

154462

**YILDIZ TEKNİK ÜNİVERSİTESİ  
FEN BİLİMLERİ ENSTİTÜSÜ**

**LOJİK FONKSİYONLARIN GEÇİŞ  
TRANSİSTÖRLERİ BAZLI GERÇEKLENMESİ İÇİN  
GÖRSEL BİR YAZILIM GELİŞTİRİLMESİ**

Elektronik ve Haberleşme Müh. M.Cem DİKBAŞ

FBE Elektronik ve Haberleşme Anabilim Dalı Elektronik Programında  
Hazırlanan

**YÜKSEK LİSANS TEZİ**

Tez Danışmanı : Doç. Dr. Tülay YILDIRIM

Prof. Dr. Galip Cansever

Prof. Dr. Oruç Bilgiç

*[Handwritten signatures of Prof. Dr. Tülay Yıldırım, Prof. Dr. Galip Cansever, and Prof. Dr. Oruç Bilgiç]*

**İSTANBUL, 2004**

# İÇİNDEKİLER

|                                                                       | Sayfa |
|-----------------------------------------------------------------------|-------|
| SİMGE LİSTESİ.....                                                    | iv    |
| KISALTMA LİSTESİ.....                                                 | v     |
| ŞEKİL LİSTESİ.....                                                    | vi    |
| ÇİZELGE LİSTESİ.....                                                  | viii  |
| ÖNSÖZ.....                                                            | ix    |
| ÖZET.....                                                             | x     |
| ABSTRACT.....                                                         | xi    |
| 1. GİRİŞ.....                                                         | 1     |
| 1.1 Elektronik Entegre Devre Tasarım Disiplinleri.....                | 1     |
| 1.2 Lojik Stilleri Hakkında Genel Bilgiler.....                       | 3     |
| 1.3 Çalışmanın Amacı ve Kapsamı.....                                  | 4     |
| 2. GEÇİŞ TRANSİSTÖR LOJİK HAKKINDA GENEL BİLGİLER.....                | 6     |
| 2.1 Geçiş Transistör Çeşitleri.....                                   | 7     |
| 2.1.1 CPL-TG.....                                                     | 7     |
| 2.1.2 DPL.....                                                        | 7     |
| 2.1.3 DVL.....                                                        | 8     |
| 2.1.4 CPL.....                                                        | 8     |
| 2.1.5 SPL.....                                                        | 9     |
| 2.1.6 PPL.....                                                        | 9     |
| 2.2 CMOS Transmisyon Kapıları (Geçiş Kapıları).....                   | 10    |
| 3. SAYISAL DEVRE TASARIMININ MODELLENMESİ.....                        | 17    |
| 3.1 İkili Karar Diyagramları (BDD's).....                             | 18    |
| 3.1.1 BDD Genel Yapısı.....                                           | 18    |
| 3.1.2 Shannon Açılımı.....                                            | 20    |
| 3.1.3 BDD İndirgeme Kuralları.....                                    | 21    |
| 3.2 BDD Kullanarak Sayısal Devre Modellenmesi.....                    | 23    |
| 3.2.1 123 Karar Diyagramı.....                                        | 24    |
| 3.2.2 CPL-TG Yapısı İçin Genelleştirilmiş Karar Diyagramı Modeli..... | 35    |
| 4. PROGRAMIN GELİŞTİRİLMESİ VE GENEL YAPISI.....                      | 40    |
| 5. UYGULAMALAR.....                                                   | 47    |
| 5.1 Lojik Ve Kapısının Gerçeklenmesi.....                             | 47    |
| 5.2 Lojik Veya Kapısının Gerçeklenmesi.....                           | 49    |

|                          |                                         |    |
|--------------------------|-----------------------------------------|----|
| 5.3                      | Keyfi Bir Fonksiyon Gerçeklenmesi ..... | 51 |
| 5.4                      | İki Bit Toplayıcı .....                 | 55 |
| 6.                       | SONUÇ .....                             | 61 |
| KAYNAKLAR.....           |                                         | 62 |
| İNTERNET KAYNAKLARI..... |                                         | 63 |
| ÖZGEÇMİŞ .....           |                                         | 64 |



