5th International Conference on Contemporary Academic Research

May 30-31, 2025: Konya, Turkey

© 2025 Published by All Sciences Academy

https://as-proceeding.com/index.php/iccar/home

Memristör Tabanlı Bellek-İçi Hesaplama Yöntemlerinde Enerji Tüketimi Analizi: SIXOR ve MAGIC Kıyaslaması

Elif Öztürk [1*], Zehra Gülru Çam Taşkıran [2]

[1] *Fen Bilimleri Enstitü, Elektronik ve Haberleşme Mühendisliği Bölümü, Yıldız Teknik Üniversitesi, Türkiye (ORCID: https://orcid.org/0009-0001-4006-3458)*

[2] *Elektrik-Elektronik Fakültesi, Elektronik ve Haberleşme Mühendisliği Bölümü, Yıldız Teknik Üniversite, Türkiye (ORCID: https://orcid.org/0000-0002-7996-7948)*

[*]*birelifozturk@gmail.com*

Özet – Memristör tabanlı bellek-içi hesaplama (IMC) sistemleri, işlemci-veri yolu darboğazlarını azaltarak enerji verimliliği açısından önemli bir potansiyel sunmaktadır. Bu sistemlerde mantıksal işlemlerin enerji tüketimi, mimari tercihlerin performansını doğrudan etkiler. Bu çalışma, IMC sistemlerinde XOR ve NOR mantık işlemlerini gerçekleştirmek için tasarlanmış SIXOR (Single-Cycle In-Memristor XOR) ve MAGIC (Memristor-Aided loGIC) yöntemlerinin enerji tüketimlerini karşılaştırmaktadır. MAGIC, NOR tabanlı evrensel bir yapıyla çapraz çubuk dizileri içinde çalışırken, SIXOR, XOR işlemini tek çevrimde tamamlayan yenilikçi bir yaklaşım sunar. Her iki yöntem, dört farklı giriş kombinasyonu (A=0, B=0; A=0, B=1; A=1, B=0; A=1, B=1) için Orcad PSpice yazılımıyla simüle edilmiştir. Simülasyon sonuçları, SIXOR'un ortalama işlem enerjisinin 0.1355815 pJ, toplam işlem enerjisinin 0.542326 pJ; MAGIC'in ise ortalama işlem enerjisinin 0.023075 pJ, toplam işlem enerjisinin 0.0923 pJ olduğunu göstermiştir. Ancak, SIXOR'un ortalama initialization enerjisi 1.9781 pJ ile MAGIC'in 11.1152 pJ'lik değerine kıyasla 5.6 kat daha düşüktür ve toplam enerji tüketiminde (8.454726 pJ) MAGIC'e (44.5531 pJ) göre 5.3 kat daha verimlidir. SIXOR, asimetrik girişlerde daha yüksek işlem enerjisi tüketse de, tek çevrimde işlem tamamlama yeteneğiyle hız ve tasarım sadeliği açısından avantaj sağlar. Buna karşılık, MAGIC, düşük işlem enerjisiyle öne çıksa da, çok çevrimli yapısı işlem süresini uzatır. Ayrıca, SIXOR'un farklı gerilim değerleri (±1.2 V) gerektirmesi çevresel devre tasarımında karmaşıklık yaratırken, MAGIC'in tek gerilim (1.3 V) kullanması bu açıdan daha sade bir yapı sunar. Yine de SIXOR'un hız ve enerji avantajları, özellikle şifreleme ve nöromorfik hesaplama gibi XOR tabanlı uygulamalarda belirgin bir üstünlük sağlar. Bu çalışma, SIXOR'un enerji verimliliği, hız ve uygulama odaklı yaklaşımıyla XOR işlemleri için daha avantajlı olduğunu ortaya koymaktadır.

Anahtar Kelimeler –Memristör, bellek-içi hesaplama, SIXOR, MAGIC, enerji tüketimi, XOR, NOR

I. GİRİŞ

Memristör tabanlı bellek-içi hesaplama (IMC) sistemleri, geleneksel Von-Neumann mimarisinin işlemci ve bellek arasındaki veri transferi darboğazını aşarak yüksek enerji verimliliği ve hız sunmaktadır [1]. Bu sistemler, özellikle enerji tüketimini azaltma ve hesaplama hızını artırma potansiyeli ile dikkat çekmektedir; çünkü Von-Neumann mimarisinde, işlemci ve bellek arasındaki sürekli veri transferi hem enerji tüketimini artırmakta hem de işlem gecikmelerine yol açmaktadır [2]. Bu bağlamda, SIXOR (Single-Cycle In-Memristor XOR) yöntemi, XOR mantık işlemini tek çevrimde gerçekleştirmek için tasarlanmış yenilikçi bir yaklaşım olarak öne çıkmaktadır. SIXOR, iki giriş memristörü (A ve B), bir çıkış

