

Sistem İç Dinamiklerinden Kaynaklanan Isınma Sorununun En İyi Çözümü İçin Bir Algoritma ve Uygulaması

H. Kılıç
Atılım Üniversitesi Bilgisayar Müh. Bölümü, Ankara
hurevren@atilim.edu.tr

Özetçe

Doğal ve yapay yaşam sistemlerinde karşılaşılan ve yerel birimlerin dinamiklerinden kaynaklanan global ısınma probleminin çözümü için genel amaçlı bir tersine mühendislik yöntemi önerilmiştir. Özdevinirler ağı modelini kullanan yaklaşımın düşük güç kullanımlı Yerleşik Kendini Sınama - YKS sinama örüntüsü üretici tasarımında başarı ile uygulanabileceği gösterilmiştir.

Abstract

In most natural and artificial life systems local unit's interaction dynamics may result in emergence of global heat problem which may prevent proper functioning of the system. In this work, a general purpose reverse engineering algorithm is proposed to construct minimum heat dissipating automata network design. It is shown that the proposed approach can be successfully used in the design of low-power Test Pattern Generator for Built-In Self-Test architectures.

1. Giriş

Doğada gözlemediğimiz davranış biçimlerini silikon ortamda yapay olarak oluşturabilmek hem doğayı daha iyi anlamamıza hem de alternatif form ve/veya tasarım tekniklerinin geliştirilmesine olanak tanıyabilir. Langton tarafından “yapay yaşam” olarak adlandırılmış bu yaklaşım, Von Neumann [1] tarafından önerilmiş özdevinir ağları modelini kullanmaktadır. Langton, verilen global davranış biçimlerini üretebilecek hücresel özdevinirlerin (varsa) bulunması problemini tanımlamıştır [2].

Sonraları, “Tersine Mühendislik” olarak da adlandırılan konu üzerinde genetik mühendisliği, Çok Büyük Çapta Tümleşik-ÇBÇT (Very Large Scale Intergration-VLSI) devre tasarımı, ekonometri gibi farklı alanlardan araştırmacılar sırasıyla “genomik ağ tersine mühendisliği”, “örüntü üretici tasarımı”, “ortak evrilen (ekonomik) kıymetler” başlıkları altında çalışmışlardır [3]-[9]. Bu araştırmalarda genel amaç, gözlemlenen global sistem davranışlarının altında yatan yerel bazdaki ilişki ya da bağımlılıkların ortaya çıkarılması veya maliyet-etkin tasarımların gerçekleştirilmesidir.

Doğada gözlemediğimiz terleme, bir araya toplanma, tüylenme, kış uykusuna yatma gibi bireysel veya kollektif davranış biçimlerini şekillendiren önemli faktörlerden biri de sıcaklıktır. Ortamdan veya sistemin kendi iç dinamiklerinden kaynaklanan sıcaklık değişimleri hem doğal hem de yapay yaşam sistemlerinde gözardı edilmemesi gereken önemli bir unsurdur [10][11]. Örneğin, yapay bir sistem olan ÇBÇT kırmıklarında, kırmık aktivasyonundan ve/veya birimlerarası bağlantı yoğunluğundan kaynaklanan güç tüketimi ve sonucunda ortaya çıkan ısınma problemi, kırmığın işlevini sağlıklı olarak yerine getirememesine neden olabilmektedir. [12]’de ikili durum değerleri alabilen özdevinir ağı ile modellenmiş ayrık bir dinamik sistemi oluşturan birimlerin aktivasyonundan kaynaklanan ısınma probleminin çözümü için “en soğuk olana yönelim” buluşsalına dayanan polinom zamanlı bir tersine mühendislik algoritması önerilmiştir. Doğada gözlemlenebilen “negatif termotropi” (“sıcaktan kaçınma”) davranışından esinlenerek önerilmiş polinom zamanlı bu algoritmanın olası en düşük ısınma değerine sahip

özdevinir ağı topolojisini bulabilmesinin garantisi yoktur. Öte yandan, verilen ikili vektör dizisi ile tanımlanmış global davranış biçimini üretebilen en az toplam çıkış yelpazesi değerlerine sahip özdevinir ağını bulma probleminin Polinom Olmayan-Tam (NP-Complete) bir problem olduğu bilinen Küme Kapsama (Set Covering) problemine çevrilebileceği [7] ve [8]'deki çalışmalarda birbirinden bağımsız olarak gösterilmiştir. Bu iki çalışma en az bağlantılı etkileşim topolojisini bulabilmekte ancak en düşük ısınma değerine sahip topolojiyi bulmayı hedeflememekte ve dolayısıyla garanti etmemektedir.

