T.C.
ERCİYES ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

FPGA TABANLI, PCI ARAYÜZLÜ, GERÇEK ZAMANLI RASGELE SAYI ÜRETECİ TEST SİSTEMİ TASARIMI VE UYGULAMALARI

Tezi Hazırlayan
Murat ERAT

Tezi Yöneten
Prof.Dr. Kenan DANIŞMAN

Elektronik Mühendisliği Anabilim Dalı
Doktora Tezi

Şubat 2008
KAYSERİ

Prof. Dr. Kenan DANIŞMAN danışmanlığında **Murat ERAT** tarafından hazırlanan **"FPGA Tabanlı, PCI Arayüzlü, Gerçek Zamanlı Rasgele Sayı Üreteci Test Sistemi Tasarımı ve Uygulamaları"** adlı bu çalışma, jürimiz tarafından Erciyes Üniversitesi Fen Bilimleri Enstitüsü Elektronik Anabilim Dalında **Doktora** tezi olarak kabul edilmiştir.

10 / 04/ 2008

JÜRİ:

Başkan: Prof. Dr. Nizamettin AYDIN

Üye : Prof Dr. Mustafa ALÇI

Üye : Prof. Dr. Kenan DANIŞMAN

Üye : Prof. Dr. Derviş KARABOĞA

Üye : Doç. Dr. Recai KILIÇ

ONAY:

Bu tezin kabulü, Enstitü Yönetim Kurulunun 13/05/2008 tarih ve 2008/14-02 sayılı kararı ile onaylanmıştır.

13 /. 05 / 2008..

N. Ayyıldız

Enstitü Müdürü
Prof. Dr. Nusret AYYILDIZ

TEŞEKKÜR

Tez çalışmam esnasında her türlü desteği veren başta hocam Sn. Prof.Dr. Kenan DANIŞMAN'a, özellikle bana çalışmalarımın her aşamasında yardımcı olan, teşvik eden mesai arkadaşım, Uzm.Arş. Salih ERGÜN'a, özverili çabalarından dolayı Uzm. Arş. Alper KANAK'a, her türlü katkılarından dolayı mesai arkadaşlarım Uzm. Arş. Tevfik NUR, Uzm.Arş.M.Uğur DOĞAN, Uzm.Arş. Tayyar GÜZEL, Uzm.Arş.Fatih KARA, Uzm.Arş. İmran ERGÜLER, Uzm.Arş.Yücel BİCİL'e, ayrıca Hüseyin EĞİTMEN'e, TUBİTAK-UEKAE' deki mesai arkadaşlarıma ve de aileme teşekkürlerimi bir borç bilirim.

FPGA TABANLI, PCI ARAYÜZLÜ, GERÇEK ZAMANLI RASGELE SAYI ÜRETECİ TEST SİSTEMİ TASARIMI VE UYGULAMALARI

Murat ERAT
Erciyes Üniversitesi, Fen Bilimleri Enstitüsü
Doktora Tezi, Şubat 2008
Tez Danışmanı: Prof. Dr. Kenan DANIŞMAN

ÖZET

Günümüzde, bilişimdeki gelişmeler ve bilgi güvenliğine duyulan ihtiyaç rasgele olarak üretilen güçlü anahtarların gerekliliğini zorunlu hale getirmiştir. Bu anahtarlar çoğunlukla yazılım tabanlı Rasgele Sayı Üretecleri (RSÜ) tarafından oluşturulmaktadır. Ancak donanım tabanlı olmayan bir ortamda Gerçek Rasgele Sayı Üretici (GRSÜ) uygulamasını gerçekleştirmek güvenli değildir.

Bu çalışmada rasgele sayı üretmek ve yeni RSÜ tasarımlarını test etmek için geliştirilen donanım tabanlı gerçek zamanlı istatistiksel test ve veri toplama sistemi sunulmaktadır. Geliştirilen FPGA tabanlı donanım, sadece GRSÜ olarak kullanılmayacak, aynı zamanda bundan sonra üretilecek GRSÜ prototiplerinin geliştirme sürelerini kısaltacak ve sonuçları test etmek için kullanılacaktır. Tasarlanan RSÜ'nün PCI arayüzüne sahip olması, donanımı bilgisayar tabanlı kripto uygulamaları için ideal bir çözüm haline getirmektedir. Tasarlanan FPGA tabanlı donanımın test ve veri toplama modundaki maksimum hızı 100 Mbps olarak tespit edilmiştir. Bu hız literatürde bulunan RSÜ'lerin en fazla 40 Mbps hızında çıkış sağladığı göz önüne alındığında yeterli seviyededir. Bununla birlikte donanımsal GRSÜ çıkış hızı 128 Kbpsdır. Üretilen Rasgele Sayılar FPGA üzerine gömülmüş FIPS 140-2 testlerine gerçek zamanda tabi tutulmuştur. Ölçüm sonuçları, geliştirilen sistemin doğruluğunu, uygulanabilirliğini ve sağlamlığını onaylamaktadır.

Bu çalışma kapsamında, biyometrik teknolojilerin donanım tabanlı GRSÜ ile bileşiminin pratik uygulamalarını göstermek amacıyla, kişilerin parmak izi ve yüz şablonunu şifrelemekte kullanılan özel anahtarların FPGA tabanlı GRSÜ tarafından üretildiği ve test edildiği 2 farklı biyometrik kimlik doğrulama sistemi uygulama olarak gerçekleştirilmiştir. Tasarlanan donanım standart veya gömülü bir PC'ye PCI ara yüzü vasıtasıyla kolaylıkla takılabilir. Özel anahtarı oluşturan rasgele sayıların gerçek olması,

iki seviyeli rassallık testi uygulanarak garantilenmiştir. Rassallık testi ilk olarak donanımda (FIPS 140-2) daha sonra PC üzerinde NIST 800-22 test takımı kullanılarak gerçeklenmiştir.

Anahtar kelimeler: Rasgele Sayı Üreteci, PCI, FPGA, İstatistiksel Testler, NIST, FIPS.

FPGA BASED REAL TIME STATISTICAL TEST SYSTEM WITH PCI FOR RANDOM NUMBER GENERATOR DESIGN & APPLICATIONS

Murat ERAT
Erciyes University, Graduate School of Natural and Applied Sciences
Ph. D. Thesis, February 2008
Thesis Supervisor: Prof. Kenan DANIŞMAN

ABSTRACT

The need for information security with the developments in the information technologies mandates the increase in the strength level of randomly generated keys. These keys are commonly generated by a software based Random Number Generator (RNG). However, the software implementation of a RNG is not considered as secure as it would be in a hardware implementation.

In this study, a hardware based real-time statistical test and data acquisition system that could be used for random number generation and test of novel RNG design is presented. The Field Programmable Gate Array (FPGA) based hardware will not only generate random numbers but also help to speed up the process of prototype development of such systems. This hardware design will be an ideal candidate for computer based crypto applications due to its Peripheral Component Interface (PCI) based implementation. Maximum test and data acquisition speed of the FPGA based hardware is observed as 100 Mbps which is very satisfactory when compared to other existing designs in the literature which feature throughput upto 40 Mbps. Together with that, the throughput of hardware based TRNG is 128 Kbps. The generated random numbers are subjected to FIPS 140-2 tests that are burned into the FPGA and the results of measurements approve and verify the feasibility and robustness of the system.

Within the scope of this study, two different biometric identity verification systems were implemented for face recognition, and fingerprint recognition as an application. The system could easily be mounted on a PC by using the PCI interface. It was guarenteed by a two-level randomness test that the numbers which generate the private key were truly random. Randomness test was performed first on the hardware (FIPS 140-2) and second on the PC (NIST 800-22). The purpose of the presentation of these

applications is to show a practical use of hardware based TRNG in biometric applications.

Keywords: Random Number Generator, PCI, FPGA, Statistical Test Suite, NIST, FIPS.

İÇİNDEKİLER

ONAY: i
TEŞEKKÜR ii
ÖZET iii
ABSTRACT v
İÇİNDEKİLER vii
KISALTMALAR ve SİMGELER x
ŞEKİLLER LİSTESİ xii
1. BÖLÜM 1
GİRİŞ 1
1.1. Rasgele Sayılar 2
1.2. Rasgele Sayıların Uygulama Alanları 2
1.3. Rasgele Sayıların Kaynağı 4
1.4. Donanımsal Rasgele Sayı Üretimi 5
1.5. Rasgele Sayıları Test Etme 5
2. BÖLÜM 7
RASGELE SAYI ÜRETECLERİ (RSÜ) 7
2.1. Giriş 7
2.2. RSÜ Çeşitleri 8
2.2.1. Yalancı Rasgelelik 9
2.2.2. Gerçek Rasgelelik 10
2.3. Rasgele Sayı Üreteci Yapıları 11
2.3.1. Gürültü Kaynakları 12
2.3.2. Bir Gürültü Kaynağının Güçlendirilmesi 14
2.3.3. Çift Osilatör Yapısı 18
2.3.4. Kaotik Tabanlı RSÜ' ler 20
2.4. Sayısal Son İşleme 24
3. BÖLÜM 26
RSÜ' LERE UYGULANAN İSTATİSTİKSEL TESTLER 26
3.1. Giriş 26
3.1.1. Rasgelelik 26
3.1.2. Tahmin Edilemezlik 27
3.2. Test Etme 27
3.3. Rasgelelik, Tahmin Edilemezlik ve Test Hakkında Düşünceler 31
3.4. RSÜ lerine Uygulanan NIST İstatistiksel Testleri 31

3.5. Testlerin Açıklaması 33
3.5.1. Frekans (Monobit) Testi 33
3.5.2. Bir Blok İçinde Frekans Testi 33
3.5.3. Yinelemeler Testi 33
3.5.4. Bir Blok İçinde En-Uzun-Bir-Yinelemesi (Longest-Run-Of-Ones) Testi 33
3.5.5. İkili Matris Rankı Testi 34
3.5.6. Ayrık Fourier Dönüşümü (Spektral) Testi 34
3.5.7. Örtüşmeyen Şablon Eşleştirme (Non-overlapping Template Matching) Testi 34
3.5.8. Örtüşen Şablon Eşleştirme (Overlapping Template Matching) Testi 35
3.5.9. Maurer'in "Evrensel İstatistik" Testi 35
3.5.10. Lempel-Ziv Sıkıştırma Testi 36
3.5.11. Doğrusal Karmaşıklık (Linear Complexity) Testi 36
3.5.12. Seri (Serial) Test 36
3.5.13. Yaklaşık Entropi (Approximate Entropy) Testi 37
3.5.14. Kümülatif Toplamlar (Cumulative Sums) Testi 37
3.5.15. Rasgele Gezinimler (Random Excursions) Testi 37
3.5.16. Rasgele Gezinimler Değişken (Random Excursions Variant) Testi 38
3.6. FIPS 140-2 İstatistiksel Test Protokolü 38
4. BÖLÜM 40
PCI (Peripheral Component Interconnect) VERİ YOLU 40
4.1. PCI Veri Yolu Özellikleri 41
4.2. PCI Veri Yolu İşaretleri 44
4.3. PCI Veri Yolu Paylaşımı 47
4.4. PCI Adres Bölgesi 48
4.5. PCI Veri Yolu Komutları 50
4.6. PCI Veri Yolu Mekanik Özellikleri 51
5. BÖLÜM 54
RSÜ'LER İÇİN PCI ARAYÜZLÜ FPGA TABANLI GERÇEK ZAMANLI İSTATİSTİKSEL TEST VE VERİ TOPLAMA DONANIMI 54
5.1. Tasarlanan Donanım 54
5.1.1. Kullanılan Devre Elemanları ve Özellikleri 56
5.2. Tasarlanan GRSÜ 57
5.3. Çevrimiçi Testler ve Veri Toplama 60
5.4. Yazılımsal Olarak Uygulanan İstatiksel Testler 61
5.5. PCI Arayüzünün Donanım Gerçekleştirimi 65
5.6. Donanımsal Olarak Uygulanan İstatistiksel Testler 66

6. BÖLÜM 73
UYGULAMALAR 73
6.1. Tasarlanan FPGA Tabanlı Kart ile Yapılan Güvenlik Uygulamaları 73
6.2. Güvenli Biyometrik Kimlik Doğrulama Sistemleri için Donanımsal Bir GRAÜ 73
6.2.1. Giriş 73
6.2.2. WFMT Öznitelikleri 75
6.2.3. Güvenli Yüz Doğrulama Yöntemi 77
6.3. Donanımsal GRSÜ Tabanlı Güvenli Parmak İzi Doğrulama Sistemi 80
6.3.1. Giriş 80
6.3.2. Halka - Dilim Tabanlı Parmakizi Öznitelikleri 82
6.3.3. Gerçekleştirilen Parmakizi Yetkilendirme Şeması 85
7. BÖLÜM 88
SONUÇLAR VE ÖNERİLER 88
KAYNAKLAR 93
ÖZGEÇMİŞ 97

KISALTMALAR VE SİMGELER

RSA	Ron Rivest, Adi Shamir, Leonard Adleman algoritması
DSA	Digital Signature Algorithm
DES	Data Encryption Standard
AES	Advanced Encryption Standard
MD5	Message-Digest Algorithm-5
OTA	Operational Transconductance Amplifier
ANSI	Amerikan Ulusal Standartlar Enstitüsü
FIPS	Federal Bilgi İşleme Standartı
NITS	Teknoloji ve Standartlar Ulusal Enstitüsü
RSÜ	Rasgele Sayı Üreteci
YRSÜ	Yalancı Rasgele Sayı Üreteci
GRSÜ	Gerçek Rasgele Sayı Üreteci
GRS	Gerçek Rasgele Sayı
SHA-1	Güvenli Hash Algoritması (Secure Hash Algorithm)
ETS	Embedded Tool Suite
RTOS	Real Time Operation System
FPGA	Field Programmable Gate Array
FPAA	Field Programmable Analog Array
PCI	Peripheral Component Interconnect
FSM	Sonlu durum makinası (Finite State Machine)
FMT	Fourier Mellin Transform
WFMT	Wavelet Fourier Mellin Transform
RST	Dönme, Ölçekleme, Kayma (Rotation, Scale, Translation)
$\llcorner x \lrcorner$	X'in tam değer fonksiyonu; pozitif bir x için, $\llcorner x \lrcorner = x-g$, $\llcorner x \lrcorner > 1$ ve $0<g<1$
α	Anlam seviyesi
E[]	Rasgele değerin beklenen değeri
ε	Test edilecek orjinal sıfır ve bir bit girdi dizileri
ε_i	ε dizisinin i. sıradaki değeri
H_0	Geçersiz Hipotez, örneğin: "dizi rasgeledir" ifadesi
Log(x)	X'in doğal logaritması $\log(x)= \log_e(x)=\ln(x)$, ln doğal logaritmadır
$\text{Log}_2(x)$	$\ln(x)/ \ln(2)$ olarak tanımlanır

M	Alt dizide test edilecek bit sayısı
N	Test edilecek M-bit blok sayısı
n	Dizideki test edilecek bit sayısı
π_i	N bit arasında birlerin ortalama sayısı
σ	Rasgele değişkenin standart sapması
σ^2	Rasgele değişkenin varyansı, (standart sapma)^2
S_{abs}	Frekans testinde istatistik olarak kullanılan gözlemlenen değer
Φ	Standart normal kümülatif dağılım fonksiyonu
ξi	Tanımlı döngüde verilen durumun toplam gerçeklenme sayısı
X^2	Teorik chi-kare dağılım (test istatistiği olarak kullanılır.)
$X^{2(obs)}$	Gözlemlenen verilerden hesaplanan chi-kare istatistiği
W	Test edilen bit dizisinde beklenen kelime (word) sayısı
W_{obs}	Bir dizideki ayrık kelimeler (word)

ŞEKİLLER LİSTESİ

Şekil 2.1. INTEL Rasgele Sayı Üreteci Yapısının Blok Diyagramı 13
Şekil 2.2. Genel Rasgele Sayı Üreteci Yapısının Blok Diyagramı 14
Şekil 2.3. Termik Gürültüyü Temel Alan Gerçek Rasgele Sayı Üreteci 15
Şekil 2.4. Kaskad CMOS Kuvvetlendirici Yapısı 16
Şekil 2.5. Doğrudan Kuvvetlendirme Tekniği 17
Şekil 2.6. Gürültü Temelli RSÜ Entegre Devresinin Tümleşik Yapısı 18
Şekil 2.7. Çift Osilatör Yapısının Eklendiği RSÜ 19
Şekil 2.8. Çift Osilatör Tasarımının Genel Yapısı (Frekans Oranı Ölçeklendirilmemiş.) 20
Şekil 2.9. Akım Aynalarıyla Gerçeklenen Kaotik Devre. ($I_{out} \approx I_{in}$, Φ_1-Φ_2.örtüşmeyen saat kaynakları) 21
Şekil 2.10. Akım Aynalarıyla Gerçeklenen Kaotik Devrenin SPICE Simulasyonundan Elde Edilen Ayrık Zamanlı Kaotik Harita 22
Şekil 2.11. Çift Sarmallı Otonom Kaotik Osilatör Devresi 23
Şekil 2.12. Deneysel Olarak v_1- v_3 Düzleminde Elde Edilen Kaotik Çekici 23
Şekil 4.1. PCI Veri Yolu İçeren Bir Bilgisayar Donanımı 43
Şekil 4.2. Barber-Pole İnterrupt Bağlantı Düzeni 46
Şekil 4.3. PCI Veri Yolu Paylaşımı 48
Şekil 4.4. PCI Veri Yolu Bağlantı Yuvası Yapıları 52
Şekil 4.5. PCI Kartı Fiziksel Boyutları 52
Şekil 4.6. Evrensel (Universal) PCI Kartı Yuvası Bağlantı Tanımları 53
Şekil 5.1. Donanımın Genel Mimarisi 55
Şekil 5.2. FPGA Tabanlı PCI Arayüzlü Donanım 56
Şekil 5.3. Zener Gürültüsünü Temel Alan RSÜ Devresi 58
Şekil 5.4. Zener Diyot Tarafından Üretilen Gürültü Gerilimi 59
Şekil 5.5. Gerilim Karşılaştırıcı Girişine Verilen (Kanal 2) ve Çıkışından Elde Edilen (Kanal 1) Rasgele İşaretler 60
Şekil 5.6. Yazılım Akış Diyagramı 64
Şekil 5.7. Tasarlanan Donanımın Blok Diyagramı 66
Şekil 5.8. Detaylı FPGA Blok Diyagramı 67
Şekil 5.9. Monobit ve Long-Run Testlerinin Blok Diyagramı 68
Şekil 5.10. Poker ve Seri Testlerin Blok Diyagramı 70
Şekil 5.11. RUNS Testinin Blok Diyagramı 70
Şekil 6.1. WFMT Özniteliklerini Üreten Blok Diyagram 76
Şekil 6.2. Kayıt Aşaması 77

Şekil 6.3. Doğrulama Aşaması....78
Şekil 6.4. Donanım Olarak Gerçekleştirilmiş GRSÜ (Çift Osilatör Yapısı) 81
Şekil 6.5. Halka ve Dilimler 83
Şekil 6.6. Kayıt Aşaması....86
Şekil 6.7. Doğrulama Aşaması....86

1. BÖLÜM

GİRİŞ

55, 42, 77, 34, 68, 14 ve 39 sayıları arsındaki ilişkiyi bulmaya çalışalım. Ama kendinizi daha fazla zorlamayın. Zira bu sayıların hepsi rasgele olarak seçildi ve aralarında herhangi bir ilişki yok. Rasgele sayılar kriptografide oldukça sık kullanılır. Bilgi güvenliği, bilgisayar kullanıcıları için önemli bir konu olmakta ve kötü güvenlik tecrübelerinden kaynaklanan problemlerin olduğu bilinmektedir. Güvenlik uzmanları, her zaman tahmin edilemeyecek şifrelerin kullanılmasını tavsiye ederler ve sık sık değiştirilmesini isterler. Kullanıcılar, antivirüs tarayıcılarını yeni virüslere karşı güncelleme ihtiyacı hissederler. Şifre kırıcılar ve spam göndericileri her zaman daha iyi bir silahla geldiğinden, ağ yöneticileri güvenlik duvarlarını (firewalls) uygulamak zorunda kalırlar. Askeri birimler gizli bilgilerin bilgisayar ile iletim güvenliğinin sağlanması için daha iyi yollar ararlar.

Bilgisayar güvenliği bugün çok masraflı bir mesele haline gelmiştir. Birçok bilgisayar güvenlik firması, yoğun araştırmalar sonrası bankacılık ve savunma sektörleri için daha iyi ve daha basit çözümler sunabilmektedir. Bilgisayar ağları için güvenlik uygulamalarının bir dalı olan kriptografi, değişim geçirmektedir. Kriptografi, bilgisayar ağları üzerinden güvenli bilgi iletiminin şifrelenmesi, şifre çözülmesi, güvenlik hizmetleri, kullanıcı kimliklerinin doğrulanması, web üzerinden güvenli elektronik işlemlerin yapılması, zararlı programlara karşı savunma programları ve güvenlik duvarlarının tasarlanması gibi konularla ilgilenmektedir.

Birçok kripto algoritması bilgi bitlerinin (sayıların) işlenmesi üzerine dayanmaktadır. Rasgele sayılar (aralarında hiçbir ilişki bulunmayacak şekilde rasgele seçilen sayılar) kriptografide sıkça kullanılırlar.

1.1. Rasgele Sayılar

Rasgele sayılar tahmin edilme veya ne olacağı bilinme olasılığı çok düşük olan sayılardır. Bu şu anlama gelmektedir; üretilen sayı daha önce üretilenlerle hiçbir şekilde bir ilişkiye sahip olmamalıdır. Bu tür sayıları seçmek çok güçtür ve birçok durumda yalancı rasgele sayılar (Pseudo Random Numbers) denilen sayılar üretilir. Yalancı rasgele sayılar, 'nüve (seed)' adı verilen bir değere dayalı bir denklemi kullanarak üretilirler ve genelde belirli bir aralıktaki sayılarla sınırlıdır. Nüve değeri tüm değerleri üretebilmesi için birçok matematiksel ve mantıksal işleme tabi tutulur. Bu ve benzeri sayılar saat gibi kaynaklardan da üretilebilirler [1,2].

Bu çalışma, daha önce üretilenlerle hiçbir şekilde bir ilişkiye sahip olmayan rasgele sayıların üretimi ve bunun gerçek zamanda yapılan testleri hakkındadır.

Rasgele Sayı Üreteci (RSÜ) rasgele sayı üretiminde kullanılan aygıttır. Birçok yazılım aracı, program ve algoritma rasgele sayı üretebilir. Aynı zamanda bu işi yapan donanımsal aygıtlar da mevcuttur. Genel olarak yazılımsal olarak üretilen rasgele sayılara 'yalancı (pseudo) rasgele', donanımsal üretilenlere ise 'gerçek (true) rasgele' denir.

Donanım tabanlı rasgele sayı üretimi, yazılımsal üretime göre iki nedenden ötürü daha üstündür: Birincisi yazılımsal olarak üretilen rasgele sayı dizileri deterministtir ve periyodik olarak belli bir zaman sonra dizi kendini tekrar eder. İkincisi ise, nüve değerine bağlı olarak düzenli bir dağılıma sahiptirler ki bu tahmin edilebilirliği artırır. Örneğin, Netscape RSÜ' sü bu dezavantaja sahiptir. Bu RSÜ rasgele sayı üretiminde üç değişken kullanmaktadır: günün zamanı, işlem ID'si (kimlik) ve ebeveyn işlem ID'si. Bir saldırgan, yüksek işlem yapma kabiliyetine sahip bilgisayarları kullanıp, kaba kuvvet saldırısını gerçekleyerek nüve değerini kolayca tahmin edebilir. Çoğu durumda yazılımsal rasgele sayı üretimi yeterli olsa da, yüksek güvenlik isteyen ve gerçek rasgele sayıya ihtiyaç duyan uygulamalarda donanımsal rasgele sayı üretimi hayati öneme sahiptir [2,3].

1.2. Rasgele Sayıların Uygulama Alanları

Rasgele sayı üretiminin ayrıntılarına girmeden önce bu sayıların nerelerde kullanılabileceğine bakalım. Rasgele sayılar bilimin diğer dallarında, örneğin matematik (denklem çözme, olasılık), eğlence (kumar, rasgele müzik tonları), fizik (simülasyonlar)

v.s. dışında kriptografide sıkça kullanılmaktadır. Kriptografide kullanıldığı bazı alanlar aşağıda listelenmektedir [2,3,4,5].

Anahtar üretimi: Eğer bir kullanıcı, herhangi bir veriyi (örneğin e-posta) bir saldırganlara karşı korumak amacıyla şifrelemek isterse, anahtar adı verilen bir çift değer (açık ve özel anahtar) üretir. Bu iki anahtar arasındaki ilişki şöyledir: Açık anahtar özel anahtardan üretilebilir. Ancak tersi doğru değildir. Açık anahtar açık olarak dağıtılabilirken özel anahtar gizli tutulur. RSA (**R**ivest, **S**hamir, **A**dleman), Diffie-Hellman, DSA (Digital Signature Algorithm) ve eliptik eğri gibi açık-anahtar kriptografik algoritmalar rasgele sayıları kullanırlar.

Sayısal imza algoritması: Bilgisayar kullanıcıları, her zaman kopya ile orijinal dosya arasındaki farkı bilmek ister. Ağ üzerinden bir mesaj gönderildiğinde, mesajın gerçekten iddia edilen kullanıcı tarafından gönderildiğini ispatlayan bir mekanizma olmalıdır. Bunun için dosyalar sayısal imza ile imzalanabilirler. Sayısal imzayı üreten birçok program sayısal imza algoritması tabanlıdır. Sayısal imza algoritmaları rasgele oluşturulmuş özel bir değer kullanmaktadır.

Anahtar dağıtımında: Alıcı ve verici arasında gizli anahtarlar dağıtılırken, gizli anahtarı içeren mesajı şifreleyecek bir oturum anahtarı üretilmelidir. Gönderenin anahtar dağıtım merkezine rasgele oluşturulmuş sayı içeren bir talep mesajı göndermesi gerekir.

Bilgisayar-tabanlı parolalar: Bazen bilgisayardan şifre üretmesi istenebilir. Birçok web tabanlı e-posta hizmet sağlayıcıları, Yahoo Mail gibi, bilgisayar tarafından üretilen örnek bir parola verir. Bu parolalar rasgele seçilen sayılar içerir. Örnek bir şifre "altınpencere24" olabilir. Fakat "altın", "pencere" ve "24" kullanıcının bilgileriyle hiçbir şekilde ilişkili değildir.

Bilgisayar oyunları: Bilgisayar oyunları, oyuncuyu zorlamak için oyundaki nesnelerin gidişatını rasgele sayılar ile değiştirirler. Rasgele sayıların bu şekilde kullanılmasıyla oyundaki farklı nesnelerin gidişatı tahmin edilemez olur.

İstatistiksel analiz: Herhangi bir veri istatistik açıdan analiz edilmek istendiğinde, örnekleme yoluyla belirli değerler alınır. İstatistikçiler bu değer alımının rasgele olmasını isterler. Böylece düzgün bir örnekleme yapılabilir.

1.3. Rasgele Sayıların Kaynağı

En büyük rasgele sayılar koleksiyonu olarak bilinen ilk kitap 'A million random digits with 100,000 normal derivatives' başlığını taşımaktadır. 1955 yılında RAND Kurumu tarafından basılan kitap, rasgele sayıların önemi, üretimi ve testi hakkında bilgiler içermektedir. Kitap 32 bölmeli bir elektronik rulet çarkı tarafından üretilen rasgele sayıları içermektedir. İyi bir rasgele dizi elde etmek için, rulet her denemede 3000 kez çevrilir ve saniyede bir sayı alınır. Bu şekilde okunan 32 değerden 20 si, kalan haneler göz ardı edilerek, ikiliden-onluya dönüştürücüye gönderilir. Dönüştürücü, tüm bitleri onluk tabandaki sayılara çevirir. Bu değerler daha sonra programlarda kullanılmak üzere kartlara kaydedilir. Kitapta bu şekilde üretilen bir milyon sayı sunulmaktadır [3,4].

Kitap, rasgele sayı kılavuzu gibi durmaktadır. Her sayfada 50 satır ve her satırda 5 haneli 50 rasgele sayı bulunmaktadır. Kitap rasgele sayı özellikleri ve bu konudaki uzmanların teknikleri hakkında, özellikle o zamanlarda dünyanın işlem ve modern kripto teknikleri hakkında pek bir bilgi sahibi olmadığı göz önüne alındığında, ciddi bilgi içermektedir.

Çoğu uzman, rasgele sayıları biriktirmek için gerçek dünya varlıklarını kullanmaktadır: Tıkanmış trafikte bekleyen iki aracın plakalarından, günün belirli aralıklarında ölçülen rüzgar hız değerlerinden, borsadaki herhangi bir hissenin rakamlarından. Madeni para, zar veya bir kart oyunundan rasgele sayı üretiminde yararlanılabilir. Örneğin madeni para atılıp tura gelirse "1", yazı gelirse "0" olacak bir seri oluşturulabilir. Bu şekilde oluşturulan bir seri, rasgele sayı bitleri olarak düşünülebilir.

İlk elektriksel rasgele sayı üreten makinelerden birisi G.B. Agnew tarafından yapılan metal yalıtımlı yarı-iletken kapasitördür [3,4]. Bu amaçla kullanılan diğer öncü aygıtlar ise boş çalışan bir osilatörden alınan frekans değerlerini kullanan bir aygıt ile M.Gude'un fiziksel telsiz fenomenini temel alan bir aygıt RSÜ olarak kabul edilebilir. Rasgele sayılar için bugün de karşılığı olan en popüler kaynak Manfield Richard tarafından önerilen ve uygulanan termal gürültüdür. M. Gude ise rasgele sayıların radyoaktif bozulmadan elde edilebileceğini ispatlamıştır [3,4].

Ayrıca rasgele sayıları bilgisayarlarımızda bulunan aygıtlardan da elde edebiliriz.

Örneğin;

- Yarı iletken diyot gürültüsünden [2,4],
- Dirençteki Johnson gürültüsünden veya osilatörden [1,6,7,8,10,14,16,17],
- Bilgisayarın saat değerlerinden [2,4],
- Belirli peryotlarla farenin pozisyonlarından [2,4],
- Normal bir klavye kullanan bir kişinin iki tuşa basımı arasındaki geçen zamandan [2,4],
- Disket sürücülerin dönüm davranışlarından [2,4],
- Bellekten rasgele alınan bir dosyanın boyutundan [2,4],
- Ağa bağlı bir bilgisayarın dinamik IP adresinin bir bölümünden [2,4],
- Bir veri paketinin boyutundan [2,4],
- CPU ve diğer kaynakların yüklenmesinden [2,4],
- Ses kartına mikrofondan gelen ses gibi analog sinyallerden (ses/video) [2,4].

Tüm durumlarda kaynaklardan gelen sinyaller analog-sayısal çeviriciye gönderilip sonra modüler indirgece çıktı sinyalini almak için iletilirler. Bu bir RSÜ kartı tarafından beslenebilir. Bir karşılaştırıcı veya Schmitt tetikleyicisi ayrıca kullanılabilir.

1.4. Donanımsal Rasgele Sayı Üretimi

Günümüzde, birçok yarı iletken üreticisi rasgele sayı üreten çip ve mikro-işlemci sağlamaktadırlar. Birçok donanımsal RSÜ, PCI kartları ile veya seri, paralel, USB gibi haberleşme portlarına gömülmüş olarak gelmektedir. Ancak saat ve çipler üzerine yapılan analizler göstermektedir ki, donanımsal aygıtlarının hepsi iyi rasgele sayı üretememektedir [3,4].

1.5. Rasgele Sayıları Test Etme

Verilen bir rasgele sayı dizisinin rasgele olup olmadığını test etmek mümkündür. Genelde Knuth tarafından bulunan metod sık kullanılsa da, aslında farklı birçok araç vardır. Knuth, değerlerin dağılımı ve farklı değer kümeleri arasındaki özilinti gibi istatistiksel analizleri önermiştir. Araştırmacılar her ne kadar farklı metodlar önerse de, hepsi matematiksel analiz yerine istatistiksel analiz üzerine yoğunlaşmaktadır. Zira matematiksel analiz hem zor hem de kriptografik açıdan makul değildir [3,4].

Yakın zamanda birçok test metodu önerilmiştir. Florida State Üniversitesi İstatistik Bölümünden George Marsaglia tarafından geliştirilen Diehard Test Suiti çok popüler olup fiilen kullanılmaktadır. Ulusal Standartlar ve Teknoloji Enstitüsü (NIST) tarafından geliştirilen Federal Bilgi İşleme Standardı (FIPS 140-2) dökümanı rasgele sayı dizilerini test etmek için Monobit Testi, Poker Testi ve Runs Testi önermektedir. FIPS 140-2 de farklı test dereceleri için dört seviye mevcuttur (Seviye 1, Seviye 2, Seviye 3 ve Seviye 4). Bütün yeni standart RSÜ'lerin, kendilerini 'gerçek' RSÜ olarak niteleyebilmeleri için bu testleri geçmeleri gerekmektedir [3,5].

Bu tez çalışmasında temel amaç, bir mikro işlemcinin asli işine fazla yük getirmeden iyi rasgele sayılar üretebilen ve bu sayıları gerçek zamanda test edebilen donanımsal aygıtlar geliştirmektir.

2. BÖLÜM

RASGELE SAYI ÜRETECLERİ (RSÜ)

2.1. Giriş

Güvenli bir kripto sistemi gerçek rasgele sayılar gerektirir. Neredeyse tüm kripto protokolleri, sistemi kırmaya çalışacakların kesinlikle bilmemesi gereken gizli değerlerin üretimi ve kullanımına ihtiyaç duyarlar. Örnek olarak, RSA, DSA ve Diffie-Hellman'ı içeren asimetrik (açık anahtar) algoritmalar için açık/özel anahtar çifti üretiminde rasgele sayı üreteçleri gerekmektedir [6]. Aynı şekilde simetrik ve hibrit kripto sistemleri için kullanılacak anahtarlar rasgele şekilde üretilmektedir. Rasgele sayı üreteçleri aynı zamanda kimlik doğrulamada, yalnız bir olay için kullanılacak sayıların, veri dizilerinin sonuna eklenen dolgu baytların üretiminde de kullanılmaktadır. Rasgele sayı üreteçlerinin neredeyse en yaygın kullanıldığı uygulama alanı, kripto sistemleri arasında yapılan kimlik doğrulama işlemidir. Terminal kimlik doğrulama yapacağı cihazdan rasgele bir sayı bekler. Kimlik doğrulama yapılacak cihaz bu sayıyı şifreleyip terminale gönderir. Eğer terminal bu sayıyı çözebilirse kimlik doğrulama başarıyla gerçekleşir.

Anahtar üretim işlemi, ispatlanacak şekilde güvenli tek kripto sistemi olan tek seferlik dolgu yapısında olduğu kadar karmaşık olmalıdır. Bu yapıdaki anahtar dizisinin de gerçek rasgele sayı üretme metoduyla oluşturulması gerekmektedir.

Güvenlik protokollerinin güvenilirlikleri, kullandıkları anahtarların önceden tahmin edilemez oluşlarına bağlı olduğu için, kripto sistemlerinde kullanılacak rasgele sayı üreteçleri çok sıkı gereksinimleri karşılamalıdır. En önemli gereksinim, rasgele sayı üreteci tasarımı hakkında bilgisi olan, sistemi kırmaya çalışabilecek kişilerin RSÜ çıkışları hakkında doğruluk oranı yüksek tahmin yapmasını engellemektir. Özellikle, RSÜ çıkışının görünen entropisi, bit uzunluğuna mümkün olduğu kadar yakın olmalıdır.

Shannon'a göre herhangi bir mesaj veya durumun entropisi Denklem 2.1' de verilmiştir:

$$H = -K\sum_{i=1}^{n} p_i \log p_i \qquad \textbf{(2.1)}$$

Burada *Pi*: olası n durum dışında i durumunun olasılığı, K, birimleri sağlamak için isteğe bağlı kullanılan bir sabittir. Bir rasgele sayı üretecinin *k*-bit ikili sonuç üretmesi durumunda P_i, $0 \le i < 2^k$ aralığı için bir çıkışın *i* 'ye eşit olma olasılığıdır. Böylece *mükemmel* bir RSÜ için $P_i = 2^{-k}$ 'dır ve çıkışın entropisi *k* bit'e eşittir [6]. Bunun anlamı, olası tüm sonuçların aynı derecede uygun olması veya olmaması ve ortalama olarak çıkışta bulunan bilginin *k* bitten daha kısa bir dizide temsil edilememesidir. Kripto uygulamalarında kullanılacak bir RSÜ, *mükemmel* RSÜ ile mümkün olduğu kadar rekabet edebilecek düzeyde hesapsal olarak çok sıkı gereksinimleri karşılayacak özellikleri göstermelidir.

