T.C. İSTANBUL TİCARET ÜNİVERSİTESİ

FEN BİLİMLERİ ENSTİTÜSÜ

VERİLOG İLE TAUSWORTHE DENKLEMİNE DAYANAN YENİ BİR RASTGELE SAYI ÜRETECİ TASARIMI

Minare HASANBEYLİ

Danışman
Dr. Öğr. Üyesi Vedat TAVAS

YÜKSEK LİSANS TEZİ
ELEKTRONİK VE HABERLEŞME MÜHENDİSLİĞİ ANABİLİM DALI
İSTANBUL - 2021

KABUL VE ONAY SAYFASI

Minare HASANBEYLİ tarafından hazırlanan **"Verilog ile Tausworthe Denklemine Dayanan Yeni Bir Rastgele Sayı Üreteci Tasarımı"** adlı tez çalışması 07/07/2021 tarihinde aşağıdaki jüri üyeleri önünde başarı ile savunularak, İstanbul Ticaret Üniversitesi Fen Bilimleri Enstitüsü **Elektronik ve Haberleşme Mühendisliği Anabilim Dalı**'nda **YÜKSEK LİSANS TEZİ** olarak kabul edilmiştir.

Danışman	**Dr.Öğr.Üyesi Vedat TAVAS** İstanbul Ticaret Üniversitesi	.....
Jüri Üyesi	**Dr.Öğr.Üyesi M.Alper ÖZPINAR** İstanbul Ticaret Üniversitesi	.....
Jüri Üyesi	**Doç.Dr. Indrit MYDERRIZI** İstanbul Gelişim Üniversitesi	.....

Onay Tarihi : 29.07.2021

İstanbul Ticaret Üniversitesi, Fen Bilimleri Enstitüsünün 29.07.2021 tarih ve 2021/317 numaralı Yönetim Kurulu Kararının 2. maddesi gereğince, ders yüklerini ve tez yükümlülüğünü yerine getirdiği belirlenen "Minare HASANBEYLİ" adlı öğrencinin mezun olmasına oy birliği ile karar verilmiştir.

Prof. Dr. Necip ŞİMŞEK
Enstitü Müdürü

AKADEMİK VE ETİK KURALLARA UYGUNLUK BEYANI

İstanbul Ticaret Üniversitesi, Fen Bilimleri Enstitüsü, tez yazım kurallarına uygun olarak hazırladığım bu tez çalışmasında,

- tez içindeki bütün bilgi ve belgeleri akademik kurallar çerçevesinde elde ettiğimi,
- görsel, işitsel ve yazılı tüm bilgi ve sonuçları bilimsel ahlak kurallarına uygun olarak sunduğumu,
- başkalarının eserlerinden yararlanılması durumunda ilgili eserlere bilimsel normlara uygun olarak atıfta bulunduğumu,
- atıfta bulunduğum eserlerin tümünü kaynak olarak gösterdiğimi,
- kullanılan verilerde herhangi bir tahrifat yapmadığımı,
- ve bu tezin herhangi bir bölümünü bu üniversitede veya başka bir üniversitede başka bir tez çalışması olarak sunmadığımı

beyan ederim.

07/07/2021

Minare HASANBEYLİ

İÇİNDEKİLER

ÖZET

Yüksek Lisans Tezi

VERİLOG İLE TAUSWORTHE DENKLEMİNE DAYANAN YENİ BİR RASTGELE SAYI ÜRETECİ TASARIMI

Minare HASANBEYLİ

İstanbul Ticaret Üniversitesi
Fen Bilimleri Enstitüsü
Elektronik ve Haberleşme Mühendisliği Anabilim Dalı

Danışman: Dr. Öğr. Üyesi Vedat TAVAS
2021, 70 sayfa

Rastgele sayılar şifreleme, bilgisayar benzetimi, rastgele tasarım gibi birçok alanda kullanılmaktadır. Rastgele sayılar herhangi bir öngörülebilirlik içermeyen rastgele süreçlerden elde edilir. Rastgeleliğin yetersizliği tüm sistemin güvenliğini etkileye bilir. Bu yüzden rastgele sayıların tahmin edilememesi gerekir. Bu çalışmada ayrık zamanlı rastgele sayı üreteci tasarlanmış ve FPGA ile gerçeklenmiştir. Alanda Programlanabilir Kapı dizisi tasarımcının ihtiyaç duyduğu mantık işlevlerini gerçekleştirme maksatlı üretilmiştir. Bu yöntemle üretilen bit dizilerinin rastgeleliğini belirlemek için FIPS test kümesi kullanılmış ve diziler bu testlerden başarıyla geçmiştir. Tasarım Xilinx yazılımı kullanılarak Verilog programlama dili ile yapılmıştır.

Anahtar Kelimeler: Alanda programlanabilir kapı dizileri, geribeslemeli kaydırmalı yazmaç, rastgele sayı üreteci, Tausworthe yöntemi, Verilog.

ABSTRACT

M.Sc. Thesis

A NEW RANDOM NUMBER GENERATOR DESIGN BASED ON TAUSWORTHE EQUATION WITH VERILOG

Minare HASANBEYLİ

Istanbul Commerce University
Graduate School of Applied and Natural Sciences
Department of Electronics and Communications Engineering

Supervisor: Assist. Prof. Dr. Vedat TAVAS
2021, 70 pages

Random numbers are used in many fields such as encryption, computer simulation, random design. Random numbers are derived from random processes that cannot contain any predictability. The inadequacy of randomness may affect the security of the entire system. Therefore, the random numbers should not be predicted. In this study, a discrete time random number generator was designed and implemented with FPGA. Field Programmable Gate array (FPGA) has been produced to perform the logic functions required by the designer. FIPS test set was used to determine the randomness of the bit strings produced by this method, and the strings passed these tests successfully. The design was made with Verilog programming language using Xilinx software.

Keywords: Feedback shift register, field programmable gate array (FPGA), random number generator (RNG), Tausworthe method , Verilog.

TEŞEKKÜR

Yüksek lisans tezim boyunca değerli bilgi ve deneyimlerinden yararlandığım, her konuda bilgi ve desteğini aldığım, araştırmanın planlanmasından yazılmasına kadar tüm aşamalarında yardımlarını esirgemeyen değerli danışman hocam Dr. Öğr. Üyesi Vedat TAVAS' a teşekkürlerimi sunarım**.**

Yüksek lisans tezim boyunca bana sürekli yardımcı olan, karşılaştığım zorluklarda destek vererek tüm bilgisini benimle paylaşan, katkılarıyla ve yönlendirmeleriyle çabasını esirgemeyen değerli arkadaşım Elmir MUSAZADE' ye teşekkürlerimi sunarım.

Araştırmanın kodlama kısmında sağladığı yardımlardan dolayı Burak ACAR'a teşekkürlerimi sunarım.

Son olarak bana destek olan tüm arkadaşlarıma ve aileme teşekkürlerimi sunarım.

Minare HASANBEYLİ
İSTANBUL, 2021

ŞEKİLLER

ÇİZELGELER

Sayfa

SİMGELER VE KISALTMALAR

ANSSI	National Agency for the Security of Information Systems (Ulusal Bilgi Sistemleri Güvenliği Ajansı)
ASIC	Application Specific Integrated Circuit (Uygulamaya Özel Tümleşik Devre)
AWGN	Additive white Gaussian noise (Toplamsal Beyaz Gauus Gürültüsü)
BSI	British Standards Institution (İngiliz Standartları Enstitüsü)
CLB	Configurable Logic Block (Yapılandırılabilir Mantık Blokları)
CPU	Central Processing Unit (Merkezi İşlem Birimi)
DSP	Digital Signal Processing (Sayısal Sinyal İşleme)
EPROM	Erasable Programmable Read Only Memory (Silinebilir Programlanabilir Salt Okunur Bellek)
FIPS	Federal Information Processing Standards (Federal Bilgi İşleme Standartları)
FPGA	Field Programmable Gate Array (Alanda Programlanabilir Kapı Dizileri)
GFSR	Global Financial Stability Report (Küresel Finansal İstikrar Raporu)
GPU	Graphics Processing Unit (Grafik İşleme Birimi)
GRSÜ	Gerçek Rastgele Sayı Üreteci
HD	High Definition (Yüksek Çözünürlük)
HDL	Hardware Description Language (Donanım Tanımlama Dili)
IEEE	Institute of Electrical and Electronics Engineers (Elektrik ve Elektronik Mühendisleri Enstitüsü)
JTAG	Joint Test Action Group (Ortak Test Eylem Grubu)
LCG	Linear Congruential Generator (Doğrusal bir Eşleşik Üretici)
LFSR	Linear-Feedback Shift Regis (Doğrusal Geribeslemeli Kaydırmalı Yazmaç)
LUT	LookUp Table (Arama Tablosu)
NIST	National Institute of Standards and Technology (Ulusal Standartlar ve Teknoloji Enstitüsü)
PC	Personal Computer (Kişisel Bilgisayar)
PRNG	Pseudo Random Number Generator (Sözde Rastgele Sayı Üreteci)
PROM	Programmable Read-Only Memory (Programlanabilir Salt Okunur Bellek)
RAM	Random Access Memory (Rastgele Erişimli Bellek)
RNG	Random Number Generator (Rastgele Sayı Üreteci)
SRAM	Static Random Access Memory (Statik Rastgele Erişimli Bellek)
SRSÜ	Sözde Rastgele Sayı Üreteci)
TRNG	True Random Number Generator (Gerçek Rastgele Sayı Üreteci)
VHDL	VHSIC Hardware Description Language (VHSIC Donanım Tanımlama Dili)
XOR	Exclusive Or

1. GİRİŞ

Rastgele Sayı Üreteçlerinin (RNG - Random Number Generator) günümüzdeki önemi çok fazladır (Pınar, 2006). Rastgele sayılar şifreleme (Özkaynak vd., 2015), istatistiksel örnekleme (Robinson ve Dessart, 1998), tamamen rastgele tasarım (Elbaşı ve Eskicioğlu, 2006), bilgisayar simülasyonu (Schoukens vd., 1988) gibi öngörülemeyen rastgele sayıların istendiği alanlarda önemli uygulamalara sahiptir. Rastgele sayılar bir rastgele sayı üreteci tarafından üretilir. Rastgele sayı oluşturmanın birçok farklı yolu vardır (Selman, 2019). Rastgele sayıların en önemli özelliği sayılar arasında hiçbir ilişki kurulamamasıdır (Özdemir, 2008). Böyle olması ardışık sayılar arasında hiçbir ilişki kurulmamasına neden olur (Selman, 2019).

Rastgele sayılar gerçek ve sözde rastgele sayılar olmakla ikiye ayrılır (ICYSCIENCE, 2021). Bunlar da kendi aralarında zayıf, güçlü, hibrid, basit ve güvenilir rastgele sayılara ayrılır (Slepovichev, 2017). Rastgele sayılar birçok kriptografik uygulamanın temelini oluşturur (Özkaynak, 2015). İyi bir şifreleme iyi bir RSÜ gerektirir. Rastgele sayı üreteçleri, birçok farklı matematiksel yöntem kullanmaktadır. Rastgele Sayı Üreteçlerin bir olasılık dağılımını takip ettiği söylenebilir (Özdemir, 2008). Bunlara genel olarak doğrusal eşzamanlı oluşturucu, orta kare yöntemi, olasılık dağılımına dayanan ters çevirme, kabul-ret yöntemi gibi örnekler verilebilir (Başlıgil, 2010; Atay, 2016). Bu yöntemlere bağlı olarak Mersenne Twister, Monte Carlo, Tausworth' e gibi farklı algoritmalar kullanılmaktadır. (L'ecuyer, 2017).

Güvenlik gereksinimleri için, neredeyse her zaman yüksek kaliteli bir rastgele sayı oluşturma sürecine ihtiyaç vardır (Özkaynak, 2015). Rastgele sayıların üretim sürecindeki kalite eksikliği, genellikle sistem saldırıya uğradığında güvenlik açığına yol açar ve bu da kriptografik sistemin güvenilir olmamasına veya hatta tamamen güvensiz olmasına yol açar (Özdemir, 2008).

RSÜ sürecini tahmin edebileceği sözde rastgele bir sayı ile değiştirebilirse, güvenlik tamamen tehlikeye atılır (L'ecuyer, 2017). Daha güvenli, kaliteli RSÜ

tasarlamak için daha karmaşık yöntemlere ihtiyaç duyulmaktadır. Ayzı zamanda kullanılan basit yöntemleri karıştırarak hibrit RSÜ tasarımları da gerçeklene bilir. Bu çalışmada da Tausworthe yöntemini kullanan çoklu LFSR yapısı tasarlanmıştır (L'ecuyer, 2017).

2. LİTERATÜR ÖZETİ

Bu bölümde rastgele sayı üreteçleri, programlama dilleri ve alanda programlanabilir kapı dizileri ile ilgili teorik bilgilere yer verilmiştir. Bundan amaç çalışmada kullanılan yöntemi daha iyi kavrayabilmek adına gerekli olan bilgilere yer vermektir.

2.1 Rastgele Sayı Üretecleri

Rastgele Sayı Üreteçlerinin günümüzdeki önemi çok fazladır (Pınar, 2006). Rastgele sayılar şifreleme (Özkaynak vd., 2015), istatistiksel örnekleme (Robinson ve Dessart, 1998), tamamen rastgele tasarım (Elbaşı ve Eskicioğlu, 2006), bilgisayar simülasyonu (Schoukens vd., 1988) gibi öngörülemeyen rastgele sayıların istendiği alanlarda önemli uygulamalara sahiptir. Rastgele sayılar bir rastgele sayı üreteci tarafından üretilir. Rastgele sayı oluşturmanın birçok farklı yolu vardır (Selman, 2019). Rastgele sayıların en önemli özelliği sayılar arasında hiçbir ilişki kurulamamasıdır. Böyle olması ardışık sayılar arasında hiçbir ilişki kurulmamasına neden olur (Selman, 2019). Temel olarak rastgele sayı üretecinin yapısı Şekil 2.1' de verilmiştir (Tavas, 2011).

Şekil 2.1 RSÜ temel yapısı

Rastgele sayılar birçok kriptografik uygulamanın temelini oluşturur. İyi bir şifreleme iyi bir RSÜ gerektirir (Erkek, 2015). Güvenlik gereksinimleri için, neredeyse her zaman yüksek kaliteli bir rastgele sayı oluşturma sürecine ihtiyaç vardır (Shihai, 2021).

Rastgele sayılar günlük yaşamda çok önemli bir rol oynar ve bilimsel simülasyon ve kuantum güvenli iletişim gibi bilim ve mühendislik alanlarında

yaygın olarak kullanılır. Rastgelelik yaratmak, rastgele sayılar üretme ve kullanma sürecidir. Bilimsel simülasyonda, kullanılan rastgele sayının rastgeleliği iyi değilse, bilimsel simülasyonun sonucu yanlış olabilir. Kuantum güvenli bir iletişim sisteminde, rastgele dizilerin rastlantısallığı ve üretilme hızı, sistemin güvenliğini ve anahtar dağıtım oranını doğrudan belirler. Rastgele sayılar oluşturmanın bir çok farklı yolu vardır. Rastgele sayıların en önemli özelliği, oluşturulduktan sonraki sayının önceki sayı ile hiçbir ilgisinin olmamasıdır(Safa, 2019; Bilal, 2019). GRSÜ(True Random Number Generato (TRNG),Gerçek Rastgele Sayı Üreteci)'lerin tasarımı sonanımsal olarak gerçekleştirilirken, SRSÜ (Pseudo Random Number Generator (PRNG), Sözde Rastgele Sayı Üreteci)'lerin tasarımı yazılımsal olarak gerçekleştirilmektedir (Akgül, 2015).

Rastgele sayılar, modern kriptografiye dayalı bilgi güvenliği sistemlerinin temelidir. Kriptografinin doğru kullanımı, istatistiksel olarak yüksek kaliteli bir rastgele sayı kaynağına dayanır. Rastgele sayı üreteci, tüm sistemin güvenliğinde belirleyici bir rol oynayan bilgi güvenliği sistemindeki donanım şifreleme sisteminin temel bileşenidir. Rastgele sayılar, bir dizi rastgele sayı oluşturmak için çekirdek değerleri ve diğer parametreleri kullanan bir algoritma tarafından üretilir. Böyle bir dizi rastgele sayı, rastgele sayı akışı olarak adlandırılır. Rastgele sayılarla bir model oluştururken, rastgele sayıların kaynağını izlemek çok önemlidir. Bazen, tüm farklı işlemler için rastgele sayılar aynı akıştan gelebilirken, diğer zamanlarda her işlem benzersiz bir akış gerektirebilir (Yong, 2009)

2.1.1 Rastgele sayı üreteçlerin özellikleri

Bir ve daha fazla rastgele sayı üreteci tarafından üretilen rastgele sayı dizileri bir dizi gereksinimi karşılamalıdır. İlk olarak, sayılar belirli bir kümeden seçilmelidir. İkinci olarak dizi belirli bir kümede belirli bir dağılıma uymalıdır. Rastgele sayı üreteci'nin kalitesi genellikle elde edilen sayıların 'rastgeleliğine' göre değerlendirilir (Shihai vd. 2021).

2.1.2 Rastgele sayı üreteçlerinin sınıflandırılması

Rastgele sayılar sözde rastgele sayılara, gerçek rastgele sayılara, hibrid rastgele sayılara, basit rastgele sayılara ve güvenilir rastgele sayılara bölünmüştür. Farklı üretme yöntemlerine göre, rastgele sayı üreteçleri, sözde rastgele sayı üreteçlerine ve fiziksel rastgele sayı üreteçlerine bölünür. Sözde rastgele sayı üreteci, istatistiksel olarak yaklaşık rastgele bir dizi oluşturmak için deterministik matematiksel algoritmalar kullanır. Rastgele olması, gerçekten öngörülemeyen bir diziye değil, tamamen ilk tohuma bağlıdır. Fiziksel rastgele sayı üreteçleri, öngörülemeyen fiziksel gürültüye dayanır ve kuantum gürültüsüne dayalı klasik gürültü ve rastgele sayı üreteçlerine dayalı rastgele sayı üreteçlerine bölünür (Shihai vd. 2021).

2.1.2.1 Sözde rastgele sayı üreteci

İlk sözde rastgele sayı üreteci, 1946'da von Neumann tarafından oluşturuldu. Pratik uygulamalarda, sözde rastgele sayıları genellikle yeterlidir. Bu sayı dizileri 'görünüşte' rastgele sayılardır, ancak gerçekde sabit ve tekrarlanabilir bir hesaplama yöntemiyle üretilirler. Bu tür algoritmalar gerçek rastgele sayı dizilerinin bazı özelliklerini takribi olarak taşır. SRSÜ simülasyon, video oyunları ve kriptografi gibi uygulamaların çekirdeğidir (Genç ve Tuncer, 2019). Sözde rastgele sayılar algoritmalar tarafından üretilir, çıkış sırası periyodiktir ve uygulama yüksek güvenlik ve gizlilik gerektirmeyen alanlarla sınırlıdır (İsmail, 2013).

Gerçekte rastgele sayılar olmadığından, programda sözde rastgele sayılar kullanılır. SRSÜ kullanımın tercih edilmesinin nedeni GRSÜ'lere göre donanımsal olarak çok daha kolay gerçeklene bilmeleri ve daha yüksek hızları suna bilmeleridir. Tamamen matematiksel modellerle ifade edilebilir ve öngörülebilirler. SRSÜ kullanımın literatürde tercih edilmesinin asıl nedeni GRSÜ'lere göre donanımsal olarak çok daha kolay gerçeklenebilmesi ve daha yüksek hızları sunabilmesidir. Doğrusal eşleşik üreteçler ve doğrusal geribeslemeli kaydırmalı kaydediciler en çok bilinen ve kullanılan rastgele sayı

üreteçlerinin başında gelmektedir (İsmail, 2013) . Pek çok uygulama için sözde rastgele sayılar oldukça tatmin edicidir. Bununla birlikte kriptografik uygulamalar için bile tahmin edilemeyecek sözde rastgele bitlerin kullanılması çok önemlidir (İhsan, 2008).

Herhangi bir sözde rastgele sayı üretiminin belirli bir döngüsü olmasına rağmen, von Neumann'ın kare yönteminin döngüsü yeterince uzun değildir, bu nedenle bu yöntem hızla geliştirilir ve değiştirilir. 1949'da, matematikçi DHLehmer bu fikri doğrusal bir eşleşik üreteci (LCG-Linear Congruential Generator) kullanarak gerçekleştirdi (Özdemir, 2008).

$$X_{n+1} = (a \times X_n + c) mod\ m \tag{2.1}$$

Doğrusal uyumlu jeneratörler aslında 20. yüzyılda uzun zamandır popülerdi, ancak İnternetin hızlı gelişmesiyle insanlar PRNG için daha yüksek gereksinimler ortaya koydu. Bundan önce, rastgele sayı üreteçleri büyük ölçüde tohumlara dayanıyordu. 1990'ların ortalarında, CPU (Merkezi İşlem Birimi, Central Process Unit) 'larda yerleşik rastgele sayı üretme talimatları yoktu. Sürekli olarak iyi tohumlar oluşturmak çok nadirdir. Bu nedenle, 1990'ların ortalarında dünya, rastgele sayı üreticilerinin fanatik bir yenilik dönemine girdi ve sonsuz bir akışta çeşitli rastgele sayı üreteçleri ortaya çıktı. Aynı zamanda pek çok PRNG arasında öne çıkan The Mersenne Twister adlı bir yazılım da var. 1997 yılında Matsumoto Masayoshi ve Nishimura Twister tarafından icat edildi (Shihai vb. 2021).

