

# Birleşimsel Devreler için Fourier Analizi Tabanlı Otomatik Test Örüntüsü Oluşturma

## Fourier Analysis-based Automatic Test Pattern Generation for Combinational Circuits

Tolga Ayav  
Bilgisayar Mühendisliği Bölümü  
İzmir Yüksek Teknoloji Enstitüsü  
İzmir, Turkey  
tolgaayav@iyte.edu.tr

**Özetçe** —Boole işlevlerinin Fourier analizi son on yılda bilgisayar bilimlerinde oldukça ilgi çeken bir konu olmasına rağmen henüz çok az uygulama alanı bulabilmiştir. Bu çalışma birleşimsel mantık devrelerinin testi için Fourier analizi tabanlı bir otomatik test örüntüsü oluşturma metodu sunmaktadır.

**Anahtar Kelimeler**—Birleşimsel devre, Fourier analizi, Walsh dönüşümü, otomatik test örüntüsü oluşturma.

**Abstract**—Fourier analysis of boolean functions has attracted great attention from computer scientists in the last decade but it still has few application areas. This work presents a Fourier analysis-based automatic test pattern generation method for combinational circuits.

**Keywords**—Combinational circuit, Fourier analysis, Walsh transformation, automatic test pattern generation.

### I. GİRİŞ

Tümleşik devreler üretim hataları nedeniyle üretim sonunda testlere tabi tutulur. Sayısal tümleşik devrelerde takılma hatası, gecikme hatası, köprüleme hatası gibi farklı türde hatalar olabilir [1].  $n$  girişli bir birleşimsel devrenin tam kapsamlı testi için  $2^n$  farklı girişi denemek gerekmektedir. Tam kapsamlı test büyük devreler için çok büyük test zamanı anlamına gelir. Hızlı seri üretim açısından en fazla hata kapsama oranına sahip en küçük test alt kümesini bulmak bu anlamda büyük önem taşır. Roth'un 1966'da önerdiği D-Algoritması bu alanda bir kilometre taşı sayılabilir, öyle ki D-Algoritması veya türevleri günümüzde endüstride hala kullanılmaktadır. Günümüze kadar D-Algoritmasına, ikili karar diyagramlarına, Boole kısmi türevlerine dayanan bir çok otomatik test örüntüsü oluşturma metodu geliştirilmiştir [2] [3] [4]. Bu çalışmada, Fourier analizine dayanan bir metod önerilecektir.

Uygulama alanı çok geniş olan Boole işlevleri birleşimsel mantık devrelerinin çıkışlarını girişleri cinsinden belirtmekte kullanılırlar.  $n$  girişli ve  $m$  çıkışlı bir devre aşağıda belirtildiği şekilde  $m$  adet Boole işlevi ile belirtilir:

$$f_i(\mathbf{x}) : \mathbb{B}^n \rightarrow \mathbb{B} \quad \forall i \in \{1, 2, \dots, m\}, \mathbb{B} = \{0, 1\}.$$

Boole işlevlerinin spektral analizi daha çok bilgisayar bilimlerinde anahtarlama işlevlerini incelemekte kullanılmıştır ve son on yılda çok çalışılmıştır [5]. Bir kaç çalışma dışında da devrelerde ve özellikle de test alanında pek uygulama alanı bulamamıştır. Bu çalışmada birleşimsel devrelerin testi için Fourier analizine dayalı bir yöntem sunulmaktadır.

Bölüm II'de Boole işlevlerinin Fourier analizi anlatılmaktadır. Bölüm III test girişlerinin bulunmasını, Bölüm IV ise test girişleri kümesinin eniyilenmesini anlatmaktadır. Çalışma Bölüm V ile sonuçlandırılmaktadır.