2.2. RSÜ Çeşitleri

Elektronik resmi ve finansal işlemlerin de ortaya çıkmasının doğal bir sonucu olarak bilgi gizliliği ihtiyacı sürekli artmaktadır. Bundan dolayı, kripto uygulamalarının tabanı olarak RSÜ' leri, sayısal haberleşme cihazlarında kullanılmaya başlanmıştır.

Rasgele sayı üreteçleri iki ana gruba ayrılabilir:

- Yalancı Ragele Sayı Üreteçleri (YRSÜ),
- Gerçek Rasgele Sayı Üreteçleri (GRSÜ).

GRSÜ'ler gerçek rasgele sayı üreten bilinemeyen kaynakların (entropi kaynakları) avantajına sahiptirler. GRSÜ çıktısı doğrudan rasgele sayı dizisi olarak kullanılabileceği gibi bir YRSÜ'yü de besleyebilir. Gerek asimetrik algoritmalar için kullanılan açık/özel anahtar çiftlerinin üretiminde, gerekse simetrik ve hibrit kripto sistemlerinde kullanılan anahtarların üretilmesinde rasgele sayılara ihtiyaç vardır.

YRSÜ'ler bit dizileri üretmek için kestirilebilen bir algoritmayı kullanırlar. GRSÜ tarafından üretilmiş gibi göstermek için YRSÜ'ler kısa gerçek rasgele bir dizi tarafından ilklendirilirler ve bu dizi ile üretilen çıktı arasında herhangi bir ilintinin olmasını önlerler [2,4].

Yukarıda bahsedilenlere ek olarak; yüksek kaliteli GRS'lerin (Gerçek Rasgele Sayı) oluşturulması zaman kaybına sebebiyet vereceği için, yüksek miktarda rasgele sayı

istendiğinde gerçeklenebilir de olmayacaktır. Bundan dolayı yüksek miktarda rasgele sayı üretimi için YRSÜ'ler daha uygun olabilirler.

RSÜ tasarımı bilinmesine rağmen çıktı hakkında kullanılabilir bir kestirim yapmak mümkün olmamalıdır. Kripto uygulamalarında istenen güvenlik gereksinimlerini yerine getirmek için, GRSÜ şu özellikleri sağlamalıdır: Çıktı bit dizisi bütün rasgele istatistiksel testleri geçmelidir; rasgele bitler ileri ve geri yönde tahmin edilemez olmalıdır; GRSÜ'nün çıktı bit dizisi tekrar üretilememelidir [4]. GRS üretmenin en iyi yolu gerçek hayattaki doğal rasgele olaylardan düzenli olarak meydana gelenleri bularak işlemektir. Bu olaylara örnek olarak radyoaktif ışıma sırasındaki yarılanma süresi, ısı ve vuruş gürültüsü, osilatör jiteri veya yarı iletken bir kapasitörün yük miktarı verilebilir [2].

2.2.1. Yalancı Rasgelelik

Birçok rasgele sayı kaynağı aslında yalancı rasgele üreteçlerden faydalanır. Günümüzde, birçok sistemde kullanılan rasgele sayı üreteçleri, deterministik algoritmaları temel alarak rasgele bir kaynağı genişletirler, yani başlangıçtaki kaynak durumdan bir seri çıkış üretmek için deterministik yöntemleri kullanırlar, bu yüzden üretilen sayılar "yalancı (pseudo) rasgele sayı" olarak isimlendirilirler. Bu tür rasgele sayı üreteçlerinin kullanıldığı sistemlerin güvenirliği hiçbir zaman istenilen düzeyde garanti edilemez. Çıkış, tamamen kaynak verinin bir fonksiyonu olduğu için çıkışın gerçek entropisi hiçbir zaman kaynağın entropisini geçemez. Fakat, hesapsal olarak iyi bir YRSÜ' yü, *mükemmel* RSÜ'den ayırmak mümkün değildir.

Örnek olarak 256 bitlik entropi ile kaynaklı bir YRSÜ 256 bitten daha fazla gerçek rasgelelik üretemez. Sistemi kırmaya çalışabilecek bir kişi, kaynak veriyi tahmin etmekle, YRSÜ tüm çıkışını tahmin etmiş olur. 256 bitlik kaynak verinin tahmin edilmesinin hesapsal olarak mümkün olmadığı kabulüyle, ancak böyle bir YRSÜ' nün kripto uygulamalarında kullanılabileceği sonucuna varılabilir. Uygulamalarda, sırasıyla 128 bit ve 160 bit tutan MD5 ve SHA1 içeren kripto uygulamaları için tasarlanmış YRSÜ örnekleri bulunmaktadır [6,7].

Gerektiği gibi uygulanmış ve kaynaklandırılmış YRSÜ birçok kripto uygulaması için uygun olsa da geliştirme, test ve YRSÜ algoritması seçiminde hat safhada dikkatli davranılmalıdır. YRSÜ'de güvenilir bir giriş verisi kullanılarak üretecin esaslı bir

şekilde kaynaklandırılması kritik bir noktadır. Örnek olarak, standart yazılım kütüphanelerinde bulunan birçok YRSÜ, önceden tahmin edilebilir kaynak veri veya durum değeri kullanırlar veya rasgele veriden ayırt edilebilen çıkış üretirler.

2.2.2. Gerçek Rasgelelik

GRSÜ, rasgelelik üretmek için deterministik olmayan kaynakları kullanır. Birçoğu önceden tahmin edilemeyen termik gürültü, atmosferik gürültü veya nükleer bozulma gibi doğal prosesleri ölçerek çalışır. GRSÜ'nün entropisi, güvenilirliği ve performansı tasarımın uygun olmasına bağlıdır.

Kaynak için GRSÜ olmaksızın, tek başına bir YRSÜ emniyetsiz olacaktır. Deterministik bir sistem içinden gerçek rasgelelik üretmek imkansız olduğundan kaynak için gerçek rasgele bir kaynak gerekmektedir.

Programcılar genellikle bilgisayarda, donanımsal RSÜ olmaksızın mevcut çevresel elemanları kullanarak kaynak veri için entropi elde etmeye çalışırlar. En yaygın teknik, kullanıcı zamanlama yöntemini içermektedir. Fakat bu yöntemler çok hantal ve yavaştır. Örnek olarak PGP 6.02 versiyonunda yeni bir anahtar üretmek için, kullanıcının klavyeden rasgele tuş darbesi girmesi veya fare hareketi oluşturması esnasında yaklaşık 15 sn' ler mertebesinde zaman harcaması gerekmektedir [6,7]. Bahsedilen yöntemler, kullanıcının anlık aktivitesine ve çalışacağı sistemin performansına bağlı olup, otomatik programlarla kontrol edildiklerinde kullanılamazlar veya emniyetsiz hale gelirler. Bazı uygulamalar sabit disk tarama sürelerini kullanmaktadır fakat disk teknolojisi ve sistem saat çözünürlüğü gibi etkenler birçok faktörün çok özenle hesaba katılmasını gerektirmektedir. Sistem aktivitesinin ölçülmesi, gecikmeler ve konfigürasyon verisi gibi kaynaklar da bazı yöntemlerde kullanılmaktadır.

Kapsamlı bir şekilde bakılırsa, basmakalıp donanımlar kullanılarak gerçeklenen gerçek rasgele sayı üreteçleri, yavaş çalışma eğiliminde, gerçeklenmesi zor, kullanıcının ilgisini gerektirmekte ve çoğu kez bilinmeyen miktarda gerçek entropi sağlamaktadır. Bu metotlar aynı zamanda, garanti edilmemiş donanımlar hakkında varsayımlar oluşturmaktadır. Örnek olarak işlem zamanı ölçümü, tüm sistem konfigürasyonlarında tahmin edilen miktarda rasgelelik içermeyebilir. Kullanıcının meydana getirdiği olayları kaynak alan teknikler sunucular gibi başıboş sistemlerde güvenilir olmayabilir. Birçok

uygulama için kendilerine ait güvenli rasgele veriyi üretmek mümkün olsa da birçoğunda bu yapılmaz [6,7].

Kripto araştırmaları neticesinde sık sık dile getirilen eleştirilerin ortaya koyduğu teşhis rasgele sayı üretimi konusundaki zayıflıktır. Aynı şekilde, yazılımsal güvenlik sistemleri üzerinde yapılan araştırmalar neticesinde yapılan eleştirilerde en yaygın dile getirilen problem, rasgele sayı üretimi için kaynak oluşturmayla ilgilidir. İyi bir RSÜ' nin tasarımı zordur. Bunun sebebi de üreteçlerin güvenirliklerinin donanım ve yazılımdaki ayrıntılara bağlı olmasıdır.

John Von Neumann'ın (1951) dediği gibi "Rasgele sayı üretmek için matematiksel metotlar üzerinde düşünen biri tabii ki çok yanlış bir yoldadır."

Örneğin Netscape'in kullandığı RSÜ, kaynağını üç nicelikten almaktadır: Bunlar o günkü saat, işlem kimlik numarası ve ana işlem kimlik numarası. Bu yüzden, bu üç değeri önceden tahmin edebilecek biri için iyi bilinen MD5 algoritmasını uygulayarak üretilen kaynak veriyi hatasız bir şekilde hesaplamak çok kolay olacaktır. Rasgele sayı üreteçlerinin birçok kusuru anlatılmamakla beraber problemler ortaktır. Kısmen bunun sebebi kripto yazılım kütüphanelerinin güvenilir kaynak materyal bulma işini yazılımcıya bırakmış olmasıdır [6,7].

Bir çok durumda sistem tasarımcıları, güvenliği elde etmek için kolaylıktan vazgeçme durumuyla karşı karşıya kalmaktadırlar. Örnek olarak, program her yüklendiğinde yeni kaynak verisi toplamayı önlemek için, birçok yazılım uygulaması kaynak materyali sabit sürücüde saklarlar ki bu uyuşma riski taşımaktadır. Önceden tahmin edilemeyen veri için en iyi kaynaklar, kullanıcı zamanlama metodunu içermektedir ki bu kullanıcı ara yüzüne istenilmeyen bir karmaşıklık katmaktadır.

2.3. Rasgele Sayı Üreteci Yapıları

Literatürde bilinen birkaç RSÜ tasarımı mevcuttur. Ancak rasgele sayı üretiminde üç farklı teknik bilinmektedir:

- Bir gürültü kaynağının güçlendirilmesi [7,8],
- Çift osilatör yapısı [6,9,10],
- Kaotik tabanlı RSÜ' ler [14,15,16].

Bilgisayar tabanlı kripto uygulamalarında GRSÜ işlevleri; damgalanmış disket sürücüsündeki hava türbülansına (bu disk sürücüsünün gizli okuma zamanlarında rasgele iniş çıkışlara sebep olacaktır), bir mikrofunun sesine, sistem saatine, iki tuş dokunması arasında geçen zamana, fare hareketine, girdi/çıktı tampon içeriklerine, kullanıcı girdi ve işletim sistemi değerlerine, örneğin sistem yükü ve ağ istatistiklerine, bağlı olarak gerçekleşebilmektedir. Bu işlevlerin davranışı bir çok faktöre bağlı olarak farklılık gösterebilmektedir. Örneğin kullanıcı, işlem etkinliği, bilgisayar ortamı v.s. gibi. Bu durum ise yüksek ve sabit veri oranları açısından önerilmeyeceğinden dezavantaj getirmektedir.

Verilen örneklere ek olarak, rasgele sayıları kullanan farklı uygulama alanları mevcuttur; örneğin sayısal imzaların üretimi, kimlik doğrulama protokolleri, bir kripto modülün ilklendirilmesi için ilk değerin rasgeleleştirilmesi, modelleme ve simülasyon uygulamaları, v.s.

2.3.1. Gürültü Kaynakları

Gerçek rasgele sayı üretmek için deterministik olmayan kaynaklar kullanılır. Muhtemel rasgele gürültü kaynakları şu şekilde sıralanabilir;

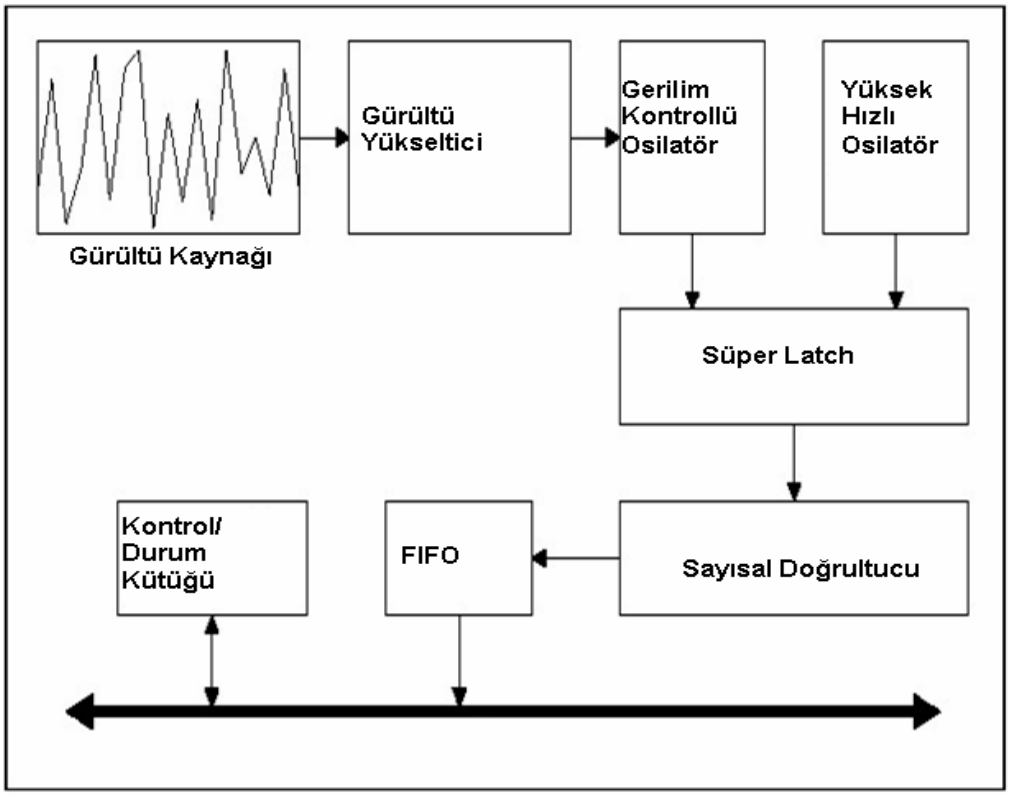
- Bir osilatörün frekans kararsızlığı,
- Yarıiletken elemanların gürültüsü,
- İki radyoaktif emisyon arasında geçen sure,
- Direncin termik gürültüsü,
- MOS kapasite yapısındaki yüklerin sayısı.

Genellikle termik gürültü olarak bilinen Johnson gürültüsü, shot ve flicker gürültüleri tüm dirençlerde bulunmaktadır ve gerçek rasgele sayı üreteçlerinde en sık kullanılan rasgele kaynaktır. Tüm bu gürültüler elektriksel olarak ölçülebilen karakterlere sahiptir ve rasgele elektron ve materyal davranışları neticesinde meydana gelmektedir [6].

Termik gürültünün kullanıldığı yöntemlerde esasen yapılan, sürülmemiş direnç üzerinden ölçülen gerilimin kuvvetlendirilerek termik gürültünün örneklenmesidir. Rasgele kaynağa ek olarak bu ölçümler elektromanyetik radyasyon, sıcaklık ve güç kaynağı dalgalanmaları gibi sınırlı yalancı rasgele çevresel faktörlerle de ilişkili olmaktadır. İki farklı direnç üzerindeki işaretlerden alınan örneklerin birbirinden

çıkarılması (fark kuvvetlendiricisi yapısı) ile elde edilen yapı sayesinde kullanılan eleman sayısı kayda değer bir şekilde azalmaktadır.

Rasgele sayı üreteci tasarımında dikkat edilmesi gereken noktalardan biri de uygulama veya işaret işleme için kullanılan devrelerdeki elemanların işaretin rasgeleliğini mümkün olduğu kadar bozmadan sürdürmesidir. Dikkat edilmesi gereken diğer önemli noktalar ise tasarımın yüksek derecede lineer olması ve sonraki katlara yüksek frekanslı işaretlerin geçmesini sağlamak için yüksek bir bant genişliğine sahip olmasıdır. Şekil 2.1' de örnek olarak gürültü kaynağını baz alan INTEL tarafından üretilen RSÜ yapısı verilmiştir [6].



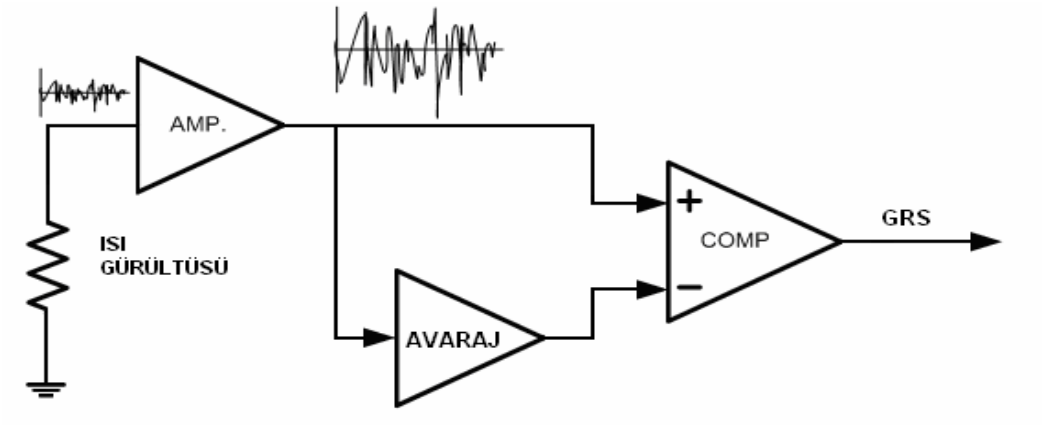
Şekil 2.1. INTEL Rasgele Sayı Üreteci Yapısının Blok Diyagramı

Rasgele işaret olarak yarıiletken elemanların; diyot veya transistorların oluşturduğu gürültünün kullanılması, bu gürültünün entegre devrenin kendi üzerinde oluştuğu düşünülürse diğer kaynakların kullanılmasına göre daha mantıklıdır. Özellikle düşük gerilim uygulamalarında bu gürültünün kullanılması çok daha kolaydır [6]. Gürültüyü

oluşturmak için zener diyot, tek katlı bir işlemsel kuvvetlendirici veya OTA (Operational Transconductance Amplifier) yapısı kullanılabilir.

2.3.2. Bir Gürültü Kaynağının Güçlendirilmesi

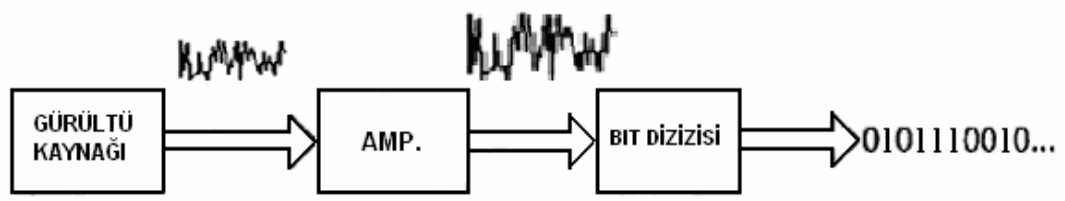
RSÜ modunda donanım tabanlı GRSÜ, çok bilinen bir teknik olan ısı gürültüsünü kullanmaktadır. Bu süreç, arttırıcı etkiye sahip olup rasgele gürültü tepeleri dizilerine sebep olmaktadır. Bir dirençten elde edilen bu gürültü beyaz tayfa sahiptir. Op-Amp, dirençteki gürültü gerilimini kuvvetlendirir. Çıktı sinyali referans olarak güçlenmiş gürültünün ortalamasını kullanan gerilim karşılaştırıcıya gönderilir. Ortalama değerden büyük olan pozitif değerler '1' değerini alırken diğerleri ise '0' değerini alır. Gerilim karşılaştırıcının çıkış sinyali, bit dizileri elde etmek için uygun frekansta örneklenir. Ancak üretilen ikili dizilerin 0, 1 dengesi bozuk olabilir. Dizideki bilinmeyen bu bozukluğu ortadan kaldırmak için iyi bilinen Von Neumann'ın düzeltici tekniği uygulanabilir [17]. Bu teknik 01 bit çiftini '0' çıktısına, 10 bit çiftinide ise '1' çıktısına çevirir. 00 ve 11 ise gözardı edilir. Yaklaşık olarak 4 bitten 1 bit üretildiğinden bu işlem rasgele sinyalin frekansını, örnekleme frekansının ¼'üne düşürür. Bahsedilen donanım Şekil 2.2'de gösterilmektedir.



Şekil 2.2. Genel Rasgele Sayı Üreteci Yapısının Blok Diyagramı

Literatürde bulunan çalışmalardan bir tanesi 2001 yılında Chen Hongyi ve Huang Zhun tarafından yapılmıştır [12]. Bu çalışmada direncin termik gürültüsünü temel alan basit bir GRSÜ devresi, simülasyon sonuçlarıyla birlikte sunulmuştur. Standart CMOS proses kullanılarak bu devrenin üretimi yapılabilir. Chen Hongyi ve Huang Zhun tarafından yapılan bu tasarım üç kısımdan oluşmaktadır: Şekil 2.3'de görüldüğü gibi fiziksel

gürültü kaynağının oluşturulması, gürültünün kuvvetlendirilmesi ve gürültü dalga şeklinin değerlendirilerek bit dizisi şekline dönüştürülmesi. Çıkış bit dizisinin, daha yüksek bir kaliteyi sağlamak için, rasgeleliği artırma prosedürleri uygulanarak tekrar gözden geçirilmesi gerekebileceği rapor edilmiştir.



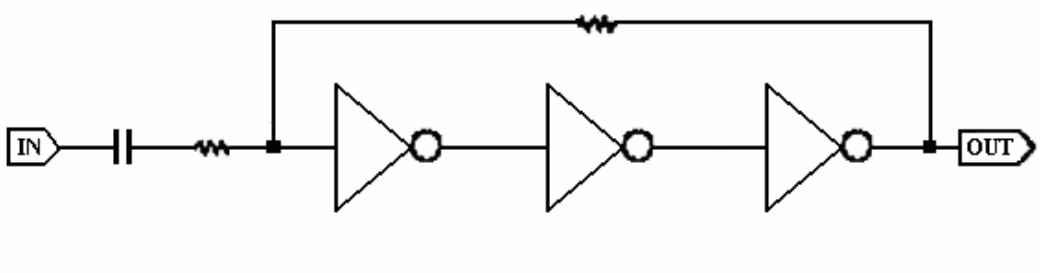
Şekil 2.3. Termik Gürültüyü Temel Alan Gerçek Rasgele Sayı Üreteci

Doğada gözlemlenen birçok fiziksel gürültü fenomeni bulunmaktadır. Bunların içinde termik gürültü performans açısından idealdir ve GRSÜ tasarımında kolaylıkla uygulanabilir.

Direncin iki terminali arasındaki gerilim, elektronların Brownian hareketinden dolayı rasgele bir dalga şekli alacaktır. Direncin veya herhangi bir iletkenin bu yükselip alçalan gerilimi Johnson gürültüsü olarak isimlendirilir ve bu gerilim iki terminal arasından akan akımdan bağımsızken sadece sıcaklık ve dirence bağlıdır. Bu normal dağılıma sahip bir beyaz gürültüdür. 1928 yılında Nyquist buna ait güç spektral yoğunluğunun $S(f)=4kRT$ şeklinde verilebileceğini ispatlamıştır. Burada k Boltzmann sabitini, R direnç değerini ve T Kelvin olarak sıcaklığı simgelemektedir. Bu denklemi kullanarak bir örnek verelim, 150 KΩ'luk bir direnç (Genellikle 50 KΩ'luk dirençler kullanılmaktadır) oda sıcaklığında, 50 nV/√Hz'lik gürültü gerilimine sahiptir. 500 KHz bant genişliğe sahip bir kuvvetlendirici kullanarak bu gürültü 500 katına kuvvetlendirilirse, yaklaşık olarak 18 mV'luk bir çıkış gerilimi ölçülebilir. Kuvvetlendiricinin kazancı ve bant genişliği ayarlanarak uygun çıkışlar elde edilebilir [6,12]. Burada en önemli nokta kuvvetlendirici kısmından sonra, gürültünün "Beyaz gürültü" özelliği gösterip göstermediğidir. Kuvvetlendiricinin bant genişliğinin sınırlı olması gürültünün "Beyaz gürültü" özelliğini sürdürmesine engel olacak başlıca etkendir. Kuvvetlendirici kısmından sonra, çıkış işareti sadece gürültü kaynağının ürettiği gürültüyü değil aynı zamanda kuvvetlendiricinin kendisinden kaynaklanan gürültüleri de içermektedir ki pek çok durumda kuvvetlendiriciden kaynaklanan gürültü daha baskındır. Tüm bu etkenler

sonuçta RSÜ' nün kalitesini etkileyecektir. Burada kritik olan nokta bu tür etkilerin nasıl azaltılacağıdır.

Chen Hongyi ve Huang Zhun tarafından yapılan tasarımda gürültünün kuvvetlendirilmesi için Şekil 2.4'de verilen kaskad (arka arkaya bağlantı) CMOS kuvvetlendirici yapısı kullanılmıştır. Buradaki temel sorun, yine yukarda bahsedildiği gibi bant genişliğinin sınırlı olması ve kuvvetlendiricinin kendisinden kaynaklanan gürültünün, üretilen gürültüden daha baskın olması ihtimalidir. Yapılan tasarımda gürültü dalga şeklinin değerlendirilerek bit dizisi şekline dönüştürülmesi için karşılaştırıcı kullanılmıştır. Gürültü, referans gerilimle karşılaştırılacak, referanstan büyük değerler için karşılaştırıcı "1" aksi taktirde "0" üretecektir. Referans gerilim, gürültünün ortalama değeri olarak seçilirse çıkış bit dizisi yaklaşık olarak eşit miktarda "0" ve "1" bitlerini içerecektir. Aynı zamanda "0" ve "1" bitlerinin, gürültünün dağılımından bağımsız bir dağılım gösterdiği rapor edilmiştir.

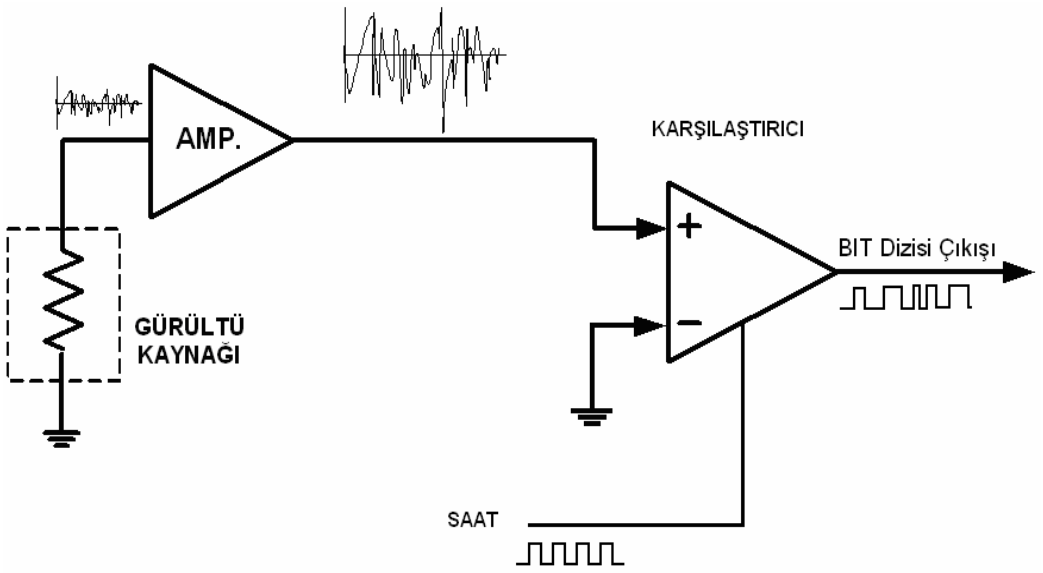


Şekil 2.4. Kaskad CMOS Kuvvetlendirici Yapısı

Yapılan tasarımın dezavantajları kaskad kuvvetlendiricinin bant genişliğinin sınırlı olmasından dolayı gürültünün "Beyaz gürültü" özelliğinin bozulması, devre CMOS proses kullanılarak üretilmediğinden gerçek sonuçların alınamaması ve simülasyon sonuçlarının FIPS 140-2 kriptografik teknik şartnamesinde tanımlanan RSÜ testlerine tabi tutulmaması şeklinde sıralanabilir [5].

Literatürde bulunan diğer bir çalışma ise Mayıs 2000 tarihinde J. Alvin Connelly ve Craig S. Petrie tarafından yapılmıştır [13]. Kriptografik uygulamalar için geliştirilen gürültü temelli RSÜ entegre devresinde, gürültü üretimi için Şekil 2.5'de gösterilen doğrudan kuvvetlendirme tekniği kullanılmıştır. Doğrudan kuvvetlendirme tekniğinde, termik veya shot gürültü (schottky noise) gibi gürültü kaynakları tarafından üretilen düşük seviyeli AC gerilimi işlemek için, yüksek kazançlı ve yüksek bant genişliğine

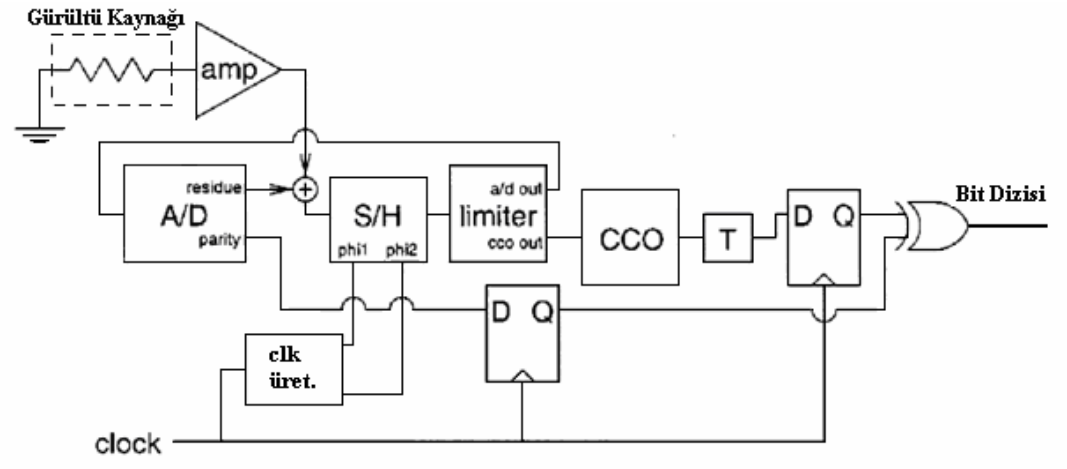
sahip kuvvetlendiriciler kullanılır. Gürültü, saat bağlanmış bir karşılaştırıcıdan etkilenmeksizin doğru bir şekilde eşik seviyelerinin belirlenebileceği bir seviyeye kuvvetlendirilmelidir. Doğrudan kuvvetlendirme tekniği, gürültü kaynağının elektromanyetik dalgalara karşı korunmasının mümkün olduğu tek kırmık veya plaket seviyesindeki çözümler için en popüler RSÜ tekniğidir. Yapılan çalışmada entegre devre yapısında, güç kaynağından ve alt tabaka işaretlerinden kaynaklanacak elektromanyetik ışımaya karşı yeterli koruma sağlanmazsa, bu metodun entegre devre bazlı kriptografik sistemlerde kullanımının sakıncalı olacağı rapor edilmiştir.



Şekil 2.5. Doğrudan Kuvvetlendirme Tekniği

J. Alvin Connelly ve Craig S. Petrie tarafından tasarlanan prototip RSÜ sisteminin tümleşik yapısı Şekil 2.6'da gösterilmiştir [13]. Yapılan tasarımda, üç farklı RSÜ metodunun birleşimi kullanılmıştır. Doğrudan kuvvetlendirme tekniği ile kuvvetlendirilen termik gürültü A/D bazlı kaotik sistemde toplanarak akım kontrollü osilatörü sürmek için kullanılmıştır. Akım kontrollü osilatör çıkışı, *D flip-flop* kullanılarak, kullanıcı tarafından belirlenen daha yavaş saat frekansıyla örneklenmektedir. A/D çevirici giriş işaretini, donanımsal olarak [0, I_{ref}] arasında sınırlamak için RSÜ yapısına sınırlama devresi eklenmiş olup böylece kaotik çevrimin sabit durum doyma koşulunda çakılıp kalması engellenmiştir. Sisteme, görev çevrimini dengelemek için akım kontrollü osilatör çıkış frekansını ikiye bölen bir *T flip-flop* eklenmiştir. Daha sonra eğilimleri azaltmak için, elde edilen çıkış, tüm A/D çevirici

dijital çıkışları XOR'lanarak elde edilen A/D çevirici parite işareti ile XOR işlemine tabi tutulmuştur [13].



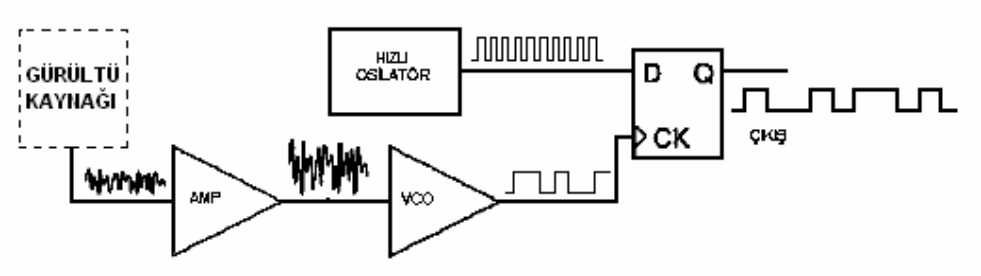
Şekil 2.6. Gürültü Temelli RSÜ Entegre Devresinin Tümleşik Yapısı

Bu sistemde, birlikte uygun bir şekilde çalışan analog alt devreler kullanılarak kararlı çalışan bir tasarım elde edilmiş, tasarım 2 μ CMOS teknolojisi kullanılarak üretilmiş ve gerekli testlere tabi tutulmuştur. Sistem rasgele olmayan etkilerden arındırılmış, gürültü bazlı, sağlam tekil bir mimariye sahiptir. Elde edilen bit dizileri, 1.4 MHz bit hızlarına kadar tüm rasgelelik testlerini geçmiştir ve 4.7 MHz'e kadar benzer rasgelelik özelliği göstermiştir. Sistem tek 3 V'luk kaynakla beslenmiş ve 1 MHz saat frekansında 3.9 mW güç tüketmiştir. Devre önemli miktarda rasgele olmayan etkilere maruz bırakıldığı halde rasgelelik performansında bir değişme gözlemlenmemiştir. Düşük güç tüketimi, küçük devre boyutları ve sağlamlığı, tasarımı entegre devre bazlı kriptografik sistemlerle bütünleştirilmesi açısından ideal hale getirmektedir. Sistemin tek dezavantajı, çok fazla karmaşık yapıyı içermesi olarak belirtilebilir. Tasarım üzerinde çalışılarak, sistemin sadece FIPS 140-2 kriptografik teknik şartnamesinde tanımlanan RSÜ testlerini geçen daha basit bir yapıya indirgenmesi sağlanabilir.