memristörü (F) ve iki yardımcı memristör (C ve D) içerir; devre, $+V_x$ ve $-V_x$ gerilim kaynaklarıyla çalışır. Öte yandan, MAGIC (Memristor-Aided loGIC) yöntemi, NOR tabanlı evrensel bir yapıya sahip olup, mantık işlemlerini çapraz çubuk dizileri içinde gerçekleştirmektedir [3]. Her iki yöntem, memristörlerin direnç durumlarını kullanarak çalışır: yüksek direnç durumu (HRS) mantıksal 0'ı, düşük direnç durumu (LRS) ise mantıksal 1'i temsil eder. Memristörlerin bu özelliği, enerji verimli hesaplama sistemleri için önemli bir temel oluşturmakta ve özellikle nöromorfik hesaplama gibi yenilikçi alanlarda kullanım potansiyeli sunmaktadır [4]. Bu çalışma, SIXOR ve MAGIC yöntemlerinin enerji tüketimlerini dört farklı giriş kombinasyonu (Case 1: A=0, B=0; Case 2: A=0, B=1; Case 3: A=1, B=0; Case 4: A=1, B=1) üzerinden analiz ederek kıyaslamayı amaçlamaktadır. Simülasyonlar, Orcad PSpice yazılımı kullanılarak gerçekleştirilmiş ve enerji tüketim değerleri 1 ns (initialization) ve 2 ns (computation) zaman noktalarında ölçülmüştür. Çalışmanın temel hedefi, SIXOR'un hız ve tasarım sadeliği açısından MAGIC'e kıyasla sunduğu avantajları ortaya koymaktır; çünkü SIXOR'un tek çevrimde işlem tamamlama yeteneği, özellikle zaman kritik uygulamalarda önemli bir fark yaratabilir [5]. Ayrıca, her iki yöntemin enerji tüketim profilleri detaylı bir şekilde değerlendirilerek, bu yöntemlerin pratik uygulamalardaki potansiyel avantajları ve sınırlamaları tartışılmıştır. Bu bağlamda, memristör tabanlı sistemlerin büyük ölçekli uygulamalarda karşılaşabileceği zorluklar, örneğin döngüden döngüye değişkenlikler ve kaçak akım sorunları, da dikkate alınmıştır [6].

II. MATERYAL VE YÖNTEM

Bu çalışma, memristör tabanlı bellek-içi hesaplama (IMC) sistemlerinde XOR ve NOR mantık işlemlerini gerçekleştirmek için tasarlanmış SIXOR (Single-Cycle In-Memristor XOR) ve MAGIC (Memristor-Aided loGIC) yöntemlerinin enerji tüketimlerini karşılaştırmayı amaçlamaktadır. Her iki yöntem, Orcad PSpice simülasyon yazılımı kullanılarak dört farklı giriş kombinasyonu (Case 1: A=0, B=0; Case 2: A=0, B=1; Case 3: A=1, B=0; Case 4: A=1, B=1) üzerinden analiz edilmiştir. Simülasyonlarda, memristörlerin yüksek direnç durumu (HRS) mantıksal "0"ı, düşük direnç durumu (LRS) ise mantıksal "1"i temsil etmiştir. Enerji tüketim ölçümleri, 1 nanosaniye (ns) zaman noktasında başlangıç (initialization) enerjisi olarak alınmış ve işlem enerjisi, 2 ns ile 1 ns arasındaki fark (2 ns - 1 ns) üzerinden hesaplanmıştır. Initialization enerjisi, memristörlerin başlangıç durumlarını ayarlamak için harcanan enerjiyi ifade ederken, işlem enerjisi, hesaplama sırasında tüketilen enerjiyi temsil eder. Tüm ölçümler piko joule (pJ) birimiyle ifade edilmiştir ($1\ pJ = 10^{-12}\ J$). Simülasyonlar sırasında, memristörlerin eşik gerilimi (Vthon) 0.7 V, uygulanan gerilimler ise +Vx ve -Vx olarak sırasıyla 1.2 V ve -1.2 V şeklinde belirlenmiştir. Bu parametreler, memristör tabanlı devrelerin tipik çalışma koşullarını yansıtmak için literatürden seçilmiştir [4]. Her bir giriş kombinasyonu için enerji tüketim değerleri, devrenin dinamik davranışını ve enerji verimliliğini değerlendirmek amacıyla detaylı bir şekilde analiz edilmiştir.

A. SIXOR Yöntemi

SIXOR (Single-Cycle In-Memristor XOR) yöntemi, memristör tabanlı bellek-içi hesaplama (IMC) sistemlerinde XOR mantık işlemini tek bir çevrimde gerçekleştirmek için tasarlanmış yenilikçi bir devre yaklaşımıdır. Devre, iki giriş memristörü (A ve B), bir çıkış memristörü (F) ve iki yardımcı memristör (C ve D) içerir. SIXOR'un temel çalışma prensibi, memristörlerin direnç durumlarını kullanarak mantıksal işlemleri gerçekleştirmektir; burada yüksek direnç durumu (HRS) mantıksal "0"ı, düşük direnç durumu (LRS) ise mantıksal "1"i temsil eder. Devre, A memristörüne +Vx (örneğin, 1.2 V), B memristörüne ise -Vx (örneğin, -1.2 V) gerilimlerini uygularken, çıkış memristörü F, ortak düğüm gerilimi (Vn) üzerinden durumu belirler. İşlem öncesinde F, C ve D memristörleri HRS ("0") olarak başlatılır. Eğer Vn, F'nin eşik gerilimini (Vthon, örneğin 0.7 V) aşarsa, F memristörü HRS'den LRS'ye geçiş yapar ve mantıksal "1" üretir; aksi halde HRS'de kalarak mantıksal "0" verir. Yardımcı memristörler C ve D, asimetrik girişlerde (örneğin, A=0, B=1 veya A=1, B=0) ve her iki girişin LRS olduğu durumlarda (A=1, B=1) devrenin stabilitesini sağlamak için kritik bir rol oynar. Örneğin, C memristörü asimetrik durumlarda Vn'yi düzenlerken, D memristörü, her iki girişin LRS olduğu durumda istenmeyen durum kaymalarını önler. Bu

tasarım, XOR doğruluk tablosuna uygun çıkışlar üretir: A ≠ B olduğunda (örneğin, A=0, B=1 veya A=1, B=0) çıkış "1", A = B olduğunda (örneğin, A=0, B=0 veya A=1, B=1) ise çıkış "0" olur. SIXOR'un en önemli avantajı, bu işlemi tek çevrimde tamamlayabilmesi ve böylece hesaplama hızını artırırken enerji tüketimini optimize etmesidir.