### II. FOURIER ANALİZİ

Boole işlevlerinin spektrum analizinde Fourier, Walsh, Walsh-Hadamard dönüşümlerinin hepsi aynı anlama gelmektedir ve literatürde birbirinin yerine kullanılmaktadır. Bu çalışmada özellikle bilinen sürekli-zaman Fourier analizinden farkını ortaya koymak için Walsh ifadesi sıklıkla tercih edilmiştir. Herhangi bir  $x(t)$  sinyalinin Fourier (Walsh) açılımı aşağıdaki gibi yazılır:

$$x(t) = a_0 + \sum_{i=1}^{\infty} (a_i \text{sal}(i, t) + b_i \text{cal}(i, t)) \quad (1)$$

Burada  $a_0$  doğru akım bileşeni,  $a_i$  ve  $b_i$  ise Walsh spektrum katsayılarıdır.  $\text{sal}$  ve  $\text{cal}$  işlevleri ise Fourier açılımındaki sinüs ve kosinüs işlevlerine olan benzerlikleri nedeniyle bu şekilde isimlendirilmiş olup aşağıda gösterilen Walsh işlevlerine denk düşerler:

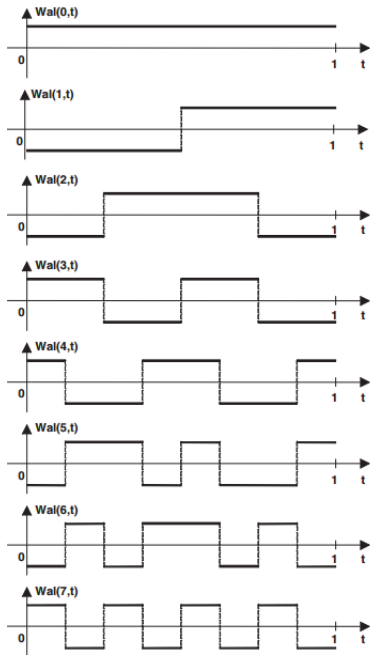
$$\text{sal}(\omega, x) = \text{wal}(2\omega - 1, x) \quad (2)$$

$$\text{cal}(\omega, x) = \text{wal}(2\omega, x) \quad (3)$$

Walsh işlevleri  $[0, 1]$  zaman aralığında tanımlanmış olup ilk 8 işlev  $\text{wal}(0, t), \dots, \text{wal}(7, t)$  örnek olarak Şekil 1'de görülmektedir. Doğal sıralı Walsh işlevleri aşağıda verilen Walsh dönüşüm dizeyince de ifade edilebilirler:

$$\mathbf{W}(n) = \bigotimes_{i=1}^n \mathbf{W}(1), \quad \mathbf{W}(1) = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}.$$

Burada  $\bigotimes$  Kronecker çarpımını ifade etmektedir. Dizeyin her bir satırı ayrı bir Walsh işlevine aittir.  $\mathbf{W}(2)$  ve  $\mathbf{W}(3)$  dizeyleri



Şekil 1: İlk 8 Walsh işlevi

de örnek olarak aşağıdaki verilmiştir:

$$\mathbf{W}(2) = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix} \begin{matrix} 1 \\ x_1 \\ x_2 \\ x_1 \oplus x_2 \end{matrix}$$

$$\mathbf{W}(3) = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \\ 1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 \\ 1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 \\ 1 & -1 & -1 & 1 & -1 & 1 & 1 & -1 \end{bmatrix} \begin{matrix} 1 \\ x_1 \\ x_2 \\ \oplus x_{1,2} \\ x_3 \\ \oplus x_{1,3} \\ \oplus x_{2,3} \\ \oplus x_{1,2,3} \end{matrix}$$

Dizeyin her bir satırının denk geldiği Walsh işlevi/bileşeni dizeyin yanında belirtilmiştir. Burada  $\oplus x_{1,2}$  ifadesi  $x_1 \oplus x_2$  ile aynıdır. Walsh dizeyi görüldüğü gibi elemanları sadece +1 ve -1'den oluşan bir kare dizeydir ve satırlar karşılıklı dikgendir. Öyle ki,  $I^n$  bir  $n \times n$  birim dizey olmak üzere  $\mathbf{W}(n) \times \mathbf{W}(n)^T = n\mathbf{I}^n$  denilebilir. Walsh dönüşümü ve bunun ters dönüşümü aşağıdaki gibi tanımlanırlar:

$$S_f(\omega) = 2^{-n} \sum_{x=0}^{2^n-1} (1 - 2f(x)) wal(\omega, x), \quad (4)$$

$$f(x) = \frac{1}{2} - \frac{1}{2} \sum_{\omega=0}^{2^n-1} S_f(\omega) wal(\omega, x). \quad (5)$$

