

İSTANBUL TEKNİK ÜNİVERSİTESİ ★ FEN BİLİMLERİ ENSTİTÜSÜ

**KAOTİK OSİLATÖR GİRİŞLİ ADC TABANLI
RASTGELE SAYI ÜRETECİ**

**YÜKSEK LİSANS TEZİ
Müh. Ahmet Şamil DEMİRKOL
504051201**

**Tezin Enstitüye Verildiği Tarih : 07 Mayıs 2007
Tezin Savunulduğu Tarih : 11 Haziran 2007**

Tez Danışmanı : Doç. Dr. Serdar ÖZOĞUZ

Diğer Jüri Üyeleri Prof. Dr. Cem GÖKNAR (Doğuş Ü.)

Yrd. Doç. Dr. Müştak Erhan YALÇIN

HAZİRAN 2007

ÖNSÖZ

Akademik yaşantım ve tez çalışmam boyunca bana sürekli yol gösteren ve yardımlarını esirgemeyen, her fırsatta bilgisinden çokça yararlandığım sayın Doç. Dr. Serdar Özoğuz'a saygı ve teşekkürlerimi sunarım.

Mayıs 2007

Ahmet Şamil Demirkol

İÇİNDEKİLER

TABLO LİSTESİ	v
ŞEKİL LİSTESİ	vi
ÖZET	viii
SUMMARY	ix
1. GİRİŞ	1
1.1 Rastgele Sayı Üreteçleri	1
1.2 Kaos	1
1.3 Tezde İzlenen Yol	2
2. KAOS	4
2.1 Elektronik Sistemlerde Kaos	6
3. RASTGELE SAYI ÜRETEÇLERİ	8
3.1 Giriş	8
3.2 Sözde Rastgele Sayı Üreteçleri	8
3.3 Gerçek Rastgele Sayı Üreteçleri	9
3.4 İstatistiksel Testler	9
3.4.1 FIPS 140-1	10
3.4.1.1 Monobit Testi	10
3.4.1.2 Poker Testi	10
3.4.1.3 Runs testi	10
3.4.1.4 Long Runs testi	12
3.4.2 NIST 800-22	11
3.5 Mevcut Tümdevre Tasarımları	13
3.5.1 Gürültünün Doğrudan Kuvvetlendirilmesi	13
3.5.2 Çift Osilatör Yapısı	14
3.5.3 Intel RSÜ	14
3.5.4 Kaos Tabanlı Yapılar	15

3.5.4.1	Ayrık Zamanlı Kaos Tabanlı Yapı	15
3.5.4.2	Çift Sarmallı Kaotik Osilatör Yapısı	17
4. KAOTİK OSİLATÖR GİRİŞLİ ADC TABANLI RASTGELE SAYI		
ÜRETECİ		20
4.1	Gerçeklenen Kaotik Osilatör Yapısı	20
4.1.1	Kaotik Osilatörün Yapısı	20
4.1.2	Devrenin Benzetimi ve Deneysel Sonuçlar	25
4.2	Kaotik İşaret Girişli ADC Tabanlı RSÜ Yapısı	28
4.2.1	Rastgelelik Bloğunun Tasarlanması	28
5. SONUÇ		33
KAYNAKLAR		34
ÖZGEÇMİŞ		36

TABLO LİSTESİ

	<u>Sayfa No</u>
Tablo 3.1. Runs testi koşulları.....	11
Tablo 4.1. NIST testi sonuçları	32

ŞEKİL LİSTESİ

Sayfa no

Şekil 2.1: Kaotik Bernoulli dönüşümü	5
Şekil 2.2: Jerk sisteminde $a=0.7$ için oluşan çekici	6
Şekil 3.1: Gürültünün doğrudan kuvvetlendirilmesi yöntemi	13
Şekil 3.2: Çift osilatör yapısı	14
Şekil 3.3: Intel RSÜ'sünün blok diyagramı	15
Şekil 3.4: (a) Kaotik bir dönüşüm	16
Şekil 3.4: (b) Dönüşüm sonucu oluşan çıkış	16
Şekil 3.5: $M(x)$ dönüşümüne ait durum diyagramı	16
Şekil 3.6: Yazı-tura atma olayına karşılık gelen durum diyagramı	17
Şekil 3.7: Bernoulli dönüşümü	17
Şekil 3.8: Çift sarmallı yapı ve seçilen eşik değerleri	18
Şekil 3.9: Karşılaştırıcı ve bit oluşturma blokları	18
Şekil 3.10: Von Neumann algoritması	19
Şekil 4.1: Nonlineer kayıplı integratörü sağlayan temel yapı	20
Şekil 4.2: İkinci temel hücre	22
Şekil 4.3: Üçüncü temel hücre	22
Şekil 4.4: Çift sarmallı çekicinin faz portresi	23
Şekil 4.5: Denklem (4.10)'u sağlayan kaotik devre	24
Şekil 4.6: Gerilim kontrollü akım aynası	25
Şekil 4.7: ε parametresine ait dallanma diyagramı	25
Şekil 4.8: Cadence benzetimi sonucu oluşan I_X - I_Y çekicisi	26
Şekil 4.9: Cadence benzetimi sonucu oluşan I_X 'in frekans spektrumu	27
Şekil 4.10: Ayrık olarak gerçekleştirilen devreden elde edilen V_X - V_Y çekicisi	28
Şekil 4.11: Çift osilatör yapısı	29

Şekil 4.12: Osilatör çıkışlarının zaman diyagramı	29
Şekil 4.13: ADC tabanlı yapının matematiksel modeli	30
Şekil 4.14: RSÜ blok diyagramı.....	31

KAOTİK OSİLATÖR GİRİŞLİ ADC TABANLI RASTGELE SAYI ÜRETECİ

ÖZET

Bu çalışmanın amacı kaotik osilatör girişli bir gerçek rastgele sayı üretici tasarlamaktır. Kullanılan osilatör, parametreleri kontrol edilebilen, tümleştirmeye uygun ve yüksek frekanslarda çalışabilen türdendir. Tasarlanan rastgele sayı üretici, literatürde sık kullanılan bir matematiksel modele uygun, analog/dijital dönüştürücü tabanlı bir yapıya sahiptir. Bu yapının özelliği, aynı modele uygun diğer yapılara göre daha hızlı çalışması ve kolay tasarlanabilir olmasıdır. Rastgele sayı üreticinin başarımını test etmek üzere, kaotik osilatör ve analog/dijital dönüştürücü tabanlı yapı, ayırık elemanlar kullanılarak gerçekleştirilmiştir. Sayı üreticinin çıkışından alınan veriler, National Institute of Standards and Technology (NIST) kurumu tarafından geliştirilmiş, en geçerli testlerden biri olan NIST 800-22 testine tabi tutulmuştur. Üreticinin testlerden başarıyla geçtiği görülmüştür. Literatürde bu testten geçen çok az sayıda yapının olması, tasarlanan üreticinin başarısını göstermektedir. Kullanılan kaotik osilatörün çalışma frekansı, sayı üreticinin çıkış hızını doğrudan etkilemektedir. Bu sebeple osilatör devresi mümkün olduğunca yüksek frekanslarda çalışmalıdır. Spectre simülatörüyle yapılan benzetim sonucu, kaotik osilatörün çalışma frekansının 60 Mhz mertebesinde olduğu görülmüştür. Bu da rastgele sayı üreticinin çıkış hızının birkaç on Mbit/s mertebesinde olduğu anlamına gelir. Bunun yanı sıra, analog/dijital dönüştürücü tabanlı yapı, mevcut modele uygun olacak şekilde tümdevre olarak gerçekleştirildiğinde, ayırık tasarıma göre çok daha basit olarak gerçekleştirilebileceği unutulmamalıdır.

CHAOTIC OSCILLATOR INPUT ADC BASED RANDOM NUMBER GENERATOR

SUMMARY

The purpose of this study is to design a chaotic oscillator input random number generator. The oscillator used has controllable parameters, is appropriate for IC design and can operate at high frequencies. The proposed random number generator, convenient with a mathematical model used frequently in literature, has an analog/digital convertor based structure. The specialty of this structure is that, it works faster and can be designed easier with respect to the other structures having the same model. In order to test the random number generator, the chaotic oscillator and the analog/digital convertor based structure are implemented using discrete components. The data got from the output of the number generator is tested by one of the most acceptable tests suite named NIST 800-22 which is developed by National Institute of Standards and Technology (NIST). It is seen that the generator passed the tests successfully. It shows the success of the designed generator that there is only a few number of generators passing the test in the literature. The operation frequency of the used oscillator affects directly the speed of the random number generator. For this reason, the oscillator must work at as high frequencies as possible. From Spectre Spice simulator, it is observed that the operating frequency of the oscillator is at the order of 60 Mhz which means that the output rate of the random number generator is at the order of tens of Mbit/s. Additionally, it must be noted that the analog/digital converter based structure can be implemented as an integrated circuit easier than the discrete design, still convenient with the same mathematical model.

1. GİRİŞ

1.1 Rastgele Sayı Üreteçleri (RSÜ)

Rastgele sayı üreteçleri, çıkışındaki sayılar istatistiksel olarak birbirinden bağımsız olan sistemlerdir. Rastgele sayı üreteçleri bilgisayar benzetimleri, sayısal analiz uygulamaları, istatistiksel analiz, Monte Carlo metodunun kullanıldığı uygulamalar ve özellikle şifreleme gibi alanlarda sıkça kullanılmaktadır. Örneğin, şifreleme algoritmalarının güvenilirliği, rastgele sayı üreteçlerinin ürettiği sayılara bağlıdır. Bu sayılar istatistiksel olarak rastgele ise, yani önceki çıkışlara bakarak daha sonrakiler tahmin edilemezse üreticimizin istatistiksel özelliği iyi demektir. Başka bir deyişle iyi bir şifreleme iyi bir RSÜ gerektirir.

RSÜ'leri kendi aralarında gerçek ve sözde RSÜ'ler şeklinde ikiye ayırmak mümkündür. Uygulamanın amacına göre bu iki yapıdan biri tercih edilmektedir. Gerçek RSÜ'lerin çalışması gürültü gibi doğal süreçlerin ölçümüne dayanırken, sözde RSÜ'ler ise sayısal algoritmalar gibi deterministik süreçleri kullanmaktadır. Şifreleme gibi güvenliğin önemli olduğu uygulamalarda gerçek RSÜ'lerin kullanılması zorunlu iken, bilgisayar benzetimlerinde kullanılmak üzere sözde RSÜ'lerin başarımları yeterli olmaktadır [12].

1.2 Kaos

Kaos terimi basit dinamik sistemlerin, görünürde karmaşık ve öngörülenden çok farklı şekilde ortaya çıkan dinamik davranışını tanımlamak için kullanılmaktadır. Bu sistemlerin davranışları periyodik olmayan bir özelliğe sahip olmakta ve kolaylıkla rastgele davranış ile karıştırılabilmektedir. Son yirmi yılda incelenen çoğu kaotik sistem, düşük dereceli (en fazla dördüncü dereceden) ve gürültü benzeri hiçbir terim içermeyen deterministik sistemlerdir. Kaotik sistemlerin bu determinizm özelliği sayesinde bunların bir andaki durumu tam olarak bilindiğinde, sonraki herhangi bir andaki durumu da tam olarak belirlenebilmektedir. Ancak bu durum aperiodyklik ve rastgele benzeri dinamik davranış ile çelişmektedir. Bu özellik, kaotik sistemlerin

başlangıç koşullarına üstel duyarlı olması ile açıklanabilmektedir. Kaosu doğal işaretlerden ayıran en belirgin özelliği, deterministik olmasına rağmen doğal işaretlerin aksine nasıl davranacağını tahmin edilemez oluşudur. Bu sayede, kaos işaretlerinin entropi kaynağı olarak kullanıldığı GRSÜ'ler önerilmiştir [3,7,11,12].

Kaos teorisi, ilk olarak 1960'lı yıllarda ortaya çıkmıştır. O yıllarda, hava tahmini yapabilmek için kendi meteorolojik modelini oluşturmaya çalışan Edward Lorentz, bilgisayarla yaptığı sayısal analizlerden elde ettiği sonuçlar, modelinin iyi çalıştığını gösterince, çalışmasını hızlandırmak için, aldığı verileri yuvarlayarak kullanmaya başlar. Ancak bu durumda sonuçlar çok hızlı bir şekilde farklılaşmakta ve artık tahmin edilemez hale gelir. Bu olayı tekrarladığında yine aynı farklılaşma ve belirsizlik ortaya çıkar. Böylece Lorentz, farkında olmadan kaos teorisinin temellerini atmış olur.

