadar hesaplanmış rezonans frekans değeri için elde edilen örnek toplamını tutan değişken ile karşılaştırılır. Eğer yeni gelen örnek toplamı daha büyük ise rezonans frekansını ve rezonans frekansını veren en büyük toplam değerini tutan değişkenler güncellenir. Yoksa eski değerleri korunur. Bu işlem taranacak frekans bandının en son frekans değerine gelene kadar devam eder. Ardından, hızlı tarama sürecinde belirlenen yaklaşık rezonans frekansı bir başka değişken ile ikincil tarama olan detaylı tarama sürecine iletilir. Elde edilen bu rezonans frekans değerini merkez kabul ederek, yeni bir dar frekans bandı belirlenir (Burada 2KHz’lik bir frekans bandı belirlenmiştir). Sonrasında, hızlı tarama sürecinde gerçekleştirilen işlemlerin tamamı detaylı tarama sürecinde de tekrar edilir. Detaylı tarama sürecinin hızlı tarama sürecine göre tek farkı, frekans bandı tarama adım aralığıdır.



Şekil 3: VHDL kodu akış diyagramı

Ultrasonik temizlik sistemlerinin çalışma anı rezonans frekansının belirlenmesi sürecinde geliştirilen VHDL kodu, sadece tarama sürecinden oluşmamaktadır. Bu amaç doğrultusunda tasarlanan ADC’li geri besleme devresine ait sayısal devre üç temel kısımdan oluşmaktadır. Kod geliştirilirken, her biri ayrı bir süreci gerçekleyen üç ayrı process bloğu tasarlanmıştır. Bu process bloklarından bir tanesi, bu makalenin de konusu olan iki aşamalı tarama yöntemini gerçeklemektedir. Diğer bir process ise, geri besleme devresinde kullanılan ADC donanımının kontrolünü sağlamaktadır. Son olarak, üçüncü process bloğumuz da kullanılan FPGA donanımının üzerinde bulunan lokal osilatör’ün bize sunduğu saat sinyalini referans olarak ADC’nin kontrolü ve frekans bandının taranması işlemleri için gerekli olan saat sinyallerinin üretilmesinden sorumludur.

### 4. Sonuçlar

Ultrasonik temizlik sistemlerinde rezonans frekansı tespiti için geliştirilen yinelemeli frekans bandı tarama tekniğinin

FPGA donanımı üzerinde gerçekleştirilmesi sonucu elde edilen tasarım sonucunun lojik kapasite kullanımı bilgisi tablo olarak Tablo1’de verilmiştir. Tablodan da görülebileceği üzere, Xilinx Spartan3 donanımı üzerinde gerçekleştirilen algoritma lojik kapasite kullanımı olarak verimli bir yapıdadır.

Sonuç olarak, bu çalışma ile ultrasonik temizlik sistemlerinde temizlik verimini düşüren kararsız yük yapısı ve fiziksel koşullar temelli olumsuz etkiler, değişken rezonans frekansının belirli zaman aralıklarıyla periyodik olarak belirlenmesi ile minimize edilmiştir. Diğer yandan geliştirilen iki aşamalı tarama yöntemi sayesinde tarama süreci kullanıcıdan gelen talebe göre daha hızlı ve dinamik bir yapıya kavuşturulmuştur.

Tablo 1: Lojik kapasite kullanımı

Device Utilization Summary (estimated values)			
Logic Utilization	Used	Available	Utilization
Number of Slices	335	1920	17%
Number of Slice Flip Flops	239	3840	6%
Number of 4-input LUTs	564	3840	14%
Number of bonded IOBs	41	173	23%
Number of GCLKs	3	8	37%

## 5. Teşekkür

Bu çalışma TÜBİTAK tarafından 7080355 ve 109E295 nolu projeler kapsamında desteklenmiştir.

## 6. Kaynakça

- [1] [http://en.wikipedia.org/wiki/Ultrasonic\\_cleaning](http://en.wikipedia.org/wiki/Ultrasonic_cleaning) (Visiting date 24.March.2010)
- [2] <http://www.bluewaveinc.com/reprint.htm> (Visiting date 24.March.2010)
- [3] <http://www.natclo.com/dp/ultra.html> (Visiting date 20.March.2010)
- [4] M. Yakut, A. Tangel, C. Tangel, “A microcontroller-based generator design for ultrasonic cleaning machines” *ISTANBUL UNIVERSITY-Journal of Electrical and Electronics Engineers*, Vol.9 (1), pp. 853-860, 2009.
- [5] J. Ishikawa, Y. Mizutani, T. Suzuki, H. Ikeda, H. Yoshida, “High-frequency drive-power and frequency control for ultrasonic transducer operating at 3 MHz”, *Industry Applications Conference, 32. IAS Annual Meeting, IAS '97*, Vol.2, pp. 900 – 905, 1997.
- [6] L. Svilainis, G. Motiejunas, “Power amplifier for ultrasonic transducer excitation” *ULTRAGARSAS*, Nr.1(58), pp.30-36, 2006
- [7] C. Buasri, A. Jangwanitlert, “Comparison of switching strategies for an ultrasonic cleaner”, 5th International Conference on *Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology, ECTI-CON 2008*, pp. 1005 – 1008, 2008.
- [8] E.Koutroulis, A. Dollas and K. Kalaitzakis, “High-frequency pulse width modulation implementation using FPGA and CPLD ICs”, *Journal of Systems Architecture*, Vol.52 (6), pp.332-344, June 2006.
- [9] A. M. Omar, N. A. Rahim, and S. Mekhilef, “ Three-Phase Synchronous PWM for Flyback Converter With Power-Factor Correction Using FPGA ASIC Design” *IEEE Transactions on Industrial Electronics*, Vol.51(1), Feb. 2004.



# Kablosuz Algılayıcı Ağları ile Yangın Tespit Sistemi

Çağdaş Döner<sup>1</sup>, Gökhan Şimşek<sup>2</sup>, K.Sinan Yıldırım<sup>3</sup>, Aylin Kantarcı<sup>4</sup>

<sup>1,2,3,4</sup>Bilgisayar Mühendisliği Bölümü  
Ege Üniversitesi, İzmir.

<sup>1</sup>donercagdas@gmail.com, <sup>2</sup>gokhansimsek87@gmail.com, <sup>3</sup>sinan.yildirim@ege.edu.tr,  
<sup>4</sup>aylin.kantarcı@ege.edu.tr.

## Özetçe

Bu çalışmada, kablosuz algılayıcı düğümleri ile çeşitli ortamlarda kullanılabilen ve erken evrede yangın tespiti yapabilen bir sistem gerçekleştirilmiştir. Gerçekleştirim sırasında Dağıtık Kapsama Ağacı Oluşturma ve Konum Belirleme protokolleri oluşturulmuştur. Sistem TOSSIM ortamında benzetim yolu ile denenip son haline getirildikten sonra gerçek algılayıcı düğümlerine taşınmıştır. Deney sonuçları sistemin başarı ile kullanılabileceğini göstermektedir.

## 1. Giriş

Kablosuz algılayıcı ağları basit işlemci, güç tüketimi düşük anten ve çeşitli algılayıcıları içeren algılayıcı düğümlerinden oluşmaktadır. Algılayıcı ağları kablolu iletişim altyapısı gerektirmediği için çevreye zarar vermeden kolay ve ucuz bir şekilde oluşturulabilir. Algılayıcı düğümleri sahip olduğu yazılım ve donanım altyapısı ile yerel veri depolama ve işleme yeteneklerine sahiptirler. Birbirleri ile iletişime geçebilmeleri, veri alışverişi yapabilmeleri yanı sıra karmaşık işlemlerin gerçekleştirilmesinde işbirliği yapabilmelerini sağlar. İletişim olanaklarının düşük güç tüketimine sahip olması algılayıcı düğümlerinin ömrünün uzun olmasına olanak verir. Düğümlerin ortama yerleştirildikten sonra programlanabilir olması da büyük avantaj sağlar [1].

Maliyetlerinin düşük olması ve kullanım esnekliklerinden dolayı kablosuz algılayıcılar birçok askeri, sivil ve çevresel uygulamada kullanılmaya elverişlidirler [3]. Bu çalışmada kablosuz algılayıcıların yangınların erken evrede saptanması amacıyla kullanımına yönelik bir sistem geliştirilmiştir. Yangınların erken evrede belirlenebilmesi, kısa sürede söndürülmelerini sağlayacak, can ve mal kaybını büyük oranda önleyecektir. Geliştirilen sistem, yerleşim yerleri, iş yerleri gibi kapalı alanlarda kullanılabileceği gibi açık alanlarda da kullanılabilir.

Kapalı alanlar için kullanılan geleneksel yangın tespit sistemleri, kablolu altyapı ve merkezi bir kontrol ünitesi içerirler [10]. Kablolu altyapı kurulum harcamalarına neden olur. Merkezi kontrol ünitesinin bozulması ve kabloların hasar görmesi sistemi iş görmez hale getirebilir.

Sistemin ekolojik dengenin bozulmasına neden olan orman yangınlarının erken tespitinde de kullanılabilmesi çalışmanın önemini arttırmaktadır. Orman yangınlarını belirleme işlemi gezici korucular, izleme kuleleri ve uydu sistemleri ile sağlanabilmektedir. Korucu ve izleme kuleleri dikkatsizlik, görevlilerin her zaman iş başında olamaması, tüm

alanın gerçek zamanlı olarak izlenememesi gibi risklere sahiptir. Uydu izleme sistemlerinin dezavantajları tarama periyodunun yüksek oluşu ve uydu görüntülerinin düşük çözünürlüğe sahip olmasından dolayı yangınları belli bir büyüklüğe eriştikten sonra tespit edilmesine olanak vermektedir. Uydu ve yeryüzü arasına bulutların girmesi de açık ortamlarda yangın tespitini güçleştirecektir. Tüm bu nedenlerden dolayı orman yangını tespiti için en elverişli ve ucuz çözüm kablosuz algılayıcı ağlarıdır [5, 6].

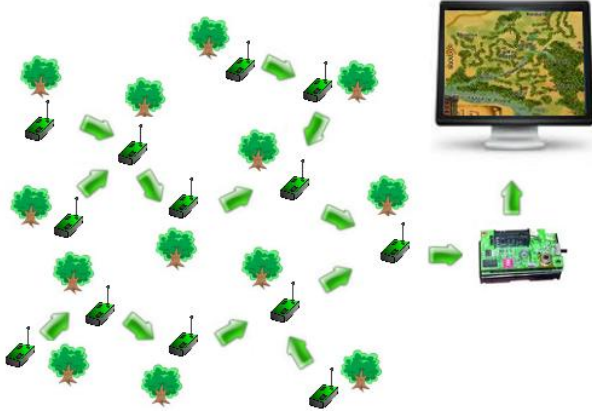
Gaz algılayıcılarına sahip düğümlerin kullanılmasıyla sistem, maden yangınlarının önceden tespit edilmesinde de yarar sağlayacaktır. Mevcut maden güvenlik sistemlerinde fiber optik, elektrik kablosu gibi kablolu iletişim mekanizmaları kullanılmaktadır. Maden ocaklarında kablolu iletişim altyapısını kurabilmek oldukça güçtür. Kablonun bir bölümünün hasar görmesi tüm sistemi kullanırsız hale getirebilir. Bu nedenle, kablosuz algılayıcı ağları maden yangınlarının da önceden belirlenmesi için en uygun çözümdür [4].

Bu çalışmada, erken evrede yangın tespiti yapabilen bir kablosuz algılayıcı ağı sistemi tasarlanmıştır. Bu tasarım Genetlab Sensenode v1.2 algılayıcı düğümleri, nesC programlama dili ve TinyOS açık kaynak kodlu işletim sistemi kullanılarak gerçekleştirilmiştir. Araştırmalarımıza göre, geliştirilen sistem, Türkiye'de açık kaynak kodlu, pratik olarak kodlanmış ve fiziksel olarak test edilmiş olan ve bir lisans tezi kapsamında gerçekleştirilmiş ilk yangın tespit sistemidir.

## 2. Gerçekleştirilen Sistem

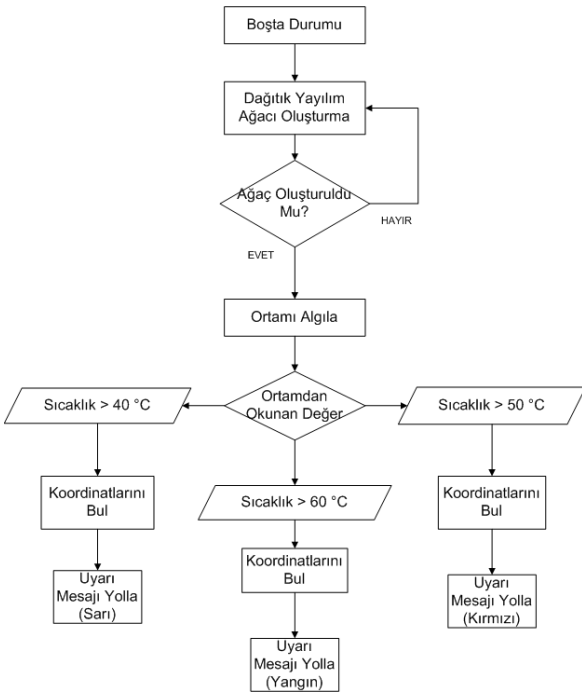
Gerçekleştirdiğimiz sistem, herhangi bir ortama rasgele yerleştirilmiş sıcaklık algılayıcıları barındırma düğümlerden oluşmaktadır (Şekil 1). Sistemin işleyiş adımları Şekil 2'de verilmektedir. Sistem ilk çalışmaya başladığı anda tüm düğümler işbirliği halinde veri iletiminde kullanılmak üzere *Yayılm Ağacı Protokolü* kullanarak sanal bir patika oluştururlar. Daha sonra, düğümler periyodik olarak ortamdaki sıcaklık değerini örneklemeye başlarlar. Ortamın sıcaklığında kayda değer bir yükselme olduğunda, algılayıcı düğümleri ölçümlerini içeren birer veri paketi oluşturur. Algılanan sıcaklık değerine göre 3 tür veri paketi kullanılmaktadır. Sıcaklık EŞİK1'den büyükse SARI, EŞİK2'den büyükse KIRMIZI uyarı paketleri gönderilir. Sıcaklığın EŞİK3'ü aşması durumunda yangın başladığına emin olunur ve YANGIN paketi gönderilir. Eşik değerleri kullanılan ortama özgü olarak belirlenebilir. Veri paketleri, oluşturduğumuz yayılım ağacı ile "ana" düğüm olarak adlandırılan, bilgisayara bağlı özel bir düğüme gönderilirler. Bilgisayarda, paketleri

alan bir uygulama alınan mesajları görselleştirerek kullanıcıya sunmaktadır.



Şekil 1: Algılayıcı Ağının Genel Yapısı

Alt bölümlerde çalışma kapsamında veri iletiminde kullanılmak üzere geliştirilen *Dağıtık Yayılım Ağacı Protokolü* ve yangın koordinatlarının belirlenmesinde kullanılan *Konum Bulma Protokolü* tanıtılmaktadır.



Şekil 2. Sistemin işleyişi

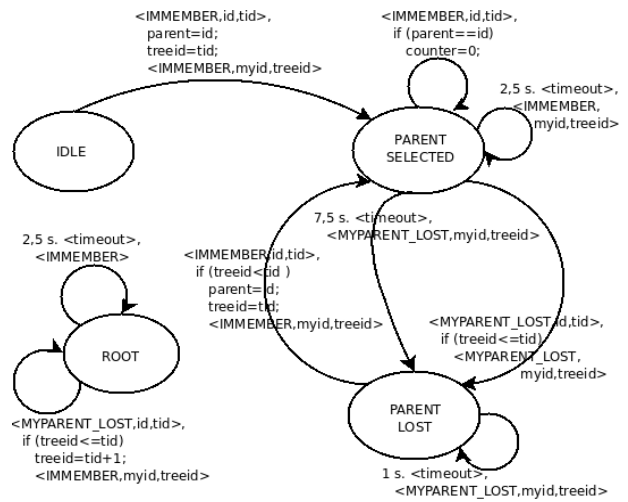
## 2.1. Dağıtık Yayılım Ağacı Oluşturma Protokolü

Dağıtık yayılım ağacı, birbirinden kaynak ve görev olarak ayrılmış fakat birbiriyle iletişimde bulunan düğümlerimizin, çoğu zaman veri taşıma amacıyla kullandıkları ve taşınacak veriyle ilgili çeşitli yönlendirme kurallarını barındıran sanal bir yoldur. Bu veri taşıma yolunun dağıtık bir şekilde oluşturulması için kullandığımız kural, her düğümün veriyi kendisinin belirlediği ve kapsama alanındaki düğümlerden seçtiği "baba" adı verilen özel bir düğüme iletmesidir.

Dağıtık yayılım ağacı oluşturma işlemi başlangıcında, her düğüm, uyanıp kendisinin hangi işi üstlendiğinin farkına varır.

Bu özellikler gerçekleştirimin içine gömülen *durum* ifadesiyle belirtilmektedir. Yayılım ağacı oluşturma işlemi başlatan özel düğüm ROOT durumunda diğerleri IDLE durumda gözlerini açar. ROOT düğümünden gelen bir mesajla yayılım ağacının oluşturulmasına başlanır. Protokol tasarımımızın Sonlu Durum Makinası (SKM) Şekil 3'te verilmiştir.

Kök düğüm açılır açılmaz kurulumu başlatmak için belirli periyotlarla *IMMEMBER* mesajını yayınlar. Bu mesajı alan *IDLE* düğüm ya da düğümler, durumlarını *PARENT\_SELECTED* olarak değiştirip *baba* düğüm olarak kök düğümünü seçerler. SKM'da da gösterildiği gibi, *PARENT\_SELECTED* durumuna geçen her düğüm, kapsamında olan düğümlerin de kendisini *baba* olarak seçilebilmesi *IMMEMBER* mesajını yayınlarlar. Tüm *IDLE* düğümler baba seçene kadar bu işlem devam eder. Görüldüğü gibi kök dışındaki her düğüm seçme ve seçilebilme hakkına sahiptir. En sonunda ağacımız yaptığı dallanmalarla dengeye ulaşır ve sanal patikalar üzerinden veri taşımaya hazır olur.



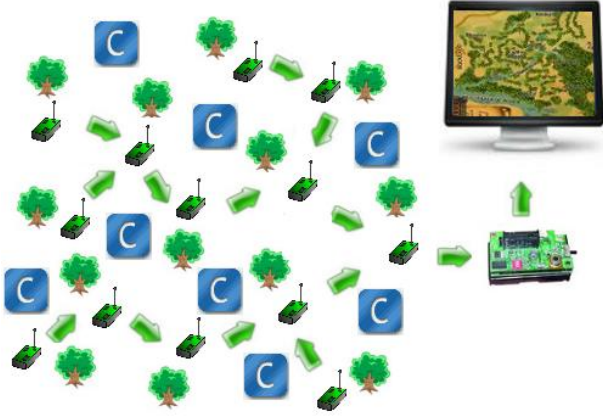
Şekil 3: Dağıtık yayılım ağacı protokolü sonlu durum makinası

Sistemimiz dinamik ve gerçek zamanlı -etkiye bir an önce tepki veren- olarak geliştirilmiştir. Zaman içinde çevresel ya da teknik nedenlerle yayılım ağacımız güvenilirliğini yitirebilir. Örneğin, sistemimiz bir ormanda kullanılacaksa, hayvanlar düğümlere zarar vererek düğümleri devre dışı bırakabilir. Teknik sorunlar ise düğümlerin pillerinin bitmesi veya algılayıcılarda anlık hataların oluşması gibi örneklendirilebilir. Bu durumlarda yayılım ağacının bir kısmı ulaşılamaz hale gelebilir. Bir düğüm, babasından belirli bir süre mesaj alamazsa protokol bir sorun olduğunu tespit eder ve aşağı doğru dağıtık yayılım ağacı kurma işlemini yeniden başlatır. Böylece aşağıya doğru yayılım tekrarlanmış ve ağaç eski sağlığına kavuşmuş olur.

## 2.2. Dağıtık Konum Bulma Protokolü

Kablosuz algılayıcı düğümleri bir ortama rasgele yerleştirildikleri için, dağıtık çalışacak bir konum belirleme protokolüne ihtiyaç vardır. Bunun için kullanılacak yöntemlerden biri GPS yardımı ile her sensörü elle yerleştirmek, ikincisi ise sensörlerin kendi koordinatlarını kendilerinin bulmasını sağlamaktır. Tasarladığımız sistemin pratik ve hatalara karşı daha dirençli olması için düğümler

kendi koordinatlarını bir konum belirleme protokolünü kullanarak hesaplamaktadırlar.



Şekil 4: Çapa düğümler, diğer düğümler ve kapsama ağacı

Düğümler, kendi konumlarını daha önceden bilen 'çapa' düğümlerin periyodik olarak yayınladıkları mesajları kullanarak konumlarını hesaplamaktadırlar. Şekil 4'te bu işleyiş gösterilmiştir. Sistem ilk çalışmaya başladığında tüm algılayıcılar kendi konumları hakkında bir bilgi sahibi değildir. Tehlike durumlarında bir algılayıcı düğümü kendine en yakın, diğer bir deyişle algılanan sinyal seviyesi en yüksek olan üç çapa düğümünün konumlarını kendi tablosuna yerleştirerek konum hesaplama için gerekli bilgileri elde eder. Bir düğüm bu tabloda yer alan çapa düğümlerin koordinatlarını kullanarak konumunu (1) deki gibi hesaplar.

$$(x_i, y_i) = ((x_1+x_2+x_3)/3, (y_1+y_2+y_3)/3) \quad (1)$$

Konum belirlemede kullanılan bu yöntemin en büyük avantajı, konumların sadece gerekli durumlarda hesaplanması ve düğümlerin çeşitli nedenlerle yer değiştirmesi sonucu ile oluşabilecek yanlış konum bildirimlerinin önüne geçilmiş olmasıdır.

### 3. Gerçekleştirim

Yangın tespit sisteminin geliştiriminde kullanılan donanım platformu Şekil 5'te gösterilen Genetlab Sensenode v.1.3 [8] 'dır. Bu platform 16-bit low-power MSP430 [11] mikrodenetleyicisini içermektedir. Mikrodenetleyici 10kB RAM, 48kB program flash ve 1024kB external flash belleğe sahiptir. Platformda bulunan Chipcon CC2420 [12] haberleşme yongası 2.4 GHz frekansta çalışmaktadır ve 250kbps veri iletim hızına sahiptir. SHT11 [13] sıcaklık algılayıcısı ise çevreden sıcaklık verisini almak için kullanılmıştır.



Şekil 5: Genetlab Sensenode v.1.3

Sistem açık kaynak kodlu bir işletim sistemi olan TinyOS ve nesC [9] programlama dili kullanılarak geliştirilmiştir. TinyOS kablosuz algılayıcı ağları için tasarlanmıştır ve

bileşen tabanlı bir mimariye sahiptir. TinyOS'un bileşen kütüphaneleri ağ protokolleri, dağıtık servisler, algılayıcı sürücüler ve veri toplama araçlarını içermektedir. Ancak tasarladığımız sistemde TinyOS'un hiçbir hazır protokolünü kullanmadık ve geliştirdiğimiz protokoller kendi özgün tasarımımızı barındırmaktadır.

#### 3.1. TinyOS Gerçekleştirimi

Sistem gerçekleştiriminde TinyOS işletim sisteminin sağladığı temel bileşenler kullanılmıştır. Bunlar veri yayını, sinyal seviyesini elde edebilmek ve ortamdan sıcaklık değerini almak için kullanılan bileşenlerdir. Veri yayını için ActiveMessageC, AMSenderC ve AMReceiverC bileşenleri kullanılmıştır. TempSensorC bileşeni sunduğu Read arayüzü ile SHT11 sıcaklık algılayıcısından sıcaklık verisi almayı sağlamaktadırlar.

Her algılayıcıda, sinyal seviyesini saklamak için 4x3'lik bir dizi tutulmaktadır. CC2420Packet bileşeni sinyal seviyesini elde edebilmek için temel bir arayüz sunmaktadır.

#### 3.2. Java Uygulaması

Tasarlanan sistem tüm öğeleriyle tamamlanıp bir ortamda çalışmak için kurulduğunda, sistemin tepkileri ve tehlikeli olarak belirtilmiş durumların bir bilgisayar ya da başka bir cihaz ile izlenmesi gerekmektedir. Bu cihazlar bilgisayar ya da bir PDA [15] olabilir. Platformdan bağımsız, kolay güncellenebilir bir ortam sunması ve zengin görselleştirme kütüphanelerine sahip olmasından dolayı uygulama geliştirmek için Java [14] dilini seçtik.

Yangın tespit sistemi çalışırken ortamda dolaşan verilerin bilgisayara aktarılma işlemi, ana düğüm adı verilen özel bir düğüm ile gerçekleştirilir. Bu özel düğüm, bilgisayara seri porttan bağlanmaktadır. Kendisine yönlendirilmiş olan mesajları algılayan ana düğüm, aldığı verileri seri porta ileterek uygulamaların kullanılmasını sağlar. Başka bir alternatif ana düğümüne bir SIM kartı yerleştirmek ve kablosuz iletişim şebekesi üzerinden bir merkezle iletişime geçmektir.

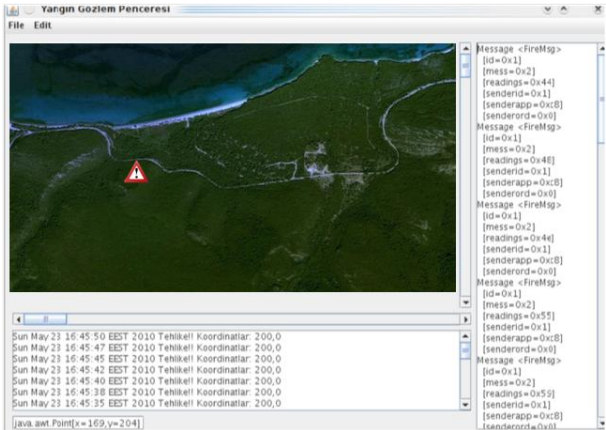
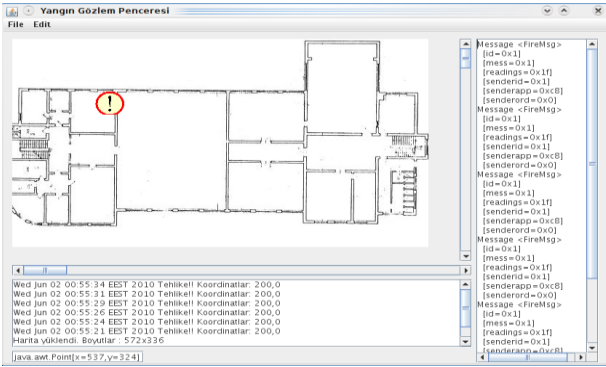
Sistemdeki mesajlardan *YELLOW* ve *RED* tipindeki mesajlar sıcaklığın yüksek olduğu durumlarda alınırken *INFO* tipindeki mesajlar periyodik olarak alınan ve uygulamada tüm düğümlerin yerinin gösterilmesi için kullanılan bir mesajdır. Düğümlerin tahmini koordinatları da mesajların taşıdığı verilerin içinde yer alır. Java uygulaması bu verileri kullanarak düğümlerin yerlerini ekranda belirtir. Bu işlem sırasında görselleştirme adına Java Swing [14] kütüphaneleri kullanılmıştır. Şekil 6'da kritik mesaj alınan bir bölge, basitçe ekranda gösterilmiştir. Görüldüğü gibi son kullanıcıya bir arayüz gerçekleştirilmiş ve sistemin denetim altında tutulması sağlanmıştır.

### 4. Deneyler

Yangın tespit sisteminin geliştirimi sürecinde, temel gerçekleştirim TOSSIM [2] benzetim aracı kullanılarak yapılan benzetimlerle test edilmiştir. TOSSIM açık kaynak kodlu ve fiziksel ortamı gerçeğe çok yakın bir biçimde modelleyen bir benzetim aracıdır. Algılayıcı düğümleri ile protokol ve uygulama geliştirimi dağıtık bir mantıkta yapıldığı için, sistemin hatalarının bulunması ve bunların ayıklanması oldukça zahmetlidir. Tüm sistemin bir bütün olarak gerçek

platformda test edilmesinden önce, her bir sistem parçası TOSSIM üzerinde hazırlanan test durumları ile sınanmıştır. TOSSIM aracı ile sistemin temel hataları ayıklanmıştır.

Sistem için tasarlanmış protokol modülleri, birbirinden bağımsız bir şekilde fiziksel düğümler üzerinde test ettik ve hatalardan ayıkladıktan sonra, tüm sistemi fiziksel ortamda test ettik. Test için, önceden konumlarını belirlediğimiz çapa düğümleri ve konumlarını bilmeyen düğümleri bölümümüzün özel bir katına rastgele yerleştirdik. Java uygulamasını çalıştırarak, herhangi bir düğümün sıcaklığının yükselmesini sağlayarak sistemdeki mesaj iletişimini takip ettik. Tüm bunlara ek olarak bazı düğümleri kapatarak ve bazılarının yerlerini değiştirerek genel testler yaptık. Deney sonuçları sistemin başarı ile çalıştığını göstermiştir.



Şekil 6. Java uygulamasından görüntüler

## 5. Sonuçlar ve Öneriler

Bu bildiriye, yangınların erken evrede belirlenmesini sağlayan bir kablosuz algılayıcı ağı sistemi tanıtılmıştır. Tasarlanan ve geliştirilen sistem, Türkiye'de bir lisans tezi kapsamında gerçekleştirilen açık kaynak kodlu ve gerçek ortamda denenmiş ilk sistemdir. Bundan sonraki hedeflerimizden biri, nem, basınç gibi farklı türdeki algılayıcıları içeren düğümleri de kullanarak sistemi zenginleştirmektir. Diğer hedeflerimiz, sisteme bir saat eşzamanlama protokolü ve bir noktanın k adet düğümce kapsanmasını sağlayacak bir k-kapsama protokolü eklemektir. Ayrıca kablosuz yazılım güncelleme ile oldukça yapılandırılabilir bir sistem gerçekleştirmeyi de hedeflemekteyiz.

## 6. Kaynaklar

1. K. Holger and W. Andreas: *Protocols and Architectures for Wireless Sensor Networks*, John Wiley & Sons Ltd., 2005, ISBN: 0-470-09510-5.
2. P. Levis, N. Lee, M. Welsh, D. Culler: *TOSSIM: Accurate and Scalable simulation of Entire TinyOS Applications*. Proceedings of the 1st international Conference on Embedded Networked Sensor Systems, SenSys '03, pp. 126-137, 2003.
3. Y. Li, M T. Thai, W. Wu: *Wireless Sensor Networks and Applications*, Springer, 2008, ISBN 978-0-387-49591-0.
4. W. Tan, Q. Wang, H Huang, Y Guo, G Zhan.: *Mine Fire Detection System Based on Wireless sensor Networks*, Proceedings of Conference on Information Acquisition, ICIA'07, 2007.
5. L. Yu, N. Wang, X. Meng: *Real-Time Forest Fire Detection with Wireless Sensor Networks*, Proceedings of Wireless Communications, Networking and Mobile Computing, pp.1214-1217, 2005.
6. J. Zhang, W. Li, N. Han, J. Kan: *Forest Fire Detection System Based on a Zigbee Wireless Sensor Network*, Front. For. China, Springer Verlag, pp.369-374, 2008.
7. [http://en.wikipedia.org/wiki/Spanning\\_tree](http://en.wikipedia.org/wiki/Spanning_tree)
8. <http://www.genetlab.com/>
9. <http://www.tinyos.net>
10. [http://en.wikipedia.org/wiki/Fire\\_alarm\\_system](http://en.wikipedia.org/wiki/Fire_alarm_system)
11. "MSP430 Ultra-Low-Power Microcontroller", Texas Instruments, <http://www-s.ti.com/sc/techlit/slab034.pdf> 2010
12. "2.4 GHz IEEE 802.15.4 / ZigBee-ready RF Transceiver", Texas Instruments, <http://focus.ti.com/lit/ds/symlink/cc2420.pdf> 2010
13. SHT11-Digital Humidity Sensor (RH&T), Sensirion, [http://www.sensirion.com/en/pdf/product\\_information/Datasheet-humidity-sensor-SHT1x.pdf](http://www.sensirion.com/en/pdf/product_information/Datasheet-humidity-sensor-SHT1x.pdf) 2010
14. Java Swing Toolkit, <http://java.sun.com>
15. Personal Digital Assistant (PDA) [http://en.wikipedia.org/wiki/Personal\\_digital\\_assistant](http://en.wikipedia.org/wiki/Personal_digital_assistant)

# Gömülü Çoklu İşlemcili Sistemlerde Yeniden Betimlenebilir Haberleşme Protokolleri

Salih Bayar, Arda Yurdakul

Boğaziçi Üniversitesi, Bilgisayar Mühendisliği Bölümü  
P.K. 2 TR-34342 Bebek, İstanbul, TÜRKİYE  
{salih.bayar, yurdakul}@boun.edu.tr

## Özetçe

Gömülü çoklu işlemcili sistemlerde işlemciler arası haberleşme tıkanıklığını engellemek için son zamanlarda Yonga-üstü-Ağ(YüA) sıklıkla kullanılmaya başlanmıştır. Oysaki YüA'larda yönlendirme tablololu anahtarlar için gerekli olan ilave alan ihtiyacı, paketleme, yönlendirme ve anahtarlama için gerekli olan ilave gecikme zamanları YüA'ların haberleşme tıkanıklığını engellemek için en elverişli seçenek olmadığını göstermektedir. Bunun yerine, fazladan bir gecikmeye yol açmayan, bire bir doğrudan bağlantı bu tarz sistemlerde daima en iyi çözüm olmuştur. Örneğin [1] ve [2]'de kısmi yeniden betimlenebilir uçtan uca (U2U) haberleşme protokolleri ortaya atılmıştır. Bu çalışmada, [1] ve [2]'de anlatılan incelemeler bir adım daha ileriye götürülerek, kısmi yeniden betimleme işleminin devrenin çalışma esnasında yani dinamik bir şekilde, isteğe bağlı olarak, çoklu kez yapılması sağlanmıştır. Böylece, çalışma esnasında, devrenin geride kalan kısmına zarar vermeden, U2U haberleşme protokol yapılandırılmaları isteğe bağlı olarak değiştirilerek, YüA'a rakip bir yapı ortaya çıkarılmıştır. Bu yapı, APKD (Field Programmable Array Logic-Alan Programlanabilir Kapı Dizisi) tabanlı, otomotiv uygulamalarında kullanılan Çoklu Araç Takibi (ÇAT, Multi Target Tracking (MTT)) için optimize edilmiş gömülü çoklu işlemcili Yonga-üstü-Sistem (YüS) mimarisinde test edilmiştir [3].

## 1. Giriş

Gömülü çoklu işlemcili sistemler, tek işlemcili sistemlere göre, uygulamanın yapısına da bağlı olarak daha yüksek performans sağlayabilmektedir. Teorik olarak, YüS'deki paralel çalışan işlemci sayısı arttıkça performansında aynı oranda artması beklenmektedir. Ancak, işlemci sayısı arttıkça, var olan işlemciler arasındaki haberleşme protokolün yapısının da karmaşıklaşacağından dolayı, bu tarz sistemlerin tasarımı zorlaşmaktadır. İşte bu noktada, YüA yapısı, bu problemleri aşmak için literatür de önerilen çözümlerden [4]. Yakın ve diğer çözümler de esnek ve uyarlanabilir haberleşme yapılarıdır [5][6][7]. Lakin YüA'larda yönlendirme tablololu anahtarlar için gerekli olan ilave alan ihtiyacı, paketleme, yönlendirme ve anahtarlama için gerekli olan ilave gecikme zamanları YüA'ların haberleşme tıkanıklığını engellemek için en elverişli seçenek olmadığını göstermektedir. Bunun yerine, fazladan bir gecikmeye yol açmayan, bire bir doğrudan bağlantı bu tarz sistemlerde

daima en iyi çözüm olmuştur. Zaman içerisinde değişen farklı yapılandırmaların oluşturulması da APKD'nin, devrenin çalışma esnasında kısmi yeniden betimleme özelliği ile U2U haberleşme protokol yapılandırmalarının isteğe bağlı olarak değiştirilmesi ile sağlanmıştır.

İşlenen bilgi miktarı yoğun olan, gömülü çoklu işlemcili sistemlerde, aktörler arası haberleşme trafiği de oldukça fazladır. Haberleşme trafiğinin büyüklüğü de, uygulamanın yapısına ve aktör sayısına bağlı olarak değişiklik göstermektedir. Yine uygulamaya bağlı olarak, aktörler arası haberleşmenin önceden bilindiği durumlarda, bir sonraki yapılandırma, aktörler birbirleri ile haberleşme işlemine ara verip bilgi işlerken yapılabilir. Burada dikkat edilmesi gereken husus, dinamik kısmi yeniden betimleme (DKYB) işlemi ile yapılan yapılandırma değiştirme işlemi için gerekli sürenin, aktörlerin o anlık bilgi işleme sürelerinden daha kısa olması gerektiğidir. Anlatılan bu durum, aktörler arası haberleşme için mümkün olabilecek en iyi durumdur ve olası diğer bütün uygulamalardan (YüA, paylaşılmış veri yolu vb.) çok daha hızlıdır. Çünkü böyle bir durumda 2 ucun birbiri ile haberleşmesinde arada hiç bir fazladan devre olmadığından, fazladan herhangi bir gecikme olmayacaktır, var olan tek gecikme kablo gecikmesi olacaktır. Bu tarz bir gecikme, bütün değişken haberleşme yapıları için en ideal ve ulaşılabilecek mümkün olmayan bir durumdur. Bu yüzden adı geçen yöntemden en etkili şekilde faydalanılabilecek uygulamalar, bilgi işleminin yoğun olduğu ve DKYB süresinin aktörlerin o anlık bilgi işleme sürelerinden daha kısa olduğu uygulamalardır.

APKD'lerin kısmi yeniden betimlenmesi, APKD üzerinde devre çalışır durumda iken APKD'nin bir kısmının, diğer kısımların çalışmasını kısıtlamadan yapılandırmanın değiştirilebilme yeteneğidir [8]. Yani, genel olarak sistemde bir performans kaybı olmamıştır ya da tasarımın değişmeyen kısımların işlevselliğinde bir eksiklik olmamaktadır. Kısmi yeniden betimlenmenin avantajı kullanılarak, donanım kaynakları farklı uygulamalar arasında paylaşılabilir, böylece kaynak faydalanma oranı artırılarak devrenin toplam güç sarfiyatı düşürülebilmektedir [9]. Burada dikkat edilmesi gereken yeniden betimlenme için gereken gücün de göz ardı edilmemesi gerçeğidir. Kısmi yeniden betimlenmenin başka bir avantajı ise bu tarz sistemlerin herhangi bir zamanda güncellenebilmesi ve böylece devamlı bir şekilde donanım desteği sağlanabilmektedir [9].

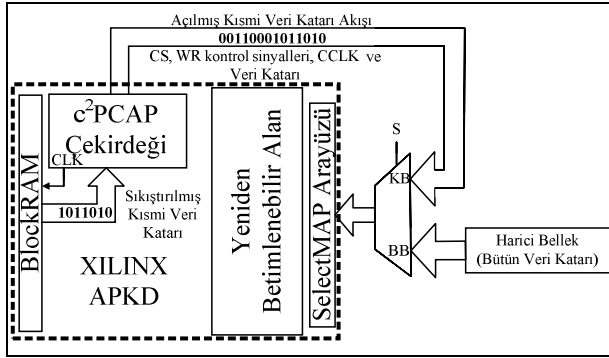
Bu çalışmada, dinamik kısmi yeniden betimlenmenin akışının kontrolü, daha önceden geliştirdiğimiz, cPCAP (compressed

Parallel Configuration Access Port) çekirdeğinin[10] geliştirilmiş hali olan c<sup>2</sup>PCAP (double compressed Parallel Configuration Access Port) çekirdeği[11] sayesinde sağlanmaktadır. c<sup>2</sup>PCAP çekirdeği önceden yaratılmış ve belirlenmiş olan sıkıştırılmış kısmi bit katarını çok hızlı Yonga-Üstü-Bellek (YüB) olan BlockRAM'de tutmaktadır. Böylece harici bir belleğe erişim ihtiyacı duyulmadan, c<sup>2</sup>PCAP çekirdeği her DKYB için YüB'de saklanan sıkıştırılmış kısmi bit katarını, zaman kaybına uğratmadan açarak, o tasarımda kullanılan APKD yapılandırma ara yüzü (SelectMAP ya da ICAP) aracılığıyla dinamik bir şekilde yapılandırma belleğine yazılmaya yazılmaktadır.

## 2. c<sup>2</sup>PCAP Mimarisi

Adı geçen c<sup>2</sup>PCAP çekirdeği, PCAP çekirdeğinin[12] geliştirilmiş bir şekli olup, önceden yaratılmış ve belirlenmiş olan, BlockRAM'de saklanan sıkıştırılmış kısmi bit katarlarının, isteğe bağlı olarak, zamanla kullanılan APKD yapılandırma ara yüzü aracılığıyla dinamik bir şekilde yapılandırma belleğine yazılmasını sağlayarak, DKYB için gerekli olan kontrol sinyallerini üretmektedir.

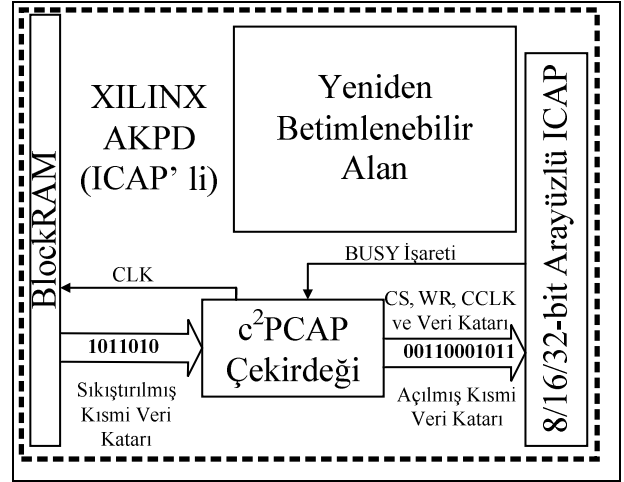
Şekil 1: Harici SelectMAP ara yüzü ile sistem donanımının mimarisi(KB: Kısmi Betimleme, BB: Bütün Betimleme)



XILINX APKD'leri c<sup>2</sup>PCAP çekirdeğinin kontrolü altında paralel olan harici SelectMAP APKD yapılandırma ara yüzü aracılığıyla, kendi kendini yeniden betimleyebilir. Burada, SelectMAP ara yüzü kullanılarak, kısmi veri katarı bu ara yüz tarafından kabul edilir ve kısmi yeniden betimleme işlemi aynı hedef olan APKD ile yapılır. Şekil 1'de görüldüğü gibi tek bir çip olan, kısmi olarak yeniden betimlenen APKD sadece edilgen değil aynı zamanda etkindir. SelectMAP ara yüzü sadece kısmi yeniden betimleme işlemi için ayrılmamıştır. Kısmi Betimlenmenin (KB) olmadığı zamanlarda, bu ara yüz aynı zamanda harici bellek ile APKD arasında Bütün Betimlenme(BB) amacı ile de kullanılabilir.

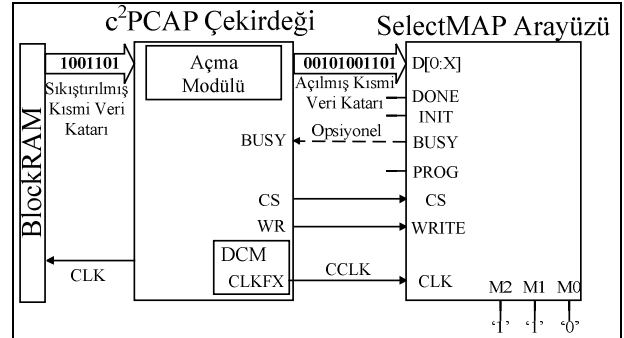
Benzer bir yapı İçsel Konfigürasyon Erişim Ara Yüzü'ne (İKEA, Internal Configuration Access Port(ICAP)) sahip olan (örneğin: Virtex-II, Spartan3A, Spartan-6 vb. ) XILINX APKD'leri için kullanılmaktadır. Şekil 2'den görüleceği üzere, ICAP, SelectMAP benzeri olup, tamamen aynı görevi gören ancak yonga içinde bulunan bir APKD yapılandırma ara yüzüdür.

Şekil 2: Dâhili Yapılandırma Ara yüzü Üzerinden Sistem Donanım Yapısı



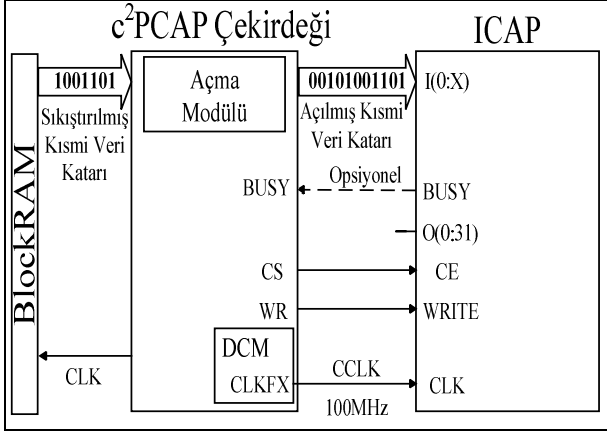
Burada dikkat edilmesi gereken ICAP'ın ara yüz genişliği (8/16/32-bit) yongadan yongaya farklılık gösterebilmektedir. Örneğin Virtex-II için 8-bit[13][14], Spartan-6 için 16-bit [15], Virtex-4 için 8/32-bit[16], Virtex-5 ve Virtex-6 için 8/16/32-bit[17][18] şeklindedir. Şekil 2'de görüldüğü gibi, yapılandırma ara yüzü için fazladan ne bir harici tele ne de harici bir elektriksel bağlantıya ihtiyaç vardır.

Şekil 3: c<sup>2</sup>PCAP Çekirdeği ve SelectMAP ara yüzü (X:7/15/31)



Şekil 3 ve Şekil 4'den anlaşılacağı üzere Yapılandırma Saat Frekansı'nı (YSF, Configuration Clock (CCLK)) üretmek için APKD'de mevcut bulunan Sayısal Saat Yönetici (SSY, Digital Clock Manager (DCM)) bloğu kullanılmaktadır. c<sup>2</sup>PCAP çekirdeği her bir saat döngüsünde yapılandırma ara yüzünün genişliğine bağlı olarak 1/2/4 Baytlık kısmi yeniden betimleme veri katarını BlockRAM'den okumaktadır. CS, WRITE ve CCLK sinyallerinin kontrolü altında, okunan bu veri kullanılan yapılandırma ara yüz tipine bağlı olarak SelectMAP ya da ICAP ara yüzüne gönderilir.

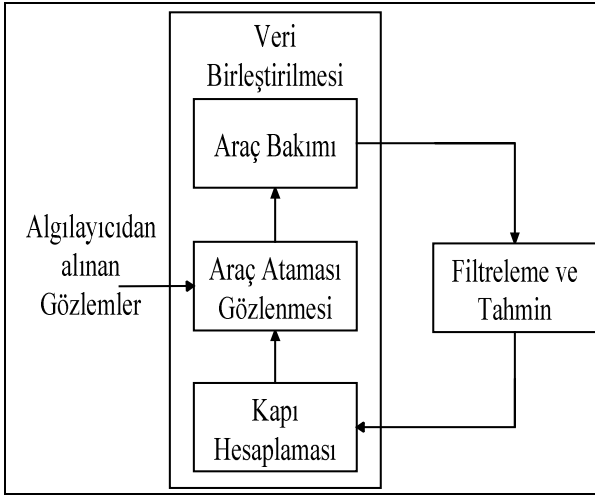
Şekil 4: c<sup>2</sup>PCAP Çekirdeği ve ICAP ara yüzü (X:7/15/31)



### 3. Örnek Olay İncelemesi: ÇAT

[3]'te APKD (Field Programmable Array Logic-Alan Programlanabilir Kapı Dizisi) tabanlı, otomotiv uygulamalarında kullanılan Çoklu Araç Takibi (ÇAT) için optimize edilmiş gömülü çoklu işlemcili Yonga-üstü-Sistem(YüS) mimarisi tasarlanmıştır. Şekil 5'te ÇAT sisteminin geliştirilmiş bir görünümü gösterilmektedir. Bu sistem Veri Birleştirilmesi ve Filtreleme & Tahmin olmak üzere 2 ana bloğa ayrılmıştır. Bu iki blok kapalı bir döngü içinde çalışmaktadır. Veri Birleştirilmesi bloğu kendi içinde 3 alt bloğa ayrılmıştır: Araç Bakımı, Araç Ataması Gözlenmesi ve Kapı Hesaplaması.

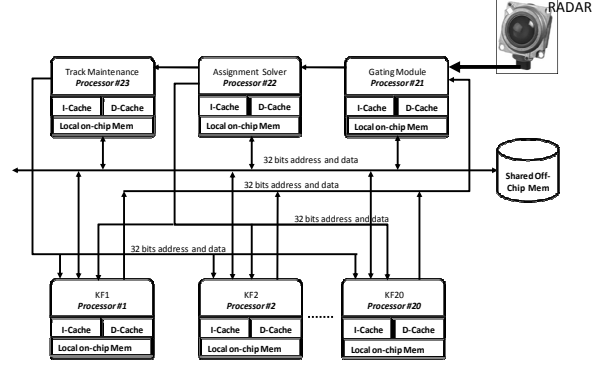
Şekil 5: ÇAT sisteminin basitleştirilmiş bir görünümü



Şekil 6'dan görüldüğü üzere bu yapıda toplam 23 adet Nios-II gömülü işlemci kullanılmış olup, işlemciler arası haberleşme değişmez zaman paylaşımı şeklindedir. Bu sistemde Filtreleme & Tahmin bloğu 20 adet Nios-II/s (Processor#1- Processor#20), Kapı Hesaplaması bloğu 1 adet Nios-II/f (Processor#21), Araç Ataması Gözlenmesi bloğu 1 adet Nios-II/s (Processor#22) ve Araç Bakımı bloğu 1 adet Nios-II/e (Processor#23) üzerinde gerçekleştirilmiştir.

Şekil 6'da ÇAT için optimize edilmiş gömülü çoklu işlemcili YüS mimarisindeki ilk kullanılan haberleşme yapısı da gösterilmektedir. Gömülü işlemciler arasındaki haberleşmeyi sağlayan bu yapı 3 adet paylaşımlı veri yolundan oluşmaktadır. Adı geçen bu örnek olay çalışması ALTERA APKD'leri üzerinde gerçekleştirilmiştir. Bizim çalışmamızda ise biz bu yapıyı XILINX APKD'leri üzerine taşıdık.

