İSTANBUL TEKNİK ÜNİVERSİTESİ ★ FEN BİLİMLERİ ENSTİTÜSÜ

MEKATRONİK SİSTEMLERDE KLONLAMAYA KARŞI ELEKTRONİK ÇÖZÜMLER

YÜKSEK LİSANS TEZİ

Emrah ABTİOĞLU

Mekatronik Mühendisliği Anabilim Dalı

Mekatronik Mühendisliği Programı

MAYIS 2014

İSTANBUL TEKNİK ÜNİVERSİTESİ ★ FEN BİLİMLERİ ENSTİTÜSÜ

MEKATRONİK SİSTEMLERDE KLONLAMAYA KARŞI ELEKTRONİK ÇÖZÜMLER

YÜKSEK LİSANS TEZİ

Emrah ABTİOĞLU
(518121034)

Mekatronik Mühendisliği Anabilim Dalı

Mekatronik Mühendisliği Programı

Tez Danışmanı: Prof. Dr. Müştak Erhan YALÇIN

MAYIS 2014

İTÜ, Fen Bilimleri Enstitüsü'nün 518121034 numaralı Yüksek Lisans Öğrencisi **Emrah ABTİOĞLU**, ilgili yönetmeliklerin belirlediği gerekli tüm şartları yerine getirdikten sonra hazırladığı **"MEKATRONİK SİSTEMLERDE KLONLAMAYA KARŞI ELEKTRONİK ÇÖZÜMLER"** başlıklı tezini aşağıdaki imzaları olan jüri önünde başarı ile sunmuştur.

Tez Danışmanı :	**Prof. Dr. Müştak Erhan YALÇIN** İstanbul Teknik Üniversitesi	.....
Jüri Üyeleri :	**Yrd. Doç. Dr. Pınar BOYRAZ** İstanbul Teknik Üniversitesi	.....
	Yrd. Doç. Dr. Nerhun YILDIZ Yıldız Teknik Üniversitesi	.....

Teslim Tarihi : **7 Temmuz 2014**
Savunma Tarihi : **14 Temmuz 2014**

Aileme,

ÖNSÖZ

Bu tez çalışmasında danışmanlığımı yapan değerli hocam Prof. Dr. Müştak Erhan Yalçın'a teşekkür ederim.

Aileme her an yanımda olup bana sağlamış oldukları maddi ve manevi tüm desteklerden dolayı teşekkür ederim.

Tez çalışması süresince yardımlarını benden esirgemeyen Yük. Müh. Ramazan Yeniçeri'ye ve Yük. Müh. Emre Göncü'ye teşekkürü bir borç bilirim.

Bu tez çalışması TÜBİTAK 2210-C Öncelikli Alanlara Yönelik Yurt İçi Yüksek Lisans Burs Programı kapsamında TÜBİTAK tarafından desteklenmiştir.

Mayıs 2014

Emrah ABTİOĞLU
Araştırma Görevlisi

İÇİNDEKİLER

Sayfa

KISALTMALAR

AES : Gelişmiş Şifreleme Standartı
ASIC : Uygulamaya Özgü Tüm Devre
CLB : Konfigüre Edilebilir Mantık Bloğu
CPLD : Karmaşık Programlanabilir Mantık Aygıtı
FPGA : Sahada Programlanabilir Kapı Dizileri
GAL : Genel Dizi Mantığı
ICAP : İç Konfigürasyon Erişim Portu
IP : Fikri Mülkiyet
PAL : Programlanabilir Dizi Mantığı
PLA : Programlanabilir Mantık Dizisi
PLD : Programlanabilir Mantık Aygıtları
PUF : Fiziksel Klonlanamayan Fonksiyon
VHDL : Yüksek Hızda Tümdevre Donanım Tanımlama Dili

ÇİZELGE LİSTESİ

Sayfa

ŞEKİL LİSTESİ

Sayfa

MEKATRONİK SİSTEMLERDE KLONLAMAYA KARŞI ELEKTRONİK ÇÖZÜMLER

ÖZET

Sahtecilik ve taklitçilik faaliyetlerinin kapsamı ve boyutu dünya genelinde gün geçtikçe artış göstermektedir. Otomotiv, elektronik ve tekstil pazarları başta olmak üzere, bu faaliyetler neredeyse tüm pazarlarda karşımıza çıkmaktadır. Fikri mülkiyet haklarını doğrudan etkileyen bu durum, yeni ürün ortaya çıkaran ve ar-ge faaliyetlerine yatırım yapan şirketlerin pazar paylarının azalmasına sebebiyet vererek, bu şirketlerin kazançlarını etkilemektedir. Sahtecilik ve taklitçilik faaliyetlerinin ekonomik boyutunun yanında, bu ürünlerin standartların altında üretilmesi, beraberinde sağlık ve güvenlik risklerini de ortaya çıkarmaktadır. Sağlık ve ekonomi açısından bu faaliyetlerden dolayı oluşan kayıplar küçümsenemeyecek kadar fazladır.

Teknolojinin ilerlemesi ile sahte ve taklit ürünlerin orjinal ürünlerden gözle ayırt edilmesi neredeyse imkansız hale gelmiştir. Üreticiler bu noktada ürünlerinde kimlik doğrulama teknolojileri kullanarak, ürünlerini korumaya çalışmaktadırlar. Bu yöntemin uygulanamadığı ürünlerde ise, şirketler ürün tedarik zincirlerini sıkı bir kontrol altına alıp, tasarımlarının veya kullandıkları malzemelerin üçüncü şahısların eline geçmesini engellemeye çalışmaktadırlar.

Sahtecilik ve taklitçiliğin tehdit oluşturduğu alanlardan biride robot endüstrisidir. Hızla gelişen robot endüstrisinde üretim, yoğun bir mühendislik ve tasarım sürecinin sonucunda gerçekleştirilebilmektedir. Robotların sahip oldukları hız ve güçlerini dikkate aldığımızda, herhangi bir parçasının standartlarının altında bir klonu ile değiştirilmesi, hem insan sağlığı ve güvenliği hemde robotun güvenliği açısından ciddi ve tehlikeli sonuçlara neden olabilir.

Robotların güvenliğini arttırmak adına, robotları kişiselleştirmek ve onlara klonlanamayan kimlikler verilmesi ile bu problemin önüne geçebilir. Fiziksel olarak klonlanamayan fonksiyonlar, klonlanamayan kimliği üretmek için ideal bir yöntem olarak son yıllarda tercih edilmeye başlanmıştır. Karakteristik özelliklerinin modellenmesi zor olan bir fiziksel sistem tarafından kolay bir şekilde hesaplanan bu fonksiyonlar belli bir sayıdaki giriş setini, fiziksel sistemin iç yapısına bağlı olarak rastgele ve eşsiz bir çıkış setine dönüştürürler. Bu özellikleri sayesinde iki farklı yapı üzerinde oluşturulan aynı yapıdaki fiziksel olarak klonlanamayan fonksiyonlar birbirinden farklı sonuçlar üretirler. Fiziksel olarak klonlanamayan fonksiyonların devre üzerinde gerçeklenmesi, şifreleme anahtarının taşınması problemini de ortadan kaldırılmış olur.

Literatürde bulunan fiziksel olarak klonlanamayan fonksiyon çeşitlerinden, halka osilatör tabanlı fiziksel olarak klonlanamayan fonksiyon, şifreleme algoritmasının anahtar verisini üretmek için kullanılmıştır.

Çevreden kaynaklı oluşabilecek sıcaklık farkı ve gerilim dalgalanması gibi gürültülerden dolayı fiziksel olarak klonlanamayan fonksiyonların sonuçları sürekli olarak

değişim gösterebilir. Bu etkilerden kurtulmak adına etiketleme adı verilen bir yöntem kullanılmıştır. Etiketleme yöntemi ile üretilen kimlik doğrulama verisinin üretimi kararlı bir yapıya ulaşmıştır.

Esnek tasarım imkanı sunan sahada programlanabilir kapı dizileri bu çalışmadaki tasarım ortamı olarak belirlenmiştir. Sahip oldukları özelliklerden biri olan kısmi yeniden yapılandırma, bu çalışmada gerçeklenmek istenen güvenlik sisteminin otomatik olarak çalıştırılabilmesi için kullanılmıştır.

Şifreleme algoritması olarak Gelişmiş Şifreleme Standardı tercih edilmiştir. Galois Sayaç Kipi, paralelleştirmeye uygun olması ve kimlik doğrulama algoritması olarak da işlev görmesi nedeniyle, bu algoritmanın çalışma kipi olarak kullanılmıştır.

ELECTRONIC SOLUTIONS AGAINST CLONING IN MECHATRONIC SYSTEMS

SUMMARY

Worlwide fraud and counterfeiting activities and the size of their scope is increasing day by day. These activities are encountered in almost all markets especially in automotive, electronics and textile markets. Due to direct effects on intellectual property rights, this situation causes a reduction of market shares of companies which produce new products and invest in R&D activities. Besides the economic dimension of fraud and counterfeiting activities, these products that are produced with low standards pose health and safety risks. Health and economic losses resulting from these activities can not be underestimated.

With the advancement of technology, it has become almost impossible to distinguish counterfeit and fake products from the original products . Manufacturers are trying to protect their products by using authentication technologies . Companies that have products which are not suitable for authentication, apply strict controls on their supply chains and design processes.

Counterfeiting and fraud activities are also a threat for the robotics industry. Intensive engineering and design process results in rapid evolution in the robotics industry. Considering the speed and power of robots, the parts that are replaced with a substandard clone can cause serious and dangerous consequences for both human safety and safety of the robot .

In the name of improving the safety of the robot, personalizing robots and giving them unclonable identities may prevent such activities and threats. In recent years, physical unclonable functions have been preferred to produce unclonable identities. Physical unclonable functions depend on static environment parameters and the internal structure of the physical systems. Thanks to these features, two different physical functions that are formed according to same structure produce different results. Due to being almost impossible to be cloned practically, the implementation of physical unclonable functions eliminates the storage problem of encryption key in cryptographic systems.

Several physical unclonable function circuit types exists in literature. SRAM, butterfly and ring oscillator based physical unclonable functions are well known. Ring oscillator based physical unclonable function is used in this work. Due to its easy implementation on FPGA chip, it has been chosen to be used in this work. Ring oscillator based physical unclonable function circuit has two ring oscillators. One bit output is produced according to the sign of the diffrence of the frequencies of these ring oscillators. Ring oscillator pairs must have mathematically identical path delays so that frequency difference depends on characteristics of FPGA chip.

Since physical unclonable functions are produced based on nonlinear characteristics of materials, they are mostly affected by changes in temperature, voltage fluctuations

and radiation. Due to this nonlinear characteristic, physical unclonable functions are not deterministic, and it is not possible to produce same outcome for every trial. In order to eliminate this problem error correction codes are being used. Error correction codes gather statistical data about physical unclonable functions, and then produces outcomes according data that they have gathered. In this work an error correction method called labeling has been used. Labeling method gathers statistical data about physical unclonable functions under different conditions, and produces a label for each ring oscillator pair. Outcome of ring oscillator based physical function is calculated by using obtained label and measured difference of frequency. One important feauture of labeling method is that it does not reveal any information about physical unclonable function.

Field programmable gate array (FPGA) offers a flexible design environment. FPGAs have the second best performance after ASIC designs, but in recent years performance gap between FPGA and ASIC has been shrinking due to developments in IC technology. In this study, FPGA has been chosen as design environment. Xilinx ML505 evaluation platform has been used. It has Xilinx Virtex-5 XC5VLX50T FPGA chip on it. One of the most important features of FPGA is partial reconfiguration. Partial reconfugiration allows to make changes on a partial area while FPGA still operates. In partial reconfiguration designs, resources are used in a time-multiplexed manner. This time-multiplexed manner reduces power consumption and area usage.

In this study, partial reconfiguration is used to load ciphered bitstream from a non-volatile memory into FPGA. Partial reconfiguration has been used to automate authentication process. Each time FPGA is powered up, first design that is loaded into FPGA starts authentication process. During this authentication process, ciphered partial bitsream is deciphered. If deciphered partial bitstream succesfully passes authentication, this deciphered partial bitstream is loaded into pre-defined partially reconfigurable partition.

Partial bitstream which is stored in non-volatile memory needs to be ciphered to prevent cloning. In this work, Advanced Encryption Standard which is a specification for the encryption of electronic data is used as crypto algorithm. FPGA configuration process is very sensitive to bit errors in bitstreams, one bit error can cause a faulty reconfiguration, and even further FPGA can be damaged. Therefore, a checksum algorithm which checks correctness of bitstreams has to be used. Considering this problem Advanced Encryption Standard is used with Galois Counter Mode of operation. Galois Counter Mode has authentication feature which also performs the checksum task. Galois Counter Mode is highly parallelable mode of operation. Therefore, it is very suitable for hardware implementations on FPGA fabric. Due to the fact that Galois Counter Mode provides both message confidentiality and authenticity, it has been used in this study.

A BPI PROM on Xilinx ML 505 board is used as non-volatile memory. Two bitsreams are stored in this BPI PROM. First bitstream is loaded into FPGA each time it is powered up. This bitstream performs authentication and partial reconfiguration processes. Second bitstream contains cipehered partial bitstream. Partial bitstream is ciphered with 128-bit key which is obtained from ring oscillator based physical unclonable function circuit. Partial reconfiguration process is triggered by a push-button on ML505 evaluation platform. When it is pressed, first design which is loaded into FPGA starts to read ciphered partial bitstream from BPI PROM.

This ciphered partial bitstream is deciphered with 128-bit key which is produced by ring oscillator based physical unclonable function circuit which is implemented in first bitstream. After decipher process, if deciphered partial bit stream succesfully passes authentication process, partial reconfiguration process takes place. To pass authentication process, both keys that are used to cipher partial bitstream and produced from physical unclonable function circuit have to be exactly same.

1. GİRİŞ

Dünyada sahtecilik ve taklitçilik faaliyetlerinin kapsamı ve boyutu gün geçtikçe artış göstermektedir. Bu faaliyetler otomotiv, elektronik ve tekstil pazarları başta olmak üzere, neredeyse tüm pazarlarda karşımıza çıkmaktadır. Doğrudan fikri mülkiyet haklarını etkileyen bu durum, ar-ge faaliyetlerine yatırım yaparak yeni ürünler ortaya çıkaran şirketlerin pazar paylarının azalmasına sebebiyet vererek, bu şirketlerin büyüme süreçlerini de yavaşlatmaktadır. Ekonomik İşbirliği ve Kalkınma Örgütü tarafından 2005 yılında yayınlanan rapora göre sahte ve taklit ürünlerin pazar payı yaklaşık olarak 200 milyar dolar olarak belirlenmiştir. Bu rakam, 2005 yılı verilerine göre 150 ülkenin gayri safi milli hasılasından daha büyüktür. Ayrıca bu belirtilen rakam içerisinde, internet ortamındaki sahte ve taklit ürünlerin değeri yer almamaktadır. Bu ürünlerin değerleri ile birlikte, sahte ve taklit ürünlerin değerinin yüzlerce milyar doları aştığı belirtilmiştir. Çizelge 1.1 en büyük 20 ekonominin sahte ürün pazarındaki paylarını göstermektedir. Bu noktada yüzde 69.7 gibi yüksek bir oranla Asya en ön sırada yer almaktadır [1].

Çizelge 1.1: En büyük 20 ekonomi arasında sahte ürün kaynaklarının bölgelere göre dağılımı [1].

Bölge	Ülke Sayısı	Yüzde
Asya	12	69.7
Orta Doğu	2	4.1
Afrika	2	1.8
Avrupa	2	1.7
Kuzey Amerika	1	1.1
Güney Amerika	1	0.8
Toplam	20	79.2

Sahtecilik ve taklitçilik faaliyetlerinin ekonomik boyutunun yanında, bu ürünlerin standartların altında üretilmesi, beraberinde sağlık ve güvenlik risklerini de ortaya çıkarmaktadır. Uluslararası Ticaret Odası'nın başlattığı Sahtecilik ve Korsana Karşı İş Hareketi tarafından 2009 yılında yayınlanan rapora göre Hindistan'da, otomobillerde kullanılan sahte ve taklit parçalar, 25,400 kişinin ölümüne ve 93,000 kişinin de yaralanmasına sebebiyet vermiştir. Ayrıca bu parçaların kullanımından dolayı,

her yıl ektsra olarak 109 milyon litre petrol ve 8 milyon litre dizelin tüketildiği belirtilmiştir. Bu parçaların kullanımı ile sağlık ve ekonomi açısından oluşan kayıplar küçümsenemeyecek kadar fazladır [2].

1.1 Sahtecilik ve Taklitçiliğe Karşı Alınan Önlemler

Hükümetler arası kuruluşlar -Dünya Ticaret Örgütü, Dünya Fikri Mülkiyetler Örgütü, Interpol- bu problemle başa çıkabilmek adına, yaptırımları geliştiren özel programları uygulamaya almışlardır. Bazı hükümetler ise bu konuda uzmanlaşmış fikri mülkiyet güçlerini ve mahkemelerini kurmuşlardır. Hükümetlerin bu konuda attıkları adımlardan birisi, tüketici tarafındaki farkındalığı arttırmaktır. Sahteciliğin ve taklitçiliğin, dünya ekonomisine etkisini, bireysel seviyede sebep olduğu sorunları, yürütülen çeşitli kampanyalar ve organizasyonlar ile tüketiciye aktarmaya çalışmaktadırlar.

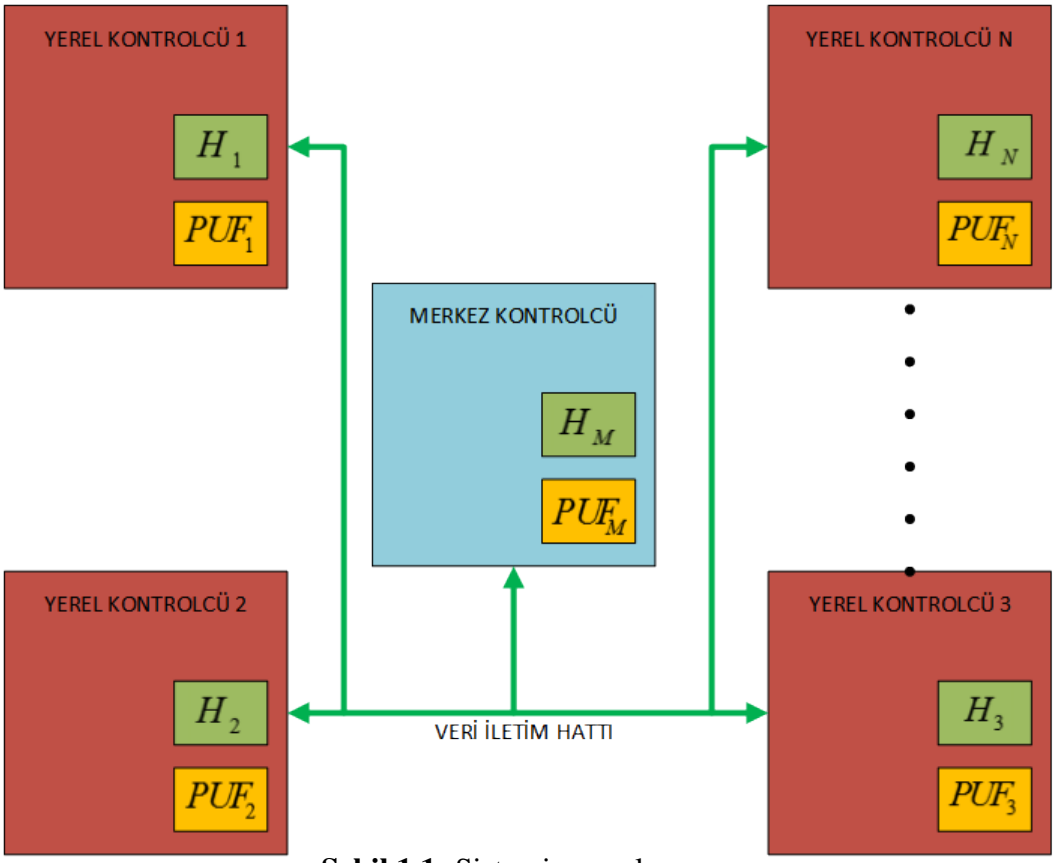
Bu problemle başa çıkabilmek için endüstri ayağında ise farklı adımlar atılmaktadır. Teknolojinin ilerlemesi ile sahte ve taklit ürünlerin orjinal ürünlerden gözle ayırt edilmesi neredeyse imkansız hale gelmiştir. Üreticiler bu noktada ürünlerinde kimlik doğrulama teknolojileri kullanarak, ürünlerini korumaya çalışmaktadırlar. Bu yöntemin uygulanamadığı ürünlerde ise, şirketler ürün tedarik zincirlerini sıkı bir kontrol altına alıp, tasarımlarının veya kullandıkları malzemelerin üçüncü şahısların eline geçmesini engellemeye çalışmaktadırlar [1].

1.2 Tezin Amacı

Sahtecilik ve taklitçiliğin tehdit oluşturduğu alanlardan biride robot endüstrisidir. Hızla gelişen robot endüstrisinde üretim, yoğun bir mühendislik ve tasarım sürecinin sonucunda gerçekleştirilebilmektedir. Gelecekte hayatımızın büyük parçası olacak olan robotlar için sahte ve taklit ürünler günümüz koşullarında yok denecek kadar az olsa dahi, yakın bir zamanda bu sorunla karşılaşacağımız aşikardır. Robotların sahip oldukları hız ve güçlerini dikkate aldığımızda, herhangi bir parçasının standartlarının altında bir klonu ile değiştirilmesi, hem insan sağlığı ve güvenliği hemde robotun güvenliği açısından ciddi ve tehlikeli sonuçlara neden olabilir. Bu tez çalışmasının ilk amacı, robotun herhangi bir parçasının orjinal olmayan bir klonu ile değiştirildiğinde,

robotun bu durumu sezmesini ve gerekli önlemleri -çalışmayı tamamen durdurmak ya da söz konusu klonlanmış parçayı kullanmamak- almasını sağlamaktır.