Kaos, daha sonraları fizik ve matematik gibi temel alanlarda çok geniş bir şekilde incelenmeye başlanmıştır. Elde edilen gelişmeler sonucu kaos, hava tahmininden sigara dumanının havada yayılışına, nüfus biliminden tek hücreli organizmaların yaşayışına kadar birçok araştırma konusunda model olarak kullanılmaya başlandı. Kaos, günümüzde birçok alanda hiç farkedilmeden kullanılmaktadır [1].

Elektronik olarak kaotik bir sistem, ilgili dinamik denklemleri sağlayan devrelerin gerçekleşmesiyle elde edilebilir. Bunun ilk uygulaması 1983'te ortaya çıkan Chua devresidir. Bu devre daha sonra bir çok kez incelenmiş ve çeşitli uygulamalarda kullanılmıştır. Chua devresi gibi kaotik işaret üreten devrelerin en büyük özelliği, bu devrelerde lineer olmayan bir elemanın ve negatif direnç gibi enerji sağlayan aktif bir elemanın kullanılmasıdır [4].

1.3 Tezde İzlenen Yol

Bu tezin amacı, sürekli zaman kaotik osilatör girişli yeni bir gerçek rastgele sayı üretici tasarlamaktır.

İkinci bölümde kaotik sistemlerin genel özellikleri tartışılacak, elektronik olarak ayrık ve sürekli zaman kaotik sistemlerin gerçekleşmesinden bahsedilecektir.

Üçüncü bölümde rastgele sayı üreticilerinden detaylı olarak bahsedilecek, rastgeleliğin sağlanıp sağlanmadığını kontrol eden istatistiksel testler açıklanacak ve mevcut örneklerle yer verilecektir.

Dördüncü bölümde gerçekleşen kaotik osilatör yapısı ve gerçekleşen rastgele sayı üretici matematiksel modeliyle anlatılacak ve RSÜ devresinin çıkışının istatistiksel test sonuçları verilecektir.

Beşinci bölümde çalışmayla ilgili sonuçlara yer verilecek ve bu sonuçlarla ilgili yorum yapılacaktır.

2. KAOS

Kaosun, günümüzde herkes tarafından kabul edilen kesin bir tanımı olmamasına rağmen kabul edilen başlıca özellikleri vardır. Kaotik sistemler, uzun vadede periyodik olmayan ve başlangıç koşullarına hassas deterministik sistemlerdir. Mevcut bir kaotik sistemin tüm başlangıç değerleri ve denklemleri bilinse dahi, fiziksel olarak sonsuz hassasiyette ölçüm yapılamayacağından başlangıç değerleri arasında farklar olacaktır ve böylece uzun vadede davranışlar üstel olarak farklılaşacaktır. Kaotik sistemlerin başka bir özelliği de nonlinear olmaları ve karmaşık bir çözüme sahip olmalarıdır [1].

Yukarıda belirtilen özelliklerin belirlenebilmesi için çeşitli yöntemler geliştirilmiştir. Kaotik sistemlere ait başlıca özellikler, otokorelasyon fonksiyonunun üstel olarak azalması, güç spektrumunun gürültü yapısına sahip olması, Poincare kesit düzleminin bir bölümünün tamamen ve düzensiz bir şekilde dolması, Lyapunov üstellerinin pozitif olmasıdır [1].

Kaotik sistemleri matematiksel modellerine göre ayrık ve sürekli zaman sistemler olarak ikiye ayırmak mümkündür.

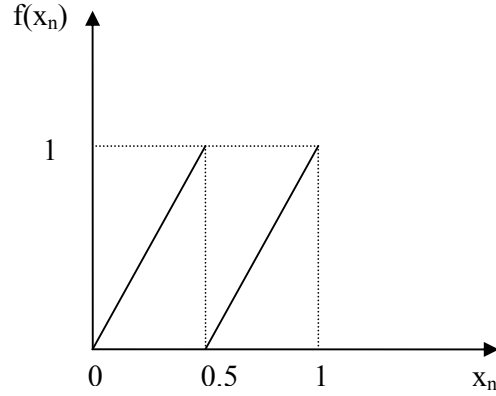
Ayrık zamanlı yapılarda, kaotik olduğu bilinen bir $f(x)$ dönüşümüyle, $x_{n+1} = f(x_n)$ olacak şekilde ayrık ve kaotik x_n değerleri üretilir. $f(x)$ fonksiyonunun kaotik olup olmadığı, başlangıç koşullarını değiştirdikçe, x_n değerlerinin çok farklı şekilde değiştiğini gözlemleyerek anlaşılabılır. Ayrık kaotik dönüşüme literatürde çok iyi bilinen Bernoulli dönüşümü (2.1) örnek olarak verilebilir [11].

$$x_{n+1} = 2x_n \mod 1, x_0 \in [0,1) \quad (2.1)$$

Denklem (2.1)'deki ifadeyi daha genel bir gösterim olan parçalı sürekli zaman bir fonksiyonla, (2.2)'deki gibi yazabiliriz.

$$f(x) = \begin{cases} 2x & 0 \leq x < 0.5 \\ 2x-1 & 0.5 \leq x < 1 \end{cases}, x_0 \in [0,1) \quad (2.2)$$

Aşağıda bu parçalı sürekli zaman dönüşüme ait grafik verilmiştir.



Şekil 2.1: Kaotik Bernoulli dönüşümü

Bernoulli dönüşümünü elektronik olarak uygulamak için (2.2)'deki fonksiyon yerine, daha basit bir dönüşüm mevcuttur. İlk olarak $[0,1)$ aralığında seçilen x_n değeri ikilik sayı tabanında, örneğin $x_n=0,782265625 = (0,01100101)_2$ şeklinde, ifade edilir. Daha sonra ikilik sistemde, x_n değerini iki ile çarpmaya denk gelen sola bir bit öteleme işlemi yapılır. Son olarak, mod alma işlemini gerçeklemek için en anlamlı bit 0 olarak yazılır.

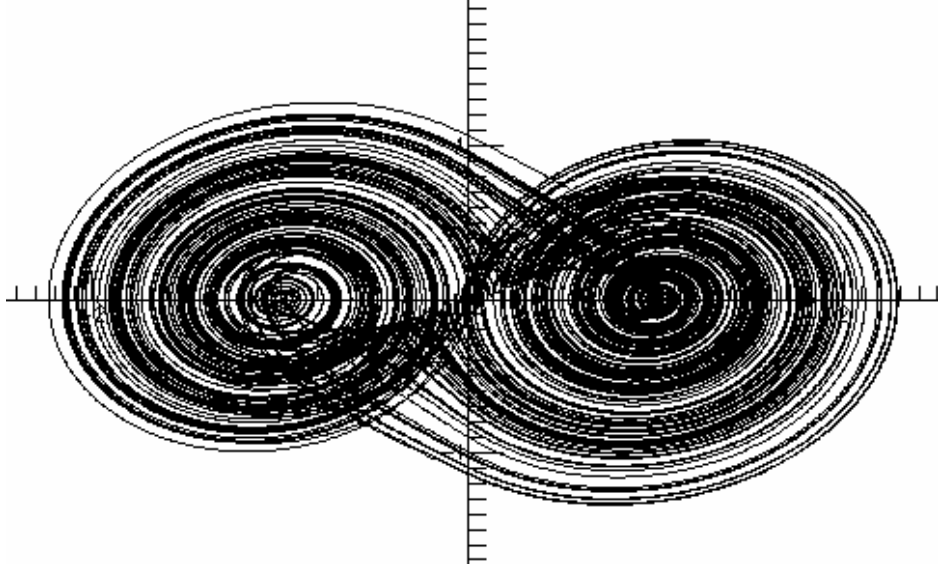
Bernoulli dönüşümünde x_0 başlangıç değeri rasyonel bir sayı seçildiğinde, bu değere karşılık düşen çözüm her zaman periyodiktir. Bunun aksine, başlangıç değeri irrasyonel seçildiği zaman ise çözümler aperiodyiktir. Bu sayede, bu ayrık dönüşümün iki temel özelliğe sahip olduğu gösterilebilir [11].

1. $f(x)$ 'in sayılabilir sonsuz tane periyodik, sayılamaz sonsuz tane aperiodyik çözümü vardır.
2. $f(x)$ fonksiyonu başlangıç değerlerine çok duyarlıdır.

Sürekli zaman kaotik sistemler, $x(t)$ bir vektör olmak üzere $\frac{d(x(t))}{dt} = f(x(t))$ olacak şekilde adi diferansiyel denklemlerle (ODE) ifade edilirler. Literatürdeki çalışmalarda görülmüştür ki kaotik sistemler oluşturmak için basit yapıda üçüncü dereceden diferansiyel bir denklem takımı çoğu zaman yeterli olmaktadır [4]. Bilinen bir denklem takımıyla oluşturulan sistemin faz portresi çizdirilerek, $f(x)$ 'in parametrelerine bağlı olarak kaotik davranış detaylı olarak incelenebilir. Buna örnek olarak jerk sistemini verebiliriz.

$$\frac{d}{dt} \begin{bmatrix} x_1(t) \\ x_2(t) \\ x_3(t) \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -a & -a & -a \end{bmatrix} \begin{bmatrix} x_1(t) \\ x_2(t) \\ x_3(t) \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ a \operatorname{sgn}(x_1(t)) \end{bmatrix} \quad (2.3)$$

Denklem (2.3)'te görülen sgn fonksiyonu, $x_1(t) < 0$ için -1 ve $x_1(t) > 0$ için $+1$ üreten, işaret fonksiyonudur. Denklemde ayrıca değişken bir a parametresi vardır. Bu parametrenin değerini değiştirerek farklı faz portreleri elde etmek mümkündür. Şekil 2.2'de buna bir örnek verilmiştir.



Şekil 2.2: Jerk sisteminde $a=0.7$ için oluşan çekici

2.1 Elektronik Sistemlerde Kaos

Kaotik işaretlerin, başlangıç koşullarına aşırı hassaslık ve aperiodyk davranış gibi, belirtilen özelliklerinden dolayı elektronik sistemlerde kullanımı gitgide yaygınlaşmaktadır. Kaosun başlıca kullanıma amacı şifrelemedir. Bunun yanı sıra, analog işaret işleme, güç elektroniği ve sayısal haberleşme alanlarında gürültü özelliği taşıyan işaretlere ihtiyaç duyulmakta ve kaotik işaretler gürültü işareti yerine kullanılmaktadır [2,3].

Elektronik devrelerde ayrık veya sürekli zaman kaotik sistemler, ilgili denklemleri sağlayan devrelerle gerçekleştirilir. Kaos, RSÜ'lerde kullanılmak üzere, elektronik devrelerde ilk olarak ayrık olarak gerçekleştirilmeye başlanmıştır. Ayrık kaotik

sistemleri gerçeklemek için ilk olarak parçalı sürekli zaman gibi uygun bir kaotik dönüşüm belirlenir ve bu dönüşüm aracılığıyla kaotik işaret üretilir. Fakat ayrıık kaotik fonksiyonların oluşturulması zor olduğundan ve tasarlanan devreler dışarıdan bir saatle kontrol edildiğinden hızları yetersiz kalmaya başlamış ve alternatif yöntemler aranmaya başlanmıştır.

Sürekli zaman kaotik devreler, çok yüksek hızda çalışabildiğinden ve çoğu sürekli zaman sistemlerin kolay tasarlanabilmesinden dolayı dikkat çekmiş ve yeni bir inceleme alanı oluşturmuşlardır. Böylece yakın zamanda yeni sürekli zaman kaotik osilatörler tasarlanmaya başlanmıştır [5,8,9,12]. Sürekli zaman kaotik bir osilatörün çıkışı, periyotları birbirinden farklı sonsuz sinüs bileşeni içerir. Çıkıştaki bileşen sayısı arttıkça işaret gürültüye benzemekte ve osilatörün de kalitesi artmaktadır. Bunun yanı sıra, her temel kaotik osilatör içinde bir sinüs osilatörü vardır. Bu sinüs osilatörünün denklemlerinde yeni parametreler oluşturmak için, devreye yeni elemanlar eklenir. Bu sayede sinüs osilatörü kaotik osilatöre dönüştürülür. Bu nedenle çıkışta ana bir sinüs bileşeni olduğu unutulmamalıdır. Bu iki özellik, frekans spektrumu incelenerek kolayca görülebilir. Kaotik osilatörün performansını belirleyen başka bir faktör de, osilatörün kontrol edilebilir ve birbirinden bağımsız parametrelerinin olması ve sistemi kaosta tutacak bu parametrelerin geniş bir aralığa sahip olmasıdır.

