İSTANBUL TEKNİK ÜNİVERSİTESİ

ELEKTRİK – ELEKTRONİK FAKÜLTESİ

GÜVENLİ BİR RFID UYGULAMASININ GERÇEKLENMESİ

BİTİRME ÖDEVİ

Mehmet SOYBALI

040050314

Bölümü: Elektronik ve Haberleşme Mühendisliği

Programı: Elektronik Mühendisliği

Danışmanı: Yrd. Doç. Dr. Sıddıka Berna Örs Yalçın

MAYIS 2010

ÖNSÖZ

Bitime çalışmam boyunca bana vakit ayırıp yol gösteren, bilgi ve tecrübesinden yararlandığım, birlikte çalışmaktan büyük mutluluk duyduğum saygı değer danışmanım Yrd. Doc. Dr. S. Berna Örs Yalçın'a sonsuz teşekkürlerimi sunarım.

Başta Araş. Gör. Yük. Müh. Ramazan Yeniçeri olmak üzere bütün Gömülü Sistem Tasarımı Laboratuvarı ekibine de sınırsız yardım ve arkadaşlıklarından ötürü teşekkürü borç bilirim.

Mayıs 2010 Mehmet SOYBALI

İÇİNDEKİLER

GÜVENLİ BİR RADYO FREKANSLI TANIMA SİSTEMİ (RFID) UYGULAMASININ GERÇEKLENMESİ

ÖZET

Gelişen teknoloji bir yandan insan hayatını kolaylaştırırken diğer yandan da yeni tehlikeler doğurmaktadır. Kötü niyetli kimselerce kullanılan gelişmiş teknoloji, gündelik hayatın her yönünde güvenlik açıkları oluşturmaktadır. Tam bu noktada kriptoloji bilimi devreye girmekte, bu açıkların giderilmesinde başrol oynamaktadır. Bahsi geçen güvenlik açıklarından bazıları da Radyo Frekanslı Tanıma Sistemlerinin (Radio Frequency İdentification - RFID) payına düşmekte, buna karşılık yine kriptolojik yöntemler ile güvenlik önlemleri alınmaktadır. Bu çalışma da bir RFID uygulamasının nasıl güvenli hale getirilebileceği konusunda çözüm önerisi sunmaktadır. Çalışma sadece bu yönüyle kalmamakta, teorik olarak kullanılan bir yapının pratik gerçeklemesini icra ederek, bu yapının bir uygulama alanında nasıl kullanılacağına da yer vermektedir.

Açık anahtarlı kriptolojide oldukça sık kullanılan tek yönlü fonksiyonlara dayanarak yola çıkılmış, RFID güvenliğini sağlamak için Fiziksel Kopyalanamaz Fonksiyon (Physical Unclonable Function-PUF) kullanılması öngörülmüştür. Kullanılacak fonksiyonun birden fazla fiziksel gerçeklemesi olmakla beraber, bu çalışmada veri seçici(multiplexer) tabanlı devre yapısı kullanılmıştır.

Çalışmada ilk olarak dayanak noktası olan teori hakkında kısa bilgi verilmiştir. Arkasından PUF devresinin gerçeklenme aşamaları ve bu aşamalarda karşılaşılan sorunlar anlatılmıştır. Daha sonra devreden ölçüm almak için gerekli olan düzeneğin hazırlanmasına ve alınan ölçüm sonuçlarına yer verilmiştir. Son olarak da elde edilen sonuçların yorumlanarak hangi devre yapısının kullanıma uygun olduğu belirlenmiştir.

IMPLEMENTATION OF A SECURE RADIO FREQUENCY IDENTIFICATION APPLICATION

SUMMARY

Developing technology facilitates human life, but on the other hand raises new dangers. Advanced technology used by malicious people, constitute vulnerability in every aspect of daily life. At this point, the science of cryptology is engaged in the matter and it has played a leading role in eliminating deficits. Radio Frequency Identification System (RFID) shares some of the mentioned vulnerability, as opposed this situation the security measures are stil provided by cryptological methods. This study offers solutions on an RFID application how to be safer. It is also includes a practical reality of the structure used as a theoretical exercise and also includes how to use these structure in the application areas.

One-way functions generally used in public-key cryptography were selected as a beginning point, after that Physical Unclonable Functions (PUF) were suggested to use for ensuring the RFID safety. Although more than one physical implementation of the function is available, in this study, multiplexer based circuit is used.

Firstly, a brief information was given about the main theory of the study. Secondly, the PUF circuit implementation stages and the problems encountered were explained. Then the preparation of the measurement and measurement results were mentioned.

1. GİRİŞ

Bu çalışmada Radyo Frekanslı Tanıma Sistemleri(Radio Frequency İdentification - RFID) sistemlerin güvenli hale gelmesi için kullanılabilecek bir devre gerçeklenmesi ve gerçeklenen devrenin pratikte kullanılabilirliğinin incelenmesi amaçlanmıştır.

Radyo Frekanslı Tanıma Sistemleri günümüzde oldukça yaygın olarak kullanılan ve Auto-ID sistemlerinde barkodun yerini alacağı öngörülen kablosuz sistemlerdir. Giriş ve çıkışlarda kart okutma sistemleri, RFID etiketi ile etiketlenmiş ürünlerin tanımlanması en çok bilinen uygulamalarıdır. Bu sistemlerin daha da yaygınlaşmasının önündeki en büyük engel ise güvenliktir. Okuyucu ile etiket arasında veri iletişimi bulunması kullanılan verilerin mümkün olduğunca sağlam şekilde korunmasını gerektirir. Aksi takdirde bahsi geçen verilerin kopyalanması, tahrip edilmesi veya kötüye kullanılması muhtemeldir. Bu tür sonuçların önüne geçebilmede alıcı ve verici arasında kimlik doğrulama işlemi bulunması önemli bir rol oynamaktadır. Yani iletişimin öteki ucunda kimin olduğunun bilinmesi, işlem yapma yetkisine sahip olup olunmadığı öğrenilmelidir. İşte kimlik doğrulama işleminde kullanılacak bir devre gerçeklenmesi ve kullanılabilirliğinin araştırılması tezin konusunu oluşturmaktadır.

2. FİZİKSEL TEK YÖNLÜ FONKSİYON

Tek yönlü fonksiyonlar açık anahtarlı kriptolojinin dayanak noktasıdır [1]. Tek yönlü fonksiyon kavramı kendini çok pratik bir bağlamda göstermiştir. Birden çok kullanıcı hesabı içeren bir bilgisayar sistemi düşünün. Hesap oluşturulduğunda kullanıcı, sistemin şifre dosyasına kaydedilecek bir şifre seçer. Her başarılı giriş işlemi için kullanıcıya şifre sorulur ve alınan şifre kayıtlı olan ile karşılaştırılır. Kötü niyetli kimselerin şifreyi elde etme ihtimaline karşı şifre sır olarak saklanmalıdır. Bu durumda kullanıcı doğrulama sisteminin güvenliği şifre dosyasının güvenliğine bağlıdır. Needham, gerçekten şifreyi bilmeden kimlik doğrulama işleminin yapılabileceğini fark etti. Onun sistemine göre bir kullanıcı ilk şifre PW'yi girdiğinde, sistem otomatik olarak $f(PW)$ fonksiyonunu hesaplar ve şifre yerine bu değeri kaydeder. Bir kullanıcı başarılı bir giriş işlemi gerçekleştirmek istediğinde şifre X'i girer. Bilgisayar $f(X)$ ve $f(PW)$'yi karşılaştırır ve eşitse giriş gerçekleşir. Burada önemli nokta şudur: f fonksiyonu tek yönlü fonksiyon ise herhangi bir argüman için hesaplaması kolay fakat tersini bulmak oldukça zordur. Böylece f ve $f(PW)$ bilinse dahi şifreyi hesaplamak neredeyse imkansızdır.

Basit bir şekilde ifade etmek gerekirse; "Tek yönlü fonksiyon, hesaplanmsı kolay fakat tersinin bulunması zor olan bir fonksiyondur."

Birden fazla tek yönlü fonksiyon ailesi mevcuttur. Örneğin mod alma işlemi bu tip bir fonksiyondur.

$$f(x) = x \,\%\, 8 \tag{2.1}$$

olarak tanımlanan bir fonksiyona 15 değerinin giriş olarak verildiğini düşünelim. Sonuç olarak 7 elde edilir. Bu durumda 7 sonucundan 15 giriş değerini bulmak oldukça zordur. Çünkü 8'e bölümünden 7 kalanı veren birçok sayı mevcuttur. Bu sayılardan hangisinin kullanıldığı, fonksiyonu ve sonucu bilen bir kişi için bile meçhuldür.

Bahsi geçen tek yönlü fonksiyonlar matematiksel nesnelerdir. Bunlar tek yönlü fonksiyonların algoritmik yönünü temsil eder. Ancak bazı fiziksel yapılar da vardır ki bu fonsiyonların özelliklerine sahiptir. İşte böyle yapılara fiziksel tek yönlü fonksiyon adı verilmektedir.

Bu durumda verilen basit ifade şu hale gelir; "Fiziksel tek yönlü fonksiyon, yapılması kolay fakat kopyalanması zor olan fonksiyondur."[1]

Projenin konusunu da fiziksel tek yönlü fonsiyon özelliğine sahip bir devre teşkil etmektedir. Devrenin açıklaması ve fiziksel tek yönlü fonksiyon özelliklerini nasıl sağladığı 3. ana başlık altında açıklanacaktır.

3. FİZİKSEL KOPYALANAMAZ FONKSİYON

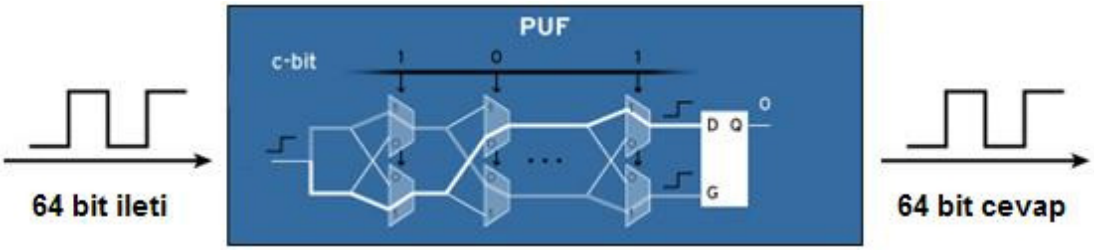
Fiziksel kopyalanamaz fonksiyon(Physical Unclonable Functions - PUF) devresi ileti(challange) girişleri ile cevap(response) çıkışları arasındaki ilişkiyi tayin eden ve fiziksel olarak gerçeklenen bir fonksiyondur [2]. Takip eden özellikleri ihtiva eder:

1. Gerçeklemesi kolaydır: fiziksel olarak kısa bir zamanda gerçeklenebilir.
2. Karakterize etmesi zordur: sınırlı bir zamana ve kaynağa sahip bir saldırgan rastgele seçilmiş ileti'ler ile önemsiz miktarda cevap bilgisi edinebilir.