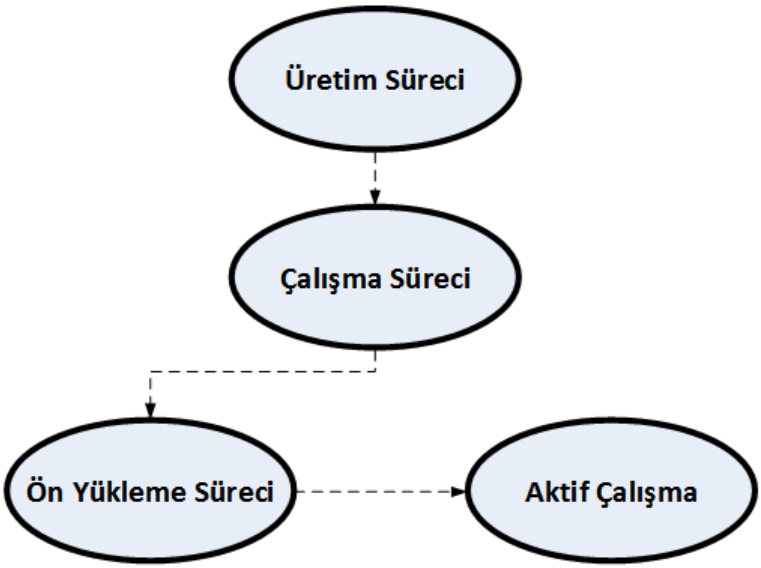
Bu konuda [3]'te yapılan çalışmada robotların güvenliğini arttırmak adına, robotları kişiselleştirmek ve onlara klonlanamayan kimlikler verilmesi için bir çalışma konsepti önerilmiştir. Bu konsepte göre robotlar zamanda belli bir anda doğmalı ve klonlanamayan bir kimliğe sahip olmalıdır. Bu kimlik robotun içerisindeki mümkün olan tüm iletişim kanallarına nüfuz ederek güvenliği sağlayan, çalışma prensibi insan DNA'sına benzeyen bir yapı olarak düşünülmüştür. Klonlanamayan kimliği üretmek için [4]'te önerilen *'elektronik mutasyon'* yöntemine başvurmuşlardır. Bu tez çalışmasında ise robot üzerindeki parçaların klonlanmasını önlemek amacıyla klonlanamayan kimlik üretilmesi konsepti farklı bir yöntem ile gerçeklenmiştir.



Şekil 1.1: Sistemin genel yapısı.

Şekil 1.1'de sistemin genel yapısına yer verilmiştir. Sistemde iki farklı kontrolcü yer almaktadır. Bu kontrolcüler merkez kontrolcü ve yerel kontrolcü olarak adlandırılmıştır. Merkez kontrolcü, robot uzuvlarının kimlik doğrulamasından ve yerel kontrolcülerin denetiminden sorumludur. Yerel kontrolcüler ise üzerinde bulundukları uzuvların kontrolünden sorumludur. Tüm kontrolcüler üzerinde, fiziksel olarak

klonlanamayan fonksiyonlar yardımıyla üretilen 128-bitlik PUF_N değerleri ve bu değerlerin özet fonksiyonundan geçirilmesiyle elde edilen 128-bitlik H_N değerleri yer almaktadır. Merkez kontrolcü uzuvların kimlik doğrulamalarını H_N değerlerini kullanarak gerçekleştirir. Bu kimlik doğrulama sürecine 4. bölümde ayrıntılı olarak yer verilmiştir. Bu tez çalışmasının diğer amacı ise, robot üzerindeki kontrol algoritmalarını içeren tasarımın kopyalanmasını engellemektir. Bu konuda Gövem'in [5]'de yapmış olduğu çalışmadan yararlanılmıştır. Gövem çalışmasında fiziksel olarak klonlanamayan fonksiyonları kullanarak FPGA üzerinde yapılan tasarımların kopyalanmasını engellemeye yönelik bir çalışma gerçekleştirmiştir.



Şekil 1.2: Sistem süreçleri.

Şekil 1.2'de sistemin çalışır hale gelebilmesi için geçmesi gereken süreçler gösterilmiştir. İlk olarak üretim süreci safhasında FPGA üzerinde fiziksel olarak klonlanamayan fonksiyonlar çalıştırılır. Fiziksel olarak klonlanamayan fonksiyonlardan elde edilen veriler toplanarak istatistiki bir değerlendirme yapılır. Bu değerlendirme sonunda 128-bitlik PUF_N değeri elde edilir. Bu işlem tüm uzuvlar için ayrı ayrı yerine getirilir. Üretim süreci sonunda her bir uzuvun kendine ait 128-bitlik PUF_N değeri elde edilmiş olur. Elde edilen PUF_N değerleri uzuvların üzerinde kısmi yeniden yapılandırılabilen bölgede yer alacak olan tasarımları şifrelemek için kullanılır. Şifrelenen tasarımlar ilgili uzuvların üzerinde yer alan BPI PROM'lar üzerinde saklanarak üretim süreci sonlandırılır.

Çalışma süreci, iki ardışıl süreçten oluşmaktadır. Bunlardan ilki ön yükleme sürecidir. Ön yükleme sürecinde merkez kontrolcü H_N değerlerini kontrol ederek

uzuvların kimlik doğrulamasını yapar. Kimlik doğrulamasından geçen uzuvlar üzerlerinde yer alan fiziksel olarak klonlanamayan fonksiyonları çalıştırarak 128-bitlik bir anahtar üretirler. Eğer üretilen bu anahtar ile üretim sürecinde elde edilen 128-bitlik PUF_N değeri aynı ise, uzuv ilgili tasarımı BPI PROM'dan okuyup, kısmi yeniden yapılandırılabilen bölgeye yükler. Bu süreçte kısmi yeniden yapılandırma kullanılmasının sebebi, şifreleme anahtarının güvenli bir şekilde saklanması problemidir. Şifreleme anahtarı hiç bir şekilde dış ortama çıkmayıp, tamamen FPGA üzerinde yer alan fiziksel olarak klonlanayan fonksiyonlar yardımıyla üretilmektedir. FPGA üzerinde çalışma sırasında anahtar üretilene kadar tasarımı çözmek için herhangi bir anahtar ortamda yer almamaktadır. Bu sebeple ilk önce şifreleme anahtarı üretilip, sonrasında ilgili tasarım bu üretilen anahtar ile çözülerek sistem üzerine yüklenmektedir. Kısmi yeniden yapılandırmaya 3. bölümde ayrıntılı olarak yer verilmiştir. Bu işlem sonucunda ön yükleme süreci biter ve aktif çalışma süreci başlar.

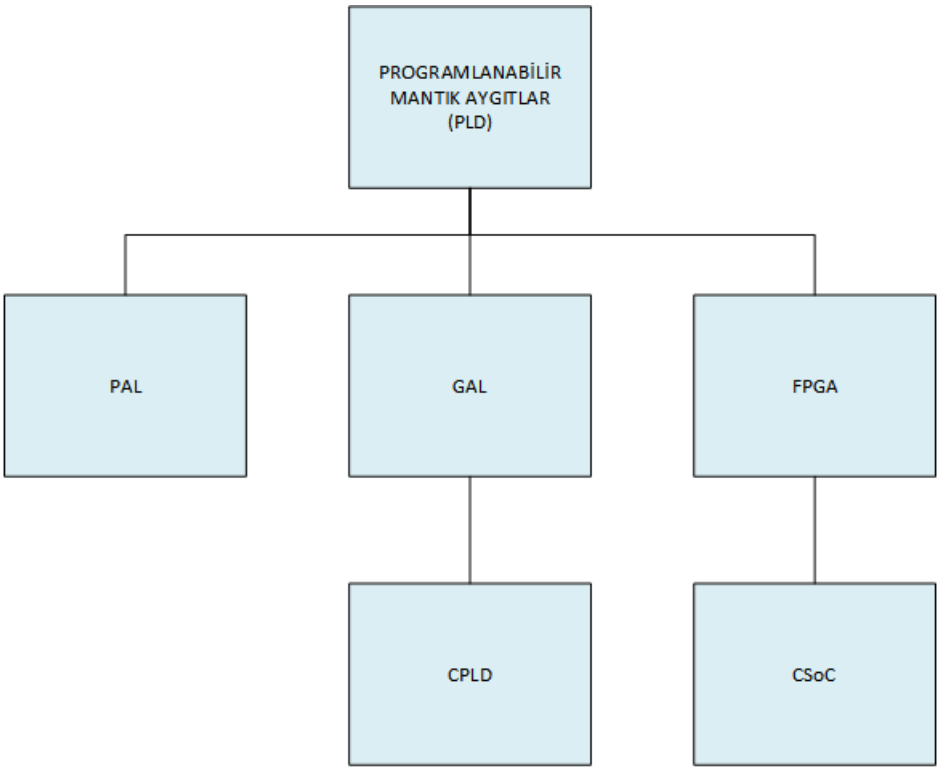
Aktif çalışma süreci, güvenli bir şekilde yüklenmiş olan kontrol algoritmalarının yardımıyla robotun istenilen davranışı gösterdiği çalışma sürecidir. Eğer robot bu sürece ulaşmışsa, robot uzuvlarının kimlik doğrulaması yapılmış ve şifrelenmiş tasarımlar güvenli bir şekilde iligili uzuvlara yüklenmiştir.

Bu süreçler sonucunda kendi uzuvlarının kimlik doğrulamasını yapabilen ve tasarımların kopyalanmasını önleyen bir sistem ortaya çıkar. Bu sistemi oluşturmak için kullanılan donanımlar, yazılımlar ve yöntemlere tezin sonraki bölümlerinde ayrıntılı bir şekilde yer verilmiştir.

2. FPGA

2.1 Genel Bilgi

FPGA'lar 1980'lerin ortasında Xilinx firması tarafından icat edilmiştir. Bu firma var olan programlanabilen mantık aygıtlarını bir adım ileriye taşıyarak sahada programlanabilen ilk yongayı yapmıştır. ASIC devrelerin aksine üretildikten sonra tekrar programlanarak, kaynaklarına bağlı olarak farklı görevleri yerine getirebilirler. FPGA'lar programlanabilir mantık aygıtları (PLD) ailesinin bir üyesidir. Şekil 2.1'de PLD çeşitleri verilmiştir.



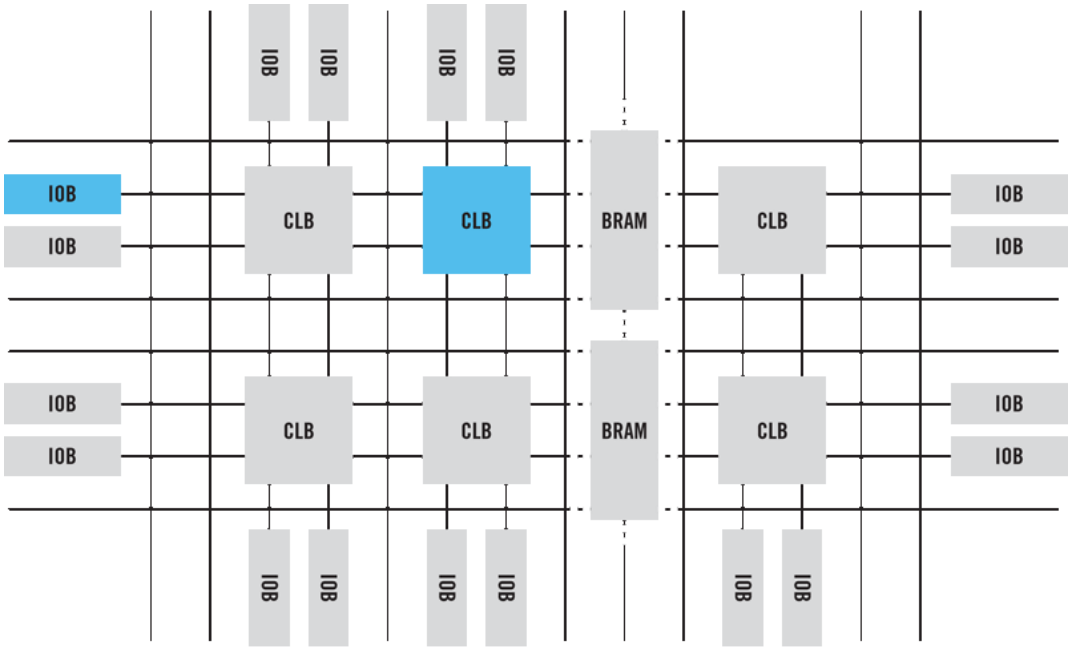
Şekil 2.1: PLD ailesi [6].

PLD'ler ilk olarak 1970'lerin ortasında üretilmişlerdir. O dönemde mantıksal yapılarına göre PLA (Programlanabilir Mantık Dizisi) veya PAL (Programlanabilir Dizi Mantığı) olarak adlandırılıyorlardı. PLD'ler ilk zamanlarda senkron yapılar içermeyen komnbinezonsal sayısal devrelerden oluştukları halde, ilerleyen zamanda

bu devrelerin çıkışlarına tutucu eklenerek senkron hale getirilmiş versiyonlarıda üretilmeye başlamıştır. PAL'lar sadece bir defa programlanabilen yapılardır. GAL (Genel Dizi Mantığı) aygıtlarının üretilmesi sektörde önemli bir adım olmuştur. PAL'lerin aksine bu aygıtlar tekrar programlanabilme özelliğine sahiplerdir. İlerleyen dönemlerde yarıiletken teknolojisindeki gelişmeler ile içerisinde çok sayıda mantık kapısı barındıran CPLD'ler üretilmiştir. Günümüzde ise içerisinde milyonlarca mantık kapısını ve işlemci çekirdeklerini içersinde barındıan CSoC aygıtları üretilmektedir [6].

2.2 FPGA Mimarisi

FPGA'lar içerisinde iki boyutta matris olarak dizilmiş konfigüre edilebilir mantık blokları (CLB) yer almaktadır. CLB'lerin içerisindeki mantık fonksiyonları dışarıdan programlanabildiği gibi, aynı zamanda CLB'lerin arasındaki bağlantılarda kullanıcı tarafından programlanabilir. CLB'lerin yapısı üreticiden üreticiye değiştiği gibi, aynı üreticinin farklı aygıtları arasında da değişebilir. Şekil 2.2'de FPGA içerisinde yer alan yapılar gösterişmiştir.



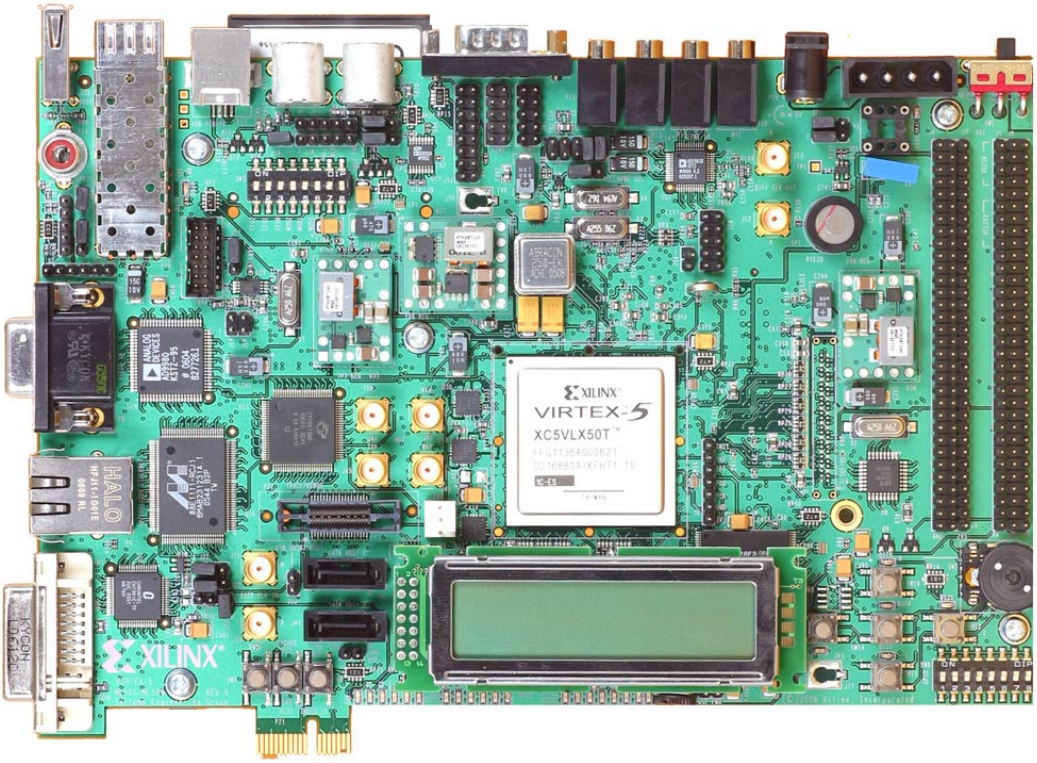
Şekil 2.2: FPGA iç yapısı [7].

CLB'ler dışında BRAM ve giriş çıkış bloklarıda (IOB) FPGA içersinde yer almaktadır. Veri depolama ihtiyacı duyulan tasarımlarda, FPGA üzerindeki BRAM'lar günümüzde 10Mbit'e kadar hafıza alanı sağlamaktadırlar. IOB'lar FPGA yongası üzerindeki tasarımın çevre elemanlar vasıtasıyla dış dünya ile iletişim kurmasını sağlarlar. IOB'lar

sadece giriş, sadece çıkış, ya da çift yönlü çalışacak şekilde programlanabilirler. Bu elemanların dışında ihtiyaca göre FPGA içersinde farklı bloklar yer alabilir. Çarpma bloğu, Fourier transformu alabilen DSP blokları ve saat frekansı kontrol etmek için sayısal saat yönetici bloğu gibi farklı bloklar yer alabilir.

2.2.1 ML505 geliştirme kartı ve Virtex-5 mimarisi

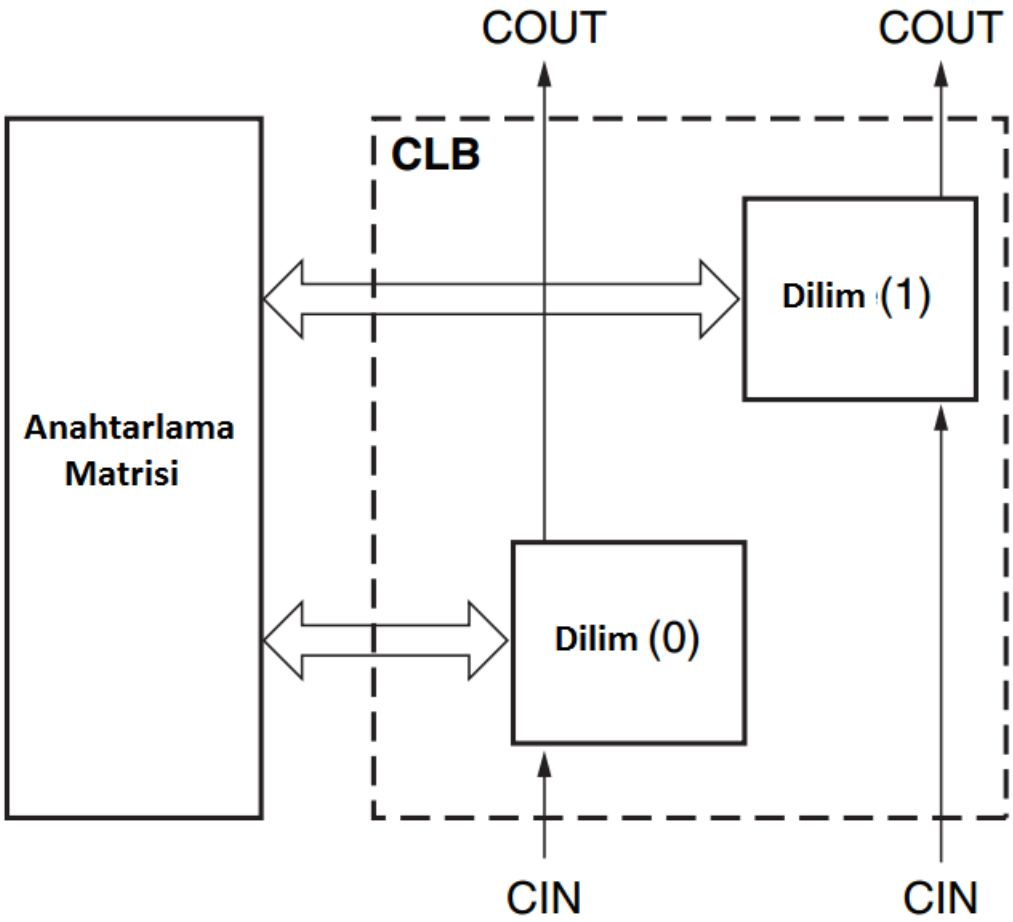
Şekil 2.3'te bu tez çalışmasının üzerinde gerçeklenmiş olduğu Xilinx tarafından üretilen ML505 geliştirme kartı yer almaktadır. Bu geliştirme kartı üzerinde, Virtex-5 XC5VLX50T FPGA yongası, 8 adet genel amaçlı elektrik anahtarı, 5 adet buton, 8 adet LED, 32 adet giriş-çıkış, ses giriş-çıkış hattı, RS-232 seri haberleşme portu, 16x2 LCD ekran, VGA girişi, DVI çıkışı, 32MB kapasiteye sahip BPI PROM, CompactFlash okuyucu, ethernet portu, 100MHz ve 33MHz olmak üzere 2 adet saat üreteci bulunmaktadır. ML505 üzerinde bulunan bu kaynaklar ile sağladığı imkanlardan dolayı gömülü sistem uygulamaları geliştirmek ve hayata geçirmek için ideal bir platformdur [8].