Rastgele sayıların performansını ve kalitesini mükemmel bir şekilde dengeler ve bu güne kadar hala iyi bir yöntemdir. Temel fikir, çok uzun bir döngü periyodu ile deterministik bir dizi oluşturabilen ve döngü periyodu ulaşabilen doğrusal geri besleme kaydırma yazmacına dayanmaktadır. Pek çok sözde rastgele sayı üreteci sorgulanmış olsa da, şimdiye kadar insanlar bu verimli sözde rastgele sayı üreteçlerine olan bağımlılıktan kurtulamadılar (Shihai vb. 2021).

Sözde Rastgele Sayı Üretecinin uygulamalarında algoritma bilindiği takdirde herhangi bir andaki değerine bakarak sonraki çıkışlar tahmin edilebileceğinden yüksek güvenlik, gizlilik ve öngörülememe gerektiren şifreleme işlemleri için yetersiz kalmaktadır (Koray,2008).

Sözde rastgele sayı üreteçleri başka bilgisayarlara taşınabilir olmaları ve hızlı olmaları sebebiyle çokça tercih edilmektedirler. Fakat, bu üreteciler kriptografik uygulamalar için uygun değillerdir. Matematiksel karmaşıklıktan kaynaklı olan bu üreteçler için ortaya çıkan en büyük sıkıntı, şifrelerin sonlu bir sürede hesaplanabilir olmasıdır. (Hanköylü, 2019). SRSÜ'lerinde entropi kaynağı olarak belli bir algoritma, matematiksel denklem ya da önceden belirlenmiş tablolar kullanılmaktadır. Sözde rastgele sayı bir makine tarafından üretildiği için, doğası gereği fiziksel bir süreç değildir, süresi çok uzun olmasına rağmen, teorik olarak hala düzenli ve öngörülebilirdir (Özdemir, 2008).

2.1.2.1.1 Zayıf sözde rastgele sayı

Zayıf sözde rastgele sayı, belirli bir kurala göre rastgele bir değer üreten saf bir yazılım algoritması ile karakterize edilir. Ancak, adından da anlaşılacağı gibi, bu tür rastgele sayılar yalnızca belirli bir rastgelelik derecesiyle karakterize edilir ve öngörülemez, yani gerçek bir rastgele sayı değildir (Slepovichev, 2017).

2.1.2.1.2 Güçlü sözde rastgele sayı

Güçlü sözde rastgele sayı (gerçek rastgele sayıya) daha yakındır. Bu tür rastgele sayı, rastgeleliği ve öngörülemezliği tatmin edebilen daha güçlü rastgele sayı özelliklerine sahiptir (Slepovichev, 2017).

2.1.2.2 Gerçek rastgele sayı üreteci

Gerçek Rastgele Sayı Üreteçleri (GRSÜ) kontrol edilemeyen ve tahmin edilemeyen gerçek fiziksel süreçleri kullanarak sayı üretir. Bu rastgele sayı üreteçlerinin gerçek rastgeleliği tamamıyla entropi kaynağına bağlıdır. Entropi

kaynağı nitelikli olursa üretilen rastgele sayılarda nitelikli olur (Büyüksaraçoğlu ve Buluş, 2021).

Gerçek rastgele sayılar, periyodiklik, öngörülemezlik ve tekrar edilememe özelliklerine sahiptir. Gerçek rastgele sayıların gerçekte var olmadığını söylemek de mantıklıdır, çünkü dizi oluşturulduktan sonra, doğru rastgele sayının tanımı ile uyumlu değildir, bu nedenle gerçek rastgele sayı bir veri oluşturma mekanizmasıdır. Eğer bir rasgtele sayı üreteci 1) öngörülemez ve 2) tekrar tekrar üretilemezse, bu bir gerçek rastgele sayı üreteçitir. Gerçek rastgele sayı üreteci, gerçek rastgelelik özelliklerine sahiptir. Üretilen rastgele sıra öngörülemez ve periyodik değildir. Güvenlik şifreleme sistemleri, iletişim sistemleri ve istatistiksel simülasyon sistemleri gibi yüksek rastgelelik kalitesi ve güvenliği gerektiren alanlarda kullanılabilir (Özdemir, 2008).

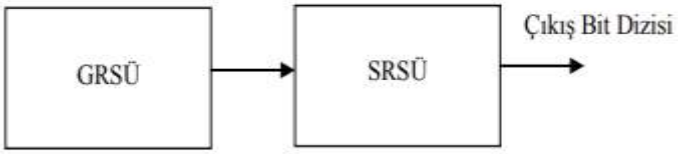
GRSÜ'ler donanım ve yazılım tabanlı olmak üzere iki farklı teknikle gerçeklenebilirler (Özdemir, 2008). GRSÜ'ler daha düşük bit aralıklarında sayı ürettikleri için SRSÜ'ler kadar hızlı değillerdir (Hanköylü, 2019) GRSÜ'ler çıkışında tamamen rastgele diziden oluşan sayılar üretilmektedir (Özkaynak, 2016). Gerçek rastgele sayılar ile sözde rastgele sayılar arasındaki farkı açıklamak gerekir. Önce zar atmak ve yazı tura atmak gibi bu ilkel yöntemlerin hepsi gerçek rastgele sayılardır. Gerçek rastgele sayılar fiziksel fenomenler kullanılarak üretilir: yazı tura atma, zar, tekerlekler, elektronik bileşenler kullanan gürültü, nükleer bölünme vb. gibi, bu tür rastgele sayı üreticilerine fiziksel rastgele sayı üreteçleri denir ve dezavantajları teknik gereksinimler nispeten yüksektir. " Bazı filozoflar, "dünyada gerçek bir rastgelelik olmadığını" iddia ediyorlar (Özdemir, 2008).

Dünyada gerçek rastgelelik olup olmadığına gelince, bu gerçekten ilgi çekici bir sorudur, ancak pratik uygulamalar ve istatistiksel ihtiyaçlar için, bu felsefi soruyu geçici olarak görmezden gelmemiz gerekir (Özdemir, 2008). Ayrıca gerçek rastgele sayıların şu üç özelliğe sahip olduğunu görebiliriz: rastgelelik (istatistiksel sapma yoktur, tamamen dağınık bir sayı dizisidir), öngörülemezlik (aşağıdaki sayıları geçmiş sayılardan tahmin edemez), tekrar üretilemezlik (dizi

kendisi kaydedilir, aynı sıraya sahip olmak imkansızdır). Gerçek rastgele sayı deneysel süreçte gösterilen dağıtım olasılığına göre belirli bir üretim sürecinde rastgele oluşturulur ve sonuç tahmin edilemez ve görünmezdir. Belirli bir algoritma simülasyonuna göre oluşturulmuş ve sonuç kesin ve görünür. Bu öngörülebilir sonucun olasılığının %100 olduğunu düşünebiliriz. Yani bilgisayarın rastgele işlevi tarafından oluşturulan "rastgele sayı" gerçek rastgele değil, sözde rastgele sayıdır. Bununla birlikte, fiziksel olaylardan tahmin edilemeyen verileri toplayan gerçek rastgele sayı üreteçlerinin bile dezavantajları vardır: yavaş ve üretimi pahalı olabilirler (Shihai vb. 2021).

2.1.2.3 Hibrid rastgele sayı üreteci

Hibrit rastgele sayı üreteçleri, GRSÜ ve SRSÜ her iksinin beraber kullanıldığı üreteçlerdir. SRSÜ olarak kaotik sistemler, kriptografik yapılar gibi deterministik yöntemler kullanılabilir. Hibrid rastgele sayı üreteci tasarımı Şekil 2.2' deki gibidir (Avaraoğlu , 2014).



Şekil 2.2 Hibrit rastgele sayı üreteci tasarımı (Avaraoğlu, 2014)

İlave girdiler ise GRSÜ kullanılarak üretilebilir. Bu üreteçlerin güvenliği kullanılan deterministik yönteme ve ilave girdilere dayanır. SRSÜ'de kulanılan deterministik yöntemin karmaşıklığı bu üreteçlerin güvenliği için belirleyicidir. Özellikle güçlü kriptografik yapılar üretecin güvenliğini garanti eder (Koçdoğan, 2015).

2.1.2.4 Basit rastgele sayı üreteci

Basit bir rastgele sayı üreteci, sözde rastgele sayılar oluşturmak için dijital teknolojiyi kullanır. Sözde rastgele sayılar temelde veya daha yüksek derecede

rastgele sayılardır ve dijital süreçler tarafından oluşturulur. Dijital teknoloji sözde rastgele sayılar üretebilir ve sözde rastgele sayılar, dijital yapıları nedeniyle belirleyicidir. Diğer bir deyişle, dijital girdi değerleri aynı olduğunda, iki özdeş sayısal rastgele sayı üreteci aynı sözde rastgele sayıları üretir. Sonuç olarak, dijital teknoloji çok istenmeyen sonuçlar üretir çünkü sözde rastgele sayılar tahmin edilebilirdir (Slepovichev, 2017).

2.1.2.5 Güvenilir rastgele sayı üreteci

Güvenilir rastgele sayılar sağlamak için geleneksel sistemler analog rastgele sayı üreticilerine dayanır. Analog rastgele sayı üreteçleri, rastgele sayıların akışlarını üretir ve sayısal rastgele sayı üreteçleri gibi periyodik yörüngeye sahip değildirler. Bu analog rastgele sayı üreteçleri, genellikle rastgele bir analog voltajı rastgele bir veri akışına dönüştürür. Bununla birlikte, bu rastgele sayı oluşturma oranları, kısa bir süre içinde çok sayıda rastgele bitin gerekli olduğu çok yüksek performanslı kriptografi için uygun değildir (Slepovichev, 2017).

Bir analog rastgele sayı üretecinin ömrü, rastgele sayılar kullanan bir dijital devreden çok daha kısadır. Rastgele sayıyı oluşturmak için kullanılan analog voltajın kaynağı bir saldırgan tarafından kolayca kontrol edilir. Bu nedenle, saldırganlara karşı savunmasız olmayan daha güvenilir analog rastgele sayı oluşturuculara ihtiyaç vardır (Slepovichev, 2017).

Analog rastgele sayı üreteci, genellikle rastgele sayılar kullanan dijital devre dışında ayrı bir entegre devrede bulunur, çünkü dijital yarı iletken süreçleri genellikle analog yarı iletken işlemleriyle uyumsuzdur (Slepovichev, 2017).

2.1.3 Rastgele sayı üreteçlerinin gelişimi

Doğadaki rastgelelik her yerde görülebilir ve çoğu zaman insanoğlunun öngörülemeyen şeylerinin belirli bir rastgelelik derecesi vardır. Zamanın gelişmesiyle birlikte, rastgele sayılar üretmenin bu orijinal yöntemi insanların ihtiyaçlarını karşılayamadı. Modern matematik çok sayıda rastgele simülasyon

gerektirir ve zar gibi manuel yöntemler gerçek rastgele sayılar üretebilse de, bu yöntem verimsizdir ve çok zaman alır ve yaygın olarak kullanılamaz (Shihai vb. 2021).

1940'ların ortalarında, RAND şirketi (American Rand Corporation), rastgele bir puls üreteci aracılığıyla çok sayıda rastgele sayı üretebilen bir makine icat etti. 1951'de rastgelelik nihayet resmi olarak standartlaştırıldı ve Ferranti Mark 1 bilgisayarına entegre edildi. Ferranti Mark No. 1, elektriksel gürültü kullanarak bir seferde 20 rastgele bit oluşturabilen yerleşik rastgele sayı üretme talimatına sahiptir. Bu özellik Alan Turing tarafından tasarlanmıştır (Shihai vb. 2021).

Rastgele sayı üreteci, tüm sistemin güvenliğinde belirleyici bir rol oynayan, bilgi güvenliği sistemindeki donanım şifreleme sisteminin temel bileşenidir (Yong, 2009). Rastgele sayılar günlük yaşamda çok önemli bir rol oynar ve bilimsel simülasyon ve kuantum güvenli iletişim gibi bilim ve mühendislik alanlarında yaygın olarak kullanılır. Elektronik bilgi teknolojisinin hızlı gelişimi ve hesaplama gücündeki önemli artışla birlikte, insanlar rastgele sayıların rastlantısallığı için gittikçe artan gereksinimlere sahiptir. Bilimsel simülasyonda, kullanılan rastgele sayının rastgeleliği iyi değilse bilimsel simülasyonun sonucu yanlış olabilir. Kuantum güvenli bir iletişim sisteminde, rastgele dizinin rastgeleliği ve üretilme hızı, sistemin güvenliğini ve anahtar dağıtım oranını doğrudan belirler. Bu nedenle, yüksek hızlı, yüksek kaliteli gerçek rastgele sayıların nasıl elde edileceği güncel araştırma noktalarından biridir (Shihai vb. 2021).

2.1.4 Rastgele sayıların kullanım alanları

Rastgele sayı üreteçlerinin önemli bir uygulamasıda ağ güvenliğinde kullanılmasıdır. Çeşitli sözde rastgele sayı üreteçlerinin ortaya çıkmasının belirli bir dereceye kadar yükseltilmesinin tam olarak ağ güvenliğinin ihtiyaçları nedeniyle olduğu söylenebilir (Shihai vb. 2021).

Mesaj doğrulamaya ek olarak, güvenli ve güvenilir iletişimin, aldatma gibi saldırıları önlemek için veri kaynağının güvenilirliğini ve iletişim varlığının gerçekliğini doğrulamak için bazı standartlaştırılmış protokoller oluşturması da gerekir. Paylaşılan bir anahtar oluşturun. Aynı zamanda, paylaşılan anahtarın gizli ve gerçek zamanlı olması gerekir. Oturum anahtarının sahtesini veya sızmasını önlemek için, oturum anahtarı iletişim kuran iki taraf arasında değiştirildiğinde şifreli metin biçiminde olmalıdır, böylece iletişim kuran iki taraf uygulama için anahtara veya genel anahtara sahip olmalıdır (Shihai vb. 2021).

Gerçek zamanlı, mesaj tekrarlama saldırılarını önlemek için son derece önemlidir. Gerçek zamanlı elde etmenin bir yolu, alınıp gönderilen her mesaja bir sıra numarası eklemektir ve bir mesaj ancak doğru sıra numarasına sahipse alınabilir. Mesajın gerçek zamanlı doğasını sağlamak için yaygın olarak "sorgulama-yanıtlama" yöntemi kullanılır, yani A kullanıcısı B'ye sorgu olarak tek seferlik rastgele bir sayı gönderir. B tarafından gönderilen mesaj da şunu içeriyorsa doğru bir kerelik rastgele sayı, A B'nin gönderdiği mesajın yeni ve kabul edilmiş olduğu düşünüldüğünden, iletişim taraflarının gerçek zamanlı yapısını sağlamak için rastgele sayılar önemli bir araçtır (Shihai vb. 2021).

Literatura baktığımızda Koçdoğan (2015), yaptığı çalışmada bir boyutlu hücresel otomat yapısı tasarlamıştır. İlk olarak temel hücresel otomat yapıları sonra hafızalı hücresel otomat yapısı incelenmiş ve daha sonra rastgele hafızalı hücresel otomat yapısı incelenmiştir. Hücrenin önceki değerlerine rastgele olarak bakılan yeni hücresel otomat yapısı tasarlanmıştır. Bu nedenle, hücrenin o anki değeri belirlenirken bir önceki ve şimdiki değerine rastgele olarak bakılır. Komşuları için de bu işlem yapılır ve hücrelerin değerleri belirlenir. Böylece bu değerler muayyen bir kurala göre etkileşerek hücrenin bir sonraki değeri belirlenir. Tasarımın FPGA (Field Programmable Gate array, Alanda Programlanabilir Kapı dizisi) üzerinde sayısal gerçeklemesi yapılmıştır. Gerçek rastgele sonuçlar alındığı görülmüş ve sonra da bu sistem üzerinden rastgele sayı üreteci tasarımı yapılmıştır (Koçdoğan , 2015).

Özkaynak vd. (2015), yaptığı çalışmada mobil cihazlar için sağlam yapılı bir gerçek rastgele sayı üreteç algoritması önermişlerdir. Algoritmanın bir uygulaması iki seviyeli kimlik doğrulama uygulamasında gösterilmiştir. Güvenlik tahlilleri önerilen algoritmanın iyi performans öz yapısına sahip olduğunu göstermiştir (Özkaynak vd. 2015).

Arathy vd. (2018) yaptığı çalışmada karmaşıklığı düşük, esnek, toplamsal beyaz Gauss gürültüsü (AWGN-Additive white Gaussian noise) kanal emülatörü yapmışlar. Yapılan işte tek tip rastgele sayılar elde etmek için çok sayıda gelişmiş Tausworthe üreteci kullanmışlar. Daha sonra Gauss rastgele sayılar oluşturmak için merkezi limit teoremi kullanılarak 12 ve 48 rastgele sayı üreteci ile AWGN kanal emülatorunun tasarımını ve uygulamasını gerçekleştirmişlerdir (Arathy vd, 2018).

2.1.5 Rastgele sayı üreteçleri için istatistiksel testler

Bu testler, rastgele sayı üreteclerinin ürettiği rastgele sayıların hakikate yakın olma durumunu ölçer. Bir sayı dizisinin rastgele olabilmesi için bu testlerden başarı ile geçmesi gerekmektedir. Bu testlerden bir tanesi bile başarılı olamadığı halde bu dizi rastgele sayılmaz. Bu dizileri FIPS, NIST, AIS3 Diehard paketlerinden biri veya bir kaçı ile test etmek mümkündür (Gürevin, 2019).

2.1.5.1 FIPS testi

Üretilen bitlerin rastgeleliği NIST ve FIPS testleri kullanılarak test edilebilir. Bu çalışmada üretilen bitlerin rastgeleliği FIPS testleri ile incelenmiştir. FIPS test kümesi dört testten oluşmaktadır. Bu testler Monobit, Poker, Koşu ve Uzun Koşu testleridir. Test edilen bitler bu dört testin hepsinden geçerse rastgele sayılır olduğu söylenebilir (Gürevin , 2019; Koyuncu, 2014; Akkaya , 2016; Çiçek, 2016).

Bu testlere sırasıyla Bölüm 2.1.5.1.1, Bölüm 2.1.5.1.2, Bölüm 2.1.5.1.3 ve Bölüm 2.1.5.1.4'te yer verilmiştir.

FIPS 140- 1 testi genellikle 20 Kbit gibi küçük boyutlu blok uzunluğuna sahip bit dizilerinin testinde kullanılmaktadır (Koyuncu, 2014).

2.1.5.1.1 Monobit testi

Bu testin amacı, 20.000 bitlik akışta 0 ve 1 sayısını tespit etmektir (Erkek, 2015). (Song, 2003). Monobit testinin başarılı olabilmesi için 20.000 bitlik X Dizesindeki '1'lerin sayısının 9654 <X<10346 aralığında olması gereklidir. (Çiçek , 2016).

2.1.5.1.2 Poker testi

Bu testin amacı rastgele bir diziden beklenen, tüm m bitlik blokların k uzunluklu bir dizide aynı sayıda birbirini tekrar etmesidir (Erkek, 2015). Poker testinin başarılı olabilmesi için X değerinin 20.000 bitlik dizide alınan değerlerin 1.03<X<57.4 aralığında olması gereklidir (Çiçek , 2016).

2.1.5.1.3 Uzun koşu testi

Bu testte üretilen rastgele sayı dizisinde uzun 0 veya 1 bit dizilerinin olup olmadığı kontrol edilmektedir. Rastgele sayı dizisi, içerisinde 34'ten daha fazla bit dizisi içerirse, rastgele sayı üreteci uzun koşu testini geçemez (Koçdoğan, 2015). Uzun koşu testlerinin başarılı olabilmesi için alınan 20.000 bitlikde dizide '1' veya '0' lardan oluşan blokların sayısı <=34 değerine eşit veya bu değerden küçük olmalıdır (Çiçek , 2016).

2.1.5.1.4 Koşu testi

Bu testten başarıyla geçmesi için, dizide ardarda gelen '1' ve '0'lardan oluşan çeşitli uzunluktaki blokların (runs'ların) sayısının tabloda belirtildiği gibi olması beklenir. 6 bitten daha uzun bloklar 6 bitlik olarak kabul edilmektedir (Erkek, 2015). Koşu testinin başarılı olabilmesi için ard arda gelen '1' veya '0'

bitlerinden oluşan blok sayısının alınan Çizelge 2.1 de verilen değer aralıklarında olması gereklidir (Çiçek , 2016).

Çizelge 2.1 Koşu testi için değer aralığı

1- 2267 <X< 2733
2- 1079 <X< 1421
3- 502 <X<748
4- 223 <X< 402
5- 90<X<223
6- 90<X<223

2.1.6 NIST testi

NIST testi FIPS testine göre daha güçlü bir testtir. FIPS testinden geçen bir test NIST testinden geçmeyebilir. Bitler tüm testlerden başarıyla geçerse test başarılı sayılır (Andrew, 2010).