PUF devreleri birçok fiziksel sistem ile gerçeklenebilirken, bu tezde kendine özgü gecikme ve zamanlama değerlerine sahip entegre tabanlı silikon PUF devrelerine odaklanılmıştır. Üretim koşullarının birbirine oldukça yakın olduğu durumlarda dahi entegreler arasında ciddi gecikme farkları oluşmaktadır [3]. Silikon PUF devreleri dijital sırlarını, entegre devreler içindeki transistör ve kabloların karmaşık gecikme karakteristiklerinden almaktadır. Bu karakteristikler üretim sürecinde oluşan rastgele değişkenlikten faydalanır. Böylece silikonun kendisine has olan sırların önceden tahmin edilmesi veya belli bir program dahiline tekrar edilmesi oldukça zor hale gelmektedir.

3.1. VERİ SEÇİCİ TABANLI PUF DEVRELERİ

Şekil 3.1 veri seçici(multiplexer) tabanlı bir PUF devresini göstermektedir.



Şekil 3.1: Veri Seçici Tabanlı PUF Devresi [4]

Şekil 3.1'deki devrede iki dizi veri seçici sırası görülmektedir. Bu seçicilerden üst üste gelenlerin seçici bitleri aynı girişe bağlıdır. Bu seçici bitlere bağlanan giriş dizisine ileti(challange) adı verilir. Örnek olarak verilen devrede ileti uzunluğu 64-bittir. İlk veri seçici çiftinin girişi ortak olup bu girişten devreye yükselen kenar verilmektedir. Yükselen kenar girişten sonra ikiye ayrılarak, ileti girişlerinin değerine göre çaprazlanan yollarda yarışır. İki yolda yarışan yükselen kenar, en son adımda pozitif kenar tetiklemeli D tipi flip floba ulaşır. İşaretin flip flobun girişlerine farklı zamanlarda ulaşması çıkışı belirler. Eğer işaret flip flobun veri girişine ilk önce gelirse çıkış lojik-1, saat girişine önce gelirse çıkış lojik-0 değerini alır.

Devrenin ileti girişleri değişken uzunlukta olabilirken, çıkış değeri 1-bit uzunluğa sahiptir. Bu durumda istenilen uzunlukta çıkış dizisi elde edebilmek için devre istenilen sayıda çalıştırılmalıdır.

Bu noktada veri seçici tabanlı PUF devresinin tezin ilk ana başlığında bahsedilen fiziksel tek yönlü fonksiyon ile bağlantısının incelenmesi yerinde olacaktır.

$$\begin{bmatrix} a_{1,1} & a_{1,2} & \cdots & a_{1n} \\ a_{2,1} & a_{2,2} & \cdots & a_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m,1} & a_{m,2} & \cdots & a_{m,n} \end{bmatrix}_{\text{mxn}}$$

Şekil 3.2: İleti Matrisi

Şekil 3.2'de, n-bitlik ileti dizileri ile m-bitlik cevap istendiğinde kullanılabilecek bir ileti matrisi görülmektedir. Basit matemetik hesaplamalar yapılırsa:

m-bit response $\Rightarrow$ $2m$ adet response üretilebilir

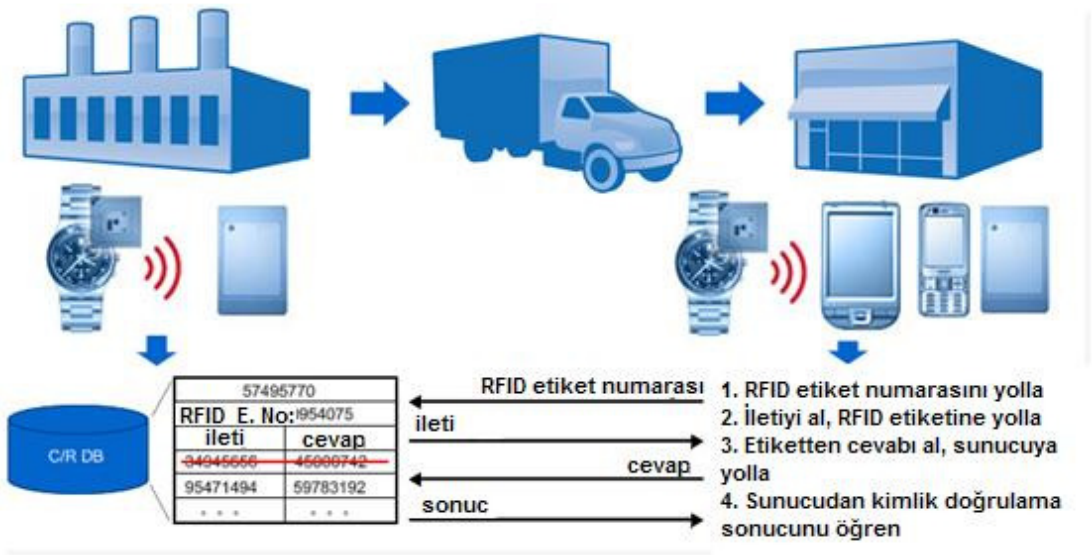
n-bit ileti $=>$ $2n$ adet ileti üretilebilir

$m\ x\ n$ boyutunda ileti matrisi $=>$ $2nm$ adet farklı ileti matrisi üretilebilir

Öyleyse $2nm\ /\ 2m\ =\ 2m(n-1)$ adetlik, aynı response dizisine karşılık gelen, ileti matris grupları oluşacaktır. Yani devrenin bilinen bir response dizisine karşılık birden fazla ileti matrisi denk gelecektir. Bu durum devrenin gerçeklediği fonksiyonun tersinin hesaplanmsını oldukça güç hale getirmektedir. Çünkü bilinen çıkışın elde edilebileceği $2m(n-1)$ adet giriş mevcuttur. Ancak tüm hesaplamalar alınan çıkış dizisinin eşit sayıda 1 ve 0 içerdiği varsayımı ile yapılmıştır.

3.2. PUF DEVRELERİ İLE KİMLİK DOĞRULAMA

PUF devreleri basit bir ileti-cevap tabanlı kimlik doğrulama işlemi sunar. Şekil 3.3'de gösterildiği gibi üretici ilk aşamada ürettiği PUF devresinden ileti-cevap çiftleri elde eder. Hangi ileti'ye hangi cevap'ın denk geldiğini liste halinde kaydeder. Bu liste daha sonra RFID okuyucusuna yüklenir.



Şekil 3.3: PUF Tabanlı RFID Etiketi İle Kimlik Doğrulama [4]

Sonraki aşamada devre RFID etiketine monte edilir. Her RFID etiketinin de bir kimlik numarası bulunmaktadır. Bu numara RFID etiketine monte edilen devrenin ileti-cevap listesinin de numarası olmaktadır. Son olarak RFID etiketi kullanılacağı ürün üzerine yerleştirilir ve kullanıma hazır hale gelir.

RFID etiketine sahip bir ürünün kimlik doğrulama işlemi sırasında ilk olarak etiket numarası okuyucuya yollanır. Okuyucu bu numara ile ilgili PUF devresinin üzerinde kayıtlı ileti-cevap listesine ulaşır. Etikete cevap üretmek üzere listeden seçtiği bir ileti yollar. İleti'yi alan etiket o an ürettiği cevap'ı okuyucuya geri yollar. Okuyucu aldığı cevap'ı kendi listesinde, yolladığı ileti'ye karşılık gelen cevap ile karşılaştırır. Eğer iki cevap eşit ise kimlik doğrulama işlemi başarılı bir şekilde gerçekleştirilmiş demektir. Aksi takdirde etiketin hizmet almaya yetkisinin olmadığı kabul edilir. Okuyucu kullanılan ileti-cevap çiftini bir daha kullanmaz.

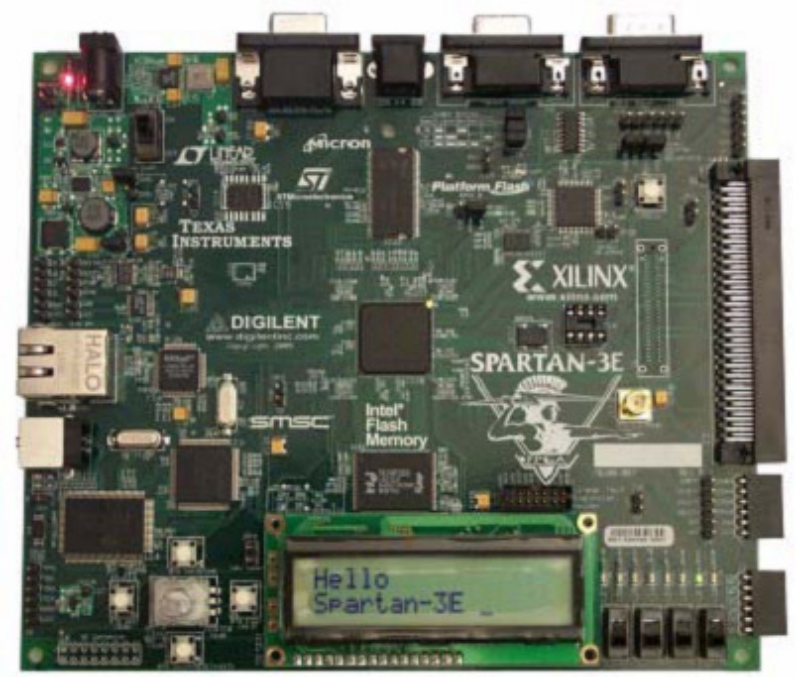
4. PUF DEVRESİNİN FPGA ÜZERİNDE GERÇEKLENMESİ VE ÇALIŞTIRILMASI

Sahada Programlanabilir Kapı Dizileri (Field Programmable Gate Array - FPGA) programlanabilen lojik (mantık) devre blokları ve ayarlanabilir ara bağlantıları içeren sayısal tümdevrelerdir [5]. Tasarımcının ihtiyaç duyduğu mantık işlevlerini gerçekleştirme amacına yönelik olarak üretilmiştir. Dolayısıyla her bir mantık bloğunun işlevi kullanıcı tarafından düzenlenebilmektedir. FPGA ile temel mantık kapılarının ve yapısı daha karmaşık olan devre elemanlarının işlevselliği artırılmaktadır. Alanda programlanabilir ismi verilmesinin nedeni, mantık bloklarının ve ara bağlantıların imalat sürecinden sonra programlanabilmesidir. Bu yönleri FPGA kullanımını oldukça avantajlı hale getirdiği gibi projede kullanılmasının da nedenlerini oluşturmaktadır.