2.3.3. Çift Osilatör Yapısı

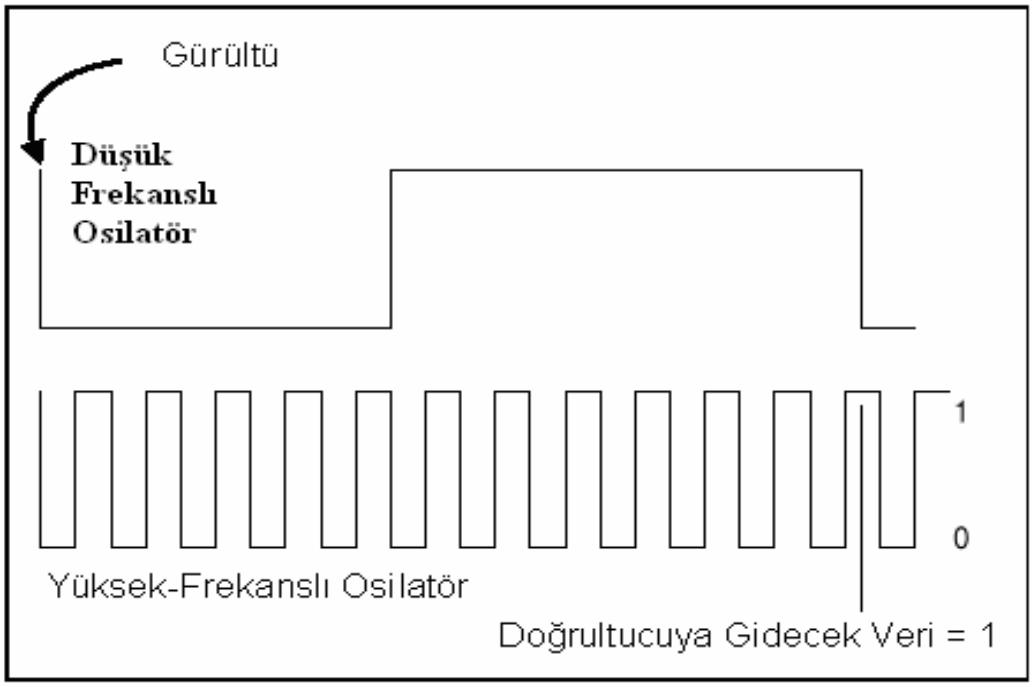
Rasgele sayı üreteçlerinde kullanılan diğer bir rasgele kaynak ise çift osilatör yapısıdır. Bu rasgele kaynak, biri hızlı diğeri daha yavaş serbest çalışan iki osilatörden elde edilmektedir. Şekil 2.7'de çift osilatör yapısı blok şeması verilmiştir. Gürültü kaynağı, daha yavaş olan saat frekansını modüle etmek için kullanılmaktadır. Değişken, gürültü

ile modüle edilmiş yavaş saat, hızlı saatin ölçümünü tetiklemek için kullanılır. İki saat arasındaki sürüklenme böylece rasgele ikili sayılara kaynak sağlamaktadır [10]. Bu yönteme benzer şekilde, bağımsız osilatörlerin kullanıldığı RSÜ tasarımları da sık kullanılmaktadır.



Şekil 2.7. Çift Osilatör Yapısının Eklendiği RSÜ

Yalancı rasgele özelliği gösteren çevresel, elektriksel ve üretimden kaynaklanan koşullara ek olarak, gürültü kaynağında yavaş osilatör frekansı da kullanılarak rasgelelik kaynakları önemli ölçüde karıştırılmalıdır. Modüle edilmiş frekansın kaydedilmiş histogramları normal dağılıma benzemektedir. Modüle edilmiş frekans, yaklaşık 10-20 kat daha yüksek frekans saat periyodunu kapsayan standart sapmaya sahip olmalıdır. Bu göstermektedir ki örnekleme işlemi rasgele kaynakla kayda değer bir şekilde değişmektedir. Çift osilatör tasarımının genel yapısı şekil 2.8'de verilmiştir. Gerçek rasgelelik üretecek bir elemanın düşük hızlı osilatörde mevcut bulunması kaydıyla, ilave edilen rasgele olmayan etkiler, RSÜ çıkışının niteliğini düşürmeyecektir. Örnek olarak, çevresel girişim, sonuçlara önceden tahmin edilemezlik özelliği katacaktır. Modülasyon bant genişliği yaklaşık olarak değişken osilatör merkez frekansının iki katıdır. Bu yapıda kullanılan osilatörler 1:100 tarzında merkez frekans oranlarına sahip olmalıdır. Eğer her iki osilatör de kaymaksızın çalışırlarsa, örneklenen bitler periyodik bir şekilde oranın katlarındaki darbelerden etkilenebilirler.



Şekil 2.8. Çift Osilatör Tasarımının Genel Yapısı (Frekans Oranı Ölçeklendirilmemiş.)

2.3.4. Kaotik Tabanlı RSÜ' ler

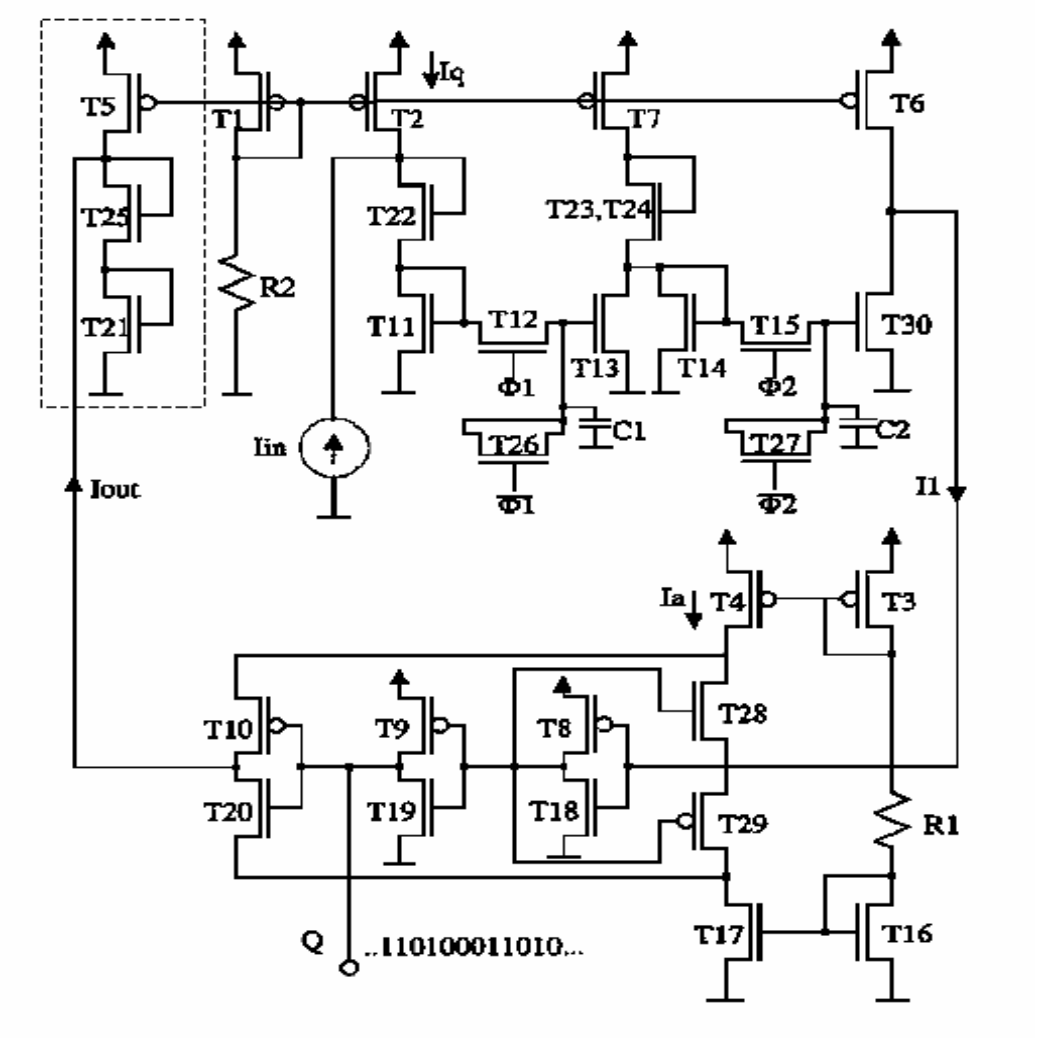
Kaotik tabanlı RSÜ' leri aşağıdaki gibi 2 gruba ayırmak mümkündür.

- Ayrık zaman kaotik planlama tabanlı RSÜ' ler [14,15,18],
- Sürekli zaman kaotik osilatör tabanlı RSÜ' ler [16].

Yakın zamana kadar ayrık zaman kaotik planlama tekniği kullanımı çok bilinse de, artık GRSÜ gerçekleşimi için sürekli zaman kaotik osilatörleri de kullanılmaktadır. Kaos temeline dayanan bir RSÜ yapmak için kaos yapısının küçük ölçeklerde deterministik özelliği ve büyük ölçeklerde rasgele özelliği kullanılır.

Ayrık zamanlı kaosu temel alan RSÜ yapısına örnek olarak [18]'de önerilmis olan ve Denklem 2.2'de verilen parçasal lineer ve tek boyutlu (PL1D) kaotik fonksiyonunu gerçekleyen tasarım ele alınmıştır.

$$x_n + 1 = f(\chi_n) = \begin{cases} q_1 + k_1 x_n, & x_n < 0 \quad için \\ -1 + k_2 x_n, & x_n \geq 0 \quad için \end{cases} \qquad \textbf{(2.2)}$$

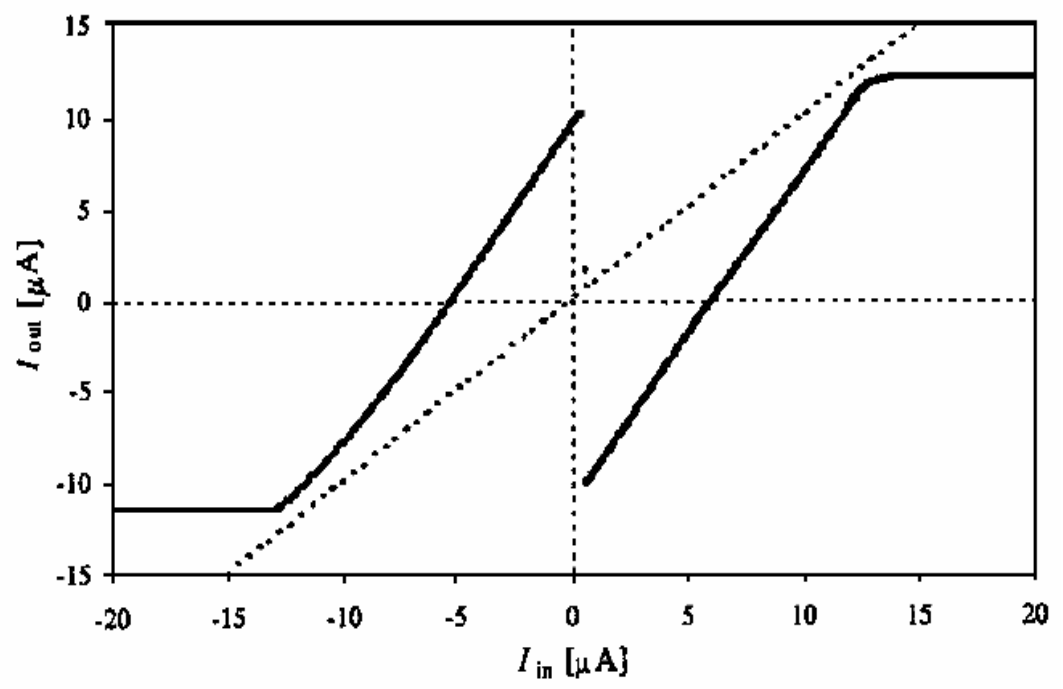


Şekil 2.9. Akım Aynalarıyla Gerçeklenen Kaotik Devre. ($I_{out} \approx I_{in}$, Φ_1-Φ_2.örtüşmeyen saat kaynakları)

Şekil 2.9' da, Denklem 2.2' de verilen kaotik fonksiyonun akım aynalarıyla gerçeklendiği devre yapısı verilmiştir. Yapı, anahtarlamalı akım aynalarına dayanmaktadır [18]. Devrenin üst kısmı eğim çarpma ve depolama işlemini yerine getirirken, alt kısmı lineer olmayan ayırma işlemini gerçekler. Ayırma işlemi iki modda çalışır:

$I_1 > 0$ için T18 ve T19 iletimdedir, T20 ve T29 da iletime girerek T17 üzerinden $I_{out}=I_1-I_a$ akımı akar. $I_1 < 0$ için T8 ve T9 iletimdedir, T10 ve T28 de iletime girerek T4 üzerinden $I_{out} = I_1 + I_a$ akımı akar. Φ_1 ve Φ_2 örtüşmeyen iki saat kaynağıdır. I_q,16 μA, I_a

ise 12 μA olarak seçilmiştir. Şekil 2.10' daki grafikte, Şekil 2.9'da verilen devrenin SPICE simulasyonundan elde edilen kaotik harita verilmiştir.

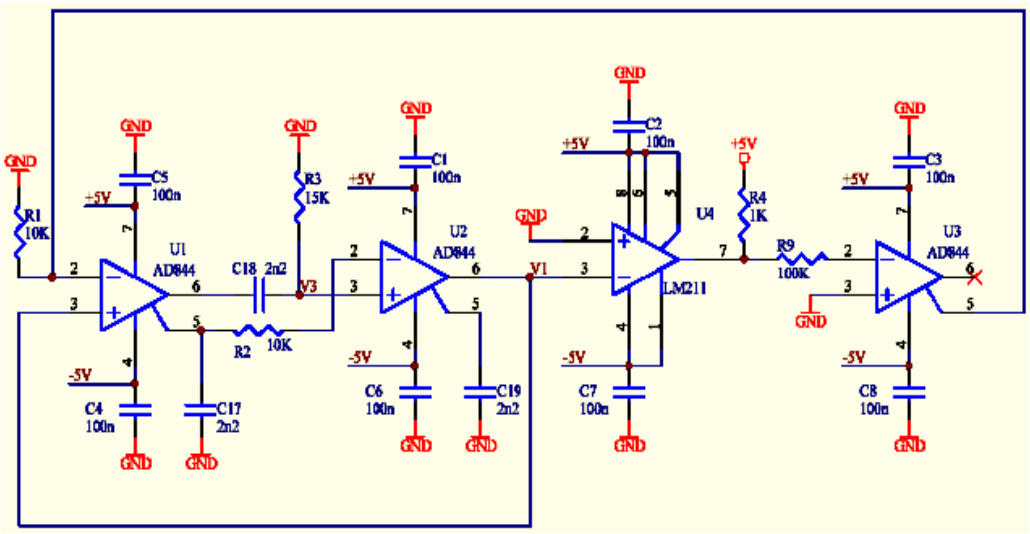


Şekil 2.10. Akım Aynalarıyla Gerçeklenen Kaotik Devrenin SPICE Simulasyonundan Elde Edilen Ayrık Zamanlı Kaotik Harita

Literatürde birçok kaotik osilatör bulunmakla birlikte, bunların pek azı yüksek frekansta performanslı bir şekilde çalışma ve entegre devre haline getirilebilme gibi tasarım kriterleri dikkate alınarak tasarlanmıştır. Bugüne kadar literatürde ayrık zamanlı kaotik haritaların rasgele sayı üretimi amacıyla kullanımına ilişkin bir çok çalışma mevcutken, sürekli zamanlı Kaotik osilatörlerin RSÜ gerçeklemek amacıyla kullanılabileceği çok az çalışmayla gösterilmiştir [16,19]. Sürekli zaman kaotik osilatörlerine örnek olarak, ayrık elemanlar kullanılarak gerçekleştirilen bir devre Şekil 2.11' de verilmiştir [19]. Nüve olarak kullanılacak olan çift sarmallı otonom kaotik osilatör, literatür [20]'de önerilmiş olan kaotik denklemden yola çıkılarak elde edilmiştir. Denklemdeki doğrusalsızlık, sürekli zamanlı doğrusalsızlıkla yer değiştirildiğinde, sistem niteliksel olarak Chua osilatörüne birebir benzemektedir.

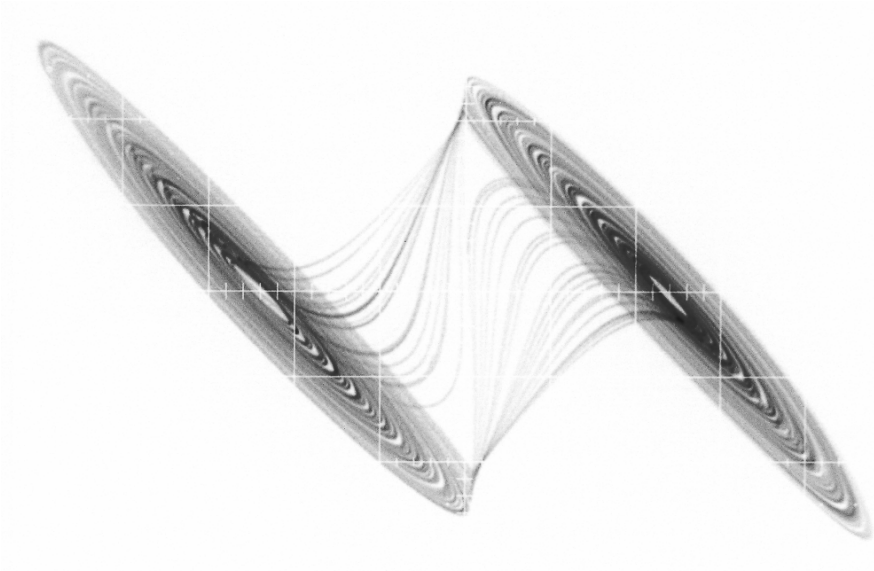
Devrede AD844, yüksek bant genişliğine sahip işlemsel kuvvetlendirici olarak ve LM211 gerilim karşılaştırıcısı gerekli doğrusalsızlığı sağlamak amacıyla kullanılmıştır. Pasif eleman değerleri: $R_1=R_2=aR_3=R=10$ KΩ, $R_3=15$ KΩ ($a=0.666$), $C_{17}=C_{18}=C_{19}=C$,

C=2.2 nF ve R_9=100 KΩ, için gerçek devreden elde edilen kaotik çekici Şekil 2.12' de verilmiştir.



Şekil 2.11. Çift Sarmallı Otonom Kaotik Osilatör Devresi

Ayrık elemanlarla kurulan devrenin, parazitik kapasitelerden etkilenmesini engellemek amacıyla kaotik devrenin temel çalışma frekansı f_{OSC}=(1/2πRC) 7.234 KHz gibi düşük bir değere ayarlanmıştır.



Şekil 2.12. Deneysel Olarak v_1- v_3 Düzleminde Elde Edilen Kaotik Çekici

Sürekli zamanlı kaotik bir sistemden rasgele bit dizileri üretmek için geriye dönüşü mümkün olmayan uzaydaki bir kesitin kullanıldığı yeni yöntemler de mevcuttur. Geriye dönüşün mümkün olmaması rasgele sayı üretimi için kilit özelliklerden bir tanesidir. Buna örnek olarak Poincare dönüşümü olarak bilinen, bir durum değişkeninin diğer bir durum değişkeninin belirli seviye geçişlerinde örneklenmesi yani kaotik bir işaretin başka bir kaotik işaretle örneklenmesi verilebilir [19].

2.4. Sayısal Son İşleme

Başta yapılan rasgele ölçümler, John Von Neumann'ın "0" ve "1" bitlerinin dengeli dağılımını oluşturmak için önerdiği bir görüşü temel alan donanım bazlı bir doğrultucu ile işlenecektir. Von Neumann doğrultucusu (0,1) bit çifti için "1", (1,0) bit çifti için "0" ve (0,0) veya (1,1) bit çiftleri için hiçbir çıkış vermeyerek bit çiftlerini çıkış bitlerine dönüştürmektedir (Tablo 2.1).

Doğrultucu kullanmak, artma eğilimindeki veriden istatistiksel olarak dengeli çıkış üretmek için kolay bir yoldur. Doğrultucu, özellikle çıkışı etkileyerek hızlı saatin çalışma devrindeki dengesizlikleri engeller. Çift osilatör kaynağında oluşabilecek bitler arası korelasyonun etkilerini azaltmak için Von Neumann doğrultucusu geliştirilebilir.

Tablo 2.1 Von Neumann Doğrultucusu

Giriş Bitleri		**Çıkış**
0	0	Hiçbiri
0	1	1
1	0	0
1	1	Hiçbiri

Doğrultucunun önemli olmasının sebeplerinden bir tanesi RSÜ' nün değişken bit hızına sahip olmasıdır. Doğrultucu ortalama olarak her 4 ham ikili sayı örneği için 1 bit üretir. Von Neumann doğrultucusu kullanılarak geliştirilen RSÜ' nün olağanüstü performansı (Yaklaşık olarak doğrultucudan sonra 75 Kbit/s'nin üzerinde) tüm standart kripto uygulamaları için GRSÜ gereksinimlerini aşmaktadır. Fakat değişken hız, bazı olağandışı kullanım senaryolarını güçleştirebilir. Rasgele sayı üretecinin sonlu bir zaman periyodu içinde bir çıkış bit'i üreteceğini ispatlamak mümkün olmasa da, uzun bir gecikme ihtimali ihmal edilebilecek düzeyde küçüktür [6].

Doğrultucu çıkışı 32 bitlik kütükte sıraya sokulmalı ve yazılım katmanı için hazır hale getirilmelidir. Rasgele çıkışların iki kez okunamayacağı ve her okuma işleminin yeni rasgele veri sağlayacağı ara yüz tarafından garanti edilmelidir.

3. BÖLÜM

RSÜ' LERE UYGULANAN İSTATİSTİKSEL TESTLER

3.1. Giriş

Rasgele ve yalancı rasgele sayılar için duyulan ihtiyaç birçok kriptografik uygulamada kendini göstermektedir. Örneğin, yaygın kripto sistemlerin kullandığı anahtarların, rasgele bir şekilde üretilmesi gereklidir. Birçok kriptografik protokoller çeşitli noktalarda yalancı rasgele veya rasgele girdilere gereksinim duyarlar.

Bu bölümde, rasgele sayı ve yalancı rasgele sayı üreteçlerinin rasgelelik testleri incelenmektedir. Bu üreteçler kriptografik, modelleme ve simulasyon uygulamalarını da içeren birçok amaç için kullanılabilir. Bu kısmın ilgi odağı, rasgeleliğin, kriptografik amaçlar için gerektiği uygulamalardadır. Rasgelelik için istatistiksel testlerin bir kısmı, bu bölümde anlatılmaktadır. Ulusal Standartlar ve Teknoloji Enstitüsüsü (NIST), bu prosedürlerin, ikili bir dizinin rasgele olup olmadığının algılanmasında, faydalı olduğunu düşünmektedir. Bununla beraber , testi yapan, test edilen ikili dizide görünen rasgelelikten büyük sapmaların, hem kötü bir şekilde tasarlanmış bir üreteç nedeniyle hem de anormaliklere bağlı olabileceğine dikkat etmelidir (Örneğin, belli bir sayıdaki başarısızlık belirli bir üreteç tarafından üretilen rasgele serilerde beklenebilir). Test sonuçlarının uygun yorumlanması testi yapana bağlıdır.

3.1.1. Rasgelelik

Bir rasgele bit dizisi, hilesiz bir madeni paranın yüzlerine"0" ve "1" etiketi yapıştırılıp yazı-tura atılmasının sonucunun kaydedilmesi ile oluşturulabilir. Her bir yazı-tura atma olayında "0" veya "1" gelmesinin olasılığı tam ½' dir. Ayrıca her atma olayı birbirden bağımsızdır: Herhangi bir önceki madeni para atılışının sonucu, gelecek madeni para atma olaylarına tesir etmez.

Hilesiz madeni para, bu nedenle mükemmel rasgele bit katarı üretecidir. Çünkü "1" ve "0" değerlerinden herbiri, rasgele dağıtılmıştır ("0" ve "1" ler eşdeğer dağılıma sahiptir).

Burada dizinin bütün elemanları birbirini etkilemeden üretilir ve dizideki sonraki elemanın değeri önceden kaç tane eleman üretildiğinden bağımsız bir şekilde tahmin edilemez. Fakat kriptografik amaçlar için hilesiz madeni paraların kullanılması, elverişsizdir. Bununla beraber bunun gibi ideal bir üretecin çıktısı olan gerçek bir rasgele sayı dizisi, rasgele ve yalancı rasgele sayı üreteçlerinin değerlendirilmesi ve karşılaştırması için bir değerlendirme aracı olarak da kullanılabilir.

3.1.2. Tahmin Edilemezlik

Kripto uygulamalar için üretilen rasgele ve yalancı rasgele sayılar tahmin edilemez olmalıdır. YRSÜ' ler söz konusu olduğunda , eğer nüve bilinmiyorsa, dizideki sonraki çıktı sayı, dizide önceki rasgele sayılarla ilgili herhangi bir bilgiye rağmen tahmin edilemez olmalıdır. Bu özellik, sonraki durum tahmin edilemezlik olarak bilinir. Ayrıca nüveyi üretilen değerlerin herhangi birinin bilgisinden belirlemek mümkün olamaz (Geçmiş durum tahmin edilemezliği de gereklidir). Bir nüve ve bu nüveden üretilen herhangi bir değer arasında hiçbir korelasyon olmamalı, dizinin her bir elemanı, olasılığı ½ olan bir bağımsız rasgele olayın neticesi olarak görünmelidir [21].

Sonraki durum tahmin edilemezliği sağlamak için nüveler elde etmede dikkat sarf edilmelidir. Eğer nüve ve üretim algoritması biliniyor olursa, bir YRSÜ tarafından üretilen değerler tamamen tahmin edilebilirdir. Birçok durumda üretim mekanizması açık olduğundan, nüve sır olarak tutulmalı ve ürettiği yalancı rasgele dizisinden tekrar elde edilebilir olmamalı ve ayrıca nüvenin kendisi de tahmin edilemez olmalıdır.

3.2. Test Etme

Çeşitli istatistiksel testler, gerçek rasgele dizinin rasgele olup olmadığını anlamak ve değerlendirmek için o diziye uygulanabilir. Rasgelelik, bir olasılıksal özelliktir. Bir rasgele dizinin özellikleri olasılık açısından tanımlanıp nitelendirilebilir. İstatistiksel testlerin muhtemel sonucu, gerçek bir rasgele diziye uygulandığında, testen önce bilinen olasılık terimleriyle tanımlanır. Sonsuz sayıda olası istatistik test vardır. Bu testlerin herbiri incelenip dizinin rasgele olmadığına işaret edecek örüntülerin var olup

olmadığını değerlendirir. Değerlendirme için o kadar çok test vardır ki, acaba bir dizi rasgeledir veya değildir diye karar vermek için testlerin hiçbir kesin sonlu kümesi, "tam" sayılamaz. Ayrıca belirli bir üreteç hakkında yanlış kararlardan kaçınmak için istatistiksel testlerin sonuçları biraz daha dikkatle incelenmelidir.

İstatistiksel bir test, belirli bir geçersiz hipotezi (H_0) test etmek için formülleştirilir. Buradaki amaç, test altındaki geçersiz hipotez, test edilen dizinin rasgele olduğudur. Bu geçersiz hipotezle ilintili alternatif hipotez (H_a) dizinin rasgele olmadığıdır. Her bir uygulamalı test için , bir karar veya sonuç geçersiz hipotezini kabul edip etmediği çıkarılır [21]. Acaba üreteç üretilen diziyi temel alarak, rasgele değerler üretip üretmiyor mu?

Her bir test için , uygun bir rasgelelik istatistiği seçilmiş olmalı ve geçersiz hipotezin onaylanması veya reddedilmesi için belirlerlenmelidir. Rasgelelik varsayımı altında, bunun gibi istatistiğin olası değerlerinin bir dağılımı vardır.

Geçersiz hipotez altında bu istatistiğe ait teorik bir referans dağılım matematiksel yöntemler tarafından belirlenebilir. Bu referans dağılımdan kritik bir değer tespit edilir (bu değer, genellikle "istisna" dağılımın sonlarında, genellikle % 99 noktasındadır). Bir test süresince , test istatistiğine ait değer, veride hesaplanır (dizi test ediliyor). Bu test istatistiğine ait değer, kritik değerle karşılaştırılır. Eğer test istatistiğe ait değer kritik değeri geçerse, rasgelelik için geçersiz hipotez reddedilir. Aksi takdirde geçersiz hipotez (Rasgelelik hipotezi) reddedilmez (H_a hipotezi kabul edilir) [21,22].

Uygulamada istatistiksel hipotez testinin çalışmasının sebebi referans dağılımın ve kritik değerin rasgeleliğe bağımlı olması ve rasgelelik varsayımının altında üretilmiş olmasıdır. Eğer rasgelelik varsayımı el altındaki veri için gerçek olursa, veride hesaplanan test istatistik değeri çok düşük bir olasılığa sahip olacak ve (% 0.01) kritik değeri geçecektir.

Diğer taraftan, olaya istatistiksel hipotez test etme açısından bakıldığında, hesaplanmış test istatistiğine ait değerin kritik değeri geçmesi, düşük olasılık olayının meydana gelmemiş olduğunu gösterir. Bu yüzden, hesaplanmış test istatistiğine ait değer kritik değeri geçince, rasgelelik hakkındaki orijinal varsayım şüpheli veya hatalı olduğu sonucuna varılır. Bu durumda, istatistiksel hipotez testi şöyle ifade edilir: H_0' ı reddedin (rasgelelik) ve H_a' yı kabul edin (rasgele olmama).

İstatistiksel hipotez testi, bir sonuç-üretim prosedürü olup bunun iki olası neticesi vardır: Ya H_0 (Veri, rasgeledir) ya da H_a (Veri rasgele değildir). Aşağıdaki tablo test prosedürü kullanılarak ulaşılan sonuç ile eldeki verinin gerçek (bilinmeyen) durumunu ilişkilendirir (Tablo 3.1).

Tablo 3.1 Sonuç Veri İlişkisi

GERÇEK DURUM	SONUÇ	
	H_0 Kabul edilir	H_a Kabul edilir
Veri rasgeledir (H_0 doğrudur)	Hata yok	Tip I hata
Veri rasgele değildir (H_a doğrudur)	Tip II hata	Hata yok

Eğer veri gerçekten rasgele ise geçersiz hipotezin red kararını alması (verinin rasgele olmadığına karar verilir) çok kısa bir zaman alacaktır. Bu çıkarım Tip I hatası olarak kabul edilir. Eğer veri gerçekte rasgele değilse (verinin rasgele olduğuna karar verilir), geçersiz hipotezi kabul etmek Tip II hatası olarak adlandırılır. Veri gerçekten rasgele iken H_0 hipotezini kabul eden veya veri gerçekten rasgele değilken H_0 hipotezini reddeden kararlar doğrudur.

Tip I hatasının olasılığı testin ayırt edilebilme düzeyini gösterir. Bu olasılık, testte öncelikle ayarlanır ve bu olasılık α olarak gösterilir. α, dizi gerçekte rasgele iken testin 'rasgele değil' sonucunu işaret etme olasılığını ifade eder. Bu, 'iyi' bir üreteç diziyi oluşturduğunda bile oluşturulan dizinin rasgele olmayan özelliklerinin bulunabileceğini gösterir. Kriptografide *α' nın* ortalama değeri 0.01 civarındadır.

Tip II hatasının olasılığı β ile gösterilir. β, gerçekte rasgele testin 'rasgele' sonucunu işaret etme olasılığını gösterir. Bu, kötü bir üreteç diziyi oluşturduğunda, oluşturulan dizinin rasgele özellikler gösterebileceğini ifade eder. α'dan farklı olarak β sabit bir sayı değildir. β birçok farklı değer alabilir çünkü bir veri dizisinin rasgele olmaması için sonsuz yol vardır ve her yol değişik bir β üretir. Tip II hatasının hesaplanması, birçok olası rasgele olmama durumu olabileceğinden, Tip I hatasının hesaplanmasına göre daha zordur. Aşağıdaki testlerin en birincil amaçlarından biri Tip II hatasının olasılığını minimize etmektir. Örneğin, kötü bir üreteç tarafından üretilen dizinin iyi bir üreteç

tarafından üretilmiş gibi kabul edilmesinin olasılığını minimize etmek gibi. α ve β olasılıkları birbiriyle ilintili olduğu kadar test edilen dizinin boyuyla (n) da ilintilidir. Eğer bu üç parametreden ikisi açıkça tanımlı ise üçüncüsü otomatik olarak belirlenebilir. Uzmanlar genelde örnek dizi boyunu (n) ve α değerini kendileri belirler. Ardından en küçük β değerini verecek kritik nokta belirlenir. Bu, kötü bir üretecin gerçekten rasgele bir dizi üretme olasılığını kabul edilebilir bir çerçevede belirleyen uygun bir dizi boyu seçmek anlamına gelir. O halde doğru kabul etme için uygun bir kestirim noktası, bir diziyi yanlışlıkla rasgele kabul etme olasılığının minimum olası değerini verecek biçimde seçilebilir [21,22].

Her bir test, verinin bir fonksiyonu olan hesaplanmış bir test istatistiğine ait değere dayanır. Eğer test istatistiğine ait değer S ve kritik değer *t* olarak alınırsa, Tip I hata olasılığı $P(S>t \parallel H_0 \text{ doğru}) = P(H_0 \text{ red} \mid H_0 \text{ doğru})$, ve Tip II hata olasılığı da $P(S \leq t \parallel H_0 \text{ yanlış}) = P(\text{kabul } H_0 \mid H_0$ yanlış) olarak bulunur. Test istatistiği Geçersiz hipoteze karşı delil kuvvetinin özeti olan *P*-değeri hesaplanırken kullanılır**.** Bu testler için, *P*-değeri test edilen diziden daha düşük rasgelelikte dizi üretebilen mükemmel bir RSÜ' nün olasılığını belirtir ve test tarafından değerlendirilirken rasgele olmama (non-randomness) çeşidini verir. Eğer bir test için *P*-değeri "1" 'e eşit elde edilirse, o zaman dizinin mükemmel rasgelelikte olduğu söylenir. "0" olan bir *P*-değeri de dizinin tamamiyle rasgele olmadığını belirtir. Testler için bir anlam seviyesi (α) seçilebilir. Eğer *P*-değeri > α ise o zaman Geçersiz hipotez kabul edilir. Örneğin dizi rasgeledir gibi. Eğer *P*-değeri < α ise, o zaman Geçersiz hipotez reddedilir. Örneğin dizi rasgele değildir gibi. α parametresi, Tip I hatasını belirtir. Tipik olarak α [0.001, 0.01] aralığında seçilir.

α değerinin 0.001 olması, eğer dizi rasgele ise testte her 1000 diziden bir tanesinin reddedileceğinin beklendiğini gösterir. *P*-değeri > 0.001 için, bir dizinin % 99.9 güvenlikli rasgele olması beklenir. *P*-değeri < 0.001 için ise bir dizinin % 99.9 güvenlikli rasgele olmaması beklenir.

α değerinin 0.01 olması, eğer dizi rasgele ise testte her 100 diziden bir tanesinin reddedileceğinin beklendiğini gösterir. *P*-değeri > 0.01 için, bir dizinin % 99 güvenlikli rasgele olması beklenir. *P*-değeri < 0.01 için ise bir dizinin % 99 güvenlikli rasgele olmaması beklenir.

Buradaki örnekler için α=0.01 olarak seçilmiştir. Şu da unutulmamalıdır ki, çoğu durumda, örneklerdeki parametreler tavsiye edilen değerlerle uyuşmaz. Örnekler sadece tanım amaçlı verilmiştir.

3.3. Rasgelelik, Tahmin Edilemezlik ve Test Hakkında Düşünceler

Aşağıdaki varsayımlar, test edilen ikili dizilerle ilgili olarak yapılmıştır.

1. Tekdüzelik: Rasgele ve yalancı restgele bit dizilerinin oluşturulması sırasındaki her bir noktada, "0" ve "1" oluşu hemen hemen eşittir. Örneğin, her birinin olasılığı ½ civarındadır. "0" ve "1" lerin beklenen sayısı $n/2$ dir ve burada n dizinin uzunluğuna eşittir.

2. Ölçeklenebilirlik: Bir dizi için kabul edilen bir test aynı zamanda rasgele çıkarılmış alt dizilere de uygulanabilir. Eğer bir dizi rasgele ise, o zaman çıkarılmış her alt dizi de rasgeledir. Bundan dolayı, her alt dizi de rasgelelik testini geçmelidir.

3. Tutarlılık: Bir üretecin davranışı başlangıç değerlerine (nüve) karşı dayanıklı olmalıdır. Tek bir nüveden alınan çıktıyı temel alan YRSÜ'yü test etmek veya fiziksel bir çıktıdan üretilen çıktıyı temel alan RSÜ'yü test etmek için yetersizdir.

