

**T.C.
GEBZE TEKNİK ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ**

**GECİKME TEMELLİ YAN-KANAL ANALİZİ
YÖNTEMLERİ KULLANARAK DONANIMSAL
TRUVA ATI TESPİTİ**

**FATMA NUR ESİRCİ
YÜKSEK LİSANS TEZİ
BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI**

**GEBZE
2018**

T.C.
GEBZE TEKNİK ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

GECİKME TEMELLİ YAN-KANAL
ANALİZİ YÖNTEMLERİ KULLANARAK
DONANIMSAL TRUVA ATI TESPİTİ

FATMA NUR ESİRCİ
YÜKSEK LİSANS TEZİ
BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI

DANIŞMANI
YRD. DOÇ. DR. ALP ARSLAN BAYRAKÇI

GEBZE
2018

T.R.
GEBZE TECHNICAL UNIVERSITY
GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES

**HARDWARE TROJAN DETECTION USING
DELAY BASED SIDE-CHANNEL ANALYSIS
METHODS**

FATMA NUR ESİRCİ

**A THESIS SUBMITTED FOR THE DEGREE OF
MASTER OF SCIENCE
DEPARTMENT OF COMPUTER ENGINEERING**

**THESIS SUPERVISOR
ASST. PROF. ALP ARSLAN BAYRAKÇI**

**GEBZE
2018**

GTÜ Fen Bilimleri Enstitüsü Yönetim Kurulu'nun 10/01/2018 tarih ve 2018/03 sayılı kararıyla oluşturulan jüri tarafından 05/02/2018 tarihinde tez savunma sınavı yapılan Fatma Nur Esirci'nin tez çalışması Bilgisayar Mühendisliği Anabilim Dalında YÜKSEK LİSANS tezi olarak kabul edilmiştir.

JÜRİ

ÜYE

(TEZ DANIŞMANI)

: Yrd. Doç. Dr. Alp Arslan BAYRAKÇI



ÜYE

: Yrd. Doç. Dr. Mustafa ALTUN



ÜYE

: Yrd. Doç. Dr. Önder ŞUVAK



ONAY

Gebze Teknik Üniversitesi Fen Bilimleri Enstitüsü Yönetim Kurulu'nun

...../...../..... tarih ve/..... sayılı kararı.

ÖZET

Yan kanal analizi temelli donanımsal truva atı tespit modellerinin tamamı varyasyon değışikliklerinden derinden etkilenmektedir. Varyasyonun yol açtığı negatif etkilerden kurtulmak için, “Konuma Bağlı Varyasyon Modeline rağmen İlişkili Yol Gecikmesi Tabanlı DTA Tespiti” yöntemi geliştirilmiştir. Bu yöntem, devre üzerinde donanımsal truva atı içerdiğinden şüphelenilen her bir kenar için birbiriyle yüksek korelasyona sahip iki yolun akıllıca seçilmesidir. Bahsedilen yollardan ilki, şüpheli kenarı içinde barındıran en kısa yolu ifade etmektedir. İkincisi, ilk yol ile oldukça ilişkili (yüksek korelasyona sahip) yol olarak tanımlanmıştır. Bu iki yolun gecikme değerlerinin oranlanmasıyla, üzerine donanımsal truva atı eklenmiş devrelerin tespit edilmesi sağlanmaktadır. Test sonuçları yöntemimizde kullanılan konuma bağlı varyasyon modeli nedeniyle hem çip içi hem de çipler arası varyasyonlara maruz kalmaktadır. Varyasyon etkisine rağmen tespiti zor olan küçük donanımsal truva atı kullanıldığında bile, donanımsal truva atının neredeyse hatasız tespit edilmesini sağlamaktadır.

Anahtar Kelimeler: donanım güvenliği, yol gecikmesi, kötü amaçlı devre

SUMMARY

Hardware Trojan detection methods based on the side channel analysis deeply suffer from the process variations. In order to suppress the effect of the variations, we devise a method that smartly selects two highly correlated paths for each interconnect (edge) that is suspected to have an hardware trojan on it. First path is the shortest one passing through the suspected edge and the second one is a path that is highly correlated with the first one. Delay ratio of these paths avails the detection of the hardware trojan inserted circuits. Test results reveal that the method enables the detection of even the minimally invasive trojans in spite of both inter and intra die variations with the spatial correlations.



Key Words: hardware security, path delay, malicious circuit.

TEŐEKKÜR

Yüksek lisans eğitimim süresince tez konuma yönelik kullanılacak yöntemlerin belirlenmesinde, yeni yöntem geliştirilmesinde ve tez çalışmalarına uygulanmasında bana sağladığı imkânlardan, gösterdiği özenden, duyduğu güvenden ve sonuç odaklı çalışmaya yönelik kazandırdığı özelliklerden dolayı minnettar olduğum danışmanım Yrd. Doç. Dr. Alp Arslan Bayrakçi 'ya teşekkürlerimi sunarım.

Eğitim hayatım süresince bana her zaman güvenen, ellerinden gelen tüm imkânları sağlayan ve toplumun kadına olan bakış açısını değil, birey olmayı temel esas belirleyen aileme teşekkür etmeyi borç bilirim.

Hem yüksek lisans eğitimimi hem de tez çalışmamın parçası olan konferans katılımımı destekleyen TÜBİTAK'a teşekkür ederim.

İÇİNDEKİLER

	<u>Sayfa</u>
ÖZET	v
SUMMARY	vi
TEŞEKKÜR	vii
İÇİNDEKİLER	viii
SİMGELER ve KISALTMALAR DİZİNİ	x
ŞEKİLLER DİZİNİ	xi
TABLOLAR DİZİNİ	xiii
1. GİRİŞ	1
1.1. DTA Problemi	1
1.2. Problemin Zorluğu	2
1.3. Tezin Amacı	3
2. LİTERATÜR ÇALIŞMALARI	5
2.1. DTA Sınıflandırması	5
2.2. DTA Tespit Yöntemleri	8
3. METOT	13
3.1. Varyasyon Modeli	13
3.2. Problem Formülasyonları	16
3.2.1. DTA Modeli ve Devreye Etkisi	18
3.2.2. Yol Gecikmesine Bakarak DTA Tespiti Zorluğu	19
3.3. Konuma Bağlı Varyasyon Modeline rağmen İlişkili Yol Gecikmesi Tabanlı DTA Tespiti	21
3.3.1. Düğüm Tahmini Gecikme Hesabı	22
3.3.2. Şüpheli Yol Seçimi	23
3.3.3. İlişkili Yol Seçimi	26
3.3.4. Yol Sentezleme	29
3.3.5. DTA Tespiti	33
4. SONUÇLAR	37
4.1. Deneysel Kurulumlar	37
4.2. Testler	38
4.3. Gelecek Çalışmalar için Öneriler	51

KAYNAKLAR

52

ÖZGEÇMİŞ

55



SİMGELER ve KISALTMALAR DİZİNİ

<u>Simgeler ve</u>	<u>Açıklamalar</u>
<u>Kısaltmalar</u>	
BNF	: Bağlayıcı Normal Formülasyonu
DAC	: Design Automation Conference
DATE	: Design & Automation Test in Europe
DTA	: Donanımsal Truva Atı
ISCAS	: International Symposium on Circuits and Systems
ISQED	: International Symposium on Quality Electronic Design
İZA	: İstatiksel Zamanlama Analizi
L	: Kapı Genişliği
OSÖÜ	: Otomatik Sınama Örüntüsü Üretimi
SAT	: Boolean Satisfiability Problem
SLE	: Stokastik Lojik Efort
TD	: Tümlşik Devre
TÜBİTAK	: Türkiye Bilimsel ve Teknolojik Araştırma Kurumu
V_{th}	: Eşik Gerilimi
YKA	: Yan Kanal Analizi

ŞEKİLLER DİZİNİ

<u>Sekil No:</u>	<u>Sayfa</u>
2.1: Tümlleşik Devrelerin Yaşam Döngüsü.	5
2.2: DTA Sınıflandırılması.	7
2.3: DTA Tespit Yöntemlerinin Sınıflandırılması.	9
3.1: Dördün Ağaç Yöntemi ile Konuma Bağlı Korelasyon Modeli.	14
3.2: Devrenin Lojik Kapılarla Gösterimi.	16
3.3: Devrenin Tek Yönlü Çevrimsiz Çizge Formatında Gösterimi.	17
3.4: Yayılım Gecikmesi Hesabı.	17
3.5: DTA Yapısı.	18
3.6: Örnek Çiplerin Yol Gecikme Değerleri.	20
3.7: Şüpheli Yol Seçme Algoritması.	24
3.8: İlişkili Yol Adayları Kümesi Örneği.	25
3.9: İlişkili Yol Seçme Algoritması.	26
3.10: Yol Bulma Algoritması.	26
3.11: Devrenin Konuma Bağlı Çizge Formatında Gösterimi.	28
3.12: Kontrol Eden ve Kontrol Etmeyen Girdi Değerleri Örneği.	30
3.13: Statik Yol Sentezleme Örneği Gösterimi.	31
3.14: Yol Sentezleme Algoritması Yaşam Döngüsü.	31
3.15: DTA Tespitinde Amaçlanan Sonucun Gösterimi.	33
3.16: İkili Yol Seçme Algoritması.	34
3.17: DTA Tespiti Algoritması.	35
3.18: DTA İçeren ve İçermeyen Gecikme Oranlarının Dağılımı.	36
4.1: c7552 SE1 – DTA Sınıflandırması Dağılımları.	42
4.2: c7552 SE2 – DTA Sınıflandırması Dağılımları.	42
4.3: c7552 SE3 – DTA Sınıflandırması Dağılımları.	42
4.4: c5315 SE1 – DTA Sınıflandırması Dağılımları.	43
4.5: c5315 SE2 – DTA Sınıflandırması Dağılımları.	43
4.6: c5315 SE3 – DTA Sınıflandırması Dağılımları.	43
4.7: c3540 SE1 – DTA Sınıflandırması Dağılımları.	44
4.8: c3540 SE2 – DTA Sınıflandırması Dağılımları.	44
4.9: c3540 SE3 – DTA Sınıflandırması Dağılımları.	44
4.10: c2670 SE1 – DTA Sınıflandırması Dağılımları.	45

4.11: c2670 SE2 – DTA Sınıflandırması Dağılımları.	45
4.12: c2670 SE3 – DTA Sınıflandırması Dağılımları.	45
4.13: c1908 SE1 – DTA Sınıflandırması Dağılımları.	46
4.14: c1908 SE2 – DTA Sınıflandırması Dağılımları.	46
4.15: c1908 SE3 – DTA Sınıflandırması Dağılımları.	46
4.16: c1355 SE1 – DTA Sınıflandırması Dağılımları.	47
4.17: c1355 SE2 – DTA Sınıflandırması Dağılımları.	47
4.18: c1355 SE3 – DTA Sınıflandırması Dağılımları.	47
4.19: c880 SE1 – DTA Sınıflandırması Dağılımları.	48
4.20: c880 SE2 – DTA Sınıflandırması Dağılımları.	48
4.21: c880 SE3 – DTA Sınıflandırması Dağılımları.	48
4.22: c499 SE1 – DTA Sınıflandırması Dağılımları.	49
4.23: c499 SE2 – DTA Sınıflandırması Dağılımları.	49
4.24: c499 SE3 – DTA Sınıflandırması Dağılımları.	49
4.25: c432 SE1 – DTA Sınıflandırması Dağılımları.	50
4.26: c432 SE2 – DTA Sınıflandırması Dağılımları.	50
4.27: c432 SE3 – DTA Sınıflandırması Dağılımları.	50

TABLÖLAR DİZİNİ

<u>Tablo No:</u>	<u>Sayfa</u>
3.1: Kapıların BNF İfadeleri	32
4.1: Test Sonuçlarının Detaylı Gösterimi.	39



1. GİRİŞ

1.1. DTA Problemi

Devreye bilgimiz ve isteğimiz dışında eklenen kötü niyetli yapılara donanımsal truva atı (DTA) (ing. hardware trojan) denir. Bu yapı, devrenin güvenlik mekanizmalarını devre dışı bırakabilir, devrenin fonksiyonel yapısını bozabilir. Böylece kritik verilerin işlenmesinde, iletiminde veya saklanmasında kullanılmak için tasarlanan bir ürünün, verileri istenmeyen kişilere sızdırmasına neden olabilir [7]. Devreyi savunmaz hale getirir.

Donanım güvenilirliği kavramının popülerleşmesinin nedeni askeri ve finansal sistemlerde kullanılan tümleşik devrelerin, güvenilir olmalarının gerekliliği ve bu alanlarda donanımsal truva atlarının potansiyel varlıklarıdır. Kritik sistemler güvenilir fabrikalarda tasarlanmış ve üretilmiş elektronik cihazları kullanmaktadırlar. Bu fabrikalardan elde edilen sistemler güvenilir olmaktadır. Fakat kritik sistemlerin her biri öncelikle devletlerin ulusal güvenlik problemidir [7,8,9,10].

Truva atı problemi, dünya genelinde 2010 yılından itibaren ayrı bir başlık altında incelenen ve gittikçe önem kazanan bir konu olmuştur. 2015 ocak ayında IEEE Spectrum bu konuya “Parçalardaki Donanımsal Truva Atlarını Durdurma” başlığında yer vermiştir [6]. ISQED 2015 programına “Donanım ve Sistem Güvenliği” başlığı eklenmiştir. Dünyada kabul görmüş DAC, DATE gibi büyük konferanslar tarafından DTA konusunda ayrı başlıklar açılmıştır. Örneğin, DAC 2014 programında Güvenlik başlığı altında donanım güvenliği ayrı başlık olarak incelenmeye başlamıştır [32].

DTA konusuna ait ilk uygulamalar ABD-Rusya arası soğuk savaş zamanına dayanmaktadır. Bu dönemde Rusya ABD’ne gidecek olan daktiloları ele geçirip, üzerlerine anahtar kaydedici yapılar eklemiştir. Avrupalılar ise işlemcilere, Suriye radarlarının İsrail tarafından algılanmasını önleyici yapılar eklemiştir [1].

Donanım bazında tespit edilen ilk arka kapı, 2012 yılında Actel FPGA üzerinde bulunmuştur [4]. Yakın zamanlarda Avusturya güvenlik kurumu SEC Consult, Barracuda firmasına ait birçok üründe raporlanmamış arka kapılar

bulduğunu açıklamıştır. Firma, bunların üretim aşamasındaki deneysel çalışmalarda kullanıldığını ve sonradan unutulduğunu dile getirmiştir. Bu şekilde dolaylıda olsa arka kapıların varlığını kabul etmiştir [2].

Bir diğer örnekte ise Amerika Ulusal Güvenlik Ajansı(NSA) tarafından, piyasada yaygın olarak kullanılan Huawei ve ZTE haberleşme cihazlarının DTA içermesine yönelik bir araştırma raporu yayınlanmış ve bu cihazların kritik projelerde kullanılması yasaklanmıştır [3].

Ülkemizde 2012 yılında Savunma Sanayi Müsteşarlığı ve TÜBİTAK [5] tarafından yayınlanan bildiri ve yayınlarda, donanım güvenliği ve DTA tespiti öncelikli alanlara dahil edilmiştir. Donanım güvenliği ve DTA tespiti üzerine yapılan çalışmalar sonucunda doğru yaklaşım bulunarak, saldırıların zararlı etkilerini ortadan kaldırmak mümkündür.

1.2. Problemin Zorluğu

Devrenin fabrikada üretimi sırasında engellenemeyen ve öngörülemeyen, aslında transistör üretiminin doğası gereği oluşan ve bu nedenle üretilen her bir devrenin birbirinden farklı voltaj, saat frekansı (ing.clock-freq) ve ya yol-gecikmesi (ing.path delay) değerlerine sahip olmasına neden olan etkiye varyasyon (ing. process variation) denir.

Gelişen teknolojiyle birlikte transistör boyutları gittikçe küçülmekte, bir çip içine sığabilen transistör sayısı artmaktadır. Tümüleşik devre tasarımı da gittikçe karmaşık bir yapıya ulaşmıştır. Donanımsal truva atı tespiti problemini zorlaştıran etmenlerden biri de budur. Transistör boyutlarının küçülmesiyle devre, varyasyon olarak tanımlanan, engellenemeyen üretim hatalarına daha duyarlı bir hal almıştır. Özellikle transistör boyutlarının küçülmesiyle, transistör parametreleri olan kapı genişliği (L) (ing. gate length), eşik gerilimi (V_{th}) (ing. threshold voltage) gibi önemli devre parametrelerinin varyasyonu artmaktadır [26]. Bu nedenle, hız/yol gecikmesi ve güç tüketimi gibi nihai devre performans göstergelerindeki varyasyon daha etkili hale gelmektedir.

Problemi zorlaştıran etmenlerden bir diğeri de devrelerin kapalı kutu şeklinde paketlenmiş olarak elimize geçmesidir. Bu yüzden devrenin iç katmanlarına erişilememektedir. Aslında devre çok sayıda katmandan oluşmaktadır ve her bir katman kendi içinde oldukça karmaşıktır. Üretilen devrenin tersine mühendislik (ing.

reverse engineering) ile şemasının çıkarılmasını, devrenin bu kapalı kutu hali oldukça zorlaştırmaktadır.

Küçük boyutlu, tespiti oldukça zor olan DTA'ların bile devreye etki etmesidir. DTA'lar devreye eklendikleri zaman pasif haldedirler. Pasif haldeyken, devrenin fonksiyonel yapısını bozamaz yani beklenen çıktı değerini değiştirmezler. DTA'yı devrede aktifleştirecek girdi değerleri, nadiren oluşacak şekilde tasarlanmıştır. DTA'lar aktif haldeyken, devrede beklenen çıktı değerini değiştirirler. Bu nedenle, DTA tespiti yapabilmek için, DTA'nın aktifleştirilmesi gerekmektedir. Oysa ki Yan kanal analiz (YKA) (ing. side-channel analysis) yöntemleri buna ihtiyaç duymazlar. Çünkü DTA'lar pasif haldeyken bile, çıktı değerini değiştirmemelerine rağmen, gecikme/hız ölçümü, güç tüketimi gibi değerlerini etkilerler.

YKA yöntemleri devrenin güç tüketimi, yol gecikmesi, sıcaklık değerleri gibi fiziksel parametreleri üzerinde ölçümler yaparlar. Yapılan ölçümleri altın modelden ya da DTA olmadığından emin olunan devre üzerinden elde edilen verilerle karşılaştırırlar. Çalışmamızda gecikme temelli yan kanal analiz yöntemlerinden yol gecikme metodunu kullanmaktayız.

YKA yöntemlerinin en büyük zafiyetlerinden biri de varyasyon etkisidir. Literatürde çokça kullanılan gecikme ve güç tüketimi ölçümlerine dayalı DTA tespitinin en zayıf noktası, varyasyonların etkisinin DTA etkisini saklamasıdır. Bu problemi ortadan kaldırmak için, varyasyonların etkisi ile DTA etkisini birbirinden ayıracak “Konuma Bağlı Varyasyon Modeline rağmen İlişkili Yol Gecikmesi Tabanlı DTA Tespiti” yöntemini geliştirdik.

1.3. Tezin Amacı

Tezin amacı, Gecikme temelli YKA yöntemleri kullanarak DTA tespit etmektir. Bu sebeple gecikme temelli YKA yöntemlerinin başarı olması için varyasyonların karıştırıcı etkisi giderilmelidir. Gecikme temelli yöntemler, devrenin gecikme değer ölçümlerini kullanırlar. Devrede oluşan değişiklikleri gecikme değer değişiklikleriyle tespit ederler. Gecikme temelli yöntemler, üretim hatalarından kaynaklanan varyasyon etkisi nedeniyle DTA etkisini tespit etmede sıkıntı çekmektedirler [8,14,15,17]. Bölüm 2'de bulunan literatür kısmında çalışmaların bir kısmı özetlenmiştir.

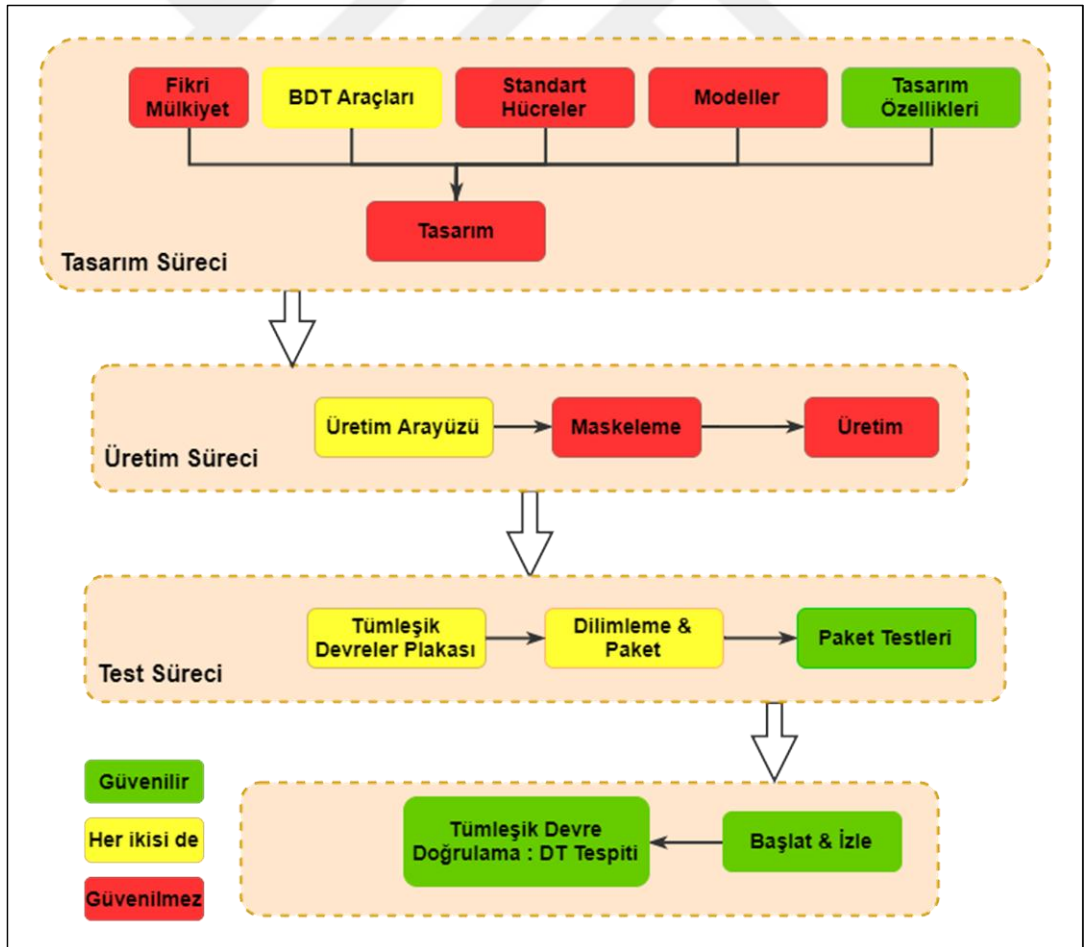
Bizim çalışmamızda devre üzerinde gecikme ölçümlerinde varyasyon etkisinin, DTA etkisinden ayrılması hedeflenmiştir. Varyasyon etkisini DTA etkisinden ayırmak için yapılan çalışmalar sonucunda, varyasyonların yapısal özelliklerinden ve konuma bağlı korelasyon (ing. spatial correlation) bilgilerinden faydalanılmasına karar verilmiştir. Bu yöntem oluşturulurken çoklu parametre kullanımı [14], en kısa yol üzerinde ölçümler yapma [17] ve stokastik kapı gecikme modeli [20] fikirleri bir araya getirilmiştir. Tezde bahsedilecek olan yöntem varyasyon etkisini DTA etkisinden başarıyla ayırabilmektedir. Temel fikir, devre içinde varyasyonlardan benzer şekilde etkilenen iki değer bulup, bu değerleri oranlayarak varyasyon etkisinden kurtulmaktır. Bu oranının devreden devreye çok değişmeyeceğine ama DTA eklendiğinde değişmesine dayanmaktadır. Anahtar nokta böyle iki değer bulunabilmesidir. Yöntem aynı zamanda, konuma bağlı korelasyon bilgisinden faydalanarak varyasyon etkisini DTA etkisinden ayıran literatürdeki ilk çalışma olmuştur.