Projede kullanılmak üzere tercih edilen kart SPARTAN 3E kartıdır. Devrenin gerçeklenmesi için VHDL donanım programlama dili ve tasarım ortamı olarak Xilinx 10.1 Desing Suite kullanılmıştır.

4.1. SPARTAN 3E KARTI

Bu bölümde üzerinde çalışılan ortamı kısa bir şekilde tanıtmak amaçlanmıştır. Spartan 3E kartı, üzerinde çeşitli uygulamalar gerçekleştirilmek için tasarlanmış çok fonksiyonel bir deneme kartıdır. Üzerinde bir VGA çıkışı, 2 adet RS 232 çıkışı, 1 adet 100 pinlik Hirose çıkışı, 12 adet programlanabilir giriş/çıkış pini, 4 adet sürgülü anahtar, 4 adet bas-çek tuş, 8 adet led, 1 adet çevirmeli ve bas-çek tuş, Usb ve ethernet girişi, PS2 girişi, SMA anten girişi, saat osilatör girişi vs. bulunmaktadır [6]. Bunların yanında 1 adet 16x2 karakter LCD ekranı, 1'er adet de ADC ve DAC bulunmaktadır. Ayrıca 512 Mbir DDR SDRAM, 128 Mbit Flash PROM, 16 Mbit serial flash ve 50 MHz saat osilatörü de kartın üzerinde bulunanlar arasındadır.



Şekil 4.1: Spartan 3E Kartı [6]

Gerçeklenen devrede SPARTAN 3E kartının FPGA çekirdeğinin yanısıra üzerinde bulunan ledlerinden, tuşlarından ve seri iletişim portlarından yararlanılmıştır.

4.3. PUF DEVRE ELEMANLARININ GERÇEKLENMESİ

PUF devrenin gerçeklenmesi için iki elemana ihtiyaç duyulmaktadır. Bu elemanlar pozitif kenar tetiklemeli D tipi flip flop ile iki giriş bir çıkışlı veri seçici(multiplexer) elemanıdır. Bu bölümde bahsi geçen iki elemanın gerçeklemensinden bahsedilecektir.

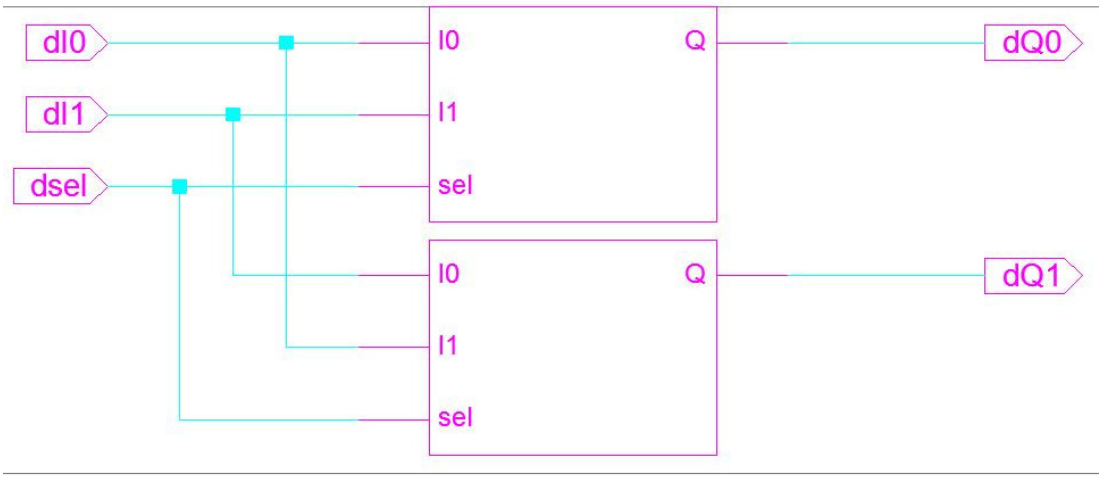
İkili veri seçici gerçeklenmesi:

İkili veri seçici, seçici bitleri aynı olan iki adet veri seçici elemanının aynı modüle yerleştirilmesi ile oluşturulan bir modüldür. PUF devresinin zincir yapısı, arka arkaya dizilen veri seçici elemanları ve çapraz bağlantılar devrenin gerçeklenmesinde oldukça fazla karmaşıklık meydana getirmekte idi. Bu karmaşıklığın önüne geçilmek için ikili veri seçici tasarlanmıştır.

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_ARITH.ALL;
use IEEE.STD_LOGIC_UNSIGNED.ALL;

entity doublemux is
   port( dI0, dI1, dsel : in std_logic;
         dQ0, dQ1 : out std_logic);
end doublemux;

architecture Behavioral of doublemux is

   component mux2to1 is
      port( I0, I1, sel : in std_logic;
            Q : out std_logic);
   end component mux2to1;

begin
mux1 : mux2to1
   port map(dI0, dI1, dsel, dQ0);

mux2 : mux2to1
   port map(dI1, dI0, dsel, dQ1);

end Behavioral;
```

Şekil 4.2: İkili Veri Seçici İçin Yazılan Kod

Şekil 4.2'de başka bir modülde gerçeklenen veri seçici elemanının "component" olarak eklemesi ve çapraz bağlantıların yapıldığı kod satırları görülmektedir. Bu satırlarda bir kere yapılan çapraz bağlantılar, zincir yapıda birçok defa tekrarlanma zorunluluğundan kurtulmuştur. Verilen kodların gerçeklediği yapı ise şekil 4.3'de görülmektedir.

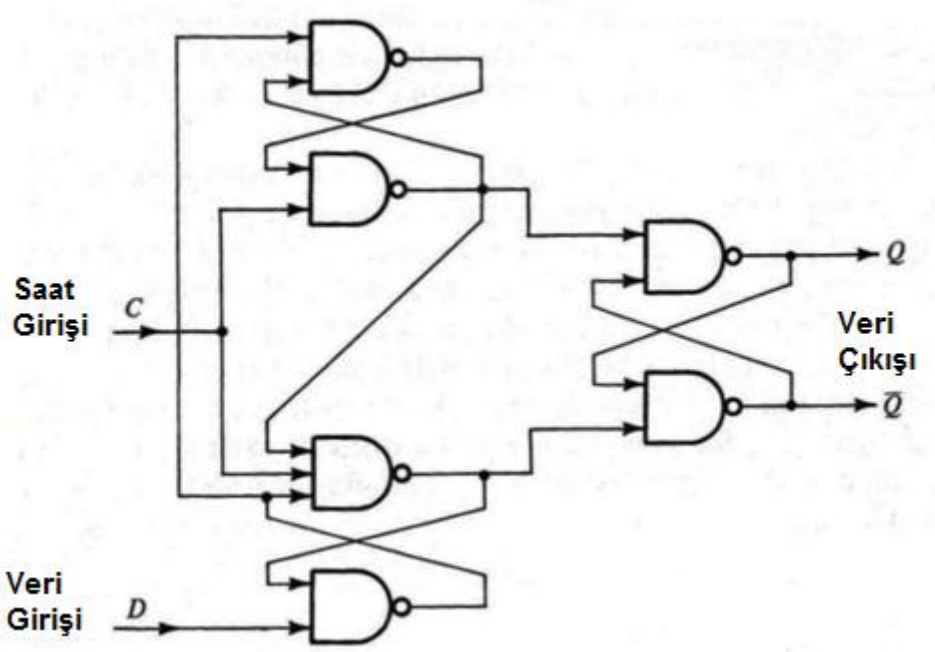


Şekil 4.3: İkili Veri Seçicinin RTL Şeması

Pozitif Kenar Tetiklemeli D Tipi Flip Flop:

Bu bölümde FPGA'de bulunan lojik yapılardan Look Up Table(LUT) yapıları ile pozitif kenar tetiklemeli D tipi flip flop gerçeklemesinin adımları anlatılacaktır. Flip flobun neden LUT yapıları ile gerçeklendiğine Karşılaşılan Problemler bölümünde değinilecektir.

Gerçeklenen pozitif kenar tetiklemeli D tipi flip flop yapısı şekil 4.4'de gösterilmiştir.

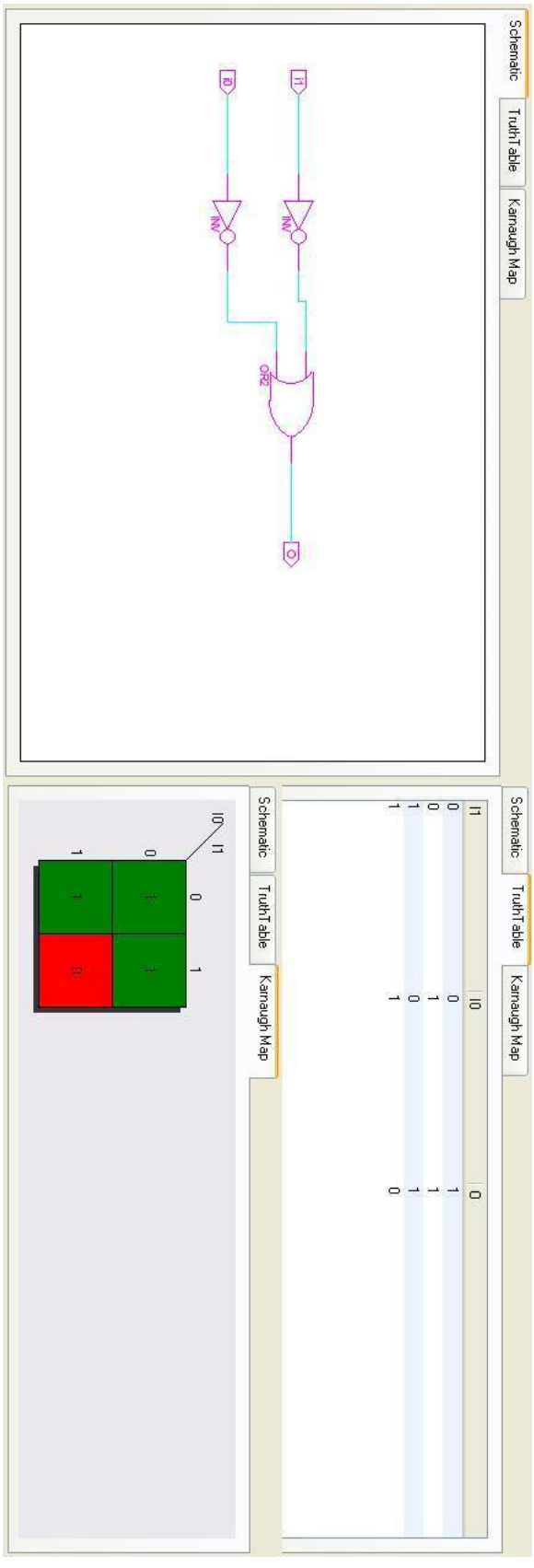


Şekil 4.4: Pozitif Kenar Tetiklemeli D Tipi Flip Flop [7]

Şekil 4.4'de verilen yapıyı gerçeklemeye LUT ile NAND kapısı gerçeklemekle başlandı. NAND kapısı gerçeklemek için yazılan kod örneği ve gerçekleme sonucunda oluşan yapı şekil 4.5 ve şekil 4.6'da görülmektedir. Sonraki aşamada sadece kapıların birbirleriyle olan bağlantıları yapılarak işlem tamamlanmıştır.