Şekil 2.3: ML505 geliştirme kartı [8].

Şekil 2.4'te Virtex-5 FPGA'larındaki CLB yapısı yer almaktadır. CLB'ler ardışıl ve kombinezonsal devre tasarımlarında kullanılan temel kaynaklardır. Her bir CLB

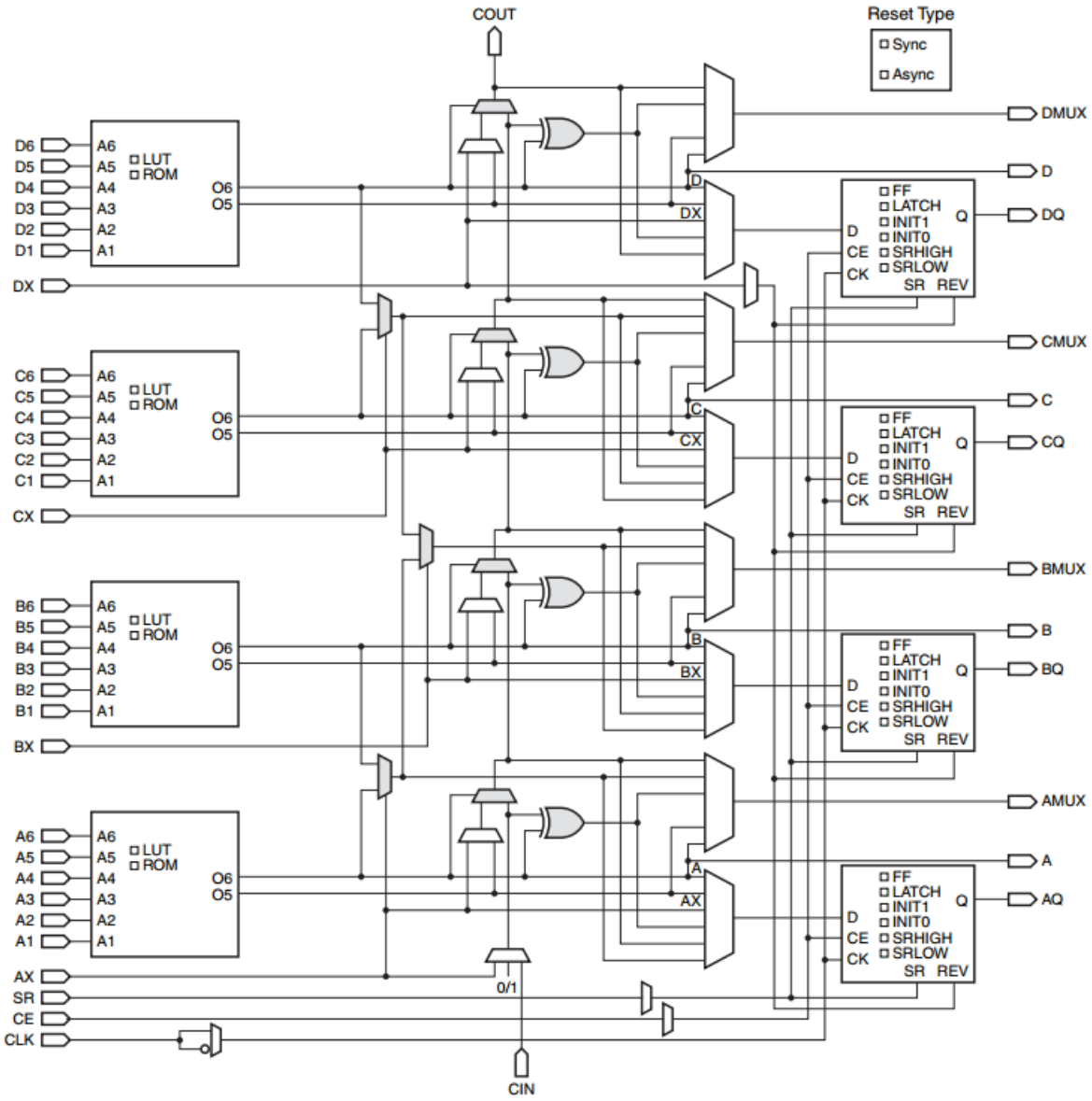
tasarım içindeki sinyallerin yönlendirilmesini sağlayan anahtarlama matrisine bağlıdır. Her CLB'de iki çift dilim bulunmaktadır. Bu iki dilim Şekil 2.4'te de gözüktüğü gibi birbirlerine direkt olarak bağlı değillerdir. Her dilimin kendine ait elde zinciri bulunmaktadır. Bu elde zincirinin görevi, başka bir CLB'den gelen eldeyi ilgili dilime aktarmaktır. Ayrıca bu dilimde elde varsa, bu elde de ilgili CLB'ye aktarılır.



Şekil 2.4: Virtex-5 FPGAlarında CLB yapısı [9].

Şekil 2.5'te dilimlerin iç yapısı gösterilmiştir. Her bir dilim 4 adet 6 girişli 2 çıkışlı taramalı tablo (LUT), 4 adet hafıza elemanı, çoklayıcılar ve elde mantığı içerir. LUT'ların çalışma mantığı RAM'lerle benzerdir. 6 girişi sayesinde, 2^6 adet veri depolayabilir. LUT'ların içine uygun değerler yüklenerek, VE, VEYA gibi mantık kapılarının davranışları gerçeklenebilir. LUT'lar 6 girişe kadar tanımlanan her türlü Boolean fonksiyonunu gerçekleyebilirler. Dilimler bu elemanları, mantıksal ve aritmetik fonksiyonları yerine getirebilmek için kullanır. Bunlara ek olarak veri depolama işlemi de gerektiğinde dilimler tarafından gerçeklenebilir. FPGA'lar bu yapılar sayesinde paralel işlem yapma yetisine sahiptirler. Yüksek işlem hızı gerektiren uygulamalarda gün geçtikçe FPGA kullanımı artmaktadır. Savunma sanayii, otomotiv

sektörü, video ve görüntü işleme gibi yüksek işlem hızına ihtiyaç duyan gerçek zamanlı sistemlerde FPGA'lar pazar payı giderek artış göstermektedir. [9].

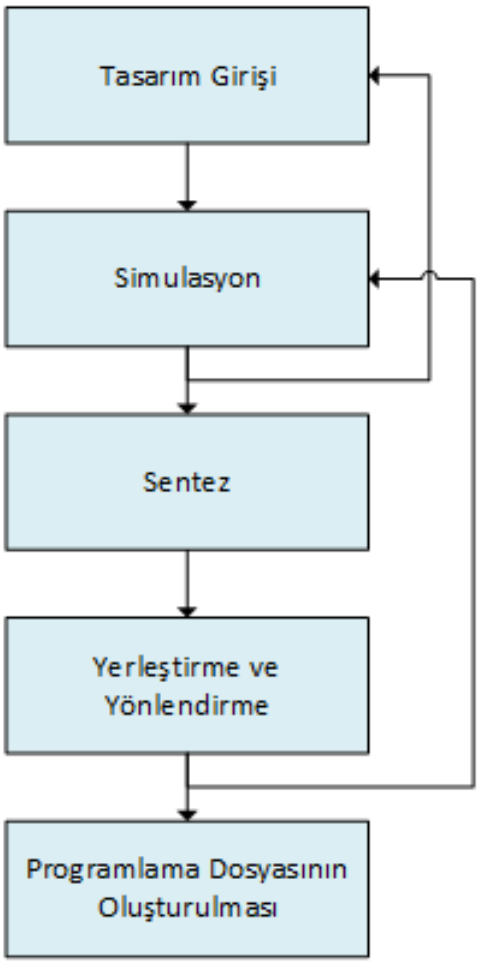


Şekil 2.5: Virtex-5 FPGAlarında dilim yapısı [9].

2.3 FPGA'da Tasarım Süreci

Sayısal mantık devrelerinin tasarım işlemi gömülü sistem programlama işlemine benzememektedir. Donanımsal yapılar, davranışsal ya da yapısal olarak genellikle yüksek seviyeli donanım tanımlama dilleri olan VHDL ya da Verilog dilleri ile yazılırlar. Bu dillerin yanında şematik tasarım da kullanılabilmektedir. Özellikle kompleks ve büyük tasarımlarda şematik tasarım tercih edilmemektedir. Bu tasarımlarda, tasarım üzerinde değişiklik yapmak donanım tanımlama dillerinde çok

daha kolaydır. Yine de tasarımın görsel olarak ifade edilmesi gereken durumlarda özellikle üst modül olarak şematik tasarım kullanılabilmektedir.



Şekil 2.6: Tasarım süreci akış diyagramı.

FPGA üzerinde tasarım geliştirme süreci Şekil 2.6'da gösterilmektedir. Yazılım geliştirme ile donanım geliştirme arasındaki temel fark tasarımcının problemler karşısındaki düşünce yapısıdır. Yazılım geliştiriciler, yazılımlarının sıralı kod işleyen bir işlemci üzerinde gerçekleştirileceğini bildiklerinden sıralı şekilde yazdıkları komutların program sırasına göre işleneceğini bilirler. Donanım geliştiriciler ise üzerinde çalıştıkları yapı gereği paralel çalışabilecek şekilde düşünmeleri ve bu şekilde tasarım yapmaları gerekmektedir. Tüm giriş sinyalleri aynı anda paralel olarak alınır ve çıkışa kadar ilgili LUT ve anahtarlama matrisleri üzerinden paralel şekilde ilerlerler. Donanım tanımlama dilleri ile gerektiğinde kendi içerisinde sıralı işlenecek kod blokları oluşturmak mümkündür. Tasarımın donanım tanımlama dilleri kullanılarak tanımlanması ile tasarım girişi aşaması tamamlanmış olur. Tasarım girişi aşamasının tamamlanmasının ardından fonksiyonel simülasyon kısmına geçilir. Bir

simülatör yardımı ile tasarım çalıştırılarak belirlenen giriş değerleri için tasarımın doğru sonuçlar üretip üretmediğine bakılır. Alan ve zamanlama ile ilgili problemler değerlendirilmez. Çünkü bu problemler kullanılacak teknolojiye bağımlıdır. İlk yapılan simulasyon ile sadece yazılan kodun fonksiyonelliği test edilmiş olur. Daha sonra klasik olarak kodun derlenmesi ve yüklenebilir dosyanın oluşturulması toplamda iki farklı aşamada gerçekleşir. İlk olarak kod bir ara gösterim haline çevrilir. Bu işleme sentez denilmektedir ve sonucunda bir netlist oluşturulur. Bu netlist programlamada kullanılacak aygıttan bağımsızdır. Bu işlem yapılırken tasarım kütüphanelerinden kullanılan yapılar hakkında bilgi alınır ve varsa belirli tasarım sınırlamaları dikkate alınır. İkinci kısım ise gerçeklemedir. Bu aşamada belirli bir yongaya yani teknolojiye göre yerleştirme ve yönlendirme işlemi yapılır. Sentez işlemi sonrasında netlist haline gelen yapı belirli bir FPGA yongası için CLB ve anahtarlama matrislerinde kullanılacak hale getirilir. Bu işlem sonrasında belirli bir yonga için içerisinde giriş-çıkış tanımlamaları ve bağlantıları tanımlayan bir dosya elde edilir. Bu dosya yalnızca belirtilen FPGA yongasına özeldir ve başka bir FPGA yongasında kullanılamaz. Yerleştirme ve yönlendirme aşamasından sonra elde edilen bit katarı dosyasının hatasız üretiminden sonra bu dosyanın FPGA içerisine yüklenmesi ile tasarım tamamlanmış olur.

2.3.1 Donanm tanımlama dilleri

Geleneksel programlama dilleri bilgisayarlar için geliştirilmiştir. Geleneksel programlama dillerinde işlemler sırayla çalıştırılır. Sıralı çalışmaları, algoritmaların adım adım geliştirilmesine olanak sağlayarak, programlayan kişilerin tasarım süreçlerini kolaylaştırır. Tasarım süreci sonunda ortaya çıkan yazılım, istenilen algoritmayı gerçekleyecek komutların mantıklı bir şekilde sıralanmış halidir. Donanım tanımlama dilleri ise yapı olarak geleneksel programlama dillerinden farklıdır. Tasarım süreci sonunda ortaya çıkacak ürün bir yazılıma değil donanıma karşılık gelir. Donanımlar ise yapıları gereği aynı anda birden fazla işlem yaparlar. Örneğin donanımın bir tarafında toplama işlemi yapılırken, diğer tarafında çarpma işlemi yapılabilir. Bu noktada donanımların paralel işlem kabiliyetlerini modelleyebilmek için donanım tanımlama dillerine ihtiyaç duyulmaktadır.

Günümüzde en çok kullanılan donanım tanımlama dilleri Verilog ve VHDL'dir. Verilog, 1984 yılında Gateway Design Automation tarafından geliştirilmiştir. 1995 yılında ise, IEEE tarafından bir standart olarak kabul edilmiştir [10]. VHDL, Amerika Birleşik Devletleri Savunma Bakanlığı'nın talebi doğrultusunda geliştirilmiş bir donanım tanımlama dilidir. İlk versiyonu 1987 yılında çıkarılmış ve aynı yıl içersinde IEEE standardı olarak kabul edilmiştir [11]. Bu tez çalışmasında Verilog ve VHDL dillerinin ikiside kullanılmıştır.

2.4 FPGAnın Kontrolcü Olarak Seçilmesi

Bölüm 1.3'te önerilen mimariye ulaşmadan önce, geliştirilecek olan sistemin iletişim hattının şifrelenmesi gerektiğine karar verilmiştir. Bu karar doğrultusunda şifreleme algoritması olarak AES şifreleme algoritması belirlenmiştir. Sistemde birden fazla kontrolcü olduğu için, bu kontrolcülerin birbirleriyle anlaşabilmelerinin tek şartı, her kontrolcünün şifreleme anahtarlarına sahip olmasıdır. Bu durumda şifreleme anahtarlarının doğrudan olarak her kontrolcü üzerinde tutulması güvenlik riskleri doğuracağından, bankamatiklerde kullanılan veri iletişim sisteminin bir benzerini kurabilmek için özet fonksiyonları kullanılmasına karar verilmiştir.

AES şifreleme algoritması ve özet fonksiyonu yoğun işlem yükü gerektiren yapılardır ve sistemin gerçek zamanlı çalışmasını engelleyebilirler. Bu yapıların kullanımına karar verilmesinin ardından, bu yapıları gerçek zamanlı olarak çalıştıracak sistemler araştırılmıştır. Çizelge 2.1'de bu yapıların FPGA ve mikroişlemci üzerindeki performansları verilmiştir.

Çizelge 2.1: AES ve MD-5 özet fonksiyonun farklı donanımlarda performans karşılaştırması [6].

Algoritma	FPGA	Mikroişlemci
	İşlem Hacmi	İşlem Hacmi
MD-5	5.86Gbps	1.27Gbps
AES	25.1Gbps	0.8Gbps

Çizelge 2.1'deki veriler 2Ghz Intel Pentium IV işlemcisi baz alınarak elde edilmiştir. Bu verilere göre MD-5 özet fonksiyonun hesaplanmasında FPGA'nın işlem hacmi yaklaşık olarak 5 kat daha fazladır. Bu fark AES algoritmasında ise neredeyse 30 kat

daha fazladır. Bu veriler doğrultusunda, AES şifreleme algoritmasının ve FPGA'nın kontrolcü olarak seçilmesine karar verilmiştir.

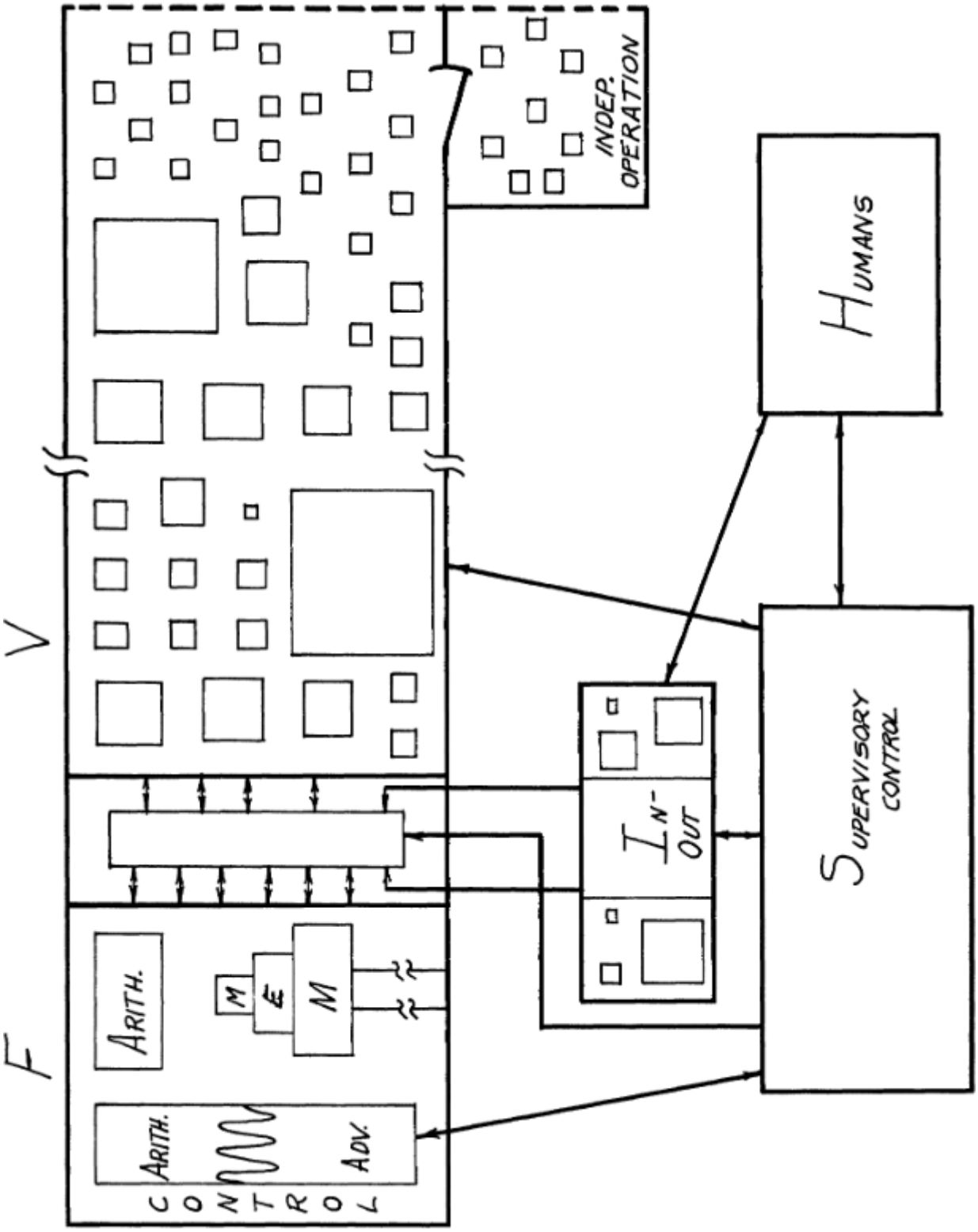
3. KISMİ YENİDEN YAPILANDIRMA

3.1 Genel Bilgi

Yeniden yapılandırma konsepti, 1960 yılında Gerald Estrin'in *Sabit ve Değişken Yapılı Bilgisayar Yapısı* adlı makalesinde ortaya çıkmıştır. Önermiş olduğu sistemin amacı yüksek hızlı altyapılardan oluşan bir envantere sahip olan, o günün şartlarıyla mümkün olmayan hesaplamalara olanak sağlamaktır. Şekil 3.1'de Gerald Estrin'in önermiş olduğu sistemin şematik gösterimi yer almaktadır. *F* ile adlandırılan blok sabittir ve içerisinde aritmetik işlemler yapabilen modüller ve hafıza elemanları yer alan genel amaçlı bir işlemcidir. *V* bloğu ise sistemin değişken kısmıdır ve içerisine özel işlemleri yüksek hızlarda gerçekleyebilecek altyapılar ihtiyaç doğrultusunda yüklenmektedir. Sistemde yer alan denetleyici kontrol ünitesi, en iyi performansı sağlayacak altyapıların *V* içerisine yüklenmesinden sorumludur [12]. Estrin bu çalışması ile yazılımın esnekleğine ve donanımın hızına sahip bir sistem konsepti önermiştir. Önermiş olduğu bu sistem, günümüzdeki yeniden yapılandırabilen aygıtlar ile birbirine benzemektedir.