Testlerin içeriği aşağıda maddeler halinde belirtildiği gibidir;

- Frekans testi
- Bir blok içinde frekans testi
- Akış testi
- Bir blok içerisindeki en uzun bir dizi testi
- İkili matris sıra testi
- Ayrık fourier dönüşümü testi
- Çakışmayan şablon eşleme testi
- Çakışan şablon eşleme testi
- Maurer'in evrensel istatistiksel testi
- Seri test
- Yaklaşık entropi testi
- Birikerek artan toplamlar testi
- Rastgele gezinimler testi
- Rastgele gezinimler değişken testi

- NIST testinde en önemli parametre P değeridir. P değeri 0'a yakın olursa rastgele özelliği azalır, 1'e yakın olursa rastgele özelliği artar (Andrew, 2010).

2.1.7 AIS31 testi

AIS31 metodolojisine göre Fransa Ulusal Bilgi Sistemleri Güvenliği Ajansı (ANSSI) tarafından doğrulanmıştır. Alman (BSI- Bilgi Güvenliği Ofisi) AIS31 metodolojisi, üç farklı gerçek rastgele sayı üreteci sınıfını tanımlar. Bu gerçek rastgele sayı üreteci sınıfları aşağıda maddeler halinde sıralandığı gibidir;

- PTG.1 - Entropi kaynağının toplam arızasını tespit eden dahili testlere sahiptir.
- PTG.2 - PTG.1 sınıfı, ek olarak entropi kaynağının stokastik bir modeli ve ham rastgele sayıların canlı istatistiksel testlerine sahiptir.
- PTG.3 - PTG.2 sınıfı, ek olarak ham rastgele sayı çıktısının kriptografik son işlemesine sahiptir (Matthias vd. 2021).

2.1.8 Diehard testi

Diehard testi birkaç yıl içinde George Marsaglia Tarafından geliştirildi ve ilk olarak 1995'te rastgele sayılardan oluşan bir CD- ROM' da (Compact Disc Read-Only Memory, Kompakt Disk-Salt Okunur Bellek) yayınlandı. Dıehard'daki testlerin, girdi dosyasyı gerçekten bağımsız rastgele bitler içeriyorsa (0,1)'de aynı olması gereken bir p değeri döndürür. Bu p değeri, p=F(X) ile elde edilir (Robert, 2021).

2.1.9 Rastgele Sayı Üreteclerinin gerçekleme yöntemleri

Literatürde Rastgele sayı üreteçleri, birçok farklı matematiksel yöntem kullanmaktadır. Rastgele Sayı Üreteçlerinin bir olasılık dağılımını takip ettiği söylenebilir (Koray, 2008). Bunlara genel olarak doğrusal eşzamanlı oluşturucu, orta kare yöntemi, olasılık dağılımına dayanan ters çevirme, kabul-ret yöntemi,

Mersenne Twister, Monte Carlo, Tausworthe gibi farklı algoritmalar örnek verilebilir (L'ecuyer, 2017).

2.1.9.1 Doğrusal eşzamanlı oluşturucu

Doğrusal eşzmanalı oluşturucu 1951 yılında Lehmer tarafından önerilmiştir (Atay, 2016). Doğrusal eşlik üreteci belirlenen 0 ve m-1 arasındaki (X_0 , X_1) tam sayılar dizisi tanımlar:

$$X_{i+1}\ (aX_i + C) \bmod m, \quad i = 0,1,2, \ldots. \tag{2.2}$$

X_0 : dizinin başlangıc değeri
a : sabit çarpan katsayısı
c : artış miktarı
m :modulus

X_0 , a, c, m tamsayılardır. Bu sayılat istatistiksel özelliklerde ve çevrim uzunluğunda büyük etkiye sahiptir. Her hangi bir X_i değeri için rassal sayı şu şekildedir: (Atay, 2016).

$$U_i = X_1/m \tag{2.3}$$

2.1.9.2 Orta kare yöntemi

Orta kare yöntemi 1946 'da Von Neumann ve Metropolis tarafından önerilmiştir. Orta kare yönteminde bir m basamaklı sayının karesi alınır ve alınansayının ortasında yer alan m basamak alınarak, yeni bir sayı üretilir. Yöntemin aşamaları şu şekildedir: İlk olarak 4 basamaklı bir sayı seçilir ve sayının karesi alınır. 8 basamağı doldurmak için gerekirse sayının sol tarafına sıfır konulur. Ortadaki 4 basamak seçilir ve bu sayılar rastgele sayı olarak kullanılır. İstenen sayıda rastgele sayı elde edene dek 3. ve 4. adım tekrar edilir (Başlıgil, 2010).

2.1.9.3 Mersenne Twister yöntemi

Mersenne Twister, 1997 yılında Makoto Matsumoto ve Takuji Nishimura tarafından geliştirildi. Mersenne Twister yöntemi sözde rastgele sayılar özde rastgele tamsayıların hızlı bir şekilde üretilmesini sağlar. Mersenne Twister yöntemi, diğer rastgele sayı üreteçlerine kıyasla çok uzun bir periyoda sahipdır. Yöntem 16.777.216 yeni 2^{24} periyoda sahiptir . 16.777.216 sözde rastgele sayı üretildikten sonra, aynı sayı dizisi tekrarlanır. Mersenne Twister yönteminin $2^{19937} - 1$ gibi çok daha uzun bir periyodu vardır ve bu, günümüzün hesaplama standartlarına göre neredeyse sonsuzdur (Matsumoto vd. 1998).

2.1.9.4 Monte Carlo yöntemi

1940'larda bilim adamları Stanislav Ulam ve Nicolas Metropolis , Los Alamos Ulusal Laboratuvarı'nda nükleer silah programı için çalışırken Monte Carlo yöntemini icat ettiler. Birçok hesaplama problemini çözmek için sözde rastgele sayıların kullanımında önemldir. Genellikle Monte Carlo yöntemi, belirli kurallara uyan rastgele sayılar oluşturarak çeşitli matematiksel problemleri çözer ve birçok alanında yaygın olarak kullanılan bir sayısal hesaplama algoritmasıdır (Anderson, 1986).

2.1.9.5 Tausworthe yöntemi

1965 yılında Tausworthe'un yazdığı makaleden hareketle yaratılmıştır. Şifreleme metotlarıyla ilgili olan bu yöntemde rastgele sayılar art arda gelen sayı çiftlerinin tekrarlanmasıyla üretilir (Tausworthe, 1965).

2.2 Donanım Tanımlama Dilleri

Verilog HDL (HDL- Hardware Description Language) ve VHDL (Very High Speed Integrated Circuit Hardware Description Language), her ikisi de 1980'lerin ortalarında geliştirilmiş, dünyadaki en popüler iki donanım tanımlama dilidir. Verilog dili aslında, 1983 yılında simülatör ürünleri için Gateway Design

Automation tarafından geliştirilen bir donanım modelleme diliydi. Verilog dili 1995 yılında Verilog 95 olarak da bilinen IEEE (Institute of Electrical and Electronics Engineers (Elektrik ve Elektronik Mühendisleri Enstitüsü) Std 1364-1995 adında bir IEEE standardı haline geldi. Verilog 2001, Verilog- 95'in geliştirilmiş büyük sürümüdür. Çok boyutlu diziler, ifade blokları oluşturma vb bazı yeni işlevlere sahiptir. Verilog-2001 Verilog' un en uygun sürümüdür (Başak, 2011).

Verilog dili genel bir donanım tanımlama dilidir, sözdizimi C diline benzer isimler, giriş parametreleri ve çıkış parametreleri vardır. Ancak bunlar HDL açıklama dilinde modül adları, giriş sinyalleri ve çıkış sinyalleri olarak adlandırılır. Ayrıca öğrenmesi ve kullanması kolaydır. Bu nedenle Verilog, yayın programında VHDL'den yaklaşık iki yıl sonra yayınlanmış olmasına rağmen, Verilog'un popülaritesi VHDL'yi çok aştı. Verilog simülatörü, kullanıcılara çok esnek bir simülasyon ortamı sağlar (Thomas vd. 2002). Verilog HDL, dijital mantık devrelerinin tasarımı için kullanılan bir dildir . Verilog HDL'de açıklanan devre tasarımı, devrenin Verilog HDL modelidir (Savran, 2017).

Verilog programlama dili arayüzünün gelişimi üç nesilden geçmiştir.Birinci nesil, C programı ile Verilog tasarımı arasında veri transfer edebilen bir görev veya fonksiyon alt rutinidir; ikinci nesil, tarafından özelleştirilebilen bir erişim alt rutinidir. Program ve Verilog'un dahili veri gösterim arayüzü kullanılır, üçüncü nesil Verilog süreç arayüzüdür ve ilk iki nesil programlama dili arayüzlerinin fonksiyonlarını daha da genişletir (Başak, 2011).

2.2.1 FPGA Programlama

FPGA iç bağlantılarının yapılması ancak programlanması ile mümkündür. FPGA ile donanım tasarım yapabilmek amacıyla geliştirilmiş bir çok dil mevcuttur. FPGA'in devresinin tasarımı 2 yolla yapılmaktadır. Birincisi şematik tasarımla, ikincisi ise Donanım tanımlama dili olan Verilog veya VHDL'den biridir. Bunlardan yaygın olarak kullanılan VHDL ve Verilogdur (Savran , 2017).

2.2.1.1 Şematik tasarım

Şematik devre elemanlarının teker teker yerleştirilip aralarındaki bağlantıların şematik üzerinde yapılmasıdır (Savran, 2017).

2.2.1.2 VHDL dili

Üst düzey bir donanım davranışı tanımlama dili olan VHDL dili, FPGA / ASIC tasarımında yaygın olarak kullanılmaktadır (Savran, 2017).

VHDL dilinin istenilen yere taşınılması ve tekrar tekrar kullanılabilmesi avantajlardan birkaçıdır. VHDL sıradan bilgisayar programlama dillerine benzemez. Günümüz donanım tasarımının gelişme yönünü temsil eden donanım davranışının doğrudan tanımlanmasıyla donanımın fiziksel olarak gerçekleştirilmesini gerçekleştirir. VHDL, mantık tasarım sürecinde çeşitli ihtiyaçları karşılamak için tasarlanmıştır (Savran, 2017).

İlk olarak, mantık tasarımında kaç tane alt mantık olduğu ve bu alt mantıkların nasıl bağlantılı olduğu gibi mantık tasarımının yapısını tanımlamak için kullanılabilir. Ek olarak, VHDL, belirli bir mantığın nasıl uygulandığını pek önemsemez, ancak geliştiricinin enerjisini mantık tarafından uygulanan işlevlere odaklanır (Savran, 2017).

İkincisi, VHDL, donanım davranışının açıklamasını tamamlamak için yüksek seviyeli bir dile benzer bir cümle formatı kullanır, daha güçlü modülerliğe sahiptir ve iyi okunabilirlik ve program taşınabilirliğine sahiptir (Savran, 2017).

Üçüncüsü, VHDL tarafından verilen mantığın simülasyonu ve hata ayıklaması, tasarım çalışması için en büyük alanı sağlar. VHDL hata ayıklama süreci oldukça esnektir: bir yandan, geleneksel dalga biçimi uyarımı uygulamak veya test vektörleri yazmak gibi geleneksel hata ayıklama yöntemlerini kullanabilirsiniz; diğer yandan, büyük ölçüde hızlandırabilen bazı VHDL kaynak kodu hata ayıklayıcılarını kullanabilirsiniz (Savran, 2017).

VHDL dili şu anda esas olarak dijital devrelerin tasarımını tanımlamaktadır ve analog devrelerin tasarımı henüz iyi ifade edilmemiştir. VHDL dili programlama sırasında daha standart hale getirilmeli, program yapısı tüm sistemin donanım yapısına uygun olmalı ve her modülün sinyal zamanlama ilişkisi veri akışının eğilimiyle tutarlı olmalıdır. VHDL dilinin tasarım formatı, belirli donanım nesnelerine yönelik bir dildir, bu nedenle donanım öğelerinden bağımsız herhangi bir programlama anlamsızdır (Savran, 2017).

2.2.1.3 Verilog dili

Verilog HDL hem bir davranış tanımlama dili hem de bir yapı tanımlama dilidir. Bu, tasarlanan devrenin bir Verilog HDL modelini oluşturmak için hem devrenin işlevsel açıklamasını hem de bileşenler ve bunlar arasındaki bağlantıları kullanabileceğiniz anlamına gelir. Verilog HDL'nin yapıcı cümleleri sinyali doğru bir şekilde modelleyebilir (Savran, 2017).

2.2.2 Verilog dilinin özellikleri

Verilog dilinin birkaç özelliği vardır. Bu özellikler aşağıda maddeler halinde belirtildiği gibidir;

- Sıralı yürütmenin veya paralel yürütmenin program yapısını tanımlayabilir
- Sürecin başlangıç zamanını açıkça kontrol etmek için gecikme ifadeleri veya olay ifadeleri kullanın
- Komut verilen olaylarla diğer işlemlerin aktivasyonunu veya durma davranışını tetikler
- Koşul / döngü gibi mantıksal kontrol yapısı sağlar
- Parametreli ve sıfır olmayan süreye sahip bir görev programı mekanizması sağlar
- İfadeler oluşturmak için aritmetik operatörler, mantıksal operatörler ve bitsel operatörler sağlar

- Kombinatoryal mantığın temel unsurlarını temsil eden tam bir ilkel uygulanır.
- Çift yönlü yolların ve dirençlerin açıklamalarını sağlar
- MOS cihazları için şarj paylaşımı ve zayıflatma modelleri oluşturur
- Yapılandırılmış cümleler aracılığıyla sinyal modellerini doğru bir şekilde oluşturabilir (Savran, 2017).

Verilog HDL hem davranış tanımlama dili hem de yapı tanımlama dilidir. Bu, tasarlanan devrenin bir Verilog HDL modelini oluşturmak için hem devrenin işlevsel tanımını hem de bileşenler ve bunlar arasındaki bağlantıları kullanabileceğiniz anlamına gelir. Verilog modeli, gerçek devrenin farklı soyutlama seviyeleri olabilir. Bu soyut düzeylerin beş türü ve bunlara karşılık gelen model türleri vardır:

- Sistem düzeyi (sistem): Tasarım modülünün dış performans modelini gerçekleştirmek için üst düzey bir dil yapısı kullanılır.
- Algoritma düzeyi (algoritma): Tasarım algoritmasının modeli, üst düzey dil yapısı ile uygulanmaktadır.
- RTL seviyesi (Register Transfer Level): Verilerin kayıtlar arasında nasıl aktığını ve bu verilerin nasıl işleneceğini açıklayan bir model.
- Kapı düzeyi: Mantık kapılarını ve mantık kapıları arasındaki bağlantıları açıklayan bir model.
- Anahtar düzeyi: Cihazdaki transistörleri ve depolama düğümlerini ve bunlar arasındaki bağlantıları açıklayan bir model (Thomas vd. 2002).

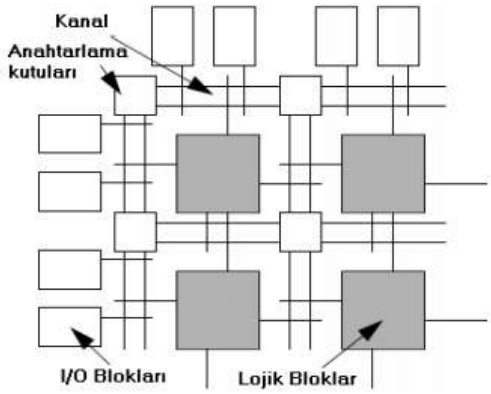
Yapılandırılmış bir dil olarak Verilog HDL dili, kapı seviyesi ve anahtar seviyesi model tasarımı için de çok uygundur. Yapısal özellikleri nedeniyle aşağıdaki işlevlere sahiptir:

- Eksiksiz bir birleşik ilkel kümesi sağlar;
- Çift yönlü yollar ve dirençli aygıtlar için temel öğeler sağlar;
- MOS cihazlarının dinamik şarj paylaşımı ve şarj düşüşü modeli oluşturulabilir (Thomas vd. 2002)

Verilog HDL'nin yapıcı cümleleri, sinyali doğru bir şekilde modelleyebilir. Bunun nedeni, Verilog HDL'de, son derece doğru bir sinyal modeli oluşturmak için gecikme ve çıkış gücünün ilkellerinin sağlanmasıdır. Sinyal değeri farklı yoğunluklara sahip olabilir ve çok çeşitli bulanıklık değerleri ayarlanarak belirsiz koşulların etkisi azaltılabilir (Thomas vd. 2002).

2.3 Alanda Programlanabilir Kapı Dizileri

Programlanabilir yarı iletken çipler 1970'lerden beri araştırılmış ve geliştirilmiştir. FPGA yani alanda programlanabilir kapı dizisi 1985 yılında Ross Freema tarafından icat edildi. Tasarım sırasında paralel işlem yapabilme nedeniyle FPGA kullanımı günümüzde oldukça yaygınlaşmıştır. FPGA'nın önemli bir özelliği gerekli uygulama veya fonksiyonel gereksinimlere göre yeniden programlanabilme yeteneğidir (İçer**,** 2016). Bu elektronik bileşen, televizyon, cep telefonları, bilgisayarlar ve bu gibi bir elektronik cihazın beyni olarak daha iyi bilinir. FPGA, bir devre tarafından programlanan bir çiptir. FPGA'lar ayrıca yoğun bir şekilde kullanmaktadır. Genel FPGA yapısı Şekil 2.3'deki gibidir (Yılmaz, 2008).



Şekil 2.3 FPGA mimarisi (Yılmaz, 2008).

FPGA, programlama ile devrelerin sentezlenmesini ve gerçeklenmesini sağlayan bir çiptir. FPGA'nın avantajları aşağıda maddeler halinde sıralandığı gibidir;

- Yüksek hızlı iletişim arayüzü tasarımı. FPGA, yüksek hızlı sinyal işleme için kullanılabilir. Genel olarak, örnekleme hızı yüksekse ve veri hızı yüksekse, verileri işlemek için verileri kesmek ve filtrelemek, veri hızını düşürmek gibi FPGA gerekir ve sinyalin işlenmesini, iletilmesini ve depolanmasını kolaylaştırır.
- Dijital sinyal işleme. Bunun avantajı, gerçek zamanlı performansın iyi olması ve alanın CPU'dan çok daha hızlı olan hız için kullanılmasıdır.
- Daha büyük paralellik. Bu, esas olarak iki teknolojiyle elde edilir: eşzamanlılık ve ardışık düzen. Eşzamanlılık, birden fazla modülün aynı anda bağımsız hesaplamalar yapabilmesi için hesaplama kaynaklarının tekrarlanan tahsisini ifade eder.

FPGA'nın temel özellikleri aşağıda maddeler halinde sıralandığı gibidir;

- ASIC (Application Specific Integrated Circuit, Uygulamaya Özel Tümleşik Devre), devresini tasarlamak için FPGA kullanarak, kullanıcılar döküm üretmeden uygun bir çip elde edebilirler.
- FPGA, diğer tamamen özelleştirilmiş veya yarı özelleştirilmiş ASIC devrelerinin örnek bir parçası olarak kullanılabilir.
- FPGA içerisinde bol miktarda flip-flop ve I / O pinleri bulunmaktadır.
- FPGA, ASIC devrelerinde en kısa tasarım döngüsüne, en düşük geliştirme maliyetine ve en düşük riske sahip cihazlardan biridir (Yılmaz, 2008).

FPGA'lar piyasaya 1980 tarihinden itibaren girdiklerinden dolayı genel amaçlı Merkezi İşlem Birimi (Central Process Unit , CPU), Uygulamaya Özgü Tümleşik Devreler (Application Specific Integrated Circuit, ASIC) ve hatta Grafik İşlemci Birimi (Graphics Processing Unit, GPU) rekabet ettikten sonra bir arada var olabildiler (Sass vd. 2010).

Şekil 2.4 Donanım platformlarının karşılaştırılması (Wikipedia, 2021).

CPU bilgisayar sisteminin çekirdeğidir ve bilgisayarın önemli bir bileşenidir. Bu nedenle, merkezi işlem biriminin güç kaynağı sistemi, bilgisayarın normal çalışmasını sağlamada hayati bir rol oynar. Genel olarak CPU'nun çalışma durumu işletim sistemi tarafından kontrol edilir ve işletim sistemi hesaplanan CPU kullanım oranına göre her bir CPU'nun çalışma frekansını kontrol eder (Sass vd. 2010).

