**İSTANBUL TEKNİK ÜNİVERSİTESİ**

**ELEKTRİK-ELEKTRONİK FAKÜLTESİ**

# GÜVENLİ BİR YAKIN ALAN HABERLEŞME SİSTEMİNİN FPGA ÜZERİNDE GERÇEKLENMESİ

**BİTİRME ÖDEVİ**

**AHMET ÇAĞRI BAĞBABA**

**040080533**

**Bölümü: Elektronik ve Haberleşme Mühendisliği Bölümü**

**Programı: Elektronik Mühendisliği**

**Danışmanı: Doç. Dr. Sıddıka Berna ÖRS YALÇIN**

**MAYIS 2013**

## ÖNSÖZ

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

Bu çalışma sırasında sabırla ve her an yardımda bulunan Yük. Müh. Ahmet Arış ve değerli arkadaşım Ahmet Turan Erozan'a teşekkür ederim.

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

MAYIS 2013 Ahmet Çağrı BAĞBABA

**İÇİNDEKİLER**

## KISALTMALAR

**NFC** : Near Field Communication
**RFID** : Radio Frequency Identification
**FPGA** : Field Programmable Gate Array
**LUT** : Look-up Table
**EDK** : Embedded Development Kit
**ISE** : Integrated Synthesis Environment
**XPS** : Xilinx Platform Studio
**TEA** : Tiny Encryption Algorithm
**XOR** : Exclusive Or
**IEEE** : The Institute of Electrical and Electronics Engineers
**IBS** : Identity Based Signature

## ŞEKİL LİSTESİ

## GÜVENLİ BİR YAKIN ALAN HABERLEŞME SİSTEMİNİN FPGA ÜZERİNDE GERÇEKLENMESİ

## ÖZET

Günümüzde Yakın Alan Haberleşme Sistemi (Near Field Communication, NFC) uygulamalarının kullanımı hızla yaygınlaşmaktadır. NFC, yeni nesil bir kablosuz iletişim teknolojisidir. NFC teknolojisi temelde, NFC standartlarına uyumlu elektronik cihazlar arasında yakın mesafeli haberleşmeyi sağlar. Bu sistemlerin kullanımının yaygınlaşması güvenlik sorununu da beraberinde getirmektedir. Birçok uygulama sahasına girmiş bu sistemlerin güvenlik açıklarına sahip olması ve bu güvenlik açıklarının kötü amaçlı insanlar tarafından kullanılması ciddi bir problem teşkil etmektedir. Bu nedenlerden dolayı NFC sistemlerinin güvenli şartlar altında kullanılması zorunluluk haline gelmiştir. Bu güvenli şartları sağlamak için NFC uygulamalarında kriptoloji algoritmalarının kullanılması en akılcı çözümdür.

Bu projede güvenli bir NFC sistemi oluşturabilmek için bir kimlik doğrulama protokolü gerçeklenmiştir. Bu protokolün nasıl gerçeklenişini daha iyi kavrayabilmek adına önbilgiler olarak öncelikle RFID sistemlerinden ve gerçekleme sırasında kullanılan tasarım araçları tanıtılmıştır. Son olarak, güvenli NFC sistemi tasarlamak için kullanılan protokol anlatılmıştır. Protokol gerçeklenirken gömülü sistemlere olan uygunluğu sebebiyle şifreleme algoritması olarak Küçük Şifreleme Algoritması (Tiny Encryption Algorithm, TEA) kullanılmıştır.

Protokolü gerçekleme aşamasında, Verilog donanım tasarlama dilinde şifreleme bloğu, kaydedici ve özel veya bloğu, VHDL dilinde ise Montgomery Modüler Çarpıcısı kullanılmıştır. 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 NFC sistemi Sahada Programlanabilir Kapı Dizinleri (Field Programmable Gate Array, FPGA) üzerinde gerçeklenmiştir.

## IMPLEMENTATION OF A SECURE NEAR FIELD COMMUNICATION SYSTEM ON FPGA


## SUMMARY

Nowadays, using of Near Field Communication System applications become rapidly widespread. NFC is a new generation wireless communication technology. NFC technology fundamentally provides near field communication between electronic devices which are compatible with NFC standarts. Becoming widespread of these systems brings with security problems. Having security vulnerabilities of these systems which entered many applications, and using of these security vulnerabilities by malicious people constitutes a serious problem. Because of these reasons, the using of NFC systems under secure conditions has became necessity. The most rational solution to ensure secure conditions for NFC application is the use of cryptology algorithms.

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

In the implementation section, encryption, register and exclusive or blocks are used in Verilog hardware description language and montgomery modular multiplication block is used in VHDL hardware description language. After testing of the hardware is fully operational, in order to control whole system and blocks the Microblaze microprocessor software design has been made on the C language. Finally, the secure NFC System with hardware and software design is implemented on FPGA.

# 1. GİRİŞ

Son yıllarda hızla gelişen kablosuz iletişim yöntemleri birçok alanın vazgeçilmezleri olmuşlardır. Bu amaçla geliştirilen teknolojilerden olan NFC ve RFID sayesinde, ödeme ve toplu taşıma biletleri gibi temassız işlemler, mobil cihazlar arası veri aktarımı gibi günlük hayatta ihtiyaç duyulan işlemler kolaylıkla yapılabilmektedir [1]. Bu uygulamaların gelişimi güvenlik sorunlarını da beraberinde getirmektedir. Veri iletişiminde kötü niyetli üçüncü kişilerin bilgiyi elde edememesi için çeşitli kriptografi yöntemleri ve algoritmalar kullanılabilmektedir. NFC için ise bu yöntemlerin uygulamalarına dijital imza adı verilmektedir.

Bu çalışma kapsamında birer RFID etiketi olan, "etiket" ve "okuyucu" adı verilen iki etiket arasındaki veri aktarımının dijital imza yoluyla güvenli bir şekilde sağlanması hedeflenmiştir. Dijital imza yöntemi olarak "Identity-Based Signature" seçilmiştir [2]. Buradaki doğrulama mekanizması ile ilgili temel hedef, gönderilen ve alınan mesajın karşılıklı olarak doğrulama işlemine tabi tutulmasıdır. Bu işlem yapılırken mesajın özetinin alınması ya da şifrelenmesi, gizli ve açık anahtarların kullanılması ve üs alma kullanılmıştır. Şifrelenen mesaj ve gizli anahtarla oluşturulan imzanın karşı etikete yollanması, okuyucu etikette ise mesajın şifrelenmesi ve imza ile birlikte açık anahtar kullanılarak doğrulamanın yapılması hedeflenmiştir.

Çalışma kapsamında verinin şifrelenmesi işlemi Küçük Şifreleme Algoritması (Tiny Encryption Algorithm, TEA) ile sağlanmıştır. Üs alma yöntemi olarak ise Montgomery Exponentiation kullanılmıştır. Giriş biti sayısı belirsiz olan mesajın gerekli şifreleme işleminin yapılması bir kaydedici ve özel veya bloğuyla sağlanmıştır. Bu doğrultuda şifreleme ve Montgomery Exponentiation FPGA üzerinde donanımsal olarak kullanılmıştır. Bu donanımları ve tüm sistemi kontrol etmek amacıyla FPGA'nın içerisinde bulunan Microblaze yazılımsal mikroişlemcisi kullanılmıştır.

Tezin ikinci bölümünde önbilgiler olarak NFC'nin temel yapısı olan radyo frekansı ile tanımlama ve donanım- yazılım tasarımı sırasında kullanılan araçlar anlatılmıştır. Tezin üçüncü bölümünde gerçeklenecek olan kimlik doğrulama protokolünden ve

kullanılan algoritmalardan bahsedildikten sonra, dördüncü ve beşinci bölümlerde bu protokolü gerçekleme aşamaları tüm detaylarıyla anlatılmıştır.

# 2. ÖNBİLGİLER

## 2.1. Radyo Frekansı ile Tanımlama Sistemleri

NFC teknolojisinin altında yatan teknoloji RFID'dir. İki teknoloji arasındaki en temel fark iletişim mesafesidir.

Durağan ve ya hareket halindeki nesnelerin tekil ve otomatik olarak tanımlanması işleminde Radyo Frekansı ile Tanımlama Sistemleri yaygın olarak kullanılır. Eskiden beri klasik olarak kullanılan barkot sistemleri kolay kullanıma sahip olmalarına karşın, saklayabildikleri veri miktarlarının az olması ve barkot üzerindeki etiketin değerinin 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 [3].

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 [4].

RFID sistemleri temel olarak, etiket ve okuyucu yapılarından oluşmaktadır. Etiket, tanımlanmak istenen nesnenin üzerine yerleştirilir, okuyucu ise anteni yardımıyla bu etiket ile haberleşmeyi sağlar.

RFID etiketi, radyo frekansını kullanarak okuyucudan gelen sinyalleri alan, sorgulayan - 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 [4].

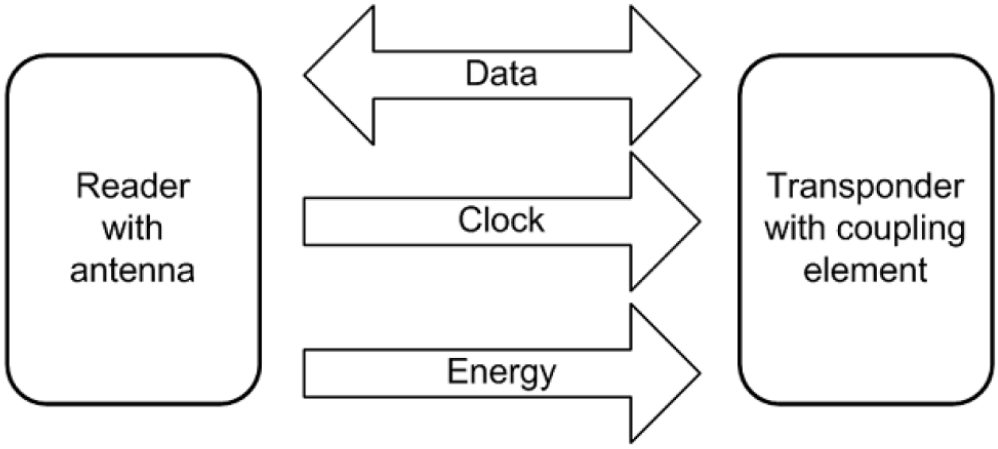
Aktif etiketler kendi yapılarında bulunan güç kaynağı sayesinde devrelerinin çalışmasını ve haberleşme için sinyal üretimlerini sağlarlar. Kendi içinde

barındırdıkları piller yani güç kaynakları sayesinde daha uzak haberleşme mesafeleri ve daha iyi çalışma performanslarına sahiptirler.

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

Pasif etiketler ise, üzerlerinde güç kaynağı barındırmazlar. Okuyucudan aldıkları elektromanyetik dalga yardımıyla devrelerini ve haberleşmesini beslemektedirler. Güç kaynakları içermemeleri daha kısa mesafeli haberleşmeler için kullanılmasına neden olur. Ucuz ve basit yapıda olduklarından tercih edilirler. Bu sebeple güç kaynağının uygulanamadığı, pil ömrünün daha öncelikli olduğu ve işlem kapasitesinin ikinci planda olduğu alanlarda kullanılırlar.