Kaotik devreler arasında en çok inceleneni ve en iyi bilineni, bir sürekli zaman sistemi olan Chua devresidir. Bu devrenin temel özelliği, lineer olmayan bir eleman ve negatif direnç gibi enerji üreten aktif bir eleman bulundurmasıdır. Bunun yanı sıra, yukarıda bahsedildiği gibi Chua devresinin içinde de bir LC sinüs osilatörü bulunmaktadır. Yani bir sinüs osilatörü kaotik bir osilatöre dönüştürülmüştür. Bu yöntemle başka uygulamalarda da başvurulmaktadır. Chua devresinin diğer bir özelliği de denklemlerindeki parametrelerin bağımsız oluşu ve bu sayede de çok çeşitli kaotik çekicilerin üretilebilmesidir [4].

3. RASTGELE SAYI ÜRETEÇLERİ

3.1 Giriş

Rastgele sayı üreteçleri, çıkışı rastgele sayılar olan sistemlerdir. Rastgele sayı dizileri, aralarında korelasyon bulunmayan, istatistiksel olarak birbirinden bağımsız sayılardan oluşurlar. Rastgele sayılara, bu özelliklerden dolayı, rastgele süreçlerin modellerinin oluşturulması, sayısal analiz yöntemleri, Monte Carlo metodu uygulamaları gibi bilgisayar uygulamalarında ihtiyaç duyulmaktadır. Bununla beraber, sayısal haberleşmenin çok hızlı bir şekilde yaygınlaşması ve bu nedenle gizlilik içeren verilerin internet gibi herkese açık sistemler kullanılarak iletilmesi şifrelemenin ve bunun bir sonucu olarak rastgele sayılara olan ihtiyacın artmasına neden olmuştur. Böylece rastgele sayılar ve onları üreten rastgele sayı üreteçleri büyük önem kazanmıştır [12].

Rastgele sayı üreteçleri, kullanma alanının yaygınlığından dolayı çok çeşitli şekilde gerçekleştirilmektedirler. Bu nedenle çalışma prensiplerine göre RSÜ'leri kendi aralarında iki temel gruba ayırabiliriz: Gerçek rastgele sayı üreteçleri (GRSÜ) ve sözde rastgele sayı üreteçleri (SRSÜ).

3.2 Sözde Rastgele Sayı Üreteçleri

SRSÜ'ler, sonlu durum makinalarıyla gerçekleştirilen deterministik bir algoritmayla, sayı üreten sistemlerdir [7]. GRSÜ'lerle karşılaştırıldığında, kolay gerçekleştirilme ve düşük maliyetle üretilme gibi avantajları vardır. Fakat kullanılan algoritmalar deterministiktir ve bu nedenle çıkışlar istenen şekilde tam rastgele değildir. Girişteki algoritma bilindiğinde, herhangi bir andaki değerine bakarak sonraki çıkışlar tahmin edilebilmektedir. Bu da gizlilik isteyen şifreleme algoritmalarında kullanımını kısıtlar. Kısaca GRSÜ'lere kıyasla istatistiksel olarak başarımları düşük RSÜ'lerdir. Bunun yanı sıra sayısal analiz ya da fiziksel süreç modelleme gibi daha düşük istatistiksel kalitede rastgeleliğin yeterli olduğu durumlarda tercih edilmektedirler.

3.3 Gerçek Rastgele Sayı Üreteçleri

GRSÜ'ler, girişleri deterministik olmayan, doğal süreçlerin rastgeleliğini kullanan bir algoritmayla sayı üreten sistemlerdir. Çıkıştaki sayılar rastgele olduğundan şifrelemede çokça kullanılırlar. Donanım ve yazılım tabanlı olmak üzere iki farklı teknikle gerçekleştirilebilirler [12].

Donanım tabanlı üreteçler, yarıiletken bir diyodun ya da bir direncin ısı gürültüsü, bir osilatörün faz gürültüsü, radyoaktif bir bozulma esnasında parçacıkların yayılmaları arasında geçen süre gibi fiziksel olayların rastgeleliğini kullanır. Bu süreçler sonucu oluşan işaretler de kendi aralarında ilintili olabileceğinden dolayı, tam rastgeleliği sağlamak için çıkış yeniden basit bir algoritmaya tabi tutulabilir [10,12].

Yazılım tabanlı üreteçler, sistemin saati, mouse hareketleri arasındaki süre, sabit diske erişim gibi bilgisayar tabanlı olayları temel alır. Yazılım tabanlı üreteçleri gerçeklemek, donanım tabanlıları gerçeklemekten daha zahmetli ve daha az güvenilirdir. Örneğin, Netscape'in rastgele sayı üreticinin temel aldığı verinin, günün saati ve süreç numarası olduğu tespit edilebilmiştir. Bu nedenle yazılım tabanlı üreteçler ikinci planda kalmıştır.

3.4 İstatistiksel Testler:

Sayı üreteçlerinin çıkışının rastgele olduğu, matematiksel olarak kanıtlanamamasına rağmen, geçerli istatistiksel testleri uygulayarak, sayı dizilerinin rastgele olmadığını veya bunun tersini söyleyebiliriz. Bu testler, üreticinin çıkışının gerçek bir rastgele diziden beklenenleri karşılayıp karşılamadığını söyler. Ayrıca testlerin sonuçlarına bakılarak rastgele sayı üreticinin kalitesi hakkında yorum yapılabilir. Bir sayı dizisinin rastgele olduğunu söylemek için, tüm testlerden geçmesi gerekir. Sadece bir tane test başarısız olsa bile dizi rastgele kabul edilemez.

Bilinen testlerden başlıca iki tanesi, Ulusal Standartlar ve Teknoloji Enstitüsü tarafından yayınlanan (NIST) FIPS 140-1 ve NIST 800-22'dir [13]. FIPS 140-1 testi blok uzunluğu küçük olan verilerin testinde kullanılır. NIST 800-22 testi ise görece uzun bloklardan oluşan verileri test etmek için kullanılır ve FIPS 140-1 testine göre kriterleri daha zorlayıcıdır. Bu nedenle kullanım amacına uygun olarak bu testlerden biri tercih edilebilir.

3.4.1 FIPS 140-1

FIPS 140-1 testi dört tane ayrı testten oluşur. RSÜ'nün çıkışından alınan ve 20000 tane bit içeren bir bit dizisi dört teste birden tabi tutulur ve dizinin rastgele olabilmesi için tüm testlerden geçmesi gerekir. Bu dört test aşağıda açıklanmıştır.

3.4.1.1 Monobit Testi

Bu testin amacı, bit dizisindeki 0 ve 1 sayısının rastgele bir diziden beklendiği gibi olup olmadığını tespit etmektir. Testin başarılı olabilmesi için 20000 bitlik bir dizideki '1' sayısının $9654 < n < 10346$ aralığında olması gerekir.

3.4.1.2 Poker Testi

Bu testte, 'k' bit içeren dizi, $k \geq 5 \cdot 2^m$ olacak şekilde, üstüste çakışmayan m-bitlik parçalara ayrılır ve i. parça n_i diye adlandırılır. Rastgele bir diziden beklenen, tüm m-bitlik blokların k uzunluklu bir dizide aynı sayıda birbirini tekrar etmesidir. Testin başarılı olabilmesi için,

$$X = \frac{2^m}{k} \left(\sum_{i=1}^{2^m} n_i^2 \right) - k \quad (3.1)$$

formülüyle hesaplanan X değerinin, $k=20000$ ve $m=4$ için, $1.03 < X < 57.4$ aralığında olması gerekir.

3.4.1.3 Runs Testi:

Elimizdeki bit dizisinin bu testten başarıyla geçmesi için, dizide ardarda gelen '1' ve '0' lardan oluşan çeşitli uzunluktaki blokların (runs'ların) sayısının tablo 3.1'de belirtildiği gibi olması beklenir. 6 bitten daha uzun bloklar 6 bitlik olarak kabul edilmektedir.

3.4.1.4 Long Runs Testi

Long runs testinin başarılı olabilmesi için 20000 bitlik bir bit dizisindeki blok (runs) uzunluklarının hepsinin 34'ten küçük olması gerekir.

Tablo 3.1: Runs testi koşulları

Blok uzunluğu	Blok sayısı aralığı
1	2267-2733
2	1079-1421
3	502-748
4	223-402
5	90-223
6+	90-223

3.4.2 NIST 800-22

NIST 800-22 testi, FIPS 140-1 testine göre çok daha güçlü bir testtir. FIPS 140-1 testini geçen bir bit dizisi NIST 800-22 testinden kalabilir. Bu nedenle ciddi uygulamalarda NIST 800-22 tercih edilmektedir. NIST 800-22 kendi içinde 15 tane ayrı testten oluşur. Teste tabi tutulan bit dizisinin başarılı olabilmesi için tüm testleri başarıyla geçmesi gerekmektedir. Aşağıda bu testlerin hepsi kısa açıklamasıyla birlikte verilmiştir:

1. *Frequency test:* Bit dizisindeki 1 ve 0 dengesini inceler.
2. *Block frequency test:* m bitlik bit bloklarının 0 ve 1 dengesini inceler.
3. *Runs test:* Dizideki 0 ve 1 bloklarının (runs) sayısını inceler.
4. *Longest run test:* Dizideki 0 ve 1 bloklarının (runs) uzunluklarını inceler.
5. *Rank test:* Sabit uzunluklu bit blokları kullanılarak, her biri bir satırı belirtecek şekilde, bir matris oluşturulur ve matrisin rankı hesaplanarak bloklar arasındaki lineer bağımlılık incelenir.
6. *FFT test:* Mevcut bit dizisinin ayrık Fourier dönüşümünü alır ve periyodikliği inceler.
7. *Non-overlapping Template Matching test:* m bitlik bir bloğun dizi içinde tekrarını inceler. Tekrar edilmesi halinde, tekrar edilen bloktan itibaren yeni bir m bitlik blok oluşturulur.

8. *Overlapping Template Matching test*: m bitlik bir bloğun dizi içinde tekrarını inceler. Tekrar edilmesi halinde, blok 1 bit ötelenerek yenisi oluşturulur.
9. *Universal test*: Dizinin veri kaybı olmadan ne kadar sıkıştırılabileceğini inceler.
10. *Linear Complexity test*: Bit dizisinin LFRS (linear feedback shift register) uzunluğuna bakarak kompleksliğini inceler.
11. *Serial test*: Tekrar eden m bitlik 2^m tane bloğun tekrar sayısının dağılımını inceler. $m=1$ için, birinci teste denktir.
12. *Approximate Entropy test*: Tekrar eden m ve $(m+1)$ bitlik blokların entropisini inceler.
13. *Cumulative Sums test*: Bit dizisini ardışık uzunluklu bloklara ayırıp blokların 1 ve 0 dengesini belirler ve bloklar arasındaki dengesizlik farkına bakar.
14. *Random Excursion test*: Bit dizisini ardışık uzunluklu bloklara ayırıp blokların 1 ve 0 dengesini belirler ve daha sonra blokların dengesinin dağılımını inceler.
15. *Random Excursion Variant test*: Bit dizisini ardışık uzunluklu bloklara ayırıp blokların 1 ve 0 dengesini belirleyip ortalama değerden sapma miktarını belirler.