Bu makalede, yapay yaşam bazlı tasarımlarda karşılaşılan “yerel birimlerin faaliyetlerinden kaynaklanan global ısınma problemi”nin çözümü için bir tersine mühendislik yöntemi önerilmiştir. Makalenin orjinal katkısı, verilen bir örüntü dizisi ile tanımlanmış herhangi bir davranış biçimini, “en az ısınma” ile üretebilen özdevinir ağını bulma yöntemini tanımlanmış olmasıdır. Dikkat edilmesi gereken önemli bir nokta, üzerinde çalışılan sistemin kapalı ve gerekirci bir sistem olduğu kabullenmesidir. Ayrıca, dışsal faktörlerden kaynaklanabilecek ısınma sorununun değil, sistemin iç dinamiklerinden kaynaklanan ısınma sorunu üzerinde çalışılmıştır. Makalenin pratikteki orjinal katkısı ise ÇBCT devrelerinin sınanmasında kullanılmak üzere verimli bir YKS sinama örüntüsü üreteci tasarım yönteminin önerilmiş olmasıdır.

Bölüm 2’de özdevinir ağı modeline ilişkin ön bilgiler verilmiştir. Bölüm 3’te önerilen algoritma ve ayrıntıları anlatılmıştır. Bölüm 4’te, algoritmanın pratik bir uygulama alanı olan Yerleşik Kendini Sınama – YKS (Built-In Self-Test) için örüntü üreteci tasarımı problemi tanımlanmış ve çözüme ilişkin elde edilen deneysel sonuçlar verilmiştir. Bölüm 5 makalenin sonuç kısmıdır. Doküman boyunca *örüntü/vektör* ve *ısınma/güç harcanması* terimleri dönüşümlü olarak kullanılmıştır.

2. Model

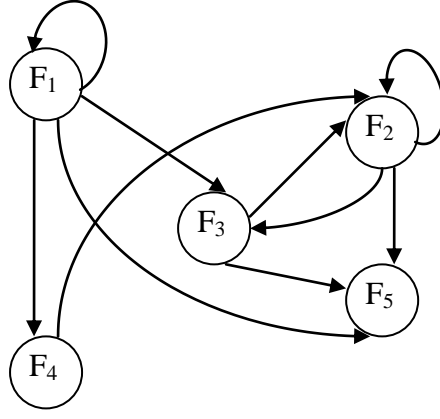
Bir özdevinir ağı, sonlu durum uzayı S ve birbirine bağlı özdevinirlerin sayısı n olmak üzere, S^n tanım kümesinden S^n değer kümesine tanımlanmış gerekirci bir F eşlemesi olarak tanımlanabilir. Özdevinirlerin bağlantı topolojisi F eşlemesi tarafından belirlenir. F eşlemesinin i 'inci özdeviniri j 'inci özdevinire bağımlı ise i 'inci özdevinir j 'inciden girdi kabul eder. Özdevinirler arasındaki bu tip bir etkileşim yapısı yönlendirilmiş çizge ile gösterilebilir. Ağın t

zamanındaki durumu, S^n uzayında tanımlı X_t vektörüyle gösterilir. Buna göre, ağ dinamikleri $X_{t+1} = F(X_t)$ eşitliği ile tanımlanır. Her bir zaman adımında, i 'inci özdevinir bir sonraki durumunu F_i eşlemesinin durum geçiş kuralına göre belirler. Senkronize bir şekilde gerçekleşen özdevinir durum değişiklikleri, ağın global evrimi etkisini oluşturur.

Sonlu durum uzayı S için $|S|=d$ ise d^n evrim adımı sonucunda özdevinir ağı durum uzayında ya bir döngüye girer ya da sabit bir noktaya yakınsar. Özdevinir ağının diğer bir özelliği ise sistemi oluşturan özdevinirlerin bir sonraki adımdaki durumlarının komşuluklarında bulunan özdevinirlerin şu anki durumlarından çıkarılabiliyor olması, bir başka deyişle gerekirci olmasıdır. Şekil 1’de örnek bir özdevinir ağının etkileşim topolojisi, yönlendirilmiş çizge şeklinde gösterilmiştir.

Şekil 1’den, F_3 özdevinirinin bir sonraki durumuna F_1 ve F_2 özdevinirlerinin şu anki durumlarına bakarak karar verdiği anlaşılmalıdır. Tablo 1’de, Şekil 1 için geçerli olan örnek durum geçiş kuralları görülmektedir. Örnekteki beş özdevinir (yani $n=5$), aşağıdaki Z matrisi ile tanımlanmış olan a, b veya c durum değerlerini alabilmektedir (yani $S=\{a, b, c\}$).

$$Z = \begin{array}{|c|c|c|c|c|} \hline a & c & c & b & c \\ \hline b & a & a & c & c \\ \hline c & b & c & b & a \\ \hline a & a & b & b & c \\ \hline b & a & c & c & b \\ \hline \end{array}$$