```
nand_lut1 : LUT2
generic map (
    INIT => X"7")
port map (O => s3,
          I0 => s1,
          I1 => s2);
```

Şekil 4.5: LUT İle NAND Kapısı Gerçekleme



Şekil 4.6: NAND Kapısının Yapısı, Lojik İfadesi

4.3. GENERATE İFADESİ İLE DEĞİŞKEN YAPI GERÇEKLENMESİ

PUF devresinin ileti girişinin değişkenlik göstermesi tekrar tekrar devre gerçekleme yükünü de beraberinde getirmekte idi. Bu durumdan kurtulmak için devreyi değişken şekilde gerçekleme yoluna gidilmiştir. Bu yöntem için VHDL dilinin "generate" ifadesinden yararlanılmıştır.

İki çeşit "generate" ifadesi bulunmaktadır [8].

1) For...generate
2) If...generate

For...generate ifadesi:

For...generate ifadesi dizi şeklinde veya düzgün dizilmiş yapıların gerçeklenmesinde kullanılmaktadır. İfadenin sözdizimi aşağıdaki gibidir.

label: for *identifier* in *range* generate

{ *concurrent_statement* }

end generate [*label*] ;

• "label", ifade için gerekli olan isimdir ve içiçe geçmiş generate ifadelerinde kullanışlılık sağlar.

• "identifier", for...generate ifadesine özgüdür. Başka bir yerde tanımlanmasına gerek yoktur, ifade tarafından otomatik olarak tanımlanır. Bu nesnenin değeri ancak ifade içinde geçerlidir. İfade dışında okunamaz ya da yeni değer atanamaz.

• "range" aşağıdaki iki formadan birinde olmalıdır.

integer_expression to *integer_expression*

integer_expression downto *integer_expression*

If...generate ifadesi:

If...generate ifadesi dizi şeklinde veya düzgün dizilmiş yapıların gerçeklenmesinde koşula bağlı olarak kullanılmaktadır.

İfadenin sözdizimi aşağıdaki gibidir.

label: if *expression* generate

{ *concurrent_statement* }

end generate [*label*] ;

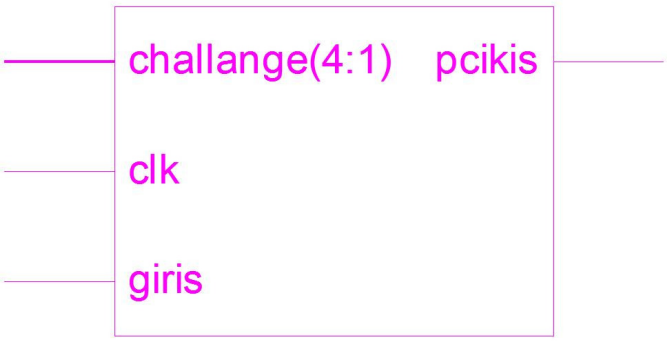
• For...generate ifadesinin sözdiziminden farklı olarak “expression” bulunmaktadır. “Expression” veri türlerinden Boolean değere sahip olmalıdır.

Takip eden şekilde generate ifadesinin PUF devresini gerçeklemede kullanımı verilmiştir.

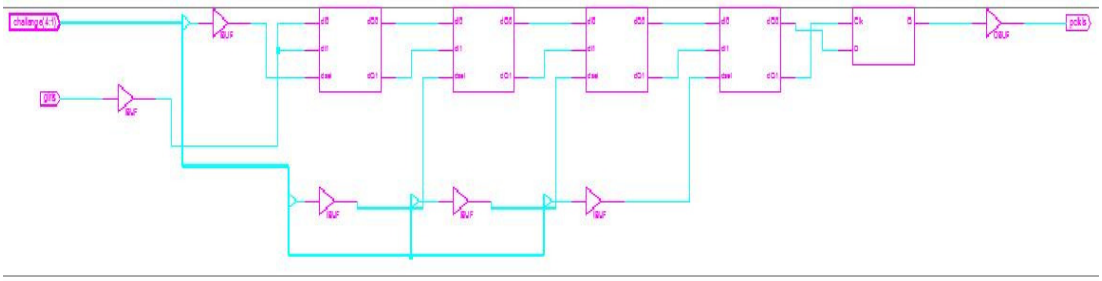
```
L1: for i in 1 to n generate

-- ilk mux çifti
   first : if(i = 1) generate
      inst_first : doublemux
         port map(sign(i-1),sign(i-1),challenge(i),sign(i),sign2(i));
   end generate first;

-- son mux çifti
   last : if(i = n) generate
      inst_last : doublemux
         port map(sign(i-1),sign2(i-1),challenge(i),sign(i),sign2(i));
   end generate last;

-- ortada kalan mux çiftleri
   general : if(i > 1 and i < n) generate
      inst_general : doublemux
         port map(sign(i-1),sign2(i-1),challenge(i),sign(i),sign2(i));
   end generate general;

end generate L1;
```

Şekil 4.7: Generate İfadesinin Kullanımı

Verilen kodların gerçeklediği 4-bit challange uzunluğuna sahip devre şeması şekil 4.8 ve şekil 4.9’daki gibidir.



Şekil 4.8: 4-bit Uzunluklu PUF Devresi Dış Görünümü



Şekil 4.9: 4-bit Uzunluklu PUF Devresi İç Görünümü

4.5. PUF DEVRESİNİN ÇALIŞTIĞININ GÖZLENMESİ

PUF devresinin çalıştığının gözlenmesi için devre,Spartan 3E kartına yüklenmiş ve kart üzerindeki donanımlar ile test edilmiştir.

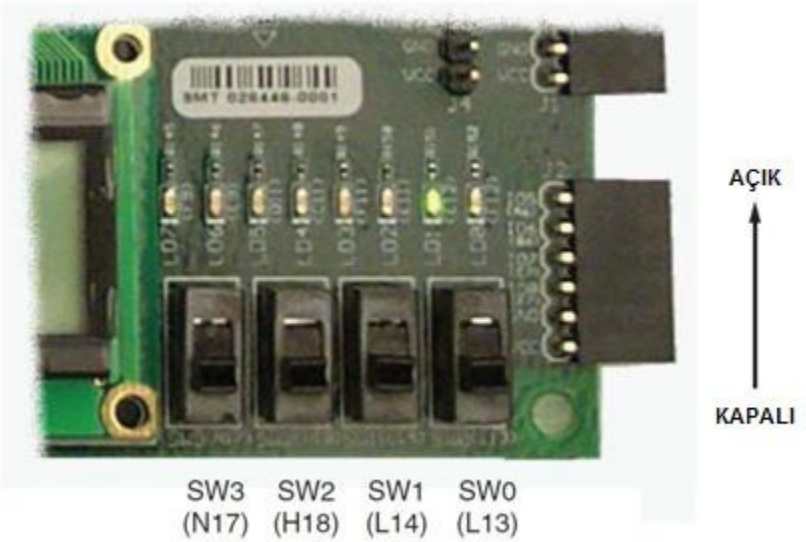
İlk adım olarak "user constraints file" oluşturuldu. User constraints file (UCF) zamanlama ve yer kısıtları tuttan bir ASCII dosyasıdır [9]. Bu dosya sayesinde devrenin giriş ve çıkışlarının kart üzerinde test için kullanılacak donanımlar ile bağlantısı gerçekleştirilmiştir. Şekil 4.10'da oluşturulan UCF dosyasının içeriği verilmiştir.