Yöntemin varyasyon etkisini DTA etkisinden başarıyla ayırdığını göstermek için ISCAS'85 devreleri [27] üzerinde testler yapılmıştır. İlgili devreler üzerinde iki ayrı yöntem kullanılarak DTA tespiti yapılmıştır. Yöntemlerden ilki en kısa yol üzerinde gecikme ölçümleri yaparak DTA tespit eden [17], ikincisi ise bu tezde önerilen yöntemdir. Test devreleri üzerinde yapılan testler sonucunda, ilk yönetime göre toplamda 9000 çipten 2944 tanesi yani yaklaşık %33'ü yanlış sınıflandırılıyorken, bu tezde önerilen yöntemle bu sayı 17 taneye, yaklaşık % 0.2'ye düşürülmüştür.

2. LİTERATÜR ÇALIŞMALARI

2.1. DTA Sınıflandırması

Şekil 2.1’de gösterilen tümleşik devre (TD) (ing. integrated circuit) üretiminin hayat döngüsü, tasarım, üretim ve test süreci olmak üzere 3 kısımdan oluşmaktadır. Şekilde yeşil renkli kısımlar güvenilir, kırmızılar güvenilirmez, sarılar ise ne güvenilir ne güvenilirmez sayılan kısımlardır. Güvenilir olan kısımlar sadece paketlenme ve son kontrole ait test ve görüntüleme kısımlarıdır. Tasarım özellikleri de bizim tarafımızdan belirlendiği için güvenilir olarak seçilmiştir ama bazı anlatımlarda tasarım gereksinimleri kısmı da sarı renkle ifade edilmektedir [7]. Biz üretim sürecindeki problemlere odaklandık. Tasarım ve test süreçlerinin güvenilir hale gelmesi ayrıca incelenmesi gereken konulardır.



Şekil 2.1: Tümleşik Devrelerin Yaşam Döngüsü.

Mikroelektronik cihazların üretiminde eskiden, üretilebilirlik ve test edilebilirlik kavramları üzerine yoğunlaşıyordu. Şu an yoğunlaşmak zorunda olunan başlık güvenilirliktir. Mikroelektronik cihazların güvenilirliği, donanım güvenliği (ing. hardware trust) veya TD güvenirligi olarak analiz edilmektedir [10]. TD güvenirligini saglayabilmek yapılması gerekenler üç maddeyle özetlenmiştir. Maddelerin tamamının yapılabilmesi çok zor olduğundan, TD güvenliğini bu şekilde önlem olarak sağlamak mümkün olmamaktadır.

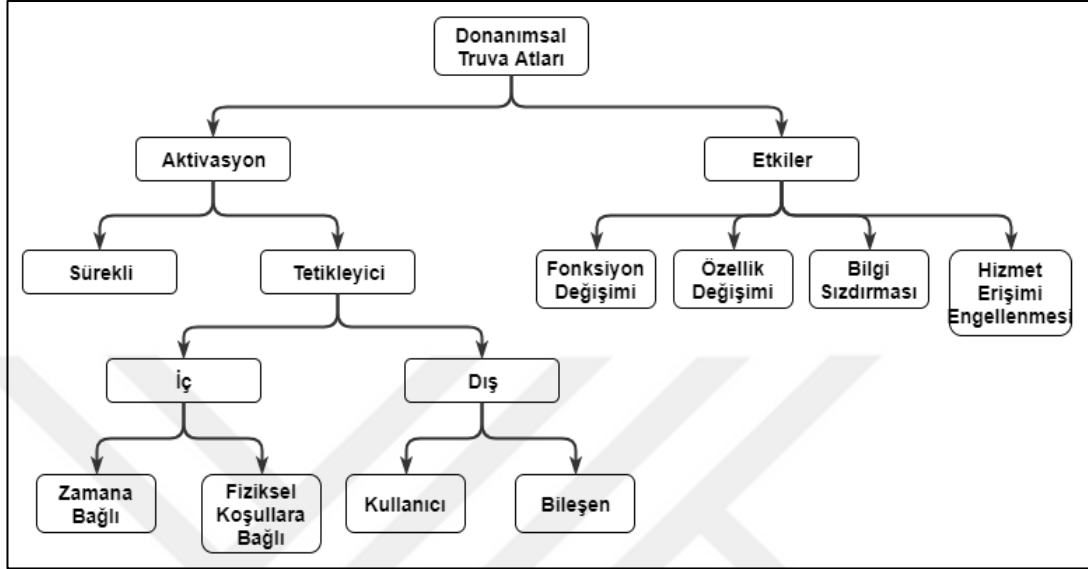
- Devre istenilen fonksiyonelliđi ne eksik ne de fazla, tam olarak sağlamalıdır.
- Devre, içinde yapılan işlemlerle ilgili bilgileri gizlemelidir. Bu bilgilere yan kanal verilerinden gecikme veya akım analizleriyle ulaşlamamalıdır.
- Sadece devre tasarımcılarına karşı şeffaf olunmalıdır. Takımın geri kalanları tasarım ve iç aşamalarla ilgili hiçbir şey bilmemelidirler.

TD güvenliğini sağlamak için öncelikle DTA'nın ne olduğu ve nasıl sınıflandırıldığıının tam olarak bilinmesi gerekmektedir. Sınıflandırmalar sayesinde benzer karakteristik yapılarıdaki DTA'lar aynı grupta yer alır. Böylece DTA tespit etmeye, önlemeye ya da korumaya karşı daha sağlıklı yöntemler geliştirilebilir. DTA sınıflandırması yapılırken dikkat edilen iki kavram [10] tüm DTA'ların sınıflandırılabilir olması (ing. coverage) ve aynı sınıf altındaki DTA'ların benzer yapıda olmasıdır (ing. resolution). Araştırmacılar tarafından kabul görmüş üç farklı sınıflandırma yöntemi vardır.

- Ortaya konan ilk sınıflandırma yapısıdır. DTA'ları fiziksel, aktivasyon ve fonksiyonel karakteristiklerine bakarak sınıflandırılır [8,9].
- Tetikleyici (ing. trigger) ve yük (ing. payload) karakteristiklerine bakarak sınıflandırmadır [7,11].
- Bu yeni sınıflandırmanın diğerlerinden farkı, sadece fabrikasyon aşamasında değil herhangi bir aşamada ve farklı fonksiyonellikleri etkilemesine göre DTA incelemesidir [10,36].

Sınıflandırma yöntemlerinin içinden en geniş kapsamlı olanı 3.yöntemdir ve Şekil 2.2 de bu sınıflandırma yapısının bizi ilgilendiren iki alt bölümü anlatılacaktır.

Sınıflandırmanın tamamı beş alt bölümden [10] oluşmaktadır fakat çalışmamızda kullanılan DTA'yı anlamlandırabilmek için sınıflandırma alt başlıklarından aktivasyon ve etkiler sınıflarının incelenmesi yeterli olacaktır.



Şekil 2.2: DTA Sınıflandırılması.

- i. Aktivasyon Mekanizması: DTA'ların hangi koşullarda aktifleşeceğine göre sınıflandırmaktadır. DTA'ların bir kısmı devrede sürekli aktif haldedir, bir kısmı da devrede belirli tetikleyici koşullar sağlandığında aktifleşirler. Aktifleşmek için belirli koşulları bekleyen DTA yapıları da kendi içinde ikiye ayrılırlar. Bunlar iç değişikliklere (ing. internal event) ve dış değişikliklere (ing. external event) göre incelenmektedirler. İç değişikliklere göre aktifleşen yapılardan zamana bağlı olanlara örnek olarak zaman bombası (ing. time bomb) verilebilir. Zaman bombasının tetikleyici yapısı belirli zaman değerine göre aktifleşmektedir. Fiziksel koşula bağlı olanlar devre iç koşullarındaki değişimlere göre aktifleşirler. Örnek olarak devrede yüksek sıcaklık ölçümleriyle aktifleşen DTA yapısı verilebilir. Dış değişikliklere göre aktifleşen yapılar, kullanıcı girdi değeriyle yapılan değişiklikler veya devredeki bileşenlere yapılan değişiklikler olmak üzere iki kısımda incelenirler.
- ii. Etki Mekanizması: Devreye eklenen DTA'nın oluşturduğu istenmeyen etkilere göre yapılan sınıflandırmadır. Bu etki grupları;

- Devrenin fonksiyonelliğinin değişmesi (ing. change function) : Devre çıktılarının beklenenden farklı olmasıdır. Devrenin fonksiyonel yapısı etkilendiği için yapması gereken işi gerçekleştirememiş ve beklenen çıktıyı üretememiştir.
- Devrenin Özelliklerinin Değişmesi (ing. change specification) : Devre hızının ve ya güç tüketimi değerlerinin olması gerekenden farklı çıkmasıdır. Bu etki de devrenin tasarlanırken sağlaması beklenen özelliklerinin değişmesine yol açar.
- Bilgi Sızdırması (ing. leak information) : Devre içindeki mahrem, saklanan bilgilerin DTA aracılığıyla dışarı sızdırılmasıdır.
- Hizmet Erişimi Engellenmesi (ing. DoS) : Devredeki bazı kesmelerin (ing. interrupt) işleme alınmamasına neden olabilir.

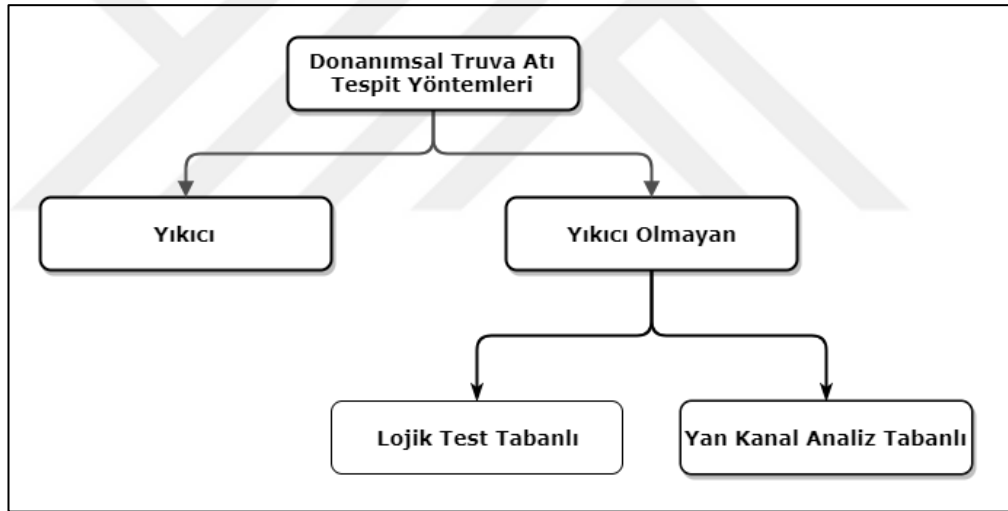
Bu tezde kullanılan tek bir DTA yapısı vardır. (Şekil 3.5) Kullanılan DTA, ilgili sınıflandırma yapılarından aktivasyona göre, kullanıcı girdileri ile aktifleşen yapıdadır. Etkisine göre incelendiğinde ise fonksiyonelliğinin değişmesi etkisine sahip yapıdadır. DTA, aktifleştğinde devrenin fonksiyonelliğini yani beklenen sonucunu da değiştirmektedir. Bu DTA'nın seçilmesinin nedeni, DTA oldukça küçük olduğu için devreye etkisinde küçük olmasıdır. Bu nedenle YKA yöntemleriyle tespiti de oldukça zordur. DTA'nın gecikme ve güç tüketim gibi yan kanal sinyallerine etkisi küçük olduğu için varyasyon etkisinden ayırt edilememektedir. Bu şartlar altında DTA'nın tespit edilebilmesi için kullanılacak YKA yönteminin, DTA etkisi ile varyasyon etkisini birbirinden ayırması gerekmektedir. Ayrıca bu DTA tipinin bir diğer zorluğu da sürekli aktif olmamasıdır. Böylece sadece çok nadir gelebilecek olan bir girdide aktif olacak şekilde modellenenabilir.

2.2. DTA Tespit Yöntemleri

Olası DTA'ların çok çeşitli olmasından dolayı, DTA'ların her çeşidini tespit edebilecek tek bir yöntem bulunmamaktadır. Tespit yöntemleri yıkıcı ve yıkıcı olmayan olarak adlandırılan iki ana başlık altında incelenirler. Yıkıcı yöntemde çeşitli kimyasallar kullanarak, üretilmiş devre katman katman soyulur. Her bir aşamada elektron mikroskopuyla yapısı gözlemlenerek tersine mühendislik yöntemleriyle analiz edilir. Analiz sonucunda devrenin şeması elde edilir. Elde

edilen şemaya bakarak, sadece yıkılmış olan devrede DTA olup olmadığı anlaşılabilir. Elde edilen şema tasarlanan devre şeması aynıysa DTA içermemektedir, şema aynı değilse yani ekleme varsa yıkıma uğrayan devre DTA içermektedir. Bu yöntem, kullanılan gereçler nedeniyle oldukça pahalı ve zaman gerektiren bir yöntemdir. Ve işlem sonucunda tüm devre yıkıma uğradığı için elimizde kullanılabilir bir yapı kalmamaktadır. Bununla birlikte üzerinde DTA bulunmayan devre altın çip verilerini elde etmek için kullanılabilir. Yıkıcı yöntem uygulanmadan önce, devre üzerinde YKA yöntemleriyle gecikme ve güç tüketimi gibi gerekli ölçümler yapılır. Ölçümler elde edildikten sonra, yıkıcı yöntem uygulanarak ve devrenin DTA içermediğinden emin olunursa, bu çip altın çip olur.

Dezavantajları ise uzun zamana ihtiyaç duyması, pahalı araçlar gerektirmesi ve sonucunda yıkıcı yöntem uygulanmayan tümleşik devreler hakkında herhangi bir garanti verememesi sayılabilir.



Şekil 2.3: DTA Tespit Yöntemlerinin Sınıflandırılması.

Yıkıcı olmayan yöntemlere geçmeden önce altın model ve altın çipin tespit yöntemleriyle ilişkisi incelenecektir. Altın model elimizdeki doğru tasarım, altın çip ise içinde DTA olmadığından emin olunan çiptir. Altın model elimizde olsa bile altın çipten gelecek olan ölçüm verilerine ihtiyaç duyulmaktadır. Çünkü tasarım yani altın model, üretilmiş çipin gecikme değerinin ne olması gerektiğini söylemez. Çipin sağlaması gereken hız ve güç tüketimi gibi değerlerin ölçümleri altın çip üzerinde yapılır.

Yıkıcı olmayan yöntemler, mantıksal teste dayalı (ing. logic testing based) ve YKA dayalı olmak üzere iki ana başlıkta incelenir. Mantıksal test yönteminde fonksiyonel test yapılarından otomatik sınaama örüntüsü üretimi (OSÖÜ) (ing. automatic test pattern generation) yapısı kullanılır. Aslında bu yapı devreler üzerinde hata analizinde, devrenin fonksiyonelliğinin kontrolünde kullanılır. Bu nedenle devrede bulunan DTA'nın aktifleşmesiyle ve tespitiyle ilgilenir. Devrede üssel sayıda bağlantı olabileceği varsayılmaktadır. Mantıksal testler de devredeki her bir bağlantı noktası için analiz yaparak çalışıklarından, tüm devre bağlantıları üzerinde çalışarak DTA tespit etmesi mümkün değildir. Bunun yerine devrenin küçük ama önemli parçalarında kullanılarak kontrol yapılması önerilir.

DTA, mantıksal test sırasında verilen girdilerde aktifleşmeyecek şekilde devreye yerleştirilmiştir. Amaç, testlerde tespit edilememeleridir. Fakat mantıksal test yöntemlerine yeni bir bakış açısı getiren VERITRUST [13] yöntemi DTA'yı aktifleştirecek olası girdileri, devrede belirlediği inceleme bölgelerini kullanarak belirler. Bölgeleri belirlerken de testlerde aktifleştirilmeyen yapılara odaklanır. Bu sayede fonksiyonel yöntemlerle tespit edilemeyen girdileri belirleyerek DTA tespiti yapmış olur.

YKA yöntemleri devrenin performansına yönelik gecikme, güç tüketimi, ısı gibi verilerin ölçümlerinde kullanılırlar. Bu ölçüm sonuçları ile, altın çip üzerinde yapılan ölçüm sonuçlarını kıyaslayarak, ne kadar sapma olduğunu ve devrede istenmeyen başka etkilerin oluşup oluşmadığını kontrol ederler. Devreye eklenen DTA'lar aktifleşmese bile YKA sonuçlarında sapmaya neden olurlar. YKA yöntemlerinin en büyük avantajı DTA'nın aktifleşmesiyle ilgilenmemeleridir. YKA yöntemlerinin en büyük dezavantajı ise üretimden kaynaklanan ve önlenemeyen varyasyon etkisidir. Varyasyon etkisi de ölçüm sonuçlarında sapmaya neden olur. DTA'ların da ölçüm sonuçlarında sapmaya neden olduklarını belirtmiştik. Ölçüm sonucunda oluşan sapmanın nedeninin varyasyon mu DTA mı olduğu anlaşılammamaktadır. Bir problem olarak, varyasyon etkisi ile DTA etkisini birbirinden ayırmak çok zordur. Varyasyon etkisi ile DTA etkisini birbirinden ayırıştırılabilmek için, çalışmalarda varyasyon etkisinin gizleyemeyeceği büyüklükte bir DTA yapısı kullanılır [12].

Yol gecikme değeri, yola ait girdi sinyalindeki değişiklik zamanı ile çıktı sinyalindeki değişiklik zamanı arasındaki süredir. Yol gecikme değerinin parmak izi

ölçümü, seçilen bir yola ait çeşitli çip örneklerinin gecikme ölçümleridir ve DTA içermediğinden emin olunan örneklerdir. Gecikme Analizinde, DTA'nın eklendiği yola ait ölçülen gecikme değeri değişir. Bu değişim, parmak izi ölçümleriyle kıyaslanarak tespit edilebilir. [15] yönteminde amaçlananlar, DTA'ları kategorize ederek tespiti çok zor olanları bile tespit edebilmek ve gecikme değerinin parmak izini analiz ederek varyasyon etkisinden kurtulup DTA tespit edebilmektir. Bu yönteme ek olarak [16] da bahsedilen yöntemden yardım alarak yol gecikme değeri hızlıca ölçülebilir. [16] yönteminde, çok sayıda yazmaçlardan (ing. register) oluşan devreler üzerinde, DTA eklenmesi gibi tasarımda yapılan değişiklikleri algılamak için, gecikme değerini hızlı ve devre fonksiyonelliğine etki etmeden ölçer.

Bir başka gecikme ölçümüne dayalı yöntemde ise DTA'nın devrenin gecikme değerine olan etkisinin artırılması üzerinde durulmuştur. Bunun için yapılacak tüm ölçümlerde DTA'yı içeren en kısa yol seçilmiştir [17]. Fakat bu yöntemdeki sonuçlar, çiplerin tamamının DTA içermeyen ve DTA içeren hallerinden oluşmaktadır. Bu yöntemde amaçlananlar, test maliyetini en aza indirmek için DTA barındıran en kısa yolu seçmeye dayanan yeni bir yöntem oluşturmak, verimli bir şekilde DTA tespit edebilmek için DTA'ları kategorilere ayırmaktır.

Bu tezde kullanılan DTA, devreye ek bir yapı kullanılarak eklenenlerdendir. Şekil 3.2'deki DTA'nın hangi sınıflara girdiği Bölüm 2.1'de anlatılmıştır. Bu tipte bir DTA seçme nedenimiz, hem fonksiyonel testlerle aktifleştirilmesi zor olduğu hem de gecikme ölçümlerine etkisi çok az olduğu için tespitinin zor olmasıdır. Gecikme tabanlı ölçümlerde DTA tespit oranını arttırmak için, en kısa yolun seçilmesi [17] gibi güç tabanlı ölçümlerde de tespit oranını arttırmak için, bölgesel güç ölçümleri üzerinde çalışmak işe yarayacaktır [14].

DTA tespit yöntemlerinin nasıl ve ne zaman kullanılacağı, devrenin ve DTA yapısına göre değişmektedir. Bazı DTA çeşitleri güç tüketimine bakarak kolayca bulunabilirken, bazıları da gecikme analiziyle bulunabilir. Aslında amaç, DTA etkisini ölçüm yaptığımız değerinden etkisinden bağımsız hale getirebilmektir. Böylece DTA etkisi gizlenemez. Bazen tek bir tespit yöntemi kullanarak DTA etkisini tespit etmek mümkün olmayabilir bu gibi durumlarda çoklu parametre (ing. multiple-parameter) yöntemleri kullanılır. Özellikle YKA yöntemlerinin birlikte kullanılmasıyla daha kesin ölçüm sonuçlarına ulaşılmış olur.

Çoklu parametre kullanan yöntemlerden [14] akım ve frekans değerlerinin korelasyonunu incelemektedir. Yöntem elindeki altın çipe göre test vektörleri üretip, akım ve frekans değerlerini ölçer. Devrede frekans değerlerini ölçmek için arka arkaya bağlanmış tek sayıdaki DEĞİL (ing. NOT) kapılarından oluşan halka devresi eklenmektedir. Bu osilatör devresinin eklendiği yol üzerine DTA eklenmeyeceğini varsaymaktadır. Ve bu sayede farklı devrelerde ölçülecek olan tüm frekans değerlerinin değişiminin nedeni sadece varyasyon etkisi olacaktır. Bunun yanında devreden ölçülen akım değeri, DTA'dan etkilenecek ve DTA içerip içermediğine göre farklı değerler verecektir. Frekans değeri ve akım değeri arasında beklenen bir korelasyon vardır. Biri arttıkça diğerinin de artması beklenir. DTA içeren ve içermeyen devreler, akım-frekans grafiğinde farklı dağılım gösterirler ve böylece korelasyonu bozan dağılımdaki devreler, DTA içeren devreler olarak tespit edilir. Yöntemde akım hem frekans değerlerini kullanmasının nedeni yan kanal analiz yöntemlerinin birkaçını bir arada kullanarak, sadece tek bir yöntem kullanmanın oluşturduğu eksikleri gidermektir. Eksiklerden kastedilen, YKA yöntemlerinin en büyük sıkıntısı olan varyasyon etkisinin DTA etkisini gizlemesidir.