Şekil-1: Örnek bir özdevinir ağı topolojisi

Tablo-1: Örnek durum geçiş kuralları

F ₁	F ₂	F ₃	F ₄	F ₅
a→b	ccb→a	ac→a	a→c	acc→c
b→c	aac→b	ba→c	b→b	baa→a
c→a	bc b→a	cb→b	c→b	cbc→c
	abb→a	aa→c		aab→b

3. Algoritma

Bu bölümde tanımlanan algoritmanın amacı, her birinin boyutu n olan ve farklı durum değerleri alabilen, birbirinden farklı (X_1, X_2, \dots, X_m) vektör ile tanımlanmış gerekirci bir özdevinir ağı durum dizisini verilen sırada üretebilen en az çıkış yelpazesi değerine sahip, tam veya kısmi tanımlı özdevinirleri (varsa) bulmaktır. Ağı oluşturan özdevinir sayısı n^m 'dir. Algoritmanın girdisi, m satır ve n sütundan oluşan iki boyutlu aktivite matrisidir $(Z_{m \times n})$. Algoritmanın çıktısı ise girdi matrisinin içeriğine göre tam veya kısmi olarak tanımlanabilen özdevinirler ve bunların etkileşim topolojisidir. En az ısınan etkileşim topolojisini bulabilmemiz için öncelikle sistemde ısınma yaratan nedenin ne olduğunu tanımlamamız gereklidir.

3.1 Isınma Metriği

Sistemin ısınması, Z ile tanımlı aktivite matrisindeki "değer değişimleri" ve her bir özdevinirin dinamiğini etkilediği özdevinirlerin sayısı (yani çıkış yelpazesi

değeri) ile ilişkilidir. Bir sütun boyunca değişen değerlerin (örneğin $a \rightarrow b, b \rightarrow c$ gibi) sayısı ne kadar az ise ve çıkış yelpazesi değerleri ne kadar küçük olursa bu birimin sistemin ısınmasına olan katkısı o derece az olacaktır. Ardışık olarak değişmeyen değerler (örneğin, $a \rightarrow a, b \rightarrow b$) enerji gerektirmediği için sistemde ısınmaya sebep olmaz. Devre terminolojisinde Ağırlıklı Anahtarlama Aktivitesi (Weighted Switching Activity) olarak bilinen tanımlama ısınma metriği tanımımız için de uygundur. Buna göre, her bir i vektörünün uygulanması sonucunda sistemin ısınmasına olan katkısı (A^i) tanımlayan eşitlik şöyledir [13]:

$$A^i = \sum_{j=1}^n Y_j D_{ij}$$

Eşitlik (1)

Eşitlikteki Y_j : j 'inci özdevinirin çıkış yelpazesi değeridir. D_{ij} değişkeni, j 'inci özdevinirin i 'inci adımda $(i-1)$ 'inci adımdaki değerinden farklı bir değer alıp almamasına göre ısı açığı çıktığı/çıkmadığı bilgisini içerir. D_{ij} , eğer $Z_{ij} = Z_{(i-1)j}$ ise 0, aksi halde 1 değerini alır. Sistemin m adet vektörü üretmesi sonucunda toplam ısınması (I) ise

$$I = \sum_{i=1}^m A^i$$

Eşitlik (2)

ile tanımlanabilir.

3.2 Önerilen Algoritma ve Ayrıntıları

Isınma miktarını enazlayacak özdevinir ağını elde etmek için, verilen vektör dizisini üretebilen, her bir özdevinir için en az çıkış yelpazesi değerine sahip olup anahtarlama aktivitesi en az olan özdevinirler hedeflenmektedir. n sayıda özdevinirden oluşan bir ağ için 2^{n^2} farklı özdevinir etkileşim topolojisi bulunabilir. Öte yandan, verilen bir $Z_{m \times n}$ aktivite matrisini en az ısınmayla üretebilecek özdevinir ağı sayısı birden fazla olabilir. Ayrıca, k tane komşusu olan ve d farklı durum değeri alabilen bir özdevinir için d^{k^*} farklı tam tanımlı özdevinir durum geçiş fonksiyonu vardır. Sonuç olarak, arama uzayı boyutu oldukça büyük bir değer olan $2^{n^2} * d^{k^*}$ 'dir.

Tanım: Verilen bir $Z_{m \times n}$ aktivite matrisi için, $1 < p, q \leq m, 1 \leq j \leq n$ olmak üzere, eğer $Z_{pj} \neq Z_{qj}$ ise C_{pq}^j 'e

j 'inci özdevinir için p, q ile tanımlı bir "değişiklik olayı" denir.

