İSTANBUL TEKNİK ÜNİVERSİTESİ

ELEKTRİK-ELEKTRONİK FAKÜLTESİ

GÜVENLİ RFID SİSTEMLERİ İÇİN BİR KİMLİK DOĞRULAMA PROTOKOLÜNÜN GERÇEKLENMESİ

BİTİRME ÖDEVİ

SEMİH ALPARSLAN

040070321

Bölümü: Elektronik ve Haberleşme Mühendisliği Bölümü

Programı: Elektronik Mühendisliği

Danışmanı: Doç. Dr. Sıddıka Berna ÖRS YALÇIN

MAYIS 2012

ÖNSÖZ

Öncelikle bitirme projem süresince benden bilgilerini, önerilerini ve desteğini hiçbir zaman esirgemeyen saygıdeğer hocam Doç. Dr. Sıddıka Berna Örs Yalçın'a sonsuz teşekkürlerimi sunmayı bir borç bilirim.

Başta İstanbul Teknik Üniversitesi Gömülü Sistemler Tasarımı Laboratuarı sorumlusu Arş. Gör. Ramazan Yeniçeri olmak üzere laboratuardaki tüm arkadaşlarıma yardımlarından ötürü teşekkür ederim.

Son olarak, bütün hayatımda olduğu gibi bitirme tezimin gerçekleşmesi aşamasında da sürekli yanımda olan ve destek veren aileme sonsuz minnettarlığımı sunarım.

Semih Alparslan

Mayıs 2012

İÇİNDEKİLER

KISALTMALAR v
ŞEKİL LİSTESİ vi
ÖZET viii
SUMMARY ix

1. GİRİŞ 1

2. RADYO FREKANSI İLE TANIMLAMA SİSTEMLERİ 3
2.1. Etiket 3
2.2. Okuyucu 4

3. KULLANILAN TASARIM ARAÇLARI 6
3.1. Sahada Programlanabilir Kapı Dizinleri 6
3.2. Spartan3E Başlangıç Kiti 8
3.3. Microblaze Mikroişlemcisi 9
3.4. Verilog Donanım Tanımlama Dili 10
3.5. ISE Ortamı 10
3.6. EDK Ortamı 10
3.7. SDK Ortamı 12

4. GERÇEKLENECEK PROTOKOL 14
4.1. ISO/IEC 18000 Standardı 14
4.2. Doğrulama Mekanizması 15
4.3. TEA Şifreleme Algoritması 17
4.4. Rastgele Sayı Üreteci 22
4.5. Mikroişlemci 22

5. KİMLİK DOĞRULAMA PROTOKOLÜNÜN GERÇEKLENMESİ 24
5.1. Donanım Tasarımı Aşamaları 24
5.1.1. Toplama Bloğu 25
5.1.2. Çıkarma Bloğu 25
5.1.3. Mantıksal Kaydırma Bloğu 26
5.1.4. Özel veya Bloğu 27
5.1.5. Şifreleme Bloğu 27
5.1.5.1. Saat İşaretli Şifreleme Bloğu 28
5.1.5.2. Saat İşaretsiz Şifreleme Bloğu 30
5.1.6. Şifre Çözme Bloğu 31
5.1.7. Rastgele Sayı Üreteci Bloğu 33
5.2. Yazılım Tasarımı 33
5.2.1. Etiket Kısmı 34
5.2.2. Okuyucu Kısmı 36

6. SONUÇLAR **39**
KAYNAKLAR **40**
EKLER **41**
ÖZGEÇMİŞ **44**

KISALTMALAR

RFID :Radio Frequency Identification
FPGA :Field Programmable Gate Array
LUT :Look-up Table
RISC :Reduced Instruction Set Computing
FPU :Floating Point Unit
MMU :Memory Management Unit
ISE :Integrated Software Environment
EDK :Embedded Development Kit
BSB :Base System Builder
XPS :Xilinx Platform Studio
SDK :Software Development Kit
IP :Internet Protocol
ASK :Amplitude Shift Keying
SOF :Start of Frame
EOF :End of Frame
CRC :Cyclic Redundancy Check
UID :Unique Identifier
AES :Advanced Encryption Standard
TEA :Tiny Encryption Algorithm
XOR :Exclusive Or
LFSR :Linear Feedback Shift Register
PLB :Processor Local Bus
UART :Universal Asynchronous Receiver Transmitter

ŞEKİL LİSTESİ

Şekil 2.1 : Etiket ve okuyucu arasındaki hatlar [2]. 5
Şekil 3.1 : Mantık hücresi yapısı [6]. 6
Şekil 3.2 : FPGA'nın iç yapısı [6]. 7
Şekil 3.3 : Spartan3E başlangıç kiti [7]. 8
Şekil 3.4 : Microblaze mimarisi [8]. 9
Şekil 3.5 : EDK sistem geliştirme araçları [9]. 11
Şekil 3.6 : Sistem tasarımı akışı [11]. 13
Şekil 4.1 : Kimlik sorma sistemi [2]. 16
Şekil 4.2 : Doğrulama protokolü [2]. 16
Şekil 4.3 : İstek çerçevesi [2]. 17
Şekil 4.4 : Cevap çerçevesi [2]. 17
Şekil 4.5 : TEA şifreleme rutini [13]. 19
Şekil 4.6 : TEA i. döngüsü [13]. 20
Şekil 4.7 : TEA şifre çözme rutini [13]. 21
Şekil 4.8 : Mikroişlemcili okuyucu yapısı. 23
Şekil 4.9 : Mikroişlemcili etiket yapısı. 23
Şekil 5.1 : Toplama bloğu. 25
Şekil 5.2 : Çıkarma bloğu. 26
Şekil 5.3 : Mantıksal kaydırma bloğu. 26
Şekil 5.4 : Özel veya bloğu. 27
Şekil 5.5 : Saat işaretli şifreleme bloğu. 29
Şekil 5.6 : Şifreleme bloğunun bir döngüsü. 30
Şekil 5.7 : Saat işaretsiz şifreleme bloğu. 31
Şekil 5.8 : Şifre çözme bloğunun bir döngüsü. 32
Şekil 5.9 : Şifre çözme bloğu. 32
Şekil 5.10 : Rastgele sayı üreteci bloğu. 33
Şekil 5.11 : Etiketin FPGA üzerinde gerçeklenmesine ilişkin ekran çıktısı. 35
Şekil 5.12 : Okuyucuya gönderilen çerçeveye ilişkin ekran çıktısı. 36
Şekil 5.13 : Rastgele sayı üretecinin Microblaze ile FPGA üzerinde gerçeklenmesine ilişkin ekran çıktısı. 37

Şekil 5.14 : Okuyucuya ait şifre çözme bloğunun FPGA üzerinde gerçeklenmesine ilişkin ekran çıktısı. 38

GÜVENLİ RFID SİSTEMLERİ İÇİN BİR KİMLİK DOĞRULAMA PROTOKOLÜNÜN GERÇEKLENMESİ

ÖZET

Günümüzde Radyo Frekansı ile Tanımlama (Radio Frequency Identification, RFID) sistemlerinin kullanımı hızla yaygınlaşmaktadır. Bu sistemlerin kullanımının yaygınlaşması güvenlik açıklarını da beraberinde getirmektedir. Birçok uygulama sahasına girmiş bu sistemlerin güvenlik açıklıklarına sahip olması ve bu güvenlik açıklıklarının kötü amaçlı insanlar tarafından kullanılması ciddi bir problem teşkil etmektedir. Bu nedenlerden dolayı RFID sistemlerin güvenli şartlar altında kullanılması zorunluluk haline gelmiştir. Bu güvenli şartları sağlamak için RFID uygulamalarında kriptoloji algoritmalarının kullanılması en akılcı çözümdür.

Bu projede güvenli bir RFID sistem oluşturmak için bir kimlik doğrulama protokolü gerçeklenmiştir. Bu protokolün nasıl gerçeklendiği daha iyi kavrayabilmek adına ön bilgiler olarak öncelikle RFID sistemlerinden ve onun ana öğelerinden bahsedilmiştir. Sonrasında, gerçekleme aşamasında kullanılan tasarım araçları tanıtılmıştır. Son olarak, güvenli RFID sistemi tasarlamak için kullanılan protokol detaylı bir şekilde anlatılmıştır. Protokol gerçeklenirken gömülü sistemlere olan uygunluğu sebebiyle şifreleme algoritması olarak Küçük Şifreleme Algoritması (Tiny Encryption Algorithm, TEA) kullanılmıştır.

Protokolü gerçekleme aşamasında, öncelikle, sistemde kullanılacak olan donanım parçaları olan şifreleme, şifre çözme ve rastgele sayı üreteci blokları Verilog donanım tanımlama dilinde tasarlanmıştır. Donanımların hepsinin tamamen çalışır durumda olduğu test edildikten sonra tüm sistemi ve bu donanım bloklarını kontrol etmek amacıyla Microblaze mikroişlemcisi üzerinde C dili ile yazılım tasarımı yapılmıştır. Son olarak donanım ve yazılım tasarımı yapılan güvenli RFID sistem Sahada Programlanabilir Kapı Dizinleri (Field Programmable Gate Array, FPGA) üzerinde gerçeklenmiştir.

IMPLEMENTATION OF AN AUTHENTICATION PROTOCOL FOR SECURE RFID SYSTEMS

SUMMARY

Today, with Radio Frequency Identification systems usage is expanding rapidly. Widespread use of this systems brings with it vulnerabilities. Having the security vulnerabilities of this systems which have entered many applications, and that security vulnerabilities have to be used by malicious people constitutes a serious problem. Because of this reasons, the using of RFID systems under secure conditions has became necessity. The most rational solution to ensure secure conditions for RFID application is the use of cryptology algorithms.

In this project, an authentication protocol to create a secure RFID system is implemented. Firstly, in order to better comprehend how this protocol was implemented, RFID systems and its main components are described as preliminary information. Then, the design tools used in the implementation section is introduced. Lastly, the protocol used for secure RFID system design is described in detail. In the implementation section, due to the compliance with embedded systems TEA is used.

In the implementation section, primarily, the hardware components used in the system with encryption, decryption and random number generator blocks are designed with Verilog hardware description language. After testing all of the hardware is fully operational, in order to control the entire system and blocks the Microblaze microprocessor software design has been made on the C language. Finally, The secure RFID system with hardware and software design is implemented on a FPGA.

1. GİRİŞ

Son yıllarda otomatik tanımlama işlemleri satın alma, dağıtım lojistiği, sanayi, üretim şirketleri ve malzeme akış sistemleri gibi birçok sektörde çok popüler hale gelmiştir. Ayrıca insanlar, hayvanlar, mallar ve nakledilen ürünler hakkında bilgi tutulması da otomatik tanımlama işlemleri ile daha işlevsel hale gelmiştir [1]. Bu nesnelerin otomatik tanımlanması işlemleri için Radyo Frekansı ile Tanımlama Sistemleri kullanmak oldukça etkili bir yaklaşımdır. Her gün birçok nesneye onları tanımlamak için RFID sistemler eklenmektedir.

RFID sistemleri arasındaki haberleşmeyi sağlamak için ISO/IEC 18000 standardı bulunmaktadır. Ancak bu Standard radyo frekansı ile doğrulamanın nasıl yapılacağı hakkında bilgi vermemektedir. RFID sistemlerin doğrulama işlemi hakkında bilgi vermemesi güvenlik açısından sorgulanmalarına yol açmaktadır. RFID sitemlerine doğrulama uygulamasının eklenmesinin en iyi yolu şifreli doğrulamanın kullanılmasıdır [2].

Bu bitirme tezinde güvenli bir RFID sistemin şifreleme algoritması da kullanılarak seçilen protokol doğrultusunda nasıl tasarlanacağı gösterilmektedir. Doğrulama mekanizması ile ilgili olarak temel hedef, gönderilecek ya da alınacak olan mesajın karşılıklı olarak doğrulama işlemine tabii tutulmasıdır. Doğrulama işlemi basamakları; okuyucu biriminin etikete rastgele mesaj göndermesi, daha sonra etiketten bu mesajı şifreli olarak alması ve aldığı mesajın şifresini çözmesi şeklindedir. Eğer alınan şifresi çözülen mesaj ile gönderilmiş olan rastgele mesaj aynı ise etiket ve okuyucunun aynı şifreleme anahtarına sahip olduğu anlaşılmaktadır. Yani bu mekanizma aynı anahtara sahip olmayan etiket ve okuyucunun haberleşememesini sağlamaktadır.

Protokolde şifreleme işlemi Küçük Şifreleme Algoritması (Tiny Encryption Algorithm, TEA) ile sağlanacaktır. Doğrulama mekanizmasının okuyucu ve etiket birimleri Sahada Programlanabilir Kapı Dizinleri (Field Programmable Gate Array, FPGA) üzerinde gerçekleneceklerdir. Sistem gerçeklenirken etiket birimi genellikle taşınabilir oldukları için az güç tüketmesi ve az alan kaplaması amaçlanmıştır.

Okuyucu birimi için ise birden çok doğrulamayı aynı anda yapabilmesi için hızlı sistem bir sistem tasarımı amaçlanmıştır. Bu doğrultuda şifreleme, şifre çözme ve rastgele sayı üreteci algoritmaları FPGA üzerinde donanım olarak gerçeklenmişlerdir. Bu donanımları ve tüm sistemi kontrol etmek için FPGA'nın içerisinde bulunan Microblaze mikroişlemcisi kullanılmıştır.

Tezin ikinci bölümünde, radyo frekansı ile tanımlama konusundan ve onun ana öğeleri olan etiket ve okuyucu yapılarından kısaca bahsedilmiştir. Donanım ve yazılım tasarımı sırasında kullanılan araçlar ise tezin üçüncü bölümünde anlatılmıştır. Tezin dördüncü bölümünde gerçeklenecek olan kimlik doğrulama protokolünden bahsedildikten sonra, beşinci bölümde bu protokolü gerçekleme aşamaları tüm detaylarıyla anlatılmıştır.

2. RADYO FREKANSI İLE TANIMLAMA SİSTEMLERİ

Radyo frekansı ile tanımlama sistemleri, durağan ya da hareket halindeki nesneleri tekil ve otomatik olarak radyo frekansı ile tanımlamak için kullanılır. Günümüzde tanımlama işlemi için birçok yöntem mevcut olsa da, son zamanlarda ilgiler radyo frekansı ile tanımlama üzerine yoğunlaşmış durumdadır. Geçtiğimiz dönemlerden beri klasik olarak kullanılan barkot sistemleri kolay kullanıma sahip olmalarına karşın, saklayabildikleri veri miktarlarının az olması ve barkot üzerindeki etiketin değerin değiştirilmesinin imkansız olması barkot sistemini dezavantajlı kılmaktadır. Bu esnek olmayan kullanımın çözümü, içerisinde akıllı kartlar barındıran ve tanımlama için kablosuz cihazlar aracılığıyla radyo frekansını kullanan RFID sistemler ortaya çıkmıştır [2].

RFID sistemler ilk olarak 1940'lı yılların başlarında İngiltere'de dost ve düşman uçakların tanımlanmasında kullanılmıştır. Bunu 1970'li yıllarda nükleer malzeme izleme uygulamaları takip etmiş, ticari uygulamaları ise 1990'lı yıllarda başlamıştır. RFID sistemlerin uygulama alanlarına örnek olarak: ürün dağıtım zinciri uygulamaları, hasta tanımlama, kütüphane, müze, sanat galerisinde ürün tanımlama, taşımacılıkta değerli ürün izlenmesi gibi örnekler verilebilir [3].

RFID sistemleri temel olarak, tanımlanmak istenen nesnenin üzerine yerleştirilen "etiket" ve içerisinde bulunan anten vasıtasıyla bu etiket ile haberleşen "okuyucu" olmak üzere temel olarak iki ana öğeden meydana gelmektedir.

2.1. Etiket

RFID etiketi, radyo frekansını kullanarak okuyucudan gelen sinyalleri alan, sorgulayan daha sonra da cevaplayan ve tanımlamak istenen nesnelerin bilgilerini taşımak üzere nesnelere yerleştirilen sistem bileşenleridir. Taşınabilir oldukları için sınırlı hafıza kapasitelerine sahiptirler. Etiketler fonksiyonları bakımından aktif, yarı pasif ve pasif olmak üzere üçe ayrılır [3].

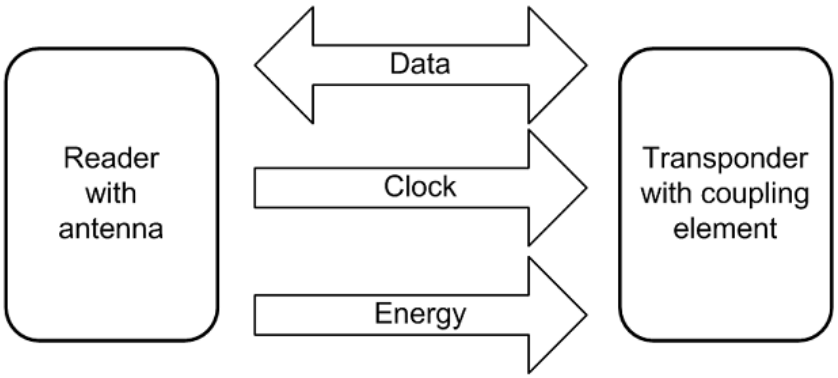
Aktif etiketler devrelerinin çalışmasını ve haberleşme için sinyal üretimlerini kendi güç kaynakları içlerinde barındırdıkları güç kaynağından sağlarlar. Kendi içinde barındırdıkları piller yani güç kaynakları sayesinde daha uzak haberleşme mesafeleri ve daha iyi çalışma performanslarına sahiptirler.

