Stokastik Hesaplamada Hata Oranlarını Azaltmak için Rastgele Bit Karıştırma Yöntemi

Random Bit Shuffling Method for Reducing Error Rates in Stochastic Computing

Serter Yavuz[1], Mustafa Altun[2]

[1,2]Elektronik ve Haberleşme Mühendisliği Bölümü
İstanbul Teknik Üniversitesi
yavuzsert@itu.edu.tr, altunmus@itu.edu.tr

Özet

Stokastik hesaplamanın sahip olduğu etkileyici avantajların yanısıra dezavantajları da vardır. Bunlar temelde süre ve hata oranlarıdır. Bu bildiride, hata oranlarını azaltmak amacıyla, adını rastgele bit karıştırma metodu koyduğumuz yeni bir yöntem öneriyoruz. Bu yöntem, stokastik hesaplamada geleneksel olarak kullanılan rastgele bit üreteçleri ile oluşturulan bit dizileri ile (bu yönteme de rastgele bit atama metodu adını koyduk), lojik kapılar ve bunlardan oluşan devreler kullanılarak hata oranları üzerinden karşılaştırılmıştır. Karşılaştırma yapmak için çeşitli giriş kombinasyonları ile hatayı veren iki ve üç boyutlu grafikler çıkarılmış, bit dizilerinin değişik uzunlukları dikkate alınmış, lojik kapılar için genel hata denklemleri oluşturulmuş, yazılımlar yardımıyla oluşturulan algoritmalar gerçeklenmiştir. Elde edilen veriler ışığında, önerilen rastgele bit karıştırma metodu, temel lojik kapılar için geleneksel metottan daha iyi hata oranları vermiştir.

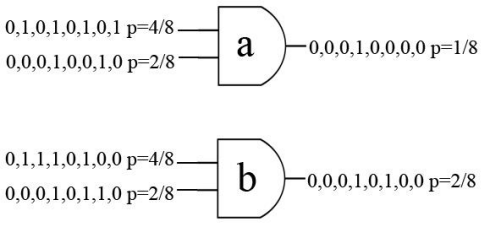
Abstract

Along the great advantages, stochastic computing has disadvantages. These are mainly time and error rates. In this paper, we are introducing a new method that we named it random bit shuffling in order to reduce error rates. This method has been compared with the bit sequence generated by random bit generators (we named it random bit assigning method) traditionally used in stochastic computing by using logic gates and circuits in terms of error rates. In order to compare, 2D and 3D graphs that shows the error were drawn by realising different combinations of two inputs, different length of bit streams were taken into consideration, general error equations for logic gates were obtained and algorithms were realised with the aid of softwares. In the light of obtained data, the random bit shuffling method that is recommended, has given better error rates that traditional method for main logic gates.

1. Giriş

Stokastik hesaplama (SH) [1], rastgele bit dizileriyle zamanda kesintisiz değerleri temsil eden teknikler topluluğudur ve ilk olarak 1953 yılında John von Neumann tarafından hazırlanan bir makale ile takdim edilmiştir [2]. Bu kesintisiz değerler olasılıklar olarak değerlendirilir. Dolayısıyla geçerli aralık [0,1] aralığıdır. Örneğin, 16 bit uzunluklu bir dizi 12 adet 1 içeriyorsa, bu dizi 1'lerin dizideki konumundan bağımsız olarak p = 0.75'i temsil eder. Örneğin (0,0,0,0,1,1,1,1,1,1,1,1,1,1,1,1) ve (0,1,0,1,1,1,1,1,1,0,1,1,1,1,1,0). Farklı bit dizisi uzunluklarıyla da aynı olasılık elde edilebilir.