Şekil 2'deki çevrim algoritması, verilen bir $Z_{m \times n}$ aktivite matrisinden bir küme kapsama problemi üretir. Üretilen problemin bilinen "dallan ve sınırla" yöntemleriyle çözümü ise aranan özdevinir ağı etkileşim topolojisini verir.

```

CEVIR(matrix Z, int m, int n)
{
  /* Isınma katsayılarını hesapla */
  j=1 den n'ye kadar {
    aj=0;
    i=2' den m'ye kadar {
      Eğer  $Z_{(i-1),j} \neq Z_{ij}$  ise aj = aj+1;
    }
  }
  /*Aşağıdaki amaç fonksiyonunu oluştur */
  MINIMIZE
   $Z = a_1E_{11} + a_2E_{12} + \dots + a_nE_{1n} + \dots + a_1E_{n1} + \dots + a_nE_{nm}$ 
   $G = \emptyset$  /* Kısıt kümesini sıfırla */
  Herbir Fj özdeviniri için
    Herbir (Zpj, Zqi) ikilisi için {
      T kısıt dizgisini boşalt;
      Eğer Cpqj bir değişiklik olayı ise {
        Herbir Fs özdeviniri için
          Eğer  $Z_{(p-1),s} \neq Z_{(q-1),s}$  ise
            Ejs değişkenini T kısıt dizgisine ekle;
          T'nin sonuna kısıt " $\geq 1$ " kısıtını ekle;
          G = G  $\cup$  T;
        }
      }
  }
}

```

Şekil-2: Çevrim algoritması sözde programı.

Algoritma herhangi p, q ikilisi için mevcut değişiklik olayının sebebini özdevinirlerin bir önceki adımdaki durumlarına dayandırır. Bir başka deyişle, değişiklik olayının sebebi bir önceki adımdaki global sistem durumlarındaki farklılıklar ile açıklanabilir. Örneğin ikili değer alan bir Z matrisinde $Z_{3,2}$ ve $Z_{4,2}$ değerleri farklı ise (yani değişiklik olayı olmuş ise) bunun sebebi matrisin ikinci (yani $p-1$) ve üçüncü (yani $q-1$) satırlarındaki vektörlerin farklılıklarında aranmalıdır. Söz konusu iki vektör sırasıyla (00111) ve (00001) ve $n=5$ olsun. Bu durumda değişiklik olayının sebebi

üçüncü veya dördüncü özdevinirlerden en az biri olabilir. Bu durum ikinci özdevinir için tanımlı $p=3$ ve $q=4$ değişiklik olayı için $E_{23}+E_{24} \geq 1$ kısıtının mevcut kısıt kümesine eklenmesiyle ifade edilir (Bkz. Şekil 3). Burada E_{ij} değişkeni j 'inci özdevinirin i 'inci özdevinir üzerindeki değişiklik etkisini göstermektedir. Sistemin geneline bakıldığında E_{ij} değerlerinin ağırlıklı toplamının minimum olması hedeflenmektedir. Hedef aynı zamanda en soğuk olan tasarımı bulmak olduğu için i 'inci özdevinir, anahtarlama aktivitesi en az olan özdeviniri tercih etmelidir. Bu sebeple algoritmanın ilk sekiz satırında her bir j özdeviniri için anahtarlama sayısı a_j hesaplanır ve bu değer amaç fonksiyonuna E_{ij} değişkeninin çarpanı olarak girer (yani $a_j \cdot E_{ij}$).

Çevrim algoritmasının zaman karmaşıklığı $O(m^2n^2)$ 'dir. Verilen bir Z matrisi için algoritma tarafından üretilen küme kapsama problemi ve bu problemin çözümü sonucunda elde edilmiş olan etkileşim topolojisi ve durum geçiş kuralları sırasıyla Şekil 3 ve 4'te görülmektedir. Herhangi bir Z matrisi için en az ısınma değerini üreten birden fazla etkileşim topolojisi bulunabilir. Şekil 4'te bunlardan biri verilmiştir. Elde edilen sonuçlara Eşitlik (1) ve (2) uygulanırsa toplam ısınma miktarının $I=4 \cdot 3 + 0 \cdot 3 + 0 \cdot 2 + 3 \cdot 1 + 1 \cdot 2 = 17$ olduğunu görülür. Bu miktar, örnekte verilen Z matrisini en az ısınarak üretebilen özdevinir ağıının ısınma değeridir.

$$Z = \begin{bmatrix} 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 \\ 1 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 \end{bmatrix}$$

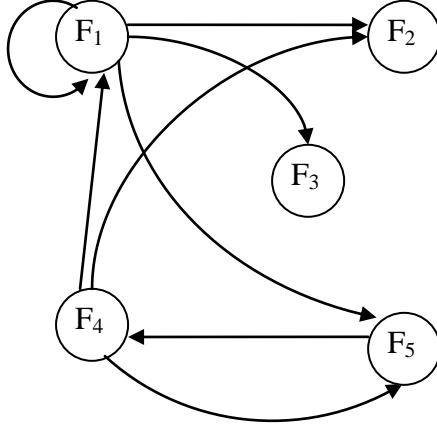
$$\begin{aligned} \text{MINIMIZE } Z = & 3E_{11} + 3E_{12} + 2E_{13} + E_{14} + 2E_{15} + \\ & 3E_{21} + 3E_{22} + 2E_{23} + E_{24} + 2E_{25} + \\ & 3E_{31} + 3E_{32} + 2E_{33} + E_{34} + 2E_{35} + \\ & 3E_{41} + 3E_{42} + 2E_{43} + E_{44} + 2E_{45} + \\ & 3E_{51} + 3E_{52} + 2E_{53} + E_{54} + 2E_{55} \end{aligned}$$