Yarı pasif etiketler de kendi güç kaynaklarını içerirler ancak içerdikleri bu güç kaynağı sadece kendi devresine güç sağlamaktadır. Haberleşme okuyucudan gelen sinyallerdeki güç ile sağlanmaktadır.

Pasif etiketler ise, üzerlerinde güç kaynağı barındırmazlar. Devrelerini ve haberleşmesini okuyucudan elektromanyetik dalga aracılığıyla aldığı güç ile beslemektedirler. Güç kaynakları içermedikleri için daha kısa mesafeli haberleşmelere olanak sağlarlar. Tercih edilme sebepleri, ucuz ve basit yapılı olmalarıdır. Bundan ötürü pasif etiketlerin en fazla yer aldığı uygulama alanları güç kaynağının uygulanamadığı, pil ömrünün daha öncelikli olduğu ve işlem kapasitesinin ikinci planda olduğu alanlardır.

2.2. Okuyucu

Bir RFID sistemde okuyucunun görevi, antenini kullanarak etiketleri uyarmak, etiketlerin içerdiği veriyi okumak ve bir ağ aracılığıyla bu veriyi bir sunucu bilgisayara göndermektir [4]. Okuyucular kullanım alanlarına göre mobil ya da sabitlenmiş olabilirler. RFID sistemlerde okuyucular aynı zamanda veritabanlarına bağlı olarak çalışırlar. Okuyucu, etiketi sorgulamak amacıyla etikete bir işaret gönderir ve bu işaret neticesinde uyarılan etiket kendi verisini okuyucuya geri gönderir. Okuyucu aldığı bu cevabı veritabanına göndererek veritabanında etiketin kayıtlı olup olmadığı sorgulanabilir. Sistemin iki ana öğesi olan okuyucu ve etiket arasındaki 3 hattan oluşan haberleşme Şekil 2.1'de görülmektedir. Bu haberleşme belirli frekanslarda gerçekleştirilmektedir. Bu hatlar; iki yönlü olmak üzere veri, sadece okuyucudan etikete doğru olan saat bilgisi ve aktif olmayan etiketler için yine sadece okuyucudan etikete doğru olan ve etiketi beslemek için gönderilen enerji olarak tanımlanır [2].



Şekil 2.1 : Etiket ve okuyucu arasındaki hatlar [2].

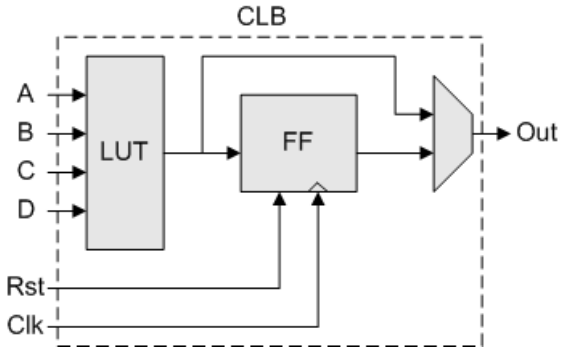
Etiket ve okuyucunun hangi formatlarda haberleşeceği, haberleşirken hangi modülasyonu kullanacakları, girişim engelleme metotları ve protokol parametreleri ISO/IEC 18000-3 standardında tarif edilmiştir. Bu standarda göre, RFID okuyucu ve etiket 13.56 Mhz frekansında haberleşmektedir [5]. Okuyucuları etiketlerden ayıran en önemli özellik, genellikle taşınamaz yapıda oldukları için içlerinde güç kaynağı barındırmalarıdır. Bir diğer fark ise aynı anda birden çok etiketle haberleşme yapabilme kapasiteleridir.

3. KULLANILAN TASARIM ARAÇLARI

3.1. Sahada Programlanabilir Kapı Dizinleri

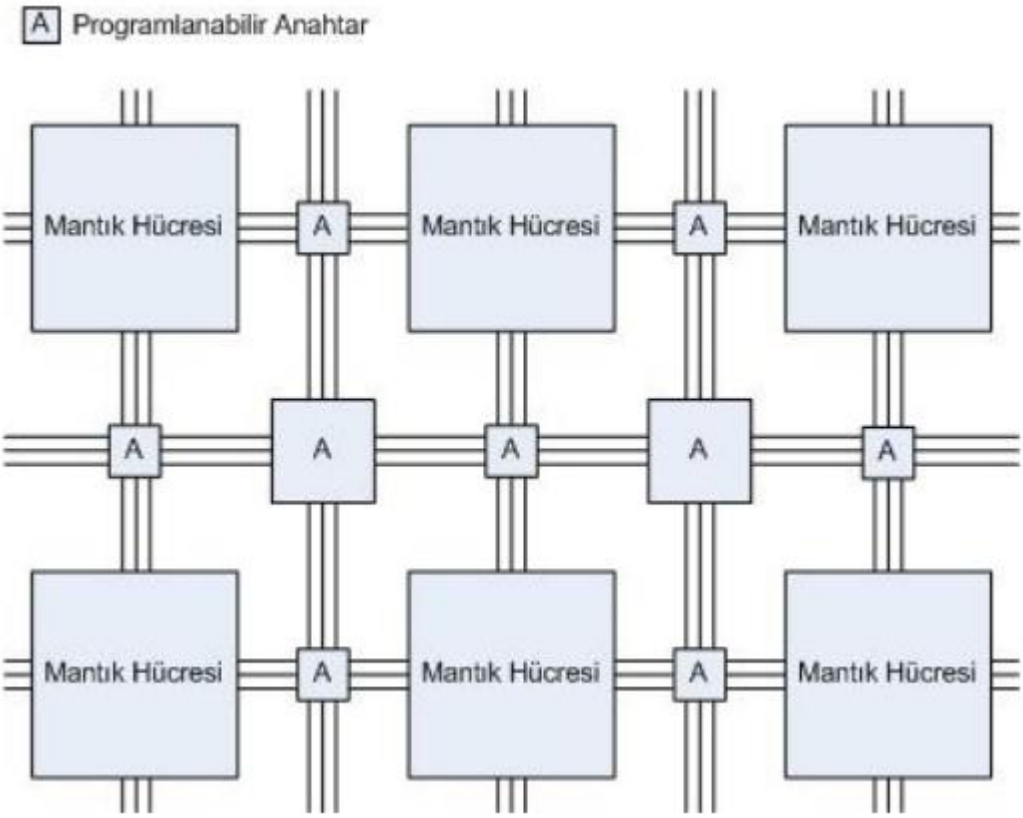
Sahada Programlanabilir Kapı Dizinleri herhangi bir sayısal fonksiyonu gerçekleştirebilmek için kullanıcı tarafından programlanabilen tümleşik devrelerdir [15]. FPGA yönetilebilir anahtarların ve programlanabilir mantık hücrelerinin iki boyutlu olarak dizilmesiyle oluşturulur. Mantık hücreleri basit bir fonksiyonu gerçeklemek üzere yapılandırılabildiği gibi programlanabilir anahtarlar ile mantık hücreleri arasında bağlantılar kurulabilir. Bu şekilde mantık hücreleri ve anahtarların programlanmasıyla sayısal donanımlar gerçeklenir. Donanım tanımlama dilleri kullanılarak devrenin tasarımı yapıldıktan ve sentezlenmesinin ardından istenilen lojik hücre ve anahtar yapılandırılmasının yer aldığı veri dizisi kablo yardımıyla FPGA'ya gönderilerek devre gerçeklenmiş olur [6].

Mantık hücreleri Şekil 3.1'de görüldüğü gibi programlanabilir kombinezonsal devre ve bir adet D tipi flip-flop içerir.



Şekil 3.1 : Mantık hücresi yapısı [6].

Programlanabilir ara bağlantılardan ve iç yapısı Şekil 3.1'de gösterilen mantık hücrelerinden oluşan FPGA'nın genel yapısı Şekil 3.2'de gösterilmektedir.



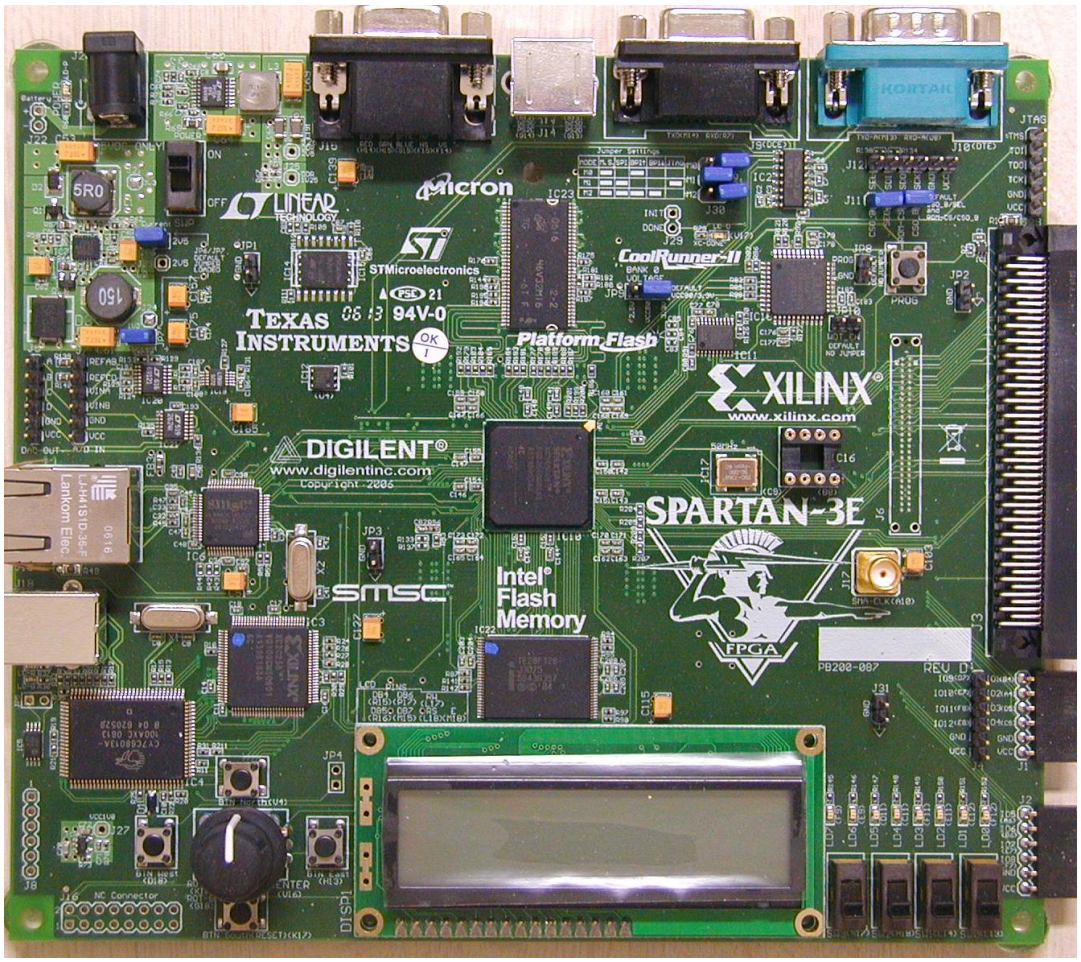
Şekil 3.2 : FPGA'nın iç yapısı [6].

Mantık hücrelerindeki LUT (Look-up Table) yapılandırılabilen kombinezonsal devreleri gerçeklemek için kullanılmaktadır. LUT'lar aslında bir mantık işlemini yerine getiren küçük belleklerdir. N girişli bir LUT 2^N boyutlu bellek elemanına karşılık düşmektedir. Binlerce LUT elemanı yan yana getirilerek daha kompleks kombinezonsal fonksiyonlar gerçekleştirilmesine imkan tanır [6].

FPGA'nın paralel işlem yapabilme kapasitesine sahip olması çok daha hızlı sistemlerin tasarlanmasına olanak sağlamaktadır. Bununla birlikte, mikroişlemciler de birer mantık devresi oldukları için FPGA üzerinde kullanılabilirler. Dolayısıyla tek bir tümleşik devre içerisinde kontrol birimi olarak hem işlemci hem de kullanıcıya özgü donanımsal fonksiyonları gerçekleyen fonksiyonlar tanımlamak mümkündür. Tüm sistemin aynı yerde yer alması bağlantılar arası gecikmeler azalacağından FPGA, üzerinde daha da hızlı sistemlerin tasarlanmasına da olanak sağlamaktadır. Bütün bu özelliklerin tasarım sırasında büyük esneklik sağlaması ve ayrıca FPGA'nın paralel işlem yapabilme kapasitesine sahip olduğundan ötürü bu tezin FPGA üzerinde tasarlanması tercih edilmiştir.

3.2. Spartan3E Başlangıç Kiti

Bu tezin yapılma aşamasında Spartan3E Başlangıç Kiti kullanılmıştır. Spartan3E Başlangıç Kiti, Xilinx firmasının Spartan3E FPGA kullanıcılarına hızlı bir başlangıç yapmaları için hazırlanmış geliştirme kartı çözümüdür. FPGA kitinin genel görünümü Şekil 3.3'te görülmektedir. Üzerinde 50 Mhz kristal saat üreteci, paralel flash, 64 MByte Çift Veri Oranlı Senkron Dinamik Rastgele Erişimli Hafıza, Ethernet, iki adet seri port, 4 adet kayan anahtar, 8 adet lamba, 4 adet anlık temaslı buton, 100-pin genişlemeli bağlantısı ve 500.000 kapıya sahip FPGA bulunmaktadır [7]. Tez kit üzerinde gerçeklenirken FPGA ve seri port kullanılmıştır.



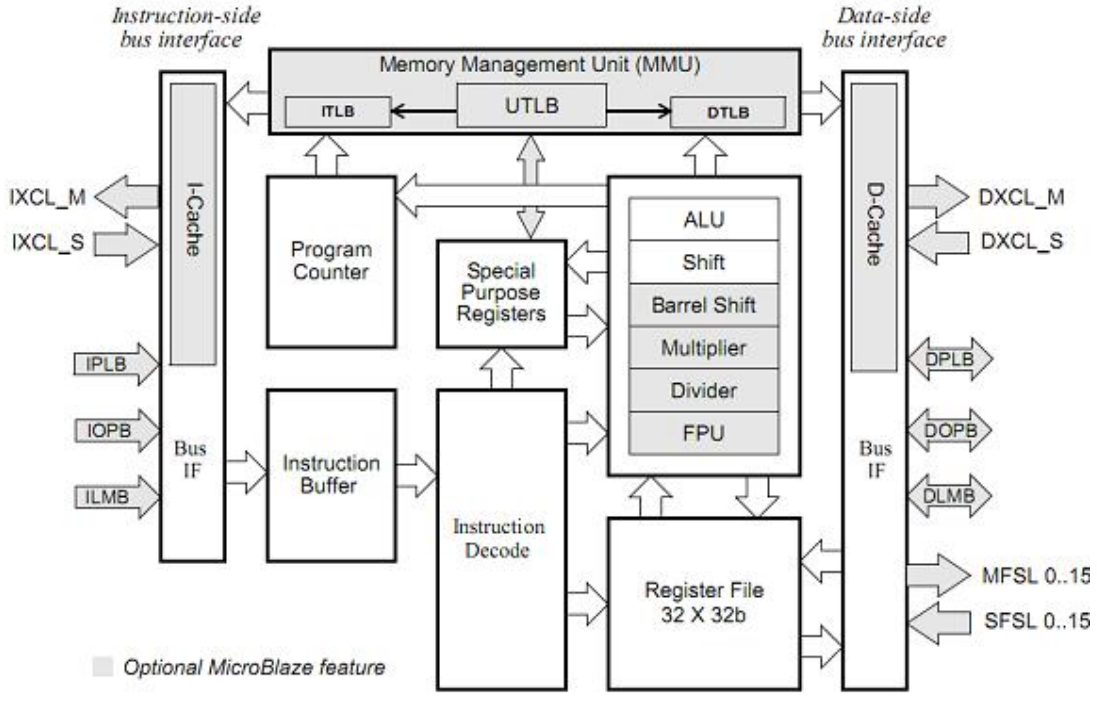
Şekil 3.3 : Spartan3E başlangıç kiti [7].

3.3. Microblaze Mikroişlemcisi

Mimari yapısı Şekil 3.4'te gösterilen Microblaze mikroişlemcisi, FPGA üzerinde yazılım ile kontrol edilebilir gömülü sistemler tasarlamaya olanak sağlamak üzere FPGA bloklarının uygun şekilde programlanması ile oluşturulur. Microblaze, tek FPGA üzerinde kullanılacak çevre birimleri, hafıza ve ara yüz özelliklerinin seçiminde esneklik sağlayarak kullanıcının isteğine tam olarak cevap veren gömülü sistemler tasarlamaya olanak sağlar.