Dizey simgelemindeyse dönüşümler şu şekilde ifade edilebilirler:

$$\mathbf{S}_f = 2^{-n} \mathbf{W}(n)(\mathbf{1} - 2\mathbf{F}) \quad (6)$$

$$\mathbf{F} = \frac{1}{2}(\mathbf{1} - \mathbf{W}(n)\mathbf{S}_f) \quad (7)$$

Formül (6)'da görülen  $\mathbf{1} - 2\mathbf{F}$  ifadesi dönüşümden önce Boole değişkenlerinin aldığı  $\{0, 1\}$  değerlerini  $\{1, -1\}$  değerlerine dönüştürmeye yarar. Benzer şekilde ters dönüşümde de  $\mathbf{W}(n)\mathbf{S}_f$ ,  $\{1, -1\}$  değerini üretir ve bu değer  $\{0, 1\}$ 'e dönüştürülür. Herhangi bir  $f(x)$  işlevinin Walsh açılımı aşağıdaki şekilde yazılır:

$$f(x) = S_f(0) + S_f(1)x_1 + S_f(2)x_2 + S_f(3)(x_1 \oplus x_2) + S_f(4)x_3 + \dots + S_f(2^n - 1)(x_1 \oplus \dots \oplus x_n) \quad (8)$$

Walsh katsayıları sinyal çıkışıyla ilgili bileşene ait Walsh işlevi arasındaki ilintiyi gösterirler. Örneğin,  $k \in \{1, 2, \dots, n\}$  olmak üzere  $wal(2^{k-1}, t)$  devrenin  $x_k$  girişine karşılık gelen bileşendir ve  $S_f(2^{k-1})$  sinyal ile, diğer bir deyişle Boole işlevinin doğruluk yöneyi ile bu bileşen arasındaki ilintidir. Örnek devre olarak, ISCAS'85 [6] karşılaştırma devreleri havuzundan C-17 devresinin şeması Şekil 2'de verilmiştir. C-17, 5 girişli ve iki çıkışlı bir devre olup 7 adet NAND kapısından oluşmaktadır.  $y_1$  çıkışının doğruluk yöneyi aşağıdaki gibi hesaplanabilir:

$$\mathbf{F}_1 = [0, 0, 1, 1, 0, 1, 1, 1, 0, 0, 1, 1, 0, 1, 0, 1, 0, 0, 1, 1, 0, 1, 0, 1, 0, 0, 1, 1, 0, 1, 1, 0, 0, 1, 1, 0, 1, 0, 1]^T$$

Formül (6)'da verilen dönüşüm hesaplaması yapıldığıdaysa aşağıdaki spektral katsayılar elde edilir:

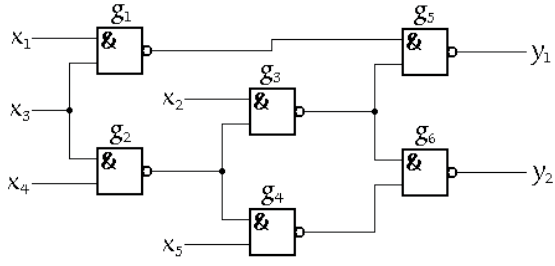
$$\mathbf{S}_{f_1} = [-0.125, 0.375, 0.625, 0.125, 0.125, -0.375, 0.375, -0.125, -0.125, -0.125, 0.125, 0.125, 0.125, 0.125, -0.125, -0.125, -0.125, 0, \dots, 0]^T$$

Bu sonuçlar hakkında ilk olarak şu yorumları yapmak olasıdır: 1)  $S_f(16)$  ve takip eden katsayıların sıfır olması işlevin  $x_5$ 'le ilintisi olmadığını söyler ki bu durum devre şemasından kolaylıkla doğrulanabilir, 2)  $S_f(2) = 0.625$  değeri  $y_1$ 'in en fazla  $x_2$  ile ilintisi olduğunu söyler. Görüldüğü üzere bir işlevin bazı bileşenlerle ilintisi daha fazla olabilir. Walsh açılımı sadece ilintisi belirli bir eşik değerinin üzerinde olan bileşenler cinsinden ifade edilip geri kalan bileşenler ihmal edildiğinde bir yaklaşık işlev elde edilebilir. Eşik değer için Spektral Güç Yoğunluğu (SGY) kullanılabilir. Spektral güç yoğunluğu spektral katsayıların karelerinin beklenen değeridir ve aşağıdaki formülle hesaplanır:

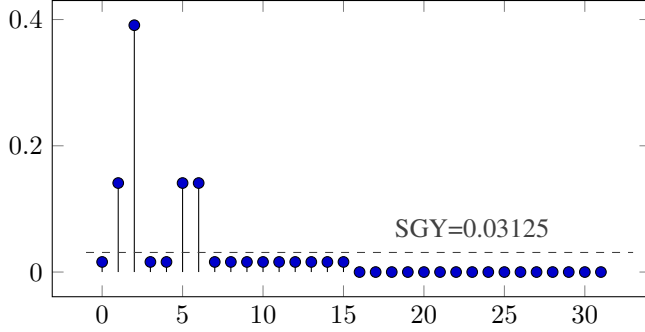
$$SGY = \frac{1}{2^n} \sum_{i=0}^{2^n-1} |S_f(i)|^2. \quad (9)$$

Şekil 3 C-17'nin  $y_1$  çıkışına ait güç spektrumunu göstermektedir. Parseval teoremi [5] bize bir Boole işlevinin spektral katsayıları toplamının 1'e eşit olduğunu söylemektedir. Bu durumda C-17 için  $SGY = 1/2^5 = 0.03125$  olarak hesaplanır. Buna göre  $y_1$ 'in katsayıları  $S_{f_1}(1), S_{f_1}(2), S_{f_1}(5)$  ve  $S_{f_1}(6)$  olan dört Walsh bileşeni, temel bileşenlerdir ve yaklaşık açılım aşağıdaki gibi yazılabilir:

$$y_1 \approx 0.375x_1 + 0.625x_2 - 0.375(x_1 \oplus x_3) + 0.375(x_2 \oplus x_3). \quad (10)$$



Şekil 2: C-17 karşılaştırma devresi



Şekil 3:  $y_1$ 'in Walsh güç spektrumu

### III. TEST GİRİŞLERİNİN BULUNMASI

Mantık devrelerinin test girişlerini belirleyebilmek için öncelikle öngörülen hata modeline karar vermek gerekir. Devre testlerinde en yaygın kullanılan hata modeli takılma (stuck-at) hata modelidir. Bu modele göre devredeki herhangi bir nokta, kalıcı olarak 0'a veya 1'e takılmış olabilir. Bir devrede kaç farklı takılma hatası olduğunu bulabilmek için tüm kapıların giriş çıkış sayılarını toplayıp ikiyle çarpmak yeterlidir. Örneğin, C-17 karşılaştırma devresinin 17 noktası olduğuna göre, devrede toplam 34 farklı hata oluşabilir. Ancak devre şemasından da açıkça görüleceği üzere, bir çok nokta birbirine bağlı olduğundan ( $g_2$ 'nin çıkışının  $g_4$  ve  $g_5$ 'in birer girişine bağlı olması gibi) eşlenik hatalar baştan gözardı edilebilir. Böylelikle C-17'nin toplam 11 farklı noktası olup 22 farklı takılma hatasına sahip olacağı görülebilir. Hatalı bir devrenin Boole işlevini  $g^{(i)}$  ile gösterirsek ve devrenin  $k$  farklı hataya sahip olabileceğini varsayarsak, her bir hatalı durum için doğruluk yöneylerini aşağıdaki gibi hesaplayabiliriz:

$$\mathbf{G}^{(i)} = [g^{(i)}(0), g^{(i)}(1), \dots, g^{(i)}(2^n - 1)]^T, \quad \forall i \in \{1, \dots, k\}.$$

#### A. Test İşlevleri

$n$  giriş,  $m$  çıkış ve  $k$  farklı potansiyel hataya sahip bir devre için aşağıda görülen  $k$  farklı test işlevi oluşturalım:

$$t_i(x) = (f_1(x) \oplus g_1^{(i)}(x)) + \dots + (f_m(x) \oplus g_m^{(i)}(x)), \quad \forall i = \{1, \dots, k\}. \quad (11)$$