Subject to:

$$\begin{aligned} E_{11}+E_{12}+E_{13}+E_{14}+E_{15} & \geq 1 \\ E_{13}+E_{14} & \geq 1 \\ E_{11}+E_{12} & \geq 1 \\ E_{21}+E_{22}+E_{23}+E_{24}+E_{25} & \geq 1 \\ E_{23}+E_{24} & \geq 1 \\ E_{21}+E_{22} & \geq 1 \end{aligned}$$

$$\begin{aligned}
E31+E32+E35 &\geq 1 \\
E31+E32+E33+E34+E35 &\geq 1 \\
E31+E32+E33+E34 &\geq 1 \\
E31+E32 &\geq 1 \\
E41+E42+E45 &\geq 1 \\
E41+E42+E43+E44+E45 &\geq 1 \\
E43+E44+E45+E53+E54+E55 &\geq 1 \\
E51+E52+E53+E54 &\geq 1 \\
E51+E52 &\geq 145 \geq 1
\end{aligned}$$

Şekil-3: Örnek Z aktivite matrisi ve bu matris için algoritmanın ürettiği küme kapsama problemi.



Şekil-4: Üretilen küme kapsama probleminin çözümü sonucunda elde edilen özdevinir ağı etkileşim topolojisi.

Tablo-2: Üretilen küme kapsama probleminin çözümü sonucunda elde edilen özdevinir ağı durum geçiş kuralları.

F ₁	F ₂	F ₃	F ₄	F ₅
11→0	11→0	1→1	0→1	11→1
01→0	01→0	0→0	1→0	01→1
00→1	00→1			00→1
10→0	10→0			10→0

4. Uygulama ve Deneysel Sonuçlar

Bu kısımda, algoritmanın pratik uygulama alanlarından biri olan YKS için örüntü üretici

tasarımı problemi tanımlanmış ve elde edilen deneysel sonuçlar verilmiştir.

4.1 Uygulama

YKS yaklaşımı ÇBÇT devrelerini sınamak için uygulanan popüler bir sınama tasarım tekniğidir. Tipik bir YKS yaklaşımındaki adımlar [14]:

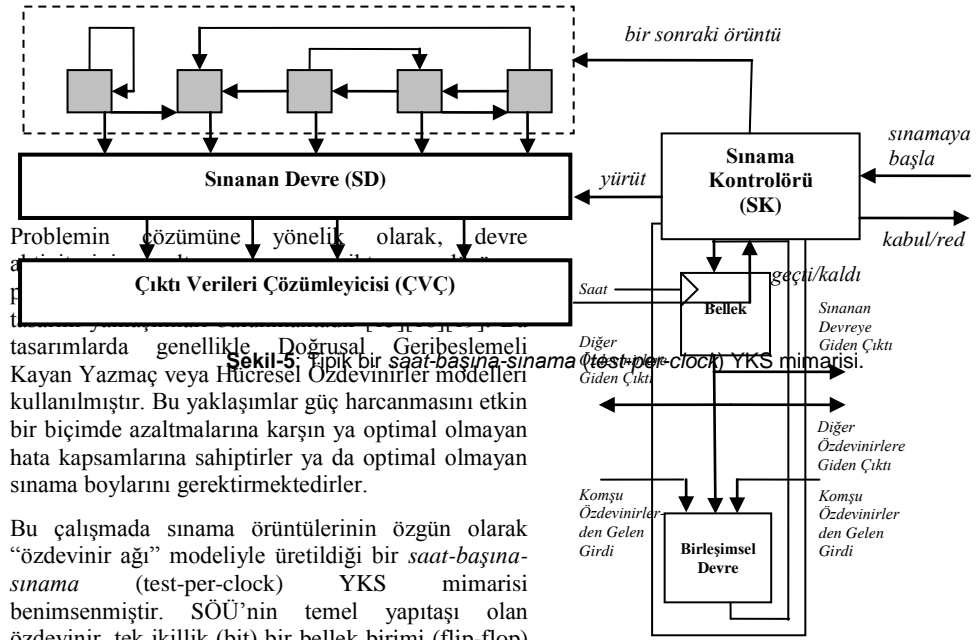
- 1) Kırmık üzerinde sınama örüntüsü üretimi,
- 2) Üretilen örüntülerin Sınanan Devreye (SD) uygulanması,
- 3) SD çıktılarının yine kırmık üzerindeki Çıktı Verileri Çözümleyicisi (ÇVÇ) kullanılarak çözümlenmesi,
- 4) Sınama Kontrolörü'nün (SK) kırmığın hatalı olup olmadığına karar vermesidir (Şekil-5).