Şekil 6: İlk Gerçekleştirme: 3 paylaşımlı veri yolu



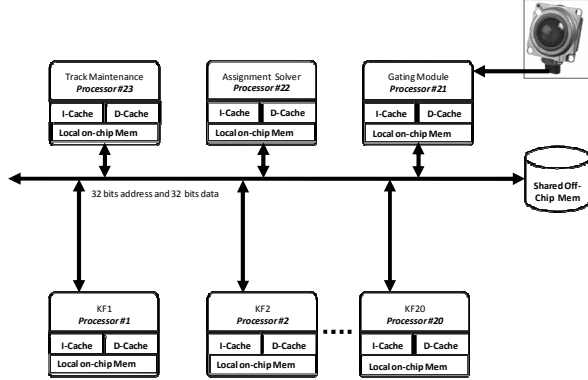
### 4. ÇAT'ın DKYB ile U2U Haberleşme Yapısının Gerçeklenmesi

Şekil 7-Şekil 11'de detaylı bir şekilde anlatıldığı üzere, sıralı, arka arkaya ve zamanla tekrarlı bir şekilde çalışan 5 farklı senaryo bulunmaktadır. Burada sadece 1 adet senaryo seçilmeyip, her bir senaryo zamanla, tekrarlı bir şekilde kısmi kendi kendine yeniden betimlenme ile gerçekleştirilmektedir. Uygulamanın yapısı gereği, işlemciler arası haberleşme yapısı zamanla Şekil 7'deki senaryodan başlayıp sırasıyla Şekil 11'deki senaryodaki gibi değişmektedir. Bizim burada gerçekleştirdiğimiz, işlemciler arası haberleşme yapısını senaryolarda belirtildiği şekilde sırası ile çalışma zamanında kısmi kendi kendine yeniden betimlenme ile değiştirmemizdir. Senaryolar arasındaki geçiş bütün betimlenme ile değil, çalışma zamanında kısmi kendi kendine yeniden betimlenme yöntemi ile gerçekleşmektedir. Çünkü var olan sistemde, çalışma zamanında hemen hemen bütün işlemciler etkin bir şekilde çalışmaktadır. Eğer kısmi yerine bütün yeniden betimlenme tercih edilseydi, sistemde performans düşüklüğü gerçekleşirdi. Bilindiği üzere yeniden betimlenebilir bir mimaride sabit, yani yeniden betimlenmeyen kısım ve bir de yeniden betimlenebilir kısım olmak üzere iki farklı kısım bulunmaktadır. Bizim oluşturduğumuz bu yapıda, 23 adet Nios-II gömülü işlemciler ve kontrol amaçlı kullanılan c<sup>2</sup>PCAP çekirdeği mimarinin sabit kısmında bulunurken, Nios-II gömülü işlemciler arasındaki bulunan farklı haberleşme yapıları yeniden betimlenebilir alanda durmaktadır. Çalışmanın detayları için ilgili makale[3] incelenebilir.

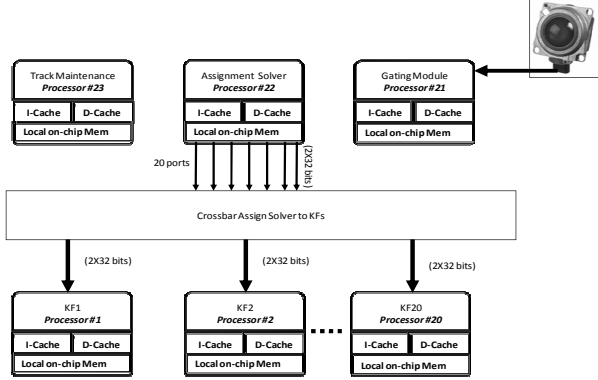
Bu çalışmada, ardışık 2 radar tarama zaman aralığı olan Darbe Yenileme Zamanı(DYZ, Pulse Repetition Time (PRT)) 25 milisaniyedir. Bu yüzden, işlemciler arası 5 farklı haberleşme yapısının değişimi için kullanılan toplam kısmi yeniden betimlenme süresi DYZ'den küçük olmalıdır. Buna ek olarak, 1 haberleşme yapısının değişimi için gerekli olacak

en uzun yeniden betimlenme süresi, herhangi bir işlemci için geçerli olan en kısa işlem zamanından daha kısa olmalıdır.

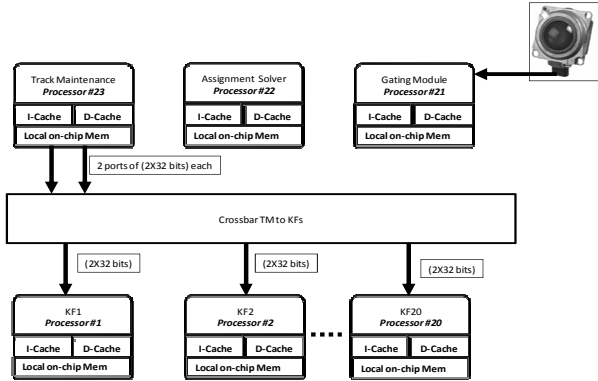
Şekil 7: [K1]: Veri ve kod erişimi için paylaşılmış veri yolu



Şekil 8:[K2]: Karşılıklı, birebir: Görev Çözücü'den KF Haberleşmesine



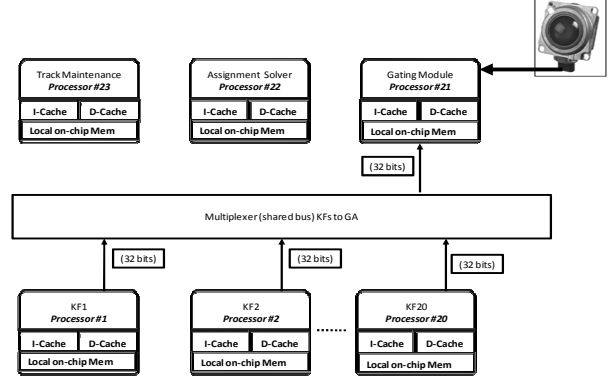
Şekil 9:[K3]: Karşılıklı, birebir: Araç Bakımı'ndan KF Haberleşmesine



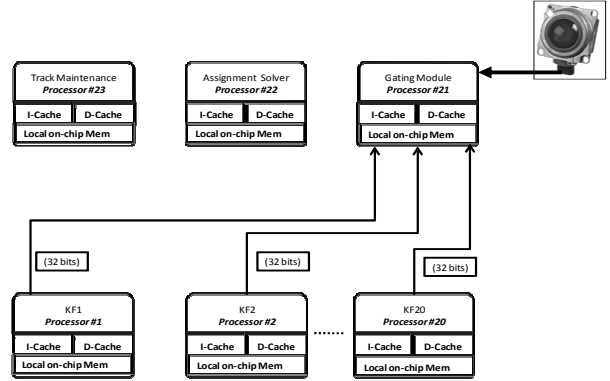
Hesap işleme zamanı, kısmi yeniden betimlenme süresinden daha uzun olduğundan dolayı ( $T_{comp} \geq T_{reconf}$ ), hesap işleme zamanı ile kısmi yeniden betimlenme süresinin tamamen örtüştüğünü söylemek emniyetli olup, yeniden betimlenebilir

U2U haberleşme mimarisinin yapısının en elverişli şekilde kullanıldığı söylenebilir.

Şekil 10: [K4]: Çoğullayıcı: KF'den Sinyal Ayırıcıya



Şekil 11: [K5]: KF'den Sinyal Ayırıcı'ya Çoklu Alçak Frekans Veri Yolları



## 5. Deneysel Sonuçlar ve Değerlendirilmesi

Tablo 1: ÇAT Uygulamasındaki 5 farklı haberleşme senaryosu için kullanılan yeniden betimlenme süreleri

Kısmi Veri Katarı	Yeniden Betimlenme Süresi [ms]			
	ICAP 8-Bit		ICAP 32-Bit	
	50MHz	100MHz	75MHz	100MHz
K1	5.36	2.68	0.894	0.670
K2	5.36	2.68	0.894	0.670
K3	5.77	2.88	0.962	0.721
K4	5.47	2.73	0.911	0.683
K5	5.21	2.60	0.868	0.651
Total	27.17	13.57	4.529	3.395

Bu çalışmada, biz, Şekil 7-Şekil 11'de gösterilen çalışmanın haberleşme yapısını XILINX firmasının Virtex-4LX100 APKD'sinde gerçekleştirdik. Her bir farklı senaryo için yeniden betimlenme süreleri Tablo 1'de özetlenmiştir.



Tablo 1’den görüldüğü üzere en uzun süre örneğin K2’den K3’e geçişte 2,88 ms olup (100MHz, ICAP 8-Bit), en hızlı hesap işleme zamanından daha kısa süreli olması gerekir.[3] ‘te Tablo 3’ten anlaşılacağı üzere, en kısa süreli hesap işleme zamanı 8 ms olup Araç Bakım Bloğuna aittir. Bu vesileyle bakıldığında, Tablo 1’ de detaylı bir şekilde gösterildiği gibi bu koşulu (<8ms) bizim uyguladığımız herhangi bir frekans değeri ve herhangi bir ara yüz sağlamaktadır.

ICAP 8-bit, 50MHz seçeneğinde herhangi bir yeniden betimlenme süresi en kısa süreli hesap işlemeyen daha kısa olmasına rağmen, toplam yeniden betimlenme süresi(27.17 ms) bu seçenek için DYZ’ den (25 ms) büyük olduğu için, bu seçeneğin (ICAP 8-bit 50 MHz) uygulanması imkânsız gözükmektedir. Oysaki ICAP 8-bit, 100MHz seçeneğinde hem toplam yeniden betimlenme süresi DYZ’ den daha küçük, hem de her bir kısmı yeniden betimlenme süresi en kısa hesap işleme süresinden daha az vakit almaktadır. Bunlara ek olarak, ICAP 32-bit modu takriben 15MHz den yüksek olmak koşuluyla herhangi bir frekans değeri için uygulanabilir durumdadır.

Bu deneyde her bir kısmı veri katarı takriben 260Kbyte civarındadır. Veri katarlarını sıkıştırarak, takriben %70 alan tasarrufu ile her bir veri katarı 79Kbyte’a inmiştir. 79Kbyte’lık bir veri takriben 40 adet yonga üstü BlockRAM bloğunu gerektirmektedir. Sonuç olarak, toplam kısmi veri paketlerinin (K1, K2, ...) miktarı 400Kbyte olup, 200/240 adet BlockRAM bloğu gerektirmektedir.

Uygulamanın ihtiyaçlarına göre (örneğin yüksek hız ya da kısıtlı miktarda YüB kullanım izni), bütün kısmi veri katarlarının saklanması YüB’ de olması yerine sadece 1 tanesinin YüB’ de, diğerlerinin harici bellekte saklanması ve YüB’ nin önbellek şeklinde kullanılmasıyla, YüB kullanımı azaltılabilir. Eğer önbellek yaklaşımı kullanılırsa takriben 40/240 BlockRAM bloğu (Toplamın %17 si) kullanılmış olur.

## 6. Sonuç

Bu çalışmada c<sup>2</sup>PCAP çekirdeği kullanılarak ICAP’ e sahip olan ya da olmayan bütün XILINX APKD’ lerin kendi kendine kısmi yeniden betimleme işlemini nasıl yaptığını, kısmi veri katarlarının nasıl YüB’ de tutulduğunu tartıştık. Bu yapı, APKD tabanlı, otomotiv uygulamalarında ÇAT için optimize edilmiş gömülü çoklu işlemcili Yonga-üstü-Sistem(YüS) mimarisinde test edilmiştir.

## 7. Kaynakça

1. *Partially Reconfigurable Point-to-Point Interconnects in Virtex-II Pro FPGAs*. **Jae Young Hur, Stephan Wong ve Stamatis Vassiliadis**. Mangaratiba, Brazil : s.n., March 27-29, 2007. Reconfigurable Computing: Architectures, Tools and Applications, Third International Workshop, ARC 2007.
2. *Partially Reconfigurable Point-to-Point FPGA Interconnects*. **Jae Young Hur, Stephan Wong ve Stamatis Vassiliadis**. 7, s.l. : International Journal of Electronics, Temmuz 2008, Cilt 95, s. 725 - 742.

3. *Trade-Off Exploration for Target Tracking Application in a Customized Multiprocessor Architecture*. **Jehangir Khan, Smail Niar, Mazen A. R. Saghir, Yassin El-Hillali ve Atika Rivenq-Menhaj**. 2009, EURASIP Journal on Embedded Systems.
4. *Networks on Chips: A New SoC Paradigm*. **L. Benini ve G. De Micheli**. Ocak 2002, IEEE Computers, s. 70-78.
5. *Dynamic Interconnection of Reconfigurable Modules on Reconfigurable Devices*. **Christophe Bobda ve Ali Ahmadinia**. 5, s.l. : IEEE Design & Test, September 2005, Cilt 22.
6. *Topology adaptive network-on-chip design and implementation*. **T. A. Bartic, J.-Y. Mignolet, V. Nollet, et al.** 4, 2005, IEE Proceedings: Computers and Digital Techniques, Cilt 152, s. 467-472.
7. *A study on communication issues for systems-on-chip*. **C. A. Zeferino, M. E. Kreutz, L. Carro ve A. A. Susin**. 2002, Symposium on Integrated Circuits and Systems Design Proceedings IEEE.
8. *Xilinx Development System Reference Guide 12.1i*. **Xilinx**. 2010.
9. *Benefits of partial reconfiguration*. **Cindy Kao**. s.l. : Xilinx, 2005, Xcell Journal, s. 65-67.
10. *Self-Reconfiguration on Spartan-III FPGAs with Compressed Partial Bitstreams via a Parallel Configuration Access Port (cPCAP) Core 2*. **Salih Bayar ve Arda Yurdakul**. s.l. : PRIME 2008 - 4th Conference on Ph.D. Research in Microelectronics and Electronics.
11. *An Efficient Self-Reconfiguration Core for Runtime Reconfigurable FPGA Interconnects*. **Salih Bayar ve Arda Yurdakul**. s.l. : IEEE Transactions on Very Large Scale Integration Systems, TVLSI (under revision), 2010.
12. *Dynamic Partial Self-Reconfiguration on Spartan-III FPGAs via a Parallel Configuration Access Port (PCAP)*. **Salih Bayar ve Arda Yurdakul**. Goteborg, Sweden : Proceedings of HiPEAC Workshop on Reconfigurable Computing, January 27, 2008.
13. *Virtex-II Pro and Virtex-II Pro X FPGA User Guide*. **Xilinx**. March 28 2007, UG012, (v4.1) edition.
14. *Virtex-II Platform FPGA User Guide*. **Xilinx**. March 28 2007, UG002 (v2.1) edition.
15. *Spartan-6 FPGA Configuration*. **Xilinx**. : s.n., February 22 2010, UG380, (v2.1) edition.
16. *Virtex-4 Configuration Guide*. **Xilinx**. June 9 2009, UG071, (v1.11) edition.
17. *Virtex-5 FPGA Configuration User Guide*. **Xilinx**. August 14 2009, UG191, (v3.8) edition.
18. *Virtex-6 FPGA Configuration*. **Xilinx**. January 18 2010, UG360, (v3.0) edition.



# Küçük Ölçekli Gömülü Sistemler İçin Bir Yazılım Geliştirme Sistemi

*İsmail Savaşkan<sup>1</sup>, Feza Buzluca<sup>2</sup>*

<sup>1</sup>{isavaskan}@neta.com.tr

<sup>2</sup>Bilgisayar Mühendisliği Bölümü  
İstanbul Teknik Üniversitesi, İstanbul.  
{buzluca}@itu.edu.tr

## Özetçe

Modern yazılım mühendisliği tekniklerinin “küçük ölçekli gömülü sistemler” (KÖGS) üzerinde uygulanması sık rastlanan bir çalışma şekli değildir. Bu durum, fiziksel kaynak kısıtlılığı, donanımlar arası bir standardın olmaması ve yazılım mühendisliği yaklaşımlarının KÖGS’e yönelik oluşturulmamasından ileri gelmektedir. Bu çalışmada ise yeni bir geliştirme süreci tasarlanarak bilinen yazılım mühendisliği teknikleri KÖGS’e uyarlanmıştır. Oluşturulan yazılım geliştirme sistemi ile özel kavramlar cinsinden modellenen sistem, sınırlandırılmış nesneye dayalı bir dille jenerik olarak gerçekleştirilmekte, daha sonra uygun araçlarla istenen donanımın yönelik derlenebilir kaynak kod elde edilmektedir. Böylece donanımlar arası taşınabilir, yazılım mühendisliği kriterlerini yerine getiren, aynı zamanda etkin kaynak kullanımlı ve KÖGS doğasına uygun yazılımlar geliştirilebilmektedir. Metinde sistemin kuramsal altyapısı ve pratikte nasıl gerçekleştirilebileceği üzerinde durulmuştur.

## 1. Giriş

"Küçük ölçekli gömülü sistem" tanımlaması genellikle 8 ya da 16 bit mikrodeneleyici tabanlı, donanım karmaşıklığı düşük elektronik dizgeler için kullanılmaktadır.[1] Kullanım alanları incelendiğinde ise dış dünyayla etkileşimleri yoğun, reaktif sistemler olarak değerlendirilebilirler. KÖGS yazılımları da aynı bakışla işlem performansından çok çevre birimleri denetimine ağırlık verilerek "olaya dayalı" olarak tasarlanırlar.

Mevcut yöntemlerle bu tür sistemler için yazılım geliştirmede bir dizi sorunla karşılaşılır. Geliştirmenin donanım üreticileri tarafından sağlanan geliştirme ortamları üzerinde yapılmasından ötürü; tüm yazılım ekibinin öncelikle kullanılacak donanımın teknik özelliklerine ayrıntılarıyla hakim olmaları gereklidir. Geliştirilen yazılım platforma özgüdür ve hiçbir düzeyde başka bir platforma taşınamaz. Kaliteli bir yazılımda olması beklenen tekrar kullanılabilirlik, okunabilirlik, hataların kolay bulunabilmesi gibi kriterleri sağlamak için gerekli araçlar ve materyal genellikle sağlanmamıştır. Eşzamanlı çalışmayı ve kaynak paylaşımını kotarabilmek oldukça güçtür. Basit işletim sistemleri veya geliştirme ortamlarıyla birlikte verilen SDK ve API gibi hazır yazılımlar belli çözümler getirirse de modellemeden kodlama aşamasına tüm süreci kuşatan, standart bir geliştirme sistemi mevcut değildir.

Bu çalışmada bahsedilen problem irdelenmiş ve bir çözüm önerilmiştir. Temel yaklaşım, geliştirmenin kaynak kod seviyesinden bir üst katmanda, jenerik (donanımdan bağımsız) olarak, hem modern yazılım kriterlerini hem de gömülü bilgisayar sistemlerinin özel yönlerini dikkate alınarak yapılmasıdır. Geliştirilen yazılım istenen hedef donanım için derlenebilir kaynak koda dönüştürülebilecektir.

Bu çalışma İTÜ Bilgisayar Mühendisliğinde hazırlanan “Gömülü Sistemler İçin Bir Geliştirme Ortamı” *“An Advanced Software Development System for Small Scale Embedded Systems”* (ADSES1) bitirme çalışmasına dayanmaktadır. Bitirme çalışmasında kuramsal yapının oluşturulmasının yanı sıra geliştirilen bir masaüstü uygulaması ile basit yazılımlar hazırlanmış ve bir mikrodeneçti üzerinde test edilmiştir. Bu konudaki çalışmalar sürdürülerek yazılımın özellikle kuramsal altyapısı iyileştirilmiş ve bu metnin kapsamını oluşturan ADSES 2.0 geliştirilmiştir.

Sonraki bölümlerde konuyla ilgili önceki çalışmalar ele alınmış, geliştirilen sistemin temel unsurları birer bölümle açıklanmış, ADSES2.0 ile örnek bir proje tasarımı yapılmış ve sonuçlar ortaya konulmuştur.

## 2. İlgili Çalışmalar

Gömülü sistemlere yönelik yazılım geliştirme konusunda literatürde birçok yayın bulunmaktadır. Ancak bu çalışmaların geneli karmaşık sistemlere yöneliktir ve uygulanmaları için bir işletim sistemine ihtiyaç duyulur. Gömülü sistem geliştiricilerine model üzerinde tasarım, ve benzetim olanağı veren, tasarlanan modelden Java kodu oluşturabilen Ptolemy projesi bu çalışmalara örnek verilebilir. [2]

Küçük ölçekli gömülü sistemlere yönelik öne çıkan çalışmalardan biri Processor Expert projesidir. Processor Expert bileşen tabanlı ve nesneye dayalı yazılımlar geliştirmeye olanak verir. Ancak geliştirilen yazılımlar donanım bağımlıdır ve yalnızca Freescale donanımları desteklenmektedir. [3]

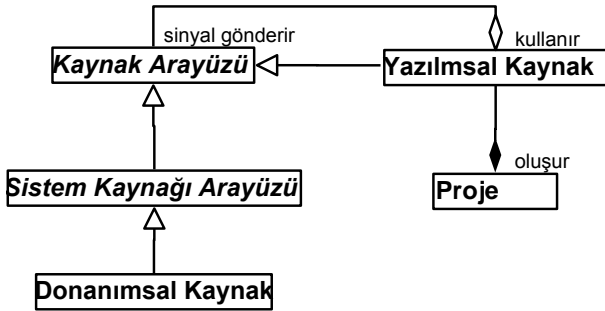
Mathworks firmasının Real-Time Workshop, Stateflow ve Simulink araçları ile model tabanlı mimari kullanarak platformlar arası taşınabilir, tekrar kullanılabilir yazılımlar geliştirilebilmekte, bilgisayar ortamında test ve benzetim

yapılabilmektedir.[4] Oldukça yüksek lisans ücretlerine sahip bu araçlar, genellikle belirli endüstrilerde işaret işleme projelerinde kullanılmaktadır.

### 3. Yazılım Modelleme Yöntemi

İyi bir modelleme ve tasarım için bazı kısıtlamalarla nesneye dayalı yaklaşım temel alınmıştır. Geliştirilecek yazılımların bileşenleri “kaynak” adı verilen aktif nesnelere olacaktır. Genel amaçlı nesneye dayalı yazılımlardaki “içerme” ilişkisinin yerine (bu ilişki de semantik olarak bir nesnenin başka bir nesneye sahip olması üzerinde durulur: kitap ve yazarı gibi) “kullanma” ilişkisi konulmuştur (bu ilişki de ise semantik olarak bir nesnenin başka bir nesneyi kullanması vurgulanır: klima ve termometre gibi).

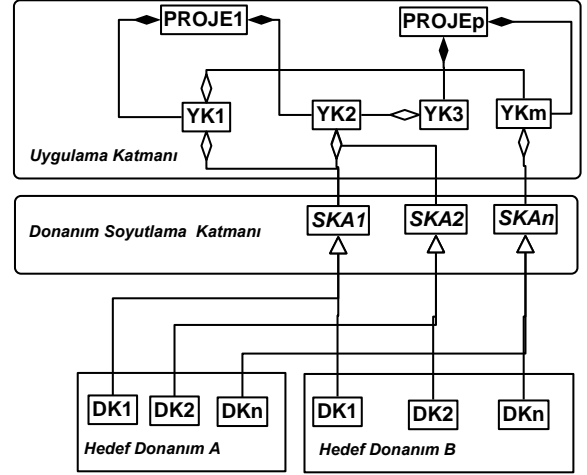
#### 3.1. Modelleme Elemanları



Şekil 1: Modelleme elemanları arası ilişkiler.

Şekil 1’de modelleme elemanları arası ilişkiler UML diyagramı şeklinde gösterilmiştir. Kaynak arayüzü, dışarıdan çağrılan metotlardan ve kendisini kullanan kaynağa gönderilen sinyallerden oluşur. Yazılımsal kaynaklar, başka kaynakları kullanabilir, böylece sınırsız sayıda kaynak oluşturmak mümkün olur. Sistem kaynağı arayüzleri donanımsal modüllere (seri iletişim birimi, ADC gibi) karşı düşer. Bir donanımsal kaynak ise bir sistem kaynağı arayüzünün belli bir hedef donanım için gerçekleşmesidir. Yazılım kaynaklarının birleşiminden oluşan projeler çalıştırılabilir yazılım birimleridir.

Geliştirilen jenerik yazılımların belli bir donanımla kullanılması için sistem kaynaklarının arayüzü kullanılabilen donanımların benzerliklerinden yararlanılarak standartlaştırılmalıdır. Bu kaynakların gerçekleşmesi ise kullanılacak her donanım için farklı olacaktır. Bu yapı işletim sistemi terminolojisindeki “donanım soyutlama katmanı”na karşı düşer.



Şekil 2: Yazılım Katmanları

Yazılımsal kaynaklar ve geliştirilen projeler uygulama katmanını oluşturur. Şekil 2’de yazılım/donanım katman yapısı temsili olarak gösterilmiştir.  $n$  adet sistem kaynağı arayüzü (SKA) donanım soyutlama katmanını oluşturmaktadır. A ve B hedef donanımları için donanım kaynakları (DK) ile sistem kaynakları gerçekleştirilmiştir. Uygulama katmanındaki ilişkiler ise örnek mahiyetindedir. Burada belirtilen yazılım kaynağı (YK) sayısı  $m$  ve proje sayısı  $p$ , gerçekte sonsuzdur.

#### 3.2. Kaynaklar Arası Etkileşim

Bir kaynak nesnesine metotları üzerinden erişilebileceği gibi bu nesne de kendisini kullanan kaynağa sinyal gönderilebilir. Bu etkileşim yöntemi “Qt C++ Application Framework” sinyal/slot mekanizmasının daha basit bir uyarlamasıdır. [5]

```
class Kaynak1{
public:
    Kaynak1();
    void method1(int m){
        //gerekli işlemler
        emit sthHappened;
    }
signals:
    void sthHappened();
private:
    int m_value;
};

class Kaynak2{
    Kaynak1 myKaynak1;
public slots:
    void handler1();//@{myKaynak1.sthHappened}
public:
    Kaynak2();
};
```

Yukarıdaki temsili bildirimde göre, myKaynak1 nesnesinin sthHappened sinyali yayınlandığında (emit sthHappened ile) bu nesneyi kullanan Kaynak2 nesnesinin handler1 metodu tetiklenir. Sinyal-slot bağlantısından dördüncü bölümde açıklandığı üzere çekirdek sorumludur.

#### 3.3. Analiz ve Tasarım

Yeni bir projeye başlanıldığında genel amaçlı nesneye dayalı yazılımlarda olduğu gibi öncelikle istekler belirlenir ve problem domeninin analizi yapılır. Uygulamayı

(gerçeklenecek olan sistemi) oluşturan kavramlar, bunların özellikleri ve aralarındaki ilişkiler tespit edilir. [6]

Tasarım aşamasında ise analiz aşamasında belirlenen kavramların yazılım domeni karşılıkları bulunur. Problem domeni kavramlarına mümkünse eldeki yazılımsal kaynaklar karşı düşürülür. Aksi halde mevcut yazılımsal kaynaklar ve/veya sistem kaynaklarından yenileri oluşturulur.

Tasarımda, bir yazılımsal kaynağa yalnızca onu kullanan nesne tarafından erişileceği göz önüne alınmalıdır. Yazılımsal kaynağın arayüzü (kamusal metodlar ve sinyaller) “bu kaynağı kullanan neleri değiştirmeye ve nelerden haberdar olmaya ihtiyaç duyar” sorusuna göre tasarlanmalıdır. Kaynakları hiyerarşik bir yapıda modelleyerek projeyi büyükten küçüğe parçalara ayırmak (*decompositon*) tasarımı kolaylaştırır.

Tasarlanan yazılımsal kaynaklar ileriki projelerde kullanılmak üzere saklanabilirler. arayüzleri kesişen kaynaklar birbirlerini yerini alabilirler.

Bu şekilde hedef donanımdan bağımsız, tekrar kullanılabilen parçalardan oluşan projeler geliştirilebilir.

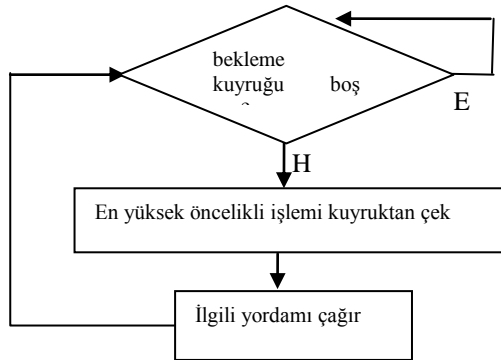
#### 4. Çekirdek

ADSES2.0 ile geliştirilen yazılımlarda yazılım kaynakları arasındaki mesajlaşmayı yürütmek, kesmeleri kotarmak, açılış işlemlerini gerçekleştirmek için bir çekirdeğe (*kernel*) ihtiyaç duyulur.

##### 4.1. Çekirdek Olay Döngüsü

Problemin doğası gereği yazılımın akışı, fonksiyon çağrıları ve koşullu dallanmalardan çok gelen asenkron sinyaller ile tetiklenen yordamların çalışması şeklinde olacaktır. Çekirdek bu iş için çok düzeyli FIFO (ilk giren ilk çıkar) yapısında bir kuyruk içerir.

Çekirdek içinde sinyallerle bunları kataracak yordamların adreslerini tutan bir tablo yer alır. Sinyaller yazılım kaynaklarından yayınlanabileceği gibi kesme altıordamlarından da yayınlanabilirler. Bir olay döngüsü çekirdeğe gelen bir sinyal ile başlar. Çekirdek sinyale karşı düşen yordamın adresini tablodan bulup varsa parametrelerle birlikte bekleme kuyruğuna ekler. Sırası geldiğinde yordam yürütülür ve kuyruktan çıkarılır. Böylece bir çevrim sona erer.



Şekil 3: Olay döngüsü akış şeması

#### 4.2. İş Sıralama

Yukarıdaki akış şemasında görüldüğü üzere kesintisiz iş sıralama yöntemi tercih edilmiştir. Kesintili iş sıralama yöntemine göre çekirdeği basitleştirdiği ve işlem yükünü hafiflettiği için bu yöntem seçilmiştir.[7] Ayrıca semafor ve mutex gibi işletim sistemi yapılarını kullanmaya gerek kalmaz.

Kesintisiz iş sıralamanın en büyük riski çalışmaya başlayan bir yordamın çekirdek tarafından durdurulmamasıdır. Ancak KÖGS'de işlemci yoğunluğunun düşük, G/Ç yoğunluğunun yüksek olduğu göz önüne alındığında tetiklenen bir yordamın çalışma zamanının kısa olacağı varsayılabilir. Bu tezi doğrulamak için donanım kaynaklarının tüm G/Ç işlemleri blokesiz (*non-blocking*) gerçekleştirilmelidir. Ayrıca, yoklama yöntemi yerine her zaman kesmeli çalışma tercih edilmelidir.

Eğer geliştirilecek sistemin işlemci yoğun görevleri varsa işlemler zamana yayılarak sorun giderilebilir. Bunun için görev alt yordamlara ayrılır, bu yordamlar doğrudan çağrılmadan sinyaller ile tetiklenir. Böylece daha yüksek öncelikli görevler araya girebilir.

### 5. Hedef Donanım Betimleme Yöntemi

Belirli bir donanıma özgü kaynak kod oluşturabilmek için hedef donanımı belli bir standarda göre betimleyen veriye sahip olunması gerekir.

Bir hedef donanım betimlemesi; sistem kaynaklarının gerçekleştirilmesi (donanımsal kaynaklar), donanımsal modül yapısı ve birbirleriyle ilişkileri ve kesme hizmet programı tanımlamalarından oluşur.

Geliştirilen yazılımın performansı yapılan jenerik tasarım kadar hedef donanım için yapılan gerçeklemlerin ve konfigürasyonların kalitesine de bağlı olacaktır.

#### 5.1. Donanımsal Kaynak Gerçekleşmesi

Donanım soyutlama katmanındaki sistem kaynağı arayüzlerinin her biri gerçekleştirilmelidir. Eğer bir sistem kaynağının işlevini doğrudan yerine getirecek donanım modülü yoksa diğer modüller tarafından işlev öykünür.

#### 5.2. Donanım Modülleri Konfigürasyonu

Her bir sistem kaynağının hedef donanımın bir donanım modülüne karşı düşürülmesi gerekir. Bunun için hedef donanım konfigürasyonu hedef cihazın sahip olduğu donanım modüllerini ve bunların özellikleri içerir. Bir donanım modülü başka bir donanım modülünü gerektiriyorsa ya da bir modül birden fazla kaynak tarafından kullanılabiliriyorsa bunlar konfigürasyonda belirtilir.

ADSES1'de donanım modülü konfigürasyonları fiziksel olarak, belli arayüzleri gerçekleyen sınıflardan oluşan Java kütüphaneleri şeklinde temsil edilmişlerdir. Konfigürasyonları XML dosyaları şeklinde ifade etmek de uygun bir yöntem olacaktır.

### 5.3. Kesme Konfigürasyonu

KÖGS'de kesme rutinleri farklılık göstermektedir. Bazı donanım üreticileri kesme vektörü kullanırken bazıları kesme kaynağını yoklama yöntemi ile tespit ederler. İlgili kesme bayrağı bazılarında yazılım tarafından temizlenmeli iken bazılarında bu işlem bazı saklayıcıların okunması ile yapılır. Bunun için donanım betimlemesine bir kesme rutini şablonu eklenmesi gerekir.

## 6. Kaynak Kod Üretici

ADSES2 sisteminin çıktısı hedef donanım derleyicileri tarafından derlenebilen kaynak koddur. Kaynak kod üretici aracı, bir projeyi çekirdek ve hedef donanım betimlemesi ile birleştirerek kaynak kod üretir.

### 6.1. Yazılım-Donanım Eşleşmesi

Projenin yazılımsal kaynakları teorik olarak sınırsızdır ancak bunları karşılayacak donanım modülleri ise kısıtlıdır. Ayrıca geliştirici bazı yazılımsal kaynaklara atanacak fiziksel donanım modüllerini kendi belirlemek isteyebilir. Bu sebeple, kaynak kod oluşturulurken projenin yazılımsal kaynaklarına donanım modülleri atanması için bir kaynak paylaşırma algoritması yürütülür. Algoritmanın temel prensibi bir istemci (yazılımsal kaynak ya da donanım modülü) için birden fazla modül uygun ise toplamda en az istenen modülün bu istemciye verilmesidir. Eğer kullanılan sistem kaynakları donanımsal kaynaklarla karşılanıyorsa kod oluşturulamaz.

### 6.2. Programlama Dili Seçimi

Tasarlanan yazılım nesneye dayalı olduğu için C++ dilinde kod oluşturmak daha kolaydır. Bu şekilde proje geliştirme aşamasında yapılan kodlama büyük ölçüde son koda taşınabilir. Öte yandan C++ desteği veren mikrodenetçi sayısı artmaktadır. ADSES1 projesinde MC9S12C32 mikrodenetleyicisi için belirtilen kısıtlamalarla C++ dilinde kaynak kod oluşturulmuş, kod derlenmiş ve çalıştırılmıştır.

Ancak son kodu C dilinde oluşturmak hem taşınabilirlik hem de durağanlık açısından daha iyi sonuçlar verecektir. Bunun için nesneye dayalı yapıdan yordamsal yapıya kodu taşıyan araçlara ihtiyaç duyulur. Bu işlem genel olarak çok zor olsa da bu çalışmada nesneye dayalı programlamanın çok şekillilik (polymorphism), şablonlar (templates) gibi özellikleri kullanılmadığı için teorik olarak mümkündür.

## 7. Örnek Proje

Örnek proje olarak "derslikler için aydınlık kontrol sistemi" tasarlanmıştır.

### 7.1. İstekler ve Kullanım Senaryoları

Dersliklerde elektronik kontrol edilebilir aydınlatma elemanları olduğu varsayılmıştır. Sistem, derslikteki kişileri giriş ve çıkışlarda sayacak içeride kimse kalmadığında aydınlatmayı kapatacaktır. İçeride birileri varsa aydınlığı istenen düzeyde tutacaktır. Derslik içine doğal ışık da gireceğinden aydınlatma elemanlarının çalışma seviyesi aydınlık seviyesi ölçülerek hesaplanmalıdır.

İstenen aydınlık seviyesinin ayarı, içerideki kişi sayısının sorgulanması gibi kullanıcı işlemleri I2C hattı üzerinden yapılan iletişimle gerçekleştirilecektir.

### 7.2. Analiz

Problemin tanımından sistemin varlıklarının elemanlarının kapı sayacı, aydınlık ölçer, aydınlatma elemanı sürücü, kullanıcı arabirimi olduğu anlaşılmaktadır. Ayrıca bunları denetleyecek bir ana denetleyiciye ihtiyaç duyulur.

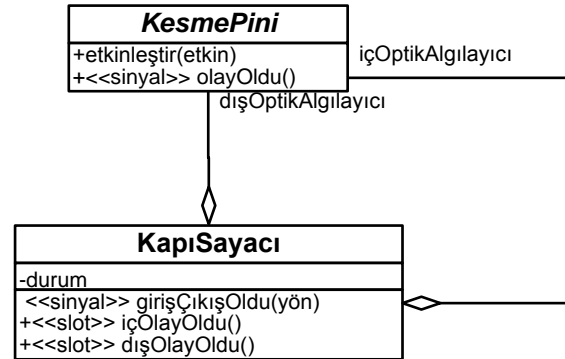


Şekil4:Problem domeni UML diyagramı

### 7.3. Tasarım

Analiz bölümündeki her bir varlık yazılımsal kaynak olarak ifade edilmelidir. Bunun için gerekli sistem kaynakları kullanılmalı ve ilgili sinyallerle metodlar eşleştirilmelidir.

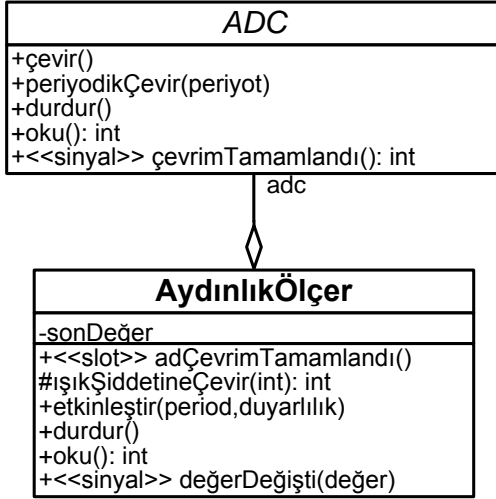
Kapıdaki hareketleri fark etmek ve hareket yönünü anlamak için kapılara bir bit lojik çıkışlı ikişer optik algılayıcı konulmuştur. Algılayıcı önüne bir engel geldiğinde çıkışı değişmektedir. Bu çıkışlar sistemin kesme girişlerine bağlanmalıdır.



Şekil 5:KapıSayacı UML diyagramı

KapıSayacı, KesmePini sistem kaynağını kullanarak optik algılayıcıdan gelen sinyalleri alır. Bir giriş/çıkış tespit ettiğinde ise bunu giriş/çıkış oldu sinyali ile bunu bildirir.

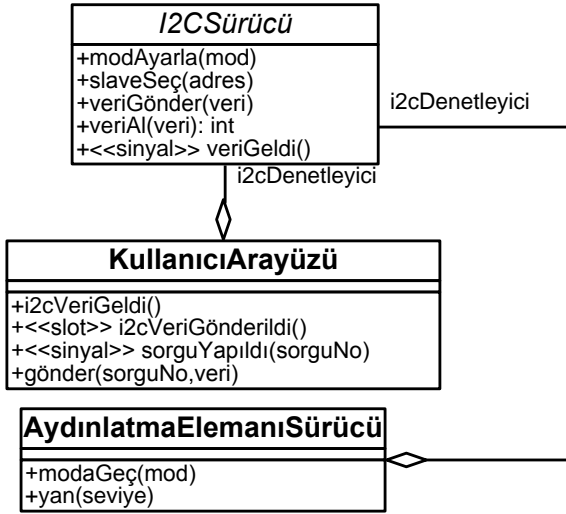
Aydınlık seviyesi bir fototransistörün iki ucu arasındaki gerilimden ölçülecektir. Bunun için bir ADC kanalına ihtiyaç duyulur.



Şekil 8: AydınlikÖlçer UML diyagramı

AydınlikÖlçer yazılımsal kaynağı bir ADC sistem kaynağı kullanır. AydınlikÖlçer etkinleştirdiğinde ADC'yi periyodik ölçüm yapacak şekilde koşullar. Ölçümün bittiğini adÇevrimTamamlandı metodunun tetiklenmesi ile anlar. Eğer son değerle yeni değer arasında anlamlı bir fark varsa değerDeğiştirdi sinyalinin yayınlar.

Kullanıcı işlemleri ve aydınlatma elemanı ile iletişim için I2C protokolü kullanılmıştır.



Şekil 7:KullanıcıArayüzü ve AydınlatmaElemanıSürücü UML diyagramı

Bu sebeple KullanıcıArayüzü ve AydınlatmaElemanıSürücü yazılımsal kaynakları I2CSürücü sistem kaynağını kullanır. AydınlatmaElemanıSürücü yalnızca verilen komutu I2C üzerinden aydınlatma elemanına iletir. KullanıcıArayüzü ise gelen veriyi ayrıştırıp istenen sorguyu sorguYapıldı sinyali ile iletir. gönder metodu ile de istenen veriyi I2C üzerinden iletir.



Şekil 8: AnaDenetleyici UML Diyagramı

AnaDenetleyici yazılımsal kaynağı bir KapaSayacı dizisi, birer AydınlikÖlçer, AydınlatmaElemanıSürücü ve KullanıcıArayüzü kullanır. Bir giriş çıkış algılandığında kişi sayısını günceller, ilk kişi girişi ve son kişi çıkışında aydınlatma elemanın açar/kapatır. Aydınlik değeri değiştiğinde durumu değerlendirerek istenen seviyeye göre aydınlaticiya yeni değerini bildirir.

Şekil 5'den 8'e kadar olan diyagramlarda sinyaller ve bunları kataracak metodlar (slotlar) stereotip şeklinde gösterilmiştir.

#### 7.4. Örnek Kodlama

Örnek olarak KapıSayacı::içOlayOldu ve AnaDenetleyici::aydınlıkDeğiştirdi metodlarının kodlaması yapılmıştır.

```
void KapıSayacı::içOlayOldu(){
    //dış algılayıcıdan geçildiyse:
    if(durum == ICERI GIRILİYOR){
        // g/ç tamamlandı:
        durum = IDLE;
        //içeri girildiğini bildir:
        emit girişÇıkışOldu(ICE);
    }
    //muhtemel dışarı çıkış:
    else if(durum == IDLE)
        durum = DISARI_CIKILIYOR;
}

void AnaDenetleyici::aydınlıkDegisti
(int yeniAydinlik){
    //ölçülen aydınlık yeterli ise:
    if(yeniAydinlik >=
        istenenAydinlikSeviyesi)
        aydinlatici.son();
    //değilse yeteri kadar yak:
    else{
        int fark = istenenAydinlikSeviyesi -
            yeniAydinlik;
        aydinlatici.yan(fark);
    }
}
}
```

## 8. Sonuçlar ve Yorumlar

Küçük ölçekli gömülü sistemler, ihtiyaç duyulmasına rağmen yazılım mühendisliği tekniklerinin fazlaca kullanılmadığı bir alandır. ADSES ile bu ihtiyacı karşılamak için jenerik ve donanıma özgü işlemleri ayıran bir geliştirme süreci ileri sürülmüş, hem modelleme hem de gerçekleştirme aşamaları için özel yöntemler ortaya konulmuştur. Bu ileri geliştirme sistemi KÖGS yazılımlarına donanımlar arası taşınabilirlik özelliği

katmakta ve geliřtiricilere yazılım modüllerinin tekrar kullanılması, yazılımların daha anlaşılabilir olması, hataların daha kolay bulunması gibi bir dizi kolaylık sunmaktadır. İleriki çalışmalarda ADSES'in tam olarak standarda oturtulması ve gerekli geliştirme araçlarının sağlanmasıyla kapsamlı deneysel sonuçlar da elde edilebilir.

## 9. Kaynakça

- [1] R. Kamal, *Embedded Systems : Architecture, Programming and Design*, McGraw-Hill, 2003
- [2] H. Lig ve diğeri, "Overview of The Ptolemy Project", Memorandum UCB/ERL, 2003, <http://ptolemy.eecs.berkeley.edu/>
- [3] Processor Expert, <http://www.processorexpert.com/>
- [4] R. Colgren, Basic MATLAB®, Simulink® and Stateflow®, AIAA, 2007
- [5] J. Blanchette, M. Summerfield, *C++ GUI Programming with Qt4, 2nd Edition*, Prentice Hall, 2008
- [6] F. Buzluca, "Yazılım Modelleme ve Tasarımı Ders Notları", 2010, <http://www.buzluca.info/ymt>
- [7] M. Short, "The Case For Non-preemptive, Deadline-driven Scheduling In Real-time Embedded System", Proceedings of the World Congress on Engineering, London, UK, 2010, s:399



## Çoklu Referans Çerçevesi Kullanan Uyarlanı H.264 Hareket Tahmini Donanımları

Aydın Aysu<sup>1</sup>, Onur Can Ulusel<sup>1</sup>, İlker Hamzaoğlu<sup>1</sup>

<sup>1</sup>Mühendislik ve Doğa Bilimleri Fakültesi  
Sabancı Üniversitesi, İstanbul.

{aydinaysu,onurcansel,hamzaoglu}@sabanciuniv.edu

### Özetçe

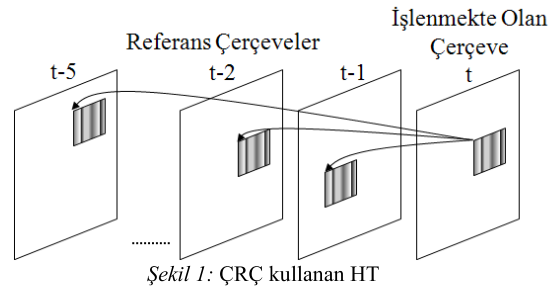
Çoklu referans çerçevesi (ÇRÇ) kullanmak ve yarım piksel hassaslığında (YPH) hareket tahmini (HT) yapmak, video kodlama verimliliğini arttırmakla birlikte işlem karmaşıklığını ve güç kullanımını arttırmaktadır. Bu bildiriye, ÇRÇ kullanan ve işlem karmaşıklığı düşük uyarlanı H.264 HT algoritması, bu algoritmayı gerçekleyen ÇRÇ kullanan düşük enerjili tam piksel hassaslığında (TPH) ve YPH uyarlanı H.264 HT donanımları önerdik. Önerilen ÇRÇ kullanan HT algoritması, her işlenen Makroblok (MB) için uyarlanı olarak belirlediği referans çerçeve sayısı ve erken sonlandırma tekniği ile işlem karmaşıklığını azaltmaktadır. Önerilen ÇRÇ kullanan H.264 HT donanımları Verilog HDL ile gerçekleştirildi. Önerilen ÇRÇ kullanan TPH H.264 HT donanımı her MB için 5 referans çerçeve kullanan TPH H.264 HT donanımına göre ihmal edilebilir bir PSNR kaybı ile Xilinx Spartan 6 FPGA'sı üzerinde %29-72 daha az enerji harcamaktadır.

### 1. Giriş

Hareket tahmini (HT) video sıkıştırma sistemlerinin en çok işlem yapılan ve en çok güç harcanan kısmıdır. HT video sıkıştırma sistemlerinde ardışık çerçeveler arasındaki zamansal fazlalıklardan faydalanarak bit-hızını düşürmek için kullanılır. HT MPEG4 ve H.264 gibi video sıkıştırma standartlarında kullanılmaktadır.

Tam arama (TA) blok eşleştirme HT algoritması, işlenmekte olan blokla Mutlak Fark Toplamı (MFT) kriterine göre en iyi eşleşen referans bloğu arama penceresindeki her arama noktasında arayarak bulur. Daha hızlı blok eşleştirme HT algoritmaları olmasına rağmen, TA algoritması düzenli veri akışı ve iyi sıkıştırma performansı nedeniyle donanım gerçekleştirilmesi için iyi bir adaydır [1].

Çoklu Referans Çerçevesi (ÇRÇ) kullanan HT Şekil 1'de gösterilmiştir. Tekrarlı hareketler, kapsanmamış arka plan, değişken kamera açıları, kamera titremesi, örnekleme hataları, ışık değişmesi, kaynak sinyalindeki gürültü gibi nedenlerden ÇRÇ kullanan HT'nin tek referans çerçeve kullanan HT'den daha iyi video kodlama verimliliği vardır [2]. Fakat, HT'de yapılan işlem miktarı, kullanılan referans çerçeve sayısı ile birlikte artmaktadır. Bu nedenle, HT'de ÇRÇ kullanmak kodlama verimliliğini arttırmakla birlikte işlem karmaşıklığını ve güç kullanımını arttırmaktadır.



Şekil 1: ÇRÇ kullanan HT

Tam piksel hassaslığında (TPH) HT'nin kodlama verimini arttırmak için, yarım piksel hassaslığında (YPH) HT yapılmaktadır [2]. YPH HT, TPH HT'nin bulunduğu hareket vektörünün gösterdiği tamsayı pikselin etrafında oluşturduğu [-1,1] büyüklüğündeki yarım piksel arama penceresinde tamsayı pikselin etrafındaki 8 yarım piksel arama noktasında arama yapar. Yarım piksel arama penceresindeki yarım pikseller, (1/32, -5/32, 5/8, 5/8, -5/32, 1/32) katsayılarından oluşan bir sonlu dürtü yanıtı (FIR) süzgeç kullanılarak oluşturulur. YPH HT TPH HT'ne göre daha fazla işlem içerdiği için kodlama verimliliğini arttırmakla birlikte HT donanımın güç kullanımını arttırmaktadır.

Az PSNR kaybı ile ÇRÇ kullanan HT'nin işlem karmaşıklığını azaltan algoritmalar literatürde önerilmiştir [3-10]. [4-7]'de, komşu blokların en iyi referans çerçevesi (EİRÇ), işlenmekte olan bloğun EİRÇ'sini öngörmekte kullanılmıştır. Bir bloğun EİRÇ o blokla en iyi eşleşen bloğu içeren referans çerçevesidir. [4, 5]'de bazı durumlarda EİRÇ zamansal olarak en yakın olan referans çerçeve olarak alınıp, zamansal ilinti kullanılmıştır. [3]'de hareket vektörlerinin uzamsal ve zamansal ilintisi hareket vektör doğrultusunu hesaplamada kullanılmış ve doğrultu etrafında daha küçük bir arama aralığında arama yapılmıştır. [4, 7]'de hareket vektörünün uzamsal ilintisi aynı zamanda EİRÇ'nin uzamsal ilintisini belirleyen bir kriter olarak kullanılmıştır. Erken sonlandırma, hızlı hareket araması ve simplex azaltması teknikleri de sırasıyla [8], [9] ve [10]'da önerilmiştir.

[3-10]'da önerilen ÇRÇ kullanan HT algoritmalarının donanım gerçeklemeleri yapılmamıştır. ÇRÇ kullanan HT donanımları [11, 12]'de önerilmiştir. Fakat bu ÇRÇ kullanan HT donanımları, HT için sabit sayıda referans çerçeve kullanmakta ve işlem karmaşıklığını azaltmak yerine verileri tekrar kullanma ve çerçeve seviyesinde sıralama teknikleri ile

gereken bellek boyutunu ve bellekten veri okuma miktarını azaltmaya çalışmaktadır.

Bu bildiride, ÇRÇ kullanan ve işlem karmaşıklığı düşük uyarlanı H.264 HT algoritması, bu algoritmayı gerçekleyen ÇRÇ kullanan düşük enerjili TPH ve YPH uyarlanı H.264 HT donanımları önerdik. Önerilen ÇRÇ kullanan HT algoritması, her işlenen Makroblok (MB) için uyarlanı olarak belirlediği referans çerçeve sayısı ve erken sonlandırma tekniği ile işlem karmaşıklığını azaltmaktadır.