1970'lerden önce, merkezi işlem birimi birden fazla bağımsız birimden oluşuyordu. Daha sonra, entegre devrelerden oluşan merkezi işlem birimi geliştirildi. Merkezi işlem birimi genel olarak karmaşık bilgisayar programlarını çalıştırabilen bir dizi mantıksal makineyi ifade eder ve çok sayıda programlama dili ve de geliştirme ortamı nedeniyle onu kullanmak daha kolaydır. İlk merkezi işlemciler genellikle büyük ölçekli ve uygulamaya özel bilgisayarlar için özelleştirildi. Bununla birlikte, belirli uygulamalar için CPU'ları özelleştirmenin bu pahalı yöntemi, büyük ölçüde bir veya daha fazla amaç için ucuz, standartlaştırılmış ve uygun işlemciler geliştirmeye yol açmıştır. CPU, bilgi işlem cihazını tanımlayan temel bileşendir, ancak tek bileşen değildir, sadece beyindir. Çip, cihazın içindeki ana devre kartında özel bir yuvaya yerleştirilmiştir. Bilgileri geçici olarak depolayan bellekten önemli ölçüde farklıdır. CPU, tek bir bilgisayar çipine milyarlarca minik transistör yerleştirilerek oluşturulmuştur. Bu transistörler, sistem belleğinde depolanan programları çalıştırmak için hesaplamalar yapmasına izin verir. Modern

cihazlarda, bir masaüstü veya dizüstü bilgisayar, sistem için birçok işleme işlevini gerçekleştirebilen özel bir CPU'ya sahiptir (Sass vd. 2010).

Başlangıçta CPU'nun tek bir işlemci çekirdeği vardı. Günümüzün modern CPU'ları birden çok çekirdekten oluşuyor ve paralel hesaplamayı başarmak için aynı anda birden fazla komutu yürütmelerine izin veriyor. CPU'nun özü, programlardan veya uygulamalardan talimatlar almak ve hesaplamalar yapmaktır. Süreç üç ana aşamaya bölünmüştür: edinme, kod çözme ve yürütme. CPU, komutları RAM (Random Access Memory, Rastgele Erişimli Bellek) 'den alır, komutların gerçek anlamını çözer ve ardından komutları yürütmek için CPU'nun ilgili kısımlarını kullanır. Gerçekleştirilen talimatlar veya hesaplamalar temel aritmetik, belirli sayıların karşılaştırılması veya hafızaya taşınmasını içerebilir. Bilgi işlem cihazındaki her şey 0, 1 sayısı ile temsil edildiğinden, bu basit görevler CPU'nun çalışmasına eşdeğerdir (Sass vd. 2010).

Modern sistemlerde, CPU tüm görevleri tamamlayamaz, ancak yine de gerekli işlev numaralarını özel donanıma sağlamalıdır. CPU, genel sistem performansı için eskisi kadar önemli olmasa da, ekipmanın çalıştırılmasında hala önemli bir rol oynamaktadır. Yalnızca programdaki komutların yürütülmesinden sorumlu olduğu için, CPU ne kadar hızlı olursa, birçok uygulamanın çalışma hızı da o kadar yüksek olur. Kısacası, CPU her derde deva değil ama çok önemli. Genellikle daha hızlı bir CPU, sisteminizin veya cihazınızın daha hızlı çalışacağı anlamına gelir (Sass vd. 2010).

GPU bilgisayardan bağlı monitöre iletilen tüm içeriğin işlenmesinden sorumludur. GPU , grafik işleme birimi, NVIDIA tarafından 1999'da piyasaya sürülen GeForce256 grafik kartında ortaya konan bir konsepttir. Derinlik arabelleklerini işlemede uzmanlaşmış ve dokulara ve diğer arabelleklere hızla erişen özel bir yongaya sahiptir. Daha da önemlisi, paralel olarak çalışan programlanabilir gölgelendiricileri destekleyen bir işlemci sağlar. Günümüzde, GPU'lar programlanabilir ve daha esnek yönde yavaş yavaş gelişmektedir. GPU, paralel görevleri işlemeye odaklanır ve bu konuda hızda büyük bir avantaja

sahiptir. GPU, bir grup benzer veriyi düzenli bir şekilde işleyen bir akış işlemcisidir (Sass vd. 2010).

Hem CPU hem de GPU silikon tabanlı mikro işlemciler olmasına rağmen, özünde ikisinin dağıtım rolleri tamamen farklıdır. CPU, çeşitli karmaşık görevleri işlemek için kullanılan bilgisayarın beynidir ve GPU verimli bir şekilde çalışamaz. CPU ve GPU, insan beyni gibidir. İlki çok sayıda farklı hesaplama türünü işleyebilirken, GPU'nun görevi grafikleri oluşturmak ve mevcut tüm çekirdekleri belirli görevlere odaklamaktır. Tek bir görev için çok sayıda karmaşık grafik ve geometrik işlem gerektiğinde, GPU ilgili işi yapar (Sass vd. 2010).

Diğer tarafta ASIC var. Uygulamaya özel bir entegre devre, belirli bir uygulama veya amaç için özel olarak tasarlanır ve üretilir. Programlanabilir mantık cihazları veya standart mantık entegre devreler ile karşılaştırıldığında, ASIC'ler hızı artırabilir çünkü bunlar özellikle bir şeyi başarmak için tasarlanmıştır ve bunu yapabilirler. Ayrıca küçültülebilir ve daha az elektrik tüketebilir. Bu devrenin dezavantajı, özellikle birkaç üniteye ihtiyaç duyulduğunda, tasarım ve üretimin daha pahalı olabilmesidir (Sass vd. 2010).

FPGA'lar genellikle ASIC' lerden daha yavaştır , daha karmaşık tasarımları tamamlayamaz ve daha fazla güç tüketemez. Bununla birlikte, FPGA'nın birçok avantajı vardır, örneğin, hızlı bir şekilde bitirilebilir ve programdaki hataları düzeltmek için tasarımcı tarafından iç mantığı tekrar tekrar değiştirilebilir. Ayrıca, FPGA kullanarak hata ayıklama maliyeti nispeten düşüktür. Üreticiler ayrıca sınırlı düzenleme yeteneklerine sahip ucuz FPGA ürünleri sağlayabilir. Bu yongaların bazılarının düzenlenebilirliği zayıf olduğundan, bu tasarımların geliştirilmesi sıradan bir FPGA üzerinde tamamlanır ve ardından tasarım uygulamaya özel bir entegre devreye benzer bir çipe aktarılır. FPGA'lar evrensel değildir, ancak üzerlerinde bir CPU veya hatta bir GPU'dan daha iyi performans gösterecek bir dizi algoritma ve görev vardır (Sass vd. 2010).

2.3.1 FPGA'ın yapısı

LUT (LookUp Table, Arama Tablosu) tabanlı FPGA yüksek bir entegrasyon derecesine sahiptir ve cihaz yoğunluğu on binlerce kapıdan on milyonlarca kapıya kadar değişir. Sekans ve mantık kombinasyonlarının son derece karmaşık mantık devresi işlevlerini tamamlayabilir. FPGA tasarımı ve üreticileri arasında Xilinx, Altera, Lattice, Actel, Atmel ve QuickLogic bulunur. Şu anda, en yüksek FPGA pazar payına sahip iki büyük şirket - Xilinx ve Altera, SRAM(Static Random Access Memory, Durağan Rastgele Erişimli Bellek) teknolojisine dayalı FPGA'lar üretiyor ve kullanım sırasında programı kaydetmek için harici bir belleğe ihtiyaç duyuyor. FPGA'ların yapısına aşağıdakılar dahildir (Özpolat, 2017).

- Alanda Programlanabilir lojik bloklar
- Alanda Programlanabilir ara bağlantılar
- Alanda Programlanabilir giriş/ çıkış blokları
- Alanda Progranabilir alıcı/ verici bloklar

2.3.1.1 Progranabilir lojik bloklar

Progranabilir lojik blokları, FPGA'nın çeşitli dijital devre yapılarını elde etmesini sağlayan FPGA'ların çekirdeğidir. Blok tasarımının kalitesi, FPGA'nın performansını ve FPGA'nın içerdiği maksimum sistem kapılarını doğrudan etkiler. Bu nedenle, Progranabilir lojik bloğu, FPGA cihazlarının tasarımında en önemli şeydir (Özpolat, 2017).

2.3.1.2 Programlanabilir ara bağlantılar

Programlanabilir ara bağlantı kaynakları, belirli işlevlere sahip devreler oluşturmak için CLB'ler(Yapılandırılabilir Mantık Blokları) arasında bağlanan çeşitli uzunluklarda kablo segmentleri ve bazı programlanabilir bağlantı anahtarlarını içerir (Özpolat, 2017).

2.3.1.3 Programlanabilir giriş/ çıkış blokları

Programlanabilir giriş / çıkış modülü esas olarak çip üzerindeki mantık ile harici pinler arasındaki arayüzü tamamlar ve genellikle çipin etrafında düzenlenir (Özpolat, 2017).

2.3.1.4 Programlanabilir alıcı/ verici bloklar

FPGA ile bağlı sistemler arasında yapılan yüksek hızlı veri alışverişi sağlamaktadır. Alıcı- verici blokları tek yönlü ya da çok yönlü kullanılabilen bloklardır. İçerisinde seri- paralel ya da paralel- seri dönüştürücüleri bulunmaktadır. FPGA'nın gelişimi geleneksel PC'lerin (Personal Computer, Kişisel Bilgisayar) ve mikrodenetleyicilerin geliştirilmesinden çok farklıdır (Özpolat, 2017).

2.3.2 FPGA'ın uygulama alanları

FPGA'nın ilk uygulama alanı aynı zamanda geleneksel uygulama alanı, iletişim alanıdır (Özpolat, 2017).

FPGA'nın uygulama alanlarına aşağıdakiler dahildir;

- ASIC tasarımı.
  ASIC, özel kullanıcı ihtiyaçları, ASIC toplu üretimi ve genel entegrasyon devresi, daha düşük ve daha düşük güç tüketimi avantajları ile karşılaştırıldığında, güvenilirliği artıran, gelişmiş performans, gelişmiş gizlilik, maliyet azaltma için karakterize edilmiştir. ASIC'i tasarlamak için FPGA kullanmak en düşük tasarım maliyetine sahiptir, ancak çip fiyatı en yüksek olanıdır ve küçük parti ASIC ürünleri için uygundur. FPGA'nın ASIC ile arasındaki fark, kullanıcıların çip düzenine ve süreç sorunlarına müdahale etmesine gerek olmaması ve mantık işlevlerini istedikleri zaman değiştirerek kullanımı esnek hale getirmesidir (Özpolat, 2017).
- Sayısal işaret işleme.

FPGA’ larda sinyal işleme algoritmalarını uygulamak için, donanım ve yazılım tasarım seviyelerinde geleneksel sınırlamalar vardır. FPGA donanım yapısı, gelişmiş DSP (Digital Signal Processing, Sayısal Sinyal İşleme) bloklarını uyguluyor. Bu daha fazla işleve ve daha güçlü performansa sahiptir. Sistem düzeyinde tasarım yazılımı, blok seviyesinden donanım tanımlama diline dönüşümü basitleştiriyor. Bazı sistem araçları, FPGA’larda sinyal işleme algoritmalarını uygulama sürecini böyük ölçüde basitleştiren popüler DSP algoritma geliştirme araçlarını (MATLAB gibi) entegre eder (Özpolat, 2017).

- Yazılımsal (soft) mikroişlemci tasarımı

Mikroişlemci özel bir programlanabilir entegre devredir. Birçok kişi FPGA' ya yalnızca bir mikroişlemci koysa da, FPGA çok çekirdekli bir işlemci oluşturmak için yeterli kaynaklara sahip daha fazla mikroişlemciyi barındırabilir . Tek bir FPGA üzerindeki mikroişlemcilerin sayısı yalnızca FPGA kaynaklarının sayısıyla sınırlıdır. Bazı insanlar tek bir FPGA' ya yüzlerce yumuşak çekirdekli işlemci koyar. Bu, büyük ölçüde paralel hesaplama elde etmek için bir yöntemdir ve aynı zamanda bellek içi hesaplamaya da uygulanabilir. Bir veya daha fazla uçta kodlanmış talimatları kabul eden, talimatları yürüten ve durumunu açıklayan sinyaller çıkaran entegre bir devre. Bu talimatlar dahili olarak girilebilir, merkezileştirilebilir veya saklanabilir. Yarı iletken merkezi işlem birimi olarak da bilinen bu birim , bir mikro bilgisayarın ana bileşenidir. Bir mikroişlemcinin bileşenleri genellikle tek bir yongaya veya aynı bileşene monte edilir, ancak bazen birkaç farklı yongaya dağıtılır (Özpolat, 2017).

- Haberleşme ağlarında

FPGA’ lar fiziksel katman ile yüksek seviyeli haberleşme protokol katmanları arasında arayüz bağlantısını gerçeklemektedirler. FPGA’ lar haberleşme ve ağ oluşturma fonksiyonlarını tek cihazda birleştiren yüksek hızlı alıcı- verici birimlerini içerebilmektedir (Özpolat, 2017).

- Donanım simülasyonlarında

Donanım simülasyon katman yazılım vasıtasıyla donanım komut seti ve işlemci davranışını simüle eder (Özpolat, 2017).

- Veri Merkezi (Data Center)

Veri Merkezleri çok büyük boyutlarda verinin toplanması, işlenmesi, depolanması ve dağıtılması amacıyla, bilgisayar ve ağ donanımlarının bir merkezde toplandığı yerlerdir (Özpolat, 2017).

2.3.3 FPGA Sınıflandırmaları

FPGA sınıflandırması na Bölüm 2.3.3.21 ve Bölüm 2.3.3.2'de yer verilmiştir.

2.3.3.1 Programlanabilir mantık blokları olarak sınıflandırma

Programlanabilir mantık blokları olarak sınıflandırılan mantıksal fonksiyon bloklarının boyutuna basın FPGA temel mantık yapı taşlarıdır. Mantık fonksiyon bloklarının farklı boyutlarına göre, FPGA' lar iki türe ayrılabilir: ince taneli yapı ve kaba taneli yapı (Lortoğlu, 2019).

İnce taneli FPGA' ların mantık fonksiyon blokları genellikle küçüktür, sadece birkaç transistörden oluşur ve yarı özel bir kapı dizisinin temel birimine çok benzer. Avantajı, fonksiyon bloğunun kaynaklarının tam olarak kullanılabilmesidir, ancak dezavantajı karmaşık mantığı tamamlamasıdır. Fonksiyon çok sayıda bağlantı ve anahtar gerektirir, bu nedenle hız yavaştır (Lortoğlu, 2019).

Kaba taneli FPGA mantık bloğu ölçek olarak büyük ve güçlüdür ve yalnızca daha az fonksiyon bloğu ve dahili kablolamaya ihtiyaç duyar karmaşık mantığı tamamlar, böylece daha iyi performans elde edebilir. Dezavantajı ise işlevsel blokların kaynaklarından bazen tam olarak yararlanılamamasıdır. Son yıllarda

teknolojinin sürekli iyileştirilmesi ve FPGA entegrasyonunun sürekli geliştirilmesi ile birlikte donanım tanımlama dili (HDL) tasarım yöntemi yaygın olarak kullanılmaktadır. Mantıksal sentez araçlarının çoğu geçit dizisi yapısı için geliştirildiğinden, ince taneli FPGA' lar kaba taneli FPGA' lara göre daha iyi mantıksal sentez sonuçları elde edebilir. Bu nedenle, birçok üretici daha yüksek entegrasyona sahip bazı ince taneli FPGA' lar geliştirmiştir (Lortoğlu, 2019).

2.3.3.2 Ara bağlantı yapısı sınıflandırma

Ara bağlantı yapısı sınıflandırmasına göre FPGA içindeki farklı kablolama yapısına göre, bölümlü ara bağlantı tipi ve sürekli ara bağlantı tipi olarak ikiye ayrılabilir. Segmentli ara bağlantı FPGA' da farklı uzunluklarda birden fazla metal hat vardır ve her bir metal hat segmenti bir anahtar matrisi veya anti-sigorta programıyla bağlanır. Bu tür bir kablolama yapısı esnektir ve çeşitli uygulanabilir çözümlere sahiptir, ancak kablolama gecikmesi yerleşimin belirli işleme süreciyle ilgilidir ve tasarım tamamlanmadan önce tahmin edilemez ve tasarım değişikliği gecikme performansının değişmesine neden olacaktır (Akchurin, 2016). Sürekli ara bağlantı FPGA'ları, mantık fonksiyon blokları arasındaki ara bağlantıyı gerçekleştirmek için genellikle tüm yonga boyunca uzanan uzun kablolar olan aynı uzunlukta metal teller kullanır ve bağlantının mesafe ile hiçbir ilgisi yoktur. Bu kablolama yapısında, mantık hücrelerinin farklı yerlerdeki bağlantı hatları belirlenir, böylece kablolama gecikmesi sabit ve öngörülebilirdir (Lortoğlu, 2019).

2.3.4 FPGA'nın Programlama özellikleri

Programlanabilir aygıtlar sinyal işleme, ses, kontrol, bilgi sistemleri, düzenleme alanlarında kullanılmaktadır. FPGA'lara birçok program yerleştirilerek çalıştırılabilmektedir (Yılmaz, 2008). Programlama özelliklerine göre sınıflandırma kullanılan farklı anahtarlama elemanlarına göre, FPGA'lar iki türe ayrılabilir: tek seferlik programlama tipi ve yeniden programlanabilir tip (Alçin, 2011).

2.3.4.1 Tek seferlik programlama

Tek seferlik programlama FPGA, sigorta önleyici anahtarlama elemanlarını kullanır. İşlem teknolojisi, bu cihazın küçük boyut, yüksek entegrasyon, düşük ara bağlantı karakteristik empedansı ve daha yüksek hız avantajlarına sahip olduğunu belirler. Ayrıca, özellikleri de vardır. Şifreleme biti, kopyalama önleme, radyasyon önleme ve parazit önleme gibi ve PROM (Programlanabilir Salt Okunur Bellek) veya EPROM'u(Silinebilir programlanabilir salt okunur bellek) başlamanıza gerek yoktur. Ancak, tasarım verileri çipe yazıldıktan sonra yalnızca bir kez programlanabilir, tasarım artık değiştirilemez, bu nedenle nihai ürünler ve toplu uygulamalar için daha uygundur (Alçin, 2011).

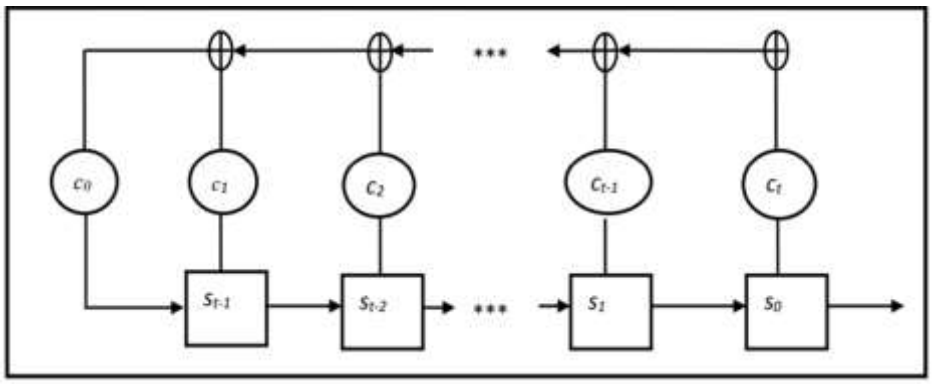
2.3.4.2 Yeniden programlama

Yeniden programlanabilir FPGA, SRAM anahtarlama elemanlarını veya flaş EPROM kontrollü anahtarlama elemanlarını kullanır. FPGA yongasında, her mantık bloğunun işlevi ve bunlar arasındaki ara bağlantı modu, yongadaki SRAM veya flash EPROM' da depolanan veriler tarafından belirlenir. SRAM tipi anahtarın FPGA' sı geçicidir ve gücün yeniden uygulandığı her seferde, FPGA yapılandırma verilerini yeniden yüklemelidir. SRAM FPGA' nın olağanüstü avantajı, tekrar tekrar programlana bilmesidir. Sistem açıldığında, farklı donanım işlevlerini tamamlamak için FPGA' ya farklı yapılandırma verileri yükleyin. Bu konfigürasyon değişikliği, sistem işlevlerinin dinamik olarak yeniden yapılandırılmasını sağlamak için sistemin çalışması sırasında bile gerçekleştirilebilir. Flaş EPROM kontrol anahtarları kullanan FPGA'lar, geçici olmayan ve tekrarlanabilir programlamanın ikili avantajlarına sahiptir, ancak yeniden programlamada SRAM tabanlı FPGA'lardan daha az esnektirler ve dinamik yeniden yapılandırma gerçekleştiremezler. Ayrıca statik güç tüketimi, anti-sigorta ve SRAM FPGA'lardan daha fazladır (Alçin, 2011).

3. TAUSWORTHE YÖNTEMİ

Bu çalışmada 1965 yılında kabul edilen Tausworth'e yöntemini temel alıp FPGA ile rastgele sayı üreteci tasarımını gerçekleştirilmiştir. Şifreleme metotlarıyla ilgili olan bu yöntemde rastgele sayılar art arda gelen sayı çiftlerinin tekrarlanmasıyla üretilir (Tausworthe, 1965). Ayrıca Doğrusal Geri Besleme Kaydırma Yazmacı (Linear Feedback Shift Registers (LFSR) rastgele sayı üreteçleri Tausworthe üreteçleri olarak adlandırılır (L'ecuyer, 1999).