## SİMGE LİSTESİ

|              |                                    |
|--------------|------------------------------------|
| $k_n$        | NMOS geiş iletkenlięi parametresi |
| $k_p$        | PMOS geiş iletkenlięi parametresi |
| $I_D$        | Drain akımı                        |
| $I_{DS}$     | Drain-Source akımı                 |
| $I_{SD}$     | Source-Drain akımı                 |
| $R_{eq}$     | Eşdeęer diren                     |
| $V_{CC}$     | Pozitif besleme gerilimi           |
| $V_{DD(dd)}$ | Pozitif besleme gerilimi           |
| $V_{DS}$     | Drain-Source gerilimi              |
| $V_{GS}$     | Gate-Source gerilimi               |
| $V_{IH}$     | Minimum giriş gerilim seviyesi     |
| $V_{IL}$     | Maksimum giriş gerilim seviyesi    |
| $V_{OH}$     | Minimum ıkış gerilim seviyesi     |
| $V_{OL}$     | Maksimum ıkış gerilim seviyesi    |
| $V_T$        | Eşik gerilimi                      |
| $V_{th}$     | Eşik gerilimi                      |



## KISALTMA LİSTESİ

|        |                                                     |
|--------|-----------------------------------------------------|
| AND    | Lojik Ve Kapısı                                     |
| BDD    | Binary Decision Diagram                             |
| BDT    | Binary Decision Tree                                |
| CPL    | Complementary Pass Transistor Logic                 |
| CPL-TG | Complementary Pass Transistor Logic with Pass Gates |
| CVSL   | Cascade Voltage Switch Logic                        |
| DPL    | Double Pass Transistor Logic                        |
| DTL    | Diode Transistor Logic                              |
| DVL    | Dual Value Logic                                    |
| ECL    | Emitter Coupled Logic                               |
| EEPL   | Energy Economised Pass Transistor Logic             |
| EVBDD  | Edge Valued Binary Decision Diagram                 |
| MTBDD  | Multi Terminal Binary Decision Diagram              |
| NAND   | Lojik Tüve Kapısı                                   |
| NOR    | Lojik Tüveya Kapısı                                 |
| OBDD   | Ordered Binary Decision Diagram                     |
| OR     | Lojik Veya Kapısı                                   |
| PPL    | Push Pull Pass Transistor Logic                     |
| PTL    | Pass Transistor Logic                               |
| ROBDD  | Reduced and Ordered Binary Decision Diagram         |
| RTL    | Resistor Transistor Logic                           |
| SPL    | Single Ended Pass Transistor Logic                  |
| SRPL   | Swing Restored Pass Transistor Logic                |
| TG     | Transmission Gate                                   |
| TSPC   | True Single Phase Clocked                           |
| VLSI   | Very Large Scale Integrated                         |
| XOR    | Lojik Ayrıcalı Veya Kapısı                          |