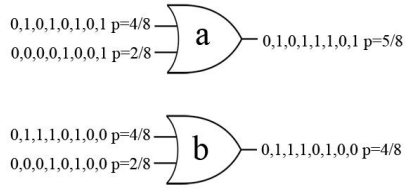
Diziler üzerinde karmaşık hesaplamalar basit devrelerle bit bit işlemlerle yapılabilir. Çarpma ve ölçeklenmiş toplama gibi temel işlemlerin yanısıra SH, bölme ve karekök alma [3], matris işlemleri [4] ve polinom aritmetiğine uygulanabilir [5]. Çarpma işlemi giriş dizisinin uzunluğundan bağımsız olarak tek bir lojik VE (AND) kapısıyla gerçeklenebilmektedir [6]. Klasik yöntemle çarpma yapmak için bit dizisi uzadıkça devre de çok büyük hale gelmektedir. İki adet girişte bulunan bit dizisindeki 1 sayısına bağlı olarak temsil edilen olasılıklar sırasıyla p_1 ve p_2 olsun. Çıkışta elde edilmesi beklenen olasılık $p_1 \; x \; p_2$ 'dir. Ancak girişteki aynı olasılık farklı permütasyonlarla elde edilebileceği için çıkışta beklenen sonuç her zaman elde edilemeyebilir. Bu noktada hata oranı devreye girer.



Şekil 1: VE kapısı ile stokastik işlem örneği.

Şekil 1'de a ile gösterilen VE kapısı ile beklenen sonuç elde edilebilirken b ile gösterilen kapıda sonuç hatalı elde edilmiştir.

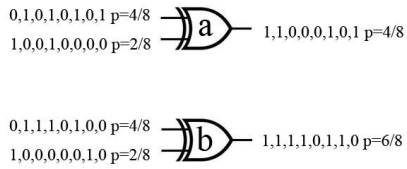
VEYA (OR) kapısı için iki adet girişte bulunan bit dizisindeki 1 sayısına bağlı olarak temsil edilen olasılıklar sırasıyla p_1 ve p_2 olsun. Çıkışta elde edilmesi beklenen olasılık 1 - [(1 - p_1) x (1 - p_2)] 'dir.



Şekil 2: VEYA kapısı ile stokastik işlem örneği.

Şekil 2'de a ile gösterilen VEYA kapısı ile beklenen sonuç elde edilebilirken b ile gösterilen kapıda sonuç hatalı elde edilmiştir.

AYRICALIKLI-VEYA (EXOR) kapısı için iki adet girişte bulunan bit dizisindeki 1 sayısına bağlı olarak temsil edilen olasılıklar sırasıyla p_1 ve p_2 olsun. Çıkışta elde edilmesi beklenen olasılık $p_1 x (1 - p_2) + p_2 x (1 - p_1)$] 'dir.



Şekil 3: AYRICALIKLI-VEYA kapısı ile stokastik işlem örneği.

Şekil 3'te a ile gösterilen AYRICALIKLI-VEYA kapısı ile beklenen sonuç elde edilebilirken b ile gösterilen kapıda sonuç hatalı elde edilmiştir.

Stokastik devrelerde çıkış, dizideki 1'ler sayılarak elde edildiği için deterministiktir. Ancak rastgele bit atama metodunda girişler ise olasılıksaldır. Bu nedenle hata oranları yüksek çıkmaktadır. Hatayı azaltmak için girişin de deterministik olduğu rastgele bit karıştırma methodu araştırılmıştır, detaylı bir analiz yapılmıştır.

Bildirinin akışında ilk olarak karşılaştırılan metotlarla ilgili ayrıntılı bilgiler verilecek, karşılaştırma analizleri yapılacak, örnek bir devre uygulaması gerçekleştirilecek, çalışmanın elde edilen sonuçlarından bahsedilecek ve yapılan çalışmanın gelecekte izleyeceği yol kakkında bilgi verilecektir.

2. Karşılaştırılan Metotlar

Bu bildiride SH'de kullanılan iki metot için karşılaştırmalar yapılmıştır. Bu iki metot arasındaki fark, girişe gelen zamanda sürekli bit dizilerinin temsil ettiği değerler aynı olmasına rağmen dizilerin fiziksel olarak farklılık göstermesidir.

2.1. Rastgele Bit Atama Metodu

SH'da bit dizisi oluşturmak için rastgele bit üreteçleri kullanılır [6]. Bu şekilde dizi oluşturma yöntemine rastgele bit atama metodu (RBAM) adını koyduk. RBAM için girişte istenen olasılık değeri p_1 olsun. Rastgele sayı üreteci de [0-1] arası binom dağılıma sahip değerler üretsin. Üretilen sayı p_1 'den küçük ise girişteki dizide 1 değeri, p_1 'den büyük ise girişteki dizide 0 değeri gözlenir. Bu sebepten girişte istenen değer ile elde edilen arasında farklar oluşmaktadır.