Microblaze 32-bit İndirgenmiş Komut Takımı Bilgisayarı (Reduced Instruction Set Computing, RISC) Harvard bellek mimarisine sahiptir. Program ve veri erişimi ayrı bellek alanlarından sağlanır. Her bir adres alanı 32 bit ile adreslenir [8].

32 bitlik 32 adet genel amaçlı kaydedicileri ve 32 bit adres yolu gibi özellikleri sabit iken, iş hattı (pipeline) derinliği, veri yolu sayısı ve türleri, kayan noktalı sayı birimi (Floating Point Unit, FPU) ve bellek idare birimi (Memory Management Unit, MMU) gibi özellikleri ile FPGA için optimize edilmiş bir mikroişlemcidir [8].



Şekil 3.4 : Microblaze mimarisi [8].

3.4.Verilog Donanım Tanımlama Dili

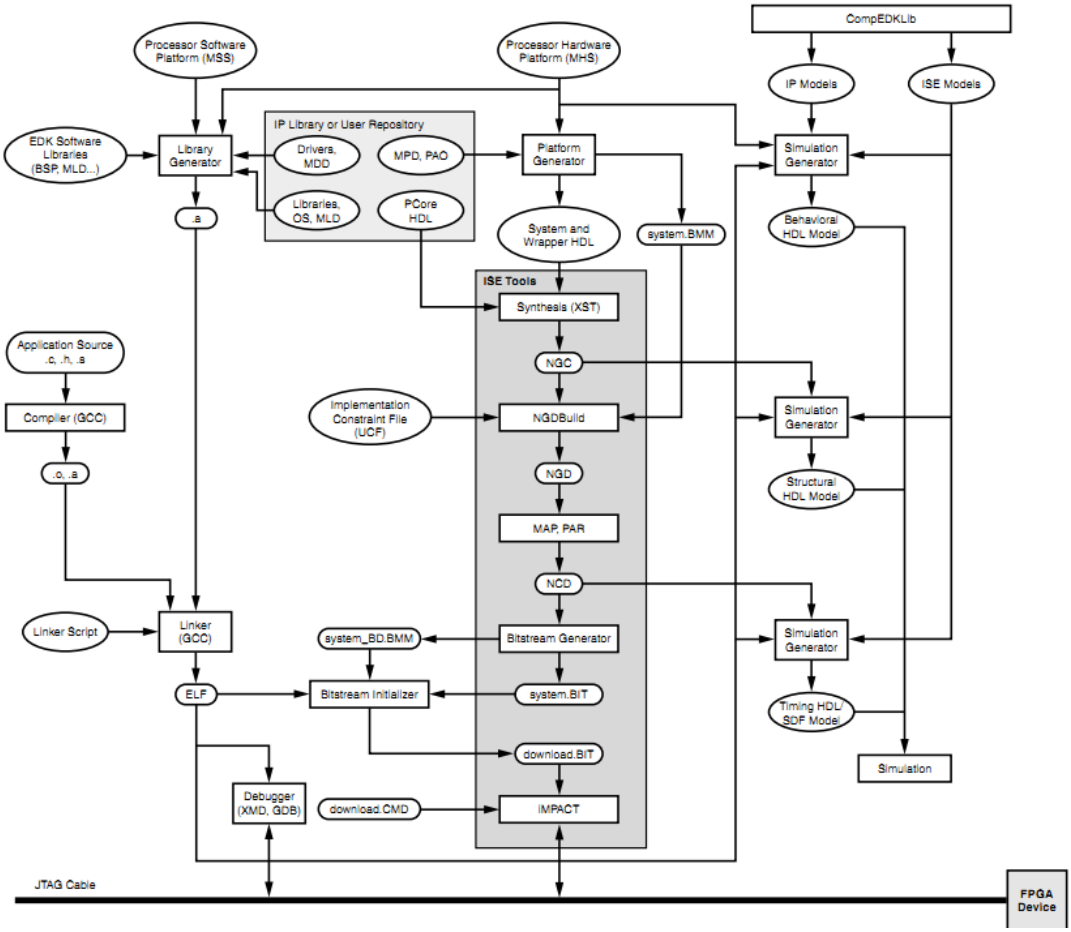
Verilog donanım tanımlama dili 1984-1985 yıllarında Philip Morby tarafından sayısal devreleri modelleme, test etme ve analiz etme amacıyla kolay, basit ve etkili bir şekilde ifade etmeyi hedefleyen bir donanım tanımlama dili olarak geliştirilmiştir. Yapısal olarak C dili ile olan yakınlığı nedeniyle sayısal sistem tasarımı geliştiricilerinin sıklıkla tercih ettiği donanım tanımlama dili haline gelmiştir.

3.5. ISE Ortamı

Tümleşik Yazılım Ortamı (Integrated Software Environment, ISE) FPGA'ları programlamak için kullanılmak üzere Xilinx firmasının geliştirdiği bir ara yüz yazılımıdır. ISE ortamı, donanım tanımlama dilleri ya da şematik çizimler sayesinde FPGA'da çalıştırmak üzere sistemler tasarlamaya olanak sağlar. ISE ortamında donanım tanımlama dilleriyle oluşturulan tasarım sentezleme ve gerçekleme aşamalarından geçirerek kablo aracılığıyla FPGA içine yerleştirilebilir. Ayrıca, donanım üzerinde çalıştırılmadan hata ayıklamak ve hatanın nereden kaynaklandığını görmek üzere ISE ortamında test yapmaya da olanak sağlamaktadır. Ayrıca ISE kullanıcıya tasarladığı sistemlerin ne kadar hat gecikmesine sahip olduğunu, donanım üzerinde ne kadar yer kapladığını ve yapılan tasarımın FPGA'nın hangi bölgesine yerleştirileceğine kadar detaylı bilgiler sunmaktadır.

3.6. EDK Ortamı

EDK (Embedded Development Kit) ortamı Xilinx firmasının ürettiği FPGA'lar üzerinde mikroişlemci tabanlı sayısal sistemler geliştirmek üzere kullanıma sunulmaktadır. EDK çevre birimlerinin ve FPGA donanımlarının bağlanması, sistemin adreslenmesi, haberleşme protokollerinin yazılması gibi işlerle uğraşmak yerine sadece donanım ve yazılım tasarımına odaklanmayı sağlar [9]. Şekil 3.5'te FPGA içerisindeki mikroişlemciyi kullanarak tasarlanan bir sistemin tasarım akış diyagramı görülmektedir. EDK ortamı bu geliştirme aşamaların hepsini tek bir ara yüz programı ile kullanıcıya sunarak çok zahmetli ve karmaşık sistemleri daha kolay tasarlanabilir hale getirir ve proje süresini önemli ölçüde kısaltır.



Şekil 3.5 : EDK sistem geliştirme araçları [9].

EDK geliştirme ortamında “soft core” (Microblaze) veya “hard core” (PowerPC) gibi mikroişlemci temelli donanım projelerine FPGA kartı üzerinde bulunan çevre birimleri ve giriş-çıkış birimleri eklenebileceği gibi Xilinx tarafından geliştirilmiş donanımlar ve kullanıcının ISE aracılığıyla oluşturduğu kendi donanımları eklenebilmektedir [9].

EDK ortamı Temel Sistem Oluşturucu (Base System Builder, BSB) ile tasarımcıya kendi tasarım sisteminin tabanını oluşturma imkanı sağlar. BSB yardımıyla tasarımcı kolaylıkla FPGA kartı üzerindeki istediği donanımların hazır İnternet Protokollerini (Internet Protocol, IP) seçerek tasarımına ekleyebilmektedir. Tasarımcı kendi donanımları ve eklediği bu IP’leri EDK’nın sunmuş olduğu veri yolları ile kolayca bağlayarak sistemi oluşturmuş olur.

EDK donanım projesinin yapılandırılmasında XPS (Xilinx Platform Studio) programı kullanılmaktadır. XPS’de Microblaze mikroişlemcisine bağlanan çevre

birimleri Microblaze tarafından adreslenmektedir. Sistemin adres haritası üretildikten sonra XPS ortamında ya da ISE ortamında tasarlanan projeye sentezleme ve gerçekleme aşamaları uygulanmaktadır. Son olarak bu aşamadan sonra donanım tasarımı bitirilerek bu donanımı kontrol etmek için kullanılan Microblaze yada PowerPC gibi mikroişlemcilerin yazılımının tasarlanması aşamasına geçilmektedir.

3.7. SDK Ortamı

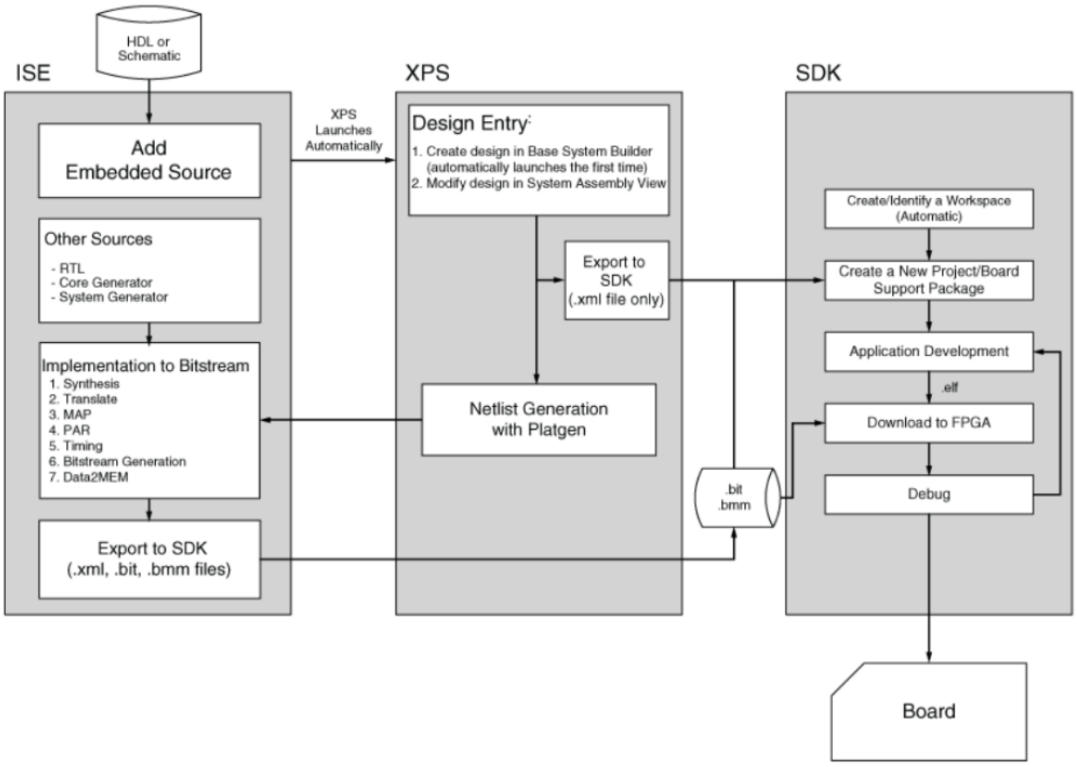
Yazılım Geliştirme Kiti (Software Development Kit, SDK) Xilinx firması tarafından EDK ortamında tasarlanan mikroişlemci merkezli sayısal sistem tasarımlarının yazılım tasarımını gerçeklemek için geliştirilen ara yüz ortamıdır. Xilinx'in tasarım ortamlarının eski sürümlerinde SDK, XPS geliştirme ortamı içerisinde yer almaktaydı. ISE13 sürümünden sonra Xilinx firması SDK'yı XPS ortamından ayırarak XPS'yi sadece donanım tasarlama ortamına dönüştürmüştür. SDK ise sadece tasarlanan donanımlara yazılım tasarımı yapmak amacıyla kullanılmaktadır. EDK ortamında tasarlanan sisteme ait kullanıcı donanımları ve çevre birimlerinin kütüphaneleri üretilerek yazılım tasarımına ilk adımın atılması sağlamaktadır. Aynı zamanda, SDK tarafından üretilen kütüphanelerin söz konusu söz konusu yazılım projesine eklenmesiyle kullanıcıya mikroişlemciyi kolayca kontrol etme olanağı sağlanmaktadır.

- Zengin özellikli C/C++ kod editörü ve derleme ortamı
- Proje yönetimi
- Tasarım yapılandırması uygulaması ve otomatik Makefile üretimi
- Hata navigasyonu
- Kaynak düzeyinde hata ayıklama ve gömülü hedeflerin görünüşü için iyi tümleştirilmiş ortam
- Kaynak kodu sürümü kontrolü

SDK tarafından kullanıcılarına sunulmuş başlıca özelliklerdir [10].

ISE, EDK ve SDK ortamları kullanılarak sıfırdan bir sistem tasarımının akışı Şekil 3.6'da gösterilmektedir. Tasarım akışından bahsedilecek olursa öncelikle ISE ortamında donanım tanımlama dilleri ya da şematik çizimlerle tasarlanan donanımlar EDK ortamında kullanıcı donanımı olarak tanımlanır. EDK ortamında kullanıcı donanımına IP verilerek mikroişlemci merkezli sayısal sistem tasarımına eklenmek

istenen diğer hazır IP'lerle birlikte eklenmektedir. EDK ortamında donanım yapısı tamamlanan sistem SDK ortamına gönderilerek bu aşamada otomatik olarak kütüphaneleri üretildikten sonra yazılım tasarımı yapılmaktadır. Son olarak, donanım ve bu donanımları kontrol etmek için yapılan yazılım da tamamlandıktan sonra SDK aracılığıyla, donanım bilgilerini içeren "bit" uzantılı donanım dosyası ve "elf" uzantılı yazılım dosyası birleştirilerek FPGA'ya gönderilir.



Şekil 3.6 : Sistem tasarımı akışı [11].

4. GERÇEKLENECEK PROTOKOL

Bitime tezinde Martin Feldhofer'ın daha güvenli RFID sistemler tasarlanması adına yazmış olduğu "An Authentication Protocol in a Security Layer for RFID Smart Tags" kimlik doğrulama protokolü FPGA üzerinde gerçeklenecektir. Martin Feldhofer bu protokolde RFID sistem protokollerine yeni bir yaklaşım sunmaktadır. Günümüzde şifreli tanımlama sistemleri ürünlerin güvenliklerini sağlamak için zorunlu hale gelmiştir.

Bu protokol günümüzdeki diğer protokollerden ayrı olarak şifreli şekilde haberleşmeyi yeni bir yaklaşım olarak sunmaktadır. Protokolde etiket için sınırlı işlem kapasitesi, alan kısıtlaması, düşük güç tüketimi gibi kısıtlamalar sebebiyle iki yönlü meydan okuma-yanıt (challenge-response) kimlik doğrulaması şeması önerilmiştir. Paket ve çerçeve biçimleri var olan protokollerde olduğu gibi ISO/IEC 18000 standardını içermektedir [2].

4.1. ISO/IEC 18000 Standardı

Bölüm 2.2'de de belirtildiği gibi ISO/IEC 18000-3 Standardı okuyucu ile etiket arasındaki haberleşmenin hangi frekansta, hangi özellikler veya kısıtlamalar altında olacağını bildirmiştir.

Etiket ile okuyucu arasındaki haberleşme modülasyon ile çalışmaktadır. Okuyucu haberleşmek için endeksi %10 ve %100 olan Genlik Kaydırmalı Anahtarlama (Amplitude Shift Keying, ASK) modülasyonu kullanmaktadır. Veri şifrelemesi "256'da 1" ya da "4'te 1" veri şifreleme biçimi ile mümkün olur. Veri şifreleme biçimine göre çıkış yolu oranı (uplink rate) saniyede 26.69 kbit hızına ulaşabilmektedir [5].

Söz konusu iletişim protokolü okuyucu ve etiket arasındaki komutların ve verilerin iki yönlü olarak aktarılma şeklini tanımlar. Bu protokol "okuyucu önce konuşur" ilkesine dayanmaktadır. Bu ilke, hiçbir etiketin okuyucudan direktif alıp bunu tam anlamıyla çözmeden iletime başlamaması gerektiğini açıklar. Her komut,

okuyucudan etikete olmak üzere ‘istek’ ve etiketten okuyucuya olmak üzere ‘yanıt’ içermelidir. Bu istek ve yanıtlar bir çerçeve içinde Çerçeve Başlangıcı (Start of Frame, SOF) ve Çerçeve Sonu (End of Frame, EOF) ile sınırlandırılmıştır. Bu başlangıç ve bitiş sınırlarının arasında her bir istek ve yanıt çerçevesi; Bayrak, Komut Kodu, Parametreler ve Çevrimsel Hata Denetimi (Cyclic Redundancy Check, CRC) içermektedir [5].