SIXOR'un çalışma prensipleri, dört farklı giriş kombinasyonu (Case 1: A=0, B=0; Case 2: A=0, B=1; Case 3: A=1, B=0; Case 4: A=1, B=1) üzerinden detaylı bir şekilde analiz edilmiştir. Case 1 (A=0, B=0) senaryosunda, her iki giriş memristörü (A ve B) HRS'de olduğundan, ortak düğüm gerilimi (Vn) sanal toprak seviyesinde (yaklaşık 0 V) kalır ve F'nin eşik gerilimini (Vthon) aşamaz. Bu nedenle, F memristörü HRS'de (mantıksal "0") kalır, XOR doğruluk tablosuna uygun bir çıkış üretir. Case 2 (A=0, B=1) durumunda, A memristörü HRS'de, B memristörü LRS'de olduğundan, Vn başlangıçta -Vx'e (örneğin, -1.2 V) yaklaşır. Ancak, yardımcı memristör C'nin durumu HRS'den LRS'ye değişerek Vn'yi artırır (yaklaşık +Vx, örneğin 1.2 V), böylece F'nin eşik gerilimini aşması sağlanır ve F, HRS'den LRS'ye (mantıksal "1") geçiş yapar. Case 3 (A=1, B=0) senaryosu, Case 2'nin simetrik tersidir; A memristörü LRS'de, B memristörü HRS'de olduğundan, Vn bu kez +Vx'e (örneğin, 1.2 V) yaklaşır ve F'nin eşik gerilimini aşarak F'yi LRS'ye (mantıksal "1") geçirir. Son olarak, Case 4 (A=1, B=1) durumunda, her iki giriş memristörü LRS'de olduğundan, Vn sanal toprak seviyesinde (yaklaşık 0 V) kalır; ancak, yardımcı memristör D'nin etkisiyle Vn, F'nin eşik gerilimini aşamaz ve F, HRS'de (mantıksal "0") kalır. Bu dört senaryo, SIXOR'un XOR mantık kapısının doğruluk tablosuna uygun çıkışlar ürettiğini ve tek çevrimde stabil bir şekilde çalıştığını doğrulamaktadır. SIXOR'un sade tasarımı, çapraz çubuk dizisi kullanılmadan çalışmasını sağlar ve bu da potansiyel kaçak akım (sneak path) sorunlarını azaltır.

Tablo 1: SIXOR Yönteminin Case Bazında Enerji Tüketim Değerleri

Case	**İnitialize Enerjisi (pJ)**	**İşlem Enerjisi (pJ)**
Case 1	0.0675	0.066026
Case 2	1.9739	0.151
Case 3	1.9732	0.255
Case 4	3.8978	0.0703
Toplam	7.9124	0.542326
Ortalama	1.9781	0.1355815

SIXOR'un enerji tüketim performansı, her bir Case için başlangıç (initialization) ve işlem enerjileri üzerinden değerlendirilmiştir. Tablo 1, dört farklı giriş kombinasyonu için 1 ns'de ölçülen initialization enerjisini ve işlem enerjisini (2 ns - 1 ns farkı) özetlemektedir. Case 1 (A=0, B=0) durumunda, initialization enerjisi 0.0675 pJ, işlem enerjisi ise 0.066026 pJ'dir; bu düşük değerler, her iki girişin HRS'de olması ve F memristörünün durumunda değişiklik olmaması nedeniyle enerji tüketiminin minimal olduğunu gösterir. Case 2 (A=0, B=1) senaryosunda, initialization enerjisi 1.9739 pJ, işlem enerjisi 0.151 pJ'dir; F memristörünün HRS'den LRS'ye geçişi enerji tüketimini artırır. Case 3 (A=1, B=0) durumunda, initialization enerjisi 1.9732 pJ, işlem enerjisi 0.255 pJ'dir; bu senaryoda F memristörünün geçişi, işlem enerjisinde en yüksek değeri (0.255 pJ) ortaya çıkarır. Son olarak, Case 4 (A=1, B=1) durumunda, initialization enerjisi 3.8978 pJ ile en yüksek seviyededir. Böylelikle, SIXOR yönteminde 4 case için harcanmış olan toplam İnitialize enerjisi 7.9124 pJ olmuştur. Ancak işlem enerjisi 0.0703 pJ ile düşüktür, çünkü F memristörü HRS'de kalır. Bu sonuçlar, SIXOR'un enerji tüketiminin giriş kombinasyonlarına bağlı olarak değiştiğini, özellikle asimetrik girişlerde (Case 2 ve Case 3) F memristörünün durum değişiminin işlem enerjisini artırdığını göstermektedir. Ortalama initialization

enerjisi 1.9781 pJ, ortalama işlem enerjisi ise 0.1355815 pJ'dir; bu değerler, yöntemin enerji verimliliği açısından dengeli bir performans sunduğunu doğrulamaktadır.