Çizelge 1: $p_1 = 0.5$ için RBAM ile üretilen 8 bitlik örnek bir dizi

Üreteçte Elde Edilen Sayı	Bit Dizisi
0.89967	0
0.45847	1
0.81537	0
0.20818	1
0.71289	0
0.43615	1
0.64214	0
0.78744	0

Çizelge 1'deki örnekte üretilen 8 bitlik dizide üç adet 1, beş adet 0 bulunmaktadır. Bu nedenle girişte elde edilen dizi $p_e = 0.375$ olmaktadır. Çıkışta elde edilecek sonuçta da hatalar oluşacaktır. Devreler karmaşıklaştıkça hata oranları da kabul edilebilir seviyelerin üzerine çıkacaktır.

2.2. Rastgele Bit Karıştırma Metodu

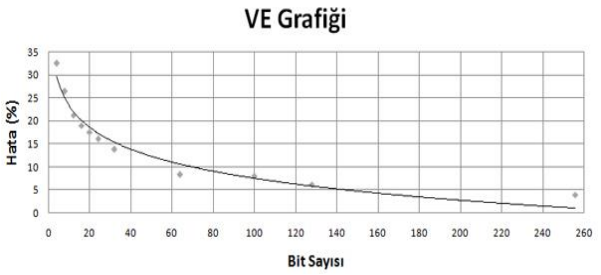
Adını koyup, SH için önerdiğimiz rastgele bit karıştırma metodu (RBKM) için girişte istenen olasılık değeri p_1 olsun. Bu metotta istenen olasılık değeri ve bit dizisi uzunluğuna göre gerek ve yeter miktarda 1 içeren dizi oluşturulur ve Fisher–Yates karıştırma algoritmasının [7] Durstenfeld tarafından bilgisayar kullanımına uyarlanan versiyonu [8] uygulanır. Bu nedenle girişte istenen değerle elde edilen değerin olasılıksal olarak aynı olması garanti edilir. Bu sayede elde edilecek çıkıştaki hata oranlarında bir düşüş elde edilmiş olacaktır.

3. Karşılaştırma Analizleri

İki metot arasında karşılaştırma yapmak için temel lojik kapılar üzerinde çalışılmıştır. Girişlerden ikisi sabit tutularak, her iki girişte [0 - 1] aralığında değiştirilerek ve farklı bit uzunluklarıyla sonuçlar elde edilmiştir.

3.1. p_1=1/2, p_2=1/2 Bit Dizileri için RBKM Kapı Sonuçları

Bu analizde giriş olasılıkları sabit tutularak farklı bit uzunlukları için üç farklı lojik kapıda grafikler elde edilmiştir.

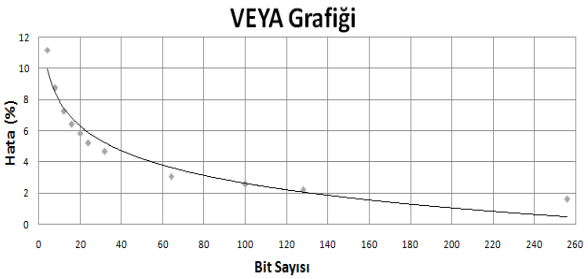


Şekil 4: VE kapısı Hata-Bit Sayısı Grafiği.

VE Kapısı için RBKM ile elde edilen sonuçlar gösteriyor ki, bit dizisinin uzunluğu arttıkça hata oranı düşmektedir. 64 bitten sonrası için hata oranı %10'un altında olduğu için çalışılabilecek bit uzunluğu da bu noktadan sonrasıdır.

p_1, p_2=1/2 ve x=bit dizisinin uzunluğu iken RBKM için VE kapısı genel hata denklemini çıkarttık.

$$\text{Hata}(x) = \frac{2 * \sum_{s=0}^{\frac{x}{4}-1} \binom{x/2}{s} * \binom{x/2}{\frac{x}{2}-s} * \binom{\frac{x}{4}-s}{x/4}}{\binom{x}{x/2}} \quad (1)$$

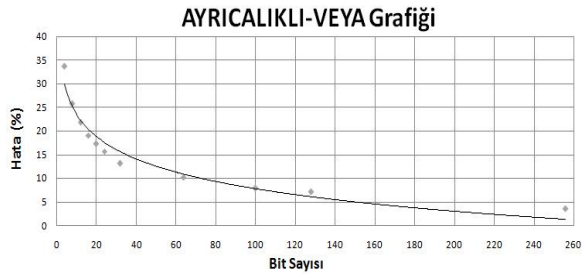


Şekil 5: VEYA kapısı Hata-Bit Sayısı Grafiği.