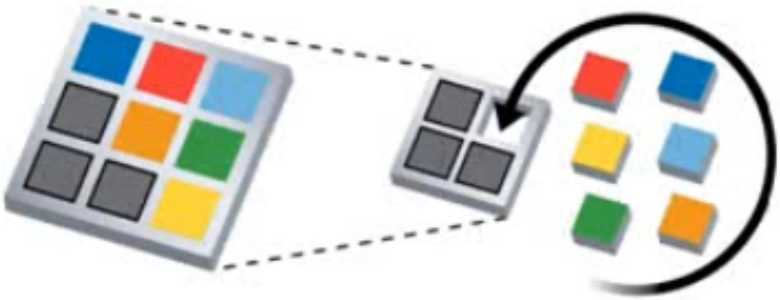
Yeniden yapılandırma konseptinde amaç, iki farklı sistemin artılarını bir araya getirmektir. Bu artılar yazılımın tasarımcıya sağlamış olduğu esneklik ve donanımın yani sayısal mantık devrelerinin hızıdır. Esnekliğe ve hıza sahip olan donanım bir çok konuda kazanç sağlamaktadır. 1991'de üretilen ilk yeniden yapılandırılabilir bilgisayar Algotronix tarafından üretilen Algotronix CHS2X4'tür. Algotronix firmasının ve çalışanlarının Xilinx ile birleşmesiyle yeniden yaplıandırma alanında bir ivmelenme başlamıştır [6].

FPGA teknolojisi, yonga üretimi tamamlandıktan sonra, sahada tekrar programlanabilme esnekliğine sahiptir. Kısmi yeniden yapılandırma özelliği, bu esnekliği bir adım daha ileriye taşıyarak, faal olan bir FPGA yongası üzerinde değişiklik yapmaya olanak sağlar.



Şekil 3.1: Gerald Estrin tarafından önerilen sistem [12].

Sayısal mantık devrelerin tasarımlarında farklı görevleri olan bloklar bir araya getirilerek sistem tasarımı yapılır. Bu blokların tamamı fiziksel olarak gerçeklenir. Ancak sistemin çalışması sırasında tüm bloklara sürekli ihtiyaç olmayabilir. Bazı blokların çalışma zamanları kesişmez. Yani bir blok çalışırken diğer bloğa ihtiyaç olmayabilir. Bu kategoriye giren tasarımlarda var olan bloklar çalışma esnasında belli zaman aralıklarıyla pasif olsalar dahi, donanım olarak gerçeklendiklerinden dolayı belli bir alan kaplarlar, ve belli bir güç tüketimi söz konusudur.



Şekil 3.2: Kısmi yeniden yapılandırma konsepti.

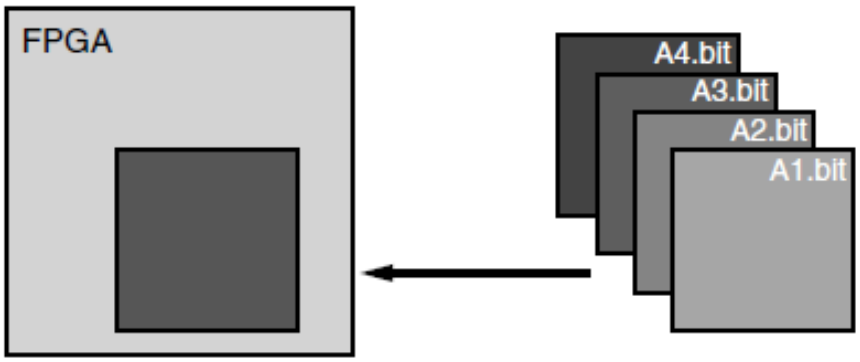
Anlatılan bu duruma benzer bir yapı Şekil 3.2'de gösterilmiştir. Gri renk ile gösterilen bloklar tasarımın sürekli ihtiyaç duyduğu bloklardır. Gri renk dışındaki diğer renkte gösterilen blokların çalışma zamanları zaman uzayında kesişmemektedir. İlk durumda gösterilen tüm blokların tek bir yonga üzerinde gerçeklenmesi yerine, sabit blokların yanına yeniden yapılandırılabilme özelliği olan bir blok yerleştirilerek alandan ve güçten tasarruf edilebilir. Böylece bir bir bloğa ihtiyaç duyulduğunda, yeniden yapılandırma bölgesine yüklenerek, sistem herhangi bir aksaklık yaşamadan çalışmasına devam edebilir.

Kısmi yeniden yapılandırmanın ihtiyaç duyulabileceği durumlar şu şekilde sıralanabilir. İlk olarak, tasarımın kullanılacak olan yongaya sığmadığı durumlar örnek verilebilir. Böyle bir durumda, çalışma süreleri zamanda kesişmeyen bloklar yeniden yapılandırma bölgesinde çalıştırılarak, tasarımın boyutu küçültülerek yongaya sığması sağlanabilir. Yeniden yapılandırmanın ihtiyaç duyulabileceği başka bir durum ise, güç yönetimi gerektiren sistemlerdir. Güç yönetimiyle ilgili internet ağı üzerinden bir örnek verilebilir. Elimizde yonga üzerinde tasarımını yaptığımız iki adet ethernet modülü olsun. Bu modüllerden ilki 1Gbps veri aktarımına olanak sağlayıp daha az enerji tüketmektedir. İkinci modül ise 10Gbps veri aktarımına olanak sağlamakla birlikte, enerji tüketimi daha fazladır. İnternet ağı dinlenerek veri aktarımının 1Gbps'yi

geçmediği durumlarda ilk modülü kullanan, 1Gbps'yi geçtiği durumlarda da ikinci modülü kullanan yeniden yapılandırma özellğine sahip bir sistem ile daha az enerji tüketimi sağlanabilir.

3.2 Xilinx FPGAlarda Kısmi Yeniden Yapılandırma

Xilinx tarafından üretilen FPGA'ların Virtex serisi ve 7 serisi FPGA'larında kısmi yeniden yapılandırma özelliği bulunmaktadır. FPGA üzerine bit katarı dosyası yüklendikten sonra, tasarım çalışmaya başlar. Bu tasarım çalışırken, FPGA üzerinde daha önceden belirlenmiş bir bölgeye, kısmi bit katarı dosyası yüklenerek, yeni bir donanım eklenebilir. Şekil 3.3'te koyu renk gösterilen bölge, yeniden yapılandırılabilen bölge olarak belirlenmiştir. Bu bölgeye daha önceden hazırlanmış olan *A1.bit*, *A2.bit*, *A3.bit*, *A4.bit* gibi kısmi bit katarı dosyaları, sistem çalışırken, sistemin geri kalanını etkilemeyecek bir şekilde yüklenebilir.

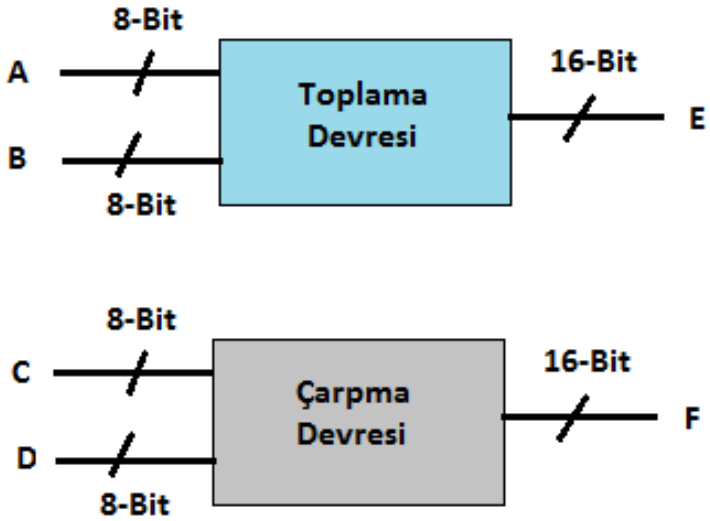


Şekil 3.3: Kısmi yeniden yapılandırma konsepti [13].

Kısmi yeniden yapılandırma içeren tasarımları daha iyi anlayabilmesi için kullanılan terimler ve açıklamaları aşağıda verilmiştir. FPGA üzerinde kısmi yeniden yapılandırma içeren tasarımlarda, tasarım alanı iki farklı bölgeye ayrılır. Bu bölgeler, statik bölge ve yeniden yapılandırılabilen bölgedir. Şekil 3.3'te koyu renkli bölge yeniden yapılandırılabilen bölgeyi, açık renkli bölge ise statik bölgeyi temsil etmektedir. Statik bölge FPGA çalışmaya başladıktan sonra sabit kalır. Herhangi bir kısmi bit katarı dosyasının yüklenmesi, statik bölgenin içeriğini değiştirmez veya etkilemez. Yeniden yapılandırılabilen bölge ise, kısmi bit katarı dosyası yüklenerek tamamen değiştirilebilir. Bir tasarım birden fazla yeniden yapılandırılabilen bölge içerebilir. Kısmi yeniden yapılandırılabilen bölgenin içerisine yüklenecek olan her bir tasarım yeniden yapılandırılabilen modül olarak adlandırılır. Kısmi

yeniden yapılandırma içeren tasarımlarda en az iki tane yeniden yapılandırılabilen modül bulunur. Şekil 3.3'te gösterilen bit katarı dosyalarının tamamı yeniden yapılandırılabilen modüllerdir. Yeniden yapılandırılabilen modül ve statik bölgenin tamamı konfigürasyon olarak adlandırılır. Bir tasarımdaki konfigürasyon sayısı, yeniden yapılandırılabilen modüllere bağlı olarak değişiklik gösterir. Örneğin, bir adet kısmi yeniden yapılandırılabilen bölgesi ve üç adet yeniden yapılandırılabilen modülü olan bir tasarımda toplam üç farklı konfigürasyon mevcuttur [13].

Kısmi yeniden yapılandırma tasarımlara esneklik ve hız kazandırsa da, her tasarım kısmi yeniden yapılandırma içerecek şekilde yeniden tasarlanamayabilir. Kısmi yeniden yapılandırma işleminde bazı kısıtlamalar söz konusudur. Bunlardan en önemli olanı yeniden yapılandırılabilen modüllerin girişlerinin ve çıkışlarının boyutları aynı olmak zorundadır.



Şekil 3.4: Yeniden yapılandırılabilen modül örnekleri.

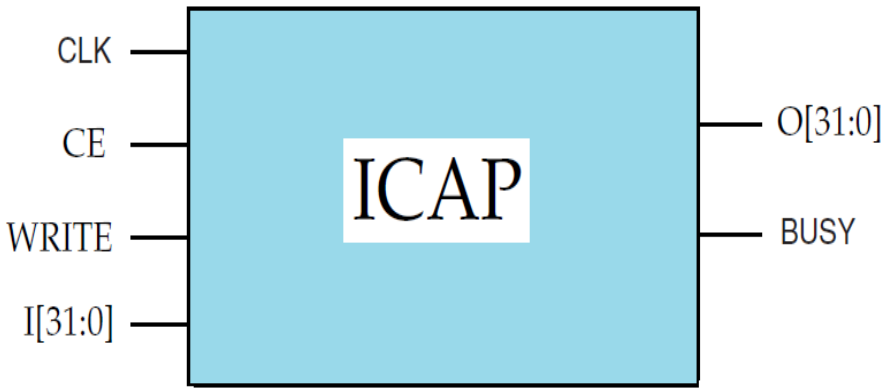
Şekil 3.4'te gösterilen yeniden yapılandırılabilen modüllerin giriş ve çıkışları, kısmi yeniden yapılandırma kısıtlarına göre hazırlanmıştır. Toplama devresi iki adet 8-bitlik sayıyı toplamakta, çarpma devresi ise iki adet 8-bitlik sayıyı çarpmaktadır. Bu iki devrenin FPGA üzerindeki bir tasarımda yeniden yapılandırılabilen bölgeye yüklenecek olan yeniden yapılandirilebilen modüller olacağını varsayalım. Toplama işlemin sonucu ikilik sistemde en fazla 9-bitlik bir sayıya karşılık gelirken, çarpma devresinin sonucu en fazla ikilik sistemde 16-bitlik bir sayıya karşılık gelmektedir. Toplama devresinde 9-bitlik bir çıkış yeterli olduğu halde, kısmi yeniden yapılandırmadaki giriş ve çıkış sayılarının ve boyutlarının aynı olmasını gerektiren

kısıt sebebiyle, toplama devresinin çıkışının boyutu da 16-bit olmak zorundadır. Bu kısıt sebebiyle tasarlanacak olan yeniden yapılandirilebilen modüllerin,giriş ve çıkışların bazılarına ihtiyaçları olmasa dahi, giriş ve çıkışları aynı olmak zorundadır.

Xilinx FPGA'lara bit katarı yüklemek için çeşitli yöntemler mevcuttur, seri yapılandırma, JTAG, SelectMap, SPI flash, BPI flash bellekler aracılığıyla yapılabilir. Seri olarak bit katarı yüklemek için FPGA üzerinde bulunan pinlerden veya yine seri olarak JTAG kapısı ile bilgisayardan direk olarak FPGA'ya bit katarı dosyası atılabilir. SelectMap ara yüzü kullanılarak paralel olarak çift yönlü 8, 16 veya 32 bit veri kapısından bit katarı yüklenebilir veya FPGA üzerindeki tasarım geri okunabilir. Bu çalışmada kullanılan FPGA programlama arayüzü ise ICAP'tir. ICAP, SelectMap ile aynı şekilde bit katarı yükler, ancak SelectMap'ten farklı olarak ICAP FPGA'nın içinde bulunur ve FPGA üzerinde bulunan tasarıma (kullanıcı tarafından tasarlanıp FPGA'ya yüklenmiş tasarım) FPGA'nın yapılandırma devresine erişim imkanı sağlar. Böylece bu çalışmada amaçlanan kimlik doğrulama sonrası kısmi bit katarı dosyasının FPGA'ya yüklenmesi işlemi FPGA içindeki devre tarafından gerçekleştirilir.

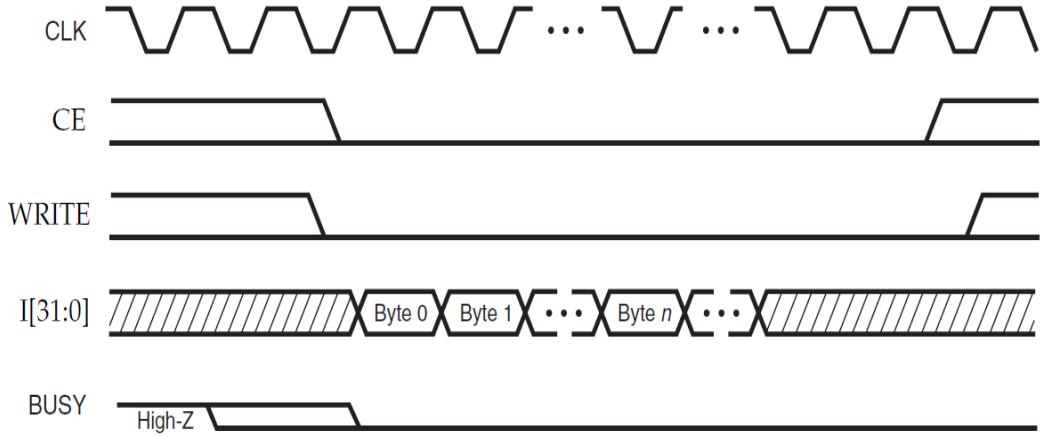
3.2.1 ICAP konfigürasyon portu

FPGA üzerindeki donanım yapılandırma işlemini mümkün kılan konfigürasyon portlarından biride ICAP'tir. ICAP'i diğer konfigürasyon portlarından farklı kılan özelliği, FPGA üzerinde çalışan tasarım tarafından kontrol edilebilir olmasıdır. Bu özelliği sayesinde, FPGA üzerindeki devre uygun tasarlandığı durumda, ICAP konfigürasyon portu kısmi yeniden yapılandırma işlemi için kullanılabilir. Şekil 3.5'te Virtex-5 FPGA'sı üzerinde yer alan ICAP konfigürsayon portunun blok diyagramına yer verilmiştir.



Şekil 3.5: ICAP konfigürasyon portu.

CLK girişi saat sinyalinin girişi, *CE* genel izin sinyali girişi, *WRITE* yazma izin sinyali girişi, *I* veri girişi, *O* veri çıkışı, *BUSY* meşgul sinyali çıkışıdır. *I* veri girişinin boyutu 8, 16 vey 32-bit olarak tasarımcı tarafından değiştirilebilir. ICAP konfigürasyon portunun çalışma zaman diyagramına Şekil 3.6'da yre verilmiştir.



Şekil 3.6: ICAP konfigürasyon portu çalışma zaman diyagramı [14].

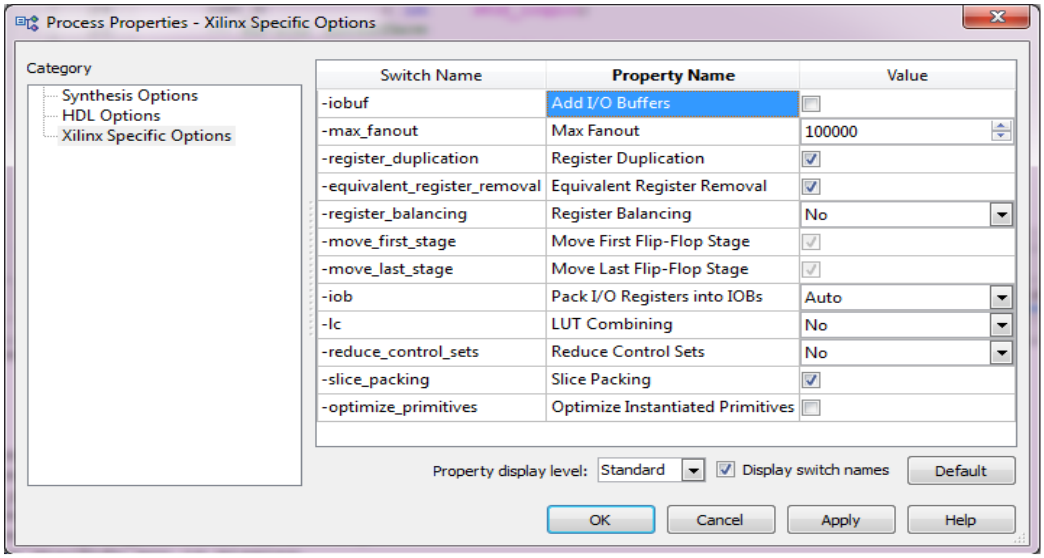
CE ve *WRITE* sinyal girişleri aktif-düşük sinyallerdir. ICAP konfigürasyon portu ile FPGA yapılandırma işlemine başlamadan önce, bu iki sinyalin sıfır seviyesine çekilmesi gerekir. Bu işlem yapılırken dikkat edilmesi gerek nokta, *WRITE* sinyali *CE* sinyalinden önce sıfır seviyesine gelmelidir. Aynı şekilde yapılandırma işlemini sona erdirmek için, ilk önce *CE* sinyali bir seviyesine getirilmeli, daha sonra ise *WRITE* sinyali bir seviyesine getirilmelidir. Eğer bu işlemler yukarıda belirtilen sırayla yapılmazsa, FPGA üzerinde yapılandırma süreci başarısızlıkla sonuçlanır ve FPGA çalışmasını durdurur. ICAP modülünün maksimum saat frekans değeri katalogta 100Mhz olarak verilmiştir. Bu kısıtlamadan dolayı tasarlanan sistemin saat frekansı 100Mhz olarak belirlenmiştir [14].

ICAP konfigürayon portu tasarımda kullanılacak ise, tasarımda mutlaka verilerin doğru alındığını kontrol eden sınama toplamı benzeri bir yapı bulunmalıdır. ICAP konfigürsayon portu girişinden aldığı verilerin doğruluğunu kontrol etmez. Bu verilerin herhangi birinin yanlış olması sistemin yanlış yapılandırılmasına veya FPGA yongasının yanmasına sebep olabilir. Bu gibi sonuçları engellemek adına verilerin doğruluğu mutlaka kontrol edilmelidir. Bu tez çalışmasında verilerin doğruluğunu kontrol etmek için, AES şifreleme algoritmasının GCM modu kullanılmıştır. GCM

modu kimlik doğrulama sağlayarak bir yandan da verilerin doğruluğunu test etmektedir.

3.3 Kısmi Yeniden Yapılandırma Tasarım Süreci

FPGA tasarım sürecine bu tez çalışmasında Bölüm 2.3'te yer veirlmiştir. Xilinx FPGA'larını kullanarak hazırlanacak olan tasarımlarda, bu süreçlerin tamamı (tasarım girişi, simulasyon, sentez, yerleştirme, yönlendirme ve programlama) Xilinx tarafından sağlanan Xilinx ISE geliştirme platformu üzerinde gerçekleştirilebilmektedir. Kısmi yeniden yapılandırma içeren tasarımlarda tasarım sürecini tamamlayabilmek için Xilin ISE geliştirme platformu yeterli değildir. Kısmi yeniden yapılandırma içeren tasarımlarda, tasarım süreci şu şekilde ilerlemektedir. Şekil 2.6'da gösterilen tasarım sürecinin sentezleme işlemine kadar olan kısmı Xilinx ISE geliştirme platformu üzerinde gerçeklenir.