## ŞEKİL LİSTESİ

|                                                                                                                                                                                      |    |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|
| Şekil 1.1 Sayısal işaretler için gürültü marjı (Leblebici, 1996) .....                                                                                                               | 2  |
| Şekil 2.1 Geçiş transistör lojik xor kapısı (Munteanu, 2001).....                                                                                                                    | 6  |
| Şekil 2.2 CPL-TG ile xor lojik kapısı tasarımı (Munteanu, 2001).....                                                                                                                 | 7  |
| Şekil 2.3 DPL ile xor lojik kapısı tasarımı (Munteanu, 2001) .....                                                                                                                   | 7  |
| Şekil 2.4 DPL ve DVL ile lojik nand kapısı (Munteanu, 2001).....                                                                                                                     | 8  |
| Şekil 2.5 (a) CPL lojik nand2 kapısı , (b) CPL lojik nor2 kapısı (Munteanu, 2001) .....                                                                                              | 8  |
| Şekil 2.6 CPL lojik xor kapısı (Munteanu, 2001) .....                                                                                                                                | 9  |
| Şekil 2.7 SPL ile lojik xor kapısı (Munteanu, 2001).....                                                                                                                             | 9  |
| Şekil 2.8 PPL ile lojik xor ve nxor kapıları (Munteanu, 2001).....                                                                                                                   | 10 |
| Şekil 2.9 CMOS transmisyon kapısı gösterim şekilleri (Leblebici, 1996).....                                                                                                          | 10 |
| Şekil 2.10 TG'nin bias koşulları ve karşılık düşen çalışma bölgeleri (Leblebici, 1996) .....                                                                                         | 11 |
| Şekil 2.11 TG eşdeğer direnç eğrisi (Leblebici, 1996) .....                                                                                                                          | 13 |
| Şekil 2.12 TG'nin eşdeğer modeli (Leblebici, 1996).....                                                                                                                              | 14 |
| Şekil 2.13 TG'ler ile gerçekleştirilen iki girişli çoğullayıcı devresi (Leblebici, 1996).....                                                                                        | 14 |
| Şekil 2.14 TG yapıları kullanan ve sekiz transistör ile gerçekleştirilen lojik xor kapısı<br>(Leblebici, 1996) .....                                                                 | 15 |
| Şekil 2.15 TG yapısı kullanan altı transistör ile gerçekleştirilen lojik xor kapısı (Leblebici, 1996).....                                                                           | 15 |
| Şekil 2.16 $F=AB+A'C'+AB'C$ lojik fonksiyonu gerçekleştirilmesi (Leblebici, 1996) .....                                                                                              | 16 |
| Şekil 3.1 Doğruluk tablosu ile verilen lojik fonksiyon ve ona ilişkin ve BDD gösterimi [2].                                                                                          | 19 |
| Şekil 3.2 Doğruluk tablosu verilen bir lojik fonksiyon ve ilişkin BDT (Bryant, 1992).....                                                                                            | 19 |
| Şekil 3.3 (a)Terminal düğümlerinin indirgenmeleri, (b)Terminal olmayan düğümlerin<br>indirgenmeleri, (c)Terminal olmayan gereksiz düğümlerin yok edilmesi<br>(Bryant, 1992).....     | 20 |
| Şekil 3.4 $f = ac+bc$ fonksiyonuna ait BDD haritalaması.....                                                                                                                         | 21 |
| Şekil 3.5 Birinci indirgeme kuralının gösterimi.....                                                                                                                                 | 22 |
| Şekil 3.6 İkinci indirgeme kuralının gösterimi.....                                                                                                                                  | 22 |
| Şekil 3.7 Temel lojik kapılarının BDD gösterimleri [1] .....                                                                                                                         | 23 |
| Şekil 3.8 123 karar diyagramı ve geçiş transistörü ile gerçekleştirilmesi (Jaekel, 1998).....                                                                                        | 24 |
| Şekil 3.9 123 karar diyagramı gösterimi (Jaekel, 1998) .....                                                                                                                         | 24 |
| Şekil 3.10 Liste gösterimi akışı (Jaekel, 1998) .....                                                                                                                                | 27 |
| Şekil 3.11 Çocuk düğüm oluşturma (Jaekel, 1998) .....                                                                                                                                | 29 |
| Şekil 3.12 Yolların birleştirilmesi 1 (Jaekel, 1998) .....                                                                                                                           | 30 |
| Şekil 3.13 Yolların birleştirilmesi 2 (Jaekel, 1998) .....                                                                                                                           | 30 |
| Şekil 3.14 Terminal olmayan düğümün, terminal düğüme dönüştürülmesi (Jaekel, 1998).....                                                                                              | 31 |
| Şekil 3.15 Alt kümelerin birleştirilmesi .....                                                                                                                                       | 32 |
| Şekil 3.16 Beş değişkenli, 00001001000000111111100011111001 ile kodlanmış lojik<br>fonksiyona ilişkin, 123 karar diyagramı oluşturma ve indirgeme basamakları<br>(Jaekel, 1998)..... | 34 |
| Şekil 3.17 $F=a'b'c+ab'+abc'$ lojik fonksiyonuna ait doğruluk tablosu ve kodlama stratejisi<br>(Avcı ve Yıldırım, 2003) .....                                                        | 37 |
| Şekil 3.18 (3.8)'de verilen lojik fonksiyonuna ve fonksiyona ilişkin elde edilen 10001101<br>kodu ve ikil karar diyagramı yapısı (Avcı ve Yıldırım, 2003) .....                      | 37 |
| Şekil 3.19 $F=a'b'c+ab'+abc'$ lojik fonksiyonun CPL-TG devre yapısı.....                                                                                                             | 38 |
| Şekil 4.1 Program Genel Akış Diyagramı.....                                                                                                                                          | 40 |
| Şekil 4.2 Fonksiyon girişi ve kod oluşturma paneli .....                                                                                                                             | 42 |
| Şekil 4.3 Lojik Fonksiyon Giriş Paneli .....                                                                                                                                         | 43 |
| Şekil 4.4 Lojik fonksiyona ait kodun oluşturulmasını gösteren ekran çıktısı .....                                                                                                    | 43 |
| Şekil 4.5 BDD haritalamasını gösteren ekran çıktısı .....                                                                                                                            | 44 |
| Şekil 4.6 Geçiş transistör devre modelini gösteren ekran çıktısı .....                                                                                                               | 45 |