NIST 800-22 testinde, teste tabi tutulacak blok uzunluğu ve blok sayısı kullanıcı tarafından belirlenir. Ayrıca yukarıda belirtilen 2, 7, 8, 9, 10, 11, 12 nolu testlerde adı geçen blok uzunluğu parametreleri olarak, kullanıcı tarafından varsayılan değerden başka bir değer belirlenebilir. Bu parametrelere göre her bir test sonucu olarak, p -değeri ve test geçme oranı olarak iki farklı değer ortaya çıkar. Teste tabi tutulan bloklardan testi geçenlerin sayısı testi geçme oranını belirler ve bu oran teste sokulan blok sayısına göre değişir. p -değeri ise testi geçen blokların performansının bir ölçütüdür. Bu değeri belirlemek için, testi geçen bloklar, başarılarına göre on gruba ayrılır. Daha sonra bu grupların sayısının dengesiyle p -değeri belirlenir. Tüm bloklar testi çok başarılı bir şekilde tamamladığında p -değeri 1'e yakın olur. Bloklar testi düşük bir derecede geçerse, p -değeri 0'a yakın bir değer alır. Test programı, hesaplanan p -değeri ve testi geçme oranı değerlerini belirleyerek test edilen veri için

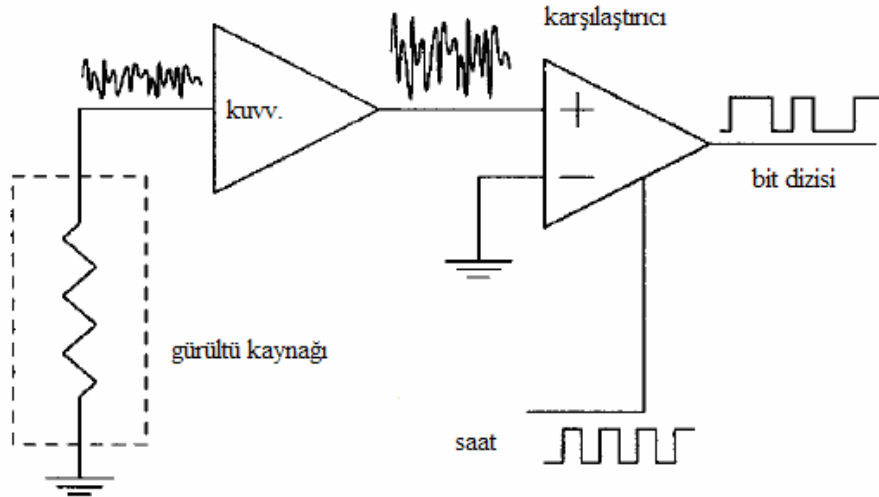
başarılı veya başarısız şeklinde sonuç belirtir. Bunun yanı sıra kullanıcı, test geçme oranı ve p-değerini aynı anda yorumlayarak test edilen dizilerin başarısı hakkında yorum yapabilir.

3.5 Mevcut Tümdevre Tasarım Örnekleri

Sayısal haberleşmenin artması ve şifrelemenin önem kazanmasıyla birlikte tümdevre olarak tasarlanan RSÜ'lere olan ihtiyaç da artmıştır. Böylece RSÜ devreleri daha kararlı çalışan, daha az güç tüketen ve daha az yer kaplayan devreler olarak gerçekleştirilmeye başlamıştır. Çeşitli tümdevre RSÜ tasarımları arasında öne çıkan bazı tasarımlar aşağıda açıklanmıştır.

3.5.1 Gürültünün Doğrudan Kuvvetlendirilmesi

Literatürde iyi bilinen gürültünün doğrudan kuvvetlendirilmesi yöntemi [10] şekil 3.1'de gösterilmiştir.



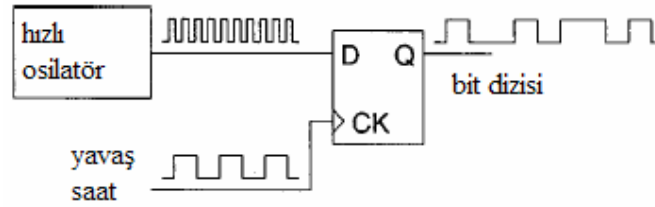
Şekil 3.1: Gürültünün doğrudan kuvvetlendirilmesi yöntemi

Bu yöntemde, gürültü kaynağı olarak bir direncin ısı gürültüsü seçilir. Bu gürültü işaretini işlenebilir kılmak için yüksek kazançlı ve bant genişlikli, saat girişi olan bir kuvvetlendirici kullanılır. Böylece gürültü işareti kuvvetlendirildikten sonra referansla karşılaştırılıp '1' ve '0' dizisi elde edilir. Yöntemin avantajı yapının gerçekleştirilmesinin kolay olmasıdır fakat besleme gürültüsü, kullanılan elemanların

gürültüsü ve benzeri gürültü etkenlerinin yalıtılması zorluk çıkardığı için bu yöntem yaygın olarak uygulanamamaktadır.

3.5.2 Çift Osilatör Yapısı

Çift osilatör yapısında, osilatörlerin faz gürültüsü rastgelelik için esas alınır. Bunun yanısıra osilatörlerin çıkışının, örneğin gerilim kontrollü osilatörlerde olduğu gibi, başka gürültü etmenleriyle oluşturulduğu da yapılan uygulamalar arasındadır. Şekil 3.2’de en temel uygulama bloğu verilmiştir [2].

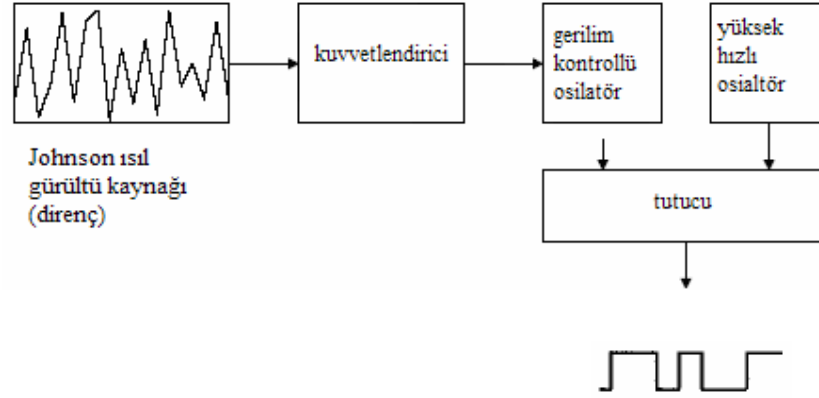


Şekil 3.2: Çift osilatör yapısı

Bu yöntemde biri hızlı diğeri yavaş iki tane osilatör bulunmakta ve D tipi flip-flop kullanılarak yavaş osilatörle hızlı osilatör örneklenmektedir. Osilatörlerin faz gürültüleri her bir örnekleme esnasında rastgelelik sağlamaktadır. Bu yapıda önemli hususlardan biri, iki osilatörün frekansları arasındaki oranı iyi bir şekilde belirlemektir. Bazı yapılarda bu oran 1:100 civarındadır. Yavaş osilatörün frekansını düşük tutmak çıkış hızını düşürmekte, hızlı osilatörün frekansını yüksek tutmak ise tasarımı zorlaştırmaktadır. Bu da tasarımın dezavantajlarından biridir. D tipi flip-flopun çıkışından elde edilen bitler ise, düşük frekans oranlarında, istenilen rastgeleliği oluşturamadığından çıkış tekrar sözde rastgele bir algorithmadan geçirilerek tam rastgele bit dizileri elde edilir.

3.5.3 Intel RSÜ

Intel firmasının ürettiği RSÜ, yukarıda bahsedilen iki ayrı yöntemin birleşmesinden elde edilmiştir [6]. Yöntemin blok diyagramı şekil 3.3’te gösterilmiştir.



Şekil 3.3: Intel RSÜ'sünün blok diyagramı

Blok diyagramdan görüldüğü gibi, rastgelelik için ilk yöntemdeki gibi bir direncin ısı gürültüsü kuvvetlendirilip, bit üretmek için kuvvetlendirici yerine gerilim kontrollü bir osilatörün girişine uygulanır. Gerilim kontrollü osilatör, çift osilatör yapısındaki yavaş osilatörün yerini tutmaktadır. Daha sonra hızlı bir osilatör, yavaş osilatörle örneklenir. Çift osilatör yapısındaki ve gürültünün kuvvetlendirilmesindeki dezavantajlar, iki bloğun beraber kullanılmasından dolayı, bu yapıda kısmen etkisini yitirmiştir. Fakat çıkış hala tam olarak rastgele olmamaktadır. Bu nedenle örnekleme sonucu oluşan bitler tekrar bir algorithmadan geçirilir ve tam rastgele bit dizisi elde edilmiş olur.

3.5.4 Kaos Tabanlı Yapılar

Düzensiz davranışları ve başlangıç koşullarına aşırı hassas oluşları nedeniyle, kaotik işaretler de rastgelelik kaynağı olarak değerlendirilmekte ve RSÜ yapımında kullanılmaktadırlar [11,12].

Kaotik sistemleri ayırık ve sürekli olarak ikiye ayırmak mümkündür. Her iki tip sistemle oluşturulan RSÜ yapıları vardır. Aşağıda her yapı için birer örnek verilmiştir.

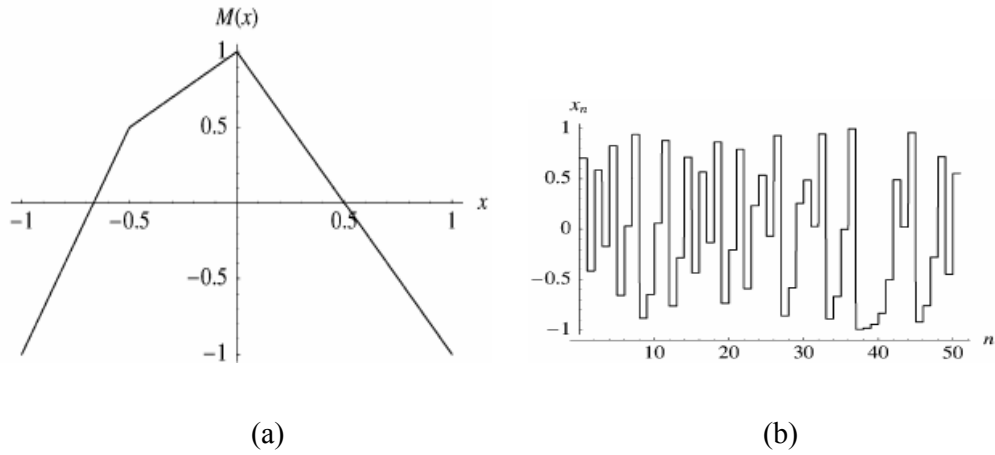
3.5.4.1 Ayırık Zaman Kaos Tabanlı Yapı:

Ayrık kaotik sistemlere dayalı yapılar, yakın zamanda fazlaca incelenmiş ve RSÜ yapılarında kullanılmışlardır [3,7,11]. Ayrık zamanlı kaosa örnek olarak [3]'teki çalışma incenebilir. İlk olarak parçalı sürekli bir Markov dönüşümü incelenmiş, bu dönüşümün durum diyagramı belirtilmiştir. Daha sonra ise, ideal rastgeleliğe iyi bir

örnek olan yazı-tura atma olayının durum diyagramı verilmiş ve bu diyagrama denk düşen dönüşümün parçalı sürekli zaman Bernoulli dönüşümü olduğu belirtilmiştir. Şimdi ilk dönüşüm örneğini ele alalım.

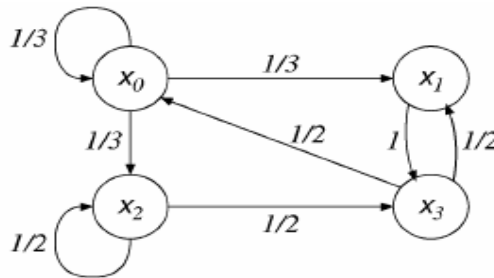
$$x_{n+1} = M(x_n), \quad M(x) = \begin{cases} 3x + 2, & x < -0.5 \\ x + 1, & -0.5 < x < 0 \\ -2x + 1, & \text{diğer} \end{cases} \quad (3.2)$$

Aşağıda $M(x)$ kaotik dönüşümünün ve bu dönüşüm sonucu oluşan x_n çıkışlarının grafiği gösterilmiştir. Bu dönüşümün özelliği, $x_i \cap M(x_i) = \emptyset$ ya da $x_i \subseteq M(x_j)$ olmasıdır. Şekil 3.4'te $M(x)$ 'e ait grafik ve bu sisteme ilişkin kaotik bir x_n çıkışı görülmektedir.