3. METOT

3.1. Varyasyon Modeli

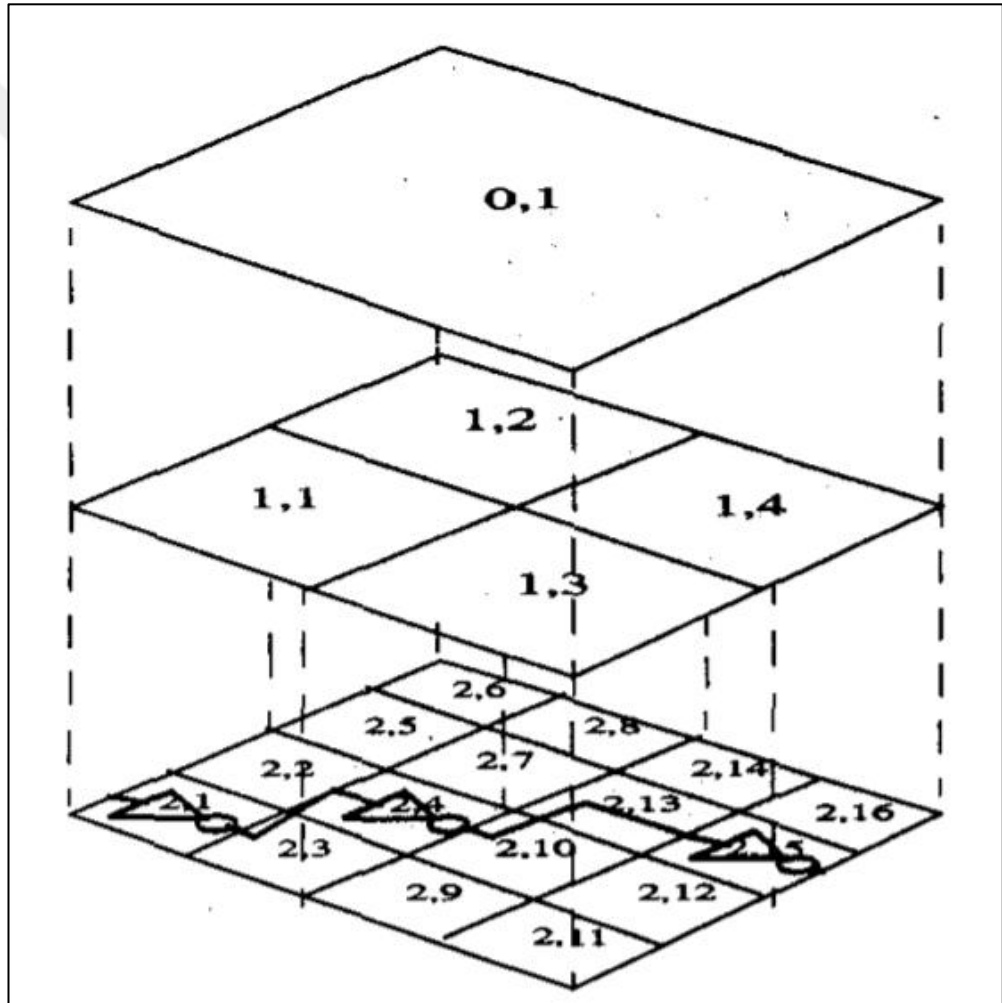
Tümleşik devre üretim süreci varyasyonu (ing. process variation), gecikme temelli yan kanal analizinin zaâfiyete uğramasına neden olan ana unsurdur. Varyasyonun devreye yapacağı ilave etki DTA'nın devreye yapacağı ilave etkiden çok daha fazla olabildiğinden, özellikle DTA boyutu küçük olduğunda, DTA etkisi ayırt edilememektedir [13,14,15,17]. Varyasyon etkisinin DTA etkisini gizlediği ve bu yüzden DTA etkisinin ayırt edilemediği Şekil 3.6'daki örnekte de gösterilmektedir.

Varyasyon etkisi ne kadar gerçeğe yakın modellenirse, simülasyon ölçümleriyle devre üzerinde yapılan ölçümler arasındaki ilişki o kadar sağlıklı olur. Bu yüzden yan-kanal analiz temelli DTA tespit yöntemleri için kullanılan varyasyon modeli oldukça önemlidir. Modern çip üretim teknolojilerinde, çip içi varyasyon bileşenleri çipler arası varyasyon kadar önemlidir [14]. Çip içi ve çipler arası varyasyon modellenirken, gerçeğe yakın olması için, konuma bağlı korelasyonun göz önünde bulundurulması gerekmektedir.

Konuma bağlı korelasyon modeline göre, varyasyon modelinde kullanılan rastgele parametreler (ing. random parameters) aslında birbirleriyle ilişkilidirler. Konuma bağlı korelasyonun sonucu olarak, tümleşik devrede birbirine yakın karelerdeki kapılar, birbiriyle ilişkili rastgele parametrelere sahiptir (ing. correlated random parameters). Bu korelasyonlar üretim safhasında birbirine yakın devre elemanlarının, üretim hatalarından benzer şekilde etkilenmesi sonucu oluşurlar. Bu nedenle kapılar arasındaki uzaklık arttıkça konuma bağlı korelasyon değeri de azalmaktadır.

Konuma bağlı korelasyon modelinde [19], çip Şekil 3.1'deki gibi çeşitli bölgelere ayrılır. Bölgelere ayrılma işlemi dördün ağaç (ing. quad-tree) yapısına göre yapılır. Çip önce dördün ağaç yapısına göre katmanlara bölünür. Şekilde katman sayısı üçtür ve l harfi ile ifade edilir. Her bir katman, $2^l \times 2^l$ tane kare olacak şekilde tekrar bölümlenir. 0. katman $2^0 \times 2^0 = 1$ kare parça halindedir. 1. katman $2^1 \times 2^1 = 4$ kare parça halindedir.

Her bir kare parça ITRS raporunda [26] belirlenen, devreye dağıtılmış değerler içerir. Örneğin, L kapı genişliğinin etkisi 45nm teknolojisine göre %12 olarak belirtilmiştir. Bu %12 değeri dördün ağaç yapısındaki her bir katmana eşit şekilde dağıtılır. Böylece her bir katmana %4 etki oluşturacak şekilde dağılır. Olasılık yoğunluk fonksiyonuna göre $3\sigma/\mu$ %4 olacak şekilde çizdirilmiş sürekli grafikten, eğer 2. katman ise $2 \times 2 = 4$ rastgele değişken seçilir, eğer 3.katman ise $4 \times 4 = 16$ rastgele değişken seçilir. Böylece dördün ağaç yapısındaki her bir kare parça için bir rastgele değişken seçilmiş olur. Bu örnek, varyasyon parametrelerinden sadece biri olan L içindir.



Şekil 3.1: Dördün Ağaç Yöntemi ile Konuma Bağlı Korelasyon Modeli.

Her bir kare parçaya ait değişken önce katman numarası sonra katman içindeki parça numarasını gösteren $(\Delta L_{l,r})$ şekilde ifade edilir. Şekil 3.1'de (2,1), (2,4), (2,15)

karelerindeki kapılara ait varyasyon modelleri sırasıyla Denklem 3.1, 3.2, 3.3'de gösterilmiştir. $\Delta L_{l,r}$ ile gösterilenler bağımlı rastgele değişkenlerdir. ΔL_r ise bağımsız rastgele değişkendir ve varyasyon modelinden bağımsız olarak ilgili kapı üzerine etki etmektedir.

$$\Delta L_1 = \Delta L_{2,1} + \Delta L_{1,1} + \Delta L_{0,1} \quad (3.1)$$

$$\Delta L_2 = \Delta L_{2,4} + \Delta L_{1,1} + \Delta L_{0,1} \quad (3.2)$$

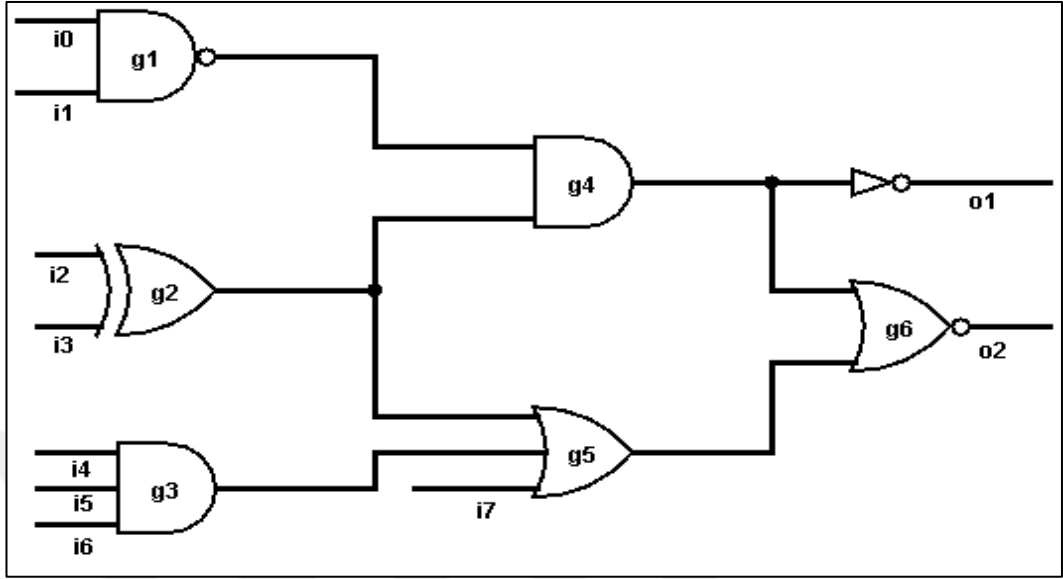
$$\Delta L_3 = \Delta L_{2,15} + \Delta L_{1,4} + \Delta L_{0,1} \quad (3.3)$$

Denklemlerde de görüldüğü üzere, 1. kapı ve 2. kapı birbiriyle kuvvetli ilişkilidir, ortak olarak $\Delta L_{1,1}$ ve $\Delta L_{0,1}$ değişkenlerine sahiptirler. Ama 3. kapı, ortak olarak sadece $\Delta L_{0,1}$ sahip olduğu için diğer ikisiyle daha zayıf ilişkilidir.

Birbirine yakın ama farklı kare parçalarda bulunan kapılar, bire bir aynı kare parçalarda bulunan kapılara göre daha az ilişkilidirler. Ama çip içindeki kapıların birçoğu aynı kare içine düşüyor ve birbirine göre kıyaslama yapılamıyorsa, dördün ağaç yapısına 1 katman daha ekleyip, çipin bölümlendiği kare sayısını arttırmak sorunu çözecektir. Şekil 3.1 de 3 katmanlı yapı örnek verilmişti ama bizim çalışmamızda varyasyon modelini gerçeğe daha da yakınlaştırmak ve kapılar arasındaki ilişkiyi daha iyi sınıflandırmak için 4 katmanlı yapıyı kullanarak, konuma bağlı korelasyon modellenmiştir.

Elimizde çipe ait tasarım bulunmaktadır. Simülasyon ölçüleri için, tasarıma varyasyon modeli eklenerek, istenildiği kadar çip kopyası oluşturulabilir. Oluşturulan bu çip kopyalarının her biri, sonraki kısımlarda örnek olarak adlandırılacaktır. Ve her bir örnek için bir \mathbf{X} vektörü vardır. \mathbf{X} vektörü her bir transistör için, varyasyon modelinden gelen kapı genişliği ve eşik gerilimi parametreleri için, konuma bağlı korelasyon modeline ait rastgele değişkenleri içermektedir. \mathbf{X} vektörü toplamda 170 rastgele değişken barındırmaktadır. Varyasyon modelindeki parametreler olan \mathbf{L} ve \mathbf{V}_{th} 'ın her biri için 85 tane rastgele değişken kullanılır. Bu 85 rastgele değişken ise dördün ağaç yapısındaki her bir kare parça için bir rastgele değer bulunmasından gelir. 85 rastgele değişken, $8 \times 8 + 4 \times 4 + 2 \times 2 + 1$ hesabı ile oluşur.

3.2. Problem Formülasyonları

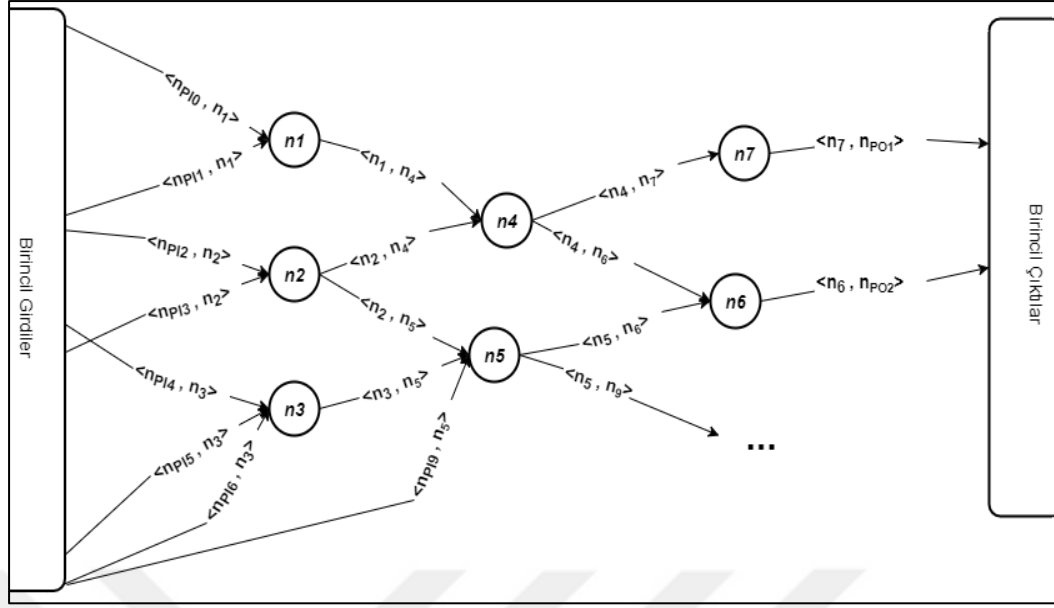


Şekil 3.2: Devrenin Lojik Kapılarla Gösterimi.

Netlist formatında tasarlanan devre yapıları, çizge yapılarına dönüştürülmüştür. Çalışmanın tamamında devrenin çizge formatına dönüştürülmüş yapısı üzerinden işlemler yapılmaktadır. Şekil 3.2’de netlist formatının ifade ettiği devre yapısı gösterilmiştir. Şekil 3.3’de aynı devrenin çizge hali gösterilmektedir.

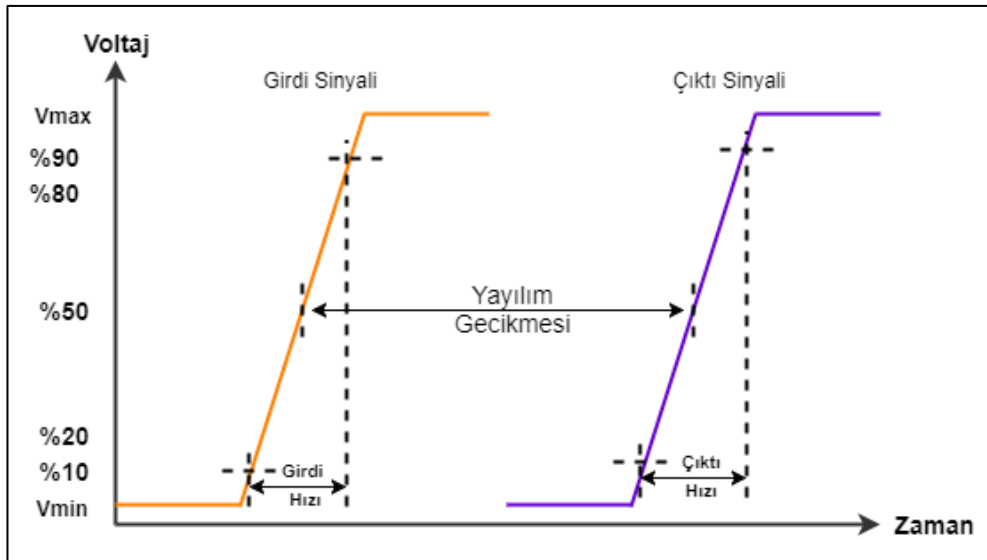
Netlistte devre kapısı olarak ifade edilen yapılar çizgede düğüm olarak, 2 devre kapısı arasında bağlantı olarak ifade edilen yapılar kenar olarak dönüştürülmüştür. Düğümü ifade eden yapı n_q şeklinde gösterilir. Çizgede q adlı düğümü temsil eder. Kenarı ifade eden yapı $\langle n_p, n_q \rangle$ şeklinde gösterilir.

Çizgede yol P harfi ile sembolize edilir, k tane düğümden oluşan yol $P = \langle n_{q1}, n_{q2}, n_{q3}, \dots, n_{qk} \rangle$ şeklinde gösterilir. n_{q1} düğümünün sembolize ettiği devre kapısı birincil devre girdilerinden (ing. primary inputs) birine bağlıdır. n_{qk} düğümünün sembolize ettiği devre kapısı birincil devre çıktılarından (ing. primary outputs) birine bağlıdır. Şekil 3.3’de gösterilen örnek çizgede, $P1 = \langle n_{p10}, n_1, n_4, n_6, n_{p02} \rangle$ örnek yol olarak gösterilebilir.



Şekil 3.3: Devrenin Tek Yönlü Çevrimsiz Çizge Formatında Gösterimi.

Yayılm gecikmesi (ing. propagation delay) (Şekil 3.4), bir sinyalin belirli ağ veya kapı üzerinde yayılması için gereken zamanı ifade eder. Belirli ağ olarak, devreden seçilen yolları kullanmaktayız. Bu nedenle yol için yayılım gecikmesi, seçili yola ait girdi sinyalindeki değişiklik zamanı ile çıktı sinyalindeki değişiklik zamanı arasındaki süredir. Girdi ve çıktıda oluşan değişiklik zamanları, ölçülebilen en fazla değerinin yüzde ellisine ulaştığı zamandan başlar.



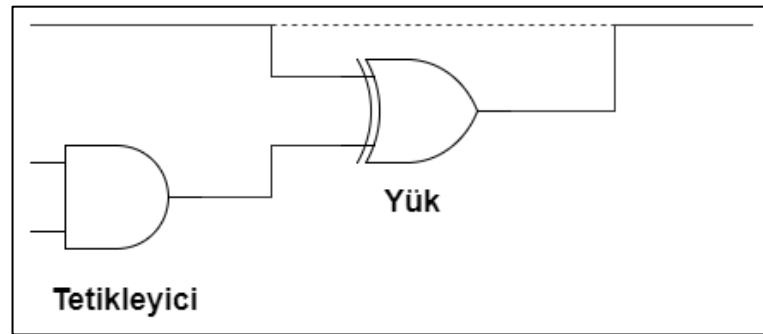
Şekil 3.4: Yayılm Gecikmesi Hesabı.

Yol (yayılm) gecikmesi, yol üzerindeki düğümlerin gecikmelerinin toplamıdır. Denklem 3.4 yol gecikmesi değeri için ifade etmektedir. Bir P yoluna ait ve rastgele parametreleri Bölüm 3.1’de anlatılan X vektörüyle verilmiş olan bir devrede, yol Gecikmesinin gösterimi $d(P, X_j)$ sembolize edilir. Bu gösterim tek bir devredeki yol gecikme değeri ifade eder. l tane devre üzerinde test yapabilmek için, l tane yol gecikme değeri ihtiyacı vardır. 1’den l ’ye kadar her bir örnek için, bir X vektörü oluşturulur ve gecikme değeri hesaplanır. l tane yol gecikme değeri $d(P, X_{1...l})$ ile gösterilir. $d(n_i, X)$ devredeki n_i düğümünün, X_j vektöründe bulunan rastgele parametrelere göre hesaplanan gecikme değeridir.

$$d(P, X_{1...l}) = \sum_i^k d(n_i, X_j) \quad (3.4)$$

3.2.1. DTA Modeli ve Devreye Etkisi

Kullandığımız DTA devresi en küçük bileşimli (ing. combinational logic) yapılardan biridir ve Şekil 3.5’de gösterilmiştir. DTA’nın tetikleyici kısmında VE (ing. AND) kapısı, yük kısmında D-YA (ing. XOR) kapısı kullanılmıştır. Bu DTA, yapısı gereği 2 girdi değeri de 1 olduğunda aktifleşmektedir ve çıktı olarak giriş sinyalinin (d-ya kapısına giren değerin) tersini vermektedir. DTA’nın yük kısmı D-YA kapısı ile devreye bağlanmıştır ve bağlandığı yolun gecikme değerine etki eden kısmı da burası oluşturmaktadır. Kesikli çizgi normalde DTA içermeyen devrenin bağlantısını göstermektedir. Eklenen DTA yüzünden o bağlantı kesilmiş ve D-YA kapısından geçirilmiştir.



Şekil 3.5: DTA Yapısı.

Şüpheli kenar, üzerinde DTA barındırdığından şüphelenilen kenarı tanımlamaktadır. Şüpheli kenar, çalışmamızda rastgele seçilmiştir ama devre üzerinde daha az kenar inceleyerek, devrenin tamamı hakkında sonuca varmak için akıllıca şüpheli kenar seçen yöntemler kullanılabilir.

DTA'lar devrelere aktifleştiklerinde zarar verirler. Verdikleri zarar, tipine ve işlevine bağlıdır. Kullandığımız DTA aktifleştğinde, devrede bağlandığı yola ait beklenen çıktı değerine etki etmektedir. Böylece devrenin fonksiyonel olarak yapması gereken işlerden bazıları yapılmamış olur. DTA'lar çok nadir olarak aktifleşecek şekilde tasarlanırlar. Tetikleyici kısımlarının girdi değerleri de buna göre ayarlanır. Nadir olarak aktifleşmeye odaklanmalarının nedeni, fonksiyonel testlerde ortaya çıkmamaktır.

Kullandığımız DTA sofistike ve karmaşık yapıda değildir ama küçük yapıda olması, gecikme değerine olan etkisini de küçültmektedir. Bu da YKA ile tespitini zorlaştırmaktadır. YKA yöntemlerinin dezavantajlarından biri de küçük yük etkisinin, varyasyon etkisi içinde kaybolmasıdır.

Üzerinde DTA barındırdığından şüphelenilen bağlantı, şüpheli kenar (ing. suspected edge) olarak adlandırılır. İçinde şüpheli kenarı barındıran yol Denklem 3.5 ile ifade edilir. $d(n_{load}, X)$ DTA'nın devreye eklenen kısmının (yük) gecikme değerini ifade eder, $d(P, X_{1..l})$ değerinden tek farkı da budur.

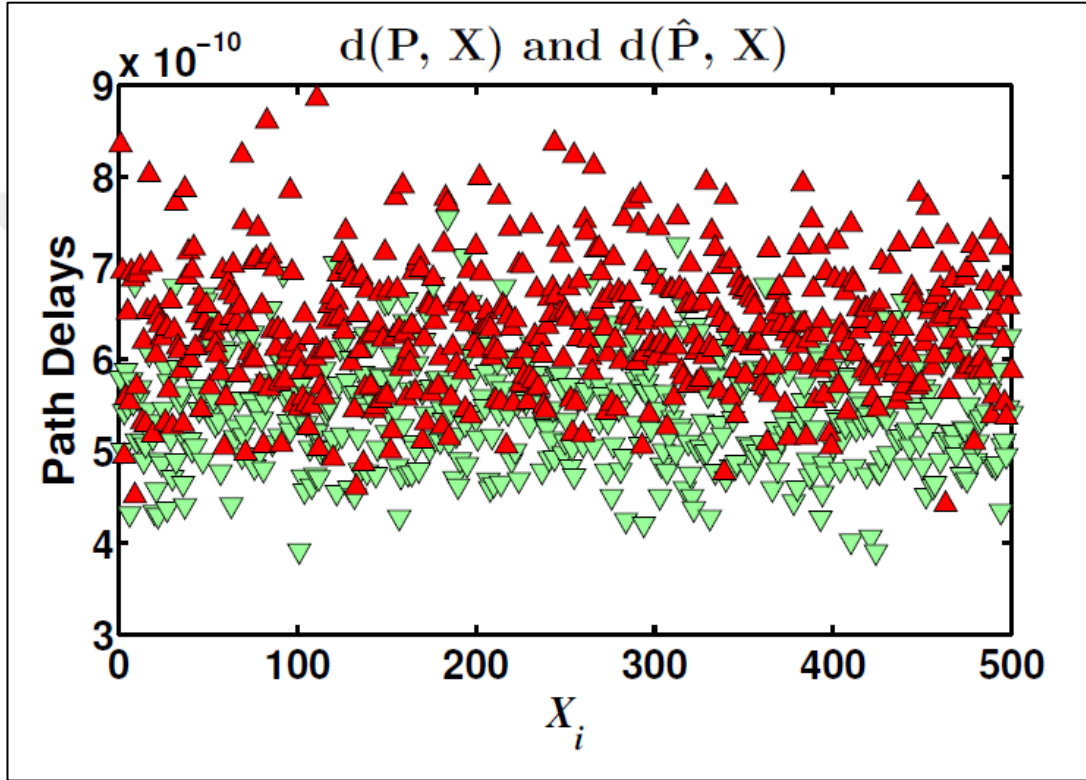
$$d(\hat{P}, X_{1..l}) = \sum_i^k d(n_i, X_j) + d(n_{load}, X_j) \quad (3.5)$$

Gecikme temelli yöntemler kullanarak DTA tespitinin amacı, DTA'nın devreye eklenen kısmının (ing. payload) gecikme değeri ile varyasyondan dolayı eklenen rastgele değeri birbirinden ayırabilmektir.