Bir RFID sistemde okuyucunun görevi, antenini kullanarak etiketleri uyarmak, etiketlerin içerdiği veriyi okumak ve bir ağ aracılığıyla bu veriyi bir sunucu bilgisayara göndermektir [5]. Okuyucular kullanım alanlarına göre mobil ya da sabitlenmiş olabilirler. RFID sistemlerde okuyucular aynı zamanda veri tabanlarına 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 veri tabanına göndererek veri tabanında 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 [3].



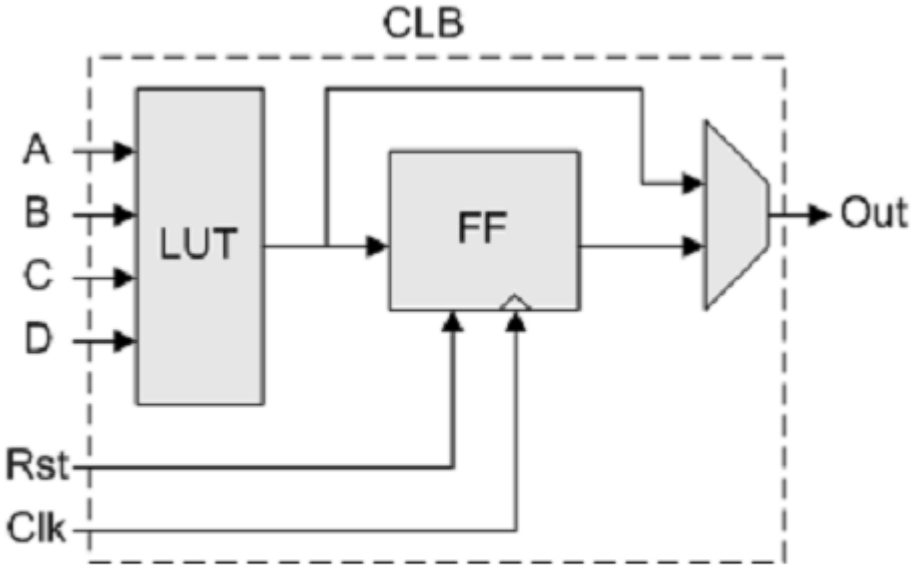

**Şekil 2.1 :** Etiket ve okuyucu arasındaki hatlar [3].

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 [6]. 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.

## 2.2. Sahada Programlanabilir Kapı Dizileri

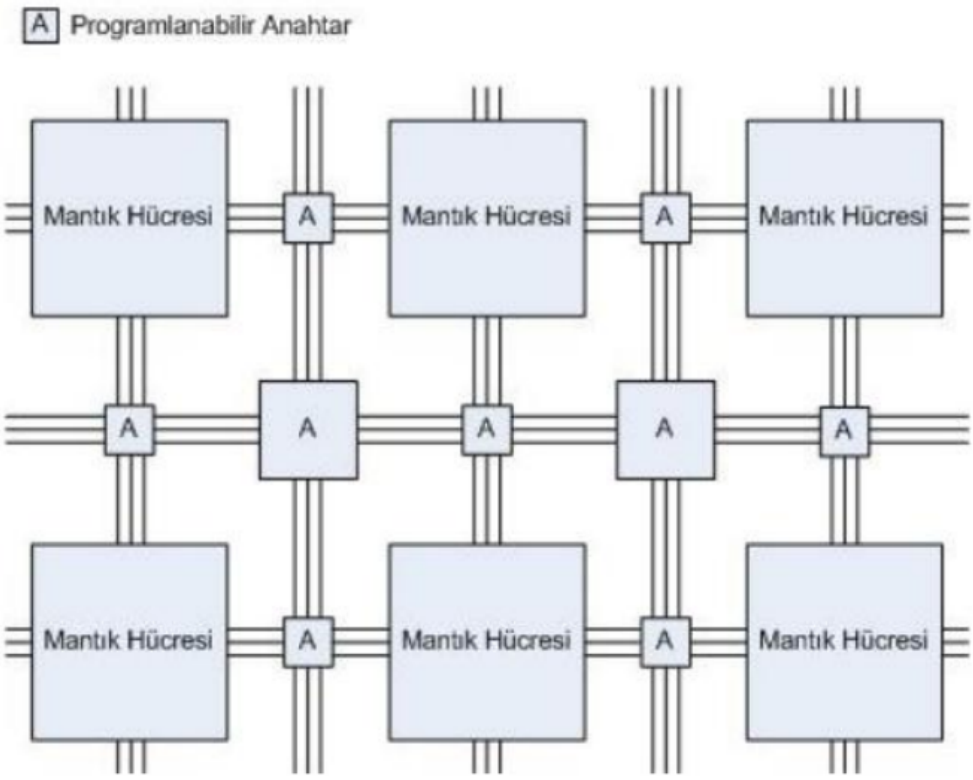
Sahada Programlanabilir Kapı Dizinleri herhangi bir sayısal fonksiyonu gerçekleştirebilmek için kullanıcı tarafından programlanabilen tümleşik devrelerdir [7]. FPGA yönetilebilir anahtarların ve programlanabilir mantık hücrelerinin iki boyutlu olarak dizilmesiyle oluşturulur. Mantık hücreleri basit bir 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 [8].

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




**Şekil 2.2 :** Mantık Hücresi Yapısı [8].

Programlanabilir ara bağlantılardan ve içyapısı Şekil 2.2"de gösterilen mantık hücrelerinden oluşan FPGA'nın genel yapısı Şekil 2.3"de gösterilmektedir.



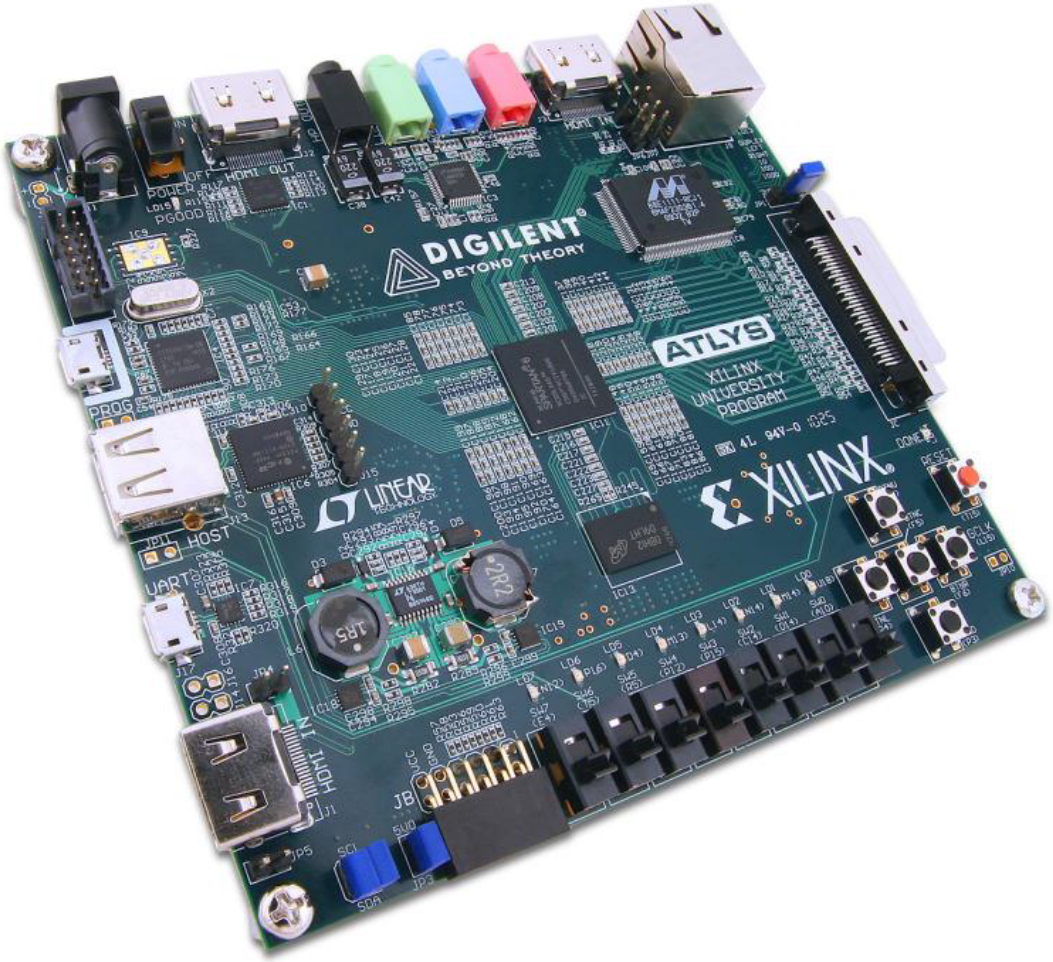

**Şekil 2.3 :** FPGA'nın İç Yapısı [8].

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 karmaşık kombinezonsal fonksiyonlar gerçekleştirilmesine imkân tanır [9].

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.

### 2.3. Xilinx Spartan-6 LX45 FPGA

Çalışmada kullanılan kart üzerinde Xilinx firmasının Spartan-6 LX45 FPGA'yi bulunmaktadır. Bu FPGA, her biri 4 adet 6 girişli LUT ve 8 adet flip flop içeren 6822 dilim, 58 DSP dilimi içermektedir. Ayrıca 500 MHz'e kadar saat hızı sunmaktadır [10].




**Şekil 2.4 :** Kullanılan FPGA Kartı [10].

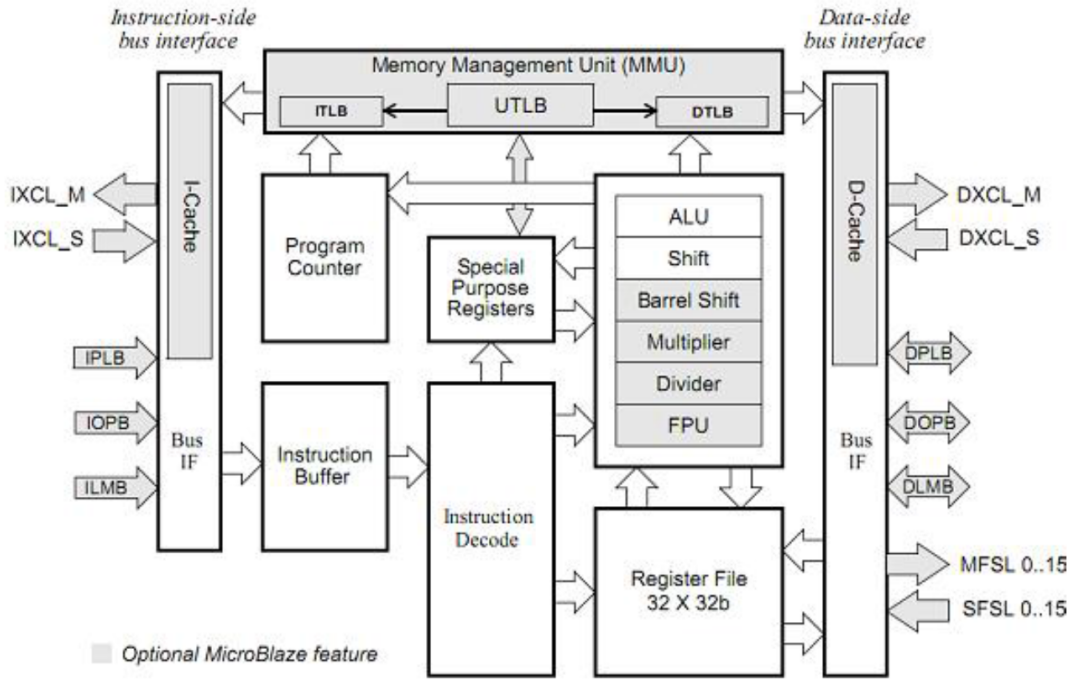
## 2.4. Microblaze İşlemcisi

Mimari yapısı Şekil 2.5'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 [11].