Önerilen ÇRÇ kullanan işlem karmaşıklığı düşük HT algoritmasını seri ve paralel olarak gerçekleyen ÇRÇ kullanan değişken blok boyutlu (DBB) TPH H.264 HT donanımları tasarladık. Seri HT donanımı 256 işlem birimi (İB) içermekte ve referans çerçevelerde seri olarak arama yapmaktadır. Paralel HT donanımı ise 512 İB içermekte ve aynı anda iki referans çerçevede paralel olarak arama yapmaktadır. Ayrıca, her MB için 5 referans çerçevede arama yapan DBB TPH bir H.264 HT donanımı (Ref-5) tasarladık. Bu HT donanımı 256 İB içermekte ve TA algoritmasını gerçeklemektedir.

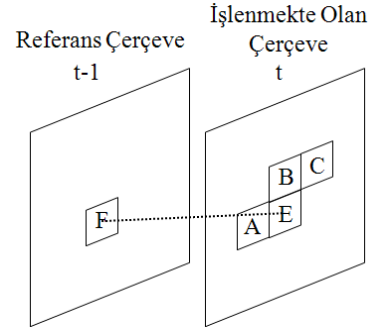
Bu üç ÇRÇ kullanan HT donanımını Verilog HDL ile gerçekledik. Verilog kodları Xilinx Spartan 6 FPGA'sına Synopsys Synplify ile sentezlendi ve Xilinx ISE 11.4 ile en kötü PVT durumunda 58 MHz'de çalışacak şekilde yerleştirildi. Ref-5, seri ve paralel ÇRÇ kullanan HT donanımlarının CIF (352x288) büyüklüğündeki Foreman video dizisi için 50 MHz'deki güç kullanımları Xilinx XPower 11.4 yazılımı kullanılarak belirlendi. Sonuçlar, Xilinx Spartan 6 FPGA'sı üzerinde, ÇRÇ kullanan seri HT donanımının Ref-5'e göre ihmal edilebilir bir PSNR kaybı ile %29-72 daha az enerji harcadığını, ÇRÇ kullanan paralel HT donanımının ise Ref-5'e göre ihmal edilebilir bir PSNR kaybı ile %21-64 daha az enerji harcadığını göstermiştir.

ÇRÇ kullanan seri TPH HT donanımını kullanarak ÇRÇ kullanan YPH uyarlanı bir HT donanımı geliştirdik. Bu donanımın güç kullanımını azaltmak için bir teknik önerdik ve bu tekniği HT donanımında gerçekledik. Bu teknik Foreman video dizisindeki ilk 100 çerçeve için 0.075 dB PSNR kaybıyla YPH HT donanımının güç kullanımını %7.5 azalttı.

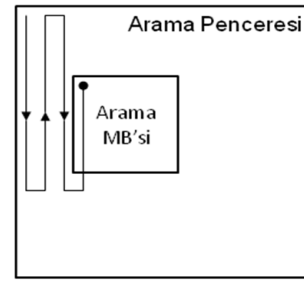
Bildirinin geri kalan kısmı şu şekilde organize edilmiştir. 2. bölümde önerilen ÇRÇ kullanan işlem karmaşıklığı düşük H.264 HT algoritması anlatılmış ve simülasyon sonuçları verilmiştir. 3. bölümde önerilen ÇRÇ kullanan TPH H.264 HT donanımlarının mimarileri anlatılmış ve gerçekleştirme sonuçları verilmiştir. 4. bölümde önerilen ÇRÇ kullanan YPH H.264 HT donanımının mimarisi anlatılmış ve gerçekleştirme sonuçları verilmiştir. 5. bölümde sonuçlar sunulmuştur.

## 2. Önerilen ÇRÇ Kullanan HT Algoritması

Önerilen ÇRÇ kullanan HT algoritması uzamsal ve zamansal ilintileri kullanarak işlenmekte olan MB'nin EİRÇ'ni öngörmeye çalışmaktadır. Uzamsal ve zamansal ilinti Şekil 2'de gösterilmiştir. Komşu A, B ve C MB'lerinin EİRÇ'lerinin ortancası işlenmekte olan MB için uzamsal ilintili EİRÇ adayı olarak seçilir. Önerilen algoritmanın zamansal ilintiyi kullanması [3-5]'de kullanılan zamansal ilintiden farklıdır. F MB'nin EİRÇ işlenmekte olan MB için zamansal ilintili EİRÇ adayı olarak kullanılır. F MB'si E MB'ye zamansal olarak en yakın olan referans çerçevede E MB'nin pozisyonundaki MB'dir.



Şekil 2: Uzamsal ve Zamansal İlinti



Şekil 3: Zig-Zag Arama Akışı

Önerilen ÇRÇ kullanan uyarlanı HT algoritması:

Adım 1 - Zamansal ilintiyeye göre EİRÇ adayında TA HT'ni yap.

Adım 2 - Uzamsal ilintiyeye göre EİRÇ adayında TA HT'ni yap.

Adım 3 - 1 ve 2'de elde edilen MFT'lerden Lagrangian maliyet kriterine göre en iyi sonucu seç. Eğer seçilen sonuç eşik değerinden ( $T_1$ ) küçükse bu sonucu veren referans çerçevesi EİRÇ seç ve 5. adıma git.

Adım 4 - Ek iki referans çerçevede TA HT'ni yap, ek iki referans çerçeveden elde edilen ve 3. adımda elde edilen MFT'lerden Lagrangian maliyet kriterine göre en iyi sonucu seç.

Adım 5 - İşlenmekte olan MB için EİRÇ seçildi, sonraki MB'ye geçip 1. adıma git.

MB'nin 8x8, 8x16 ve 16x8'lik alt blokları farklı referans çerçevelerden seçilebilir. Fakat işlem karmaşıklığını azaltmak için sonraki MB'lar tarafından uzamsal ve zamansal ilintilerde sadece 16x16'lık MB'nin EİRÇ kullanılmaktadır. Sadece uzamsal ve zamansal ilintiyi kullanmak bir süre sonra EİRÇ adayının hep aynı çerçeveye yakınsamasına neden olacağı için, önerilen algoritma uzamsal ve zamansal ilintili çerçevelerdeki arama sonuçlarının  $T_1$  eşik değerinden büyük olması durumunda ek iki referans çerçevede daha arama yapmaktadır.

Bu algoritmanın 1, 2 ve 4'üncü adımlarında yapılan HT'de, TPH HT donanımı Şekil 3'te gösterildiği gibi referans çerçevedeki arama penceresinin sol-üst köşesinden başlayıp

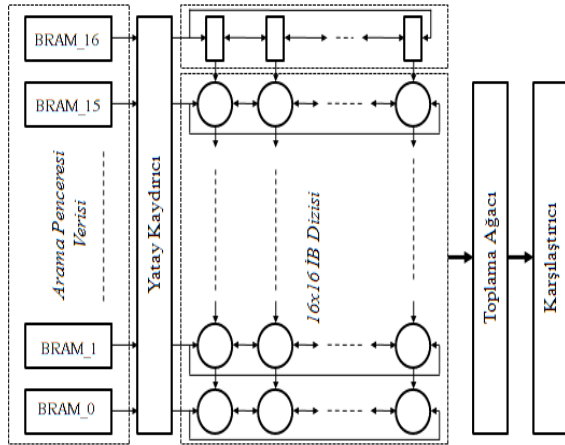
zigzag şeklinde sağ-alt köşesine kadar arama işlemini bitirip kip seçimi donanımını çalıştırmaktadır. Fakat çoğu zaman, arama penceresinin ortalarına doğru en iyi sonuçlar (en düşük MFT'ler) bulunmakta, sağ alt köşeye doğru aramaya devam etmek ancak büyük hareket olduğu zaman PSNR değerini arttırmaktadır. Bu nedenle TPH HT donanımı arama penceresinde aramasına devam ederken, 16 x 16 blok boyutunun MFT değerinin bir eşik değerinden ( $T_2$ ) düşük olması durumunda arama penceresinin tamamı aranmadan erken sonlandırma yapılabilir. Geliştirdiğimiz algoritmanın 1, 2, ve 4'üncü adımlarında erken sonlandırma durumu olursa, algoritma 5. adıma atlayıp bir sonraki MB için arama işlemini yapmaya başlar. Erken sonlandırma tekniği HT algoritmasının işlem karmaşıklığını azaltmakla birlikte, en iyi sonucu veren arama noktasında arama yapılmadan işlemin erken sonlandırılması durumunda PSNR kaybına neden olabilir.

Her MB için tek referans çerçeve kullanan HT algoritması (Ref-1), her MB için 5 referans çerçeve kullanan HT algoritması (Ref-5) ve önerilen ÇRÇ kullanan uyarlanı HT algoritması MATLAB'da gerçekleştirildi ve aşağıdaki ayarlar ile simülasyon yapılarak PSNR sonuçları alındı.

- TPH HT yapıldı.
- [-16,+16] arama penceresinde TA yapıldı.
- $T_1$  için {0, 500, 1000, 1500, 2000} olmak üzere 5 farklı değer kullanıldı.
- $T_2$  için 500 değeri kullanıldı.
- Video dizisindeki ilk çerçevede çerçeve içi diğer çerçevelerde ise sadece çerçeveler arası sıkıştırma yapıldı.
- 100 tane CIF boyutunda (352 x 288) çerçeve kodlandı.
- Hata ölçütü olarak parlaklık PSNR'ı kullanıldı.

Tablo 1: Önerilen Algoritmanın ve Ref-1'in Ref-5'e göre Ortalama PSNR kaybı

	Önerilen $T_1=0$	Önerilen $T_1=500$	Önerilen $T_1=1000$	Önerilen $T_1=1500$	Ref-1
Foreman	0.050	0.255	0.387	0.502	1.003



Şekil 4: 256 İB DBB Hareket Tahmini Donanımı

Ref-1 algoritmasının ve önerilen ÇRÇ kullanan uyarlanı HT algoritmasının farklı eşik değerleri için Ref-5 algoritmasına göre PSNR kaybı Tablo 1'de gösterilmiştir. Erken sonlandırma kullanılmazsa ( $T_2 = 0$ ), önerilen HT algoritmasının PSNR kaybı ortalama 0.01dB artmaktadır.

Önerilen ÇRÇ kullanan uyarlanı HT algoritmasıyla farklı  $T_1$  ve  $T_2$  eşik değerleri kullanılarak farklı miktarlarda işlem yapıp farklı kalitede sonuçlar almak mümkündür. Uygun eşik değerleri kullanılarak, önerilen algoritma ihmal edilebilir bir PSNR kaybıyla işlem miktarını büyük oranda (%30-75) azaltabilir. Önerilen algoritmanın  $T_1$  eşik değeri düşüğe HT'de kullandığı ortalama referans çerçeve sayısı artmaktadır. Önerilen algoritma eşik değerine bağlı olarak 1.2 ile 3.5 arasında referans çerçeve kullanmaktadır.

### 3. Önerilen ÇRÇ Kullanan TPH HT Donanımları

[1]'de önerdiğimiz tek referans çerçeve kullanan DBB TPH H.264 HT donanımının blok diyagramı Şekil 4'te gösterilmiştir. Bu HT donanımı [-16,15] arama penceresinde TA yapmaktadır ve 256 İB'den oluşan 2 boyutlu sistolik İB dizisi kullanılmaktadır. Şekildeki her daire bir İB'yi temsil etmektedir. Bütün İB'ler verileri yukarı, aşağı ve sola kaydırma özelliğine sahiptir.

Her İB işlenmekte olan MB'dan bir piksel ile arama penceresindeki bir pikselin mutlak farkını hesaplamaktadır. Bir arama noktasının MFT'si İB'lerde hesaplanan mutlak farkların toplama ağacı ile toplanmasıyla elde edilmektedir. Bu HT donanımı boruhatlıdır ve gecikme süresi sekiz saat çevrimdir. Bir çevrim BRAM'den okuma için, bir çevrim yatay kaydırma için, bir çevrim 2 boyutlu sistolik İB dizisinde mutlak farkları oluşturmak için, iki çevrim toplama ağacında 4x4'lük blok boyutundaki MFT değerlerinin hesaplanması için ve üç çevrim toplama ağacında yedi farklı blok boyutu için 41 hareket vektörünü oluşturmak için harcanmaktadır.

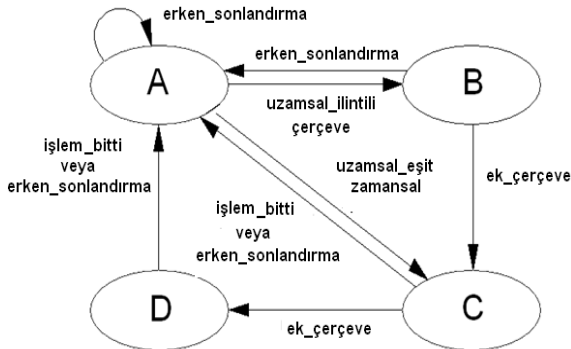
Bu DBB HT donanımı her saat çevriminde bir arama noktasının MFT sonucunu oluşturduğu için, çıkan MFT değerini  $T_2$  eşik değeriyle karşılaştırıp gerekli durumlarda donanımda erken sonlandırma yapılması oldukça kolaydır.

Önerilen ÇRÇ kullanan HT algoritmasının seri gerçekleştirmesinin sonlu durum makinesi (SDM) Şekil 5'de gösterilmiştir. Seri gerçekleştirilmenin her durumunda işlenmekte olan MB ile sadece bir referans çerçeve arasında HT yapılır. A durumunda donanım algoritmanın 1'inci adımındaki zamansal ilintili çerçevedeki HT işlemini yapmaktadır. Bu durumdan sonra eğer bir erken sonlandırma yoksa donanıma B adımını yapmak için gerekli kontrol sinyalleri gönderilir. B durumunda donanım algoritmanın 2'inci adımındaki uzamsal ilintili çerçevedeki HT işlemini yapmaktadır.

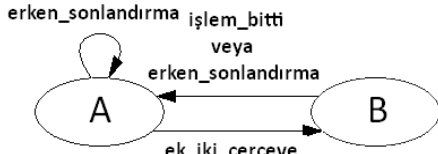
Belirlenen eşik değeri  $T_1$ 'e göre ek çerçeve istenirse ek çerçeve sinyali üretilir ve donanım C durumuna geçer. C ve D durumlarında donanım algoritmanın 4'üncü adımındaki ek çerçevelerdeki HT işlemini yapmaktadır. Eğer uzamsal ve zamansal ilintili çerçeveler aynı çerçeveyse çerçeve kararı donanıma uzamsal\_eşit\_zamansal sinyalini kullanarak B durumunu atlayıp, A durumundan C durumuna geçer. Herhangi bir durumda erken sonlandırma olursa veya D durumunda işlem\_bitti sinyali gelirse donanım A durumuna geri döner ve bir sonraki işlenmekte olan MB için HT işlemini yapmaya başlar.

Önerilen ÇRÇ kullanan HT algoritmasının paralel gerçekleştirilmesinin sonlu durum makinesi (SDM) Şekil 6'da gösterilmiştir. A durumunda donanım algoritmanın 1 ve 2'nci adımlarındaki uzamsal ve zamansal ilintili çerçevelerdeki HT işlemlerini aynı anda paralel yapar. Belirlenen eşik değeri  $T_1$ 'e göre ek çerçeve istenirse donanım tarafından ek\_iki\_çerçeve sinyali üretilir ve donanım B durumuna geçer. B durumunda donanım algoritmanın 3'üncü adımındaki ek iki referans çerçevedeki HT işlemlerini aynı anda paralel yapar. Herhangi bir durumda erken sonlandırma olursa veya B durumunda işlem\_bitti sinyali gelirse donanım A durumuna geri döner ve bir sonraki işlenmekte olan MB için HT işlemini yapmaya başlar.

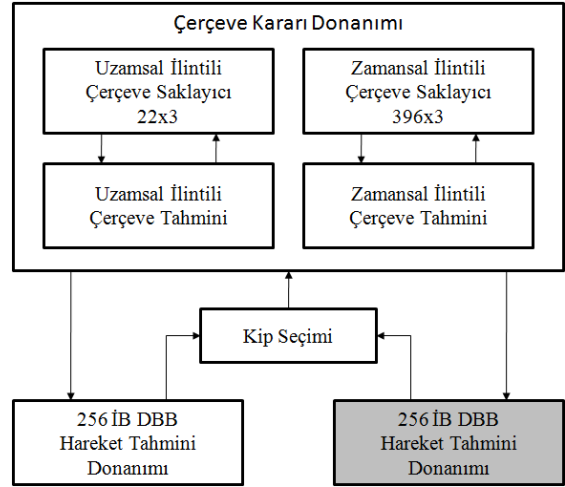
Önerilen seri ve paralel ÇRÇ kullanan TPH uyarlanabilir HT donanımları Şekil 7'de gösterilmiştir. Bu HT donanımlarında [1]'de önerilen 256 İB'si olan ve her MB için tek referans çerçeve kullanan DBB HT donanımı kullanılmıştır. Gri renkli



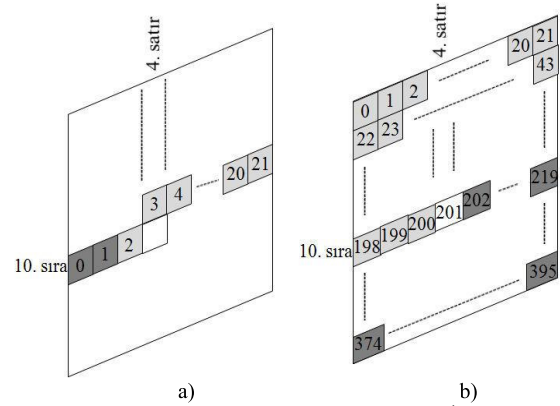
Şekil 5: ÇRÇ Kullanan Seri HT Donanımının SDM



Şekil 6: ÇRÇ Kullanan Paralel HT Donanımının SDM



Şekil 7: ÇRÇ Kullanan TPH Uyarlanabilir HT Donanımlarının Blok Diyagramı



Şekil 8: a) Uzamsal ilintili b) Zamansal ilintili EİRÇ'lerin Bellekte Saklanması

Tablo 2: ÇRÇ Kullanan TPH HT Donanımlarının Güç ve Enerji Kullanımları

	Önerilen Donanım T1=500 T2=0		Önerilen Donanım T1=500 T2=500		Önerilen Donanım T1=1000 T2=0		Önerilen Donanım T1=1000 T2=500		Önerilen Donanım T1=1500 T2=0		Önerilen Donanım T1=1500 T2=500		Ref-5 Donanımı
	Seri	Paralel	Seri	Paralel	Seri	Paralel	Seri	Paralel	Seri	Paralel	Seri	Paralel	
Saat (mW)	80	137	80	137	80	137	80	137	80	137	80	137	76
Lojik (mW)	47	109	41	98	40	84	37	76	35	74	34	69	49
Sinyal (mW)	76	178	73	176	66	135	65	134	61	121	60	119	78
BRAM'ler (mW)	15	33	14	30	14	30	13	29	13	29	12	28	16
Toplam Güç (mW)	218	457	208	441	200	386	195	376	189	361	186	353	219
Zaman (s)	0.032	0.017	0.03	0.015	0.022	0.013	0.02	0.012	0.016	0.01	0.015	0.01	0.045
Enerji (mJ)	6.97	7.77	6.24	6.62	4.4	5.02	3.9	4.51	3.02	3.61	2.79	3.51	9.86
Enerji Farkı	29%	21%	37%	33%	55%	49%	60%	54%	69%	63%	72%	64%	-

256 İB'si olan tek referans çerçeve kullanan DBB HT donanımı ve bu donanımın diğer bloklara olan bağlantısı, ÇRÇ kullanan seri HT donanımında bulunmamaktadır. ÇRÇ kullanan paralel HT donanımı iki adet 256 İB'si olan tek referans çerçeve kullanan DBB HT donanımı kullandığı için, işlenmekte olan MB'nin iki referans çerçevesindeki TA HT'nini aynı anda paralel yapmaktadır. Uzamsal ve zamansal ilintili referans çerçeveler aynıysa 256 İB DBB HT donanımlarının sadece birisi çalıştığı için diğerine saat kapılama yöntemi uygulanarak donanımın harcadığı güç azaltıldı.

Çerçeve kararı (ÇK) donanımı uzamsal ve zamansal ilintili EİRÇ adaylarını belirler ve bu bilgiyi 256 İB DBB HT donanımına gönderir. Kip seçimi donanımı 256 İB DBB HT'nin sonuçlarını kullanarak en iyi blok boyutunu ve EİRÇ'yi seçer. EİRÇ'nin maliyeti  $T_1$  eşik değerinden büyükse, ÇK donanımı ek iki referans çerçeve ister. Kip seçimi donanımı tekrar 256 İB DBB HT'nin sonuçlarını kullanarak en iyi blok boyutunu ve EİRÇ'yi seçer.

Şekil 8 a)'da gösterildiği gibi, işlenmekte olan MB'nin uzamsal ilintili EİRÇ adayı sol, üst ve sağ-üst komşu MB'lerin EİRÇ'leri kullanılarak belirlendiği için bütün bir sıradaki MB'lerin EİRÇ'leri uzamsal ilintili çerçeve saklayıcısında saklanır. ÇRÇ kullanan HT donanımı 5 önceki referans çerçevenin 4'ünü referans çerçeve olarak kullanabildiği için, CIF büyüklüğündeki çerçevelerde uzamsal ilintili EİRÇ bilgisinin saklanması için  $3 \times 22 = 66$  bit yonga üzeri bellek kullanılır. Örneğin, 4. sütunun 10. sırasındaki MB'ye ait uzamsal ilintili EİRÇ adayının bulunması için uzamsal ilintili EİRÇ kararı donanımı 2, 3 ve 4'deki EİRÇ bilgilerini okuyup ortancayı bulur.

Şekil 8 b)'de gösterildiği gibi, işlenmekte olan MB'nin zamansal ilintili EİRÇ adayı, bir önceki referans çerçevede aynı pozisyondaki MB'nin EİRÇ olduğu için, bütün bir çerçevedeki MB'lerin EİRÇ'leri zamansal ilintili çerçeve saklayıcısında saklanır. ÇRÇ kullanan HT donanımı 5 önceki referans çerçevenin 4'ünü referans çerçeve olarak kullanabildiği için, CIF büyüklüğündeki çerçevelerde zamansal ilintili EİRÇ bilgisinin saklanması için  $3 \times 396 = 1188$  bit yonga üzeri bellek kullanılır. Örneğin, 4. sütunun 10. sırasındaki MB'ye ait zamansal ilintili EİRÇ adayının bulunması için zamansal ilintili EİRÇ kararı donanımı 201'deki EİRÇ bilgisini okur.

Önerilen ÇRÇ kullanan HT donanımı her MB için kullanacağı referans çerçeve sayısını dinamik olarak belirler. Örneğin, eğer uzamsal ve zamansal ilintili EİRÇ adayları aynıysa ve bu referans çerçevesinde yapılan arama sonucunda  $T_1$  eşik değerine bakılarak ek iki referans çerçeve istenmezse ÇRÇ kullanan HT donanımı işlenmekte olan MB için sadece bir referans çerçeve kullanır. Eğer uzamsal ve zamansal ilintili EİRÇ adayları farklıysa ve bu referans çerçevelerde yapılan aramalar sonucunda  $T_1$  eşik değerine bakılarak ek iki referans çerçeve istenirse ÇRÇ kullanan HT donanımı işlenmekte olan MB için dört referans çerçeve kullanır.  $T_1$  ve  $T_2$  eşik değerleri ÇRÇ kullanan HT donanımının girdileri oldukları için, her MB için dinamik olarak değiştirilebilirler.

İşlenmekte olan her MB için 5 referans çerçeve kullanan bir TPH H.264 HT donanımı da (Ref-5) tasarladık. Ref-5 donanımı da 256 İB kullanmaktadır ve bu donanımda [1]'de önerdiğimiz tek referans çerçeve kullanan DBB HT donanımını temel almaktadır. Seri, paralel ve Ref-5 ÇRÇ kullanan TPH HT donanımları Verilog HDL ile gerçekleştirildi. Verilog kodları Xilinx Spartan 6 FPGA'sına Synopsys

Synplify ile sentezlendi ve Xilinx ISE 11.4 ile en kötü PVT durumunda 58 MHz'de çalışacak şekilde yerleştirildi.

Ref-5, seri ve paralel ÇRÇ kullanan HT donanımları Xilinx Spartan 6 FPGA'sında sırasıyla 5k, 5.5k ve 10.2k dilim kullanılmaktadır. Ref-5 donanımı saniyede 9 VGA (648x480) çerçevesi kodlayabilmektedir. Seri donanım en kötü durumda saniyede 11 en iyi durumda ise saniyede 45 VGA çerçevesi kodlayabilmektedir. Paralel donanım ise en kötü durumda saniyede 22 en iyi durumda ise saniyede 45 VGA çerçevesi kodlayabilmektedir.

Ref-5, seri ve paralel ÇRÇ kullanan HT donanımlarının Xilinx Spartan 6 FPGA'sında CIF (352x288) büyüklüğündeki Foreman video dizisi için 50 MHz'deki güç kullanımları Xilinx XPower 11.4 yazılımı kullanılarak belirlendi. Güç ve enerji kullanım sonuçları Tablo 2'de gösterilmiştir. Sonuçlar, Xilinx Spartan 6 FPGA'sı üzerinde, ÇRÇ kullanan seri HT donanımının Ref-5'e göre %29-72 daha az enerji harcadığını ve ÇRÇ kullanan paralel HT donanımının Ref-5'e göre %21-64 daha az enerji harcadığını göstermiştir. Bu nedenle HT donanımının kullanılacağı uygulamanın hız ve güç kullanım gereksinimlerine bağlı olarak seri veya paralel ÇRÇ kullanan HT donanımı kullanılabilir.

#### 4. Önerilen ÇRÇ Kullanan YPH HT Donanımı

ÇRÇ kullanan seri TPH HT donanımını kullanarak ÇRÇ kullanan YPH uyarlanır bir HT donanımı geliştirdik. Bu HT donanımının blok diyagramı Şekil 9'da gösterilmiştir. Bu donanımda, 256 İB kullanan DBB TPH HT donanımı bir referans çerçevede aramayı bitirdikten sonra, 41 farklı hareket vektörünü ve bunların MFT'lerini kip seçimi modülüne gönderir ve kip seçimi modülü en iyi kip ve alt-kipleri seçer. YPH HT donanımı seçilen kiplerin blok boyutları için arama penceresinde YPH aradığı ve arama yapar. Çerçeve kararı modülü ise YPH HT sonucunda elde edilen MFT değerlerini kullanarak en iyi referans çerçevesini seçer.

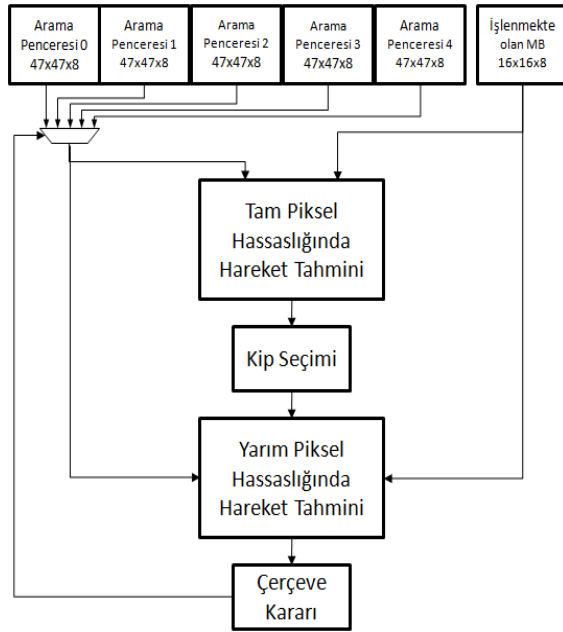
YPH HT sonucunda elde edilen MFT değerleri TPH HT sonucunda elde edilen MFT değerlerinden küçük veya eşit oldukları için önerilen ÇRÇ kullanan HT algoritmasının 3. adımında ek iki referans çerçeve kullanıp kullanılmamaya karar verilirken  $T_1$  eşik değeri ile karşılaştırılan değerler daha küçük olacaktır. Bu nedenle ÇRÇ kullanan YPH uyarlanır HT donanımı ÇRÇ kullanan TPH uyarlanır HT donanımına göre daha az ek referans çerçevede arama yapacaktır. Buda HT donanımının işlem sayısını ve güç kullanımını azaltacaktır.

ÇRÇ kullanan YPH uyarlanır HT donanımının güç kullanımını azaltmak için bir teknik önerdik ve bu tekniği HT donanımında gerçekleştirdik. Önerilen teknik ÇRÇ kullanan HT algoritmasının 4. adımında seçilen ek iki referans çerçevede YPH HT yapılıp yapılmayacağına bu iki referans çerçevede yapılan TPH HT'nin sonuçlarını 1. ve 2. adımlarda zamansal ve uzamsal ilintili referans çerçevelerinde yapılan TPH HT'nin sonuçları ile karşılaştırarak karar veriyor. Eğer sonuçların arasındaki fark bir eşik değerinden ( $T_3$ ) büyükse, bu iki referans çerçevede YPH HT yapmıyor. Çünkü bu durumda ek iki referans çerçevede YPH HT yapılsa bile büyük ihtimalle 1. ve 2. adımlarda yapılan YPH HT'den elde edilen sonuçtan daha iyi bir sonuç elde edilmeyecek.

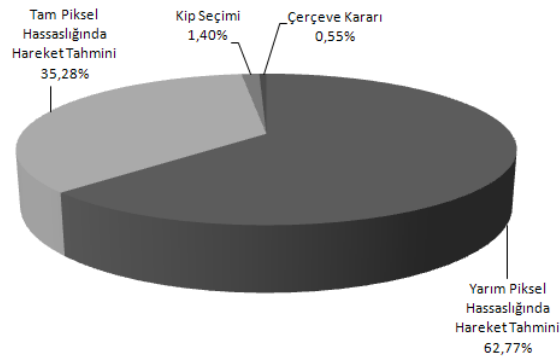
Bu teknik çok az bir PSNR kaybıyla YPH HT donanımının işlem miktarını ve güç kullanımını azaltıyor. Önerilen teknik Foreman video dizisindeki ilk 100 çerçeve için ortalama 0.075 dB PSNR kaybına neden oldu. Bu sonuç

önerilen tekniğin ihmal edilebilir bir PSNR kaybına neden olduğunu göstermiştir. Bu PSNR karşılaştırmasında  $T_3$  değeri  $T_2$ 'ye eşit (500) kullanılmıştır. Farklı  $T_3$  değerleri kullanılarak farklı miktarda PSNR kaybıyla YPH HT donanımının işlem miktarı ve güç kullanımı farklı miktarda azaltılabilir.

ÇRÇ kullanan YPH uyarlan HT donanımı Verilog HDL ile gerçekleştirildi ve Xilinx Virtex-6 FPGA'ya yerleştirildi. Donanım toplam 12k dilim kullanmaktadır. Donanımdaki modüllerin dilim kullanımları Şekil 10'da gösterilmiştir. Önerilen teknik küçük bir donanımla gerçekleştirilmiştir. ÇRÇ kullanan YPH uyarlan HT donanımının Xilinx Virtex-6 FPGA'sında Foreman video dizisinin bir çerçevesi için 50 MHz'deki güç kullanımı Xilinx XPower 11.4 yazılımı kullanılarak belirlendi. Önerilen tekniğin uygulanmadığı HT donanımı 81 mW güç harcamaktadır. Önerilen tekniğin uygulandığı HT donanımı ise 75 mW güç harcamaktadır. Dolayısıyla önerilen teknik YPH HT donanımının güç kullanımını %7.5 azalttı.



Şekil 9: ÇRÇ kullanan YPH Uyarlan HT Donanımının Blok Diyagramı



Şekil 10: ÇRÇ Kullanan YPH HT Donanımının Alan Kullanımı

## 5. Sonuç

Bu bildiriye, ÇRÇ kullanan ve işlem karmaşıklığı düşük uyarlan H.264 HT algoritması ve bu algoritmayı gerçekleyen düşük enerjili TPH ve YPH uyarlan H.264 HT donanımları önerilmiştir. Xilinx Spartan 6 FPGA'sı üzerinde, önerilen ÇRÇ kullanan TPH H.264 HT donanımı her MB için 5 referans çerçeve kullanan TPH H.264 HT donanıma göre ihmal edilebilir bir PSNR kaybı ile %29-72 daha az enerji harcamaktadır.

## 6. Teşekkür

TUBITAK'a 108E239 sayılı proje kapsamında bu çalışmayı desteklediği için teşekkür ederiz.

## 7. Kaynakça

- [1] C. Kalaycioglu, O. Ulusel and I. Hamzaoglu, "Low Power Techniques for Motion Estimation Hardware", Int. Conference on FPL, pp. 180-185, Eylül 2009.
- [2] T. Wiegand, G. J. Sullivan, G. Bjøntegaard, and A. Luthra, "Overview of the H.264/AVC Video Coding Standard", IEEE Trans. on CAS for Video Technology, Temmuz 2003.
- [3] Y. Su and M.-T. Sun, "Fast multiple reference frame motion estimation for H.264/AVC," IEEE Trans. on CAS for Video Technology, vol. 16, no. 3, pp. 447-452, Mart 2006.
- [4] Wu, P., Xiao, C.-B., "An adaptive fast multiple reference frames selection algorithm for H.264/AVC", IEEE International Conference on Acoustics, Speech and Signal Processing, pp. 1017-1020, Nisan 2008.
- [5] L. Shen, Z. Liu, Z. Zhang, and G. Wang, "An Adaptive and Fast Multiframe Selection Algorithm for H.264 Video Coding," IEEE Signal Processing Letters, vol. 14, pp. 836-839, Kasım 2007.
- [6] H. Li, C. Hsu and M. Chen, "Fast Multiple Reference Frame Selection Method for Motion Estimation in JVT H.264", IEEE Asia-Pacific Conference on Circuits and Systems, Aralık 2004.
- [7] SW Ho, SD Kim, MH Sunwoo, "Fast multiple reference frame selection methods for H.264/AVC", IEEE Workshop on Signal Processing Systems, 2008.
- [8] Y. Shen and C. Huang, "Fast multi-frame motion estimation algorithm in H.264", Int. Conference on Signal Processing, 2004.
- [9] C.W. Ting, L. M. Po, and C. H. Cheung, "Center-biased frame selection algorithms for fast multi-frame motion estimation in H.264," IEEE International Conference on Neural Networks and Signal Processing, pp. 1258-1261, Aralık 2003.
- [10] M E Al-Mualla, C N Canagarajah, D R Bull, "Simplex Minimization for Single- and Multi-Reference Motion Estimation", IEEE Trans. on CAS for Video Technology, vol. 11, no. 12, pp. 1209-1220, Aralık 2001.
- [11] T.-C. Chen, C.-Y. Tsai, Y.-W. Huang, and L.-G. Chen, "Single Reference Frame Multiple Current Macroblocks Scheme for Multiple Reference Frame Motion Estimation in H.264/AVC," IEEE Trans. on CAS for Video Technology, vol. 17, no. 2, pp. 242-247, Şubat 2007.
- [12] K. Y. Min, J. W. Chong, "A novel frame reordering scheme and a high speed VLSI architecture of multiple reference frame motion estimator for H.264/AVC", IEEE Trans. on Consumer Electronics, pp. 2394-2400, Kasım 2009.

## FPGA Her Beklentiyi Karşılatabilir mi? – Türkiye’de ASIC Üretimi

*Aziz U. Çalışkan*

TÜBİTAK BİLGEM UEKAE, Enstitü Müdür Yardımcısı  
aziz@uekae.tubitak.gov.tr

### Özetçe

Ülkemizde de dünyadaki gelişmelere paralel olarak FPGA kullanımı oldukça yaygınlaşmıştır. Bununla beraber aynı sayısal devreyi hem FPGA ve hem de ASIC ile gerçekleyip sonuçlar karşılaştırıldığında; FPGA ortalama 4 kez daha yavaş çalışırken, 14 kez daha yüksek dinamik güç tüketmektedir. Bunlara ilave olarak FPGA’in kırmık alanı ASIC’e göre 35 kez daha büyük olmaktadır. Bu durum standart hücre ASIC için bile geçerli iken, özel tasarım ASIC ile FPGA farkı daha da artmaktadır. Eğer devre tasarımı savunma sanayi için yapılıyorsa, tasarım performansından beklentilerin artacağı açıktır. Ülkemizde ASIC üretimi, 1999 yılında tasarımın -algoritmanın- güvenliğini sağlamak için TÜBİTAK BİLGEM bünyesindeki YİTAL’deki kripto tümdevresi üretimi ile başlamıştır. Güvenlik ihtiyacına paralel olarak ; - yüksek hız, - düşük güç tüketimi, - küçük kırmık alanı ve - düşük gürültülü analog uygulamalar için YİTAL’de ASIC üretimi sürdürülmektedir. Bu amaçla halen kullanılmakta olan 0.7um Sayısal CMOS teknolojisine ilave olarak 0.25um SiGe HBT BiCMOS teknolojisi ile de 2011 yılı içinde ASIC üretimi sürdürülecektir. Bu bildiride YİTAL’de gerçekleştirilen çeşitli ASIC uygulamaları tanıtılacaktır.





## **Gömülü Sistemler ile Geleceğin Akıllı Cihaz Çözümleri**

*Uygar Doyuran*

Intel Orta doęu, Türkiye, Afrika Bölgesi İş Geliştirme Müdürü  
uygar.doyuran@intel.com

### **Özetçe**

Gömülü sistemler geleceęi nasıl şekillendirecek? İnternet iletişiminin gücünü neredeyse limitsiz çeşitlilikte gömülü cihazlarla genişlettiğimizde ortaya çıkacak etki bir çok teknoloji uzmanının bile hayal edemeyeceęi boyutlara ulaşıyor. Yapılan araştırmalar 2020'de internete baęlı cihaz sayısının 31milyar'a çıkacağı yönünde. Milyarlarca akıllı ve her an iletişimde olan cihaz hayatımızın bir çok alanında yer alarak, tüketici elektronięinden telekomünikasyona, savunma sanayii'nden havacılıęa, saęlıktan endüstriyel uygulamalara, otomotivden enerjiye, kamera ve görüntüleme sistemlerinden digital signage'a kadar bir çok alanda teknolojik dönüşümü sağlayacak. Bu sunumda gömülü sistemler ve tüketici elektronięindeki gelişmeler ele alınacak, yazılım ve donanım tasarımı yapan kurumların kullanabileceęi çözümler hakkında bilgiler verilecektir. Türkiye'deki firma ve üniversitelerin gelişmiş yazılım ve donanım çözümlerini inovatif iş modelleri ile birleştirerek, teknolojik dönüşümde etkin şekilde yer almaları ve öncü konuma gelmeleri için fikirler paylaşılacaktır.



## Savunma Sanayisinde FPGA Kullanımı

*Kutsal Anıl*

PAVO Tasarım Üretim Elektronik Ticaret A.Ş.  
kutsal.anil@pavo.com.tr

### Özetçe

Savunma Sanayi projelerinde FPGA uygulamaları Savunma sanayinde geliştirilen sistemlerin

- Esnek,
- Modüler,
- Yeniden kullanılabilir,
- Reaksiyon süresinin en az ve benzer özelliklerdeki sistemlere göre ucuz,
- Geliştirilebilir, değiştirilebilir,
- Ortaya çıkabilecek yeni istelere ve sorunlara en kısa sürede cevap verebilir

mimariye sahip olmaları hayati önem arz etmektedir. Bu özelliklere sahip sistemlerin karşı sistemlere olan üstünlükleri, hem bugün ve ihtiyaç halinde yeniden düzenlenebilir olabilmeleri sebebi ile yakın gelecekte ortaya çıkabilecek sorunların çözümünde ülkemizin güvenliği bakımından çok önemlidir. Ayrıca Savunma sanayi projelerinde gerçekleştirilen ürünler, birçok örnekte de görülebileceği üzere sonrasında ticari kullanıma sunulmaktadırlar. Aynı zamanda da bu ürünlerin kişisel tüketime uygun fiyatlarda mal edilebilmesi de önem arz etmektedir. FPGA → ASIC dönüşümünün çok hızlı ve nispeten ucuz gerçekleştirilebilmesi sivil uygulamalara olan geçişin gerçekleştirilebilmesini kolaylaştırmaktadır. Sunumda örnekleri ile FPGA kullanılarak gerçekleştirilen sistemler ile DSP ve/veya işlemci kullanılarak gerçekleştirilen sistemler karşılaştırılacak ve verilecek örnekler ile her iki yaklaşımın birbirlerine göre avantaj ve dezavantajları konusu işlenecektir.



## Küp Uydu için Yer İstasyonunda Yazılım Tanımlı Radyo Tasarımı

Seyyid M. Dilek<sup>1</sup>, Osman Ceylan<sup>2</sup>, H. Bülent Yağcı<sup>3</sup>

Elektronik ve Haberleşme Mühendisliği Bölümü  
İstanbul Teknik Üniversitesi, İstanbul.  
{dileks<sup>1</sup>, ceylanos<sup>2</sup>, bulent.yagci<sup>3</sup>}@itu.edu.tr

### Özetçe

Yer İstasyonları ile küp uydu iletişimini sağlayacak sistemler, donanımsal fonksiyonları yazılım tanımlı olarak gerçekleştirebilir. Bu sistemlerde modülasyon, demodülasyon ve kodlama gibi işlemleri sayısal işaret işleyebilen tek bir işlemci ile yapılması donanıma duyulan ihtiyacı azaltmaktadır. Yazılımsal olarak gerçekleştirilecek işlemler hem maliyeti düşürmekte hem de esnek bir yapıyı oluşturmaktadır. Sistemin ileriki aşamalarında daha farklı bant genişliğindeki modem ihtiyacı için yazılım tanımlı radyonun tekrardan programlanabilmesi veya tekrardan değiştirilebilir olması sistemin tasarımında etkinlik yaratır. Test amaçlı olarak yer istasyonları için düşünülen yazılım tanımlı radyo sistemini Analog Devices firmasının ürettiği Blackfin 533 işlemcisi kullanarak dördüncü evre kaydırmalı anahtarlama ( QPSK ) modülasyon tasarımı yapılmıştır.

### 1. Giriş

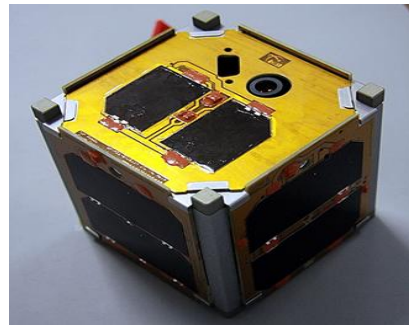
Uydu sistemlerinin en önemli bölümlerinden biri haberleşme sistemidir. Uyduların yerden kilometrelerce uzakta olması iletişimde bazı sorunları ortaya çıkarmaktadır. Yer istasyonları ile uydular arasındaki haberleşmeyi sağlayacak sistemler daha çok klasik RF haberleşme devreleri olup, yeni nesil küp uydu yer istasyonları için yüksek maliyet sıkıntısı oluşturmaktadır. Yeterli esnekliği sağlayamamaları da haberleşme sorunundan dolayı küp uyduların işlevlerini kısıtlamaktadır. Yazılım tabanlı radyolar önemli bir esneklik sağladığı için küp uyduların yer istasyonlarında kullanılması son dönemde gündeme gelmiş ve bazı çalışmalar yapılmaya başlanmıştır. IEEE ve SDR forumu yazılım tabanlı radyo (SDR-YTR) şöyle tanımlamaktadır; “Fiziksel katman fonksiyonlarının hepsinin ya da bir kısmının yazılım tabanlı olduğu radyodur”, bu tanım ile beraber sayısal işareti antene olabildiğince yaklaştırma ihtiyacı doğmaktadır. Çalışmamızda sayısal işaret işleyici (DSP-Sİİ) kullanarak yer istasyonunda donanımı en aza indirmek ve yer istasyonuna esneklik kazandırmak amacıyla uydu haberleşmesinde kullanılan YTR için yazılım tabanlı QPSK modülasyon gerçekleştirilmiştir.

### 2. YTR ve Küp Uydular

YTR teknolojisi birçok uygulama alanında esnekliği ve kolay adaptasyonu sayesinde dikkatleri üzerine çekmeyi başarmıştır[1]. Temel amacı radyo sinyallerini işlemek olan

sistemin basit tanımı; donanımdan olabildiğince kaçınmak ve donanımsal fonksiyonları yazılım tabanına çekmektir. Bu bağlamda donanım ve yazılım maliyetlerini azaltmak amacıyla yüksek veri hızındaki modemleri desteklemek için uydu haberleşme sistemlerinde kullanılmaktadır. Bunun yanında yer istasyon sisteminin farklı frekans bantlarında ekstra donanım gerektirmeden sadece yazılım güncellemeleri yardımı ile çalışmasına olanak sağlayabilmektedir.

Maliyeti makul seviyelerde tutabilmek için YTR ile yapılan tasarımların kullanımı daha fazla önem taşımaktadır. Küp uydu piko-uydular için bir standarttır. Küp uydu standardı California Polytechnic State Üniversitesi ve Stanford Üniversitesi tarafından belirlenmiş olup çok sayıda üniversite ve kurum tarafından küp uydular üzerine çalışma yapılmaktadır[2]. Bu standart düşük maliyetli uydu fırlatma olanağı sağlamaktadır. İstanbul Teknik Üniversitesi'nin geliştirdiği İTÜ-pSAT I (Şekil 1) uydusu küp uydular standartlarına uygun bir şekilde 2009 yılında uzaya fırlatılmıştır. Proje Türkiye'de Küp uydu programına kabul edilen ilk küp uydu projesidir.



Şekil 1: İTÜ-pSAT I küp uydusu.

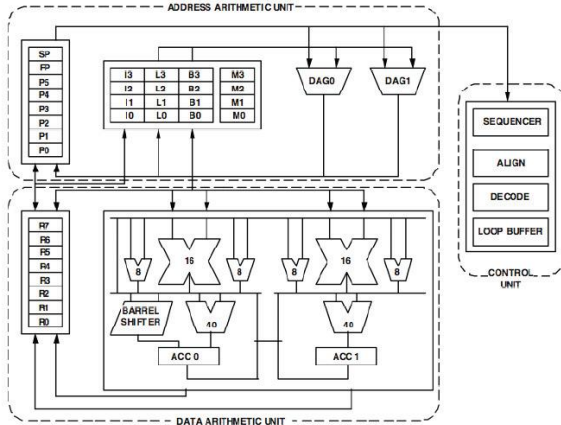
YTR'nin sahip olduğu modülasyon, kodlama, bant genişliği, yeniden kullanılabilirlik ve uzaktan kontrol özellikleri yardımı ile küp uydular ile yer istasyonları arasında ki haberleşmeyi daha ekonomik hale dönüştürmektedir. Küp uydular projeleri genel itibarıyla öğrenci projeleri olduğundan ekonomik sıkıntıları azaltmak için donanımı yazılım tabanına çekebilmek önem taşır.

### 3. Küp Uydular ile Haberleşmede QPSK

Evre kaydırmalı anahtarlama (PSK) genel olarak frekans kaydırmalı anahtarlama (FSK) sistemlerinden daha iyi bit hata oran (BER) performansına sahiptir. Bu bağlamda ikili evre kaydırmalı anahtarlama (BPSK) ve QPSK karşılaştırması yapmak gerekmektedir. QPSK aynı bant genişliği ile BPSK dan bit hızı iki kat daha fazladır, aynı BER performansına sahip olmakla beraber çözünürlüğü daha fazladır. Bununla beraber QPSK modemler güç ve spektrum etkinliğinin yanında uydular haberleşme sistemleri açısından çok etkilidir. QPSK geniş ölçüde mevcut haberleşme sistemlerinde, CDMA içeriğinde, kablosuz yerel ağlarda ve dijital video yayını yapan uydularda kullanılır. Çalışmamızda yer istasyon sistemlerinde kullanılması düşünülen modemlerde Sİİ kullanarak yazılım tabanlı QPSK modülasyonunun yapılması daha uygun görülmüştür.

### 4. Blackfin 533 DSP özellikleri

ADSP-BF533 işlemcisi Analog Devices firması tarafından üretilen 16-bit sabit noktalı bir işlemcidir. Blackfin tek işlemde birçok veri (SIMD) özelliğine sahip. İşlemci çekirdeğinde Şekil 2 de, iki 16-bit çarpıcı, iki 40-bit toplayıcı, iki 40-bit aritmetik lojik ünite (ALU), dört 8-bit video ALU ve 40-bit kaydırıcı içermektedir [3].



Şekil 2: Blackfin Çekirdeği.

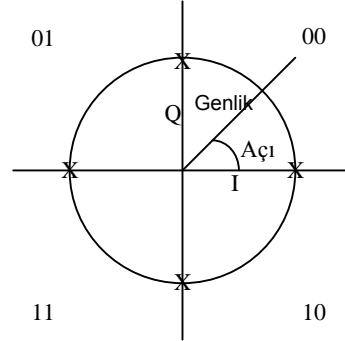
İşlemci 0.8 voltta 100 MHz' den 1.45 voltta 750 MHz hıza kadar işlem yapabilmektedir.

### 5. Sİİ ile QPSK Modülasyon Tasarımı

Dijital uygulama alanında ara frekans (IF) işaret işlemleri için Sİİ kullanımı birçok avantaj sağlamaktadır. Bunlardan bir kaç; maliyeti düşürmesi, donanım karmaşıklığını önlemesi, güvenilirliği artırması ve donanım kopyalamayı basitleştirmesi, genlik dengesini ve doğrusallığı geliştirmesi,

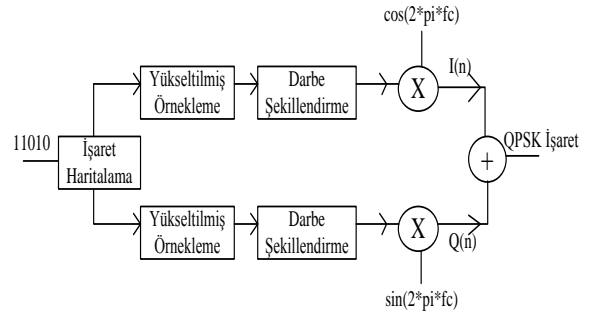
bant genişliği ve giriş işareti genliği konusunda esnek bir kullanım sağlamasıdır.

İşaret eş-evre(I) ve dördül-evre(Q) bileşenlerine sahip olup (Şekil 3) Sİİ sistemlerinde sayısal işaret işleminin bu bileşenler üzerinden yapılması tasarıma esneklik ve doğruluk katmaktadır.



Şekil 3: İşaretin I-Q Bileşenleri ve Sembol oluşturma

QPSK modülasyon gerçekleştirilmesinde kullandığımız Sİİ işlemcisi sabit noktalı bir işlemcidir. İşlemlerimizi işlemciyi daha etkin kullanabilmek için kendisinin tasarladığı değişken değerleri ve fonksiyonlar kullanıldı. Bit işlemlerinde IQ modülasyonlarından yararlandı. IQ modülasyon dijital haberleşme sistemlerinde yaygın olarak kullanıldığı için tercih edilmiştir.