3.4. RSÜ lerine Uygulanan NIST İstatistiksel Testleri

NIST test suiti, yazılım veya donanım tabanlı kriptografik rasgele veya YRSÜ' ler tarafından üretilen yeterince uzun ikili dizilerin rasgeleliğini ölçmek için geliştirilmiş, 16 test içeren istatistiksel bir pakettir. Bu testler bir dizi içinde rasgele olmayan durumların çeşitliliğine odaklanır. Bazı testler çeşitli alt testlere ayrılabilir. Bu 16 test aşağıda verilmiştir [21]:

1. Frekans (monobit) testi
2. Bir blok içinde frekans testi
3. Yinelemeler (Runs) testi
4. Bir blok içinde en-uzun-bir-yinelemesi (longest-run-of-ones) testi
5. İkili matris rankı testi
6. Ayrık fourier dönüşümü (spektral) testi
7. Örtüşmeyen şablon eşleştirme (Non-overlapping template matching) testi
8. Örtüşen şablon eşleştirme (Overlapping template matching) testi

9. Maurer'in "Evrensel İstatistik" testi

10. Lempel-Ziv sıkıştırma testi

11. Doğrusal karmaşıklık (linear complexity) testi

12. Seri (serial) test

13. Yaklaşık entropi (approximate entropy) testi

14. Kümülatif toplamlar (cumulative sums – cusums) testi

15. Rasgele gezinimler (random excursions) testi

16. Rasgele gezinimler değişken (random excursions variant) testi

Test suitindeki testlerin uygulamalarının sırası tamamen isteğe bağlıdır. Ancak, bir dizide rasgele olmayan bölgelerin varlığı ile ilgili temel ipuçları veren frekans testinin öncelikle uygulanması önerilmektedir. Eğer bu test başarısız olursa, diğer testlerin de başarısız olma olasılığı yüksektir. Zaman kriteri açısından en karmaşık test doğrusal karmaşıklık testidir.

Test süitindeki bazı testler, referans dağılım olarak standard normal ve chi-square (χ^2) dağılımlarından oluşmaktadır. Eğer test altındaki dizi, gerçekte rasgele değilse, hesaplanan test istatistiği, referans dağılımın ekstrem bölgelerinde başarısız olur. Standart normal dağılım, RSÜ'den elde edilen test istatistik değeri ile rasgelelik varsayımı altındaki istatistiğin beklenen değerinin karşılaştırılmasında kullanılır. Standard normal dağılımın test istatistiği $z = (x - \mu)/\sigma$ biçimindedir. Burada x, örnek istatistik test değeri, σ^2 ve μ ise varyans ve beklenen değer olarak verilir. χ^2 dağılımı ise örnek bir ölçümün gözlemlenen frekansları ile buna denk düşen farzedilen dağılımın beklenen frekansları arasındaki uyumun-iyiliği (goodness-of-fit) kriterinin karşılaştırılmasında kullanılır. Test istatistiği $\chi^2 = \sum\left((o_i - e_i)^2 / e_i\right)$ biçimindedir. Burada o_i ve e_i, meydana gelen ölçümlerin gözlenen ve beklenen frekanslarıdır [21].

Bu suitteki testlerde, dizi uzunluğu n, 10^3 ila 10^7 mertebesinde varsayılır. Bunun gibi *n*' in büyük örnek değerleri için asimptotik referans dağılım türetilmiş ve testlerde uygulanmıştır. Testlerin çoğu genel olarak daha küçük *n* değerleri için uygulanabilmektedir. Ancak, küçük *n* değerlerinin kullanılması asimptotik referans dağılımları için çok uygun olmamakla beraber bu dağılımların yerine çok daha zor hesaplamalar gerektiren tam doğru dağılımlar kullanılabilir.

3.5. Testlerin Açıklaması

3.5.1. Frekans (Monobit) Testi

Bu test, tüm dizideki sıfırlarla birlerin oranının gözlemlenmesi ilkesine dayanır. Bu testin amacı, gerçek bir RSÜ'de olması beklenen bir özellik olan bir dizideki "0" ve "1" sayılarının yaklaşık olarak aynı olup olmadığının ölçülmesidir. Bu testi takip eden diğer tüm testlerin uygulanması, bu testi geçme ön koşuluna bağlıdır. Bu testin sonucu, dizinin rasgele olup olmadığı ile ilgili ciddi bilgi verir [21,23].

3.5.2. Bir Blok İçinde Frekans Testi

Bu test, M bitlik bir blok içindeki "1" lerin oranının gözlemlenmesi ilkesine dayanır. Bu testin amacı, rasgeleliğin varsayıldığı bir dizi içindeki M bitlik blokta bulunan "1" lerin frekansının M/2 olup olmadığının belirlenmesidir. M=1 uzunluğundaki bir blok için yapılan test aslında tam olarak Frekans (Monobit) Testine denk düşer [21,24].

3.5.3. Yinelemeler Testi

Bu test, bir dizi içindeki yinelemelerin gözlemlenmesine odaklıdır. Bir dizi içindeki yineleme (run) kesintiye uğramayan özdeş bit dizisi olarak tanımlanmaktadır. k uzunluğundaki bir yineleme tam olarak k özdeş bit içerir ve bu bit dizisi başında ve sonunda kendinden farklı bir bit ile sınırlıdır. Bu testin amacı, rasgele olduğu iddia edilen dizideki "1" ve "0" yineleme sayılarının beklendiği gibi mi yoksa beklenenden farklı mı olduğunun araştırılmasıdır. Bu test özellikle sıfırlar ve birler arasındaki osilasyonun hızlı veya yavaş olup olmaması ile ilgili çok önemli bilgiler verir.

Klasik parametrik olmayan testin bu biçimi birbirini takip eden "1"lerin ve "0" ların bulunduğu koşturmaları inceler ve bu homojen alt diziler arasındaki geçişlerin hızına karar verir. Burada özgül test toplam takip etme dağılım tabanlı bir testtir [21,25].

3.5.4. Bir Blok İçinde En-Uzun-Bir-Yinelemesi (Longest-Run-Of-Ones) Testi

Bu test, bir dizi içindeki en-uzun-bir-yinelemelerinin gözlemlenmesine odaklıdır. Bu testin amacı, rasgele olduğu iddia edilen dizideki "1" lerden oluşan en uzun yineleme uzunluğunun, bir rasgele dizide beklenen değerle tutarlı olup olmadığının araştırılmasıdır. Ancak unutulmamalıdır ki, test dizisindeki en-uzun-bir-

yinelemesindeki düzensizlikler en-uzun-sıfır-yinelemesinde de görülebilir. O nedenle sadece en-uzun-bir-yinelemesi testi yeterlidir [21,26].

3.5.5. İkili Matris Rankı Testi

Bu test, tüm dizinin ayrışık altmatrislerinin ranklarına odaklı bir testtir. Testin amacı, orijinal dizinin sabit uzunluktaki altkatarları arasında doğrusal bir bağıntının varlığını araştırmaktır. Bu test ayrıca DIEHARD test suitinde de bulunmaktadır [9,27].

Rasgelelik testine başka bir yaklaşım da orjinal diziden elde edilen sabit uzunlukta alt diziler arasındaki doğrusal bağımlılığın bulunmasıdır; birbirini takip eden sıfırlardan ve birlerden matrisler oluşturup, satırlar ve sütunlar arasındaki doğrusal bağımlılığa bakılır. Rank'ın teorik olarak beklenen değerden sapması veya eksikliği istatistik olarak rasgelelik hakkında bize bilgi verir.

Bu test Kovalenko (1972) 'nun araştırmasından elde edilen sonuçlardan elde edilmiş ve Marsaglia ve Tsay (1985)'ın çalışması ile formülize edilmiştir [27].

3.5.6. Ayrık Fourier Dönüşümü (Spektral) Testi

Bu test, dizinin ayrık Fourier dönüşümündeki tepe yüksekliklerine (peak heights) odaklı bir testtir. Testin amacı, test edilen dizinin içindeki rasgelelik varsayımındaki sapmaya işaret eden periyodik özniteliklerin sezimlenmesidir. Pratik olarak tepe sayısının, % 95 eşik değerini aştığı durum rasgelelik için iyi bir ölçüt anlamına gelir.

Burada tanımlanan test, ayrık Fourier dönüşümü üzerine dayalıdır. Bu spektral bir yöntemdir. Fourier testi bit serilerindeki periyodik özellikleri belirler ki bunlar da rasgeleliğin varsayımından bir sapma belirlerler.

3.5.7. Örtüşmeyen Şablon Eşleştirme (Non-overlapping Template Matching) Testi

Bu test, önbelirtimli (pre-specified) hedef dizisinin bulunma sıklığının gözlenmesine dayanır. Testin amacı, üretecin oluşturduğu birçok periyodik olmayan (aperiyodik) örüntünün sezimlenmesidir. Bu ve Bölüm 3.5.8'de tanıtılan Örtüşen Şablon Eşleştirme testlerinde, m bitlik bir örüntüyü taramak için m bitlik bir pencere kullanılır. Eğer örüntü bulunmazsa, pencere bir bit kaydırılarak tarama işlemine devam edilir. Eğer örüntü bulunursa, pencere bulunan örüntüden sonraki ilk bite yeniden konumlanır ve

taramaya devam edilir. Bu test verilmiş periyodik olmayan bir durumda çok fazla ve ya çok az karşılaşmaların sergilendiği seriyi reddeder [28].

3.5.8. Örtüşen Şablon Eşleştirme (Overlapping Template Matching) Testi

Bu test, önbelirtimli (pre-specified) hedef dizisinin bulunma sıklığının gözlenmesine dayanır. Testin amacı, Bölüm 3.5.7'de tanıtılan testte olduğu gibi üretecin oluşturduğu birçok periyodik olmayan (aperiyodik) örüntünün sezimlenmesidir. Bu ve Bölüm 3.5.7'de tanıtılan Örtüşmeyen Şablon Eşleştirme testlerinde, gene m bitlik bir örüntüyü taramak için m bitlik bir pencere kullanılır. Eğer örüntü bulunmazsa, pencere bir bit kaydırılarak tarama işlemine devam edilir. Bu testin Bölüm 3.5.7'deki testten tek farkı, eğer örüntü bulunursa pencere bulunan örüntüden sonraki ilk bit yerine sadece o an bulunan pozisyondan bir sonraki bite yeniden konumlanır ve taramaya devam edilir [29].

3.5.9. Maurer'in "Evrensel İstatistik" Testi

Bu test, bir dizide eşleşen örüntülerin bit dizisinin gözlenmesine dayanır. Bu testin amacı, dizinin bilgi kaybı olmadan anlamlı bir şekilde sıkıştırılıp sıkıştırılamayacağının tespit edilmesidir. Anlamlı bir biçimde sıkıştırılabilen bir dizi rasgele bir dizi olarak kabul edilmez.

Bu test Princeton Üniversitesi'nin Bilgisayar Bilimi Bölümü'ndeki Ueli Maurer tarafından 1992' de sunulmuştur. Maurer' in önerdiği test istatistiği, bit başına entropi ile yakından ilgilidir ki Maurer bunun "kriptografik uygulamalarda kalite kontrolü için gizli bir anahtar" olabileceğini iddia eder. Aslında bu testin, bir bozukluğun gerçek kriptolojik anlamını ölçeceği iddia edilir. Çünkü bu bir düşmanın seçmeli anahtar arama stratejisi ile ilgilidir veya şifreleyici bir sistemin etkili anahtar boyutudur. Test çok özel bir durumu veya istatistiksel bir hatanın çeşidini belirlemek için tasarlanmamıştır. Aslında, test sonlu bellekli ergodik durağan bir kaynakla modellenebilen istatistiksel hataların çok genel sınıflarının her hangi birisini belirleyebilmek için tasarlanmıştır. Bu yüzden, Maurer testin standart istatistiksel testlerin bir sayısını içerdiğini iddia eder [30].

Test, Ziv fikrine dayanan kısaltılmış çeşitte bir test olduğundan evrensel istatistiksel bir test, evrensel kaynak kodlu bir algoritmaya dayandırılabilir. Bir RSÜ yalnız çıktı serisi

belirgin bir şekilde kısaltılamazsa testi geçmelidir. Maurer'e göre, Lempel-Ziv'e dayanan kaynak kodlu algoritmanın istatistiksel test olarak uygulanması için daha az uydurulduğudur. Çünkü dağılımın kararlı veya yaklaşık olduğu istatistiksel bir testin tanımlanmasının zor olduğu görünmektedir.

3.5.10. Lempel-Ziv Sıkıştırma Testi

Bu test, genel olarak, bir dizideki kümülatif olarak ayrık örüntülerin (sözcüklerin) sayısına odaklanır. Amaç, test edilen dizinin en çok ne kadar sıkıştırılabileceğinin belirlenmesidir. Eğer dizi anlamlı bir biçimde sıkıştırılamazsa, dizinin rasgele olduğu kabul edilir. Rasgele dizi belli sayıda karakteristik ayrık örüntüye sahiptir.

Bu test, rasgele dizi adayını Lempel-Ziv [30] algoritmasını kullanarak sıkıştırır. Eğer sıkıştırma teorik olarak beklenen sonuca göre istatistiksel olarak kayda değer miktarda fazlaysa dizinin rasgele olmadığına hükmedilir. Bir üreteci test etmek için çok sayıda dizi bu şekilde test edilir. Her dizi için anlamlılık olasılıkları hesaplanır ve anlamlılık olasılıklarının düzgün dağılımlı olduğu hipotezi, mesela Kolmogorov-Smirnoz testi kullanılarak test edilir.

Lempel-Ziv testinin frekans ve koşu testlerini, diğer sıkıştırma testlerini ve muhtemelen spektral testini kapsadığı düşünülür, ama rasgele ikili matris rankı testiyle kesişebilir. Bu test entropi testiyle özdeştir, hatta Maurer'in evrensel istatistik testiyle daha da özdeştir. Bununla birlikte, Lempel-Ziv testi doğrudan doğruya modern bilişim teorisini tanımlayan sıkıştırma olgusunu içerir.

3.5.11. Doğrusal Karmaşıklık (Linear Complexity) Testi

Bu test genel olarak, Doğrusal Geribeslemeli Kayan Yazmaç (DGKY) (Linear Feedback Shift Register (LFSR)) uzunluğuna odaklıdır. Testin amacı, rasgele olduğu iddia edilen dizinin yeterince karmaşık olup olmadığının belirlenmesidir. Rasgele diziler daha uzun DGKY ile karakterize edilir. Çok kısa DGKY'ler büyük olasılıkla rasgele olmayan dizileri gösterir [2].

3.5.12. Seri (Serial) Test

Bu test, tüm dizideki m-bitlik örtüşen olası örüntülerin frekansına odaklanır. Testin amacı, 2^m adet m-bitlik örtüşen örüntülerin sayısının, rasgele bir dizide beklenen sayıya

ne kadar yakın olduğunun araştırılmasıdır. Rasgele dizilerde tek-biçimlilik (uniformity) önemli bir özelliktir. Bu tek-biçimlilikte her *m*-bitlik örüntünün diğer *m*-bitlik örüntüler gibi ortaya çıkma olasılığının aynı olması beklenir. Unutulmamalıdır ki, *m*=1 için Seri Test Bölüm 3.5.1'de tanıtılan Frekans Testine denk düşer [2].

3.5.13. Yaklaşık Entropi (Approximate Entropy) Testi

Bölüm 3.5.12'de tanıtılan Seri Test gibi, Yaklaşık Entropi Testi de tüm dizideki *m*-bitlik örtüşen olası örüntülerin frekansına odaklanır. Testin amacı, iki ardışıl uzunlukta (*m* ve *m*+1) örtüşen bloğun frekansını, rasgele bir dizinin beklenen frekans değeri ile karşılaştırmaktır [31].

3.5.14. Kümülatif Toplamlar (Cumulative Sums) Testi

Bu test, bir dizide ayarlanmış (-1, +1) dijitlerin kümülatif toplamı olarak tanımlanan, rasgele yürüyüş (random walk) maksimal gezinimlerin araştırılmasına dayanır. Testin amacı, test edilen dizide bulunan kısmi altdizilerin kümülatif toplamlarının, rasgele olduğu bilinen bir dizinin beklenen değerine göre çok büyük veya çok küçük olup olmadığının belirlenmesidir. Bu kümülatif toplam, rasgele bir yürüyüş olarak kabul edilebilir. Rasgele dizilerde, rasgele yürüyüş gezinimleri sıfır civarındadır. Rasgele olmayan diziler için bu gezinimler sıfırdan uzakta seyreder.

Bu test +/-1'lerden oluşan bir dizinin kısmi toplamlarının maksimum mutlak değerinin hesaplanmasına dayanır. Bu istatistikteki yüksek değerler, dizinin öncül halinde çok fazla "1" veya çok fazla "0" olduğuna işaret eder. Düşük değerler ise "1" ler ve "0" ların dizide aşağı yukarı aynı oranda bulunduğunu gösterir [26].

3.5.15. Rasgele Gezinimler (Random Excursions) Testi

Bu test, kümülatif toplam rasgele yürüyüşündeki tam olarak K ziyaretlik çevrimin sayısına odaklanır. Kümülatif toplam rasgele yürüyüşü, (0,1)' lerden oluşan dizinin (-1,+1)'e ayarlandıktan sonra elde edilen kısmi toplamlarından elde edilir. Bir rasgele yürüyüş çevrimi, rasgele seçilen bir yerden tam bir çevrim olana kadar belli bir uzunluktaki adım dizisini içerir. Bu testin amacı, bu çevrim sırasında rasgele dizide beklenen sapmadan kaynaklanan özel bir durumun ziyaret edilme sayısının belirlenmesidir. Bu test aslında 8 testten ve çıkarımlarından oluşan bir seridir [26].

3.5.16. Rasgele Gezinimler Değişken (Random Excursions Variant) Testi

Bu testte, bir kümülatif toplam rasgele yürüyüşünde, özel bir durumun ziyaret edilme sayısı ölçülür. Testin amacı, bir rasgele yürüyüşte özel durumların beklenen ziyaret sayısındaki sapmalarının gözlenmesidir. Bu test aslında 18 testin (ve çıkarımların) serisidir [21,26].

3.6. FIPS 140-2 İstatistiksel Test Protokolü

Federal Bilgi İşleme Standartları olarak geçen bu testler RSÜ' lerin çıktılarının rasgele olmadıklarını belirler. Dört testen ibaret olup herhangi birisinin olumsuz olması durumunda diğer testler uygulanmayıp o dizinin rasgele olup olmadığına karar verilir. Tez çalışmasında FIPS 140-2 testleri FPGA (Alan Programlanabilir Kapı Dizileri) içine donanımsal olarak gömülmüş olup gerçek zamanda RSÜ çıktıları üzerinde rasgelelik testlerini yapılmaktadır.

Monobit Test: Alınan 20000 örnekten, "1" lerin sayısının 9725 ile 10275 arasında olması gerekir. Bu test sonucunda dizideki, "1" lerin ve "0" ların dağılımlarının biribirine yakın olması beklenir.

Poker Test: Alınan 20000 örnek, 4-bitlik 5000 parçaya bölünür. 4-bitin olası 16 durumu sayılır. Sayıların karelerinin toplamını alır, 16/5000 ile çarpıp 5000 çıkarır. Sonucun 2.16 ile 46.17 arasında olması gerekir.

Runs Test: Alınan 20000 örnekten ardışıl gelen "1" lerin ve "0" ların sayısı sayılır. Bu koşuldaki hem "0" hem de "1" ler için gözlenen ardışıl dizideki bit sayısının,

- 1 uzunluklu dizi için 2343-2657 arasında,
- 2 uzunluklu dizi için 1135-1365 arasında,
- 3 uzunluklu dizi için 542-708 arasında,
- 4 uzunluklu dizi için 251-373 arasında,
- 5 uzunluklu dizi için 111-201 arasında,
- 6 veya daha fazla uzunluklu dizi için 111-201,

arasında olması gerekir.

Örnek : 10010001101010010000110011100 dizisi için,

- 1 uzunluklu “1” sayısı = 4
- 1 uzunluklu “0” sayısı = 2
- 2 uzunluklu “1” sayısı = 2
- 2 uzunluklu “0” sayısı = 4
- 3 uzunluklu “1” sayısı = 1
- 3 uzunluklu “0” sayısı = 1
- 4 uzunluklu “1” sayısı = yok
- 4 uzunluklu “0” sayısı = 1’ dir

Long Runs Test: Alınan 20000 örneğin içinde, ardışıl 26 veya daha fazla uzunlukta “0” veya “1” dizisi olmamalıdır.

4. BÖLÜM

PCI (PERIPHERAL COMPONENT INTERCONNECT) VERİ YOLU

Bu bölümde, günümüz kişisel bilgisayarlarında, iş istasyonlarında ve endüstriyel amaçlı birçok sistemde standart olmuş PCI veri yolu hakkında genel bilgi verilmiştir. Tez çalışması sırasında gerçeklenen donanım kartının bilgisayar arayüzü bu veri yolu ile gerçeklenmiştir.

Bilgisayarlar üzerinde koşan işletim sistemleri, her geçen gün gelişen yazılım uygulamaları nedeniyle giderek daha büyük miktarlarda veri transferine gerek duymaktadır. Bu gereksinim, öncelikle, grafik kartları ve video uygulamaları ile başlamıştır. Giderek, ağ (network) kartı, ses kartı, ekran kartı gibi donanımlarda büyük veri transferleri gerçeklenmeye başlanmıştır [32].

Windows, Linux v.b gibi işletim sistemlerinde bulunan kullanıcı arayüzleri, görsel kullanım, kontrol imkanı veren pencereleri açıp kapatırken, büyültme küçültme yaparken ve özellikle fare işaretçisini ekranda gezdirirken aşırı derecede hızlı ekran görüntüsü güncellemesi gerektirmektedir. Bu güncellemelerin kullanıcı tarafından makul görülebilecek süreler içinde olması gerektiği açıktır. Görüntüyü oluşturan bilgiler sistem belleğinde bulunduğundan, işlemci her defasında yüksek miktardaki veriyi sistem belleğinden video belleğine çok hızlı bir şekilde aktarmak zorundadır [32,33].

Bilgisayarlarda bulunan I/O (giriş-çıkış) kontrol devreleri, hedef giriş-çıkış portu ile sistem belleği arasında veri transferi yapmaktadır. Bilgi saklama üniteleri, sabit disk' ler, CDROM aygıtları, yedekleme sistemleri, genellikle giriş-çıkış devreleri kontrol merkezine bağlıdır. Bu merkez ile sistem belleği arasındaki veri transferini olumlu yönde etkileyebilecek en ufak bir yenilik, sistemin genel performansını büyük ölçüde arttıracaktır. Dosya transferi ya da baskı alınması için ağ kartları kullanıldığında yine sistem belleği ile ağ kartı arasındaki veri aktarımı hızı tüm sistemin başarısı üzerinde etkili olacaktır.

Hızlı veri aktarımının önemli olduğu bir başka uygulama olarak, bilgisayar alt yapısı kullanılarak gerçeklenmek istenen canlı video konferans görüşmelerini inceleyelim. Konferans görüşmesi yapmak isteyen kullanıcının bilgisayarındaki ekran çözünürlüğü 1280x1024, nokta başına renk sayısı 16 milyon seçilmiş olsun. Video çerçevesi saniyede 10 kez tekrarlansın. Bu şartlarda, video belleğine aktarılacak veri saniyede 39.3216 Mbyte' tır. Ekranda konferans yapılan uzak sistemin görüntüsünün ön izleme penceresi (küçültülmüş görüntü) bulunsun. 320x240 çözünürlük, 256 renk ve saniyede 10 çerçeve güncellemesi ile video belleğine aktarılacak veri saniyede 2.3 Mbyte' tır. Ekranda ayrıca iki farklı uzak noktayla daha konferansa ait görüntü pencereleri olduğunu düşünelim. Bunların özellikleri de 640x480 çözünürlük, 256 renk, saniyede 30 çerçeve olursa video belleğine aktarılacak veri saniyede 9.2 Mbyte eder. Üç video görüntüsüne ait sıkıştırılmış bilgi, işlemci üzerinden akacağından bunları da hesaba katalım. Üçüne ait sıkıştırılmış video bilgisi yaklaşık 600 Kbyte olacaktır. Gereken tüm veri aktarım hızlarını toplarsak saniyede 60.516 Mbyte yapmaktadır. Bu hesabın içerisinde bilgisayarın diğer donanımlarının örneğin ağ kartının ya da sabit diskin, veri iletişim ihtiyaçları yoktur [33].

Görülüyor ki, kullanıcının isteklerini yeteri derecede karşılayacak sıradan bir bilgisayar donanımında, bilgisayarın işlemcisi ile diğer donanımlar arasında oldukça hızlı veri aktarımı yeteneği gerekmektedir. Yüksek hızlı veri aktarımı sorununa zamanla yeni çözümler, yeni mimariler tasarlanmaktadır. Bunlardan biri de PCI veri yoludur. PCI veri yolunun en önemli özelliklerinden biri, bilgisayarda bulunan donanımlar arasında, işlemciden bağımsız veri aktarımı imkanı vermesidir.

4.1. PCI Veri Yolu Özellikleri

PCI veriyolunu kullanacak donanımların sağlaması gereken standartlar PCI-SIG (Peripheral Component Interconnect - Special Interest Group) tarafından belirlenmektedir. Standartlar şu dört madde ile özetlenebilir [32,33]:

- Zamanlama uyumluluğu (Timing),
- Mekanik boyutlar,
- Protokol,
- Elektriksel uyumluluk,

PCI veriyolu şekil 4.1'de görüldüğü gibi merkezi işlemcinin (processor) çalışması sırasında hemen hemen tüm uygulamaları için kullanılmaktadır. Bilgisayara bağlı çevre birimleri arasında veri aktarımı, modern bilgisayarlarda şekilde görüldüğü gibi bir ortak kullanımlı veri yolu üzerinden olmaktadır. Şekil 4.1'de merkezi işlem birimi (CPU-Central Processing Unit), *host bridge* adı verilen bir mekanizmanın arkasında yer almaktadır. *Host bridge* merkezi işlem biriminin kendi yerel veri yolu ile sistemin ortak veri yolları arasında arabuluculuk yapmaktadır. Birden fazla PCI veri yolu olan sistemlerde benzer yapıda olmak üzere veri yolu sayısı kadar *host bridge* bulunmaktadır [32,33].

PCI veri yolunda, veri transferinde "öncü (initiator)", "hedef (target)", "agent"(aygıt) gibi anahtar kelimeler kullanılmaktadır.

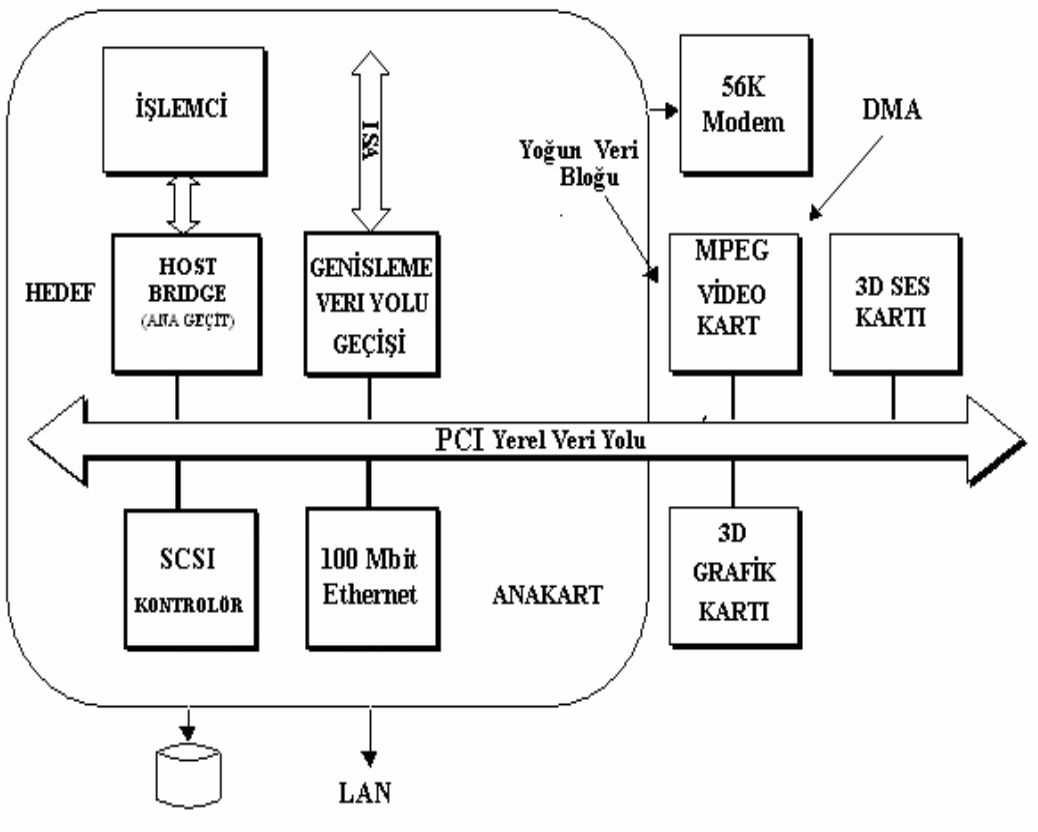
Öncü: Veri yolunu elde ettiğinde veri transferini gerçekleştirir. Her öncü aynı zamanda hedef olarak davranabilir ve "*master"* olarak da adlandırılır.

Hedef: Yazma veya okuma, veri transferinin yapıldığı aygıttır ve "*Slave*" olarakta adlandırılır.

Agent: PCI veri yolu üzerinde herhangi bir aygıt. Öncü veya hedef.

PCI veri yolunun önemli özelliklerinden biri, veri yolunun işlemciden bağımsız olarak da kullanılabilmesidir. Genellikle, iletilen verinin çıkış noktasındaki aygıt "*maste*r", varış noktasındaki ise "*slave*" olarak adlandırılır [32]. "*Master"*, adından da anlaşılacağı gibi veri transferi yapmak isteyen taraftır. Veri yolu üzerinde birden fazla sayıda "*master*" özellikli aygıt bulunabileceğinden, bunların veri yolunu paylaşımlı olarak kullanmaları gerekmektedir. Bu paylaşımı düzenleyen *host* bridge'dir.

Şekil 4.1'de işlemcinin video kartına büyük miktarda veri ileteceğini düşünelim. İşlemci, öncelikle, veri yolunu kullanmak istediğini host bridge'ye bildirir. Host bridge, veri yolu o sırada kullanımda ise istek sahibini sıraya sokar. Yol üzerindeki transfer bittikten sonra işlemciye izin verir. Diğer bütün öncü istekli transferler bu şekilde olmaktadır. Bu örnekte öncü, işlemci, hedef ise video kartıdır. Bir sonraki adımda video kartı işlemciye veri aktarmak isterse bu sefer roller değişecek ve öncü, video kartı, slave ise işlemcinin kendisi olacaktır. Görüldüğü gibi veri aktarımı kontrolü doğrudan işlemci üzerinde olmamaktadır.



Şekil 4.1. PCI Veri Yolu İçeren Bir Bilgisayar Donanımı

Tablo 4.1'de veri yolu standartlarına ait veri transfer hızları verilmiştir. Verilen hızlar veri yolunun sürekli kullanımda olduğu düşünülerek hesaplanmıştır. Oysaki, veri yolu birçok donanımın zamanda paylaştığı bir veri iletim ortamı olduğundan, hiçbir aygıtın veri yolunun hakimiyetini sürekli elinde tutması mümkün değildir. Dolayısıyla, gerçek veri hızı Tablo 4.1' de verilen değerlerden daha az olacaktır. Yine de tek bir aygıtın veri yolunu sürekli kullandığı zaman dilimlerinde bu değerler anlık hız olarak değerlendirilebilirler. Veri iletim hızı, veri yolu üzerinde bulunan aygıtların sayısına ve bu aygıtların veri yolunu kullanma sıklığına göre değişecektir [33,34,35].

Tablo 4.1'de verilen 32 bit PCI standardında her saat periyodunda 32 bitlik veri iletimine izin vermektedir. Sürekli veri iletimi durumunda veri iletim hızı 32 bit x 33 MHz = 132 Mbyte/sn olmaktadır.

Tablo 4.1 Çeşitli Veri Yollarının Karşılaştırılması

Veri yolu Standardı	**En yüksek veri hızı**
PCI-X 64 bit / 133 MHz	1.06 Gbyte'/sn
Fiber Kanal	~1 Gbyte'/sn
PCI 64 bit / 66 MHz	528 Mbyte'/sn
AGP 2X	528 Mbyte'/sn
PCI 32 bit / 33 MHz	132 Mbyte's/sn
Wide Ultra2 SCSI	80 Mbyte's/sn
IEEE 1394 (FireWire)	400 Mbits/sn
USB	12 Mbits/sn

4.2. PCI Veri Yolu İşaretleri

Bu bölümde, PCI veri yolu kullanıcıları için gereken elektriksel işaretler konusunda bilgi verilmiştir. İşaretlerin temel işlevleri anlatılmıştır. İşaretlerin iletim yönü, PCI veri yolu üzerinde bulunabilecek herhangi bir aygıt için belirtilmiştir. PCI-SIG tarafından belirlenen PCI v3.0 esas alınmıştır.

Elektriksel işaretler şu şekilde sınıflandırılabilir :

- Saat ve başlatma işaretleri (Clock and Reset),
- Veri aktarımı kontrol işaretleri (Transaction Control Signals),
- Adres ve veri işaretleri (Address and Data Signals),
- Sıralama işaretleri (Arbitration Signals),
- Hata işaretleri (Error Signals),
- İsteğe bağlı işaretler (Optional Signals).

İşaretlerin yanında bulunan semboller ve anlamları:

- **#:** düşük seviyede aktif (active low) bir işaret olduğunu göstermektedir.
- ***I/O***: giriş ve çıkış olarak kullanılabilir.
- ***I***: sadece giriş.
- ***O:*** sadece çıkış

Saat ve başlatma işaretleri:

- ***CLK*** *(Saat işareti)*

PCI giriş saati. Bütün işaretler yükselen kenarla örneklenir. Frekansı, 33 MHz'dir. Gerçekte, 30 ns periyotlu bir işarettir (frekansı 33.33333 MHz). Saat işareti, 0'dan 33 MHz' e kadar herhangi bir aralıkta olabilir.

RST# (Başlatma işareti)

Sistem saatinden bağımsız ve asenkrondur. RST# aktif olduğu sürece, PCI aygıtının tüm giriş çıkışlarını yüksek empedans (tri-state) konumunda tutması istenir.

Veri aktarımı kontrol işaretleri:

- ***FRAME#*** **– I/O:** Herhangi bir veri aktarımının başlangıcını ve bitişini belirtir.
- ***IRDY#*** **– I/O (Initiator-Ready):** Öncü'nün veri transferine hazır olduğunu gösterir.
- ***DEVSEL#*** **– I/O (Device select):** PCI'ın dağıtık adres kod çözme yapısının bir parçasıdır. Her hedef her veri aktarımına ilişkin adres çözme işleminden kendisi sorumludur.Bir hedef aygıtı, kendi adresini gördüğünde, veri aktarımında bulunduğunu belirtmek üzere DEVSEL# işaretini aktif hale getirir.
- ***TRDY#*** **– I/O (Target-Ready):** Bir hedef aygıtı bu işareti aktif hale getirirmekle karşı taraftaki öncü'ye veri transferi için hazır olduğunu belirtir
- ***STOP#*** **– I/O:** Hedef aygıtı bu işareti aktif ederek veri aktarımını sonlandırmak ihtiyacında olduğunu belirtir. Birkaç değişik sekilde sonlandırma olabilir. Bunlar ileride incelenecektir.
- ***IDSEL*** **– I:** Veriyoluna bağlı herbir aygıt (agent) için ayrı bir IDSEL işareti mevcuttur. Aygıtlar, kendi adreslerini tanımazdan önce bu işaret ile seçilebilir.

Adres ve veri işaretleri:

- ***AD[31:0]*** **– I/O:** 32-bit address/data bus ."0" biti, LSB (Least Significant Bit) bitini göstermektedir.
- ***C/BE#[3:0]*** **– I/O:** 4-bit "command/byte enable" verisi. Adres fazı sırasında PCI veri transfer komutunu belirler. Data fazı sırasında "byte enable" olarak

kullanılır. Her bit, bir byte'a karşılık düşer. Örneğin, C/BE#[0], AD[7:0] aralığını temsil eder.

- ***PAR – I/O:*** Eşlik(parity) biti. Adres, data ve "command/byte enable" verilerinin iletimini doğrulamak için kullanılır. Bu üç işaretin YA DA (XOR) fonksiyonundan geçirilmesi ile elde edilir. Başka bir deyişle bu 37 bitlik dizinin tamamındaki "1" lerin sayısı çift olmalıdır.

Sıralama işaretleri:

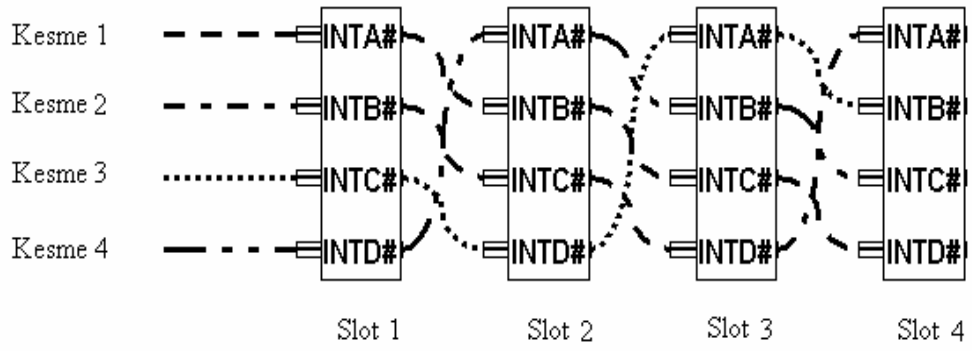
- ***REQ#* – O (Request-İstek):** Öncü tarafından veri yolunu elde etme isteğini gösterir. Her Öncü'nün kendi REQ# hattı bulunmaktadır.
- ***GNT#* – I (Grant-Onay):** Sistem tarafından Öncü'ye veri yolunun verildiği bilgisini taşır. Her Öncü'nün kendi GNT# hattı bulunmaktadır.