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 [11].




**Şekil 2.5 :** Microblaze Mimarisi [11].

## 2.5. Verilog Donanım Tanımlama Dili

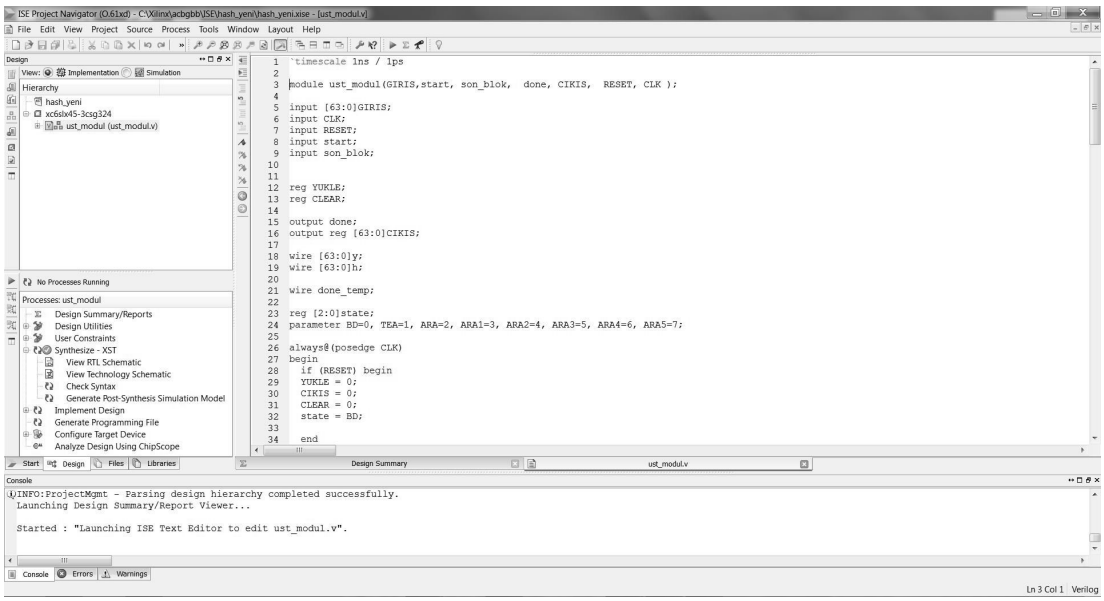
Verilog donanım tanımlama dili 1984-1985 yıllarında Philip Morby tarafından sayısal devreleri modelleme, test etme ve analiz etme amacıyla kolay, basit ve etkili bir şekilde ifade etmeyi hedefleyen bir donanım tanımlama dili olarak geliştirilmiştir. Yapısal olarak C dili ile olan yakınlığı nedeniyle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

## 2.6. VHDL Donanım Tanımlama Dili

Açılımı Very High Speed Integrated Circuit Hardware Description Language olan VHDL en çok kullanılan donanım tasarlama dillerinden biridir. Bu programlama dili 1980'lerden beri kullanılmakta olup sürekli geliştirilmiş ve IEEE tarafından da standart olarak kabul edilmiştir. Tasarımların hiyerarşili bir şekilde bileşenlerine ayrılabilmesi, her bir tasarım elemanının iyi tanımlı bir ara yüze ve hatasız davranış tanımlamasına sahip olması gibi özellikleri sayesinde VHDL oldukça yaygın olarak sayısal tasarımlarda kullanılmaktadır.

## 2.7. Xilinx 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 Verilog, VHDL gibi 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.

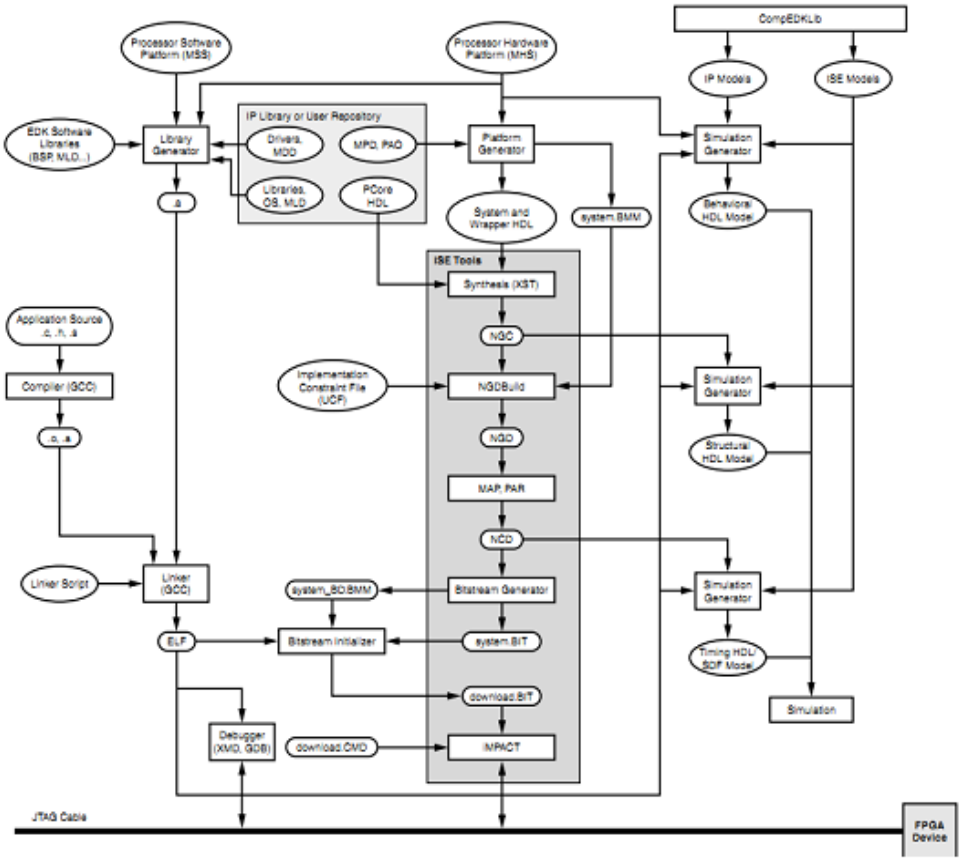



**Şekil 2.6 :** Xilinx ISE Programının Görüntüsü.

## 2.8. Xilinx 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 [12]. Şekil 2.7"de

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.



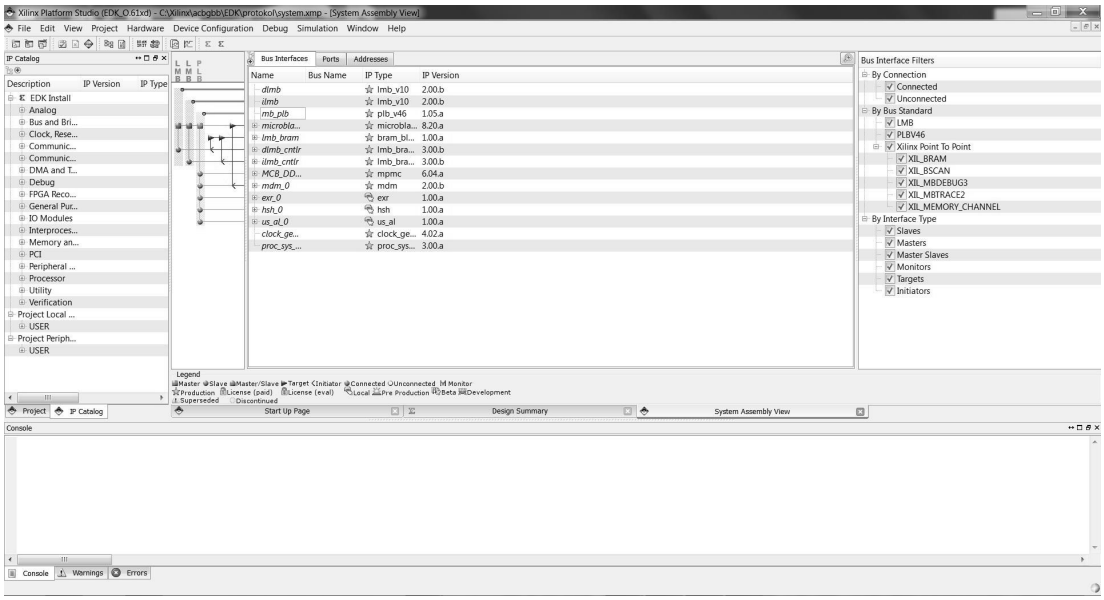

**Şekil 2.7 :** EDK Sistem Geliştirme Araçları [12].

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 [12].

EDK ortamı Temel Sistem Oluşturucu (Base System Builder, BSB) ile tasarımcıya kendi tasarım sisteminin tabanını oluşturma imkânı 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 ya da PowerPC gibi mikroişlemcilerin yazılımının tasarlanması aşamasına geçilmektedir.



**Şekil 2.8 :** Xilinx EDK Programının Görünümü.

## 2.9. Xilinx SDK Ortamı

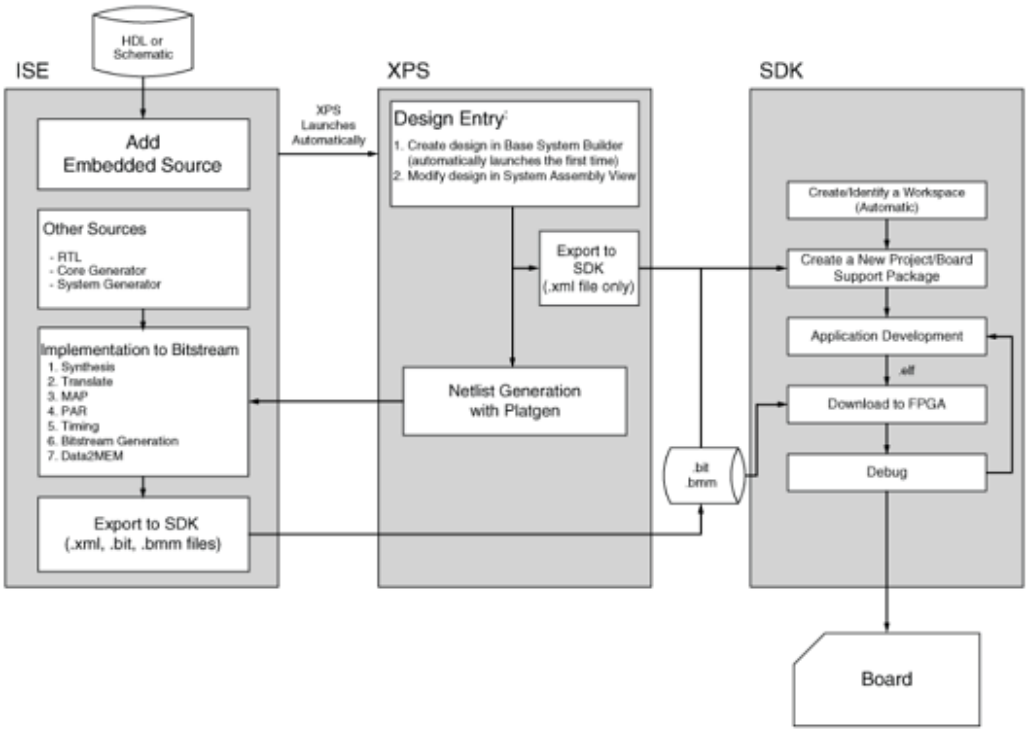
Yazılım Geliştirme Kiti (Software Development Kit, SDK) Xilinx firması tarafından EDK ortamında tasarlanan mikroişlemci merkezli sayısal sistem tasarımlarının yazılım tasarımını gerçeklemek için geliştirilen ara yüz ortamıdır. Xilinx'in tasarım ortamlarının eski sürümlerinde SDK, XPS geliştirme ortamı 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.