Şekil 3: QPSK Modülasyon

#### 5.1 İşaret Haritalama

Gelen bit dizisini iki bit olacak şekilde işaretler oluşturuldu, bu işlemden sonra I ve Q değerlerine ayırıp haritalandı. Haritalama işlemi Tablo 1 de gösterilmiştir.

Tablo 1: Gray kodlama ile QPSK işaret haritalama

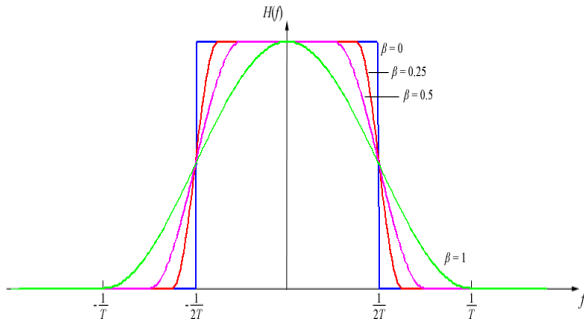
Bit Çifti	I Değeri	Q Değeri	Evre
00	1	1	$+\pi/4$
01	-1	1	$+3\pi/4$
11	-1	-1	$-3\pi/4$
10	1	-1	$-\pi/4$

## 5.2 Yükseltmiş Örnekleme

Uygulamamızda örnekleme katsayısını  $Q=16$  alında, işlem gelen  $I$  ve  $Q$  değerini 16 kez tekrarlaması sonucu örnekleme ve çözünürlüğü artırıldı. Bu işlemi yapmamızda ki temel amaç alıcıda işlemlerin doğruluğunu ve kolaylığını sağlamaktır.

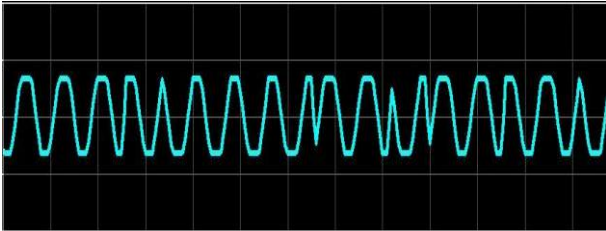
## 5.3 Darbe Şekillendirme Süzgeci

Dijital iletişimde ikili bitler darbe serileridir. Darbelerin iletimi esnasında keskin geçişlerinden kaynaklanacak sonsuz frekans oluşumu sınırlı bant sistemlerde kullanılmaz. Darbe şekillendirme süzgeci bant genişliği ihtiyacı nedeniyle gelen darbeleri süzgeçten geçirerek daha etkin bant genişliğini sağlar. Darbe şekillendirme yönteminde azalma katsayısı  $\beta=0.25$  olarak temel yükseltmiş kosinüs darbesi kullanıldı. Bu işlem sonrasında elde edilen işaret darbe işaretinden daha iyi spektral özellikler taşımaktadır.

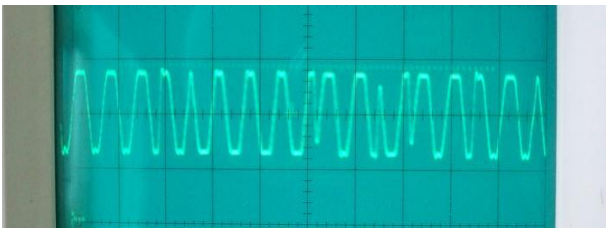


Şekil 3: Darbe Şekillendirme Süzgeci

## 5.4 İşaretin Sanal ve Gerçek Zamanlı Karşılaştırılması



Şekil 4: İşaretin VisualDSP++5.0 Programı Yardımı ile Çizdirilmesi



Şekil 5: İşaretin Osiloskoptan Görünümü

## 6. Sonuç

Yer istasyonları ile küp uydular arasında haberleşmeyi sağlaması düşünülen YTR sistemi için QPSK modülatör tasarımı gerçekleştirildi. Modülatör SİI yardımıyla tamamıyla yazılım tabanlı ve sabit nokta özelliği kullanılarak tasarlandı. Tasarım esnasında işlemlerin sabit noktalı olması,

modülasyon, süzgeç ve yükseltmiş örnekleme gibi işlemleri *fract.h* kütüphanesi kullanarak, işlemciyi daha etkin kullanmada yardımcı oldu .

## 7. Gelecek Çalışmalar

Gerçeklememiz test amacı ile olduğu için düşük frekanslar üzerine çalışılmıştır. İTÜ Uçak-Uzay Fakültesi ve Elektrik-Elektronik fakültesinin beraber yürüttüğü proje kapsamında daha yüksek frekanslarla çalışan analog dijital dönüştürücüler (ADC) ve dijital analog dönüştürücüler (DAC), paralel işlem özelliğine sahip alan programlanabilen kapı dizileri (FPGA) kullanılması düşünülmektedir. Projenin 437.25 MHz frekans bandında çalışılacak olması YTR tasarımı için yukarıda saydığımız donanımların ve daha çoklu modülasyonlar için ise yazılım ihtiyacı doğurmaktadır. Bu bağlamda FPGA/DSP beraber çalıştırılması düşünülmektedir.

## 8. Teşekkür

Proje kapsamında desteğini sürdüren Uçak Uzay Bilimleri Fakültesi, Uzay Mühendisliği Bölümü öğretim üyesi Prof. Dr. Alim Rüstem Aslan'a ve Ertan Ümit'e teşekkür ederiz. Bu uygulamada emeği geçen RF Elektronik Laboratuvarı'nda çalışmalarını sürdüren Hasan Bellikli, M. Emin Şafak ve Onur Yılmaz'a da teşekkür ederim.

## 9. Kaynakça

- [1] M. N. O. Sadiku and C. M. Akujobi, "Software-defined radio: a brief overview", IEEE Potentials, Vol. 23, No. 4, pp. 14-15, 2004.
- [2] H. Heidt, J. Puig-Suari, A. Moore, S. Nakasuka, and R. Twiggs. "CubeSat: A new Generation of Picosatellite for Education and Industry Low-Cost Space Experimentation." In Proceedings of the 14th AIAA/USU Small Satellite Conference. SSC00-V-5. August 2001.
- [3] ADSP-BF533: High Performance General Purpose Blackfin Processor, Analog Devices, 2010





## Gömülü Sistem ile Akıllı Gaz Algılayıcısı Geliştirme

Gürcan YAVUZ<sup>1</sup>, H.İbrahim ÇAKIR<sup>1</sup>, Sevcan AYTEKİN<sup>1</sup>, Ahmet ÖZMEN<sup>2</sup>

<sup>1</sup>Bilgisayar Mühendisliği Bölümü

<sup>2</sup>Elektrik-Elektronik Mühendisliği Bölümü

Dumlupınar Üniversitesi, Kütahya.

{ozmen, gurcanyavuz, cakirhal, saytekin}@dpu.edu.tr

### Özetçe

Akıllı algılayıcılar, mikrodenetleyici ve/veya gömülü sistem ile donatılmış, belirli bir iletişim kapasitesi olan, üzerinde öz denetim (self-diagnostic) donanımı barındıran algılayıcı sistemlerdir. Bu çalışmada, klasik bir gaz algılayıcısının akıllı gaz algılayıcısına dönüştürülmesi çalışmaları sunulmuştur. Klasik gaz algılayıcısı QCM (Quartz Crystal Microbalance) sensör dizisi, dönüştürücüler ve seri yoldan (RS-232) toplanan verileri gönderen mikrodenetleyicili devreden oluşmaktadır. Linksys WRT54GL kablosuz yönlendiricisinin yeniden yapılandırılması ile elde edilen gömülü sistem sayesinde, algılayıcının performansı test edilebilmekte ve sensör verileri ağ bağlantısı ile yerden bağımsız, on-line olarak işlenebilmektedir.

### 1. Giriş

Gömülü sistem, bir ya da birkaç atanmış görevi yerine getirmek üzere tasarlanmış özel bir bilgisayar sistemidir [1, 2]. Bu görevler genelde gerçek zamanlıdır. Mekanik veya elektriksel bir sistemin alt parçası olarak gerçekleştirildiğinden *gömülü sistem* olarak adlandırılmaktadır. Gömülü sistemlerde donanım olarak genellikle bir mikrodenetleyici veya sayısal işaret işlemcisi (DSP) bulunmaktadır. Sistemin genelde bir göreve atanmış olarak kullanılması, yüksek performanslı işlemci gerektirebilmektedir. Örneğin, görüntü işleme ile ilgili birçok uygulamada (uydu alıcısı gibi), ağ anahtarlarında (switch) yaygın olarak gömülü sistemlerden yararlanılmaktadır.

Gömülü sistemler sadece bir görevi yerine getirmek için tasarlandığından, tasarımcılar bu tür sistemleri fiyatını ve boyutunu küçültürerek, performansını ve güvenilirliğini artırarak optimize etmektedirler. Günümüzde bazı gömülü sistemlerde özel yapılandırılmış işletim sistemi de bulunmaktadır. Bu tür sistemlerde kullanıcılara sağlanan kütüphaneler ile yazılım geliştirmek kolaylaşmıştır [3, 4].

Akıllı algılayıcılar, mikrodenetleyici ve/veya gömülü sistem ile donatılmış, belirli bir iletişim kapasitesi olan, üzerinde self-diagnostic donanımı barındıran algılayıcı sistemlerdir. Self-diagnostic birimi, algılayıcının direk kullanıcıya veya bağlı bulunduğu gözleme sistemine çalışma etkinliğini arttırmak veya bakım giderlerini azaltmakla ilgili bilgi üreten ve gönderen donanımdır. Uzun dönemlerde yardımsız olarak çalışması planlanan uygulamalar için akıllı algılayıcılar klasik algılayıcılara göre avantajlıdır. Bu temel özellikleri sağlayan farklı tiplerde algılayıcılar, özellikle otomotiv endüstrisinde yoğun olarak kullanılmaktadır.

Bu çalışmada, klasik bir gaz algılayıcı üniteye gömülü sistem ilave edilmek suretiyle akıllı bir gaz algılayıcı sisteme

dönüştürülmüştür. Klasik gaz algılayıcı sistem RS-232 seri iletişim arabiriminden veri gönderebildiğinden, sensör verileri deney sırasında bir dosyaya kaydedilmekte ve daha sonra analiz yazılımlarıyla bu dosyalardan alınarak incelenilmektedir. Bu nedenle:

- Saatlerce süren deneyler sırasında analiz yapılamaması vakit kaybına sebep olmaktadır,
- Sensör verilerini kaydetmek için, klasik gaz algılayıcı sistem daima bir bilgisayara bağlı olarak çalışması gerekmektedir.

### 2. Gaz Algılayıcı Sistemler

Gazların analizine duyulan ihtiyaç gün geçtikçe artmakta, bu durum araştırmacıları daha ekonomik olan elektronik gaz sensörleri ve elektronik gaz analiz sistemlerini geliştirmeye yöneltmektedir. İşlev olarak canlıların burnunu taklit ettiğinden, elektronik gaz ölçüm sistemlerine “E-Burun” da denilmektedir.

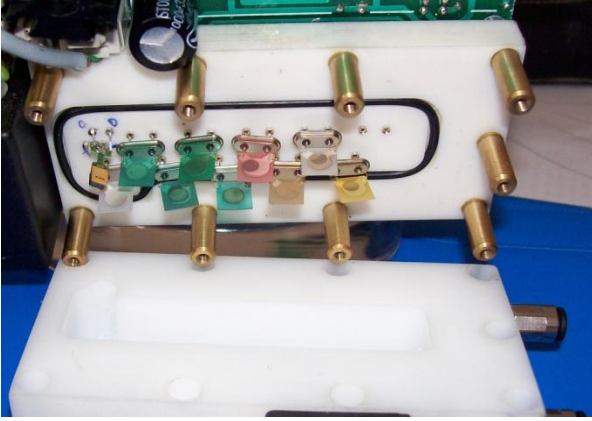
#### 2.1. Gaz Algılayıcılarının Kullanım Alanları

Gaz analizine talebin en fazla olduğu sektörler ve örnek uygulamalar şöyle sıralanabilir:

- *Sağlık sektörü:* İnsan nefesinden veya diğer unsurların kokularından (kan, idrar, ter gibi) hastalık teşhisi,
- *Gıda sektörü:* Kokulu meyvelerin sınıflandırılması, gıdaların tazeliğinin ölçümü,
- *Askeri uygulamalar:* Kimyasal silahların çıkardığı zararlı gazların önceden tespiti, mayın tespiti,
- *Kriminal araştırmalar:* Alkol muayenesi, olay ortamı havasının analizi,
- *Çevre sağlığı:* Hava kalitesi ölçümü,
- *Endüstriyel uygulamalar:* Proses kontrol (üretim kalitesi için), gazların analizi (konsantrasyon oranı, tür belirleme), erken ikaz sistemleri (güvenlik).

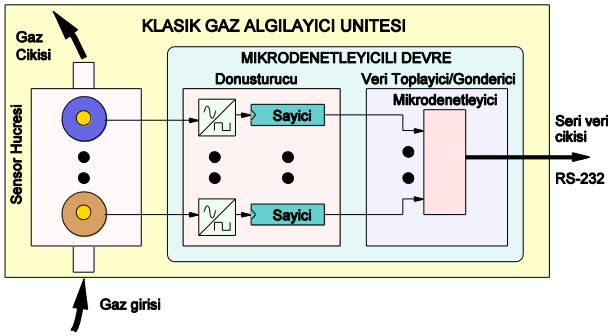
#### 2.2. Klasik Gaz Algılayıcı Sistemler

Gaz sensörleri, test edilen gazın türüne ve/veya miktarına göre bazı elektriksel veya fiziksel özelliği değişen elemanlardır. Gazların analizinde kullanılan sensörler yapısal bakımdan çok çeşitlidir. Metal oksit, iletken polimer, kuvars kristal mikrobals (QCM: Quartz Crystal Microbalance) ve optik sensörler bunlardan bazılarıdır. Bu sensörlerin avantaj ve dezavantajları onların kullanılacağı uygulamaları belirlemektedir. Yüksek konsantrasyonlu gazlara lineer tepki vermesi, neme duyarlılığının az olması sebebiyle, QCM sensörleri endüstriyel uygulamalarda daha fazla tercih edilmektedir.



Şekil 1: Gaz algılayıcı hücresi ve farklı malzemelerle kaplanmış QCM gaz sensörleri.

Gaz analizi için kullanılan klasik gaz algılayıcıları: 1) gaz sensörleri, 2) dönüştürücüler ve 3) veri toplayıcı/göndericiler olmak üzere üç temel bölümden oluşmaktadır. Şekil 1’de klasik bir gaz algılayıcı sisteminin fotoğrafı ve Şekil 2’de ise blok diyagramı görülmektedir [5, 6, 7].



Şekil 2: Klasik QCM sensörlü bir gaz algılayıcı ünite.

Bu çalışmada kullanılan gaz sensörleri kuvars kristalinin yüzeyini hedef gazlara karşı duyarlı kimyasal malzeme ile kaplamak suretiyle elde edilmiştir. Kristal yüzeyindeki kimyasal malzeme ortamdaki hedef gazlarla zayıf bağ kurarak sensör yüzeyinde ortamdaki miktarla orantılı gaz molekülünün yapışmasına sebep olmaktadır. Yüzeye yapışan maddeler, sensöre dönüştürülmüş kuvars kristali ile elde edilen osilatörün titreşim frekansında kaymalara neden olmaktadır. Frekanstaki kayma miktarını gaz kütlesi değişimine bağlayan formül Denklem 1.’de verilmiştir [8]:

$$\Delta f = \frac{-C_f f_0^2}{A} \Delta m \quad (1)$$

Bu denklemde,  $A$  duyarlı bölgenin alanını,  $C_f$  kuvars kristalinin kütle duyarlık sabitini,  $C_0$  kuvars kristalinin temel frekansını ve  $\Delta m$  ise kütle değişimini göstermektedir.

Dönüştürücüler, çoğunlukla analog olan bu değişimleri gözlemleyip sayısalaya dönüştüren kısımdır. Gaz etkileşimi ile direnci değişen sensör sistemlerinde dönüştürücü olarak analog/dijital çeviriciler kullanılmaktadır. Kuvars kristalinin

sensör olarak kullanıldığı algılayıcılarda ise sayıcılar dönüştürücü olarak kullanılmaktadır.

Ölçüm sırasında sayıcılar sensör frekansını sayar, belirli aralıklarla sayma işlemi durdurularak sayıcı içindeki değerin kopyası alınır. Bu işleme örnekleme, iki örnek alımı arasındaki süreye de örnekleme periyodu denilmektedir. Sayma ve örnekleme işlemlerinin birbirini engellememesi için, genelde sensör adedi kadar sayıcı kullanılmaktadır. Kristal frekansları 10-20 MHz civarında olduğu ve saniyelik örneklerde elde edilecek nümerik değer 10 milyonlar mertebesinde olacağı için 24 veya 32 bitlik çok sayıda hızlı sayıcılara ihtiyaç vardır. Bu tür sayıcılar genelde mikrodenetleyiciler içine kısa atanmış bir yazılım yapılarak gerçekleştirilmektedir. Daha iyi diğer bir yöntem ise programlanabilir lojik teknolojisi ile çok sayıda saklayıcının bir entegre içinde gerçekleştirilmesidir.

Veri toplayıcı ve gönderici kısım; dönüştürücülerden alınan sayısal verileri dosyalara kaydetmek üzere başka birimlere gönderen kısımdır. Bu kısım genelde bir mikrodenetleyici ile gerçekleştirilmektedir. Bu mikrodenetleyici, sayıcılardan (dönüştürücülerden) alınan sayısal verileri genelde seri yoldan algılayıcı ünite dışına göndermektedir. İletişim protokolü olarak çoğunlukla RS-232 kullanılmaktadır.

### 3. Akıllı Gaz Algılayıcı Sistemin Tasarımı

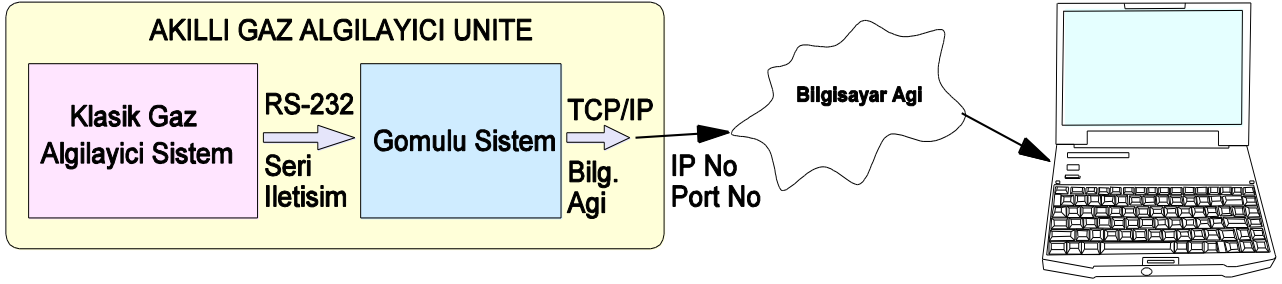
Akıllı algılayıcı sistemleri değişik sensörlerle farklı alanlarda yaygın olarak kullanılmaktadır. Bu çalışmada ise gaz analizinde kullanılabilecek uzaktan erişilebilir bir akıllı gaz algılayıcısı geliştirilmiştir. Geliştirilen sistemin yetenekleri şunlardır: 1) sensör dizilerinden gelen sayısal verileri ön işlemden geçirebilmesi, 2) sistemin kendini belirli aralıklarla test edebilmesi (self-diagnostic) ve 3) ağ bağlantısı üzerinden diğer istemcilere verilerin gönderilebilmesi. Bu amaçla sınırlı donanım üzerine özel yapılandırılmış Linux çekirdeği ve TCP/IP protokol yumağı kullanılmıştır. Algılayıcıya ağ üzerinden birden fazla istemci bağlanıp veri alabilmektedir. Bu sayede, dağıtık ölçeklenebilir bir sistem elde etmenin ötesinde, elle taşınabilir kendi kendine işlevsel sistemler de geliştirilebilecektir.

Verileri işleyen akıllı algılayıcının dışındaki kısım ise genellikle yazılım ve donanım parçalarından oluşan bir bilgisayar sistemidir. Yazılım olarak bilgi çıkarımı yapan lineer ve/veya lineer olmayan algoritmalar, donanım olarak ise masaüstü veya dizüstü bilgisayarlar kullanılmaktadır. Sistemin detaylı blok diyagramı Şekil 3’te gösterilmiştir.

#### 3.1. Sistemin Genel Tasarım Özellikleri

Bu çalışmada, QCM gaz sensörleri içeren klasik bir algılayıcı hücreesine, Linksys WRT54GL kablosuz yönlendiricisinden [9] elde edilen gömülü sistem yeniden yapılandırılarak ilave edilmiş ve akıllı algılayıcı sisteme dönüştürülmüştür. Yeni algılayıcının blok diyagramı Şekil 4.’te gösterilmiştir.

Sistem çalıştırıldığında klasik gaz algılayıcısı içindeki mikrodenetleyici, örneklenmiş sensör verilerini seri yoldan saniyede bir karakter dizisi şeklinde göndermektedir. Gömülü sistemde yapılandırılan sunucu RS-232 den gelen veriyi almakta, diğer taraftan da soket bağlantısı üzerinden bağlı istemcilere göndermektedir.



Şekil 3: Akıllı gaz algılayıcı ünite ve istemci bilgisayar.

### 3.2. Gömülü Sistem

Gömülü sistem, çok kısıtlı kaynaklara sahip özel amaçlar için geliştirilmiş bir bilgisayar olarak düşünülebilir. Gömülü sistemlerde donanım olarak bir CPU, sınırlı miktarda RAM ve FLASH bellek; yazılım olarak ise özel olarak yapılandırılmış bir işletim sistemi ve az sayıda atanmış uygulama yazılımları bulunmaktadır. Genellikle Linux sürümlerinden elde edilen işletim sistemi çekirdeği (kernel), gömülü sistem hangi amaç için kullanılacaksa ona yönelik sistem yazılımı modüllerini barındırmaktadır.

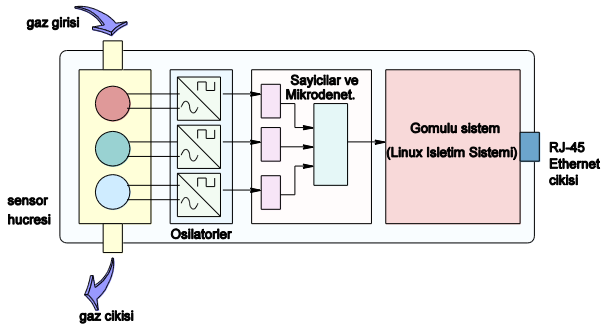
#### 3.2.1. Donanım Çalışması

Bu çalışmada gömülü sistem olarak Linksys WRT54GL kablosuz yönlendiricisi (Wireless Router) yazılımsal ve donanımsal olarak değiştirilerek kullanılmıştır. Bu yönlendiricinin anakartı aslında bir gömülü sistem olarak farklı amaçlar için kullanıma çok uygun olup; kart üzerinde 200 MHz saat frekanslı MIPS işlemcisi ile 16 MByte RAM ve 4 MByte Flash bellek bulunmaktadır.

Kart üzerinde 5 adet RJ-45 yuvası ve 1 adet RS-232 bağlantı yeri bulunmaktadır. Ancak seri haberleşme kanalının kullanılabilmesi için MAX-232 entegresi ile RS-232 soketini barındıran bir iletişim devresi kurulması gerekmektedir.

#### 3.2.2. Yazılım Geliştirme

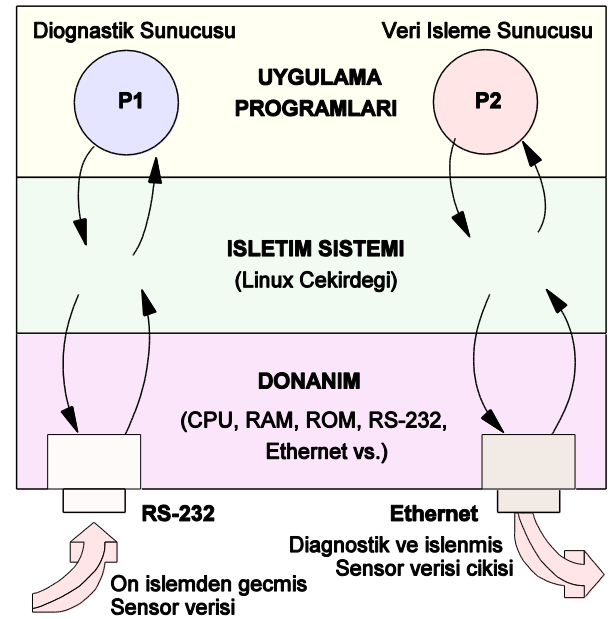
Yönlendirici satın alındığında üzerinde özel gömülü (firmware) yazılımlar bulunmaktaydı. İlk olarak, bu yazılım kaldırılmış ve yerine Linux işletim sisteminin gömülü sistemler için yapılandırılmış bir sürümü olan OpenWRT-0.9 kurulmuştur (Kernel sürüm no: 2.4.30) [10].



Şekil 4: Bilgisayar ağı üzerinden veri gönderebilen akıllı algılayıcının blok diyagramı.

Kodlamaya geçilmeden önce gömülü sistemde çalışacak sunucu programları tasarlandı. Sunuculardan birisi diagnostik sunucusu diğeri ise veri transfer sunucusu olarak belirlendi. Şekil 5'te bu sunucu programların gömülü sistem üzerindeki durumları görülmektedir. Diagnostik sunucusu (Şekilde P1

olarak gösterilmiştir) sistem açıldığında akıllı algılayıcının tüm unsurlarını (algılayıcılar, sayıcılar, RS-232 ve Ethernet iskeleleri) test etmekte bir aksilik olursa istemciye bu durumu bildirmektedir. Ancak, diagnostik taraması başarı ile geçildiğinde veri işleme sunucusu (P2) devreye girmektedir. Veri işleme sunucusu ise seri kanaldan aldığı sensör verilerini IP paketine koyarak istemcilere göndermektedir.

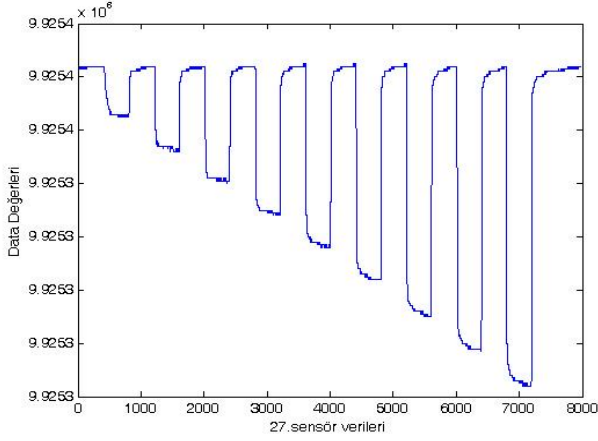


Şekil 5: Gömülü sistemde gerçekleştirilen yazılımlar.

Sunucuları geliştirmek üzere bir başka bilgisayara (host) Linux işletim sistemi kuruldu. Daha sonra OpenWRT-SDK yazılım aracı (derleyici ortamı) indirildi ve yine bu host bilgisayara kuruldu. Sunucular C dilinde kodlandı ve kolay kurulum biçimine sokularak gömülü sisteme aktarıldı. Gömülü sistem için TCP/IP bağlantı parametreleri IP No: 192.168.1.1, Port No: 5001 olacak şekilde ve en fazla 5 istemciye hizmet verecek şekilde ayarlandı.

#### 3.2.3. Test

Geliştirilen sistemi test etmek üzere Matlab'ın grafik aracından yararlanıldı [11]. Matlab programlama dosyası (M-File) için java dilinde kodlanmış yazılım parçaları yerleştirilerek soket bağlantısı ve on-line veri transferi yapılması sağlandı. Bir diziye alınan veriler Matlab grafik komutlarıyla on-line olarak ekrana çizdirildi. Şekil 6'de istemci tarafından on-line olarak çizdirilmiş bir ölçüm grafiği görülmektedir.



Şekil 6: Test sırasında alınan verilerin istemci bilgisayarda çizdirilmesi.

#### 4. Sonuçlar

Bu çalışmada, Linksys WRT54GL kablosuz yönlendiricisinin yeniden yapılandırılması ile elde edilen gömülü sistem kullanılarak, klasik bir gaz algılayıcısı, akıllı bir gaz algılayıcı üniteye dönüştürülmüştür. Gömülü sistem üzerinde yüksek seviye dilde geliştirilen yazılımlar ile algılayıcının doğru çalıştığı test edilebilmekte ve sensör verileri ağ bağlantısı ile yerden bağımsız, on-line olarak alınabilmekte ve işlenebilmektedir. Geliştirilen sistem ile gaz ölçüm deneyleri daha etkin bir şekilde yapılabilmektedir.

#### 5. Kaynakça

- [1] M. Barr. "Embedded Systems Glossary". Netrino Technical Library. <http://www.netrino.com/Embedded-Systems/Glossary>. Last retrieved: 2010-09-17
- [2] S. Heath, "An embedded system is a microprocessor based system that is built to control a function or a range of functions." Embedded systems design. EDN series for design engineers (2 ed.). Newnes. p. 2. 2003.
- [3] Hawkboard, <http://www.hawkboard.org/>.
- [4] BeagleBoard, <http://beagleboard.org/>.
- [5] A. Özmen, E. Doğan E., "Design of a Portable E-Nose Instrument for Gas Classifications", IEEE Transactions on Instrumentation and Measurement, Vol:58, No:10, pp:3609-3618, 2009.
- [6] B. Mumyakmaz, A. Özmen, M.A. Ebeoğlu, C. Taşaltın , "Predicting Gas Concentrations of Ternary Gas Mixtures for a Predefined 3-D Sample Space", Sensors and Actuators B, Vol:128, Issue:2, pp: 594-602, 2008.
- [7] A. Özmen, M.A. Ebeoğlu, F. Tekce, C. Taşaltın, Z.Z. Öztürk, "Finding the Composition of Gas Mixtures by a Phthalocyanine Coated QCM Sensor Array and an Artificial Neural Network", Sensors and Actuators B, Vol:115, Issue:1 pp: 450-454, 2006.
- [8] H.W. King, Piezoelectric sorption detector,
- [9] Linksys WRT54GL kablosuz yönlendirici, <http://www.linksysbycisco.com/US/en/promo/Promotion-Go-Wireless>.
- [10] OpenWrt, <http://openwrt.org/>.
- [11] Matlab, <http://www.mathworks.com/>.

## Balık Çiftlikleri için Uzaktan İzleme Sistemi Tasarımı

Gökhan Bölük<sup>1</sup>, Orhan Başkan<sup>2</sup>

TÜBİTAK Bilişim ve Bilgi Güvenliği İleri Teknolojiler Araştırma Merkezi (BİLGEM)  
Gebze, Kocaeli

<sup>1</sup>gokhan.boluk@ bte.tubitak.gov.tr

<sup>2</sup>orhan.baykan@ bte.tubitak.gov.tr

### Özetçe

Bu çalışmada çiftliklere yerleştirilmiş olan algılayıcılardan elde edilen fiziksel, kimyasal ve biyolojik verilerin Veri İletim Birimi (VİB) aracılığı ile GPRS bağlantısı üzerinden Tarım Bakanlığı kapsamında kurulacak kontrol merkezi uygulama sunucusuna aktarılması amaçlanmıştır. Alınan bilgiler kontrol merkezi uygulama sunucusu üzerinden bir veri yönetim sistemine aktarılmış ve Web tabanlı kullanıcı arayüzü ile görselleştirilmiştir. Bakanlık yetkilisi, çiftlik sahibi veya bilimsel çalışmalar yapacak olan yetkili kullanıcıların tüm bu sonuçlara raporlar ve grafikler ile görselleştirilerek ulaşabilecekleri bir sistem geliştirilmiştir. Algılayıcılardan gelen bilgilerin eşik değerleri geçmesi durumunda tanımlı kullanıcılara ve balık çiftliği yetkililerine SMS ile alarm bilgilendirilmesi sağlanmıştır.

Balık çiftlikleri normalde kıyıda yaklaşık 1.5-2 mil açıkta yer alırlar. Bu koşullarda enerji temininin zorlukları nedeniyle projede mümkün olduğunca az güç tüketen elemanlar kullanılmıştır. Sistem güneş paneli ile şarj edilen 12V/7Ah bir batarya ile beslenecek şekilde tasarlanmıştır. Aynı zamanda deniz içinde yaklaşık 20mt derinlikte algılayıcıların bulunduğu algılayıcı grubu da, geliştirilen VİB tarafından izole edilerek beslenmiştir. Ayrıca VİB ve algılayıcı grubu arasındaki seri veri iletişiminin de galvanik olarak izolasyonu sağlanmıştır. VİB ile kontrol merkezi arasındaki iletişim Wavecom GSM modülü ile sağlanmıştır. VİB'nin tüm çalışma parametreleri ve alarmlar için gerekli olan eşik değerleri kontrol merkezi tarafından uzaktan (bakanlık bünyesinde kurulu sunucudan) tanımlanabilmektedir.

Bu çalışmada açık denizde çalışacak bir sistemin kritik ihtiyaçları; su geçirmezlik (IP67), açık alanda güç gereksinimi, su altı sistemi ile iletişim ve su üstü sisteminin GSM altyapısını kullanarak kontrol merkezi ile iletişimi gibi problemler incelenerek bir Veri İletim Birimi (VİB) tasarımı ele alınmaktadır. Sistemin gömülü yazılım tasarımının Rhapsody tasarım aracı ile durum makineleri tabanlı gerçekleştirilmesi ve C programlama dili ile OpenAT işletim sistemi üzerinde kodlanması betimlenmiştir. GSM altyapısı kullanılarak böyle bir uzak alan veri iletim projesinin nasıl gerçekleştirilebileceği konusunda ip uçları verilmektedir.

### 1. Giriş

Bu Proje Tübitak KAMAG tarafından desteklenen 105G038 nolu proje çalışması kapsamında gerçekleştirilmiştir.

Bu çalışma “Balık Çiftliklerinin Doğal Ekosisteme Etkilerinin Araştırılması Projesi (DENEKO)” kapsamında çiftliklerin uzaktan izlenebilmesi ve online verilerin kaydedilmesi amacıyla başlamıştır. Bilindiği gibi balık çiftliklerinin sayısı günümüzde sürekli yükselen bir artış göstermektedir. Bu artış ile beraber ekosisteme etkileri de sürekli tartışılan bir konu olmuştur.

Sistem iki ayrı tanım kümesinde incelenmiştir.

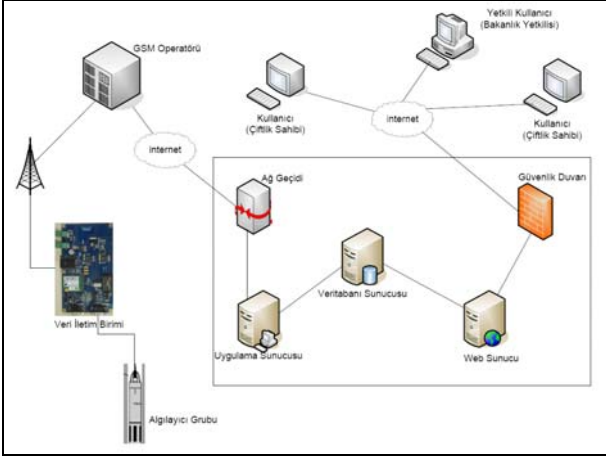
- Kontrol Merkezi: Yazılımı(KMY); Kendi içerisinde iki ayrı alanda tasarlanmıştır. Web tabanlı kullanıcı arayüzü ve VİB Sunucu Yazılımı.
- Veri İletim Birimi (VİB): VİB Donanım ve VİB Gömülü Yazılım olarak iki ayrı alanda incelenmiştir.

### 2. Yetenek ve Kısıtlar

Balık Çiftliklerine yerleştirilmiş olan algılayıcılardan elde edilen fiziksel, kimyasal ve biyolojik veriler VİB aracılığı ile GSM altyapısı kullanılarak GPRS bağlantısı üzerinden Kontrol Merkezi 'ndeki VİB Sunucu Yazılımına gelmektedir. VİB Sunucu Yazılımı'nın daha sonra bir veri tabanı yönetim sistemine ilgili verileri aktarmasıyla sisteme ait tüm verilerin web tabanlı kullanıcı arayüzünde görselleştirilmesi sağlanmıştır.

Toplanan tüm veriler (balık çiftliklerine, VİB'e, algılayıcılara, örnekleme sonuçlarına ve kullanıcılara ait bilgiler), VİB ve algılayıcılara ait alarmlar (pil bitti, kalibrasyon zamanı geldi, veri alınmıyor vs..), raporlar ve grafikler ile görselleştirilmiştir.

Şekil 1'de projenin tüm sistem bileşenleri verilmiştir.



Şekil 1: Sistem Bileşenleri

## 2.1. Veri İletim Birimi (VİB)

Projenin yetenek ve kısıtları değerlendirildiğinde piyasada birçok GSM Modem bulabilmek mümkündür. Bu kapsamda Wavecom firmasının Q2687 GSM modemi tercih edilmiştir. Bu modem üzerinde Open-AT işletim sistemi koşturmaktadır ve 2.5MB programlanabilir hafıza alanı sunmaktadır[1], [2]. Bu açıdan değerlendirildiğinde donanım tasarımı sırasında ayrıca bir mikroşemcili sistem tasarımına gerek kalmamıştır. Modem ile beraber Eclipse IDE ile C programlama dilinde geliştirme yapabileceğimiz bir SDK da mevcuttur.

Proje kapsamında algılayıcı grubu sisteminin deniz altında çalışması ve yaklaşık 100mt uzunluğunda bir kablo ile VİB'ne bağlanacak olması algılayıcı seçimi ve tasarlanan sürücü devre konusunda daha titiz çalışmamıza sebep olmuştur. Bu tip donanımların en büyük problemi tuzlu su kaynaklı problemler ve deniz altında algılayıcılara yuvalanan böcek ve canlılardır. Bu sebeple YSI firmasının 6600V2 algılayıcı grubu tercih edilmiştir[3].

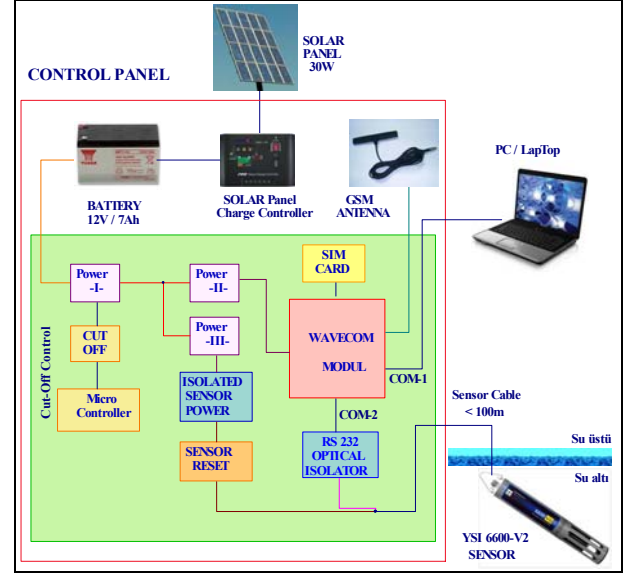
### 2.1.1. VİB Donanım Yetenek, Durum ve Kısıtları

Şekil 2'de blok şeması verilmiş olan VİB'nin genel olarak çalışması şu şekildedir:

VİB, algılayıcı grubundan gelen farklı verileri kullanıcı tarafından ilk bağlantı sırasında belirlenen parametrelerle alacak ve kendi hafızasında saklayacaktır. Yine belirlenen parametrelerle KMY ile iletişime geçecek ve ölçülen algılayıcı değerlerini gönderecektir. VİB'nin çalışması sırasında daha önceden belirlenen alarmlar oluştuğunda VİB KMY ile derhal iletişime geçip alarm bilgilerini gönderecektir.

Herhangi bir nedenden dolayı (güneş olmadığı hava şartları veya solar panelin zarar görmesi v.b.) batarya hiç şarj olamadığı durumlarda sistem ortalama 7 gün çalışabilecek şekilde tasarlanmıştır. Böyle bir durumda sistemin daha uzun süreli çalışması istenirse 12V/12Ah' lik bir batarya ile de sistem sorunsuzca çalışabilmektedir. Batarya deşarj olurken bu süreler içerisinde sisteme herhangi bir müdahalede bulunulmazsa bataryayı korumak amacıyla sisteme mikro kontrolör destekli Cut-off devresi ilave edilmiştir. Bu,

bataryanın 0Volt'a kadar deşarj olup bozulmasını engellemekte ve batarya ömrünün daha uzun olmasını sağlamaktadır. Batarya tekrar şarj olmaya başladığı zaman gerilim daha önce tanımlanan eşik değere ulaşmaya kadar sistem kapalı kalmakta, ancak şarj eşik değeri geçtikten sonra çalışmaya başlamaktadır.



Şekil 2: VİB Sistem Gösterilimi

VİB ile algılayıcı grubu arasındaki kablo maximum 100m olarak öngörülmüştür. RS232 ile sorunsuzca ve izole bir haberleşme sağlayabilmek için de ek devreler tasarlanmıştır. Aynı zamanda kablo üzerinden YSI algılayıcı grubunun beslemesi de izole edilerek sağlanmıştır.

### 2.1.2. Genel Donanım Tasarım Kararları

- VİB deniz üzerinde ve açık denizde çalışacağı için kullanılan elektronik ve mekanik bileşenler deniz koşullarında çalışabilen ve çevreye uyumlu bileşenler olmasına dikkat edilmiştir.
- Elektronik birimlerin gerektiği gibi çalışabilmesi için su yalıtımlarına özellikle dikkat edilmiştir. (IP67)
- VİB beslemesi şarj edilebilir piller üzerinden sağlanmıştır. Bu piller tam boş oldukları zaman 30W' lık bir güneş paneli ile sadece 4 saat gibi kısa bir sürede kontrol devresi üzerinden %95 şarj edilebilmektedir.
- VİB ile KMY arasında GSM altyapısı kullanılarak GPRS bağlantısı ile TCP/IP tabanlı bir iletişim gerçekleştirilmiştir.

### 2.1.3. Genel Yazılım Tasarım Kararları

- VİB Gömülü yazılımı su altında çalışan sistemi gerektiğinde (kalibrasyon ve diğer prob ayarları için) VİB ile algılayıcı grubu arasındaki kablo

bağlantılarını çıkarmaksızın diğer seri haberleşme portu üzerinden haberleşebilecek şekilde tasarlanmıştır.

- Gerektiğinde sualtında çalışan prob ve algılayıcıların çalışma arayüzlerine ulaşabilmek için VİB gömülü yazılımı üzerinde bir telnet protokolü gerçekleştirilmiştir. Böylelikle sahada çalışan tüm algılayıcı sistemlerin arayüzlerine kontrol merkezinden ulaşabilmek ve sistemleri kontrol edebilmek mümkün olmuştur.
- VİB, belirli bir süre boyunca sualtında çalışan algılayıcı grubundan bilgi alamaz ise belirlenen kurallar çerçevesinde algılayıcı grubu besleme gerilimini bir digital I/O vasıtası ile kapatıp açarak ilklendirebilmesi sağlanmıştır.

#### 2.1.4. Operasyonel Kavram ve Senaryolar

VİB, KMY ile ilk bağlantısı sırasında bağlantı başarılı ise çalışma parametrelerini alabilmektedir.

VİB, algılayıcılardan gelen farklı verileri kullanıcı tarafından belirlenen parametrelerle almakta ve kendi hafızasında saklayabilmektedir.

VİB belirlenen aralıklar ile algılayıcı grubundan ölçüm değerlerini alarak -kurulu bulunan GPRS bağlantısı üzerinden- KMY'na belirlenen arayüz üzerinden göndermektedir.

VİB'in çalışması sırasında daha önceden belirlenen alarmlar oluştuğunda VİB KMY ile derhal iletişime geçip alarm bilgilerini göndermektedir.

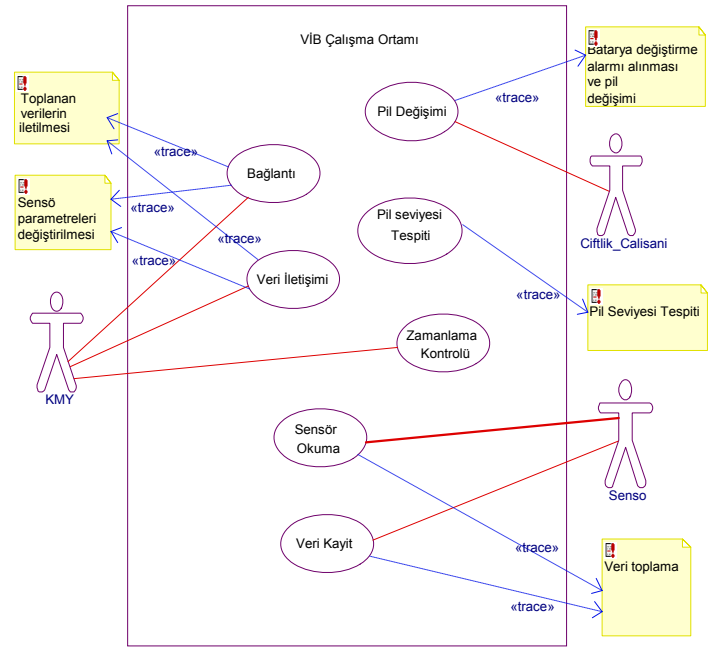
Algılayıcılardan gelen bilgilerin algılayıcı prob parametre eşik değerlerini geçmesi durumunda tanımlı kullanıcılara ve balık çiftliği yetkililerine SMS ile alarm bilgilerini göndermektedir.

Download over the air (DOTA) özelliği ile gömülü yazılımın yeni versiyonları yayınlandığında uzaktan GSM altyapısı kullanılarak GPRS bağlantısı ile yazılımın son sürümü yüklenebilmektedir.

#### 2.1.5. VİB Gömülü Yazılım Tasarımı

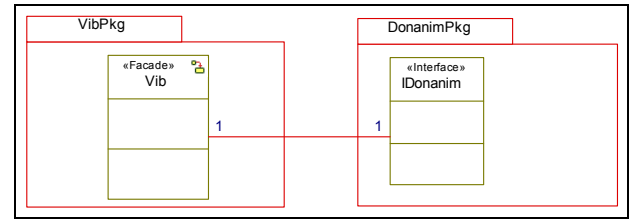
Sistemin gömülü yazılım tasarımı Rhapsody tasarım aracı ile durum makinaları tabanlı gerçekleştirilmiştir. Özellikle Rhapsody tasarımının debug edilebilirliği projenin kodlama aşamasına geçilmeden önce proje ihtiyaçlarının daha iyi anlaşılabilir olmasını sağlamıştır. Bu tasarım mantığı projenin kodlama aşamasını da kolaylaştırmıştır.

Öncelikle sistemin kullanım durumları detaylı olarak ve tasarım kısıtları ile beraber değerlendirilerek tanımlanmıştır. Şekil 3 de sistemin ana kullanım durum diagramı verilmiştir.



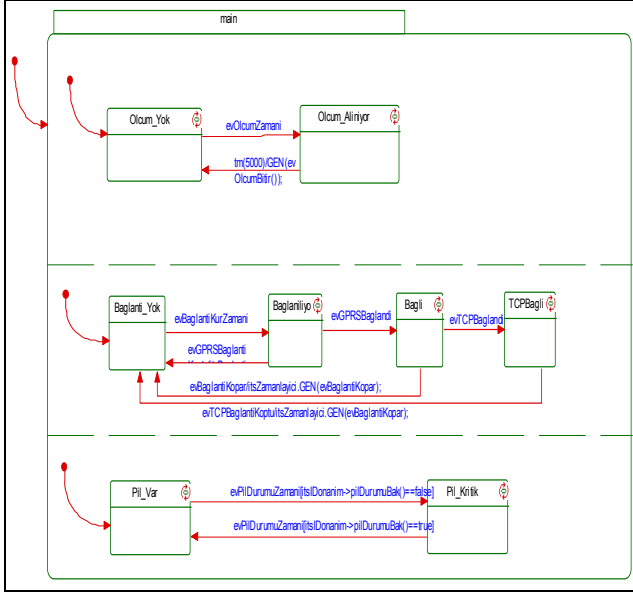
Şekil 3: VİB Kullanım Durumları Diagramı

VİB gömülü yazılımı uygulama modelinde(Şekil 4) daha sonraki dönemlerde kullanılan altyapının değiştirilebileceği düşünülerek (Wavecom, Microsoft, Linux gibi) ayrı bir donanım arayüzü tasarlanmıştır. Böylelikle kodlamanın daha esnek ve kolay anlaşılabilir olması sağlanmıştır.



Şekil 4: VİB Uygulama Modeli

Vib Sınıfı durum makinesi diagramı Şekil 5'de verilmiştir. Diagramdanda anlaşılacağı gibi üç ayrı paralel durum tasarlanmış ve uygulamanın bu durumlar içerisinde koşması sağlanmıştır.



Şekil 5: VİB Durum Makinesi Diagramı

VİB Message sınıfı etkinlik diagramı Şekil 6'daki gibi tanımlanmıştır. Algılayıcılardan gelen mesajlar VİB tarafından ayrıştırılır, değerlendirilir ve yeniden KMY arayüzü formatında oluşturularak kontrol merkezine gönderilir.

Gömülü yazılım tasarımı Wavecom firmasının sağladığı Eclipse IDE Tabanlı geliştirme aracı ile C programlama dili kullanılarak kodlanması betimlenmiştir.

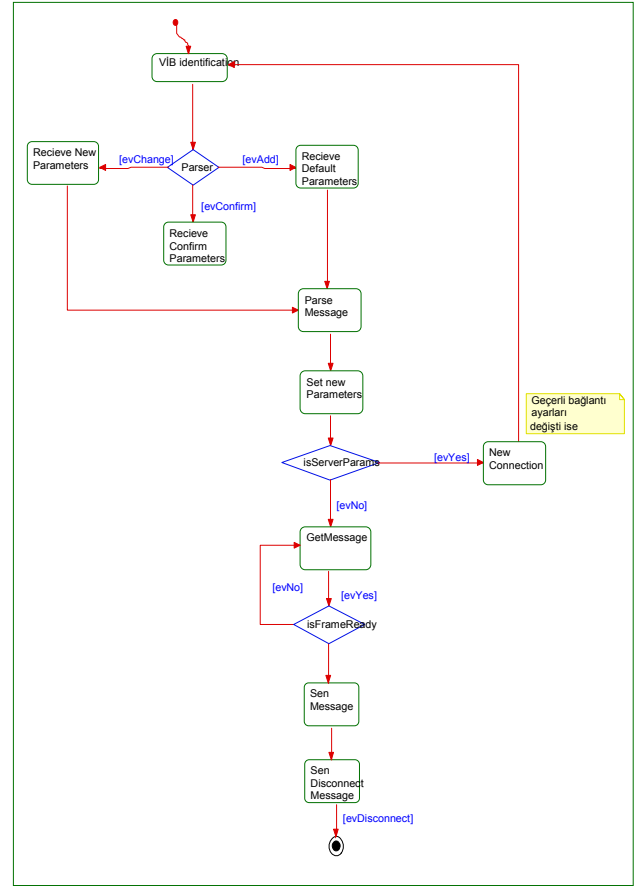
#### 2.1.6. Ağ Güvenliği ve GPRS Bağlantısı

Bu tip bir projede öncelikle kullanılacak olan GSM operatörüne karar verilmesi gerekmektedir. Çünkü seçilen operatörün kapsama alanı projenin kapsamı ve sistemin çalışacak olduğu bölge, bu konuda belirleyici kriterler olacaktır. GSM Operatörü proje için kendi sisteminde bir APN (Access Point Name) adı tanımlayacak ve bağlantılar bu APN adı üzerinden gerçekleştirilecektir.