B. Magic Yöntemi

MAGIC (Memristor-Aided loGIC) yöntemi, NOR tabanlı evrensel bir yapıya sahiptir ve çapraz çubuk dizileri içinde çalışır. MAGIC, memristörlerin direnç durumlarını kullanarak mantık işlemlerini gerçekleştirir; burada da HRS mantıksal "0"ı, LRS ise mantıksal "1"i temsil eder. Devre, A ve B memristörlerine VSET gerilimi (örneğin, belirli bir eşik gerilimi) uygulanarak çalışır; çıkış memristörü F, yalnızca A ve B'nin her ikisi de HRS'deyken (mantıksal "0") LRS'ye geçiş yapar ve mantıksal "1" üretir; diğer tüm durumlarda (A veya B'den en az biri LRS ise) F, HRS'de kalarak mantıksal "0" verir. Bu NOR tabanlı yapı, diğer mantık kapılarının (örneğin, AND, OR, XOR) türetilmesine olanak tanır, ancak bu işlem birden fazla çevrim gerektirir. Örneğin, XOR işlemini gerçekleştirmek için MAGIC, birden fazla NOR kapısı kombinasyonu (örneğin, A XOR B = (A NOR B) NOR (A NOR A) NOR (B NOR B)) kullanır ve bu da işlem süresini uzatır. MAGIC'in çapraz çubuk dizisi mimarisi, yüksek yoğunluklu hesaplama imkanı sunarak büyük ölçekli uygulamalarda paralel işlem kapasitesi sağlar; ancak, bu yapı kaçak akım sorunlarına yol açabilir ve birden fazla çevrim gerektirmesi nedeniyle enerji verimliliğini etkileyebilir.

MAGIC'in çalışma prensipleri de dört farklı giriş kombinasyonu üzerinden analiz edilmiştir. Case 1 (A=0, B=0) senaryosunda, her iki giriş memristörü HRS'de olduğundan, NOR mantık kapısının doğruluk tablosuna uygun olarak (A NOR B = NOT (A OR B)), F memristörü HRS'den LRS'ye geçiş yapar ve mantıksal "1" üretir. Case 2 (A=0, B=1) durumunda, A memristörü HRS'de, B memristörü LRS'de olduğundan, NOR işlemi sonucunda F memristörü HRS'de kalarak mantıksal "0" üretir. Case 3 (A=1, B=0) senaryosu, Case 2 ile simetrik olup aynı sonucu verir; F, HRS'de kalarak mantıksal "0" üretir. Son olarak, Case 4 (A=1, B=1) durumunda, her iki giriş memristörü LRS'de olduğundan, F memristörü HRS'de kalarak mantıksal "0" üretir. Bu sonuçlar, MAGIC'in NOR mantık kapısının doğruluk tablosuna uygun çıkışlar ürettiğini göstermektedir. Ancak, MAGIC'in çok çevrimli yapısı, özellikle XOR gibi karmaşık işlemler için işlem süresini artırır ve enerji tüketimini olumsuz etkileyebilir. Ayrıca, çapraz çubuk dizisi mimarisi, büyük ölçekli uygulamalarda kaçak akım sorunlarına yol açabilir ve memristörlerin döngüden döngüye değişkenlikleri performansı etkileyebilir.

Tablo 2: MAGIC Yönteminin Case Bazında Enerji Tüketim Değerleri

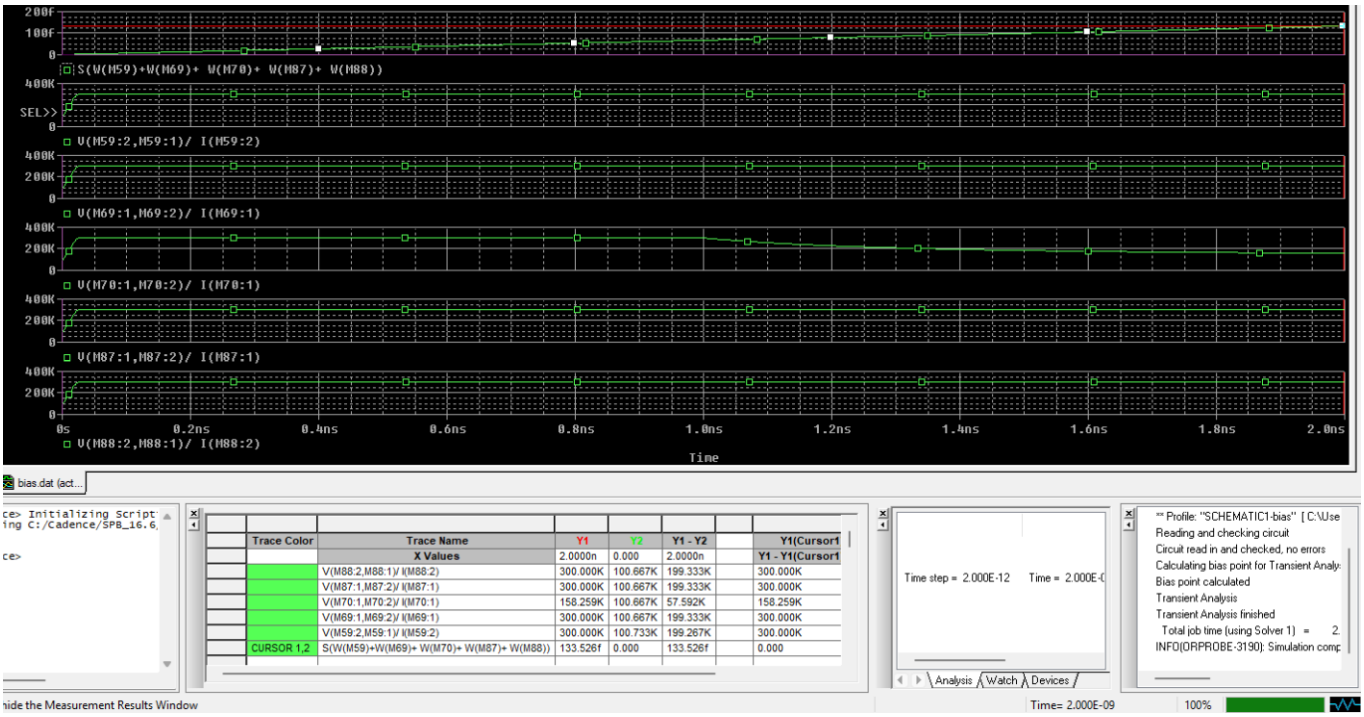
Case	İnitialize Enerjisi (pJ)	İşlem Enerjisi (pJ)
Case 1	9.3208	0.0113
Case 2	11.120	0.031
Case 3	11.120	0.032
Case 4	12.900	0.018
Toplam	44.4608	0.0923
Ortalama	11.1152	0.023075

