İSTANBUL TİCARET ÜNİVERSİTESİ FEN BİLİMLERİ DERGİSİ

İstanbul Commerce University Journal of Science

http://dergipark.gov.tr/ticaretfbd

Araştırma Makalesi / Research Article

VERİLOG İLE TAUSWORTHE DENKLEMİNE DAYANAN YENİ BİR RASTGELE SAYI ÜRETECİ TASARIMI*

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

Minare HASANBEYLİ[1] **Vedat TAVAS[2]**

Sorumlu Yazar / Corresponding Author
minare.hesenli.95@mail.ru

Geliş Tarihi / Received
12.05.2021

Kabul Tarihi / Accepted
08.06.2021

Öz

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. Rastgele sayılar oluşturmanın birçok farklı yolu vardır. Rastgele sayıların en önemli özelliği ise bağımsız olmasıdır, böyle olması ardışık sayılar arasında hiçbir ilişki kurulmamasına neden olur. Bu çalışmada Tausworthe denklemine dayanan ayrık zamanlı rastgele sayı üreteci tasarlanmıştır. Tasarımda geri beslemeli kaydırmalı yazmaçlar kullanılmıştır. Tasarım Xilinx yazılımı kullanılarak Verilog donanım tanımlama dili ile gerçeklenmiştir. Önerilen yöntemle üretilen bit dizilerinin rastgeleliğini belirlemek için FIPS test kümesi kullanılmış ve diziler bu testlerden başarıyla geçmiştir.

Anahtar Kelimeler: Geri beslemeli kaydırmalı yazmaç, rastgele sayı üreteci, Tausworthe yöntemi, Verilog.

Abstract

Random numbers are used in many fields such as encryption, computer simulation, random design. Random numbers are derived from random processes that do not involve any predictability. The lack of randomness can affect the security of the entire system. Therefore, random numbers should not be predictable. There are many different ways to generate random numbers. The most important feature of random numbers is that they are independent, which causes no relationship to be established between consecutive numbers. In this study, feedback shift registers are used in the lunar design based on the Tausworthe equation. The design was implemented with Verilog hardware description language using Xilinx software. FIPS test set was used to determine the randomness of the bit strings produced by the proposed method and the strings passed these tests successfully time random number generator is designed.

Keywords: Feedback shift register, random number generator (RNG), Tausworthe method, Verilog.

**Bu yayın Minare HASANBEYLİ isimli öğrencinin İstanbul Ticaret Üniversitesi Fen Bilimleri Enstitüsü, Elektronik ve Haberleşme Mühendisliği Programındaki Yüksek Lisans tezinden üretilmiştir.*

[1]İstanbul Ticaret Üniversitesi, Fen Bilimleri Enstitüsü, Elektronik ve Haberleşme Mühendisliği Anabilim Dalı, Küçükyalı, İstanbul, Türkiye. minare.hesenli.95@mail.ru, Orcid.org/0000-0002-5470-6194.

[2]İstanbul Ticaret Üniversitesi, Mühendislik Fakültesi, Elektronik ve Haberleşme Mühendisliği Bölümü, Küçükyalı, İstanbul, Türkiye. vtavas@ticaret.edu.tr, Orcid.org/0000-0003-2945-9846..

1. GİRİŞ

Rastgele Sayı Üreteçlerinin günümüzdeki önemi çok fazladır. 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. Rastgele sayıların en önemli özelliği sayılar arasında hiçbir ilişki kurulamamasıdır. Rastgele sayı üreteçleri "gerçek rastgele sayı üreteci (True Random Number Generator, TRNG)" ve "sözde rastgele sayı üreteci (Pseudo Random Number Generator, PRNG)" olmak üzere iki gruba ayrılmıştır. (ICYSCIENCE, 2021).

1.1. 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).

1.2. Sözde Rastgele Sayı Üreteci

Sözde Rastgele Sayı Üreteçlerinin (SRSÜ) çıktıları gerçek anlamda rastgele değildir. 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).