VEYA Kapısı için RBKM ile elde edilen sonuçlar gösteriyor ki, bit dizisinin uzunluğu arttıkça hata oranı düşmektedir. 4 bitten sonrası için hata oranı %10'un altında olduğu için çalışılabilecek bit uzunluğu da bu noktadan sonrasıdır. VE kapısı ile aradaki fark VEYA için beklenen hatanın hesabından kaynaklanmaktadır.

p_1, p_2=1/2 ve x=bit dizisinin uzunluğu iken RBKM için VEYA kapısı genel hata denklemini çıkarttık.

$$\text{Hata}(x) = \frac{2 * \sum_{s=0}^{\frac{x}{4}-1} \binom{x/2}{s} * \binom{x/2}{\frac{x}{2}-s} * \binom{\frac{x}{4}-s}{3x/4}}{\binom{x}{x/2}} \quad (2)$$



Şekil 6: AYRICALIKLI-VEYA kapısı Hata-Bit Sayısı Grafiği.

AYRICALIKLI-VEYA Kapısı için RBKM ile elde edilen sonuçlar gösteriyor ki, bit dizisinin uzunluğu arttıkça hata oranı düşmektedir. 64 bitten sonrası için hata oranı %10'un altında olduğu için çalışılabilecek bit uzunluğu da bu noktadan sonrasıdır. Hata oranları VE kapısıyla paralellik göstermektedir.

p_1, p_2=1/2 ve x=bit dizisinin uzunluğu iken RBKM için A.VEYA kapısı genel hata denklemini çıkarttık.

$$\text{Hata}(x) = \frac{2 * \sum_{s=0}^{\frac{x}{4}-1} \binom{x/2}{s} * \binom{x/2}{\frac{x}{2}-s} * \binom{\frac{x}{4}-s}{x/4}}{\binom{x}{x/2}} \quad (3)$$

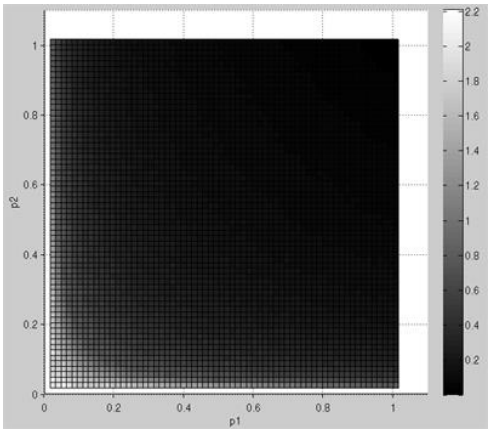
3.2. p_1=1/2, p_2=1/2 Bit Dizileri için RBKM - RBAM Karşılaştırma Sonuçları

Çizelge 2: *Üç lojik kapı için değişen bit uzunları ile elde edilen hata değerleri (%)*

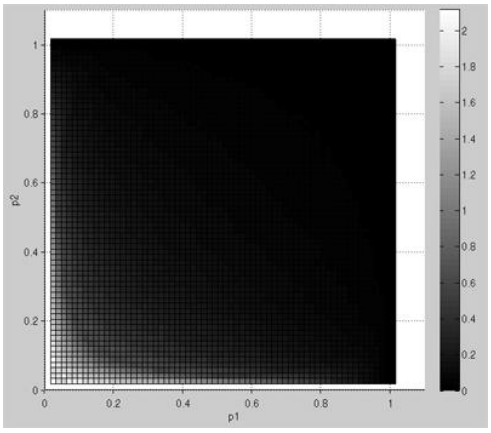
Bit	RBAM			RBKM		
	VE	VEYA	A. VEYA	VE	VEYA	A. VEYA
4	64.3	20.82	37.55	32.55	11.2	33.65
8	46.58	15.38	27.1	26.44	8.77	25.76
12	39.92	12.98	22.52	21.22	7.24	21.85
16	33.38	11.05	19.4	19.04	6.42	19.09
20	30.28	10.22	17.68	17.58	5.85	17.24
24	27.75	9.38	16.35	16.01	5.2	15.66
32	24.28	8.2	14.02	13.82	4.67	13.18
64	17.1	5.68	9.65	8.32	3.05	10.24
100	13.62	4.58	7.88	7.93	2.58	7.82
128	12.12	4.05	7.08	6.21	2.23	7.1
256	8.52	2.8	4.98	3.98	1.6	3.57
512	6.22	2	3.55	2.6	1.34	2.03
1024	4.42	1.4	2.45	1.95	0.35	1.56