MAGIC'in enerji tüketim performansı, her bir Case için başlangıç (initialization) ve işlem enerjileri üzerinden değerlendirilmiştir. Tablo 2, dört farklı giriş kombinasyonu için 1 ns'de ölçülen initialization enerjisini ve işlem enerjisini (2 ns - 1 ns farkı) özetlemektedir. Case 1 (A=0, B=0) durumunda, initialization enerjisi 9.3208 pJ, işlem enerjisi ise 0.0113 pJ'dir; bu düşük işlem enerjisi, F memristörünün HRS'den LRS'ye geçişinin minimal enerjiyle gerçekleştiğini gösterir. Case 2 (A=0, B=1) senaryosunda, initialization enerjisi 11.120 pJ, işlem enerjisi 0.031 pJ'dir; F memristörünün HRS'de kalması enerji

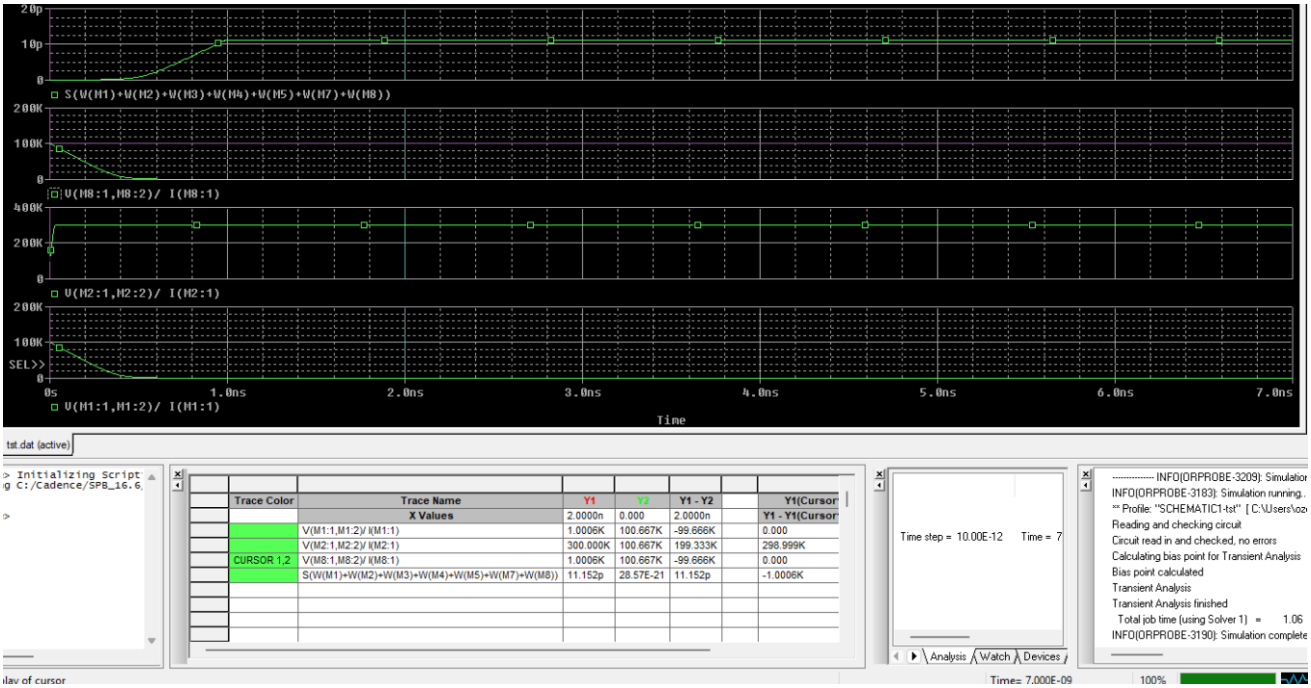
tüketimini düşük tutar, ancak initialization enerjisi diğer senaryolara kıyasla yüksektir. Case 3 (A=1, B=0) durumunda, initialization enerjisi yine 11.120 pJ'dir, işlem enerjisi ise 0.032 pJ ile en yüksek değerini alır; bu artış, asimetrik girişlerin devrenin dinamik davranışı üzerindeki etkisini yansıtır. Son olarak, Case 4 (A=1, B=1) durumunda, initialization enerjisi 12.900 pJ ile en yüksek seviyededir. Böylelikle, MAGIC yönteminde 4 case için harcanmış olan toplam İnitialize enerjisi 44.4608 pJ olmuştur. Ancak işlem enerjisi ise 0.018 pJ'dir; F memristörünün HRS'de kalması işlem enerjisini düşük tutar, ancak her iki girişin LRS'de olması initialization enerjisini artırır. Bu sonuçlar, MAGIC'in enerji tüketiminin giriş kombinasyonlarına bağlı olarak değiştiğini, özellikle her iki girişin LRS olduğu durumda (Case 4) initialization enerjisinin belirgin şekilde yüksek olduğunu göstermektedir. Ortalama initialization enerjisi 11.1152 pJ, ortalama işlem enerjisi ise 0.023075 pJ'dir; bu değerler, MAGIC'in düşük işlem enerjisi sunmasına rağmen, yüksek initialization enerjisi nedeniyle toplam enerji verimliliğinin sınırlı kaldığını ortaya koymaktadır.