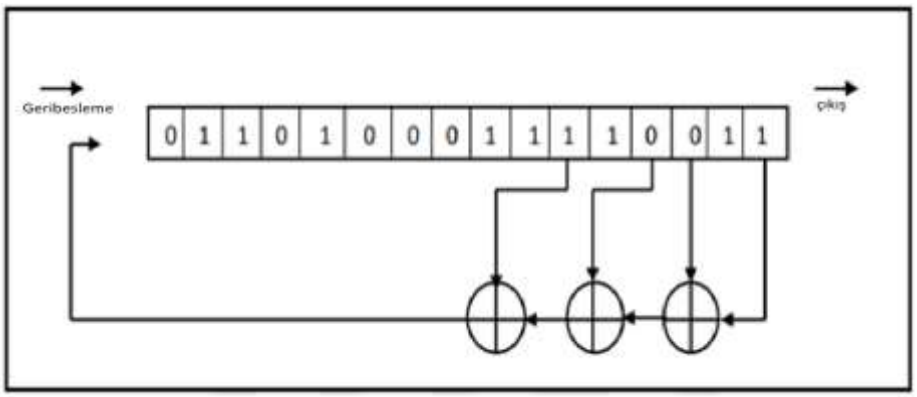
Doğrusal Geri Besleme Kaydırma Yazmacı bit üretecisidir. Verimli ve çok hızlı olabilmesi için kolay bir donanım uygulaması düşünülerek tasarlanmıştır. Uzun bir sözde rastgele dizi elde etmenin kolay bir yolu, kaydırma yazmaç sayısını (LFSR) artırmaktır. LFSR'nin donanımda uygulanması kolaydır. Doğrusal geri besleme kaydırma yazmacı Modern iletişim ve bilgisayar sistemlerinde güvenlik önlemleri çok önemli bir faktördür. LFSR' ye ait Şekil 3.5'de verilmiştir (Dilan, 2017).



Şekil 3.1 Genel LFSR yapısı (Dilan, 2017)

Şekil 3.5'de yer alan c harfleri ise geri besleme ifadesine ait gerekli katsayıları belirtirken, s harfleri LFSR 'ye ait hafıza birimlerini belirtir. Burada Yazıcı l bitlik olup c0 katsayısı 1 alınır. LFSR'nin ilk başlangıç haline tohum denilir. LFSR'nin işlevi düzenli olduğundan üretilen bit dizisi bir önceki bit dizisine bağlıdır. Nitelikli seçilen bir LFSR yapısı rastgelelik olasılığı güçlü olan ve kendini uzun süre sonra tekrarlayan bit dizisi üretebilir (Dilan, 2017).

LFSR yapısındaki yazmaçlarda katsayıların sadece 0 veya 1 değerlerinden birine ait olabilir.



Şekil 3.2 LFSR' ye örnek (Dilan, 2017).

Şekil 3.6'da gösterildiği gibi LFSR yapısında geri besleme fonksiyonunu oluşturan bitler 11., 13., 14., 16. bitlerdir. Geri besleme fonksiyonunun eşitlik ifadesini $G(x) = x^{11} + x^{13} + x^{14} + x^{16} + 1 \pmod 2$ şeklinde ifade edile bilir. LFSR sonucu çıkan rastgele bit dizisinin uzun periyotlu olması gerekir (Dilan, 2017).

Doğrusal geri besleme kaydırma yazmacının herhangi bir m sırası için, ilkel polinomiye göre, devre bağlantısı 2^m -1. M-dizisi adı verilen bir maksimum uzunluk dizisi oluşturabilir . Her m için, her zaman en az sayıda terime sahip bir polinom ve katsayısı 1 olan her terim olacaktır. Bu tür polinomlara ilkel polinomlar denir. İlkel polinomu temel alan devre en basit olanıdır. LFSR 'lar birçok anahtar dizisi üretecinde kullanılmaktadır. Bunun nedeni olarak donanımsal uygulamalarda uygunlukları, geniş periyoda sahip olmaları, üretilen serinin iyi istatistiksel özellikler göstermesi ve cebirsel tekniklerle kolayca analiz edilebilmeleri gösterilebilir (Tausworthe, 1965; L'ecuyer, 1999).

Doğrusal geri besleme kaydırma yazmacı önceki durumun çıktısını ifade eder, çıkışın doğrusal işlevi giriş kaydırma yazmacı olarak yeniden kullanılır . XOR işlemi, en yaygın tek bitli doğrusal fonksiyondur. Bir LFSR, birkaç depolama öğesinden ve bir geri bildirim yolundan oluşur (Tausworthe, 1965).
Tausworthe yöntemi aşağıdaki gibidir.

$$b_i = \left(c_1 b_{i-1} + c_2 b_{i-2} + \cdots + c_q b_{i-q}\right) \bmod 2 \quad (3.1)$$

Denklemde c_q katsayılarının en fazla 2 tanesi sıfırdan farklı olabillir. Bu nedenle denklem en basit haliyle;

$$b_i = \left(b_{i-r} + b_{i-q}\right) \bmod 2 \quad (3.2)$$

olarak yazılır. Burada r ve q tamsayı ve 0 < r < q olmalıdır.

$$b_i = \begin{cases} 1, & b_{i-r} = b_{i-q} \\ 0, & b_{i-r} \neq b_{i-q} \end{cases} \quad (3.3)$$

Giriş değerlerinin 0 veya 1 olması durumunda (3) numaralı eşitlik denklemin sonucunun XOR lojik kapısının doğruluk tablosuyla aynı olduğunu göstermektedir. Tausworthe yöntemiyle üretilen rastgele sayıların periyodu $2^q - 1$ şeklinde belirlenir. Uzun periyotlu rastgele sayı dizesi üretmek için q değerini büyük tutmak gerekmektedir. (L'ecuyer, 1999).

Bu yöntemin diğer yöntemlere göre bazı avantajları vardır. Bu yöntemde üretilen sayıların kullanılan bilgisayardan bağımsız ve çok uzun periyotlu sayılar üretmek mümkündür. Ayrıca yöntem tek bir lojik kapı ile gerçekleştirile biliyor. Bu açıdan hızlı bir algoritmaya sahiptir. Dezavantajı ise üretilen değerlerin rastgelelik özelliğinin diğer yöntemlerdeki değerler kadar iyi olmaması. Bu nedenle kullanıldığı alanlar daha basit alanlardır (Tausworthe, 1965; L'ecuyer, 1999).

3.1 LFSR ile Yapılan Çalışmalar

Dereli (2020), yaptığı çalışmada doğrusal geri beslemeli öteleyen kaydedici temelli sözde rastgele sayı üreteci tasarımı gerçekleştirmiştir. Bu rastgele sayı üretecinin bariz farkı ürettiği sayıların "0" ve "1" arasında 32-bitlik hassasiyete sahip kayan noktalı sayılar olmasıdır. O nedenle yapılan çalışmada üretilen sayıların 0.1‟den büyük ve 1.0‟dan küçük olması sağlanmıştır (Dereli, 2020).

Çakır (2012), yaptığı çalışmada akış şifreleme yöntemleri karşılaştırmalı olarak incelemiştir. Akış şifrelemede yaygın olarak kullanılan LFSR ve PRNG tabanlı algoritmaların çalışma prensiplerini araştırmıştır. Şifrelemeyi LFSR tabanlı üreteçler ve PRNG tabanlı üreteçler olarak iki farklı grupta ele almıştır. Çalışmada ilk olarak LFSR tabanlı üreteçlerden; A5/1, A5/2 ve RC4 olmak üzere üç algoritmanı seçerek incelemiş ve daha sonra PRNG tabanlı üreteçlerden BBS, LCG, ICG ve MT olmak üzere dört algoritmanı seçerek incelemiş ve MATLAB ortamında uygulamıştır (Çakır, 2012).

Huang vd. (2010) yaptığı çalışmada Tausworthe mimarisi, Box- Muller ve CORDIC IPcore kullanarak bağlantılı Lognormal dağıtılmış diziyi oluşturmak için donanım mimarisi önermişler. FPGA üzerindeki uygulama 4210 dilim, 4 blok RAM (Random Access Memory) ve 2DSP48S kullanır. Sayısal deney önerilen yöntemin Lognormal dağıtılmış diziyi doğru bir şekilde oluştura bildiğini göstermektedir. Önerilen bu yöntemin radar eko ve dağınıklık simülatörü için kullanılabilir olduğu belirtilmiştir (Huang vd. 2010).

Jianpo (2016) yaptığı çalışmada Gauss beyaz gürültü sinyali kaynağı oluşturmak için bir yöntem önermiştir. Çalışmada uzun dönemli düzgün dağılmış rastgele sayılar üretmek için FPGA'da birleştirilmiş bir Tausworth rastgele sayı üreteci önermiş ve bu birleştirilmiş Tausworthe rastgele sayı üreteci birkaç Tausworth rastgele sayı üretecinden oluşmuştur. Sayı üreteci birleştirilmiş; uzun periyotta tekdüze dağıtılmış rastgele sayı ve Gauss beyaz gürültüsü ile eşlenir; bir dizi filtre sırası Aynı sayıda filtre katsayısından oluşan birden fazla filtre seti, geniş bant Gauss beyaz gürültüsünü, ayarlanabilir parametrelerle bir ara frekanslı gürültü kaynağına dönüştürmüştür. Mevcut buluşu uygulayan bir Gauss beyaz gürültü sinyali kaynağı oluşturmaya yönelik yöntem ve cihaz, aşağıdaki faydalı etkilere sahiptir: donanım kaynaklarını koruyabildiğini ve FPGA'nın düzgün dağılmış rastgele sayıları Gauss beyaz gürültüsüne dönüştürmesini gerçekleştirmesini kolaylaştırabildiği söylenmiştir. (Jianpo, 2016).

L’Ecuyyer yaptığı çalışmada iyileştirilmiş istatistiksel özellikler elde etmek için üç LFSR tabanlı URNG’yi birleştirerek saat başına 32 bitlik tek tip rastgele sayı üretmiş ve bu büyük bir periyoda sahip olduğu görülmüştür (L’Ecuyer, 1996).

Tezukaç (1993) yaptığı çalışmada uzun dönemli rastgele sayı üreteçlerinin çoğunu açıklamıştır. Çalışmada üç tip uzun dönemli rastgele sayı üreticiden bahs etmiştir. Bu üreteçler- Kaydırma yazmaç üreteçleri, Twisted-GFSR (Global Financial Stability Report, Küresel Finansal İstikrar Raporu) üreteçleri, ve birleşik Tausworthe jeneratörleri olarak sınıflandırılmıştır. Tausworthe jeneratörleri modül 2 açısından Wichmann-Hill üreteçlerine benzediği belirtilmiştir. Çalışmada iki tür kaydırmalı kayıt rastgele sayı dizisi olan: Tausworthe dizileri ve GFSR dizilerinden bahs ederek uzun dönmeli rastgele sayı üreteçlerinden bahs edilmiştir (Tezukaç, 1993).

4. ARAŞTIRMA BULGULARI VE TARTIŞMA

Tausworthe denklemini kullanırken XOR kapısını doğrudan ikili bit içeren negatif olmayan tamsayılar dizisi üzerinde kullanmak daha uygundur (Math**,** 2021). Rastgele bitleri üretmek için ilk başta bir başlangıç bit dizesi tanımlanır. Tausworthe denklemini en az $2^q - 1$ periyotlu olduğundan rastgeleliği vermesi öngörülen denklemin derecesi en az q=15 olması gerekmektedir. Denklemde değeri 1 olan yazmaçların değerleri XOR kapısından geçirerek sisteme geri besleme değeri olarak verilmiş, en yüksek değerli yazmaçtaki değer rastgele sayı dizisini oluşturacak şekilde bit dizisi çıktısı elde edilmiştir. Aşağıda blok yapıları verilen tasarımlar Verilog donanım tanımlama dili ile Vivado HLS v.2020.2 programı ile sentezlenmiştir. Elde edilen sayı dizisi FIPS testlerinden geçirilmiştir. FIPS testlerinin kullanılabilmesi için yirmi bin bitlik bir dizeye ihtiyaç duyulmaktadır.

Yöntemi kullanırken önce XOR kapısını doğrudan ikili bit içeren negatif olmayan tamsayılar dizisi üzerinde kullanmak daha uygundur. Önce XOR kapısını kullanarak on bin bitden oluşan 10 paketten ibaret yüz bin bit üreten algoritma yazılmıştır. Üretilecek dizelerin ilk on değerini Çizelge 4.1`deki gibi belirlenmiştir. Algoritma verilen besleme değerleri ile başlayarak her dizede 10 bin bit üretdi.

Çizelge 4.1 Dize başlanğıç değerleri

Dize adı	**Değerler**	**Dize adı**	**Değerler**
X1	{0, 1, 0, 1, 0, 1, 0, 1, 0, 1…}	X6	{0, 1, 0, 1, 0, 1, 0, 1, 0, 1…}
X2	{0, 1, 0, 1, 0, 1, 0, 1, 0, 1…}	X7	{0, 1, 0, 1, 0, 1, 0, 1, 0, 1…}
X3	{0, 1, 0, 1, 0, 1, 0, 1, 0, 1…}	X8	{0, 1, 0, 1, 0, 1, 0, 1, 0, 1…}
X4	{0, 1, 0, 1, 0, 1, 0, 1, 0, 1…}	X9	{0, 1, 0, 1, 0, 1, 0, 1, 0, 1…}
X5	{0, 1, 0, 1, 0, 1, 0, 1, 0, 1…}	X10	{0, 1, 0, 1, 0, 1, 0, 1, 0, 1…}

Üretilen bit dizelerini test etmek amacıyla her biri 20000 bit içeren 5 blok halinde kaydettik. Kaydedilen `.txt` uzantılı dosyaları MatLab`da FIPS testine

kümesinden geçirilmiştir. Yalnızca XOR kapısı kullanılarak üretilen dizelerin test sonuçları Çizelge 4.2`de gösterilmiştir.

Çizelge 4.2 Tek denklemli XOR kapısı ile üretilen bitlerin FIPS test sonucu

Fıps test	Blok 1	Blok 2	Blok 3	Blok 4	Blok 5	Ortalama değer	Beklenen değer
l_runs0	**6665** **0** **0** **0** **0** **0**	**4001** **1999** **0** **0** **0** **0**	**6000** **1** **666** **0** **0** **0**	**4002** **2665** **0** **0** **0** **0**	**3336** **2664** **667** **1** **0** **0**	**4800** **1465** **266** **0,2** **0** **0**	%1 2267 - 2733 %2 1079 - 1421 %3 502 - 748 %4 223 - 402 %5 90 - 223 %6 90 - 223
l_runs1	**5332** **2** **0** **0** **0** **1333**	**2667** **1999** **669** **0** **666** **0**	**5332** **2** **0** **666** **1** **666**	**3336** **2664** **667** **1** **0** **0**	**4666** **668** **666** **667** **0** **0**	**4266** **1067** **400** **266** **133** **399**	%1 2267 - 2733 %2 1079 - 1421 %3 502 - 748 %4 223 - 402 %5 90 - 223 %6 90 - 223
long_run0	**1**	**2**	**3**	**2**	**2**	**2**	run <=34
long_run1	**9**	**2**	**6**	**4**	**4**	**5**	run <=34
mono	**13336**	**12002**	**12001**	**10669**	**10668**	**11735,2**	9654<mono< 10346
poker	**1.100**	**2.289**	**6.908**	**4.066**	**4.062**	**3,685**	1.03<X<57.4

Her biri 20000 bit içeren 5 blok halinde Tek denklemli XOR kapısı ile üretilen bitlerde blok 1, blok 3, blok 4, blok 5' de testlerden hiç biri beklenen değer aralığında değildir. Yalnız blok 2'de koşu testinin 2.kademesi beklenen değer aralığındadır. Test sonuçların olumlu olmamasının ardından karıştırma algoritması kullanıldı. Dizelerden belirli bitleri alarak 20 bin bitlik bloklar oluşturduk ve testleri yeniden gerçekleştirildi. n=2 durumu için dizelerden sadece çift sayılı indisli elemanları aldık, ardından tek sayılı indisli elemanları dizenin sonuna eklendi. Oluşturulan blokların test sonuçları Çizelge 4.3`de gösterilmiştir.

Çizelge 4.3 Karıştırma algoritması ile üretilen bitlerin FIPS test sonucu

Fıps test	Blok 1	Blok 2	Blok 3	Blok 4	Blok 5	Ortalama değer	Beklenen değer
l_runs0	7 0 0 1 1332 0	3336 1333 667 0 0 0	671 0 1 2 667 665	4000 667 1332 1 0 0	2669 666 666 0 666 1	2136 533 533 0,8 533 133	%1 2267 - 2733 %2 1079 - 1421 %3 502 - 748 %4 223 - 402 %5 90 - 223 %6 90 - 223
l_runs1	8 2 1330 1 0 1332	3337 666 1 667 2 664	9 1 667 1330 0 666	3999 669 2 1331 0 0	1337 2000 0 1331 1 0	1738 667 400 932 0,6 532	%1 2267 - 2733 %2 1079 - 1421 %3 502 - 748 %4 223 - 402 %5 90 - 223 %6 90 - 223
long_run0	5	3	6	4	6	4,8	run <=34
long_run1	11	7	10	4	5	7,4	run <=34
mono	13330	11998	11994	10667	10666	11731	9654<mono <10346
poker	1.751	1.751	7.591	1.570	1.216	2.744	1.03<X<57.4

Karıştırma algoritmasında 20 bin bitlik 5 blokdan oluşturulan blokların test sonuçları beklenen aralıkta olmadı. Sonra tek denklemli yöntemler ve ardından da ikili denklemli yöntemler uygulanmıştır. Tek denklemli yöntemde q=7, q=8, q=9, q=10, q=11, q=12, q=13, q=14, q=15' e kadar değerlerin her biri test edilmiştir. Sonuçlar Çizelge 4.5, 4.7, 4.9, 4.11, 4.13, 4.15, 4.17, 4.19, 4.21' de verilmiştir.

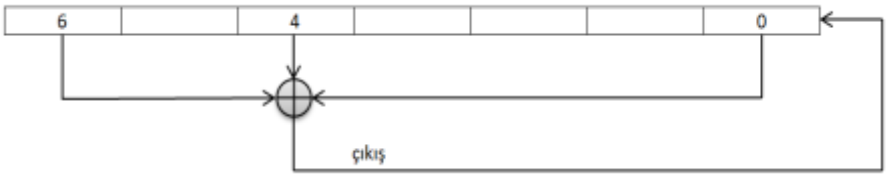
$X^7 + X^5 + 1$ denklemi için başlangıç bit dizisi çizelge 4.4'deki gibidir.

Çizelge 4.4 Tausworthe denkleminde q=7 için kullanılan denklem ve başlangıç dizesi

Kullanılan denklem	Başlangıç dizesi
X^7+X^5+1	[0101010]

X^7+X^5+1 denklemine dayanan LFSR devresi yapısı Şekil 4.1'deki gibidir.

Burada sıfırıncı yazmaç denklemdeki +1'i, dördüncü yazmaç denklemdeki X^5 i ve altıncı yazmaç denklemdeki X^7 i temsil etmektedir. Bu üç yazmaçtaki verinin (1 veya 0) XOR devresinden geçirilmesiyle oluşan sonuç sıfırıncı yazmaca geri besleme verisi olarak yazılır. Bu yazmaçtaki veri bir sonraki yazmaca aktarılır. XOR devresinden geçirilmesiyle oluşan bir aynı samanda çıkış verisini de oluşturmaktadır.



Şekil 4.1 X^7+X^5+1 denklemini gerçekleyen LFSR şeması

X^7+X^5+1 denkleminden elde edilen bit dizisinin FIPS test sonucu Çizelge 4.5'de verilmiştir. Çizelgeden görüleceği gibi koşu 0 ve koşu 1 testinin hemen hemen bütün kademelerinden kalmıştır. Sadece koşu 0 ve koşu 1 testinin 2.kademesinde beklenen değer aralığında test sonucu başarılı olmuştur. Ayrıca monobit testinden de kalmıştır. Uzun koşu 0, uzun koşu 1 ve poker testi beklenen değer aralığındadır. Bu Sonuçlardan anlaşılmaktadır ki q=7 olacak şekilde gerçeklenen tekli LFSR yapısı ile yeterli rastgelelikle bit dizisi üretilememiştir.

Çizelge 4.5 q=7 Durumu için FIPS test sonucu

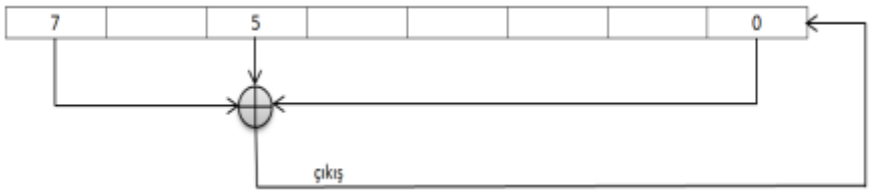
FIPS test	Tausworthe yöntemi	Beklenen değer
Koşu 0	**5335** 1333 **0** **0** **0** **0**	%1 2267 - 2733 %2 1079 - 1421 %3 502 - 748 %4 223 - 402 %5 90 - 223 %6 90 - 223
Koşu 1	**4002** 1333 **0** **1333** **0** **0**	%1 2267 - 2733 %2 1079 - 1421 %3 502 - 748 %4 223 - 402 %5 90 - 223 %6 90 - 223
Uzun koşu 0	2	<=34
Uzun koşu 1	4	<=34
Monobit	**12000**	9654 < X< 10346
Poker	3.89	1.03<X<57.4

X^8+X^6+1 denklemi için başlangıç bit dizisi Çizelge 4.6'daki gibidir.