- Bayraklar: Bir ya da iki alt taşıyıcı frekansını ve yanıt için hangi veri oranının kullanılması gerektiğini göstermektedir. Uygun görülen etiketleri adreslemek için ekstra bilgiler sunulmaktadır. Etiketin yanıtı bayrakları kullanarak haberleşme sırasında oluşan hataları göstermektedir.
- Komut Kodu: Bir baytlık sabit, hangi isteğin gönderildiğini göstermektedir. Üç adet temel komut mevcuttur. Bunlardan birincisi olan zorunlu komutlar etiket tarafından gerçekleştirilmelidir. Seçmeli komutlar uygulama için gerekli ise etiket tarafından gerçeklenebilir. Özel komutlar ise kendi komutlarını protokole eklemek isteyen üreticiler tarafından kullanılabilir.
- Parametreler ve Veri Alanları: İstek ve Yanıtı işlemek için gerekli bilgileri barındıran özel komutlardır.
- CRC: Çevrimsel Hata Denetimi kendi hariç SOF’den sonra gelen bütün baytların belli bir algoritma içerisinde hesaplanmasıyla oluşturulmakta ve haberleşme esnasında herhangi bir hata olup olmadığını ortaya çıkarmak için kullanılmaktadır [5].

4.2. Doğrulama Mekanizması

Protokolde şifrelemeli doğrulama mekanizması kullanılmaktadır. Şifreli doğrulama yöntemleri gizli anahtar ve umumi anahtar şifreleme olarak ikiye ayrılmaktadır. Bu iki şifreleme yöntemi de bu protokol kapsamında kullanılabilmektedir. Doğrulama mekanizmasında davalı (claimant) yani talep eden bir birim ve sağlayıcı (verifier) bulunmaktadır. Bu davalı ve sağlayıcının ikisi de etiket ya da okuyucu olabilir. Protokol kapsamında sağlayıcı olan kısım davalı kısma rastgele bir sayı olabilecek kimlik sorma talebi gönderir. Davalı kısım bu rastgele sayıyı sahip olduğu gizli anahtarla işleyerek kimliğini kanıtlamak üzere tekrar sağlayıcı kısma gönderir. Sağlayıcı kısım davalıdan aldığı veriyi kontrol ederek davalının gizli anahtarı bilip

bilmediğine bakar. Güvenlik için bu haberleşme esnasında gizli anahtarın üçüncü bir kişi tarafından ele geçirilmemesi gerekmektedir.

Bu protokolde Şekil 4.1'de görüldüğü gibi iki aşamalı kimlik sorma sistemi kullanılmıştır.

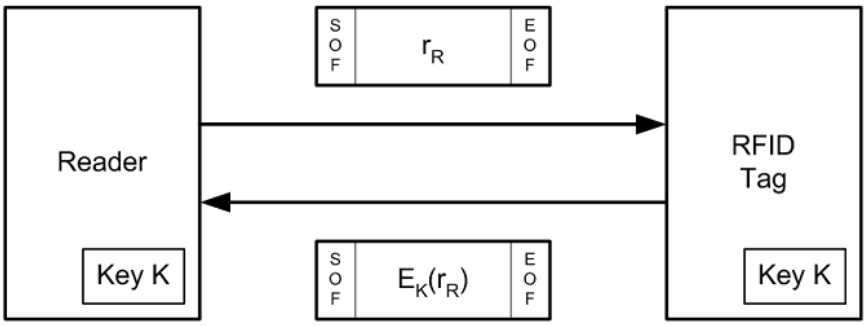
$$A \rightarrow B: \quad E_K(t_A)$$

$$A \leftarrow B: \quad r_B$$
$$A \rightarrow B: \quad E_K(r_B)$$

Şekil 4.1 : Kimlik sorma sistemi [2].

Bu sistemde öncelikle A birimi kendi içinde sakladığı zaman bilgisini şifreleyerek B birimine gönderir. Bu şifreli zaman bilgisini alan B birimi şifreyi çözerek aldığı zaman bilgisini kendi zaman bilgisiyle karşılaştırır. Eğer iki zaman birimi birbirine eşit ise protokolün ikinci aşamasına geçilir. İkinci aşamada, B birimi kendi içinde ürettiği rastgele sayıyı A'ya gönderir. A birimi aldığı bu sayıyı şifreleyerek B birimine gönderir. B aldığı bu şifreli sayıyı çözerek kendi yolladığı rastgele sayı ile karşılaştırır ve bu iki sayı eşitse kimlik sorgulama işlemi başarıyla tamamlanmış olur.

A ve B birimleri aslında Okuyucu ve Etiket bileşenleridir. Protokolün detaylı şeması Şekil 4.2' de gösterilmektedir.



Şekil 4.2 : Doğrulama protokolü [2].

Protokolde okuyucu etikete, rastgele sayı üreteci kullanarak ürettiği 128 bitlik r_R rastgele sayısını gönderir. Etiket ise rastgele sayıyı $E_k(r_R)$ şeklinde şifreleyerek okuyucuya gönderir. Okuyucu bu sayıyı alarak şifresini çözer ve yolladığı sayı ile karşılaştırır. Eğer sayılar aynı ise okuyucu etiketin kimliğini tanıdığından emin olur.

Okuyucu ve etiket arasındaki bu iletişimin sağlıklı olması için belirli bir çerçeve içerisinde sağlanmalıdır. İstek çerçevesi Şekil 4.3'te gösterildiği gibi olmaktadır.

SOF	Flags	0xA0	IC Mfg code	UID	Random number r_R	CRC	EOF
	8 bit	8 bit	8 bit	64 bit	128 bit	16 bit	

Şekil 4.3 : İstek çerçevesi [2].

İstek çerçevesi 8 bitlik bayraklar, 8 bitlik 0xA0 komutu, 8 bitlik kullanıcıya özel komut, 64 bitlik her etikete özgü UID (Unique Identifier), 128 bitlik rastgele sayı ve son olarak 16 bitlik CRC içermektedir [2].

Cevap çerçevesi de neredeyse istek çerçevesine benzer olarak Şekil 4.4'te görülmektedir.

SOF	Flags	UID	Signed data $E_K(r_R)$	CRC	EOF
	8 bit	64 bit	128 bit	16 bit	

Şekil 4.4 : Cevap çerçevesi [2].

Cevap çerçevesi ise 8 bitlik Bayraklar, 64 bitlik her etikete özgü olan UID (Unique Identifier), 128 bitlik şifrelenmiş olan rastgele sayı ve son olarak 16 bitlik CRC içermektedir [2].

4.3. TEA Şifreleme Algoritması

Bölüm 4'te söylendiği gibi günümüz RFID sistemlerinde atakları önlemek ve daha güvenli haberleşme ortamı sağlamak için etiket ve okuyucu arasındaki haberleşmenin şifreli olarak yapılması zorunluluk haline gelmiştir. Protokolde bu şifreleme işlemini gerçekleştirmek için Gelişmiş Şifreleme Standardı (Advanced Encryption Standard, AES) kullanılmıştır. Ancak günümüzde teknolojinin gelişmesiyle birlikte küçülen etiket boyutlarıyla paralel olarak yeni şifreleme standartları ortaya çıkmaktadır. Gerek etiketin boyut kısıtları gerekse daha az enerji tüketme zorunluluğundan dolayı daha az enerji tüketen ve gerçeklendiğinde daha az yer kaplayacak şifreleme standartları kullanma zorunluluğu ortaya çıkmıştır. Bu zorunluluktan ötürü protokolün gerçeklenme aşamasında Wikram Reedy Andem'in 2003 yılında gömülü

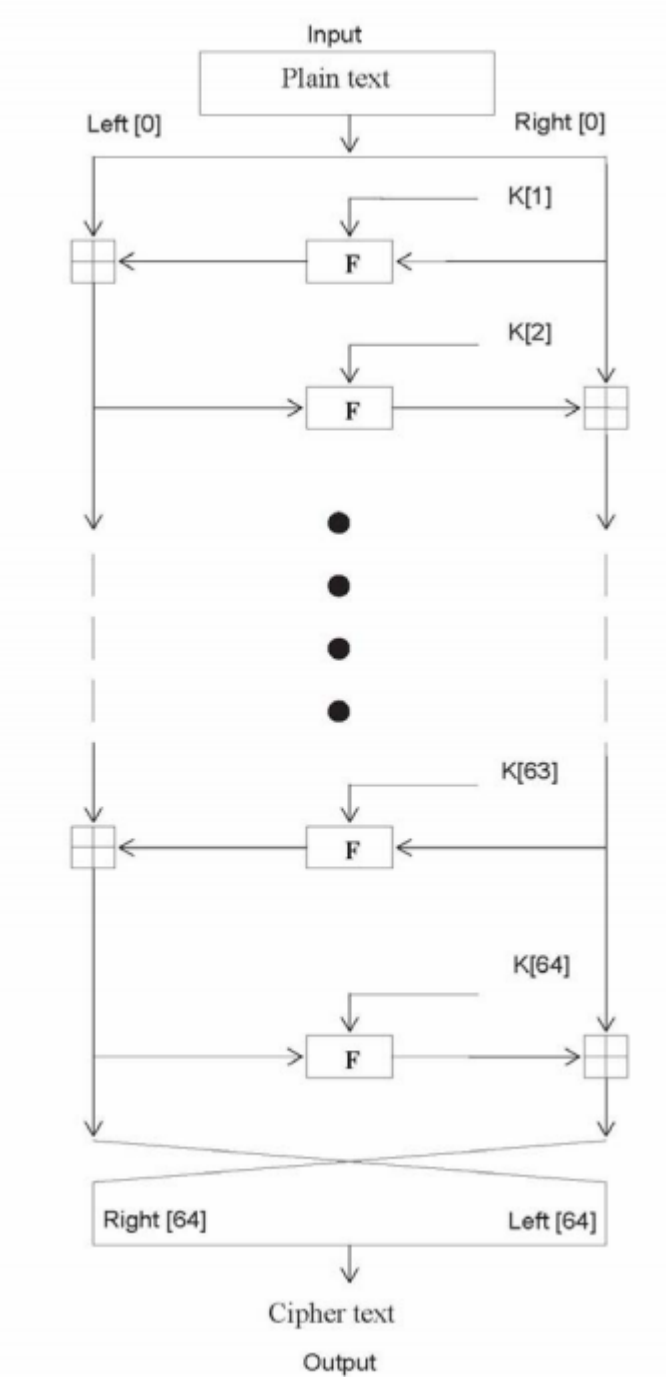
sistem uygulamaları için geliştirmiş olduğu TEA kullanılmıştır. Gömülü sistemlerdeki yüksek performansı, gerçekleme kolaylığı, hızlı olması, düşük enerji tüketimine imkan vermesi, düşük masraflı olması ve güvenli olması hafif (lightweight) olması özelliği ile TEA gömülü sistem tasarımlarına oldukça uygundur [12].

TEA minimum hafıza alanı ve maksimum hız hedeflenerek oluşturulmuş bir şifreleme algoritmasıdır. Karışık cebirsel işlemleri kullanan ve Feistel türü bir şifreleme yapan bir algoritmadır. Feistel türü şifreleme türü olduğu için blok şeklinde şifreleme yöntemine göre oluşturulmuştur. TEA ayrımsal şifre analizine (Differential Cryptanalysis) oldukça dirençlidir. Aynı zamanda sadece altı tur sonra tam yayınım sağlamaktadır. Bunun anlamı, şifrelenecek metinde 1 bit değiştirildiğinde çıkıştan alınan şifreli metine bu değişiklik 32 bit olarak yansımaktadır. Zaman performansı ise bilgisayarlarda ve iş istasyonlarında oldukça etkileyicidir [13].

TEA algoritması blok şifreleme yapısında olması sebebiyle girişine uygulanan 64 bitlik metni bit bit olarak şifrelemek yerine 64 biti tek blok olarak alarak sanki tek bir bitmiş gibi şifrelemektedir [13]. TEA simetrik yapılı şifreleme algoritması olması nedeniyle şifreleme ve şifre çözme algoritmaları yapıları birbirlerine oldukça yakındır.

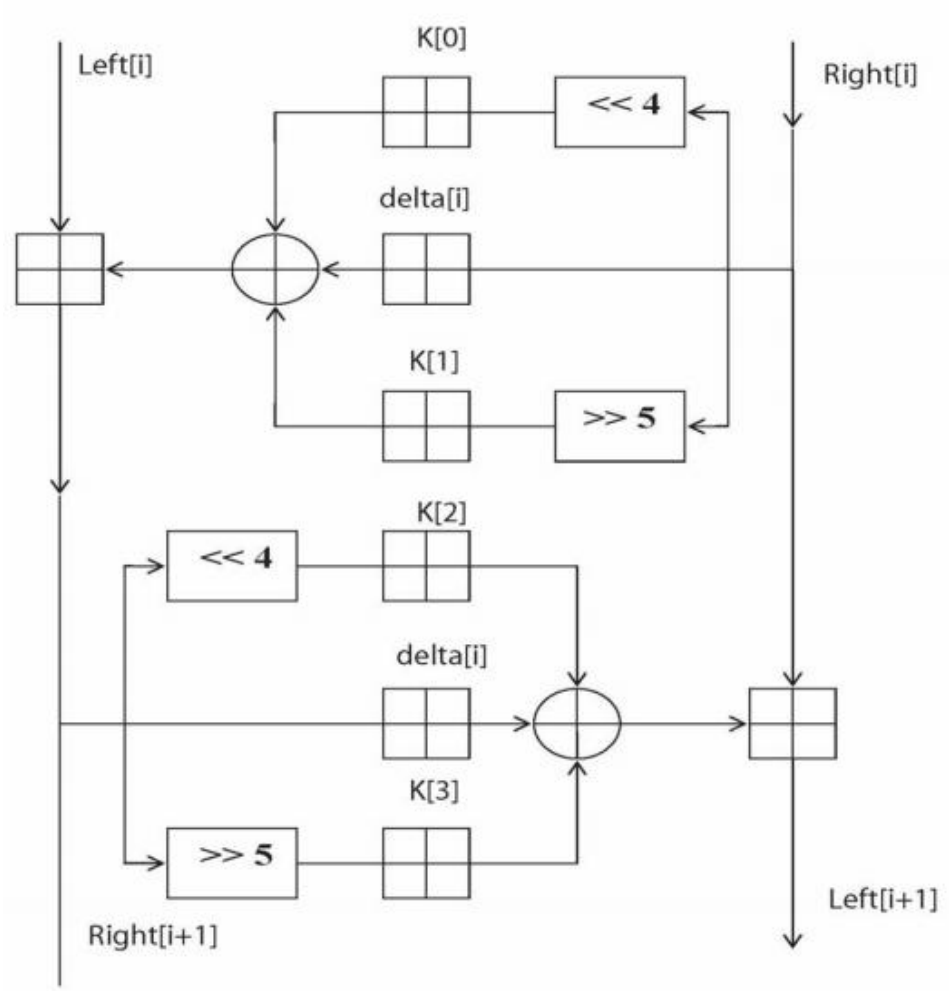
TEA şifreleme ve şifre çözme yapısında 128 bit uzunluklu şifreleme anahtarı kullanılmaktadır. Bu 128 bit uzunluklu anahtar K[0], K[1], K[2] ve K[3] şeklinde dört adet 32 bit uzunluklu anahtarlara bölünerek işlemlere sokulur [13]. Şekil 4.5'te algoritmanın şifreleme yapan kısmının blok diyagram yapısı görülmektedir. Şifreleme yapısı 64 adet Feistel döngüsünden meydana gelmektedir. Şifrelenmek istenen metin 32'şer bitlik iki kısma bölünerek sağ ve sol taraftan şifrelenmek üzere döngüye sokulur. Her bir döngüde farklı anahtar kullanılmaktadır. Girişe uygulanan Sol[0] ve Sağ[0] girişleri K[0], K[1], ..., K[64] anahtarları tarafından şifrelenerek bu 64 döngünün sonunda Sol[64] ve Sağ[64] çıkışlarından şifrelenmiş metin halinde çıkmaktadırlar [13]. Yapı, her bir döngünün girişi bir önceki döngünün çıkışına bağlanacak şekilde oluşturulmuştur. Her bir döngüye giren 64 farklı K[i] anahtarı, 128 bit şifreleme anahtarının "delta" isimli altın orandan üretilen bir sabitin işleme sokularak oluşturulmaktadır. Delta sabitinin ilk değeri Denklem 4.1'den hesaplanmaktadır [13].

$$\text{delta} = (\sqrt{5} - 1)*2^{31} = 9E3779B9_h \quad (4.1)$$



Şekil 4.5 : TEA şifreleme rutini [13].

Şifreleme kısmının iç yapısına değinilecek olunursa, 64 Feistel döngüsünde oluştuğu bilinen şifreleme yapısının Şekil 4.6'da görüldüğü gibi 32 döngüden oluşmaktadır. Yani Şekil 4.6 iki tane Feistel döngüsü içermektedir. Her bir döngüde döngüye giren metinler toplama, özel veya (exclusive or, XOR), mantıksal kaydırma işlemlerine tabii tutulmaktadır.