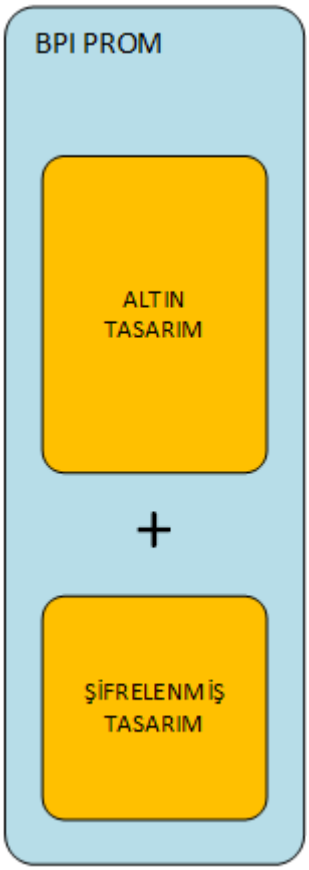
Şekil 3.7: Yeniden yapılandırılabilen modüllerin I/O bağlantıların kaldırılması.

Statik bölgenin ve yeniden yapılandırılabilen modüllerin tasarımı, simulasyonu ve sentezlemesi ayrı ayrı Xilinx ISE geliştirme platformunda yapılır. Yeniden yapılandırılabilen modüllerin sentez işlemi sırasında, Şekil 3.7'de gösterildiği gibi sentez tercihlerinden *Add I/O Buffers* seçeneği kaldırılmalıdır. Bu işlemler sonucunda statik bölge ve yeniden yapılandırılabilen modüllerin netlistleri elde edilir. Bu adımdan sonra tasarım sürecine Xilinx'in Planahead geliştirme platformunda devam edilir.Planahead yerleştirme ve yönlendirme işlemlerinin kolay bir şekilde yapılabilmesi için, Xilinx tarafından üretilmiş olan bir geliştirme ortamıdır. Şekil

2.6'da gösterilen tasarım sürecinin yerleştirme, yönlendirme ve programlama adımları Planahead ile yapılır. İlk olarak FPGA üzerinde yeniden yapılandırılabilen bölge seçilir. Bu bölge seçildikten sonra, bu bölgede çalıştırılacak yeniden yapılandırılabilen modüllerin netlistleri Planahead tarafından sisteme yüklenir. Bu aşamanın ardından tasarım sürecine devam edilerek yerleştirme, yönlendirme ve programlama süreçlerinin sonucunda her bir konfigürasyonun bit katarı dosyası ile yeniden yapılandırılabilen modüllerin kısmi bit katarı dosyaları üretilir. Planahead ile kısmi yeniden yapılandırma içeren tasarımlar yapabilmek için, Xilinx'ten kısmi yeniden yapılandırma lisansının temin edilmesi gerekmektedir [15, 16].

3.4 Hazırlanan Kısmi Yeniden Yapılandırma Tasarımı

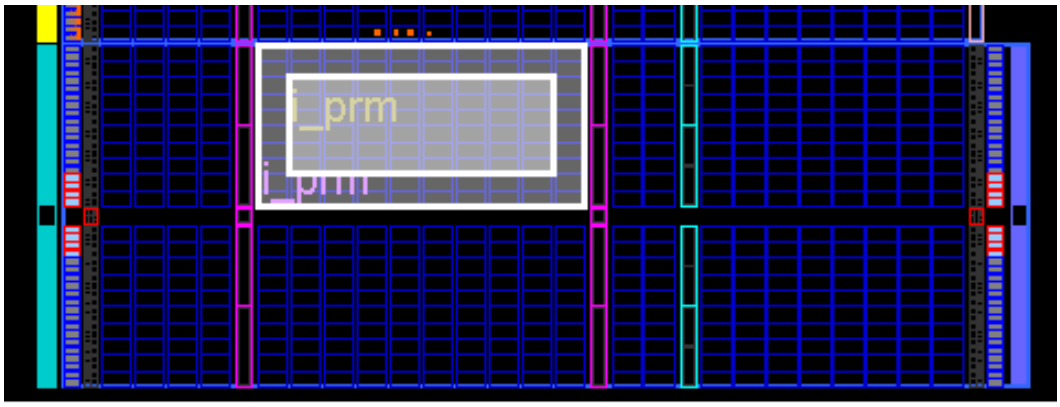
Bu tez çalışmasında önerilen sistemdeki kısmi yeniden yapılandırılabilen bölgeye yerleştirilmek üzere iki farklı yeniden yapılandırılabilir modül hazırlanmıştır. Tasarlanan sistem kendisini BPI PROM üzerinden yükleyecek şekilde tasarlanmıştır. Şekil 3.8'de BPI PROM içerisindeki bit katarı dosyaları gösterilmiştir.



Şekil 3.8: BPI ROM içerisindeki bit katarı dosyaları.

FPGA yongaları üzerindeki tasarımlar, sisteme güç verildiği sürece faaliyet gösterirler. Güç kesildiği anda FPGA üzerindeki tasarım kaybolur. Bu yüzden FPGA yongaları güç kesildikten sonra tekrar çalıştırılmak istendiklerinde yeniden yapılandırılmaları gereklidir. Bu tez çalışmasında tasarlan sisteme güç verilip ilk kez çalıştırıldığında, BPI PROM üzerindeki altın tasarımı içeren bit katarı dosyası ile FPGA üzerindeki donanım yapılandırılır. Altın tasarımın içerisinde birinci yeniden yapılandırılabilen modül yer almaktadır. Tasarlanan ikinci yeniden yapılandırılabilen modülün kısmi bit katarı dosyası şifrelenek BPI PROM üzerine kopyalanmıştır.

Tasarlanan yeniden yapılandırılabilen modüllerden birincisi, FPGA ilk kez çalışmaya başladığında kısmi yapılandırılabilen bölgenin çalıştığını gözlemlemek amacıyla kullanılmıştır. İlk modül içerisindeki tasarlanan devre, duruma göre $A \oplus B$ veya $A \wedge B$ işlemini yapmaktadır. A ve B değerleri ML505 kartı üzerinde yer alan elektrik anahtarlarından alınmaktadır, ve sonuç LED'ler üzerinde gösterilmektedir. İkinci modül içerisindeki tasarlanan devre, duruma göre $\overline{(A \oplus B)}$ veya $A \vee B$ işlemini yapmaktadır. Bu modüllerin yüklenecekleri yeniden yapılandırılabilen bölge Planahed ortamında önceden belirlemiştir. Şekil 3.9'da FPGA üzerinde belirlenmiş olan yeniden yapılandırılabilen bölge gösterilmiştir.

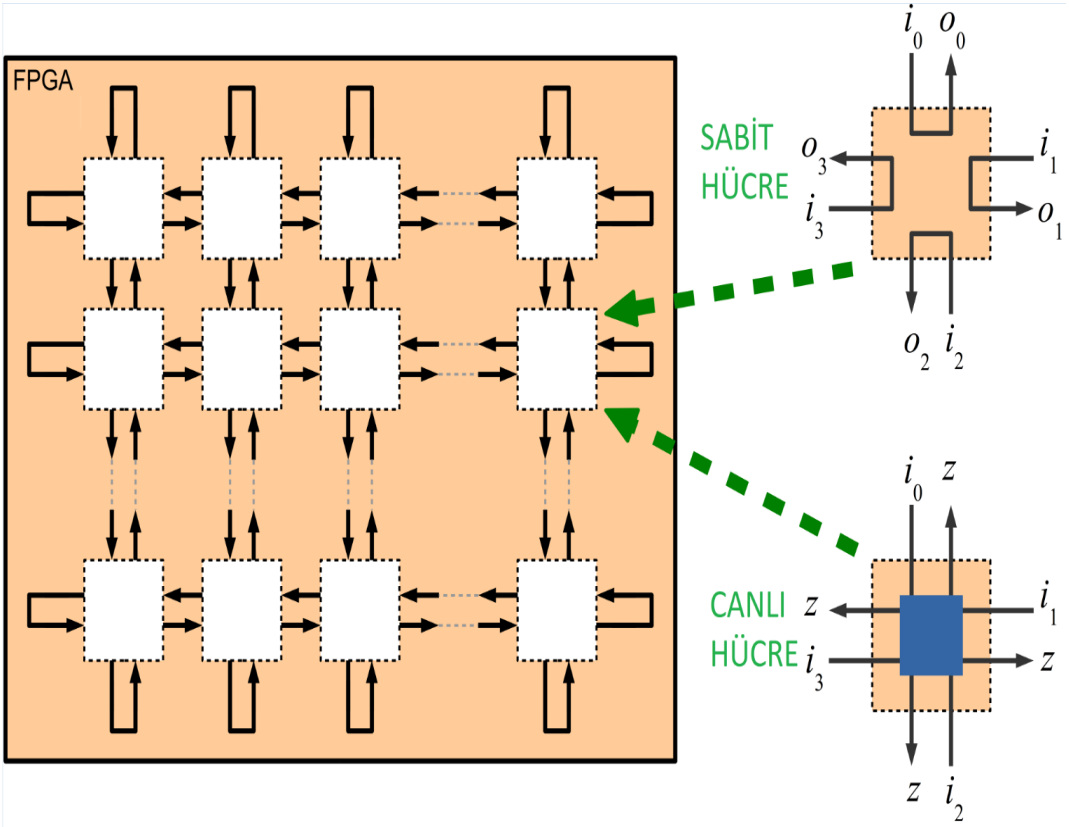


Şekil 3.9: FPGA üzerindeki yeniden yapılandırılabilen bölge.

Sistem çalışmaya başladıktan sonra dışarıdan yeniden yapılandırma yapması için elektrik anahtarlarından biri kullanılmaktadır. Elektrik anahtarı bir seviyesine getirilip sistem uyarıldığında BPI PROM üzerinden şifreli tasarım okunur, ve FPGA içerisine alınır. AES-GCM şifre çözme bloğunda BPI PROM üzerinden okunan şifrelenmiş tasarım çözülür, ve kimlik doğrulaması yapılır. Sistemin kimlik doğrulamadan geçebilmesi için, BPI PROM üzerinden okuma işlemi sırasında herhangi bir sorun yaşanmamış olması ve şifre çözme için kullanılan anahtarın doğru anahtar olması

gerekir. Bu iki şartın sağlandığı durumda, ICAP üzerinden yeniden yapılandırılabilen bölgeye tasarım yüklenerek kısmi yeniden yapılandırma işlemi başarıyla tamamlanır. Kısmi yeniden yapılandırma tamamlandığında, yeniden yapılandırma bölgesinde birinci yeniden yapılandırılabilen modül tamamen kaldırılarak, yerine ikinci yeniden yapılandırılabilen modül çalışmaya başlar.

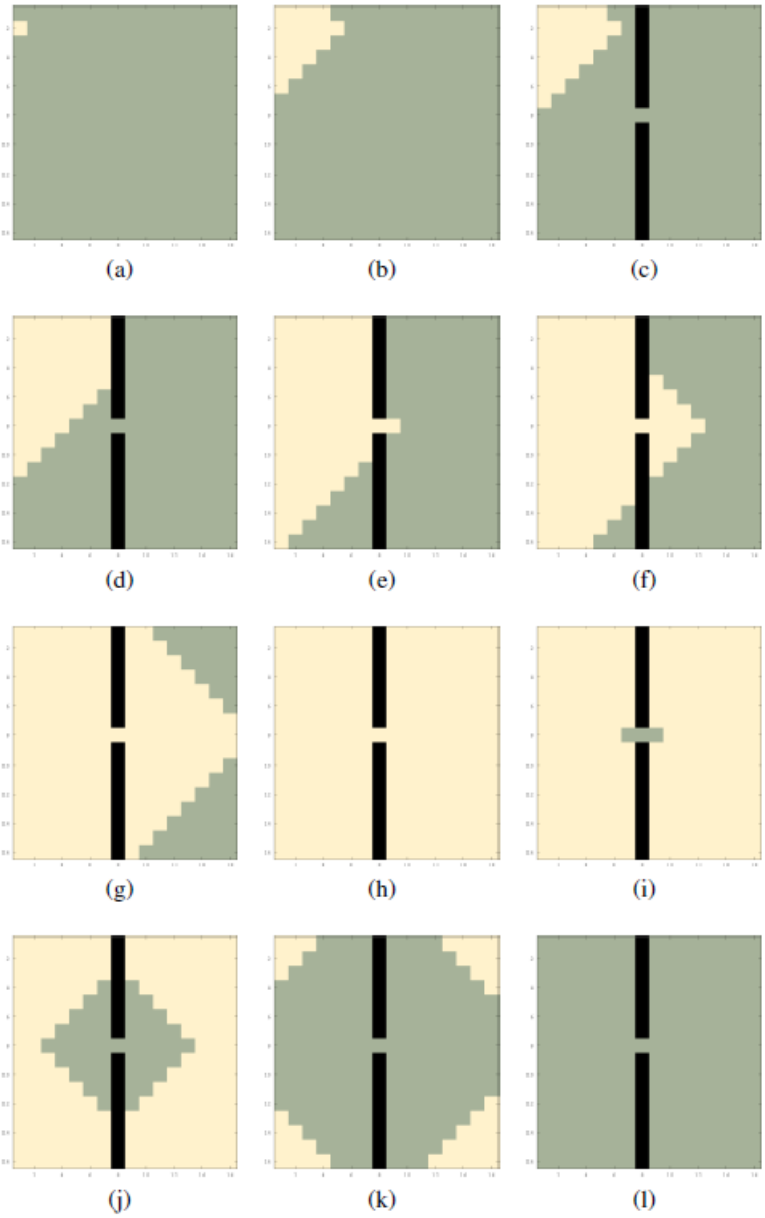
Tez çalışması sürecinde, kısmi yeniden yapılandırmaya daha iyi hakim olabilmek adına çalışmalar yapılmıştır. Bu çalışmalar sonucunda [17]'de belirtilen bildiri hazırlanmıştır. Bu bildiride kısmi yeniden yapılandırma özelliğine sahip hücresel doğrusal olmayan ağ (CNN) yapısı FPGA üzerinde tasarlanmıştır.



Şekil 3.10: Hazırlan CNN hücreleri ve ağ yapısı [17].

Şekil 3.10'da bildiride hazırlanan CNN hücreleri ve bu hücreler ile oluşturulan ağ yapısı gösterilmiştir. Bu ağ üzerinde ikili sistem üzerinde tetiklemeli dalga simulasyonu yapılmıştır. Ağ üzerindeki bir bölge kısmi yeniden yapılandırmaya uygun olarak hazırlanmıştır. Bu bölge için iki farlı hücre tasarımı yapılmıştır. Bu hücreler sabit hücre ve canlı hücre olarak adlandırılmıştır. Canlı hücre, tetiklemeli dalgaların yayılmasına olanak sağlarken, sabit hücre bir engel davranışı sergilemektedir.

Şekil 3.11'de hazırlanan sistem üzerinde belli bir senaryoya uygun simulasyon sonuçları gösterilmiştir. Beyaz renk, üzerinden dalga geçen canlı hücreyi, yeşil renk üzerinden dalga geçmeyen canlı hücreyi, siyah renk ise sabit hücreyi yani engeli temsil etmektedir.



Şekil 3.11: CNN ağı üzerinde gerçeklenen simulasyon sonuçları [17].

Sisteme sol üst köşeden bir sinyal uygulanarak, belli bir süre bu dalganın yayılması sağlanmıştır. Daha sonrasında kısmi yeniden yapılandırma işlemi ile canlı hücrelerin bir kısmı, sabit hücreler ile değiştirilmiştir. Şekil 3.11'de a'dan h'ye kadar olan şekiller bu senaryonun sonuçlarını göstermektedir. Şekil 3.11'de i'den l'ye kadar olan şekiller, ağın tam ortasına, üzerinden dalga geçmeyen canlı bir hücrenin kısmi yeniden yapılandırma ile yerleştirilmesi sonucu ortaya çıkan davranışı sergilemektedir. Bu

sonuçlara göre, canlı hücre sisteme eklenirken o bölgede ki var olan durumdan farklı bir durum ile yüklenirse bir dalga kaynağı gibi davranıp sistemde istenmeyen sonuçlara neden olabilmektedir.

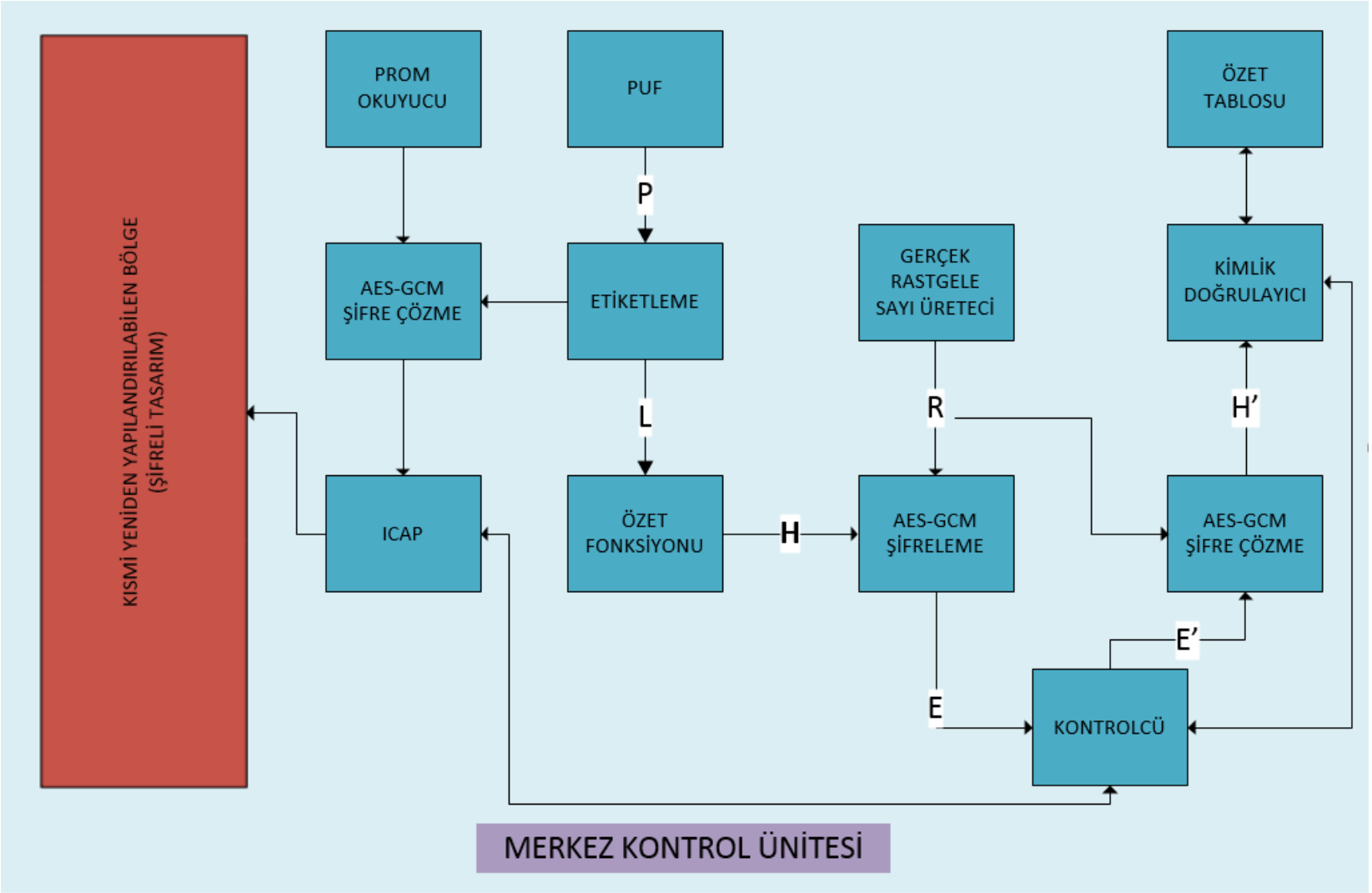
4. SİSTEM MİMARİSİ

Bu bölümde sistem mimarisi hakkında bilgiler ve açıklamar yer almaktadır.

4.1 Önerilen Sistem Mimarisi

Tezin amacında belirtilen hedefe ulaşabilmek için önerilen sistem mimarisi bu bölümde anlatılacaktır. Şekil 1.1'de gösterilen sistemin kontrolü, eyleyicilerdeki yerel kontrolcüler ve bu yerel kontrolcüleri kontrol eden merkez kontrolcüler tarafından gerçekleştirilmektedir. Bu kontrolcüler sahada programlanabilen kapı dizileri (FPGA) üzerinde yer almaktadır. Bu tez çalışmasında Xilinx firması tarafından üretilen ML505 geliştirme kartı kullanılmıştır. Bu kart üzerinde Virtex 5 serisi bir FPGA yer almaktadır. Geliştirme kartını programlamak için kullanılan bit katarı (bitstream) dosyaları, kart üzerinde yer alan 32MB kapasiteye sahip BPI PROM üzerinde tutulmaktadır. Bu çalışmada geliştirme kartı üzerindeki FPGA'nın ön yükleme sürecindeki bit katarı dosyası ile aktif çalışma sürecinde yüklenen şifrelenmiş olan bit katarı dosyası BPI PROM üzerinde tutulmaktadır. Kart her yeniden başlatma işleminde, programlama işlemini BPI PROM üzerinde yer alan bit katarı dosyası yardımı ile yapmaktadır. Şifrelenmiş olan bit katarı içerisinde saklanmak istenen fikri mülkiyet (IP) tasarımı bulunmaktadır. Bu bit katarını şifrelemek için AES şifreleme algoritması kullanılmıştır. Ayrıca BPI PROM üzerinden okunan bit katarının herhangi bir dış etkiyle değişmediğini doğrulamak adına sınama toplamı (checksum) yapan GCM modu AES algoritması ile birlikte kullanılmıştır. AES-GCM şifreleme algoritması ve gerçeklemesi hakkındaki bilgilere bu tezin bu bölümünde yer verilmiştir.