|                                                                                                                                                                                 |    |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|
| Şekil 4.7 Çizim kumanda paneli .....                                                                                                                                            | 45 |
| Şekil 5.1 $F=ab$ (ve) fonksiyonuna ilişkin kod tablosu.....                                                                                                                     | 47 |
| Şekil 5.2 $F=ab$ (ve) fonksiyonuna ilişkin BDD gösterimi.....                                                                                                                   | 47 |
| Şekil 5.3 $F=ab$ (ve) fonksiyonuna ilişkin geçiş transistör lojigi devre yapısı .....                                                                                           | 48 |
| Şekil 5.4 $F=ab$ (ve) fonksiyonuna ilişkin, kaynakları da içeren devre yapısı .....                                                                                             | 48 |
| Şekil 5.5 Şekil 5.4'deki devreye ilişkin simülasyon sonucu, (a) giriş 1, (b) giriş 2, (c) çıkış..                                                                               | 49 |
| Şekil 5.6 $F=a'b+ab'+ab$ (veya) fonksiyonuna ilişkin kod tablosu.....                                                                                                           | 49 |
| Şekil 5.7 $F=a'b+ab'+ab$ (veya) fonksiyonuna ilişkin BDD gösterimi .....                                                                                                        | 49 |
| Şekil 5.8 $F=a'b+ab'+ab$ (veya) fonksiyonuna ilişkin geçiş transistör lojigi devre yapısı.....                                                                                  | 50 |
| Şekil 5.9 $F=a'b+ab'+ab$ (veya) fonksiyonuna ilişkin, kaynakları da içeren devre yapısı.....                                                                                    | 50 |
| Şekil 5.10 Şekil 5.9'daki devreye ilişkin simülasyon sonucu, (a) giriş 1, (b) giriş 2, (c) çıkış                                                                                | 51 |
| Şekil 5.11 (5.2)'de verilen lojik fonksiyona ait BDD gösterimi .....                                                                                                            | 53 |
| Şekil 5.12 (5.2)'de verilen lojik fonksiyona ait geçiş transistör lojik devre modeli .....                                                                                      | 53 |
| Şekil 5.13 (5.2)'de verilen fonksiyona ilişkin geçiş transistör lojik devresi .....                                                                                             | 54 |
| Şekil 5.14 Şekil 5.13'deki devreye ait simülasyon sonuçları, (a), (b), (c), (d) ve (e) sırasıyla girişleri ve (f) ise çıkışı göstermektedir. ....                               | 54 |
| Şekil 5.15 $Y_0$ lojik fonksiyonuna ait karar diyagramı gösterimi.....                                                                                                          | 57 |
| Şekil 5.16 $Y_1$ lojik fonksiyonuna ait karar diyagramı gösterimi.....                                                                                                          | 58 |
| Şekil 5.17 $Y_2$ lojik fonksiyonuna ait karar diyagramı gösterimi.....                                                                                                          | 58 |
| Şekil 5.18 $Y_0$ lojik fonksiyonuna ait geçiş transistör lojik devre modeli.....                                                                                                | 59 |
| Şekil 5.19 $Y_1$ lojik fonksiyonuna ait geçiş transistör lojik devre modeli.....                                                                                                | 59 |
| Şekil 5.20 $Y_2$ lojik fonksiyonuna ait geçiş transistör lojik devre modeli.....                                                                                                | 59 |
| Şekil 5.21 İki bit toplayıcıya ilişkin geçiş transistör lojik devresi .....                                                                                                     | 60 |
| Şekil 5.22 İki bit toplayıcıya ilişkin simülasyon sonuçları, (a), (b), (c) ve (d) kaynakları, (e), (f) ve (g) sırasıyla $Y_0$ , $Y_1$ ve $Y_2$ çıkışlarını göstermektedir. .... | 60 |