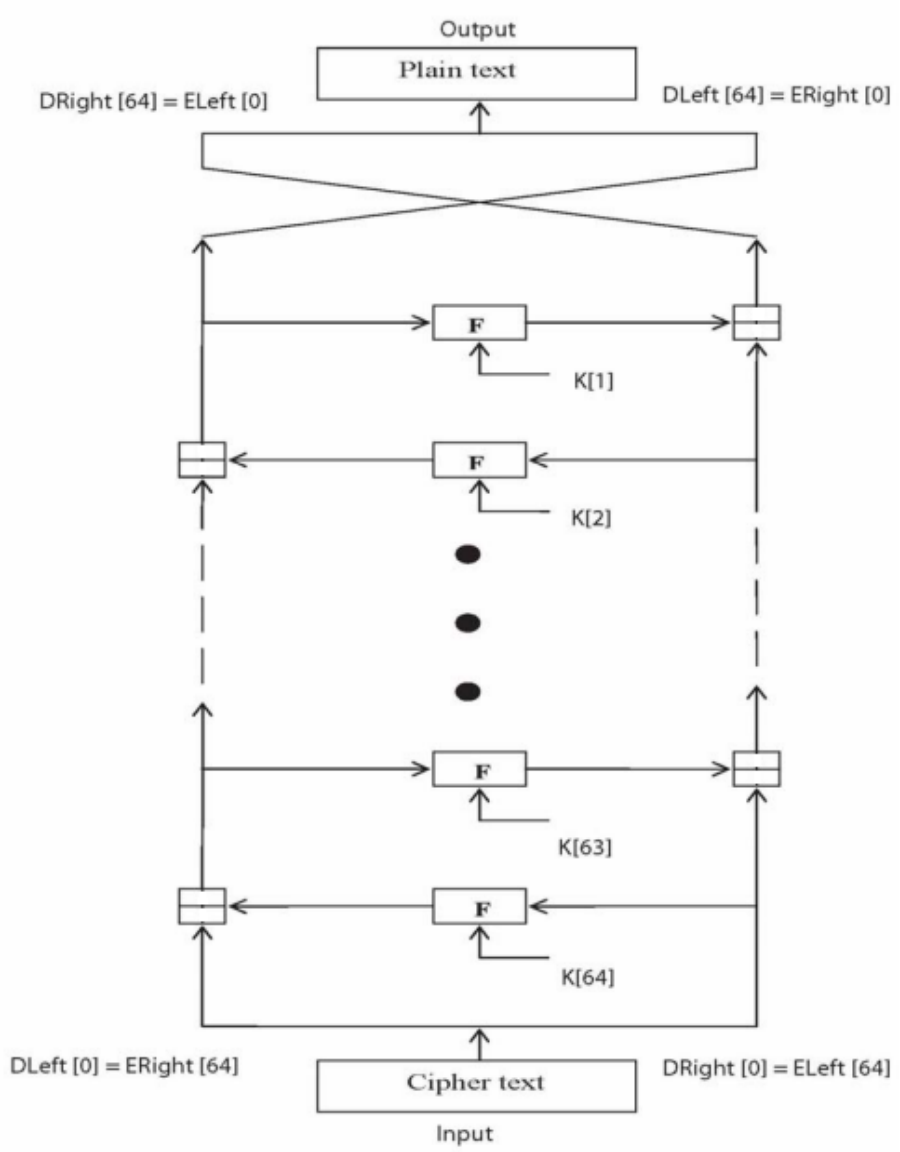


Şekil 4.6 : TEA i. döngüsü [13].

Şekil 4.6'da da görüldüğü gibi bir döngüye sağ taraftan giren şifresiz metine öncelikle 4 bit sola kaydırma işlemi uygulanır ve daha sonra K[0] anahtarıyla toplama işlemine girer. Yine aynı bilgi 5 bit sağa kaydırılarak K[1] anahtarıyla toplama işlemine girmektedir. Bir de bilginin kendisi direk olarak delta[i] sabitiyle toplama işlemine girmektedir. Daha sonra işleme giren bu üç koldan gelen veriler XOR işlemine girerek sol koldan metinle toplanmaktadır. Bu toplamın sonucu o döngünün sol taraftan verdiği çıkışı olarak bulunmaktadır. Sol taraftan çıkan bu bilgi bir Feistel turu önce sağdan girmiş olan metinle aynı işlemlere tabii tutularak bu döngünün sağ taraftan verdiği çıkış sonucu elde edilmektedir. Bu işleme bu şeklide 32 tur devam edilerek son turun çıkışında girişten verilen şifresiz metinlerin şifreli halleri elde edilmektedir.

TEA simetrik yapılı şifreleme algoritması olması sebebiyle şifre çözme yapısı da şifreleme yapısıyla benzer olmaktadır. TEA Şifre çözme yapısı Şekil 4.7'de

görüldüğü üzere şifreli metinin çıkıştan girişe doğru işlenmesi şeklindedir. Şifre çözme yapısını şifreleme yapısından ayıran bir diğer önemli nokta ise şifresi çözülecek metinin başlangıçta K[64] anahtarı kullanılarak işleme sokulmasıdır. Yani anahtarların sırası da tamamen yer değiştirmiştir. Aradaki son fark ise sağ ve solda görülmekte olan ana kollardaki toplama işlemlerin yerini çıkarma işlemleri alması şeklinde olmaktadır.



Şekil 4.7 : TEA şifre çözme rutini [13].

4.4. Rastgele Sayı Üreteci

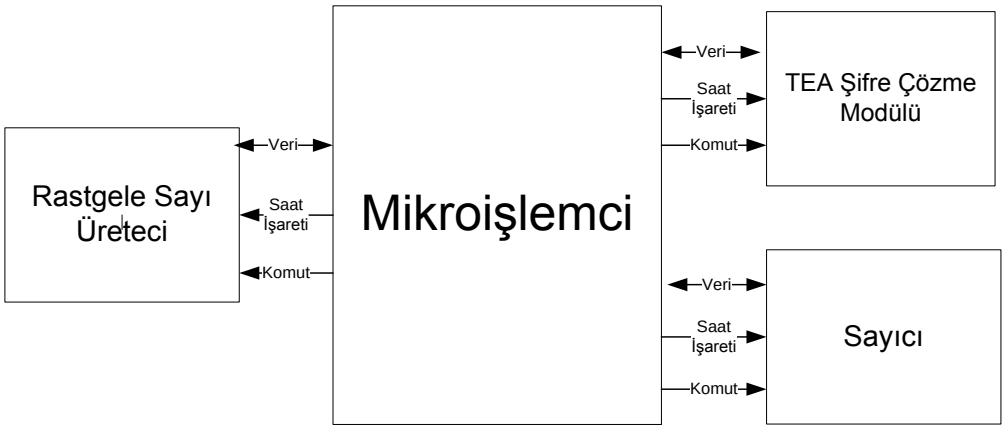
Bölüm 4.2'de bahsedildiği üzere protokolün doğrulama mekanizmasının gerçekleştirilmesi aşamasında okuyucudan etikete bir rastgele sayı gönderilmesi gerekmektedir. Bu 64 bit uzunluklu rastgele sayıyı üretmek için sistemin okuyucu kısmına rastgele sayı üreteci eklenmesi gerekmektedir. Sistemin daha güvenilir bir hale gelmesi açısından üretilen rastgele sayının tahmin edilemez olması önemlidir. Tamamen rastgele bir sayı üretmek ayrı bir tasarım yükü getireceğinden bu tez aşamasında ürettiği sayıların rastgele olduğu varsayılan doğrusal geri beslemeli ötelemeli kaydedici (Linear Feedback Shift Register, LFSR) kullanılmıştır.

LFSR donanım gerçeklemeleri için uygun olması, büyük periyotlu dizi üretimi özelliği, iyi istatistiksel özellikli dizi üretimi özelliği ve yapısının cebirsel teknikleri kullanarak basit bir şekilde ifade edilebilmesinden ötürü sıklıkta tercih edilmektedir [14].

Sözde-rastgele sayı üreteci olan LFSR girişine tohum yani başlangıç değeri uygulanarak her saat darbesi geldiğinde farklı bir rastgele sayıyı çıkışından vermektedir. L bit uzunluklu bir LFSR'nin çalışma yapısı, ilk olarak aldığı L bit uzunluklu tohum değerini her saat işareti geldiğinde bir düşük anlamlı bitine kaydırır. Her bit bir anlamsız basamağa kaydığı zaman boşta kalan en anlamlı bit olan L-1 bitine diğer bitlerin bir kaçının XOR işlemine tabii tutulmasıyla elde edilen değer atanır. LFSR belirli bir periyodu tamamlandığında başlangıç değerine geri dönmektedir. Bu sebepten dolayı daha güvenli sistemler gerçeklemek için bu periyotun mümkün olduğu kadar uzun tutulması gerekmektedir. Protokolün gerçekleme aşmasında 64 bit uzunluklu rastgele sayı gerekliliği olduğundan ötürü bu bit uzunluğunda bir LFSR için maksimum periyodu sağlayacak XNOR barındıran bir geri-besleme fonksiyonu kullanılmıştır.

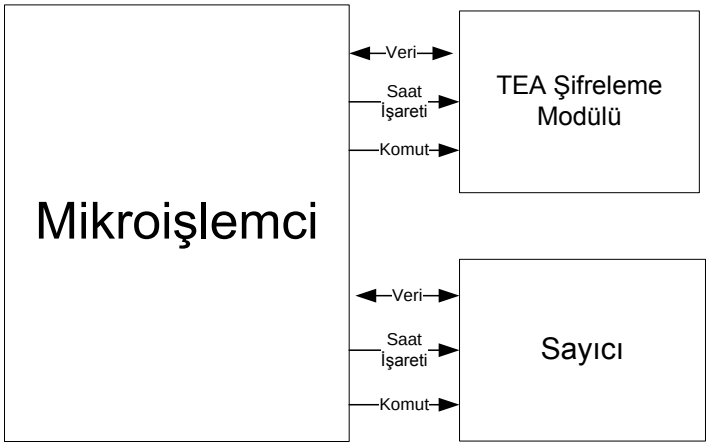
4.5. Mikroişlemci

Bitirme tezi mikroişlemci merkezli yapı ile gerçeklenecektir. Tasarlanan donanımlar ve sistemin tamamı üzerinde yazılım koşturulan mikroişlemci tarafından kontrol edilecektir. Şekil 4.8'de tasarlanacak okuyucu kısmının genel yapısı görülmektedir. Protokolde okuyucu kısmında bulunan mikroişlemci şifre çözme donanımını, sayıcı donanımını ve rastgele sayı üretecini kontrol etmekle sorumludur.



Şekil 4.8 : Mikroişlemcili okuyucu yapısı.

Şekil 4.9'da ise etiket yapısının genel hali görülmektedir. Etiket kısmındaki mikroişlemcinin görevi ise şifreleme donanımını ve sayıcı donanımını kontrol etmektir.



Şekil 4.9 : Mikroişlemcili etiket yapısı.

Her iki yapıda da mikroişlemci donanımlarla iki taraflı veri haberleşmesini ve tek taraflı komut ve saat işareti göndermeyi sağlayacaktır. Mikroişlemcinin bir diğer görevi ise okuyucu kısmı için etiketle, etiket kısmı için de okuyucuyla arada olan haberleşmeyi sağlamaktır. Bu tezde sistemi daha da hızlandırmak adına sayıcı donanımı kullanmak yerine yazılım aşamasında değişken olarak tanımlanıp kontrol edilecektir.

5. KİMLİK DOĞRULAMA PROTOKOLÜNÜN GERÇEKLENMESİ

Protokolün gerçeklenmesi aşamasına ilk olarak protokolde yer alan donanımların Verilog donanım tanımlama dili kullanılarak FPGA üzerinde tasarlanmasıyla başlanılmıştır. Protokolde bulunan donanımlar; etiket kısmında yer alan TEA şifreleme donanımı, okuyucu kısmında yer alan TEA şifre çözme donanımı ve yine okuyucu kısmında bulunan rastgele sayı üreteci donanımlarıdır. Donanım tasarımı aşaması tamamlandıktan sonra bu donanımları ve tüm sistemi kontrol etmek amacıyla FPGA içerisindeki Microblaze mikroişlemcisi üzerinde yazılım tasarımı yapılmıştır.

5.1. Donanım Tasarımı Aşamaları

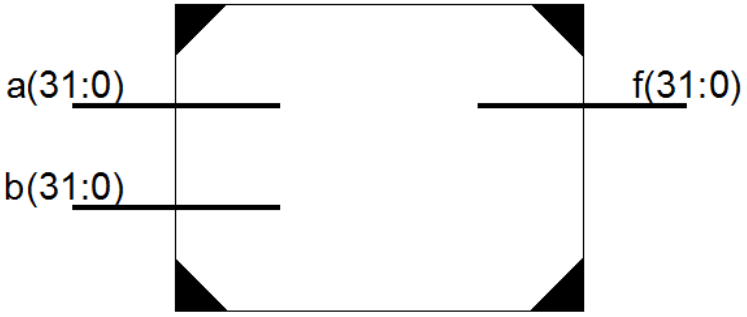
Şifreleme ve şifre çözme donanımlarının bir turunun yapısı Şekil 4.6'da gösterilmişti. Bu yapıya istinaden şifreleme ve şifre çözme donanımlarından önce bu iki donanımın alt blokları olan toplama, çıkarma, mantıksal kaydırma ve özel veya donanımları tasarlanmıştır. Daha sonra tasarlanan toplama, çıkarma, mantıksal kaydırma ve özel veya donanımları kullanılarak şifreleme, şifre çözme donanımlarının tasarlanması tamamlanmıştır. Bir diğer donanım olan rastgele sayı üreteci diğer donanımlardan bağımsız olarak tasarlanmıştır.

Donanım tasarımı yapılırken etiket ve okuyucunun gereksinimlerine uygun yapılar göz önünde bulundurulmuştur. Bunlardan birinci derecede önem arz eden gereksinimler, pasif etiketler kendi içlerinde güç kaynağı barındırmadıkları için etiket kısmı için yapılacak tasarımın oldukça az enerji tüketecek yapıda olmasıdır. Yine etiket kısmı için bir diğer önemli tasarım gereksinimi ise etiketlerin taşınabilir olması nedeniyle alanda daha az yer kaplayacak şekilde tasarlanması hedeflenmiştir. Okuyucu kısmı için tasarım hedeflerine değinilecek olunursa, birçok etiketin aynı anda okuyucu ile haberleşme yapma olasılığını göz önünde bulundurarak okuyucunun etiketlere göre çok daha hızlı sistem yapısına sahip olması gerekmektedir. Okuyucular genelde taşınamaz yapılar olmaları nedeniyle kendi içlerinde güç kaynakları barındırdıklarından Bölüm 2.2'de bahsedilmişti.

Okuyucuların güç ve kapladıkları alan konusunda sıkıntıları olmadığı için okuyucu kısımdaki donanımların tasarımı yapılırken olabildiğince hızlı sistem tasarımı hedeflenmiştir.

5.1.1. Toplama Bloğu

Şifreleme ve şifre çözme yapılarında 32 bit uzunluklu iki sayıyı toplamak amacıyla toplama bloğu tasarlanmıştır. Sistemin daha hızlı hale getirmek amacıyla seri toplama yerine paralel toplama bloğu tasarlanmıştır. Toplama bloğu tasarımında ilk olarak iki yarı toplayıcı yapısı kullanarak bir tam toplayıcı elde etme amaçlanmıştır. Ancak elde edilen 32 bit uzunluklu tam toplayıcının ISE'nin kullanıcıya sunmuş olduğu "+" operatörü kullanılarak tasarlanan toplayıcıdan daha fazla alan kapladığı görülmüştür. Bunun üzerine toplama bloğu "+" operatörü kullanılarak tasarlanmıştır. Toplama bloğu şifre çözme ve şifreleme yapılarında çok sayıda kullanılacağı için tasarımda az yer kaplaması oldukça önem arz etmektedir.

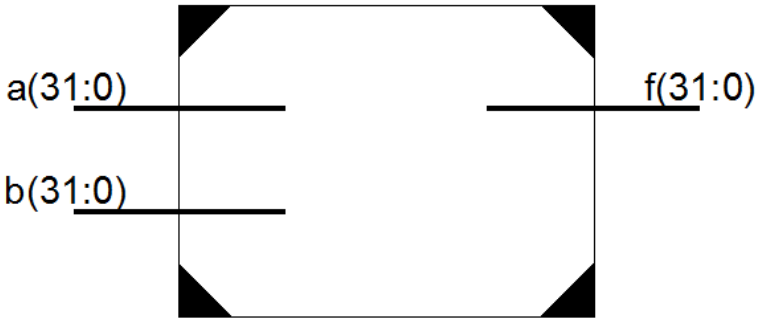


Şekil 5.1 : Toplama bloğu.

Şekil 5.1'de toplama bloğunun genel yapısı görülmektedir. Toplama bloğu iki adet 32 bit uzunluklu sayıyı girişlerinden alarak bu iki sayının toplamlarını "f" çıkışından yine 32 bit uzunluklu olarak vermektedir. Toplama işleminin sonucu belirli değerler toplandığında 33 bit uzunluklu olduğunda ise en anlamlı biti atarak diğer 32 biti çıkışa vermektedir.