3.2.2. Yol Gecikmesine Bakarak DTA Tespiti Zorluğu

Bölüm 3.2 de açıklanan varyasyon modeline göre oluşturulmuş 1000 tane X vektörü sayesinde, 1000 farklı çip oluşturulmuştur. Şekil 3.6'da, bu çiplere ait yol gecikme değerleri gösterilmektedir. Bu değerler, bir yolun DTA içeren ve içermeyen haline göre hesaplanmış gecikme değerlerini göstermektedir. Yol gecikme

değerlerinin 500 tanesi DTA içermeyen çiplere aittir ve yeşil renk aşağı bakan üçgenlerle ifade edilmiştir. DTA içeren gecikme değerleri de kırmızı renk yukarı bakan üçgenlerle gösterilmiştir. Şekilde gecikme değerleri gösterilmiş olan yol, c3540 devresinde bulunmaktadır ve rastgele seçilmiş şüpheli kenarı içine alacak ve en küçük gecikme değerine sahip olacak şekilde seçilmiştir. Küçük gecikme değerine sahip olan yol, DTA'nın kısmi etkisini arttırmak için seçilmiştir [17]. Yola eklenen DTA ise Şekil 3.5'de gösterilmiştir.



Şekil 3.6: Örnek Çiplerin Yol Gecikme Değerleri.

Şekil 3.6'da DTA içeren ve içermeyen yolları, sadece gecikme değerlerine bakarak birbirinden ayırmanın çok zor olduğunu açıkça görmekteyiz. Halbuki DTA'nın gecikme değerine olan etkisi artsın ve ayırım sağlanabilsin diye yol olarak da en küçük gecikme değeri olan seçilmiştir ama bu bile varyasyon etkisinden DTA'nın etkisini kurtaramamıştır. Sonuç olarak, sadece yol gecikmesine dayanarak yapılacak bir sınıflandırmada yanlış sınıfa konan çip sayısı oldukça fazla olacaktır.

3.3. Konuma Bağlı Varyasyon Modeline rağmen İlişkili Yol Gecikmesi Tabanlı DTA Tespiti

Bir önceki bölümde anlatıldığı üzere sadece gecikme değerlerine bakarak (Şekil 3.6), DTA tespiti yapmak, varyasyonlar yüzünden mümkün değildir. Kullanılan konuma bağlı korelasyon modeli sayesinde SLE ölçümlerinde elde edilen gecikme değerleri, SPICE ölçümlerinden elde edilen değerlerine yakın olmaktadır. Varyasyon modelini geliştirmek, ölçüm sonuçlarını gerçeğe yakınlaştırmaktadır. Peki varyasyon etkisi DTA etkisinden nasıl ayırt edilebilir? Bunun ayrımını ortaya koyabilen yeni bir değişkene ihtiyaç duyulmaktadır. Bu yeni değişken, varyasyondan az etkilenirken aynı zamanda DTA'dan çok daha fazla etkilenmelidir ki DTA etkisi ortaya çıkabilsin. Yeni değişkeni bulmak için, devre üzerinde varyasyondan aynı şekilde etkilenen iki parametre belirlenip bu parametreler oranlanmalıdır [19, 33]. Böylece bu yeni oran istenildiği gibi, varyasyona göre bir çip örneğinden diğerine geçildiğinde fazla değişmeyecektir. Bununla beraber bu öyle bir oran olmalıdır ki DTA eklendiğinde oran değeri değişsin.

Bulunması gereken yeni değeri bu şekilde tanımlandıktan sonra problem, devrede varyasyonlardan aynı oranda etkilenen parametrenin ne olduğunu bulmaya dönüşmüştür. Devre içindeki transistör parametreleri konuma bağlı korelasyona sahiptir. Yani transistörler konum olarak birbirlerine yakınlaştıkça, transistör parametrelerinin çipten çipe değişimleri birbirine benzer şekilde olmaktadır. Çünkü aralarında korelasyon vardır. Buradan yola çıkarak ortaya koyduğumuz yeni düşünce, varyasyondan aynı oranda etkilenen parametrelerin, devreden seçilen yol gecikmeleri olabileceğidir. Devreden öyle iki yol çıkartılmalıdır ki biri diğerine çok yakın kare parçalardan geçsin, böylece bir yola ait gecikme değeri ile diğer yola ait gecikme değeri, çipten çipe benzer şekilde azalır/artacaktır. Ayrıca bu yollardan biri şüpheli kenardan geçerken diğeri geçmemelidir ki DTA'dan sadece biri etkilenmiş olsun.

Özetle gecikme temelli DTA tespitinde problemin esas zorluğu, varyasyon etkisinin DTA etkisini gizlemesidir. DTA etkisini bir şekilde varyasyon etkisinden kurtaran bir değişkene ihtiyaç duyulmaktadır. Gecikme değerlerinin oranı, aranan bu değişkeni karşılamaktadır.

Herhangi bir çip örneği için bu iki yoldan herhangi biri üzerine DTA eklendiğinde tespit edebilmek için, iki yola ait gecikme değerleri ölçülür. Bu

gecikme deęerleri oranlanarak, oran deęerleri elde edilir. Artık her bir ip iin bu oran deęeri kullanılarak oran daęılımları oluřturulur. Her bir ip üzerinde, kendine ait oran daęılımları analiz edilerek, DTA olup olmadıęını tespit edebilir. ünkü bu oran deęeri sayesinde varyasyon etkisi en aza indirilmiřtir. Bu sayede varyasyon etkisinin DTA etkisini gizlemesi önlenmiřtir.

alıřmanın kalan kısımlarında, bu yolların devreden verimli bir řekilde nasıl seildięi ve DTA tespitinin nasıl yapıldıęı anlatılacaktır. Bölüm 3.2’de anlatıldıęı gibi devreler netlist yapısından izge yapısına dönüřtürülmüřtür. Devrenin sahip olduęu tüm parametre bilgileri, düęüm ve baęlantılar üzerine tařınmıřtır. Bölüm 3.3’de anlatılacak tüm iřlemler izge yapısı üzerindedir. alıřmamızı beř alt bařlıkta inceleyebiliriz.

3.3.1. Düęüm Tahmini Gecikme Hesabı

Yolların seiminden önce, izge yapısındaki her bir düęüme ait gecikme deęerlerinin hesaplanması gerekmektedir. Gecikme deęeri varyasyon modeli uygulanarak hesaplanır ve ilgili düęümün verisi olarak kaydedilir. Düęümlere ait gecikme deęeri hesabı iin stokastik lojik efor (SLE) (ing. stochastic logical effort) yöntemi kullanılır [20].

SLE yöntemi, hızlı ama tahmini sonuçlar veren kapı gecikme modelidir. SLE kullanarak tümleřik devredeki bir yol gecikme deęeri hızlı ve etkili řekilde belirlenebilir. SLE yöntemi, gecikmeyi modellerken varyasyon parametrelerini de kullanmaktadır.

$$d_r^{SLE}(X) = \tau(X)(p(X) + g(X)h) \quad (3.6)$$

Denklem 3.6’da d_r^{SLE} , r düęümünün gecikme deęerini ifade eder. Denklemdeki X , devre iindeki tüm rastgele deęiřkenleri iinde barındıran vektördür. p parazitik gecikmeyi, g mantıksal eforu ve h elektriksel eforu temsil etmektedir. Mantıksal efor g , kapının karmařıklıęının ölçüsüdür. Sadece kapının topolojisine baęlıdır ve kapının büyüklüęü ve yükünden baęımsızdır. Parazitik gecikme p , kendi i direncine baęlı olarak kapının kendi i gecikmesini ifade eder ve kapıdaki transistörlerin boyutlarından baęımsızdır. Elektriksel efor h , mantık kapısının yük direncinin belirli bir girdinin direncine oranıdır. Denkleme göre, p ve g varyasyona baęlı

parametrelerdir. SLE modeline ait $\tau(X)$, $p(X)$ ve $g(X)$ parametreleri önceden karakterize edilir. Bölüm 3.1’de anlatılan varyasyon modeline ilişkin X vektörüne göre, transistör seviyesinde SPICE ölçümleri ile parametre değerleri hesaplanır ve arama tablosuna kaydedilir [20]. Tablodan, ilgili X vektörüne göre SLE parametrelerine erişilir ve bir düğümün gecikme değeri hesaplanır.

3.3.2. Şüpheli Yol Seçimi

Bu kısımda verilen şüpheli kenarı içinde barındıran yolun nasıl seçildiği anlatılmaktadır. Şüpheli yol (P_{susp}), devre içindeki yollardan üzerinde DTA barındırdığından şüphelenilen yoldur. DTA şüpheli yolun belirli bir kısmından devreye bağlanmaktadır, bu nedenle şüpheli yol seçerken DTA’nın devreye nerden bağlanacağına dikkat edilmektedir. DTA’nın devreye bağlandığı düşünülen kenar ise şüpheli kenar olarak adlandırılmaktadır. Böylece şüpheli yol tanımı, şüpheli kenarı içinde barındıran yol olarak değişmiş olur.

Devre yapısı Şekil 3.3’de gösterildiği gibi çizge formatındadır. Çizge, birincil girdilerden birincil çıktılara doğru, içinde döngü barındırmayan tek yönlü yapıdadır. Şüpheli yol seçimi yaparken, şüpheli kenarı içinde barındıran en kısa yolun seçilmesi gerektiğine karar verilmiştir. En kısa yol seçildiğinde [17], DTA’nın yol gecikme değerine olan etkisi de artmış olacaktır. Bu artış kendi başına DTA tespitini sağlamasa da yardımcı olmaktadır. Şüpheli yol tanımı, şüpheli kenarı içinde barındıran en küçük gecikme değerine sahip yol olarak son halini almış olur.

Şüpheli kenarı içinde barındıran en kısa yolu bulma probleminin çözümü için, çizge üzerinde en kısa yol bulma algoritma incelenerek, tek kaynaklı en kısa yol (ing: single source shortest path) algoritmaları içinde Dijkstra’nın algoritması [23] uygun bulunmuştur.

Dijkstra algoritması, tek kaynağın yani çizge üzerinde başlangıç düğümünün, çizgedeki diğer tüm düğümlere olan en kısa uzaklıklarını bulur. Yeni çizge yapısında kenarların ağırlık değerlerinin olması gerekmektedir. Düğüm gecikme değerlerini, kenar gecikme değerleri olarak tanımlamak gerekmektedir. Bizim çizgemizde düğümler, tahmini gecikme değerlerine sahiptirler. Düğüm gecikme değerleri SLE kullanılarak hesaplanır, Bölüm 3.3.1’de anlatılmıştır. Yol-gecikme değerleri yolun

üzerinden geçtiği kapıların, bulunan gecikme değerlerinin toplamı ile hesaplanır. (Denklem 3.4)

Şekil 3.7 Şüpheli Yol Seçme algoritmasını ifade etmektedir. Her bir birincil girdi düğümünden, şüpheli kenarın $\langle n_s, n_t \rangle$ ilk düğümüne (n_s) olan en kısa yollar Dijkstra algoritması kullanılarak bulunur. Dijkstra algoritması birincil girdi sayısı kadar çağırılır. Bu yollar π_{fromPI} kümesinde tutulurlar (sadır 2). Şüpheli kenarın ikinci düğümü (n_t)'den her birincil çıkışa olan en kısa yolları bulmak için bir kez daha Dijkstra algoritması çağırılır. Bulunan yollar π_{toPO} kümesinde tutulurlar (sadır 3). Son olarak π_{fromPI} ve π_{toPO} kümelerinin Kartezyen çarpımı, birincil girdilerden birincil çıktılara kadar olan tüm yol kombinasyonlarını üretir (sadır 4). Bu yolların her biri, şüpheli kenarı içerir ve yollara karşılık gelen <birincil girdi – birincil çıktı> düğüm çifti için en kısa gecikme değerine sahiptir. Şüpheli yol seçimi algoritması detayları aşağıdaki örnek üzerinde anlatılacaktır.

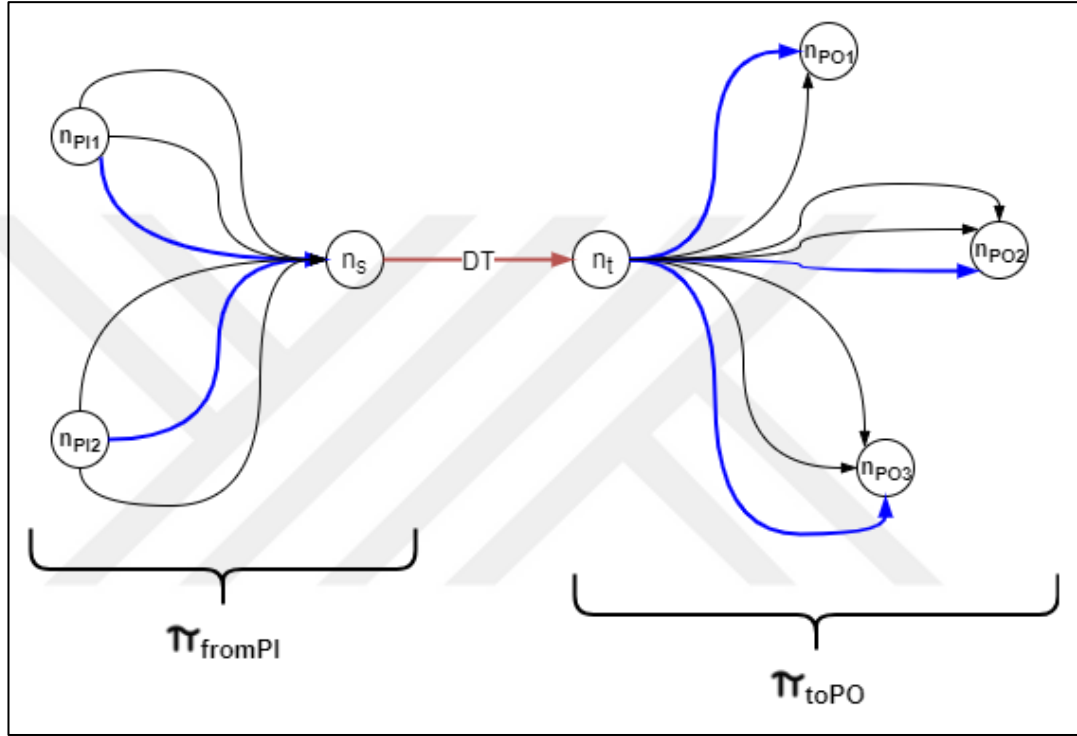
Şüpheli Yol Seçme - P_{susp} Seçimi

- 1: Her bir şüpheli kenar $\langle n_s, n_t \rangle$ için
- 2: Her bir birincil girdi (PI) düğümünden n_s düğümüne, bulunan yollardan en küçük gecikmeye sahip olanlar, π_{fromPI} aday kümesinde toplanır.
- 3: Her bir n_t düğümünden birincil çıktı (PO) düğümüne, bulunan yollardan en küçük gecikmeye sahip olanlar, π_{toPO} aday kümesinde toplanır.
- 4: $\pi_{susp} = \pi_{fromPI} \times \pi_{toPO}$ (iki yol adayları kümesinin Kartezyen çarpımı)
- 5: $P_{susp} = \pi_{susp}$ yol adayları kümesinden en küçük gecikme değerine sahip olan seçilir.

Şekil 3.7: Şüpheli Yol Seçme Algoritması.

Algoritmada satır 4'ü Şekil 3.8'de bulunan örnekle açıklamak gerekirse, π_{fromPI} kümesinde, $\langle PI_1 - n_s \rangle$ ve $\langle PI_2 - n_s \rangle$ düğüm çiftlerinden geçen en kısa gecikme değerine sahip yollar bulunmaktadır. En kısa gecikme değerine sahip yollar mavi renkle işaretlenmişlerdir. Diğer yollar, yani seçilmeyen yollar siyah renkle gösterilmiştir. $\langle PI_1 - n_s \rangle$, şüpheli kenarın ilk düğümü olan n_s 'den n_{PI1} düğümüne olan en kısa yolu ifade etmektedir. Şekilde bu iki düğümü birbirine bağlayan 3 yol bulunmaktadır, ama en kısa gecikme değerine sahip olan yol seçilmiştir. Diğer düğüm ikilileri içinde aynı durumlar geçerlidir.

Bu düğüm ikilileriyle ifade eden yollar, kartezyen çarpım kullanılarak, bir araya getirilir ve gerçek yollar oluşturulur. Bu yollar, şüpheli yol aday kümesine (π_{susp}) eklenir. Örnek olarak, $\langle PI_1 - n_s \rangle$ ile $\langle n_t - PO_2 \rangle$ düğüm ikilileri birleştirilince, $\langle PI_1 - PO_2 \rangle$ düğümleri arasında bulunan girdi ve çıktıya bağlı gerçek bir yol oluşmuş olur. Bu yol artık, $P = \langle PI_1, \dots, n_s, n_t, \dots, PO_2 \rangle$ denklemleriyle ifade edilir ve aday kümesine eklenir.



Şekil 3.8: İlişkili Yol Adayları Kümesi Örneği.

Çizge üzerinde her bir birincil girdiden her bir birincil çıktıya en kısa gecikme değerine sahip olan ve şüpheli kenarı barındıran yollar bulunmuştur. Bu yolların sayısı en fazla, girdi düğüm sayısı ile çıktı düğüm sayısının çarpımına kadar olabilir. Ama her bir girdiden veya çıktıdan, şüpheli kenara bağlantı olmayabileceği için şüpheli yol adaylarının sayısı daha az olabilmektedir. Örnekteki şüpheli yol aday kümesinde 6 yol bulunmaktadır.

π_{susp} İçinden en kısa gecikme değerine sahip olan yol P_{susp} olarak seçilir (sıra 5). Bu kümedeki herhangi bir yol, P_{susp} olarak kullanılabilir, ancak en kısa olanı kullanarak DTA tespit edilme olasılığı artar. En kısa yol sentezlenemezse veya

karşılık gelen bir ilişkili yol adayı bulunamazsa, π_{susp} kümesindeki bir sonraki en küçük gecikme değerine sahip yol P_{susp} olarak atanır.

3.3.3. İlişkili Yol Seçimi

Bir önceki bölümde şüpheli yol yani P_{susp} belirlenmişti. Bu bölümde ise P_{susp} ile ilişkili yol (ing. correlated path) belirleme anlatılacaktır. İlişkili yol P_{corr} ile ifade edilmektedir. İlişkili yolu bulmaktaki amacımız, varyasyondan benzer şekilde etkilenen iki yol gecikmesi elde etmektir. Önceden anlatıldığı üzere, eğer yol gecikmeleri varyasyondan benzer şekillerde etkileniyorsa, bu gecikmeleri oranladığımızda varyasyon etkisini ortadan kaldırmış oluruz. İlişkili yol ararken devredeki tüm yolları inceleyerek değil, daha kısa bir yöntemle bu işi yapmak istiyoruz. Çünkü devrede üssel sayıda, çok fazla yol bulunabilir. Devrede var olan konuma bağlı korelasyon nedeniyle, şüpheli yola ait transistörlerin bulunduğu karelere yakın olan diğer yollar, şüpheli yol ile daha ilişkilidir. Bu nedenle sadece şüpheli yola yakın kareleri kullanan yolları incelemek işi hızlandıran ve daha doğru sonuçlara ulaşmamızı sağlayan bir yöntem olacaktır.

İlişkili Yol Seçme - P_{corr} Seçimi

- 1: **Kırmızı kareler** = P_{susp} geçtiği kareler
- 2: Her bir düğüm için (kırmızı karelerden geçen)
- 3: **Yol Bulma (Düğüm)**
- 4: π_{corr} kümesindeki her bir yol P_i için
- 5: $r_i = \text{Corr} (d_{SLE}(P_i, X), d_{SLE}(P_{susp}, X))$
- 6: P_{corr} = En Yüksek korelasyon değeri, r_i 'ye sahip olan seçilir.

Şekil 3.9: İlişkili Yol Seçme Algoritması.

Yol Bulma (Başlangıç Düğümü)

- 1: **Başlangıç Düğümünün** tüm komşuları için
- 2: Her bir kırmızı kare parçası için
- 3: Eğer **kırmızı** kare parçası == komşu düğümün kare parçası
- 4: **Yol Bulma (Komşu Düğüm)**

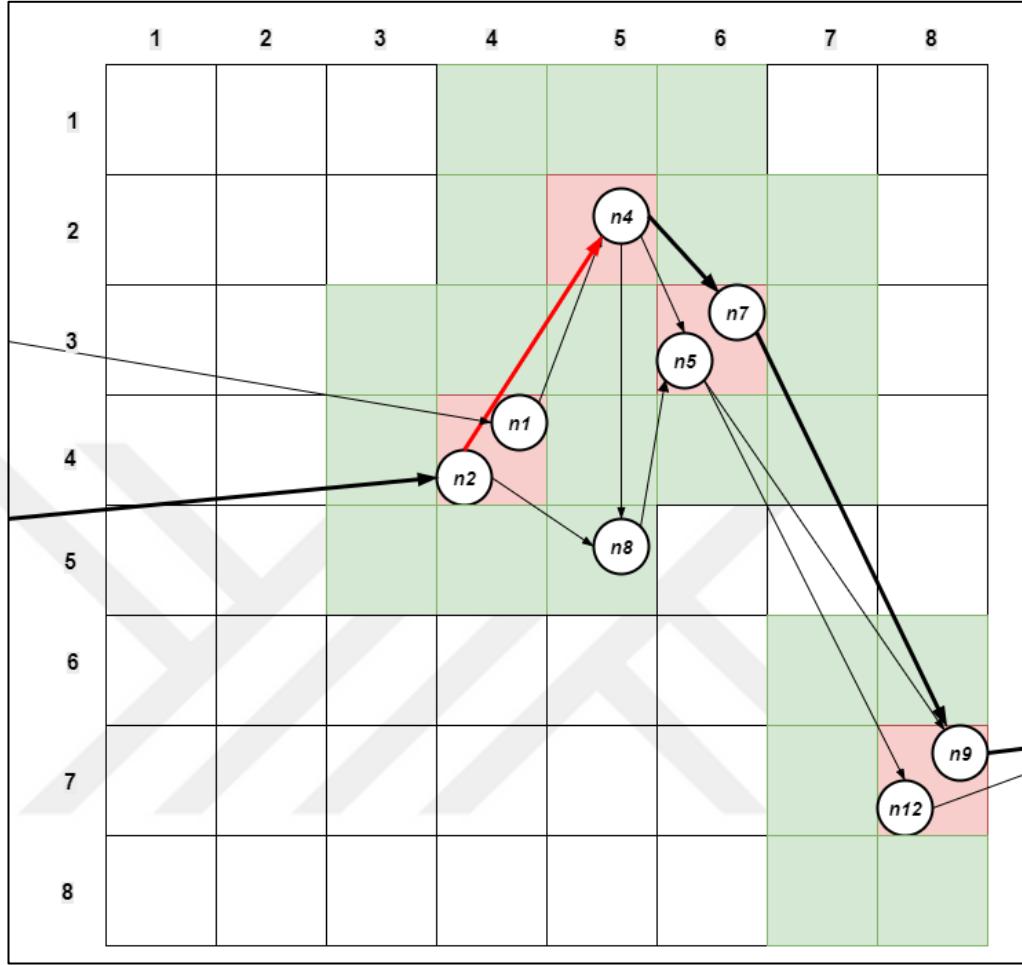
Şekil 3.10: Yol Bulma Algoritması.