Şifrelenmiş olan bit katarı FPGA üzerinde belirlediğimiz kısmi olarak yeniden yapılandırılabilen bölgeye yüklenecektir. Kısmi yapılandırma (PR) Virtex 5 FPGA'larında bulunan bir özelliktir. AES-GCM şifreleme algoritmasında, 128-bitlik şifreleme anahtarını üretmek için fiziksel olarak klonlanamayan fonksiyonlardan (PUF) yararlanılmıştır. PUF çeşitlerinden halka osilatörü kullanılmıştır.



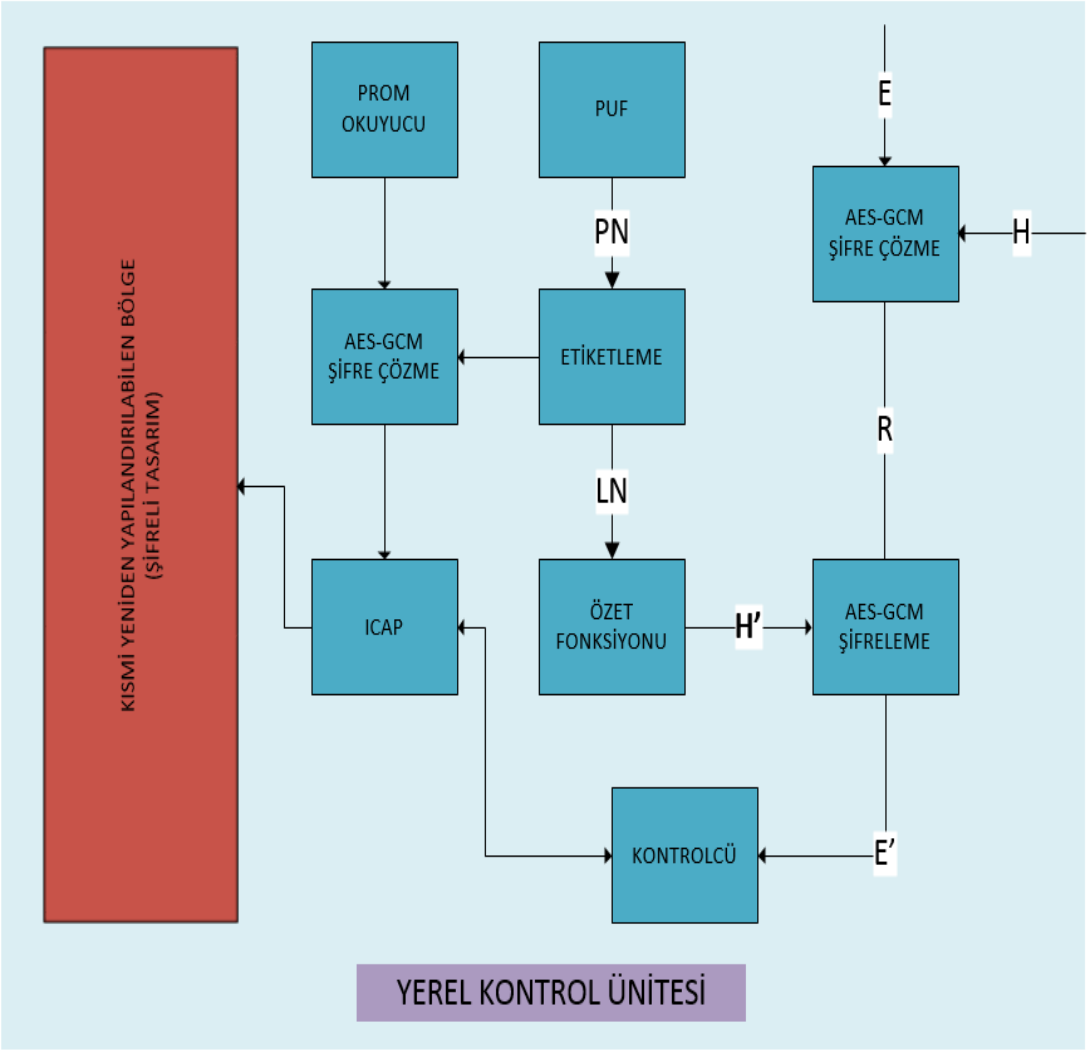
Şekil 4.1: Merkez kontrolcü iç yapısı.

Halka osilatörlerinin sıcaklığa karşı gösterdiği değişimlerden etkilenmemek için [18]'te önerilen etiketleme yöntemi uygulanmıştır. Tezin bu bölümünde fiziksel olarak klonlanamayan fonksiyonlar ve etiketleme yöntemi yer almaktadır.

Şekil 4.1'de merkez kontrolcünün iç yapısı verilmiştir. Kimlik doğrulaması ve yerel kontrolcülerin kontrolü, merkez kontrolcünün temel iki görevidir. Kimlik doğrulama işlemi, FPGA üzerindeki statik kısımda yapılır. Yerel kontrolcülerin kontrolü ise kısmı olarak yeniden yapılandırılan bölgede yapılır.

Merkez kontrolcü çalışmaya başladıktan sonra ilk olarak, halka osilatörleri çalıştırarak 128-bitlik *P* değerini elde eder. Elde edilen bu *P* değeri etiketleme işleminden geçirilerek *L* değeri elde edilir. *L* değeri, merkez kontrolcünün kısmı olarak yeniden yapılandırılabilen bölgesine yüklenecek olan bit katarını şifrelemek ve şifresini çözmek için kullanılır. BPI PROM üzerinden şifrelenmiş bit katarı okunarak, merkez kontrolcü üzerinde kısmi konfigürasyon işlemi yapılır. Kimlik doğrulama işlemi için *L* değeri özet fonksiyonundan geçirilerek *H* değeri elde edilir. *H* değeri tüm yerel kontrolcülerin içerisinde gömülü olarak yer alır. Yerel kontrolcülerden kimlik doğrulama isteği geldiğinde, gerçek rastgele sayı üretecinin çıktısı olan *R* değeri, *H* ile şifrelenerek ilgili yerel kontrolcüye gönderilir. Yerel kontrolcüden gelen *H'* değeri eğer doğruysa, ilgili yerel kontrolcü kimlik doğrulamasından geçmiş sayılır. Bu süreç sonunda kimlik doğrulamasından geçen yerel kontrolcü ile merkez kontrolcü arasındaki iletişim *R* ile şifrelenerek gerçekleştirilir. *R* değeri her yerel kontrolcü için farklıdır.

Şekil 4.2'de yerel kontrolcünün iç yapısı verilmiştir. Yerel kontrolcü çalışmaya başladığında ilk olarak kimlik doğrulama için merkez kontrolcü ile iletişime geçer. Merkez kontrolcüden gelen *E* değeri, *H* değeri ile çözülerek *R* değeri elde edilir. Etiketleme işleminin sonucu olan *LN* değeri özet fonksiyonundan geçirilerek *H'* değeri elde edilir. *H'* değeri *R* ile şifrelenerek *E'* değerine ulaşılır ve bu değer merkez kontrolcüye gönderilir. Eğer merkez kontrolcü bu değerin doğru olduğunu kabul eder ve onay mesajını gönderirse, yerel kontrolcü üzerindeki kısmı olarak yeniden yapılandırabilen bölgeye BPI PROM üzerindeki şifrelenmiş bit katarı yüklenir. Yerel kontrolcülerde, kısmı olarak yeniden yapılandırılabilen bölgeye yüklenecek olan bit katarları, her bir yerel kontrolcünün kendi etiket değeri (LN) ile şifrelenirler.



Şekil 4.2: Yerel kontrolcü iç yapısı.

Bu sistemin çalışırtırılması için, her bir yerel kontrolcünün etiket değerleri üretim süreci safhasında [18]'de belirtilen yöntem ile önceden sistemin dışına alınmalıdır. Sistem dışına alınan etiket değelerleri ile bit katarları ilgili oldukları kontrolcünün etiket değeri ile şifrelenmelidir. Burada üreticinin bu etiket değerlerini üçüncü şahıslarla paylaşmadığı varsayılmaktadır. Bu sistemde özet fonksiyonlarının kullanılmasının amacı, bit katarlarını şifrelemek için kullanılan etiket değerlerinin sadece ilgili kontrolcü tarafından bilinmesini sağlayarak, tek bir kontrolcü üzerine dışarıdan yapılacak saldırıların diğer kontrolcülerin güvenliğini etkilemesinin önüne geçebilmektir. Önerilen bu sistem ile robotun parçaları klonları ile değiştirilip yazılımları kopyalansa dahi, kimlik doğrulamanın fiziksel olarak klonlanamayan fonksiyonlara bağlı olmasından dolayı, yazılım klonlanmış donanım üzerinde çalışmayacaktır.

4.2 Fiziksel Klonlanamayan Fonksiyonlar

Fiziksel olarak klonlanamayan fonksiyon (PUF), karakteristik özelliklerinin modellenmesi zor olan bir fiziksel sistem tarafından kolay bir şekilde hesaplanan fonksiyonların genel adıdır. Bu fonksiyonlar belli bir sayıdaki giriş setini, fiziksel sistemin iç yapısına bağlı olarak rastgele ve eşsiz bir çıkış setine dönüştürürler. Bu özellikleri sayesinde iki farklı yapı üzerinde oluşturulan aynı yapıdaki PUF'lar birbirinden farklı sonuçlar üretirler. PUF'lar çok çeşitli ortamlardan elde edilebilir. Literatürde yer alan PUF çeşitlerine akustik, optik, manyetik, radyo frekansı, manyetik PUF'lar örnek olarak verilebilir [19, 20].

PUF'lar üretim süreçlerinde ortaya çıkan idealsizliklerden yararlanılarak elde edildikleri için, kullanıldıkları ortamların ve süreçlerin değişkenliği, sıcaklığın değişmesi ve gerilimdeki dalgalanmalar, güvenilir sonuçlar vermelerini etkileyebilmektedir. [19]'de bahsedilen özgün silikon PUF çeşitleri bu değişimlerin etkisini azaltmak için kullanılmaktadırlar. Özgün silikon PUF'lar aynı şekilde oluşturulmuş gecikme yolu çiftlerinin çıkışlarının birbiriyle karşılaştırılması prensibine göre çalışırlar. Bu bölümün ilerleyen kısımlarında özgün PUF'lara yer verilmiştir.

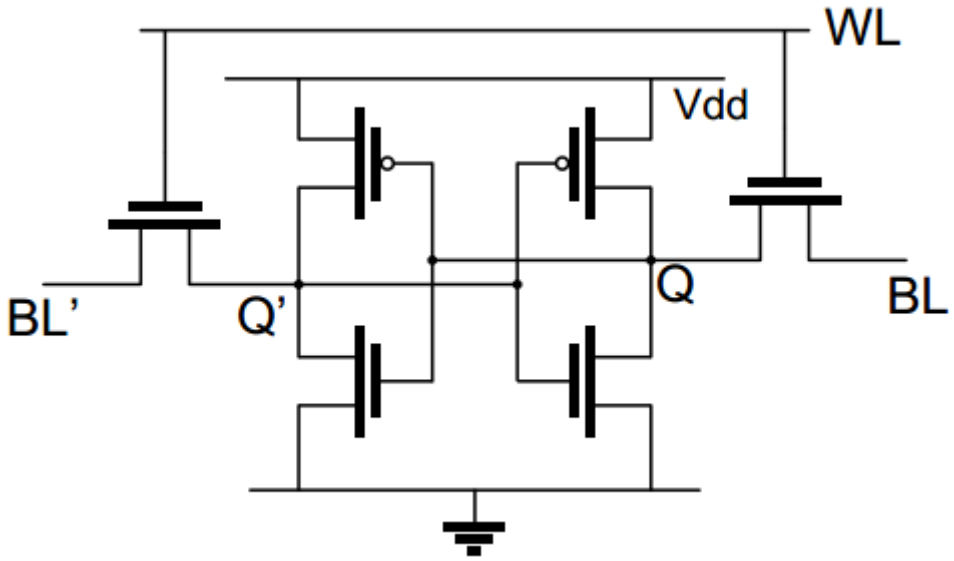
PUF'lar maliyeti düşüren ve klonlanması zor yapılar olduklarından günümüzde kimlik doğrulama süreçlerinde kullanılmaktadırlar [21]. PUF'lar, tümdevrelerin üretim sürecinde ortaya çıkan kompleks fiziksel karakteristiklerinden, kimlik doğrulama süreçlerinde kullanılmak üzere anahtar üretmek için kullanılabilirler. Örneğin, kabloların ve transistörlerin değişken gecikme karakteristiklerinden yararlanılarak, aygıta göre değişen anahtar üretimi gerçekleştirilebilir. Bu tez çalışmasında kimlik doğrulama sürecinde kullanılacak anahtarı üretmek için özgün silikon PUF kullanılmıştır. Tezin ilerleyen kısımlarında özgün silikon PUF'lardan PUF olarak bahsedilecektir.

4.3 Özgün Silikon PUF Çeşitleri

4.3.1 SRAM PUF

SRAM PUF'ları, SRAM hücrelerinden sayısal işaret üretirler. Bir SRAM hücresi çapraz bağlanmış iki adet eviriciden oluşur. Bu eviricilere bağlanan iki transistör,

okuma ve yazma operasyonlarını gerçeklemek için kullanılır. Şekil 4.3'te SRAM hücre yapısına yer verilmiştir.

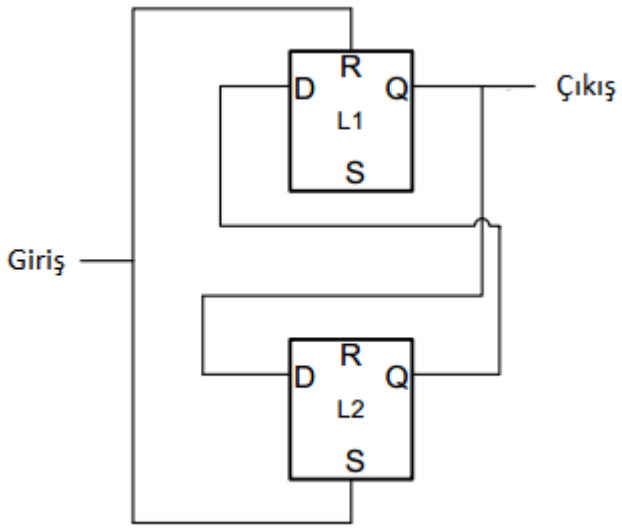


Şekil 4.3: SRAM hücre yapısı.

İdeal durumda, SRAM yapısındaki çapraz bağlanmış eviricilerin eşik gerilimleri aynıdır. Pratikte ise, kutuplama konsantrasyonlarının farklılığı, gerilim ve sıcaklık gibi dış etmenlerin etkisiyle eşik gerilimleri birbirinden farklı halde gelir. SRAM uçucu hafıza depolama birimidir. SRAM'e verilen güç kesildiğinde, daha önceden içerisinde tutmuş olduğu veri biti kaybolur. SRAM'e güç verildikten sonra, çapraz bağlanmış eviricilerin eşik gerilimlerinin farklı olması sebebiyle, SRAM hücresi sayısal 0 ya da sayısal 1 sabit durumlarından birine geçer. Çapraz bağlanmış eviricilerin eşik gerilimleri arasındaki fark sabit kaldığı sürece, her yeniden başlatma işleminde sürekli olarak aynı durumda (sayısal 0 ya da sayısal 1) ile çalışmaya başlar. Bu sabit durumlar SRAM hücreleri arasında farklılık göstermektedir. SRAM hücrelerine güç verilerek çalıştırılmaya başlandığı anda, SRAM hücrelerinin durum değerleri okunarak SRAM PUF elde edilir. SRAM PUF'larda ki bu değerleri okumak için bir mikrokontrolcüden yararlanılabilinir.

4.3.2 Kelebek PUF

Kelebek PUF hücresinin yapısı Şekil 4.4'te gösterilmiştir. Kelebek PUF hücresi çapraz bağlanmış eş iki D tutucusundan oluşur. Hücrenin çıkışında bu tutucuların arasında hız farkını ölçerek bir değer elde edlilir.



Şekil 4.4: Kelebek PUF yapısı.

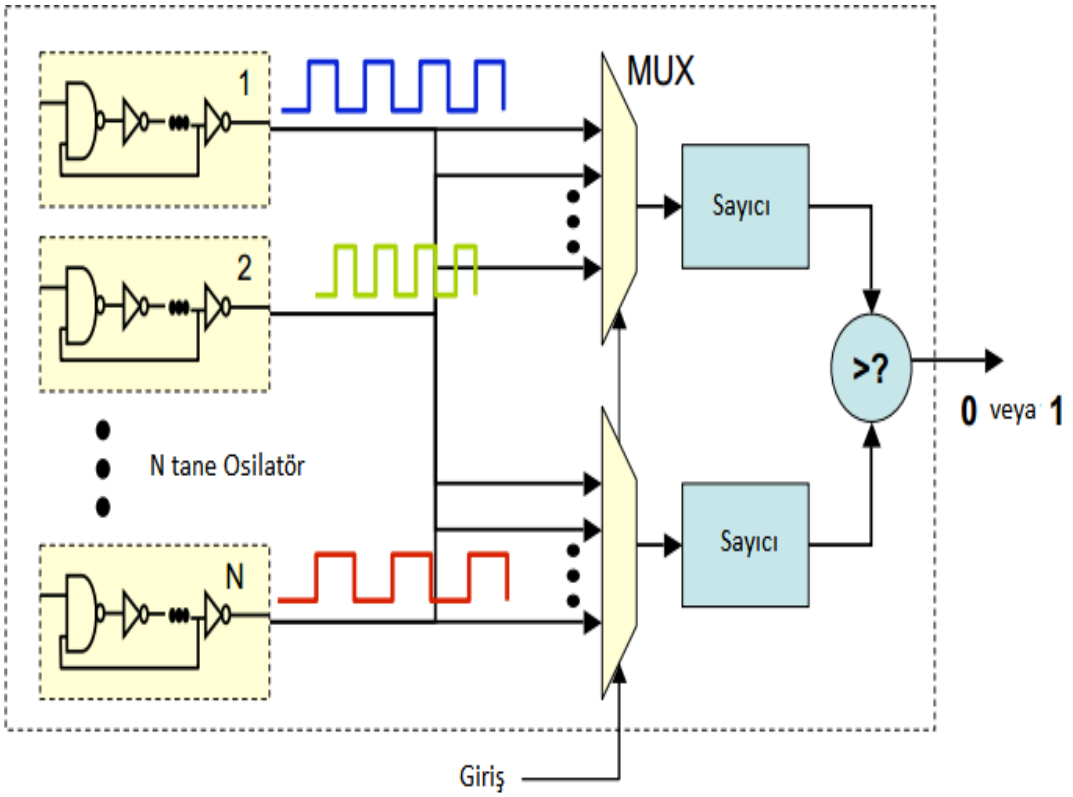
Şekilde görülen L1 tutucusunun reset pini ve L2 tutucusunun set pini Giriş sinyaline bağlıdır. Giriş sinyali sayısal bir değerine çekildiğinde, L1 tutucusunun çıkışı sayısal sıfır değerine, L2 tutucusunun çıkışı sayısal bir seviyesine gelir. Giriş sinyali tekrardan sıfıra çekildiğinde tutucular birbirlerinin çıkışlarını yakalamaya başlarlar. Bu tutucular arasında bir yarış durumu yaratarak, kelebek PUF hücresinin çıkışında sürekli olarak sayısal bir yada sıfır değerinde kalmasını sağlar.

4.3.3 Halka osilatör tabanlı PUF

Halka osilatör tabanlı PUF yapısı [21]'de önerilmiştir. Diğer PUF çeşitlerine göre FPGA üzerinde uygulanması daha kolay olmasıyla birlikte daha güvenilir bir yapıdır. Diğer bir yandan, diğer PUF çeşitlerine göre daha yavaş çalışıp daha fazla enerji tüketir. Şekil 4.5'te halka osilatör tabanlı PUF gösterilmiştir.

Halka osilatörü tek sayıda arka arkaya bağlanmış eviricilerden oluşur. En sondaki eviricinin çıkışı pozitif geri besleme yapılarak, halka osilatörünün girişinde bulunan ilk eviriciye bağlanır. Pozitif geri besleme sayesinde halka osilatörünün çıkışında bir kare dalga üretilir. Bu kare dalganın frekansı eviricilerin gecikmesine bağlı olarak değişir. Halka osilatör tabanlı PUF'un çalışma prensibi eviricilerin gecikmelerinden kaynaklanan bu değişime bağlıdır.