Hata işaretleri:

- ***PERR#* – I/O:** Parite hatasının oluştuğunu bildirir.
- ***SERR#* – I/O:** Ciddi bir sistem hatasının oluştuğunu bildirir. Örneğin, adres parite hatası. Bazı sistemlerde bilgisayarın tekrar başlamasına neden olabilir.

İsteğe bağlı işaretler:

- ***INTA#, INTB#, INTC#, INTD# :*** PCI kesme (interrupt) işaretleri Genellikle yaygın tasarım geleneği olarak INTA# kullanılır. Şekil 4.2'de "barber-pole" adı verilen interrupt bağlantı düzeni görülmektedir. Bu teknikle, yaygın olarak INTA# kullanılmasına rağmen, aygıtlar arasında çakışma olmamaktadır [32,33].

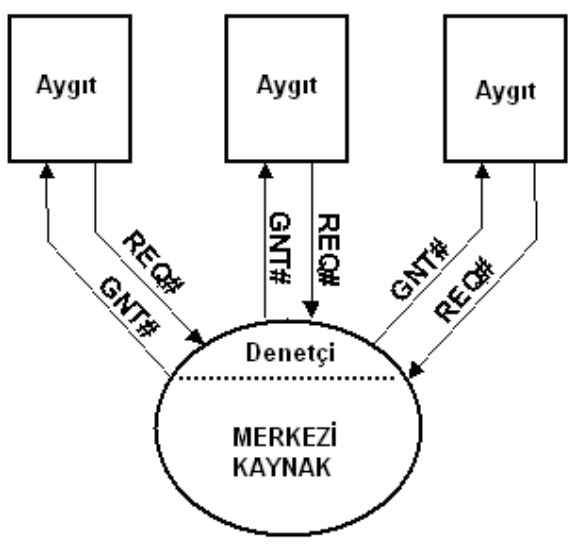


Şekil 4.2. Barber-Pole İnterrupt Bağlantı Düzeni

- ***LOCK# :***Sadece bridge'ler tarafından birinden diğerine özel bilgi aktarımı için kullanılır.
- ***SBO#, SDONE :*** PCI v2.2 – ayırılmış kullanılmayan işaretler.
- **CLKRUN#, PME# :** Güç yönetimi için ayrılmış işaretler
- ***M66EN :*** Veri yolu 66MHz hızında çalışabilecek bir ana kartta, PCI sistem saatinin 33MHz ile 66 MHz arasında seçilmesi için kullanılır. 66 MHz'i desteklemeyen donanım kartları, 66 MHz çalışabilen bir PCI slot'a takıldığında M66EN işaretini toprak potansiyeline çekerek PCI sistem saatinin 33MHz koşmasına neden olurlar. 66 MHz' i desteklemeyen bir kart bile olsa PCI veri yolu 33 MHz hızında çalışır.
- ***JTAG :*** IEEE 1149.1 boundary-scan için kullanılan işaretler.

4.3. PCI Veri Yolu Paylaşımı

PCI veri yolu üzerinde bulunan aygıtların veri yolunu sırayla kullanmaları, paylaşmaları, genellikle *host bridge'* de bulunan sıralayıcı tarafından düzenlenir. Bu sıralayıcı, tıpkı bir trafik polisinin kavşaktan geçen araçları kontrol ettiği gibi, PCI veri yolunu kullanacak olan aygıtı belirler. Şekil 4.3'de PCI veri yolunda bulunan aygıtlar ve sistem *host bridge'* i temsil eden merkezi kaynak (*central source)* görülmektedir [33]. Öncü'ler veri yolu erişimi isteklerini REQ# işaretini aktif ederek belirtirler. *Host bridge*, veri yolunu hangi Öncü'ye verecekse ona ait GNT# işaretini aktif yapar. Sistem *host bridge* tarafından belirli bir anda sadece bir Öncü'ye ait GNT# işareti aktif edilir. GNT# işareti, herhangi bir anda herhangi bir Öncü'ye verilebilir. Bir diğer Öncü o sırada veri yolunu kullansa bile bu değişmez. Bu sayede sıralamalar arasında geçebilecek boş süreler azalır.



Şekil 4.3. PCI Veri Yolu Paylaşımı

Veri yolu erişim hakkı yeni bir aygıta verilmişse ve o sırada veri yolu başka bir aygıt tarafından kullanılıyorsa, veri yolu atıl duruma geçer geçmez ilk erişim hakkı bu yeni aygıtındır. PCI v2.1, *host bridge'* de bulunan veri yolu paylaştırıcısı çalışma prensibi hakkında tavsiyede bulunmamaktadır. Bu iş için çeşitli algoritmalar kullanılmaktadır. Önemli olan, hiçbir aygıtta iletilememiş veri bulunmayacak şekilde çalışma yapılabilmesidir. Hiçbir aygıt, veri yolu erişimi istemiyorsa ve veri yolu atıl durumdaysa GNT# işareti herhangi bir aygıt üzerinde park etmiş olarak kalabilir. Bu durumda aygıt AD veri yolunu, C/BE veri yolunu ve PAR işaretini elektriksel olarak uygun seviyelerde tutmak zorundadır. Aynı zamanda FRAME, IRDY, TRDY gibi işaretleri de pasif konumlarında tutmak zorundadır .

4.4. PCI Adres Bölgesi

PCI'da hedef aygıtı olarak çalışacak donanımlar, üç farklı türden adres bölgesi bulundurabilirler. Bunlar;

Düzenleme bölgesi; aygıt hakkında temel bilgilerin bulunduğu bölgedir. İşletim sistemi veya *host bridge* tarafından bu bölgeye erişilerek aygıt için programlama yapılır. Bu bölgede aygıtın üretici firmasını ve sınıfını belirten kodlar bulunur. Aynı zamanda tak çalıştır (plug-and-play) özelliği sağlar. İçerdiği "*taban adres yazmaç*" ları sayesinde, PCI aygıtının hafıza veya I/O bölgesine dinamik olarak yerleşmesi sağlanır. Toplam 256 byte'lık bir adres bölgesidir (bkz Tablo 4.2). İlk 64 byte (00-3C h) PCI v2.1

tarafından belirlenmiş klasik konfigürasyon bölgesidir. Geriye kalan kısım kullanıcıya bırakılmıştır. Bu alan, aygıta erişen yazılım tarafından genel amaçlı kullanılabilir.

Giriş/Çıkış (I/O) bölgesi; bilgisayar çevre birimleri ile kullanılır. PCI v2.1 standardına göre, her aygıt için, 4 byte' tan 2 G byte' a kadar I/O bölgesi tanımlanabilir.

Hafıza bölgesi; diğer kalan bütün erişimler için ayrılmış bölgedir. Her PCI aygıtında 16 byte' tan 2 G byte'a kadar bellek bölgesi tanımlanabilir.

Tablo 4.2 PCI Konfigürasyon Bölgesi Haritası

	31 16		15 0	
00h	Cihaz Kimliği		Üretici Kimliği	
04h	Durum		Komut	
08h	Sınıf Kodu			Sürüm No
0Ch	BIST	Başlık Tipi	Gecikme-Zaml.	Cache Line boyutu
10h	Taban Adres Yazmaç 1			
14h	Taban Adres Yazmaç 2			
18h	Taban Adres Yazmaç 3			
1Ch	Taban Adres Yazmaç 4			
20h	Taban Adres Yazmaç 5			
24h	Taban Adres Yazmaç 6			
28h	CardBus CIS (Kart Bilgi yapısı) İşaretçi			
2Ch	Sistem Kimliği		Alt Sistem Üretici Kimliği	
30h	ROM Genişleme Taban adresi			
34h	Ayrılmış			Liste Başı İşaretlyi.
38h	Ayrılmış			
3Ch	Max_Lat	Min_Gnt	Kesme Ucu	Kesme Hattı

Tablo 4.2'de görülen Vendor ID, PCI-SIG tarafından PCI aygıtı üreten üretici firmalara verilen tekil bir sayıdır. Yazılım ya da işletim sistemi, PCI veri yolunda hangi firmalara ait aygıtlar bulunduğunu, aygıtların Vendor ID'lerini okuyarak tespit eder. Device ID ve Revision ID, üretici firma tarafından belirlenir. System ID ve Subsystem Vendor ID, PCI arayüzü işlemlerini kotaran tümdevrenin (aygıtın) değil, o PCI slot'ta bulunan karta

ait sayılardır ve PCI-SIG tarfından belirlenirler. *Header Type*, konfigürasyon adres bölgesinin tipini belirler. Standart *header* ya da PCI to PCI *bridge header* gibi farklı tipleri ifade etmek için kullanılır. *Class Code*, PCI veri yolunda bulunan donanımın ne tür bir cihaz (işlemci, grafik kartı v.b) olduğunu belirler. *Command register*, PCI aygıtının işletim sisteminden gelen komutların hangilerine, ne şekilde cevap vereceğini belirler. Örneğin, hafıza bölgesi erişimine cevap verip vermeyeceği bu register yardımıyla kontrol edilebilir. Erişim izni verilebilir ya da kaldırılabilir.

Status register, PCI aygıtının işletim sistemine veya yazılıma özel amaçlı durum bilgileri aktarmak için kullanılabilir. Örneğin, PCI aygıtında adres kod çözme işleminin (DEVSEL timing) hızı hakkında işletim sistemine bilgi verilebilir. *Geçikme Zamanlayıcı (Latency Timer)*, Max_Gnt ve Max_Lat Öncü veri transferi özellikleri için kullanılır. Max_Gnt, Öncü'nün veri yolunu ne kadar süreyle kullanacağını belirtir. Max_Lat, Öncü'nün veri yolu kullanımı için iki kullanım arasında ne kadar bekleyebileceğini belirtir. Gecikme Zamanlayıcı, veri transferi sırasında, öncü, veri yolu kullanma hakkını kaybederse (GNT# pasif olursa) ne kadarlık verinin bozulabileceğini saat kenarı sayısı olarak verir.

Taban Adres Yazmaç' ları, 16 byte'tan 2G byte'a kadar adresleme bölgesinin ilk adreslerini gösterirler. PCI aygıtı bu register'ların hepsini veya birkaçını kullanabilir. *Host bridge*, bu register'a PCI aygıtına atadığı adres bölgesinin başlangıç adresini yazar. Bundan sonra, PCI aygıtı, veri yolunda, bu adresi gördüğünde kendisine erişilmek istendiğini farkeder. BIST, *Built In Self Test* register'ıdır. *Self test* özelliği olan PCI aygıtlarında, testi başlatmak ve sonucunu izlemek için kullanılır [33,34].

4.5. PCI Veri Yolu Komutları

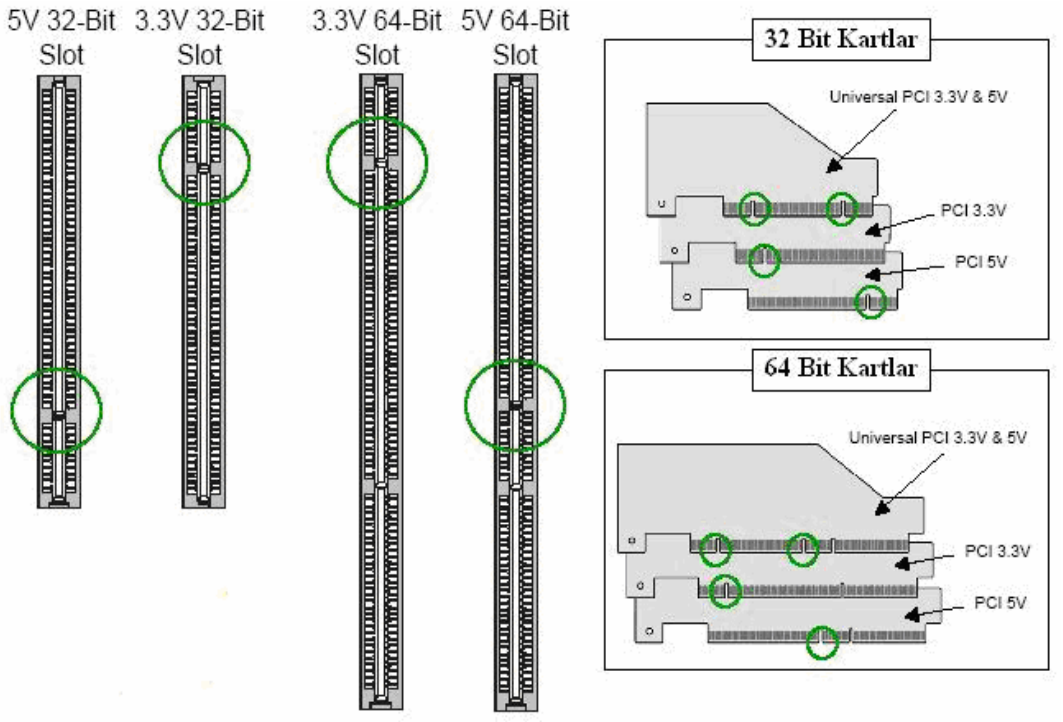
PCI veri yolunda, veri transferi yapılırken, konfigürasyon, Giriş/Çıkış (I/O), hafıza, özel amaçlı gibi çeşitli türde komutlar kullanılmaktadır. Veri transferi yapacak Öncü tarafından FRAME işaretinin aktif edilmesiyle, 4 bitlik C/BE işareti ile 16 değişik türden komut belirlenebilir. Bu komutlar tablo 4.3'de gösterilmiştir. Bir PCI veri yolu çevriminin ilk saatinde, bir hafıza veya Giriş/Çıkış cihazının adresi, Adres/Veri uçlarında görülür. Bununla birlikte C/BE uçlarında, PCI üzerinde yapılan işlem belirtilir.

Tablo 4.3 PCI Veri Yolu Komutları

C/BE	**Komutlar**
0000	Kesme İsteği
0001	Özel Çevrim
0010	I/O (Giriş/Çıkış) Okuma
0011	I/O (Giriş/Çıkış) Yazma
0100	Ayrılmış
0101	Ayrılmış
0110	Hafıza Okuma
0111	Hafıza Yazma
1000	Ayrılmış
1001	Ayrılmış
1010	Konfigürasyon Okuma
1011	Konfigürasyon Yazma
1100	Ardışıl Hafıza Okuma
1101	Çift Adresleme Çevrimi
1110	Satır Hafıza Erişimi
1111	Geçersiz kılma ve Hafıza Yazma

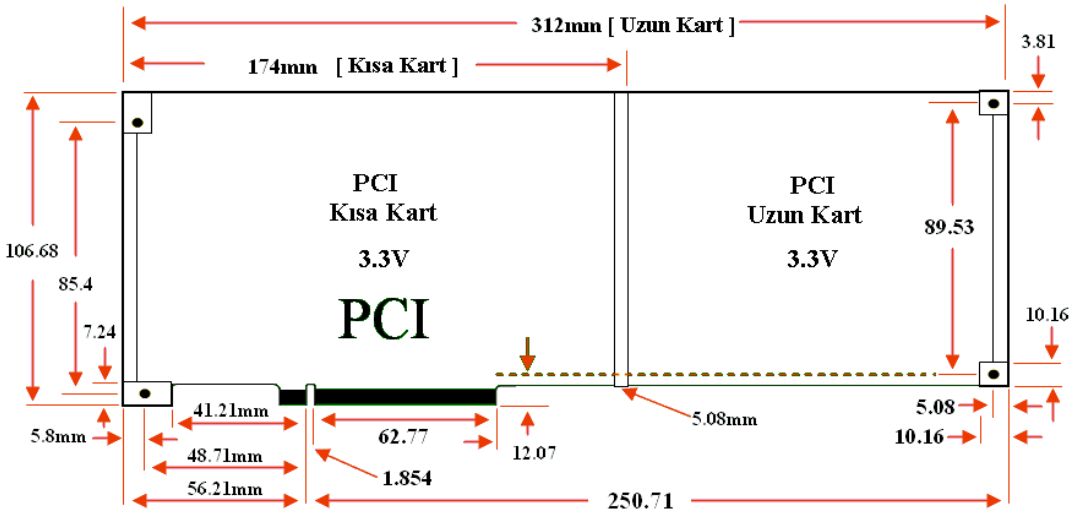
4.6. PCI Veri Yolu Mekanik Özellikleri

PCI veri yolunu destekleyen sistemlerde PCI kartlarının takılabileceği yuva (connector) yapıları Şekil 4.4'de verilmiştir. Görüldüğü gibi, yuvalarda, gerilim seviyeleri yönünden uygun olmayan kartların, sisteme takılmalarını önleyici anahtar elemanları mevcuttur. 5 Volt ve 3.3 Volt'u beraber destekleyen evrensel kartlar her iki bağlantı yuvasına uyacak şekilde çift anahtarlı yapılırlar.

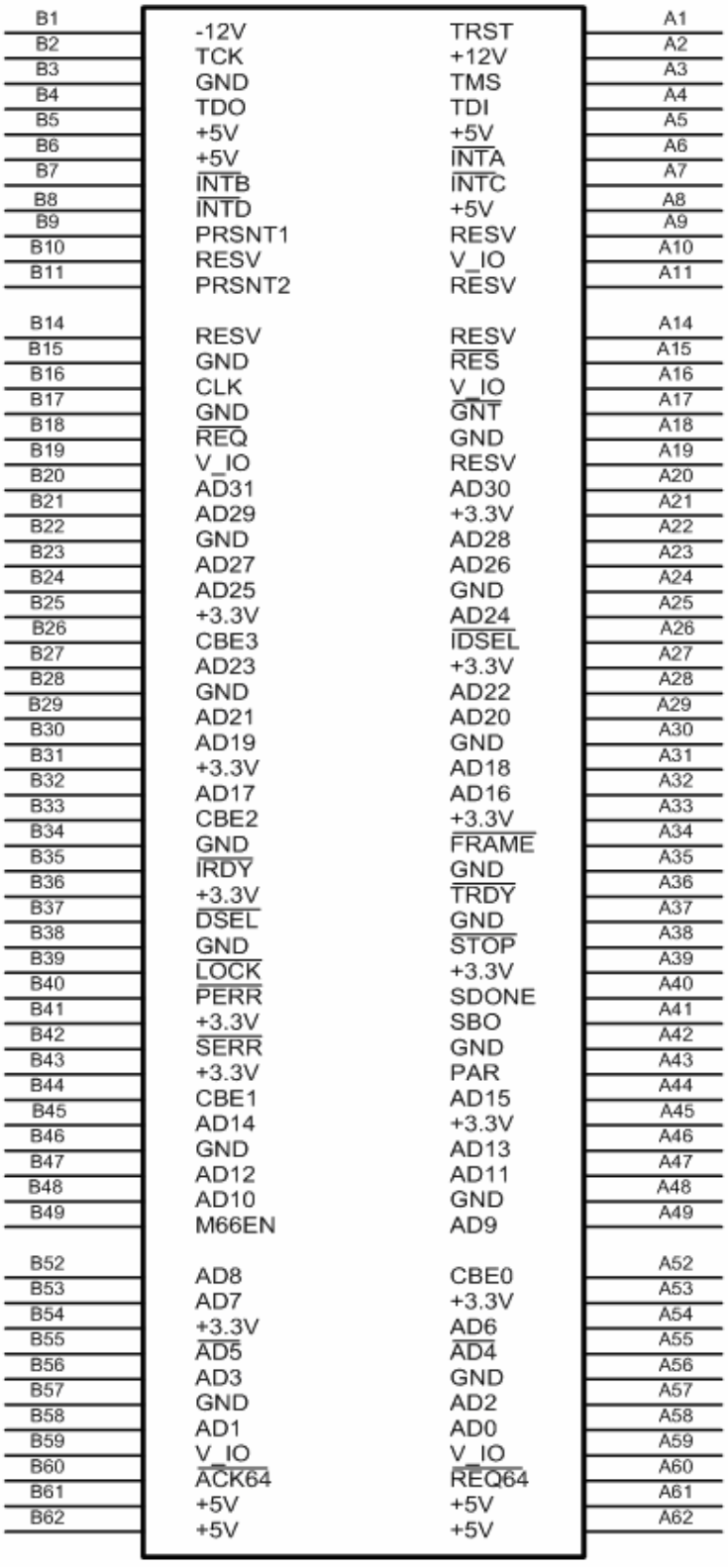


Şekil 4.4. PCI Veri Yolu Bağlantı Yuvası Yapıları

32 bit PCI yuvalarında, elektriksel işaretleri iletmek için 58 adet iletim yolu bulunmaktadır. Çift taraflı olarak dizilmiş bu elemanlar toplam 2 x 58 = 116 adet elektriksel işaret için bağlantı sağlamaktadır. Şekil 4.5'de PCI kartı fiziksel boyutları Şekil 4.6'da bağlantıların isimleri ve yuvadaki fiziksel konumları görülmektedir.



Şekil 4.5. PCI Kartı Fiziksel Boyutları



Şekil 4.6. Evrensel (Universal) PCI Kartı Yuvası Bağlantı Tanımları

5. BÖLÜM

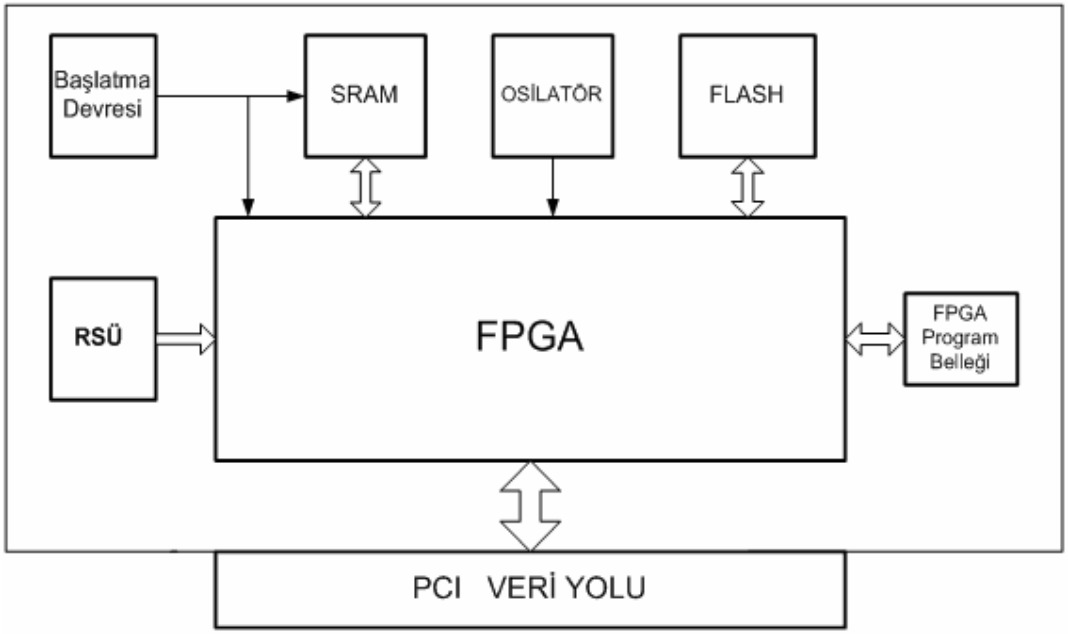
RSÜ'LER İÇİN PCI ARAYÜZLÜ FPGA TABANLI GERÇEK ZAMANLI İSTATİSTİKSEL TEST VE VERİ TOPLAMA DONANIMI

5.1. Tasarlanan Donanım

Tekrar düzenlenebilir işlem kartı tasarımı yapılırken, bilgisayarlarda ve iş istasyonlarında genel performans iyileştirmesi sağlaması ve uygulama çeşitliliği sunması istenir. Bu amaçla, tekrar düzenlenebilir sistemlerin temel yapı taşı olan SRAM (uçucu bellek elemanı) tabanlı bir FPGA elemanı kullanılmıştır. Kart ile çeşitli uygulamaların gerçeklenebileceği düşünülerek büyük kapasiteli SRAM elemanları seçilmiştir. Tekrar programlama özelliği için, FPGA'e ait programın saklanabileceği kalıcı bellek elemanları kullanılmıştır. Bilgisayar arayüzü olarak, yüksek hızlı (132 Mbyte/s) veri transferine olanak veren endüstri standardı olmuş PCI veri yolu tercih edilmiştir. Ayrıca temel yapı elemanı olarak seçilen FPGA, 200000 kapılıktır ve 200 MHz' e kadar saat hızlarında çalışabilecek güçtedir. Bu özellikler sayesinde, yüksek performans elde edilmiştir.

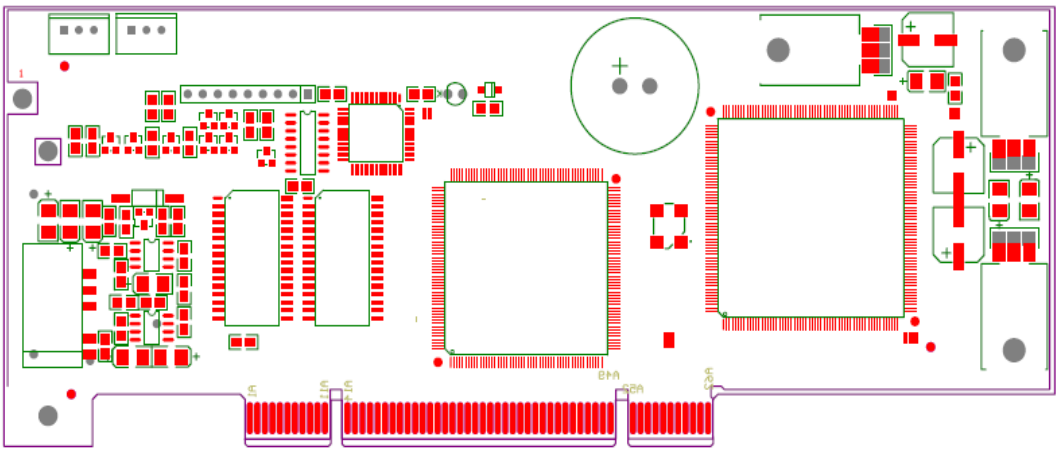
Bununla beraber farklı uygulamalar için kullanılabilecek ilaveten 1 adet Virtex-E serisi FPGA ve ona ait farklı bir program belleği de donanım üzerine konulmuştur. Bu FPGA üzerinde ihtiyaç duyulan farklı bir fonksiyonu gerçekleştirmek mümkündür. Aynı zamanda diğer FPGA ile de haberleşebilecek şekilde tasarlanmıştır.

Tasarlanan ve gerçeklenen tekrar düzenlenebilir işlem kartının genel mimarisi Şekil 5.1'de görülmektedir. Kartın, bilgisayar ile arayüzü, 32 bit, 33 MHz, PCI v3.0 veri yolu üzerinden gerçeklenmiştir. PCI veri yolu ile arayüz devresi FPGA içinde gerçeklenmiştir. Ayrıca aynı FPGA içine, FIPS 140-2 test suiti, test sonucunu gösteren LED kontrol devreleri harici RSÜ okuma girişleri ve Von Neumann doğrultucusu da gömülmüştür. Von Neumann doğrultucusunun aktif edilmesi ve yasaklanması içinde aynı FPGA den bir giriş ucu kullanıcıya çıkartılmıştır.



Şekil 5.1. Donanımın Genel Mimarisi

Kart üzerinde istenilen sayısal tasarımın koşacağı ortam FPGA'lerdir. FPGA'lere ait programların saklanabileceği iki adet kalıcı bellek elemanları FPGA'lerin programlanma ve yeniden başlatılma işlerini idare etmektedir. Uygulama ihtiyaçları göz önünde bulundurularak, 2 adet 8 bit x 512 K büyüklüklerinde uçucu bellek (SRAM) elemanları kullanılmıştır. PCI veri yolundan gelen sistem saatine ilave olarak, istenilen frekansta osilatör takılması için donanım üzerinde yer bırakılmıştır. Tasarımda besleme gerilimi gözleyicisi ve başlatma devresi elemanı kullanılmıştır. Bu elemanlar her besleme verildiğinde ve besleme hatalarında FPGA' yi yeniden koşullandırır. Kullanılan FPGA 3.3 Volt besleme gerilimi ile çalışabilmektedir. Giriş çıkış kapılarının elektriksel özellikleri PCI veri yoluna uygundur. Günümüzde yaygın olarak kullanılan bilgisayarların hepsinde PCI veri yollarında 3.3 Volt gerilim kaynağı bulunmadığından kart üzerinde bu iş için 5 Volt'dan 3.3 Volt gerilim elde edebilen gerilim dönüştürücüsü kullanılmıştır. RSÜ devresi ise ayrı bir blok olarak kart üzerine yerleştirilmiş ve dış etkilerden korumak için ekranlanmıştır. Tasarıma ait baskılı devre üst görünüşü Şekil 5.2'de verilmiştir.



Şekil 5.2. FPGA Tabanlı PCI Arayüzlü Donanım

5.1.1. Kullanılan Devre Elemanları ve Özellikleri

Kulanılan Devre Elemanları ve özellikleri şöyledir:

- 1 adet Xilinx XC2S200-5PQ208 Spartan II 2.5 V FPGA
- 1 adet Xilinx XCV400E-6PQ240C Virtex-E 1.8 V FPGA
- 1 adet XC18V04VQFP44C 4 Mbit Spartan II FPGA program belleği
- 1 adet XC18V04VQFP44C 4 Mbit Virtex-E FPGA program belleği
- 1 adet Am29LV040B 512K x 8 bit kalıcı bellek (flash), erişim süresi 70 ns
- 2 adet K6T4008V1C-VB70 512 K x 8bit uçucu bellek, erişim süresi 70 ns
- 1 adet MAX793TCSE başlatma ve gerilim gözleyici devresi

Xilinx XC2S200-5PQ208 FPGA tümdevresi;

0.18 mikrometre teknolojisi ürünüdür. 3.3 V (± %5 tolerans) besleme gerilimi ile çalışmaktadır. Giriş port'ları 5 V TTL işaretlere uyumludur. Giriş-çıkış port'ları 3.3 V LVTTL standardındadır. 5 ve 3.3 V PCI giriş/çıkış işaretlerine uyumludur. 8 adet birbirinden bağımsız saat dağıtım ağı içermektedir. Dahili üç durumlu (tri-state) işaretleşmeye izin vermektedir. Giriş-çıkış portlarında 3.3 V gerilimi yönünde programlanabilir sınırlama diyodu vardır.

Tablo 5.1 Xilinx XC200 FPGA Tümdevresi Özellikleri

Yonga	Lojik Hücre	Kapı Sayısı	CLB Matris	Toplam CLB	Program Uzunluğu (bit)
XCS200	5292	200000	28x42	1176	1335840

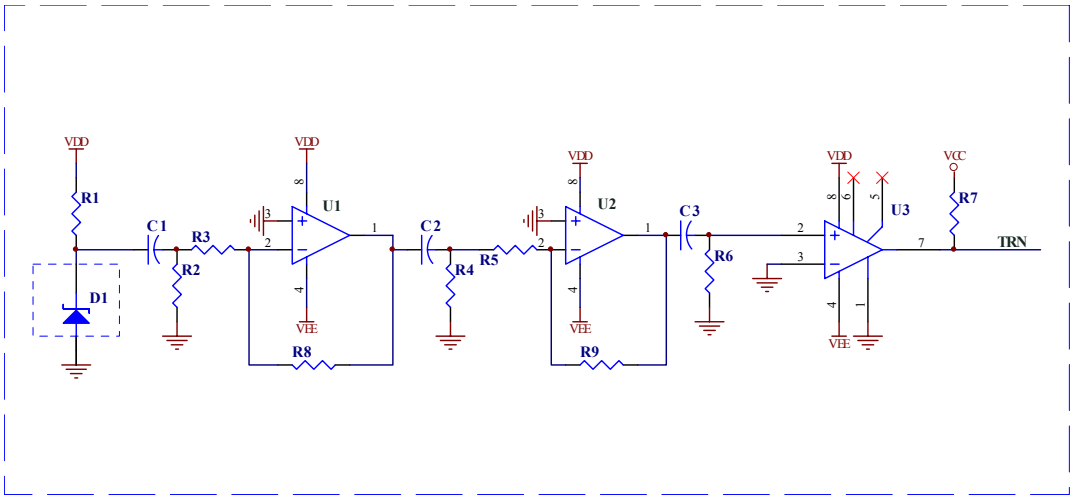
Bilgisayara ve dolayısıyla PCI karta, besleme gerilimi ilk verildiğinde, gerilim izleyici devresi, besleme gerilimi kararlı bir seviyeye oturduktan 200 ms sonra başlatma işareti gönderir. Bu işaretle birlikte FPGA flash 1 belleğindeki programı otomatik olarak çeker. FPGA'in program belleği olarak 4 Mbit lik seri bellek kullanılmıştır. FPGA, bu program ile programlanıp normal çalışmaya başlar. PC bu ilk konfigürasyonla açılır. Bu aşamadan sonra, kart üzerinde çalıştırılmak istenen herhangi bir donanım uygulaması PCI veri yolu üzerinden, flash 2 belleğine herhangi bir F(x) fonksiyonunu gerçekleyen FPGA programını yükleyebilir.

PCI veri yoluna erişim dahil olmak üzere yapılacak tüm sayısal devreler aynı FPGA yapısı içinde tümleştirilmiştir. Uygulamaların çalışma hızı, kullanılan FPGA' in teknolojisine ve gerçeklenebilecek en yüksek frekanslı saat frekansına bağlı olarak değişecektir.

5.2. Tasarlanan GRSÜ

Zener gürültüsünü temel alan RSÜ tasarımı Şekil 5.3'de gösterilmiştir. Çığ gürültüsü gerilimi kuvvetlendirme yöntemi, yarı iletken elemanlar kullanarak rasgele sayı üretmek için ideal bir yöntemdir. Çığ gürültüsü, zener veya PN jonksiyonunda çığ olayından kaynaklanan bel verme esnasında üretilmektedir. Fakirleşmiş bölgedeki delik ve elektronlar kafesle çarpışarak delik-elektron çifti oluşturabilmek için yeterli enerji miktarını elde etmelidir. Zener diyot kullanılarak elde edilen bu gürültü de beyaz spektruma sahiptir.

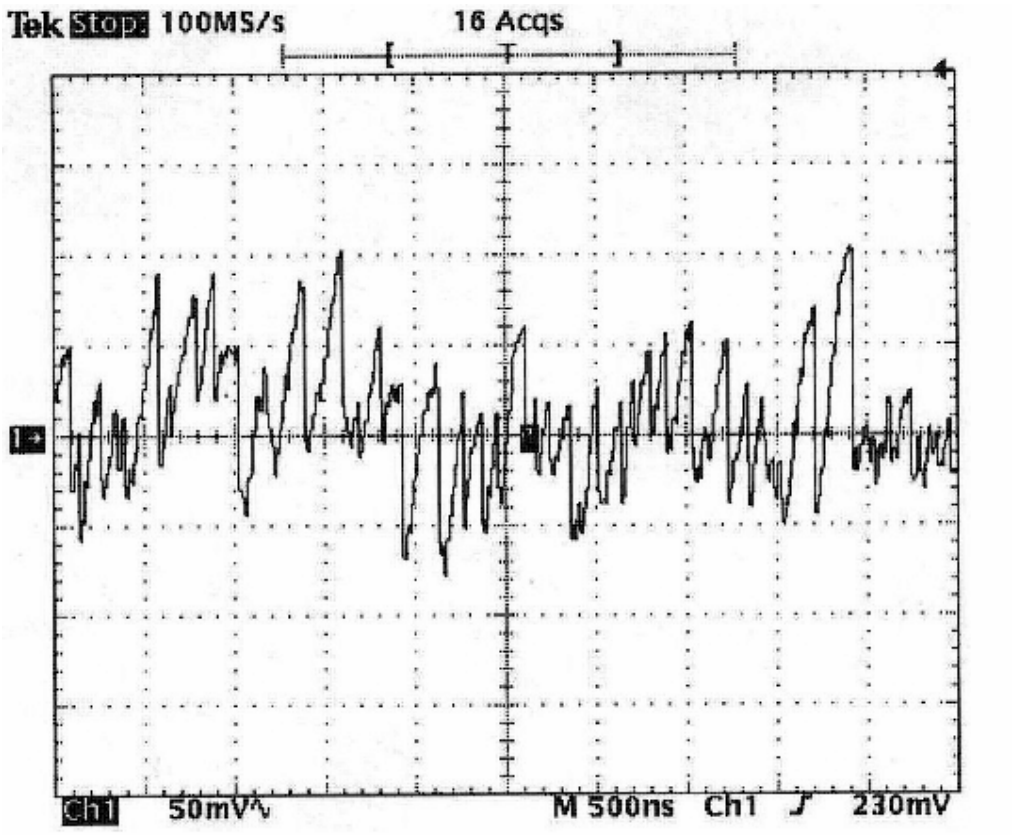
Uygulamada bu gürültü yapısı temel alınmış, gürültü işaretleri üzerinde işlemler yapılırken, gürültünün karakteristiğini bozmamaya özen gösterilmiştir.