## ÇİZELGE LİSTESİ

|                                                                      |    |
|----------------------------------------------------------------------|----|
| Çizelge 5.1 Örneği verilen lojik fonksiyona ilişkin kod tablosu..... | 52 |
| Çizelge 5.2 İki bit toplayıcıya ilişkin doğruluk tablosu .....       | 55 |
| Çizelge 5.3 İki bit toplayıcıya ilişkin kod tablosu.....             | 56 |



## ÖNSÖZ

Çalışmam boyunca bana yol gösteren, her zaman olumlu yönde sabırla destek veren ve sürekli fikir alış verişinde bulunduğum, değerli hocam Doç. Dr. Tülay Yıldırım'a, bilgileri ve birikimlerinden yararlandığım değerli arkadaşım ve meslektaşım Ar.Gör. Mutlu Avcı'ya ve yazılım geliştirme aşamasında birikimlerinden sıklıkla faydalandığım YTÜ Bilgisayar Mühendisliği araştırma görevlisi arkadaşlarıma, sonsuz teşekkürlerimi sunarım.



## ÖZET

Bu çalışmada, Avcı ve Yıldırım'ın (2003) önerdikleri, genelleştirilmiş bir sayısal devre tasarım metodu referans alınarak, lojik fonksiyonların geiş transistörleri bazlı gereklenmesi için görsel bir yazılım geliştirilmiştir.

Şu ana kadar yapılan tasarımlarda, klasik metodlar kullanılarak, iki düzeyli geiş transistör devreleri tasarlamak ve bunları blok olarak kullanan kompleks gereklemeler yapmak mümkün olmuştur. Bu çalışma ile birlikte, çok daha kompleks lojik fonksiyonlara ilişkin devrelerin, ikil karar diyagramları kullanılarak, daha az sayıda transistör ile gereklenmesi bilgisayar ortamında sağlanmıştır.

Bazı lojik kapıların ve birtakım lojik fonksiyonların devre gereklemeleri, devre yapıları ve simülasyon sonuçları ile birlikte sunulmuştur.

Geliştirme ortamı olarak, Borland C++ Builder 6.0 kullanılmış ve elde edilen devre yapılarının simülasyonları, TÜBİTAK-YİTAL 1.5µm teknolojisi model parametreleri kullanılarak, PSPICE ortamında gerekleştirilmiştir.

**Anahtar kelimeler:** Geiş Transistör Lojik, BDD, sayısal devre tasarımı, CPL-TG, TG, lojik fonksiyonların gereklenmesi, lojik fonksiyonların gösterimleri.

## **ABSTRACT**

In this study, a visual software for the pass transistor based implementation of logic functions has been developed, with reference to a generalized digital circuit design method, proposed by Avci and Yildirim in 2003.

In the designs made so far, it has been possible to design two level pass transistor circuits and to make complex implementations that use these as blocks, using classical methods. With this study, the implementation of circuits related to even more complex logic functions has been provided in the computer environment, using binary decision diagrams.

The circuit implementations of basic logic gates and some logic functions have been presented with their circuit structures and simulation results.

Borland C++ 6.0 was used as the development environment and the simulations of the circuit structures obtained were developed in the PSPICE environment using the TÜBİTAK-YİTAL 1.5 $\mu$ m technology model parameters.

**Keywords:** Pass Transistor Logic, BDD, digital circuit design, CPL-TG, TG, implementation of logic functions, demonstration of logic functions.

## 1. GİRİŞ

Yirminci yüzyılın ortalarında tanımlanan ve üretilen ilk transistör ile başlayan süreç, yine yirminci yüzyılın son çeyreğinde taleplerin hızlı artışı ve üreticilerin taleplere cevap verebilme kapasitelerinin yükselmesi ile birlikte, yüksek ivmeli bir yarış halini almıştır. Günümüz elektronik pazarının ihtiyaçları, sektör çalışanlarını yeni teknolojileri araştırmaya teşvik etmektedir. Yeni teknolojilerin, eski teknolojiler ile ilişkilendirilmesi, birlikte sentezlenmesi veya eski teknolojilerin yerini alması şeklinde ortaya çıkan bu çabaların sonucunda pazar ihtiyaçları karşılanmaya çalışılmaktadır.