Çizelge 4.6 Tausworthe denkleminde q=8 için kullanılan denklem ve başlangıç dizesi

Kullanılan denklem	Başlangıç dizesi
X^8+X^6+1	[01010101]

X^8+X^6+1 denklemini kullanan LFSR yapısı Şekil 4.2'deki gibidir



Şekil 4.2 X^8+X^6+1 denklemini gerçekleyen LFSR şeması

$X^8 + X^6 + 1$ denklemiyle oluşturulan bit dizisinin FIPS test sonucu Çizelge 4.7'de verilmiştir. Çizelgeden görüleceği gibi fips testlerinin bütün testlerinden kalmıştır. q=8 olarak seçilen denklemden de yeterli rastgelelik sağlanamamıştır.

Çizelge 4.7 q=8 Durumu için FIPS test sonucu

FIPS test	Tausworthe yöntemi	Beklenen değer
Koşu 0	**10000** **0** **0** **0** **0** **0**	%1 2267 - 2733 %2 1079 - 1421 %3 502 - 748 %4 223 - 402 %5 90 - 223 %6 90 - 223
Koşu 1	**10001** **0** **0** **0** **0** **0**	%1 2267 - 2733 %2 1079 - 1421 %3 502 - 748 %4 223 - 402 %5 90 - 223 %6 90 - 223
Uzun koşu 0	**1**	<=34
Uzun koşu 1	**1**	<=34
Monobit	**10001**	9654 < X< 10346
Poker	**75000**	1.03<X<57.4

$X^9 + X^7 + 1$ denklemi için başlangıç bit dizisi Çizelge 4.8' deki gibidir.

Çizelge 4.8 Tausworthe denkleminde q=9 için kullanılan denklem ve başlangıç dizesi

Kullanılan denklem	Başlangıç dizesi
$X^9 + X^7 + 1$	[010101010]

$X^9 + X^7 + 1$ denklemini kullanan LFSR yapısı Şekil 4.3'deki gibidir

Şekil 4.3 $X^9 + X^7 + 1$ denklemini gerçekleyen LFSR şeması

$X^9 + X^6 + 1$ denklemiyle oluşturulan bit dizisinin FIPS test sonucu Çizelge 4.9'da verilmiştir. Çizelgeden görüleceği koşu 1 testinin tüm kademlerinden kalmış koşu 0 testinde sadece 3.kademesinden geçmiştir. q=9 olarak seçilen denklemden de yeterli rastgelelik sağlanamamıştır.

Çizelge 4.9 q=9 Durumu için FIPS test sonucu

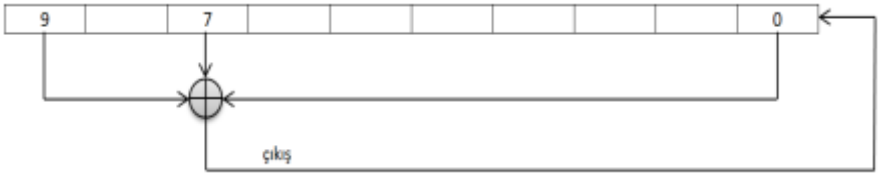
FIPS test	Tausworthe yöntemi	Beklenen değer
Koşu 0	**2825** **1881** 471 **0** **236** **235**	%1 2267 - 2733 %2 1079 - 1421 %3 502 - 748 %4 223 - 402 %5 90 - 223 %6 90 - 223
Koşu 1	**3767** **940** **235** **471** **235** **0**	%1 2267 - 2733 %2 1079 - 1421 %3 502 - 748 %4 223 - 402 %5 90 - 223 %6 90 - 223
Uzun koşu 0	6	<=34
Uzun koşu 1	5	<=34
Monobit	9411	9654 < X< 10346
Poker	5.88	1.03<X<57.4

$X^{10} + X^8 + 1$ denklemi için başlangıç bit dizisi Çizelge 4.10'daki gibidir.

Çizelge 4.10 Tausworthe denkleminde q=10 için kullanılan denklem ve başlangıç dizesi

Kullanılan denklem	Başlangıç dizesi
$X^{10} + X^8 + 1$	[0101010101]

$X^{10} + X^8 + 1$ denklemini kullanan LFSR yapısı Şekil 4.4'de verilmiştir.



Şekil 4.4 $X^{10}+X^{8}+1$ denklemini gerçekleyen LFSR şeması

$X^{10}+X^{8}+1$ durumu için FIPS test sonucu Çizelge 4.11'de verilmiştir. Çizelgeden görüleceği gibi fips testlerinin bütün testlerinden kalmıştır. Bu durumda da yeterli rastgelelik elde edilememiştir.

Çizelge 4.11 q=10 Durumu için FIPS test sonucu

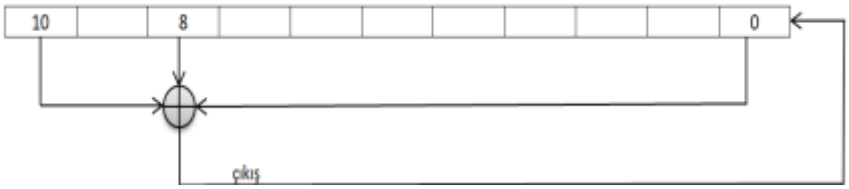
FIPS test	Tausworthe yöntemi	Beklenen değer
Koşu 0	**10000** **0** **0** **0** **0** **0**	%1 2267 - 2733 %2 1079 - 1421 %3 502 - 748 %4 223 - 402 %5 90 - 223 %6 90 - 223
Koşu 1	**10001** **0** **0** **0** **0** **0**	%1 2267 - 2733 %2 1079 - 1421 %3 502 - 748 %4 223 - 402 %5 90 - 223 %6 90 - 223
Uzun koşu 0	**1**	<=34
Uzun koşu 1	**1**	<=34
Monobit	**10001**	9654 < X< 10346
Poker	75000	1.03<X<57.4

$X^{11}+X^{9}+1$ denklemi için başlangıç bit dizisi Çizelge 4.12'deki gibidir.

Çizelge 4.12 Tausworthe denkleminde q=11 için kullanılan denklem ve başlangıç dizesi

Kullanılan denklem	Başlangıç dizesi
$X^{11}+X^{9}+1$	[01010101010]

$X^{11}+X^{9}+1$ denklemini kullanan LFSR yapısı Şekil 4.5'teki gibidir.



Şekil 4.5 $X^{11}+X^{9}+1$ denklemini gerçekleyen LFSR şeması

$X^{11}+X^{9}+1$ durumu için FIPS test sonucu Çizelge 4.13'de verilmiştir. Çizelgeden görüleceği gibi fips testlerinin bütün kademelerinden geçdiği halde yalnız poker testinden kalmış. Bu durumda da yeterli rastgelelik elde edilememiştir.

Çizelge 4.13 q=11 Durumu için FIPS test sonucu

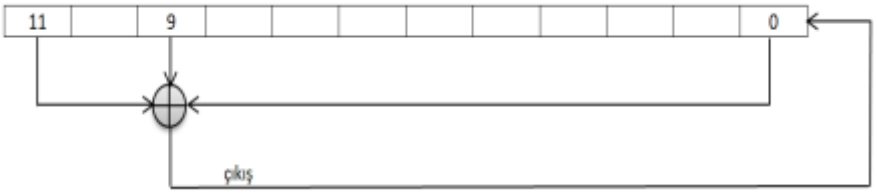
FIPS test	Tausworthe yöntemi	Beklenen değer
Koşu 0	2513 1253 642 314 158 158	%1 2267 - 2733 %2 1079 - 1421 %3 502 - 748 %4 223 - 402 %5 90 - 223 %6 90 - 223
Koşu 1	2516 1273 625 313 156 156	%1 2267 - 2733 %2 1079 - 1421 %3 502 - 748 %4 223 - 402 %5 90 - 223 %6 90 - 223
Uzun koşu 0	10	<=34
Uzun koşu 1	9	<=34
Monobit	9984	9654 < X< 10346
Poker	**0.54**	1.03<X<57.4

$X^{12}+X^{10}+1$ denklemi için başlangıç bit dizisi Çizelge 4.14'teki gibidir.

Çizelge 4.14 Tausworthe denkleminde q=12 için kullanılan denklem ve başlangıç dizesi

Kullanılan denklem	Başlangıç dizesi
$X^{12}+X^{10}+1$	[010101010101]

$X^{12}+X^{10}+1$ denklemini kullanan LFSR yapısı Şekil 4.6'daki gibidir.



Şekil 4.6 $X^{12}+X^{10}+1$ denklemini gerçekleyen LFSR şeması

$X^{12}+X^{10}+1$ durumu için FIPS test sonucu Çizelge 4.15'de verilmiştir. Çizelgeden görüleceği gibi fips testlerinin uzun koşu 0 ve uzun koşu 1 kademelerinde testi geçmiş kalan bütün kademelerinde testten kalmıştır. Bu durumda da yeterli rastgelelik elde edilememiştir.

Çizelge 4.15 q=12 Durumu için FIPS test sonucu

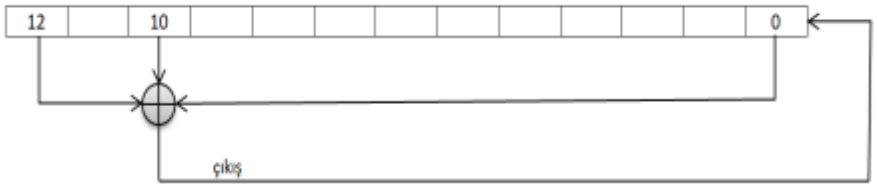
FIPS test	Tausworthe yöntemi	Beklenen değer
Koşu 0	**10000** **0** **0** **0** **0** **0**	%1 2267 - 2733 %2 1079 - 1421 %3 502 - 748 %4 223 - 402 %5 90 - 223 %6 90 - 223
Koşu 1	**10001** **0** **0** **0** **0** **0**	%1 2267 - 2733 %2 1079 - 1421 %3 502 - 748 %4 223 - 402 %5 90 - 223 %6 90 - 223
Uzun koşu 0	**1**	<=34
Uzun koşu 1	**1**	<=34
Monobit	**10001**	9654 < X< 10346
Poker	**75000**	1.03<X<57.4

$X^{13}+X^{11}+1$ denklemi için başlangıç bit dizisi Çizelge 4.16'taki gibidir.

Çizelge 4.16 Tausworthe denkleminde q=13 için kullanılan denklem ve başlangıç dizesi

Kullanılan denklem	Başlangıç dizesi
$X^{13}+X^{11}+1$	[0101010101010]

$X^{13}+X^{11}+1$ denklemini kullanan LFSR yapısı Şekil 4.7'deki gibidir.



Şekil 4.7 $X^{13}+X^{11}+1$ denklemini gerçekleyen LFSR şeması

$X^{13}+X^{11}+1$ durumu için FIPS test sonucu Çizelge 4.17'de verilmiştir. Çizelgeden görüleceği gibi fips testlerinin uzun koşu 0 ve uzun koşu 1 kademelerinde testi geçmiş kalan bütün kademelerinde testten kalmıştır. Bu durumda da yeterli rastgelelik elde edilememiştir.

Çizelge 4.17 q=13 Durumu için FIPS test sonucu

FIPS test	Tausworthe yöntemi	Beklenen değer
Koşu 0	**10000** **0** **0** **0** **0** **0**	%1 2267 - 2733 %2 1079 - 1421 %3 502 - 748 %4 223 - 402 %5 90 - 223 %6 90 - 223
Koşu 1	**10001** **0** **0** **0** **0** **0**	%1 2267 - 2733 %2 1079 - 1421 %3 502 - 748 %4 223 - 402 %5 90 - 223 %6 90 - 223
Uzun koşu 0	**1**	<=34
Uzun koşu 1	**1**	<=34
Monobit	**10001**	9654 < X< 10346
Poker	**75000**	1.03<X<57.4

$X^{14}+X^{12}+1$ denklemi için başlangıç bit dizisi Çizelge 4.18'teki gibidir.

Çizelge 4.18 Tausworthe denkleminde q=14 için kullanılan denklem ve başlangıç dizesi

Kullanılan denklem	Başlangıç dizesi
$X^{14}+X^{12}+1$	[01010101010101]

$X^{14}+X^{12}+1$ denklemini kullanan LFSR yapısı Şekil 4.8'teki gibidir.

Şekil 4.8. $X^{14}+X^{12}+1$ denklemini gerçekleyen LFSR şeması

$X^{14}+X^{12}+1$ durumu için FIPS test sonucu Çizelge 4.19'da verilmiştir. Çizelgeden görüleceği gibi fips testlerinin uzun koşu 0 ve uzun koşu 1 kademelerinde testi geçmiş kalan bütün kademelerinde testten kalmıştır. Bu durumda da yeterli rastgelelik elde edilememiştir.

Çizelge 4.19 q=14 Durumu için FIPS test sonucu

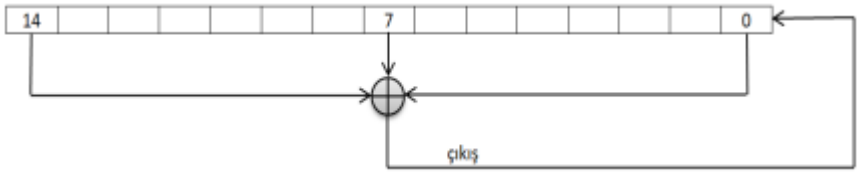
FIPS test	Tausworthe yöntemi	Beklenen değer
Koşu 0	**10000** **0** **0** **0** **0** **0**	%1 2267 - 2733 %2 1079 - 1421 %3 502 - 748 %4 223 - 402 %5 90 - 223 %6 90 - 223
Koşu 1	**10001** **0** **0** **0** **0** **0**	%1 2267 - 2733 %2 1079 - 1421 %3 502 - 748 %4 223 - 402 %5 90 - 223 %6 90 - 223
Uzun koşu 0	**1**	<=34
Uzun koşu 1	**1**	<=34
Monobit	**10001**	9654 < X< 10346
Poker	**75000**	1.03<X<57.4

$X^{15}+X^{13}+1$ denklemi için başlangıç bit dizisi Çizelge 4.20' deki gibidir.

Çizelge 4.20 Tausworthe denkleminde q=15 için kullanılan denklem ve başlangıç dizesi

Kullanılan denklem	Başlangıç dizesi
$X^{15}+X^{13}+1$	[010101010101010]

$X^{15}+X^{13}+1$ denklemini kullanan LFSR yapısı Şekil 4.9' da ki gibidir.



Şekil 4.9 $X^{15}+X^{3}+1$ denklemini gerçekleyen LFSR şeması

$X^{15}+X^{13}+1$ durumu için FIPS test sonucu Çizelge 4.21'de verilmiştir. Çizelgeden görüleceği gibi uzun koşu 0, uzun koşu 1 ve poker testi geçmiştir. Koşu o testinin 4 ve 5.kademesinde, koşu 1 testinin 3.kademesinden testten geçmiş diğer kademelerinde kalmıştır. Ayrıca mono bit testi de testi geçememiştir. Bu durumda da yeterli rastgelelik elde edilememiştir.

Çizelge 4.21 q=15 Durumu için FIPS test sonucu

FIPS test	Tausworthe yöntemi	Beklenen değer
Koşu 0	**5036** **944** **158** **316** **158** **0**	%1 2267 - 2733 %2 1079 - 1421 %3 502 - 748 %4 223 - 402 %5 90 - 223 %6 90 - 223
Koşu 1	**4093** **1576** **472** **472** **0** **0**	%1 2267 - 2733 %2 1079 - 1421 %3 502 - 748 %4 223 - 402 %5 90 - 223 %6 90 - 223
Uzun koşu 0	**5**	<=34
Uzun koşu 1	**4**	<=34

Monobit	**10549**	9654 < X< 10346
Poker	**2.41**	1.03<X<57.4

Tausworthe denkleminde en fazla iki katsayı sıfırdan farklı olması öngördüğünden ve aynı denklemin çıktısını kendine girdi olarak almasından dolayı LFSR yapısı yeterince rastgele sayı üretememektedir. Rastgeleliği artırmak için Tausworthe denkleminde ikiden fazla katsayının sıfır seçilmesi ve/veya birden fazla LFSR yapısının eşzamanlı olarak kullanılması gibi tasarım kullanılmıştır (Arathy, 2018).

Bu çalışmada da rastgeleliği artırmak için birden fazla Tausworthe denklemi eşzamanlı olarak kullanılmıştır. İki denklemli yöntemler uygulanmış, litaratürde olan eşzamanlı doğrusal geri beslemeli kaydırmalı yazmaç (LFSR) yapısı ile ve önerilen yöntem ile elde edilen bit dizelerinin rastgelelik analizleri yapılmıştır. Literatürde olan doğrusal geri besleme yerine geri besleme yapısı her bir denklem diğerini besleyecek şekilde çapraz olarak değiştirilmiştir. Çünkü LFSR yapısı aynı denklem çıktısını kendine geri besleme olarak aldığından sistemde bir periyodiklik oluşmakta, bu da çıktıların rastgele olmamasına neden olmaktadır. Çapraz besleme kullanılarak LFSR yapısının doğasındaki periyodikliğin kırılması amaçlanmıştır.

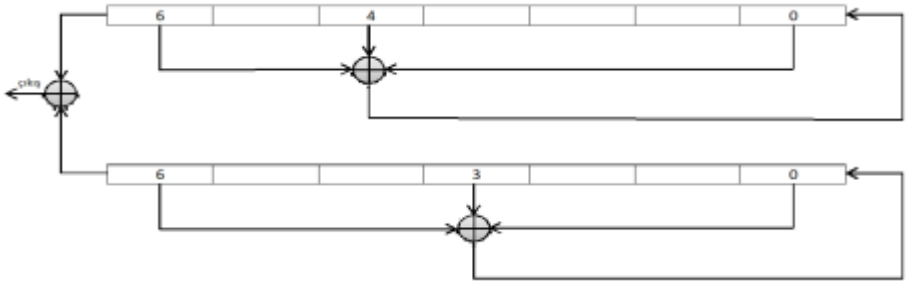
Oluşan bit dizilerinde rastgeleliği artırmak için eş kuvvetli iki farklı denklemi sağlayan kaydırmalı yazmaç yapıları eşzamanlı kullanılmıştır. Her bir denklemi sağlayan FSR yapılarının çıktısı XOR kapısından geçirilerek bit dizisi elde edilmiş, bu dizinin rastgeleliği incelenmiştir. Yapılan tasarımlarda kullanılan ikincil denklemlerin dereceleri birincileri ile aynı seçilmiştir. Burada 'q' Tausworthe denkleminin derecesini belirtmek üzere q=7, q=9 ve q=15 için uygulamalar gerçekleştirilmiştir.

Çizelge 4.22'te q=7 için kullanılan denklemler ve başlangıç dizeleri verilmiştir.

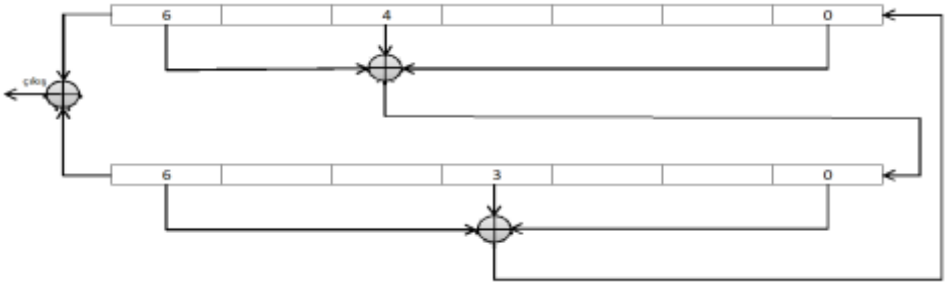
Çizelge 4.22 Tausworthe denkleminde q=7 İçin kullanılan denklemeler ve başlangıç dizesi

Kullanılan denklem	Başlangıç dizesi
X^7+X^5+1	[0101010]
X^7+X^4+1	[0101010]

Bit üretimi için kullanılan eşzamanlı LFSR devresinin blok yapısı Şekil 4.10'da gösterildiği gibidir. Önerilen çapraz beslemeli devrenin blok yapısı Şekil 4.11'de gösterilmiştir.



Şekil 4.10 q=7 için kullanılan doğrusal geri beslemeli rastgele sayı üreteci şeması



Şekil 4.11 q=7 için kullanılan çapraz beslemeli rastgele sayı üreteci şeması

X^7+X^5+1 ve X^7+X^4+1 denklemi için çapraz beslemeli ve doğrusal geri beslemeli yapılardan elde edilen bit dizilerinin FIPS test sonucu Çizelge 4.23'deki gibidir. Çapraz beslemeli yapıdan elde edilen bitler FIPS testlerinin hepsinden geçerken doğrusal geri beslemeli yapıdan elde edilen bitlerin poker testinden geçemediği gözlemlenmiştir.