Matematiksel modeli aynı olan iki tane halka osilatör çifti aynı anda çalıştırılarak bir hız yarışına sokulurlar. Eviricilerin gecikmeleri farklı olduğundan, her iki çiftin çıkışında gözlemlenen kare dalgaların frekansı farklı olacaktır. Üretilen bu kare

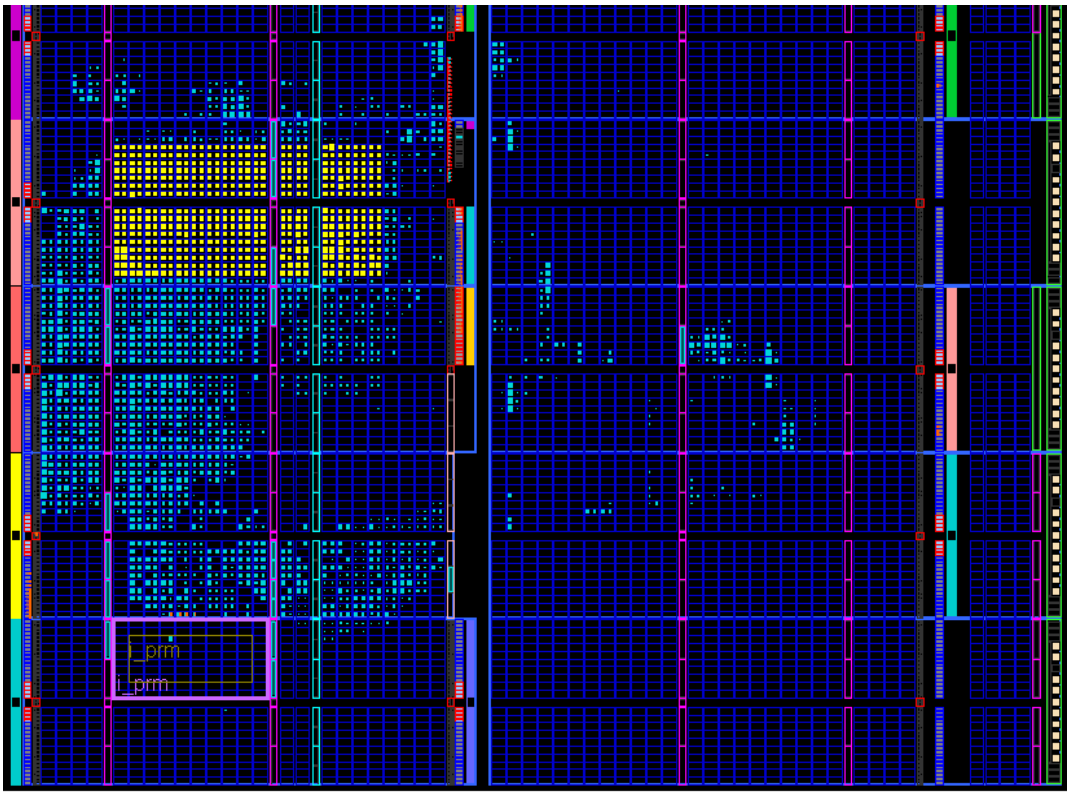


Şekil 4.5: Halka osilatör tabanlı PUF yapısı [21].

dalgalar ile sayıcı devreleri saydırılır. Sayıcı devrelerinden herhangi biri önceden belirlenen bir değere ulaştığında, halka osilatörleri durdurularak sayıcıların çıkışları karşılaştırılır. Karşılaştırma işleminde büyük veya küçük eşit durumlarına göre çıkışta sayısal bir ya da sayısal sıfır değeri üretilir. Bu şekilde halka osilatör çiftleri kullanılarak istenilen uzunlukta (96-bit, 128-bit, 256-bit vs.) rastgele ve yongaya özgü bir anahtar üretmek mümkündür. Halka osilatör tabanlı PUF'un yavaş çalışmasının sebebi, her bir halka osilatör çiftinin belli bir değere kadar sayması, ve bu işlemin her bir çift için sırayla gerçekleşmesidir.

4.4 FPGA Üzerinde Halka Osilatör Tabanlı PUF Gerçeklemesi

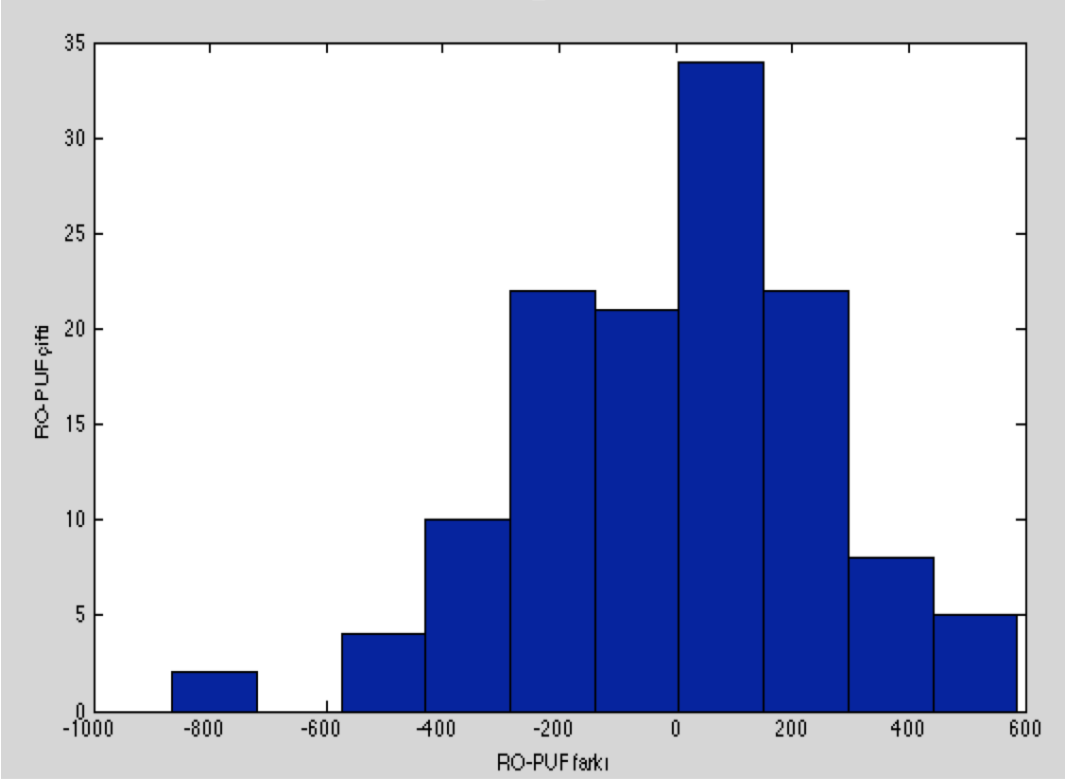
Bu tez çalışmasında kimlik doğrulama işleminde kullanılacak anahtarı üretmek için halka osilatör tabanlı PUF kullanılmıştır. 7 tane eviriciden oluşan halka osilatör modeli kullanılmıştır. Tasarlanan halka osilatör PUF devresinde 128-bitlik bir anahtar üretebilmek için 128 adet halka osilatör çifti kullanılmıştır. Halka osilatör çiftlerinin uçlarındaki sayıcılar 32768 değerine ulaşana kadar çalıştırılmışlardır ve bu değere ulaştıklarında iki sayıcı arasındaki fark kaydedilmiştir. Şekil 4.6'da sarı renkli gösterilen bloklar halka osilatör PUF devresini göstermektedir.



Şekil 4.6: FPGA üzerinde halka osilatör tabanlı PUF devresi.

Halka osilatör tabanlı PUF devresinin FPGA yongası üzerindeki yeri önceden sabitlenmiştir. Bu sabitleme işleminin sebebi PUF devresinin çıkışında elde edilecek değerlerin değişiklik göstermesini engelleyebilmektir. Tasarlanan PUF devresi FPGA üzerine yüklenerek ölçüm yapmak için test ortamı oluşturulmuştur. ML505 geliştirme kartı, RS-232 seri haberleşme portu üzerinden bilgisayara bağlanarak MATLAB ile haberleştirilmesi sağlanmıştır. MATLAB ortamında hazırlanan bir program ile yaklaşık olarak bir gün boyunca 128 çift halka osilatörü çalıştırılmıştır. Bu test sonucunda her bir osilatör çifti 140.000 adet fark verisi elde edilmiştir. Şekil 4.7'de elde edilen bu fark verilerinin ortalamasının histogramı çizdirilmiştir.

Histogramda görüldüğü üzere halka osilatörlerin farkları Gauss dağılımına benzer bir yapıdadır. Elde edilen veriler incelendiğinde 128 halka osilatör çiftinin yaklaşık 20 tanesinin kararsız davranış gösterdiği gözlemlenmiştir. Bu 20 çiftte elde edilen sonuçlarda iki halka osilatörü arasındaki yarışın kazananı sürekli bir değişim sergilemektedir. Ortam sıcaklığındaki değişiklikler ve gerilimdeki dalgalanmlar bu duruma sebep olmaktadır. Ortam sıcaklığı arttıkça halka osilatörlerin ürettikleri kare dalgaların frekanslarında bir azalma meydana gelmektedir. Sıcaklığa bağlı frekanstaki azalma miktarı halka osilatör çiftleri arasında farklılık göstermektedir. Örneğin



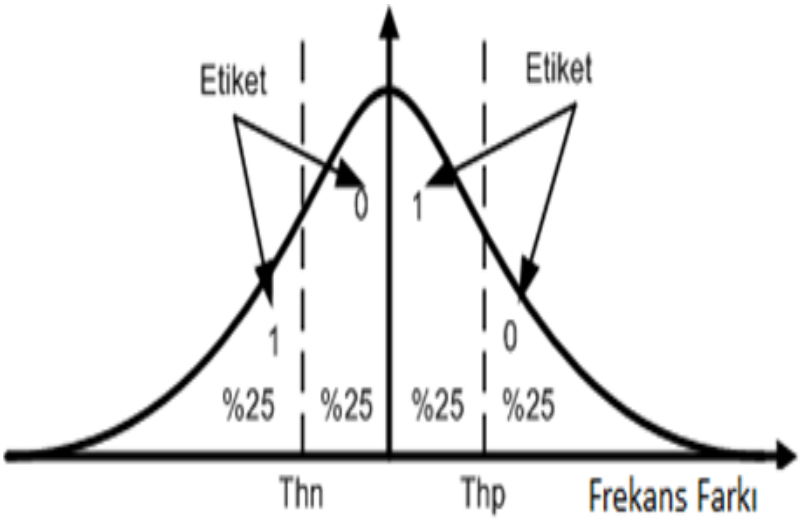
Şekil 4.7: PUF devresinden elde edilen farkların ortalamasının histogramı.

düşük sıcaklıklarda daha yüksek frekansa sahip olan halka osilatörü, sıcaklığın artmasıyla frekansı diğer halka osilatörün frekans değerinin altına düşebilmektedir. Bu istenmeyen bir davranış olup sistemin kimlik doğrulama sürecinden geçmesine engel olacak bir durumdur. Kimlik doğrulama sürecinde kullanılacak anahtar üretimi için, halka osilatör çiftlerinde gözlemlenen bu durumu engellemek adına Kaya'nın [18] 'de önermiş olduğu etiketleme yöntemine başvurulmuştur.

4.5 Etiketleme Yöntemi

Halka osilatör tabanlı PUF devresinin çıkışı, halka osilatör çiftlerinin frekans farkına göre belirlenmektedir. Halka osilatörlerinin çıkışına bağlı olan sayıcıların çıkışları karşılaştırılarak o halka osilatör çifti için çıkış değeri hesaplanmaktadır. Bu süreç genel olarak, ilk sayıcının çıkış değerinden ikinci sayıcının çıkış değerini çıkartarak elde edilen farkın işaretini halka osilatör çiftinin çıkışı olarak kabul etmektir. Ancak bu farkın çok az olduğu durumlarda, çevreden kaynaklı oluşabilecek gürültülerden dolayı PUF çıkışının sürekli değişimi gözlenebilir. Kaya [18]'de önermiş olduğu etiketleme yöntemi ile bu sorunu çözmeye çalışmıştır.

Etiketleme yönteminde, ilk olarak kullanılacak olan FPGA yongası üzerinden ölçüm değerleri toplanır. Bu ölçümlerin istatistiklerine dayanarak karar verme mekanizmaları oluşturulur.



Şekil 4.8: Halka osilatör çiftlerinin etiketlenmesi.

Şekil 4.7'de görüldüğü üzere halka osilatör çiftlerinin frekans farkları Gauss dağılımına benzer bir yapı sergilemektedir. Etiketleme yönteminde de, halka osilatörlerin frekans farklarının Gauss dağılımı sergiledikleri varsayımıyla Şekil 4.8'de gösterilen işlem uygulanmaktadır. İlk olarak *Thn* ve *Thp* belirlenir. Bu değerler Gauss eğrisinin altında kalan alanı dört eşit parçaya bölecek şekilde belirlenir. Bir halka osilatör çiftinin frekans farkı *Thp* değerinden büyük veya *Thn* ve '0' değeri arasında ise, etiket değeri '0' olurken, diğer durumlarda ise etiket değeri '1' olarak belirlenir. Bu işlemler sonucunda elde edilen etiket değerleri yardımcı veri olarak adlandırılır. Yardımcı veriyi h_i ile, halka osilatör çiftlerindeki frekans farkını x_i ile belirtirsek, yukarıda anlatılan yöntemin özeti aşağıdaki gibidir.

$$h_i = \begin{cases} 1 & \text{if} \quad x_i < thn \quad or \quad 0 <= x_i < thp \\ 0 & \text{if} \quad thn <= x_i < 0 \quad or \quad thp <= x_i \end{cases} \tag{4.1}$$

Halka osilatör çiftlerinin etiketleri PUF çıkışı hakkında herhangi bir bilgi vermemektedir. Etiketleme yöntemi, çevresel etkilere bağlı PUF çıkışındaki değişimleri başarılı bir şekilde ortadan kaldırabilir. Yardımcı veri üretildikten sonra aşağıdaki adımlar uygulanarak PUF çıkışı elde edilir,

- Halka osilatörlerin frekans farkları tekrardan ölçülür.

- Denklem 4.1'de ki kurala göre yeni etiket değerleri üretilir.
- Üretilen yeni etiket değerleri, önceden elde edilen yardımcı veri ile aynıysa bu etiket değeri doğru kabul edilir. Yanlış ise en yakın bölgeye taşınır ve PUF çıkışı yeni bölgeye göre düzeltilir.

Yukarıda anlatılan süreç aşağıda verilen kural ile özetlenebilir.

$$z_i = \begin{cases} 1 & \text{if} \quad y_i < thn_M \\ w'_i & \text{if} \quad thn_M <= y_i < thn_p \\ 0 & \text{if} \quad y_i > thp_M \end{cases} \tag{4.2}$$

Burada $thn_m = Thn/2$ ve $thp_m = Thp/2$'dir.

4.5.1 Etiketleme yönteminin sistem üzerinde uygulanması

Yukarıda anlatılan etiketleme yöntemi Bölüm 4.4'te karşılaşılan PUF çıkışlarının sürekli değişimini engellemek için kullanılmıştır. Elde edilen fark verilerinin ortalama değerleri Denklem 4.1'de verilen kuralda x_i değeri yerine yazılarak kullanılmıştır. Bu işlem sonucunda kullandığımız ML505 kartı üzerindeki Virtex-5 FPGA'sı için yardımcı veri elde edilmiştir. Yardımcı veri elde edildikten sonra Denklem 4.2 ile belirtilen yöntem uygulanarak etiketlenmiş PUF çıkışı elde edilmiştir.

Çizelge 4.1: Yardımcı veri ve PUF çıkışı.

	Değer
Yardımcı Veri	01DA6442B910ADF6467E16E24FF66D74
PUF Çıkışı	5BC94FBC3221A5DB94FAE95AE7121A47

Buraya kadar uygulanan etiketleme yöntemi, bilgisayar ortamında yapılmış, yardımcı veri ve PUF çıkışı değerleri üretilmiştir. Kimlik doğrulamanın sağlıklı bir şekilde yapılabilmesi için yukarıda bahsedilen etiketleme yöntemini uygulayan devrenin FPGA üzerinde gerçeklenmesi gerekmektedir. Elde edilen yardımcı veri, *Thn* ve *Thp* değerleri kullanılarak etiketleme yöntemini gerçekleyen devre FPGA üzerinde tasarlanmıştır. Elde edilen PUF çıkışı, FPGA üzerindeki yeniden yapılandırılabilen bölgeye yüklenecek olan ikinci yeniden yapılandırılabilen modülü şifrelemek için kullanılmıştır. Şekil 3.8'de gösterilen şifreli tasarımın şifreleme anahtarı, tez çalışmasının bu aşamasında elde edilen PUF çıkışı değeridir.

4.6 AES-GCM Şifreleme Algoritması

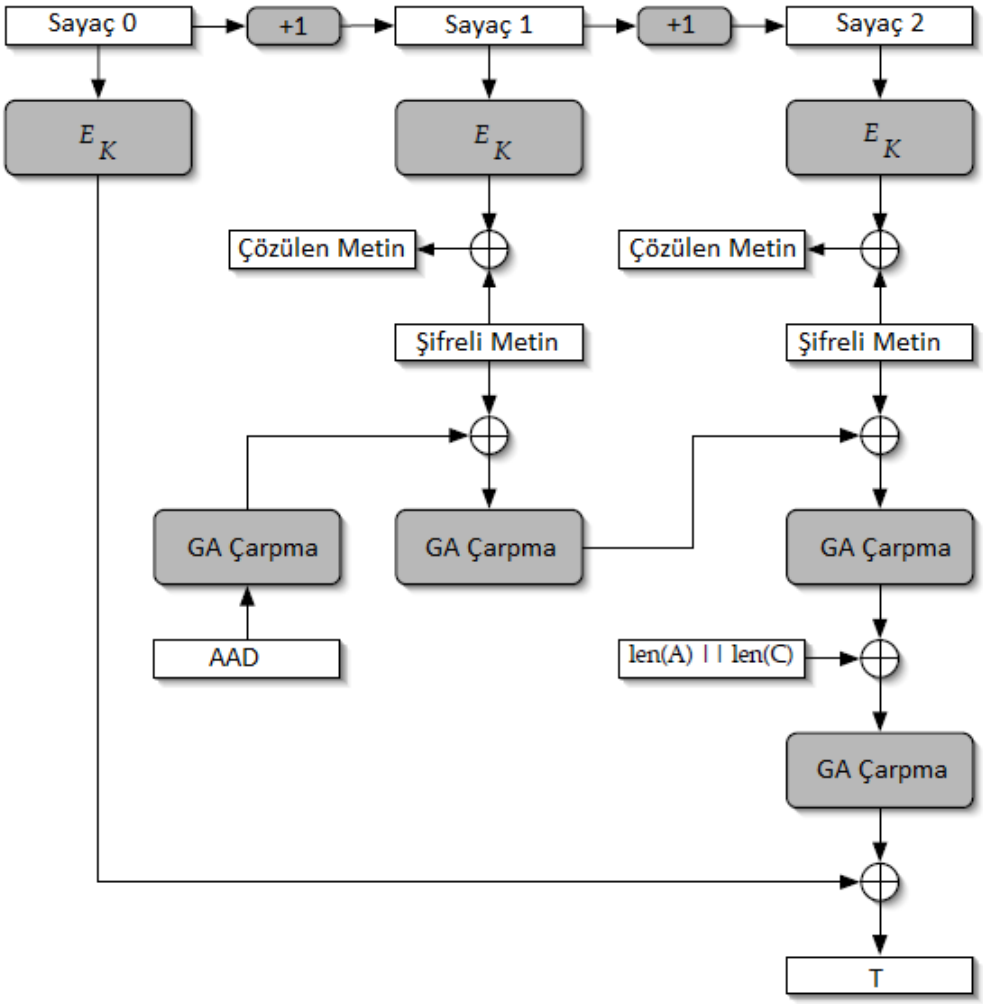
AES şifreleme algoritması, Amerika Birleşik Devletleri Ulusal Teknoloji ve Standartlar Enstitüsü (NIST) tarafından 2001 yılında yayınlanmıştır [22]. Joan Daemen ve Vincent Rijmen tarafından NIST'e sunulan *Rijndael şifreleme bloğu* temel alınarak AES şifreleme algoritması geliştirilmiştir [23]. Rijndael'de blok ve anahtar boyutları 32 sayısının katları şeklinde tanımlanabilirken, AES'te blokların boyutu 128 ile sınırlandırılmış, anahtar uzunlukları ise 128, 192 ve 256-bit olabilmektedir. İlk olarak Amerika Birleşik Devletleri'nde kullanılmaya başlanan AES, 1977'de önerilen Veri Şifreleme Standardı'nın (DES) yerini alarak günümüzde dünya genelinde kullanılmaya başlamıştır. AES şifreleme algoritması simetrik bir algoritmadır. Simetrik algoritmalarda şifreleme ve şifre çözme işlemi için tek bir anahtar kullanılır [24]. Bu çalışmada kullanılan AES algoritması 128-bit anahtar uzunluğuna sahiptir.

Şifreleme algoritmaları blok olarak adlandırılan belirli uzunluktaki verileri şifrelemek için kullanılırlar. Daha uzun blok dizilerini şifrelemek için, şifreleme algoritmalarına farklı çalışma kipleri eklenir. Çalışma kipi, blokların ne şekilde şifreleneceğini belirler. Kullanılacak verinin yapısına ve gerçekleme ortamına göre (yazılım, donanım, vs.) çalışma kipleri güvenliği ve performansı oldukça etkileyebilirler. Güvenlik ve performans kriterleri göz önüne alındığında bu tez çalışmasında [25]'te belirtlimiş olan Galois Sayaç Modu'nun (GCM), AES şifreleme algoritması ile birlikte kullanılmasına karar verilmiştir. Galois Sayaç Modu'nun çalışma prensibi bir sonraki bölümde anlatılmıştır.

4.7 GCM Çalışma Kipi

GCM çalışma kipi, ikili Galois alanı üzerinde evrensel özet fonksiyonu kullanan ve kimlik doğrulama sağlayan bir şifreleme modudur. Donanım üzerinde gerçeklenerek, düşük gecikmeye sahip yüksek hızlı çözümler sunmaktadır. GCM'nin iki farklı işlemi vardır. Bunlar kimlik doğrulamalı şifreleme ve kimlik doğrulamalı şifre çözmedir. Bu tez çalışmasında FPGA üzerinde kimlik doğrulamalı şifre çözme işlemini yapan devre gerçeklenmiştir. GCM'nin kimlik doğrulamalı şifreleme işlemi ise simulasyon ortamında kısmi bit katarını şifrelemek için kullanılmıştır.