5.1.2. Çıkarma Bloğu

Şifre çözme donanımının tasarlanması aşamasında gerekli olan bir diğer blok ise 32 bit uzunluklu iki sayının farkını veren çıkarma bloğudur. Tasarım aşamasında toplama bloğunda olduğu gibi çıkarma bloğunda da en az alan kaplayan ve en az kapı gecikmesine sahip blok "-" operatörünü kullanarak elde edilmiştir. Çıkarma bloğunun genel hali Şekil 5.2'de görüldüğü gibi olmaktadır.

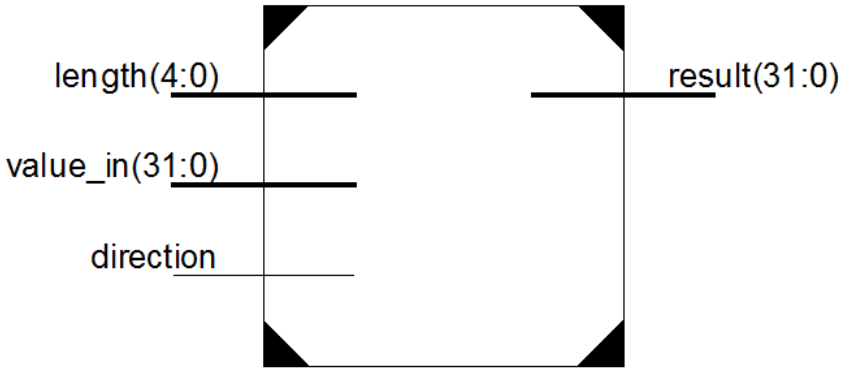


Şekil 5.2 : Çıkarma bloğu.

Çıkarma bloğunun çalışma prensibi bloğun ‘a’ ve ‘b’ girişlerine uygulanan 32 bit uzunluklu sayıların farkını alarak yine çıkışa 32 bit uzunluklu sonucu vermektir.

5.1.3. Mantıksal Kaydırma Bloğu

Mantıksal Kaydırma Bloğu hem şifreleme hem de şifre çözme yapısında kullanılmıştır. Şekil 4.6’da da görüldüğü üzere her bir döngüde iki tane sola 4 bit kaydırma işlemi 2 tane de sağa 5 bit kaydırma işlemi bulunmaktadır. Bu sağa ve sola bit kaydırma işlemleri tek bir blok tasarlanarak gerçekleştirilmiştir. Sadece içerisindeki parametreler değiştirilerek bu blok istenilen uzunlukta istenilen tarafa doğru kaydırma işlemi yapabilmektedir. Mantıksal kaydırma bloğunun genel yapısı Şekil 5.3’te görüldüğü gibidir.



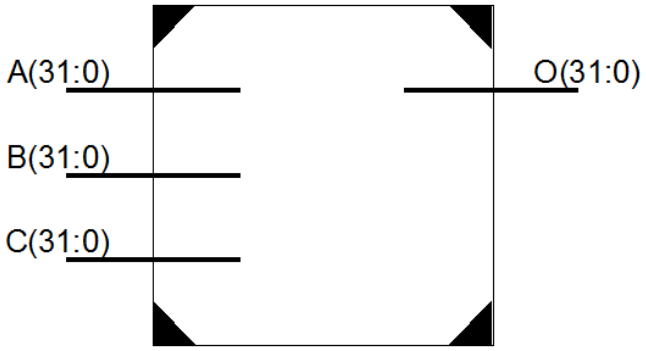
Şekil 5.3 : Mantıksal kaydırma bloğu.

Blokta 5 bit uzunluklu ‘lenght’, 32 bit uzunluklu ‘value_in’ ve 1 bit uzunluklu ‘direction’ girişleri bulunmaktadır. Bloğun çıkışı ise 32 bit uzunluklu ‘result’ değeridir. Blokta yön seçme girişi yüksek iken blok sol tarafa kaydırma işlemi yön seçme girişi düşük iken sağ tarafa doğru kaydırma işlemi yapmaktadır. Bloğun ne kadar uzunlukta kaydırma yapacağı ise bloğun ‘lenght’ girişi değiştirilerek ayarlanmaktadır. Blok 4 bit sola kaydırma işlemi için kullanıldığında seçme girişi

yüksek seçilip kaydırma uzunluğu girişi de 4 seçilerek kaydırılmak istenen 32 bit uzunluklu veri girilir. Blok 4 bit sola kaydırılmak istenen sayının her bir bitini 4 defa bir yüksek anlamlı bite taşır ve en sağda kalan 4 boş bitleri de sıfır değeri ile doldurur. Yine 5 bit sağa kaydırma işleminde ise seçme girişi düşük seçilip kaydırma uzunluğu 5 seçilerek kaydırılmak istenen 32 bit uzunluklu veri girişe uygulanır. Blok 5 bit sağa kaydırılmak istenen sayının her bir bitini 5 defa bir düşük anlamlı bite kaydırır. Bu işlem sonucunda en sağda kalan 5 bite sıfır değeri atanır.

5.1.4. Özel Veya Bloğu

Şifreleme ve şifre çözme yapılarının ikisinde de özel veya bloğu bulunmaktadır. İki yapı içerisinde de bir tur içinde iki adet özel veya bloğuna ihtiyaç duyulmaktadır. Şekil 4.6'daki yapıda da görüldüğü üzere tasarım aşamasında kullanılacak özel veya bloğu 32 bit uzunluklu üç adet veri girişi barındırmalıdır. Şekil 5.4'te özel veya bloğunun genel yapısı görülmektedir. Bu blok girişine uygulanan üç verinin özel veya işlemine girmesiyle 32 bit uzunluklu çıkış verisi üretmektedir.



Şekil 5.4 : Özel veya bloğu.

5.1.5. Şifreleme Bloğu

Şifreleme ve şifre çözme donanımlarının alt donanım blokları tasarımları tamamlandıktan sonra TEA şifreleme ve TEA şifre çözme üst bloklarının tasarımına geçilmiştir. Belirlenen tasarım hedefleri göz önünde tutularak en uygun tasarımlar yapılmıştır.

Şifreleme donanımının mümkün olduğunca az alan kaplaması ve az güç tüketmesi gerekliliği Bölüm 5.1'de belirtilmişti. Bu tasarım hedefi doğrultusunda TEA şifreleme donanımının saat işareti ile eş zamanlı olarak çalışmasına karar verilmiştir. Şekil 4.6'da şifreleme yapısının 32 turundan sadece 1 tanesi görülmektedir.

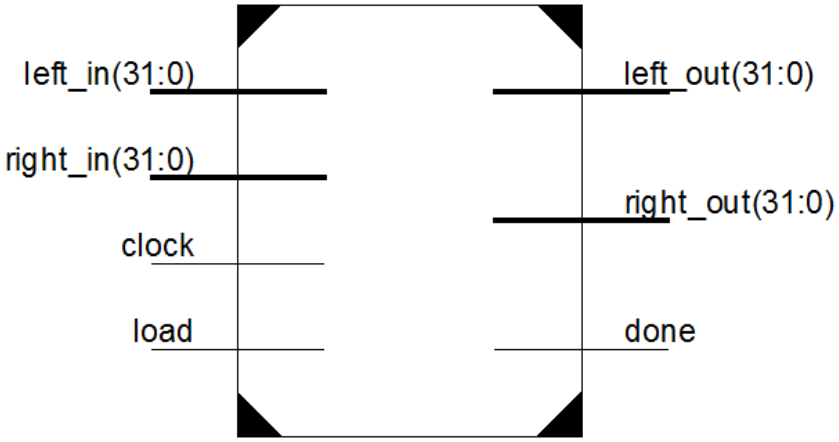
Şifreleme kısmında önce bu bloktan 1 tane tasarlanarak daha sonra gerekli bağlantılar yapılarak şifrelenmek için girişe uygulanan verinin 32 defa aynı bloğa uygulanması yöntemi kullanılmıştır.

Şifre çözme donanımının olabildiğince hızlı çalışması gerekliliği yine Bölüm 5.1'de belirtilmişti. Bu tasarım hedefi doğrultusunda şifre çözme bloğunun tamamen kombinezonsal olarak tasarlanmasına karar verilmiştir. Şekil 4.6'da görülen şekilden 32 adet arka arkaya bağlanarak oluşturulmuştur. Yani saat işaretinden bağımsız halde girişine şifresi çözülmek üzere verilen verinin şifresini çözerek çıkışa vermektedir.

5.1.5.1. Saat İşaretli Şifreleme Bloğu

Saat işaretine sahip TEA şifreleme bloğu tasarımda oldukça az alan kaplamakta ve buna bağlı olarak oldukça az güç tüketmektedir. Bu şifreleme donanımı FPGA üzerinde sadece 175 dilim kaplamaktadır. Bu tez aşamasında kullanılan başlangıç kitinin toplam 4656 dilime sahip olduğu düşünülürse 175 dilim kullanarak şifreleme donanımını gerçeklemek tasarım açısından oldukça başarılıdır.

Şifreleme bloğunun genel yapısı Şekil 5.5'te görüldüğü gibidir. Şifreleme bloğu donanımı 4 adet mantıksal kaydırma, 8 adet toplama ve 2 adet özel veya bloğu içermektedir. Verilog dili ile bu alt bloklar birbirlerine Şekil 4.6'da gösterildiği şekilde bağlanmıştır. Bu bağlama işlemleri yapılırken her bir bağlantıya birer kablo ya da kaydediciler atanarak bu kablolar ve kaydediciler yapı sağlanacak şekilde alt donanımların giriş ve çıkışlarına atanmışlardır. Şifreleme donanımının anahtar değeri içeride 128 bit uzunluklu bir kablo ile saklı tutulmuştur. Anahtar değerinin 32 bit uzunluklu 4 parçasının gerekli alt bloklara bağlantıları atama işlemleriyle sağlanmıştır. Şifreleme kısmında her bir döngüde farklı olmak üzere kullanılacak delta değişken değerleri ise hesabı fazladan alan yükü getirmemesi amacıyla Denklem 4.1'den Matlab programı yardımı kullanılarak hesaplanmışlar ve dizi olarak kodun içerisinde tutulmuşlardır.



Şekil 5.5 : Saat işaretli şifreleme bloğu.

Şifreleme donanımının yükleme girişi yüksek seviyede iken şifrelenecek olan 32 bit uzunluklu iki veri bloğun 32 bit uzunluklu iki girişine uygulanarak şifreleme işlemine ilk adım atılır. Bu arada saat işareti sürekli olarak donanıma uygulanmaktadır. Şifrelenecek olan sayılar donanım tarafından alındığında ve yükleme girişi yüksekte iken alınan sayılar donanım içerisinde iki tane 32 bit uzunluklu kaydediciye atılmaktadır. Bunun sebebi şifrelenecek sayıların saat ile eş zamanlı biçimde işlenmesi için değişken bir değere atama gerekliliğidir. Aynı zamanda daha önceden hesaplanan delta değişken değerleri de başlangıçta bir kaydediciye atılmıştır. Bu kaydedicilere değerler atandığında yükleme girişi düşük seviyeye getirilerek şifreleme işlemi başlatılmaktadır. TEA'nın yapısından da görüldüğü üzere şifreleme mekanizması 32 döngü sonunda tamamlanmaktadır. Döngü sayısını kontrol etmek için bloğun içerisine bir sayıcı eklenmiştir. Sayıcıya başlangıçta 1 değeri atanmakta ve her saat darbesinde sayıcı bir değer arttırılmaktadır. Sayıcı değişkeni 32 değerini aldığında donanımın 'done' çıkışı kendiliğinden yüksek değere çıkarak şifreleme işleminin bittiğini haber vermektedir.

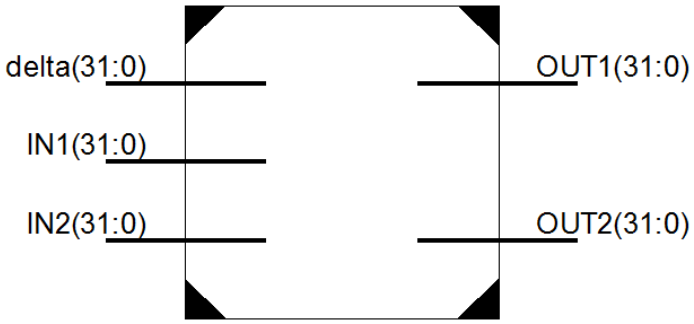
Tasarlanan bloğun ISE aracı yardımıyla yerleştirme ve hat çizimi sonrası benzetimi EK A'da görülmektedir.

Benzetim sonuçlarının doğruluğu şifreleme algoritmasının anlatıldığı kaynaklarda mevcuttur. Test aşamasında makalede de bahsedildiği gibi girişlerin her ikisine de $(00000000)_{16}$ verileri uygulanarak çıkışlardan $(9327C497)_{16}$ ve $(31B08BBE)_{16}$ şifreli verileri elde edilmiştir. Benzetim sonuçlarında da görüldüğü üzere donanım 32 saat darbesi sonunda doğru sonucu vermektedir. Doğru sonucu verdiğinde 'done' çıkışının seviyesi yükselmektedir.

5.1.5.2. Saat İşaretsiz Şifreleme Bloğu

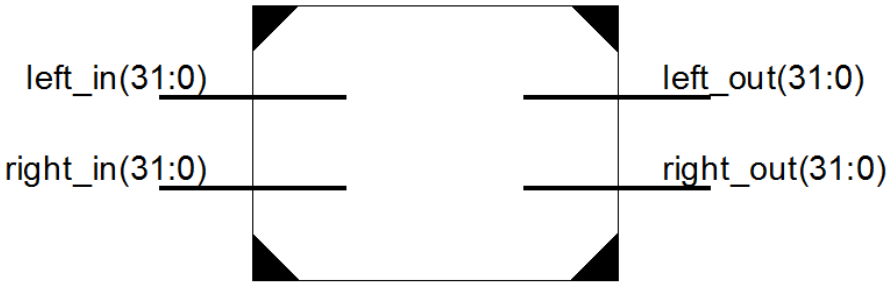
Yazılım tasarımı esnasında Microblaze mikroişlemcisi ile saat işaretli şifreleme bloğu sağlıklı bir şekilde kontrol edilememesinden dolayı bir de saat işaretsiz bir başka deyişle tamamen kombinezonsal olarak şifreleme bloğu tasarlanmıştır. Tamamen kombinezonsal olarak tasarlanan şifreleme bloğu saat çevrimi olmayacağı 32 döngü beklemek zorunda kalmayarak tek seferde şifreleme yapacaktır. Bu da saat işaretli şifreleme bloğuna göre en az 2 kat daha hızlı bir sistemi ifade etmektedir. Kapladığı alan konusunda kaybettiği bu avantajı en az iki kat daha hızlı bir sistem olarak hızdan kazanmaktadır.

Bu şifreleme bloğu tasarımında ilk şifrelemenin bir döngüsü Şekil 5.6'daki gibi tasarlanmıştır. Şifreleme için gerekli olan 128 bit uzunluklu anahtar bu blok içerisinde bir kablo değişkeniyle tutulmaktadır. Bloğun üç girişi ve iki çıkışı mevcuttur. Bunlardan ikisi her bir döngüye girmekte olan verilerin girişidir. Daha önce hesaplanmış olan delta değişkeninin değerleri de üst blokta tutularak her bir döngüye kendi delta değeri bu delta girişi vasıtasıyla ulaştırılacaktır.



Şekil 5.6 : Şifreleme bloğunun bir döngüsü.

Üst blok yani saat işaretsiz şifreleme bloğu oluşturulurken Şekil 5.6'da görülmekte olan bloklardan 32 tanesi bir alt bloğun çıkışı diğerinin girişi olacak şekilde en üst blokta art arda bağlanmıştır. Yine en üst blokta bu 32 adet alt bloğun delta girişlerine daha önce hesaplanıp tutulmuş delta değerleri verilmiştir. En üst bloğun yani şifreleme bloğunun genel yapısı şekil 5.7'de görüldüğü gibidir.

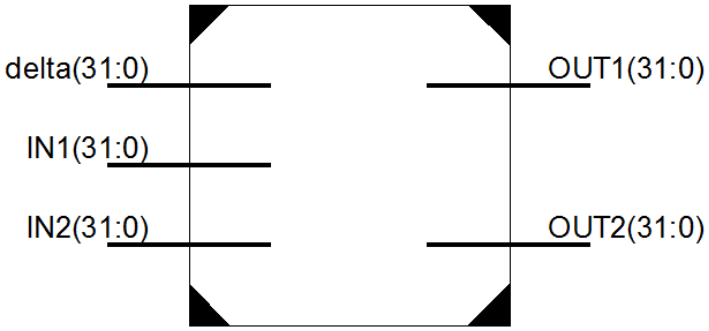


Şekil 5.7 : Saat işaretsiz şifreleme bloğu.

Blok girişine verilen şifresiz veriler saat işareti beklemeksizin hesaplanarak çıkışa verilmektedirler. Şifreleme bloğu FPGA üzerinde toplam 2048 adet dilim kaplamaktadır. Hat gecikmeleri ve mantık bloklarının gecikmelerinin toplamı yaklaşık olarak $290*10^{-9}$ saniyedir. Yani Bloğun girişine şifrelenmek üzere veri girdiğimizde blok bütün işlerini $290*10^{-9}$ saniyede bitirerek şifreli veriyi çıkışa vermektedir. Şifreleme bloğunun hat bağlantıları ve yerleştirme işlemi tamamlandıktan sonraki benzetim sonuçları EK B'da görülmektedir. Girişlerden çıkışlara ne kadar gecikme olduğu EK B'da açıkça görülmektedir.