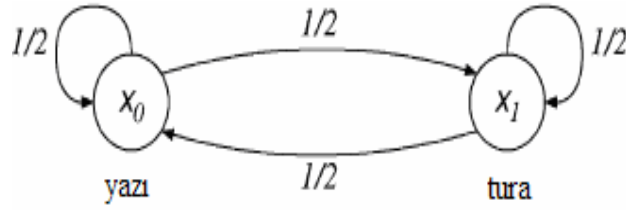
Şekil 3.4: (a) Kaotik bir dönüşüm (b) Dönüşüm sonucu oluşan çıkış

Görüldüğü gibi x_n ve $M(x)$ için $[-1, -0.5)$, $[-0.5, 0)$, $[0, 0.5)$ ve $[0.5, 1]$ şeklinde dört tane olası aralık vardır. Her x_n aralığı için yine belli sayıda $M(x)$ aralığı vardır. Buna ilişkin durum diyagramı şekil 3.5'te görülmektedir.



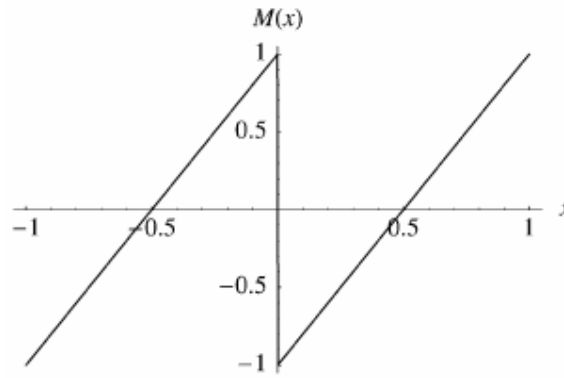
Şekil 3.5: $M(x)$ dönüşümüne ait durum diyagramı

Şekil 3.6’da ise yazı-tura atma olayına ait durum diyagramı görülmektedir.



Şekil 3.6: Yazı-tura atma olayına karşılık gelen durum diyagramı

Yazı-tura durum diyagramına karşılık gelen ayırık fonksiyon ise 2. bölümde açıklanmış olan ve şekil 3.7’de gösterilen Bernoulli dönüşümüdür.

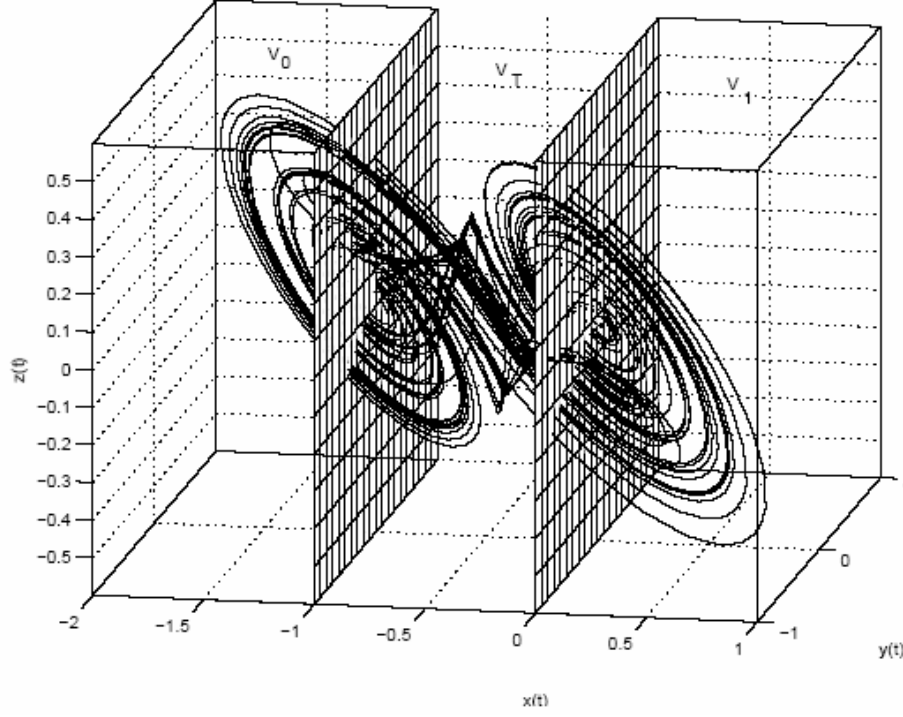


Şekil 3.7: Bernoulli dönüşümü

Bu çalışmada, Bernoulli dönüşümünü elektronik olarak gerçeklemek için, pipeline yapıda bir A/D kullanılmıştır. Bunun sebebi, pipeline türü yapıda, dönüştürücünün kuantalama hatasının matematiksel ifadesinin Bernoulli dönüşümüyle aynı olmasıdır.

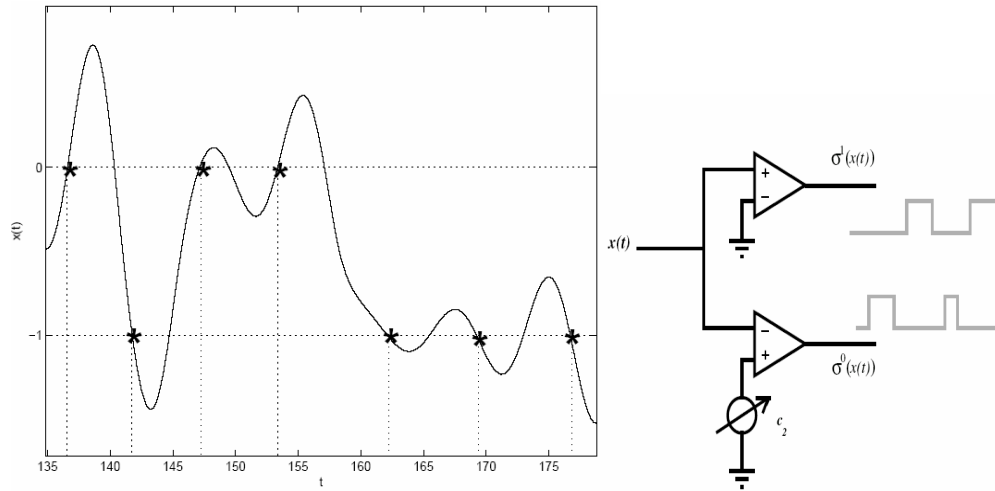
3.5.4.2 Çift Sarmallı Kaotik Osilatör Yapısı

Bu yapı, sürekli zaman çift sarmal üreten kaotik osilatör kullanılarak yapılan ilk çalışmadır [12]. İlk önce çift sarmal üreten kaotik bir devre tasarlanmış ve gerçekleştirilmiştir. Daha sonra kaotik osilatör devresinin çıkışı, biri denge noktasında ve ikincisi ise diğer denge noktası civarında olmak üzere, iki tane eşik değeriyle (0 ve -1) karşılaştırılmıştır (şekil 3.8).

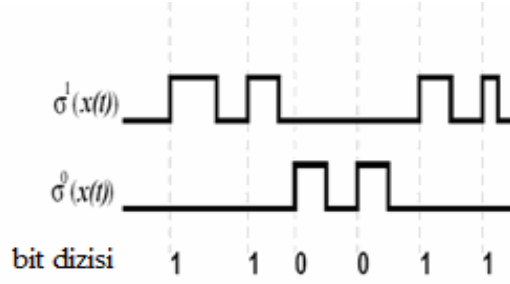


Şekil 3.8: Çift sarmallı yapı ve seçilen eşik değerleri

Kaotik osilatör devresinin eşik değerleriyle karşılaştırılan $x(t)$ işareti ve karşılaştırma bloğu şekil 3.9'da görülmektedir. $x(t)$ çıkış gerilimi pozitif değer aldığı anda ilk karşılaştırıcı 1, aksi halde 0 üretir. Çıkış gerilimi $-1V$ seviyesinin altında iken ikinci karşılaştırıcı da 1 üretmektedir. Daha sonra bu iki bit dizisinden iyi bilinen Von Neumann algoritmasıyla rastgele bit dizisi elde edilir (şekil 3.10).



Şekil 3.9: Karşılaştırıcı ve bit oluşturma blokları



Şekil 3.10: Von Neumann algoritması

Von Neumann algoritmasını, şekil 3.10'dan yararlanarak açıklayabiliriz: ilk bit dizisi 1 ve ikinci bit dizisi 0 iken çıkış 1, ilk bit dizisi 0 ve ikinci bit dizisi 1 iken çıkış 0, diğer durumlarda (00 ve 11) ise çıkış üretilmektedir.

Bu yapının dezavantajı, karşılaştırılacak eşik değerlerinin belirlenmesidir. Bu zorluğu aşmak için, eşik parametresi değiştirilirken bir yandan da çıkışın entropsinin değişimini incelenerek doğru eşik değerinin tespiti kolaylaştırılmıştır [12]. Yapının başka bir dezavantajı, çıkışta uygulanan Von Neumann algoritmasının çıkış hızını yaklaşık 4 kat azaltmasıdır. Bu nedenle bu metodu kullanmak için yüksek hızda çalışan kaotik osilatörler tercih edilmelidir.

4. KAOTİK OSİLATÖR GİRİŞLİ ADC TABANLI RASTGELE SAYI ÜRETECİ

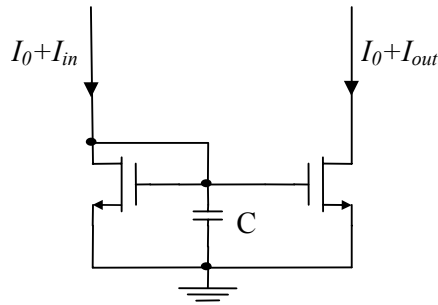
Bu tezin amacı, girişinde kaotik bir işaret olan ADC tabanlı rastgele sayı üretici tasarlamaktır. Kaotik osilatör olarak [8]'deki osilatör seçilmiş ve rastgele sayı üretmek için [6]'daki matematiksel model esas alınarak yeni bir RSÜ yapısı tasarlanmıştır.

4.1 Gerçeklenen Kaotik Osilatör

Bu çalışmada, yeni bir çift sarmallı sürekli zaman kaotik osilatör [8] kullanılmıştır. Bu osilatör devresi, CMOS üretim sürecine uygun olması, yüksek frekanslarda çalışabilmesi, parametrelerinin kontrol edilebilir oluşu ve tasarımının kolay olması gibi özellikleriyle öne çıkmaktadır.

4.1.1 Kaotik Osilatörün Tasarlanması

Gerçeklenen kaotik osilatör üç temel hücre ve bir tane karşılaştırmacı bloğundan oluşur. Temel hücreler şekil 4.1'deki temel yapıya dayanır. Burada transistörler eşlenmiş ve doyma bölgesindedir.



Şekil 4.1: Nonlinear kayıplı integratörü sağlayan temel yapı

Temel hücreyi incelersek, kapasite ve transistörün akım-gerilim bağıntılarından (4.1) ve (4.2) nolu denklemleri elde ederiz.

$$C \frac{d}{dt} V_C = i_C = -I_{out} + I_{in} \quad (4.1)$$

$$\frac{\beta}{2} (V_C - V_{th})^2 = I_o + I_{out} \quad (4.2)$$

(4.2) denkleminde V_C 'yi çekersek (4.3)'ü elde ederiz.

$$V_C = \sqrt{\frac{2(I_o + I_{out})}{\beta}} + V_{th} \quad (4.3)$$

Daha sonra (4.3)'teki V_C ifadesini (4.1)'de yerine koyarsak, (4.4) elde edilir.

$$C \frac{d}{dt} V_C = C \frac{2 I_{out}^o / \beta}{2 \sqrt{\frac{2(I_o + I_{out})}{\beta}}} = -I_{out} + I_{in} \quad (4.4)$$

(4.4)'teki ifade düzenlenirse,

$$\frac{C I_{out}^o}{\sqrt{2\beta(I_o + I_{out})}} = -I_{out} + I_{in} \quad (4.5)$$

genel denklemi elde edilir. Daha sonra $u \equiv I_{in} / I_o$, $v \equiv I_{out} / I_o$ ve $t_{norm} = t \sqrt{2\beta I_o} / C$ normalizasyonu yapılırsa (4.5) denklemi (4.6)'ya dönüşür:

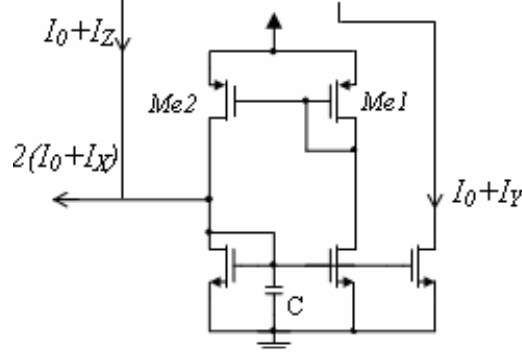
$$\dot{v} = (-v + u) \sqrt{1 + v} \quad (4.6)$$

(4.6)'da verilen denklemin $|v| \ll 1$ veya $1+v \approx 1$ alındığında kayıplı bir integratör denklemine dönüştüğüne dikkat etmekte fayda vardır. Normalize eşitliği göz önünde bulundurarak (4.1) ve (4.6) denklemlerine dikkat edersek, C kapasitesinin üzerinden akan akım denklemin sağ tarafında görülmektedir.