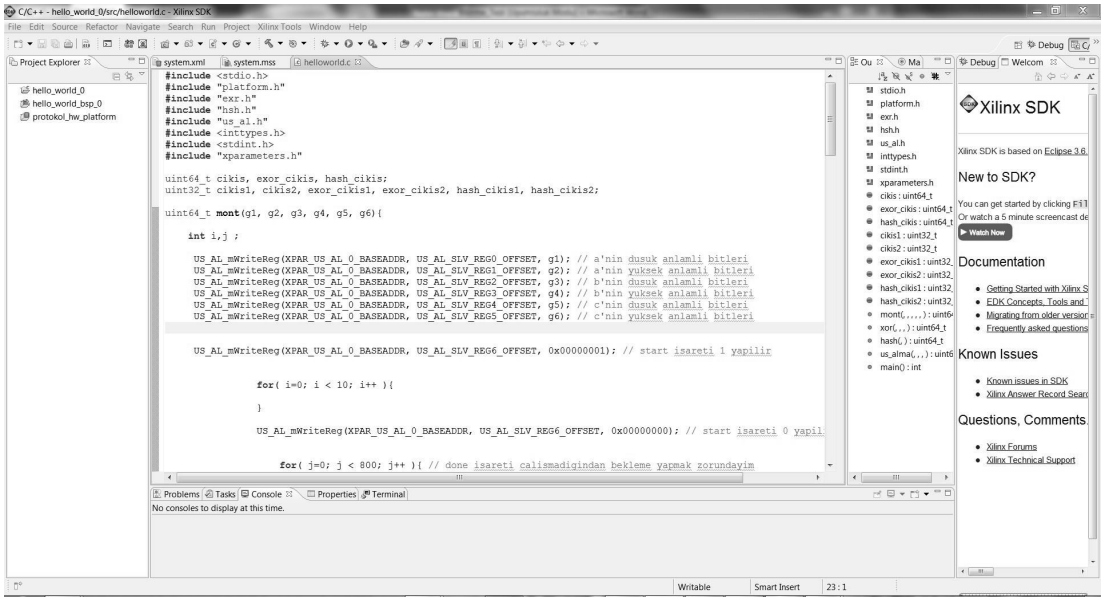
- Zengin özellikli C/C++ kod editörü ve derleme ortamı
- Proje yönetimi
- Tasarım yapılandırması uygulaması ve otomatik Makefile üretimi
- Hata navigasyonu
- Kaynak düzeyinde hata ayıklama ve gömülü hedeflerin görünüşü için iyi tümleştirilmiş ortam
- Kaynak kodu sürümü kontrolü

SDK tarafından kullanıcılarına sunulmuş başlıca özelliklerdir [13].

ISE, EDK ve SDK ortamları kullanılarak sıfırdan bir sistem tasarımının akışı Şekil 2.9'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 2.9 :** Sistem Tasarım Akışı [14].




**Şekil 2.10 :** Xilinx SDK Programının Görüntüsü.

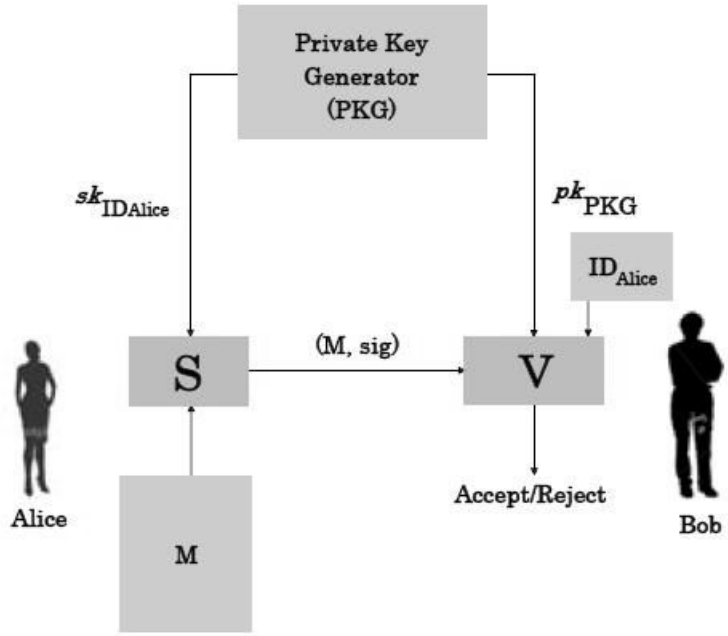
# 3. GERÇEKLENECEK PROTOKOL VE ALGORİTMALAR

Bu çalışmada güvenli bir NFC haberleşmesi sağlamak için Piseth Ith, Yoshihito Oyama, Atsuo Inomata ve Eiji Okamoto'nın yazmış olduğu "Implementation of ID-Based Signature in RFID System" adlı makaleden faydalanılmıştır.

## 3.1. Protokol

İlgili makalede Identity Based Signature yöntemi kullanılmaktadır. Bu yöntem Public Key Cryptography'nin başka bir uygulaması olmakla birlikte zaman ve hata bakımından olumlu yanları daha çoktur. Public Key Cryptography'de kullanıcıların iletişim sağlanmadan önce şifreleme ve imza ikililerini üretme zorunluluğu, kimlik doğrulaması ve şifreli mesajların değişimi için gerekli olan Certificate Authority (CA) zorunluluğu bu sistemlerin hem zaman olarak hem de hata oranı olarak IBS sistemlerinin gerisinde kalmasına sebep olmaktadır [15]. IBS'e ait mesajın imzalanması ve imzanın doğrulanmasına dair örnek anlatım Şekil 3.1'de verilmiştir. Genel olarak IBS sistemler dört ana parça halinde incelenir [2].

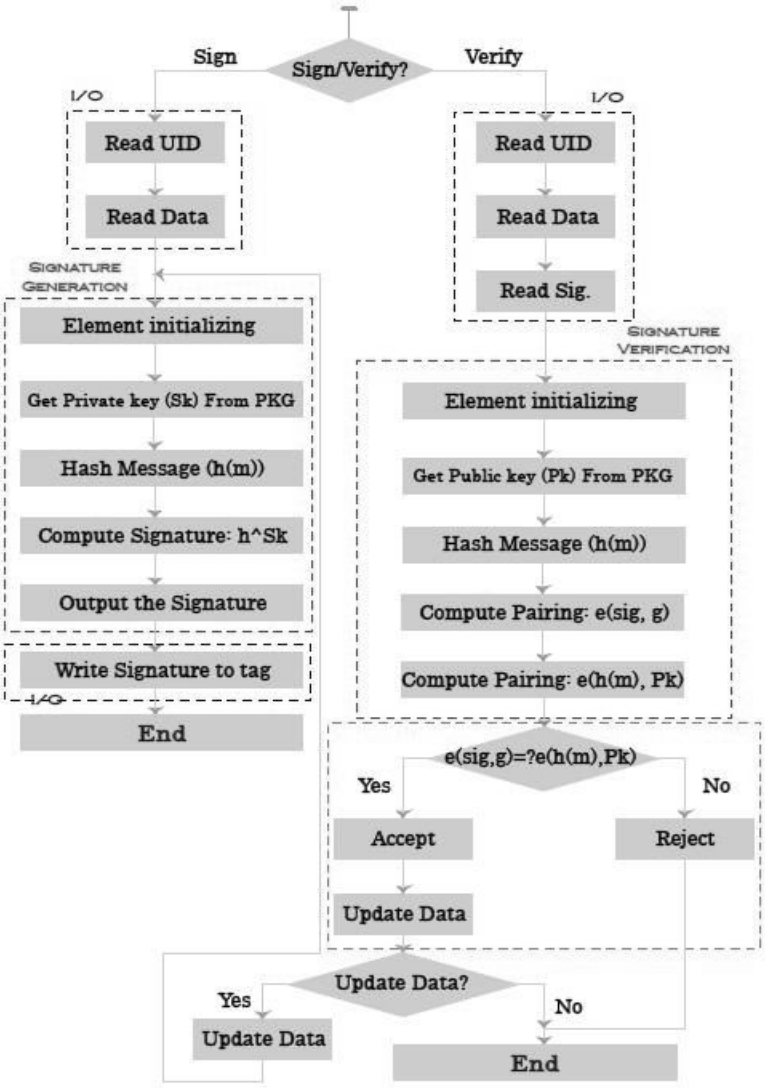
- Setup: Private Key Generator'nin (PKG) gizli ve açık anahtar ikililerini üretmesidir. Açık anahtar tüm kullanıcılara dağıtılır ve sabit parametredir.
- Private Key Extraction: Mesajı imzalayacak olan Alice, PKG'den kendi ID'sine ait gizli anahtarı alır.
- Signing: Alice, gizli anahtarıyla mesajı imzalar ve mesaj-imza ikilisini Bob'a yollar.
- Verification: Bob açık anahtarını ve gönderici olan Alice'in ID'sini kullanarak imzayı doğrular.




**Şekil 3.1 :** Identity Based Signature Scheme [2].

Bu çalışma kapsamında Dan Boneh, Ben Lynn ve Hovav Shacham'ın önerdiği "Kısa İmzalı Şema" gerçeklenecektir. Burada kısa imza kullanılmasındaki amaç zamandan avantaj elde etmektir. Kullanılan algoritma Şekil 3.2'de verilmiştir. IBS sistemlerinde olduğu gibi burada da setup, key generation, signing ve verification aşamaları söz konusudur.