Şekil 5.3. Zener Gürültüsünü Temel Alan RSÜ Devresi

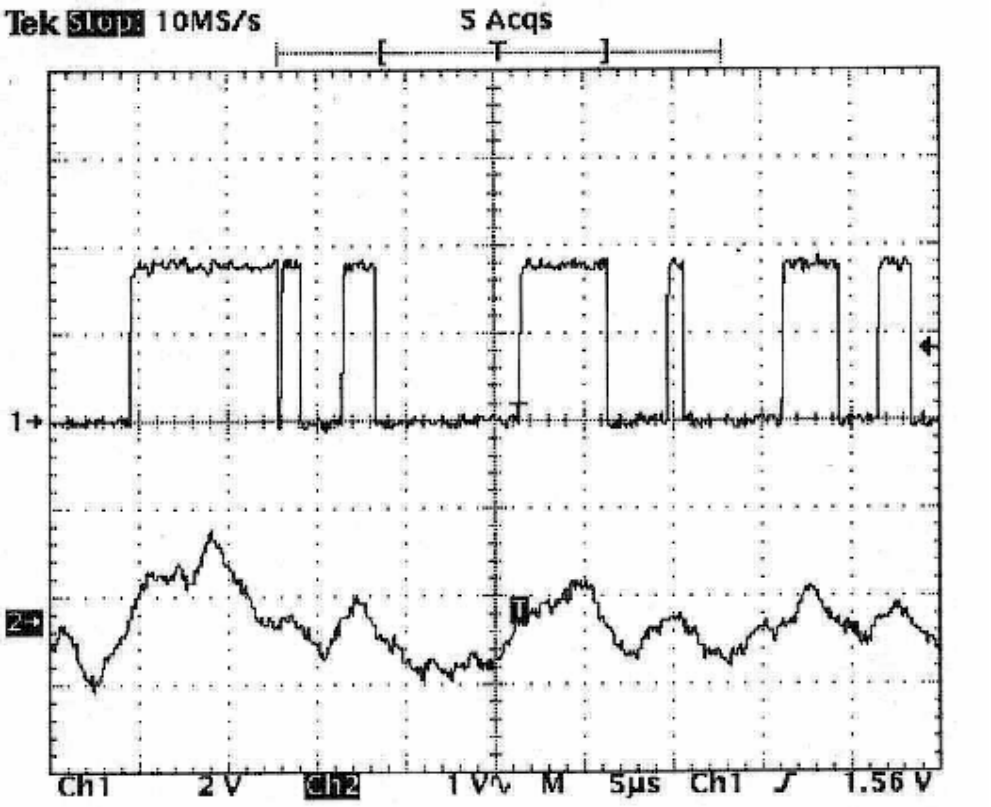
Tasarımda Micronetics firması tarafından üretilen bir zener diyot kullanılmıştır. Tüm diyotlar, simetrik beyaz Gauss gürültüsü özelliğine sahip olması açısından Micronetics tarafından test edilmiştir, bu bakımdan rasgele sayılar üretmek için ideal bir başlangıç noktasıdır. Tasarlanan RSÜ devresinde asıl amaç gürültü üretmek olduğu için özellikle ters zener geriliminin (I_z) 100 µA'in altında tutulması avantajlı olacaktır. Düşük akım bölgelerinde gürültü yoğunluğu, zener diyota bağlı olarak daha yüksektir. Micronetics, mümkün olduğu kadar yüksek gürültü gerilimi üretilmesi için ters zener geriliminin (I_z) 40 µA'de tutulması gerektiğini tavsiye etmektedir. Tasarlanan devredeki R_1 direnci, ters kutuplanmış D_1 zener diyotu üzerinden yaklaşık olarak 30-60 µA akım akmasını sağlayacak şekilde seçilmelidir.

Kullanılan zener diyot 10 Hz – 500 KHz frekans bölgesinde çalışmaktadır. Devrede iki adet U_1 ve U_2 işlemsel kuvvetlendiricisi kaskad bağlanmıştır. Tek bir işlemsel kuvvetlendiricinin kazancı 10 iken kaskad yapı ile zener diyot üzerindeki gürültü gerilimi 100 katına kuvvetlendirilmiştir. Kuvvetlendirici katı 10 Hz – 200 KHz frekans bölgesindeki işaretleri geçirmektedir, bu kattan alınan çıkış işareti, referans gerilim olarak GND'nin kullanıldığı U_3 gerilim karşılaştırıcıya verilir. GND'den büyük işaret seviyeleri lojik 1, küçük işaret seviyeleri ise lojik 0 olarak değerlendirilir. Şekil 5.4'de I_z=45 µA için U_2 işlemsel kuvvetlendirici girişinde osiloskopla ölçülen gürültü gerilimi gösterilmiştir, Şekil 5.5'de ise yine osiloskopla ölçülmüş olan kuvvetlendirilerek U_3 gerilim karşılaştırıcı girişine verilen ve U_3 çıkışından elde edilen rasgele işaretler görülmektedir.



Şekil 5.4. Zener Diyot Tarafından Üretilen Gürültü Gerilimi

Artık gürültü işareti, rasgele ikili sayılar halinde işlenmeye hazırdır. Gerilim karşılaştırıcıdan alınan bu çıkış FPGA'e verilerek 128 KHz'de örneklenir. Daha önce sayısal son işleme bölümünde tanımlanan Von Neumann doğrultucusu, tasarımda kullanılan FPGA içersinde gerçeklenmiştir. Fakat bu durum işaretin rasgeleliğini artırmakla beraber frekansını yaklaşık 32 KHz'e düşürecektir. Bu devrenin çalışmasına, harici bir port vasıtası ile izin verilir veya yasaklanır. Von Neumann doğrultucusunun rasgeleliği artırma etkisini önlemek için RSÜ' lere uygulanan FIPS 140-2 testlerinden önce, bu devre yasaklanmış olup testler bu şekilde yapılmıştır. Bu durumda RSÜ frekansı örnekleme frekansı ile aynı olacaktır. İsteğe bağlı olarak kullanıcı RSÜ tasarımından emin olması durumunda, üretim hızının yavaşlamasını da göz önüne alarak Von Neumann devresinin çalışmasına izin verebilir.



Şekil 5.5. Gerilim Karşılaştırıcı Girişine Verilen (Kanal 2) ve Çıkışından Elde Edilen (Kanal 1) Rasgele İşaretler

Uzun ve sürekli bit dizilerinin toplanarak, ölçülen çıkışların rasgelelik test programlarına uygulanması için genellikle bilgisayar bazlı veri elde etme sistemleri kullanılır.

5.3. Çevrimiçi Testler ve Veri Toplama

Önerilen sistemde yeni RSÜ tasarımlarının uzun prototipleme sürelerinden kaçınmak için, gerçek-zamanlı istatistiksel testler ve veri toplama işlemleri PCI arayüzü ile gerçeklenmiştir.

Bu modda, lojik "0"'dan ve lojik "1" 'den oluşan test altındaki yeni RSÜ'nün çıkış sinyali FPGA tarafından 100 Mbps' e kadar örneklendirilebilir. Bu örnekleme oranı, önerilen RSÜ tasarımlarının maksimum 40 Mbps'a çıkmayı başardığı [1,36] ve ümit verici olan sürekli-zaman kaotik osilatör tabanlı RSÜ'ler ile 100 Mbps' ler düzeyine çıkılabileceği [37] düşünülürse günümüzde yeterlidir.

Rasgele sayı üretim modunun tersine, Von Neumann'ın de-skewing tekniği, üretilen bit dizisinin asıl karakteristiğini belirlemek istendiğinden, FPGA içindeki bu modülün çalışması yasaklanmıştır. Eğer gerekirse, Von Neumann de-skewing tekniğinin çalışmasına yazılımla izin verilebilir.

Çevrimiçi test modunda ise, kart üzerindeki RSÜ çıkışından değil, dışarıdan bağlanan yeni RSÜ'nün çıkışından ardışık 20000 bit toplanır ve FPGA tarafından gerçeklenen 5 temel istatistiksel teste tabi tutulur. Testler bittiğinde, üretilmiş tüm bit dizileri PCI arayüzü üzerinden PC' nin RAM'ine yüklenir ve gerçek zamanlı 5 temel teste ait sonuçlar, donanım üzerindeki LED'lerden gözlenebilir. Bu, tasarımcıya parametre değişikliklerinin istatistiksel testlerin sonuçları üzerindeki etkilerini gerçek zamanda görme imkanı tanır. Sonuç olarak bu özellik, yeni prototip geliştirme zamanının kısalmasını sağlayacaktır. Bu tasarımda, FPGA tabanlı donanımın en yüksek veri depolama hızı 100 Mbps iken donanımsal olarak testlerin gerçeklenme hızı 31.5 Mbps olmaktadır.

5.4. Yazılımsal Olarak Uygulanan İstatiksel Testler

Donanımsal olarak hayata geçirilmiş GRSÜ ya da yazılımda test altındaki yeni RSÜ tasarımı tarafından üretilen gelişigüzel uzunluktaki ikili dizilerin rasgeleliğini test etmek için NIST test program demetindeki istatistiksel test paketi kullanılmıştır. Bu program demeti 16 testten oluşur ve bu testler bir dizi içinde var olabilen rasgele olmama durumunun değişik tiplerinin türleri üzerine odaklanır [21].

Bazı testler, çeşitli alt test gruplarına ayrıştırılabilirler. NIST test dizisinin odak noktası, kriptografik amaçlar için rasgeleliğin gerekli olduğu bu uygulamalar üzerinedir. Mevcut durumdaki 16 test Bölüm 3.5' te anlatılmıştır. Bu testlerin uygulama sırası, test dizisinde gelişigüzel seçilmektedir. Buna rağmen, bir dizideki dağılımın en belirgin rasgele ve düzgün olmama karakteristiğini belirlediği için frekans testinin ilk sırada koşturulması tavsiye edilmektedir. Eğer bu test başarısız olursa, diğer testlerinde başarısız olma ihtimali oldukça yüksektir. Yukarda bahsedilen testlerden en fazla zaman alan istatistiksel test lineer karmaşıklık testidir.

Parametreleri çağırmak yerine, bazı girdiler, ANSI C de geliştirilen test kodundaki bazı global değerler olarak seçilir. Test dizisinde kullanılan bir grup referans dağılım testleri standart normal ve chi-kare (χ^2) dağılımlarıdır. Eğer test edilen dizi rasgele değilse,

hesaplanan test istatistiği referans dağılımın uç bölgelerine düşecektir. Standart normal dağılım (çan eğrisi), rasgelelik kabulünün altında, beklenen istatistiksel değerle elde edilen RSÜ' nün değerinin karşılaştırılmasında kullanılır. Standart normal dağılımdaki test istatistiği $z = (x-\mu)/\sigma$ şeklindedir. Burada x örnek test istatistiği değeridir, μ ve σ^2 beklenen değer ve test istatistiğinin değişintisidir. χ^2 dağılımı (sola kayık eğri), örneğin gözlemlenmiş frekansların hipotez olarak kabul edilmiş dağılımın beklenen frekanslarıyla uygunluğunu karşılaştırmak için kullanılır. Test istatistikleri $(\chi^2)=\sum((o_i-e_i)^2/e_i$ biçimindedir ve bu eşitlikte o_i ve e_i gözlemlenmiş ve beklenen olma sıklığının ölçütüdür. Bu test suitindeki testlerin çoğu için, dizi uzunluğu değeri n'nin $10^{3'}$ ten 10^7'ye kadar geniş değerler aldığı varsayılmıştır. Böyle geniş bir örnek hacmi için, asimptotik olarak referans dağılımlar türetilmiş ve bu dağılımlar testleri gerçekleştirmek amacı ile kullanılmıştır. Testlerin çoğu n'nin daha küçük değerleri için uygulanabilir olmasına rağmen, n için daha küçük değerler kullanıldığında, referans dağılımı uygunsuz olabilmekte ve genellikle hesaplanması daha zor olan gerçek dağılımla yer değişitirmek zorunda kalınabilmektedir. Bu tez çalışmasında rasgele sayı üretme, çecrimiçi test ve veri toplama modlarında, testlerin herbiri 10^6 düzeyinde bit içeren diziler üzerinde uygulanmıştır.

Rasgele Sayı Üretim Modunda üretilen rasgele sayılardan donanımsal olarak gerçeklenmiş testleri geçip "Aday Rasgele Sayı Havuzu" nda depo edilenler yazılımsal olarak NIST'in testlerine tabi tutulurlar ve başarılı dizilerden başarısız olan sayılar hariç "Rasgele Sayı Havuzuna" aktarılırlar. "Rasgele Sayı Havuzu"'ndaki sayı miktarı 125 Kbyte'ın altında olduğunda, test edilen sayı miktarı 1250 Kbyte'a ulaşıncaya kadar testler tekrarlanır ve veriler tekrar örneklenir. Test sonuçları olumlu olursa, havuzdaki rasgele sayı miktarı test edilen değerler kullanılarak 1250 Kbyte'a tamamlanır. Sonuçta, donanımsal olarak kararsız bir şekilde üretilen rasgele sayıların sadece donanımsal olarak gerçeklenmiş 5 istatistiksel testi değil aynı zamanda yazılımsal olarak gerçeklenmiş NIST testlerini de geçmesi gerekmektedir.

Örnek olarak, 335 Mbit uzunluğundaki ikili dizi donanımsal olarak gerçeklenmiş GRSÜ'den elde edilmiş ve NIST test suitine tabi tutulmuştur. *P*-değerlerinin dağılımlarının düzgünlüğünü ve dizilerin geçme oranlarını gösteren sonuçlar Tablo 5.1'de verilmiştir. 335 Mbit örnekleme uzunluğunda, rasgele gezinim (random excursion) testi dışındaki tüm testler için en düşük başarım oranının yaklaşık olarak

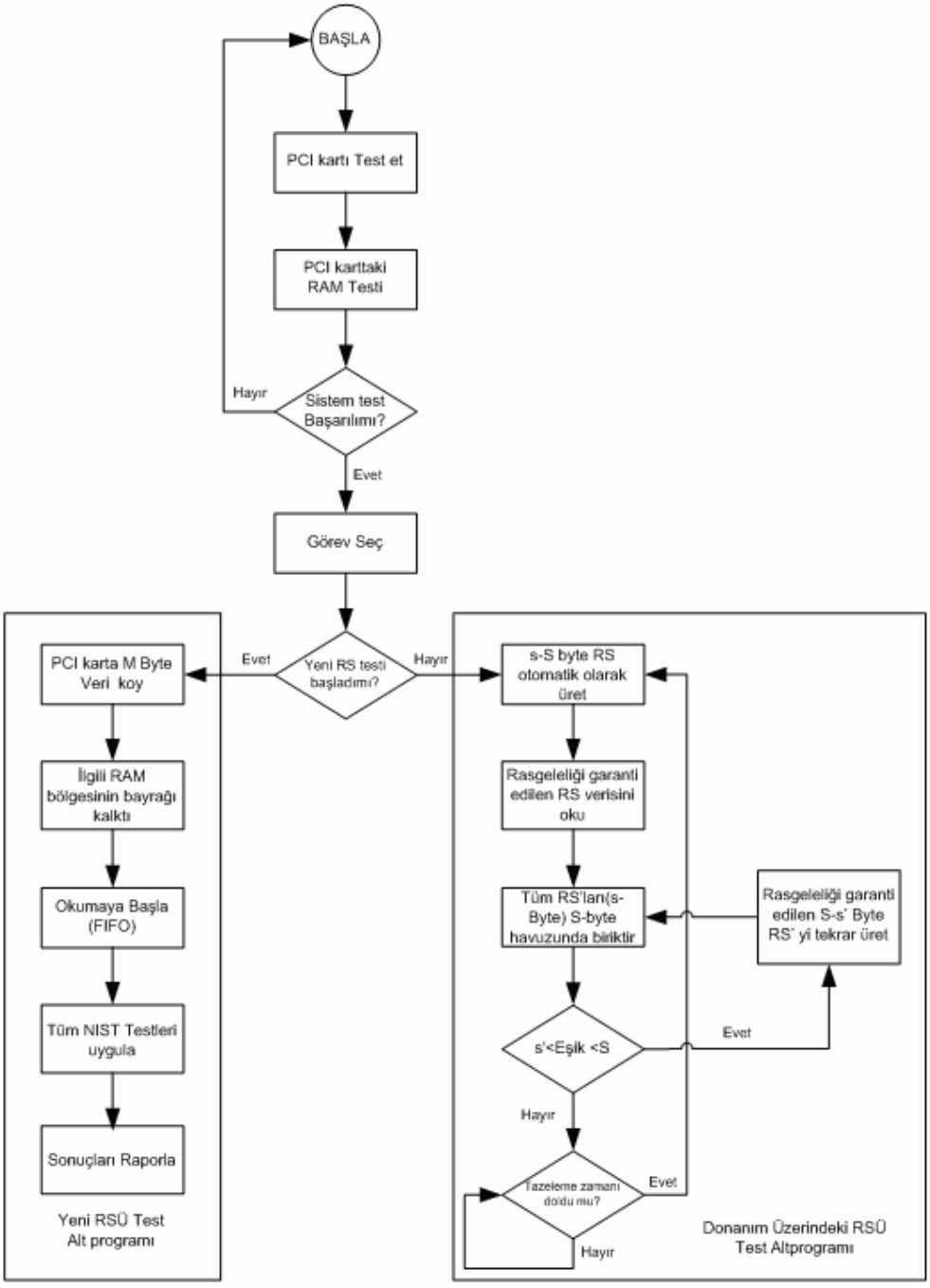
0,973691 olduğu gösterilmiştir. Çevrimiçi test ve veri toplama modunda, donanımsal olarak gerçeklenmiş testlerin sonuçlarını gözardı ederek, tüm üretilen bit dizileri PCI arayüzü üzerinden yüklenmiş ve tüm NIST test suitine tabi tutulmuştur. Üretilen rasgele dizinin tespit edilen boyutu ve test sonuçları hafızada ilgili dizine kaydedilmiştir. Son olarak donanımsal olarak gerçeklenen istatistiksel testlerin, daima yazılımsal olarak uygulanan NIST rasgele sayı test suiti ile paralel sonuçlar ürettiği deneysel olarak gösterilmiştir.

Tablo 5.1 Donanımsal Olarak Gerçeklenen GRSÜ için NIST Test Sonuçları.

C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	P-Değeri	Geçme Oranı	İstatiksel Test
46	31	34	27	37	29	31	27	38	35	0.399069	0.9738	Frekans
28	37	23	37	39	22	35	36	39	39	0.208233	1.0000	Blok-Frekans
44	37	35	28	38	25	30	42	25	31	0.185800	0.9761	Kümlt. Topl.
49	42	38	29	28	30	28	34	30	27	0.101421	0.9761	Runs
28	33	39	40	31	31	36	26	41	30	0.580520	0.9940	Longest-Runs
46	20	43	39	41	25	28	34	21	38	0.004068	0.9821	Rank
29	37	31	30	41	26	34	37	36	34	0.778606	0.9940	FFT
31	42	34	35	34	28	33	31	35	32	0.924811	1.0000	Periyodik Olm.
28	29	35	32	29	31	28	36	43	44	0.373012	0.9940	Örtüşen
41	34	20	36	39	34	32	33	33	33	0.483177	0.9851	Üniversal
36	39	36	33	37	27	39	34	28	26	0.681194	0.9910	ApEn
22	9	22	20	19	19	20	23	11	26	0.157901	1.0000	Rasgele Gezn.
19	13	22	21	18	21	18	16	21	22	0.907914	0.9948	Rasgele Gezn. Var.
39	33	35	32	29	36	32	29	31	39	0.928429	0.9881	Seri
40	29	33	31	32	41	31	32	37	29	0.818179	0.9940	Doğrusal Karmaş.

Uygulama olarak, istatistiksel rasgele sayı testleri, Intel 32-bit x86 işlemciler için gömülü geliştirme ortamı olan ve Windows uyumlu bir gerçek-zamanlı işletim sistemi (RTOS) ve gerçek-zamanlı Gömülü Araç Demeti (ETS) çekirdeği içeren Venturcom'un Phar Lap ETS üzerinde gerçeklenmiştir. Her nekadar PCI kartı Phar Lap işletim sistemi içeren bir gömülü PC üzerine takılmışsa da, OnTime gibi başka bir gömülü işletim sistemi üzerine veya uygun bir sürücü ayarlanarak Windows, Linux veya UNIX kullanan herhangi bir standart PC ortamına da takılabilir. Fakat testlerde kullanılan Gerçek zamanlı ETS Çekirdeği endüstriyel güçte gömülü uygulamaların ihtiyaçlarını karşılamak üzere üretilmiştir. ETS, gömülü sistemler için üretilen en kompakt Win-32 tabanlı işletim sistemi olarak kabul edilmekte ve Windows İşletim Sisteminin uygulanabilirliğini taşıyarak tasarımlar için çok kompak bir seçenek olmaktadır.

Bütün rasgele sayı test ve üretici altprogramları Phar Lab işletim sistemli gömülü bir kişisel bilgisayarda (PC) yürütülmüştür. Yazılımın akış diyagramı Şekil 5.6' da sunulmuştur.

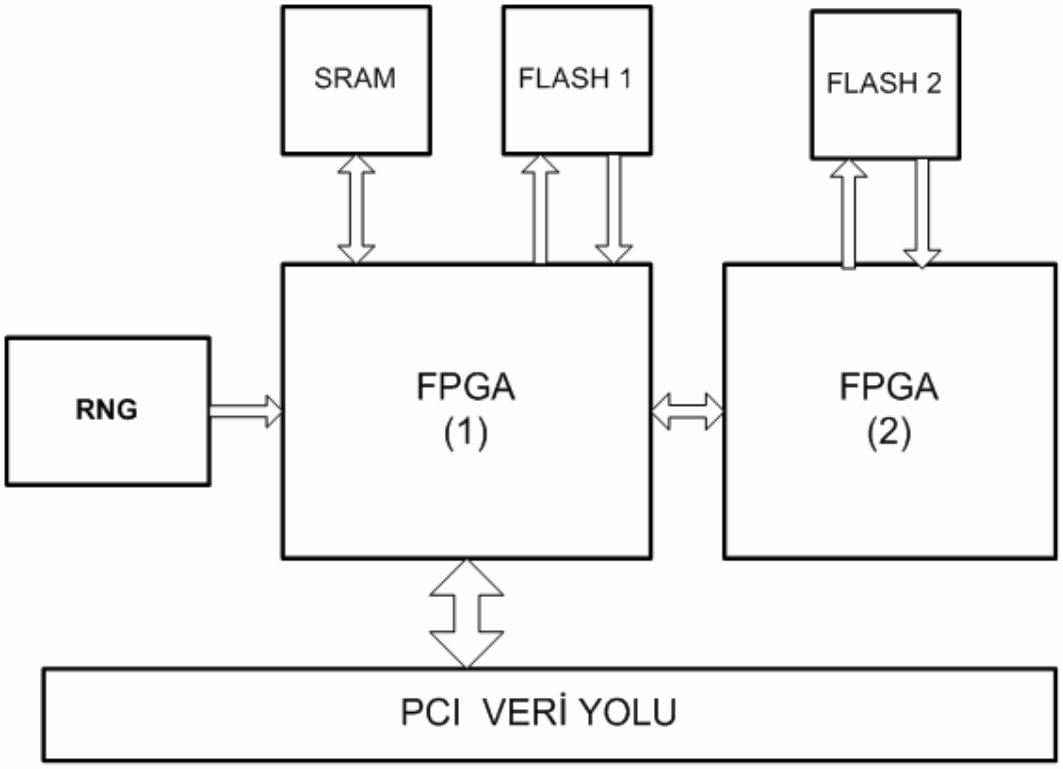


Şekil 5.6. Yazılım Akış Diyagramı

Test algoritması, standart çevre birimleri bağlantı kartı (PCI) denetim işlemleri ve bellek denetimleriyle başlar. Donanım, kartın ilklendirme aşamasında bir problem ortaya çıkarsa uyaracak şekilde tasarlanmıştır. Test işlemleri, PCI veriyoluna ve kartın rasgele erişimli belleğine (RAM) uygulanan kontrollü okuma/yazma işlemleri tarafından gerçekleştirilir. Bir hata oluşmazsa, yazılım geliştiricinin tercihine göre yeni RSÜ test yada rasgele sayı üretim modu seçilir. PCI kart donanımı, hem rasgele sayı testine, hem de çevrimiçi rasgele sayı üretimine imkan sağlar. Testlerin olumlu olduğu durum, üretilen sayıların rasgele olmasını garanti eder. PC ortamındaki yazılımda gerçekleştirilmiş testler, hızlı ve tam NIST rasgelelik testlerine ve yazılım geliştirici için açık bir raporlama hizmetine olanak sağlar. Diğer yandan, üretilen rasgele sayılar çevrimiçi rasgelelik testlerini geçtikten sonra, gerçek zamanlı uygulamalarda kullanılabilir.

5.5. PCI Arayüzünün Donanım Gerçekleştirimi

Şekil 5.7 ve 5.8' de sunulan, tasarımın ana devre ve detaylı FPGA devre şeması, RSÜ Sonlu Durum Makinesi (FSM), Von Neumann, Örnekleme Saati Üretici, Blok Transfer FSM' si ve Blok Transfer FSM FIFO'su üst düzey bloklarından oluşur. Burada RSÜ FSM, test makinelerini ve onların kontrol mantıklarını içerir. Testler, uygulama yazılımının "Test Başlangıç" bitinin aktif edilmesi ile başlar. Beş temel test paralel olarak çalışırken, rasgele sayı adayları Blok Transfer FSM FIFO'suna aktarılır. Bu aday sayılar aynı zamanda kart üzerindeki RAM' de de tutulmaktadır. Aday rasgele sayılar rasgelelik testlerinden geçmesi durumunda RAM' de ve PC'nin kullanması için de FIFO' da tutulurlar. Bir sonraki RSÜ çıktıları yeni test için tekrar aday havuzuna alınır ve teste tabi tutulurlar. Bu durum rasgelelik testlerini geçen sayıların tutulduğu RAM dolana kadar devam eder. RAM' in dolması durumunda, belirli periyotlarda yeni rasgele sayılar ile RAM deki eski rasgele sayılar tazelenir. Böylece RAM' deki Rasgele sayılar sürekli yeni üretilen rasgele sayılarla yer değiştirmiş olur.

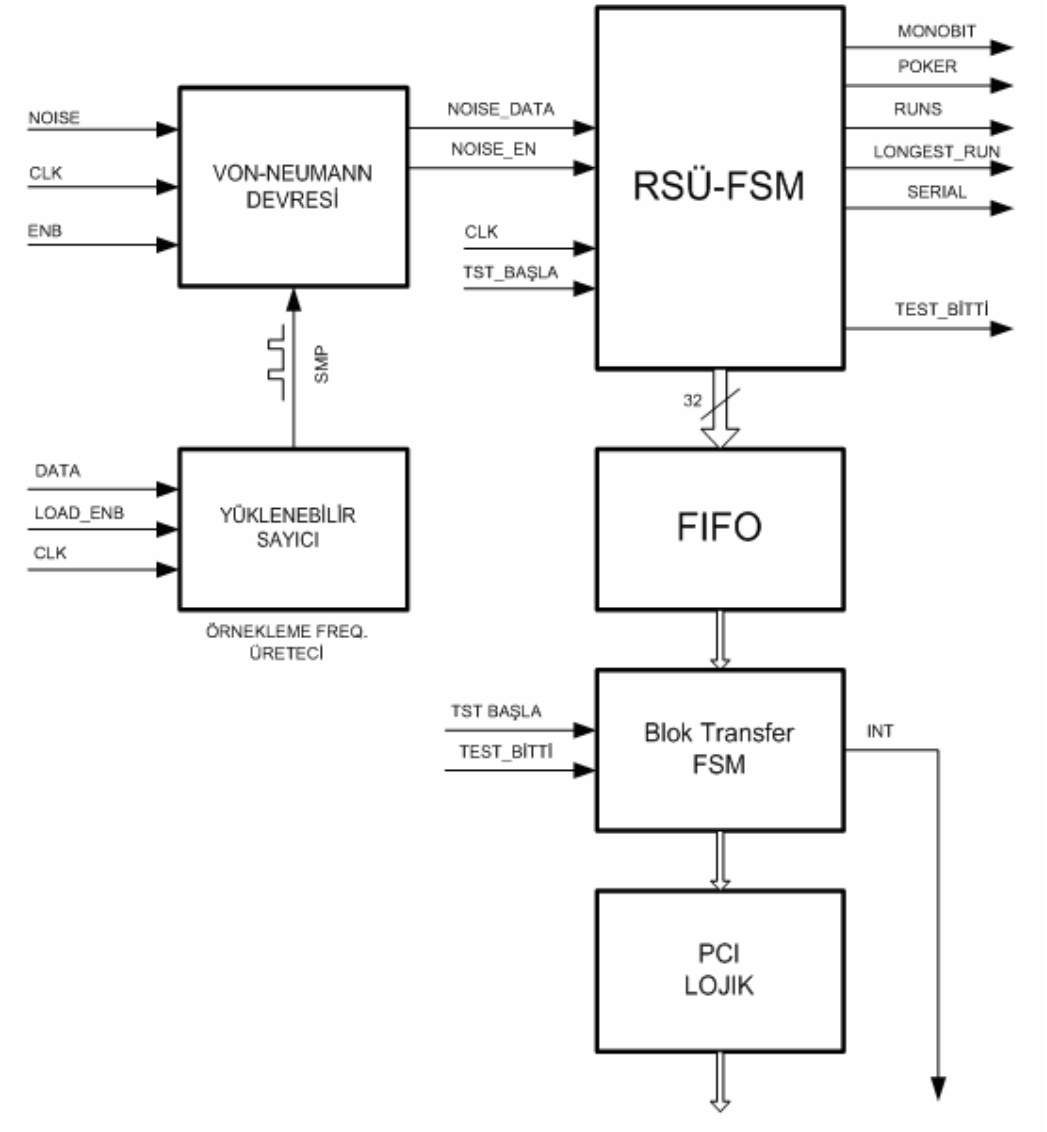


Şekil 5.7. Tasarlanan Donanımın Blok Diyagramı

5 temel test olan frekans (mono-bit), poker, koşular (runs), uzun-koşu (long-run) ve seri testleri kriptografik modüllerin güvenlik şartlarını sağlamakla beraber önerilen herhangi bir RSÜ donanımının özelliklerini belirlemede etkindir. FIPS 140-2 Test Suitinin [22] önerdiği kabul aralığı bu testler temel alınarak belirlenmektedir.

5.6. Donanımsal Olarak Uygulanan İstatistiksel Testler

Frekans (mono-bit) testi: Bu testin amacı, üretilen bit dizisindeki "0"ların ve "1" lerin sayısının aşağı yukarı eşit olduğu durumların belirlenmesidir. Kabul aralığı 9725 < M1 < 10275 iken burada M1 üretilen bit dizisindeki "1" lerin sayısını ifade eder. Verilen bu değerler 20000 bitlik dizi içindir.

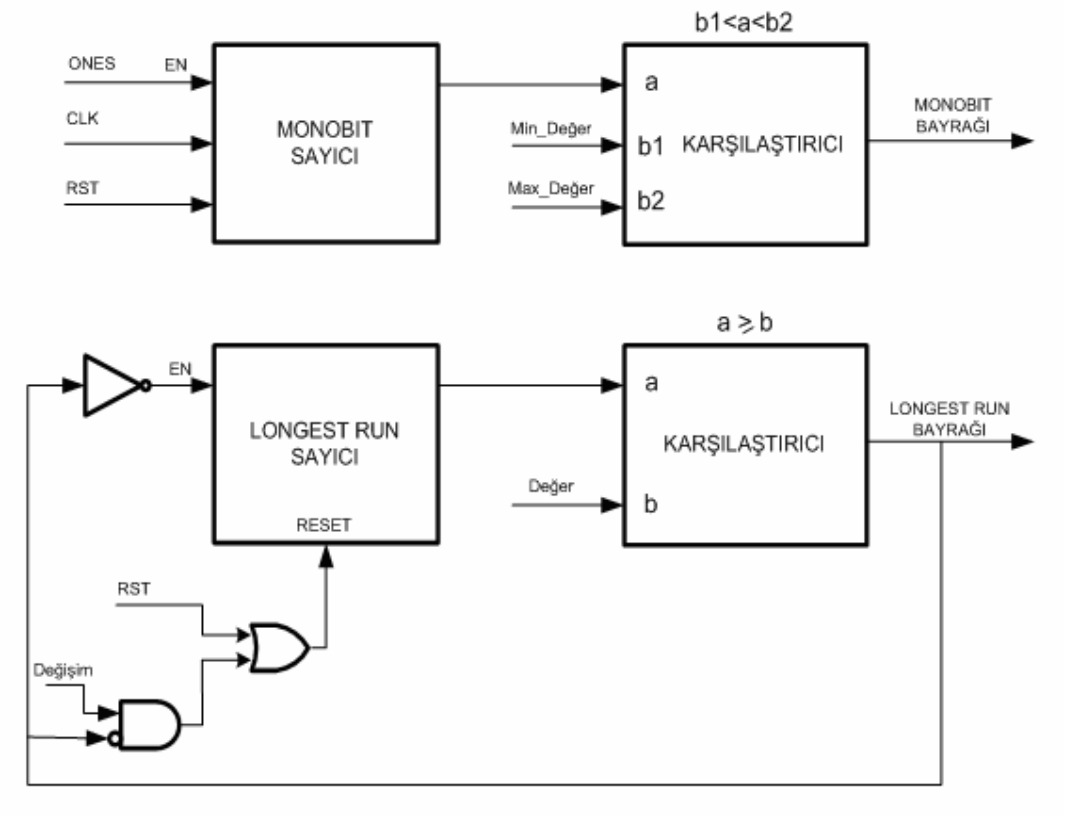


Şekil 5.8. Detaylı FPGA Blok Diyagramı

Uzun-koşu (Long-Run Test) testi: Bu testin amacı, üretilen bit dizisinde L1 uzunluğunda "0" dizisi veya "1" dizisinin varlığını araştırmaktır. Örneğin, FIPS 140-2 test suitinde L1=26' dır.

Monobit ve uzun-koşu testlerinin blok diyagramı Şekil 5.9' da gösterilmiştir. Bu iki testin gerçeklenmesinde bir sayaç ile karşılaştırma blokları kullanılmaktadır. Monobit testi "1" leri saymak için bir ikili sayaç kullanırken, uzun-koşu testi yine bir ikili sayacı bu defa FIPS 140-2 test suitinde 26 olarak tanımlanan uzun-koşu limitine ulaşılıp ulaşılmadığını test etmekte kullanır. Örneklenmiş verideki değişimler uzun-koşu

sayacını sıfırlar. Eğer 26 veya daha uzun bir koşuda değişim yoksa sayaç sona erer ve test başarısız olur.