İşlevden anlaşılacağı üzere,  $t_i(x)$  sadece ve sadece orijinal devre ile hatalı devrenin çıkışlarından en az birinin farklı olması durumunda 1 değeri üretecektir. Dolayısıyla, hata  $i$ 'nin belirlenebilmesi için test işlevinin 1 değeri üreteceği giriş

dizilimleri bizim için test girişleridir. Devrenin testi için gerekli olacak test girişleri,  $x \in \{0, 1\}^n$  olmak üzere,

$$t_i(x) = 1, \quad \forall i = \{1, 2, \dots, k\}$$

koşulunu sağlayan girişler test girişleri kümesini oluşturur.

Test girişlerinin belirlenebilmesi için test işlevlerinin spektrumunu incelemek ve yaklaşık işlevleri bulmak bu çalışmada önerilen metodun özünü oluşturmaktadır. Bir test işlevinin Walsh katsayıları aşağıdaki formülle hesaplanabilir:

$$\mathbf{S}_{T_i} = 2^{-n} \mathbf{W}(n) \left( 1 - 2 \sum_{j=1}^m (\mathbf{F}_j \oplus \mathbf{G}_j^{(i)}) \right). \quad (12)$$

C-17'nin 22 farklı hata durumu için test işlevleri bulunup bunların spektral katsayıları hesaplanır ve temel bileşenleri dikkate alınarak yaklaşık işlevler bulunur. Aşağıda örnek olarak  $t_1, t_2, t_3$  ve  $t_4$  işlevleri görülmektedir.  $t_1, g_3$ 'ün çıkışının 1'e takılma hatasına;  $t_2, y_1$ 'in 0'a takılma hatasına;  $t_3, x_2$ 'nin 0'a takılma hatasına;  $t_4$  ise  $x_1$ 'in 1'e takılma hatasına karşılık gelen test işlevleridir.

$$t_1 = 1.000 \quad (13)$$

$$t_2 = 0.375x_1 + 0.625x_2 - 0.375(x_1 \oplus x_3) + 0.375(x_2 \oplus x_3) \quad (14)$$

$$t_3 = 0.375 + 0.625x_2 - 0.375x_3 + 0.375(x_2 \oplus x_3) \quad (15)$$

$$t_4 = 0.375 - 0.625x_2 - 0.375x_3 - 0.375(x_2 \oplus x_3) \quad (16)$$

Dikkat edilmesi gereken bir nokta; yaklaşık işlevlerin üreteceği değerler gerçel bir sayı olduğudur, diğer bir deyişle  $t_i : \{1, -1\}^n \rightarrow \mathbb{R}$ . Bu durumda test girişleri kümesi aşağıdaki gibi tanımlanabilir:

$$T_i = \{x : t_i(x) \leq 0\} \quad (17)$$

Görüldüğü gibi,  $t_1$  sadece DC bileşene sahip olup hiç bir girişe bağlı değildir ve sıfırdan küçük bir değer üretmez. Bu sonuç bize bu hatayı ortaya çıkartabilmek için herhangi bir test girişi bulunamayacağını söyler, diğer bir deyişle devre bu hataya karşı dayanıklıdır.  $t_2$  içinse aşağıdaki eşitsizliği yazabiliriz:

$$0.375x_1 + 0.625x_2 - 0.375x_1x_3 + 0.375x_2x_3 < 0$$

Burada Boole değişkenlerinin alacağı değerlerin  $\{1, -1\}$  olarak kabul edilmesi durumunda  $xy \equiv x \oplus y$  olduğuna dikkat edilmelidir. Eşitsizliği sağlayan  $x$  girişleri test girişlerini oluşturur. Örneğin,  $x_1 = x_2 = x_3 = -1$  için eşitsizliğin sağlandığı görülmektedir. Diğer bir çözüm de  $x_1 = x_3 = 1, x_2 = -1$  değeridir. Burada,  $-1 \rightarrow 0.5 - 0.5(-1) = 1$  ve  $+1 \rightarrow 0.5 - 0.5(+1) = 0$  olduğuna dikkat edilmelidir. Dolayısıyla, test girişleri  $(x_1, x_2, x_3, x_4) = (0, 1, 0, X)$  ve  $(x_1, x_2, x_3, x_4) = (1, 1, 1, X)$  olarak bulunur. Burada  $X$  önemsenmez giriş anlamına gelmektedir.  $t_3$  için eşitsizlik