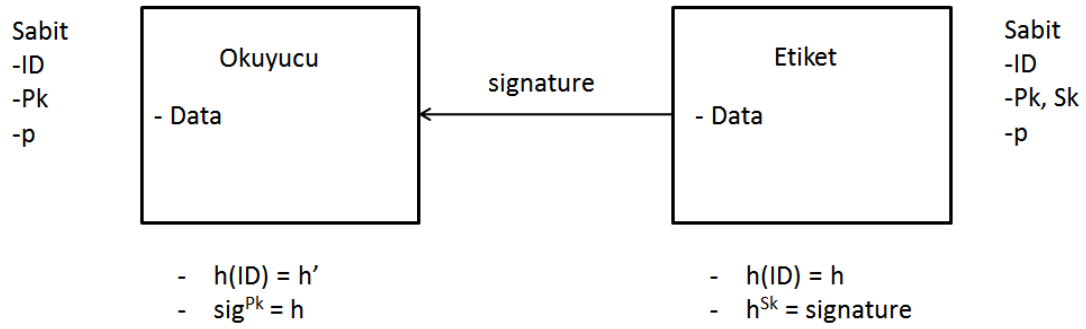
- Setup:
- Key Generation:
- Signing:
- Verification:




**Şekil 3.2 :** Boneh Şeması [2].

Algoritma bloklarına bakıldığında hash ve üs almaya ihtiyaç duyulduğu görülmektedir. Hash algoritması TEA ile gerçeklenmiştir. Üs alma ise donanımda gerçeklenen Montgomery Algoritması yardımıyla MicroBlaze üzerinde gerçeklenmiştir. Protokolün MicroBlaze üzerindeki gerçeklemesi Şekil 3.3'te görülmektedir. $P_k$ ile belirtilen açık anahtar, $S_k$ ile belirtilen gizli anahtarın çarpma işlemine göre tersidir. Daha önceden belirlenmiş olan $P_k$ açık anahtarı ve p asal sayısı yardımıyla, Denklem 3.1 kullanılarak Maple programı aracılığıyla gizli anahtar $S_k$ elde edilir. Mesajın onaylanması Şekil 3.3'te gösterilen h' = h olması durumda sağlanacaktır.

$$P_k = S_k^{-1} \bmod p \qquad (3.1)$$




**Şekil 3.3 :** Protokolün MicroBlaze üzerinde gerçeklenmesi.

### 3.2. TEA Şifreleme Algoritması

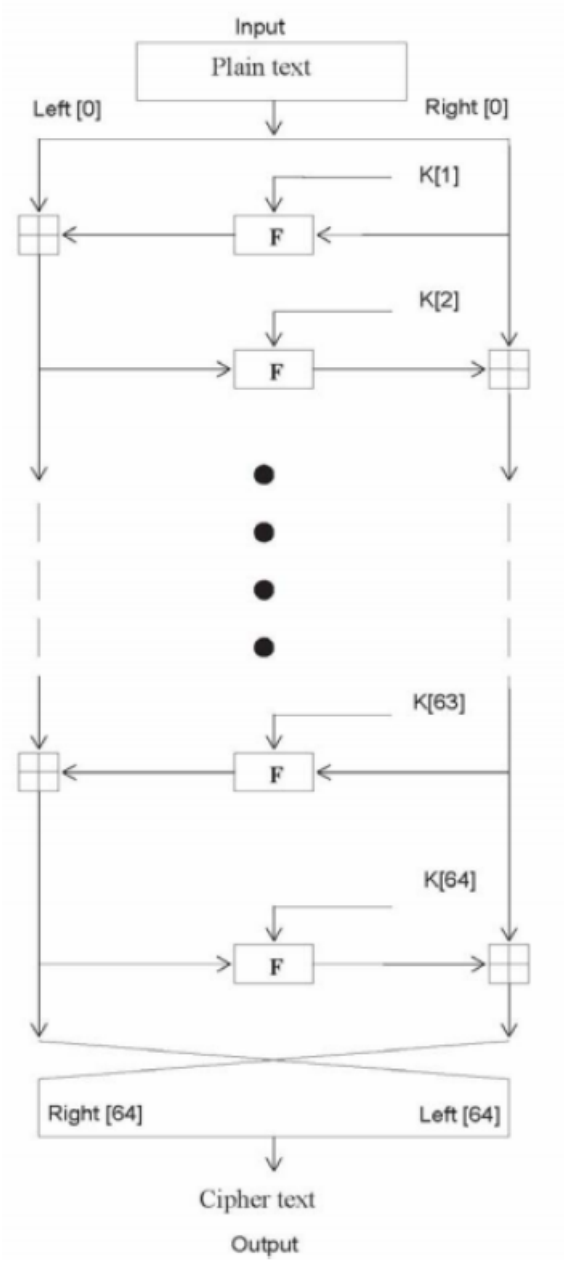
Günümüz RFID sistemlerinde etiket ve okuyucu arasındaki haberleşmenin şifreli yapılması atakları önlemek açısından son derece önemlidir. Ancak günümüzde teknolojinin gelişmesiyle birlikte küçülen etiket boyutlarıyla paralel olarak yeni şifreleme standartları ortaya çıkmaktadır. Daha az enerji tüketimi ve daha az yer kaplanması zorunluluğu bu gereksinimlere uygun şifreleme standartlarının kullanılmasını mecbur kılmıştır. Bu zorunluluktan ötürü bu çalışmanın şifreleme 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 [16].

TEA, karışık cebirsel işlemleri kullanan, Feistel türü şifreleme yapan, minimum hafıza alanı ve maksimum hız hedeflenerek oluşturulmuş bir şifreleme algoritmasıdır. Feistel türü şifreleme blok şeklinde şifreleme yapılmasını gerektirir ve sadece altı tur sonra tam yayınım sağlamaktadır. Böylece, şifrelenecek metinde 1 bit değiştirildiğinde çıkıştan alınan şifreli metine bu değişiklik 32 bit olarak yansıyacaktır. Zaman performansı ise bilgisayarlarda ve iş istasyonlarında oldukça etkileyicidir [17].

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 [17]. TEA şifreleme yapısında 128 bit uzunluklu

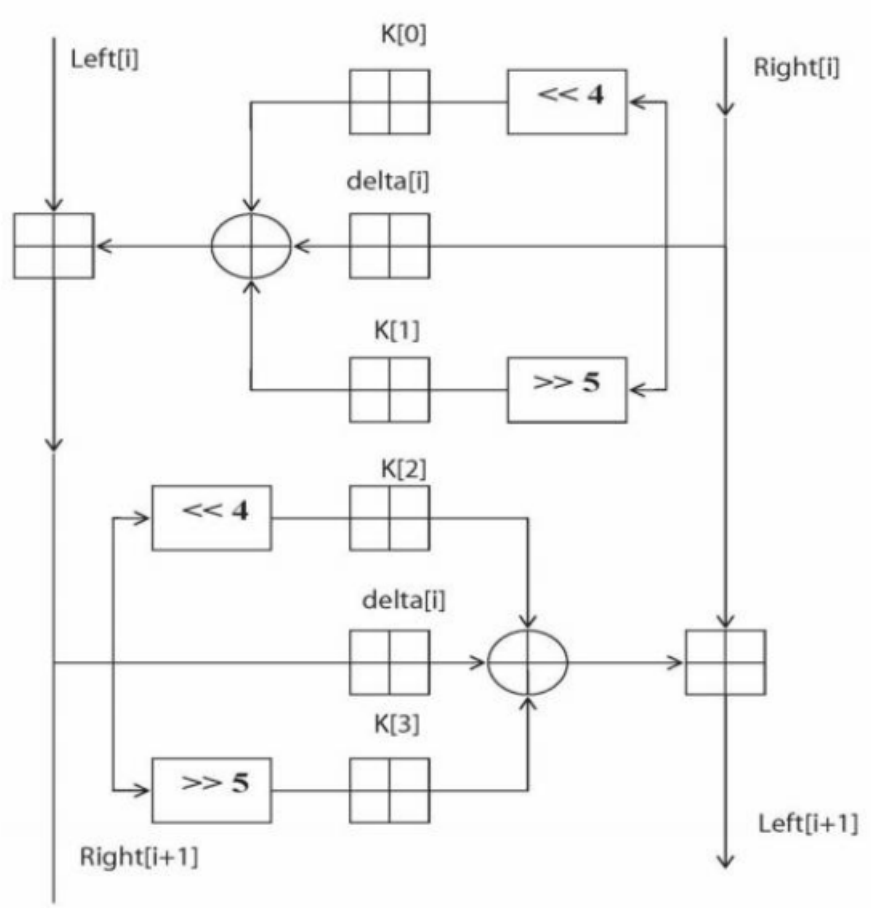
şifreleme anahtarı kullanılmaktadır. Bu 128 bit uzunluklu anahtar K[0], K[1], K[2] ve K[3] şeklinde dört adet 32 bit uzunluklu anahtarlara bölünerek işlemlere sokulur [13]. Şekil 3.4'de algoritmanın şifreleme yapan kısmının blok diyagram yapısı görülmektedir. Şifreleme yapısı 64 adet Feistel döngüsünden 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 [17]. 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 3.2'den hesaplanmaktadır [17].

$$\text{Delta} = ( \sqrt{5} - 1 ) * 2^{31} = 9E3779B9_h \qquad (3.2)$$




**Şekil 3.4 :** TEA Şifreleme rutini [17].

64 Feistel döngüsünde oluştuğu bilinen şifreleme yapısının Şekil 3.5'de görüldüğü gibi 32 döngüden oluşmaktadır. Yani Şekil 3.5 iki tane Feistel döngüsü içermektedir. Her bir döngüde döngüye giren metinler toplama, özel veya (exclusive or, XOR), mantıksal kaydırma işlemlerinden geçer.




**Şekil 3.5 :** TEA i. Döngüsü [17].

Şekil 3.5'de de görüldüğü gibi bir döngüye sağ taraftan giren şifresiz metin öncelikle 4 bit sola kaydırılır ve daha sonra K[0] anahtarıyla toplama işlemine girer. Yine aynı bilgi 5 bit sağa kaydırılarak K[1] anahtarıyla toplama işlemine girmektedir. Ayrıca bilginin kendisi de direkt olarak delta[i] sabitiyle toplama işlemine girmektedir. Daha sonra işleme giren bu üç koldan gelen veriler XOR işlemine girerek sol tarafta 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.

### 3.3. Montgomery Çarpımı Algoritması

Modüler çarpımla ilgili birçok öneri ve geliştirme vardır. İndirgeme yaklaşımlarına göre bu yöntemler ikiye ayrılabilir: soldan sağa indirgeme ve sağdan sola indirgeme [18]. Montgomery çarpımı sağdan sola indirgemeli yöntemlere dahildir. Peter L. Montgomery tarafından geliştirilen Montgomery Çarpımı [19] sağdan sola indirgemeli yöntemlerin tek örneğidir ve soldan sağa indirgeme metodlarıyla karşılaştırıldığında daha çok tercih edilir. Montgomery Çarpımı'nda klasik modüler çarpım yöntemlerindeki karşılaştırma ve indirgeme işlemlerinin yerine, toplama ve kaydırma işlemleri kullanılır.

A, B ve N k bit ikili sayı olarak düşünülürse R = AB mod N işlemi hesaplanacaktır. N'nin k bit olduğu düşünülerse, bu sayı $2^{k-1} \leq N \leq 2^k$ aralığında olacaktır ve $R = 2^k$ olarak kabul edilir. Aynı zamanda Montgomery indirgeme algoritması için tek sayı olan N, R ile aralarında asal olmalıdır ($\gcd(R,N) = \gcd(2^k,N) = 1$).

Seçilen tam sayı A'nın ($0 \leq A < N$) R'ye göre tanımlanışı şu şekildedir:

$$A = A.\ R \bmod N \tag{3.3}$$

A modulo N işleminin R'ye göre Montgomery indirgemesi ise şu şekildedir:

$$A.\ R^{-1} \bmod N \tag{3.4}$$

Montgomery indirgemesinin önemli bir adımı $ABR^{-1} \bmod N = ABR \bmod N$ eşitliğidir ve bu eşitlik modular exponentiation işleminde kullanılmaktadır [20]. Örnek olarak, $A^5 \bmod N$ hesaplanmak istenirse öncelikle $A = A.R \bmod N$ hesaplanır. Daha sonra AA'nın Montgomery indirgemesi olan $X = A^{-2}R^{-1} \bmod N$ hesaplanır. Bu işleme göre $X^2$'nin indirgemesi $X^2R^{-1} \bmod N = A^{-4}R^{-3} \bmod N$ olmuştur. Son olarak, $(X^2R^{-1} \bmod N)A$'nün indirgemesi olarak $(X^2R^{-1})AR^{-1} \bmod N = A^{-5}R^{-4} \bmod N = A^5\ R \bmod N$ hesaplanır. Bu işlem $R^{-1} \bmod N$ ile çarpılarak istenilen $A^5 \bmod N$ işlemi elde edilmiş olur.