Tanımlanan APN adı üzerindeki güvenlik ve kullanıcı hakları iki ayrı şekilde gerçekleştirilebilmektedir.

- Birincisi, kullanılan Sim kartların, kullanıcı adlarının ve şifrelerinin GSM operatörünün kendi sunucularında tanımlanması;
- İkincisi ise tüm bu tanımlamaların GSM operatöründen ayrı olarak kurulacak proje sunucularında tanımlanması olacaktır. Bu çalışma yönteminde tüm GPRS kullanıcı yönetimi tamamen proje sunucuları üzerinde proje yöneticilerinin kontrolünde gerçekleşiyor olması olacaktır.

Biz bu proje kapsamında ikinci yöntemi tercih ederek kullanıcı yönetimini kendimiz yapmayı tercih ettik. Bu kapsamda GSM operatörü ile bir tünel protokol ile bağlı Radius Server kurulmuş, Sim kart ve kullanıcılar bu Radius Server üzerinde tanımlanarak yetkilendirilmiştir.



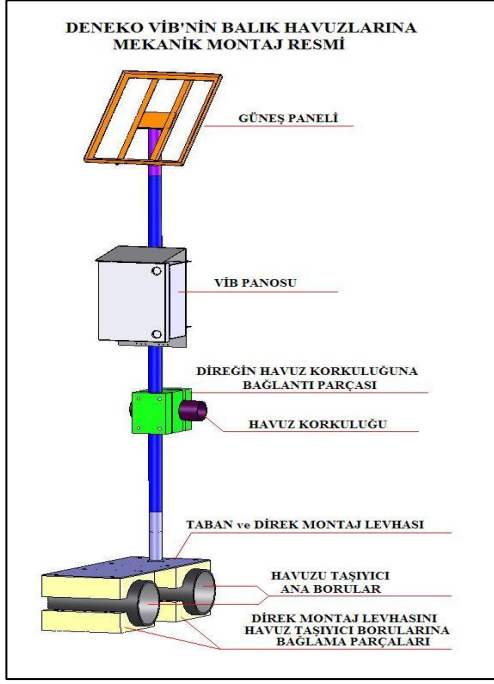
Şekil 6: VİB Message Sınıfı Etkinlik Diagramı

#### 2.1.7. Proje Mekanik Tasarım ve Görüntüleri

Sistemin balık havuzlarına montajının zorlukları dikkate alınarak en uygun tasarım gerçekleştirilmiştir. Mekanik montaj parçaları taşıma ve kaldırılması bir kişi tarafından kolaylıkla yapılabilecek büyüklük ve ağırlıkta tasarlanmıştır.

Şekil 7' de gösterildiği gibi mekanik sistem toplam 5 ana parçadan oluşturulmuştur. Doğrudan su altında kalan, havuzun ana taşıyıcı borularına monte edilen direk montaj levhası taşıyıcı parçaları korozif etkilerden etkilenmemesi, denizin dalgalı olduğu zamanlarda daha kolay esneme yapabilmesi ve dayanıklı olması için kestamitten döküm+talaslı imalat ile ürettirilmiştir. Bunların hemen üzerine hem çalışma sırasında bir iskele oluşturmak hemde direği monte etmek üzere paslanmaz levha kullanılmıştır. Bu levhanın üzerine direk monte edilmiştir. Fırtınalı havalarda veya büyük dalga olan deniz durumlarında mekanik sistemin zarar görmemesi için polyamid malzemeden üretilmiş montaj parçaları ile direk orta noktasından da havuz korkuluklarına bağlanmıştır. Oluşturulan çalışma platformu gözönüne alınarak VİB panosu direğe en uygun yüksekliğe bağlanmıştır. En tepe noktasına da güneş paneli bağlanmıştır. Montaj yapılan havuzun coğrafi konumu dikkate alınarak güneş enerjisinden en fazla yararlanılabilecek yön ayarlanabilmektedir. Mekanik bağlantılarda kullanılan tüm bileşenler deniz suyu veya diğer doğal etkenlerden etkilenmeyecek şekilde paslanmaz seçilmiştir.





Şekil 7: VİB Mekanik Tasarım Çizimi



Şekil 8: VİB Sistem Görüntüsü

VİB panosu cam elyaf destekli polyester malzemeden hazır ürün olarak temin edilmiştir. Kutu IP67 standardında olup,

seçilmesindeki önemli nedenlerden biriside GSM antenin kutu içinde bırakılmasına rağmen herhangi bir anten güç kaybı yaratmıyor olmasıdır.

VİB Sistemi Şekil 8'de fotoğraflandığı gibi kurulmuştur.

### 3. Sonuçlar

Proje mimari olarak klasik araç takip projelerine çok benzemektedir. Projenin bileşenleri kolaylıkla ayrıştırılabilir olup, bu alt bileşenler algılayıcı ağların uzaktan izlenmesine yönelik projelerde kolaylıkla uygulanabilmektedir.

Bu projenin başlangıç amacı çok sıkça tartışılan Balık çiftliklerinin doğal ekosisteme etkilerine dair, bilimsel ve gerçekçi bir cevap oluşturabilmek olmuştur. Projenin başarılı olması aynı zamanda sayıları binleri geçen tüm balık çiftliklerinde uygulanarak çiftliklerin denizel ekosisteme daha az zarar vermelerini, daha bilinçli yemleme yapmalarını sağlayacak ve daha başarılı bir balık üretim potansiyeli doğuracaktır.

Proje yaklaşık 1 yıla yakın bir süre boyunca biri Ege Denizi diğeri Karadeniz olmak üzere iki ayrı sahada test edilmiştir. Tüm sistem bu ortamlarda her türlü hava şartı ve çalışma şekli ile denenmiştir. Sonraki aşamada Bakanlık bünyesine kurulumu ve yaygınlaştırılması çalışmaları devam etmektedir.

### 4. Teşekkür

Projenin gerçekleştirilmesinde destek olan Tarımsal Üretimi Geliştirme Genel Müdürlüğü'ne (TKB-TUGEM), TÜBİTAK, TÜBİTAK MAM Çevre Enstitüsü'ne ve tüm proje ekibi arkadaşlarımıza teşekkürlerimizi iletiriz.

Ayrıca projenin tasarım ve gerçekleşmesi aşamasında destek olan proje yürütücümüz Sn. Fatma Telli Karakoç ve değerli çalışma arkadaşlarımız, Sn. Fikret Şanlı, Sn. Ferhat Yıldız, Sn. Uğur Kavza ve Sn. Ömür Kalkan'a teşekkürlerimizi iletiriz.

### 5. Kaynakça

- [1] AirPrime\_Q2687\_Refreshed\_Product\_Technical\_Specification\_and\_Customer\_Design\_Guideline-Rev003, <http://www.sierrawireless.com/>.
- [2] AirPrime\_Q2687\_Refreshed\_Migration\_Guide-r5, <http://www.sierrawireless.com/>.
- [3] YSI-6-Series-Manual-RevF, YSI 6600 V2 Sonde Spec Sheet.



# ULUSAL AKILLI KART TÜRMEVRESİ ve UKİS İşletim Sistemi

Mustafa Başak, Koral Ataman, Ercan Doğan<sup>1</sup>

<sup>1</sup>TÜBİTAK-UEKAE

Ulusal Elektronik ve Kriptoloji Araştırma Enstitüsü, Gebze.  
{mbasak, koral.ataman, ercan.dogan}@uekae.tubitak.gov.tr

## Özetçe

Akıllı kartlar taşıdığı bilgiler ve kullanım kolaylığı açısından, her geçen gün hayatımıza daha fazla girmektedir. Taşıdıkları bilgi miktarıyla koştur olarak kullanım alanları da artmaktadır. Tersi de doğrudur. Bu döngü böylece sürüp gitmektedir. Son zamanlarda bilginin niceliğiyle birlikte niteliğinin de yükseldiği ve değerinin arttığı ortaya çıkmaktadır. Taşınan bilgi kredi kartlarındaki gibi maddi değer de ifade edebilir, elektronik kimlik kartlarındaki gibi kişinin ömür boyu değiştirilemeyecek olan biyometrik bilgisini de taşıyabilir. Ya da kişiye ait özel bilgileri saklar. Bu nedenle akıllı kartlar artık kişisel doğrulama amaçlı bilgi teknolojileri alanında kullanılmaya başlanmıştır. Bu yazıda özellikle akıllı kart tabanlı kimlik kartları için geliştirilen ulusal akıllı kart türmevresi, bu türmevresinde çalışın işletim sistemi ve çalışmalar sırasında hesaba katılması gereken güvenlik önlemleri üzerinde durulmuştur.

## 1. Giriş

Akıllı kartlar, insanların belli bir mal ve hizmetten yararlanmak amacıyla kişiye özel olarak basılmış plastik kartların evrimi sonucu ortaya çıkmıştır.

Plastik kartların akıllı kartlara dönüşmesinin altında yatan en önemli etken güvenlik eksikliği ve kopyalanma riskidir. Bu iki olumsuzluğu engellemek ancak kişinin bildiği bir bilginin (PIN) veya sahip olduğu bir özelliğın (biyometrik veri) güvenli bir şekilde saklanması ve sorgulanması ile gerçekleştirilir. Bu gereksinimin işlem gücü gerektirmesi nedeniyle nitelsiz plastik kartlar yerine elektronik türmevresine içeren akıllı kartlar kullanılmaya başlanmıştır. Aşağıda Türkiye Cumhuriyeti kimlik kartları için geliştirilmiş ulusal akıllı kart türmevresi ve işletim sistemi ile ilgili bilgiler verilmektedir.

## 2. Ulusal akıllı kart türmevresi ve genel yapısı

Akıllı kart teknolojisindeki gereksinim gözönünde bulundurulduğunda ülkemizde de üzerinde işletim sistemi ve bazı uygulamaların koşacağı akıllı kart yongasının tasarlanması ve gerektiğinde kullanıma alınması amaçlanmıştır. Bu çerçevede TÜBİTAK-UEKAE Enstitüsü Yarı İletken Teknolojileri Araştırma Laboratuvarında (YITAL) ulusal akıllı kart türmevresi tasarlanmış ve bu türmevresinde UKİS işletim sistemi ve elektronik kimlik uygulaması yüklenerek başarı ile çalıştırılmıştır. Geliştirilen

ulusal akıllı kart türmevresinin blok yapısı Şekil 1'de verilmiştir.

Türmevresinde, 8051 tabanlı bir mikroişlemci ile bu işlemcinin eriştiği farklı türde bellek yapıları (ROM, SRAM, Flaş Bellek), kriptoloji algoritma blokları (RSA2048, DES-3DES, AES256), kart okuyucu arayüzü (UART), rasgele sayı üretici ve dış saldırıları sezme/engellemeye yönelik güvenlik devrelerinden oluşmaktadır.

### 2.1. 8051 Mikroişlemci

Standart bir 8051 işlemcisinin tüm komutlarını gerçekleştiren mikroişlemci bloğu, türmevresinin sağlaması gereken güvenlik koşullarına göre özel olarak tasarlanmıştır. Mikroişlemci ROM'dan okuduğu işletim sistemi ve uygulama komutlarını işlemekte, bu sırada gerek duyduğu bellek işlemleri için kendi 256 B'lık iç belleği ile 8 KB'lık statik RAM belleği kullanmaktadır.

### 2.2. ROM Bellek

Enstitümüz tarafından proje çerçevesinde geliştirilen Ulusal İşletim sistemi yazılımı UKİS, 64 KB'lık ROM bloğuna üretim sırasında yerleştirilmektedir.

### 2.3. Flaş Bellek

Kişisel bilgiler, sertifikalar, anahtarlar gibi besleme gerilimi olmadığı durumda da sürekli saklanması gereken veriler 64 KB'lık flaş bellekte tutulmaktadır.

### 2.4. Kriptoloji Algoritma Blokları

Türmevresinde, işletim sistemine simetrik kriptoloji algoritmaları DES-3DES ve AES 256 ile, asimetrik kriptoloji algoritması RSA 1024 / 2048 işlemlerini tam donanımsal olarak gerçekleştirme hizmeti vermektedir. Gerek kriptoloji bloklarına gerek flaş belleğe UKİS'in erişebilmesi için gerekli erişim kütüphane yazılımları da türmevresinde tasarımıyla beraber geliştirilmiştir.

### 2.5. Rasgele Sayı Üretici

Rasgele sayı üretme bloğu, akıllı kart türmevresinin çalışma sırasında gerek duyduğu FIBS-120 standartlarına uygun rasgele sayıları tam donanımsal olarak üretmektedir.

### 2.6. Saldırı Algılayıcıları

Saldırı algılayıcı bloğun işlevi, türmevresinde saklı gizli bilgileri ortaya çıkarmayı hedefleyen bir saldırganın müdahalesini sezdiği anda türmevresinin çalışmasını durdurup saldırıyı engellemesidir.

## 2.7. UART Bloğu

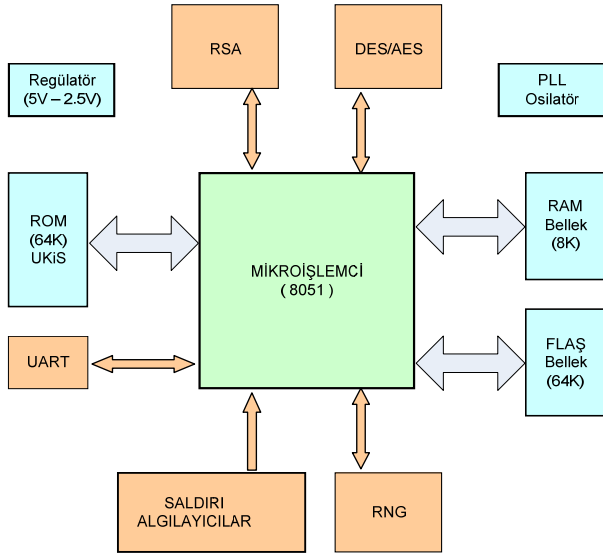
UART bloğu, akıllı kart tümdevresi ile kart okuyucu arasındaki ISO/IEC 7816-3 standartlarına uygun haberleşmeyi sağlamaktadır.

## 2.8. İç osilatör PLL blokları

Tümdevrenin çalışması 5 MHz'lik iç saat osilatörü ile sağlanmaktadır. PLL bloğu ile bu saat işareti gerektiğinde hızlandırılarak işlem süreleri kısaltılmaktadır.

## 2.9. Gerilim Regülatörü

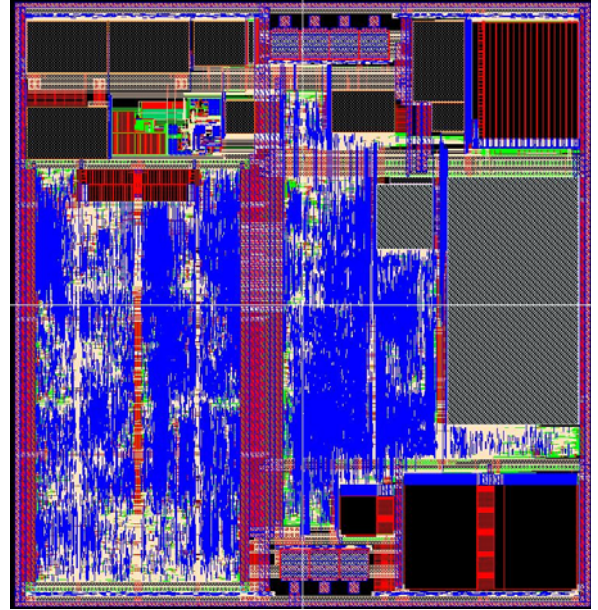
Gerilim regülatörü, devreye dışarıdan uygulanan 3.3V - 5V arasındaki besleme geriliminden yararlanarak, devre içinde kullanılan 2.5 V'luk iç besleme gerilimini üretmektedir.



Şekil 1 : TÜBİTAK-UEKAE UKTÜM yongası

TÜBİTAK – UEKAE’de tasarlanan ve Şekil 1’de blok olarak gösterilen akıllı kart tümdevresi 0.25µm 5-metal CMOS e-Flaş teknolojisi kullanılarak gerçekleştirilmiştir. Ulusal akıllı kart tümdevresinin serimi Şekil – 2’de verilmiştir. Bu tümdevre içerisinde 8051 mikroişlemcisi, DES-3DES, AES ve RSA kriptolojileri, rasgele sayı üretici, saldırı algılayıcıları, UART bloğu proje gerekliliklerine göre ve CC sertifikasyon süreci göz önünde bulundurularak tamamen özgün olarak tasarlanmıştır. Statik bellek, ROM, flaş bellek, iç osilatör ve gerilim regülatörleri yonganın seri olarak üretilmesi için üreticiden hazır olarak alınmaktadır. Çünkü üretici firma yonganın bu bloklarını kendi üretim teknolojisi ile hazırlamaktadır.

Şekil 2’de serimi gösterilen UKTÜM tümdevresi flaş bellek hariçinde 2.2 milyon tranzistör içermektedir.



Şekil 2 : Akıllı Kart Tümdevresi Serim Tasarımı

## 3. Ulusal akıllı kart işletim sisteminin (UKİS) geliştirilmesi

Ulusal bir kimlik kartı ihtiyacı oluşmasından sonra kişinin kimlik bilgilerini taşıyacak güvenli bir kimlik kartına gereksinim duyulmuştur. Bu durumda kimlik kartı projesi olarak ta adlandırılan *elektronik kimlik doğrulama sistemi* projesi başlatılmıştır. Kimlik kartı projesinin temelinde yine TÜBİTAK-UEKAE tarafından geliştirilen açık anahtar altyapısı (AAA<sup>1</sup>, PKI) ve sayısal imza projeleri bulunmaktadır. Akıllı kart işletim sistemi (AKİS), ilk olarak sayısal imza uygulamasında başarılı bir şekilde kullanılmıştır. Daha sonra akıllı kart tabanlı elektronik kimlik doğrulama sistemi (EKDS) çalışmalarına başlanmıştır. Bu çalışmalar, AAA (PKI) projesinde sayısal imza taşıma aracı olarak kullanılan AKİS işletim sistemli akıllı kartlar üzerine eID uygulaması konularak devam etmiştir. Böylece geliştirme aşamasının daha sağlıklı ve güvenli olması planlanmıştır. AAA ve EKDS projelerinde, güvenlik onayı CC EAL5+ seviyesinde olan firmaların donanımları kullanılması kararlaştırılmıştır. Zamanla, EKDS projesi kapsamında, milli yonganın geliştirilmesine başlanmış ve AKİS’in bazı temel yapıları kullanılarak Ulusal Kart İşletim Sistemi (UKİS) geliştirilmiştir. Böylece TÜBİTAK-UEKAE tarafından geliştirilen akıllı kart tümdevresindeki mikroişlemci ünitesinin ve bağlı bulunduğu çevre birimlerinin kullanılması sağlanmıştır. Sonuç olarak, proje kapsamında ulusal kart işletim sistemiyle çalışan bir akıllı kart yongası elde edilmiş oldu. UKİS’te ayrıca eID uygulaması da ROM<sup>2</sup> bellekte bulunmaktadır. O da AKİS gibi sonradan değişmez “native” yapıdadır.

Elektronik kimlik uygulamaları, zamanla değişim gerektiren uygulama kodları içermediğinden var olan veri

<sup>1</sup> AAA: Açık Anahtar Altyapısı

<sup>2</sup> ROM: Salt okunabilir bellek

yapısının değiştirilmesi gerekmez. Bu nedenle sonradan yüklenebilen veya değiştirilebilen Java uygulamaları eID'de kullanılmamıştır. Bu tercihin en önemli gerekçesi güvenlidir. Sonradan yüklenebilen veya değişken kodlar içeren uygulamalar akıllı kartların EEPROM belleğini kullanır. Bu bellek hem fiziksel ömür hemde saldırılar açısından daha risklidir. Ayrıca yüklenen uygulamaya bağlı olarak zaten sınırlı olan EEPROM<sup>1</sup> alanı daha da azalır. Bu durum, uygulaması da EEPROM'a yüklenen sistemlerde kişisel bilgiler, fotoğraf veya biyometrik veri gibi uygulama verileri için bellek kısıtı oluşturmaktadır. Bu nedenlerden dolayı UKiS'te AKiS gibi değişmez "native" yapıda tasarlanmıştır.

Akıllı kart donanımı, üzerinde bir işletim sistemi ve onu kullanan bir uygulama olmadan hiç bir işlem görmez. Bu nedenle güvenliği sağlanmış bir donanım üzerinde mutlaka bir işletim sistemi ve en az bir uygulama bulunmalıdır. Genellikle ve yukarıda açıklanan nedenlerden, işletim sistemleri akıllı kart yongalarının ROM'unda bulunurlar. Bu sistemlerin yönetimi bir bölümü EEPROM'a yüklenen işletim sistemlerine göre çok daha basittir. Çünkü uygulamanın EEPROM'a yüklenmesi işlemi, uygulamanın sertifikalandırılması adı verilen, güvenlik açısından karmaşık bir sürecin tamamlanması sonrasında gerçekleştirilir. Bu nedenle, AKiS ve UKiS değişmez "native" bir işletim sistemi olarak tasarlanmıştır. Tamamıyla ROM üzerinde bulunmaktadır. Ancak, kullanılacak uygulamaya göre, uygulama verileri EEPROM olarak adlandırılan bellek alanında, AKiS/UKiS Dosya/Bellek yönetim sistemi tarafından, gelişmiş güvenlik önlemleri alınarak saklanmaktadır. Akıllı kart işletim sistemleri dış dünya ile iletişimini iki adet iletişim ucu üzerinden, APDU olarak adlandırılan uygulama protokol veri paketleri aracılığıyla sağlarlar. Bu iletişim protokolü ISO7816-2/3/4 standartlarıyla tanımlanmıştır. EKDS projesinde geliştirilen AKiS/UKiS işletim sistemleri üzerinde çalışan elektronik kimlik uygulaması da akıllı kartlar ve uygulamaları için tanımlı tüm standartları sağlamaktadır.

### 3.1. Akıllı Kart işletim sistemi yazılım yapısı

Akıllı kartlar için geliştirilen donanımlar, üzerinde işletim sistemi ve bir uygulama olmadan tek başına bir anlam taşımadığı belirtilmiştir. Çünkü akıllı kart uygulamaları dış birimler ile etkileşimli çalışmaktadır. Akıllı kart kullanan uygulamalar tarafından gereksinim duyulan birçok işlevin çalıştırılması ve kullandığı EEPROM belleğinin belli bir yapıda düzenlenmesini gerekir. Bu işlemleri akıllı kart işletim sistemi gerçekleştirir.

Bu aşamada akıllı kart işletim sistemi ile akıllı kart uygulamaları arasındaki farkı belirtmekte yarar var. Akıllı kart işletim sistemi akıllı kart içerisindeki sistem verilerinin, uygulama ise dış birimdeki uygulamanın gereksinim duyduğu verilerin yönetimi görevlerini üstlenmektedir.

Elektronik kimlik kartı için AKiS/UKiS işletim sistemi ile birlikte elektronik kimlik uygulamasının da ROM'da olduğunu belirtilmiştir. Elektronik kimlik uygulamasının belli bir sistematiğe çalışabilmesi için AKiS/UKiS işletim sistemi gerekli hiyerarşik yapıyı sağlar. İşletim sistemi yapısında dört temel yazılım bileşeni bulunmaktadır:

1. Bellek yöneticisi (Memory manager)
2. Dosya yöneticisi (File manager)

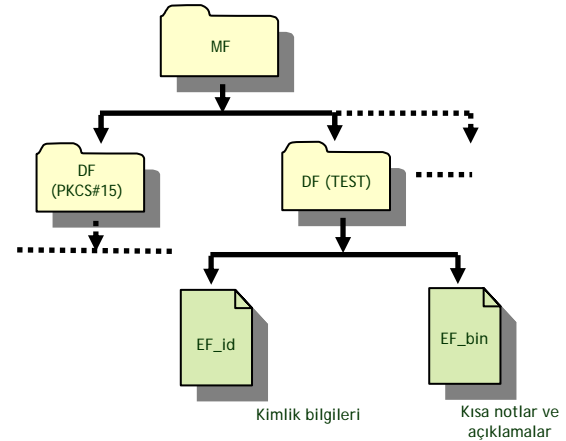
3. Komut yorumlayıcı (Command interpreter)
4. İletişim kotarıcı (Communication handler)

Bunlar haricinde donanıma ait bazı arayüz işlevleri de vardır. Ancak bunlar yonga tasarımcıları tarafından sağlandığından işletim sistemi bileşenlerine dahil edilmemiştir.

#### 3.1.1. Dosya bellek sistemi

AKiS/UKiS işletim sisteminin en önemli özelliği, özgün ve kolay yönetilebilir bir Dosya Bellek Yönetim Sistemine (DBYS) sahip olmasıdır. Bu sistemi kullanan akıllı kartlarda aşağıda bir örneği gösterilen dosya ve dizinlerin oluşturulması, oluşturulan dosyalara değişik evrelerde verilerin kaydedilmesine olanak sağlar. Böylece akıllı karttaki dosya sistemi önkışıselleştirme birimi, kişiselleştirme ise ayrı bir kişiselleştirme birimi tarafından gerçekleştirilebilir.

Aşağıda tipik bir akıllı kart uygulamasına ait dosya yapısı gösterilmektedir. AKiS/UKiS işletim sisteminin yönettiği EEPROM bellekteki dosyalar (EF) ve dizinler (DF) örnekteki gibi mantıksal bir yapıda tutulurlar. Bu dosya ve dizinlere erişim, anahtarlar ve erişim koşulları ile kısıtlanmaktadır.



AKiS/UKiS tabanlı akıllı kartlar üzerine, AAA uygulaması için PKCS#15 standartında veri yapısı, elektronik kimlik kartı uygulaması için de TUBITAK-UEKAE'de oluşturulan EKDS veri yapısı yüklenerek her iki uygulamanın da aynı anda çalışması sağlanabilir.

#### 3.1.2. Komut yorumlayıcı ve iletişimin kotarılması

AKiS/UKiS işletim sisteminde dış uygulamalar APDU olarak adlandırılan komutlar ile gerçekleştirilmektedir. Bu komutların içerisindeki verileri şekil 1'de UART olarak adlandırılan donanım birimi kotarak APDU veri paketini oluşturur ve oluşan APDU veri paketinin hata denetimlerini yaparak gelen komutu komut yorumlayıcıya aktarır. Komut yorumlayıcı da istenen işlemi gerçekleştirmek için program denetimini ilgili uygulamaya geçirerek görevini gerçekleştirmiş olur.

<sup>1</sup> EEPROM: Elektriksel silinebilir bellek

#### 4. Güvenli bir akıllı kart nasıl olmalıdır?

Akıllı kartlara yönelik geliştirilen saldırılara karşı akıllı kart tarafında hem donanımsal hem de yazılımsal önlemler alınmaktadır.

Bu bölümde AKİS/UKİS'in sağlıklı ve güvenilir çalışması için gerekli donanımsal güvenlik önlemleri ve daha sonra da işletim sistemi düzeyinde uygulanan güvenlik önlemleri iki ayrı başlıkta sıralanmıştır.

##### 4.1. Akıllı kart tümdevresindeki güvenlik özellikleri

Proje çalışmasında en önemli isteklerden biri, üretilen akıllı kart tümdevresinin CC EAL 5+ sertifikası alacak şekilde tasarlanmasıydı. Bu sertifika, tümdevrenin en yüksek profilli saldırganın dahi saldırılarına karşı dayanıklı olmasını gerektirmektedir. Tümdevre tasarımında en fazla emek ve zaman güvenlik önlemlerinin geliştirilmesi için harcanmıştır. Çünkü bu kartlar yaygın olarak tüm ülke vatandaşlarının elinde olacak ve çok ucuz olarak elde edilebilecektir. Bundan dolayı elde edilecek donanım gerekli tüm güvenlik gereklerini sağlamak zorundadır.

Tümdevre içerisinde saklanan gizli bilgileri elde etmeyi hedefleyen bir saldırganın yapabileceği saldırılar ve tasarımda bunlara karşı alınan önlemler aşağıda özetlenmiştir.

*4.1.1. Tümdevrenin çalıştığı dış ortam koşullarının izin verilen sınırların dışına çıkartılıp tümdevreye hata yaptırılmaya çalışılması*

Bir saldırgan tümdevrenin çalıştığı ortamın sıcaklığını, tümdevreye verilen besleme geriliminin değerini ya da dışarıdan uygulanan saat işaretinin sıklığını izin verilen çalışma değerleri dışına çıkararak tümdevreye hata yaptırabilir. Tümdevre bu tür saldırıların sezilmesi halinde çalışmasını durduran algılayıcılara sahip olmalıdır.

Sıcaklık algılayıcısı, tümdevrenin çalıştığı ortamın sıcaklığını ölçerek, ortamın izin verilen sıcaklık aralığından yüksek ya da düşük olup olmadığını sezmelidir.

Besleme gerilimi algılayıcıları gerek dışarıdan uygulanan besleme geriliminin gerekse bu gerilimden yaranılarak devre içinde regülatörler tarafından üretilen iç besleme geriliminin izin verilen sınır değerlerden yüksek ya da düşük olması durumunda devrenin çalışmasını durdurmaktadırlar. Saldırganın devreye hata yaptırmak üzere dış besleme gerilimi üzerine bindireceği çentik biçimdeki işaretlerin sezilmesi ve yok edilmesi yine besleme gerilimi algılayıcıları tarafından gerçekleştirilmektedir.

Saldırganın tümdevreye dışarıdan uygulanan saat işaretinin frekansını sınırların üzerine yükselterek tümdevreyi hatalı çalıştırmasına, ya da düşürerek tümdevreyi adım adım çalıştırmasına engel olmak için farklı yöntemler uygulanmıştır. Tümdevre dış saat işareti ile değil, bu işaretten bağımsız olarak kendi içinde ürettiği, saldırganın dışarıdan müdahale edemeyeceği iç saat işareti ile çalışmaktadır. Dış saat işareti yalnızca kart okuyucu ile kart arasındaki haberleşmede kullanılmaktadır. Bununla birlikte, gerek dış saat işaretinin gerekse iç saat işaretinin frekanslarının izin verilen frekans aralığından yüksek ya da düşük olduğunda devrenin çalışmasını durduran frekans algılayıcıları da devrede bulunmaktadır.

##### 4.1.2. Tersine mühendislik saldırıları

Saldırgan tümdevreyi oluşturan tabakaları çeşitli kimyasal ya da fiziksel yöntemlerle adım adım kaldırarak, her tabaka kaldırıldıktan sonra tümdevreyi mikroskop altında inceleyerek tasarım ve güvenlik önlemleri konusunda bilgi edinmeye çalışabilir. Çok emek ve zaman gerektiren ve tümdevreleri tahrip eden bu tür saldırıların yanında, FIB (Focused Ion Beam) türü çok pahalı aygıtlar kullanılarak, tümdevre tahrip edilmeden üzerinde istenilen bölgeler aşındırılıp, istenilen bağlantılar yapılarak, kritik bilgi taşıdığından şüphelenilen hatlar tümdevre yüzeyinde oluşturulan adacıklara bağlanabilir ve bu adacıklara mikroskop altında iğnelerle dokunularak işaretler osiloskopa incelenebilir. Bu tür saldırılara karşı tümdevre tasarımında çeşitli önlemler alınmıştır.

- Tümdevre yüzeyi en üst düzey metal olan metal-5 seviyesinde paralel hatlarla kaplanmıştır. Tümdevre çalışırken aktif kalkan olarak adlandırılan bu hatlarda kısa devre ve açık devre denetimleri yapılmaktadır. Saldırganın tümdevrenin alt düzeylerindeki bağlantılara ulaşmak için bu paralel hatlarda oluşturmak zorunda kalacağı kısa devre ya da açık devreleri tümdevre sezdiği anda çalışmasını durdurmaktadır.
- Saldırganın UKİS'in tutulduğu ROM bloğunu oluşturan tabakaları inceleyip işletim sistemi yazılımı hakkında bilgi edinmesini engellemek amacıyla, işletim sistemi şifrelenerek ROM'a yerleştirilmiştir. Mikroişlemci ROM'dan okuduğu şifreli verileri çözdükten sonra işlemektedir.
- Saldırganın tümdevre serimini incelediğinde blokların yerleri konusunda bilgi sahibi olmasını engellemek amacıyla, tüm sayısal blokların serimleri birbirinin içine geçmiş şekilde karmaşık yapılmıştır. Bloklar arasında veri ve adreslerin taşındığı yollar ayırt edilemez şekilde karışık durumdadır. Buna karşın, ROM, statik bellek, flaş bellek gibi bloklar ise yapıları gereği tümdevre serimi incelendiğinde kolayca ayırt edilebilirler. Dolayısı ile bu tür blokların adres girişleri ile veri giriş ve çıkış yollarının saldırgan tarafından incelenmesi tehlikesi bulunmaktadır. Bu tehlikeyi engellemek için, verilerin ROM'da olduğu gibi statik bellek ve flaş bellek yapılarında da şifreli olarak tutulması sağlanmıştır. Mikroişlemci statik bellek ve flaş belleğe verileri şifreleyerek yazmakta, buradan okuduğu verileri de şifrelerini çözdükten sonra işlemektedir. Mikroişlemci ile kripto blokları arasındaki veri ve adres yolları da, serimde ayırt edilememelerine rağmen yine de şifrelenmişlerdir. Saldırganın bu yollardaki işaretleri izleyebildiği durumda bile haberleşmeyi incelemesi şifreleme nedeniyle mümkün olmayacaktır.

#### 4.1.3. Lazer saldırıları

Saldırgan tümdevreye mikroskop altında lazer atışları yaparak, lazer ışınının üzerine düştüğü bir kütüğün değerini değiştirebilir. Değeri değişen kütüğün mikroişlemcinin ya da kriptoloğunun çalışmasını etkileyen kritik bir kütük olması ve lazer atışının doğru zamanda yapılması durumunda tümdevre gizli bir bilgiyi açığa çıkarabilecek şekilde hatalı çalışabilir.

Bu tehlikeyi gidermek amacıyla, mikroişlemcinin kritik kütükleri, birbirini denetleyecek şekilde ve çift olarak gerçekleştirilmişlerdir. Lazer saldırıları her iki kütüğü aynı anda ve aynı şekilde bozamayacağı için, her zaman aynı değere sahip olması gereken kütüklerin değerleri arasında bir farklılık oluştuğunda bir saldırı olduğu anlaşılmakta ve tümdevre çalışmasını durdurmaktadır. Benzer yaklaşım kriptoloğlarının tasarımında da kullanılmıştır.

Lazer saldırılarına karşı bir diğer önlem de gerek mikroişlemcinin gerekse kriptoloğlarının işlemlerinin çalışma sırasında rasgele duraklatılmasıdır. Böylece saldırıların lazer atışlarının zamanlamasını istediği işlem adımına denk düşürecek şekilde ayarlaması zorlaştırılmaktadır.

#### 4.1.4. Yan kanal analizi

Yan kanal analizi çalışmaları, devrenin çalışması sırasında beslemeden çektiği akımın ya da dışarı yaydığı elektromanyetik yayınının kayıt edilip istatistiksel yöntemlerle analiz edilmesi yoluyla devrede saklı gizli bilgilerin ortaya çıkarılması yaklaşımına dayanır. Bu tür analizlerin ve bunlara karşı önlemlerin geliştirilmesi günümüzde son derece güncel çalışmalardır. UEKAE-YİTAL tümdevre tasarım grubu da 2002-2004 tarihleri arasında AB 6. Çerçeve programı çerçevesinde SCARD – Side Channel Analysis Resistant Design Flow- Yan kanal Analizine Dirençli Tasarım Akışı Geliştirilmesi projesine bu alanda günümüzde öncü durumda olan endüstriyel kuruluş ve üniversitelerle beraber katılmıştır. Bu proje sırasında kazanılan bilgi birikimi ve deneyim geliştirilerek akıllı kart tümdevresinde kriptoloğları tasarımının yan kanal analizine karşı dirençli olarak gerçekleştirilmesinde kullanılmıştır.

Bir kriptoloğ bloğuna şifrelenecek ya da çözülecek veri ile bu veriyi şifreleme ya da çözmeye kullanılacak anahtar değeri giriş olarak verilmektedir. Bir saldırı, kart okuyucu üzerinden algoritmaya girilecek veriyi kontrol edebilir. Ancak anahtar değeri akıllı kartın flaş belleğinde saklıdır ve saldırıların anahtara erişme olanağı yoktur. Saldırının amacı yan kanal analizi yardımıyla anahtar ortaya çıkarmaktır. Yan kanal analizi, kriptoloğ bloğunun çalışması sırasında tümdevrenin beslemeden çektiği akımın algoritmanın işlediği veri ve anahtar değerlerine bağlı olmasına dayanmaktadır. Saldırın bilmediği anahtarla bildiği verilerin bildiği algoritma bloğu tarafından işlenmesi sırasında beslemeden çekilen akımları kaydeder. Farklı giriş verileri için topladığı akım eğrilerini istatistiksel yöntemler kullanarak analiz ederek gizli anahtar bulmaya çalışır. Analizin başarılı olması için kaydedilmesi ve işlenmesi gereken farklı eğri miktarı arttıkça kriptoloğ bloğunun yan kanal analizine karşı direnci artmaktadır.

Geliştirilen akıllı kart tümdevresinin kriptoloğlarının tasarımında yan kanal analizine karşı direnci arttıran çeşitli yöntemler uygulanmıştır. İşlenecek verilerin her seferinde ayrı bir rasgele sayı ile işleme sokulduktan sonra algoritmaya

uygulanması, algoritma çıkışında bir ters işlemle gerçek sonucun elde edilmesine dayanan maskeleyme yöntemi bunlardan biridir. Maskeleyme yöntemi, her seferinde aynı veri işlenebilecek rasgele sayı ile işleme sokulma nedeniyle kriptoloğunun her seferinde beslemeden farklı akım çekmesini sağlamaktadır. Uygulanan diğer yöntemlerden biri de, kriptoloğ bloğunun adımlarının arasına rasgele olarak sahte işlemler yerleştirilmesidir. Sahte işlemler saldırının uyguladığı veriler yerine rasgele verilerin işlenmesine yol açtığı için kaydedilen güç eğrilerinin istatistiksel analizlerini zorlaştırmaktadır.

DES, AES ve RSA kriptoloğlarının her birinin farklı yapılarına bağlı olarak farklı önlemler uygulanmış ve yukarıda sözü edilen tüm güvenlik önlemleri tümdevre alanının büyümesine yol açmıştır.

#### 4.2. Akıllı kart işletim sisteminin güvenlik özellikleri

Sadece donanım düzeyinde güvenlik önlemi alınması yeterli değildir. Yonga üzerinde çalışan işletim sistemi ve uygulamaların da bazı güvenlik önlemleri olması gerekir. AKİS/UKİS işletim sistemlerinin gerçekleştirdiği güvenlik önlemleri;

- Algoritmaların işlem süreleri sabitlenerek yan kanal ve zamanlama analizleri ile gizli bilginin açığa çıkarılması önlenir. Eğer herhangi bir işlemin gerçekleşme süresi gizli bilginin içeriğine bağlı olarak değişiyorsa, bu bilgi güç analizi ile ortaya çıkabilir. Bu nedenle giriş değerleri ne olursa olsun işlem süreleri sabit tutulmalıdır. Bunun için gerekiyorsa algoritmaya rasgele gecikmeler eklenir.
- Güvenlik açısından önemli olan verilere (anahtarlar, PIN, PUK, vs) toplama sınavı konularak verinin bütünlüğü denetlenir. Herhangi bir nedenle bütünlük bozulduğunda akıllı kart kendini korumaya alır.
- Algoritmalarla gerçekleştirilen işlemlerin işleyiş sırası değiştirilerek algoritmanın ne yaptığının saptanması güçleştirilir.
- Algoritmalarla gerçekleştirilen kritik karşılaştırma işlemlerine çifte denetim konulup sonuçlar karşılaştırılarak hatanın önüne geçilebilir.
- Güvenlik açısından önemli verilerin birden fazla kopyası birden fazla formda tutularak (verinin üssü, vs) değiştirilmesi sezişebilir.
- Yan kanal analizlerinde yanlış PIN girilmesi sonucu PIN hata sayacının azaltılma işlemi tespit edilip o sırada güç kesilerek hata sayacının azaltılması engellenebilmektedir. PIN doğrulaması yapılırken PIN'in doğruluğuna bakılmadan sayaç azaltılıp PIN doğru girilirse eski değerine çekilerek bu saldırı önlenir.
- Veri iletişiminin dinlenmesi ve iletilen verinin değiştirilmesi ile ilgili saldırılara karşı akıllı kartlar ve arabirim cihazı arasındaki veri iletişimi güvenli iletişim yöntemi kullanılarak korunabilir. Böylece giden gelen veri araya giren saldırıların tarafından anlaşılabilir. Güvenli iletişim yönteminde kart ve arabirim cihazı karşılıklı olarak bir oturum boyunca anlaşabilecekleri ortak bir simetrik şifreleme anahtarı oluştururlar. Bu

ortak anahtara *oturma anahtarı* denir ve bir oturma boyunca değişmez. Komut içerisinde yer alan veri oluşturulan oturma anahtarı ile şifrelenerek iletilir. Oturma anahtarı oluşturulmasında asimetrik yöntem kullanılması güvenlik açısından tercih edilir.

- DES algoritmasının zayıflığından dolayı veri şifreleme ve deşifreleme için 3DES ve hatta AES algoritmasının kullanılması önerilir.
- PIN ve PUK gibi yüksek güvenlik gerektiren verilere uzunluk sınırlaması getirilerek deneme yanılma yöntemiyle tahminleri güçleştirilir.

Akıllı kartlardan en temel beklenti güvenlidir. Güvenliğin yanı sıra tüm sistemin dayanıklı bir sistem olması gerekir. Akıllı kartların güvenliği için hem akıllı kart donanımının (tümdevresinin) hem de onun üzerinde çalışan işletim sisteminin uyması gereken kuralların bulunduğu vurgulanmıştır. Bu bölümde akıllı kart güvenliğini belirleyen faktörlerin neler olduğu ve, nasıl belirleneceği sorularına cevap vermeye çalışılacaktır. Akıllı kart benzeri şifreleme cihazlarının ne kadar güvenli olduğu üzerinde fikir birliği sağlamak için birçok çalışma yapılmış ve uyulması gereken bazı ölçütler oluşturulmuştur. Bu ölçütler zamanla Ortak Kriterler (Common Criteria, CC) adı ile sınıflandırılmış ve yayınlanmıştır. Bunları sağlayan donanım ve yazılımlara bulunduğu sınıfa göre CC<sup>1</sup> sertifikası verilerek gerçekleştirilen ürünün ne kadar güvenli olduğu ifade edilmiştir. Günümüzde akıllı kart donanım platformu olarak, CC EAL5+ onayı almış yüksek güvenliğe sahip mikrobilgisayarların (örneğin AKiS'in kullandığı SLE66CLX800PE ve P5CD081 tümdevreleri) kullanılması güvenlik için zorunludur. Akıllı kart ürünlerinin CC seviyesi Ortak Kriter Test Merkezlerinde belirlenmektedir. Buralarda uygulanan testlerin sonucuna göre güvenlik seviyesini belirtir CC sertifikası verilmektedir.

CC sertifikası, akıllı kart donanımına verilebileceği gibi üzerinde çalışan işletim sistemine de verilmektedir. Akıllı kart kullanarak geliştirilen uygulamalar için kart donanımı, işletim sistemi ve o işletim sistemi üzerinde çalışan uygulamalar ayrı ayrı sertifikalandırılabilir. Bu sertifikalar değişik güvenlik seviyelerine de sahip olabilir. Burada uygulamanın ve uygulamayı kullananların istediği güvenlik seviyesi önemlidir. Örneğin Türkiye Cumhuriyeti Ulusal Kimlik kartları için bu seviyeler donanım için CC EAL5+, işletim sistemi ve kimlik uygulaması için CC EAL4+ olarak belirlenmiştir.

Güvenliğin yanı sıra akıllı kartların sağlaması gereken diğer kısıtlama da standartlara uyumdur. Akıllı kartların iletişim arabirimi, protokol yapısı ve veri yapıları ile ilgili IEC/ISO 7816 ve ISO 14443 olarak tanımlanan standartlar bulunmaktadır. Bu standartlar bütün olarak, temassız ve temassız donanımsal iletişim arabirim standartını (fiziksel katmanı), veri iletişim protokolleri standartını (veri katmanı), veri şifreleme standartlarını ve veri depolama standartlarını içermektedir. Örneğin ISO7816-2/3/4 akıllı kartların fiziksel dünya ile iletişimini ve APDU olarak adlandırılan uygulama protokolü veri paketlerini tanımlamaktadır.

ISO7816-8, Açık Anahtar Altyapısında (AAA, PKI) kullanılan şifreleme/şifre çözme yöntemleri ile ilgili standartı, ISO7816-9 ise akıllı kart işletim sistemindeki dizin/dosya yapısına ilişkin standartları tanımlamaktadır.

<sup>1</sup> CC: Ortak Kriter

Yukarıda anlatılanlar, akıllı kart kullanımının yaygınlaşmasının güvenliğinin yeterli düzeye çıkması ile olanaklı olduğunu göstermektedir. AKiS ve UKiS işletim sistemli akıllı kartların kullanım alanları TÜBİTAK-UEKAE enstitüsü tarafından geliştirilen iki değişik işletim sistemi olan AKiS (Infineon ve NXP yongaları üzerinde) ve UKiS (TÜBİTAK-UEKAE YITAL yongası üzerinde) olarak adlandırılan iki değişik işletim sistemine sahip akıllı kartlar üzerinde değişik akıllı kart uygulamalarını çalıştırmak olasıdır. Bu uygulamalar akıllı kartın içerisine yüklenecek veri yapılarını belirlerler. Örneğin sayısal imza uygulaması için mutlaka PKCS#15 veri yapısının AKiS veya UKiS işletim sistemine sahip bir akıllı karta yüklenmiş ve daha sonra bu kartı kullanacak kişiye göre kişiselleştirilmiş olması gerekir. Aşağıda AKiS ve UKiS'li akıllı kartlar ile gerçekleştirilebilecek uygulamalar yer almaktadır.

#### 4.3. Elektronik kimlik (eID) uygulaması

Elektronik kimlik uygulaması, akıllı kart tümdevresi içeren bir elektronik kimlik kartının kişinin ülke sertifikası ile doğrulanması ve geçerlenmesi uygulamasıdır. Bu uygulamanın kişiselleştirilmesi ve kullanıma alınması süreci son derece önemlidir. Bu konuda TÜBİTAK-UEKAE enstitüsü oldukça iyi bir deneyime sahiptir. Türkiye Cumhuriyeti Ulusal Kimlik kartı da bu şekilde gerçekleştirilmiş bir uygulamadır.

#### 4.4. Sayısal imza Açık Anahtar Altyapısı (AAA, Public Key Infrastructure PKI) uygulaması

Kişinin ıslak imzadan daha güvenli elektronik imza ile doküman imzalaması veya gelen dokümanın doğru kişiden ve güvenli olarak geldiğinden emin olunması amacıyla geliştirilmiş bir uygulamadır. Bu uygulama sayısal imza uygulaması olarak ta adlandırılır ve imza taşıma aracı olarak AKiS işletim sistemine sahip akıllı kartlar kullanılmaktadır.

#### 4.5. Elektronik pasaport (ePasaport) uygulaması

Gümrüklerde ve ülke giriş/çıkışlarında kağıt pasaport denetimine ek olarak uygulanan elektronik pasaport denetimi uygulamasıdır. İşlemlerin daha güvenli, çok daha hızlı ve kolay yapılmasını sağlamaktadır. Bu uygulamada her ülkenin kendisine ait sertifikası yonga içerisine konulur ve pasaportu denetleyen birimin bunu geçerlemesi istenir. Eğer elektronik sertifika geçerlenir ise pasaport doğru, güvenli ve geçerli bir pasaporttur.

#### 4.6. Sürücü belgesi (eDL) uygulaması

Yakın gelecekte trafikte araç kullanımı için gereksinim duyulacak bu uygulamanın da AKiS/UKiS tabanlı işletim sistemleri üzerinde çalıştırılması son derece kolay olacaktır. Tıpkı elektronik pasaport uygulamasında olduğu gibi sürücü belgesini veren kurum kendi sertifikasını yonga içerisindeki EEPROM bellek alanına yerleştirir ve daha sonra bu sertifika denetlenerek sürücü belgesinin asıl olduğu ve geçerli olduğuna karar verilir.

### 5. Sonuçlar

Bu yazıda öncelikle ulusal akıllı kart işletim sisteminin geliştirilmesi, mimarisi, dosya bellek yönetim sistemi yapısı ve güvenlik önlemleri anlatıldı. Ayrıca AKiS/UKiS işletim



sistemleri üzerinde kořan uygulamalar hakkında da bilgiler verildi. Őurası bir gerçek ki ulusal iřletim sistemine sahip olmak ve özellikle de bunu kendi tasarladığımız bir yonga üzerinde çalıřtırmak önemli bir teknolojik seviyeyi göstermektedir. Daha ayrıntılı bilgi ve örnek yazılımlar ile, Windows® ve Linux® sürücüleri [www.akiskart.com.tr](http://www.akiskart.com.tr) adresinden bilgi alınabilir.

## 6. Teřekkür

Bu makaleye konu olan akıllı kart teknolojisinin geliřtirilmesinde TÜBİTAK-UEKAE'nin YİTAL ve AKİS proje gruplarına ve desteklerini esirgemeyen TÜBİTAK-UEKAE'ye yönetimine teřekkür ederiz.

## 7. Kaynakça

- [1] W. Rankl, W. Effing, Smart Card Handbook, Giesecke & Devrient Gmbh, Munich, Germany, 2003.
- [2] K. E. Mayes, K. Markantonakis, Smart Cards, Tokens, and Applications, University of London, UK, 2008.
- [3] S. Mangard, E. Oswald, T. Popp, Power Analysis Attacks, Graz University of Technology Graz, Austria, 2007.
- [4] M. Bařak, Bilgi Güvenliđi ve Akıllı Kartlar, Ađ ve Bilgi Güvenliđi Sempozyumu, 2008 Girne, KKTC.
- [5] M. Bařak, AKİS, Akıllı Kart İřletim Sistemi, TÜBİTAK-UEKAE Dergisi, 2009-2010.