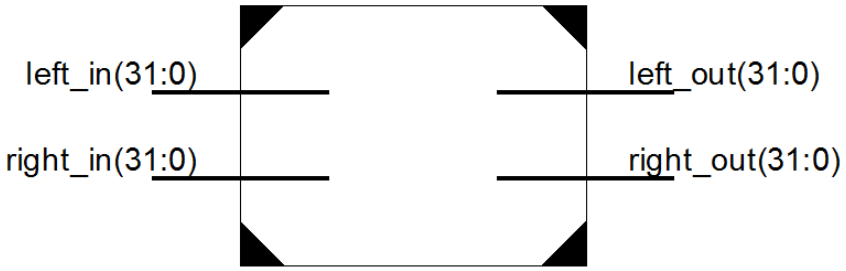
5.1.6. Şifre Çözme Bloğu

Etiket kısmına yerleştirilecek olan şifreleme bloğunun tasarımı tamamlandıktan sonra okuyucu kısmı için de bu şifrelenmiş veriyi çözmek için şifre çözme bloğuna ihtiyaç duyulmaktadır. Bölüm 5.1'de şifre çözme donanımı için tasarım hedefi olarak çok hızlı bir sistem olması belirtilmişti. Bu doğrultuda şifre çözme bloğunun tasarımının saat işaretinde bağımsız tamamen kombinezonsal olmasına karar verilmiştir. Şifre çözme bloğunun tasarımı sırasında alt bloklar olarak toplama, çıkarma, mantıksal kaydırma ve özel veya blokları kullanılmıştır. Şifre çözme bloğu da şifreleme bloğunda olduğu gibi Şekil 4.6'daki turun 32 defa döngüye girmesiyle tamamlanmaktadır. Şifreleme tarafından farklı olarak anahtarların ters dönmesi ve ana kollardaki toplama bloklarının yerlerini çıkarma bloklarının almasıdır. Saat işaretsiz şifreleme bloğu tasarımında da olduğu gibi ilk olarak alt bloklar kullanılarak şifre çözme yapısının tek bir döngüsü tasarlanmıştır. Daha sonra bu tek bir döngünün öncekini çıkışı sonrakinin girişi olacak şekilde 32 adet peş peşe bağlanmıştır. Şekil 5.8'de tek bir turun genel yapısı görülmektedir.



Şekil 5.8 : Şifre çözme bloğunun bir döngüsü.

Bu bloğun içerisinde 4 adet mantıksal kaydırma, 6 adet toplama, 2 adet çıkarma ve 2 adet özel veya bloğu uygun şekillerde ara kablolarla birbirlerine bağlanmışlardır. Anahtar değeri bu blok içerisinde tutulmuştur. Okuyucu tasarımının tamamlandığı en üst bloğun içerisinde ise bu bloklardan 32 tanesi yan yana üretilerek uygun şekilde bağlama işlemleri yapılmıştır. En üst bloğun genel yapısı Şekil 5.9'da görülmektedir. Delta değişkeninin değerleri ise bir dizi yardımıyla en üst blokta tutularak her bir alt bloğa kendi kullanacağı delta değerleri Şekil 5.8'deki delta girişinden verilmektedir.



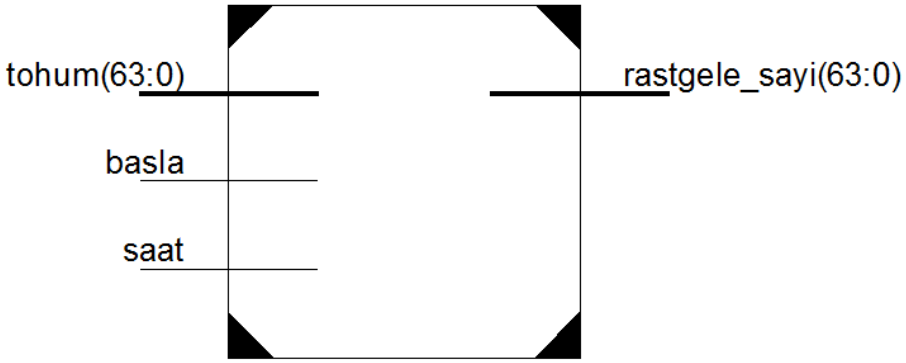
Şekil 5.9 : Şifre çözme bloğu.

Şifre çözme bloğunun FPGA üzerine yüklenmesi benzetim sonuçları EK C'de görülmektedir.

Şifreleme bloğu için benzetim sonuçlarında girişlere $(00000000)_{16}$ değeri verildiğinde çıkışlarda $(9327C497)_{16}$ ve $(31B08BBE)_{16}$ şifreli metinleri elde edilmişti. Şifre çözme bloğundan beklediğimiz ise bu şifreleri verileri girişlere verdiğimiz zaman çıkışlardan şifresi çözülmüş verileri yani $(00000000)_{16}$ elde etmemiz gerekmektedir. EK C'de de görüldüğü gibi yapılan şifre çözme bloğu tasarımı tamamen doğru olarak çalışmaktadır. Şifre çözme donanımı FPGA üzerinde 2048 dilim kaplamaktadır. Aynı zamanda şifre çözme bloğu $296*10^{-9}$ saniye gibi çok kısa bir sürede şifre çözme işlemini tamamlamaktadır. Buradan da görüleceği üzere tasarımın başında hedeflenen okuyucu kısmı için çok hızlı şifre çözme donanımı tasarımı başarıyla gerçekleştirilmiştir.

5.1.7. Rastgele Sayı Üreteci Bloğu

Bölüm 4.2'de belirtildiği üzere doğrulama mekanizması içerisinde okuyucu etiketin kimliğini tanımlama yapabilmesi için rastgele sayı üreterek etikete göndermesi gereklidir. Bundan dolayı okuyucu kısmı için rastgele sayı üreteci bloğu tasarlanmıştır. Tasarımın genel yapısı Şekil 5.10'da görülmektedir. Rastgele sayı üretecinin çalışma yapısı, basla girişi alçak seviyede iken bloğa 64 bit uzunluklu bir tohum değeri girilmektedir. Tohum değeri girildikten sonra her saat darbesinde bitler bir yüksek anlamlı bite atanır. Bunun sonucunda en sağ tarafta kalan en anlamsız bite ise 64. 62. 51. ve 50. bitlerin 'özel veya değil' işlemine sokulmasıyla elde edilen bit atanır. Tüm bu işlemler sonucunda rastgele sayı üreteci bloğu her saat darbesinde okuyucu tarafından kullanılmak amacıyla çıkışından 64 bit uzunluklu sözde-rastgele sayı üretir.



Şekil 5.10 : Rastgele sayı üreteci bloğu.

5.2. Yazılım Tasarımı

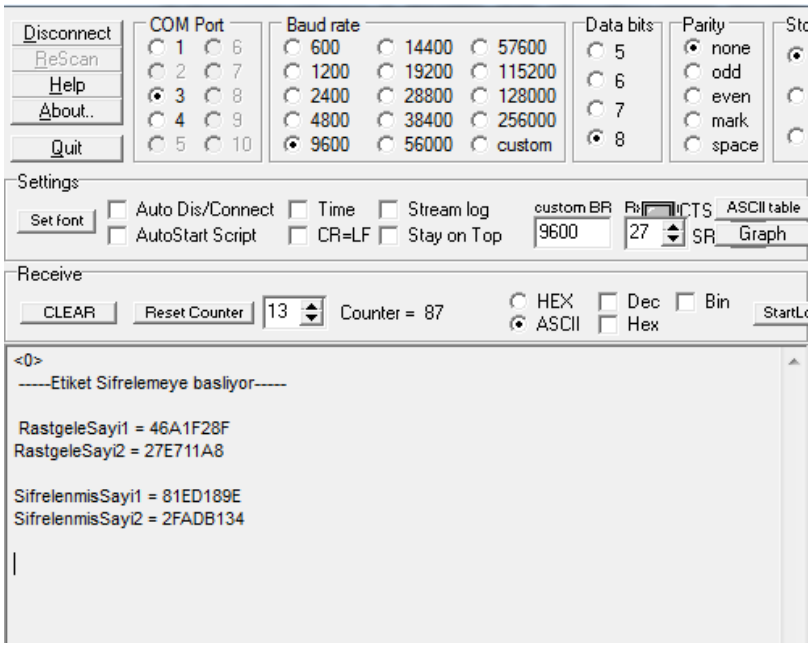
Etiket ve okuyucu tasarımı için gerekli olan şifreleme, şifre çözme ve rastgele sayı üreteci donanımları tasarımları yapıldıktan sonra bu donanımları ve tüm sistemi kontrol için FPGA üzerindeki Microblaze mikroişlemcisi kullanılmıştır. Etiket ve okuyucu kısımları için iki ayrı FPGA kartı kullanılmıştır. Mikroişlemci merkezli sistem tasarımı yapılırken öncelikle ISE programı aracılığıyla Microblaze temelli bir proje oluşturulmuştur. Daha sonra EDK programıyla donanımlar ile Microblaze arasındaki bağlantılar yapılmıştır. Ayrıca proje için gerekli olan seri Evrensel Eşzamansız Alıcı/Verici (Universal Asynchronous Receiver/Transmitter, UART) IP'leri de Microblaze ile yönetilebilir hale getirilmiştir. Son olarak tüm donanım

bağlantıları tamamlandıktan sonra SDK programında yazılım tasarımı yapılarak proje tamamlanmıştır.

5.2.1. Etiket Kısmı

Etiket kısmı için gereken donanım parçası şifreleme bloğu olduğu için etiket kısmındaki Microblaze mikroişlemcisi şifreleme bloğuna erişmeli, onu kontrol etmelidir. Bu tasarım hedefi doğrultusunda projeye başlarken ISE programından projeye Microblaze mikroişlemcisi eklenmiştir. Projenin donanım Microblaze bağlantılarının yapmak üzere EDK programına geçilmiştir. EDK programı yardımıyla tasarımın gerçekleneceği kit seçilerek, sistemin saat işareti frekansı 50 Megahertz olarak belirlenmiştir. Ayrıca Microblaze'in hafızası 32 Kilobayt olarak seçilmiştir. Proje oluşturulduktan sonra şifreleme donanımı için kullanıcı IP'si oluşturur. Kullanıcı IP'si oluşturulurken Microblaze ile şifreleme donanımı arasındaki veri alış verişi sağlamak amacıyla 16 adet 32 bit uzunluklu kaydediciler kullanılmıştır. IP oluşturulduktan sonra şifreleme kısmı için yazılan donanım bloğunun en üst blok kodu EDK'nın kullanıcı IP'si için oluşturmuş olduğu en üst verilog kodunda çağrılmıştır. Kullanıcı IP'si için Verilog koduna gerekli eklemeler yapıldıktan sonra bu eklemelerin güncellenmesi için kullanıcı IP'si yeniden projeye eklenmiştir. IP'si oluşturulan şifreleme donanımının Microblaze ile olan bağlantısı PLB (Processor Local Bus) ile sağlanmıştır. Tüm bağlantıları da yapılan şifreleme donanımı için son olarak Microblaze hafızasında otomatik olarak adresi oluşturulmuştur. Microblaze ile şifreleme donanımı tamamen birleştirildikten sonra yazılım tasarımına geçmek üzere EDK projesi ISE aracılığıyla sentez ve gerçekleme aşamalarına girdirilmektedir. Bu aşamalar da tamamlandıktan sonra Microblaze'li donanım projesi yazılım tasarımı yapılmak üzere SDK programına gönderilir. Microblaze mikroişlemcisinin kontrol mekanizması SDK ara yüzünde C dili ile programlanmıştır. Yazılım aşamasında şifreleme donanımına ulaşmak için kullanıcı IP'sinin üst bloğunda Microblaze ile iletişim oluşturmak için aralarına koyulan kaydedicilerin adresleri kullanılmaktadır. Okuyucu tarafından etikete gelen rastgele sayının etiket tarafından şifrelenebilmesi için alınan sayı, şifreleme donanımının girişine bağlanan kaydediciye gönderilerek, çıkışına bağlanan kaydediciden de şifrelenmiş olan sayı okunmuştur. Oluşturulan sistemin FPGA üzerinde gerçekleme aşamalarını görmek için FPGA seri port aracılığıyla bilgisayara bağlanarak, FPGA'nın üzerinde akmakta olan işlemler bilgisayar ekranında gözlemlenmiştir.

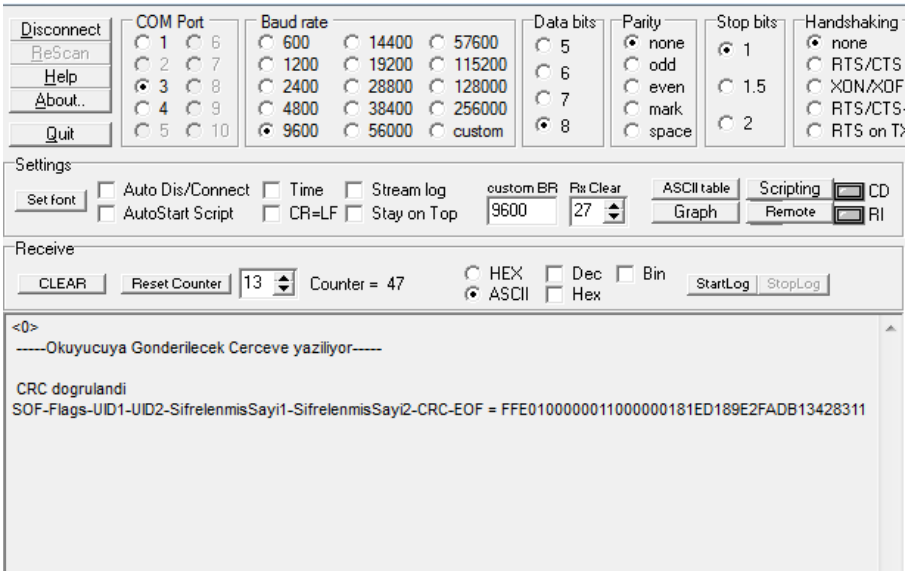
Okuyucu tarafından gönderilen ve etiket tarafından alınıp şifrelenen verinin FPGA üzerinde gerçekleştirilmesi Şekil 5.11'de görülmektedir.



Şekil 5.11 : Etiketin FPGA üzerinde gerçeklenmesine ilişkin ekran çıktısı.

Etiket tarafından oluşturulan şifrelenmiş veri okuyucu kısmına gönderilmesi için Bölüm 4.2'de belirtilen şekilde cevap çerçevesi içerisinde gönderilmelidir. Çerçeve içerisindeki SOF değeri $(FF)_{16}$, Bayraklar değeri $(E0)_{16}$, UID değeri $(1000000110000001)_{16}$ ve EOF değeri ise $(11)_{16}$ olarak belirlenmiştir. Okuyucu kısmına gönderilecek verinin hatalı olup olmadığını kontrol etmek amacıyla çerçeveye eklenmesi gereken CRC değeri yazılım içerisinde hesaplanarak $(0283)_{16}$ olarak hesaplanarak çerçeveye eklenmiştir. Okuyucuya gönderilen veri çerçevesinin değerleri Şekil 5.12'de gösterilmektedir.

Protokolün başlangıcında etiket Microblaze içinde sakladığı sayıcı değerini, şifrelenmiş sayıyı gönderdiği gibi aynı çerçevede göndermektedir. Etiketin göndermiş olduğu sayıcıyı alıp kendi sayıcısı ile kontrol eden okuyucu etikete bu rastgele sayısını göndermektedir. Daha sonra Şekil 5.12'de ki çerçeve okuyucuya gönderilerek etiket görevini tamamlamıştır.



Şekil 5.12 : Okuyucuya gönderilen çerçeveye ilişkin ekran çıktısı.