Şekil 3.9’da ifade edilen ilişkili yol seçme algoritmasının çıkış dayanağı konuma bağlı korelasyon etkileri olduğu için, ilk adım olarak P_{susp} ait kapıların kareler, kırmızı kareler olarak kaydedilir (satır 1). Kırmızı karelerden geçen her bir düğüm için yol bulma algoritması (Şekil 3.10) çağırılır. Bu algoritma, kendi içinde özyinelemeli (ing. recursive) şekilde tanımlanmıştır (satır 2-3).

Yol bulma algoritması, kendisine verilen düğümü başlangıç olarak seçer. Ve fonksiyonun çalışma şartları sağlandığı sürece kendi kendini yeniden çağırarak, devre üzerinde dallanarak, çeşitli düğüm ikilileri seçmiş olur. Bahsedilen çalışma şartları, başlangıç düğümünün tüm komşularına bakılarak, en az birinin kaydedilen kare parçalarından geçmesidir. Bu şartı sağlayan komşu düğüm ve başlangıç düğüm ikilileri fonksiyonun çıktısı olarak döndürülür. Özyinelemeli fonksiyon, çalışma şartlarını sağlayan düğüm bulunamadığında sona erer. Ve bu dallanmalar sırasında bulunan tüm verileri yani düğüm ikililerini geri döndürür. Düğüm ikilileri, arama sırasıyla geri döndüklerinden, bulunan yolu ifade etmektedirler. Bulunan yolların, devre üzerinde tanımlanan yollardan biri olup olmadıklarının kontrolü için, herhangi bir birincil çıktıya ulaşıp ulaşmadıkları kontrol edilir, ulaşmayanlar elenerek geriye kalanlar ilişkili yol adayları (π_{corr}) olarak tanımlanır. π_{corr} kümesindeki her bir ilişkili yol adayı P_i ile P_{susp} arasındaki korelasyon katsayısı, $Corr$ fonksiyonu ile hesaplanır. Korelasyon katsayısı, bu iki yol gecikme değerinin değişiminin benzer olup olmadığını ifade eder. $Corr$ fonksiyonu, her bir aday için konuma bağlı varyasyon modeline ve SLE kullanılarak hesaplanan gecikme değeri ile korelasyon katsayısını hesaplar (satır 4-5). π_{corr} içinden, şüpheli yol ile en yüksek korelasyon değerine sahip olan ilişkili yol (P_{corr}) olarak seçilir (satır 6). Eğer P_{corr} sentezlenemezse π_{corr} kümesindeki bir sonraki en büyük korelasyon değerine sahip yol P_{corr} olarak atanır. Eğer π_{corr} kümesindeki hiçbir yol sentezlenemezse, şüpheli yol seçme adımına geri dönülür. π_{susp} şüpheli yol adayları içinden sıradaki en küçük gecikmeye sahip yol P_{susp} olarak seçilir.

İlişkili yol seçme algoritmasının çizge üzerinde nasıl çalıştığı Şekil 3.11’deki örnekle anlatılacaktır. Şekil 3.11’deki kareler, Bölüm 3.1’de anlatılan dördün ağaç modeline göre en alt katmanı göstermektedir. Varyasyon modeli için 4 katmanlı yapı kullanıldığından, en alt katmanda $8 \times 8 = 64$ kare parçası bulunmaktadır. Kare parçaları x ekseninde soldan sağa, y ekseninde yukardan aşağıya olacak şekilde

numaralandırılmışlar. Örneğin, n4 düğümü x eksenine göre 5, y eksenine göre 2. Karede bulunmaktadır. Bu kare 5x2 ile ifade edilir.



Şekil 3.11: Devrenin Konuma Bağlı Çizge Formatında Gösterimi.

Şekil 3.11’da daha koyu olarak gösterilen yol P_{susp} temsil etmektedir. $P_{susp} = \langle PI_3, n_2, n_4, n_7, n_9, PO_{12} \rangle$ yoluna ait düğümlerinin geçtiği kareler bir küme altında toplanır (şekilde kırmızıyla boyanmış kareler). Bu karelerde bulunan başka yollar yani ilişkili yollar aranır. Örneğimizde, P_{susp} ait kare kümesi = 4x4, 5x2, 6x3, 8x7 şeklindedir.

$\langle n_2, n_4 \rangle$ bağlantısı DTA içerdiğinden şüphelenilen bağlantıdır. İlişkili yol ararken, yolların P_{susp} ait kare kümesinden geçmeleri ama şüpheli kenar içermemeleri gerekmektedir. Aksi halde DTA eklendiğinde, iki yolda DTA’dan etkilenecek ve bu iki yolun oranı yine sabit olacaktır. İlişkili yol bulmamızın amacı oran değerinde oluşan değişimden faydalanmaktır. İlişkili yol, şüpheli kenarı

içermediği garanti edilirse, oran değerinde DTA etkisi gözlemlenebilir. Ve bu sayede DTA tespiti yapılabilir. İlişkili yolun, şüpheli kenar içermediği garanti edilemiyorsa, önerdiğimiz yöntem çalışmayacaktır.

$P1 = \langle PI_3, n_2, n_4, n_5, n_9, PO_{12} \rangle$, $P2 = \langle PI_3, n_2, n_4, n_5, n_{12}, PO_{18} \rangle$ içlerinde şüpheli kenarı barındırdıkları için seçilemezler. $P3 = \langle PI_1, n_1, n_4, n_5, n_9, PO_{12} \rangle$, $P4 = \langle PI_1, n_1, n_4, n_5, n_{12}, PO_{18} \rangle$ ve $P5 = \langle PI_1, n_1, n_4, n_7, n_9, PO_{12} \rangle$ yolları ilişkili yol adayları kümesine (π_{corr}) seçilirler. Bu yollar P_{susp} 'i oluşturan düğümlerle, birebir aynı karelerden geçen düğümlere sahiptir.

Çizgede birebir aynı karelerden yani kırmızı karelerden geçen başka yollar olmayabilirdi. Bunun nedenlerinden biri devrede çok az sayıda kapı olması olabilir. Böyle bir durumda ilişkili yol bulma yönteminin sınırlarının biraz esnetilmesi yeterli olacaktır. P_{susp} ait kare kümesine, komşu karelerin yani yeşil renkle boyanmış karelerin de eklenmesi ve bu şekilde yol aranması sorunu çözecektir.

$P6 = \langle PI_3, n_2, n_8, n_5, n_9, PO_{12} \rangle$, $P7 = \langle PI_3, n_2, n_8, n_5, n_{12}, PO_{18} \rangle$ yollarının 3 düğümü P_{susp} ile aynı karelerde, sadece 1 düğümü komşu karededir. Bu yollar da aday kümesine seçilirler. $P8 = \langle PI_1, n_1, n_4, n_8, n_5, n_9, PO_{12} \rangle$, $P9 = \langle PI_1, n_1, n_4, n_8, n_5, n_{12}, PO_{18} \rangle$ yollarının 4 düğümü P_{susp} ile aynı karelerde, sadece 1 düğümü komşu karededir. Bu yollar da aday kümesine seçilirler. Böylece $\pi_{corr} = P3, P4, P5, P6, P7, P8, P9$ olarak tamamlanmış olur. Buraya kadar örnek üzerinde anlatılanlar Şekil 3.10'da bulunan ilişkili yol seçme algoritmasının ilk 5 satırındaki işlemlere de örnek teşkil etmektedir.

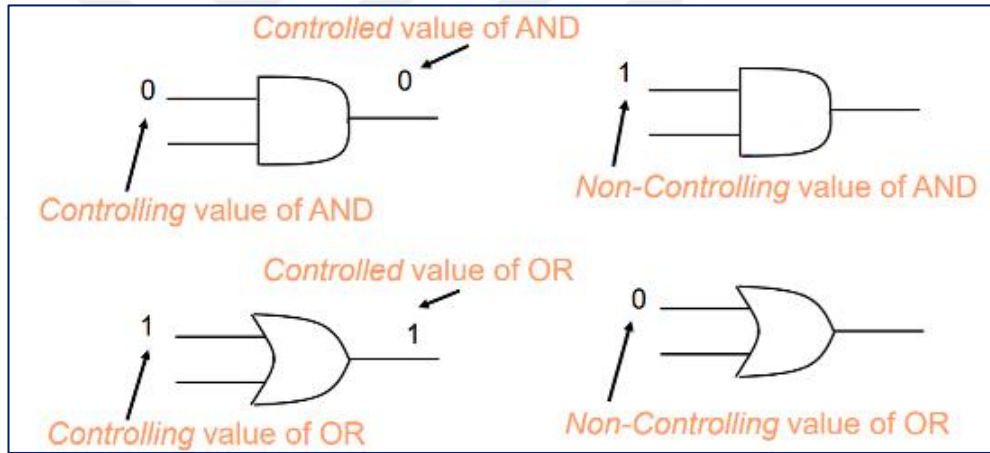
3.3.4. Yol Sentezleme

Şüpheli yol ve ilişkili yol olarak adlandırılan bu iki yolun devrede ayağa kaldırılabilir olması gerekmektedir. Ayağa kaldırılamayan yollar devrede ölçülen gecikme değerinin sorumlusu değildirler. Gecikme değerinin sorumlusu olması için bir yolun üzerinden sinyal yayılımı olması gerekmektedir. Üzerinden sinyal yayılımı olan yollara ayağa kaldırılabilir yol ya da sentezlenebilir (ing. sensitizable) yol denir.

3.3.4.1. Statik Yol Sentezleme Yapısı

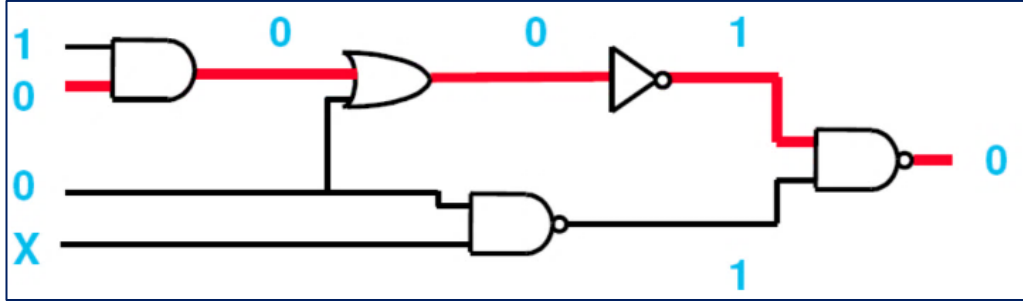
Literatürde, sentezlenebilen yollara doğru yol (ing.true path), sentezlenemeyen yollara yanlış yol (ing. false path) denir. Sentezlenebilirlik kontrolü için de literatürde iyi bilinen yöntemlerden biri olan “devre üzerinde kritik yollar için sentezleme” yöntemi [24] kullanılır. Bu tezde tatmin tabanlı (ing. SAT based) statik sentezleme yöntemi [25] kullanılarak, P_{susp} ve P_{corr} olarak seçilen yolların sentezlenip, sentezlenemediğini incelenmiştir.

Statik sentezleme, yol sentezleme yöntemlerinden bir tanesidir. Statik sentezleme yöntemi, yayılım sinyallerin varış zamanlarını göz ardı eden, gecikmeden bağımsız bir yöntem olduğu için "statik" olarak adlandırılır. Yol üzerindeki kapıların kontrol etmeyen yan girdileri (ing. non-controlling side inputs) ayarlayan bir girdi vektörü varsa, yol statik olarak sentezlenebilir.



Şekil 3.12: Kontrol Eden ve Kontrol Etmeyen Girdi Değerleri Örneği.

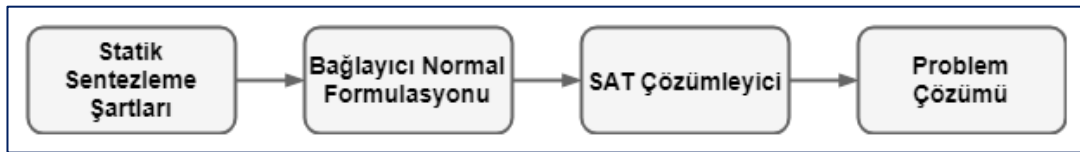
Kontrol eden değer (ing. controlling input) ve kontrol etmeyen değer (ing. noncontrolling input) kavramları yol sentezleme yöntemlerinde kullanılan kavramlardır. Bir lojik kapısının çıktısını tek başına belirleyen girdiye kontrol eden girdi denir. Bu durumda diğer girdilerin ne olduğunun önemi olmaz. Örneğin VE kapısı için kontrol eden girdi lojik-0 değeridir. Çünkü tek bir girdisi 0 olduğunda VE kapısının çıktısı da belli (lojik-0) olacaktır. Eğer bir yolda bulunan tüm lojik kapılarının yan girdileri kontrol etmeyen değerler almışsa bu durumda bu yoldan sinyal yayılımı olacak, yani bu yol gerçek yol olacaktır. İşte statik sentezleme bu koşulun sağlanmasıdır.



Şekil 3.13: Statik Yol Sentezleme Örneği Gösterimi.

Şekil 3.13'deki örnek üzerinde, sentezlenmek istenen yol daha kırmızı ile seçilmiş ve sırasıyla VE-VEYA-DEĞİL-VEDEĞİL (ing. AND-OR-NOT-NAND) kapılarından oluşan yoldur. VE kapısı için denetlenemeyen girdi değer 1'dir ve yan girdi değeri olarak verilmiştir. Denetlenen değer olarak, yayılımı sağlayacak 0 değeri verilmiştir. VE kapısının çıktı değerini, denetlenen değer olan 0 belirlemiştir. VEYA kapısının denetlenemeyen girdi değeri 0'dır. DEĞİL kapısının denetlenen veya denetlenemeyen girdi değeri yoktur. VEDEĞİL kapısının denetlenemeyen girdi değeri VE kapısı gibi 1'dir. Kırmızı renkle seçili yola ait kapıların değerlerine odaklanılır. Yola etki etmeyen değerler X ile gösterilir, hem 1 hem 0 olabileceği anlamına gelmektedir.

3.3.4.2. Statik Yol Sentezleme Denklemi



Şekil 3.14: Yol Sentezleme Algoritması Yaşam Döngüsü.

Seçili yol ve devrenin ifade edildiği çizge, sentezleme algoritmasına verilir. Seçili yol ve yolun çizgede bağlı olduğu kısımlar dikkate alınarak, mantıksal ifadelere uygun tanımlanmış, bağlayıcı normal formülasyonu (BNF) (ing. Conjunctive Normal Form) ile seçili yolu ifade eden bir denklem oluşturulur. Bu denklem mantıksal tatmin problemi (SAT) çözümleyiciye verilir. SAT çözümleyici

olarak minisat [2005/2006] kullanılmıştır. Çözümleyici denklemi optimize eder ve değişkenlere mantıksal (ing. boolean) değerler atayarak denklemin doğruluğunu sağlamaya çalışır. SAT problemi NP-complete bir problem olduğundan polinom zamanda denklemin optimal bir çözümüne ulaşmak mümkün değildir. SAT çözümleyiciler mantıksal denklemin doğruluğunu sağlayan çözümler ararlar, bunu sezgisel yöntemlerle yaptıklarından aynı denklemin farklı çalışmalar altında farklı doğruluk çözümleri elde edilebilir.

Tablo 3.1: Kapıların BNF İfadeleri.

Kapı	Denklem	BNF
VE	$o = a.b$	$(a'+b'+o) . (a+o') . (b+o')$
VEDEĞİL	$o = (a.b)'$	$(a'+b'+o') . (a+o) . (b+o)$
VEYA	$o = a+b$	$(a+b+o') . (a'+o) . (b'+o)$
VEYADEĞİL	$o = (a+b)'$	$(a+b+o) . (a'+o') . (b'+o')$
DEĞİL	$o = a'$	$(a+o) . (a'+o')$

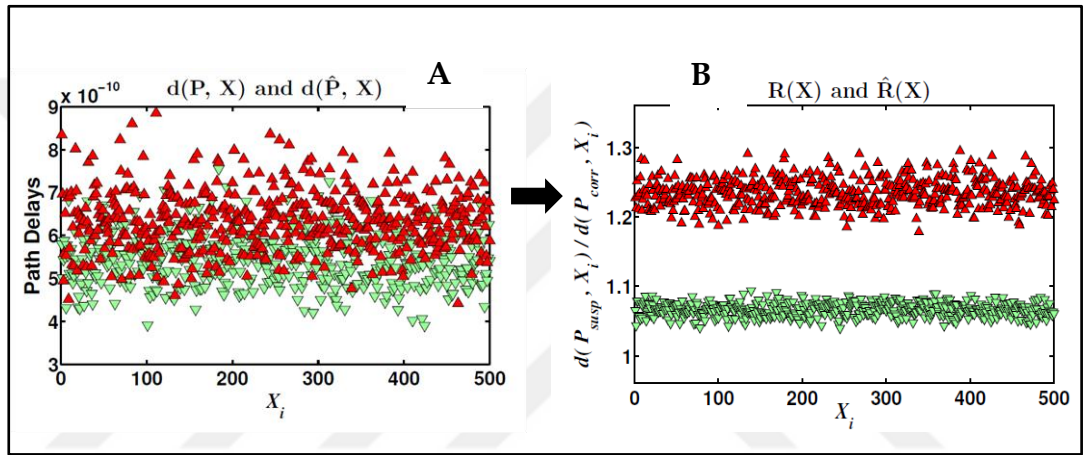
Tablo 3.1'deki BNF yapılarını incelersek, nokta(.) mantıksal VE kapısı, artı(+) mantıksal VEYA kapısıdır. Tırnak işareti (') mantıksal değil kapsıdır. Değişkenler a,b,o harfleriyle sembolize edilmektedir. BNF denkleminde, her bir parantez () sağlanması gereken şartlardan birini temsil etmektedir. BNF denklemindeki şartlarının tamamı 1 (doğru) olmalıdır ki denklemin sonucu 1 (doğru) çıksın. Denklemdaki değişkenlere mantıksal değerler olan 0 veya 1 atanarak, doğrulanıp doğrulanamadığı hesaplanır.

Her bir kapı tipi için tanımlı olan BNF denklemi kullanılır. Yolun sahip olduğu kapılar için BNF denklemlerini, hiçbir sıra gözetmeden, yan yana eklenir. Denklemlerdeki tüm şartlar VE kapısıyla bağlandığı için önemli olan sıra ya da öncelik değil, değişkenleri doğru adlandırmak ve değer atamaktır.

Devrenin BNF denklemine ek olarak, sentezlemek istenen yola ait kontrol şartları denkleme eklenir. Bu kontrol şartları belirli birincil girdiden belirli birincil çıktıya doğru yol üzerindeki kapıların girdilerine değerler atar. VE VEDEĞİL kapılarının yan girdileri bir, VEYA, VEYADEĞİL kapılarının yan girdileri sıfır olarak atanır. Bu koşullar sayesinde ilgili yolun yan girdileri sabit tutulmuş olur. Devrenin BNF denklemini ve yolun yan girdilerine ait kontrol şartlarının denklemini üzerinde SAT çözümleyici çalışır.

Çözümleyici denklemleri doğruladığında, seçili yolu ayağa kaldıracak tüm girdi değerlerinin ne olması gerektiği bulunmuş olur. Devrenin girdilerine bir değer verdiğimizde, çıktı olarak bir değer oluşmasını bekleriz. Devreye bu girdiler verilerek seçili yol ayağa kaldırılır ve böylece ölçülen gecikme değeri seçili yolun gecikmesi olur. Yol sentezleme sayesinde, devre üzerinde sadece seçili yolu ilgilendiren gecikme ölçümleri yapılmış olur.

3.3.5. DTA Tespiti



Şekil 3.15: DTA Tespitinde Amaçlanan Sonucun Gösterimi.

DTA tespitinde asıl amacımız Şekil 3.15’de **B** grafik sonucunu elde etmektir. **A** ile gösterilen şekilde iç içe geçmiş değerler yöntemimizi kullanmadan ilk ölçülebilen değerlerdi ve Bölüm 3.2.2’de anlatılmıştı. **B** ile gösterilen şekil, şüpheli yol gecikmesi ile ilişkili yol gecikmesinin oranından elde edilmiştir. DTA varlığını tespit etmemizi sağlayan şey, DTA içeren (kırmızı) ve DTA içermeyen (yeşil) değerlerin tüm örnekler için tamamen birbirinden ayrılmış olmasıdır. Şekil 3.15’de **B** grafiğin nasıl elde edildiği ve bu sayede DTA tespitinin nasıl yapılabildiği bu bölümde detaylı anlatılacaktır.

B grafiğinin elde edilebilmesi için ilk adım olarak, birbiriyle ilişkili yol gecikme değerlerine sahip yolların, devreden verimli bir şekilde nasıl seçildiği anlatılacaktır. Bu yolları devrede belirleyebilmek için, Şekil 3.16’da gösterilen **İkili Yol Seçme algoritması** kullanılmaktadır. Bu algoritma devre üzerinde DTA içerdiğinden şüphelenilen her bir bağlantı için tekrarlanarak çalışmaktadır (satır 1). Bölüm

3.3.2’de anlatılan, Şekil 3.7’de bulunan Şüpheli Yol Seçme algoritması kullanılmaktadır (satır 2). Bölüm 3.3.3’de anlatılan, Şekil 3.10’da bulunan İlişkili Yol Seçme algoritması kullanılmaktadır (satır 3).

İkili Yol Seçme - P_{susp} ve P_{corr} Seçimi

- 1: Her bir şüpheli kenar $< n_s, n_t >$ için
- 2: $P_{susp} = \text{Şüpheli Yol Seçme}$
- 3: $P_{corr} = \text{İlişkili Yol Seçme}$

Şekil 3.16: İkili Yol Seçme Algoritması.

- Şüpheli Yol Seçimi (satır 2): DTA içerdiğinden şüphelenilen bağlantıyı içinde barındıran birden çok yol olabilir. Devredeki her bir birincil girdiden birincil çıktıya olan ve içinde şüpheli kenarı barındıran en kısa yollar seçilir. Seçilenlerin tamamı şüpheli yol adayları (π_{susp}) kümesini oluşturur. Bu aday kümesi içinden, DTA ‘nın yol gecikmesine olan etkisini arttırmak için, en küçük gecikme değerine sahip yol seçilir ve şüpheli yol (P_{susp}) olarak adlandırılır. En küçük gecikme değerine sahip olan yolu seçmek, DTA’nın gecikmeye olan etkisini arttırdığından kısmen tespitini kolaylaştırır.
- İlişkili Yol Seçimi (satır 3): Devrede üssel sayıda yol olabilir. Bunlar içinden P_{susp} ile ilişkili başka bir yol bulunmalıdır. Ama ilişkili yol ararken devredeki tüm yolları inceleyerek değil, daha kısa bir yöntemle bu işi yapılmalıdır. Devrede var olan konuma bağlı korelasyon nedeniyle, şüpheli yolun kapılarının kullandığı karelere yakın olan diğer yollar, şüpheli yol ile daha ilişkilidir. Bu nedenle sadece şüpheli yola yakın kareleri kullananları incelemek işi hızlandıran ve daha doğru sonuçlara ulaşmamızı sağlayan bir yöntem olacaktır. Bu yöntemde, şüpheli yola yakın karelerdeki yolların kümesi, ilişkili yol aday kümesini (π_{susp}) oluşturacaktır. Bu aday kümesi devreden çıkartılarak ölçümleri yapıldıktan sonra, içlerinden şüpheli yol ile en yüksek ilişkiye sahip yol seçilir ve ilişkili yol (P_{corr}) olarak adlandırılır.