Günümüz elektronik pazarına bakıldığında ve geçmişle mukayese edildiğinde, çok daha yüksek hızlarda çalışabilen, az güç tüketen, yüksek veri işleme kapasitesine sahip, taşınabilir ve hacimce çok küçük cihazların raflarda yerlerini aldıklarını görmek mümkündür. Örneklendirmek gerekirse, son derece küçük hacimlere ve güç tüketimine sahip mobil iletişim araçlarını, diz üstü bilgisayarları, ses ve görüntü işleme cihazlarını vermek mümkündür. Örnekler çoğaltılacak olsa da, ortaya çıkan gerçek şudur ki, geçmişte benzer görevi yerine getiren ve benzer işlevi olan cihazlara göre günümüz elektronik cihazlarının daha fazla kullanıcı konforu gözetilerek tasarlandıklarıdır. İşte bu nokta, niçin araştırmacıların daha küçük, daha az güç tüketen, taşınabilir ve daha yüksek veri işleme kapasitesine sahip cihazları ortaya koyacak teknolojilerin peşine düştüklerini net olarak ortaya çıkarmaktadır.

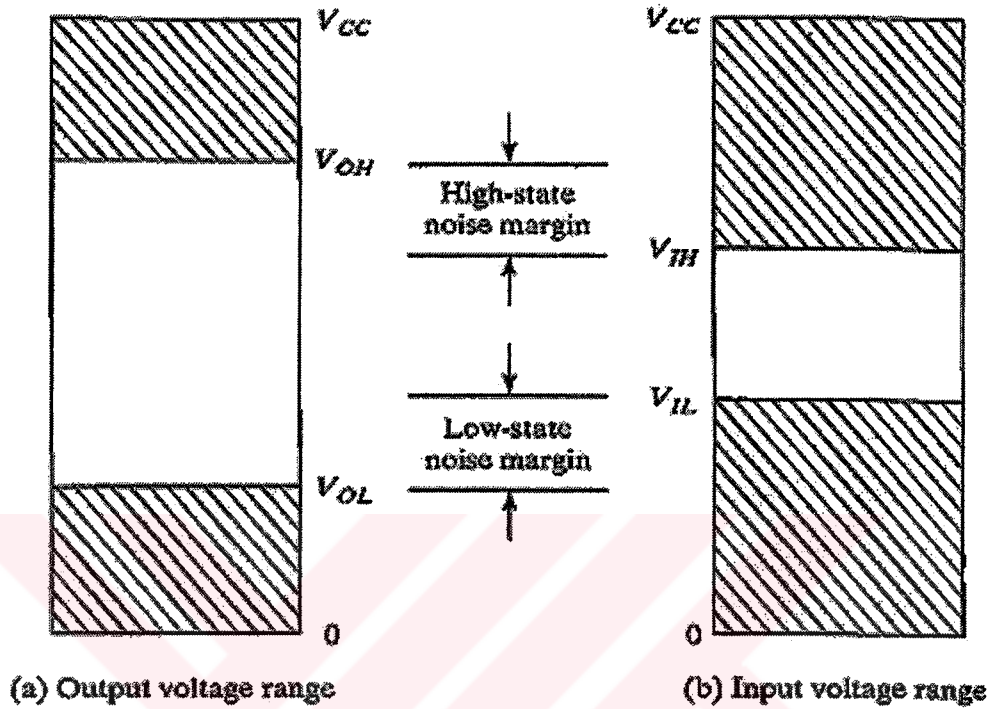
### 1.1 Elektronik Entegre Devre Tasarım Disiplinleri

Üç ana başlık altında tanımlamak mümkündür. Bunlar sırası ile, analog devre tasarımı, sayısal devre tasarımı ve analog ve sayısal devre tasarımını birlikte kullanan hibrit devre tasarımıdır. Her üç disiplin de, tasarımlarında analog devre elemanları kullanırlar.