Şekil 5.9. Monobit ve Long-Run Testlerinin Blok Diyagramı

Poker testi: Bu testin amacı m uzunluğundaki dizilerin, üretilen tüm diziler içinde bulunma sıklığının aşağı yukarı aynı sayıda olma durumunu incelemektir. Bit dizisi, her biri m uzunluğunda olan k ($k = \llcorner n/m \lrcorner$) adet üst üste binmeyen bölüme ayrılır. *i*. ($1 \le i \le 2^m$) tip dizinin gerçekleşme sayısı n_i olmak üzere, bu testin kabul edilme aralığı ($2.16 < P_1 < 46.17$) dır. Öyle ki m = 4 için,

$$P_1 = \frac{2^m}{k}\left(\sum_{i=1}^{2^m} n_i^2\right) - k \quad , \quad P_1 = \frac{16}{5000}\left(\sum_{i=1}^{16} n_i^2\right) - 5000 \qquad \text{dir.} \tag{5.1}$$

Runs testi: Runs testinin amacı rasgele bir bit dizisi içinde farklı uzunluklarda bulunabilecek birbirini takip eden "0" ya da "1" serilerinin miktarını bulmaktır. O_i birbirini takip eden "1" lerden oluşan seri miktarını ifade ediyor olsun. Z_i ise birbirini takip eden 0 lardan oluşan seri miktarını ifade ediyor olsun. e_i ($e_i = (n - i + 3)/2^{i+2}$), *n*

uzunluğundaki bir bit dizisi içinde bulunan, i ($1 \leq i \leq k$) uzunluğunda, birbirini takip eden "0" ve birbirini takip eden "1" lerden oluşan serilerin miktarını ifade eder. Bu durumda istatistik (5.2)' de verilmiştir.

$$R_1 = \sum_{i=1}^{k} \left(\frac{(O_i - e_i)^2}{e_i}\right) + \sum_{i=1}^{k} \left(\frac{(Z_i - e_i)^2}{e_i}\right) \tag{5.2}$$

Seri (three-bit) testi: Bu testin amacı bit dizisi içinde gözlemlenen 000, 001, 010, 011, 100, 101, 110, 111, 00, 01, 10 ve 11 miktarının rasgele bir diziden beklendiği gibi yaklaşık olarak eşit olup olmadığını tespit etmektir. n_{ijk} i, j ve k "0" yada "1" iken ijk' nın kaç kere gözlemlendiğini ifade ediyor olsun. Bu test için kabul aralıkları:

$$S_3 = n_{000}^2 + n_{001}^2 + n_{010}^2 + n_{100}^2 + n_{101}^2 + n_{110}^2 + n_{111}^2 \ , \tag{5.3}$$

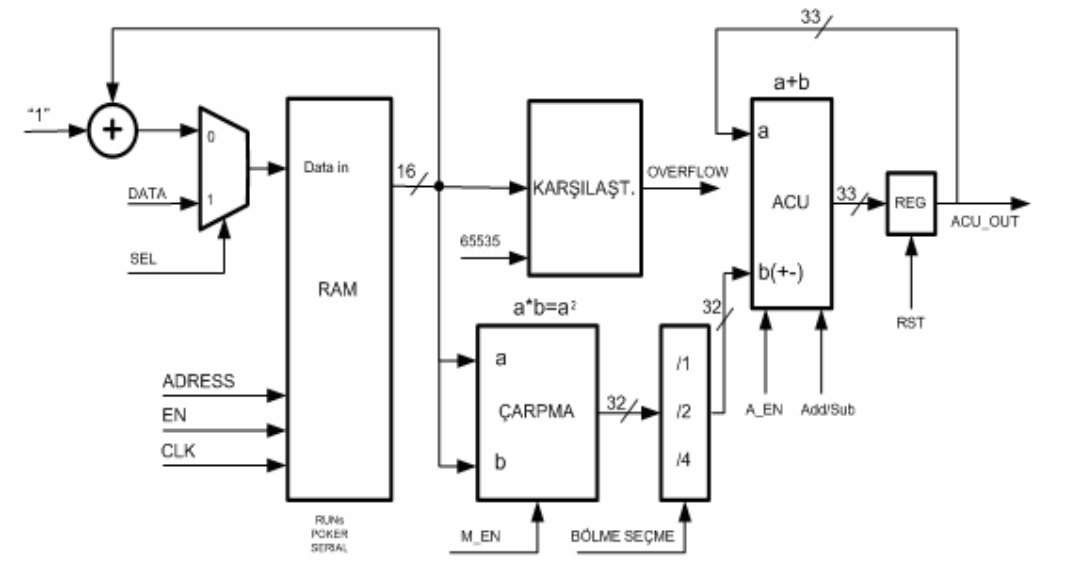
$$S_2 = n_{00}^2 + n_{01}^2 + n_{10}^2 + n_{11}^2 \ , \quad \text{ve} \quad S_1 = n_0^2 + n_1^2 \quad \text{iken} \tag{5.4}$$

$$0 \leq S_3 - \frac{S_2}{2} \leq 47698 \qquad \text{ve} \qquad 0 \leq S_3 - S_2 + \frac{S_1}{4} \leq 34555 \tag{5.5}$$

dir.

Poker, Seri ve Runs testleri öncekilerden farklı çalışır. Poker ve Seri testlerinin Şekil 5.10'da gösterilen blok diyagramları yükleyici, karşılaştırıcı, çarpıcı ve RAM bloklarından oluşmaktadır.

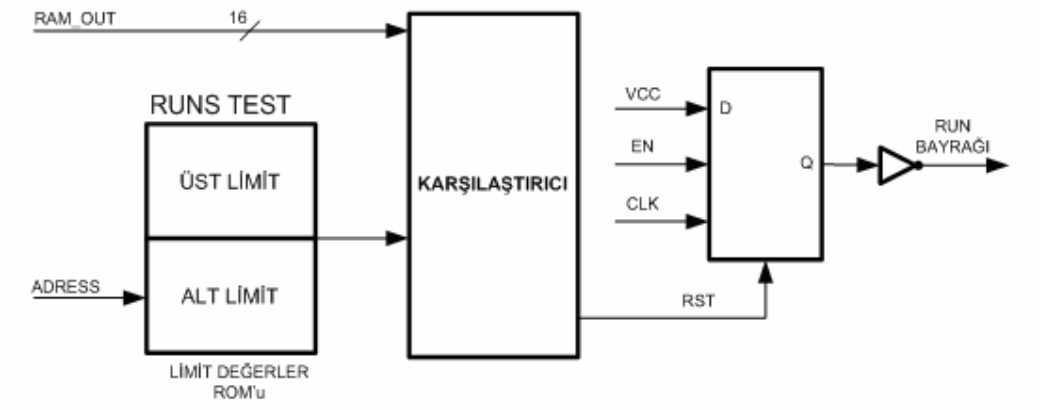
Şekil 5.10' da, Rasgele sayı bit dizilerindeki birli, ikili, üçlü, dörtlü bit dizilişlerinin olası durum sayılarının değeri RAM' de saklanır. Toplam olası durum değeri RAM çıkışındaki işlem bloklarından geçirilerek Seri ve Poker testlerine ait sonuçlar elde edilir.



Şekil 5.10. Poker ve Seri Testlerin Blok Diyagramı

Runs test bloğu Şekil 5.11' de gösterilmiştir. Yükleyici, karşılaştırıcı ve RAM bloklarından oluşmaktadır.

Bu testler değerlendirilmeden önce, bütün verinin gerekli parçaları, Poker testinde çakışmayan dört bitlik grupların tüm olası durumları ve Seri testte çakışan iki ve üç bitlik grupların tüm olası durumları sayılır. Runs testi değişmeyen veri akışlarının miktarını uzunluklarına göre sayar. Sayım bir RAM modülünde bir akümülatörle (toplayıcı) devam eder ve sonra, akümülatör lojiği poker ve seri testlerinin sonuçlarını hesaplar. Sonuçlar, doğru aralıkta olmaları için, bir karşılaştırıcı lojik tarafından kontrol edilir.



Şekil 5.11. RUNS Testinin Blok Diyagramı

Blok Transfer FSM FIFO: Testler çalışırken, aday sayılar sistem hafızasına PCI arayüzü yoluyla blok yazma metodu ile transfer edilir. Bunun için Blok Transfer FSM birimi "Arbiter" dan PCI veri yolunu talep eden portu aktif eder. Yazma işlemine başlamak için GNT işaretini bekler. Eğer veri yolu kullanılabilir değilse, bir kısım veri, Blok Transfer FSM FIFO' da beklemek zorundadır. Bu geçici hafıza (512 x 16 bit) transfer edilecek veriyi, PCI veri yolu kullanılabilir olduğunda, PC' ye anında aktarır. RSÜ bit dizilerine uygulanan testler bittiğinde bir kesme üretilir, test bitti bayrağı kaldırılır ve aday rasgele sayılar transfer edilir.

Örnekleme frekansı Üreteci: Örnekleme frekansı değiştirilebilir olup, 33 MHz PCI veri yolu saati ya da diğer dış osilatörler örnekleme saati üretmek için kullanılabilirler. Aslında, bu bir saat değil, bir periyot uzunluğunda bir saat etkinleştirme sinyalidir. RSÜ, FSM ve Von Neumann lojik devrelerinde bu saat etkinleştirme sinyalini kullanırlar.

Von Neumann Devresi: Von Neumann'ın de-skewing tekniği 01 bit çiftini "0" çıktısına, 10 bit çifitini "1" çıktısına dönüştürmekten ve 00 ve 11 çiflerini atmaktan ibarettir. Bu yazılımla etkinleştirilebilir ya da etkisiz kılınabilir. Eğer etkinleştirilirse rasgele veri üretim hızı dört kat yavaşlar. Bu mantık iki çakışmayan ardarda örneği izler.

PCI Devresi: FPGA tabanlı bir PCI lojiği, önerilen tasarım ve bilgisayar işlemcisi arasındaki arayüze uygulanmıştır. Tasarımda 32 bit 33 MHz PCI veri yolu kullanılmıştır. 5 V ve 3.3 V işaretleşme durumu kabul edilebilir olup PCI v3.0 desteklenmiştir. Aday rasgele sayılar blok yazma modunda transfer edilmiştir. Kesme işareti test bitiş durumunu göstermek için kullanılmıştır. Kesme onayı test bitiş saklayıcısının okunması ile yapılır. Her testin üst ve alt limitleri uygulamaya göre yazılım ile ayarlanabilir. FPGA FIPS 140-2 test program suitinde raporlanan varsayılan değerlerle başlar. Örneklemiş veri aynı zamanda PC' de koşan yazılım tarafından okunabilen bir saklayıcıda saklanmıştır. FPGA kapasitesinin yaklaşık % 25' i PCI devreleri için kullanılmaktadır. Kalan kısmı ise kontrol ve istatistiksel test devreleri için kullanılmıştır. Son olarak önerilen tasarımın doğru çalıştığını ve sağlamlığını test etmek amacıyla 50 adet 40 MByte uzunluğunda rasgele sayılar toplanmış ve bu sayılara ilişkin donanımsal olarak koşturulan gerçek zamanlı test sonuçları her bir 20000 bit uzunluklu dizi için sırayla kaydedilmiştir. Toplanan dizilere aynı istatistiksel testler yazılımsal

olarak da uygulanmış ve elde edilen sonuçların tasarım tarafından üretilen kaydedilmiş sonuçlarla birebir örtüştüğü doğrulanarak önerilen tasarımın doğru çalıştığı ve sağlamlığı deneysel yöntemlerle onaylanmıştır. Yani rasgele aday sayılar FPGA üzerinde donanımsal olarak çalışan FIPS 140-2 test suitine tabi tutulmuş, sonuçları olumlu çıkmış ve aynı aday sayılar PCI üzerinden PC'ye aktarılıp, PC'de koşan NIST test suitine tabi tutulmuş ve sonuçların aynı olduğu gözlemlenmiştir.

6. BÖLÜM

UYGULAMALAR

6.1. Tasarlanan FPGA Tabanlı Kart ile Yapılan Güvenlik Uygulamaları

Bu bölümde PCI arayüzlü FPGA tabanlı kart ile yapılan 2 biyometrik uygulama anlatılacaktır. Bu uygulamalar:

1. Güvenli biyometrik kimlik doğrulama sistemleri için donanım tabanlı bir Gerçek Rasgele Anahtar Üreticisi (GRAÜ) [38].
2. Donanımsal GRSÜ tabanlı güvenli parmak izi doğrulama sistemi [39].

6.2. Güvenli Biyometrik Kimlik Doğrulama Sistemleri için Donanımsal Bir GRAÜ

6.2.1. Giriş

Bu bölümde, bir kişinin yüz şablonunu şifrelemekte kullanılan özel anahtarı oluşturan FPGA tabanlı GRSÜ' yü içeren bir biyometrik kimlik doğrulama sistemi anlatılmaktadır. Tasarlanan donanım kolaylıkla standart veya gömülü bir PC'ye, PCI ara yüzüyle rasgele sayı üretimi için kolaylıkla takılabilir. Özel anahtarı oluşturan rasgele sayıların gerçek olması, iki seviyeli rasgelelik testini geçtiğinden garantilenmiştir. Rassallık testi ilk olarak donanımda daha sonra PC üzerinde NIST test suiti kullanılarak analiz edilmiştir. Sistem, kişiye ait gizli biyometrik veriyi saklamak için AES (Advanced Encryption Standard) şifreleme algoritmasını kullanmaktadır. GRSÜ tarafından üretilen özel anahtar tek ve gerçek rasgele bir parola oluşumunu garantilemektedir. Sistem Dalgacık Fourier Mellin Dönüşümü (Wavelet Fourier-Mellin Transform - WFMT) tabanlı yüz özniteliklerini indeks numaralarıyla birlikte veri tabanında tutmaktadır ki, bunlar akıllı kartlarda veya andaçlarda (token) saklanabilir. Çalışmanın amacı, biyometrik bir teknolojinin donanım tabanlı GRSÜ ile bileşiminin pratik uygulamasını sunmaktır.

GRS'lar sayısal imzaların üretiminde kullanıldığından, sistemi biyometrik tabanlı kimlik doğrulama sistemlerinde bulunan GRS tabanlı anahtar isteyen kripto tasarımları için nihai donanıma entegre etme düşüncesi ilginç durmaktadır. Çalışmada, WFMT tabanlı yüz doğrulama sistemi [40,41] anlatılmaktadır. Bu sistemde yüz şablonları özel anahtarlarla şifrelenmektedir. Çalışmanın literatüre en önemli katkısı, poza göre değişmeyen WFMT yüz özniteliklerinin güvenli yüz şablonu saklama sistemi ile entegrasyonudur. Güvenli yüz şablonu saklama sistemi AES tabanlı şifreleme ile gerçekleşmektedir. Sistem GRS'lerden üretilen özel anahtarlarla güvenilirliği garantilemektedir. FPGA tabanlı bu sistemin herhangi standart veya gömülü bir PC'ye takılabilir olması da diğer bir avantajdır.

RSÜ modunda donanım tabanlı GRSÜ çok bilinen bir teknik olan ısıl gürültüyü kullanmaktadır.

Bölüm 5' de anlatılan donanım üzerinde üretilen rasgele sayılar, FPGA üzerindeki istatistiksel değerlendirme testlerine tabi tutulurlar. Olası rasgele sayılar iki mekanizma tarafından değerlendirilir, bunlar donanımsal ve yazılımsaldır. Donanım değerlendirme mekanizması bit dizilerini beş temel testte (mono-bit, poker, runs, uzun-koşu ve seri testler) tanıtıldığı gibi saymaya başlamak için yazılım mekanizması tarafından etkinleştirilir. Bu testler kripto algoritmalarında kullanılan RSÜ' ler için gerekli güvenlik gereksenimlerini kapsadığı gibi tavsiye edilen istatistiksel özellikleri de belirtmektedir. Her test FPGA üzerinde, donanım tabanlı RSÜ' den oluşturulan 100000 bitlik ardışık dizi çıktısının işlem görmesiyle gerçekleştirilir.

Test programı koştuğunda, yazılım FPGA'yi kullanarak rasgelelik testlerini başlatır. Test süresince FPGA üzerinden rasgele olduğu kabul edilen değerler okunur ve saklanır. Testler (Von Neuman algoritması ve beş istatistik test) tamamlandığında, test sonuçlarının adresi FPGA tarafından okunur ve değerlendirilir. Eğer bütün test sonuçları olumlu ise tutulan değer bellekteki "Rasgele Sayı Aday Havuzu" na iletilir, olumsuz durumda ise değerler sayı havuzunda tutulmaz.

Eğer rasgele sayılar kripto veya güvenlik amaçlı uygulamalar için gerekiyorsa, en az ikisi fiziksel olarak bağımsız olan üçten az olmamak şartıyla farklı ortamlarda bu sayıların rasgeleliği test edilir. Bu şartı sağlamak için FPGA' den fiziksel olarak bağımsız çalışan NIST istatistik test suiti (yazılım olarak) eklenmiştir [21].

"Rasgele Sayı Aday Havuzu" nda tutulan sayılar NIST istatistik test suiti ile analiz edilerek, başarılılar "Rasgele Sayı Havuzu" na transfer edilirler. "Rasgele Sayı Havuzu" ndaki sayı miktarı 125 Kbyte' ın altına düştüğü zaman, testler tekrar başlar, veri tekrar örneklenir ve bu işlem test değerlerinin miktarı 1250 Kbyte'a çıkıncaya kadar devam eder. Eğer sonuçlar olumlu ise havuzdaki rasgele sayı miktarı test değerleri kullanılarak 1250 Kbyte'ta tamamlanır. Sonuç olarak, donanımda bilinemeyecek şekilde oluşturulan rasgele sayılar donanımda uygulanan beş temel istatistik testin yanı sıra yazılımda koşturulan NIST test suitinin de kontrolünden geçmelidir.

6.2.2. WFMT Öznitelikleri

Günümüzde popüler olarak kullanılan yüz öznitelikleri imge tabanlıdır. İmge tabanlı stratejilerin yüksek işlem verimliliği yanında düşük çözünürlüklerde bile etkin olduğu ispatlanmıştır. Buna rağmen imge tabanlı stratejiler şekil bozukluklarına, poz değişimlerine, dönme, öteleme, kayma ve ölçek varyasyonuna hassasiyetleri ile bilinmektedir. Tümleşik WFMT öznitelikleri yüzün temsilinde kullanılmak için önerilmiştir. Dalgacık dönüşümü sadece yerel kenarların korunması için değil aynı zamanda imge ayrıştırmadan sonra düşük frekans bölgesinde gürültü azaltma için de kullanılmaktadır. Böylece elde edilen yüz imgeleri şekil bozulmalarına daha az hassas bir hale getirilmiş olur. Diğer yandan, Fourier-Mellin Dönüşümü (Fourier Mellin Transform - FMT) çok iyi bilinen dönme, ölçekleme ve kayma (Rotation, Scale, Translation - RST) problemine karşı gürbüz bir özniteliktir ve gürültü altında yüksek başarım sağlamaktadır [40].

Tipik 2-B (2 Boyutlu) işaretlerde ayrıştırma algoritması 1-B işaretler için geliştirilen algoritmalar ile çok benzerdir. 2-B dalgacık dönüşümü dört eleman içinde *j-1* seviyesinde yakınsak katsayıların ayrıştırılmasını sağlar; *j* seviyesinde yakınsamalar ve üç yöndeki (Yatay, dikey ve çapraz) detaylar aşağıdaki şekildedir:

$$L_j(m,n) = [H_x * [H_y * L_{j-1}]_{\downarrow 2,1}]_{\downarrow 1,2}(m,n) \quad (6.1)$$

$$D_{j\ \text{dikey}}(m,n) = [H_x * [G_y * L_{j-1}]_{\downarrow 2,1}]_{\downarrow 1,2}(m,n) \quad (6.2)$$

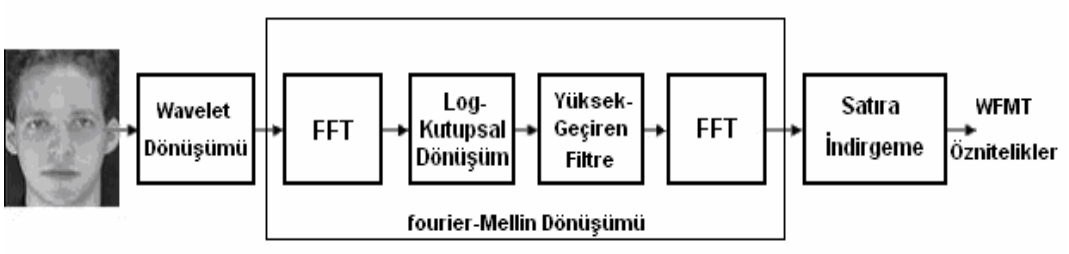
$$D_{j\ \text{yatay}}(m,n) = [G_x * [H_y * L_{j-1}]_{\downarrow 2,1}]_{\downarrow 1,2}(m,n) \quad (6.3)$$

$$D_{j\ \text{çapraz}}(m,n) = [G_x * [G_y * L_{j-1}]_{\downarrow 2,1}]_{\downarrow 1,2}(m,n)\} \quad (6.4)$$

Formülde "*" evrişim (konvolüsyon) operatörünü ve ↓2,1 (↓1,2) satır ve sütün boyunca altörneklemeyi temsil etmektedir. H alçak geçiren, G ise band geçiren filtredir.

Enerji içeriğinin düşük frekans bandında L_j bulunduğu yaygın bir gözlemdir. D_j 'ler düşük enerji içeriği ve yüksek geçiren özelliğinin kenar detaylarının iyileştirilmesi yanında gürültü ve şekil bozukluklarını artırmasından dolayı pek kullanılmaz. Buna rağmen alt band L_j orijinal yüzün çok gürültülü olmayan yumuşatılmış versiyonudur ve yerel kenarlar düşük bozulmalara karşı hassas bir şekilde saklanmıştır. Burada dikkat edilmesi gereken nokta seçilen dalgacık tabanının L_j' nin ne kadar enerji saklayabileceğini etkiliyor olmasıdır.

Şekil 5.1'de WFMT özniteliklerini elde etmek için kullanılan yordam gösterilmektedir. Önce I(x,y) imgesi dalgacık dönüşümü ile ayrıştırılır. Bu ayrıştırma L_j' ye *n* defa özyinelemeli olarak uygulanır (L_0=I(x,y), j=0,…,n). Daha sonra L_j'e FMT uygulanır. FMT öncelikle L_j 'e Hızlı Fourier Dönüşümü (Fast Fourier Transform - FFT) ile başlar ve Log-kutupsal dönüşümü ile devam eder. Örnekleme ve yuvarlamalardan kaynaklanan hataları gidermek için bir yüksek geçiren filtre, $H(x,y) = (1-\cos(\pi x)\cos(\pi y))(2-\cos(\pi x)\cos(\pi y))$, $-0.5 \le x,y \le 0.5$ olmak üzere, uygulanır. Böylece WFMT'yi elde etmek için filtrelenmiş imgeye ikinci FFT dönüşümü uygulanır. Sonuç vektörü V_{wfmt} son WFMT imgesinin satırlarını peşpeşe koyarak elde edilir. Literatür, WFMT'nin değişmeyen, bozulma ve gürültüye hassas olmayan öznitelikler ürettiğini göstermektedir. WFMT kullanmanın diğer bir avantajı da imgenin yoğunluğunu düşürüp, gerçek zaman uygulamaları için uygun hale getirmesi olabilir.



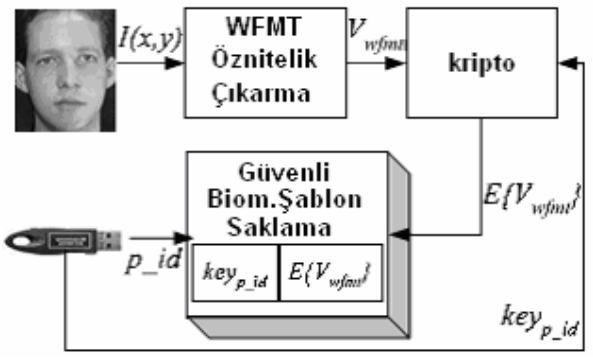
Şekil 6.1. WFMT Özniteliklerini Üreten Blok Diyagram

6.2.3. Güvenli Yüz Doğrulama Yöntemi

Güvenli yüz doğrulama yöntemi, WFMT tabanlı yüz özniteliklerini kullanarak geliştirilmiştir. Aslında bu şablon, birçok biyometrik öznitelik uygulamasına (parmak izi, iris, retina vs.) adapte edilebilir. WFMT tabanlı yüz özniteliklerinin kısıtlı kapalı bir küme içinde kullanılması, kriptografi, rasgele sayı üretimi ve biyometriklerin birbirine entegrasyonu için iyi bir başlangıç noktası sayılabilir.

Sistem her halükarda akıllı kart ve benzeri bir bellek üzerinde tutulan bir kişisel kimlik numarası p_id ve özel anahtar gerektirmektedir. Doğrudan kodlama kullanılması, şifrelerin kolayca unutulabilmesinden dolayı önerilmemektedir.

Doğrulama sistemi iki faz olarak görülebilir: Kayıt ve Doğrulama. Şekil 6.2'de görülen kayıt işleminde kullanıcılar sisteme kaydını yaptırır. Kullanıcılar ilk olarak akıllı kartlarını sisteme takarak kendilerini tanıtırlar. Akıllı kart hem p_id hem de özel anahtar key_{p_id} bilgisini tutmaktadır. Bunun üzerine WFMT özniteliklerini çıkarmak üzere bir kamera ile kullanıcının yüz imgesi I(x,y) kaydedilir. V_{wfmt} dizisi rasgele sayılar kullanan key_{p_id} tarafından kodlanmış bir kod dizisidir. Son olarak kodlanmış öznitelikler, $E\{V_{wfm}\}$, ve her bir p_id numarasına karşılık gelen özel anahtar key_{p_id} bir veritabanına kaydedilir. p_id, kullanıcıların PIN numarası olmasının yanında aynı zamanda veritabanında indeks olarak da kullanılmaktadır.

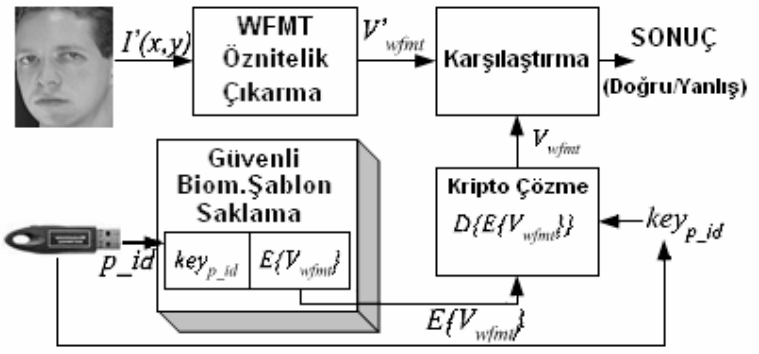


Şekil 6.2. Kayıt Aşaması

Doğrulama fazında Şekil 6.3'de görüldüğü gibi I'(x,y) kullanıcı yüzü kaydedilir ve V'_{wfmt} WFMT öznitelikleri çıkarılır. Doğrulanacak kişiye ait p_id'ye karşılık düşen $E\{V_{wfmt}\}$ seçilir. Kişiye ait key_{p_id} anahtarı yardımıyla şifrelenmiş öznitelik dizisinin şifresi çözülürek kaydedilmiş öznitelik $D\{E\{ V_{wfmt}\}\} = V_{wfmt}$ elde edilir. Son olarak

karar verme mekanizması Öklit-Uzaklığı (Euclidean-distance) tabanlı karar stratejisini kullanarak şablon özniteliği V_{wfmt} ile kodu çözülüp elde edilen sorgu özniteliği V'_{wfmt} karşılaştırır. Eğer uzaklık belirli bir eşik değerinden düşükse I'(x,y) doğrulanmış kabul edilir.

Doğrulama başarılı olursa, tam güvenli erişimi sağlamak için akıllı kart üzerinde FPGA tabanlı GRSÜ ile key_{p_id} 'de değişiklik yapılır.



Şekil 6.3. Doğrulama Aşaması

Sistemin tanıma performansı Olivetti Yüz Veritabanı (ORL) ile test edilmiştir . ORL veritabanında 40 kişinin değişik poz ve yüz ifadeleri bulunan (gülen/gülmeyen, profilden/karşıdan…) 10'ar değişik gri tonlu resim (112x92) bulunmaktadır. ORL yüzlerini en iyi temsil edecek dalgacık filtrelerini bulmak için de birçok deney yapılmıştır. Sonuçlar N adet resmin kalan N-1 adet resimle karşılaştırılması ile elde edilen Gerçek Denklik Oranı (GDO) olarak verilmiştir. Sonuçlara bakılacak olursa tanıma oranı % 87.25 ile % 96.50 arasında değişmektedir. Haar, Daubechies-3, Biorthogonal 1.1 ve Biorthogonal 2.2 diğer dalgacık tiplerinden daha iyi GDO vermektedir. Deneylerde, MATLAB araçlarında tanımlı bulunan 29 adet dalgacık tipi kullanılmıştır. Her bir dalgacık tipi ve buna ilişkin GDO değerleri Tablo 6.1.'de sunulmuştur.

Tablo 6.1 Farklı Dalgacık Tipleri İçin Elde Edilen GDO değerleri

Dalgacık Tipi	*MATLAB Parametresi*	*GDO*
Haar	*haar or db1*	***96.50***

Daubechies	*db2*	*90.50*
	db3	***96.50***
	db4	*89.75*
	db5	*96.00*
	db6	*90.00*
	db7	*96.00*
	db8	*87.25*
Coiflets	*coif1*	*96.25*
	coif2	*90.50*
	coif3	*95.75*
	coif4	*87.00*
	coif5	*93.25*
DMeyer	*dmey*	*90.50*
Biortogonal Dalgacıklar	*bior1.1*	***96.50***
	bior1.3	*96.00*
	bior1.5	*95.75*
	bior2.2	***96.50***
	bior2.4	*95.75*
	bior2.6	*96.00*
	bior2.8	*96.25*
	bior3.1	*90.75*
	bior3.3	*90.25*
	bior3.5	*90.50*
	bior3.7	*90.75*
	bior3.9	*90.25*
	bior4.4	*95.75*
	bior5.5	*90.75*
	bior6.8	*96.25*

Bu sonuçlardan çıkarılabileceği gibi tanıma başarımı, kullanılan dalgacık tipine oldukça bağımlıdır. Başarımın kimi dalgacık tiplerinde (Coiflet-4, Daubechies-8 gibi) düşmesinin nedeni bunların imgeyi ayrıştırırken yüzdeki ayırt edici özellikleri yok

etmesidir. Bu nedenle ayrıştırma yapılacaksa kullanılan biyometriğe uygun ve mümkünse biyometrik algılayıcının sağladığı imgenin özelliklerini göz ardı etmeyen bir dalgacık tipinin seçilmesi veya tasarlanması gerekmektedir. Bu çalışmada % 96.50 GDO sağlayan dalgacık tipleri yeterli güvenlik sağladığı düşünüldüğünden önerilebilir. Ancak ileri çalışmalarda, özellikle daha geniş veritabanlarında veya ışık koşullarının, poz değişimlerinin veya diğer etmenlerin etkin olabileceği gerçek hayat uygulamalarında uygulamalarında daha iyi ayrıştırma sağlayan dalgacık tiplerinin tasarlanması gerekebilir.

Kodlama tarafında AES kullanılmaktadır. Güvenlik modülü modüler bir yapıda olduğundan AES istenildiğinde DES veya Triple DES ile değiştirilebilir. AES için 128 bit anahtarlar kullanılmaktadır. AES, başarımı ve sağladığı yüksek güvenlikle sistemi cazip kılmaktadır. AES, daha büyük blok boyutları, daha uzun anahtarlar ve kriptoanalitik ataklara dayanıklılık sağlayan bir algoritmadır. Burada dikkat edilecek husus key_{p_id} 'nin 128 bit olması ve V_{wfmt} 'nin 128'in bir tamsayı katı olan 1024 Byte olmasıdır.

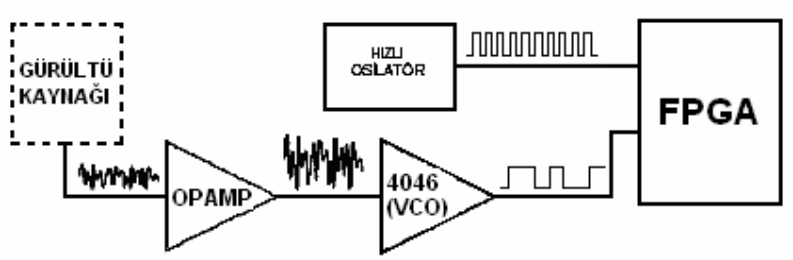
6.3. Donanımsal GRSÜ Tabanlı Güvenli Parmak İzi Doğrulama Sistemi

6.3.1. Giriş

Bu çalışmanın en büyük katkısı parmak izi doğrulama sistemindeki güvenliği arttırmak için donanım tabanlı GRSÜ'lerin kullanılmasıdır. Bu uygulamada, öznitelikleri özel bir anahtarla şifrelenmiş güvenli bir parmak izi doğrulama sistemi anlatılmaktadır. Özel anahtarlar GRSÜ'den elde edilmektedir. Oluşturulan bit dizilerini PC'ye aktarmak için donanımın PCI arabirimine sahip olması, önerilen tasarımı bilgisayar tabanlı kiriptografik uygulamalar için ideal hale getirir.

Bu bölümde anlatılan GRSÜ, ısıl gürültünün yükseltilmesi metodu ile çift osilatör yapısı birleştirilerek tasarlanmıştır. Donanımda bulunan GRSÜ'nün veri hızı 130 Kbps olup gerçek zamanlı uygulamalarda kullanmak için uygundur. Bu çalışmanın temel katkısı maliyet etkin halka-dilim yöntemi tabanlı parmakizi tanıma algoritmasının güvenli donanımsal anahtar üretim mekanizması ile birleştirilmesidir. Sistem GRS'lerden oluşan uygun özel anahtarları oluşturmayı garanti eder. Bu sistemin diğer bir katkısı da FPGA tabanlı sistemin kolayca herhangi bir PC ile tümleştirilebilir olmasıdır. Donanım blok yapısı Şekil 6.4'de gösterilmiştir.

4046 gerilim kontröllü osilatör, yükseltilmiş gürültü voltajı ile düşük saat frekanslarının ayarlanması uygulamasında kullanılır. Gürültü ayarlı düşük saatin yükselen kenarıyla FPGA içindeki D flip-flop kullanılarak hızlı saat çıkışı örneklenir. Gerilim denetimli osilatörün merkez frekansı düşük saatin frekansını belirler ve 4046 için 17 MHz'e ayarlanabilir.



Şekil 6.4. Donanım Olarak Gerçekleştirilmiş GRSÜ (Çift Osilatör Yapısı)

İki osilatör arasındaki sapma, rasgele bit üretiminin daha gürbüz olmasını sağlar. Örneklemeyle ilgili düzlemsel olmayan örtüşme olgusu nedeniyle çift osilatör mimarisi artan ham madde çıkışını ve yüksek istatistiksel kaliteyi sağlar [13]. [42]'de ilişkisiz rasgele bit akışı elde etmek için ayarlı düşük osilatör periyodunun hızlı osilatör periyodundan daha yüksek standart sapma özelliği taşıması gerektiği belirtilmiştir. Çıkış bit dizisinin etkilenmesinin kaldırılması için hızlı osilatörün dengeli doluluk boşluk oranına sahip olması gerekir. Tatmin edici sonuçlar elde etmek için hızlı osilatör, FPGA içindeki düşük seğirmeli 192 MHz kristal osilatör 4'e bölünerek uygulanır. Bu yolla yaklaşık % 50'lik doluluk boşluk oranına sahip 48 MHz hızlı osilatör elde edilmiş olur. Dengeli doluluk boşluk oranı garanti edilebilirse, hızlı osilatör frekansı yükselir.

[42] ve [6]'da kullanılan hızlı ve yavaş osilatörler sırasıyla 1:100 merkez frekans oranlarına sahiptir. Tasarımda, ham veri miktarını belirleyen düşük saat frekansı 520 KHz'e ayarlandığında tam NIST test suitinden deneysel olarak başarılı sonuçlar elde edilmiştir. Sonra, 48 MHz hızlı osilatör düşük saatin yükselen kenarında FPGA içindeki D flip-flop kullanılarak örneklenmiştir. Gürültü ayarlı osilatör tarafından başarılan yüksek seğirmeli düşey hızlı osilatör periyodundan daha fazla standart sapma özelliği gösterir. Sonuçta, bu plan ilişkisiz rasgele bit akışı sonucunu verir. Ancak, bu şekilde elde edilen ikili dizi taraflı olabilir. Bu diziden bilinmeyen etkileri çıkarabilmek için çok iyi bilinen Von Neumann'un de-skewing [17] tekniği kullanılır. Von Neumann

süreci FPGA'de de uygulanabilir. Yaklaşık olarak 4 bitten 1 bit ürettiği için, bu süreç rasgele işaretin frekansını 130 KHz'e düşürür.