1.3. Alanda Programlanabilir Kapı Dizileri ve Donanım Tanımlama Dilleri

Alanda Programlanabilir Kapı Dizilerinin (Field-programmable gate array, FPGA) önemli bir özelliği, yeniden yapılandırma yeteneğidir (İçer, 2016). FPGA'lar bir donanım tanımlama dili kullanılarak sayısal tasarım yapmaya imkân sağlayan geliştirme platformlarıdır. FPGA, devre programlamayı destekleyen ve devrenin simülasyonunu gerçekleştiren bir çiptir. 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)

FPGA iç bağlantılarının yapılması ancak programlanması ile mümkündür. FPGA'nın içerisinde olacak devrenin tasarımı 2 yolla yapılmaktadır. Bunlardan biri şematik tasarım ile devrenin çizilmesi, diğeri ise donanım tanımlama dillerinden biri ile devrenin davranışının tanımlanmasıdır. En yaygın kullanılan donanım tanımlama dilleri Verilog ve VHDL (Very High Speed Integrated Circuit Hardware Description Language, Yüksek Hızlı Tümleşik Devreler İçin Donanım Tanımlama Dili) dir.(Savran, 2017),

2. LİTERATÜR TARAMASI

Literatürde Rastgele sayı üreteçleri, birçok farklı matematiksel yöntem kullanmaktadır. Bunlara genel olarak doğrusal eşzamanlı oluşturucu, orta kare yöntemi, olasılık dağılımına dayanan ters çevirme, kabul-ret yöntemi gibi örnekler verilebilir. Bu yöntemlere bağlı olarak Mersenne Twister, Monte Carlo, Tausworthe gibi farklı algoritmalar kullanılmaktadır (L'ecuyer, 2017).

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 üzerinde sayısal gerçeklemesi yapılmıştır. Gerçek rastgele sonuçlar alındığı görülmüş ve sonra da bu sistem üzerinden rastgele sayı üreteci tasarımı yapılmıştır (Koçdoğan, 2015).

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).
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) 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).

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 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).

3. MATERYAL VE YÖNTEM

Bu çalışmada 1965 yılında kabul edilen Tausworthe 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).

$$b_i = \left(c_1 b_{i-1} + c_2 b_{i-2} + \cdots + c_q b_{i-q}\right) mod2 \quad (1)$$

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

$$b_i = \left(b_{i-r} + b_{i-q}\right) mod2 \quad (2)$$

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

$$b_i = \begin{cases} 1, & b_{i-r} = b_{i-q} \\ 0, & b_{i-r} \neq b_{i-q} \end{cases} \quad (3)$$

Giriş değerlerinin 0 veya 1 olması durumunda (3) numaralı eşitlik denklemin sonucunun XOR lojik kapısının doğruluk tablosuyla aynı olduğunu göstermektedir (Math**,** 2021). Tausworthe yöntemiyle üretilen rastgele sayıların periyodu $2^q - 1$ şeklinde belirlenir. Uzun periyotlu rastgele sayı dizesi üretmek için q değerini büyük tutmak gerekmektedir.

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. Monobit testinin başarılı olabilmesi için alınan değerin 9654 <X<10346 aralığında olması gereklidir. Poker testinin başarılı olabilmesi için alınan değerlerin 1.03<X<57.4 aralığında olması gereklidir. Uzun koşu testlerinin başarılı olabilmesi için alınan değerlerin <=34 değerine eşit veya bu değerden küçük olmalıdır. Koşu testinin başarılı olabilmesi için alınan değerlerin Tablo 1 de verilen değer aralıklarında olması gereklidir (Akkaya**,** 2016).

Tablo 1. Koşu Testi İçin 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

4. YÖNTEMİN UYGULANMASI

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. FIPS testlerinin kullanılabilmesi için yirmi bin bitlik bir dizeye ihtiyaç duyulmaktadır. 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. Elde edilen sayı dizisi FIPS testlerinden geçirilmiştir.