# CAN Protokol Tabanlı Bir Tren Simülatörünün Gerçeklenmesi

Fatih Yardım<sup>1</sup>, Oğuzhan Urhan<sup>2</sup>

<sup>1</sup>Tübitak Bilgem Bilişim Teknolojileri Enstitüsü  
fatih.yardim@bte.tubitak.gov.tr

<sup>2</sup> Elektronik ve Haberleşme Mühendisliği Bölümü  
Mühendislik Fakültesi, Kocaeli Üniversitesi  
urhano@kocaeli.edu.tr

## Özetçe

*Bu çalışmada bir tren simülatörüne ait donanım birimlerinin haberleşmesinde kullanılması düşünülen ve günümüzde gömülü ağ sistemlerinde yaygın kullanılan protokollerden biri olan CAN (Controller Area Network) veri yolu protokolünden faydalanılması amaçlanmıştır. Öncelikle, neden CAN protokolünün tercih edildiği ve diğer haberleşme protokollerine göre avantajları vurgulanmaktadır. Çalışmanın amacını gerçekleştirmek için insan döngü (man in-the-loop) simülatörü tipinde bir tren simülatörünün temel davranışlarını gösteren bir simülatör bilgisayarı ve simülatör donanımına ait bazı birimleri içine alan bir model sistem tasarlanıp gerçekleştirilmiştir. Bu sistem üzerinde CAN protokolünün uygunluğunu doğrulamak amacıyla bazı testler gerçekleştirilmiştir. Bu testlere ait deneysel sonuçlar açıkça göstermektedir ki bu sistemde CAN protokolü düzgün çalışmakta ve donanım birimlerinin haberleşmesi için uygunluğu onaylanmaktadır.*

## 1. Giriş

Daha nitelikli makinist yetiştirilmesini amaçlayan tren simülatörleri son zamanlarda bilgi teknolojilerinin gelişmesiyle büyük bir ivme kazandı. Bugün artık birçok lokomotif tipine ait simülatörleri görmek mümkündür. Öğrenciler gerçek sistemle tanıştırmadan önce simülatör kullanarak tecrübe kazanırlar. Böylelikle gerçek sistemde meydana gelebilecek ciddi kaza ihtimalleri de azaltılmış olur.

Tren simülatörleri eğitim amaçlı kullanıldığından kullanıcı müdahalesine izin verecek şekilde tasarlanmıştır. Kullanıcıdan aldığı buton, şalter, kol gibi donanım elemanların konum değişimi bilgisi ve simülasyon boyunca kullanılacak senaryodan gelen bilgilere göre trenin hızını ve trenin çalışmasına yönelik diğer nicelikleri günceller ve bunların sonuçlarını uyarı lambaları ve göstergelerle kullanıcıya bildirir. Tren simülatörleri bu yönüyle donanım döngü simülatörlerinden [1-8] farklı olarak insan döngü simülatörü [9-10] niteliğindedir.

Son zamanlarda hem insan döngü hem de donanım döngü simülatörleri ile yapılan çalışmalarda donanım birimlerinin birbirleriyle haberleşmesi CAN protokolüyle [1-10] sağlanmaktadır. CAN protokolünü bu tür haberleşmelerde avantajlı kılan en büyük özelliği CSMA/CD (Taşıyıcı sezme, çoklu erişim ve çarpışma algılama) özelliğidir. Alternatif

olabilecek diğer protokollerle karşılaştırıldığında avantaj ve dezavantajları Tablo 1'de gösterilmektedir.

Tablo 1: Haberleşme protokolleri özellikleri

Protokol	Hız (Mbps)	Çarpışma algılama	Gürültü eleme	Yazılım basitliği
RS485	1	-	+	+
CAN	1	+	+	+
SPI	10	+	-	+
I2C	3.4	+	-	+
Ethernet	100	+	+	-

Yukarıdaki tabloya göre RS 485 ve CAN gürültüye karşı bağımsızlık sağladığından SPI ve I2C'ye oranla daha uzak mesafelere iletim yapmaktadır. Ethernet ise protokol kümesi karmaşıktır ve mikrodenetleyici belleğinde çok yer kaplar. RS 485 ise çarpışma algılama mekanizması olmadığından bir ağ yöneticisine ihtiyaç duyar ve bu da etkin haberleşme hızının düşmesine yol açar.

CAN protokolü daha uzak mesafelere iletim yapabilme, yazılım basitliği ve de çarpışma algılama özelliklerinden dolayı diğer protokollere oranla ön plana çıkmaktadır.

## 2. Tren ve Simülatör Kavramları

Simülatör donanımı kol, buton, anahtar, sigorta gibi çeşitli elemanlardan oluşmaktadır. Bu çalışmada bu elemanlardan bazıları gerçeğiyle benzer veya aynı, bazıları ise sanal olarak gerçekleştirilmektedir.

Gerçeğiyle aynı veya benzer görünümde [11-12] gerçekleştirilen elemanlar trenin hattan elektrik enerjisini almasını sağlayan pantografi kaldıran pantograf anahtarı, elektrik enerjisini elektrik sistemine ileten disjonktör tutturana disjonktör anahtarı, trenin hareketini sağlayan 14 konumlu hızlandırma kolu, elektriksiz frenlemeyi sağlayan 10 konumlu dinamik fren kolu, hava freni yapmayı sağlayan 5 konumlu makinist musluğu, acil durumlarda durmayı sağlayan imdat butonu, makinistin görev başında olduğunu bildirmesini sağlayan toman pedalı ve elektronik kontrol devrelerinin enerjilenmesini sağlayan elektronik şalterden oluşur. Bunun yanında hat voltajını, cer motorlarının (trenin hareketini sağlayan elektrik motorları) çektiği akımı, batarya voltajını ve tren hızını gösteren 4 adet gösterge ile disjonktörün tutup

tutmadığını gösteren DJ açık, cer motor gruplarının devrede olup olmadığını gösteren Hızlı Devre Kesici 1 ve 2 lambaları, arıza durumunda flaş yapan arıza lambası ve elektronik kontrol devrelerinin iptal olması durumunda yanan elektronik kontrol lambası da yer almaktadır.

Sanal olarak gerçekleştirilen elemanlar [11-12] ise bataryayı koruyan, batarya voltajından faydalanmayı ve bataryanın şarj olmasını sağlayan batarya şarj girişi, şarj iptal ve batarya şalterleri ile 1. grup cer motorlarını devre dışı bırakan cer motor grup 1 iptal şalteri ve trenin servise hazırlarken devreye alınması gereken işletme anahtarıdır.

### 2.1. Trenin Çalışması [11-12]

Trenin elektrik sisteminin enerjilendirilmesi için batarya voltajı en az 95V olmalıdır. Öncelikle batarya şalteri ve işletme anahtarı devreye konulur. Daha sonra pantograf anahtarı kaldırılarak pantograf elektrik şebekesine bağlanır. Disjonktör anahtarı kaldırılarak disjonktörün tutturulması sağlanır ve trenin elektrik sistemi enerjilenir.

Treni hareket ettirmek için hızlandırma kolu kapalı konumdan herhangi bir hız kademesine getirilerek tren hareket ettirilmiş olur. Tren yavaşlatılmak veya durdurulmak istendiğinde dinamik fren kolu ile elektrikli fren, makinist musluğu ile hava freni yapılabilir.

Ayrıca makinistin ölüm, bayılma, rahatsızlanma gibi durumlarına karşı treni seri fren yaparak treni durduran totman sistemi makinist tarafından her 15 saniye içinde totman pedalı ile tetiklenmelidir. Bu durumda tren hareketine devam edebilecektir.

### 2.2. Simülasyon Yapısı

Tren simülasyonuna ait birimler genellikle yıldız topolojisinde bir ağ oluştururlar ve aradaki iletişim TCP/IP protokolüyle sağlanır.

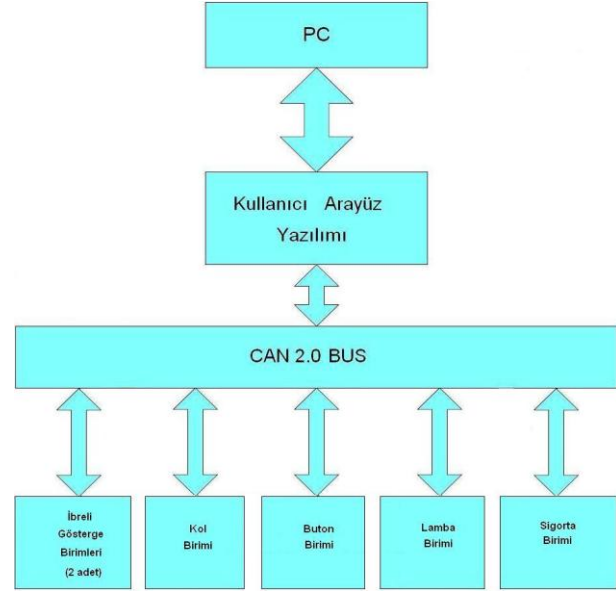
Tren simülasyonuna ait ağ içinde haberleşme trafiğini yöneten bir **benzetim motoru** bulunur. Bu birim diğer birimlerden gelen bağlantı isteklerini kabul eder, gönderilen mesajları alır ve ilgili birimlere gönderir. Simülasyon boyunca gerçekleşecek senaryo **senaryo editörü** üzerinde eğitmen tarafından hazırlanır. Eğitmen simülasyonun herhangi bir anında senaryo akışını değiştirebilir. Lokomotif içinde bulunan donanımlara ait bilgiler **kumanda masası ekranında** gösterilir. **Dinamik model** tren aksamında bulunan donanımlara ait akım, basınç, hız, ivme gibi nicelikleri geliştirilen fiziksel modeller yardımıyla hesaplar. **Donanım birimi** ise donanım elemanların konum değişimini dinamik modele bildirerek güncellenen niceliğe göre göstergeleri çalıştırır ve senaryo editöründen gelen arıza durumuna göre ilgili uyarı lambasını günceller. **Görsel sistem** senaryo editöründen ve dinamik modelden gelen verilere göre simülasyon esnasındaki animasyonu gerçekleştirir. **Ses sistemi** ise ortamdaki seslerin durumuna göre ses efektlerini çıkarır.

## 3. Model Sistem

Bu çalışmada, bir tren simülasyonunda kullanılması düşünülen CAN veri yolu protokolünün uygunluğunu incelemek amacıyla simülasyon ortamına ait bazı davranışları gerçekleştiren model

bir sistem tasarlanmış ve gerçekleştirilmiştir. Tasarlanan bu sistemin blok şeması Şekil 1'de gösterilmektedir.

Bu çalışmada dinamik model, senaryo editörü, donanım birimine ait kabin arıza ekranı ve merkezi donanım bilgisayarı tarafından yürütülen görevler bir simülasyon ortamı tarafından gerçekleştirilir. Bu simülasyon ortamı bilgisayarı kullanıcının simülasyonu başlatmasını sağlayan, senaryo parametrelerini ilklendiren, arızaları oluşturan ve kaldıran, trenin donanım elemanlarının durumunu gösteren birimdir.



Şekil 1: Sistem blok şeması

Donanım birimi ise giriş ve çıkış birimlerini kontrol eden 6 adet gömülü donanım biriminden meydana gelmektedir. Simülasyon ortamı bilgisayarı ve bu gömülü donanım birimleri arasındaki haberleşme CAN veri yolu protokolüyle sağlanmaktadır. Bununla ilgili ayrıntılı bilgi devam eden bölümlerde verilmektedir.

## 4. Donanım Tasarımı

Bu çalışma için tasarlanan sisteme ait donanım PC üzerinde çalışan bir grafik kullanıcı arayüzü tarafından kontrol edilir ve PC ile bir USB-CAN dönüştürücü üzerinden haberleşir. Donanım birimi (Şekil 2) aşağıdaki işlevsel birimlerden oluşmaktadır.

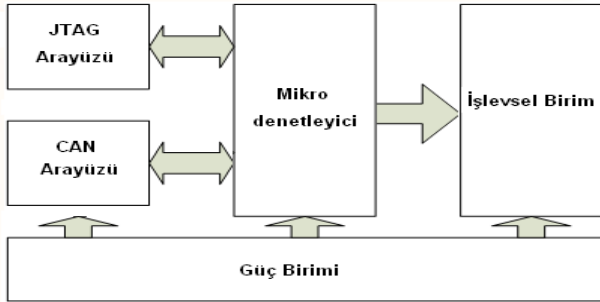
- İbrelili Göstergeler Birimleri
- Kol Birimi
- Lamba Birimi
- Sigorta Birimi
- Buton Birimi

Bu birimler tarafından gerek duyulan donanım güç ünitesi, denetleyici ünitesi ve işlevsel donanım ünitesi olmak üzere 3 bölüme ayrılır. Güç ve denetleyici üniteleri tüm işlevsel birimler için ortaktır.

#### 4.1. Güç ve Denetleyici Üniteleri Tasarımı

Güç ünitesi biri 12V besleme voltajını 5V'a diğeri 5V gerilimi 3.3V'a düşüren iki ayrı DC/DC çeviriciden meydana gelmektedir. 3.3V mikrodnetleyici ve haberleşme devrelerinin beslemesini sağlarken, 5V ve 12V gerilim seviyeleri işlevsel donanım ünitelerini beslemede kullanılır.

Denetleyici ünitesi ise giriş ve çıkış elemanlarını kontrol eden mikrodnetleyici, tüm işlevsel birimler ve simülâtör bilgisayarı arasında haberleşmeyi sağlayan CAN arayüzü, yükleme ve hata ayıklama amaçlı kullanılan JTAG arayüzü olmak üzere 3 bölümden oluşmaktadır. Bu çalışmada ST Microelectronics tarafından üretilen STM32F103VB mikrodnetleyicileri kullanılmıştır.



Şekil 2: Gömülü donanım birimi blok şeması

#### 4.2. İşlevsel Donanım Ünitesi Tasarımı

**İbrelî gösterge birimleri** her biri 2 adet göstergeden oluşan 2 birimdir. Bu göstergeler hat voltajı, cer motor grup 1 akımı, batarya voltajı ve tren hızı niceliklerini gösterir. Her bir gösterge bir DC motor tarafından sürülür ve 2 DC motor 1 adet sürücü entegresi tarafından kontrol edilir. Motor milinin pozisyonu 1 milisaniye periyotlarla ölçülür ve kaydedilir. Simülâtör bilgisayarıdan gönderilen değer ölçülen değer ile kıyaslanır. Büyük ise saat yönünde küçük ise saat yönünün tersi yönünde ibre döndürülür.

**Kol birimi** makinistin hareketi sağlayan elemanları kontrol etmesine imkan tanır. Kontrolü sağlanan elemanlar hızlandırma kolu, dinamik fren kolu ve makinist musluğuğudur. Kolun pozisyonu 1 milisaniye periyotlarla ölçülür ve değışim olmuşa simülâtör bilgisayarına gönderilir.

**Lamba birimi** makinisti uyarılar ve arızalar konusunda bilgilendirir. Her bir lamba birer LED ile temsil edilir. Kontrolü sağlanan lambalar DJ Açık, Arıza, Hızlı Devre Kesici 1-2 ve Elektronik Kontrol lambalarıdır. Lambanın durumu simülâtör bilgisayarıdan mesaj ile alınır ve bunun sonucuna göre yakılır, söndürülür veya flaş yaptırılır.

**Sigorta birimi** elektronik kontrol devrelerine enerji akışını sağlayan bir elektronik şalterin kontrolünü gerçekleştirir. Bu şalter bir röle ve röle sürücü devresi ile kontrol edilir. Şalter aşırı akıma maruz kaldığında atar. Şalterin durumu 1 milisaniye periyotlarla okunur ve herhangi bir değışiklik varsa simülâtör bilgisayarına gönderilir. Simülâtör bilgisayarıdan herhangi bir arıza durumuna karşılık şalter atırma mesajı gelmiş ise şalter atılır.

**Buton birimi** makinistin trenin kontrolünü gerçekleştirmesine imkân sağlar. Pantograf anahtarı, disjonktör anahtarı, imdat butonu ve totman pedalının kontrolünü gerçekleştirir. 1 milisaniye aralıklarla butonların

durumu okunur. Herhangi birinin durumunun değışmesinde durum simülâtör bilgisayarına gönderilir.

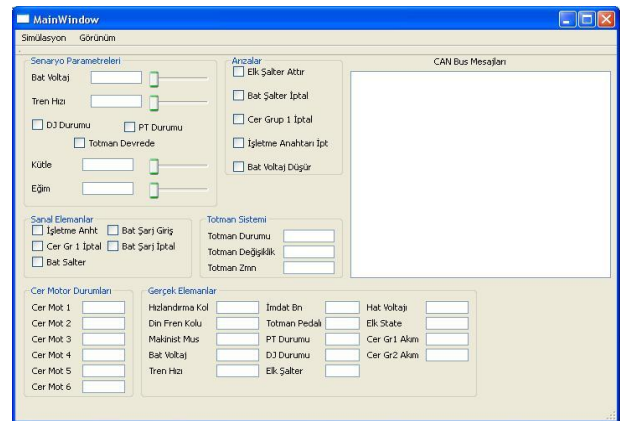
### 5. Yazılım Tasarımı

Bu çalışmada geliştirilen yazılım grafik kullanıcı arayüz (GUI) yazılımı, CAN sürücü arayüz yazılımı ve modelleme yazılımı olmak üzere 3 bölümden oluşur.

Grafik kullanıcı arayüz yazılımı grafik kullanıcı arayüzünde (Şekil 3) bulunan aşağıdaki elemanların kontrolünü gerçekleştirir.

- Senaryo parametreleri
- Cer motor durumları
- Arızalar
- Sanal elemanlar
- Gerçek elemanlar
- Totman sistemi
- CAN mesajları

Batarya voltajı, tren ilk hızı, pantograf ve disjonktör durumları, totman aktif, trenin kütlesi ve eğim senaryo tarafından ilklendirilen parametrelerdir. **Arızalar** bölümünde elektronik şalter, batarya şalteri, batarya voltajı, işletme anahtarı ve cer motor grup 1 iptal arızaları yer alır ve bu arızalar eğitmen tarafından simülasyon esnasında verilir ve bu arızalar kaldırılabilir. **Sanal elemanlar** batarya şalteri, batarya şarj iptal ve batarya şarj girişi şalterleri, işletme anahtarı ve cer motor grup 1 iptal şalterinden oluşur. Bu elemanların durumları simülasyon esnasında değıştirilebilir. **Gerçek elemanlar**, hızlandırma kolu, dinamik fren kolu, makinist musluğuğ, imdat butonu, totman pedalı, pantograf, disjonktör, elektronik şalter ve elektrik sistem durumları ile hat voltajı, batarya voltajı, tren hızı, cer motor grup 1 ve 2 akımlarını gösterir. **Cer motorları durumu** cer motorlarının devrede olup olmadığını, **totman sistemi** totman durumları arasındaki geçişleri ve toman zamanlayıcısını, **CAN mesajları** ise mesaj hatında iletilen tüm mesajları gösterir.



Şekil 3: Grafik kullanıcı arayüzü

**CAN sürücü arayüz yazılımı** uygulama programlama arayüz (API) yazılımı ve haberleşme uygulama yazılımı olmak

üzere 2 bölümden oluşur. API yazılımı kullanılan USB-CAN dönüştürücüye özeldir. Bu çalışmada Peak Systems firmasının geliştirmiş olduğu USB-CAN dönüştürücüye ait yazılım kullanılmaktadır. Bu yazılım kapalı kaynak koda sahip olup üretici tarafından hazırlanmış dll uzantılı bir dinamik bağ kütüphanesi dosyasından oluşur. Bu koda erişim yine üretici tarafından hazırlanan bir başlık dosyası ile sağlanır. API yazılımı düğüm iklendirme, sonlandırma, mesaj alma, gönderme ve filtreleme gibi işlevleri içerir. Haberleşme uygulama yazılımı ise API yazılımına ait işlevlere erişim sağlayarak düğüm oluşturma, mesaj okuma, yazma işlemlerini gerçekleştirir. Ayrıca donanımdan alınan mesajları ve kullanıcı arayüzünde yapılan değişiklikleri işleyerek model yazılımının kullanacağı girdileri elde eder. Böylelikle donanım, grafik kullanıcı arayüz yazılımı ve model yazılımları arasında birer köprü oluşturur. Haberleşme hızı 1 Mbps değerine ayarlanır ve 10 ms’de bir mesaj tamponundaki mesajlar okunur.

**Modelleme yazılımı** donanımdan ve kullanıcı arayüzünden gelen verilere göre sistemin fiziksel modelini gerçekleyen, fiziksel parametrelerini hesaplayarak güncelleyen ve her 100 ms’de bir çalışan aşağıdaki modellerden oluşur.

- Elektriksel sistem modeli
- Cer motorları modeli
- Tren hızı hesaplama modeli
- Cer akımı hesaplama modeli
- Batarya şarj modeli
- Lambalar modeli
- Totman modeli

**Elektriksel sistem modeli** elektrik sistem değişkenini günceller, pantograf ve disjonktörün durumlarını belirler. Ayrıca hat voltajını günceller ve değişim varsa bunu mesaj ile ibrelî gösterge birimine gönderir.

**Cer motorları modeli** cer motorlarının arızalar ve makinist kabinindeki elemanların durumuna göre durumunu günceller.

**Tren hızı hesaplama modeli**, tren hızını periyodik olarak günceller. Tren hızını hesaplamak için çekiş veya dinamik fren kuvveti, varsa hava freni kuvveti, rampa direnci ve aerodinamik direnç kuvvetlerini hesaplayarak ve trenin kütlesini kullanarak ivmeyi ve ivmeyi kullanarak da son hızı hesaplar.

**Cer akımı hesaplama modeli** cer motorları tarafından çekilen çekiş veya dinamik fren akımını iki grup motorları için de hesaplar ve grup 1 motorlarının akımını ibrelî gösterge birimine gönderir.

**Batarya şarj modeli** elektrik sistem durumunu, batarya şalteri, batarya şarj girişi ve batarya şarj iptal şalterlerinin durumlarını kullanarak batarya voltajını günceller. Batarya voltajında değişiklik varsa ibrelî gösterge birimine gönderir.

**Lambalar modeli** lamba durumlarını arıza durumlarına ve elektrik sistem durumuna göre günceller. Simülâtör bilgisayarından gönderilen mesaj ile lambalar yakılır, söndürülür veya flaş yaptırılır.

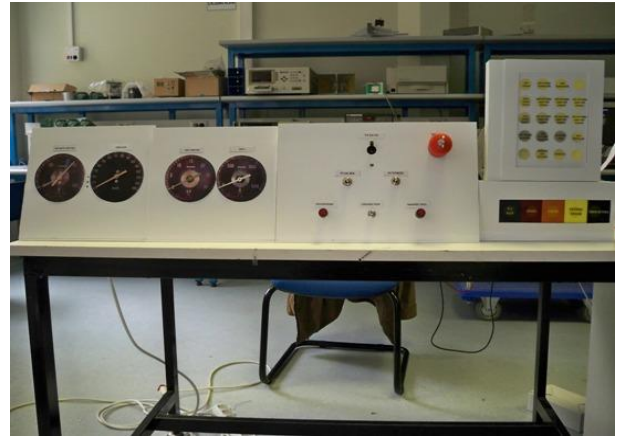
**Totman modeli** totman sisteminin durumunu günceller. Dinamik fren kolu, hızlandırma kolu, makinist musluğu ve totman pedalının durumlarına göre değişiklikleri kontrol eder ve totman durumları arasındaki geçişi belirler. Sistem totmana kaçtığında tren seri frene geçer.

## 6. Deney Sonuçları

CAN haberleşme protokolünün performansını incelemek amacıyla Şekil 4’te gösterilen deney düzeneğiyle “Trenin enerjilenmesi ve çalışması”, “Totman sistemi” ve “Arızalar” adı altında 3 tip test gerçekleştirilmiştir.

Trenin enerjilenmesi ve çalışmasını konu alan testler simülâtör donanımının davranışı ile ilgilidir. Testlerin başlatılması için ilk önce senaryo parametreleri iklendirilir.

Parametreler senaryo parametreleri kısmından batarya voltajı 110V, tren hızı 0, trenin kütlesi 120 ton, eğim 0, pantograf inik, disjonktör çözükle ve totman devrede değil olarak iklendirilir ve simülasyon başlatılır. Daha sonra pantograf ve disjonktör devreye alınarak tren elektrik sistemi enerjilenir. Bu durumda iken hızlandırma kolu açılıp kol kademesi 1’den başlayarak 7 seviyesine getirildiğinde tren hızı 78 km/h değerine kadar artar, bu değerde sabitlenir. Bu durumda iken batarya voltajı, tren hızı, hat voltajı ve cer motorları grup 1 akımı Şekil 5’de gösterilmektedir.



Şekil 4: Simülâtör donanımı (önden görünüm)

Sabit hıza ulaştıktan sonra eğim 3°’ye getirilir. Bu durumda eğimden kaynaklanan yavaşlatıcı kuvvet, trenin çekiş kuvvetinden büyük olacağından tren yavaşlamaya geçer ve 65 km/h hız (Şekil 6) değerinde kuvvetler birbirini dengelediğinden tren hızı bu değerde sabitlenir. Bu durumda iken trenin kütlesi 2 katına 240 tona çıkarıldığında kütle arttığından eğimin tren hızına etkisi daha fazla olacaktır ve tren tekrar yavaşlamaya geçecektir. Hız değeri 51 km/h değerine (Şekil 7) geldiğinde kuvvetler dengelenecek ve hız sabit kalacaktır. Her iki durumda da trene uygulanan çekiş kuvveti artacağından cer motorları da Şekil 6 ve 7’de görüldüğü gibi daha fazla akım çekecektir.

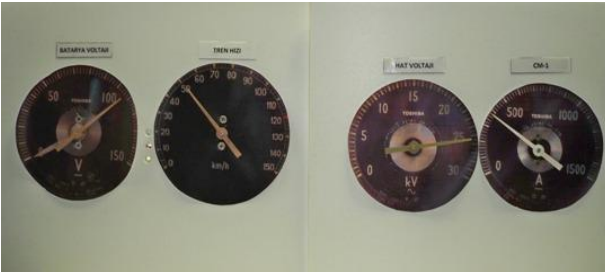


Şekil 5: Hızlandırma kolu 7. kademe gösterge durumları

Kütle tekrar 120 tona ve eğim 0'a getirildikten sonra Şekil 5'teki durum tekrar görülecektir. Bu durumda makinist musluğu ile fren yapıldığında Şekil 8'de görüldüğü gibi hız 3,5 km/h değerine kadar düşer. Daha sonra makinist musluğu ile yapılan fren çözülür ve Şekil 5'teki durum tekrar görülür. Bu durumda hızlandırma kolu kapatılıp dinamik fren yapıldığındaki etki Şekil 9'da gösterilmektedir.



Şekil 6: Eğim 3° iken 7. kademe gösterge durumları



Şekil 7: Eğim 3° ve kütle 240 ton iken 7. kademe gösterge durumları



Şekil 8: Hızlandırma kolu 7. kademede iken fren durumu

Hız 5 km/h seviyesine geldikten sonra lokomotifin elektronik kontrolünden dolayı dinamik fren devreden çıkar. Tren kendisine etki eden yavaşlatıcı kuvvetler yardımıyla yavaşlamaya devam eder



Şekil 9: Dinamik fren kullanıldığında gösterge durumları

Totman sistemi ile ilgili testler totman durumu arasındaki geçişleri inceler. Bu testler için simülasyon ilkendirme değerleriyle yeniden başlatılır. Bu defa senaryo parametrelerinden totman devrede seçeneği işaretlenir. Şekil 10 ve 11 imdat butonu çekilip sistem totmana kaçtığında ibrelü göstergelerin ve kullanıcı arayüzünün durumunu gösterir.

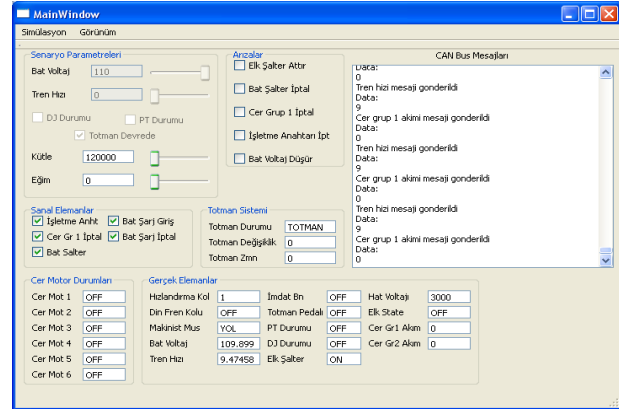
Son test ise elektronik şalter arızası ile ilgilidir. Bu arıza grafik kullanıcı arayüzünden eğitmen tarafından verilir. Bu arıza gerçekleştiğinde simülatörde aşağıdaki değişiklikler olur.

- Elektronik şalter atar.
- Disjonktör çözer.
- Cer motorları durur.
- DJ açık, elektronik kontrol ve hızlı devre kesici 1-2 lambaları yanar.
- Arıza lambası flaş yapar.

Bu arızaya ilişkin ibrelü gösterge durumları, lamba durumları, elektronik şalter durumu ve grafik kullanıcı arayüzü sırasıyla Şekil 12, 13, 14 ve 15'de gösterilmektedir.



Şekil 10: Totman durumunda iken gösterge durumları



Şekil 11: Totman durumunda iken kullanıcı arayüzü



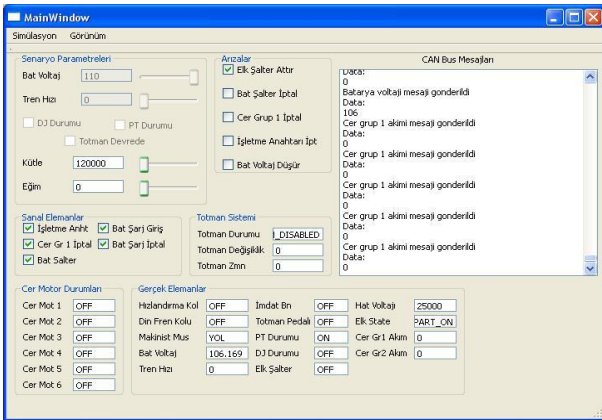
Şekil 12: Elektronik şalter arızasında ibrelü gösterge durumları



Şekil 13: Elektronik şalter arızasında lamba durumları



Şekil 14: Elektronik şalter arızasında şalter durumu



Şekil 15: Elektronik şalter arızasında kullanıcı arayüzü

## 7. Sonuç

Bu çalışmada bir tren simülöründe donanım birimlerinin haberleşmesinde kullanılması düşünülen CAN veri yolu protokolünün performansı ve sistem için uygunluğu incelenmiştir. Yapılan deneysel çalışmada elde edilen sonuçlar CAN veri yolu protokolünün model sistem için kusursuz çalıştığını göstermektedir.

Bu çalışma için 1. bölümde Tablo 1'de verilen diğer protokoller de tercih edilebilirdi. Fakat gerçekleştirilen sistemin tren simülörünün yalnızca bir bölümü olduğu göz önüne alındığında sistem büyüdüğünde ve kabiliyetleri artırıldığında CAN protokolünün diğer protokollere olan avantajı daha belirginleşmektedir. RS485'e göre ağ yöneticisine ihtiyaç duymayıp daha hızlı haberleşme sağlaması, SPI ve I2C'ye göre gürültüye karşı daha bağımsız olup uzun mesafelere iletim yapabilmesi ve Ethernet protokolüne göre maliyeti ve yazılım basitliği açısından en uygun protokol olduğu görülmektedir.

## 8. Kaynaklar

- [1] K. Yonemoto, T. Shidooka, K. Okuda, "Development and Flight Test of Winged Rocket", *27th International Symposium on Space Technology and Science*, Tsukuba City/Japan, (2009).
- [2] S. Ateş, I. Bayezit, G. Inalhan, "Design and Hardware-in-the-Loop Integration of a UAV Microavionics System in a Manned-Unmanned Joint Airspace Flight Network Simulator", *Journal of Intelligent and Robotic Systems*, volume 54, 359-386 (2008).
- [3] L. Gangyan, X. Jun, "An Information Acquisition Method of City Bus Integrated Control Network," *csie*, vol. 3, 722-725, 2009 WRI World Congress on Computer Science and Information Engineering, (2009).
- [4] S. L. Woodruff, L. Qi, M. J. Sloderbeck, "Hardware-in-the-Loop Experiments on the Use of Propulsion Motors to Reduce Pulse-Load System Disturbances", *2007 IEEE Electric Ship Technologies Symposium*, (2007).
- [5] J. Feng, H. Zhong, G. Ao, J. Wang, H. Tang, X. Mao, B. Zhuo, "Principles and Application of the Real-Time Hardware-in-the-Loop Simulation Platform Based On Multi-Thread and CAN", *IEEE International Symposium on Industrial Electronics*, 2225-2230 (2008).
- [6] Cover L. Kis, G. Regula, B. Lantos, "Design and hardware-in-the-loop test of the embedded control system of an indoor quadrotor helicopter", *2008 International Workshop on Intelligent Solutions in Embedded Systems*, 1-10 (2008).
- [7] M. Short, M. J. Pont, "Hardware in the Loop Simulation of Embedded Automotive Control Systems", *2005 IEEE Intelligent Transportation Systems Proceedings*, 426-431 (2005).
- [8] G. Tang, T. Zhang, X. Liu, W. Liu, T. Mei, "Vehicle hardware-in-the-loop simulation facility for driverless vehicle", *2nd International Conference on Power Electronics and Intelligent Transportation System*, (2009).
- [9] G. Liu, Q. Huang, B. Zhang, J. Han, E. Xu, "A Study on Quick Cockpit for Flight Simulators Prototype Systems," *icmtma*, vol. 2, 28-31 2009 International Conference on Measuring Technology and Mechatronics Automation, (2009).
- [10] Z. Lei, J. Hongzhou, L. Hongren, "PC Based High Quality and Low Cost Flight Simulator", *Proceedings of the IEEE International Conference on Automation and Logistics*, 1017-1022 (2005).
- [11] H. Soltekin, "E43000 Tipi Lokomotif Eğitimi Servise Hazırlama ve Loko Kullanma", *Eskişehir Eğitim Merkezi Müdürlüğü*, (Eylül 2005).
- [12] H. Soltekin, "E43000 Tipi Lokomotif Eğitimi Elektrik Bilgisi", *Eskişehir Eğitim Merkezi Müdürlüğü*, (Eylül 2005).



## Radyo Frekansıyla Konteyner Takip Sistemi

Muhammet Alican Güncan, Ali Yavuz Kahveci, Muhammet Erkoç, Arda Yurdakul

Bilgisayar Mühendisliği Bölümü, Boğaziçi Üniversitesi, İstanbul.

{ muhammet.guncan, yavuz.kahveci, muhammet.erkoc, yurdakul }@boun.edu.tr

### Özetçe

Hızla gelişen deniz taşımacılığı, limanların alanlarının artmasına yol açmış ve bu durum beraberinde önemli bir sorun ortaya çıkartmıştır: limana giriş yapan konteynerlerin bulunması. Radyo Frekansıyla Konteyner Takip Sistemi konteynerlerin yerini, herhangi bir ön bilgi olmaksızın, her bir konteynerin eşsiz kimlik numaraları aracılığıyla tespit etmek için kullanılan bir sistemdir. Sistem temel olarak iki alt gömülü sistemden oluşmaktadır: el cihazı ve vericiler. Vericiler konteynerlerin üzerine yerleştirilir ve el cihazı bu vericilerden sinyal alarak onların yerlerini tespit etmek ve elde edilen sonucu LCD ekran aracılığıyla kullanıcıya bildirmek için kullanılır.

### 1. Giriş

Deniz aşırı ticaretin artması ile deniz taşımacılığı da gelişmiş, artan ihtiyacı karşılamak için limanlar da gün geçtikçe büyümeye başlamıştır. Bunun bir sonucu olarak, limanlarda giriş yapan bir konteyneri bulmak bazen uzun zaman almaktadır. Bu durum maliyetleri arttırmaktadır. Bu konuyla ilgili olarak bir denizcilik şirketinde konteyner kontrol müdürü Mehmet Hacı İbrahim[6] ile yapılan görüşmelerde limanlar, konteynerler ve konteyner gemileri hakkında gerekli bilgiler alınmış ve bu bilgiler doğrultusunda, problemin çözümü için bir gömülü sistem tasarlanmasına karar verilmiştir.

### 2. Alternatif Çözümler

Belirtilen problemin çözümü için şu an, konteynerleri limanlara girdikleri andan itibaren kayıt altına alıp, liman içerisindeki hareketlerini izleyen bir sistem kullanılmaktadır. Bu sistemlerde, limana giren bir konteynerin kaydı yapılır ve bu kayıta konteynerin depolanması gereken yer de tutulur. Bu sistemlerin en büyük handikapı bu noktada ortaya çıkmaktadır. Eğer bir konteyner sistemde belirtilen yere yerleştirilmemişse ya da konteynerin yerleştirilmesi gereken yer sisteme yanlış olarak girilmişse, ortaya büyük bir sorun çıkmaktadır: limana giriş yapmış ancak yeri bilinmeyen bir konteyner. Konteyner depolama alanlarının büyüklüğü göz önüne alındığında, örneğin İstanbul Haydarpaşa Limanı'nın konteyner depolama alanı 55.000m<sup>2</sup>'dir, sorunun büyüklüğü anlaşılmaktadır. Bu durum sistemin hata toleransını son derece düşürmektedir. Sistemin doğru

çalışması, konteynerlerin doğru yerlere yerleştirilmesi ve sisteme hatasız girilmesi ile mümkündür. Bu sistem insanlar tarafından yönetileceği için, hata yapılma ihtimali yüksektir. Bu nedenle ihtiyaç duyulan çözüm hataya karşı daha toleranslı olmalıdır.

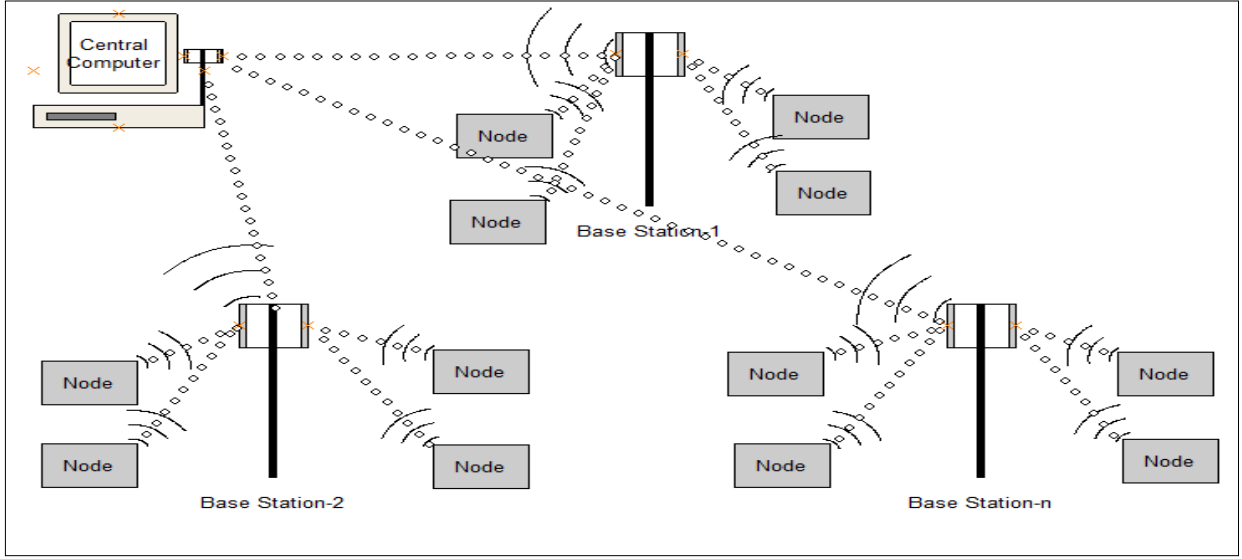
Bu çalışmamızda, problemin çözümü için, konteynerleri herhangi bir ön bilgi olmaksızın bulan bir gömülü sistem önerilmiştir. Bu doğrultuda, tasarlanacak gömülü sistemde kablosuz iletişim kullanılmasına karar verilmiş, bununla konteynerlerin yanlışlıkla olması gerektiğinden farklı bir yere yerleştirilmesi durumunda ortaya çıkacak sorunların önüne geçilmesi amaçlanmıştır. Bu durum sistemimizin hataya karşı toleransını arttırmış ve problemin mevcut çözümüyle karşılaştırıldığında, sistemimizin en önemli avantajı olmuştur. Tasarladığımız gömülü sistemle bir konteyneri bulmak için yapılması gereken tek şey, konteynerin eşsiz kimlik numarasını seçmektir. Sonrasında, konteynerin yeri hakkında bilgiler ekranda verilecektir. Sistemimizin kullanım kolaylığı bir diğer önemli avantajdır.

Sistemimizin kablosuz iletişim teknolojisi için üç farklı seçenek ele alınmıştır: GPS, RFID ve RF. Bu üç farklı teknoloji arasından seçim yaparken iki temel kriter göz önüne alınmıştır: kapsama alanı ve maliyet. 3000 m.ye kadar olan kapsama alanı ve diğer teknolojilere nazaran düşük maliyeti nedeniyle sistem iletişiminin RF alıcı/vericileri ile sağlanmasına karar verilmiştir.

Sistemin yapısı için iki farklı seçenek ele alınmıştır. Birincisinde, konteynerlerin üzerine birer verici yerleştirilmesi ve bu konteynerlerin yerlerinin limanda kurulacak, RF modülleri taşıyan sabit istasyonlar tarafından bulunması düşünülmüş ancak bu yapının mobiliteyi düşürmesi sebebiyle vazgeçilmiştir. Bu yapıyla ilgili diğer bir sorun, gelecek olan konteynerlerin yerlerinin önceden belli olmaması nedeniyle yerleştirilen istasyonların atıl kalması ihtimalidir. Böyle bir durumda sistemin verimliliği düşecektir. Bu nedenlerle bu yapıdan vazgeçilmiştir. Şekil 1'de bu yapı görülmektedir.

Sistemimizde konteynerlerin üzerine yerleştirilen vericilerle haberleşerek onların yerlerini tespit edecek bir el cihazı tasarlanmıştır. Kullanıcıyla etkileşim için el cihazına bir LCD ekran yerleştirilmiştir. Bu yapı sayesinde sistemimiz, herhangi bir altyapıya ihtiyaç duymadan, istenilen tüm limanlarda kullanılabilme imkanı sağlamaktadır. Şekil 2'de bu yapı görülmektedir.

Bir konteynerin uzaklığını hesaplamak için "Ulaşan Sinyal Gücü" yaklaşımı (RSSI approach) esas alınmıştır.[1] Ulaşan sinyal gücü, alıcı/vericilere gelen sinyalin gücünü gösteren bir çıktıdır. Konteynerin yönünü bulmak içinse, el cihazının anteni bir adım motor ile döndürülmüş ve ulaşan sinyal gücünün en yüksek olduğu yön seçilmiştir.



Şekil 1: Sabit istasyonlarla tasarlanan sistem

### 3. Sistem Yapısı ve Fonksiyonları

#### 3.1. Tasarımda Dikkat Edilen Hususlar

Sistemimiz iki gömülü sistemden oluşmaktadır: el cihazı ve vericiler. Her iki gömülü sistemde de, ortak olarak Aerocomm 2.4Ghz RF alıcı/vericileri bulunmaktadır.[2]

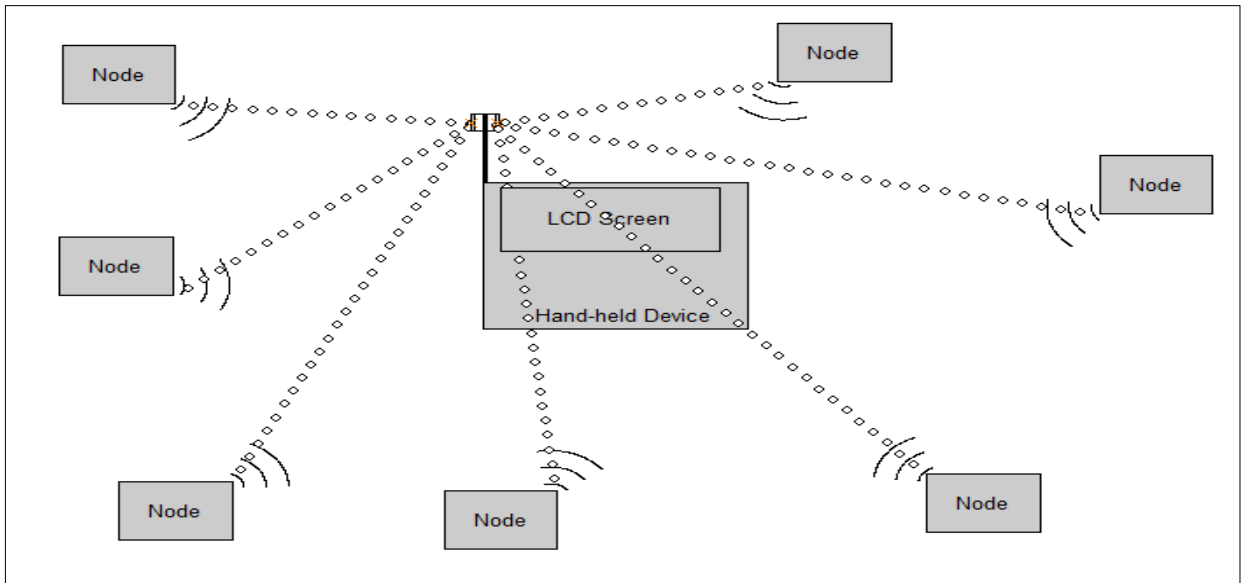
##### 3.1.1. El Cihazı

El cihazında ana kontrol kartı olarak Intel EMB-4650 kullanılmıştır.[3] El cihazında bir RF modülünün yanı sıra kullanıcıyla etkileşimi sağlamak amacıyla bir de LCD ekran bulunmaktadır. Yön tespiti için, anten bir adım motor ile döndürülmüş ve antenin arkasına gelen sinyali

sınırlandırmak için bir metal tabaka yerleştirilmiştir. El cihazının kontrolü bir bilgisayar faresi yardımıyla sağlanmaktadır. El cihazının yazılımı C# programlama diliyle yapılmıştır.

##### 3.1.2. Vericiler

Vericilerde ana kontrol kartı olarak Freescale MC9S12DP256B kullanılmıştır.[4] Bu kart ile alıcı/vericiden gelen veriler işlenmektedir. Ayrıca analog bir sinyal olan ulaşan sinyal gücü çıkışı (RSSI), bu kart üzerinde bulunan analog-dijital çevirici ile dijital bir değere çevrilmektedir. Vericilerin yazılımı C programlama diliyle yapılmıştır.



Şekil 2: El cihazı ile tasarlanan sistem

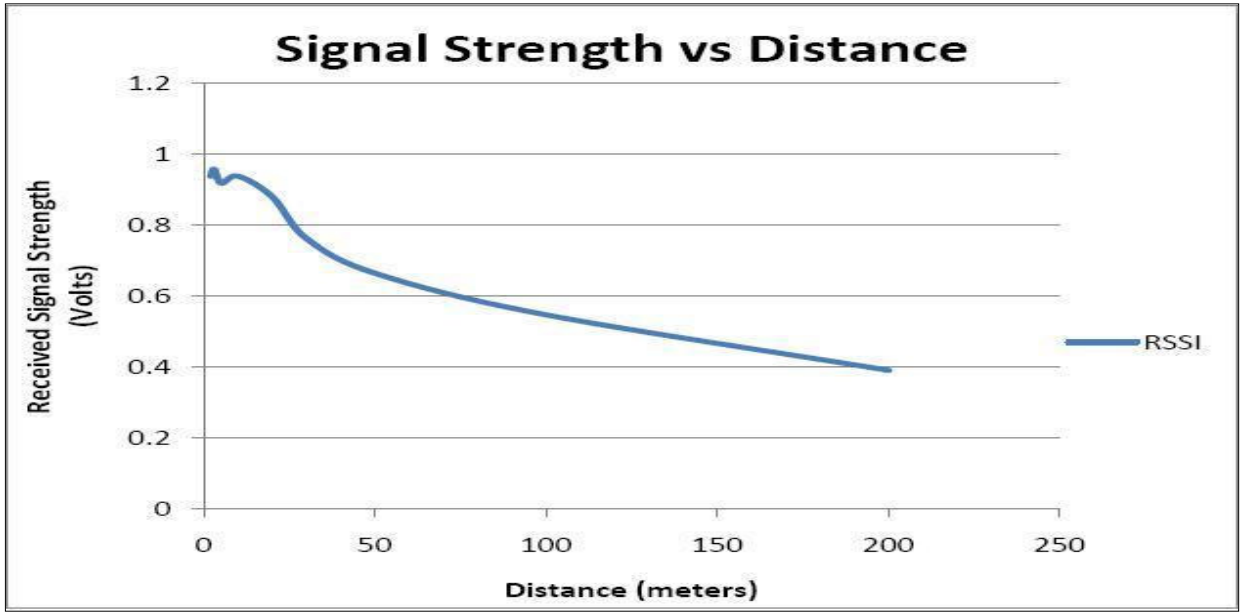
### 3.2. Sistem İletişimi

Sistemde iletişim 2.4 Ghz RF alıcı/vericiler ile sağlanmaktadır. Temel olarak, kullanıcı bulmak istediği konteyneri el cihazının ekranındaki menüden seçerek işlemi başlatır. El cihazı, bulunmak istenen konteynerin üzerindeki vericiye bir başlangıç sinyali gönderir. Sonrasında bu sinyali alan verici, ulaşan sinyal gücünü dijital bir değere çevirip, el cihazına geri gönderir. Gelen değere göre el cihazı konteynerin uzaklığını hesaplar. Bu işlem antenin adım motor ile döndüğü her adım için

tekrarlanır. Ulaşan sinyal gücü değerinin en yüksek olduğu yön konteynerin yönü olarak belirlenir.

### 4. Test Sonuçları ve Değerlendirme

Sistem için iki farklı test uygulanmıştır. Birinci deneyin amacı uzaklık ve ulaşan sinyal gücü (RSSI) arasındaki bağlantıyı belirlemektir. Bu doğrultuda 2 m. ile 200 m. arasında değişen uzaklıklarda ölçümler yapılmış ve Şekil 3'teki grafikte görülen sonuçlar elde edilmiştir.



Şekil 3: Uzaklığa bağlı sinyal gücü değişimi

Sistem konteyner alanlarında kullanılmak üzere tasarlandığından, alandaki metal yoğunluğu, el cihazı ve vericiler arasındaki iletişimde kullanılan sinyallerin yansıma olasılığını oldukça yükseltmektedir. Bu yansımalar sonucu alınan sinyal, aranan konteynerin yönünün yanlış tespit edilmesine yol açmaktadır. Bu nedenle bir konteynerin yerini tek bir tarama ile tespit etmek mümkün olmamaktadır. Ancak el cihazının gösterdiği yönde ilerleyip, taramaya devam edildiğinde bu sorun aşılma ve konteynerin yeri bulunmaktadır. Hesaplamaların ulaşan sinyalin yönünün yanında, ulaşan sinyal gücü üzerine de kurulu olması sayesinde yansımalarından kaynaklanabilecek yanlış bulguların önüne geçilmektedir.