C. Simülasyon ve Enerji Tüketim Analizi

Her iki yöntem, dört farklı giriş kombinasyonu için simüle edilmiştir: Case 1 (A=0, B=0), Case 2 (A=0, B=1), Case 3 (A=1, B=0) ve Case 4 (A=1, B=1). SIXOR'un işlem enerjisi değerleri şu şekildedir: Case 1 için 0.066026 pJ, Case 2 için 0.151 pJ, Case 3 için 0.255 pJ ve Case 4 için 0.0703 pJ. Toplam işlem enerjisi 0.542326 pJ, ortalama işlem enerjisi ise 0.1355815 pJ'dir. MAGIC yöntemi için ise işlem enerjisi değerleri Case 1 için 0.0113 pJ, Case 2 için 0.031 pJ, Case 3 için 0.032 pJ ve Case 4 için 0.018 pJ'dir. Toplam işlem enerjisi 0.0923 pJ, ortalama işlem enerjisi ise 0.023075 pJ'dir. Initialization (1 ns) aşamasındaki enerji tüketimleri de analiz edilmiştir; SIXOR'un ortalama initialization enerjisi 1.9781 pJ iken, MAGIC'in ortalama initialization enerjisi 11.1152 pJ'dir. Şekil 1, SIXOR'un Case 1 (A=0, B=0) için simülasyon sonucunu göstermektedir; bu senaryoda F memristörü HRS'de kalarak XOR doğruluk tablosuna uygun bir çıkış üretmiştir. Şekil 2 ise MAGIC'in Case 3 (A=1, B=0) için simülasyon sonucunu göstermektedir; bu senaryoda F memristörü HRS'de kalarak NOR doğruluk tablosuna uygun bir çıkış üretmiştir.



Şekil 1: SIXOR Yöntemi ile Case 1 (A=0, B=0) için 2 Nanosaniye Simülasyon Sonucu



Şekil 2: MAGIC Yöntemi ile Case 3 (A=1, B=0) için 2 Nanosaniye Simülasyon Sonucu

III. KIYASLAMA VE DEĞERLENDİRME

SIXOR yöntemi, asimetrik girişlerde (örneğin, Case 3'te 0.255 pJ) daha yüksek enerji tüketimi sergilese de, tek çevrimde XOR işlemini tamamlaması hız ve tasarım sadeliği açısından belirgin bir avantaj sunmaktadır. MAGIC yöntemi ise daha düşük enerji tüketimi (ortalama 0.023075 pJ) sergilemesine rağmen, NOR tabanlı yapısı nedeniyle birden fazla çevrim gerektirebilmekte ve işlem süresini uzatmaktadır. SIXOR'un initialization ve computation aşamalarındaki enerji tüketimi, MAGIC'e kıyasla daha düşüktür; bu da SIXOR'un enerji verimliliği açısından belirli uygulamalarda daha uygun olduğunu göstermektedir. Ancak, MAGIC'in çapraz çubuk dizisi yapısı, büyük ölçekli uygulamalarda paralel işlem kapasitesi sunarken, kaçak akım sorunları performansını etkileyebilir. Tablo 1, her iki yöntemin ortalama enerji tüketimlerini kıyaslamaktadır.

Tablo 3. SIXOR ve MAGIC Yöntemlerinin Ortalama Enerji Tüketim Değerleri

Yöntem	**Ortalama İnitialize Enerjisi (pJ)**	**Ortalama İşlem Enerjisi (pJ)**
SIXOR	1.9781	0.1355815
MAGIC	11.1152	0.023075