N, b tabanında bir sayı seçilirse R için uygun bir seçim $b^k$ olacaktır [20]. Ek olarak Montgomery indirgeme algoritmasını tanımlamak için N′ olarak belirtilen yeni bir değişken tanımlanmaktadır [21] ve daha önce tanımlanan R ile ilişkisi $R.R^{-1} - N.N' = 1$ olarak belirtilir. $R^{-1}$ ve N′ tam sayıları Euclidian algoritması yardımıyla

hesaplanabilir [20]. $AR^{-1}$ mod N işlemini yapan Montgomery indirgeme algoritması Şekil 3.6'da verilmiştir.

**Require:** Integers $A = (a_{2k-1}a_{2k-2}\cdots a_0)_b < NR$, $N = (n_{k-1}n_{k-2}\cdots n_0)_b$ with $\gcd(N,b) = 1$, $R = b^k$, $N' = -N^{-1} \bmod b$.
**Ensure:** $AR^{-1} \pmod N$

```
T := A
for i = 0 to (k − 1) do
    q_i := t_i N' mod b
    T := T + q_i N b^i
end for
T := T / b^k
if T ≥ N then
    T := T − N
end if
return T
```

**Şekil 3.6 :** Montgomery indirgeme algoritması [20].

İki tam sayının çarpımının Montgomery indirgemesine Montgomery Çarpımı adı verilmektedir ve Şekil 3.7'de belirtilmiştir. Montgomery çarpım algoritması Montgomery indirgemesi ve sıradan çarpım işlemlerinden oluşmaktadır [22]. İndirgeme adımları olarak öncelikle kısmi toplam ve kısmi çarpımın en düşük anlamlı bitleri tarafından modülüs katsayısı hesaplanır. Daha sonra bu katsayı, en düşük anlamlı biti 0 yapan kısmı toplama eklenir.

**Require:** Integers $X = (x_{k-1}x_{k-2}\cdots x_0)_b$, $Y = (y_{k-1}y_{k-2}\cdots y_0)_b$, $N = (n_{k-1}n_{k-2}\cdots n_0)_b$ with $0 \le X,Y < N$, $R = b^k$ with $\gcd(N,b) = 1$ and $N' = -N^{-1} \bmod b$.
**Ensure:** $XYR^{-1} \pmod N$

```
A := 0(Notation: A := (a_k a_{k−1} ··· a_0)_b.)
for i = 0 to (k − 1) do
    q_i := (a_0 + x_i y_0) N' mod b
    A := (A + x_i Y + q_i N) / b
end for
if A ≥ N then
    A := A − N
end if
return A
```

**Şekil 3.7 :** Montgomery çarpım algoritması [22].

Q′ değerini oluşturmak için Montgomery çarpımı boyunca biriktirilen $q_i$ değerleri söz konusudur. Daha önce de bahsedildiği üzere A, B, N ve Q′ değerleri, pozitif bir tam

sayı olan h cinsinden k = 2h bit olarak belirlenir, en yüksek ve en düşük anlamlı h bitlerine $r = 2^h$ olmak üzere şu şekilde ayrılırlar:

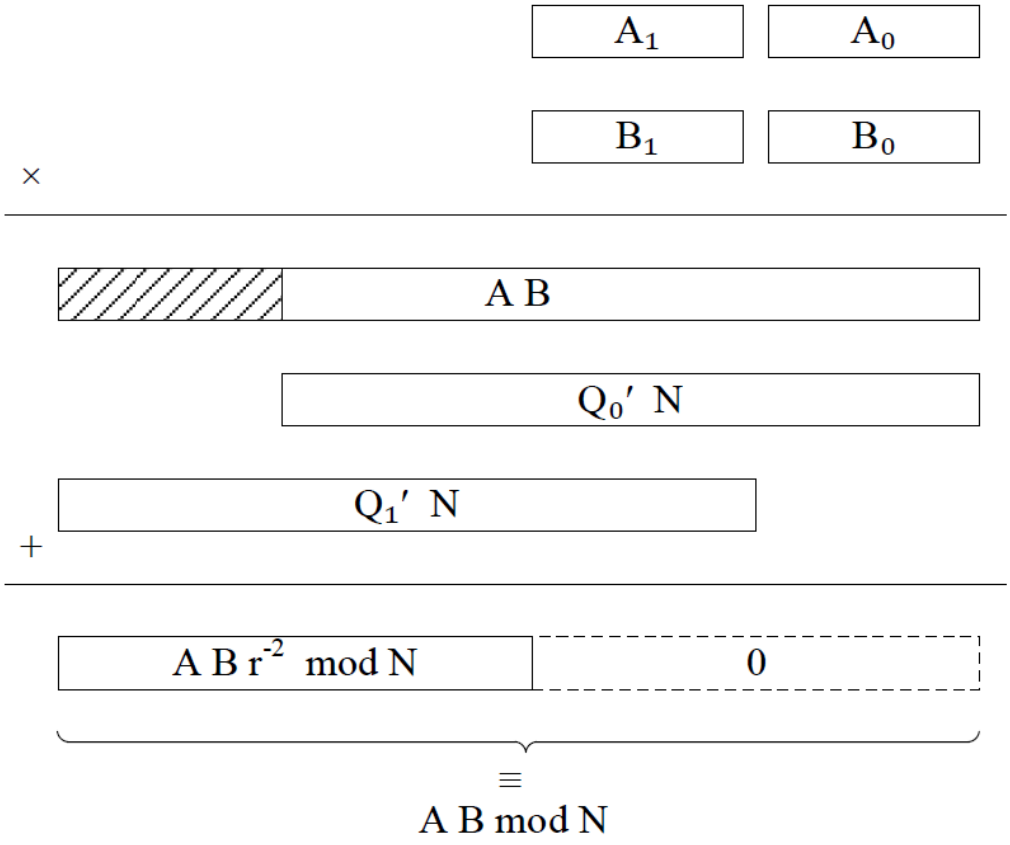
$$A = A_1.2^h + A_0 = A_1.r + A_0 \quad (3.5)$$

$$B = B_1.2^h + B_0 = B_1.r + B_0 \quad (3.6)$$

$$N = N_1.2^h + N_0 = N_1.r + N_0 \quad (3.7)$$

$$Q' = Q'_1.2^h + Q'_0 = Q'_1.r + Q'_0 \quad (3.8)$$

Soldan sağa modüler çarpım olarak da adlandırılan Montgomery çarpımının süreci Şekil 3.8'de görülmektedir. Şekil 3.8'de de anlatıldığı gibi Montgomery çarpımı çarpmanın ve indirgemenin birleşimidir. $ABr^{-2}$ mod N işleminin sonucunu elde etmek için AB çarpımına, $Q'N = Q'_0N + Q'_1Nr$ eklenir. Algoritmanın ilk h döngüsü boyunca $Q'_0N$, AB çarpımına eklenir ve AB çarpımının en anlamlı h biti taralı olarak gösterilmiştir.



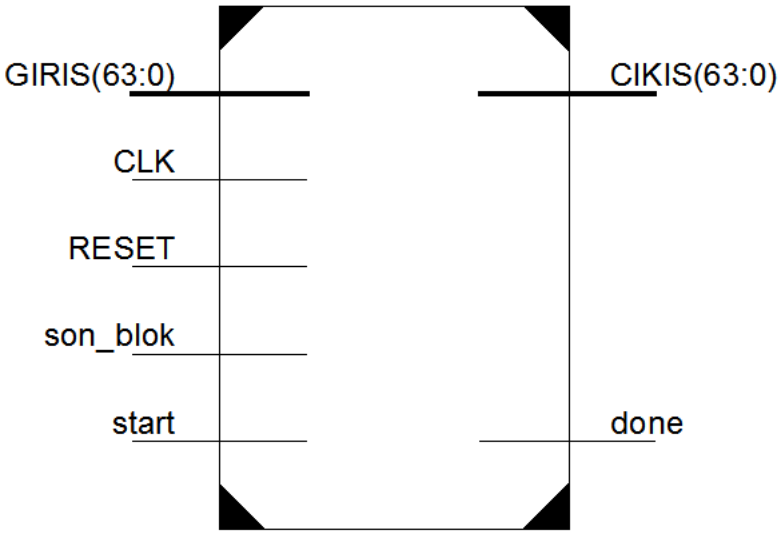

**Şekil 3.8 :** Montgomery çarpımı [18].

# 4. DONANIM TASARIMI

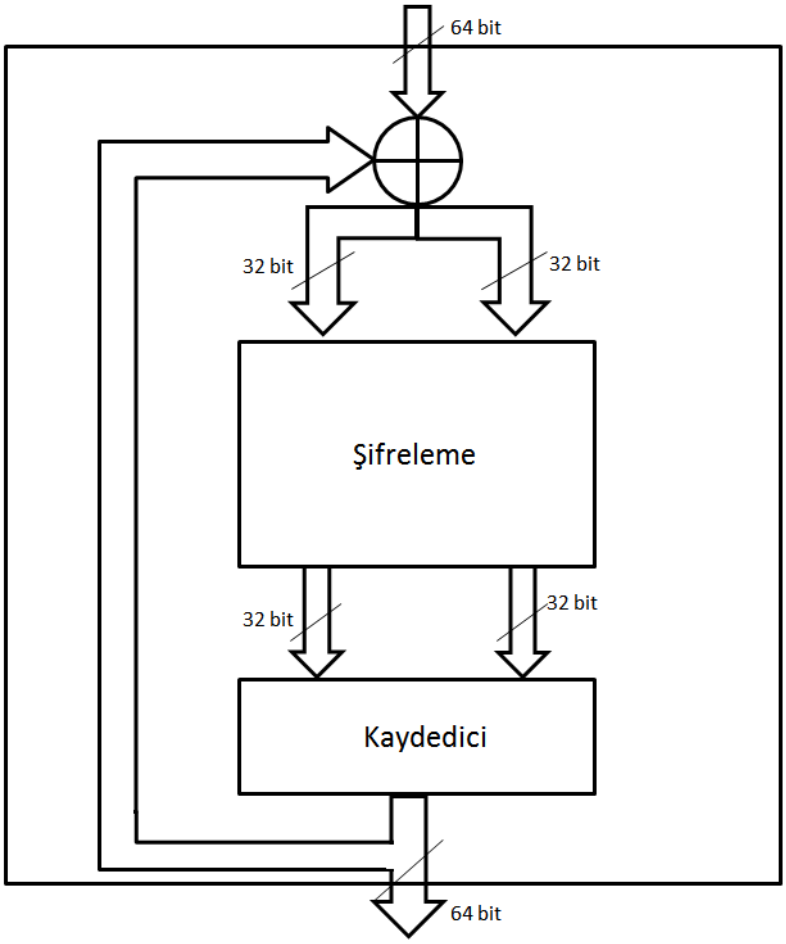
Protokolün gerçeklenmesi aşamasına ilk olarak protokolde yer alan donanımların Verilog ve VHDL donanım tanımlama dilleri kullanılarak FPGA üzerinde tasarlanmasıyla başlanmıştır. Protokolde bulunan donanımlar; daha önce hash olarak adlandırılan özet bloğu ve Montogomery Çarpıcı bloğu isimleri altında toplanmıştır. Donanım tasarımı aşaması tamamlandıktan sonra bu donanımları ve sistemi kontrol etmek amacıyla FPGA içerisindeki Microblaze mikroişlemcisi üzerinde yazılım tasarımı yapılmıştır.