```
NET "clk" LOC = "C9" | IOSTANDARD = LVCMOS33 ;

NET "chl<0>" LOC = "L13" | IOSTANDARD = LVTTL | PULLUP ;
NET "chl<1>" LOC = "L14" | IOSTANDARD = LVTTL | PULLUP ;
NET "chl<2>" LOC = "H18" | IOSTANDARD = LVTTL | PULLUP ;
NET "chl<3>" LOC = "N17" | IOSTANDARD = LVTTL | PULLUP ;

NET "cikis" LOC = "F12" | IOSTANDARD = LVTTL | SLEW = SLOW | DRIVE = 8 ;

NET "giris"  LOC = "H13" | IOSTANDARD = LVTTL | PULLDOWN ;
```

Şekil 4.10: UCF Dosyası

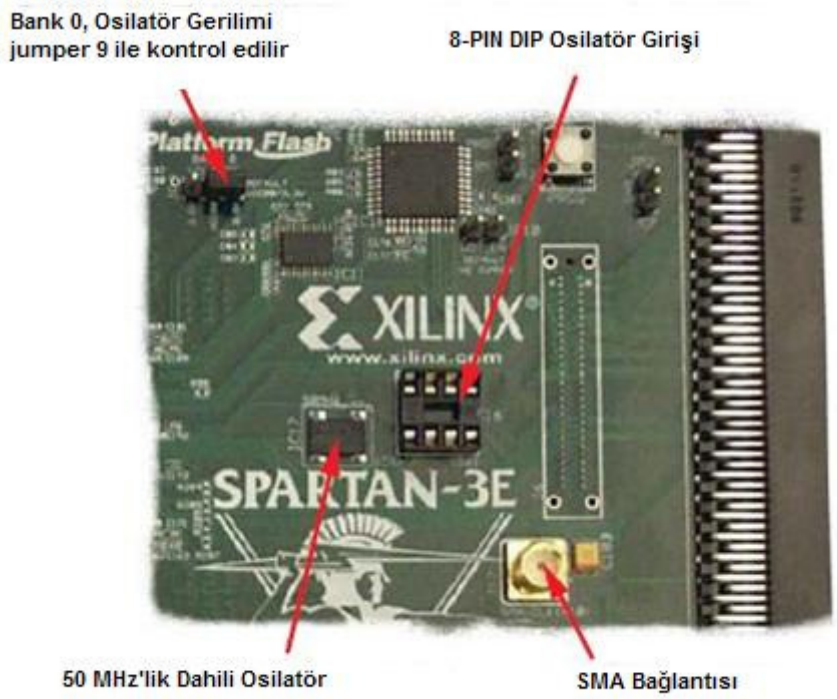
Verilen UCF dosyası ile devre giriş ve çıkışlarına hangi donanımların bağlandığı madde madde sıralanmıştır.

1. "challange(4:1)" girişlerine kart üzerinde bulunan kayar tuşlar bağlanmıştır.



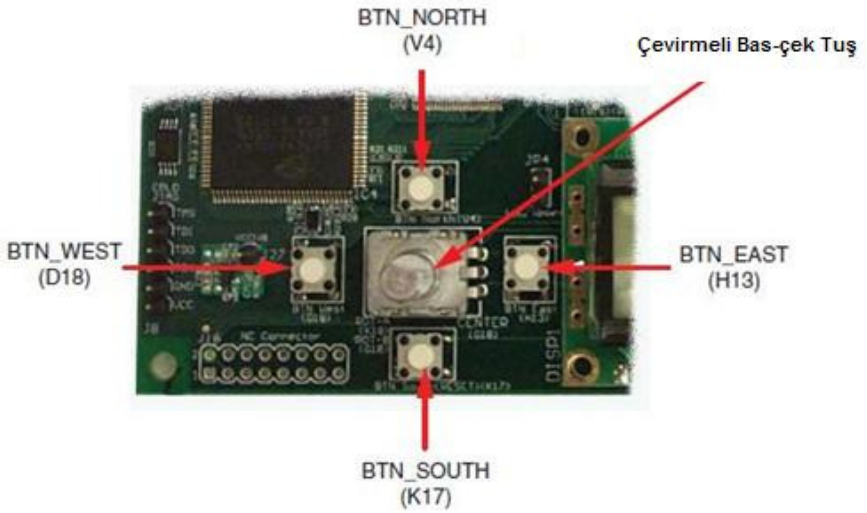
Şekil 4.11: Spartan 3E Üzerinde Kayar Tuşlar [6]

2. "clk" girişine kart üzerinde bulunan 50 MHz'lik saat bağlanmıştır.



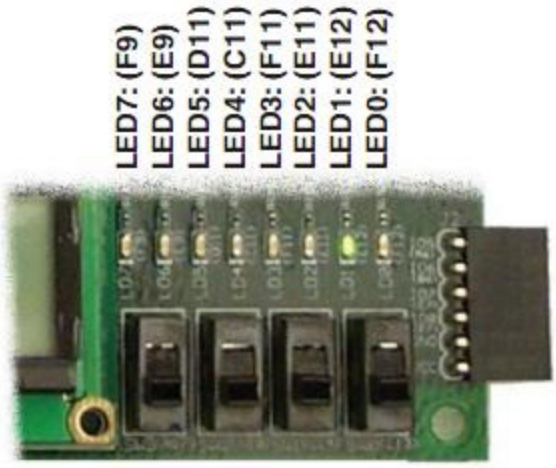
Şekil 4.12: Spartan 3E Dahili Saat Üreteci [6]

3. "giriş" isimli girişe çevirmeli bas-çek tuş bağlanmıştır. Ancak sadece bas-çek işlevi kullanılmıştır.



Şekil 4.13: Spartan 3E Çevirmeli Bas-çek Tuş [6]

4. Devrenin tek çıkışı olan “pcikis” isimli çıkışa LED7 isimli led bağlanmıştır.



Şekil 4.14: Spartan 3E Üzerindeki Ledler [6]

UCF dosyasının da eklenmesiyle ikinci adıma geçilmiş, FPGA içine yüklenecek (.bit) uzantılı programlama dosyası oluşturulmuştur. Bu dosya Xilinx ISE yardımı ile karta yüklenmiştir.

Son adımda devre test edilmiştir. Test aşamasında kayar anahtarlardan ileti girişleri ayarlanmış, çevirmeli bas-çek tuştan da yükselen kenar verilmiştir. Farklı ileti girişleri için farklı sonuçlar çıktığı, yeniden yükselen kenar verilene kadar çıkan sonucun değişmediği led üzerinden gözlemlenmiştir.

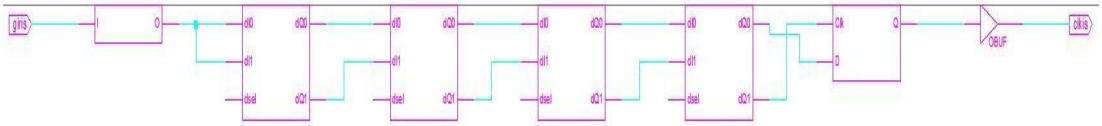
4.5. KARŞILAŞILAN SORUNLAR

Bu bölümde PUF devresinin gerçeklenerek düzgün bir şekilde çalıştırılması aşamalarında karşılaşılan zorluklardan bahsedilecektir. Çünkü yeni problemlerle yüzleşmek ve bu problemleri çözüme kavuşturmak mühendisliğin temel taşları olduğu gibi bu projenin değer kazanmasının da ana unsurlarıdır.

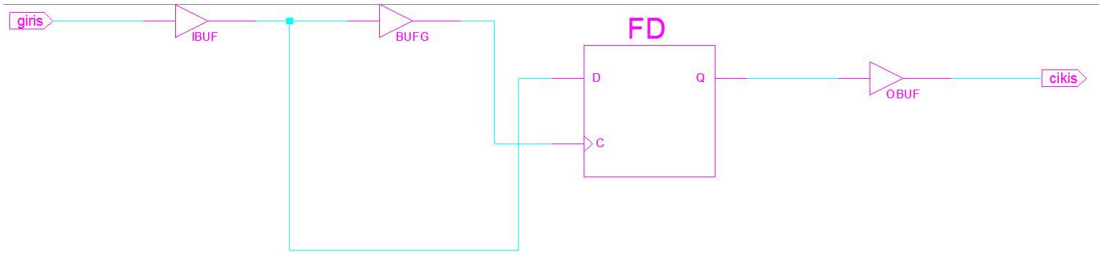
Keep Hierarchy:

Projenin ilk aşamasında PUF devresi VHDL dili kullanılarak gerçekleştirilmişti. Ancak bir türlü istenilen sonuç elde edilemedi. Zincir yapıda bir devre beklenirken çok daha kısa bir devre şeması ile karşılaşılıyordu. Şekil 4.15 ve şekil 4.16’da 4-bit

giriş uzunluğuna sahip devrenin olması gereken ve beklenmeyen durumdaki şemaları verilmiştir.

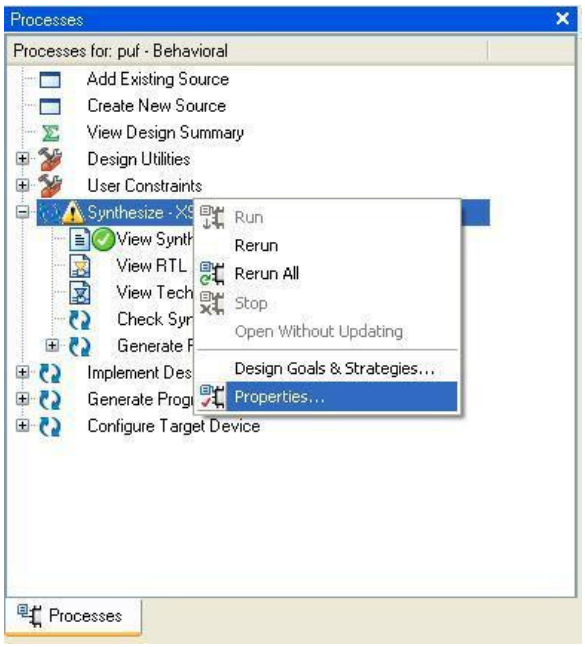


Şekil 4.15: PUF Devresinin Beklenen Şeması

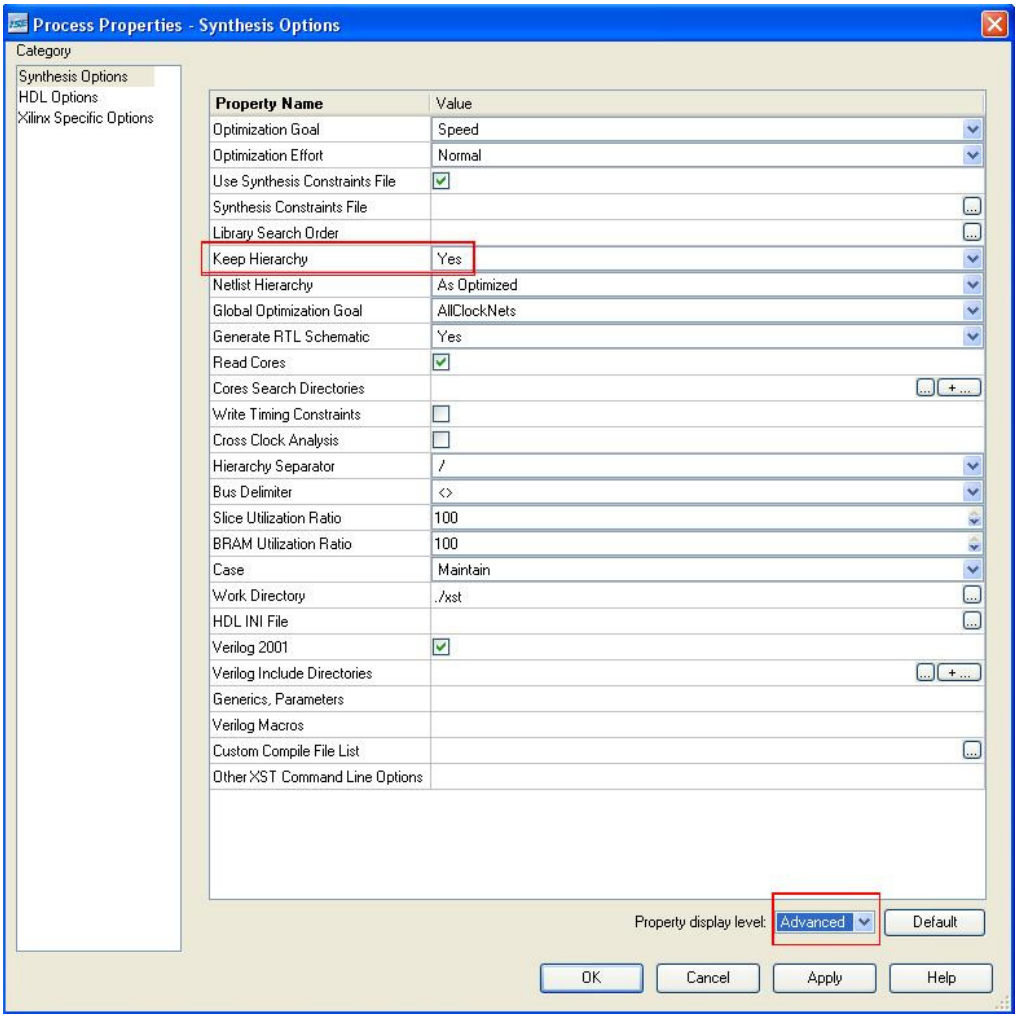


Şekil 4.16: PUF Devresinin Olmaması Gereken Şeması

Bu durumda sorunun yazılan kodlarda olduğu düşünüldü ancak yapılan değişiklik ve düzeltmelerden bir sonuç alınamadı. En sonunda problemin Xilinx ISE programının sentez ayarlarından kaynaklandığı ortaya çıktı. Xilinx ISE programı yazılan kodu optimize ederek sentezlemekte idi. Bu durumu düzeltmek için programın *Processes* penceresinden *Syntesize-XST / Properties* kısmına ulaşılır. *Property Display Level* ayarı *Advenced* seçililerek *Keep Hierarchy* ayarının açılması sağlanır. Son olarak *Keep Hierarchy Yes* seçilerek işlem tamamlanır. Aşağıda bahsedilen ayarın yapılışı gösterilmektedir.



Şekil 4.17: Xilinx ISE Sentez Ayarlarına Giriş



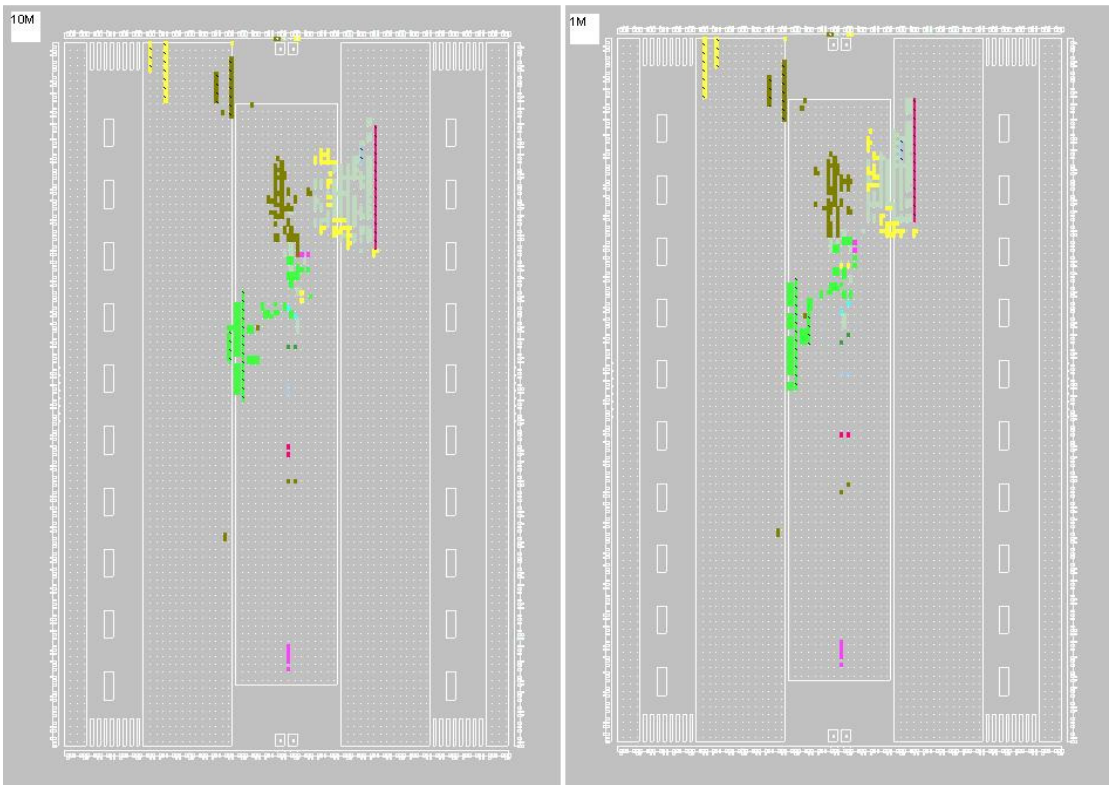
Şekil 4.18: Xilinx ISE "Keep Hierarchy" Ayarı

D tipi flip flop ve saat girişi:

Gerçeklenen PUF devresinin ilk halinde devrenin sonunda bulunan D tipi flip flop davranışsal olarak tanımlanmıştı. Bu durumda flip flobun saat girişine FPGA'in kendi saat işareti bağlanmakta ve devrenin girişine verilen işaretin yarışacağı iki yoldan biri devre dışı kalmaktaydı. Sorunun önüne geçebilmek için FPGA'in lojik bloklarından Look Up Table (LUT)'lar ile pozitif kenar tetiklemeli D tipi flip flop tasarlanmıştır.