Çizelge 2'de görüldüğü üzere karıştırma metodu VE ve VEYA kapıları için daha iyi sonuç vermiştir. Bit uzunluğunun artışı beklendiği üzere hata oranlarını azaltmış, iki method arasındaki hata oranları arasındaki fark ta azalmıştır. Beklenen sonsuzda iki metodun hata oranlarının oturmasıdır. AYRICALIKLI-VEYA kapısı için 4-128 bit arası birbirine oldukça yakın sonuçlar vermektedir, ancak bit sayısı arttıkça karıştırma methodu daha iyi sonuçlar vermektedir.

3.3. 64 Bit Uzunluklu Girişler p_1, p_2 Değişkenken RBKM - RBAM Karşılaştırma Sonuçları

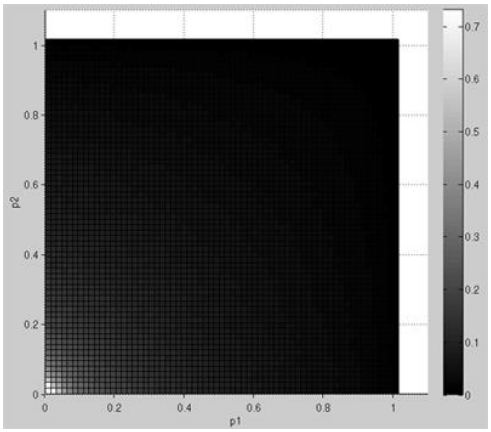


Şekil 7: RBAM VE kapısı p_1-p_2-Hata Grafiği.

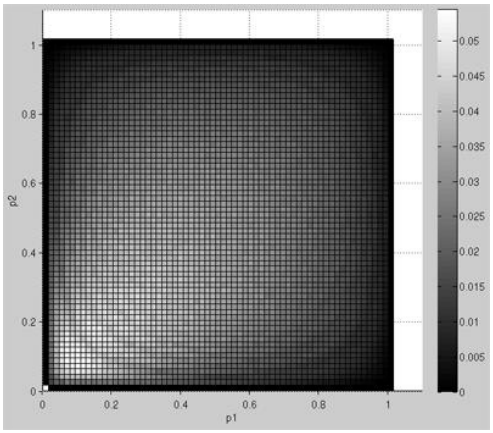


Şekil 8: RBKM VE kapısı p_1-p_2-Hata Grafiği.

VE için p_1 veya p_2 "0" olamaz. Çünkü paydada yer alacak beklenen değer de "0" olacağından hata hesaplanamaz. Grafik köşegene göre simetriktir. VE için ideal çalışma aralığı iki girişinde 0,5 ve üzeri olmasıdır. RBKM için sonuçlar daha iyidir.

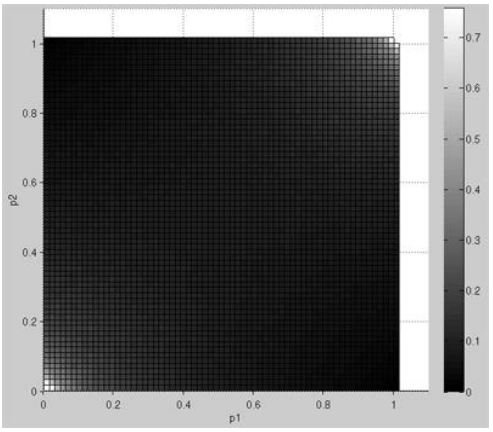


Şekil 9: RBAM VEYA kapısı p_1-p_2-Hata Grafiği.