## 4.1. Özet Bloğu

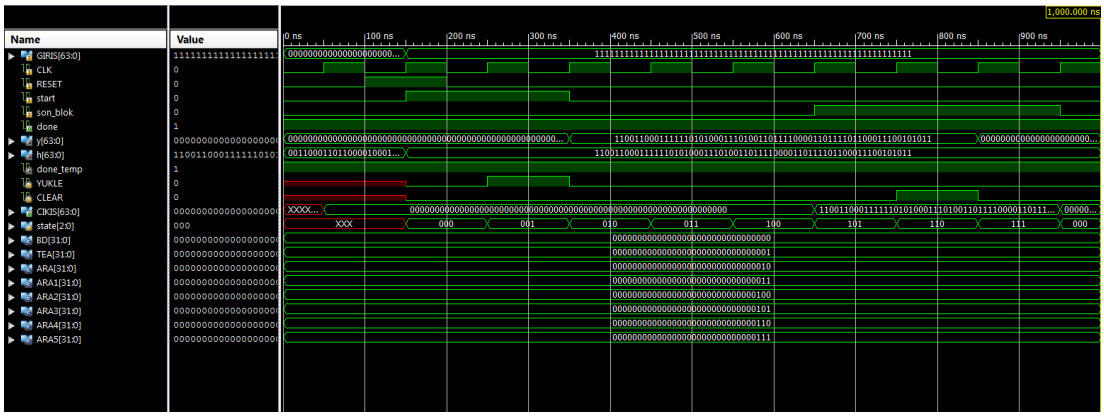
Uygulanan protokol gereği [2] sistemin girişine kaç bit veri geleceği belirsizdir. Bu amaçla özet bloğu döngü içeren bir sistem olarak tasarlanmıştır. Sistemin girişi 64 bittir ve veriler ilk olarak özel veya bloğuna gelmektedir. Özel veya bloğundan çıkan veri, TEA'nın yapısı gereği 32 bitlik iki ayrı veriye ayrılarak şifreleme bloğuna girer ve aynı şekilde 32'şer bit olarak çıkar. Şifreleme bloğundan çıkan verilerin saklanması işlemi bir kaydedici blok yardımıyla yapılmıştır. Kaydedici blok çıkışı ise tekrar özel veya bloğuna bağlanarak yeni veriyle işleme girmesi amaçlanmıştır. Böylece gelen her bit dizisi şifreleme işleminde geçecek ve uzun bit dizileri de daha önceden elde edilen şifrelenmiş bit dizileriyle tekrar şifrelenmiş olur. Özet bloğunda kullanılan start ve done işaretleri, yapının Microblaze ile haberleşmesi amacıyla kullanılmıştır. Son blok adı verilen girişle donanıma son 64 bitlik girişin verildiği kontrol edilmektedir. Özet bloğu, özel veya bloğu, şifreleme bloğu ve kaydedici bloklarından oluşmaktadır. Tüm bu bloklar sonlu durum makinesiyle kontrol edilmiştir.




**Şekil 4.1 :** Özet modülü.



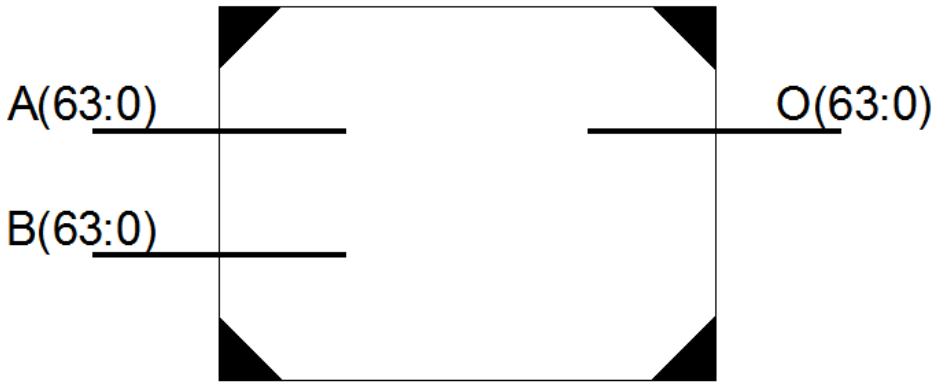

**Şekil 4.2 :** Özet Bloğu yapısı.




**Şekil 4.3 :** Özet bloğu benzetim sonucu.

### 4.1.1. Özel Veya Bloğu

Özet bloğunun girişinde kullanılmak üzere 64 bit girişi, 64 bit çıkışı olan Şekil 4.4'de görülen özel veya bloğu tasarlanmıştır. Özel veya bloğunun çıkışı şifreleme bloğunun girişine gelmektedir. TEA'nın girişinin 32'şer bitlik olması özel veya bloğunun çıkışının ikiye ayrılmasını gerektirir. Bu üst modülde sağlanmıştır.

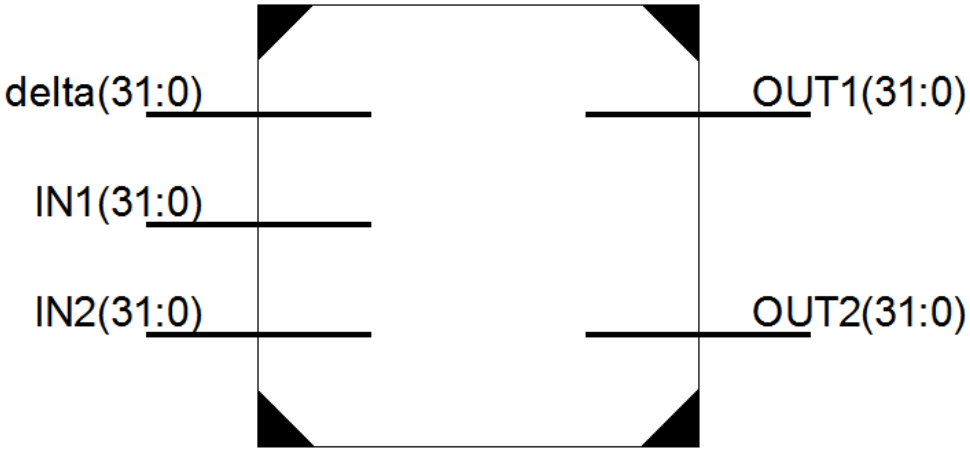



**Şekil 4.4 :** Özel veya modülü.

### 4.1.2. Şifreleme Bloğu

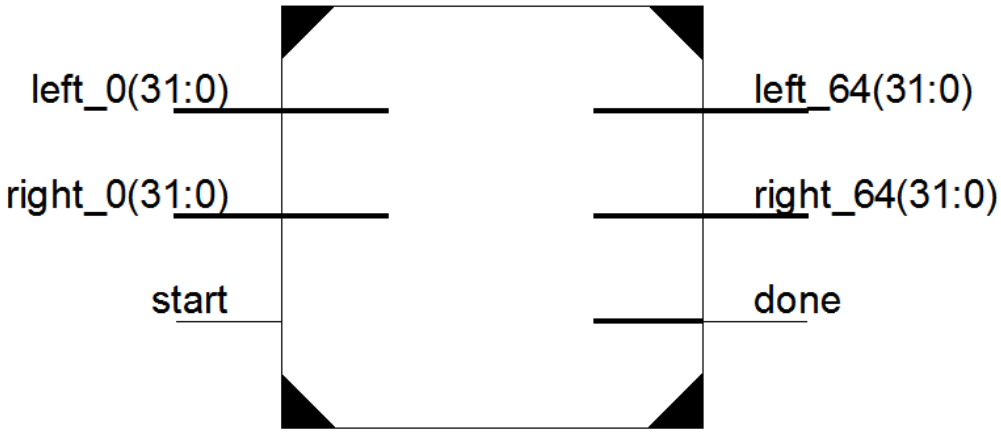
Şifreleme bloğu donanımsal olarak özel veya modülü, toplama modülü ve mantıksal kaydırma modülünden oluşmaktadır. Bu yüzden öncelikle bu alt bloklar tasarlanmıştır. Bu çalışma kapsamın Semih Alparslan'ın tasarlamış olduğu şifreleme bloğu kullanılmıştır. Yazılım tasarımı sırasında Microblaze mikroişlemcisi ile kontrol etmenin kolaylaştırılması için, şifreleme bloğu kombinezonsal olarak

tasarlanmıştır [23]. Tamamen kombinezonsal olarak tasarlanan şifreleme bloğu 32 döngü beklemek zorunda kalmayarak tek seferde şifreleme yapar. 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 4.5'deki gibidir. 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.

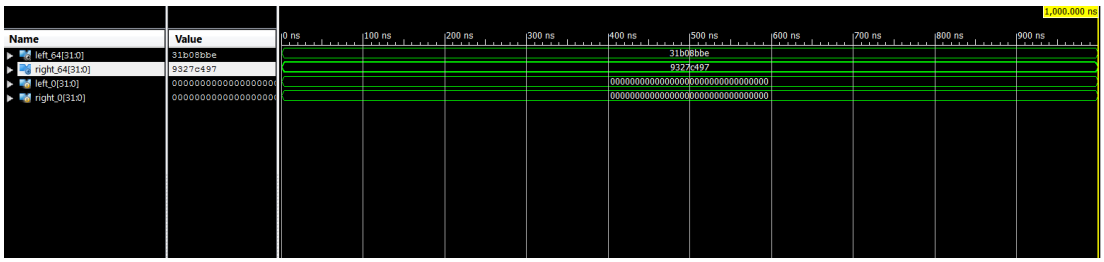



**Şekil 4.5 :** Kombinezonsal şifreleme bloğunun bir döngüsü.

Üst blok yani saat işaretsiz şifreleme bloğu oluşturulurken Şekil 4.4'de 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 4.6'de görüldüğü gibidir. Start ve done işaretleri Microblaze'in donanımı kontrol edebilmesi amaçlıdır.