İkili yol seçme işlemi sonucunda üzerinde DTA olduğundan şüphelenilen her bir kenar için, P_{susp} ve P_{corr} yol ikilileri bulunmuş olur. Bu iki yolu bulma nedenimiz, varyasyona karşı benzer şekilde değişim gösteren bu yol ikililerinin gecikme

değerlerini oranlamaktır. Bu sayede elde edilen oran değerinde, varyasyon etkisi en aza indirilmiş ve DTA etkisi nedeniyle oluşabilecek değer farkları tespit edilmiş olur. İkili Yol Seçme algoritması, tasarımdan sonra çiplerin üretimden gelmesini beklerken yapılacak işlemlerdir. Şekil 3.15’de **B** kısmında da gösterilen DTA tespitini sağlama kısmı, DTA tespiti algoritması üzerinden anlatılacaktır. Bu kısım aynı zamanda çipler üretimden geldikten sonra, uygulanacak olan DTA tespitidir.

DTA Tespiti - P_{susp} üzerinde DTA Analizi

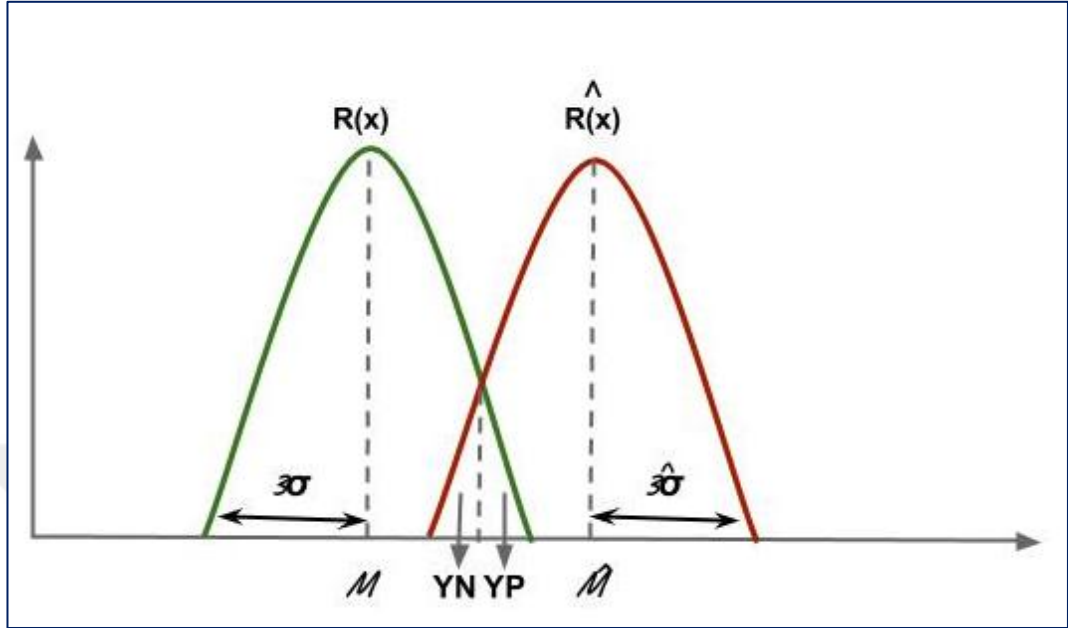
- 1: Her bir çip için
- 2: Her bir şüpheli kenar $\langle n_s, n_t \rangle$ için
- 3: $P_{susp}, P_{corr} \leq$ *İkili Yol Seçme* algoritmasını çalıştır.
- 4: $R(X) = d(P_{susp}, X) / d(P_{corr}, X)$
- 5: *pdf (R(X)) analizi*

Şekil 3.17: DTA Tespiti Algoritması.

Her bir çipin, her bir şüpheli kenarı için, *ikili yol seçme algoritması* çalıştırılarak P_{susp} ve P_{corr} yolları ve bu yolların varyasyon modeline göre hesaplanmış gecikme değerleri elde edilmiştir (satır 1-3). Bu iki yola ait gecikme ölçümleri oranlanır ve bu oran $R(X)$ olarak adlandırılır. Böylece $R(X)$ değeri varyasyon etkisinden arındırılmış olur (satır 4). DTA içermeyen şüpheli kenara ait $R(X)$ değeri ile DTA içeren şüpheli kenara ait $R(X)$ değerinin dağılım grafikleri oluşturulur ve bu grafikler üzerinde işlemler yapılarak DTA olup/olmadığı ve tespitinin ne kadar başarıyla yapıldığı söylenir (satır 5). Artık bir çipe ait şüpheli kenara DTA içerir veya içermez tanımı konulabilir. DTA tespitini hatasız (ing. zero error) yapabilmek için, dağılımların birbirleriyle kesişmeden, tamamen ayrılmasını sağlamak gerekmektedir.

Üzerinde ölçümler yaptığımız n tane örneğin bir kısmını DTA içermeyen, kalan kısmını ise DTA içeren olarak tanımlanmıştır. Böylece n örnek içinden kaç tanesinin DTA tespitinde yanlış sınıflandırıldığı gözlemlenmiş olur. Yanlış sınıflandırılan örneklerden, yanlış pozitif (YP) olanlar DTA içermeyen ama test sonuçlarında DTA içerdiği söylenen örneklerdir. Bu örneklerin DTA içerdiğinin kabul edilmesi elimizdeki güvenilir devre sayısını azaltmış olur. Yanlış negatif (YN) olanlar DTA içeren ama test sonuçlarında DTA içermediği belirtilen örneklerdir. Bu örneklerin kullanımına devam edilmemesi ve güvenilir devrelerin arasından çıkartılmaları

gerekmektedir. Bu nedenle, yanlış sınıflandırılan örneklerden YN örneklerinin tespit edilmesi hayati önem taşımaktadır.



Şekil 3.18: DTA İçeren ve İçermeyen Gecikme Oranlarının Dağılımı.

$R(x)$ değerleri her ne kadar varyasyon etkisinden arındırılmaya çalışılsa da içlerinde küçük ölçüde varyasyonu etkisini barındırabilirler ve bu etki nedeniyle birbirlerinden tamamen ayrılmasını sağlamak biraz daha zorlaşabilir. Buradan da doğru sınıflandırma için, $R(X)$ değerinin önemini, yani birbiriyle yüksek derecede ilişkili yollar seçmenin önemini bir kez daha görmekteyiz.

Geliştirilen DTA tespit yöntemi yani yol seçme algoritması, DTA içerdiğinden şüphelenilen her bir bağlantı için çalıştırılır. Tüm çipin analiz edilebilmesi için şüpheli kenar seçiminin nasıl olacağı çok önemlidir. Şüpheli kenar sayısı en fazla çizgedeki bağlantı sayısı kadar olabilir. Çizgedeki bağlantı sayısı en fazla, düğüm sayısının karesi kadar olabilir. n düğümlü çizge için bağlantı sayısı $O(n^2)$ olur. ISCAS'85 devreleri için, bağlantı sayısı en fazla düğüm sayısının 2 katı kadar olmaktadır. Yani yol seçme algoritmasının $O(2n)$ kez çalışması gerekmektedir.

4. SONUÇLAR

4.1. Deneysel Kurulumlar

Transistör parametrelerinden en önemlileri L ve V_{th} olarak tanımlanabilir [28]. Varyasyon modelinde kullanılan parametreler, ITRS raporunda belirlenen değerler kullanılarak tanımlanmışlardır [26]. 45nm teknolojisi için, varyasyon modelinde kullanılan L değeri $3\sigma/\mu$ %12, V_{th} değeri $3\sigma/\mu$ %20 dir. Toplam varyasyonun yarısı çip içi varyasyondan, kalan yarısı da çipler arası varyasyon bileşeninden oluşmaktadır [29].

Deneyslerimizde, literatürde yaygın olarak da kullanılan ISCAS'85 devreleri kullanılmıştır [27]. Devreler NanGate 45nm açık standart hücre kütüphanesi ile sentezlendi [30]. Transistör seviyesinde simülasyonlar için NgSpice [21] simülatörünün üzerinde değişiklik yapılmış versiyonu kullanıldı. Testler her biri 6 çekirdekli ve 2-GHz frekansında çalışan, 2 tane Xeon E5-2620 işlemci ve 24GB RAM bulunan bir iş istasyonunda çalıştırıldı.

Yapılan tüm testlerde, her test devresinden 1000 çip kopyası olduğu varsayılmıştır. Bunlardan 500 tanesi üzerine DTA eklenmiş ve gecikme değerleri tekrar hesaplanmıştır. Kalan 500 tanesi üzerinde herhangi bir değişiklik yapılmamış, onlar DTA içermeyenler olarak kalmışlardır.

Netlist devreler üzerinde, DTA ekleme gibi değişiklikler yapan yapılar ise C programlama dili kullanılarak oluşturulmuştur. Çizge devreler üzerinde yapılan tüm çalışmalar ve ölçümler MATLAB üzerinde gerçekleştirilmiştir. Tezde Bölüm 3'de bahsedilen tüm algoritma ve yöntemler çizge devre üzerinde çalıştığından MATLAB ortamında yazılmışlardır.

DTA tespit yönteminin ana parçası olan İkili Yol Seçme algoritmasının çalışma süresi 25 saniyedir. ISCAS'85 devrelerinden en çok düğüm ve kenar içeren c7552'nin içerdiği kenar sayısı 6997 Düğüm sayısı 3512'dir. Devredeki tüm kenarlar şüpheli kabul edildiğinde çalışma süresi, $2 * 3512 * 25sn = 48,6$ saat olmaktadır. Bu da yaklaşık 2 güne denk gelmektedir. Devrenin kaç kopyası için ölçüm yapılacağı, çalışma süresini belirleyen önemli etkenlerden biridir. Bizim tüm testlerimizde 1000 devre için ölçüm yapılmıştır. Bu işlemler devrenin tasarımı biter bitmez, üretimden gelinceye kadar olan sürede yapılmaktadır.

DTA Tespiti algoritmasının, çalışma süresi tamamen test cihazına bağlıdır. Çalışma süresini belirleyecek faktörlerden biri de yol gecikme ölçümü için verilen süredir. Her bir şüpheli kenar için, 4 * yol gecikme süresi kadar çalışacağı söylenebilir. Aynı anda birden çok yola ait gecikme değeri ölçen bir test cihazıyla bu süre çok daha kısaldır.

Yöntemimizin çalışma süresi olan 25 saniye çeşitli yöntemlerle kısaltılabilir. Bunlardan ilk akla gelen, MATLAB yerine daha hızlı çalışan programlama araçlarında yöntemi oluşturmaktır. İkinci hızlandırma önerisi, yöntemi paralel çalışabilecek hale getirmektir. Üçüncü hızlandırma önerisi, yöntemi paralelleştirdikten sonra üzerinde biraz daha değişiklikler yüksek düzede paralelleştirme ile GPU'da çalışacak hale getirmektir.

4.2. Testler

Test sonuçları olan Şekil 4.1 – Şekil 4.27 arasındaki grafiklerin tamamında transistör seviyesinde SPICE ölçümleri kullanılmıştır. SLE ile gecikme hesabı yol seçme yöntemi (Şekil 3.16) sırasında kullanılmıştır. 9 tane olan ISCAS'85 devrelerinin her biri için, devre üzerinde rastgele belirlenmiş 3 tane şüpheli kenar (SE1, SE2, SE3) seçilmiştir. Toplamda 27 farklı test yapılmıştır. Her bir test için 1000 örnek üzerinden 500 tanesine, şekil 3.5'de gösterilen DTA eklenmiştir. Örnek olarak, *Exp(c432, SE2)* c432 devresinin 2. Şüpheli kenarı üzerinde yapılan testleri ifade etmektedir. Tablo 4.1 de gösterilen 27 testin her biri için, **İkili Yol Seçme** algoritması kullanılmıştır. P_{susp} ve P_{corr} yolları bulunmuş ve yolların her birinin 1000 örnek için gecikme değerleri elde edilmiştir. Daha sonra, **DTA Tespiti** algoritması çalıştırılarak devreler üzerinde yol ikililerinin gecikme değerleri için SPICE ölçümleri yapılmıştır. P_{susp} gecikme değerinin P_{corr} gecikme değerine oranlanmasıyla elde edilen $R(X)$ değeri hesaplanmıştır.

Tablo 4.1: Test Sonuçlarının Detaylı Gösterimi.

		π_{susp} Boyutu	π_{corr} Boyutu		Yanlış Sınıflandırılan Örnek Sayısı P_{susp}	Yanlış Sınıflandırılan Örnek Sayısı $R(x)$
			Aynı Kare	Komşu Kare	(1000 örnek için)	(1000 örnek için)
C432	SE1	54	2	11	13	0
	SE2	5	1	30	42	0
	SE3	180	1	22	9	0
C499	SE1	28	3	18	151	17
	SE2	32	2	56	110	0
	SE3	40	1	2	145	0
C880	SE1	6	4	135	123	0
	SE2	16	5	73	36	0
	SE3	26	2	3	44	0
C1355	SE1	192	109	-	198	0
	SE2	105	7	25	12	0
	SE3	106	6	7	8	0
C1908	SE1	3	3	15	66	0
	SE2	252	576	-	170	0
	SE3	25	26	-	196	0
C2670	SE1	6	26	-	85	0
	SE2	16	7	64	119	0
	SE3	66	612	-	289	0
C3540	SE1	20	34	-	244	0
	SE2	245	63	-	216	0
	SE3	2	32	-	95	0
C5315	SE1	2	12	-	45	0
	SE2	24	14	-	51	0
	SE3	1	51	-	29	0
C7552	SE1	9	27	-	126	0
	SE2	6	179	-	195	0
	SE3	4	12	-	127	0

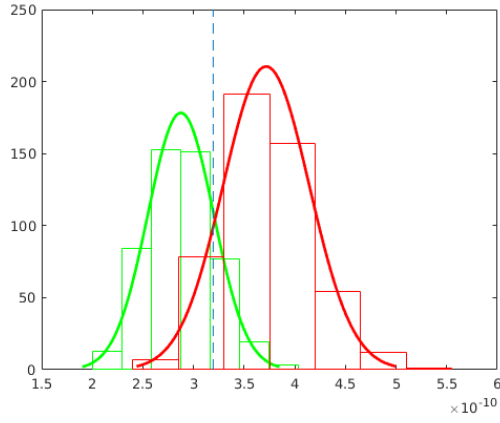
Tablo 4.1 27 teste ait tüm ölçümleri içermektedir. Tablonun ilk sütunu olan π_{susp} Boyutu, şüpheli yol adayları kümesindeki yolların sayısıdır. İkinci sütun olan Aynı kare, P_{susp} ile tamamen aynı karelerden geçen kapılardan oluşan, ilişkili yol adayları kümesindeki yolların sayısıdır. Üçüncü sütun olan Komşu Kare, P_{susp} ile komşu karelerden geçen kapılardan oluşan, ilişkili yol adayları kümesindeki yolların sayısıdır. İlişkili yol seçme Bölüm 3.3.4'te anlatılmıştır. Komşu karelerden geçen ilişkili yol arama yapısı, sadece P_{susp} ile birebir aynı karelerden geçen ilişkili yolların sayısı yetersizse veya korelasyon katsayıları küçükse kullanılır. 4. Sütunda bazı kısımların kesik çizgi (-) ile gösterilmesinin nedeni komşu karelerden geçen yollara ihtiyaç kalmamasıdır. Dördüncü ve beşinci sütun, 1000 örnek içinden yanlış sınıflandırılmış örneklerin sayısıdır. Her iki sütunda DTA tespiti için farklı yöntemler kullanılmıştır. Dördüncü sütunda, en kısa gecikmeye sahip P_{susp} gecikme değeri sınıflandırma için kullanılır. Eğer en kısa yol değil de daha uzun yollar üzerinde tespit yapılsaydı, yanlış sınıflandırma sayılarının daha yüksek olacağı akılda tutulmalıdır. Dördüncü sütun temel olarak [17] deki yöntemi ifade etmektedir. Beşinci ve son sütunda, bu çalışmada önerilen şekilde iki yola ait gecikme oranı sınıflandırma için kullanılır.

Örneğin Şekil 4.1 Exp(c7552, S1) ait histogramları göstermektedir. Bu iki histogramdan A, P_{susp} gecikme değeri dağılımını ifade eder. Yeşil renkli olanlar DTA içermeyen 500 örneği, kırmızı renkli olanlar DTA eklenmiş 500 örneği ifade eder. Dağılım sonucunda DTA içeren (kırmızı) ve içermeyen (yeşil) gecikme değeri dağılımları birbirine karışmıştır. Yani 1000 örnek içinden DTA içerip/içermemeleri yanlış sınıflandırılmış 126 örnek bulunmaktadır. Bu sınıflandırma olabilecek en iyi sınıflandırmadır. Gerçekte bu 1000 gecikme değerini vererek iki ayırım dağılım elde edilmesi mümkün değildir. Değerler çok iç içe geçtiği için, gerçekte tek bir dağılım çizilebilecek ve sınıflandırma yapılamayacaktır. Biz en kısa gecikmeye sahip P_{susp} kullanarak DTA tespiti yöntemini, olabilecek en iyi sınıflandırma ile gösterdik. Şekil 4.1'de B histogramı, P_{susp} ve P_{corr} yollarının gecikme değerlerinin oran dağılımını göstermektedir. $R(X)$ oran dağılımına göre DTA içeren ve içermeyen tüm örnekler hatasız tespit edilmişlerdir. $R(X)$ dağılımları birbirlerinden tamamen ayrılmıştır. B histogramı tez yöntemimiz olan, varyasyondan benzer şekilde etkilenen yol ikililerinin oranını kullanarak DTA tespit eden yöntemdir.

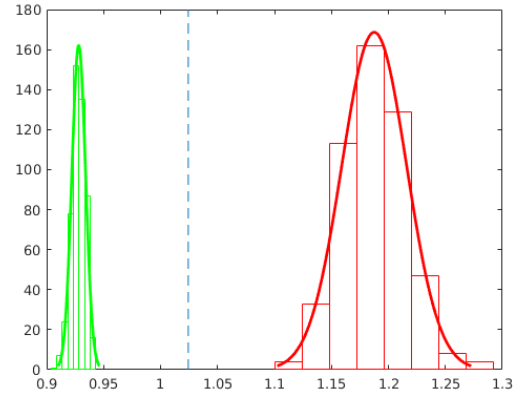
Histogram şekillerinden A ile ifade edilenler gecikme değer dağılımlarıdır. Dağılımlarda bulunan boydan kesikli çizgi, DTA içeren/içermeyen sınıflandırmasında, en az hatayı gösterecek şekilde, dağılımların kesişiminden hesaplanır. Bu kesikli çizgi ve DTA içermeyen gecikme dağılımları arasında kalan kısım yanlış negatif (YN), kesikli çizgi ve DTA içeren gecikme dağılımları arasında kalan kısım yanlış pozitif (YP) hata değerleridir. Exp(c7552, S1) için 126 olan yanlış sınıflandırma sayısı YP ve YN sayılarının toplamıdır.

Histogramları Şekil 4.22'de gösterilen Exp(c499, SE1) haricinde tüm deneylerde birbirine çok benzer histogram grafikleri elde edilmiştir. Sınıflandırma için yol gecikme oranı ($R(X)$) kullanılmış olmasına rağmen, Exp(c499, SE1)'in yanlış sınıflandırılmış örneklerin sayısı 17'dir. Fakat en kötü sınıflandırma sonucumuz olan bu test için bile, π_{susp} kümesinden en kısa yoldan başka bir yol seçtiğimizde, yanlış sınıflandırılmış örneklerin sayısı sıfır olmaktadır. Çıkarılan iki yolun korelasyonu, ilişkili yol seçme algoritmasının 6. satırına göre önceden hesaplanmıştır. Bu nedenle, tüm r_i değerleri küçük, yani sınıflandırma için yetersiz bulunursa, ilk önce P_{susp} seçimini değiştirmek daha iyidir. Algoritmanın bu küçük değişikliği ile, Exp(c499, SE1) testi yanlış sınıflandırılmış örnekler vermeden, diğer testler gibi hatasız sınıflandırma yapabilir.

Şekil 4.4, 4.20, 4.25 ve 4.27 de B histogramlarında, DTA içermeyen (yeşil) devrelere ait oran dağılımı sabit 1 olarak elde edilmiştir. Bunun nedeni, elde edilen P_{susp} ve P_{corr} yollarının birbirleriyle tamamen aynı karelerde bulunan, aynı kapı tiplerinin aynı kapı dizilimine sahip olmalarıdır. Örneğin; $P_{corr} = \langle PI_1, n_{400}, n_{496}, n_{712}, PO_{10} \rangle$, $P_{susp} = \langle PI_2, n_{600}, n_{645}, n_{698}, PO_{21} \rangle$ olsun. n_{400} ve n_{600} düğümlerinin ifade ettiği kapıların ikisi de 2 girişli VE kapısıdır. İkisi de devrede 6x8 karesinde bulunmaktadır. n_{496} ve n_{645} için, ikisi de 4 girişli VEDEĞİL kapısıdır, 4x6 karesinde bulunmaktadırlar. n_{712} ve n_{698} için, ikisi de 2 girişli VEYA kapısıdır, 6x7 karesinde bulunmaktadırlar.

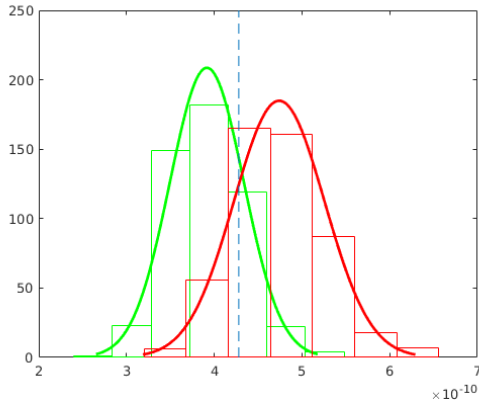


Gecikme Değerine Göre (A)

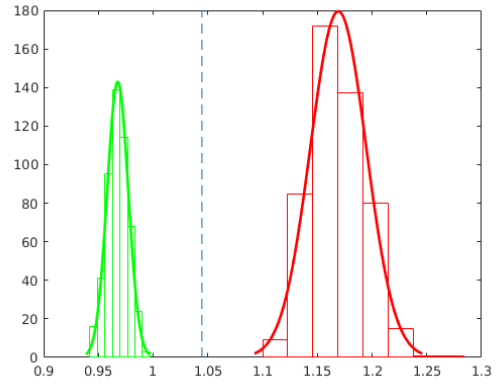


Oran Değerine Göre (B)

Şekil 4.1: c7552 SE1 – DTA Sınıflandırması Dağılımları.