Önce tek denklemli yöntemler uygulanmış ve tek denklemli yapıların yeterli rastgelelikte bir dizisi oluşturmadığı doğrulanmıştır. Daha sonra iki denklemli yöntemler uygulanmış, literatürde olan eşzamanlı doğrusal geri beslemeli kaydırmalı yazmaç (LFSR) yapısı ile ve önerilen yöntem ile elde edilen bit dizelerinin rastgelelik analizleri yapılmıştır. Tek denklemli sistemlerde kullanılan denklem değiştirilmeden ikinci bir denklem sisteme eklenerek önerilen yöntemin etkinliği gösterilmeye çalışılmıştır.

$X^7 + X^5 + 1$ denklemi için başlangıç bit dizisi Tablo 2' deki gibidir.

Tablo 2. 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 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.

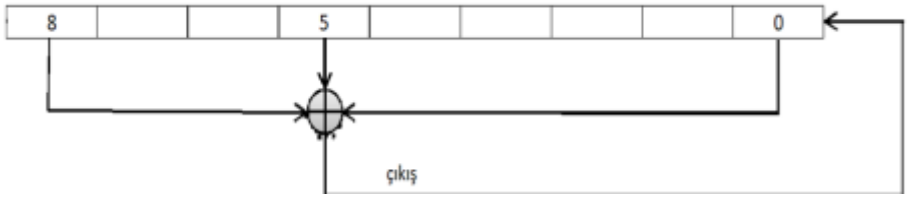
Şekil 1. X^7+X^5+1 Denklemini Gerçekleyen LFSR Şeması

X^9+X^6+1 denklemi için başlangıç bit dizisi Tablo 3' deki gibidir.

Tablo 3. Tausworthe Denkleminde q=9 için Kullanılan Denklem ve Başlangıç Dizesi

Kullanılan denklem	Başlangıç dizesi
X^9+X^6+1	[010101010]

X^9+X^6+1 denklemini kullanan LFSR yapısı Şekil 2'deki gibidir.



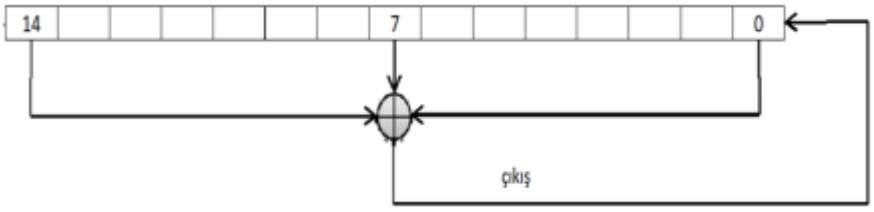
Şekil 2. X^9+X^6+1 Denklemini Gerçekleyen LFSR Şeması

$X^{15}+X^8+1$ denklemi için başlangıç bit dizisi Tablo 4'teki gibidir.

Tablo 4. Tausworthe Denkleminde q=15 için Kullanılan Denklem ve Başlangıç Dizesi

Kullanılan denklem	Başlangıç dizesi
$X^{15}+X^8+1$	[010101010101010]

$X^{15}+X^8+1$ denklemini kullanan LFSR yapısı Şekil 3'teki gibidir.



Şekil 3. $X^{15}+X^8+1$ Denklemini Gerçekleyen LFSR Şeması

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 Şekil 4'te de gösterildiği gibi eşzamanlı olarak kullanılması gibi tasarım kullanılmıştır (Arathy vd., 2018).