Devrenin FPGA Üzerinde Yerleşimi:

Gerçeklenen devrenin FPGA'in üzerine nasıl yerleştirileceği Xilinx ISE programı tarafından belirlenmektedir. Bu durum devrenin çalışmasını etkilemektedir. Şöyle ki; bazı yerleştirme işlemlerinde devrenin sonundaki flip flobun iki girişinden birine çok uzun bir kablo, diğerine çok kısa kablo gelebilmektedir. Bu da devrenin gecikmelere dayanan yapısını etkilemektedir. Aynı durum veri seçiciler arasında da görülmekte, bazı durumlarda devrenin sürekli lojik-1 ya da lojik-0 verebilmesini sağlamaktadır. Şekil 4.19'da devrenin çıkış üretmesi için beklenen sürenin değiştirilmesinin bile devre yerleşimini (özellikle sarı ve yeşil bölgeler) etkilediği görülmektedir.



Şekil 4.19: Gecikme Değerinin Değişmesiye Yerleşimde Oluşan Farklılık

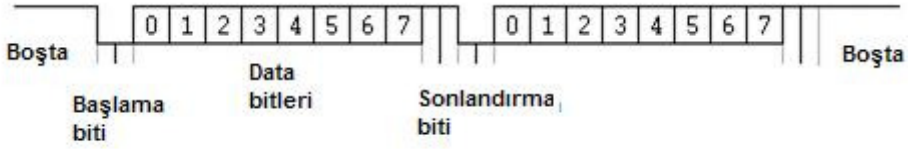
5. ÖLÇÜM DÜZENEĞİ VE ÖLÇÜMLER

Devrenden ölçüm alma işlemi girişine verilen farklı ileti dizilerine karşılık çıkış dizisi alma şeklindedir. Oldukça fazla sayıda ileti girilecek olması ölçümlerin elle yapılmasını büyük bir yük haline getirmekte idi. Bu problemden kurtulmak için hem ileti girişlerine değer yollama hem de çıkıştan değer okuma işlemini gerçekleştirebilecek MATLAB programından yararlanılmıştır. MATLAB ile FPGA kartı arasında ise seri iletişim kurularak kolay veri alış verişi sağlanmıştır.

5.1. MATLAB İLE KART ARASINDA SERİ İLETİŞİM KURULMASI

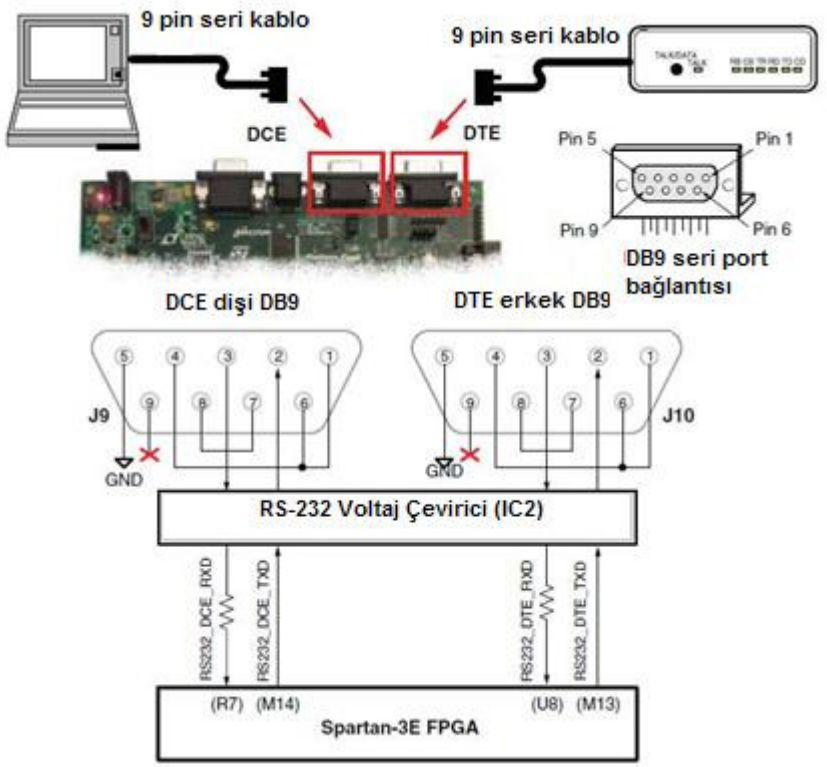
Bu bölümde seri iletişim hakkında kısa bir bilgi verilecek arkasından bu haberleşmenin gerçekleşmesi için yapılan işlemler anlatılacaktır.

Asenkron seri haberleşme protokolü şekil 5.1'de görselleştirilmiştir. Gönderici bekleme durumunda 1 olan hattı 0'a çektiğinde alıcı start bitini algılar. Takip eden 8 bit veri bitleridir ve LSB'den MSB'ye doğru yollanır. Sekizinci data bitinin gönderilmesini takiben gönderici hattı tekrar 1'e çeker. Alıcı bunu stop biti olarak algılar ve gelen bir baytlık veriyi servis etmeye hazır olur.



Şekil 5.1: Asenkron Seri Haberleşme Protokolü [10]

Şekil 5.2'de ise Spartan 3E kartı üzerindeki seri haberleşme donanımı ve bu donanımın bağlantı şeması görülmektedir.



Şekil 5.2: Spartan 3E Seri İletişim Donanımı [6]

5.1.1. ANA MODÜLE UART MODÜLLERİNİN EKLENMESİ

Projede asenkron seri iletişim kurulması için Verilog donanım programlama dilinde yazılmış hazır UART modülleri kullanılmıştır. Modüllerin ayrıntıları ve çalışma yöntemleri şu şekildedir:

1. UATv12: Evrensel Asenkron Verici [11]:

Şekil 5.3: UATv12 Modülü [11]

Özellikleri:

• Evrensel asenkron seri haberleşme protokolüne uygun olarak çalışır.

• Yalnızca bir bayt veri tamponu mevcuttur.

• İstenen haberleşme frekansında çalışması ayarlanabilir.

• Bayt gönderiliyorken meşgul işareti üretir.

Fonksiyonel Tanımı:

TxD girişi haberleşme hattına bağlanır. fbol_orani girişine uygulanacak değer ile UATv12'nin çalışacağı haberleşme frekansı ayarlanır. Üst devre tarafından bayt_gonder kesmesi geldiğinde , dataT'deki veriyi örnekler, start bitini oluşturur ve ardından baytı TxD'den seri olarak gönderir. En son veri biti MSB'dir. Bunun ardından stop bitini yollar. Bu esnada, meşgul olduğunu bildiren bayt_gonderiliyor çıkışı lojik 1'de tutulur. Stop bitini de gönderdikten sonra bayt_gonderiliyor'u 0'a çeker. Üst devre yeni bir bayt gönderme komutu vermek için bayt_gonderiliyor'un 0'a çekilmesini beklemek zorundadır. UAT, meşgul iken bayt_gonder kesmesinin ve dataT paralel bayt girişinin değişimine duyarsızdır.

Giriş / Çıkışları:

Tasarımda beş giriş iki çıkış bağlantısı vardır. Bu bağlantılar aşağıda detaylı olarak açıklanmıştır.

• clk: saat işareti.