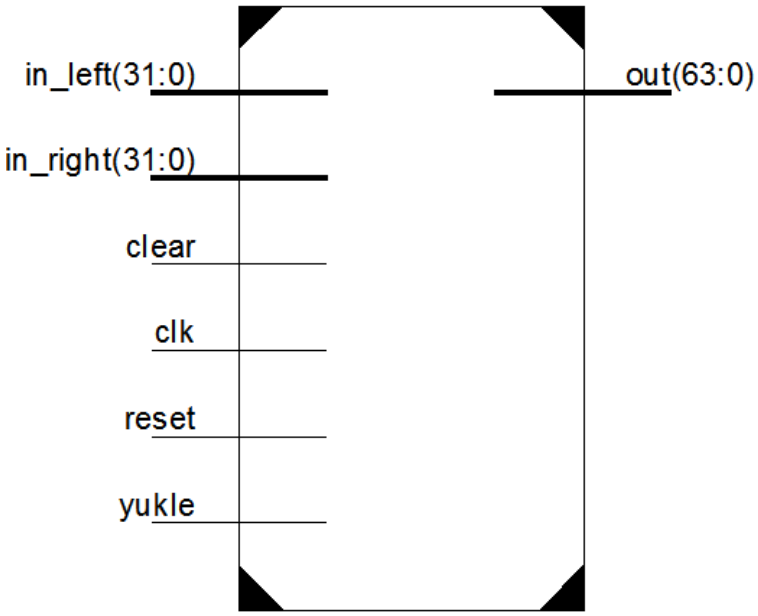

**Şekil 4.6 :** Kombinezonsal şifreleme bloğu.




**Şekil 4.7 :** Kombinezonsal şifreleme bloğu benzetim sonucu.

### 4.1.3. Kaydedici Bloğu

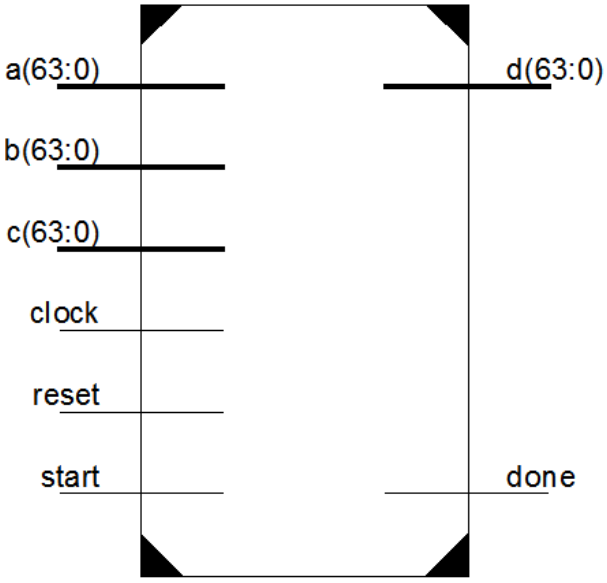
Kaydedici bloğu, şifreleme bloğundan çıkan verilerin saklanması amacıyla tasarlanmıştır. Şifreleme bloğunun 32'şer bitlik çıkış yapısından dolayı 32 bitlik veri girişlerine sahiptir. Reset ve clear girişleri sayesinde kaydedicinin temizlenebilmektedir. Yükle girişi ise kaydetme işleminin başlamasını kontrol etmektedir.



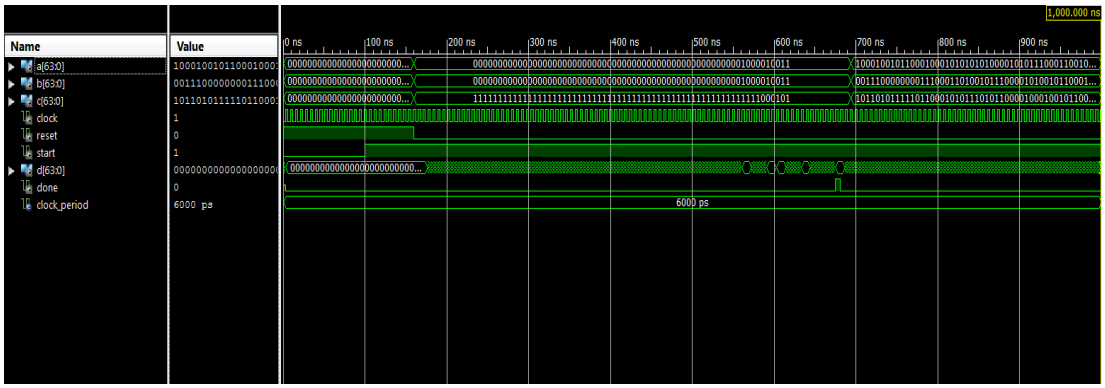

**Şekil 4.8 :** Kaydedici bloğu.

### 4.2. Montgomery Çarpıcı Bloğu

Bu çalışmada Yüksek Mühendis Ahmet Arış'ın tasarladığı Montgomery Çarpıcı Bloğu kullanılmıştır. Her adımda 1 bit işleyen modüler çarpıcı şeklindedir. 64 bitlik veriler için $d = a.b.2^{-64}$ mod c işlemini gerçekleştirmektedir. 64 bitlik modüler çarpım maksimim 83 saat çevriminde tamamlanabilir. Daha erken tamamlanması ise done sinyali ile kontrol edilebilir. 64 bitlik modüler çarpımı maksimum 83 saat çevriminde tamamlamasının nedeni Montgomery'nin gerektirdiği son toplama ve çıkarma işlemleridir. Bu yapıda çarpıcının boyu parametrik olarak verildiğinden farklı uzunluklardaki verilerin kullanımı için uygundur. Montgomery çarpıcı bloğu, saklayıcı, full adder ve flip flop yapılarından oluşmaktadır. Modüler çarpım bu şekilde donanımsal olarak tasarlanarak daha sonra Microblaze mikroişlemcisinde üs alma işlemi yapılmıştır.




**Şekil 4.9 :** Montgomery çarpıcı bloğu.




**Şekil 4.10 :** Montgomery Çarpıcı Bloğu benzetim sonucu.

# 5. YAZILIM TASARIMI

Donanımları kontrol etmek, donanımlar arası veri akışını sağlamak amacıyla Microblaze mikroişlemcisi kullanılmıştır. Yazılımsal tasarım olarak ise Şekil 4.3'te verilen yapı kullanılmıştır. Bu yapıda görülen üs alma işlemleri için öncelikle Montgomery çarpımı bloğu donanımının Microblaze üzerinde gerçeklenmesi sağlanmış, daha sonra üs almak için Şekil 5.1'de verilen algoritma C dilinde gerçeklenmiştir. Özet işlemleri için ise yapıdan ayrı tasarlanan özel veya bloğu ile şifreleme ve kaydedici bloklarının iletişimi sağlanmıştır. Bu yapı için bir FPGA kartı kullanılması yeterli olmuştur.

INPUT: $m = (m_{l-1} \cdots m_0)_b$, $R = b^l$, $m' = -m^{-1} \bmod b$, $e = (e_t \cdots e_0)_2$ with $e_t = 1$, and an integer $x$, $1 \leq x < m$.
OUTPUT: $x^e \bmod m$.

1. $\widetilde{x} \leftarrow$ Mont$(x, R^2 \bmod m)$, $A \leftarrow R \bmod m$. ($R \bmod m$ and $R^2 \bmod m$ may be provided as inputs.)
2. For $i$ from $t$ down to 0 do the following:
   2.1 $A \leftarrow$ Mont$(A, A)$.
   2.2 If $e_i = 1$ then $A \leftarrow$ Mont$(A, \widetilde{x})$.
3. $A \leftarrow$ Mont$(A, 1)$.
4. Return$(A)$.

**Şekil 5.1 :** Üs alma algoritması [20].

# 6. SONUÇLAR

Bu bitirme çalışması kapsamında güvenli bir NFC sistemi tasarlanması hedeflenmiştir. Bu amaçla iki RFID etiket Microblaze üzerinde gerçeklenerek seçilen protokol uygulanmıştır ve yazılım donanım ortaklı bir sistem elde edilmiştir. Gerek yer gerek hız bakımından daha verimli bir yöntem olduğundan TEA şifreleme algoritması seçilmiştir.

Çalışma kapsamında özet bloğunun sonlu durum makinesiyle kontrol edilmesi zorunluluğu, Microblaze ve özet bloğu donanımının haberleşmesi arasında problemler doğurmuştur. Bu problem özet bloğundaki özel veya donanımının ayrı bir donanım olarak gerçeklenmesiyle çözülmüş olsa da bu durum hata olasılığını arttırmıştır. Ayrıca Microblaze mikroişlemcisinin sonuçları yakalayabilmesi için donanıma fazladan birkaç durum eklenmiştir. Bu istenmeyen bir durum olsa da çözüme katkıda bulunmuştur.

Montgomery çarpıcı bloğunun Microblaze ile haberleşmesi sırasında ise done sinyalinin alınması konusunda problem yaşanmıştır. Bu sorun sistemi bir süre bekleterek sonucun alınması sağlanarak çözülmüştür. Fakat bu çözüm daha yavaş bir sistem ortaya çıkarmaktadır.

Özel veya bloğu ayrı olarak tasarlanarak, FPGA üzerinde 135 dilim işgal eden bir özet bloğu sistemi elde edilmiştir. Montgomery çarpıcı bloğu ise FPGA üzerinde 484 dilim işgal etmektedir.

İleri çalışmalarda yukarıda bahsedilen sonlu durum makinesi ve done sinyali problemleri daha verimli bir şekilde çözülerek daha gelişmiş bir sistem tasarlanabileceği tespit edilmiştir.

## KAYNAKLAR


[1] NFC Forum, [Alıntı Tarhi: 10 Mayıs 2013], http://www.nfc-forum.org/aboutnfc/

[2] **Ith, P.; Oyama, Y.; Inomata, A.; Okamoto, E**., 2007. Implementation of ID-based signature in RFID system , Asia-Pacific Conference on , vol., no., pp.233-236.

[3] **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.

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

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

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

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

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

[9] MicroBlaze Soft Processor Core, [Alıntı Tarihi: 3 Nisan 2013],

http://www.xilinx.com/tools/microblaze.htm

[10] Atlys™ Spartan-6 FPGA Development Board, [Alıntı Tarihi: 10 Mayıs 2013], http://www.digilentinc.com.

[11] **Xilinx**, 2007. MicroBlaze Processor Reference Guide.

[12] **Xilinx,** 2007. Embedded System Tools Reference Manual.

[13] **Xilinx,** Software Development Kit Help Contents, [Alıntı Tarihi: 11 Mayıs 2013],http://www.xilinx.com/support/documentation/sw_manuals/xilinx12_2/SDK_Doc/index.html.

[14] **Xilinx,** 2011. EDK Concepts, Tools and Techniques.

[15] **Youngblood, C.,** 2005. An Introduction to Identity-Based Cryptography, CSEP 590TU

[16] **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.

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

[18] **Saldamli, G.,** (2011). Partially Interleaved Modular Karatsuba-Ofman Multiplication, *preprint*.

[19] **Montgomery, P.L.,** (1985). Modular multiplication without trial division,Mathematics of Computation, **44(170)**, 519–521.

[20] **Menezes, A., van Oorschot, P. and Vanstone, S**., (1997). Handbook of Applied Cryptography, CRC Press.

[21] **Çetin Kaya Koç,** (1994). High-Speed RSA Implementation, Technical ReportTR201, RSA Laboratories, 73 pages.

[22] **Arış, A**., 2012. "Design and Implementation of RSA Cryptosystem Using Partially Interleaved Modular Karatsuba-Ofman Multiplier", *Yüksek Lisans Tezi,* İstanbul Teknik Üniversitesi Bilgisayar Mühendisliği, İstanbul.

[23] **Alparslan, S.,** 2012. "Güvenli RFID Sistemler İçin Bir Kimlik Doğrulama Protokolünün Gerçeklenmesi", *Lisans Tezi*, İstanbul Teknik Üniversitesi Elektronik ve Haberleşme Mühendisliği, İstanbul.

[24] **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.

## ÖZGEÇMİŞ

**Adı Soyadı:** Ahmet Çağrı BAĞBABA

**Doğum Yeri ve Tarihi:** Ankara, 1989

**Lise:** Ankara Atatürk Lisesi; 2003-2007

**Lisans:** İstanbul Teknik Üniversitesi, Elektronik Mühendisliği; 2008-2013