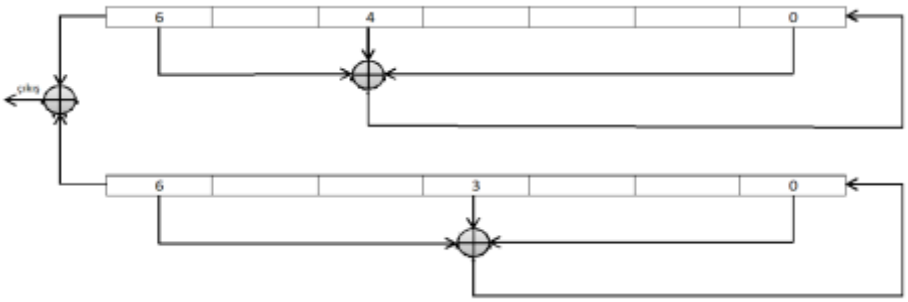
Bu çalışmada da rastgeleliği artırmak için birden fazla Tausworthe denklemi eşzamanlı olarak kullanılmıştır. Literatürde olan doğrusal geri besleme yerine geri besleme yapısı her bir denklem diğerini besleyecek şekilde çapraz olarak değiştirilmiştir. Çünkü LFSR yapısı aynı denklem çıktısını kendine geri besleme olarak aldığından sistemde bir periyodiklik oluşmakta, bu da çıktıların rastgele olmamasına neden olmaktadır. Çapraz geri besleme kullanılarak LFSR yapısının doğasındaki periyodikliğin kırılması amaçlanmıştır.

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. Tablo 5'te q=7 için kullanılan denklemler ve başlangıç dizeleri verilmiştir.

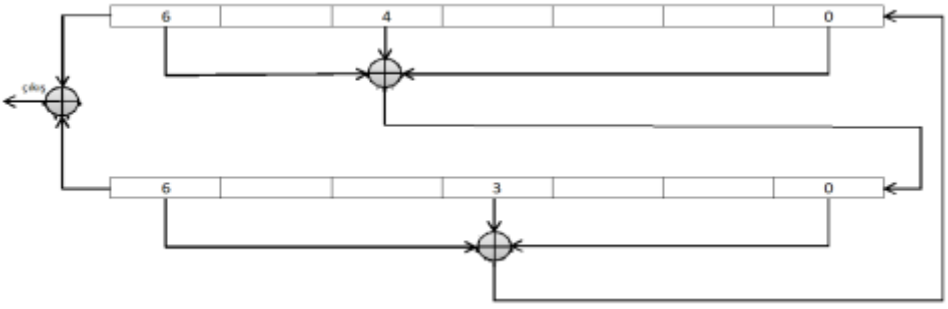
Tablo 5. Tausworthe Denkleminde q=7 için Kullanılan Denklemeler ve Başlangıç Dizesi

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'te gösterildiği gibidir. Önerilen çapraz geri beslemeli devrenin blok yapısı Şekil 5'te gösterilmiştir.



Şekil 4. q=7 için Kullanılan Doğrusal Geri Beslemeli Rastgele Sayı Üreteci Şeması



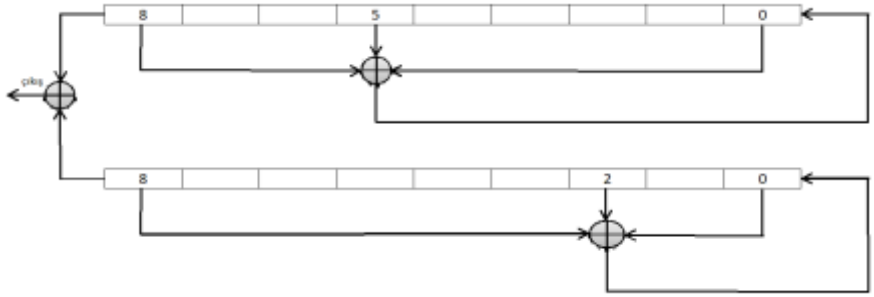
Şekil 5. q=7 için Kullanılan Çapraz Geri Beslemeli Rastgele Sayı Üreteci Şeması

Tablo 6'da q=9 için kullanılan denklemler ve başlangıç dizeleri verilmiştir.