Initialization Enerjilerinin Toplam Enerji Tüketimine Etkisi: SIXOR ve MAGIC yöntemlerinin enerji tüketimlerini karşılaştırırken, yalnızca hesaplama enerjileri değil, initialization enerjileri de dikkate alınmalıdır. Simülasyon sonuçlarına göre, SIXOR'un ortalama initialization enerjisi 1.9781 pJ iken, MAGIC'in ortalama initialization enerjisi 11.1152 pJ'dir (Tablo 1). Bu, SIXOR'un başlangıç aşamasında MAGIC'e kıyasla yaklaşık 5.6 kat daha az enerji tükettiğini göstermektedir. MAGIC'in düşük hesaplama enerjisi (0.023075 pJ), yüksek initialization enerjisiyle dengelenmekte; bu durum, SIXOR'un toplam enerji tüketiminde (initialization ve hesaplama enerjileri dahil) daha verimli bir profil sunduğunu ortaya koymaktadır. Örneğin, dört işlem için SIXOR'un toplam enerji tüketimi (initialization ve hesaplama) yaklaşık 8.45 pJ iken, MAGIC'in toplam enerji tüketimi 44.55 pJ'e ulaşmaktadır. SIXOR'un tek çevrim

tasarımı, initialization sürecini sadeleştirerek enerji tüketimini minimize ederken, MAGIC'in NOR tabanlı yapısı ve çapraz çubuk dizisi mimarisi daha karmaşık bir başlangıç hazırlığı gerektirir. Bu sonuçlar, SIXOR'un enerji verimliliği açısından özellikle zaman kritik uygulamalarda daha uygun olduğunu doğrulamaktadır.

IV. BULGULAR

Simülasyon sonuçları, her iki yöntem için dört farklı senaryoda enerji tüketim değerlerini ortaya koymuştur. SIXOR yöntemi için toplam işlem enerjisi (2 ns - 1 ns farkı) şu şekilde ölçülmüştür: Case 1 için 0.066026 pJ, Case 2 için 0.151 pJ, Case 3 için 0.255 pJ ve Case 4 için 0.0703 pJ. Toplam işlem enerjisi 0.542326 pJ, ortalama işlem enerjisi ise 0.1355815 pJ'dir. MAGIC yöntemi için ise toplam işlem enerjisi şu şekilde ölçülmüştür: Case 1 için 0.0113 pJ, Case 2 için 0.031 pJ, Case 3 için 0.032 pJ ve Case 4 için 0.018 pJ. Toplam işlem enerjisi 0.0923 pJ, ortalama işlem enerjisi ise 0.023075 pJ'dir.

Initialization (1 ns) ve computation (2 ns) aşamalarındaki enerji tüketim değerleri de analiz edilmiştir. SIXOR'un 1 ns değerleri sırasıyla 0.0675 pJ, 1.9739 pJ, 1.9732 pJ ve 3.8978 pJ; 2 ns değerleri ise 0.133526 pJ, 2.1249 pJ, 2.2282 pJ ve 3.9681 pJ'dir. MAGIC'in 1 ns değerleri 9.3208 pJ, 11.120 pJ, 11.120 pJ ve 12.900 pJ; 2 ns değerleri ise 9.3321 pJ, 11.151 pJ, 11.152 pJ ve 12.918 pJ'dir. Şekil 1, SIXOR'un Case 1 için 2 ns simülasyon sonucunu; Şekil 2, MAGIC'in Case 3 için 2 ns simülasyon sonucunu göstermektedir.

V. TARTIŞMA

MAGIC, ortalama işlem enerjisi açısından SIXOR'a kıyasla daha verimli bir performans sunar (0.023075 pJ), ancak initialization enerjisi 11.1152 pJ ile SIXOR'un 1.9781 pJ'lik değerine göre 5.6 kat daha yüksektir. Bu fark, toplam enerji tüketiminde SIXOR'u (8.454726 pJ) MAGIC'e (44.5531 pJ) kıyasla 5.3 kat daha avantajlı kılar. SIXOR'un tek çevrimde XOR işlemini tamamlaması, özellikle AES ve SHA-256 gibi şifreleme algoritmalarında veya nöromorfik hesaplama sistemlerinde düşük gecikme gerektiren uygulamalarda hem hız hem de enerji verimliliği sağlar. Ancak, SIXOR'un farklı gerilimler (±1.2 V) gerektirmesi, bu gerilimleri üretip uygulamanın zorluğu nedeniyle çevresel devre tasarımında karmaşıklık yaratır; oysa MAGIC, NOT ve NOR işlemleri için tek bir gerilim (1.3 V) kullanarak daha sade bir tasarım sunar. Buna rağmen, SIXOR'un hız ve enerji avantajları, bu tasarım zorluğunu telafi edebilir.

Sonuç olarak, SIXOR, XOR işlemleri için hem enerji verimliliği hem de hız açısından MAGIC'e kıyasla daha avantajlı bir yöntemdir. MAGIC'in düşük işlem enerjisi, yüksek initialization enerjisiyle dengelenirken, SIXOR'un düşük initialization enerjisi ve tek çevrimde işlem tamamlaması, toplam enerji tüketimini ve işlem süresini optimize eder. Bu özellikler, SIXOR'u özellikle XOR tabanlı uygulamalarda (örneğin, gerçek zamanlı veri şifreleme veya sinaptik ağırlık güncellemeleri için nöromorfik sistemler) daha uygun bir seçenek haline getirir. Ancak, memristörlerin döngüden döngüye değişkenlikleri ve malzeme özelliklerinin enerji tüketimine etkisi, her iki yöntemin uzun vadeli performansını değerlendirmek için daha ayrıntılı araştırmalar gerektirmektedir.