• reset: Senkron aktif sıfırdır. Alt modüllere de ulaşması için en az bir clk periyodu boyunca sürülmelidir.

• TxD: Haberleşme kanalının gidiş yönlü hattına bağlanır. Seri işaret çıkışıdır.

• fbol_orani: 16 bit genişlikli frekans belirleyici parametredir. Tablo 5.1'de verilen örnek değerler ile sürülebilir.

• dataT: 8 bit (1 bayt) genişlikli gidecek veri girişidir. Üst devre bayt_gonder kesmesini göndermeden önce bu girişe gidecek baytı yazmalıdır.

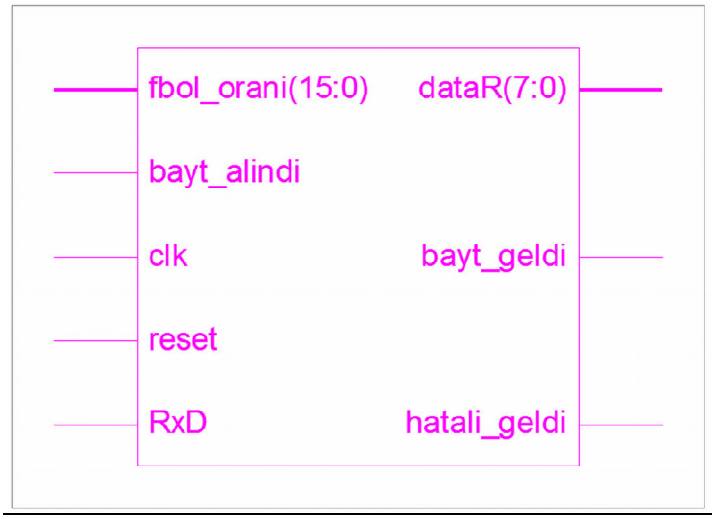
• bayt_gonder: Üst devre tarafından sürülen bayt gönderme işlemini tetikleyen işarettir. UAT, mesgul değil iken 1 algıladığında bayt göndermeye baslar.

• bayt_gonderiliyor: Bir bayt veri gönderilirken lojik 1 olan meşgul çıkışıdır.

Tablo 5.1: UART Modüllerinin Çalışma Hızlarına Göre Frekans Bölme Oranları [11]

Çalışma (clk) frekansı	Haberleşme hızı	fbol_orani değeri	
25 MHz	9600 bps	16'h0515	= 1301_{10}
25 MHz	460800 bps	16'h001A	= 26_{10}
50 MHz	38400 bps	16'h028A	= 650_{10}
50 MHz	57600 bps	16'h01B1	= 433_{10}
100 MHz	9600 bps	16'h1457	= 5207_{10}
100 MHz	38400 bps	16'h0515	= 1301_{10}
100 MHz	57600 bps	16'h0363	= 867_{10}
100 MHz	115200 bps	16'h01B1	= 433_{10}

2. UARv12: Evrensel Asenkron Alıcı [10]:



Şekil 5.4: UARv12 Modülü [10]

Özellikleri:

• Evrensel asenkron seri haberleşme protokolüne uygun olarak çalışır.

• Yalnızca bir bayt veri tampon mevcuttur. Her bayt gelişinden sonra tampon boşaltılmalıdır.

• İstenen haberleşme frekansında çalışması ayarlanabilir.

• Veri alındığında kesme üretir.

• Yanlış stop biti aldığında hata kesmesi üretir.

Fonksiyonel Tanımı:

RxD girişi haberleşme hattına bağlanır. İçerdiği ASGIRISv10 modülü sayesinde RxD hattı saat işaretine senkron hale getirilir. fbol_orani girişine uygulanacak deger ile UARv12'nin çalısacağı haberleşme frekansı ayarlanır. RxD'yi sürekli dinleyen alıcı anlamlı bir bayt veri geldiğinde bayt_geldi çıkışını 1 yapar. Üst devre bu çıkışla birlikte dataR'deki veriyi okuyabilir. USRv12'nin yeni bir baytı yakalayabilmesi için üst devrenin bayt_alindi girişine 1 vermesi ve UARv12'yi serbest bırakması gerekmektedir. RxD'den gelen verinin stop biti algılanamazsa hatali_geldi çıkışını 1 yapar. Bu durumda üst devre yine bayt_alindi girişine 1 uygulayarak UARv12'yi yeni veri için serbest bırakmalıdır. Serbest bırakmanın gerçekleşmesi için üst devre bayt_alindi işaretini, bayt_geldi ve hatali_geldi çıkışlarının 0 olmasına dek, 1 olarak sürmelidir.

Giriş / Çıkışları:

Tasarımda beş giriş üç çıkış bağlantısı vardır. Bu bağlantılar asağıda detaylı olarak açıklanmıştır.

• clk: saat işareti.

• reset: Senkron aktif sıfırdır. Alt modüllere de ulaşması için en az bir clk periyodu boyunca sürülmelidir.

• RxD: Haberleşme kanalının geliş yönlü hattına bağlanır. Asenkron seri işaret girişidir.

• fbol_orani: 16 bit genişlikli frekans belirleyici parametredir. Tablo 5.1'de verilen örnek değerler ile sürülebilir.

• bayt_alindi: Üst devre tarafından gelen baytı veya hatalı alımı yakalayan UAR'ı serbest bırakma işareti girişidir. Kesme isaretleri 0'a düşünceye kadar 1 olarak sürülmeli, kesme gelmeden de 1'e çekilmemelidir.

• bayt_geldi: Bir bayt veri alındığında lojik 1 yapılan kesme çıkışı.

• hatali_geldi: Gelen baytın stop bitinde hata olduğunda lojik 1'yapılan kesme çıkışı.

• dataR: 8 bit (1 bayt) genişlikli gelen veri çıkışı. Üst devre bayt_geldi kesmesini aldıktan sonra bu çıkışı okumalıdır.

5.1.2. SERİ İLETİŞİM İÇİN MATLAB KODLARININ YAZIMI

Bir önceki bölümde seri haberleşmenin FPGA tarafı anlatılmıştır. Haberleşmenin diğer tarafında da MATLAB programı bulunmaktadır. MATLAB programının seri port heberleşmesi için hazır fonksiyonlarının bulunması diğer tarafa kıyasla oldukça kolaylık sağlamıştır. Şekil 5.5'de 8-bitlik iletişim için kullanılan örnek bir MATLAB kodu verilmiştir.