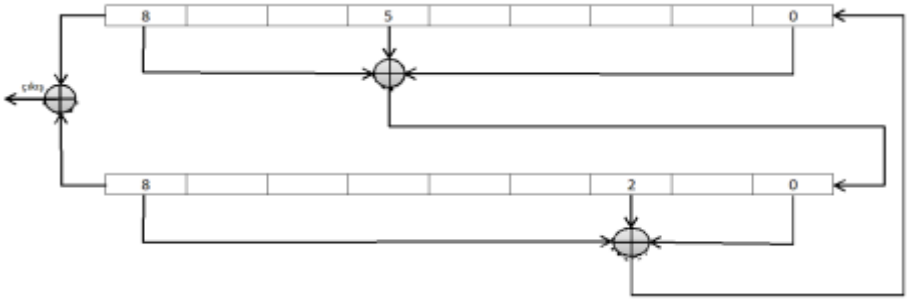
Tablo 6. Tausworthe Denkleminde q=9 için Kullanılan Denklemeler ve Başlangıç Dizesi

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

Şekil 6'da q=9 için kullanılan eşzamanlı LFSR devresinin blok yapısı gösterilmiştir Önerilen çapraz geri beslemeli yapı ise Şekil 7'de verildiği gibidir.



Şekil 6. q=9 için Doğrusal Geri Beslemeli Rastgele Sayı Üreteci Şeması



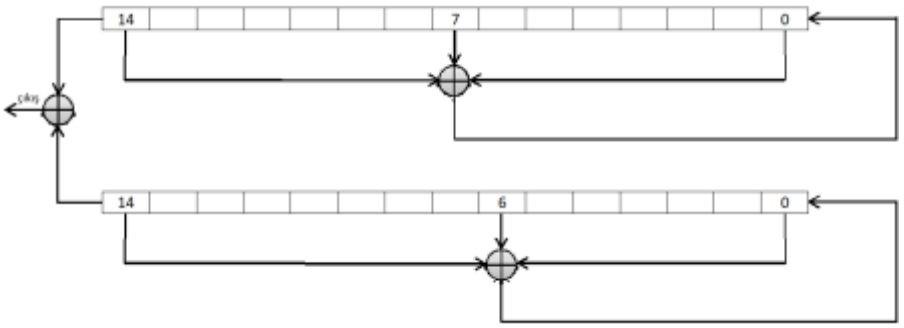
Şekil 7. q=9 için Çapraz Geri Beslemeli Rastgele Sayı Üreteci Şeması

Son olarak q=15 için üretim denklemleri ve başlangıç dizesi Tablo 7'de verilmiştir.

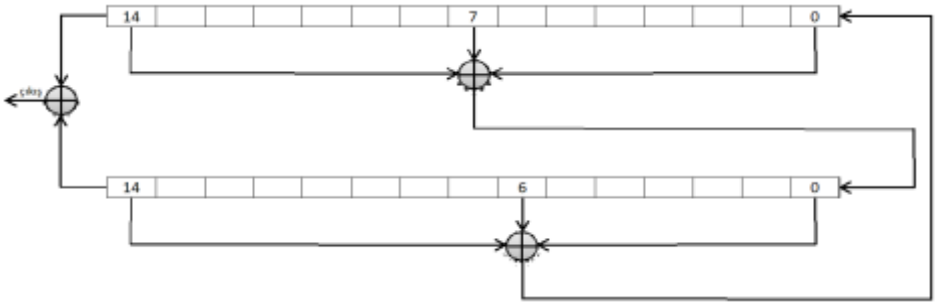
Tablo 7. Tausworthe Denkleminde q=15 için Kullanılan Denklemeler ve Başlangıç Dizesi

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

Şekil 8'de q=15 için kullanılan devrenin blok yapısı gösterilmiştir. Birinci ve ikinci dizelerin geri beslemeleri çapraz olacak şekilde değiştirilmesi ise Şekil 9'da verilmiştir.