Şimdi, şekil 4.1'dekine benzer ikinci bir temel hücreyle denklem (4.7) 'yi oluşturalım.

$$\dot{y} = ((1 + \varepsilon)y + z - 2x + \varepsilon) \sqrt{1 + y} \quad (4.7)$$

Amacımız eşitliğin sağ tarafında verilmiş olan normalize büyüklükteki akımı, şekil 4.1'deki gibi C değerli bir kapasite üzerinden akıtmaktır. Bunun için şekil 4.2'deki devreyi önerebiliriz.



Şekil 4.2: İkinci temel hücre

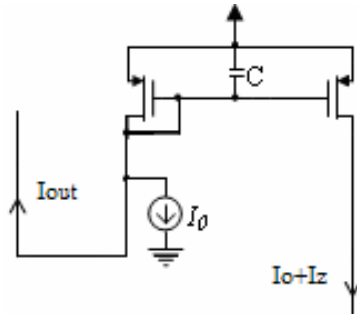
Me2 ile Me1 in arasındaki akım oranı $(2+\varepsilon)$ olmak üzere C kapasitesinin üzerinden akan akım $(1+\varepsilon)I_Y + I_Z - 2I_X + I_0\varepsilon$ olur. $y \equiv \frac{I_Y}{I_0}, x \equiv \frac{I_X}{I_0}$ ve $t_{norm} = t\sqrt{2\beta I_0}/C$ normalizasyonu yapıldıktan sonra,

$$\dot{y} = ((1+\varepsilon)y + z - 2x + \varepsilon)\sqrt{1+y} \quad (4.8)$$

elde edilmiş olur. Şimdi de üçüncü bir hücreyle denklem (4.9)'u oluşturalım.

$$\dot{z} = (-z + k \operatorname{sgn}(x - \varepsilon))\sqrt{1+z} \quad (4.9)$$

Üçüncü hücrede C değerli kapasitenin üzerinden akıtmaya çalıştığımız akım $(-I_Z + k \operatorname{sgn}(I_X - \varepsilon I_0))$ olacaktır. Bu akımı şekil 4.3'teki devreyle sağlayabiliriz.



Şekil 4.3: Üçüncü temel hücre

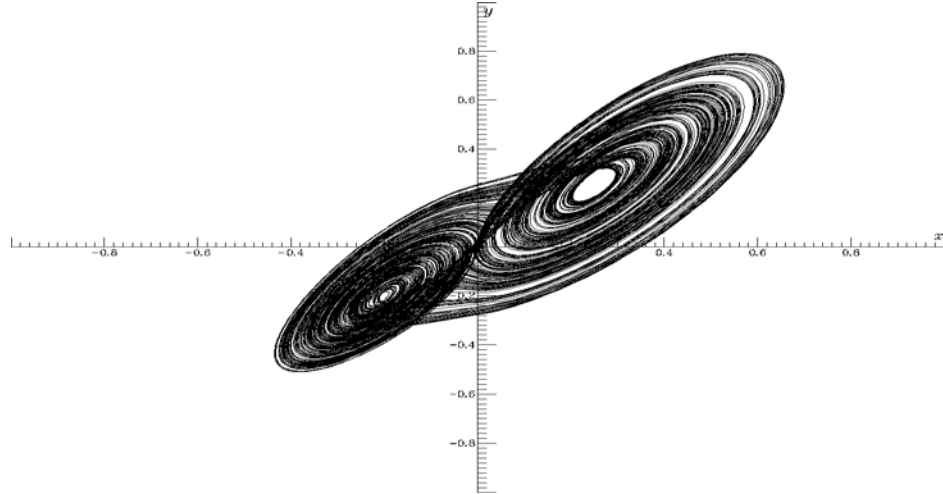
$I_{out} = kI_0 \text{sign}(I_X - \varepsilon I_0)$ olmak üzere, C kapasitesi üzerinden akmaya zorlanan akım $(-I_Z + k \text{sgn}(I_X - \varepsilon I_0))$ olur. $z \equiv \frac{I_Z}{I_0}, x \equiv \frac{I_X}{I_0}$ ve $t_{norm} = t\sqrt{2\beta I_0}/C$ normalizasyonu yapıldıktan sonra elde etmeye çalıştığımız (4.9) denklemi sağlanmış olur.

Şimdi aşağıdaki denklem takımını ele alalım:

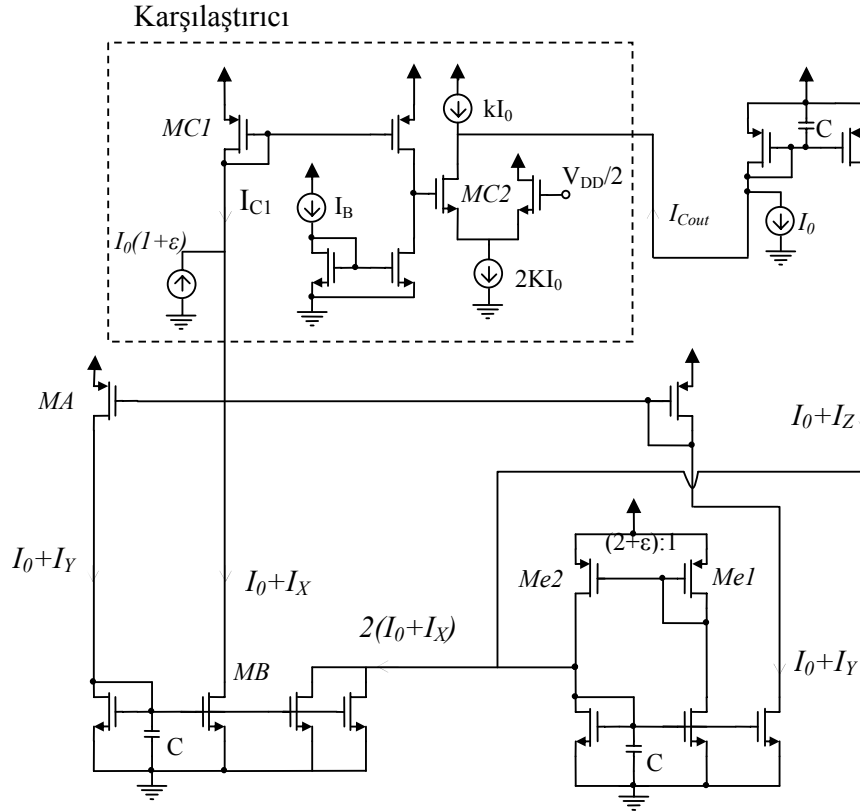
$$\begin{aligned}\dot{x} &= (-x + y)\sqrt{1+x} \\ \dot{y} &= ((1+\varepsilon)y + z - 2x + \varepsilon)\sqrt{1+y} \\ \dot{z} &= (-z + k \text{sgn}(x - \varepsilon))\sqrt{1+z}\end{aligned}\tag{4.10}$$

Görüldüğü gibi yine $|x| \ll 1$, $|y| \ll 1$ ve $|z| \ll 1$ koşulları altında (4.10)'daki üç denklem, (4.6)'yla aynı yapıda ve kayıplı integratör denklemiyle aynı formattadır. Bu sistem, değişik parametreler için kaos üretmektedir. Şekil 4.4'teki çift sarmallı çekici, $\varepsilon=0.15$ ve $k=0.2$ parametreleriyle oluşturulmuştur.

Buna dayanarak, elde ettiğimiz üç temel hücreyi birbirine bağlarsak, şekil 4.5'teki devreyi elde ederiz.



Şekil 4.4: Çift sarmallı çekicinin faz portresi

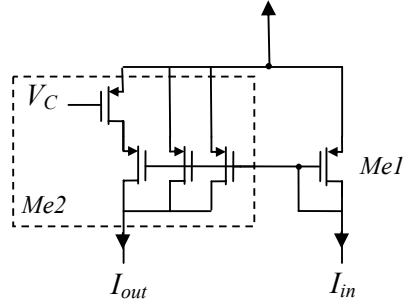


Şekil 4.5: Denklem (4.10)'u sağlayan kaotik devre

Devredeki karşılaştırıcı bloğu, (4.10)'daki işaret fonksiyonunu gerçeklemektedir. Devre analiz edilirse, M_{C1} transistörü $I_{C1} = I_X - \epsilon I_0 < 0$ için kesimde olacağından, M_{C2} de kesimde olur ve karşılaştırıcının çıkış akımı $I_{Cout} = -kI_0$ olur. $I_{C1} = I_X - \epsilon I_0 > 0$ için M_{C1} ile beraber M_{C2} de iletme geçer ve $I_{Cout} = kI_0$ olur. Başka bir deyişle karşılaştırıcının çıkışı (4.11)'deki gibidir.

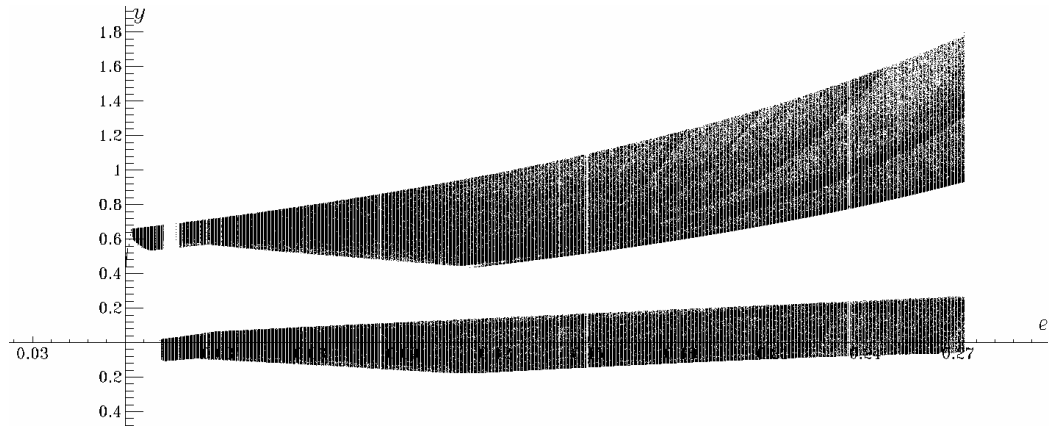
$$I_{Cout} = kI_0 \text{sign}(I_X - \epsilon I_0) \quad (4.11)$$

Denklem (4.10) incelendiğinde kaotik osilatörün en önemli parametresinin ϵ olduğu görülür. Bu nedenle ϵ parametresinin kontrol edilebilirliği açısından M_{e1} ve M_{e2} transistörlerinin $(2+\epsilon)$ akım oranı şekil 4.6'da gösterilen ayarlı akım aynasıyla sağlanabilir.



Şekil 4.6: Gerilim kontrollü akım aynası

Kritik ε parametresinin değer aralığı $[0, 0.27]$ olarak belirlenmiştir. Uygun çalışma koşulu için ε parametresinin değeri bu aralığın ortasında olacak şekilde belirlenir. Şekil 4.7’de bu parametreye ait dallanma diyagramı verilmiştir.