$$0.375 + 0.625x_2 - 0.375x_3 + 0.375x_2x_3 < 0$$

olup, test girişleri  $(x_1, x_2, x_3, x_4, x_5) = (X, 1, 0, X, X)$  olarak hesaplanır. Son olarak,  $t_4$  için

$$0.375 - 0.625x_2 - 0.375x_3 - 0.375x_2x_3 < 0$$

dir ve test girişleri  $(x_1, x_2, x_3, x_4, x_5) = (X, 0, 0, X, X)$  olarak bulunur.

#### IV. TEST ÖRÜNTÜSÜNÜN KÜÇÜLTÜLMESİ

Devredeki tüm hatalar için test girişleri elde edildikten sonra devredeki hataları belirleyecek en küçük test giriş kümesini elde etmek gerekir. Eniyilemeyi yaparken genellikle iki yaklaşım vardır: 1) Devrede herhangi bir hata olup olmadığının bulunması 2) Hataların devredeki yerinin de bulunması. Eniyileme problemine yaklaşıma göre karar vermek gerekmektedir.

C-17'nin seçtiğimiz dört hatası için test girişi kümeleri bir önceki bölümde anlatıldığı şekilde aşağıdaki gibi elde edilir:

$$\begin{aligned} T_1 &= \emptyset \\ T_2 &= \{“0100”, “0101”, “1110”, “1111”\} = \{4, 5, 14, 15\} \\ T_3 &= \{“0100”, “0101”, “1100”, “1101”\} = \{4, 5, 12, 13\} \\ T_4 &= \{“0000”, “0001”, “1000”, “1001”\} = \{0, 1, 8, 9\} \end{aligned}$$

Görüldüğü gibi, üç kümenin içerisinde 10 farklı test girişi bulunmaktadır.  $\mathbf{u}^T = [u_0, u_1, u_4, u_5, u_8, u_9, u_{12}, u_{13}, u_{14}, u_{15}]$  her bir test girişine karşılık gelen karar değişkenlerimiz olsun, öyle ki  $u_i = 1$  olması test  $i$ 'nin kabul edilmesi anlamına gelsin. Her kümeyi kapsayacak şekilde en az test girişini bulma problemi eniyileme alanında çok bilinen en küçük küme kapsama problemi ile aynıdır. 1 numaralı yaklaşım için eniyileme aşağıdaki şekilde yazılır:

$$\min \mathbf{1}^T \mathbf{u} \quad (18)$$

$$\text{s.t. } \mathbf{A} \mathbf{u} \geq \mathbf{1} \quad (19)$$

$$\text{s.t. } \mathbf{u} \in \{0, 1\}^k. \quad (20)$$

$\mathbf{1} = [1, 1, \dots, 1]^T$  ve  $\mathbf{A}$  diziyi de:

$$\mathbf{A} = \begin{bmatrix} 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \end{bmatrix}$$

olarak yazılabilir. Bu problemin çözümlerinden biri diğer değişkenler sıfır olacak şekilde  $u_0 = u_4 = 1$ 'dir. Diğer bir deyişle test girişleri kümemiz:

$$T = \{“0000”, “0100”\}$$

olarak bulunur. Devrenin hatasının bulunmasının yanısıra hatanın oluştuğu yer ve hata tipinin de bulunması yaklaşımında ise her bir hata için öyle bir test girişi bulunmalıdır ki bu test girişi diğer hatalarda kullanılsın. Bu problemin çözümü aşağıdaki doğrusal denklem sisteminin çözümüdür:

$$\mathbf{A} \mathbf{u} = \mathbf{1}. \quad (21)$$

Bu durumda problemimizin bir çözümü  $u_1 = u_{13} = u_{14} = 1$ 'dir. Test kümesiye:

$$T = \{“0000”, “1101”, “1111”\}.$$

olarak bulunur. Bazı durumlarda Denklem (21)'in çözümü olmayabilir, bu durumda bazı hataların tam yer ve tipi bulunamayabilir ancak yine de olası yer ve tip kümesi olabildiğince daraltılabilir. Bu durumda Denklem (21)'i bir eniyileme problemi olarak yazmak daha doğru olacaktır:

$$\min \mathbf{1}^T \mathbf{v} \quad (22)$$

$$\text{s.t. } \mathbf{A} \mathbf{u} - \mathbf{v} = \mathbf{1} \quad (23)$$

$$\text{s.t. } \mathbf{u} \in \{0, 1\}^k, \mathbf{v} \in \mathbb{N}^l. \quad (24)$$

#### V. SONUÇ

Bu çalışmada birleşimsel mantık devreleri için Fourier analizine dayanan bir otomatik test örüntüsü oluşturma yöntemi sunulmuştur.  $n$  girişli bir devre için Fourier analizinin zaman ve kaynak kullanımı karmaşıklığı  $O(n2^n)$  ve  $O(2^n)$  olarak ifade edilir [7]. Bu nedenle yöntemin darboğazının tüm katsayıların hesaplanmasındaki zorluk olduğu görülmektedir. Daha hızlı çevrim için ikili karar verme diyagramlarından yararlanılan ve sadece istenen katsayılar için kısmi hesaplama yapan yöntemler önerilmiştir [8] [9]. Bu yöntemlerin de eklenmesiyle metodun başarısının mevcut yöntemlerle kıyaslamalı olarak ITC'99 [10] vb. test devreleriyle sınanması gerekmektedir.

#### KAYNAKÇA

- [1] J. A. Abraham and W. K. Fuchs, “Fault and error models for VLSI,” in *Proceedings of the IEEE*, vol. 75. IEEE, 1986, pp. 639–654.
- [2] V. P. Shmerko, S. N. Yanushkevich, and V. G. Levashenko, “Test pattern generation for combinatorial multi-valued networks based on generalized d-algorithm.” in *ISMVL*, 1997, pp. 139–144.
- [3] S. Srinivasan, G. Swaminathan, J. H. Aylor, and M. R. Mercer, “Combinational circuit atpg using binary decision diagrams.” in *VTS*. IEEE, 1993, pp. 251–258.
- [4] W. Changqian and W. Chenghua, “A Method for Logic Circuit Test Generation Based on Boolean Partial Derivative and BDD,” *2009 WRI World Congress on Computer Science and Information Engineering*, pp. 499–504, 2009. [Online]. Available: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5170892>
- [5] R. O'Donnell, “Some topics in analysis of boolean functions,” in *Proceedings of the Fortieth Annual ACM Symposium on Theory of Computing*, ser. STOC '08. New York, NY, USA: ACM, 2008, pp. 569–578. [Online]. Available: <http://doi.acm.org/10.1145/1374376.1374458>
- [6] M. C. Hansen, H. Yalcin, and J. P. Hayes, “Unveiling the iscas-85 benchmarks: A case study in reverse engineering,” *IEEE Des. Test*, vol. 16, no. 3, pp. 72–80, Jul. 1999. [Online]. Available: <http://dx.doi.org/10.1109/54.785838>
- [7] P. Porwik, “Efficient spectral method of identification of linear boolean function,” *Control and Cybernetics*, vol. 33, no. 4, pp. 663–678, 2004.
- [8] D. Jankovic, R. Stankovic, and R. Drechsler, “Decision diagram method for calculation of pruned Walsh transform,” *IEEE Transactions on Computers*, vol. 50, no. 2, pp. 147–157, 2001. [Online]. Available: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=908990>
- [9] P. Porwik and E. Gramatov, “Conditions of the affine extension of an incompletely defined boolean function,” vol. 29, pp. 1073–1088, 2010.
- [10] F. Corno, M. Reorda, and G. Squillero, “Rt-level itc'99 benchmarks and first atpg results,” *Design Test of Computers, IEEE*, vol. 17, no. 3, pp. 44–53, Jul 2000.