VI. SONUÇLAR

Bu çalışma, memristör tabanlı bellek-içi hesaplama (IMC) sistemlerinde XOR ve NOR mantık işlemlerini gerçekleştirmek için tasarlanmış SIXOR (Single-Cycle In-Memristor XOR) ve MAGIC (Memristor-Aided loGIC) yöntemlerinin enerji tüketim analizlerini ve kıyaslamalarını detaylı bir şekilde sunmuştur. SIXOR, initialization enerjisi, işlem enerjisi, hız ve tasarım sadeliği açısından MAGIC'e kıyasla daha avantajlı bir yöntem olarak öne çıkmaktadır. Ortalama işlem enerjisi açısından SIXOR'un 0.1355815 pJ'lik değeri, MAGIC'in 0.023075 pJ'lik değerinden yüksek olsa da, SIXOR'un ortalama initialization enerjisi 1.9781 pJ ile MAGIC'in 11.1152 pJ'lik değerine kıyasla yaklaşık 5.6 kat daha

düşüktür. Bu durum, SIXOR'un toplam enerji tüketiminde (initialization ve işlem enerjileri toplamı: 8.454726 pJ) MAGIC'e (44.5531 pJ) kıyasla yaklaşık 5.3 kat daha verimli olduğunu göstermektedir. Ayrıca, SIXOR'un tek çevrimde XOR işlemini tamamlaması, MAGIC'in çok çevrimli NOR tabanlı yapısına kıyasla belirgin bir hız avantajı sağlar. Bu özellik, özellikle XOR tabanlı uygulamalarda (örneğin, AES veya SHA-256 gibi şifreleme algoritmaları ve nöromorfik hesaplama sistemlerinde sinaptik ağırlık güncellemeleri) düşük gecikme ve yüksek hız gereksinimlerini karşılamada SIXOR'u daha uygun bir seçenek haline getirir. SIXOR'un sade tasarımı, çapraz çubuk dizisi kullanılmadan çalışmasını mümkün kılarak potansiyel kaçak akım (sneak path) sorunlarını azaltır ve büyük ölçekli uygulamalarda güvenilirliği artırır. Buna karşılık, MAGIC'in çapraz çubuk dizisi mimarisi, paralel işlem kapasitesi sunmasına rağmen, yüksek initialization enerjisi ve çok çevrimli yapısı nedeniyle zaman kritik uygulamalarda sınırlı kalmaktadır.

Bu çerçevede, gelecekteki çalışmaların hem SIXOR hem de MAGIC yöntemlerinin büyük ölçekli sistemlerdeki performansını daha ayrıntılı bir şekilde incelemesi faydalı olacaktır. Özellikle memristör aygıt değişkenliklerinin (cycle-to-cycle variation) mantıksal doğruluk üzerindeki etkilerinin deneysel ve simülasyon tabanlı analizlerle araştırılması, bu yöntemlerin uygulanabilirliğine ilişkin daha derinlemesine bir anlayış sağlayabilir. Örneğin, direnç durumlarındaki %5'lik bir sapmanın işlem doğruluğuna etkisi veya bu değişkenliklerin enerji tüketimine yansıması detaylı bir şekilde değerlendirilebilir. Ayrıca, her iki yaklaşımın farklı mantık işlemleri (örneğin, AND, OR veya karmaşık kombinasyonlar) için genişletilebilirliği, mimari esneklikleri ve hata toleransı gibi faktörler bakımından incelenmesi, bellek-içi hesaplama sistemlerinin uzun vadeli uygulanabilirliğine katkı sunacak önemli bir araştırma yönü olarak öne çıkmaktadır. Bunun yanı sıra, SIXOR'un düşük initialization enerjisini ve hız avantajını daha da optimize etmek için yeni memristör malzemeleri (örneğin, TiO2 veya HfO2 tabanlı) veya hibrid CMOS-memristör devre tasarımlarının geliştirilmesi, yöntemin pratik uygulamalardaki etkinliğini artırabilir. Benzer şekilde, MAGIC'in yüksek initialization enerjisini azaltmaya yönelik stratejiler (örneğin, daha verimli sıfırlama mekanizmaları) üzerine çalışmalar, bu yöntemin enerji verimliliğini iyileştirebilir ve daha geniş bir uygulama yelpazesinde kullanılabilirliğini sağlayabilir.

KAYNAKLAR

[1] Li, H., Gao, B., Chen, Z., Zhao, Y., Huang, P., Ye, H., ... & Liu, X. (2015). A learnable parallel processing architecture towards unity of memory and computing. Scientific Reports, 5, 13330.

[2] Kvatinsky, S., Belousov, D., Liman, S., Satat, G., Wald, N., Friedman, E. G., Kolodny, A., & Weiser, U. C. (2014). MAGIC—Memristor-aided logic. *IEEE Transactions on Circuits and Systems II: Express Briefs, 61*(11), 895-899.

[3] Ben-Hur, R., & Kvatinsky, S. (2016). Memristive memory processing unit (MPU) controller for in-memory processing. *IEEE Transactions on Nanotechnology, 15*(5), 783-791.

[4] Yang, J. J., Strukov, D. B., & Stewart, D. R. (2013). Memristive devices for computing. *Nature Nanotechnology, 8*(1), 13-24.

[5] Shalf, J. M., & Leland, R. (2015). Computing beyond Moore's Law. *Computer, 48*(12), 14-23.

[6] Hu, M., Strachan, J. P., Li, Z., Grafals, E. M., Davila, N., Graves, C., ... & Williams, R. S. (2016). Dot-product engine for neuromorphic computing: Programming 1T1M crossbar to accelerate matrix-vector multiplication. *IEEE Transactions on Electron Devices, 63*(3), 1151-1158.