Şekil 4.7: ε parametresine ait dallanma diyagramı

Şekil 4.5 teki devrenin durum denklemleri düzenlenip yazılırsa,

$$\begin{aligned} C \dot{I}_X &= (-I_X + I_Y) \sqrt{2\beta(I_0 + I_X)} \\ C \dot{I}_Y &= ((1 + \varepsilon)I_Y + I_Z - 2I_X + \varepsilon I_0) \sqrt{2\beta(I_0 + I_Y)} \\ C \dot{I}_Z &= (-I_Z + k I_0 \operatorname{sgn}(I_X - \varepsilon I_0)) \sqrt{2\beta(I_0 + I_Z)} \end{aligned} \quad (4.12)$$

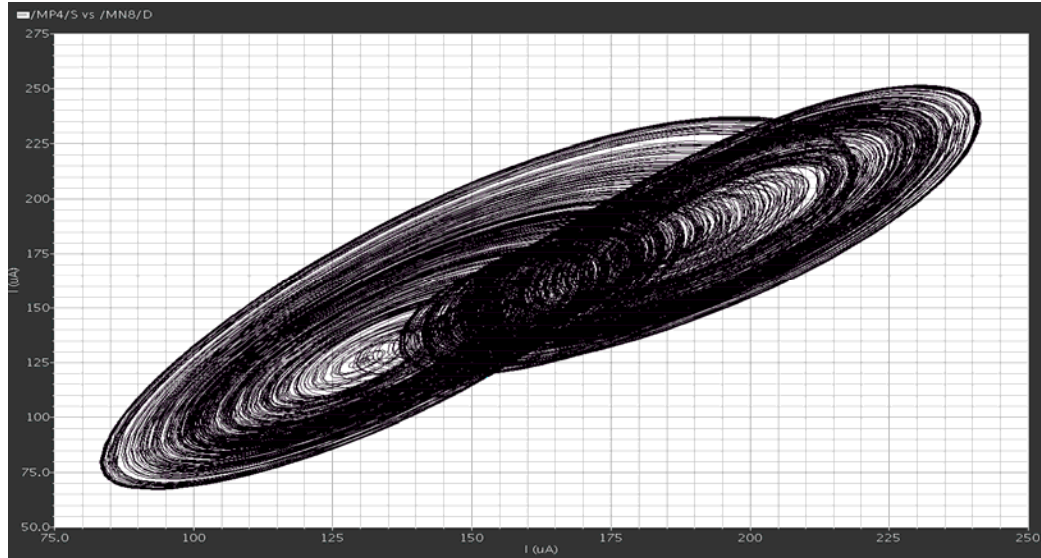
denklemleri elde edilir ve normalizasyon parametreleri olarak $x \equiv I_X / I_0$, $y \equiv I_Y / I_0$, $z \equiv I_Z / I_0$, $m = \sqrt{\beta_n / \beta_p}$ ve $t_{norm} = t \sqrt{2\beta_n I_0} / C$, yerine konursa beklendiği gibi denklem (4.8)’in elde edileceği görülür.

4.1.2 Devrenin Benzetimi ve Deneysel sonuçlar:

Şekil 4.5’teki devrenin yüksek frekansta çalışabilirliğini kontrol etmek için, CADENCE ortamında SPECTRE simülatörü kullanılarak AMS SiGe 0.35 μ

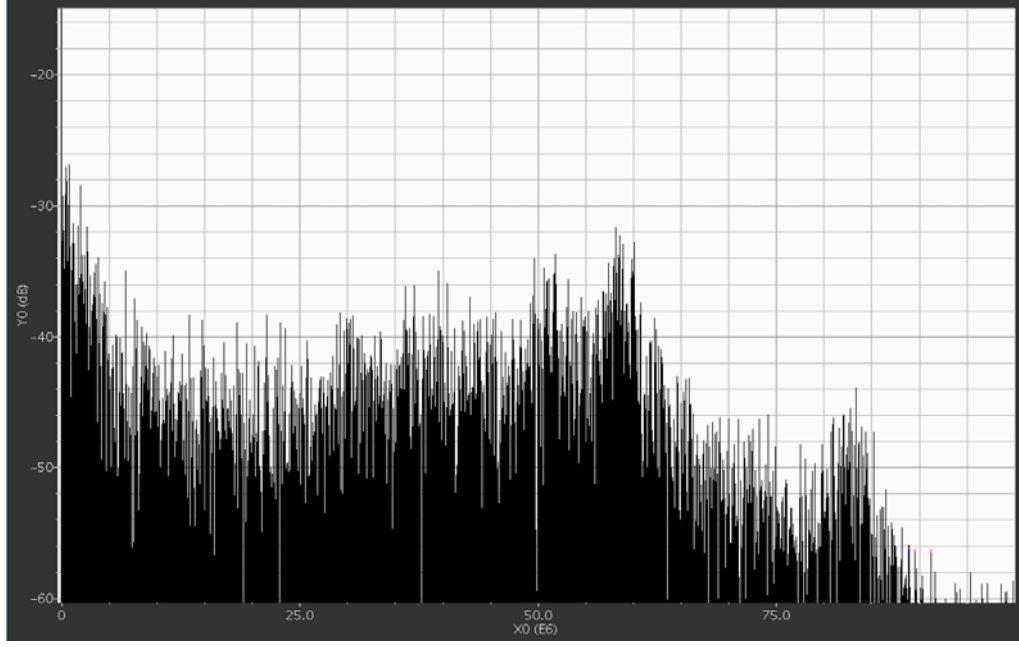
BiCMOS prosesinde benzetim yapılmıştır. Besleme gerilimi 3.3V ve tüm kapasite değerleri 1.5 pF olarak seçilmiş, k değeri 0.2 ve I_0 akımı 100 μ A olarak ayarlanmıştır. Benzetimlerde I_B akımının değerinin önemli olmadığı görülmüş ve $I_0/2$ seçilmiştir. NMOS ve PMOS transistörlerin boyutları sırasıyla 12 μ m/0.7 μ m ve 15 μ m/0.7 μ m seçilmiştir. M_{C2} 'nin boyutu ise 5/0.7 μ m olarak seçilmiştir. Bu değerler kullanılarak, V_C 'nin 2.4V ve 2.8V değerleri arasında, devrenin kaotik işaret ürettiği görülmüştür. $V_C = 2.4V$ için I_X ve I_Y nin faz portresi şekil 4.8'de görülmektedir.

Kaotik osilatör devresinin kalitesini gösteren başka bir unsur da frekans spektrumudur. Frekans spektrumu, ideal gürültünün spektrumuna ne kadar benzerse osilatör de o kadar kaliteli demektir. Fakat devrenin kaçınılmaz olarak her zaman ana bir frekans bileşeni vardır ve bu da frekans spektrumuna tepe olarak yansır. Ayrıca spektrumun sürekliliği çıkıştaki bileşen sıklığını göstermektedir.



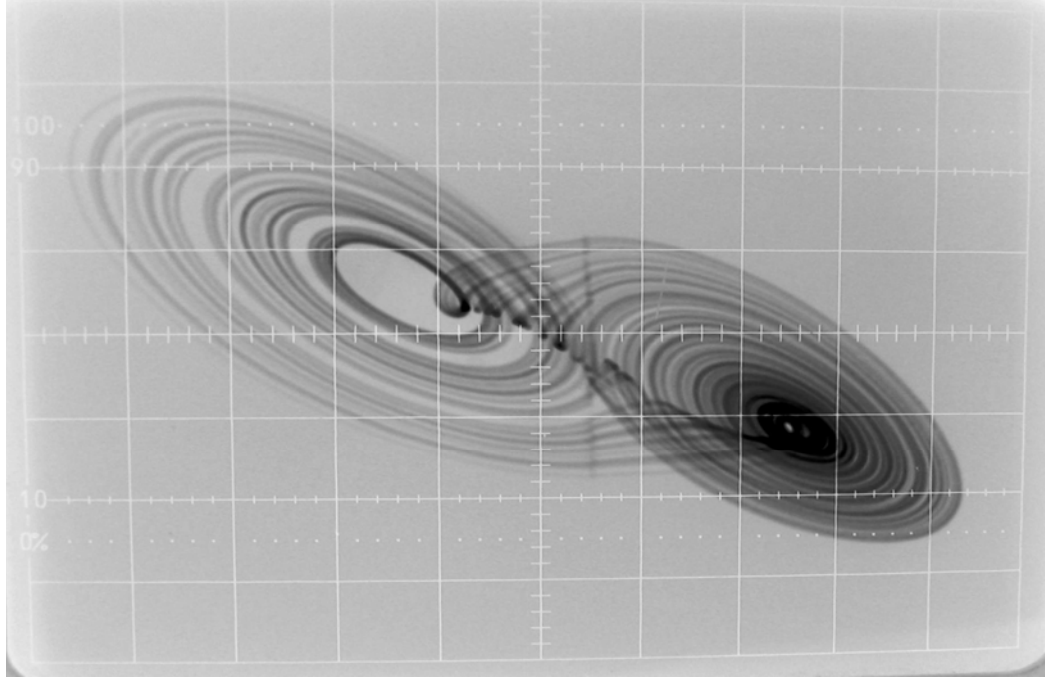
Şekil 4.8: Cadence benzetimi sonucu oluşan I_X - I_Y çekicisi

Tasarlanan kaotik osilatör devresinin frekans spektrumu incelendiğinde 60MHz'de ana frekans bileşeni olduğu ve bant genişliğinin 0-80 MHz arasında olduğu tespit edilmiştir (Şekil 4.9). Bu da devrenin çok yüksek frekanlarda çalışabildiğinin göstergesidir.



Şekil 4.9: Cadence benzetimi sonucu oluşan I_X 'in frekans spektrumu

Devrenin uygulanabilirliğini göstermek açısından, kaotik osilatör, ayrık devre elemanlarıyla gerçekleştirilmiştir. Besleme gerilimi $\pm 5V$ ve kapasite değerleri $10nF$ seçilmiştir. $I_0 = 2mA$ yapılmıştır. MOS transistörler için CD4007 tümdevresi kullanılmıştır. Bu parametrelerle, devrenin ana frekans bileşeninin $f=22\text{ KHz}$ olduğu tespit edilmiştir. I_X ve I_Y akımları kopyalanıp direnç üzerinden akıtılarak gerilime dönüştürülmüş ve şekil 4.10'daki çift sarmallı çekici elde edilmiştir. Görüldüğü gibi şekil 4.4 ve şekil 4.10 arasında uyumlu bir benzerlik vardır.



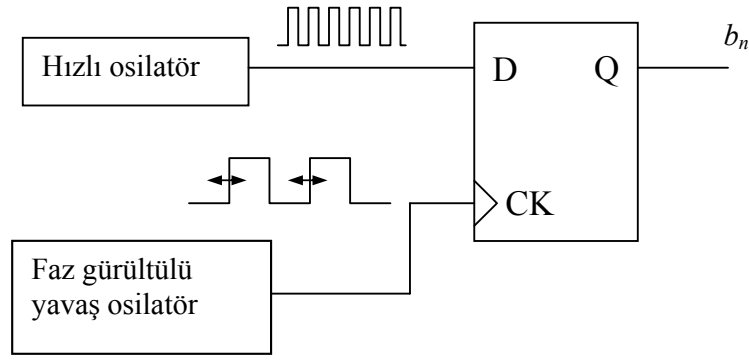
Şekil 4.10: Ayrık olarak gerçekleştirilen devreden elde edilen V_X - V_Y çekicisi

4.2 Kaotik İşaret Girişli ADC Tabanlı RSÜ Yapısı

Düzensiz davranmaları, aperiodyk olmaları, başlangıç koşullarına hassaslık ve gürültü işareti gibi davranmaları nedeniyle kaotik işaretlerin RSÜ'lerde rastgelelik sağlayıcı olarak kullanıldığına değinilmişti. Fakat kaotik işaretlerin aralarında zaman zaman korelasyon oluşmasından dolayı, kaotik işaret girişli RSÜ'ler iki ana bloktan oluşur. Bunlar, kusurlu bir şekilde rastgelelik üreten kaotik osilatör bloğu ve tam rastgelelik sağlayan ikinci bloktur.

4.2.1 Rastgelelik Bloğunun Tasarlanması

Bu çalışmada rastgelelik bloğunu gerçeklemek için literatürde iyi bilinen çift osilatör yapısının [6] matematiksel modeli kullanılmıştır (Şekil 4.11). Önceden belirtildiği gibi, bu yöntemde osilatörlerin faz gürültüsü rastgelelik esas alınmış ve yavaş osilatörün yükselen kenarında hızlı osilatör örneklenmiştir.