Çizelge 4.23 q=7 Durumu için FIPS test sonucu

FIPS test	Doğrusal geri besleme	Çapraz besleme	Beklenen değer
Koşu 0	2537 1247 602 302 172 172	2541 1243 614 318 165 160	%1 2267 - 2733 %2 1079 - 1421 %3 502 - 748 %4 223 - 402 %5 90 - 223 %6 90 - 223
Koşu 1	2496 1290 645 301 129 172	2517 1262 636 305 154 163	%1 2267 - 2733 %2 1079 - 1421 %3 502 - 748 %4 223 - 402 %5 90 - 223 %6 90 - 223
Uzun koşu 0	8	12	<=34
Uzun koşu 1	9	13	<=34
Monobit	9978	9996	9654 < X< 10346
Poker	0.30	5.13	1.03<X<57.4

Çizelge 4.23'deki sonucu veren verilog kodu Şekil 4.12 da verilmiştir.

```
.B1(b1),
.B2(b2),
.B3(b3),
.A1(a1),
.A2(a2),
.A3(a3),
.Out3(out3),
.Out2(out2),
.Out1(out1)
);

initial begin


b[0]=1; b[1]=1; b[2]=1; b[3]=0;
b[4]=1; b[5]=0; b[6]=1;

a[0]=1; a[1]=1; a[2]=1; a[3]=0;
a[4]=1; a[5]=1; a[6]=1;

for(n=7; n<=20020; n=n+1) begin

b1=b[n-7];
b2=b[n-3];

b3=b[n-1];
a1=a[n-7];
a2=a[n-4];
a3=a[n-1];

#2;

b[n]=out2;
a[n]=out1;
#2;

Rand[n-7]=out3;
end
```

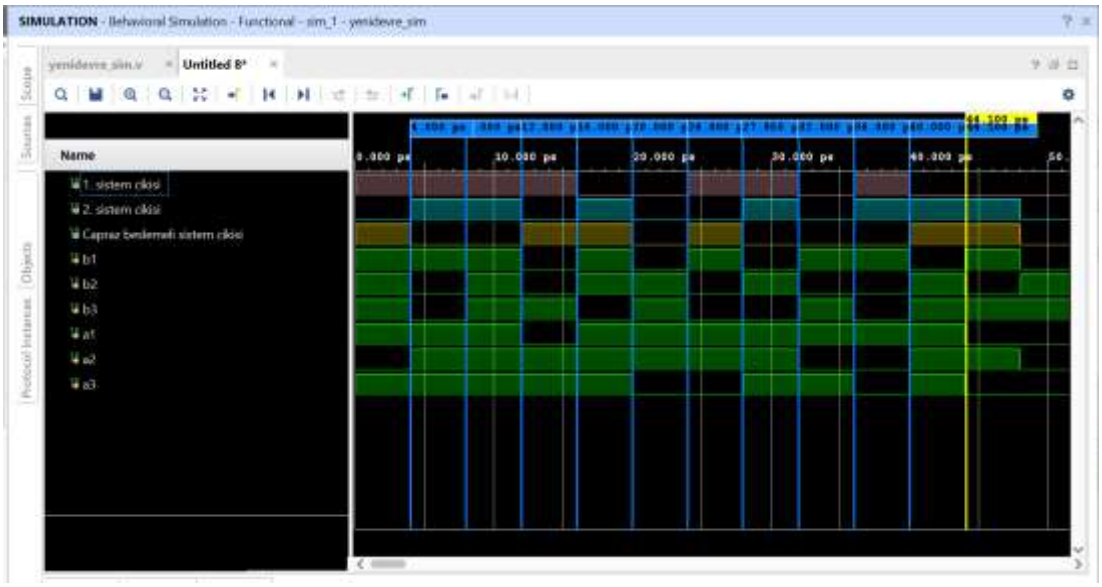
(a)

```
.B1(b1),
.B2(b2),
.B3(b3),
.A1(a1),
.A2(a2),
.A3(a3),
.Out3(out3),
.Out2(out2),
.Out1(out1)
);

initial begin


b[0]=1; b[1]=1; b[2]=1; b[3]=0;
b[4]=1; b[5]=0; b[6]=1;

a[0]=1; a[1]=1; a[2]=1; a[3]=0;
a[4]=1; a[5]=1; a[6]=1;

for(n=7; n<=20020; n=n+1) begin

b1=b[n-7];
b2=b[n-3];

b3=b[n-1];
a1=a[n-7];
a2=a[n-4];
a3=a[n-1];

#2;

b[n]=out1;
a[n]=out2;
#2;

Rand[n-7]=out3;
end
```

(b)

Şekil 4.12 Çapraz (a) ve doğrusal (b) LFSR verilog sentezleme programı

$X^7 + X^5 + 1$ ve $X^7 + X^4 + 1$ denklemi için çapraz beslemeli rastgele sayı üretecisinin için ilk 10 çıkış değeri aşağdaki gibidir.



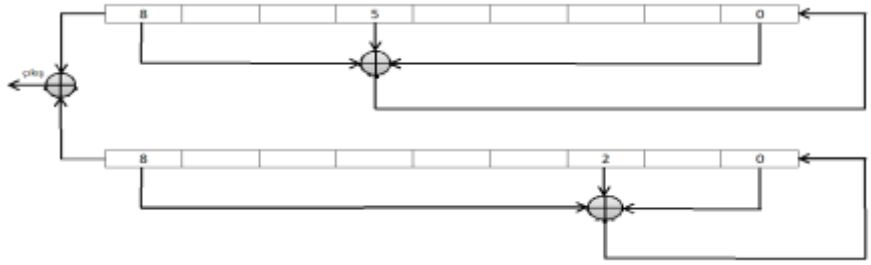
Şekil 4.13 $X^7 + X^5 + 1$ ve $X^7 + X^4 + 1$ denklemleri için çapraz beslemeli sistemin simülasyon çıktısı

Çizelge 4.24 de q=9 için kullanılan denklemler ve başlangıç dizeleri verilmiştir.

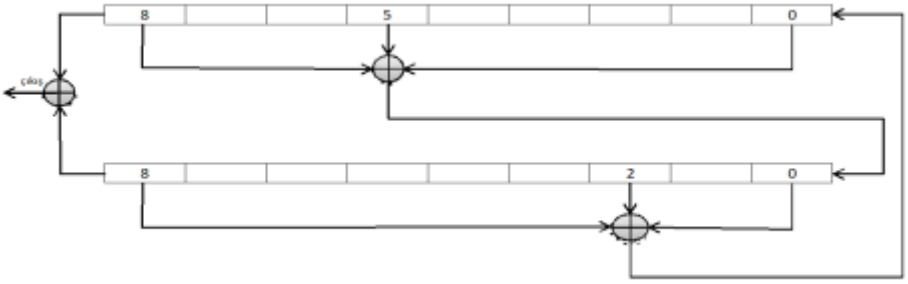
Çizelge 4.24 Tausworthe denkleminde q=9 için kullanılan denklemeler ve başlangıç dizesi

Kullanılan denklem	Başlangıç dizesi
$X^9 + X^6 + 1$	[0101010]
$X^9 + X^3 + 1$	[0101010]

Şekil 4.14'de q=9 için kullanılan eşzamanlı LFSR devresinin blok yapısı gösterilmiştir Önerilen çapraz beslemeli yapı ise Şekil 4.15'de verildiği gibidir.



Şekil 4.14 q=9 için doğrusal geri beslemeli rastgele sayı üreteci şeması



Şekil 4.15 q=9 için çapraz beslemeli rastgele sayı üreteci şeması

$X^9 + X^6 + 1$ ve $X^9 + X^3 + 1$ denklemi için çapraz beslemeli ve doğrusal geri beslemeli yapılardan elde edilen bit dizelerinin FIPS test sonucu Çizelge 4.25'deki gibidir. Çapraz beslemeli yapıdan elde edilen bitler FIPS testlerinin hepsinden geçerken doğrusal geri beslemeli yapıdan elde edilen bitlerin koşu testinin 6. kademesinden geçemediği gözlemlenmiştir.

Çizelge 4.25 q=9 Durumu için FIPS test sonucu

FIPS test	Doğrusal geri besleme	Çapraz besleme	Beklenen değer
Koşu 0	2383 1034 953 239 316 79	2501 1322 578 340 155 152	%1 2267 - 2733 %2 1079 - 1421 %3 502 - 748 %4 223 - 402 %5 90 - 223 %6 90 - 223
Koşu 1	2702 1350 477 238 158 158	2495 1238 675 294 185 141	%1 2267 - 2733 %2 1079 - 1421 %3 502 - 748 %4 223 - 402 %5 90 - 223 %6 90 - 223
Uzun koşu 0	7	12	<=34
Uzun koşu 1	8	9	<=34
Monobit	9602	10004	9654 < X< 10346
Poker	1.15	6.43	1.03<X<57.4

Çizelge 4.25 daki sonucu veren verilog kodu Şekil 4.16 de verilmiştir.

```
.B1(b1),
.B2(b2),
.B3(b3),
.A1(a1),
.A2(a2),
.A3(a3),
.Out3(out3),
.Out2(out2),
.Out1(out1)
);

initial begin


b[0]=1; b[1]=1; b[2]=1; b[3]=0;
b[4]=1; b[5]=0; b[6]=1;  b[7]=1;
b[8]=1;

a[0]=1; a[1]=1; a[2]=1; a[3]=0;
a[4]=1; a[5]=1; a[6]=1; a[7]=0;
a[8]=1;


for(n=9; n<=20020; n=n+1) begin

b1=b[n-9];
b2=b[n-4];

b3=b[n-1];
a1=a[n-9];
a2=a[n-7];
a3=a[n-1];

#2;

b[n]=out2;
a[n]=out1;
#2;

Rand[n-9]=out3;
end
```

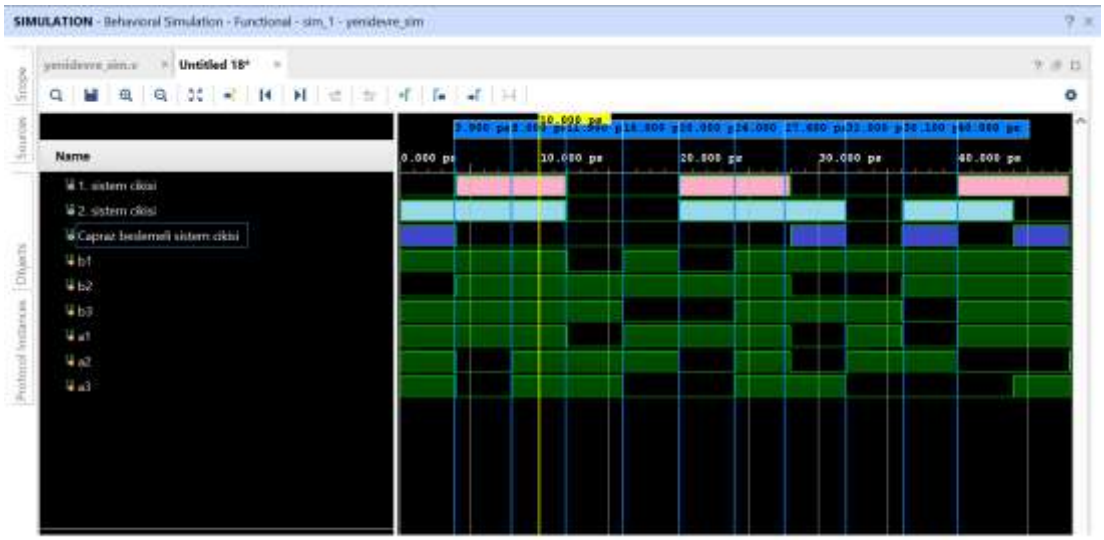
(a)

```
.B1(b1),
.B2(b2),
.B3(b3),
.A1(a1),
.A2(a2),
.A3(a3),
.Out3(out3),
.Out2(out2),
.Out1(out1)
);

initial begin


b[0]=1; b[1]=1; b[2]=1; b[3]=0;
b[4]=1; b[5]=0; b[6]=1;  b[7]=1;
b[8]=1;

a[0]=1; a[1]=1; a[2]=1; a[3]=0;
a[4]=1; a[5]=1; a[6]=1; a[7]=0;
a[8]=1;


for(n=9; n<=20020; n=n+1) begin

b1=b[n-9];
b2=b[n-4];

b3=b[n-1];
a1=a[n-9];
a2=a[n-7];
a3=a[n-1];

#2;

b[n]=out1;
a[n]=out2;
#2;

Rand[n-9]=out3;
end
```

(b)

Şekil 4.16 Çapraz (a) ve doğrusal (b) LFSR verilog sentezleme programı

$X^9 + X^6 + 1$ ve $X^9 + X^3 + 1$ denklemi için çapraz beslemeli rastgele sayı üretecisinin için ilk 10 çıkış değeri aşağdaki gibidir.



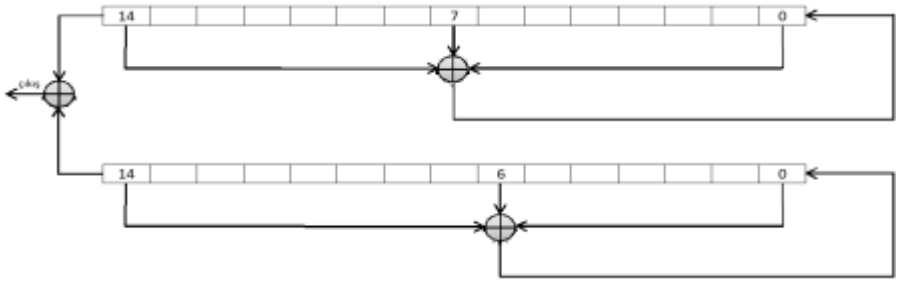
Şekil 4.17 $X^9 + X^6 + 1$ ve $X^9 + X^3 + 1$ denklemleri için çapraz beslemeli sistemin simülasyon çıktısı

Son olarak q=15 için üretim denklemleri ve başlangıç dizesi Çizelge 4.26'da verilmiştir.

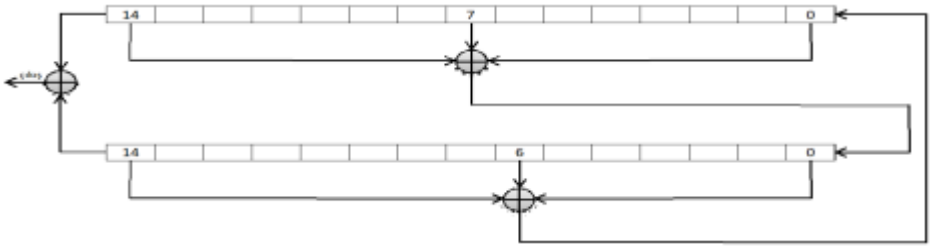
Çizelge 4.26 Tausworthe denkleminde q=15 İçin kullanılan denklemeler ve başlangıç dizesi

Kullanılan denklem	Başlangıç dizesi
$X^{15} + X^8 + 1$	[0101010]
$X^{15} + X^7 + 1$	[0101010]

Şekil 4.18'de q=15 için kullanılan devrenin blok yapısı gösterilmiştir. Birinci ve ikinci dizelerin geri beslemeleri çapraz olacak şekilde değiştirilmesi ise Şekil 4.19'de verilmiştir.



Şekil 4.18q=15 için doğrusal geri beslemeli rastgele sayı üreteci şeması



Şekil 4.19 q=15 için çapraz beslemeli rastgele sayı üreteci şeması

$X^{15}+X^{8}+1$ ve $X^{15}+X^{7}+1$denklemi için çapraz beslemeli ve doğrusal geri beslemeli yapılardan elde edilen bit dizelerinin FIPS test sonucu Çizelge 4.27'deki gibidir. Çapraz beslemeli ve doğrusal geri beslemeli yapılardan elde edilen bitler FIPS testlerinin hepsinden başarıyla geçmiştir.

Çizelge 4.27 q=15 Durumu için FIPS test sonucu

FIPS test	Doğrusal geri besleme	Çapraz besleme	Beklenen değer
Koşu 0	2593 1253 637 296 147 155	2535 1226 630 327 143 159	%1 2267 - 2733 %2 1079 - 1421 %3 502 - 748 %4 223 - 402 %5 90 - 223 %6 90 - 223
Koşu 1	2563 1243 643 308 174 152	2482 1242 634 330 159 165	%1 2267 - 2733 %2 1079 - 1421 %3 502 - 748 %4 223 - 402 %5 90 - 223 %6 90 - 223
Uzun koşu 0	13	18	<=34
Uzun koşu 1	17	16	<=34
Monobit	10071	10058	9654 < X< 10346
Poker	16.73	8.53	1.03<X<57.4

Çizelge 4.27 'deki sonucu veren verilog kodu Şekil 4.19 de verilmiştir.

```
 B1(b1),
 .B2(b2),
 .B3(b3),
 .A1(a1),
 .A2(a2),
 .A3(a3),
 .Out3(out3),
 .Out2(out2),
 .Out1(out1)
 );

initial begin


b[0]=1; b[1]=1; b[2]=1; b[3]=0;
b[4]=1; b[5]=0; b[6]=1;  b[7]=1;
b[8]=1; b[9]=0; b[10]=1; b[11]=0;
b[12]=0; b[13]=0; b[14]=1;

a[0]=1; a[1]=1; a[2]=1; a[3]=0;
a[4]=1; a[5]=1; a[6]=1; a[7]=0;
a[8]=1; a[9]=1; a[10]=1; a[11]=0;
a[12]=1; a[13]=1; a[14]=1;


for(n=15; n<=20020; n=n+1)
begin

b1=b[n-15];
b2=b[n-8];

b3=b[n-1];
a1=a[n-15];
a2=a[n-9];
a3=a[n-1];

#2;

b[n]=out2;
a[n]=out1;
#2;

Rand[n-15]=out3;
end
```

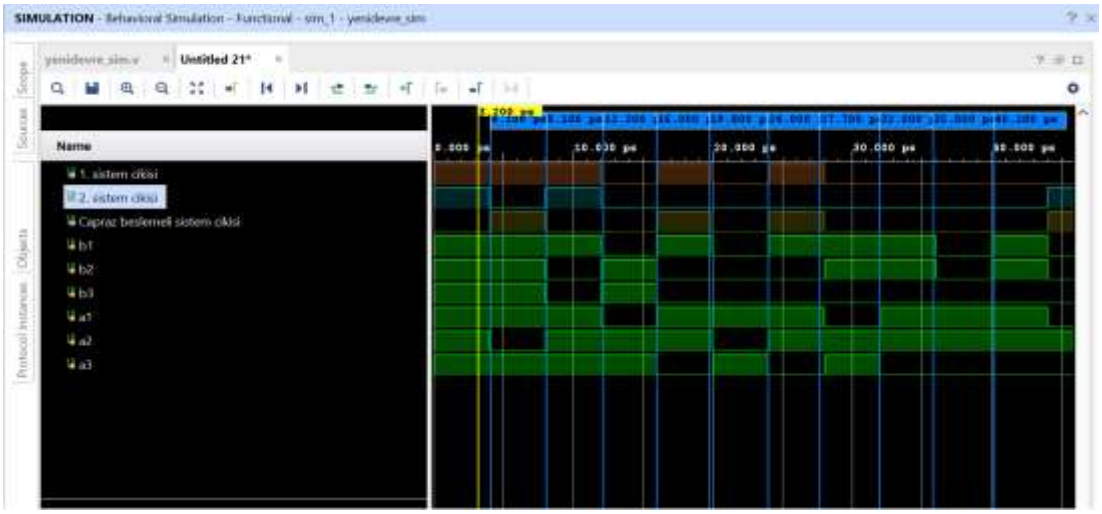
(a)

```
 B1(b1),
 .B2(b2),
 .B3(b3),
 .A1(a1),
 .A2(a2),
 .A3(a3),
 .Out3(out3),
 .Out2(out2),
 .Out1(out1)
 );

initial begin


b[0]=1; b[1]=1; b[2]=1; b[3]=0;
b[4]=1; b[5]=0; b[6]=1;  b[7]=1;
b[8]=1; b[9]=0; b[10]=1; b[11]=0;
b[12]=0; b[13]=0; b[14]=1;

a[0]=1; a[1]=1; a[2]=1; a[3]=0;
a[4]=1; a[5]=1; a[6]=1; a[7]=0;
a[8]=1; a[9]=1; a[10]=1; a[11]=0;
a[12]=1; a[13]=1; a[14]=1;


for(n=15; n<=20020; n=n+1)
begin

b1=b[n-15];
b2=b[n-8];

b3=b[n-1];
a1=a[n-15];
a2=a[n-9];
a3=a[n-1];

#2;

b[n]=out1;
a[n]=out2;
#2;

Rand[n-15]=out3;
end
```

(b)

Şekil 4.20 Çapraz (a) ve doğrusal (b) LFSR verilog sentezleme programı

$X^{15}+X^{8}+1$ ve $X^{15}+X^{7}+1$ denklemi için çapraz beslemeli rastgele sayı üretecisinin için ilk 10 çıkış değeri aşağdaki gibidir.