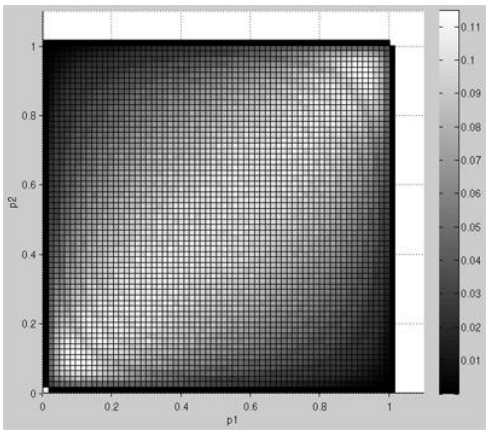


Şekil 10: RBKM VEYA kapısı p_1-p_2-Hata Grafiği.

VEYA için p_1 ve p_2 "0" olamaz. Çünkü paydada yer alacak beklenen değer de "0" olacağından hata hesaplanamaz. Grafik köşegene göre simetriktir. VE'ye göre hata oranları oldukça düşüktür. VEYA için ideal çalışma aralığı iki girişinde 0,3 ve üzeri olmasıdır. RBAM sonuçları RBKM'ye göre oldukça kötüdür.



Şekil 11: RBAM AYRICALIKLI-VEYA kapısı p_1-p_2-Hata Grafiği.



Şekil 12: RBKM AYRICALIKLI- kapısı p_1-p_2-Hata Grafiği.

AYRICALIKLI-VEYA için p_1 ve p_2 "0" veya p_1 ve p_2 "1" olamaz. Çünkü paydada yer alacak beklenen değer de "0" olacağından hata hesaplanamaz. Grafik her iki köşegene göre de simetriktir. VE'ye göre hata oranları düşük, VEYA'ya göre daha yüksektir. AYRICALIKLI-VEYA için ideal çalışma aralığı iki girişinde 0,3 ve üzeri olmasıdır,ancak iki giriş aynı anda köşegen üzerinde değerler almamalıdır. RBAM sonuçları RBKM'ye göre kötüdür.

3.4. p1=[0,1], p2=[0,1] iken RBKM için Genel Hata Denklemleri

Denklemleri elde etmek için Monte Carlo analizinden faydalandık. Bu analiz, problemlerin çözümünü rastgele sayılar kullanarak elde etmeye verilen genel bir isimdir.

n bit sayısı iken VE kapısı için elde ettiğimiz genel hata denklemi ve elde edilen çıkışın gelme olasılığı;

$$\frac{\sum_{i=\begin{cases}0, & p_1+p_2\le 1\\ n*(p_1+p_2-1), & p_1+p_2>1\end{cases}}^{n*\min(p_1,p_2)} \frac{\left|\frac{i}{n}-p_1*p_2\right|}{p_1*p_2} x \binom{(1-p_1)*n}{(n*p_2)-i} x \binom{n*p_1}{i}}{\binom{n}{n*p_2}} \quad (4)$$

$$p_{\left(\frac{i}{n}\right)} = \frac{\binom{(1-p_1)*n}{(n*p_2)-i} x \binom{n*p_1}{i}}{\binom{n}{p_2}} \quad (5)$$

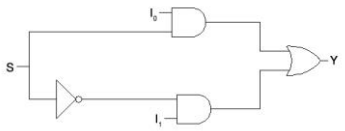
n bit sayısı iken VEYA kapısı için elde ettiğimiz genel hata denklemi ve elde edilen çıkışın gelme olasılığı;

$$\frac{\sum_{i=n*\max(p_1,p_2)}^{n*\min((p_1+p_2),1)} \frac{\left|\frac{i}{n}-(1-[(1-p_1)*(1-p_2)])\right|}{1-[(1-p_1)*(1-p_2)]} x \binom{n*p_1}{i-(n*p_2)} x \binom{(1-p_1)*n}{i-(n*p_1)}}{\binom{n}{n*p_2}} \quad (6)$$

$$p_{\left(\frac{i}{n}\right)} = \frac{\binom{n*p_1}{i-(n*p_2)} x \binom{(1-p_1)*n}{i-(n*p_1)}}{\binom{n}{p_2}} \quad (7)$$

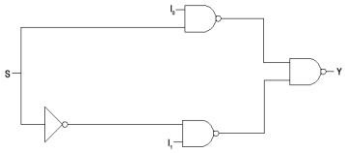
4. Örnek Bir Uygulama

Yöntemler arasında karşılaştırmanın ikinci aşamasında lojik kapı uygulamalarından çoklayıcı kullanılmıştır. Çoklayıcı bir veya daha fazla giriş hattından ikili bilgiyi seçen ve bunu tek çıkış hattına bağlayan bir kombinezonsal devredir [9]. Stokastik hesaplamada ölçekli toplayıcı olarak kullanılmaktadır.