Şekil 8. q=15 için Doğrusal Geri Beslemeli Rastgele Sayı Üreteci Şeması



Şekil 9. q=15 için Çapraz Geri Beslemeli Rastgele Sayı Üreteci Şeması

5. TESTLER

Yukarıda blok yapıları verilen tasarımlar Verilog donanım tanımlama dili ile Vivado HLS v.2020.2 programı ile sentezlenmiştir. Sentezlenen her bir tasarımdan FIPS testleri için 20 bin adet bitten oluşan veri üretilmiştir. Bu bit dizelerinin rastgele kabul edilebilmesi FIPS test kümesinde tanımlı dört testten de geçmesi gerekmektedir. Bu testler Monobit, Poker, Koşu ve Uzun Koşu testleridir. FIPS testleri ile ilgili bilgiler Bölüm 3'te verilmişti.

Bölüm 4'de belirtildiği gibi Tausworthe yöntemine dayanan tek denklemli yapılardan elde edilen verinin rastgeleliğinin kötü olduğunu göstermek için ilk önce tek denklemli tasarımların test sonuçları verilmiştir. Daha sonra literatürde kullanılan iki denklemli LFSR yapısı ile önerilen iki denklemli çapraz geribeslemeli kaydırmalı yazmaç (FSR, feedback shift register) test sonuçları verilmiştir. Önerilen çapraz geri beslemeli yöntemin etkinliğini göstermek iki denklemli yapılarda kullanılan denklemlerden biri tek denklemli LFSR yapısında kullanılan denklemle aynı seçilmiştir. Elde edilen iki denklemli doğrusal ve çapraz FSR yapıları sonuçları karşılaştırılmıştır.

$X^7 + X^5 + 1$ denkleminden elde edilen bit dizisinin FIPS test sonucu Tablo 8'de verilmiştir. q=7 olacak şekilde gerçeklenen tekli LFSR yapısı ile yeterli rastgelelikle bit dizisi üretilememiştir.

Tablo 8. 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

$X^9 + X^6 + 1$ denklemiyle oluşturulan bit dizisinin FIPS test sonucu Tablo 9'da verilmiştir. q=9 olarak seçilen denklemden de yeterli rastgelelik sağlanamamıştır.

Tablo 9. q=9 Durumu için FIPS Test Sonucu

FIPS test	Tausworthe yöntemi	Beklenen değer
Koşu 0	10000	%1 2267 - 2733
	0	%2 1079 - 1421
	0	%3 502 - 748
	0	%4 223 - 402
	0	%5 90 - 223
	0	%6 90 - 223
Koşu 1	10001	%1 2267 - 2733
	0	%2 1079 - 1421
	0	%3 502 - 748
	0	%4 223 - 402
	0	%5 90 - 223
	0	%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

$X^{15} + X^8 + 1$ durumu için FIPS test sonucu Tablo 10'da verilmiştir. Bu durumda da yeterli rastgelelik elde edilememiştir.

Tablo 10. q=15 Durumu için FIPS Test Sonucu

FIPS test	Tausworthe yöntemi	Beklenen değer
Koşu 0	10000	%1 2267 - 2733
	0	%2 1079 - 1421
	0	%3 502 - 748
	0	%4 223 - 402
	0	%5 90 - 223
	0	%6 90 - 223
Koşu 1	10001	%1 2267 - 2733
	0	%2 1079 - 1421
	0	%3 502 - 748
	0	%4 223 - 402
	0	%5 90 - 223
	0	%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

Tablolar 8-10 da görüldüğü gibi Tausworthe yöntemini kullanan tek denklemli LFSR yapısı Bölüm 4'e de belirtildiği gibi yeterli rastgeleliği sağlayamamıştır.

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. Kaydırmalı yazmaçlar Şekil 4,6,8'deki gibi doğrusal geri besleme ve Şekil 5,7,9'daki gibi çapraz geri besleme ile ayrı ayrı beslenmiş, bu şekilde her bir yapının çıktısı XOR kapısından geçirilerek bit dizisi üretilmiştir. Devrelerden elde edilen değerler aşağıdaki Tablolarda gösterilmiştir.

$X^7 + X^5 + 1$ ve $X^7 + X^4 + 1$ denklemi için çapraz ve doğrusal geri beslemeli yapılardan elde edilen bit dizilerinin FIPS test sonucu Tablo 11'deki gibidir. Çapraz geri beslemeli yapıdan elde edilen bitler FIPS testlerinin hepsinden geçerken doğrusal geri beslemeli yapıdan elde edilen bitlerin poker testinden geçemediği gözlemlenmiştir.