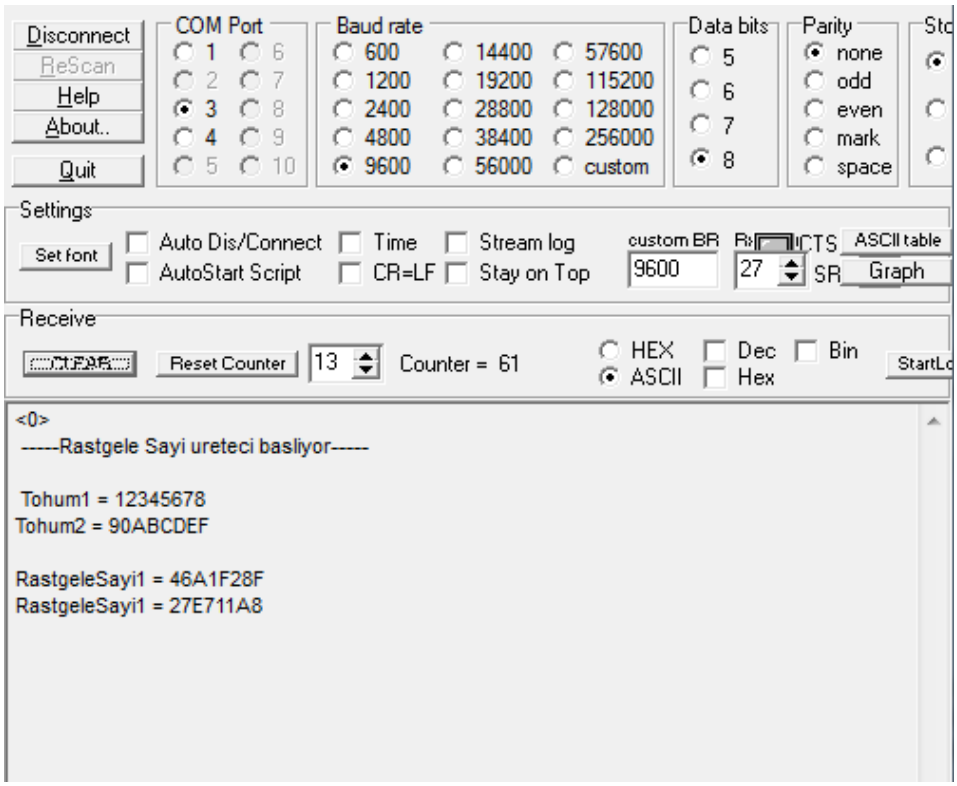
5.2.2. Okuyucu Kısmı

Okuyucu kısmı için tasarlanan şifre çözme ve rastgele sayı üreteci donanımları Microblaze mikroişlemcisi ile yönetilmesi gerekmektedir. Bu doğrultuda okuyucu kısmını yönetmek için ISE programı kullanılarak Microblaze temelli sistem oluşturuldu. Etiket kısmında olduğu gibi sistem birimlerini ve bağlantılarını oluşturulan için EDK programı kullanılmıştır. EDK programında, daha önce tasarlanmış olan rastgele sayı üreteci ve şifre çözme bloklarının Microblaze ile bağlantısını yapmak için kullanıcı IP'si oluşturulmuştur. İki donanım bloğu aynı IP içerisinde yönetilmiştir. Bu iki donanım bloğunu yönetmek için Microblaze ile donanımlar arasına IP içerisinde donanımlara erişmek amacıyla 32 bit uzunluklu kaydediciler konulmuştur. Donanımlara erişmek için kullanılacak olan kaydedici sayısı donanımları giriş ve çıkışlarının toplamı kadar olmalıdır. Rastgele sayı üreteci donanımına erişmek için 5 adet ve şifre çözme bloğuna erişmek için ise 4 adet kaydedici kullanılmıştır. IP'si oluşturulan kullanıcı donanımlarının Microblaze'in veri yoluna bağlanması PLB ile sağlanmıştır. Bu haliyle Microblaze rastgele sayı üretecine ve şifre çözme bloğuna kendi komutlarıyla ulaşabilir hale getirilmiştir.

Sisteme eklenen Microblaze ile donanım tasarımı tamamlanan okuyucunun yazılım tasarımını yapmak için ISE aracılığıyla sentezleme ve gerçekleme işlemleri yapılarak SDK programına geçilmiştir. Okuyucu kısmının yazılım tasarımı da C dili ile yapılmıştır. Okuyucunun görevi doğrultusunda öncelikli olarak etiketten aldığı sayıcı bilgisini kontrol eder, eğer bu veri kendi içerisinde tutmuş olduğu sayıcı ile aynı

değilse protokol gerçeklenmeden etiketle iletişimi kesmektedir. Eğer sayıcı bilgisi doğru ise protokolün ikinci aşaması olan rastgele sayı üretecine tohum değeri gönderir ve oradan bir rastgele sayı alarak uygun çerçeve ile etiket kısmına göndermektedir. Daha sonra etiket kısmından gelen şifrelenmiş sayının şifresini çözmek üzere şifre çözme donanımına gönderir. Bunun sonucunda şifresi çözülen sayıyı göndermiş olduğu rastgele sayı ile karşılaştırarak eğer doğruysa protokolü başarılı bir şekilde tamamlar.

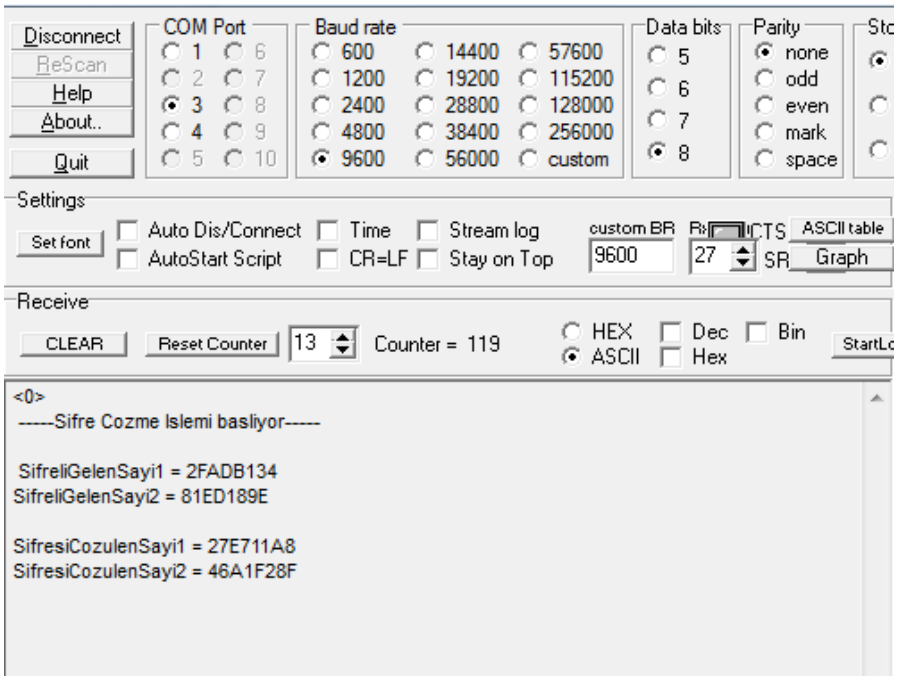
Microblaze'in rastgele sayı üretecine göndermiş olduğu tohum değeri ve bunun sonucunda elde ettiği rastgele sayıya ilişkin FPGA üzerinde gerçekleme görüntüleri Şekil 5.13'de görülmektedir. Bu değerin Şekil 5.11'de görülen etikete giden veri olduğu bilinmektedir.



Şekil 5.13: Rastgele sayı üretecinin Microblaze ile FPGA üzerinde gerçeklenmesine ilişkin ekran çıktısı.

Rastgele Sayının üretilip aynı sayının etiket tarafından şifrelenmesi Şekil 5.11'de görülmüştü. Etiket tarafından şifrelenen verilerin okuyucu tarafına gönderilecek şifresinin çözülmesine ilişkin ekran görüntüsü Şekil 5.14'te görülmektedir. Şifresi çözülmüş olan rastgele sayı ile okuyucu tarafından üretilip etikete gönderilmiş olan rastgele sayının eşit olduğu Şekil 5.13 ve Şekil 5.14 karşılaştırılarak çok net bir

biçimde görülmektedir. Bu iki verinin aynı olması okuyucunun ve etiketin protokolü başarılı bir biçimde tamamladığını göstermektedir.



Şekil 5.14 : Okuyucuya ait şifre çözme bloğunun FPGA üzerinde gerçeklenmesine ilişkin ekran çıktısı.

6. SONUÇLAR

Bu bitirme çalışmasında donanım ve yazılım içeren güvenli bir RFID sistem tasarlanmıştır. Çalışmada bahsedilen protokol ilk defa FPGA üzerinde gerçeklenerek donanım ve yazılım ortak çalışması gösterilmiştir. Etiket ve okuyucu kısımları tasarımlarının sorunsuz şekilde çalışır halde olduğu UART aracılığıyla bilgisayarla haberleştirilerek kanıtlanmıştır. Protokolde kimlik doğrulama mekanizmasını sağlamak üzere şifreleme algoritması olarak ilk defa TEA kullanılmıştır. TEA gerek yer bakımından gerek hız bakımından daha önce gerçeklenmiş olan algoritmalara kıyasla, RFID sistemlere daha verimli çalışma imkanı sağlamıştır. TEA şifreleme algoritmasının saat işaretli olarak FPGA üzerinde sadece 175 dilim işgal edilerek tasarlanması başarılmıştır. Ayrıca okuyucu kısmı için tasarlanan şifre çözme bloğunun $290*10^{-9}$ saniye gibi kısa bir sürede şifre çözme işlemini yapması sağlanmıştır.

Çalışma neticesinde TEA şifreleme algoritması kullanılarak daha hızlı ve daha az yer kaplayan RFID sistemlerin tasarlanması mümkün hale getirilmiştir. Gelecek çalışmalarda daha gelişmiş mikroişlemciler ve daha efektif şifre algoritmaları kullanılarak daha verimli sistemlerin tasarlanabileceği tespit edilmiştir.

KAYNAKLAR

[1] Finkenzeller, K., 2003. RFID-Handbook., , pp. 1, John Wiley & Sons, Ltd., second edition.

[2] Feldhofer, M., 2004. "An authentication protocol in a security layer for RFID smart tags," Electrotechnical Conference, 2004. MELECON 2004. Proceedings of the 12th IEEE Mediterranean, **2**, pp. 759- 762.

[3] Kavas, A., 2007. Radyo Frekans Tanımlama Sistemleri, **430,** s. 74-80.

[4] Lehpamer, H., 2008. RFID Design Principles, pp. 178, Artech house.

[5] ISO/IEC 18000-3, 2003. Information Technology AIDC Techniques - RFID for Item Management, International Organization for Standardization.

[6] Chu, Pong P., 2008. FPGA Prototyping by VHDL Examples. Wiley-Interscience, New Jersay.

[7] Xilinx, 2006. Spartan-3E Starter Kit Board User Guide.

[8] Xilinx, 2007. MicroBlaze Processor Reference Guide.

[9] Xilinx, 2007. Embedded System Tools Reference Manual.

[10] Xilinx, Software Development Kit Help Contents, http://www.xilinx.com/support/documentation/sw_manuals/xilinx12_2/SDK_Doc/index.html.

[11] Xilinx, 2011. EDK Concepts, Tools and Techniques.

[12] Abdelhalim, M.B., Elhennawy, A., Ayyad, M.and El-Mahallawy, M., 2011. Implementation of a Modified Lightweight Cryptographic TEA Algorithm in RFID System, VI. International Conference on Internet Technology on Secured Transactions, Abu Dhabi, 11-14 December, pp. 509-513.

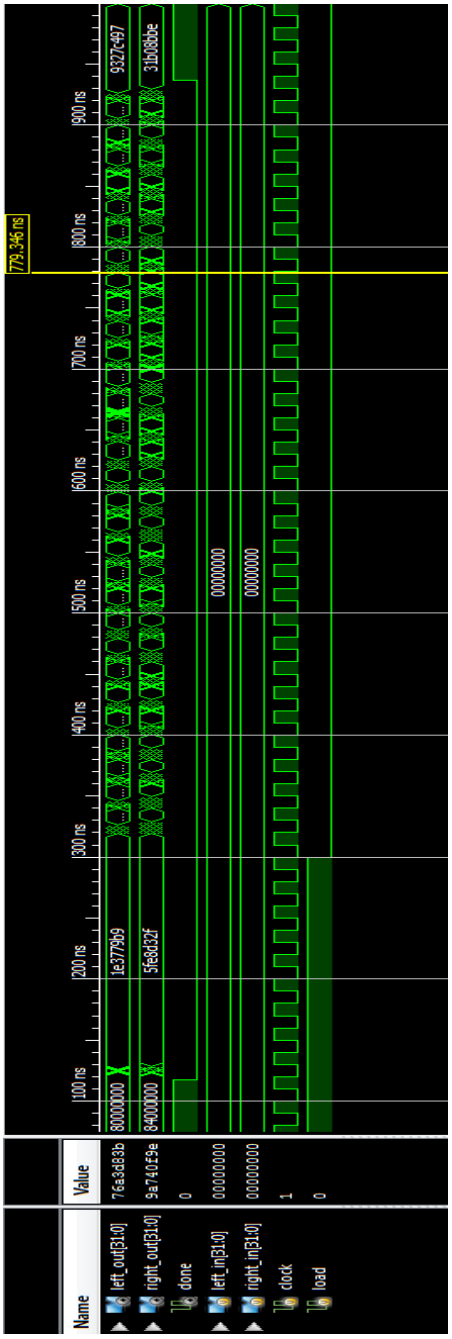
[13] Andem, V.R., 2003. A Cryptanalysis of the Tiny Encryption Algorithm, MSc. Thesis, The University of Alabama, ALABAMA.

[14] Menezes, A.J., Van Oorschot, P.C. and Vanstone, S.A., 1997. Handbook of Applied Cryptography, pp. 195-198, CRC Press.

[15] Koca, H., 2007. "Robot Manipülatör Denetimi", *Yüksek Lisans Tezi*, Gazi Üniversitesi Fen Bilimleri Enstitüsü, Ankara.

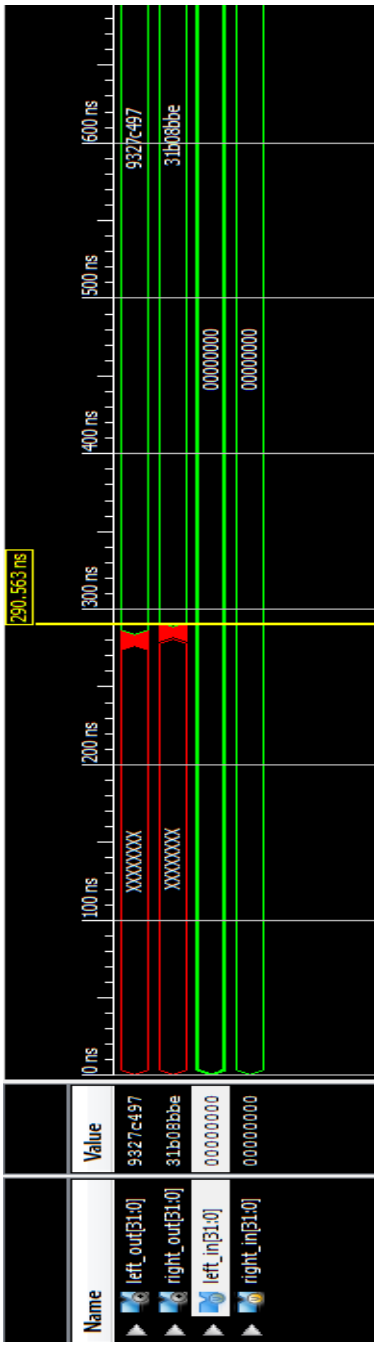
EK A

Saat İşaretli Şifreleme Bloğu Benzetim Sonuçları



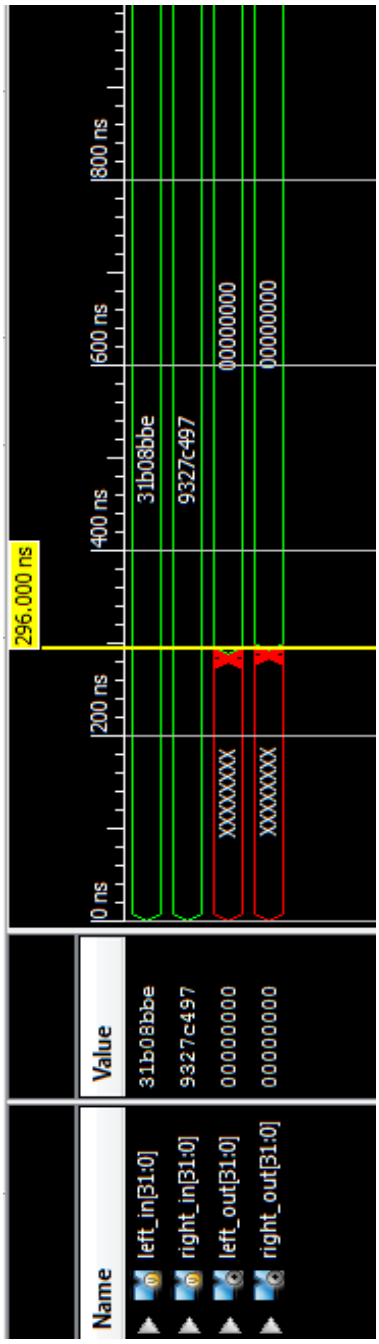
EK B

Saat İşaretsiz Şifreleme Bloğu Benzetim Sonuçları



EK C

Saat İşaretsiz Şifre Çözme Bloğu Benzetim Sonuçları



ÖZGEÇMİŞ

Adı Soyadı: Semih Alparslan

Doğum Yeri ve Tarihi: Tokat, 1989

Lise: Tokat Fen Lisesi; 2003-2007

Lisans: İstanbul Teknik Üniversitesi, Elektronik Mühendisliği; 2007-2012