Şekil 16: Çoklayıcı devresi birinci tasarım [9].

İki metot arasında karşılaştırma için iki farklı devre için 64 ve 256 bitlik diziler seçilmiştir. Girişlerin yer değiştirmesi sonucu değiştirmemektedir. Tüm karşılaştırmalarda metodun algoritmasına göre $p_S = 1/2$ alınmıştır.



Şekil 17: Çoklayıcı devresi ikinci tasarım.

Çizelge 3: 64 bitlik giriş dizileri için hata oranları

	Birinci Tasarım		İkinci Tasarım	
Girişler	RBKM Hatalar	RBAM Hatalar	RBKM Hatalar	RBAM Hatalar
$p_1 = 0.25, p_0 = 0.25$	% 13.73	% 18.48	% 6.67	% 19.28
$p_1 = 0.25, p_0 = 0.5$	% 12.08	% 16.09	% 9.09	% 15.65
$p_1 = 0.25, p_0 = 0.75$	% 10.99	% 14.34	% 10.34	% 13.97
$p_1 = 0.5, p_0 = 0.5$	% 15.06	% 16.4	% 14.29	% 16.65
$p_1 = 0.5, p_0 = 0.75$	% 17.45	% 18.42	% 17.65	% 17.9
$p_1 = 0.75, p_0 = 0.75$	% 23.04	% 23.65	% 23.08	% 23.04

RBKM'nin, RBAM'dan daha iyi sonuç verdiğini gözlemledik. İki farklı tasarım arasında RBAM açısından büyük bir fark yoktur, ancak RBKM için ikinci tasarım daha iyi sonuçlar vermiştir.

5. Sonuçlar

Bu çalışmanın sonucunda temel lojik kapılar için rastgele bit karıştırma yönteminin, rastgele bit atama yönteminden hata bazında daha iyi sonuçlar verdiği gözlemlenmiştir. Gelecekte bildiride adı geçen iki yöntem daha fazla lojik kapı içeren aritmetik elemanlar ve hafıza elemanları uygulamaları ile karşılaştırılacaktır. Olası uygulama alanları arasında sinir ağları [10] ve kontrol devreleri [11] de yer almaktadır.

6. Kaynaklar

[1] Gaines, B.R., Stochastic computing, *Proc. AFIPS Spring Joint Computer Conf.*, 1967, 149-156.

[2] von Neumann, J., "Probabilistic logics and the synthesis of reliable organisms from unreliable components", *The Collected Works of John von Neumann*, Macmillan, 1963.

[3] Toral, S.L., Quero, J.M., ve Franquelo, L.G., "Stochastic pulse coded arithmetic", *Proc. ISCAS*, 1, 599-602, 2000.

[4] Mars, P., ve McLean, H.R., "High-speed matrix inversion by stochastic computer", *Electronics Letters*, 12, 457-459, 1976.

[5] Qian, W., Li, X., Riedel, M.D., Bazargan, K., ve Lilja, D.J., "An architecture for fault-tolerant computation with stochastic logic", *IEEE Trans. Computers*, 2, 93-105, 2011.

[6] Alaghi, A., Hayes, J. P., "Survey of Stochastic Computing", *ACM Transactions on Embedded Computing Systems*, 12 (2s), 1-16, 2013.

[7] Fisher, Ronald A., Yates, Frank., *Statistical tables for biological, agricultural and medical research (3. baskı)*, Oliver & Boyd, Londra, (1948).

[8] Durstenfeld, R., "Algorithm 235: Random permutation", Communications of the ACM, 7., 420, 1964.

[9] Mano, M., *Sayısal Tasarım*, Literatür Yayıncılık, İstanbul, 2003.

[10] Brown, B.D., ve Card, H.C., "Stochastic neural computation I: computational elements", *IEEE Trans. Computers*, 50, 891-905, 2001.

[11] Marin, S.L.T., Reboul, J.M.Q., ve Franquelo, L.G., "Digital stochastic realization of complex analog controllers", *IEEE Trans. Industrial Electronics*, 49, 1101-1109, 2002.