Şekil 4.21 $X^{15}+X^{8}+1$ ve $X^{15}+X^{7}+1$ denklemleri için çapraz beslemeli sistemin simülasyon çıktısı

Çizelgelerden de göründüğü gibi iki denklemli doğrusal ve çapraz FSR sonuçları karşılaştırmalı olarak verilmiştir. Bu çizelgelerden de görüleceği gibi iki denklemli doğrusal FSR yapısı tek denklemli yapıya göre rastgeleliği artırsa da yeterli rastgeleliği vermemektedir. Literatürde LFSR yapısını kullanan RSÜ yapılarında denklemin derecesini artırmanın yanında Tausworthe tanımına aykırı olarak ikiden fazla parametrenin katsayısını 1 seçme, denklem kuvvetini artırma gibi yöntemler kullanılarak rastgelelik sağlanmaktadır (Zhang G., 2005). LFSR yapısı Tausworthe tanımına uygun olarak sadece iki parametrenin katsayısını 1 tutarak, denklem derecesini q=7 ye kadar düşürülmesi rağmen yeterli rastgelelik vermiştir.

Doğrusal geribesleme ile aynı yapının çıktısını sisteme geri vermek, iki denklem kullanılsa bile Tausworthe denkleminin tekrarlama özelliğini yeterli miktarda kaldıramamaktadır.

5. SONUÇ VE ÖNERİLER

Bu çalışmada Tausworthe tanımına dayanan yeni bir rastgele sayı üreteci tasarımı ve verilog ile gerçeklemesi yapılmıştır. Tausworthe denklemleri XOR lojik kapısı ve kaydırmalı yazmaçlar kullanılarak gerçeklenmiştir. Bit üretiminde sürekliliği sağlamak için ilgili yazmaçlardaki veriler XOR kapısından geçirilerek çıktısı devrelere geri verilmiştir. Tausworhte denkleminde aynı denklemin çıkışı girişe veri olarak verildiğinde sistem denklem kuvveti ile ters orantılı bir bellek barındırmaktadır. Dolayısıyla Tausworthe tanımı gereği çıkış verisinde tekrarlanma barındırmaktadır ve tek denklemli yapı yeterli rastgeleliği sağlayamamaktadır. Bu özelliği verilog kodu ile de gösterilmiştir.

Yeterli rastgeleliği sağlamak için iki ayrı Tausworthe denklemini gerçekleyen devre eşzamanlı olarak kullanılmış, her bir devrenin çıkışı XOR kapısından geçirilerek çıkış bit dizisi elde edilmiştir. Eşzamanlı çalışan devrelerde doğrusal ve çapraz besleme kullanılmıştır. Bu devrelerden yirmi bin bit üretilmiş ve elde edilen sayıların rastgeleliğini test etmek için FIPS testleri kullanılmıştır. Kullanılan denklemlerin derecesi q=7'ye kadar düşürülmüş ve çapraz beslemeli yazmaç yapısının çıktısının FIPS testlerinin hepsinden başarıyla geçtiği fakat eşzamanlı LFSR devresinin çıktısının bazı testlerden geçemediği gözlemlenmiştir.

LFSR sistemi yapısal olarak bellekli bir yapı olmasından dolayı iki denklem kullanılsa bile bu bellek özelliği az da olsa çıktılarda gözlenmiş, dolayısıyla denklem derecesi düştükçe rastgelelik sağlanamamıştır. Litaratürde de LFSR yapılarında rastgeleliği artırmak için Tausworthe tanımı dışına çıkıp birden fazla parametrenin katsayısının 1 yapılması, denklem derecesinin artırılması, ikiden fazla denklem kullanılması gibi çözümler mevcuttur.

Sonuç olarak çapraz beslemeli yapı başarılı sonuçlar alınmış olması çapraz beslemenin Tausworthenin yapısındaki bellekli yapısının kırıldığını düşündürmektedir. Çapraz beslemeli yapı ile daha az kaydırmalı yazmaç kullanılarak doğrusal geri beslemeli yapıdan daha etkin bir rastgelelik elde

edilebilmektedir. Daha az kaydırmalı yazmaç kullanımı, FPGA uygulamalarında kaynaklarının daha verimli kullanılmasını sağlar.

KAYNAKLAR

Akchurin, A.D., Yusupov K.M, 2016. Verilog Programlama, Journal of Geophysical Research, Kazan Federal Üniversitesi, Rusya.

Akgül, A., 2015. Yeni Kaotik Sistemler ile Rasgele Sayi Üreteci Tasarımı ve Çoklu Ortam Verilerinin Yüksek Güvenlikli Şifrelenmesi, Sakarya Üniversitesi, Fen Bilimleri Enstitüsü, Doktora Tezi, 176, Sakarya.

Akkaya, S., 2016. Yeni Bir Kaos Tabanlı Rastgele Sayı Üreteci Kullanan Banka Şifrematik Cihazı Tasarımı ve Uygulaması, Sakarya Üniversitesi, Fen Bilimleri Enstitüsü, Yüksek Lisans Tezi, 93, Sakarya.

Alçin, F.Ö., 2011. Alan Programlanabilir Kapı Dizisi ile Sigma-Delta Modülatörlerin Gerçeklenmesi, Fırat Üniversitesi, Fen Bilimleri Enstitüsü, Yüksek Lisans Tezi, 73 s, Elazığ.

Anderson, H.L., 1986. Metropolis, Monte Carlo, and the Maniac.

Andrew, R., Soto, J., James, N., Miles, S., Elaine, B., Stefan, L., Levonson, M., Vangel, M., Banks, D., Heckert, A., Dray, J., Sanvo., 2010. A Statistical Test Suite for Random and Pseudorandom Number Generators for Cryptographic Applications, NIST Pubs.

Arathy, B.N., Mondal, A., Garani, S.S ., 2018. A Low-complexity Hardware AWGN Channel Emulator on FPGA Using Central Limit Theorem, 2018 IEEE 61st International Midwest Symposium on Circuits and Systems (MWSCAS), Windsor, ON, Kanada.

Atay, D.F., 2016. Rassal Sayı ve Rassal Değişken, Erişim Tarihi: 01.06.2021, https://silo.tips/download/rassal-sayi-ve-rassal-de-ken-dd-her-ui-nin-beklenen-deeri-benzetimde-rassallk-k.

Avaraoğlu, E., 2014. Donanım Tabanlı Rastgele Sayı Üretecinin Gerçekleştirilmesi, Fırat Üniversitesi, Fen Bilimleri Enstitüsü, Doktora Tezi, 146 s, Elazığ.

Bakır, D., 2017. Rastgele Sayı Üreteçlerinin Donanımsal Olarak Gerçekleştirilmesi, Munzur Üniversitesi, Fen Bilimleri Enstitüsü, Yüksek Lisans Tezi, 61 s, Tunceli.

Başak, S., 2011. FPGA Tabanlı Sentezlenebilir İşlemci Tasarımı,Yıldız Teknik Üniversitesi, Fen Bilimleri Enstitüsü, Yüksek Lisans Tezi, İstanbul.

Başlıgil, H., 2010. Modelleme ve Simülasyon, İstanbul Üniversitesi Açik ve Uzaktan Eğitim Fakültesi, Endüstri Mühendisliği Lisans Programı, 239, İstanbul.

Büyüksaraçoğlu, F., Buluş, E., 2021. Sözde Rassal Sayı Üretiminin Kriptografik Açıdan İncelenmesi, Erişim Tarihi: 01.06.2021 https://www.emo.org.tr/ekler/3e6f423ffcbf723_ek.pdf.

Cho, J., Jin, S., Pham, X., Jeon, J., Byun, J., Kang, H., 2006. A Real Time Object Tracking System, International Conference on Intelligent Robots and Systems, Beijing-China, 2822-2827.

Çakır, B., 2012. Akış Şifreleme Algoritmalarının Karşılaştırmalı Olarak İncelenmesi, Hacettepe Üniversitesi, Fen Bilimleri Enstitüsü, Yüksek Lisans Tezi, 159 s, Ankara.

Çelik, E., 2015. Olasılık Dağılımlarından Rassal Değişkenlik Üretimi ve VBA Uygulaması, Uludağ Üniversitesi, Sosyal Bilimler Enstitüsü, Yüksek Lisans Tezi, 87 s, Bursa.

Çelik, E.H., Sezen, K., 2017. Üstel Dağılım için Rassal Değişkenlik Üretimi ve VBA Uygulaması, Uludağ Üniversitesi İktisadi ve İdari Bilimler Fakültesi Dergisi, 36(2), 23-38.

Çiçek, S., 2016. Yeni Bir Kaotik Sistem ile FPGA tabanlı Bir Kaotik Haberleşme Sistemi Tasarımı Ve Gerçekleştirilmesi, Sakarya Üniversitesi, Fen Bilimleri Enstitüsü, Doktora Tezi, 215 s, Sakarya.

Demirkol, A.Ş., Özoğuz, S., Tavas, V., Kılınç, S., 2008. A CMOS Realization of Double-Scroll Chaotic Circuit and Its Application to Random Number Generation, IEEE International Symposium on Circuits and Systems.

Dereli, S., 2020. Yüksek Hızlı FPGA ile Yeni Bir LFSR Tabanlı 32-Bit Kayan Noktalı Rastgele Sayı Üreteci Tasarımı, International Journal of Advances in Engineering and Pure Sciences, 32(3), 219-228.

Diaz, J., Ros, E., Pelayo, F., Ortigosa, E.M., Mota, S., 2006. FPGA-Based Real-Time Optical-Flow System, IEEE Transactions on Circuits and Systems for Video Technology.

Elbaşı, E., Eskicioğlu A.M, 2006. PRN Based Watermarkıng Scheme for Color Images, İstanbul Ticaret Üniversitesi Fen Bilimleri Dergisi, 5(10), 119-131.

Erkek, E., 2015. Akış Şifreleme Algoritmaları Kullanılarak Rasgele Sayı Üretilmesi ve FPGA Ortamında Gerçekleştirilmesi, Fırat Üniversitesi, Fen Bilimleri Enstitüsü, Yüksek Lisans Tezi, 72 s, Elazığ.

Gürevin, B., 2019. New Random Number Generator Design Based on Microcomputer and Encyription Application, Sakarya Uygulamalı Bilimler Üniversitesi Lisansüstü Eğitim Enstitüsü, Yüksek Lisans Tezi, 90 s, Sakarya.

Güven, P., 2006. Otonom Olmayan Kaotik Sistemlerde Rastgele Sayı Üretiminin İncelenmesi, İstanbul Teknik Üniversitesi, Fen Bilimleri Enstitüsü, Yüksek Lisans Tezi, 50 s, İstanbul.

Hanköylü, S., 2019. Quantum Random Number Generator Design and Impelementation, Hacettepe Üniversitesi, Fen Bilimleri Enstitüsü, Yüksek Lisans Tezi, 82 s, Ankara.

Slepovichev I.I, 2017. Генераторы Псевдослучайных Чисел, 21.02.2017.

Huang, D, Zeng, D.Z ,Long, T., Yu, J.Y., 2010. Design of a Correlated Lognormal Distributed Sequence Generator Based on Virtex-IV Series FPGA. International Conference on Computer Application and System Modeling (ICCASM 2010), Taiyuan, Çin, 22-24 October 2010.

İçer, Y., 2016. Temel Kenar Algılama Algoritmalarının FPGA Üzerinde Gerçeklenmesi, Fırat Üniversitesi, Fen Bilimleri Enstitüsü, Yüksek Lisans Tezi, 77, Elazığ.

Icyscıenceı, 2021. Rastgele sayı nedir? - Techopedia nedir? - gelişme – 2021. Erişim Tarihi: 02.05.2021, https://tr.icyscience.com/random-number#menu-1.

Innocente, R., 2018. Random Number, SISSA, Trieste for the joint ICTP/SISSA MHPC.it master course, April, 2018.

Jianpo, Y., 2016. Gauss Beyaz Gürültü Sinyali Kaynağı Üretmek için Yöntem, Shenzhen Kang'ao, Industrial Technology Co Ltd. Erişim Tarihi: 01.06.2021, https://patents.google.com/?assignee=Shenzhen+Kang%27ao+Industrial+Technology.

Koçdoğan, A., 2015. FPGA Üzerinde Hafızalı Hücresel Otomat Yapısı ile Rastgele Sayı Üreteci Tasarımı, İstanbul Teknik Üniversitesi, Fen Bilimleri Enstitüsü, Yüksek Lisans Tezi, 81, İstanbul.

Koyuncu, İ., 2014. Kriptolojik Uygulamalar için Fpga Tabanlı Yeni Kaotik Osilatörlerin ve Gerçek Rasgele Sayı Üreteçlerinin Tasarımı ve Gerçeklenmesi, Sakarya Üniversitesi, Fen Bilimleri Enstitüsü, Doktora Tezi, 145 s, Sakarya.

L'Ecuyer, P., 1999. Tables of Maximally- Equıdistributed Combined LFSR Generators, 1999 American Mathematics of Computation, 68(1999), 261-269.

L'ecuyer, P., 2017. History of Uniform Random Number Generatıon, 2017 IEEE Proceedings of the 2017 Winter Simulation Conference, Montreal.

Lortoğlu, M., 2019. FPGA Tabanlı Yapay Sinir Ağı Kullanarak Buğday Türlerinin Sınıflandırılması, KTO Karatay Üniversitesi, Fen Bilimleri Enstitüsü, Yüksek Lisans Tezi, Konya.

Math, 2021. Düzgün dağıtılmış rastgele sayı üretimi. Erişim Tarihi: 29.04.2021,:https://www.math.pku.edu.cn/teachers/lidf/docs/statcomp/html/_statcompbook/rng-uniform.html.

Matthias, P., Werner, S., 2021. Bundesamt fur ¨Sicherheit in der Informationstechnik (BSI) Bonn, Germany.

Matsumoto, M., Nishimura, T., 1998. Mersene Twister: a 623- dimensionally equidistributed uniform pseudo- random number generator.

NIST, 2010. A Statistical Test Suite for Random and Pseudorandom Number Generators for Cryptografic Applications, Special Publication 800-22 Revision 1a.

Robert, G.B., 2021. Dieharder: Rastgele Sayı Test Paketi, Duke Üniversitesi.

Özçelik, Mehmet F., 2012. Görüntü İşleme Algoritmalarının FPGA Üzerinde Gerçekleştirilmesi, Gazi Üniversitesi, Bilişim Enstitüsü, Yüksek Lisans Tezi, 63 s, Ankara.

Özdemir, K., 2008. Sürekli- zamanlı Kaos ile Rastgele Sayı Üreteci Tasarımı, İstanbul Teknik Üniversitesi, Fen Bilimleri Enstitüsü, Yüksek Lisans Tezi, 71 s, İstanbul.

Özkaynak, F., Özdemir H.İ., Özer, A.B, 2015. Cryptographic Random Number Generator for Mobile Devices, 23nd Signal Processing and Communications Applications Conference (SIU), 16-19 May 2015, Turkey, 24-29.

Özkaynak, F., 2015. Kriptolojik Rasgele Sayı Üreteçleri, Fırat Üniversitesi TBV-Türkiye Bilişim Vakfı Bilgisayar Bilimleri ve Mühendisliği Dergisi, 8(2), 37-44.

Özpolat, E., 2017. FPGA Tabanlı Fır Fıltre Tasarımı, Fırat Üniversitesi, Fen Bilimleri Enstitüsü, Yüksek Lisans Tezi, 64 s, Elazığ.

Öztürk, İ., 2013. Kaotik Sistem Tabanlı Sözde Rastgele Sayı Üreteçlerinin Tasarımı ve Gerçekleştirilmesi, Erciyes Üniversitesi, Fen Bilimleri Enstitüsü, Yüksek Lisans Tezi, 117 s, Kayseri.

Robinson, S.O., Dessart, D.J., 1998. Yearbook-National Council of Teachers of Mathematics, 243-250, NCTM, USA.

Sass, R., Schmidt, A.W.G., 2010. Embedded Systems Design with Platform FPGAs: Principles and Practices, Morgan Kaufmann, Amsterdam, Hollanda.

Savran, İ., 2017. Donanım Tanımlama Dili VHDL ve FPGA Uygulamaları, Papatya Bilim, 320, İstanbul.

Schoukens, J., Pintelon, R., van der Ouderaa, E., Renneboog, J., 1988. Survey of Excitation Signals for FFT Based Signal Analyzers, IEEE Transactions on Instrumentation and Measurements, 37(3), 342-352.

Shihai, W., Hayran, H., Jie,Y., Wei, H., He, Y., Dashuang, L., Bingjie,X., 2021. Haberleşme Güvenliği Laboratuvarı Bilim ve Teknoloji, Güneybatı İletişim Enstitüsü, Chengdu, Sichuan 610041, Çin.

Solak, İ., 2008. UHF RFID Etiketler için SRAM Tabanlı Donanımsal Rastgele Sayı Üreteci Tasarımı, Karadeniz Teknik Üniversitesi, Fen Bilimleri Enstitüsü, Yüksek Lisans Tezi, 66 s, Trabzon.

Song, J.K., 2003. FIPS 140-2 İstatistiksel Test Paketinin IP Çekirdeği, Keio Üniversitesi, IP SoC Tasarım.

Tausworthe, R.C., 1965. Random Numbers Generated by Linear Recurrence Modulo Two, Mathematics of Computation, 19(90), 201–209.

Tavas, V., Demirkol A.S., Özoğuz S., Kılınç S., Toker A., Zeki A., 2010. An IC Random Number Generator Based on Chaos, Uluslararası Uygulamalı Elektronik Konferansı, IEEE, 11.10.2010.

Tavas, V., Demirkol A.S., Özoğuz S., Kılınç S., Zeki A., Toker, A., 2010. An ADC Based Random Bit Generator Based on a Double Scroll Chaotic Circuit, Devreler, Sistemler ve Bilgisayarlar Dergisi, 19 (07), 1621-1639.

Tavas, V., 2011. Tümleştirmeye Uygun Rastgele Sayı Üreteçleri, İstanbul Teknik Üniversitesi, Fen Bilimleri Enstitüsü, Doktora Tezi, 134 s, Türkiye.

Tezukaç, S., 1993. A Unifed View of Long- Pedriod Random Number Generators, IBM Research, Tokyo Research Laboratory.

Tuna, M., 2017. Chaos-Based Dual Entropy Core True Random Number Generator Design and ITS Realization on FPGA, Karabük University.

Thomas, D.E., Moorby P.R., 2002. Verilog Donanım Tanımlama Dili, Kluwer Akademik Yayıncılık, US.

Tuncer, S.A, Genç Y., 2019. İnsan Hareketleri Tabanlı Gerçek Rastgele Sayı Üretimi, Bitlis Eren Üniversitesi Fen Bilimleri Dergisi, 8(1), 261-269.

Vega-Rodríguez, M.A., Sánchez-Pérez, J.M., Gómez-Pulido, J.A., 2002. An FPGA Based Implementation for Median Filter Meeting The Real-Time Requirements of Automated Visual Inspection Systems, 10th Mediterranean Conference on Control and Automation, Lisbon-Portugal.

Yakut, S, 2019. Gerçek Rastgele Sayı Üreteçlerinin Tasarlanması ve Analizi, Fırat Üniversitesi, Fen Bilimleri Enstitüsü, Doktora Tezi,111s, Elazığ.

Yılmaz, N., 2008. Alan Programlamalı Kapı Dizileri (FPGA) Üzerinde Bir YSA'nın Tasarlanması ve Donanım Olarak Gerçekleştirilmesi, Selçuk Üniversitesi, Fen Bilimleri Enstitüsü, Yüksek Lisans Tezi, Konya.

Yong, Z., 2009. Mikroelektronik ve Katı Hal Elektroniği, Harbin Teknoloji Enstitüsü, Yüksek Lisans Tezi, Çin.

Zhang, G., Leong, P.H.W., Lee, D.U., Villasenor, J.D., Cheung, R.C.C., Luk, W., 2005. Ziggurat-Based Hardware Gaussian Random Number Generator, International Conference on Field Programmable Logic and Applications, Tampere, Finland, 24-26 August 2005, Tampere, Finland, 275-280.

Wikipedia, 2021. FPGA nedir, ASIC, mikroişlemci ve mikrodenetleyici ile karşılaştırması, Entegre çeşitleri. Erişim Tarihi: 01.04.2021, https://www.slideshare.net/SerkanDereli2/fpga-ve-vhdl-ders-1.

ÖZGEÇMİŞ

Adı Soyadı : Minare HASANBEYLİ

Eğitim Durumu

Lise : 279 numaralı lise, 2012

Lisans : Milli Havacılık Akademisi, Fen Bilimleri Fakültesi, Elektrik-Elektronik Mühendisliği Bölümü, 2016

Yüksek Lisans : İstanbul Ticaret Üniversitesi,
Fen Bilimleri Enstitüsü, Elektronik ve Haberleşme Mühendisliği Anabilim Dalı, 2021

Yayınları

Hasanbeyli M., Tavas V., 2021. Verilog ile Tausworthe Denklemine Dayanan Yeni Bir Rastgele Sayı Üreteci Tasarımı, İstanbul Ticaret Üniversitesi Fen Bilimleri Dergisi, 20(39), 112-126.