Verimli bir Sınama Örüntüsü Üretici'nin (SÖÜ) tasarımı konusu ilk adımla ilgilidir ve YKS yaklaşımında önemli bir yer tutar. Yüksek Hata Kapsamlı (HK) sınama örüntü dizilerinin en az boyuttaki donanımla, en az sınama süresinde ve en az Güç Harcaması (GH) ile üretimi, verimli bir SÖÜ tasarımında temel amaçtır. Öte yandan, ÇBÇT devrelerinin sınama modunda normal çalışma moduna göre daha yüksek aktivasyon gerektirdiği, dolayısıyla daha çok güç harcanmasına sebep oldukları, sonuç olarak ortaya çıkan yüksek ısınmanın kırmığın sağlıklı bir şekilde sınanmasını engellediği bilinmektedir [15][16]. YKS için SÖÜ tasarımının ısınma özelliklerini tanımlayan üç temel parametre vardır [17]:

- 1) Sınama boyunca harcanan toplam güç (I) (Bkz. Eşitlik (2)),
- 2) Ortalama güç tüketimi (I/Sınama Boyu),
- 3) Bir saat çevrimi anında harcanan en çok güç miktarı ($\max(A^1)$) (Bkz. Eşitlik (1))

Madde 2'deki Sınama Boyu (SB) uygulanan sınama vektörlerinin sayısını vermektedir.

Sınama Örüntüsü Üretici (SÖÜ)



tasarımlarda genellikle Doğrusal Geribeslemeli Kayan Yazmaç veya Hücreli Özdevinirler modelleri kullanılmıştır. Bu yaklaşımlar güç harcanmasını etkin bir biçimde azaltmalarına karşın ya optimal olmayan hata kapsamına sahiptirler ya da optimal olmayan sınamaya boylarını gerektirmektedirler.

Bu çalışmada sınamaya örüntülerinin özgün olarak "özdevinir ağı" modeliyle üretildiği bir saat-başına-sınama (test-per-clock) YKS mimarisi benimsenmiştir. SÖÜ'nin temel yapıtaşı olan özdevinir, tek ikilik (bit) bir bellek birimi (flip-flop) ve özdevinire ait durum geçiş kurallarını bulunduran birleşimsel bir birimden (örneğin Programlanabilir Mantık Dizisi) oluşacak şekilde tasarlanmıştır. (Bkz Şekil 6).

4.2 Deneysel Sonuçlar

Deneylerde ISCAS-85 denektaşı verileri kullanılmıştır[20]. Sınama örüntü dizilerini üretmek için CADENCE yazılımının Otomatik Sınama Örüntüsü Üretim-OSÖÜ (Automatic Test Pattern Generation-ATPG) aracı kullanılmıştır. Önerilen algoritma C programlama dilinde kodlanmıştır. Üretilen küme kapsama problemleri CPLEX optimizasyon paketi kullanılarak çözülmüştür. Deneyler sırasında kullanılan donanım Pentium 4 2.00 GHz makinadır.

Şekil-6: SÖÜ'yu oluşturan özdevinirlerin iç yapısı.

Önceki kısımda önerdiğimiz yaklaşıma benzer bir yaklaşım, sınamaya örüntülerinin üretiminde "doğrusal olmayan karma hücreli özdevinirler" modelinin kullanıldığı YKS mimarisidir [13]. Optimizasyon için Rastgele Mutasyonlu Tepe Tırmanma - RMTT (Random Mutation Hill Climbing - RMHC) arama algoritmasının kullanıldığı bu yaklaşım birleşimsel devrelerin düşük güç harcamalı sınaması amacıyla önerilmiştir. Ancak ardışıl devre sınaması için doğrudan kullanılamamaktadır. Öte yandan, devre hatalarının çoğunun birleşimsel bloklardan çok ardışıl bloklarda meydana geldiği bilinmektedir [14]. Kısım 3'te önerdiğimiz algoritma hem birleşimsel hem de ardışıl devreleri sıyanan verimli SÖÜ tasarımları için kullanılabilir. Verilen bir vektör dizisini üretebilen en düşük toplam aktiviteye sahip olan özdevinir ağlarının sayısı birden fazla olabilir. Deneylerde, en düşük toplam aktiviteyi üretebilen olası topolojilerden birinin bulunması yeterli görülmüştür.