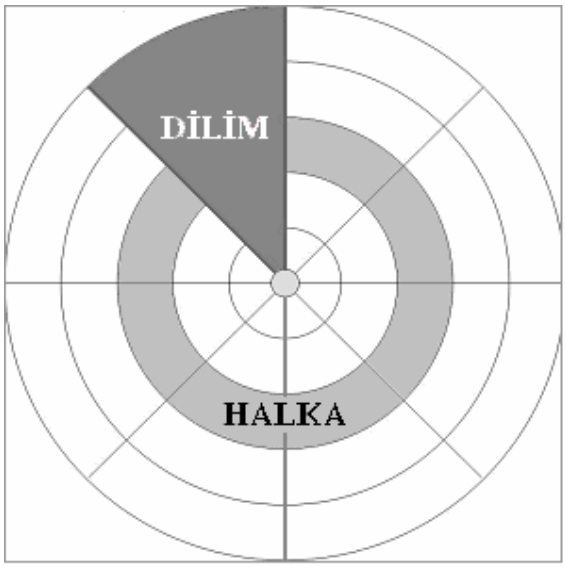
Olası rasgele sayılar donanımsal ve yazılımsal olarak uygulanan iki mekanizmayla değerlendirilir. Donanımsal değerlendirme mekanizması, yazılımsal mekanizma tarafından kriptografik modüller için güvenlik gerektirmelerini sağlayan 5 temel testte (frekans testi, bir blok içinde frekans testi, yinelemeler testi, bir blok içersinde en uzun 1 yinelemesi testi ve seri testler) tanımlanan bit akışlarını saymaya başlamak üzere etkinleştirilir. Aynı mekanizma RSÜ'ler için önerilen istatistiksel testleri de belirtir. 5 testten her biri donanımsal RSÜ'den çıkan 100000 ardışık bit üzerinde gerçekleştirilir. Test programı, çalıştırıldığında FPGA'yı kullanarak rasgelelik testlerine başlar. Testler boyunca yazılım, FPGA üzerinden rasgele olduğu varsayılan değerleri okur ve kaydeder. Von Neumann Algoritması ve 5 istatistiksel test tamamlandığında test sonuçlarının adresleri FPGA üzerinden okunur ve değerlendirilir [39]. Tüm testler pozitif olursa, kaydedilen sayı, hafızadaki "Rasgele Sayı Havuzu"na gönderilir. Ancak, başarısız olan rasgele aday sayılar havuza kaydedilmez.

Eğer rasgele sayılar kriptografik veya genellikle güvenlik konularında kullanılmak üzere gerekiyorsa, rasgele sayı üretimi 3 bağımsız hata içermemeli ve bunlardan en az iki tanesi de fiziksel olarak birbirinden bağımsız olmalıdır. Bu koşulu sağlamak için fiziksel olarak FPGA'dan bağımsız olan yazılımda gerçekleştirilen tam NIST rasgele sayı test suitinde [21] bulunan bir test mekanizması uygulanır. "Rasgele Sayı Aday Havuzu" na kaydedilen başarılı rasgele sayılar yazılımsal olarak tam NIST test suitine maruz bırakılır, başarısız olan rasgele sayılar dışındakiler son havuza gönderilir. Havuzdaki rasgele sayıların miktarı 125 Kb'ın altına düşerse test yenilenir ve veriler test edilen değerlerin miktarı 1250 Kb'a erişinceye kadar tekrar örneklenir. Test sonuçları pozitifse, havuzdaki rasgele sayıların miktarı test edilen değerler kullanılarak 1250 Kb'a tamamlanır.

6.3.2. Halka - Dilim Tabanlı Parmakizi Öznitelikleri

Geleneksel parmak izi tanıma sistemleri genellikle parmak izinin kabartı, vadi ve olay noktası (minutiae – kabartıların sonlandığı veya çatallandığı noktalar) gibi yapısal özellikleri üzerinde durmuştur [43]. Bu metodlarda eşleştirme süreci değişken durumlu olay noktası listelerinin karşılaştırılmasıyla yapılır. Ama olay noktası tabanlı yöntemler

karmaşık işlemler gerektirir ve yapısal özellikleri ortaya çıkarmak için kullanılan ön işlemler zaman kaybına neden olur. Üstelik, olay noktası tabanlı gösterimin değişken doğası nedeniyle donanıma yönelik uygulamalarda kullanılması uygun değildir. Bu problemler, araştırmacıları imgesel yöntemlere kanalize eder. Bu çalışmada ucuz donanımla düşük kaliteli parmak izlerinde tanımaya izin veren gürbüz bir FFT tabanlı algoritma uygulanmıştır. Önerilen yöntemde, öznitelikler Halka-Dilim sınıflandırıcısı kullanılarak iyileştirilmiş imgelerden çıkarılmıştır [44,45]. Bu yöntemde parmak izi imgesinde bir referans noktasına göre düzgelem (üst üste getirme) yapılır. Ardından imgenin FFT'si alınarak frekans imgesi elde edilir. Frekans imgesi dilim ve halkalardan oluşan bir dart tahtası deseni ile kaplanır (bkz. Şekil 6.5). Yöntemin anafikri şudur: Parmak izleri genel olarak periyodik bir yapıdan oluştuğu için onları frekans tanım kümesinde incelemek çok doğaldır. Parmak izi üzerindeki ayırt edici özelliğe sahip düşük frekanslı bölgeler tanımada daha etkindir.



Şekil 6.5. Halka ve Dilimler

Bu yöntem faydalıdır çünkü tüm temel formlar (kabartı yönelimi, frekans haritası ve ilgi bölgesi maskesi) parmak izinin iyileştirilmesinde kullanılan Kısa-Süreli Fourier Dönüşümü (Short-Time Fourier Transform – STFT) analizinden eşzamanlı olarak kestirilir.

Öznitelikleri çıkarmak için parmakizi imgesi önce (6.5)'teki formül uyarınca iki boyutlu Fourier dönüşümü ile uzaysal frekans domenine dönüştürülür.

$$f(x,y) = \frac{1}{MN}\sum_{u.=0}^{M-1}\sum_{v=0}^{N-1} F(u,v)\exp\{j2\pi(\frac{ux}{M}+\frac{vy}{N})\} \tag{6.5}$$

x = 0,1,2,...., M-1 için ve y = 0,1,2,....., N-1.

Frekans domeninde, parmak izi bilgisi çoğunlukla frekansın radyal bandında kapsanır ki bunun açısal boyutu baskın kabartı yönelimini izler. Parmak izinin ayırt edici karakteristikleri, kabartının baskın uzay frekansından az miktarda sapmış görünür. Bu, her biri için merkezdeki alçak uzay frekansı arka plan yoğunluğunu gösterirken uzay frekansın halka bölgesine yerleştirildiği anlamına gelir. Parmak izindeki her bir bölge frekans domeninin tamamına katkıda bulunduğu için spektrumun büyüklüğü parmağın dönmesine göre değişmez. Güç spektrumunun kullanımı yerine dönüşümün büyüklüğünün karesi alınarak elde edilen [44]' deki orjinal çalışmanın tersine, büyüklüğün dördüncü dereceden gücü daha kolay ayırt edilebilir özellikleri betimlemek için kullanılır. Halkanın özellikleri, sabit Δr kalınlığı ve çeşitli r yarıçapı değerlerinin dairesel halkada 4. dereceden gücünün tayfla ilgili değerlerinin toplamıyla biçimlendirilir. i inci halkanın dışındaki n tane halkanın özellikleri Denklem 6.6'da verilmiştir:

$$\lambda i(r) = \sum_{u}\sum_{v}\left|F(u,v)\right|^{4} \tag{6.6}$$

Burada r<(u+v)<r+r ve 0<r<N/2 iken maksimum imge boyutu N'dir. Benzer olarak delta ΔQ genişlik segmentleri ve Q yöneliminin çeşitli dağılımıyla dilim parçaları şu şekilde hesaplanır:

$$\beta_{j}(\Phi) = \sum_{u}\sum_{v}\left|F(u,v)\right|^{4} \tag{6.7}$$

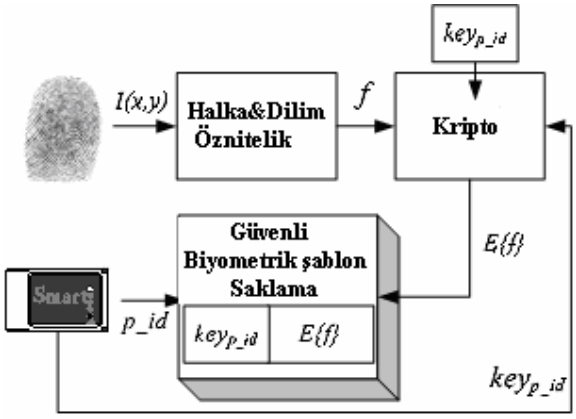
Burada Q<arctan(v/u)<Q+ ΔQ olarak alınırken *j* toplam m tane dilimin *j* inci dilimini işaret eder. Reel fonksiyonun güç tayfı simetrik olduğundan bu hesaplamalar sadece her bir spektrumun yarım düzlemi üzerinde yapılabilir.

6.3.3. Gerçekleştirilen Parmakizi Yetkilendirme Şeması

Yetkilendirme şemasının güvenliği Halka-Dilim parmakizi öznitelikleri kullanılarak geliştirilir. Gerçekte bu kalıp, kolaylıkla her tür biyometrik özniteliğe (yüz,iris, retine, vb..) uyarlanabilir. Ayrıca Halka-Dilim yöntemi yerine eğer sistem kaynakları daha karmaşık algoritmaları ele almak için yeterli ise daha gürbüz öznitelikler de uygulanabilir. Yine de Halka-Dilim özniteliklerini bir arada kapalı bir sistemde kullanmak, biyometrik, kriptoloji ve rasgele sayılar gibi popüler kavramların bütünleşmesini göstermede iyi bir başlangıçtır.

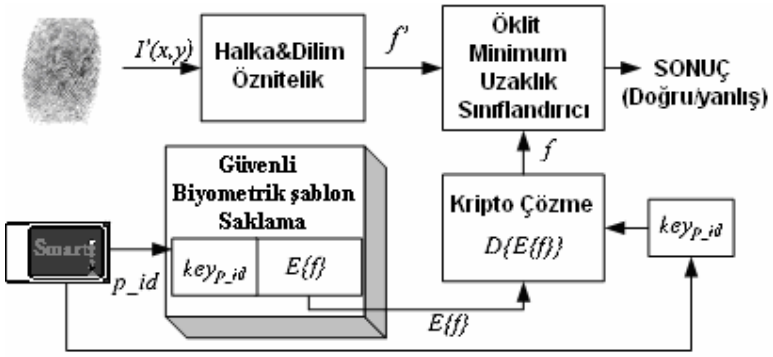
Sistemin tamamı tekil kimlik kartının (token), akıllı kartın ya da hafıza kartının (glossary) üzerinde depolanmış kişisel tanımlama numarası olan p_id gerektirir ve GRS'yi içeren özel anahtar, tasarlanan GRSÜ tarafından üretilir. Sadece bir tane şifre kullanılması tavsiye edilmez çünkü şifreler genellikle unutulur ya da kolay tahmin edilir.

Yetkilendirme sistemi Kayıt ve Doğrulama aşamalarından oluşturulur. Şekil 6.6'da gösterilen kayıt aşaması kişinin kendisini sisteme tanıttığı ve bilgilerini akıllı karta kaydettirdiği kısımdır. Şuraya dikkat çekilmelidir ki akıllı kart hem p_id hem de bireye ait özel bir anahtar kelime içermektedir. Bir kişinin parmak izi imgesi, *I(x,y)*, bir sensör yardımıyla alındıktan sonra halka-dilim özniteliği, *f*, hesaplanır. Kişiye özel bir bilgi olan *f*, raslantısal olarak sayı üreten key_{p_id} tarafından şifrelenir. Sonuç olarak şifreli özellik *E{f}* ve özel anahtar key_{p_id}, *p_id* ile uyuşan ve buna göre indekslenmiş bir veritabanı alanında depolanır. Burada *p_id*, veritabanında da dizin olarak kullanılan kişisel giriş PIN numarasıdır.



Şekil 6.6. Kayıt Aşaması

Şekil 6.7' de gösterilen doğrulama evresinde sorgusu yapılan parmakizi imgesi *I'(x,y)* elde edilir ve *f'* hesaplanır. Aynı zamanda şifreli öznitelik ile ilşkili *E{f}*, verilen *p_id* ile seçilir. Burada *p_id* parmak izi kalıbının veritabanındaki dizini olarak kabul edilir. Ardından, şifreli özniteliği tekrar elde etmek için, *D{E{f}}=f*, çözülür. Sonunda karar mekanizması *f* ile *f'* nün karşılaştırılmasına dayanır . Doğrulama, *f* ile *f'* arasındaki öklit uzaklığını karşılaştırarak tanımlama yapan Minimum Öklit Uzaklık Sınıflandırıcısına dayandırılır. Eğer doğrulama başarılı ise akıllı kartın üzerindeki key_{p_id}, tam güvenlik sağlamak için yeni özel anahtarıyla, GRSÜ tarafından düzeltilir.



Şekil 6.7. Doğrulama Aşaması

Performansı nesnel olarak ölçebilmek için UPEK TouchStrip TCS3 sensörü tarafından elde edilmiş imgelerin üzerinde karşılaştırma yapılmıştır. Bu silikon şeritli sensör özellikle el bilgisayarları, cep telefonları gibi taşınabilir aygıtlar için idealdir. Kullanıcılar, güvenilir kimlik denetimi, dijital ve fiziksel varlıklarının korunumu için

bu tip güvenlik önlemlerine sıcak bakmaktadırlar. Veritabanı 88 farklı parmaktan alınmış 440 adet 124x180 piksellik imgeden oluşmaktadır. Başarım karakteristiğinin elde edilebilmesi için gerçek-sahte sınıf karşılaştırmaları yapılmıştır. Gerçek sınıf karşılaştırması için her parmak örneği gene aynı kişiye ait diğer örneklerle (5x(5-1)/2 = 10 karşılaştırma) karşılaştırılır. Öte yandan sahte sınıf karşılaştırması için ise her parmağın ilk örnek imgesi diğer parmakların ilk örnekleriyle toplam 88x(88-1)=3828 test ile karşılaştırılır. Burada dikkat edilecek bir nokta *f'* nin çıkarılması sırasında sistemin doğruluğunun dilim ve halka sayısına bağlı olmasıdır. Önerilen katıştırılmış sistem, zaman alan şifreleme ve ön işleme süreçleri içerdiği için, deneysel olarak halka ve dilim sayısı sırasıyla 12 ve 10 olarak seçilmiştir. Dahası aynı parmaktan elde edilen birden fazla öznitelik dizisinin aritmetik ortalaması, daha iyi sonuç alınabilmesi amacıyla her kişi için deneme sürecinde hesaplanmıştır. Son olarak 440 geliştirilmiş parmak izinden % 8.12 lik eşit hata oranı elde edilmiştir, fakat daha çok parmak iziyle daha doğru bir başarım testi yapılabilir. Bu küme için doğrulama fazının ortalama süresi 1 saniyeden daha azdır. Şifreleme arka planı için Advance Encryption Standard (AES) kullanılmıştır ancak sistem başka bir şifreleme standardıyla değiştirilebilecek modülerliğe sahiptir.

7. BÖLÜM

SONUÇLAR VE ÖNERİLER

Bu tez çalışmasında, donanımsal olarak Gerçek Rasgele Sayı üretip test eden ve aynı zamanda yeni RSÜ tasarımları için gerçek zamanlı istatistiksel test donanımı olarak da kullanılan FPGA tabanlı, PCI ara yüzlü bir veri toplama sistemi sunulmuştur. PCI ara yüzüne sahip olması, önerilen tasarımı bilgisayar tabanlı şifreleme uygulamaları için ideal hale sokmuştur. Donanımda, deterministik olmayan yolla üretilen rasgele sayılara öncelikle beş tane donanımsal olarak gerçeklenen istatistiksel testler (FIPS 140-2 ve Seri test) uygulanmış ve daha sonra PCI arabağlaşımı sayesinde PC ortamında yazılımla gerçekleştirilen tüm NIST-800-22 test suiti de uygulanarak üretilen sayıların gerçek rasgeleliği doğrulanmıştır. Gerçek zamanlı RSÜ test düzeneği, üzerindeki LED göstergeler vasıtasıyla parametre değişikliklerinin istatistiksel test sonuçlarına etkilerini aynı anda değerlendirme imkanı tanımış ve yeni RSÜ tasarımının prototiplendirme zamanlarının kısaltılmasını sağlamıştır. Sonuçta, donanımsal olarak gerçeklenen GRSÜ'nin çıkış veri hızı ve gerçek zamanlı istatistiksel test etme hızı etkin olarak sırasıyla 128 Kbps ve 31.5 Mbps olarak belirlenmiştir. Önerilen tasarımın en yüksek veri toplama hızını tespit etmek amacıyla FPGA içinde farklı hızlarda çalışan sayaçlar gerçeklenmiş ve RSÜ çıkışı yerine bu sayaçların çıkışı örneklenerek tasarımın 100 Mbps' e kadar hatasız bir şekilde veri toplayabildiği deneysel olarak doğrulanmıştır. Bu hız literatürde bulunan RSÜ'lerin en fazla 40 Mbps [1,36] hızında çıkış sağladığı düşünüldüğünde yeterli seviyededir.

Önerilen tasarım ile diğer bazı klasik RSÜ uygulamaları arasındaki temel farklar Tablo 7.1'de verilmiştir. Intel yaklaşımı [6], jitter gürültüsünün kullanıldığı çift osilatör yapısı [48], ayrık [49] ve sürekli [16] zamanlı kaos temeline dayanan tasarımlar karşılaştırmalı olarak sunulmuştur. Tez çalışmasında önerilen GRSÜ ve tabloda verilen ilk iki tasarım gürültü fiziksel niceliklerin gözlenmesi üzerine kurulmuşken, diğer yapılar deterministik kaos temeline dayanmaktadır. Kaotik sistemler, başlangıç değerlerine aşırı duyarlı olmaları ve pozitif lyapunov üsteli vasıtasıyla periyodik olmayan, düzensiz ve

uzun-vadede tahmin edilemez işaretler üretmektedirler. Bu temel özellikleri sayesinde yalancı rastgele sayı üretme amacıyla kullanılsalar da, kaotik sistemlerin [46] ve [47]' de gösterildiği üzere kısa-vadede tahmin edilebilir olması şifreleme uygulamalarında gerçek rastgele sayı üretme amacıyla kullanılmalarını uygunsuz hale getirmektedir.

Verilen tabloda görüldüğü üzere, tezde önerilen yapı dışında NIST-800-22 istatistiksel test paketinden geçen tek tasarım FPAA (Field Programmble Analog Array) üzerinde gerçeklenmiş olan ve ayrık zamanlı kaos temeline dayanan [49] RSÜ'dür. Bu tasarımın entropi kaynağı olarak deterministik kaosu kullandığı düşünülürse, tezde önerilen GRSÜ teorik anlamda güvenlik açısından diğer yaklaşımlara göre büyük bir üstünlük sağlamıştır. Tasarlanan sistemin ayrık zamanlı kaosu temel alan RSÜ yapısına [49] göre dezavantajı, çıkış bit hızının düşük olmasıdır. Bu dezavantaj, önerilen GRSÜ'nün kişisel bilgisayara bağlanmasını sağlayan ve diğer tasarımlarda bulunmayan PCI arayüzüne sahip olması ve üretilen rastgele sayıların gerçek zamanlı testte tabi tutularak büyük bir bellek alanında depolanması ile giderilmiştir.

Tablo 7.1 Farklı RSÜ Uygulamaları Arasındaki Kıyaslamalar

Tasarımcı	**INTEL (1999)**	**Fischer (2002)**	**Yalcin (2004)**	**Callegari (2003)**	**Tez Çalışması (2008)**
Gürültü Kaynağı	Isısal Gürültü tabanlı	Jitter Gürültü tabanlı	Kaos tabanlı	Kaos tabanlı	Yarı iletken diyot gürültüsü
Entropi Kaynağı	Mikro kosmik (Isıl gürültü)	Mikro kosmik (Jitter gürültüsü)	Sürekli Zamanlı Kaos	Ayrık Zamanlı Kaos	PN jonksiyon Gürültüsü
Çıkış Bit Hızı	75 Kbit/s	70 Kbit/s	?	5 Mbit/s	128 Kbps
Rasgelelik Testleri	Yok	Yok	Var	Var	Var
NIST 140-2, 800-22	Sağlamıyor	Sağlamıyor	Sağlıyor, Sağlamıyor	Sağlıyor	Sağlıyor
Geliştirme Süresi	Çok Fazla	Az	Çok	Çok az	Çok çok az
Veri Toplama Arayüzü/ Hız	Yok/?	Yok/?	Paralel Port / 115 KB/s	?/?	PCI/100Mbps

Tabloda verilen ilk iki RSÜ, entegre devre olarak tümleştirilmiş tasarımlardır. Bu iki RSÜ geniş ölçekte üretim söz konusu olduğunda daha avantajlı iken hız ve istatistiksel testler açısından daha dezavantajlılardır. Küçük ölçekli bir üretim için, geliştirme süresi ve maliyet azlığı dikkate alındığında önerilen sistemde bulunan FPGA gibi alan programlama cihazlarının kullanılması daha mantıklı hale gelir ki, bu durumda FPGA'in sağlandığı esneklik ilave bir avantajdır. Programlanabilir bir cihaz, değişik tür GRSÜ'ler barındırabilir ve saniyeler mertebesinde çok kısa bir zamanda çıkış bitlerinin istatistiksel kalitesine göre herhangi bir üreteç tercih edebilir. Ayrıca, tez çalışmasında önerilen yapıda olduğu şekilde, alan programlama cihazları içeren tasarımlar, son anda ince parametre ayarı yapılmasına izin verir ve böylece daha gelişmiş GRSÜ tasarımları ortaya çıkar.

Tez çalışmasında önerilen sistemin veri toplama fonksiyonunu mevcut sistemlerle karşılaştırmak amacıyla Tablo 7.2'de veri yolu arayüzleri ve bunlara ait maksimum veri iletme hızları verilmiştir. Verilen tabloda görüldüğü üzere, veri toplama hızı açısından en avantajlı arayüz, tez çalışmasında da uygulanmış olan PCI veri yoludur. [16]'de önerilen RSÜ'de açık savaklı (Open Drain) karşılaştırıcı çıkışları paralel porttan örneklenerek çıkış bit dizisi oluşturulmuştur. Bu tasarımda, karşılaştırıcı çıkışlarının düşme ve yükselme eğimleri simetrik olmamakla birlikte kullanılan pull-up direncinin değerine bağlı olarak değişecek ve üretilen bit dizilerinin 0-1 dengesi bozulacaktır. Tez çalışmasında tasarlanan yapıda ise FPGA tarafından örneklenip 8 bitlik diziler haline getirilen veriler bozulmadan toplanabilecektir. Ayrıca PCI arayüzü vasıtasıyla ilave kabloya ihtiyaç duyulmadan veri toplanabiliyor olması tez çalışmasında önerilen tasarımı gürültü bağışıklığı açısından ve sistemi bozma amaçlı saldırılara karşı daha güvenli hale getirmiştir.

Tablo 7.2 Farklı Veri Arayüzü Hızları

Veri Yolu	**Bant Genişliği**
Seri Port	115 Kbit/s
Paralel Port	115 KB/s
ECP/EPP Paralel Port	3 MB/s
USB 1.0	1.5 MB/s
USB2.0	60 MB/s
16 Bit ISA	16 MB/s
PCI	132 MB/s

Sonuç olarak, bildiğimiz kadarıyla bu çalışma, gerçek rasgele kaynakların FPGA uygulaması konusunda bir ilktir. Ayrıca önerimiz RSÜ kaynaklarının tasarımı için de bir test ortamıdır. Literatürde, çok sayıda farklı görev için karmaşık dinamikten faydalanma ile ilgili oldukça fazla sayıda öneri vardır. Bunların kabulleri, ilgili analog devrelerin gerçeklenmesindeki güçlükler yüzünden yavaşlamıştır. Burada gösterildiği gibi ucuz alan programlama cihazları ile kaotik dinamik elde etme olasılığı, kaos-tabanlı tekniklerin denenmesi ve daha geniş anlamda kabul edilmesi için bir teşvik olabilir.

Tasarlanan FPGA tabanlı donanımın pratik uygulaması olarak iki tane biyometrik kimlik doğrulama sistemi örnek olarak sunulmuştur. İlk uygulamada, kodlanmış yüz şablonlarının güvenli bir biçimde ve herhangi bir aygıt üzerine yüklenebilecek şekilde indekslenerek veritabanında tutulduğu WFMT tabanlı bir yüz doğrulama sistemi önerilmiştir. Bu uygulamanın en önemli özelliği sistemin kullandığı özel anahtarların FPGA-tabanlı GRSÜ tarafından donanımsal olarak üretilmesidir. Sisteme yapılan saldırılarda özel anahtarlar öğrenilmedikçe kodlanmış biyometrik şablonlara erişilemeyecektir. Bu çalışmada RST'ye göre değişmezlik karakteristiğinden dolayı yüz resimlerinin WFMT tabanlı temsili kullanılmış olmakla birlikte parmakizi detayları, iris, ses özellikleri gibi biyometrik özelliklerde kullanılarak daha da iyileştirme yapılabilir. Aynı şekilde AES tabanlı şifreleme yerine eliptik eğri şifreleme veya benzeri güçlü bir yöntemle güvenlik tarafıda güçlendirilebilir. İkinci uygulamada ise şifrelenmiş parmak izi şablonlarının güvenli bir erişim aygıtına (smart veya glossary kart,jeton,vs.) indeks numarasıyla yüklendiği ve bir veri tabanında saklandığı parmak iziyle kimlik denetimi sistemini önerilmiştir. Bu uygulamanın en önemli özelliği yeni nesil donanım tabanlı GRSÜ' ler ve hızlı bir parmak izi tanıma algoritmasını bütünleştirmeyi öneren gömülü biyometrik kimlik denetimi sistemi olmasıdır. Güvenlik öncelikle daha az güvenli olan YRSÜ' lerin yerine GRSÜ' lerin kullanımıyla, ikinci olarak her başarılı girişte dinamik olarak değişen şifrelerle ve son olarak ta her kullanıcının parmak izinin kullanımıyla üst düzeyde geliştirilmiştir. Ortaya çıkan sistem kolaylıkla bir bilgisayara takılabildiği gibi PCI arabağlaşımı yardımıyla gerçek rasgele anahtar üretmek için de kullanılabilir. Sistemi kırmayı amaçlayan saldırgan, bireysel şifreyi öğrense bile, bütün veri tabanını ele geçirmediği sürece kişinin şifrelenmiş biyometrik şablonunu oluşturamaz. Bu uygulamada daha az hesap karmaşıklığı ve düşük kaliteli parmak izlerinde gösterdiği yüksek performanstan dolayı halka-dilim tabanlı parmak izi öznitelikleri kullanılmıştır. Belki de bir sonraki çalışma olarak çeşitli parmak izi öznitelikleri, çeşitli biyometrikler

veya biyometrik birimlerin bütünleştirilmesi kullanılabilir. Deneysel sonuçlar tasarlanan FPGA tabanlı devrenin gerçeklenebilirliğini doğrulamakla kalmamış aynı zamanda onun YRSÜ' leri yerine yüksek- başarımlı GRSÜ olarak ta kullanımını teşvik etmiştir.

Tez sonrasında yapılabilecek çalışmalar: NIST800-22 test süitinde verilen 16 testin ilave edilmesiyle daha güvenilir rasgele sayı üretiminin ve testlerinin gerçek zamanda sağlanması; Test devrelerinin tümleştirilmesi; PCIe standardı kullanılarak yüksek veri toplama hızlarına ulaşılması; Biyometrik güvenlik uygulamalarında farklı özniteliklerin ve karşılaştırma algoritmalarının değerlendirilmesi ve mevcut biyometrik güvenlik uygulamalarında daha geniş veritabanları üzerinde testler koşturulması şeklinde sıralanabilir.

KAYNAKLAR

1. Bucci, M., Germani, L., Luzzi, R., Tommasino, P., Trifiletti, A., Varanonuovo M., A High-Speed IC Random-Number Source for SmartCard Microcontrollers. IEEE Trans. Cir. Sys.-1, Vol. 50. No. 11 pp. 1373 - 1380, 2003.

2. Menezes, A.J., Van Oorschot, P.C, Vanstone, S.A, Handbook of Applied Cryptology. CRC Press 1996.

3. Vijaya Kittu, M., Hardware Random Number Generators. IT June (67-70), 2003.

4. Schneier, B., Applied Cryptography. 2*nd* edn. John Wiley & Sons 1996.

5. National Institute of Standard and Technology, Security Requirements for Cryptographic Modules, NIST, Boulder CO, Jan. 1994.

6. Jun, B., Kocher, P., The Intel Random Number Generator. Cryptography Research, Inc. white paper prepared for Inter Corp 1999.

7. Bagini, V., Bucci, M., A Design of Reliable True Random Number Generator for Cryptographic Applications. Proc. Workshop Cryptographic Hardware and Embedded Systems (CHES). 204-218, 1999.

8. Holman, W.T., Connelly, J.A., Downlatabadi, A.B., An Integrated Analog-Digital Random Noise Source. IEEE Trans. Circuits & Systems I, Vol. 44(6). 521-528, 1997.

9. Petrie, C.S., Connelly, J.A., Modeling and Simulation of Oscillator-Based Random Number Generators. Proc. IEEE Int. Symp. on Circuits & Systems (ISCAS), Vol.4. 324-327, 1996.

10. Dichtl, M., Janssen, N., A High Quality Physical Random Number Generator. Proc. Sophia Antipolis Forum Microelectronics (SAME). 48-53, 2000.

11. Schellekens, D., Preneel, B., Verbauwhede, I., FPGA Vendor Agnostic True Random Number Generator. 16th International Conference on Field Field Programmable Logic and Applications, 2006 (IEEE, FPL 2006).

12. Zhun, H., Hongyi, C., A Truly Number Generator Based on Thermal Noise. IEEE ASICON, October 2001.

13. Petrie, C.S., Connelly, J.A., A Noise-Based IC Random Number Generator for Applications in Cryptography. IEEE Trans. Circuits & Systems I, Vol. 47(5) 615-621, 2000.

14. Stojanovski, T., Kocarev, L., Chaos-Based Random Number Generators-Part I:Analysis. IEEE Trans. Circuits & Systems I, Vol. 48, 3. 281-288, 2001.

15. Delgado-Restituto, M., Medeiro, F., Rodriguez-Vazquez, A., Nonlinear Switched current CMOS IC for Random Signal Generation. Electronics Letters, Vol. 29(25). 2190-2191, 1993.

16. Yalcin, M.E., Suykens, J.A.K., Vandewalle, J., True Random Bit Generation from a Double Scroll Attractor. IEEE Trans. on Circuits & Systems I: Fundamental Theory and Applications, Vol. 51(7). 1395-1404, 2004.

17. Von Neumann, J., Various Techniques Used in Connection With Random Digits.Applied Math Series - Notes by G.E. Forsythe, In National Bureau of Standards, Vol. 12. 36-38, 1951.

18. Stojanovski, T., Pihl, J., Kocarev, L., Chaos-Based Random Number Generators-Part II: Practical Realization. IEEE Trans. Circuits and Systems I, Vol. 48, 3, 382-385, 2001.

19. Ergün, S., Özoğuz, S. "Truly random number generators based on a non-autonomous chaotic oscillator" 2006 Elsevier GmbH. AEÜ 10.1016/j.aeue. 05.006, 2006.

20. Elwakil, A. S., Salama, K. N., and Kennedy, M. P., "An equation for generating chaos and its monolithic implementation," Int. J. Bifurcation Chaos, vol. 12, no. 12, pp. 2885-2896, 2002.

21. National Institute of Standard and Technology., A Statistical Test Suite for Random and Pseudo Random Number Generators for Cryptographic Applications. NIST 800-22, http://csrc.nist.gov/rng/SP800-22b.pdf, 2001.

22. National Institute of Standard and Technology, FIPS PUB 140-2, Security Requirements for Cryptographic Modules, NIST, Gaithersburg, MD 20899, 2001.

23. Pitman,P., Probability., Springer-Verlag, pp.93-108, New York 1993.

24. Donald E. Knuth, The Art of Computer Programming. Vol 2: Seminumerical Algorithms. 3rd ed. Reading, Mass: Addison-Wesley, pp. 42-47, 1998.

25. Gibbons, J.D., Nonparametric Statistical Inference, 2nd ed.: Marcel Dekker, pp. 50-58. New York, 1985.

26. Revesz, P., Random Walk in Random and Non-Random Environments. World Scientific, Singapore,1990.

27. Diehard, G.M., a battery of tests of randomness. http://stat.fsu.edu/~geo/diehard.html.

28. Bracewell, R. N., The Fourier Transform and Its Applications., McGraw-Hill, NY,1986.

29. Johnson,N. J., S. Kotz, and Kemp, A., Discrete Distributions. Jonh Wiley, 2nd ed. pp. 378-379. NY, 1996.

30. Maurer, U. M., A Universal Statistical Test for Random Bit Generators, Journal of Cryptology, Vol. 5, No. 2, pp. 89-105, 1992.

31. Rukhin, A., Approximate entropy for testing randomness, Journal of Applied Probability. Vol. 37, 2000.

32. Shanley, T., Anderson, D., PCI System Architecture 3th edition, Addison Wesley, US,1997.

33. Nur, T., Tekrar Düzenlenebilir İşlem Kartı Tasarımı Yüksek Lisans Tezi, İTÜ, 2000.

34. Xilinx Customer Education Programme "Designing a PCI System" Course Notes, 1999.

35. PCI-SIG., PCI Special Interesy Group Home Page, http://www.pcisig.com.

36. Delgado-Restituto, M., Rodriguez-Vazquez, A., Integrated Chaos Generators. Proc. of IEEE, Vol. 90(5). 747-767, 2002.

37. Özoğuz, S., Ateş, Ö O., Elwakil, A.S., An integrated circuit chaotic oscillator and its application for high speed random bit generation. Proceeding of the International Symposium on Circuit and Systems (ISCAS). 4345-4348, 2005.

38. Erat, M., Danışman, K.,Ergün, S., Kanak, A., A Hardware-Implemented Truly Random Key Generator for Secure Biometric Authentication Systems, LNCS 4105, pp. 128-135, Springer Verlag, Berlin, 2006.

39. Erat, M., Danışman, K.,Ergün, S., Kanak, A., Kayaoğlu, M., An Embedded Fingerprint Authentication System Integrated With a Hardware-based Truly Random Number Generator, LNCS 4673, pp..366-373, Springer Verlag , Berlin, 2007.

40. Teoh, A.B.J, Ngo, D.C.L, Goh, A., Personalised Cryptographic Key Generation Based on FaceHashing. Jour. of Computer & Security , 2004.

41. Samaria, F. and Harter, A., Parameterisation of a Stochastic Model for Human Face Identification. 2nd IEEEWorkshop on Applications of Computer Vision, Sarasota FL, December, 1994.

42. Bucci, M., Germani, L., Luzzi, R., Trifiletti, A., Varanonuovo, M., A High Speed Oscillator-based Truly Random Number Source for Cryptographic Applications on a SmartCard IC. IEEE Trans. Comput., Vol. 52. 403-409, 2003.

43. Maio, D., Maltoni, D., Jain, A.K., and Prabhakar, S., Handbook of Fingerprint Recognition. Springer Verlag, 2003.

44. Willis, A.J. and Myers, L., A Cost-efective Fingerprint Recognition system for Use with Low-quality Prints and Damaged Fingertips. Jour. Pattern Recogn., Vol. 34. 255-270, 2001.

45. Chikkerur, S., Cartwright, A. N., and Govindaraju, V., Fingerprint Enhancement Using STFT Analysis. Jour. Pattern Recogn., Vol.40(1). 198-211, 2007.

46. Casdagli, M., Nonlinear prediction of chaotic time series, Phys. D, vol. 35, pp. 335–356, 1989.

47. Doyne, J. and . Sidorowich, J. J., Predicting chaotic time series, Phys. Rev. Lett., vol. 59, no. 8, pp. 845–848, 1987.

48. Fischer, V., Drutarovsky, M., True random number generator embedded in reconfigurable hardware. In Cryptographic Hardware and Embedded Systems CHES 2002.

49. Callegari, S., Rovatti, R., Setti, G., First direct implementation of a true random source on programmable hardware, Int. J. Circ. Theor. Appl., 33:1–16, 2005.

ÖZGEÇMİŞ

Murat ERAT

1963 yılında Terme'de doğdu. 1981 yılında Ünye Lisesi'nden mezun oldu. 1987 yılında Erciyes Ün. Müh. Fak. Elektronik Mühendisliği bölümünden mezun oldu. 1992 yılında aynı üniversitede yüksek lisans yaptı. 1987 yılında TUBİTAK-MAM Elektronik Bölümünde çalışma hayatına başladı. Asker dönüşü TÜBİTAK-UEKAE de tekrar işe başladı. Bir çok askeri ve ticari projelerde araştırmacı, proje yöneticisi yardımcısı ve proje yönetimi sorumlusu olarak görev aldı. ARM, DSP, FPGA, uC ve PC tabanlı sistemler tasarladı. Halen Güvenli Ses ve Veri iletimi konularında TUBİTAK-UEKAE' de çalışmaktadır.

Adres:
TUBİTAK-UEKAE
GEBZE
Tel:0262 648 13 69
Email:erat@uekae.tubitak.gov.tr