Analog ve sayısal devre tasarımlarının kendilerine özgü avantaj ve dezavantajları vardır. Aynı fonksiyona karşılık gelecek olan analog devre çözümü, genel bir yaklaşım ile sayısal devre gerçeklemesine göre daha az sayıda transistör ihtiva edecektir. Bu kesinleşmiş bir yargı olmamakla birlikte, yapılan çalışmalardan elde edilen sonuçlar bu fikri doğrulamaktadır. Analog devre tasarımlarının, sayısal rakiplerine göre daha hızlı çalıştıkları bilinmektedir. Her ne kadar bu sıralanan avantajları olsa da, analog tasarımın en büyük sorunu gürültü marjının son derece küçük olmasıdır, diğer bir ifade ile gürültü, analog devre tasarımcılarının üstesinden gelmek zorunda oldukları önemli bir konudur.

Sayısal devre tasarımı, her ne kadar hız ve kapladığı silikon alanı anlamında analog çözümler

kadar başarılı olmasa da, çok önemli avantajı, gürültüye karşı hassasiyetinin, analog devre tasarımı ile kıyaslandığında büyük ölçüde az olmasıdır. Bu gerçek, sayısal işaretlerin yapısını ve sayısal devrelerin girişlerinde kabul ettikleri ve çıkışlarında ürettikleri gerilim seviyelerini düşününce açığa çıkmaktadır.



Şekil 1.1 Sayısal işaretler için gürültü marjı (Leblebici, 1996)

Sayısal devre tasarımının çok geniş VLSI tasarım araçlarının olması, sayısal devre tasarımının avantajlarından bir diğeri olarak sayılabilir.

Sayısal devrelerin, analog devrelere göre daha çok sayıda transistör ile benzer fonksiyonları gerçekleştirdiği ve bu durumun sayısal devrelerin dezavantajı olduğu ifade edilmişti. Son yıllarda, birkaç mikron ve hatta mikron altı teknolojilerinin ortaya çıkması ile çok sayıda transistörü ucuz maliyet ile küçük hacimlere sığdırmak mümkün olmuş ve bu durum sayısal devrelerin bahsi geçen dezavantajını önemli ölçüde azaltmıştır. Her ne kadar, transistör sayısını ekonomik şekilde küçük hacimlerde arttırmak mümkün ise de, mühendis ve tasarımcıların önünde, aşmaları gereken diğeri bir konu ise, daha fazla transistör ile birlikte karşılaşılan iç bağlantı sayısının artmasıdır. Bu durum beraberinde yeni çözümlerin ve iyileştirmelerin ortaya konulmasını zorunlu kılar.

Hibrit devre tasarımı, analog ve sayısal devre tasarım çözümlerini birlikte öneren, her ikisinin de avantajlarını kullanmayı hedefleyen ve birinin diğeri üstün geldiği noktada üstün olan

metodu uygulayarak, maksimum performansı elde etmeyi hedefleyen bir tasarım şeklidir.

## 1.2 Lojik Stilleri Hakkında Genel Bilgiler

Sayısal entegre devreler, geçmişten günümüze teknolojinin sunduğu imkanlar sayesinde evrim geçirmiştir. RTL ile başlayan süreç günümüzde MOS yapıların sayısal entegre devre tasarımında yoğun olarak kullanılmasına kadar ulaşmış ve bu sayede daha kararlı, daha az güç tüketen ve daha az yer kaplayan fiziksel gerçeklemeler mümkün olmuştur. Başlıca sayısal entegre devre türlerini şu şekilde sıralayabiliriz.

- RTL Resistor-transistor logic
- DTL Diode-transistor logic
- TTL Transistor-transistor logic
- ECL Emitter-coupled logic
- CMOS Complementary metal-oxide semiconductor
- PTL Pass-transistor logic

Sıralanan türler arasında, özellikle ilk iki maddenin, günümüz teknolojilerine ışık tutması açısından öneminin büyük olduğunun altını çizerek, özellikle son iki madde olan CMOS ve MOS yapıların günümüz sayısal devre teknolojisi içerisindeki yerinin ve yüzdesinin çok büyük olduğunu belirtmek gerekir. Bu gerçeğin temel nedeni olarak, düşük güç harcamaları, yüksek frekanslardaki daha kararlı davranışları, mikron altı teknolojilere uygunluğu, bu sayede daha az silikon alanı kaplamaları ve yüksek gürültü marjları gösterilebilir.