Tablo 3'te örnek ISCAS-85 denektaşı verileri için alınan sonuçlar görülmektedir. Önerdiğimiz

yöntemle, %100 hata kapsamlarına olabilecek en az sınamaya boyu değerleri ile erişilmiştir. Başka bir deyişle, toplam sınamaya zamanı ve tüketilen toplam güç en aza indirgenmiştir. Tabloda birleşimsel devreler için verilmiş sonuçlar ardışıl devreler için de alınabilir. Tablodaki sonuçlar *global-en-iyi* sonuçlardır.

c880 devresinde gözlemlenebileceği gibi problem boyutu arttıkça ($n > 50$ gibi) harcanan Ana İşlem Birimi (AİB) zamanı ani artış göstermektedir. Bu sebeple aslında 10 adet olan ISCAS-85 denektaşı devrelerinin 4 tanesi için *global-en-iyi* sonuçlar gereken çalışma süreleri uzunluğu sebebiyle alınmamıştır. [12]'de önerilen ısıyönelimli (termotropik) algoritma polinom zamanlıdır ancak elde edilen çıkış yelpazesi değerlerinin büyüklüğü ağırlıklı anahtarlama aktivitesini arttırmakta sonuçta elde edilen örüntü başına ortalama ısınma ve toplam ısınma miktarları artmaktadır.

Düşük hata kapsamı değerleriyle güç tüketimi daha az olan tasarımlar elde etmek mümkündür [21]. Ancak bu durum doğal olarak sınamaya kalitesini düşürür ve tercih edilmeyen bir durumdur. Diğer taraftan [13]'de, sınamaya boyunu arttırmak suretiyle hata kapsamı %100 olan optimizasyon sonuçları elde

edilmiştir. Bu yaklaşımda çıkış yelpazesi değerleri tüm birimler için sabit aynı (=3) olmakla beraber, sınamaya boyundaki artış anahtarlama aktivitesini arttırmakta ve sonuç olarak ortalama ve toplam güç tüketimi (ısınma) değerleri artmaktadır. Bu çalışmada da uzun AİB zamanı sorunundan dolayı sadece 6 adet birleşimsel devre için sonuçlar verilmiştir. Bu sonuçlara bakıldığında, c432 devresi dışında kalan beş devre için önerdiğimiz yöntemle alınan örüntü başına ortalama ve dolayısıyla toplam ısınma değerlerinin daha iyi olduğu görülmüştür. Alan maliyetleri açısından önerdiğimiz yöntemle göre daha iyi olan [13]'teki yaklaşımın olası bir dezavantajı, rastgele seçilen ilk durum değerleri ve özdevinir durum geçiş kurallarına olan hassasiyettir. Kötü seçilmiş ilk değerler yöntemin zaman performansını önemli ölçüde düşürebilmektedir.

Önerdiğimiz yöntemin bir dezavantajı, düzgün olmayan bağlantı biçimi ve bağlantı karmaşıklığıdır. Düzgünlük ve basitlik ÇBÇT devrelerinde aranan önemli bir özelliktir. Ancak, önerdiğimiz yaklaşımda, ortalama çıkış yelpazesi değerlerinin, katlanarak artan problem boyutuna (m ve n değerleri) karşın fazlaca artmama şeklinde bir direnç gösteriyor olması, bağlantı karmaşıklığının sayısal olarak sınırlı kalması açısından olumludur.

Tablo-3: Örnek ISCAS-85 devreleri için elde edilen optimizasyon sonuçları

Devre Adı	Sınamaya Boyu (m) [sayı]	Örüntü Boyu (n) [sayı]	Hata Kapsamı [%]	Örüntü Başına Ortalama Isınma	Toplam Isınma [sayı]	Bir saat periyodundaki en çok ısınma [sayı]	AİB Zamanı [sn.]
c432	83	36	100.0	110,84	9089	215	147
c499	68	41	100.0	56,54	3788	174	17
c880	67	60	100.0	154,52	10198	263	10426
c1355	111	41	100.0	97,48	10723	257	31
c1908	162	33	100.0	143,43	23093	262	82
c6288	47	32	100.0	68,63	3157	167	18

başarı ile uygulanabileceği gösterilmiştir.

5. Sonuç

Doğal yaşam sistemlerinde de gözlemlediğimiz ısınma probleminin, yapay yaşam sistemlerinde çözümüne yönelik bir algoritma önerilmiştir. Önerilen tersine mühendislik algoritması, farklı durum değerleri ile modellenmiş ayrık dinamik bir sistemi oluşturan birimlerin aktivasyonundan kaynaklanan ısınma problemini çözmektedir. Önerilen yöntemin düşük güç kullanımlı YKS sınamaya

Özdevinir tanımlamalarının birbirlerinden bağımsız bir şekilde bulunabilmeleri sebebiyle, algoritma kolaylıkla paralel hale getirilebilir. Önerilen yöntem, problem boyutunun büyük olduğu ($n \approx 10000$) Genomik Düzenleyici Ağlar'ın (Genomic Regulatory Networks) tersine mühendisliği problemi için doğrudan kullanılamamakla birlikte problem boyutunun genellikle yüzler mertebesinde olduğu ortak evrilen ekonomik kıymet dinamiklerinin

araştırılmasında yararlı olabilir. Ayrıca, ısı değişikliği gibi yaşam ortamını tanımlayan basınç, nem gibi diğer faktörler, ilgili metriklerin ve uygun tersine mühendislik yaklaşımlarının kullanılması yoluyla alternatif tasarım tekniklerinin geliştirilmesine olanak tanıyabilir.