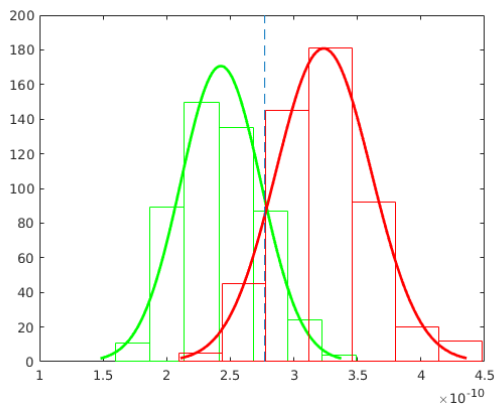


Gecikme Değerine Göre (A)

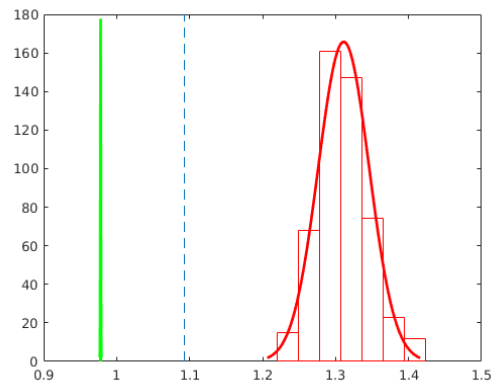


Oran Değerine Göre (B)

Şekil 4.2: c7552 SE2 – DTA Sınıflandırması Dağılımları.

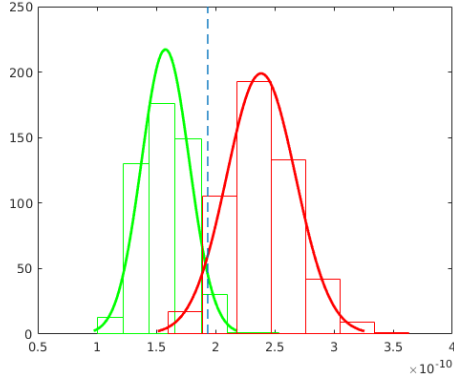


Gecikme Değerine Göre (A)

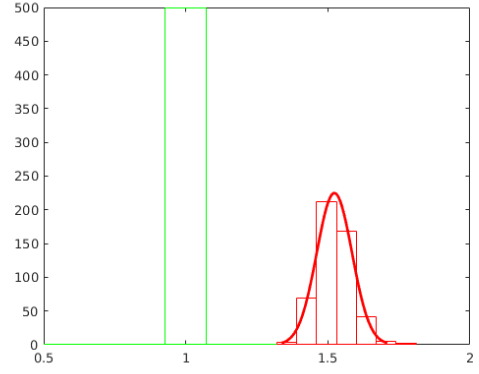


Oran Değerine Göre (B)

Şekil 4.3 : c7552 SE3 – DTA Sınıflandırması Dağılımları.

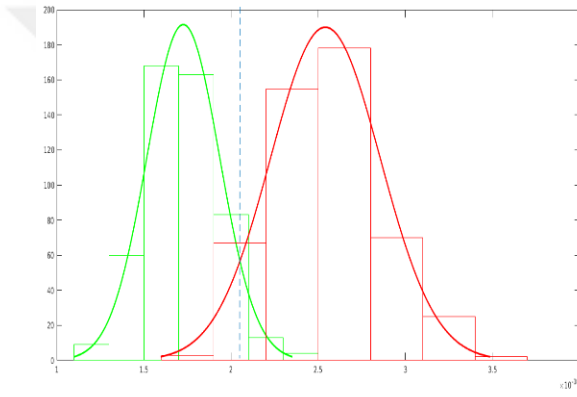


Gecikme Değerine Göre (A)

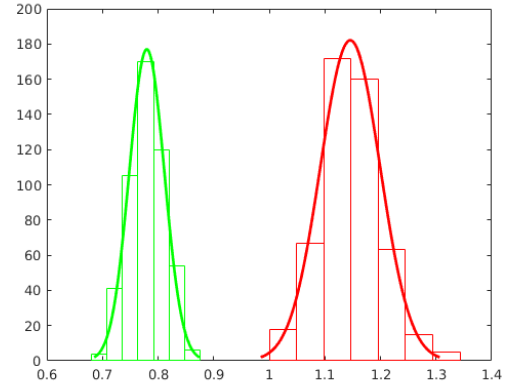


Oran Değerine Göre (B)

Şekil 4.4: c5315 SE1 – DTA Sınıflandırması Dağılımları.

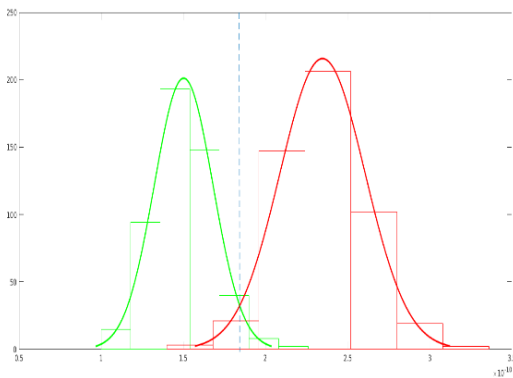


Gecikme Değerine Göre (A)

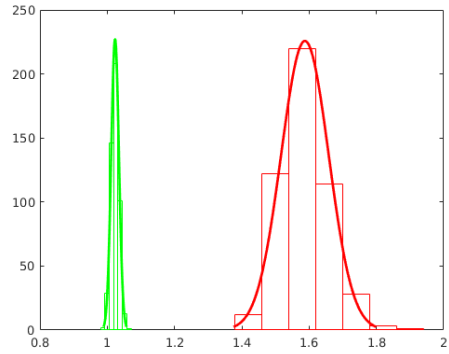


Oran Değerine Göre (B)

Şekil 4.5: c5315 SE2 – DTA Sınıflandırması Dağılımları.

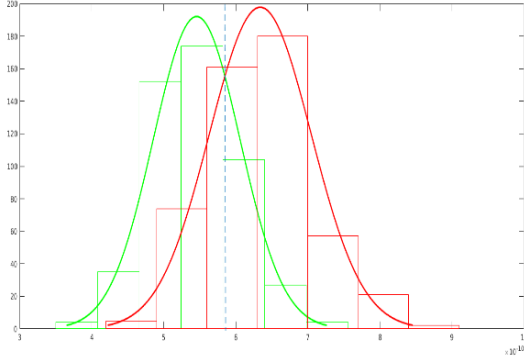


Gecikme Değerine Göre (A)

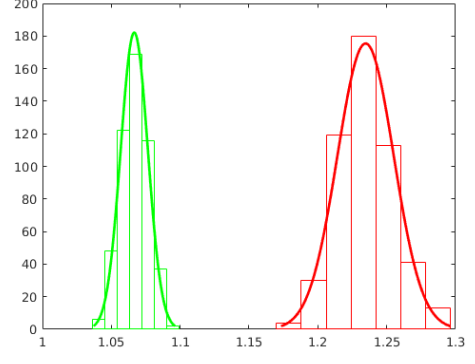


Oran Değerine Göre (B)

Şekil 4.6 : c5315 SE3 – DTA Sınıflandırması Dağılımları.

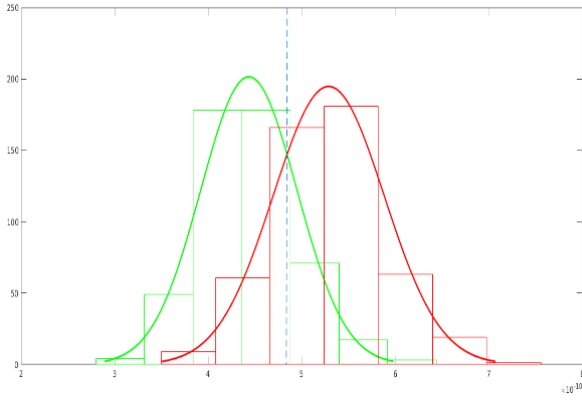


Gecikme Değerine Göre (A)

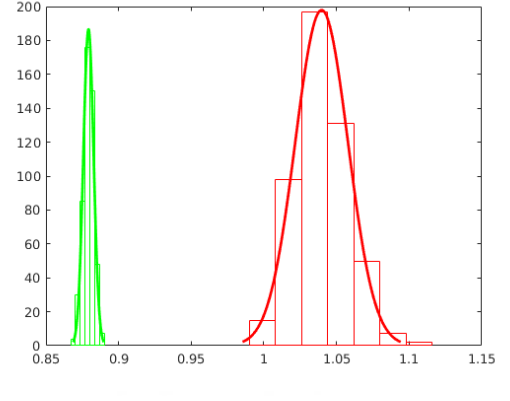


Oran Değerine Göre (B)

Şekil 4.7: c3540 SE1 – DTA Sınıflandırması Dağılımları.

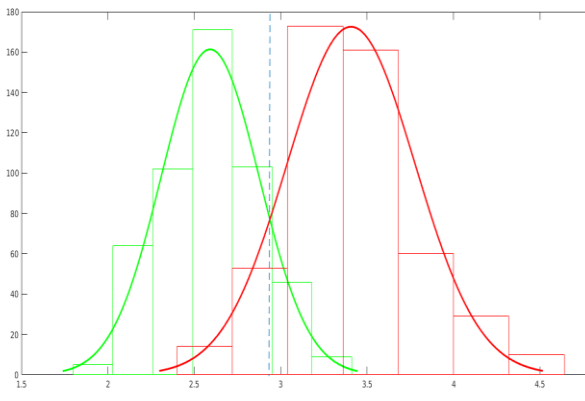


Gecikme Değerine Göre (A)

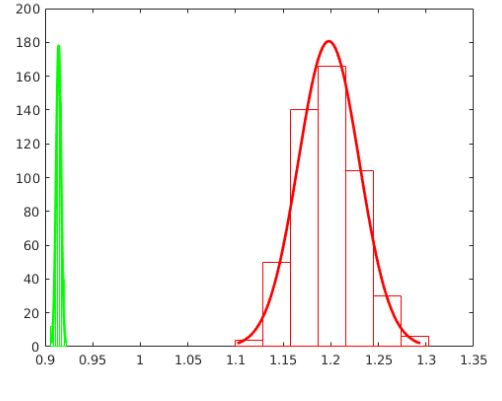


Oran Değerine Göre (B)

Şekil 4.8: c3540 SE2 – DTA Sınıflandırması Dağılımları.

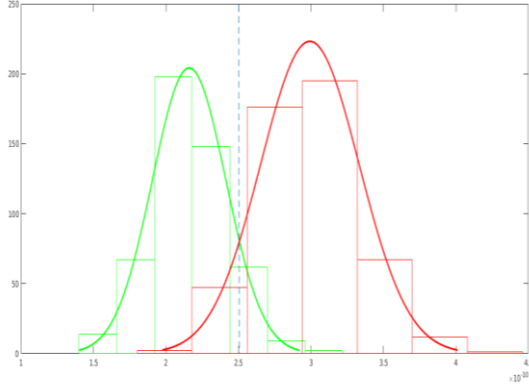


Gecikme Değerine Göre (A)

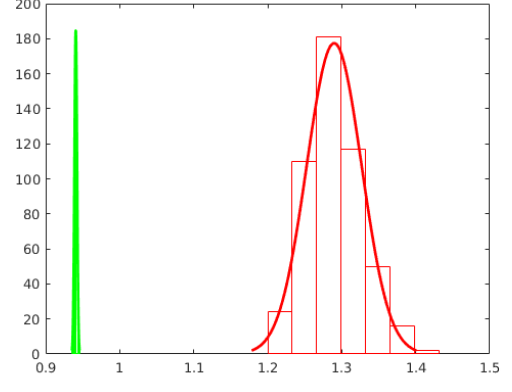


Oran Değerine Göre (B)

Şekil 4.9: c3540 SE3 – DTA Sınıflandırması Dağılımları.

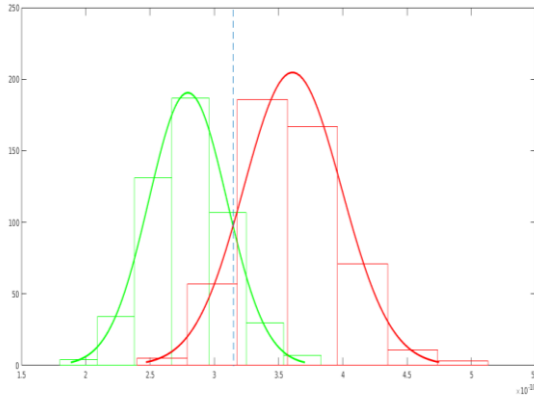


Gecikme Değerine Göre (A)

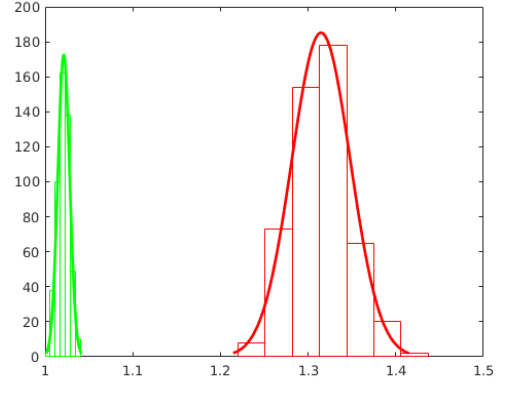


Oran Değerine Göre (B)

Şekil 4.10: c2670 SE1 – DTA Sınıflandırması Dağılımları.

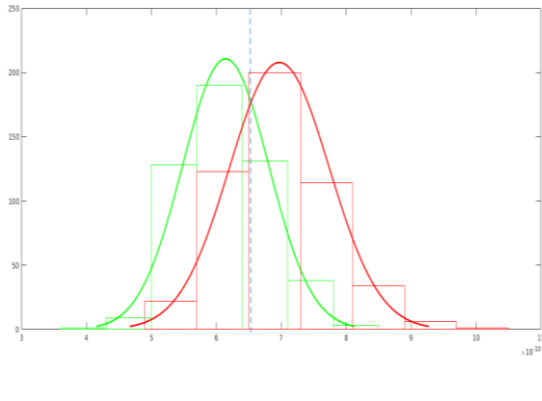


Gecikme Değerine Göre (A)

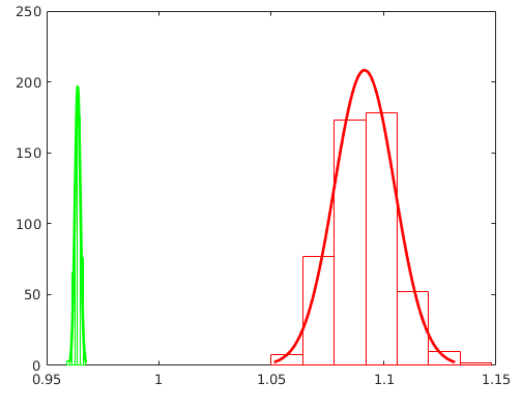


Oran Değerine Göre (B)

Şekil 4.11: c2670 SE2 – DTA Sınıflandırması Dağılımları.

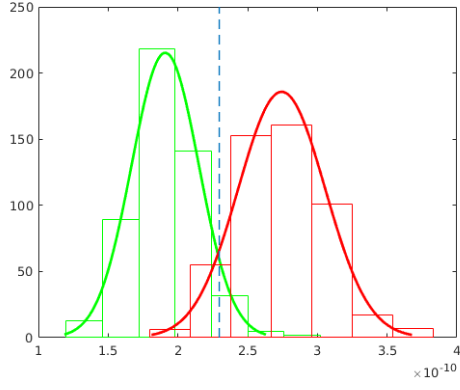


Gecikme Değerine Göre (A)

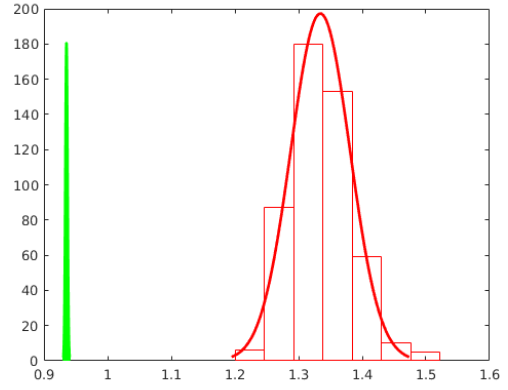


Oran Değerine Göre (B)

Şekil 4.12: c2670 SE3 – DTA Sınıflandırması Dağılımları.

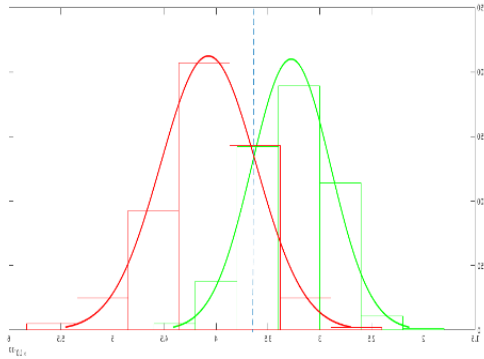


Gecikme Değerine Göre (A)

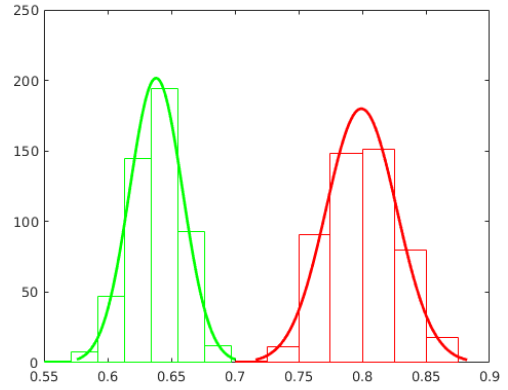


Oran Değerine Göre (B)

Şekil 4.13: c1908 SE1 – DTA Sınıflandırması Dağılımları.

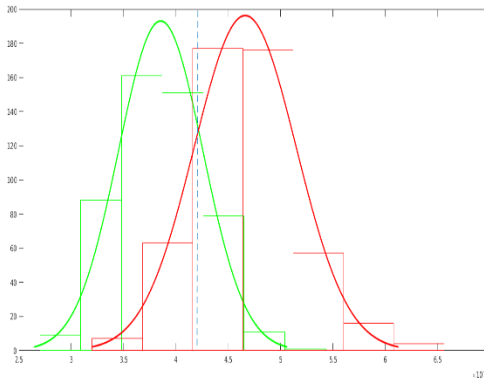


Gecikme Değerine Göre (A)

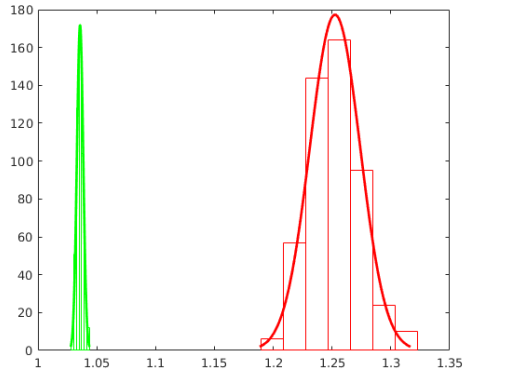


Oran Değerine Göre (B)

Şekil 4.14: c1908 SE2 – DTA Sınıflandırması Dağılımları.

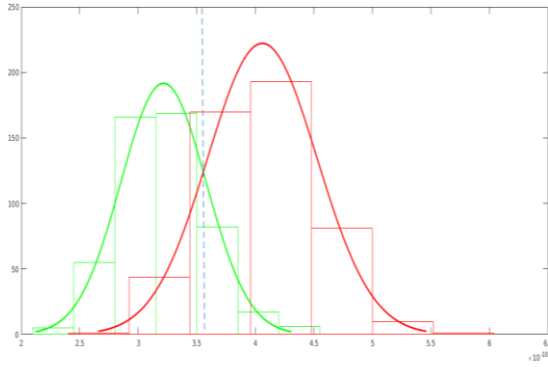


Gecikme Değerine Göre (A)

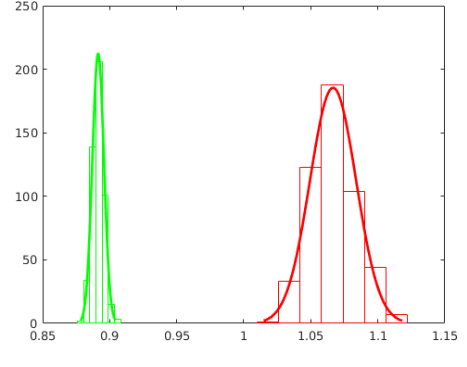


Oran Değerine Göre (B)

Şekil 4.15: c1908 SE3 – DTA Sınıflandırması Dağılımları.

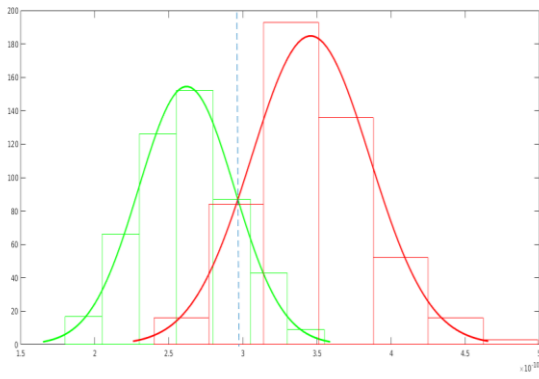


Gecikme Değerine Göre (A)

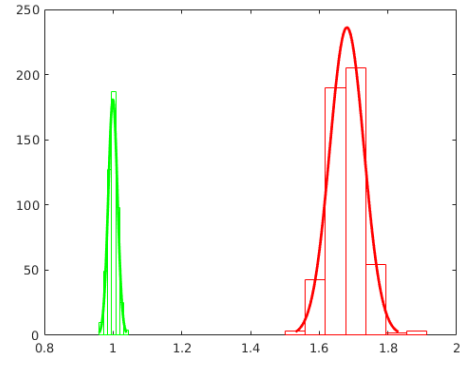


Oran Değerine Göre (B)

Şekil 4.16: c1355 SE1 – DTA Sınıflandırması Dağılımları.

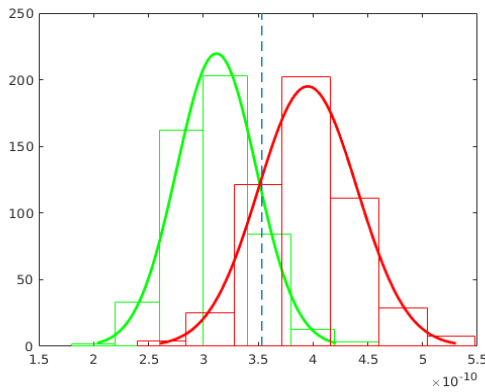


Gecikme Değerine Göre (A)

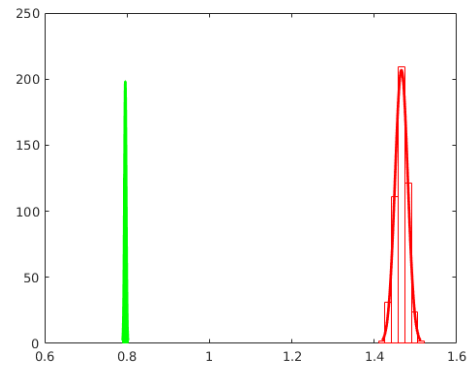


Oran Değerine Göre (B)

Şekil 4.17: c1355 SE2 – DTA Sınıflandırması Dağılımları.