Tablo 11. q=7 Durumu için FIPS Test Sonucu

FIPS test	Doğrusal geri besleme	Çapraz geri besleme	Beklenen değer
Koşu 0	2537	2541	%1 2267 - 2733
	1247	1243	%2 1079 - 1421
	602	614	%3 502 - 748
	302	318	%4 223 - 402
	172	165	%5 90 - 223
	172	160	%6 90 - 223
Koşu 1	2496	2517	%1 2267 - 2733
	1290	1262	%2 1079 - 1421
	645	636	%3 502 - 748
	301	305	%4 223 - 402
	129	154	%5 90 - 223
	172	163	%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

Tablo 11'deki sonucu veren Verilog kodu Şekil 10 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-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;

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-15]=out3;
end
```

(b)

Şekil 10: Çapraz (a) ve Doğrusal (b) LFSR Verilog Sentezleme Programı

X^9+X^6+1 ve X^9+X^3+1 denklemi için çapraz ve doğrusal geri beslemeli yapılardan elde edilen bit dizelerinin FIPS test sonucu Tablo 12'deki gibidir. Çapraz geri beslemeli yapıdan elde edilen bitler FIPS testlerinin hepsinden geçerken doğrusal geri beslemeli yapıdan elde edilen bitlerin koşu testinin 6. kademesinden geçemediği gözlemlenmiştir.

Tablo 12. q=9 Durumu İçin FIPS Test Sonucu

FIPS test	Doğrusal geri besleme	Çapraz geri besleme	Beklenen değer
Koşu 0	2383	2501	%1 2267 - 2733
	1034	1322	%2 1079 - 1421
	953	578	%3 502 - 748
	239	340	%4 223 - 402
	316	155	%5 90 - 223
	79	152	%6 90 - 223
Koşu 1	2702	2495	%1 2267 - 2733
	1350	1238	%2 1079 - 1421
	477	675	%3 502 - 748
	238	294	%4 223 - 402
	158	185	%5 90 - 223
	158	141	%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

Tablo 12'deki sonucu veren Verilog kodu Şekil 11 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-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;

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-15]=out3;
end
```

(b)

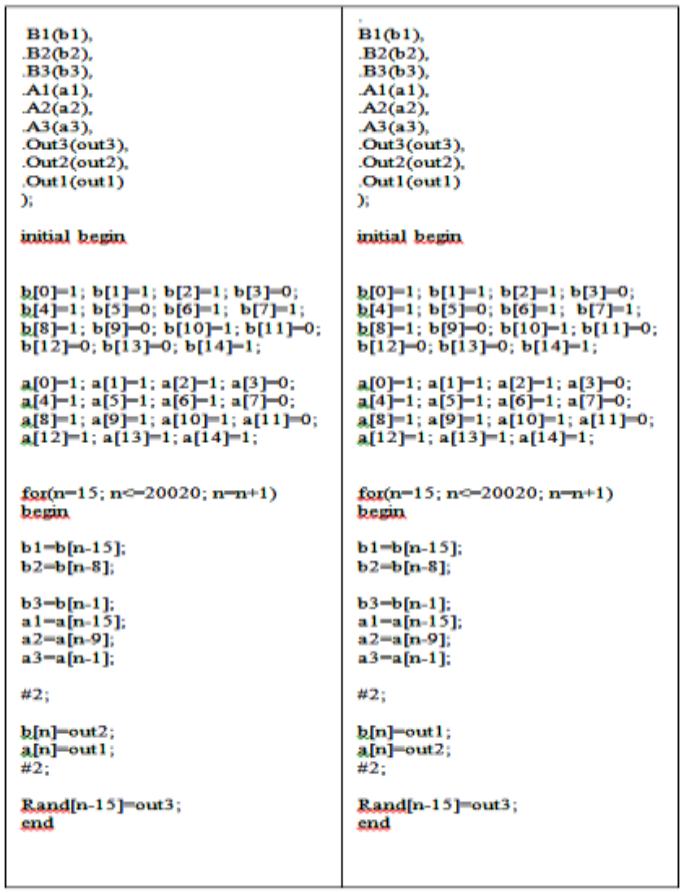
Şekil 11: Çapraz (a) ve Doğrusal (b) LFSR Verilog Sentezleme Programı