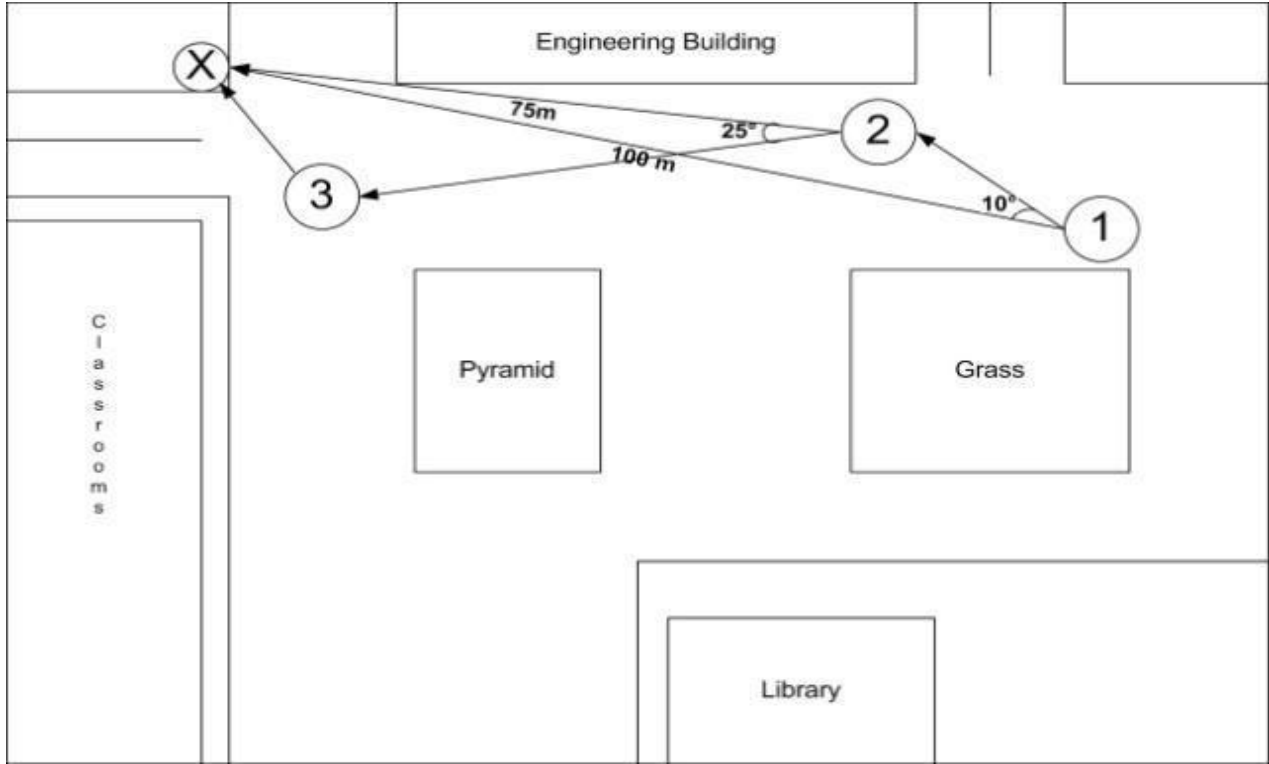
İkinci deneyde vericinin yerini bilmeyen bir kişi tarafından el cihazı kullanılarak bir verici bulunmaya çalışılmıştır. Başlangıç noktası 1 numaralı daire olup, numaralandırılan diğer noktalarda tarama tekrarlanmıştır. Bulunması hedeflenen verici "X" ile işaretli noktaya yerleştirilmiş ve kullanıcıdan vericinin bulunması istenmiştir. Her noktada

bulunan yön, uzaklıklar ve gerçek hedeften sapma açıları ve vericinin bulunması esnasında izlenen yol Şekil 4' te gösterilmiştir.

Bu deneyde verici ve el cihazı arasındaki başlangıç mesafesi 100 m.dir. Sonuç olarak sistem yeri bilinmeyen bir vericiyi 15 derecelik bir sapma ile bulmakta ve bu sapma vericiye yaklaşıldıkça azalmaktadır. Bu şekilde, yansımalarından kaynaklanabilecek problemlerin çözüldüğü görülmektedir.

### 5. Planlanan Çalışmalar

Şu anda sistem sadece istenilen bir konteynerin yerini tespit edebilmektedir. İleriki bir çalışma olarak, birden fazla konteynerin yerinin tespiti ve bir tür topoloji çıkarılması düşünülmektedir. Ayrıca bu sistemin konteyner gemilerine entegre edilerek gemi yükleme/boşaltma programlarıyla birlikte çalışması da yapılması planlanan bir çalışmadır.



Şekil 4: Vericinin bulunmasında izlenen yol.

## 6. Teşekkür

Bu bildiriye anlatılan tasarımımız, Çin'de düzenlenen Intel Gömülü Sistem Tasarım Yarışması (Intel Cup ESDC'10) kapsamında ikincilik ödülü kazanmıştır.[5] Deniz taşımacılığı ve konteynerler hakkındaki bilgilerini bizimle paylaşan CONTAZ[6] denizcilik firmasından Mehmet Hacı İbrahim'e, tasarım sürecinde bize destek olan CASLAB[7] üyesi doktora öğrencileri Salih Bayar ve Cem Ayyıldız'a, Intel kart ve LCD arasındaki kabloyu hiçbir ücret almadan bize sağlayan TTAF[8] şirketinden Erkan Sağlam'a ve projemiz için verdiği finansal destek için Mühendislik Fakültesi Dekanı Prof. Dr. Levent Akın'a teşekkür ederiz.

## 7. Kaynakça

[1] Vaidyanathan Ramadurai, Mihail L. Sichitiu.  
"Localization in Wireless Sensor Networks: A Probabilistic

Approach", International conference on Wireless Networks (ICWN03), 2003.

[2] Aerocomm RF Technologies,  
[http://www.aerocomm.com/docs/Datasheet\\_AC4424\\_HI.pdf](http://www.aerocomm.com/docs/Datasheet_AC4424_HI.pdf)

[3] [http://www.intel.com/?tr\\_TR\\_01](http://www.intel.com/?tr_TR_01)

[4] Freescale Semiconductor Corp.,  
<http://www.freescale.com/>

[5] Intel Cup Embedded System Design Contest  
[http://nuedc.sjtu.edu.cn/En/news/20100722\\_En.htm](http://nuedc.sjtu.edu.cn/En/news/20100722_En.htm)

[6] Contaz Maritime Transport and Trade S.A,  
<http://www.contaz.com/contazdeniz/index.asp>

[7] Computer Architecture and Systems Laboratory,  
[http://www.cmpe.boun.edu.tr/caslab/index\\_tr.html](http://www.cmpe.boun.edu.tr/caslab/index_tr.html)

[8] TTAF Elektronik Sanayi ve Ticaret Ltd. Şti.,  
<http://www.ttaf.com>

## Bir Gömülü Sistem Uygulaması: Elektronik Pusula

Mustafa Selman Yıldırım<sup>1</sup>, Onur Toker<sup>2</sup>, Sadık Kara<sup>3</sup>,

Elektrik-Elektronik Mühendisliği Bölümü

Fatih Üniversitesi, İstanbul.

<sup>1</sup>{mselmany}@gmail.com

<sup>2</sup>{onur}@fatih.edu.tr

<sup>3</sup>{skara}@fatih.edu.tr

### Özetçe

Bu projede iki adet manyetik alan sensörünün birbirine dik konumlandırılarak yön tayininde kullanılması çalışılmıştır. Bu elektronik sistem sayesinde mekanik pusulalardan çok daha hassas ve diğer elektronik sistemlere geri besleme yapabilecek bir pusula modülü elde edilmiştir. Bu modüle bir LCD arayüz de eklenerek kullanıcının yön bilgisini okuyabilmesi sağlanmıştır.

### 1. Giriş

Günümüzde uçak ve gemilerin kontrol sistemleri elektronik ekipmanlara dayandığı için yön bulma ihtiyacı da elektronik pusulalar tarafından sağlanmaktadır. Ülkemizde son zamanlarda yaygınlaşan navigasyon sistemlerinde de yön bulma ihtiyacı elektronik pusula sistemleri ile sağlanmaktadır. Bazı otomobil ve cep telefonlarında kullanılan sistemler buna örnek gösterilebilir. Ayrıca sadece pusula olarak kullanımda da günümüzde elektronik pusulalara rağbet artmaktadır. Bazı saatlerin pusula modülleri buna örnek verilebilir. Klasik pusulaların mekanik sürtünmeler sebebiyle hatalı sonuç vermesi, yön bilgisini elektriksiz sinyal olarak alamamamız bizi elektronik pusula kullanmaya mecbur bırakmaktadır.

Bu çalışmada temel yön bulma fonksiyonunu sağlayan, kullanıcı dostu arayüzü olan ekonomik bir pusula yapımına çalışılmıştır.

### 2. Kullanılan Enstrümanlar

#### 2.1. Manyetik Alan Algılayıcısı (Sensör)

Projede yön tayini için kullanılacak referans klasik pusulalarda olduğu gibi dünyanın manyetik alanıdır. Bu manyetik alanın ölçülmesi için manyetik alan sensörleri kullanılabilir. Manyetik alan ölçümleri için günümüzde kullanılan temel sensörler şekil-1'de verilmiştir[1].

Bu bölümde manyetik alan sensörleri tartışılacaktır.

**Squid:** Çok geniş bir ölçüm aralığına sahip olmasına rağmen 4K sıcaklığında çalışabildiğinden gelişmiş laboratuvarlarda kullanılabilir.

**Search-coil:** Çalışma prensibi Faraday kanununa dayanır. Üretimi kolay ve ucuzdur. Fakat değişken manyetik alanları ölçülebilir. Sabit manyetik alanların ölçülebilmesi için kendisinin hareket halinde olması gerekmektedir.

**Fiber-optic, optically pumped, nuclear procession sensörleri** ise yüksek doğruluk gerektiren bazı tıbbi veya laboratuvar uygulamalarında kullanılan pahalı sensörlerdir.

Magnetic Sensor Technology	Detectable Field Range (gauss)*				
	10 <sup>-8</sup>	10 <sup>-4</sup>	10 <sup>0</sup>	10 <sup>4</sup>	10 <sup>8</sup>
Squid	10 <sup>-8</sup>	10 <sup>-4</sup>	10 <sup>0</sup>	10 <sup>4</sup>	10 <sup>8</sup>
Fiber-Optic	10 <sup>-8</sup>	10 <sup>-4</sup>	10 <sup>0</sup>	10 <sup>4</sup>	10 <sup>8</sup>
Optically Pumped	10 <sup>-8</sup>	10 <sup>-4</sup>	10 <sup>0</sup>	10 <sup>4</sup>	10 <sup>8</sup>
Nuclear Precession	10 <sup>-8</sup>	10 <sup>-4</sup>	10 <sup>0</sup>	10 <sup>4</sup>	10 <sup>8</sup>
Search-Coil	10 <sup>-8</sup>	10 <sup>-4</sup>	10 <sup>0</sup>	10 <sup>4</sup>	10 <sup>8</sup>
Earth's Field					
Anisotropic Magnetoresistive	10 <sup>-8</sup>	10 <sup>-4</sup>	10 <sup>0</sup>	10 <sup>4</sup>	10 <sup>8</sup>
Flux-Gate	10 <sup>-8</sup>	10 <sup>-4</sup>	10 <sup>0</sup>	10 <sup>4</sup>	10 <sup>8</sup>
Magnetotransistor	10 <sup>-8</sup>	10 <sup>-4</sup>	10 <sup>0</sup>	10 <sup>4</sup>	10 <sup>8</sup>
Magnetodiode	10 <sup>-8</sup>	10 <sup>-4</sup>	10 <sup>0</sup>	10 <sup>4</sup>	10 <sup>8</sup>
Magneto-Optical Sensor	10 <sup>-8</sup>	10 <sup>-4</sup>	10 <sup>0</sup>	10 <sup>4</sup>	10 <sup>8</sup>
Giant Magnetoresistive	10 <sup>-8</sup>	10 <sup>-4</sup>	10 <sup>0</sup>	10 <sup>4</sup>	10 <sup>8</sup>
Hall-Effect Sensor	10 <sup>-8</sup>	10 <sup>-4</sup>	10 <sup>0</sup>	10 <sup>4</sup>	10 <sup>8</sup>

\* Note: 1gauss = 10<sup>-4</sup>Tesla = 10<sup>5</sup>gamma

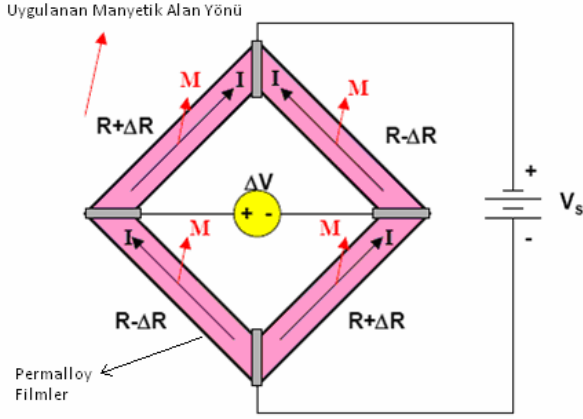
Şekil 1: Temel manyetik alan algılayıcı çeşitleri [1]

**Flux-Gate:** Dünya'nın manyetik alanını ölçmeye en uygun sensörlerden biridir. Ferromanyetik nüve üzerine sarılmış iki sarımdan oluşur. Birincil sarıma kare dalga uygulanır. Harici statik manyetik alanın etkisi ile ikincil sarğı gerilim formunda değişiklikler olur. Bu gerilim demodüle edilir, alçak geçiren filtreden geçirilerek harici statik alan bilgisi elde edilir. Bu tür sensörler büyük ve hantaldır, entegre devre olarak üretilmeye uygun değildir.

**AMR:** Dünya'nın manyetik alanını ölçmeye uygun bir diğer sensör Anisotropic Magneto-Resistive sensörlerdir. Bu sensörler permalloy denen nikel-demir alaşımından üretilmiş 4 tane ince film içerir. Filmelerin özelliği uygulanan statik manyetik alana göre direncinin %2-3 oranında değişebilmesidir. Sensörün çıkış gerilimi dört filmin Wheatstone köprüsü şeklinde konumlandırılmasıyla sağlanır. Dirençler farklı doğrultularda uzandığı için üzerlerine etkiyen manyetik alan vektörleri farklıdır. Bu da direnç değerlerinin farklı olmasına ve köprü geriliminin değişmesine sebep olur. Şekil-2'de AMR sensörün şematik yapısı gösterilmiştir[2].

AMR sensörlerin en önemli avantajı ticari entegreler olarak üretilmesidir. Bu sebeple projede AMR sensör kullanılacaktır.

Muhtelif AMR sensörler içinde seçim yaparken belirleyici kriterler: ölçüm aralığı, ofset miktarının küçük olması, gauss başına hassasiyet, dâhili S/R bobini olması ve S/R akımının küçük olmasıdır. Ayrıca lehimleme sırasında problem olmaması için boyutlarının büyük olması, surface mount tipi olmaması tercih edilmiştir.



Şekil 2: AMR sensörün yapısı[2].

Yaygın kullandığımız AMR sensörlerini üreten -bilindiği kadarıyla- üç firma vardır: Philips, HL Planar ve Honeywell. Entegre seçiminde belirleyici bazı özellikler tablo-1'de verilmiştir[3].

Tablo 1: AMR sensörlerin karakteristikleri [3]

	Hassasiyet	Çözünürlük	Ölçüm Aralığı	S/R Akımı	Ofset (tipik)
Birim:	mV/V /Gauss	μGauss	Gauss	A	mV/V
HMC100 1 /1002	3,1	27	±2	3	±15
HMC102 1 /1022	1,0	85	±6	0,5	±2,5
HMC105 1 /1052	1,0	120	±6	0,5	±0,5
KMZ51 /52	1,28	?	±2,5	1	±1,5
KMY20M	0,44±0,12	?	?	S/R yok	±1,5

Tabloda verilen entegrelerin kodlarındaki son basamak sensörün kaç eksenli olduğunu göstermektedir. Ticari ve seri üretimde maliyet ve boyutların düşmesi için pusula uygulamalarında iki eksenli sensörler daha avantajlıdır. Çünkü manyetik alan sensörleri sadece hassas oldukları eksenlerdeki manyetik alan büyüklüğünü ölçebilmektedir. Manyetik alanın yönünün belirlenebilmesi için ise yere paralel ve birbirine dik iki eksenin ölçülmesi ve birbirine trigonometrik olarak kıyaslanması gerekmektedir. Fakat bu projede işçilik elle yapılacağından tek eksenli entegrelerin boyutunun büyük olması ve iki eksenli sensörü Türkiye'de temin edemememiz sebebiyle iki adet tek eksenli sensör birbirine dik olarak kullanılmıştır. Bu entegreler tablo-1'de belirtilen üstünlükleri sebebiyle HMC1051Z olarak seçilmiştir.

## 2.2. Yükseltici Devre

AMR sensör çıkış sinyalinin işlenebilecek bir genliğe getirilebilmesi için yükseltici bir devre kullanılması gerekmektedir. Uygun fiyat, temin edilme kolaylığı, kullanım pratikliği sebebiyle bu iş için 741 opamp entegresi kullanılmıştır. Fakat opamplı sıradan bir devrenin giriş direnci yeterince büyük olmayacağından ve CMRR oranı yüksek olmayacağından opamp diferansiyel(fark) yükseltici konfigürasyonunda bağlanmıştır. Bu aynı zamanda tek opamp elde edilebilecek en basit enstrümantasyon yükseltici devredir[4]. Bu devrenin ayrıntıları bölüm-4.2'de incelenecektir.

## 2.3. A/D Çevirici ve Mikrokontrolör

Sensör çıkış voltajının yükseltici devrede yükseltilmiş halini açılı bilgisi olarak elde etmek için mikrokontrolör entegresi gereklidir. Mikrokontrolörler veriyi dijital olarak işlediklerinden, yükseltici çıkışının öncelikle dijital veriye dönüştürülmesi gerekmektedir. Bu işlemi ticari entegreler olarak satılan A/D çeviricilerle yapmak da mümkün olmasına rağmen fiyat ve pratiklik açısından dahili A/D çeviricisi olan bir mikrokontrolör kullanmak daha akıllıcadır. Bu projede dahili 10 bit A/D çeviricisi olan PIC18F452 mikrokontrolörü kullanılmıştır. Mikrokontrolörün programlanması ve çalışması için tipic [7] devresinden faydalanılmıştır

## 2.4. Arayüz

Mikrokontrolör ile açılı bilgisine çevrilen manyetik alan yönü bilgisinin kullanıcı tarafından okunabilmesi için 2x16 karakter LCD ekran modülü kullanılmıştır.

## 3. Uygulamada Karşılaşılan Problemler ve Çözüm Önerileri

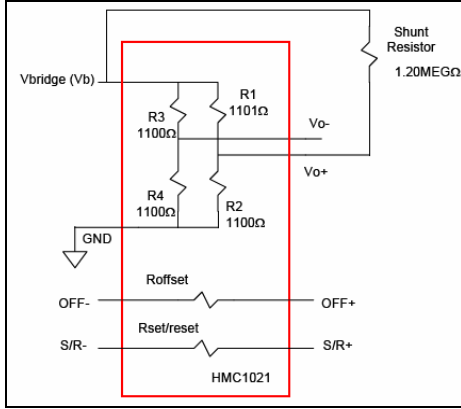
### 3.1. Ofset (sıfır) Hatası

Opamlara ait üretim toleransından kaynaklanan ofset hatası bir potansiyometre kullanılıp giriş geriliminin 0V olduğu durumda çıkış geriliminin 0V olması sağlanarak giderilebilir. Burada üzerinde durulacak olan problem opamlara ait ofset hatası değil, AMR sensörlerin üretim toleranslarından kaynaklanan ofset hatasıdır.

Gerekli enstrümanlar bölümünde anlatıldığı gibi projede kullanacağımız AMR sensörler dört adet direncin Wheatstone köprüsü şeklinde bağlanmasıyla elde edilmektedir. Bu dirençlerin nominal değeri 1100Ω'dur[5]. Şimdi üretim hatası sonucu bu dirençlerden sadece birinin 1Ω daha büyük olduğunu varsayalım. Köprü geriliminin 5V olduğunu ve herhangi bir uyarıcı alan olmadığını düşünelim. Bu durumda çıkış gerilimi 1,1mV olur. Bu ciddi bir değişim sayılmayabilir. Fakat kullanacağımız entegrenin hassasiyetinin 1,0mV/V/G olduğunu göz önünde bulundurursak, Dünyanın manyetik alanı en fazla 600mG olduğuna göre, 5V köprü geriliminde çıkışımız en fazla 0,6G\*5V\*1,0mV/V/G=3mV olur. Zaten çıkış aralığımız ±3mV iken buna 1,1mV ofsetin eklenmesi ciddi bir problemdir. AMR sensörün ofset hatasının giderilmesi için üreticinin önerdiği beş yöntem vardır[5]. Bu bölümün devamında ofset hatasının giderilmesi tartışılacaktır.

### 3.1.1. Şönt Direnci Metodu

Bu metod, direnci en büyük köprü direncine paralel bir direnç bağlayarak direnci düşürmekten ibarettir. Figür-2.1 bu metodu göstermektedir. Fakat şönt direnç metodunu uygulayabilmek için sensöre etkiyen manyetik alanın kaldırılabilmesi gerekir.



Şekil 3: Şönt Direnci Metodu [5]

### 3.1.2. Ters Ofset Metodu

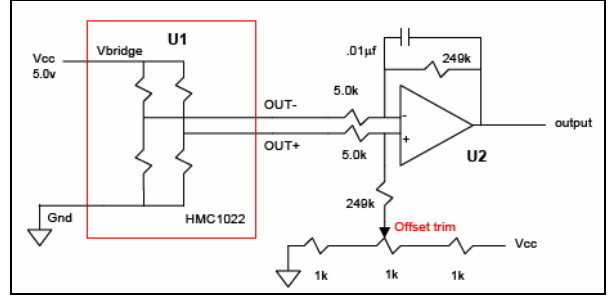
Bu metotta köprüye müdahale edilmez. Fakat köprü çıkışına konan opamp'a aksi yönde bir ofset oluşturulur. Böylece opamp çıkışında görülen değerde ofset bulunmaz. Bu metodun uygulanabilmesi için de manyetik alanın olmadığı durumdaki ofset miktarının bilinmesi gerekir. Şekil-4 bu metodu şematize etmektedir.

### 3.1.3. Anahtarlamalı Geri Besleme Metodu

Bu metod sensörün dahili olarak bulundurduğu set/reset şeridi sayesinde algılama yönünün 180° döndürülebilmesi özelliğini temel olarak çalışır. Sensör çıkışının set/reset durumları şekil-5'te görüldüğü gibidir. Eğer çıkışın ortalaması alınırsa Voff değeri bulunur. Ters ofset metodunda çıkışı sıfırlamak için uygulanan ofset değeri manuel olarak ayarlanıyordu. Bu metotta ise oluşturulacak ters ofset Voff değerinin geri beslenmesiyle ayarlanır. Bunları analog olarak yapan bir devre örneği şekil-6'da verilmiştir. Bu devrede U3 bölümü, U2 çıkışının integralini alıp U2 girişine ofset referansı olarak besleyen devredir. U4 bölümü ise çıkışta oluşan ters polarite durumunu düzeltir. Set/reset periyodu integral periyodunun %1'i kadar olursa bu devre pürüzsüz bir çıkış verecektir. Bunları sağlayabilmek için direnç ve kapasitans değerleri dikkatli seçilmelidir.

### 3.1.4. Ofset Sargısı Akımı Metodu

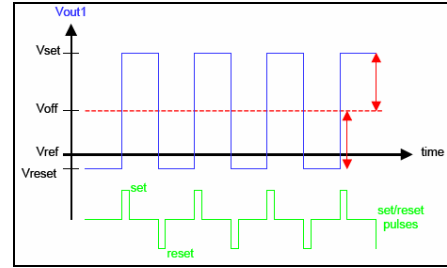
Honeywell AMR sensörlerinde bulunan dâhili ofset sargısı, ölçülen manyetik alana herhangi bir manyetik alan eklemek veya çıkış ofset gerilimini engellemek için kullanılabilir. Bu sargıdan bir akım geçirilerek ofset miktarını sıfırlayacak büyüklükte bir manyetik alan oluşturabilir. Fakat ilk iki yöntemde olduğu gibi manyetik alanın olmadığı bir bölge oluşturmamız gerekmektedir.



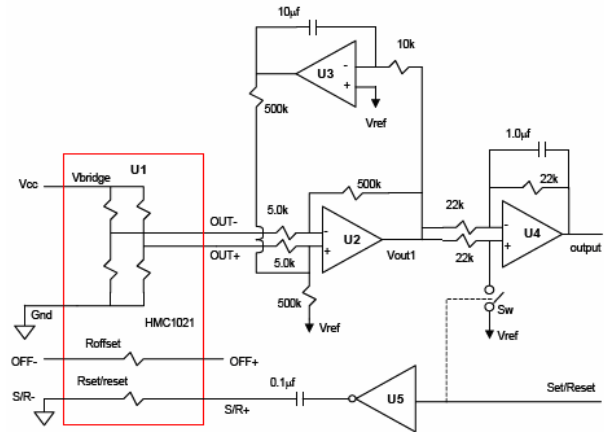
Şekil 4: Ters Ofset Metodu [5]

### 3.1.5. Dijital Çıkarma Metodu

Pusula uygulamalarında en çok kullanılan metod dijital çıkarma metodudur. Çünkü bu metod ekstra donanımlar gerektirmemektedir. Fakat dezavantajı, analog-dijital çeviricinin ölçüm aralığının büyük bir kısmının ofset yüzünden boşa kullanılmasıdır. Bu da çözünürlüğün azalmasına sebep olur.



Şekil 5: Sensör Çıkışında S/R etkisi [5]



Şekil 6: Anahtarlamalı Geri Besleme Metodu [5]

Dijital çıkarma miktarının belirlenebilmesi için değişik yöntemler olabilir. Mesela manyetik alanın olmadığı bir ortam oluşturularak dijital çıkışın değeri ölçülebilir. Veya set/reset özelliği kullanılarak çıkışın dijital ortalaması hesaplanıp bulunan değerden çıkarılabilir. Fakat bu durumda da 0,5A'lık set/reset akımını mikro kontrolörün yönetebilmesini

sağlayacak akım yükseltici bir devreye ihtiyaç vardır. Ve her iki durumda da ADC ölçüm aralığı ofsete feda edilmiştir.

### 3.1.6. Bu Çalışmada Kullanılan Metot

Buraya kadar üreticinin önerdiği beş farklı metottan bahsettik. Birinci, ikinci, dördüncü ve beşinci metotlar için herhangi bir manyetik alanın olmadığı özel bir ortam oluşturmamız gerekmektedir. Bu helmholtz coil denen bir aletle veya özel korunmuş kapalı bir kafesle sağlanabilir. Fakat bunlar pahalı çözümlerdir.

Üçüncü ve beşinci metod 0,5A büyüklüğünde bir akımın mikrokontrolörle anahtarlanmasını sağlayacak bir devreye ihtiyaç duymaktadır. Ayrıca üçüncü metod hassas opamp'lar, direnç ve kapasitörlerle yapılan karmaşık bir devre de gerektirmektedir. Bu metodlardan da faydalanarak bizim geliştirdiğimiz metod ise sensörü yere paralel düzlemde düşük bir hızda çevirmek suretiyle manyetik alanın maksimum ve minimum değerlerini mikro denetleyiciye kaydetmek, bunların ortalamasını alarak salt ofset değerini bulmak, son olarak da bu ofseti opamp aşamasında sıfırlayarak ADC aralığının çoğunun kullanılabilmesini sağlamaktır. Mekanik çevirmeyi önermemizin sebebi 0,5A akımı mikrokontrolör yönetiminde set/reset pinlerine anahtarlamak için gereken devreden kurtulmaktır.

### 3.2. Köprü Manyetizasyonunun Bozulması

AMR sensörlerin içindeki permalloy malzemede dipol mıknatıslar, harici bir manyetik alana maruz kaldığında tekrar dizilir. Bunun sonucunda sensör hassasiyetinin azalması veya sensörün yanlış sonuç vermesiyle karşılaşılabilir. Normal şartlarda kararlı düzenini yıllarca koruyabilen manyetik dipollerin yönelimini bozmak için 20G büyüklüğünde manyetik alan yeterlidir[6]. Bu alana kalıcı mıknatıslar, yüksek akım kullanan cihazlar, hoparlörler, gerilim hatları, trafolar, CRT monitörler sebep olabilir. Ayrıca deneylerimizde köprü manyetizasyonunun kendi içinden geçen akımla da bir miktar bozulduğunu gözlemledik.

Permalloy malzemede bu bozulmanın kalıcı olmayacağını üretici 10000Gauss'a kadar garanti etmektedir[3]. Daha küçük etkilerle oluşan bozulmaları düzeltmek için sensörün set/reset pinlerine 0,5A akım verilmesi yeterlidir. Bu sayede dipoller tekrar doğru yönelime geçerler.

Devremizde set/reset akımını sağlamak için TinyPic devresinde regüle edilmiş 5V gerilimi birbirine seri bağladığımız S/R bobinlerine bağladık. S/R sarımlarının nominal direnci 5Ω kadardır. İki S/R bobininin birbirine seri bağlanması ile 10Ω'luk direnç olur. 5V gerilimi bu dirence verdiğimizde üreticinin önerdiği S/R akımı olan 0,5A elde edilmiş olur. Bu akımı sadece ölçümden hemen önce kısa bir süreliğine vereceğiz. Aksi takdirde entegre aşırı ısınmaktadır. Ayrıca devreye enerji sağlaması için 0,5A'lık DC adaptör kullanılmıştır.

### 3.3. Eğim Etkisi

Ölçümlerde eğer pusula yere paralel düzlemde bulunmazsa bir veya iki bileşen yanlış ölçülebilir. Bunu engellemek için projeye bir eğim sensörü (tilt sensor) eklenebilir. Fakat bu maliyeti artırır. Bu nedenle manyetik sensörü ölçüm esnasında yere paralel durumda tutmak daha pratik bir çözüm olacaktır.

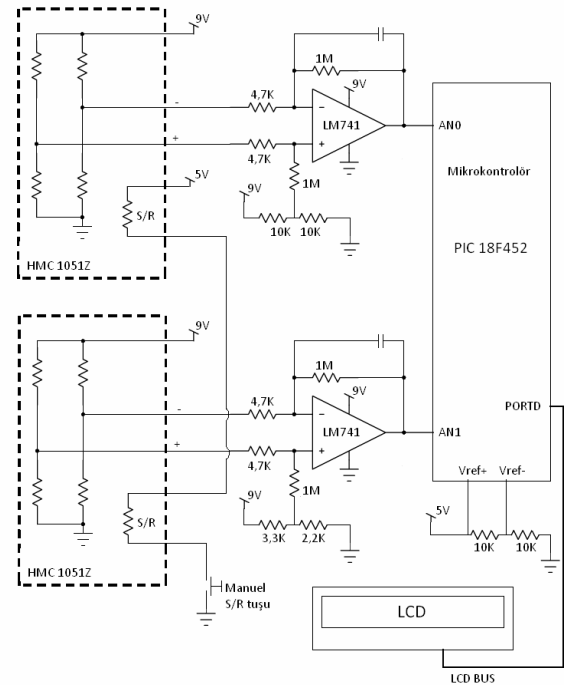
### 3.4. Manyetik Kuzey Sapması

Dünya üzerinde bulunulan konuma bağlı olarak Dünya'nın manyetik kuzey kutbuyla coğrafi kuzey kutbu birbirinden 10° veya daha fazla, batıya veya doğuya sapabilir. Ayrıca bu sapma miktarı yavaş da olsa sürekli değişmektedir[8]. Bu durumdan klasik mekanik pusulalar da aynı oranda etkilenmektedir. Eğer pusulanın kullanılacağı bölge belliye o bölgenin sapma bilgisi güncel olarak öğrenilip manyetik yönden çıkarılabilir. Fakat ölçümdeki bu sapmanın kritik olmadığı durumlarda manyetik kuzey kutbunun ölçülmesi de yeterli olmaktadır. Ayrıca bir farklı çözüm de Güneş'in görüldüğü zamanlarda Güneş yönüne göre kuzeyi hesaplayan programlar kullanarak elektronik pusulayı kalibre etmektir. Fakat pusulanın kullanıldığı konum değişirse bu kalibrasyonu tekrar yapmak gereklidir

## 4. Uygulama

### 4.1. Manyetik Alanın Ölçülmesi

Manyetik alanın ölçülmesinden hemen önce S/R bobinlerine kısa bir süreliğine akım vererek manyetik dizilimin doğru yönde olmasını ve hassasiyetin artırılmasını sağlıyoruz. Devreyi çalıştırınca manuel S/R düğmesine basmamızı LCD aracılığıyla mikrokontrolör hatırlatıyor. Ek-1'deki devrede görüldüğü gibi entegrelerin köprü girişine bağlı pinler paralel olarak 9V gerilime bağlanmıştır. Köprü çıkışlarına bağlı pinler de ayrı ayrı yükseltilme katmanlarına bağlanmıştır. Ayrıca S/R bobinleri de bir düğme üzerinden birbirine seri olarak 5V'a bağlıdır. Devrenin çalışmasıyla beraber köprü çıkışlarında görülen gerilim ofseti giderilmemiş ve yükseltilmemiş manyetik alan büyüklüğü bilgisidir.



Şekil 7: Sensörler, yardımcı devreler ve mikrokontrolör



## 4.2. Sinyalin Yükseltilmesi

Milivoltlar seviyesinde elde edilen manyetik alan büyüklüğü verilerinin mikrokontrolör tarafından doğru ölçülebilmesi için voltlar seviyesine yükseltilmesi gerekir. Bunu sağlamak için fark yükseltici devresi kullanılmıştır. Bu devre aynı zamanda tek opamp enstrümantasyon yükseltici elde etmenin basit bir yoludur. Bizim devremizde yaklaşık kazanç 213'tür.

## 4.3. A/D Çevrim

Bu aşamada yükseltici çıkışlarının mikrokontrolörün analog girişlerine bağlanmasıyla, yükseltilmiş ve ofseti kabaca azaltılmış yön büyüklüğü bilgisinin mikrokontrolör tarafından ölçülüp dijital büyüklüğe çevrilmesi sağlanmıştır. AN0 kanalından birinci opamp çıkışı, AN1 kanalından da ikinci opamp çıkışı ölçülmüştür. Ayrıca pozitif referans voltajı olarak 5V, negatif referans voltajı olarak da 2,5V seçilmiştir. A/D çeviricimizin çıkışı 10 bit olmasına rağmen, mikrokontrolörün 32Kbyte'lik hafızası çok büyük tablolara izin vermeyeceğinden çevrim sonucunun en önemli 8 biti kullanılmıştır. 8 bit bir ölçüm sonucu yaklaşık 1° lik ölçüm hassasiyeti sağladığı gibi, kullanım kolaylığı da sağlamış, ayrıca gerekli tablo da mikrokontrolör hafızasının büyük bir kısmını kaplamıştır. Tablonun fonksiyonu, büyüklüğü ve hazırlanması sonraki bölümlerde tartışılacaktır.

Ölçüm esnasında düşük frekanslı gürültülerden kaynaklanan ölçüm hatalarını filtre etmek için 10ms aralıklarla her opamp için 16'şar ölçüm yapıp ortalaması alınmıştır. 16'ya bölünen sonuç tekrar 8 bite düşmüş, bu sonuç da IC1 registerine kaydedilmiştir. Bölme işleminde kaybedilen basamakların en büyüğünün '1' olması durumunda sonuç bir artırılarak yuvarlama hatası yarıya indirilmiştir. Bu adımların aynısı diğer analog kanalın ölçülmesinde de tekrarlanmış ve sonuç IC2 registerine kaydedilmiştir.

## 4.4. Çıkış Aralığının Belirlenmesi

Bölüm 3.1.6'da belirtildiği gibi opamp çıkışlarındaki ofsetin tamamen giderilmesi için 360° lik tüm spektrumda elde edilen çıkışların her bir entegre için minimum ve maksimum değerlerinin ortalamasını almamız gerekmektedir. Bu değer bize manyetik alanın olmadığı durumdaki çıkışı bildirir. Referansımızı da bu değere göre almamız gerekir. Minimum ve maksimum değerdeki hataları elimine etmek için ölçümler 16 kez yapıp ortalaması alınmıştır. Program iki analog kanalın da en büyük ve en küçük değerlerini LCD ekrana yazdırmaktadır. Bu değerler ekrandan okunup ortalamaları alınmıştır. Bulunan ortalama değerler referans değerlerimizdir. Tablo-2'de en büyük ve en küçük değerler verilmiştir.

Tablo 2: Analog Kanalların Ölçüm Sonuçları

	En Büyük	En küçük	Ortalama	Fark
AN0	206	102	154	104
AN1	155	59	107	96

Not: Değerler onluk tabanda verilmiştir.

## 4.5. Ölçümlerin Program Hafızasında Adres Olarak Kullanılması

Buraya kadar manyetik alan büyüklüğü bilgilerinin dijital

olarak elde edilmesini inceledik. Fakat bu bilgiler bir kullanıcı tarafından kolayca yön olarak algılanamaz. Dolayısıyla kullanıcı arayüzünde manyetik alan büyüklüğü bilgileri yerine manyetik alan yönünün derece cinsinden açısı verilmelidir. Yani mikrokontrolörümüzün iki tane büyüklük vektörünü açığa çevirmesi gerekmektedir. Burada iki problem çözülmelidir. Birinci problem iki büyüklük vektörünün birbirine bölünerek açının tanjantının hesaplanmasıdır. PIC 18F452 mikrokontrolörü bölme işlemi için özel bir komut desteklememektedir. Böyle durumlarda önerilebilecek bir çözüm mikrokontrolörün program hafızasında veri tablosu oluşturmaktır. Mesela bölme işlemi için kullanılacak bir veri tablosu bölünen ve bölen sayıları bir adres gibi almayı, bölümü de bu adreste bulunan bilgi olarak bize verebilir.

Yön hesaplamadaki ikinci problem ise tanjant bulunan açının ters tanjant fonksiyonuyla kendisinin hesaplanmasıdır. Mikrokontrolörümüzün ters tanjant fonksiyonu için de bir komutu yoktur. Yapılabilecek şey ters tanjant için de bir tablo oluşturmaktır. Fakat hem bölme işlemi için, hem de ters tanjant fonksiyonu için tablo kullanılması durumunda pic'in hafızası kısıtlı olduğu için tablolarda bazı kısıtlamalar yapmak gerekir. Hâlbuki bizim bölme sonuçlarına ihtiyacımız yoktur. Öyleyse tek bir veri tablosunda, büyüklükleri bilinen vektörlerin oranlarının ters tanjantlarını saklayabiliriz. Tablo büyüklüğü için bizi kısıtlayan pic'in hafızasıdır. Pic'in program hafızası 32Kbyte'dir. Yani bu hafızanın adres göstergesi 15 bittir. ( $32K=2^{15}$ ). Analog kanallardan ölçtüğümüz veriler ise ölçülen veri aralıkları 128'den küçük olduğu için  $2 \times 7 = 14$  bittir. Öyleyse ölçtüğümüz verileri basit bir öteleme ile tablo adreslemesi için kullanabiliriz. Ardışık her iki baytı da birleşik olarak kullanırsak açılı bilgisini yazmak için ikişer baytımız olur. Açığı ekrana yazdırmak istediğimiz için BCD formatında saklarsak tekrar çevirmek zorunda kalmayız. Açılı büyüklüğü 0~360 aralığında olduğundan zaten her açı için en az üç bayt gereklidir. Ardışık her iki baytı beraber kullanacağımız için adreste 1 bit de bunun için kullanmamız gerekir. Ölçtüğümüz verilerin 14 bitlik bilgisine bu bit de ekleyerek 15 bitlik adresin tamamını kullanmış oluruz.

## 4.6. Veri Tablosunun Oluşturulması

Bölüm 4.5'te açıklandığı gibi büyüklük vektörlerini açığa çevirmek için veri tablosu oluşturulmalı ve mikrokontrolörün program hafızasına kaydedilmelidir.

Pic'in program hafızasından yapılan okumalarda iki adet register işaretçi olarak kullanılır. Bunlar TBLPTRH ve TBLPTRL'dir. Okuma esnasında adres bu registerlere yüklenecektir. TBLPTRH registerine yüklenecek adres IC1-102+15 olarak seçilmiştir. 102 sayısı birinci kanalın ofset değeri olup sunucun 7 bite sığmasını sağlamak için çıkarılmıştır. 15 ise öteleme olup assembly programının yazılacağı ilk kısmı boş bırakmak içindir. Bu durumda tablo program hafızasında  $15 \times 256 = 3840$  byte ötelenmiştir. TBLPTRL registerine yüklenecek adres ise (IC2-59)x2 olarak seçilmiştir. 59 ikinci analog kanalın ofset değeri olup yine sonucu 7 bite sığdırabilmek için çıkarılmıştır. 2 ile çarpma işlemi ise sonucu 1 bit sola kaydırmak demektir. Bu da son bitini ardışık byte'lar arasında seçim yapabilmemizi sağlar.

Adreslere yüklenecek açılar ise aşağıdaki formül ile hesaplanır.

$$\alpha_i = \arctan \left( \frac{IC1 - 154}{IC2 - 107} \times \frac{96}{104} \right) \times \frac{180}{\pi} \quad (1)$$

Bu formüldeki 154 ve 107 sırasıyla birinci ve ikinci kanalda manyetik alanın olmadığı durumda okunan değerlerdir. Ölçülen aralık tablo-2’de de görüldüğü gibi iki kanal için birbirinden biraz farklıdır. Bu farkı telafi etmek için vektörler 96/104 ile normalize edilmiştir. Formülün sonunda ise radyan olarak bulunan açı değerinin dereceye çevrilmesi sağlanmıştır. Pic’e yüklememiz gereken veri tablosu yaklaşık 10300 satırdan oluşur ve her satır için formül-1’de verilen işlemin sonucunu hesaplamak gerekir. Yani elle teker teker yazmak mümkün değildir. Projede bu tabloyu oluşturmak için C dilinde bir program yazılmıştır. Bu program veri tablosunu oluştururken analog kanalların muhtemel bütün durumlarına göre formül-1’i hesaplar, ayrıca bulduğu açının hangi çeyrek bölgeye olduğuna göre farklı işlemler yaparak sonucu tabloya BCD formatında kaydeder. Hangi bölgede hangi verinin yazılacağı tablo-3’te verilmiştir.

#### 4.7. Yönün Ekranda Yazdırılması

Projede kullanıcı arayüzüne yazdırılacak açının doğru ile yapılan pozitif yönlü açı olması tercih edilmiştir. Ekran yazdırma işlemi için öncelikle pic’in program hafızasından açının okunması gerekmektedir. Açı okunurken adres registerlerine yazılacak veri analog kanallardan ölçülen büyüklüktür. Bu büyüklük herhangi bir bozucu etki sebebiyle –mesela çok yakındaki bir mıknatıs– içinde olması gereken sınırı aşabilir. Böyle bir durumda adres işaretçisine beklenmedik bir veri yüklenebilir ve program kararsız kalabilir. Bunu engellemek için analog kanalların büyüklüğü

Tablo 3: Veri Tablosu İçin Durumlar

ŞART		SONUÇ
IC1<154	IC2<107	VERİ= 180+AÇI
	IC2>107	VERİ= AÇI
IC1>154	IC2<107	VERİ= 180+AÇI
	IC2>107	VERİ= 360+AÇI
IC1=154	IC2>107	VERİ= 0000
	IC2<107	VERİ= 0180
IC2=107	IC1>154	VERİ= 0270
	IC1<154	VERİ= 0090

adres registerine yazılmadan önce kontrol edilmeli ve sınırları aşma durumunda sınırdaki en yakın sayıya çevrilmelidir. Daha sonra adres işaretçilerine gerekli bilgiler yüklenip hafızadan okuma yapılmalı ve ekrana yazdırılmalıdır.

#### 5. Sonuçlar

Bu projede temel olarak iki adet magneto-resistive sensör ve bir mikrokontrolör kullanılarak düşük maliyetli ve oldukça hassas, kullanımı kolay bir pusula yapımının mümkün olduğu gösterilmiştir. Çok küçük miktarlardaki lineerlik, sıcaklık, yineleme hatalarını göz ardı edersek, 8 bit analog ölçümden kaynaklanan hata 1/512 seviyesindedir. Fakat A/D çevrim sonucunun tamamı yerine bir sensör için 96, diğeri için de 104 birimlik bir aralığı kullandığımız için bu hata biraz daha fazla olacaktır. Analog ölçümdeki hatanın açığa en çok etkisi

X ve Y eksenleri üzerinde olur. Ayrıca ikinci sensör çıkışı daha az hassas olduğu için Y eksenindeki hatalar en büyük hatalardır. Mesela X bileşeninin ‘0’ olduğu bir durumda A/D çevrimden kaynaklanan hatanın da maksimum halini aldığını düşünelim. A/D çevrim sonucu en kötü ihtimalle 96 birim olarak kullandığımızdan yarım birimlik hatalı ölçüm sonucu formül-1’den  $0,597^\circ$  olarak bulunur. Yani pusulanın ölçüm hatası en kötü ihtimalde bile  $1^\circ$ den küçüktür. Bu hassasiyette bir pusula birçok uygulamada kullanılabilir.

Bu çalışma sonunda oluşturulan pusula modülünün performansını test etmek için yaptığımız deneylerde gerek ana yönlerde, gerekse ara yönlerde standart bir analog pusula ile yön farklılığı gözlenmemiştir. Ayrıca pusula modülünün köşelerinin 90 derece olmasından faydalanılarak sabit cisimlerin kenarları üzerinde de denemeler yapılmış ve sistemin gösterdiği değerler olması beklendiği gibi birbirinden 90 derece farklı bulunmuştur. Cihazın performansını daha objektif yöntemlerle sınamak için ise kalibrasyonu yapılmış özel sistemler ve ortamlar gerekmektedir.

Pusulanın çıkışına etki eden önemli bir hata kaynağı da yakındaki metal kütlelerdir[1]. Büyük metal kütlelere yakın ölçüm yapmamak en iyi çözümdür. Eğer pusulanın konumu gereği yakında sabit bir metal kütle bulunacaksa veri tablosu oluşturma aşamasında bu durum dikkate alınıp metalin etkisi giderilebilir.

Sonuç olarak magneto-resistive sensörler kullanılarak mekanik pusulalardan çok daha başarılı ve geniş kullanım alanı olan pusula sistemleri yapılabildiği bu çalışma ile gösterilmiştir.

#### 6. Teşekkür

Bu çalışmanın yapılması sırasında değerli desteğini esirgemeyen Doç. Dr. Onur Tokar’e ve Prof. Dr. Sadık Kara’ya teşekkürü bir borç bilirim.

#### 7. Kaynakça

- [1] Michael J. Caruso, Dr. Carl H. Smith, Tamara Bratland, Robert Schneider; A New Perspective on Magnetic Field Sensing
- [2] HONEYWELL; AN211 Applications of Magnetic Position Sensors
- [3] Philips, Honeywell, HL Planar; Üreticilerin Yayınladıkları Datasheet’ler
- [4] Intersil Corporation; Instrumentation Amplifier Application Note 1298
- [5] HONEYWELL; AN212 Handling Sensor Bridge Offset
- [6] HONEYWELL; AN213 Set/Reset Function For Magnetic Sensors
- [7] <http://www.fatih.edu.tr/~onur/PIC18F452/TinyPIC/>
- [8] [http://en.wikipedia.org/wiki/Magnetic\\_north](http://en.wikipedia.org/wiki/Magnetic_north)

# Telsiz Duyurga Ağları için Şifreleme ve Hata Tespiti İşlemlerinin Düşük Güç Tüketimiyle Gerçeklenmesi ve Ortak Analizi

Abdullah Usta<sup>1</sup>, Arda Yurdakul<sup>2</sup>

<sup>1</sup>Elektronik ve Haberleşme Mühendisliği Bölümü  
İstanbul Teknik Üniversitesi, İstanbul.  
ustaab@itu.edu.tr

<sup>2</sup>Bilgisayar Mühendisliği Bölümü  
Boğaziçi Üniversitesi, İstanbul.  
yurdakul@boun.edu.tr

## Özetçe

Telsiz Duyurga Ağları (TDA), dış dünyadan verilerin toplanması, işlenmesi, kümelenmesi ve iletilmesini kapsayan bir dizi işlemi gerçekleştirebilen sistemlerdir. Sınırlı pil ömrü ve kablosuz bağlantıya sahip birimlerden oluşan bu ağlarda, verilerin güvenli ve hatasız ulaştırılması, kullanım alanına bağlı olmak kaydıyla önemlidir. Bu çalışmada TDA üzerinde yapılması beklenen şifreleme ve hata tespiti gibi hesaplama işlemlerinin düşük güç tüketimiyle nasıl yapılabileceğini sunacağız. Düşük güç tüketimi için tasarlanmış şifreleme yöntemleri, hatasız veri aktarımı ile ilgili algoritmalar açıklanacak, değerlendirmeleri yapacak, bir model kurup üzerinde örnek çalışma yapacağız.

## 1. Giriş

TDA kullanılarak birbirinden ayrık sesnörlerin beraber çalışması ile sıcaklık, ses, titreşim, basınç veya hareket gibi fiziksel veya çevresel verilerin izlenmesi yapılabilir [1-2]. TDA'nın başlıca kullanıldığı bina, nakliye, endüstriyel otomasyon vb. bir çok alanda uygulama yapılırken, sistemde yer alan sensörlerin pil ömürlerinin kısıtlı olması, sistemin tasarımında enerji etkin bir yöntemin izlenmesi gerektiğini ortaya çıkarmaktadır.

TDA'nın düşük güç tüketimi ile çalışabilmesi için alınabilecek önlemler, ağda kullanılan sensör veya benzeri birimlerin mimarisi, ağ tasarımı ve hesaplamaların yapıldığı donanım veya yazılım seviyelerinde olabilir [3]. Bu seviyelerin tamamı hesaplama ve haberleşme adında ikiye ayrılabilir. Düşük güç tüketimine sahip sistem tasarımında optimize edilmeye çalışılan başlıca iki fonksiyon bunlardır.

TDA ilk olarak araştırmacılar tarafından doğal çevre ve hayvan topluluklarının yaşam alanları ile ilgili bilgi toplanması amacı ile kullanılmaya başlanmıştır. TDA benzeri bir teknolojiye sahip olan RFKT (Radyo Frekanslı Kimlik Tanımlama) sistemleri de aktif stok takibi amacıyla kullanılmaya başlanmıştır [4]. Bu iki sistemde de güvenlik kaygısından pek bahsedilemez. Bununla beraber her iki teknolojinin de artan uygulamaları ile birlikte güvenlik ihtiyacı duyulmaya başlanmıştır. Kimlik doğrulama, veri bütünlüğünün korunması ve gizlilik gibi başlıca güvenlik

gerekliklerinin yerine getirilebilmesinde kullanılan en temel yapı, şifreleme fonksiyonlarıdır. TDA içerisinde bu şifreleme fonksiyonlarının, diğer hesaplama birimleri gibi düşük güç tüketimi merkezli tasarlanması gerekmektedir. Bu alanda yapılan araştırmalara bakıldığında; ÜTD (Uygulamaya Özgü Tümdevre), APKD (Alan Programlamalı Kapı Dizisi) ve yazılım ortamlarında belirli önlemlerin alınması, mevcut şifreleme algoritmalarının düşük güç tüketimine sahip gerçeklemelerinin yapılması ve düşük güç tüketimine sahip yeni şifreleme algoritmalarının tasarlanması gibi çalışmaların yapıldığı görülmektedir[5-6].