Kaynakça

- [1] **Aspray, W., and Burks, A.W.**, 1987. Papers of John Von Neumann on Computing and Computer Theory, MIT Press, Cambridge Mass.
- [2] **Langton, C.**, 1986. Studying Artificial Life with Cellular Automata, *Physica D*, vol **22**, pp. 120-149.
- [3] **Somogyi, R.**, 2000. Genetic Network Inference: from co-expression clustering to reverse engineering, *Bioinformatics*, 16, No. **8**, pp. 707-726.
- [4] **Akutsu, T., Miyano, S., and Kuhara, S.**, 2000. Inferring qualitative relations in genetic networks and metabolic pathways, *Bioinformatics*, 16, No. **8**, pp. 727-734.
- [5] **Chiusano, S., Corno, F., Prinetto, P., Reorda, M.S.**, 1997. Cellular automata for Deterministic Sequential Test Pattern Generation, *IEEE VLSI Test Symp.'97*, pp. 60-65.
- [6] **Güler, M. and Kılıç, H.**, 1988. Built-In Self-Test Generator Design Using Non-Uniform Cellular Automata Model, *IEE Proceedings - Circuits, Devices and Systems*, 145, No: **3**, pp 155-161.
- [7] **Ideker, T.E., Thorsson, V., and Karp, R.M.**, 2000. Discovery of Regulatory Interactions Through Perturbation: Inference and Experimental Design, *Proc. of Pacific Symposium on Biocomputing 5*, pp. 302-313.
- [8] **Kılıç, H.**, 2003. An Automata Network Reverse Engineering Algorithm Using System Statics And Dynamics, *Proc. of 7th World Multiconference on Systemics, Cybernetics and Informatics, Orlando, USA, in July 27-30, vol 2*, pp. 284-287.
- [9] **Kılıç, H., Altay-Salih, A. and Güler M.**, 1999. A Reverse Engineering Methodology for the Interpretation of Stock Price Dynamics, *International ICSC Symposium on Soft Computing in Financial Markets, SCFM'99, June, Rochester, New York, USA*, pp 598-604.
- [10] **Thompson, A.**, 1997. Temperature in Natural and Artificial Systems, *Proc. of 4th European Conference on Artificial Life*, pp. 388-397.
- [11] **Hoar, W.S.**, 1983. *General and Comparative Physiology*, Prentice-Hall, 3rd Edition.
- [12] **Kılıç, H.**, 2003. Yapay Yaşam Sistemlerinin Dinamiklerinden Kaynaklanan Isınma Problemi İçin Bir Algoritma, *Proceedings of International Twelfth Turkish Symposium on Artificial Intelligence and Neural Networks, (in press)*.
- [13] **Corno, F., Rebaudengo, M., Reorda, M. S., Squillero, G., and Violante, M.**, 2000. Low power BIST via non-linear hybrid cellular automata, *18th IEEE VLSI Test Symposium*, pp. 29-34.
- [14] **Steininger, A.**, 2000. Testing and built-in self-test – A survey, *Journal of Systems Architecture* vol **46**, pp. 721-747.
- [15] **Bardell, P.H., Mcanney, W.H and Savir, J.**, 1987. *Built-In-Self-Test for VLSI: Pseudorandom Techniques*, (Wiley & Sons, New York)
- [16] **Zorian, Y.**, 1993. A Distributed BIST Control Scheme for Complex VLSI Devices, *IEEE VLSI Test Symposium*, pp. 4-9.
- [17] **Virazel, A. and Wunderlich, H.-J.**, 2002. Third VIVA Schwerpunkt-Kolloquium, Chemnitz, Germany, March 18-19, pp. 128-135.
- [18] **Zhang, X., Roy, K. and Bhawmik, S.**, 1999. *POWERTEST: A Tool for Energy Concious*

Weighted Random Pattern Testing, IEEE Int. Conf. On VLSI Design.

[19] **Wang, S. and Gupta, S.K.**, 1999. LT-RTPG: A New Test-Per-Scan BIST TPG for Low Heat Dissipation, IEEE International Test Conference, pp 85-94.

[20] **Brglez, F. and Fujiwara, H.**, 1985. A Neutral List of 10 combinational circuits and a target translator in FORTRAN. IEEE ISCAS, ISCAS'85, pp. 663-698.

[21] **Zhang, X. and Roy, K.**, 1999. Design and Synthesis of Programmable Low Power Weighted Random Pattern Generator, 5th IEEE International On-Line Testing Workshop.