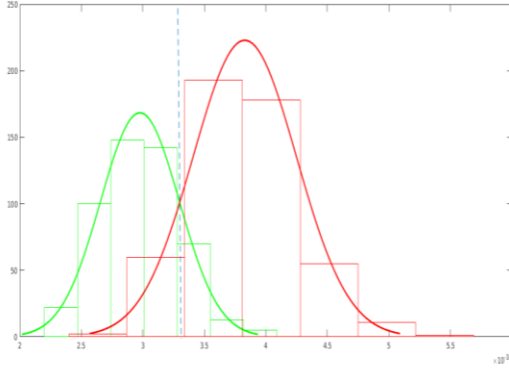


Gecikme Değerine Göre (A)

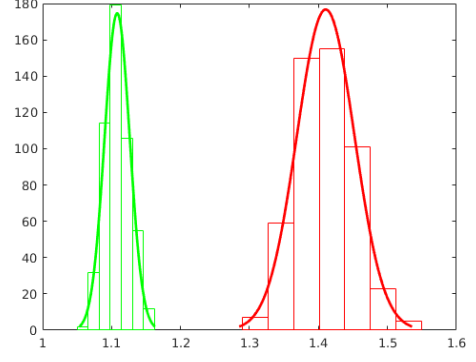


Oran Değerine Göre (B)

Şekil 4.18: c1355 SE3 – DTA Sınıflandırması Dağılımları.

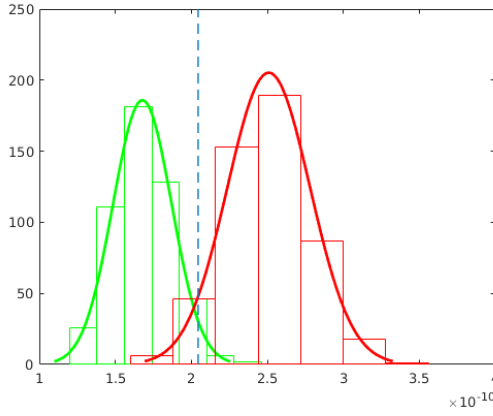


Gecikme Değerine Göre (A)

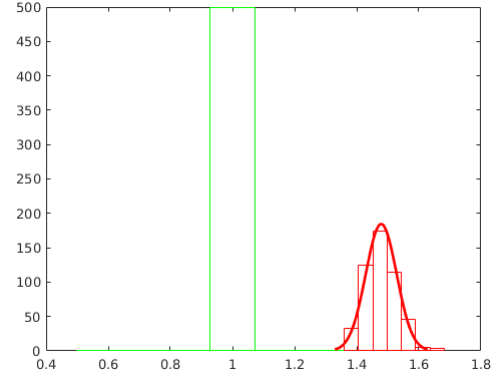


Oran Değerine Göre (B)

Şekil 4.19: c880 SE1 – DTA Sınıflandırması Dağılımları.

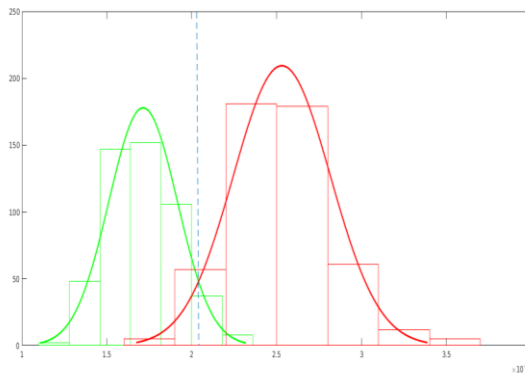


Gecikme Değerine Göre (A)

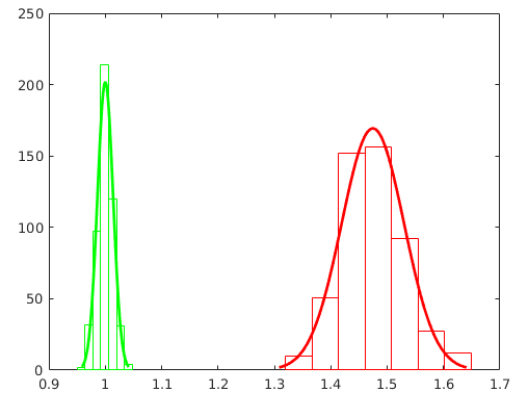


Oran Değerine Göre (B)

Şekil 4.20: c880 SE2 – DTA Sınıflandırması Dağılımları.

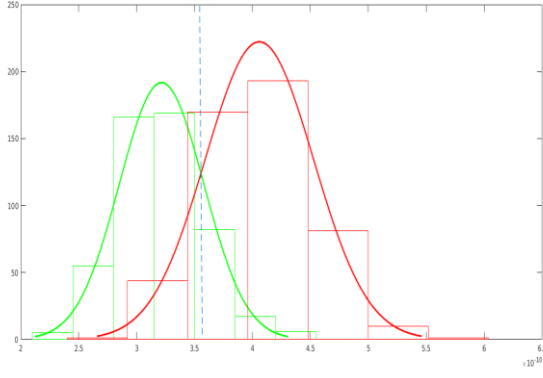


Gecikme Değerine Göre (A)

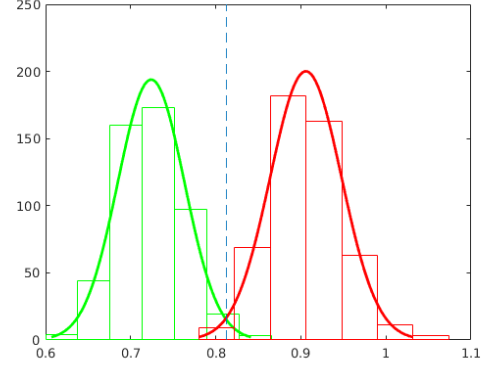


Oran Değerine Göre (B)

Şekil 4.21: c880 SE3 – DTA Sınıflandırması Dağılımları.

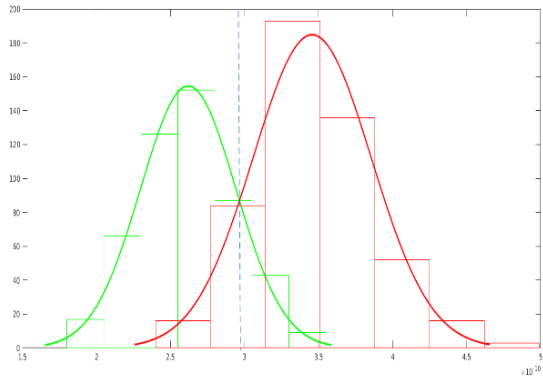


Gecikme Değerine Göre (A)

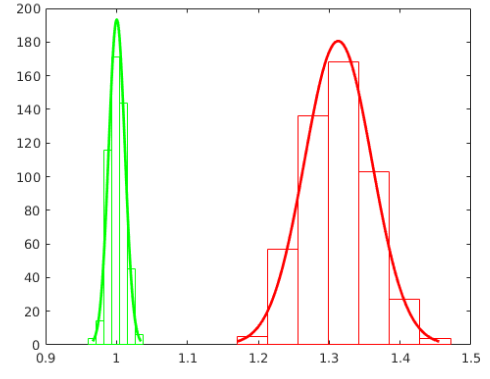


Oran Değerine Göre (B)

Şekil 4.22: c499 SE1 – DTA Sınıflandırması Dağılımları.

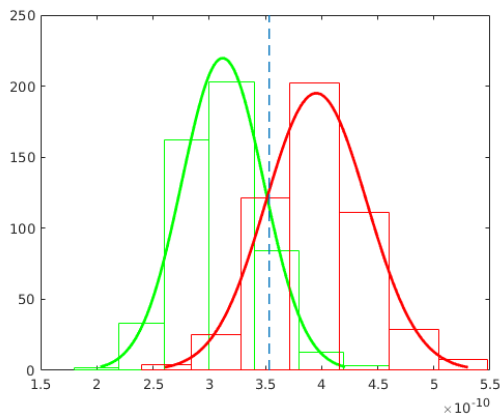


Gecikme Değerine Göre (A)

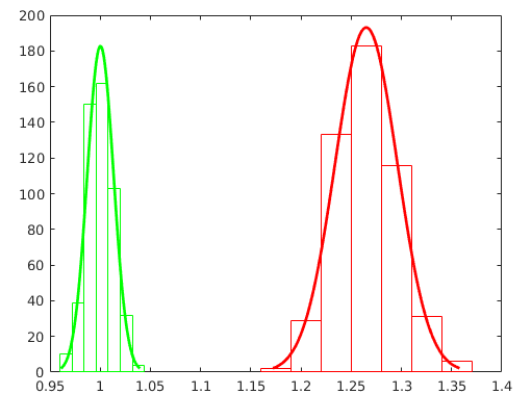


Oran Değerine Göre (B)

Şekil 4.23: c499 SE2 – DTA Sınıflandırması Dağılımları.

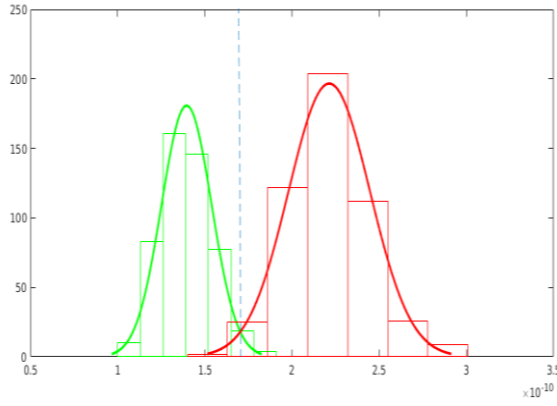


Gecikme Değerine Göre (A)

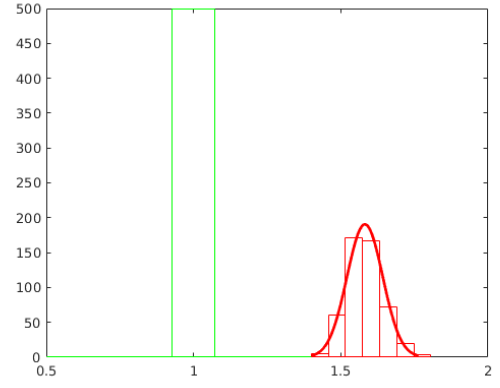


Oran Değerine Göre (B)

Şekil 4.24: c499 SE3 – DTA Sınıflandırması Dağılımları.

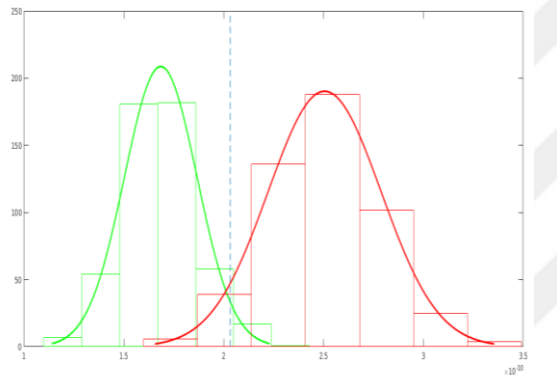


Gecikme Değerine Göre (A)

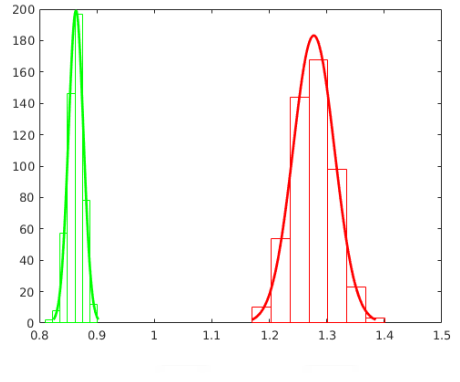


Oran Değerine Göre (B)

Şekil 4.25: c432 SE1 – DTA Sınıflandırması Dağılımları.

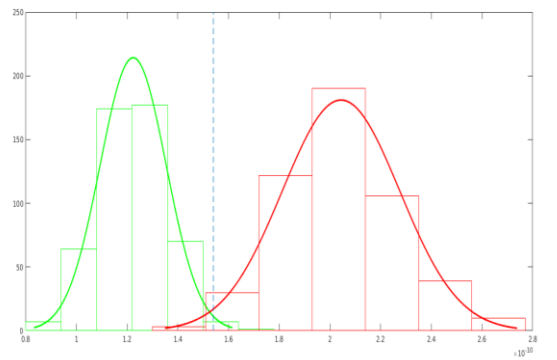


Gecikme Değerine Göre (A)

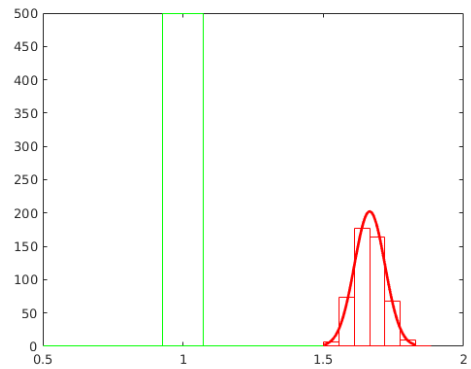


Oran Değerine Göre (B)

Şekil 4.26: c432 SE2 – DTA Sınıflandırması Dağılımları.



Gecikme Değerine Göre (A)



Oran Değerine Göre (B)

Şekil 4.27: c432 SE3 – DTA Sınıflandırması Dağılımları.

4.3. Gelecek Çalışmalar için Öneriler

- Varyasyon Modeli Aşaması için: Gerçeğe daha yakın varyasyon modeli elde etmek için, varyasyon modeli denklemine rastgele parametre eklenmelidir. Böylece birebir aynı karelerde bulunan iki aynı tip kapının bile gecikme değerleri birbirinden farklı olacaktır. İkinci bir iyileştirme olarak, konuma bağlı rastgele değişkenleri tanımlayan dördün ağaç yapısının katman sayısı artırılmalıdır. Katman sayısı arttıkça, devredeki kare parça sayısı artmaktadır. Böylece daha hassas bir varyasyon modeli elde edilecektir. Fakat bu kare sayısındaki artış, bir dezavantajı da beraberinde getirmektedir. O da kare parça başına düşen kapı sayısının azalmasıdır. Bu sayı azaldığı için ilişkili yol bulma yöntemi zora düşebilir.
- İlişkili Yol Bulma Aşaması için: Korelasyon değeri hesabı için *Corr* fonksiyonu değil de $R(X)$ değeri dağılımlarının standart sapma ve medyan değerlerine göre belirlenen $SD = \max((\hat{M} - M) - (3\delta + 3\hat{\delta}))$ formülü kullanılabilir. Formüldeki SD değeri R değer dağılımlarının birbirleri ile aralarındaki mesafedir. SD değeri sıfırdan büyük ise, dağılımlar arasında ayırım sağlanmış demektir. Ayırımı sağlayan herhangi bir yol, ilişkili yol olarak seçilebilir.
- Yol Sentezleme Aşaması için: Statik sentezleme yöntemi tek başına yeterli gelmemektedir. Statik sentezleme yöntemleri, gecikme değerinin yol üzerinde yayılımıyla ilgilenmezler. Bu yüzden ölçüm yapmak istediğimiz yol yerine başka bir yola ait sentezleme değerlerini de elde etmiş olabilir. Ama devreden seçilen kritik yollar gibi uzun yollar için böyle bir sorun oluşmamaktadır. Uzun yollar, devrede kendilerinden daha büyük gecikme değerine kimsenin sahip olmadığı yollardır. Bu nedenle o yolun çıkışındaki beklenen değişimden, uzun yol sorumludur. Devrede kısa yollar üzerinden değil de uzun yollar üzerinden işlem yaparak, statik sentezleme yönteminin istenen yol için sentezleme yapması garanti altına alınmış olur. Ama tüm yol seçme algoritmasının değiştirilmesi ve şüpheli kenar belirlemeye yönelik incelemeler yapılması gerekmektedir.

KAYNAKLAR

- [1] Huffmire T., Irvine C., Nguyen T. D., Levin T., Kastner R. , (2010) , “Handbook of FPGA Design Security”, 1, Springer
- [2] Web 1, (2013), https://www.seconsult.com/fxdata/secons/prod/temedia/advisories.txt/201301240_Barracuda_Appliances_Backdoor_wo_poc_v10.txt/, (Eriřim Tarihi: 28/11/2016)
- [3] Web 2, (2012), [http://intelligence.house.gov/sites/intelligence.house.gov/files/documents/Huawei-ZTE%20Investigative%20Report%20\(FINAL\).pdf/](http://intelligence.house.gov/sites/intelligence.house.gov/files/documents/Huawei-ZTE%20Investigative%20Report%20(FINAL).pdf/), (Eriřim Tarihi: 16/01/2018)
- [4] Skorobogatov S., Woods C., (2012), “Breakthrough silicon scanning discovers backdoor in military chip”, Cryptographic Hardware and Embedded Systems Workshop (CHES), 23-40, Leuven, Belgium, 9-12 September
- [5] Web 3, (2012), http://www.tubitak.gov.tr/sites/default/files/tubitak_2012_faaliyet_raporu_web.pdf/, (Eriřim Tarihi: 16/01/2018)
- [6] Web 4, (2015), <https://spectrum.ieee.org/semiconductors/design/stopping-hardware-trojans-in-their-tracks/>, (Eriřim Tarihi: 16/01/2018)
- [7] Chakraborty R.S., Narasimhan S., Bhunia S., (2009), “Hardware Trojan: Threats and Emerging Solutions”, IEEE International High Level Design Validation and Test Workshop, San Francisco/CA/ USA, 4-6 November
- [8] Tehranipoor M., Koushanfar F., (2010), “A survey of hardware trojan taxonomy and detection”, IEEE Design Test of Computers, 27(1), 10–25
- [9] Wang X., Tehranipoor M., Plusquellic J., (2008), “Detecting Malicious Inclusions in Secure Hardware: Challenges and Solutions”, IEEE International Workshop on Hardware-Oriented Security and Trust, 15-19, Anaheim/CA/USA, 9 June
- [10] Karri R., Rajendran J., Rosenfeld K., Tehranipoor M., (2010), “Trustworthy Hardware: Identifying and Classifying Hardware Trojans”, Computer, 43(10), 39-46
- [11] Bhunia S., Hsiao M.S., Banga M., Narasimhan S., (2014), “Hardware Trojan Attacks: Threat Analysis and Countermeasures”, Proceedings of the IEEE, 102(8) , 1229 - 1247

- [12] Amin B., Taghi S.M., Afshin A.M, (2016), “Trojan Counteraction in Hardware: A Survey and New Taxonomy”, Indian Journal of Science and Technology, 9(18), 12-21
- [13] Zhang J., Yuan F., Wei L., Liu Y., Xu Q., (2015), “VeriTrust: Verification for Hardware Trust”, IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 34(7), 1148 - 1161
- [14] Narasimhan S., Du D., Chakraborty R. S., Paul S., Wolff F. G., Papachristou C. A., Roy K., Bhunia S, (2013), “Hardware trojan detection by multiple-parameter side-channel analysis”, IEEE Transactions on Computers, 62(11), 2183–2195
- [15] Jin Y., Makris Y., (2008), “Hardware trojan detection using path delay fingerprint”, IEEE International Workshop on Hardware-Oriented Security and Trust, 51–57, Anaheim/CA/USA, 9 June
- [16] Li J., Lach.J., (2008), “At-speed delay characterization for ic authentication and trojan horse detection”, IEEE International Workshop on Hardware-Oriented Security and Trust, 8–14, Anaheim/CA/USA, 9 June
- [17] Cha B., Gupta S. K., (2013), “Trojan detection via delay measurements: A new approach to select paths and vectors to maximize effectiveness and minimize cost”, Design, Automation Test in Europe Conference Exhibition (DATE), 1265–1270, Grenoble/France, 18-22 March
- [18] Zhang L., Chang C. H., (2014), “Hardware trojan detection with linear regression based gate-level characterization”, IEEE Asia Pacific Conference on Circuits and Systems (APCCAS), 256–259, Ishigaki/Japan, 17-20 November
- [19] Agarwal A., Blaauw D., Zolotov V., (2003), “Statistical timing analysis for intra-die process variations with spatial correlations”, International Conference on Computer Aided Design (ICCAD), 900–907, San Jose/CA/USA, 9-13 November
- [20] Bayrakci A. A., (2015), “Stochastic logical effort as a variation aware delay model to estimate timing yield” Integration, the VLSI Journal, 48(C) ,101– 108
- [21] Web 5, (2018), <http://ngspice.sourceforge.net/> (Erişim Tarihi: 16/01/2018)
- [22] Sharifi E., Mohammadiasl K., Havasi M., Yazdani A., (2015), “Performance analysis of Hardware Trojan detection methods”, International Journal of Open Information Technologies, 3(5), 39-44
- [23] Dijkstra E.W., (1959), “A note on two problems m connexion with graphs”, Numerische Mathematik, 1(1), 269-271

- [24]Chen H. C., Du D. H.-C., (1993), “Path sensitization in critical path problem [logic circuit design]”, IEEE Council on Electronic Design Automation, 12(2), 196–207
- [25]Guerra e Silva L., Marques-Silva J., Silveira L. M., Sakallah K. A., (2002), “Satisfiability models and algorithms for circuit delay computation”, ACM Transactions on Design Automation of Electronic Systems (TODAES), 7(1), 137–158
- [26] Web 6, (2011), <http://www.itrs.net/> , (Erişim Tarihi: 16/01/2018)
- [27]Brglez F., Fujiwara H., (1985), “A neutral netlist of 10 combinational benchmark circuits”, IEEE International Symposium on Circuits and Systems, 695-704, Kyoto/Japan, 5-7 June
- [28]Sylvester D., Agarwal K., Shah S., (2008), “Variability in nanometer cmos: Impact, analysis, and minimization”, Integration the VLSI Journal, 41(3), 319–339
- [29]Cong J., Gupta P., Lee J., (2010), “Evaluating statistical power optimization”, IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 29(11), 1750–1762
- [30]Web 7, (2010), <http://www.nangate.com/> , (Erişim Tarihi: 16/01/2018)
- [31]Chang H., Sapatnekar S.S., (2005), “Statistical timing analysis under spatial correlations”, IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 24(9), 1467 – 1482
- [32]Web 8 , (2014), https://dac.com/sites/default/files/files/51DAC_web_book.pdf/, (Erişim Tarihi: 16/01/2018)
- [33]Esirci F.N, Bayrakçı A.A, (2017), “Hardware Trojan Detection Based on Correlated Path Delays in Defiance of Variations with Spatial Correlations”, Design, Automation & Test in Europe Conference & Exhibition (DATE), 163-168, Lausanne/Switzerland, 17-31 March
- [34]Eggersgluß S., Fey G., Glowatz A., Hapke F., Schloeffel J., Drechsler R., (2010), “MONSOON: SAT-based ATPG for Path Delay Fault Using Multiple-Valued Logics”, Journal of Electronic Testing, 26(3), 307-322
- [35]Bell C.W, (2013), “A Multi-Parameter Functional Side Channel Analysis Method for Hardware Trojan Detection in Untrusted FPGA Bitstreams”, Thesis (MSc), University of South Florida
- [36]Rajendran J., Gavas E., Jimenez J., Padman V., Karri R., (2010), “Towards a comprehensive and systematic classification of hardware Trojans”, IEEE International Symposium on Circuits and Systems, 1871–1874, Paris/France, 3 May-2 June

ÖZGEÇMİŞ

Fatma Nur Esirci 1993 yılında Kırıkkale’de doğdu. 2010 yılında başladığı Gebze Teknik Üniversitesi Mühendislik Fakültesi Bilgisayar Mühendisliği Bölümünü 2015 yılında başarıyla tamamlayarak aynı yıl yüksek lisans eğitimine Gebze Teknik Üniversitesi Fen Bilimleri Enstitüsü Bilgisayar Mühendisliği Anabilim Dalında başladı. Yüksek lisans eğitimi boyunca TÜBİTAK bursiyeri olarak çalışmalarını sürdürdü. Gebze Teknik Üniversitesi Bilgisayar Mühendisliği Bölümünde, 2017 yılından bu yana araştırma görevlisi olarak çalışmaktadır.