```
close all;
clear all;

s = serial('COM5','BaudRate',38400);
fopen(s);
flushinput(s);
readasync( s, 1);

for i=1:1:255
fwrite( s, i, 'async' );

while( s.TransferStatus(1) ~= 'i')
end

while s.BytesAvailable < 1;
end

if s.BytesAvailable == 1
    cevap7(i) = fread( s, 1);
end

end

stopasync(s);
fclose(s);
delete(s);
clear s;
```

Şekil 5.5: 8-bit Seri İletişim İçin Yazılmış MATLAB Kodu

Verilen kodda ilk önce çalışma ortamı temizlenmektedir. Ardından "serial" fonksiyonu ile bir seri port nesnesi oluşturulmaktadır. Fonksiyonun parametreleri kaçıncı porttan ve hangi hızda iletişim yapılacağını belirlemektedir. Nesne iletişime açıldıktan sonra giriş tamponu gelecek veriler için boşaltılmaktadır. Daha sonra iletişimin asenkron olacağı belirtilmiştir. Programın ana döngüsünde program sayacı her döngüde 8-bitlik ifade olarak yollanmakta ve cevap beklenmektedir. Cevap geldiğinde "cevap" isimli dizinin içine kaydedilmektedir. Cevap yalnızca 1-bitliktir.

Böylelikle döngü tamamlandığında 1'den 255'e kadar birer birer artan değerler 8-bitlik ifadeler halinde yollanmış, bu ifadelere karşılık gelen 1'er bitlik cevaplar 255 boyutlu bir dizinin içerisine alınmış olmaktadır. Bahsi geçen işlem tamamlandığında asenkron iletişim sona erdirilmekte ve seri iletişim nesnesi silinmektedir.

5.2. ÖLÇÜMLER VE DEVRE UZUNLUĞUNUN BELİRLENMESİ

PUF devresinin giriş uzunluğunun belirlenmesinde birkaç parametre rol oynamaktadır. Bu devrelerin RFID etiketleri üzerinde kullanılacağı düşünülürse küçük ve az güç tüketmesi için küçük bir giriş uzunluğuna sahip olması gerektiği düşünülebilir. Diğer taraftan söz konusu güvenlik olduğunda devre uzunluğunun arttırılarak karmaşıklığın arttırılması gerektiği de söylenebilir. Ancak devre uzunluğunu belirleyen en önemli etken devrenin dayandığı teoride yatmaktadır. Bölüm 3.1'de m uzunluklu cevap ve n uzunluklu ileti için düşünüldüğünde $2^{m(n-1)}$ 'lik matris gruplarının oluşacağı belirtilmişti. Ancak bu durum m uzunluklu ileti'nin yarı yarıya 1 ve 0'dan oluştuğu şart altında gerleşmektedir. PUF devresinin yapısı ise çıkışın eşit sayıda 1 ve 0 içereceği konusunda garanti vermemektedir. Düzgün dağılım olmadığı durumda fonksiyonun tersi hesaplanmaya kalkışılırsa her çıkışa karşılık düşebilecek ileti sayısı farklılık gösterecektir. Yani kullanılan ileti-cevap çiftlerinin güvenlik düzeyleri farklılık gösterecektir. Bu farklılığın büyük olması durumunda belki de bazı çiftlerin kullanılmaması gerekliliği ortaya çıkacaktır. Bahsedilen nedenlerden ötürü devrenin uzunluğunu belirlemek için 8-bitten başlayarak 16-bit, 32-bit ve 64-bit uzunluğundaki devreler için ölçümler alınmıştır. Alınan çıkış dizilerinin içinde en yakın 0-1 oranına sahip olan devre uzunluğu kullanılmaya en uygun devre uzunluğu olarak belirlenmiştir.

Tablo 5.2: Değişen Bit Uzunluğuna Göre Çıkıştaki 1 Sayıları

BİT SAYISI	**ALINAN ÖLÇÜM SAYISI**	**BİR SAYISI**
8-bit	256	212
16-bit	1000	507, 495, 498
32-bit	1000	836, 794, 817
64-bit	1000	758, 771, 762

Tablo 5.2'de alınan ölçüm sonuçları verilmiştir. Bu noktada ölçüm alınırken izlenen yöntemden bahsetmek yerinde olacaktır. Çünkü ölçüm alınırken 8-bit için bütün giriş olasılıklarını denemek sorun olmazken diğer bit uzunlukları için sıkıntı oluşmuştur.

Uart modülleri ile 8-bitlik iletişim yapılabilmesi yüksek bit uzunluklarında kaydırma işlemini devreye sokmuştur. Örneğin 16-bit için düşünülürse, ilk 8-bit alınmakta ve

yüksek anlamlı bitlere kaydırılmakta ardından düşük anlamlı bitlere yeni bir 8-bit alınmaktadır. Yollanan bu 8-bitlik sayılar ise MATLAB ortamında rastgele seçilerek yollanmıştır.

MATLAB ortamında rasrtgeleliğin sağlanması için "randperm" fonksiyonundan yararlanılmış, 1'den 255'e kadar değişen değerler içeren 1000 uzunluklu diziler oluşturulmuştur. Kaydırma işlemine gönderilen her 8-bitlik paket farklı dizilerden seçilmiştir. Yüksek bit uzunluğuna sahip devreler için bu diziler her seferinde yeniden üretilip tekrar ölçüm alındığında 1-0 oranının çok mühim olarak değişmediği görülmüş, böylece alınan ölçümlerin devre davranışı hakkında çok da yanıltıcı olmadıkları görülmüştür.

Ancak devre uzunluğu belirlemede bozucu etken sadece istatiksel hata değildir. Karşılaşılan sorunlar başlığı altında incelenen, devrenin FPGA üzerine yerleşimi de oldukça büyük bir bozucu etken olarak karşımıza çıkmaktadır. Bu şartlar altında bit uzunluğu arttıkça 1-0 oranı artar veya azalır gibi bir kanıya varmak mümkün değildir. Sadece şartlar dahilinde alınan ölçümler, 16-bit uzunluğuna sahip PUF devresinin oldukça yakın 1-0 çıkış değerine sahip olduğunu göstermektedir.

Sonuç olarak, projede gerçeklenen çeşitli uzunluktaki PUF devrelerinin arasında kullanılmaya en elverişli devrenin 16-bit uzunluğuna sahip olan PUF devresi olduğu anlaşılmıştır.

6. SONUÇ VE TARTIŞMA

Bu çalışmada Fiziksel Kopyalanamaz Fonksiyon'un, birçok alanda kullanımı gittikçe yaygınlaşan FPGA üzerinde tasarımı ve gerçeklenmesi yapılmıştır. Gerçeklenen veri seçici tabanlı devre deneme kartında test edilmiş ve çalıştığı gözlemlenmiştir. Basit yapısı ile birçok uygulama alanına entegre olabilecek potansiyele sahip bu devre, RFID sistemlerin ihtiyacı olan birçok özelliği de bünyesinde barındırmaktadır. RFID sistemler için az yer kaplama, az güç tüketme, hızlı çalışma gibi özelliklerin vazgeçilmez hale geldiği bir ortamda bu ihtiyaçların hemen hepsine cevap verebilecek Fiziksel Kopyalanamaz Fonsiyon gerçeklemesi güvenlik açısından oldukça büyük bir önem taşımaktadır.

Gerçeklenen devrenin giriş bit uzunluğu ile çıkışındaki 0-1 oranı arasındaki ilişkinin daha hassas olarak incelenmesi ve devrenin çalışmasının ortam koşullarından etkilenme seviyesinin saptanması devrenin kullanılabilirliğini belirlemede daha fazla yardımcı olabilecek çalışmalardır.

Kimlik doğrulama işlemleri, gündelik hayatın güvenliği açısından birçok yerde karşımıza çıkmaktadır. Fiziksel Kopyalanamaz Fonksiyonlar ise bu işleminin güvenliğini ve işlevselliğini arttırmaktadır. Sonuç olarak, bu fonksiyonların teoriden çıkarak pratik hayata uygulanmasını amaçlayan çalışma ile gündelik hayatın biraz daha güvenli hale gelmesi yönünde önemli bir adım atılmış oldu.

KAYNAKLAR

[1] **Ravikanth**, P.Srinivasa, 2001. Physical One-Way Functions, Massachusetts Institute of Technology, USA

[2] **Tuyls P., Batina L.**, 2006. RFID-Tags for Anti-counterfeiting, Springer, Berlin

[3] **Devadas, S., Suh, E., Paral, S., Sowell, R., Ziola, T., Khandelwal, V.,** 2008. Design and Implementation of PUF-Based "Unclonable" RFID ICs for Anti-Counterfeiting and Security Applications, *RFID, 2008 IEEE International Conference,* Las Vegas, 16-17 Nisan, s. 58-64

[4] http://www.verayo.com

[5] **Küçükgüzel, E.,** 2005. FPGA (Field Programmable Gate Array) lere Giriş, Kısa Tarihçe ve Örnek Uygulama, TOLA Elektronik, İstanbul

[6] **UG230**, 2006. Starter Kit Board User Guide

[7] http://www.plctalk.net

[8] **Xilinx, Inc.,** VHDL Reference Guide

[9] http://www.xilinx.com

[10] **Yeniçeri, R.**, 2009. UARv12:Evrensel Asenkron Alıcı, İTÜ Gömülü Sistem Tasarımı Laboratuvarı, İstanbul

[11] **Yeniçeri, R.**, 2009. UATv12:Evrensel Asenkron Verici, İTÜ Gömülü Sistem Tasarımı Laboratuvarı, İstanbul

ÖZGEÇMİŞ

1987 yılında Afyonkarahisar'da dünyaya gelen Mehmet SOYBALI ilkokulu Çanakkale'de, ortaokulu İzmir'de okudu. Lise öğrenimini Kütahya Fen Lisesi'nde yatılı şekilde tamamlayarak, 2005 senesinde İstanbul Teknik Üniversitesi Elektronik Mühendisliği bölümünü kazandı. Üniversite 3. ve 4. sınıfta İTÜ Gömülü Sistem Tasarımı Laboratuvarı'nda çalışarak sayısal sistem tasarımı, gömülü sistemler ve kriptoloji hakkında bilgi ve deneyim sahibi oldu.