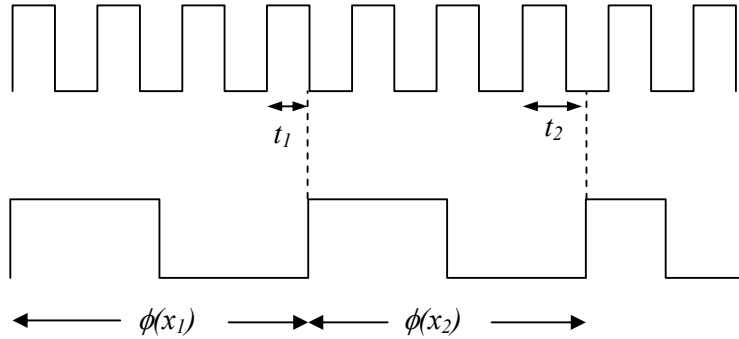


Şekil 4.11: Çift osilatör yapısı

Şekil 4.12’de görüldüğü gibi t_n , örnekleme anlarında iki osilatör arasındaki faz farkını göstermektedir. Matematiksel ifadeyle, $\phi(x_n)$ yavaş osilatörün periyotlarını ve T_f ’de hızlı osilatörün periyodunu belirtmek üzere,

$$t_n = \left(\sum_{j=0}^n \phi(x_j) \right) \bmod T_f \quad (4.12)$$

olduğu görülmektedir.

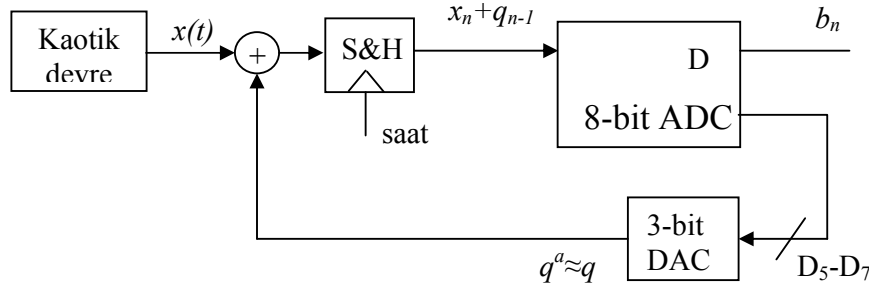


Şekil 4.12: Osilatör çıkışlarının zaman diyagramı

Modüler aritmetik kullanılarak (4.12) ifadesinin (4.13)’e eşit olduğu kolayca gösterilebilir. Örnekleme sonucu oluşan b_n çıkışı da (4.14) ‘te belirtilmiştir.

$$t_n = (t_{n-1} + \phi(x_n)) \bmod T_f \quad (4.13)$$

$$b_n = \begin{cases} 1, & t_n < T_f / 2 \\ 0, & \text{diğer} \end{cases} \quad (4.14)$$



Şekil 4.14: RSÜ Blok Diyagramı

Bu yapıda, adım aralıkları birbirleriyle eşit, 8 bitlik bir ADC ve DAC kullanılmıştır. Dönüştürücülerin bitleri, en çok anlamlıdan en az anlamlıya doğru, $D_0, \dots, D_7$ şeklindedir. Yapılan bilgisayar analizleri sonucu, ADC' nin çıkışından veri okunan bitin anlamı azaldıkça verinin kalitesinin arttığı, geribesleme bloğunun etkisi ortadan kaldırıldığında ise verinin kalitesinin belirgin olarak düştüğü görülmüştür. Kullanılan ADC'nin 8 bit olduğu göz önüne alınırsa, gerçekleştirme esnasında veri okunan bitin anlamı azaldıkça geribesleme yapılacak bit sayısı da azalacağı için, veri 6. bitten alınmıştır. Bu nedenle geribeslemenin çözünürlüğü için en az anlamlı son iki bit kullanılmıştır. ADC'nin bit sayısı arttıkça RSÜ'nün ideal modele daha da yaklaşacağı açıktır. Modelin başarımını test etmek açısından, devreden geribesleme bloğu çıkarılarak devre teste tabi tutulmuştur. Bu şekilde çıkışlar, NIST 800-22 testlerinden başarısız olmuş fakat FIPS 140-1 testleri hâlâ iyi sonuçlar vermeye devam etmiştir.

Devre ayrıık olarak, *successive approximation* yapılı AD7892 ve AD667 DAC tümdevreleriyle gerçekleştirilmiştir. Tümdevrelerin adım ve giriş salınım aralıkları aynıdır. Toplayıcı blok olarak AD844 (Current Feedback Amplifier) tümdevrelerinden oluşan bir blok tasarlanmıştır. Yapılan ölçümler sonucu, ADC'nin örnekleme frekansı arttıkça kaotik osilatörün çıkışından alınan örnekler arasındaki korelasyonun arttığı görülmüştür. bu nedenle kaotik osilatörün çalışma frekansının 22KHz civarında olduğu göz önünde bulundurularak ADC'nin saat frekansı 4KHz seçilmiştir. Bu frekansın altındaki değerler için de aynı sonuçlar geçerli olmuştur.

Tasarlanan RSÜ'nün ürettiği bitler dijital bir kart aracılığıyla bilgisayara aktarılmış ve NIST 800-22 testine tabi tutulmuştur. Bu amaçla 60Mbit veri, her bir blokta 100,000 bit olacak şekilde 600 blok halinde test edilmiştir. Test sonucunda tüm verilerin testi geçtiği görülmüştür. Tablo 4.1'de, sağdaki sütun, testi geçen blok

sayısının teste giren blok sayısına oranını ve soldaki sütun da bu blokların testi geçme başarısına ilişkin p-değerini göstermektedir.

Tablo 4.1: NIST Testi Sonuçları (600 blokluk verinin testi geçme oranı, random-excursions testi hariç, 0.977814'den büyük olmalıdır.)

Statistical Test	p-değerleri	testi geçen blokların oranı
Block-frequency	0.007787	0.9950
linear-complexity	0.245072	0.9767
Runs	0.000115	0.9800
Fft	0.074945	0.9867
Apen	0.000527	0.9850
Serial	0.875539	0.9883
Serial	0.342451	0.9883
cumulative-sums	0.785562	0.9900
cumulative-sums	0.247171	0.9900
longest-run	0.065502	0.9850
Frequency	0.407091	0.9883
Rank	0.345115	0.9983
Universal	0.8055	1.0000
overlapping-templates	0.955835	0.9967
nonperiodic-templates	0.006582	0.9933
random-excursions	0.122325	0.9630
random-excursions-variant	0.010988	0.9818

Testler birkaç defa daha tekrarlanmış ve sonuçların aynı şekilde başarılı olduğu görülmüştür. Kaotik osilatörün benzetim sonuçlarına göre 60MHz'de çalışabileceği düşünülürse RSÜ'nün çıkış hızı bir kaç on Mbit/s olacaktır.

5. SONUÇ

Bu tezin amacı, girişinde yeni sürekli zaman çift sarmallı kaotik osilatörden elde edilen bir işaret olan, ADC tabanlı yeni bir gerçek rastgele sayı üretici (GRSÜ) tasarlamaktır. Tezde kullanılan kaotik osilatör, tümleştirmeye uygun, yüksek hızlarda çalışabilen ve parametreleri kontrol edilebilen türdendir. Tasarlanan RSÜ ise, literatürde iyi bilinen çift osilatör yapısının matematiksel modeliyle uyumlu, ADC tabanlı bir yapıya sahiptir. Gerçeklemede kullanılan ADC ve DAC'ler 8 bitlik olmasına rağmen sadece 3 bit aktif olarak kullanıldığı için, tümdevre yapımında adım aralığı 8 bitlik bir ADC'ninkine eşit olan 3 bitlik bir ADC ve DAC kullanılması tasarımı kolaylaştıracaktır.

RSÜ'nün çıkışından alınan 60Mbit veri, herbirinin uzunluğu 100000 bit olan 600 blok halinde, en geçerli rastgelelik testlerinden biri olan NIST 800-22 testine tabi tutulmuştur. NIST 800-22'yi oluşturan onbeş testten dördünün sonucunda çok küçük p-değerleri oluştuğu gözlenmiştir. Bu sonuç, test edilen verinin o testlere ait özelliklerinin sınır değerlerde olduğunu gösterse de, veri tüm testlerden geçmiştir. Devrenin tümleştirilmesi halinde RSÜ'nün çıkış hızı birkaç on Mbit/s seviyesinde olacaktır.

Literatürde NIST 800-22 testini başarıyla geçen çok az sayıda RSÜ yapısı vardır. Ayrıca tasarlanan RSÜ'nün çıkışı herhangi bir algoritmaya tabi tutulmadığından ve girişinde yüksek frekanslı bir işaret olmasından dolayı, çıkış hızı mevcut yapılardakinden daha yüksektir. Kullanılan osilatörün frekansından daha yüksek frekanslı bir kaotik osilatör tasarlandığında çıkış hızının da çok yüksek olacağı açıktır.

KAYNAKLAR

- [1] **Strogatz S.**, 2001. Nonlinear Dynamics and Chaos, *Westview Press*, Cambridge.
- [2] **Bucci M., Germani L., Luzzi R., Tommasino P., Trifiletti A. and Varanonuovo M.**, 2003. A High-Speed IC Random-Number Source for SmartCard Microcontrollers, *IEEE Trans. Circuits and Systems I*, **50**, 1373-1380.
- [3] **Callegari S., Rovatti R. and Setti G.**, 2005. Embeddable ADC-Based True Random Number Generator for Cryptographic Applications Exploiting Nonlinear Signal Processing and Chaos, *IEEE Trans. on Signal Processing*, **53**, 793-805.
- [4] **Chua L., Wu W., Huang A. and Zhong G.**, 1993. A Universal Circuit for Studying and Generating Chaos-Part I: Routes to Chaos, *IEEE Trans. Circuits and Systems I*, **40**, 732-744.
- [5] **Demirkol Ş., Tavas V., Özoğuz S. and Toker A.**, High Frequency Chaos Oscillators with Applications, *ECCTD 2007*.(Hakem değerlendirmesinde)
- [6] **Jun B. and Kocher P.**, 1999. The Intel Random Number Generator, *White Paper Prepared for Intel Corporation*, Available at: <http://www.cryptography.com/resources/whitepapers/IntelRNG.pdf>.
- [7] **Kocarev L. and Jakimoski G.**, 2003. Pseudorandom Bits Generated by Chaotic Maps, *IEEE Trans. Circuits and Systems I*, **50**, 123-26.
- [8] **Özoğuz S., Tavas V. and Demirkol Ş.**, An Integrated Double-Scroll Chaotic Circuit with its Application to Random Number Generation, *Analog Integrated Circuits and Signal Processing*. (Hakem değerlendirmesinde)
- [9] **Özoğuz S., Elwakil A. and Ergun S.**, 2006. Cross-Coupled Chaotic Oscillators and Application to Random Bit Generation, *IEE Proc.-Circuits Devices Syst.*, **153**, 506-510.
- [10] **Petrie C. and Connelli J.**, 2000. A Noise-Based IC Random number Generator for Applications in Cryptography, *IEEE Trans. Circuits and Systems I*, **47**, 615-621.
- [11] **Stojanovski T. and Kocarev L.**, 2001. Chaos-Based Random Number Generators-Part I: Analysis, *IEEE Trans. Circuits and Systems I*, **48**, 281-288.
- [12] **Yalçın M., Suykens J. and Vandewalle J.**, 2004. True Random Bit Generation from a Double Scroll Attractor, *IEEE Trans. Circuits Syst. I*, **51**, 1395-1404.

- [13] **National Institute of Standard and Technology**, A Statistical Test Suite for Random and Pseudo Random Number Generators for Cryptographic Applications, NIST800-22, 2001. Available at: [*http://csrc.nist.gov/rng/SP800-22b.pdf*](http://csrc.nist.gov/rng/SP800-22b.pdf)

ÖZGEÇMİŞ

Ahmet Şamil Demirkol, 1983 yılında Erzurum’da doğdu. Lise öğrenimini 2001 yılında İstanbul Atatürk Fen Lisesinde, lisans öğrenimini ise 2005 yılında İstanbul Teknik Üniversitesi Elektronik Mühendisliği bölümünde tamamladı. Aynı yıl, İ.T.Ü. Fen Bilimleri Enstitüsü Elektronik Mühendisliği programında yüksek lisans eğitimine başladı. 2005 yılından beri, Devreler ve Sistemler Anabilim dalında, araştırma görevlisi olarak çalışmaktadır.