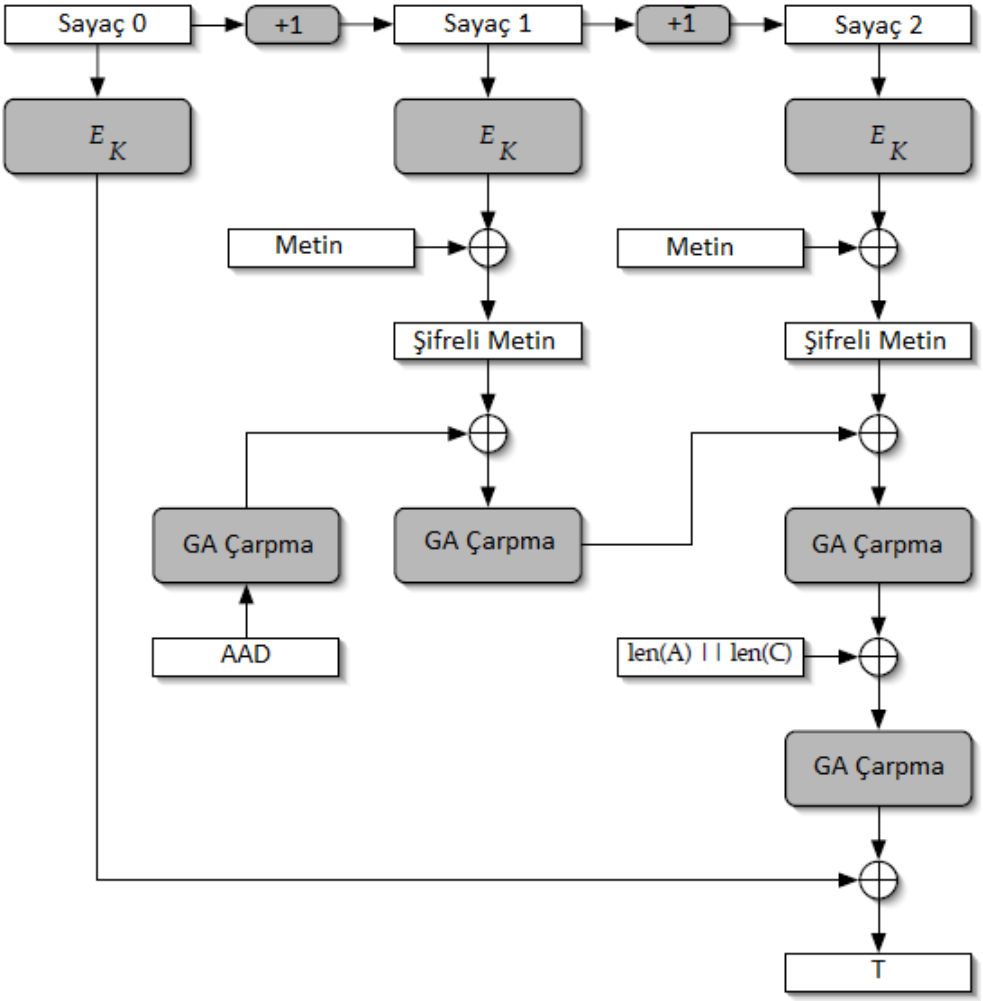
Kullanılan GCM kimlik doğrulamalı şifreleme bloğunun dört girişi vardır. Bu girişler, 128-bit uzunluğunda şifreleme anahtarı K, uzunuluğu 1 ve 2^{64} arasında değişen başlangıç vektörü IV, uzunluğu 0 ve $2^{39} - 256$ arasında değişen şifrelenecek metin P ve uzunluğu 0 ve 2^{64} ek doğrulama verisi AAD'dir. Çıkışta ise, uzunluğu P ile aynı olan şifrelenmiş metin C ve uzunluğu 0 ve 128 arasında değişen kimlik doğrulama etiketi T elde edilir. GCM kimlik doğrulamalı şifre çözme bloğunun girişleri ise K, IV, C, AAD, ve T'dir.



Şekil 4.9: GCM kimlik doğrulamalı şifre çözme bloğu akış diyagramı [25].

Şekil 5.1'de GCM kimlik doğrulamalı şifre çözme bloğu akış diyagramı verilmiştir. Burdaki gösterimde şifreli metin dizisinin boyutunun 256-bit olduğu varsayılmıştır aynı anda iki adet 128-bitlik verinin şifre çözüm işlemi gerçekleşitirilmektedir. E_K ile gösterilen bloklar, sayaçlardan okudukları değeri K ile şifrelerler. Sayaç 1'in altındaki ilk 128-bitlik şifreli metin ile Sayaç 1'in değerinin şifreleme sonucu toplanır. Galois alanında toplama işlemi EXOR işlemiyle aynı olduğundan devre gerçeklemesinde bu

iki değer EXOR kapılarında işleme sokulurlar. Bu işlemin sonucunda şifreli metin çözülmüş olur. Şekilde görülen *GA Çarpma* blokları Galois alanında tanımlanmış özel bir çarpma işlemidir. Bu çarpma işleminin kullanılmasının amacı donanım gerçeklemesinin kolay olmasıdır. Şifreli metin dizisindeki T değerini elde etmek için, şifreli metinler GA çarpma bloklarından geçirilirler. GCM çalışma kipindeki kimlik doğrulama verisi, Şekil 5.1'de görüldüğü üzere şifreli metinlerin GA çarpma bloklarından geçirilmesinden elde edilen veriler ile birlikte ilk sayaçtan ve son sayaçtan gelen veriler kullanılarak elde edilir.



Şekil 4.10: GCM kimlik doğrulamalı şifreleme bloğu akış diyagramı [25].

Şekil 5.2'de GCM kimlik doğrulamalı şifreleme bloğu akış diyagramı verilmiştir. Görüldüğü üzere şifreleme ve şifre çözme blokları birbirine çok benzerdir. Sadece E_K bloklarının sonuçlarının toplama sokulduğu yerde, şifreleme bloğunda metin toplama işlemine girip şifreli metin sonuç olarak çıkarken, şifre çözme bloğunda şifreli metin toplama işlemine girip sonuç olarak çözülen metin elde edilmektedir.

4.8 AES-GCM Şifreleme Algoirtmasının FPGA Üzerinde Gerçeklenmesi

Bu tez çalışmasında kullanılan şifreleme algoritmasının donanım tanımını yapan devrenin kodu Gövem'in [5]'te yapmış olduğu tez çalışmasından alınmıştır. Gövem ise donanım kodunu [26]'te belirtilen Open Cores projesi üzerinde gerekli değişiklikleri ve eklemeleri yaparak hazırlamıştır.

AES-GCM çalışma kipi 2048-bitlik bir metin üzerinde çalışmak üzere ayarlanmıştır. Şekil 5.1'de gösterilen sayaçlardan 17 tane kullanılmıştır. Aynı anda 16 tane 128-bitlik şifrelenmiş verinin çözüm işlemi ve kimlik doğrulaması yapılacak şekilde devre ayarlanmıştır.

BPI PROM üzerinde şifrelenmiş tasarımın yer aldığı 3. Bölüm'de belirtilmiştir. Şifrelenmiş tasarımın tek seferde okunarak FPGA üzerinde tutulması çok fazla kaynak kullanımına neden olacağı için, kısmi bit katarı dosyası 2048 kelimelik (128-bit) bloklara ayrılmış, ve bloklar ayrı ayrı şifrelenerek BPI PROM üzerinde yüklenmiştir. Kısmi yeniden yapılandırma işleminin başlama isteğiyle, FPGA üzerindeki tasarım BPI PROM üzerinden şifrelenmiş tasarımı 2048 kelimelik bloklar halinde okumaya başlar. Aynı anda 16 tane şifrelenmiş metin çözen AES-GCM şifre çözme bloğu, 2048 kelimelik bloğu on altılı bloklar halinde çözer ve kimlik doğrulamasını yapar. 2048 kelimelik blok kimlik doğrulama işleminden geçerse, bir sonraki 2048 kelimelik blok BPI PROM üzerinden okunarak şifre çözme işlemine devam edilir. Bu işlemler şifrelenmiş tasarımın tamamı okunana kadar veya okunan bloklardan birinin kimlik doğrulaması başarısız olana kadar devam eder. Eğer tüm bloklar kimlik doğrulamadan başarıyla geçerlerse kısmi yeniden yapılandırma işlemide başarıyla tamamlanmış olur.

5. SONUÇ VE ÖNERİLER

Bu tez çalışması Xilinx firması tarafından üretilmiş olan ML505 geliştirme kartı üzerinde gerçekleştirilmiştir. ML505 geliştirme kartı üzerindeki sistem tasarımı Xilinx ISE v13.2 ve Planahead v13.2 yazılımları kulllanılarak yapılmıştır. Tasarımın benzetimleri için Modelsim 6.5d yazılımı, sistem üzerinden toplanan verilerin değerlendirilmesi için MATLAB yazılımı kullanılmıştır.

Önerilen sistem mimarisinin çalıştığını göstermek amacıyla kısmi yeniden yapılandırılabilen bölgelere basit mantık devreleri yerleştirilmiştir. Sistem ilk kez çalıştırıldığında birinci kısmi yeniden yapılandırılabilen modül, statik tasarım ile birlikte BPI PROM üzerinden FPGA üzerine yüklenir. Bu yükleme tamamlandıktan sonra kısmi yeniden yapılandırılabilen bölgedeki tasarım, ML505 kartı üzerinde bulunan elektrik anahtarlarından aldığı giriş sinyalleri ile EXOR veya VE sayısal mantık işlemlerini gerçekleştirmektedir. Bu işlemlerin sonuçları ML505 üzerinde yer alan LED'lerde gözlemlenebilmektedir.

Kısmi yeniden yapılandırma işleminin başlamasını belirten sinyalin kart üzerinde yer alan buton ile tetiklenmesi sağlanmıştır. Bu butona basıldığında, sistem kısmi yeniden yapılandırma sürecini başlatır. Bu süreç şu şekilde işlemektedir. İlk olarak halka osilatör çiftleri çalıştırlır ve bağlı oldukları sayıcılardan biri 32768 değerine ulaşana kadar çalıştırılıp, sayıcılar arasındaki fark kaydedilir. Bu işlem 128 adet halka osilatör çifti için sırayla gerçekleştirilir. İşlem sonucunda elde edilen sayaç farkları kullanılarak sistem yeni bir etiket değeri üretir. Bu yeni üretilen etiket değeri ve Çizelge 4.1'de gösterilen yardımcı veri kullanılarak sistem yeni bir PUF çıkışı değeri üretir. Üretmiş olduğu bu PUF çıkışı değeri AES-GCM şifre çözme bloğunda şifre çözme anahtarı olarak kullanılır. Yapılan ölçümler ve elde edilen sonuçlar doğrultusunda bazı halka osilatör çiftlerinin etiketleme yönteminde dahi kararlı bir davranış sergilemediği gözlemlenmiştir. Kararsız davranış gösteren 4, 5, 8, 25, 62, 64 ve 111 nolu halka osilatör çiftlerinin PUF çıkışları '0' değerine sabitlenmiştir.

Sistem kısmi yapılandırma sürecinde PUF çıkışı ürettiken sonra BPI PROM üzerindeki şifrelenmiş tasarımı okumaya başlarlar. BPI PROM üzerindeki şifreli tasarımın içerisinde ikinci kısmi yeniden yapılandırılabilen modülün bit katarı yer almaktadır. Bu modül EXNOR veya VEYA sayısal mantık işlemlerini gerçekleştirmektedir. Çizelge 4.1'de gösterilen PUF çıkışı değeri ile ikinci kısmi yeniden yapılandırılabilen modülün bit katarı şifrelenmiştir. Bu şifreleme işleminde kısmi bit katarı 2048 kelimelik parçalara ayrılmış ve ayrı ayrı şifrelenerek BPI PROm üzerine yazılmıştır. Sistem kısmi yeniden yapılandırma sürecinde, PUF çıkışını ürettikten sonra BPI PROM üzerinden şifrelenmiş tasarımı 2048 kelimelik bloklar halinde okumaya başlar. Bu bloklar AES-GCM şifre çözme bloğunda devrenin üretmiş olduğu yeni PUF çıkışı değeri ile çözülüp, kimlik doğrulama işleminden geçerler. Eğer devrenin üretmiş olduğu yeni PUF çıkışı değeri, Çizelge 4.1'de verilen PUF çıkışı değeri ile aynı ise kimlik doğrulama işlemi başarıyla sonuçlanır. ML505 kartı üzerindeki butona basarak kısmi yeniden yapılandırma işlemi başlatıldığında, sistemin kısmi yeniden yapılandırma işlemini başarıyla tamamladığı gözlemlenmiştir.

Çizelge 5.1: FPGA üzerindeki kaynakların kullanımı.

	LUT	FF	36K BRAM	ICAP
Kullanılan	6984	3832	6	1
Mevcut	69120	69120	148	2

Çizelge 5.1'de tasarlanan sistemin kaynak kullanımına yer verilmiştir. [27]'da yapılan çalışmada AES-GCM ve AES-SHA algoritmalarının kaynak kullanımları karşılaştırılmıştır. Virtex-5 üzerindeki AES-GCM devresinin kaynak kullanımı 3040 tane LUT, 2166 tane FF, ve 13 tane BRAM olarak verilmiştir. Aynı çalışmada ek olarak ICAP üzerinden FPGA üzerinde kısmi yeniden yapılandırma işleminin, işlemci ve donanım olarak tasarlanmış kontrolcü tarafından gerçeklenmesi kıyaslanmıştır. Elde ettikleri sonuçlarda donanım olarak tasarlanan kontrolcü çözümlerinin, işlemci içeren çözümlerden daha hızlı çalıştığı belirtilmiştir. [27]'da belirtilen kaynak kullanım değerleri, bu tez çalışmasının kaynak kullanımlarından daha azdır. Aradaki bu fark, [27]'de tasarlanan devrede şifreleme anahtar üretimi için kullanılan ek bir devre olmamasından kaynaklanmaktadır. Bu tez çalışmasında şifreleme anahtarı üretimi için

halka osilatör tabanlı PUF devresi ve etiketleme yöntemini gerçekleyen devrelerde yer almaktadır.

5.1 Çalışmanın Uygulama Alanı

Bu tez çalışmasında gerçeklenen sistem ile klonlamaya karşı elektronik bir sistem geliştirilmeye çalışılmıştır. Çizelge 5.1'deki kaynak kullanımı değerlendirildiğinde, kullanılan Virtex-5 FPGA'sının kaynaklarının yaklaşık olarak %10'u kullanılmıştır. Geri kalan kaynaklar, robotlar vb. sistemlerde kullanılacak olan kontrol algoritmaları için yeterli olacaktır. Donanım ve IP güvenliğine ihtiyaç duyulan uygulamalarda, kısmi yeniden yapılandırma ve halka osilatör tabnlı PUF devresi ile anahtar üreten sistem, ihtiyaçları karşılayabilecek durumdadır.

KAYNAKLAR

[1] **OECD**, http://www.oecd.org/industry/ind/38707619.pdf, alındığı tarih: 14.04.2014.

[2] **BASCAP**, http://www.iccwbo.org, alındığı tarih: 14.04.2014.

[3] **Adi, W.** (2009). Mechatronic Security and Robot Authentication, *Bio-inspired Learning and Intelligent Systems for Security, 2009. BLISS '09. Symposium on*, s.77–82.

[4] **Adi, W. ve Soudan, B.** (2007). Bio-Inspired Electronic-Mutation with genetic properties for Secured Identification, *2007 ECSIS Symposium on Bio-inspired, Learning, and Intelligent Systems for Security (BLISS 2007)*.

[5] **Gövem, B.** (2013). Klonlanamaz Fonksiyonlar ve Yardımcı Bilgiler Kullanılarak Patent Hakları Korunması, Özgün Algoritma ve Donanımın Güvenlğinin Sağlanması, *İstanbul Teknik Üniversitesi Fen Bilimleri Enstitüsü(İTÜ)*.

[6] **Rodríguez-Henríquez, F.**, **Pérez, A.D.**, **Saqib, N.A. ve Koç, C.K.** *Cryptographic Algorithms on Reconfigurable Hardware*.

[7] **Url-1**, http://www.xilinx.com/fpga/index.htm, alındığı tarih: 16.04.2014.

[8] **ML505UG**, http://www.xilinx.com/support/documentation/boards_and_kits/ug347.pdf, alındığı tarih: 16.04.2014.

[9] **Virtex5UG**, http://www.xilinx.com/support/documentation/user_guides/ug190.pdf, alındığı tarih: 16.04.2014.

[10] **Verilog**, http://en.wikipedia.org/wiki/Verilog, alındığı tarih: 16.04.2014.

[11] **VHDL**, http://en.wikipedia.org/wiki/VHDL, alındığı tarih: 16.04.2014.

[12] **Estrin, G.** (1960). Organization of Computer Systems: The Fixed Plus Variable Structure Computer, *Papers Presented at the May 3-5, 1960, Western Joint IRE-AIEE-ACM Computer Conference*, IRE-AIEE-ACM '60 (Western), ACM, New York, NY, USA, s.33–40.

[13] Partial Reconfiguration User Guide, http://www.xilinx.com/support/documentation/sw_manuals/xilinx12_1/ug702.pdf, alındığı tarih: 20.04.2014.

[14] Virtex-5 Configuration User Guide, http://www.xilinx.com/support/documentation/user_guides/ug191.pdf, alındığı tarih: 20.04.2014.

[15] ISE In-Depth Tutorial, http://www.xilinx.com/support/documentation/sw_manuals/xilinx12_4/ise_tutorial_ug695.pdf, alındığı tarih: 20.04.2014.

[16] Planahead User Guide, http://www.xilinx.com/support/documentation/sw_manuals/xilinx13_4/PlanAhead_UserGuide.pdf, alındığı tarih: 20.04.2014.

[17] **Yeniçeri, R.**, **Abtioğlu, E.**, **Gövem, B. ve Yalçın, M.E.** (2014). A 16x16 Cellular Logic Network With Partial Reconfiguration Feature, *14th International Workshop on Cellular Nanoscale Networks and their Applications(Accepted)*, Notre Dame, IN, USA.

[18] **Kaya, G.** (2012). Reliability Enhancement of Ring Oscillator Based Physically Unclonable Functions, *İstanbul Teknik Üniversitesi Fen Bilimleri Enstitüsü(İTÜ)*.

[19] **Verbauwhede, I.**, **Tuyls, P. ve Maes, R.** (2008). Intrinsic PUFs from flip-flops on reconfigurable devices, *3rd Benelux Workshop on Information and System Security (WISSec 2008)*.

[20] **Tuyls, P.**, **Skoric, B.**, **Stallinga, S.**, **Akkermans, A. ve Ophey, W.** (2005). Information theoretical security analysis of physical unclonable functions, *Proceedings of Conference on Financial Cryptography and Data Security 2005, volume 3570 of Lecture Notes in Computer Science*.

[21] **Suh, G.E. ve Devadas, S.** (2007). Physical Unclonable Functions for Device Authentication and Secret Key Generation, *2007 44th ACM/IEEE Design Automation Conference*, 9–14.

[22] **FIPS**, Announcing the Advanced Enryption Standart, http://csrc.nist.gov/publications/fips/fips197/fips-197.pdf, alındığı tarih: 13.04.2014.

[23] **Daemen, J. ve Rijmen, V.**, AES Proposal: Rijndael, http://csrc.nist.gov/archive/aes/rijndael/Rijndael-ammended.pdf#page=1, alındığı tarih: 13.04.2014.

[24] Advaced Encyption Standard, http://en.wikipedia.org/wiki/Advanced_Encryption_Standard#cite_note-fips-197-4, alındığı tarih: 13.04.2014.

[25] **Viega, J. ve McGrew, D.**, The Galois/Counter Mode of Operation (GCM), http://en.wikipedia.org/wiki/Advanced_Encryption_Standard#cite_note-fips-197-4, alındığı tarih: 13.04.2014.

[26] **Ahmad, T.**, OpenCores Projesi:Galois Counter Mode Advanced Encryption Standard GCM-AES, http://opencores.org/project,gcm-aes.

[27] **Hori, Y.**, **Satoh, A.**, **Sakane, H. ve Toda, K.** (2008). Bitstream encryption and authentication with AES-GCM in dynamically reconfigurable systems, *Field Programmable Logic and Applications, 2008. FPL 2008. International Conference on*, s.23–28.

ÖZGEÇMİŞ

Ad Soyad: Emrah ABTİOĞLU

Doğum Yeri ve Tarihi: KIRCALI, 1988

Adres:

E-Posta: eabtioglu@itu.edu.tr

Lisans: Anadolu Üniversitesi Elektrik-Elektronik Mühendisliği, Haziran 2012

Mesleki Deneyim ve Ödüller:

Yayın ve Patent Listesi:

TEZDEN TÜRETİLEN YAYINLAR/SUNUMLAR

▪ Yeniçeri R., **Abtioğlu E.**, Gövem B. and Yalçın M. E., 2014: A 16x16 Cellular Logic Network With Partial Reconfiguration Feature(Sunuma kabul edilmiştir), *14th International Workshop on Cellular Nanoscale Networks and their Applications*, July 29-31, 2014 Notre Dame, IN, USA.