Haberleşme, TDA'nın işlevini yerine getirmesini sağlayan en hayati fonksiyonlardan biridir. Haberleşme sisteminin başlıca görevi olan güvenilirliği belirli bir seviyede tutmaktır. Güvenirlik, aktarılan bilginin ve haberleşmenin yapıldığı ortamla doğrudan ilgilidir. Çalışma ortamları ve düşük güçle çalışan anten yapıları sebebiyle, TDA içindeki birimlerin kendi arasındaki haberleşmede kaynaktan çıkan bilgi ile alıcıya ulaşan bilgi aynı olmayabilir. Burada meydana gelen hata oranı, haberleşme kanalının işaret-gürültü oranı ile ters orantılıdır. Bu hatanın giderilmesi için daha fazla güç harcayarak işaret gürültü oranının artırılması yerine, güç tüketimini düşük tutmak için hata tespit ve/veya düzeltme kodları kullanılmaktadır. Bununla ilgili [7]'deki çalışmalarda farklı kodlama sistemleri kullanılarak aynı işaret gürültü oranlarına sahip ortamlarda elde edilen hata oranı değerleri görülmektedir.

Şimdiye kadar, düşük güç tüketimi odaklı şifreleme ve hata düzeltme kodları üzerine ayrı ayrı çalışmaların, analizlerin yapıldığı, bahsedildiği üzere bilinmektedir [4-7]. Bu çalışmada bu iki güç minimize etme araştırması, birlikte ele alınarak yapılmıştır. TDA gibi uygulama merkezli tasarıma uygun yapılarda, güç tüketimini azaltabilmek için, sistemin başlıca güç maliyetlerini doğuran etmenleri (bizim incelememizde şifreleme ve hata tespit/düzeltilme) beraber optimize etmeye çalışmak, ayrı ayrı ele alınan yaklaşımlara göre daha iyi sonuçlar verecektir. 2. bölümde düşük güç tüketimli şifreleme ve hata tespit/düzeltilme algoritmaları verilecek, değerlendirmeleri yapılacak ve bizim yaklaşımımız ifade edilecek, 3. bölümde iddia edilen yaklaşım ve kurulan modelle

ilgili bir örnek gerçekleştirme incelenecek, 4. bölümde ise sonuçlar verilecektir.

## 2. Yaklaşımlar

Düşük güç tüketimli şifreleme işlemi için yapılabileceklerden giriş bölümünde bahsedilmiştir. TDA kullanılarak sistem tasarımı yapılırken, uygulama özelinde yaklaşımda bulunulması, daha etkin güç tasarrufu yapılmasını sağlayacaktır. Bu sebeple mevcut algoritmaların özel gerçeklemleri yerine tamamen düşük güç tüketimi amaçlı tasarlanmış şifreleme algoritmalarına yönelmek daha iyi sonuçlar vermektedir.

### 2.1. Düşük güç tüketimli şifreleme

Düşük güç tüketimli ve az yer kaplayan popüler şifreleme algoritmalarından birisi TEA (Tiny Encryption Algorithm, küçük şifreleme algoritması)'dır [8]. Bu algoritma Feistel yapısında olup sadece xor, toplama ve kaydırma fonksiyonlarını kullanmaktadır. Veri Şifreleme Standardı (DES) ve İleri Şifreleme Standardı (AES) ile kıyaslandığında, bu algoritma S-kutusu veya P-kutusu gibi karmaşık yapılara ihtiyaç duymamaktadır. Anahtar uzunluğu 128-bit olup, 64-bit blok uzunluğuna sahiptir.

[9]'da yapılan çalışmada TEA algoritmasının üç farklı gerçekleştirme üzerine durulmuştur. Paralel, ardışıl ve seri gerçeklemleri yapılmıştır. Bu üç gerçeklemeden daha az güç tüketen ikisi ile İleri Şifreleme Standardının, RFKT etiketler için yapılmış AES-1 [10] ve akıllı kartlar için alan etkin yapılmış AES-2 [11] olmak üzere iki farklı gerçekleştiriminin kıyaslanması Tablo 1'de verilmiştir.

Tablo 1: TEA ve AES karşılaştırması [9]

	AES-1	AES-2	TEA Paralel	TEA Dijiit
Anahtar (bit)	128	534	128	128
Veri uzunluğu (bit)	128	128	64	64
Çevrim sayısı	10	10	32	32
128 bit şif. için kullanılan saat sayısı	1016	226	64	512
İşlem Türü	Şif.	Şif./Çöz.	Şif.	Şif.
Eşdeğer kapı sayısı	3,595	6,177	6,918	3,872
Serim alanı (mm <sup>2</sup> )	0,25	0,337	0,378	0,211
Makimum hız Mbps	9,9	11,33	48	6,25
Akım $\mu$ A @ 12.8 kbps	8,15	11,1	0,55	2,9
CMOS teknolojisi	0,35	0,35	0,35	0,35

Tablo 1'de görüldüğü üzere paralel TEA gerçekleştirme diğerlerinden belirgin bir şekilde daha az güç tüketmektedir. Bir diğer düşük güç tüketim amaçlı tasarlanan şifreleme algoritması ise PRESENT ("hazır,mevcut" anlamına gelen bu kelime, şifreleme bloğunun birçok uygulamaya uyumlu olduğunu göstermesi için tercih edilmiştir) isimli algoritmadır[12]. PRESENT algoritması, düşük güç tüketimine ihtiyaç duyan, donanımsal şifreleme yapması

beklenen, belirli bir seviye güvenliğinin (80-bit anahtar uzunluğu gibi) yeterli olduğu, çok büyük veri şifrelemesine ihtiyaç duymayan ve alan etkin olması beklenen sistemler için tasarlanmıştır. [12] PRESENT, TEA'nin aksine AES gibi SPN(Substitution and Permutation Network, Yer Değiştirme ve Kaydırma Ağı) yapısına sahiptir. [13]'te yer alan çalışmada, bazı blok şifreleyicilerin UÖTD gerçeklemlerinin hız, alan, güç vb. alanlarda kıyaslaması yapılmıştır, değerler Tablo 2'deki gibidir.

Tablo 2: PRESENT ve diğer blok şifreleyicilerin karşılaştırması

Tasarım	Saat çevrimi	Blok uzunluğu (bit)	Anahtar uzunluğu (bit)	Alan (Eşdeğer kapı(GE))	Hız/Alan (Kbps/GE)	Güç( $\mu$ W)	Bit başına enerji (pJ)
AES 8-bit[14]	534	128	128	4070	24,0	23,8	994
AES 8-bit[15]	1016	128	128	3595	12,6	26,9	2135
AES 8-bit[16]	1032	128	128	3400	12,4	4,5	363
DESXL[17,18]	144	64	184	2168	44,4	1,6	36
Camelia[19]	21	128	128	11350	609,5	-	-
Camelia[20]	44	128	128	6511	290,1	-	-
PRESENT-80[21]	32	64	80	1570	200,0	5,0	25
PRESENT-80[21]	32	64	128	1886	200,0	-	-

Tablo 2'de görüldüğü gibi, diğer blok şifreleyicilere göre PRESENT bir bitin şifrelemesinde çok daha az enerji harcamaktadır. Bu çalışma kapsamında PRESENT şifreleme algoritmasının APKD gerçekleştirme ve güç analizleri yapılmıştır. 3. bölümde gerçekleştirme sonuçları ifade edilecektir.

### 2.2. Düşük güç tüketimli hata tespit ve düzeltme

Düşük güç tüketimli TDA tasarımında, giriş bölümünde bahsedildiği üzere haberleşme çok önemlidir. Güvenirliği koruyabilmek için haberleşme esnasında oluşabilecek hatalara karşı, hatanın tespit edilip verinin yeniden iletilmesi veya hatanın alıcıda düzeltilmesi gibi farklı yöntemler kullanılabilir. Hangi yöntemin daha uygun olabileceği sorusunu sormadan önce, haberleşmede ne tür hataların olabileceğini göz önüne almak gerekir. Farklı hata çeşitleri ise haberleşme sisteminin birimleri ve ortama doğrudan ilgilidir. Bu çalışmada bir model ortaya koyabilmek için, haberleşmede bir bit hatanın oluştuğunu kabul ettik. Yapılan incelemeler ve güç optimizasyonu çalışmaları da bu kabul merkeze alınarak yapılmıştır. Farklı çeşitlerde hata durumlarında, benzer bir metodoloji kullanılarak uygun modeller oluşturulabilir.

Bir bit hatanın tespitinde kullanılacak en az güç maliyetli yöntem, verinin yanında verinin eşliğinin (paritesinin) de gönderilmesidir. Alıcı gelen verinin eşliğini hesaplayıp gelen eşlik değeri ile kıyaslayarak bir bit hatanın var olup olmadığını kontrol edebilir. Bu yöntem kullanılarak hatanın hangi bitte meydana geldiği bilinmeyeceği için hatanın alıcı tarafında düzeltilmesi söz konusu değildir.

Hata düzeltme kodları içinde en sık kullanılan yöntem Hamming Kodları'dır [22]. Hamming kodlama sisteminde taşınmak istenen verinin yanına, verinin bit sayısına bağlı sayıda kontrol bitleri eklenir. Bu kontrol bitleri kullanışlı veriden üretilmiştir. Kullanışlı veri ile kontrol verisi

birleştirilerek kod verisi meydana getirilir. Haberleşme esnasında iletilen veri budur. Alıcı bu veriyi aldıktan sonra kullanışlı veriden tekrar kontrol verisi elde edip, gelen kontrol verisi ile karşılaştırır. Aynı değerlerin elde edilmesi, haberleşme esnasında (bir bit) hatanın oluşmadığı anlamına gelir. Bu işlemle, haberleşmede meydana gelen bir bitlik hatalı iletimin tespiti yapılabilir. Ayrıca hatanın tespit edildiği bu iki kontrol işaretinin XOR işlemine tabi tutulması ile sendrom vektörü oluşturulur. Bu sendrom vektörü eğer bir bitin değeri alıcıya yanlış gelmişse ('1' yerine '0' veya '0' yerine '1'), o bitin, kod verisinin kaçınıcı biti olduğu bilgisini verir. Böylece o bitin değeri terslenerek gönderilmek istenen veri tekrar elde edilmiş, hata düzeltilmiş olur.

Hata düzeltme işlemi yapan devrenin karmaşıklığı, sistemin çalışmasında ek güç maliyetleri getirmektedir. Buna göre hata düzeltme yerine, hata tespit edildiğinde verinin tekrar gönderilmesinin talep edilmesi güç açısından daha uygun olabilir. Bu incelemenin yapılabilmesi için kullanılan kodlama sisteminin kodlama, kod çözüme ve düzeltme güç maliyetleri ile veri göndermenin güç maliyetlerinin birleştirilerek değerlendirilmesi gerekir.

### 2.3. Önerilen güç analiz modeli

Bu çalışmada hata tespit veya hata düzeltme sistemi kullanılarak tasarlanan farklı sistemlerin ihtiyaç duyacakları gücü kıyaslayabilmek için bir model geliştirilmiştir. Bu modelde haberleşme işleminde harcanan enerjinin bir bit için normalize edilmiş değerleri kullanılmıştır. İlerleyen bölümlerde bu değerler için aşağıdaki kısaltmalar kullanılacaktır:

- $E_S$  : Bir bitin gönderilmesi için harcanan enerji.
- $E_E$  : Bir bitin kodlanması için harcanan enerji.
- $E_D$  : Bir bitin kodunun çözülmesi için harcanan enerji (hata tespiti).
- $E_C$  : Bir bitin kodunun çözülüp hata düzeltilmesinin yapılması için harcanan enerji.

Haberleşmede kullanılacak sistemi tanımlayıp yukarıda ifade edilen enerji değerleri kullanılarak belirli boyutta bir verinin doğru şekilde gönderilmesi için ne kadar enerji harcanacağı hesaplanabilir.

- Birinci Senaryo

Birinci sistemde kurulan haberleşme senaryosu, verinin eşliğinin hesaplanıp gönderilmesi ve alıcı tarafında tekrar eşliğinin hesaplanıp kontrol edilerek hatanın tespit edilmesi şeklinde olsun. Hatalı verinin gönderici tarafından tekrardan gönderilmesi istensin. Burada verinin tekrar gönderilmesi için fazladan haberleşme ve gönderici tarafında fazladan bellek ihtiyaçları ekstra güç harcaması gerektirecektir. Fakat bu değişkenlerin modele alınması, modelin karmaşıklığını artıracığından bu çalışmada ihmal edilmiştir.

Gönderilmek istenen verinin paketler halinde gönderildiğini varsayalım.  $p$  adet veri paketinin gönderilmesi için harcanması gereken enerji hesaplanmak istensin.  $p$  adet paketin gönderilmesinde (ardı ardına da olabilmesi şartıyla)  $m$  adet paketin hatalı gittiğini varsayalım. Bu senaryoda  $n$  bitlik veri paketleri yanlarına bir bitlik eşlikleri de eklenerek  $n+1$  bitlik  $c$ , kod verisi oluşturulur. Buna göre, bu sistemde gönderilmek istenen  $p$  paketlik, diğer bir ifade ile  $p*n$  bitlik veri için harcanması gereken enerji Denklem 1'deki gibi olur.

$$c = n + 1$$

$$E_{ENC} = (p + m) * c * E_E$$

$$E_{SEND} = (p + m) * c * E_S$$

$$E_{DEC} = (p + m) * c * E_D$$

$$E_{TOPLAM} = E_{ENC} + E_{SEND} + E_{DEC} \quad (1)$$

- İkinci Senaryo

İkinci bir senaryo olarak eşlik kontrolü yerine kodlama sistemi olarak Hamming kodlama sistemi kullanılsın ve hatalı paketler birinci senaryoda olduğu gibi tekrar gönderilsin. Burada kullanılacak kod verisi içerisinde gerçek veri ile  $k$  bitlik kontrol verisi olacağı için, kod verisinin boyutu

$$c = n + k \quad (2)$$

bit şeklinde olacaktır.  $p$  paketlik verinin iletilmesi için ihtiyaç duyulan toplam enerji miktarı yine Denklem 1 kullanılarak hesaplanabilir.

- Üçüncü Senaryo

Önerilen üçüncü ve son haberleşme sistemine göre, veri ikinci senaryodaki gibi Hamming kodlama sistemi ile kodlansın ve alıcıda bir bit hata düzeltme uygulansın. Buna göre  $p$  paketin gönderilmesi için harcanması gereken toplam enerji Denklem 3'teki gibi olur.

$$c = n + k$$

$$E_{ENC} = p * c * E_E$$

$$E_{SEND} = p * c * E_S$$

$$E_{CORR} = p * c * E_C$$

$$E_{TOPLAM} = E_{ENC} + E_{SEND} + E_{CORR} \quad (3)$$

Önerilen bu üç model kullanılarak, bir uygulamanın karşı karşıya olduğu bit hata oranına, gerçekleştiği ortamda hata kodlama, kod çözüme ve hata düzeltme devrelerinin yapısına bağlı olarak farklı güç tüketim değerleri hesaplanabilir. Buna göre o uygulama için hangi sistemin daha uygun olduğu tercih edilebilir. 3. bölümde bu çalışma kapsamında yapılan bir gerçekleştirme ve modelin uygulaması yer almaktadır.

## 3. Uygulama

Bu bölümde, 2. bölümde bahsi geçen yaklaşımlar ve modellerin uygulaması sonucu elde edilen sonuçlar paylaşılacak ve yorumlanacaktır. Çalışma kapsamında PRESENT şifreleme algoritması, eşlik hesaplama, Hamming (21,16) kodlama, kod çözüme ve hata düzeltme devrelerinin APKD gerçeklemeleri yapılmıştır. Xilinx Spartan XC3S700A APKD modeli kullanılarak, "ISE XPower Analyzer" güç analiz programı yardımı ile devrelerin ne kadar güç tüketeceği hesaplanmıştır. Buna göre gerçekleştirilmesi yapılan devrelerin güç, hız ve alan değerleri Tablo 3'te yer almaktadır.

Tablo 3: Gerçekleşmesi yapılan devrelerin alan, hız ve güç değerleri

	Alan [Slice]	Devrenin hızı [MHz]	Güç [mW] @ 50 Mhz	Veri işleme hızı [Mbps]	Bir bit veri işlemek için gerekli enerji [fJ]
PRESENT	357	130,68	15,03	253	1186
Eşlik Hesaplama	4	314,66	0,17	5035	0,68
Hamming (21,6) kodlama	6	291,97	0,37	4672	1,58
Hamming (21,6) kod çözme	9	156,79	0,35	2509	2,79
Hamming (21,6) hata düzeltme	22	147,49	0,35	2360	2,97

Tablo 3'te yer alan veriler incelendiğinde, gerçekleşmesi yapılan devrelerin ne kadar yer kapladığı, hangi hızlarda çalışabildiği ve 50 MHz'de çalışırken ne kadar güç tüketeceği ile ilgili benzetim değerleri görülmektedir. 33 saat çevriminde 64 bit şifreleme yapan PRESENT devresinin, bir saat çevriminde 16 bitlik verinin eşlik hesabı, Hamming kodlama, Hamming kod çözme ve Hamming hata düzeltme yapan devrelerin güç değerleri tablodaki gibidir. Buradaki güç değerleri içinde giriş çıkış padlerinin güç değerleri dâhil edilmemiştir. Sadece gerekli fonksiyonu yerine getiren lojik yapının dinamik güç tüketim değeri kullanılmıştır. Dolayısıyla bu güç değerlerini, sistemlerin birbirleri ile kıyaslanması anlamında düşünmek gerekir.

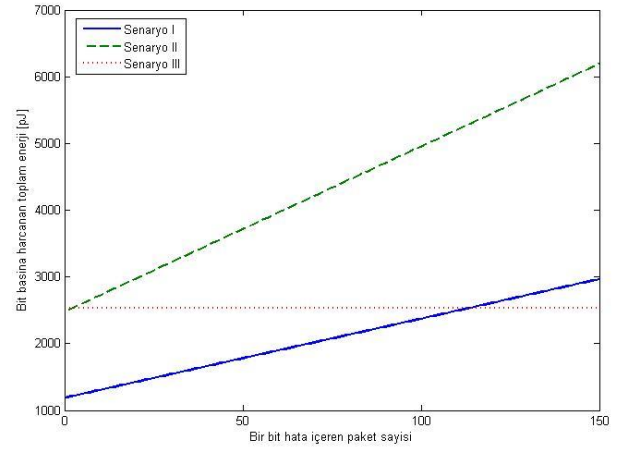
Tablo 3'te ifade edilen bir bit veri işlenmesi için gereken enerji değerleri kullanılarak 2. bölümde tanımlanan güç modeli kurulabilir. Bu model sayesinde yine 2. bölümde ifade edilen üç farklı haberleşme senaryosu için güç ihtiyaçları hesaplanabilir. Buna göre tablo 3'ten yararlanılarak güç modelinde kullanılacak değerler Tablo 4'teki gibidir.

Tablo 4: İşlemler için bit başına harcanan enerji

	Eşlik Hesaplama	Hamming (21,6) kodlama	Hamming (21,6) kod çözme	Hamming (21,6) hata düzeltme
$E_E$ [pJ]	0,68	1,58	-	-
$E_D$ [pJ]	0,68	-	2,79	-
$E_C$ [pJ]	-	-	-	2,97

Güç modelinde Tablo 4'teki enerji değerlerinin yanı sıra, verinin gönderilmesi için ihtiyaç duyulan enerji  $E_S$ 'nin de girilmesi gerekir. Bu değer ortalama ilgilidir ve  $E_S$ 'nin diğer enerji bileşenleri ile arasındaki büyüklük ilişkisine bağlı olarak, güç modelinde yer alan senaryolardan biri diğerinden daha az güç tüketimi yapabilir.

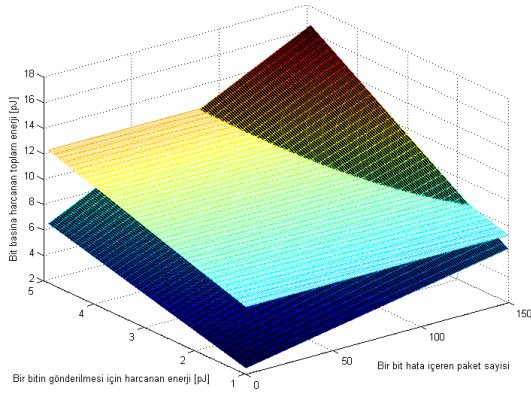
Bir örnek olarak  $E_S = 3$  pJ alınsın. 16 bitlik veri paketlerinden 100 paketin güvenilir şekilde gönderilmesi istensin. Bir bitlik hatanın kaç defa gerçekleşeceğine göre bu senaryolarda güç tüketimi değişmelidir. Bu duruma göre, güç modeli kullanılarak hesaplanan farklı güç eğrileri, Şekil 1'de verilmiştir.



Şekil 1: Değişen hata oranına göre, farklı senaryoların enerji tüketim değerleri.

Şekil 1'de görüldüğü üzere, veri göndermek için  $E_S$  sabit alındığında hata oranı arttıkça, 1 ve numaralı senaryoda, veri tekrar gönderildiği için harcanan enerji miktarı artmaktadır. Fakat 2 numaralı senaryoda kullanılan Hamming kodlama ve kod çözme eşlik hesaplamadan daha fazla güç tükettiği için, 1 numaralı senaryodan daha fazla güç tüketim değeri görülmektedir. 3 numaralı senaryonun güç tüketim değerine baktığımızda ise harcanan enerjinin hata sayısı ile değişmediği görülmektedir. Bu senaryoda gelen tüm veri doğrulama işlemine tabi tutulduğu için, verinin hatalı olup olmaması, harcanan enerjiyi değiştirmiyor. Bu durumda en az güç tüketimine ihtiyaç duymasını beklediğimiz sistem belirli bir hata oranına kadar 1 numaralı sistem olurken, o hata değerinden sonra 3 numaralı sistem daha az güce ihtiyaç duymaktadır.

Şekil 1'deki grafikte sabit gönderme enerjisi için durum değerlendirildi, fakat farklı  $E_S$  değerleri için birinci senaryoya göre artan hata oranı, harcanan enerjiyi de artıracaktır. Böylece denge yine değişecektir. Buna göre değişen  $E_S$  ve hata (m) değerleri için 1 ve 3 numaralı sistemlerin bit başına harcanan enerji değerleri Şekil 2'deki gibi olur.



Şekil 2: Değişen hata oranı ve  $E_s$  değerine göre, 1 ve 3 numaralı senaryoların enerji tüketim değerleri.

Şekil 2’de görüldüğü üzere 3 numaralı sistemin güç tüketimi, sadece  $E_s$  ile (lineer) değişirken, bir numaralı sistem hem  $E_s$ , hem de hata değerinin yükselmesi ile artmaktadır. Buna göre en az güç tüketim değerini veren, Şekil 3’te görüldüğü gibi, düşük hata ve  $E_s$  değerleri için 1 numaralı sistem, yüksek hata ve  $E_s$  değerleri için 3 numaralı sistem olmaktadır.

#### 4. Sonuçlar

TDA tasarımında güç tüketiminin önemi gittikçe artmaktadır ve buna göre tasarımcılar, sistemin çalışacağı koşullara bağlı olarak uygulama özelinde çözümler sunmalıdır. Biz bu çalışmamızda, Telsiz Duyarga Ağlarında şifreleme ve hata tespiti işlemlerinin nasıl daha az güç tüketimi ile yapılabileceğinin araştırılmasında kullanılacak bir model önerdik. Çalışmamızda ön plana çıkardığımız nokta, şifreleme ve hata tespiti gibi güç tüketimi anlamında şimdiye kadar ayrı ayrı değerlendirilen işlemlerin, ortak değerlendirilerek güç tüketim değerlerinin düşürülmeye çalışılması oldu. Bu model, temelinde güç tüketimi bileşenlerinin ayrı ayrı değerlerinin elde edilip, birbirine alternatif sistemlerde bu değerlerle elde edilen toplam gücün ortam parametrelerine bağlı olarak incelenmesi ile oluşturuldu. Modelde kullanılan değişkenler, farklı fonksiyon gerçekleştiren veya farklı ortamda çalışması beklenen sistemler için değiştirilerek, o şartlar için de güç minimizasyonunda kullanılabilir. Hatta bu çalışmada ihmal edilen bazı parametrelerin de modele eklenmesi ile çok daha hassas sonuçlar veren bir model de oluşturulabilir.

#### 5. Kaynakça

[1] Römer, Kay, Friedemann Mattern (December 2004), "The Design Space of Wireless Sensor Networks", IEEE Wireless Communications 11 (6): 54–61, doi:10.1109/MWC.2004.1368897

[2] Lewis, F. L., "Wireless Sensor Networks", Smart Environments: Technologies, Protocols, and Applications edited by Cook, D.J. and Das, S.K., John Wiley, New York, USA, 2004.

[3] Rex Min, Manish Bhardwaj, Seong-Hwan Cho, Amit Sinha, Eugene Shih, Alice Wang, and Anantha

Chandrakasan, "Low-Power Wireless Sensor Networks", VLSI Design 2000, January 2001.

[4] Kaps, J. -P. "Cryptography for Ultra-Low Power Devices", unpublished thesis (PhD), Worcester Polytechnic Institute, 2006

[5] C. Rolfes, A. Poschmann, G. Leander, and C. Paar, "Ultra-Lightweight Implementations for Smart Devices-Security for 1000 Gate Equivalents", The 8th Smart Card Research and Advanced Application IFIP Conference - CARDIS 2008, LNCS 5189, G. Grimaud and F.-X. Standaert (eds.), Berlin, Germany: Springer-Verlag, pp. 89-103, 2008.

[6] A. C. Atici, L. Batina, J. Fan, I. Verbauwhede, and S. B. Ors, "Low-cost Implementations of NTRU for pervasive security", In 19th IEEE International Conference on Application-specific Systems, Architectures and Processors (ASAP), IEEE, pp. 79-84, 2008.

[7] S. L. Howard, C. Schlegel, and K. Iniewski, "Error control coding in low-power wireless sensor networks: when is ECC energy-efficient?" EURASIP Journal on Wireless Communications and Networking, pp. 1–14, 2006.

[8] Wheeler D.J., Needham R.M. "TEA, a tiny encryption algorithm." In: Second International Workshop on Fast Software Encryption. Lecture Notes in Computer Science 1008, pp. 363–366, 1994

[9] P. Israsena and S. Wongnamkum, "Hardware Implementation of a TEA-Based Lightweight Encryption for RFID Security", 2008

[10] Feldhofer M., Dominikus S., Wolkerstorfer J. (2004) Strong authentication for RFID systems using the AES algorithm. In: Cryptographic Hardware and Embedded Systems. Lecture Notes in Computer Science 3156, pp. 357–370

[11] Pongjit J. (2003) Power–Area efficient advanced encryption standard IP core targeting smart card applications. Master Thesis, Asian Institute of Technology

[12] A. Bogdanov et al., "PRESENT: An Ultra-Lightweight Block Cipher," Proc. Workshop Cryptographic Hardware and Embedded Systems (CHES 07), LNCS 4727, Springer, 2007, pp. 450-466.

[13] J.-P. Kaps, "Chai-Tea, Cryptographic Hardware Implementations of xTEA", The 9th International Conference on Cryptology in India - INDOCRYPT 2008, LNCS 5356, D.R. Chowdhury, V. Rijmen, and A. Das (eds.), Berlin, Germany: Springer-Verlag, pp. 363-375, 2008.

[14] Kaps, J.P., Sunar, B.: Energy comparison of AES and SHA-1 for ubiquitous computing. In et al., X.Z., ed.: Embedded and Ubiquitous Computing (EUC-06)

Workshop Proceedings. Volume 4097 of LNCS., Springer (Aug 2006) 372–381

- [15] Feldhofer, M., Dominikus, S., Wolkerstorfer, J.: Strong authentication for RFID systems using the AES algorithm. In: Cryptographic Hardware and Embedded Systems – CHES 2004. Volume 3156 of LNCS., Springer (Aug 2004) 357–370
- [16] Feldhofer, M., Wolkerstorfer, J., Rijmen, V.: AES implementation on a grain of sand. Information Security, IEE Proceedings 152(1) (Oct 2005) 13–20
- [17] Leander, G., Paar, C., Poschmann, A., Schramm, K.: New lightweight DES variants. In: Fast Software Encryption, FSE 2007. Volume 4593 of LNCS., Springer (2007) 196–210
- [18] 24. Poschmann, A., Leander, G., Schramm, K., Paar, C.: New light-weight crypto algorithms for RFID. In: International Symposium on Circuits and Systems (ISCAS). (May 2007) 1843–1846
- [19] 25. Aoki, K., Ichikawa, T., Kanda, M., Matsui, M., Moriai, S., Nakajima, J., Tokita, T.: Camellia: A 128-bit block cipher suitable for multiple platforms – design and analysis. In: Selected Areas in Cryptography, SAC 2000. Volume 2012 of LNCS., Springer (2001) 39–56
- [20] Satoh, A., Morioka, S.: Hardware-focused performance comparison for the standard block ciphers AES, Camellia, and Triple-DES. In: 6th Information Security Conference, ISC'03. Volume 2851 of LNCS., Springer (2003) 252–266
- [21] Bogdanov, A., Knudsen, L., Leander, G., Paar, C., Poschmann, A., Robshaw, M., Seurin, Y., Vikkelsoe, C.: PRESENT: An ultra-lightweight block cipher. In: Cryptographic Hardware and Embedded Systems – CHES 2007. Volume 4727 of LNCS., Springer (2007) 450–466
- [22] [http://en.wikipedia.org/wiki/Hamming\\_code](http://en.wikipedia.org/wiki/Hamming_code)



# Gömülü Sistemlerde Medya Uygulamaları ve Performans Media Applications on Embedded Systems and Performance

Halid Ziya Yerebakan<sup>1</sup>, İsmail Uzun<sup>2</sup>

<sup>1</sup>Elektrik- Elektronik Mühendisliği Bölümü  
Yüksek Lisans Öğrencisi  
Boğaziçi Üniversitesi, İstanbul  
halidziya@gmail.com

<sup>2</sup>C Tech Bilişim  
Gebze, Kocaeli  
ismail.uzun@ctech.com.tr

## Özetçe

Gömülü sistemlerde kullanılan teknolojilerin gelişmesine paralel olarak uygulama alanları da sürekli genişlemektedir. Günümüzde gömülü sistemler özellikle işlemci gücü ve hafıza birimlerinin gelişmesi ve birçok medya arayüzlerini desteklemesi neticesinde üzerinde işletim sistemi bulunan küçük birer bilgisayar halini almıştır. Bu gelişmelere paralel olarak gömülü sistemler medya (ses ve video) uygulamalarında yaygın olarak kullanılmakta ve medya teknolojilerinin kullanımına farklı boyut kazandırmaktadır. Bu çalışmada örnek olarak seçilen gömülü sistem üzerinde Linux tabanlı işletim sistemi koşturulmuş ve başta medya uygulamaları dahil olmak üzere farklı uygulamaların işlemci gücü gereksinimleri ve hafıza kullanım durumları analiz edilerek karşılaştırılmıştır. Yapılan çalışmanın akademik alanda çalışan araştırmacılara ve endüstriyel uygulama geliştiriciler için sistem ve uygulama tasarımlarında yararlı olacağını umuyoruz.

## Abstract

Embedded systems and their application area are expanding continuously based on evolution of underlying technology. These recent evolutions lead to have powerful embedded systems in terms of processor speed and memory. New power input approximates embedded systems as a small personal computer which is capable to maintain operating systems. Furthermore, embedded systems broaden their usage including media (speech and video) applications and bring a different aspect to multimedia domain. In this paper, a sample embedded system is used for running Linux based operating system and speech/video applications to analyze and compare processor, memory usages and power consumptions. This study may be a good reference for future studies both for academicians and industry professionals while designing a system or developing an application.

## 1. Giriş

Gömülü sistemler, aynı yapı ile bir veya birden fazla bilgisayar işlevini gerçekleştiren sistemler olarak tanımlanmaktadır. [1],[2]. Bilgisayarlar ile ilgili teknolojilerin gelişmesine bağlı olarak gömülü sistemleri artık gömülü bilgisayar sistemi halini almıştır. Gömülü sistemlere örnek olarak Elektronik ev eşyaları (mikrodalga fırın, çamaşır makinesi, TV, DVD çalar), savunma sistemleri, uçaklarda ve füzelerde kullanılan uçuş kontrol sistemleri ve endüstriyel otomasyon ve izleme sistemlerindeki PC'ler gösterilebilir [2].

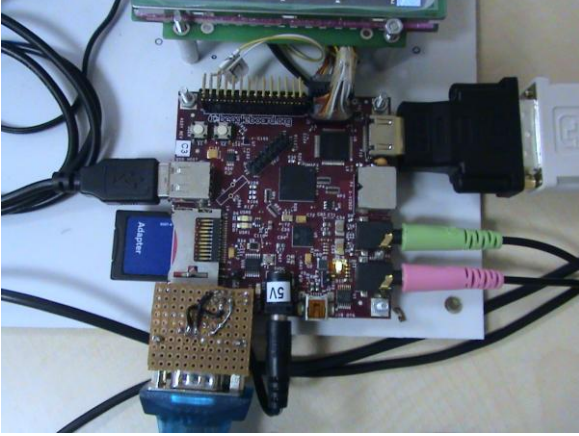
Günümüzde gömülü sistemler artık IP üzerinden ses iletimi (Voice Over IP - VoIP) uygulamalarında hatta IP üzerinden video transferlerinde de kullanılabilir hale gelmiştir [3],[4],[5]. Gömülü sistemlerde uygulamalar çeşitlendikçe ve taşınabilirlik (mobility) ihtiyacı arttıkça güç kullanımı çok daha önemli bir gereksinim haline gelmektedir.

Bu çalışmada IP üzerinden iletilen video verisini alıp çözümleyebilen ve bunu HDMI ekranda ya da LCD ekranda gösterebilen bir gömülü sistem uygulamasının güç ve işlemci yükü sonuçları ölçülmüş ve paylaşılmıştır. Çalışmada gömülü sistem donanımı olarak BeagleBoard (rev. C3) [6] kullanılmıştır. Beagleboard, üzerinde 600MHz ARM mikro işlemci bulunduran OMAP3530 tabanlı bir işlemciye sahiptir [7], tek karttan oluşan bir gömülü sistemdir. Arayüz olarak HDMI, USB, MMC ve ses arayüzleri mevcuttur. USB arayüzünü bir hub yardımı ile genişletmek mümkündür. Rev C3 ve sonrasında, LCD arayüzü ile doğrudan ekrana bağlanabilir. 14 pin JTAG bağlantısı ile donanım seviyesine yakın bir şekilde hata ayıklamaya olanak sağlar. Grafik arayüzü kullanılmadığı takdirde RS232 bağlantısı ile konsol kullanılabilir. Bütün bu özellikleri ile BeagleBoard işlemcisini, anakartını, ses kartını ve ekran kartını bir kartta barındıran küçük bir bilgisayarı andırmaktadır.

Bu çalışmada, gömülü sistem üzerinde koşturulan uygulamaların gerektirdiği işlemci gücü ölçümü için sistem üzerinde voltaj ve akım ölçümleri alınarak harcanan güç hesaplanmıştır.

## 2. Gömülü Sistem

BeagleBoard 2A kadar akım verebilen 5V güç kaynağı ile çalışmaktadır. Alternatif olarak USB'den güç almaktadır ancak USB yüksek akım veremeyeceği için tercih edilmemektedir. Bununla birlikte bir adet USB hub ile USB sayısını artırarak klavye, fare ve ağ dönüştürücüsü takılabilmektedir. 600 MHz'e çıkabilen işlemci hızı ve 256 MB RAM sayesinde bir gömülü sisteme göre gayet yüksek bir performans elde edilebilmektedir. Test ortamı için DVI-HDMI bağlantı kablosu ile BeagleBoard'ın DVI çıkışı monitöre bağlanmaktadır. Ses testleri için gerektiğinde mikrofon ve kulaklık da takılabilmektedir. Gerekli bağlantılarının olduğu örnek test sistemi Şekil 1'de gösterilmiştir.



Şekil 1: Beagleboard ve test ortamı

Bu çalışmada da BeagleBoard üzerine işletim sistemi olarak açık kaynak kodlu bir işletim sistemi olan Angström Linux dağıtımını seçilmiştir [8]. Bu dağıtımın önemli bir avantajı gömülü sistemlere destek vermesi ve BeagleBoard üzerinde yaygın olarak kullanılmasıdır. Açık kaynak kodlu işletim sisteminde kodları inceleme ve değiştirme hakkınız olduğu için araştırma geliştirme için en uygun sistem açık kaynak kodlu sistemler olarak değerlendirilebilir.

Donanım dışında sistemin kullanılabilmesi için bazı yazılımsal ayarların yapılması gereklidir. Sistemin boot olabilmesi için öncelikle bir sistem yükleyicinin kurulması gereklidir [9]. Uboot adlı yükleyici bu iş için uygundur. Bu yükleyici BeagleBoard'ın kendi NAND hafızasına yüklenmesi kolay kullanım açısından önemlidir. Fakat yüklenecek dosya sisteminin MMC gibi harici bir hafıza biriminden alınması ilerde yaşanması muhtemel kapasite problemlerini önleyecektir. Bu ayarlar yükleyicinin çevre değişkenleri tarafından belirlenir. İstenilen konfigürasyonlara göre bu çevre değişkenlerini değiştirmek mümkündür.

Şekil 2'de Uboot yükleyicisinin bir ekran görüntüsü verilmiştir. Şekilde Uboot yükleyicisinin konsolu açılmıştır. Buradan çeşitli ayarlar yaparak el ile dosya sistemi mümkün olmakla birlikte, otomatik olarak *bootcmd* çevre değişkenini bir betik gibi çalıştırmak daha pratik olacaktır. *printenv* komutu ile çevre değişkenleri de görülebilir.

```
Texas Instruments X-Loader 1.41

Starting OS Bootloader...

U-Boot 1.3.3 (Jul 10 2008 - 16:33:09)

OMAP3530-GP rev 2, CPU-OPP2 L3-165MHZ

OMAP3 Beagle Board + LPDDR/NAND

DRAM: 128 MB

NAND: 256 MiB

In: serial

Out: serial

Err: serial

Audio Tone on Speakers ... complete

OMAP3 beagleboard.org #
```

Şekil 2 : Uboot yükleyicisi ve ekran çıktısı

## 3. Medya Uygulamaları

Çalışmada uygulamaların en güncel hallerini test edebilmek için Angström işletim sistemi üzerinde çalışan uygulamalar güncelleştirilmiştir. Güncelleme için BeagleBoard'ın internet bağlantısı USB-Ethernet dönüştürücü ile sağlanmıştır. IP üzerinden video transferi için de gömülü sistemimizin internete bağlı olması ve gerekli video kodeklerin bulunması gerekmektedir. Sistem ile ilgili bu ayarlamalar yapıldıktan sonra görüntü yayını yapan sunucunun IP adresinde akışa erişilebilir.

Ayrıca bu çalışma kapsamında geliştirilen bir I/O (Giriş/Çıkış) sürücüsü ile herhangi bir tuşu açma kapatma komutu olarak sisteme tanıtmak mümkündür. Bu şekilde uygulamaların ses/video iletimine başlaması ve durdurulması daha rahat kontrol edilebilmektedir.

### 3.1. Gstreamer

Gstreamer birçok kodek destekleyen, API yada uygulama olarak kullanılabilen konsol üzerinde çalışan bir medya aracıdır. Boruyolu (pipeline) mimarisi sayesinde çok çeşitli medya kodlama ve çözme kombinasyonları yapılması mümkündür. Bu mimaride bulunan görüntü kaynakları (sources), ara elemanlar (bin) ve alış noktaları (sink) sayesinde hiç kodlamaya ihtiyaç duymadan medya biçimlerini çözümlenmek mümkündür. Gstreamer uygulamasını Angström işletim sistemi üzerinde paket yükleyicisi (opkg) ile yüklemek mümkündür. Şekil 3'de Gstreamer uygulamasının çalıştırılması ile ilgili komutlar gösterilmiştir.

Şekilde görüldüğü gibi IP tabanlı görüntü aktarımı için öncelikle *rtsp* kaynağından gelen ağ akışının *rtspsrc* ile yakalanması ardından *rtmp4vdepay* ara elemanı ile çözümlenmesi gerekiyor. Alınan video *mp4* formatında olup otomatik çözümlenme yapan *decodebin* elemanı ile çözümlendikten sonra *xwindows* penceresinde gösterilmek üzere *xvimagesink* elemanına gönderiliyor.

```
gst-launch rtspsrc
location=rtsp://192.168.2.248:8554/mpeg4 !
rtppmp4vdepay ! decodebin ! xvimagesink
sync=false display=:0.0
```

Şekil 3: Gstreamer çalıştırılması

### 3.2. Aplay

Sadece ses aygıtını test etmek için alsa araçlarından [10] *aplay* ve *arecord* programları kullanılabilir. Bu programların yardım metinlerinden parametre detayları öğrenilebilir. Parametre olarak dosya formatı, örnekleme frekansı ve dosya ismi belirtilmelidir. Basit bir mikrofonlu kulaklık yardımı ile arecord ile kaydettiğiniz “.wav” dosyasını aplay ile çalabilirsiniz. Bu uygulama için 44100 Hz frekansını tavsiye ederiz. BeagleBoard 44100 Hz frekansında sorunsuz bir şekilde çalışmaktadır.

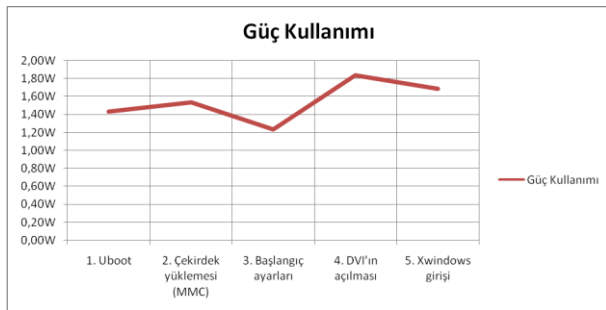
## 4. Performans Ölçümleri

### 4.1. Test Ortamı ve Testler

Bu çalışmada, Linux tabanlı bir işletim sistemi olan Angström üzerinde koşutulan bazı açık kaynak kodlu uygulamaların gerektirdiği işlemci gücü ölçümü için sistem üzerinde voltaj ve akım ölçümleri alınarak harcanan güç hesaplanmıştır. Ölçümlerimizi açılış sırasında ve kullanıcı etkileşimi gerekli olduğu anlardaki durumu değerlendirerek elde ettik.

### 4.2. Ölçüm Değerleri

Güç kullanımını uygulamalar için gerekli voltaj ve çekilen akım ölçülerek hesap edildi. Ölçüm değerlerinin daha sağlıklı olması için 3 ölçüm yapılmış ve ortalaması alınmıştır. Çevre birimlerine (arayüzlere) verilen güç ile birlikte standart bir sistem açılışında güç değişimi Şekil 4’te gösterilmiştir. Şekilde görüldüğü gibi ilk açılış esnasında gömülü sistemimizin 1,5W civarında güç tüketmeye başlayıp bu tüketim grafik ekranına geçince biraz daha yükseliyor. Serbest durumda grafik ekranı açık iken güç tüketimi 2W değerine çok yakın ve uygulamaların ne kadar güç tükettiğini gözlemlemek için bu değer referans alınabilir.



Şekil 4. Gömülü sistemin açılış sırasında harcadığı güç değerleri

Grafik ekran açık durumda iken çeşitli uygulamalar için elde edilen uygulama güç tüketimi de Tablo 1’de gösterilmiştir. Bu değerler Şekil 4’de elde edilen güç değerleri ile aynı donanım konfigürasyonlarında elde edilmiştir.

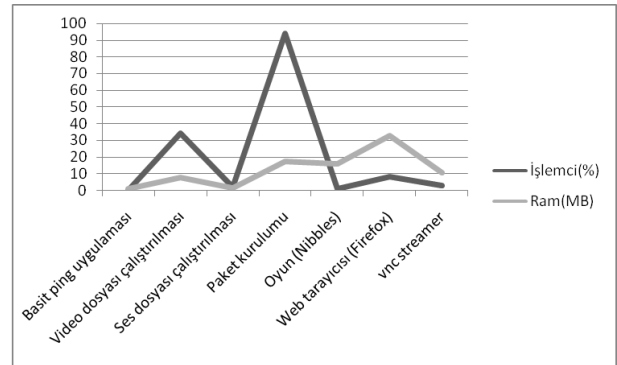
Tablo 1 : Çeşitli uygulamalar için elde edilen uygulama güç tüketimi (Grafik ekran açık durumda)

Uygulama	Güç Kullanımı (W)
Gnome açılışı	2,159
Basit ping uygulaması	2,15985
Video dosyası çalıştırılması	4,1002
Ses dosyası çalıştırılması	4,0426
LED'in yakılması	2,1844
Paket kurulumu	2,25552
Oyun (Nibbles)	2,17503
Web tarayıcısı (Firefox)	2,15475
VNC akışı	2,1801

Tabloda görüldüğü gibi ses çıkışının olmadığı uygulamalarda güç tüketimi 2W civarında seyrederken ses çıktısı verildiği takdirde güç tüketimi iki katına çıkmaktadır. Video uygulamasında yine ses çıktısı da olduğundan ses uygulamasına yakın bir sonuç elde ediyoruz. Fakat işlem yükünün biraz daha fazla olması neticesinde en yüksek güç tüketimini burada yapıyor. Paket kurulumu en fazla güç harcayan uygulama olarak gözlemlenmektedir. Ayrıca uzaktan kontrol yazılımı ve VNC sunucusu ile gömülü sistemi kontrol ettiğimizde çok büyük güç artışının olmadığını görüyoruz.

Bu çalışma da güç ölçümleri dışında Linux altında *top* aracı kullanılarak elde edilmiş işlemci (CPU) gücü ve hafıza (RAM) kullanımları da gözlemlenmiştir.

Şekil 5’te işlemci gücü yüzde olarak ve hafıza kullanımı MB cinsinden belirtilmiştir. Şekilde görüldüğü gibi uygulama ağacından bir programın kurulması işlemcinin neredeyse tamamının kullanılmasını gerektirmektedir. Bunun dışındaki uygulamalarda, web tarayıcısı ve Gstreamer diğer uygulamalara oranla yüksek miktarda işlemci gücü kullanan uygulamalardır. Hafıza kullanım oranı web tarayıcısında 30 MB ile en yüksek değeri almaktadır. Farklı web tarayıcılarında bu değer değişebilir.



Şekil 5. Uygulamaların işlemci ve hafıza kullanım oranları

## 5. Değerlendirme

Kullanılan gömülü bir sistemde çevre aygıtlarının çektiği güç miktarı uygulamalara göre oransal olarak daha fazladır. Uygulamalar arasında ise web tarayıcısı güç ve hafıza gereksinimi en yüksek olan uygulama olarak görülmektedir. Video uygulaması için mevcut hazır bir uygulama kullanılmış ve güç/hafıza gereksinimleri elde edilmiştir. Uygulamanın içeriği hafıza ve güç tüketimi açısından değişkenliğe sebep olacak olsa da genel bir fikir vermesi adına standart kullanımlar referans alınabilir. Ayrıca analizlerden anlaşılacağı üzere sistemin ses çıkış aygıtlarını iptal etmek batarya kullanım süresini uzatmak için mantıklı bir çözüm olacaktır. İlerleyen çalışmalarda uygun kapasitede bir batarya ile gömülü sistemimizin ne kadar süre kullanılabilceği, iletişimi kablosuz hale getirdiğimiz takdirde artan güç ihtiyacının ne oranda olacağı gibi analiz ve çalışmalar hedeflenmektedir. Ayrıca, kablosuz iletim arayüzlerini kullanan ve medya iletimi yapan bir sistemin ekonomikliği de araştırılabilecektir. Bununla birlikte bu şekilde bir gömülü sistemde LCD ekran modülü ve çevre aygıtları ile sistemin güç tüketimi ve batarya ömrüne etkisi muhtemel araştırma konularıdır.

## 6. Teşekkür

Bu çalışmaya yardım ve desteklerinden dolayı Uğur Yıldırım'a teşekkür ederiz

## 5. Kaynakça

- [1] Heath, S., *Embedded Systems Design*, Second Edition, Elsevier Science, 2003.
- [2] Kaya, Ahmet., "Gömülü Sistemler ve Uygulama Alanları", *GÖSİS'08, Gömülü Sistemler ve Uygulamaları Sempozyumu, 3-4-5 Kasım 2008, İstanbul*.
- [3] <http://www.soekris.com/net5501.htm>
- [4] <http://sourceforge.net/projects/porting-voip/>,
- [5] Yijiang, Y., Fiyu, L., "Design and Implementation of an EmbeddedVoIP System using Bluetooth Technique", *2nd International Conference on Future Computer and Communication, June 2010, China*.
- [6] <http://beagleboard.org/>
- [7] [http://focus.ti.com/docs/prod/folders/print/omap3530.htm](http://focus.ti.com/docs/prod/folders/print/omap3530.html)
- [8] <http://beagleboard.org/project/angstrom/>
- [9] [http://code.google.com/p/beagleboard/wiki/Beagleboard\\_RevC3Validation](http://code.google.com/p/beagleboard/wiki/Beagleboard_RevC3Validation)
- [10] [http://www.alsa-project.org/main/index.php/Main\\_Page](http://www.alsa-project.org/main/index.php/Main_Page)

## Yazar İndeksi

Adıbelli, Yusuf .....	5	Yıldırım, K. Sinan .....	25
Anıl, Kutsal .....	51	Yıldırım, Mustafa Selman .....	85
Ataman, Korall .....	67	Yurdakul, Arda .....	11, 29, 81, 91
Aysu, Aydın .....	41	Yüzbaşıç, Reşat .....	3
Aytekin, Sevcan .....	57		
Başak, Mustafa .....	67		
Bayar, Salih .....	29		
Baykan, Orhan .....	61		
Bölük, Gökhan .....	61		
Buzluca, Feza .....	35		
Ceylan, Osman .....	53		
Çakır, Halil İbrahim .....	57		
Çalışkan, Aziz U. ....	47		
Dilek, Seyyid Muhammed .....	53		
Doğlan, Ercan .....	67		
Doyuran, Uygur .....	49		
Döner, Çağdaş .....	25		
Erbaş, Bülent .....	3		
Erkoç, Muhammet .....	81		
Güncan, Muhammet Alican .....	81		
Güvenç, Ulvi .....	21		
Hamzaođlu, İlker .....	5, 41		
Kahveci, Ali Yavuz .....	81		
Kantarci, Aylin .....	25		
Kaya, Gürkan .....	17		
Özdemir, Serhat .....	1		
Özmen, Ahmet .....	57		
Parlak, Mustafa .....	5		
Saldamlı, Gökay .....	17		
Savaşkan, İsmail .....	35		
Şimşek, Gökhan .....	25		
Tangel, Ali .....	21		
Tükel, Mehmet .....	11		
Ulusel, Onur Can .....	41		
Urhan, Ođuzhan .....	75		
Usta, Abdullah .....	91		
Uzun, İsmail .....	97		
Yağcı, H. Bülent .....	53		
Yakut, Mehmet .....	21		
Yalçın, Müştak Erhan .....	17		
Yardım, Fatih .....	75		
Yavuz, Gürcan .....	57		
Yerebakan, Halid Ziya .....	97		