$X^{15}+X^8+1$ ve $X^{15}+X^7+1$denklemi için çapraz ve doğrusal geri beslemeli yapılardan elde edilen bit dizelerinin FIPS test sonucu Tablo 13'deki gibidir. Çapraz ve doğrusal geri beslemeli yapılardan elde edilen bitler FIPS testlerinin hepsinden başarıyla geçmiştir.

Tablo 13. q=15 Durumu İçin FIPS Test Sonucu

FIPS test	Doğrusal geri besleme	Çapraz geri besleme	Beklenen değer
Koşu 0	2593	2535	%1 2267 - 2733
	1253	1226	%2 1079 - 1421
	637	630	%3 502 - 748
	296	327	%4 223 - 402
	147	143	%5 90 - 223
	155	159	%6 90 - 223
Koşu 1	2563	2482	%1 2267 - 2733
	1243	1242	%2 1079 - 1421
	643	634	%3 502 - 748
	308	330	%4 223 - 402
	174	159	%5 90 - 223
	152	165	%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

Tablo 13'deki sonucu veren Verilog kodu Şekil 12 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
```

```
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
```

(a) (b)

Şekil 12: Çapraz (a) ve Doğrusal (b) LFSR Verilog Sentezleme Programı

Tablolar 11-13'te iki denklemli doğrusal ve çapraz FSR sonuçları karşılaştırmalı olarak verilmiştir. Bu Tablolardan da 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, 2005).

Tablolar 11-13'ten görüleceği gibi çapraz LFSR yapısı Tausworthe tanımına uygun olarak sadece iki parametrenin katsayısını 1 tutarak, denklem derecesini q=7 ye kadar düşürülmesi rağmen yeterli rastgelelik vermiştir.

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

6. SONUÇ

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. Tausworthe 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 geri 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 geri 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. Literatürde 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 geribeslemeli yapı başarılı sonuçlar alınmış olması çapraz geri beslemenin Tausworthe'nin yapısındaki bellekli yapısının kırıldığını düşündürmektedir. Çapraz geribeslemeli 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.

Yazarların Katkısı

Bu çalışmada Minare HASANBEYLİ fikir, araştırma, veri toplama, analiz, kaynak taraması ve makalenin yazımı konusunda katkı sağlamıştır. Vedat TAVAS fikir, eleştiri, yorum ve makalenin yazımı konusunda katkıda bulunmuştur.

Çıkar Çatışması Beyanı

Yazarlar arasında herhangi bir çıkar çatışması bulunmamaktadır.

Araştırma ve Yayın Etiği Beyanı

Yapılan çalışmada araştırma ve yayın etiğine uyulmuştur.

KAYNAKÇA

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.

Büyüksaraçoğlu, F., Buluş E., (2021), "Sözde Rassal Sayı Üretiminin Kriptografik Açıdan İncelenmesi", 30.04.2021, https://www.emo.org.tr/ekler/3e6f423ffcbf723_ek.pdf.

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.

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

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), 22-24 October 2010, China.

ICYSCIENCE, (2021), Rastgele sayı nedir? - Techopedia nedir?, 02.05.2021, https://tr.icyscience.com/random-number#menu-1.

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

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.

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

L'ecuyer, P., (2017), "History of Uniform Random Number Generation",2017 IEEE Proceedings of the 2017 Winter Simulation Conference, 3-6 December 2017, Montreal, Kanada.

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.

Nair, A.B., 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), 5-8 August 2018, Windsor, ON, Kanada, 428-431.

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.

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.

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

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

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