MOS sayısal entegre devrelerini kendi aralarında statik ve dinamik sayısal entegre devreler olarak iki grupta tanımlamak mümkündür. Dinamik ve statik sayısal entegre devreler de kendi içlerinde gruplara ayrılırlar. Dinamik sayısal entegre devrelerine örnek olarak Domino Logic devreleri, NORA Logic devreleri ve TSPC devreleri verilebilirken, statik sayısal entegre devreleri ise, CMOS, geçiş transistörleri ve CVSL devreleri olarak örneklendirebiliriz.

Statik sayısal devreler, lojik fonksiyonları, sürekli hal davranışlarını esas alarak gerçeklerler. Diğer bir ifade ile, statik sayısal devrelerin tüm çıkış büyüklükleri, sürekli hal devre çıkış değerleridir. Böylece statik sayısal devre yapıları, çıkışını belli bir gecikme ile birlikte, tamamen girişine uygulanan gerilime bağlı olarak üretir ve besleme kesilmedikçe ve yeni girişler tanımlanmadıkça çıkış değerini korur. Sayısal devre yapılarında, hız ve silikon alanı kriteri ise, dinamik sayısal devre teknikleri devreye girer. Özellikle, sadece NMOS kullanarak gerçekleştirilen dinamik sayısal devre tasarımları ile daha az sayıda transistör kullanarak

fonksiyon gerçeklemek mümkündür. Dinamik sayısal devreler, devrenin sürekli hal davranışını değil, geçici hal davranışını esas alır. Dinamik devrelerde, iki faz bulunur. İlk yükleme öncesi fazı (precharge), diğeri ise, yüklenme sonrası veya değerlendirme fazı (evaluation phase) olarak verilir. Bu fazlar bir saat ile kontrol edilir. Dinamik devre yapıları, hızlı çalışmalarının yanında, özellikle işaret aktivitesinin yoğun olduğu durumlarda, şarj öncesi fazı sebebiyle fazla güç harcamaktadır.

Zimmermann ve Fichtner (1997) yaptıkları çalışmada, CMOS ve çeşitli geçiş transistör lojiği stillerini birlikte ele almış, birbirleri arasında karşılaştırmalar yapmış ve sonuçlarını değerlendirmiştir.

Geçiş transistörleri ile sayısal devre tasarlanmasında, Jaekel, Bandyopadhyay ve Jullien (1998) yaptıkları çalışmada, bir lojik fonksiyona karşılık gelen bir kod ile birlikte, farklı bir BDD indirgemesinden bahsetmiş ve indirgedikleri BDD'lerin geçiş transistörü ile sayısal devre tasarımına uygunluğunu işlemiştir. Çalışmalarında, dinamik tasarıma örnekler vermişlerdir. Ancak bu çalışmada, ileride de değinileceği gibi, kodlama ile ilgili hiçbir açıklama yer almamaktadır. Kodlamanın nasıl gerçekleştirildiğini de ihtiva eden, daha sade bir BDD indirgeme algoritmasına sahip ve statik bir tasarım metodunu öneren bir çalışma Avcı ve Yıldırım (2003) tarafından yapılmıştır.

### 1.3 Çalışmanın Amacı ve Kapsamı

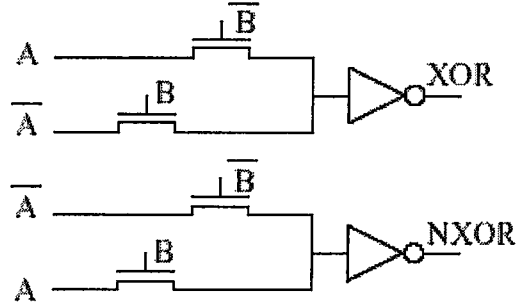
İkil karar diyagramlarının (BDD), sayısal entegre devre tasarımının bir dalı olan geçiş transistörü lojiği ile devre gerçekleştirmeye oldukça uygun bir model oluşu ve BDD'lerin bilgisayar ortamına olan uyumluluğu, beraberinde, model olarak BDD'yi kullanan geçiş transistör lojik devre tasarımı için yazılım geliştirilmesi ihtiyacını gündeme getirmiştir. Yapılan çalışmada, geçiş transistör lojiği ile tasarım modeli olarak, Avcı ve Yıldırım (2003)'ın yaptıkları çalışma ele alınmıştır. Bu statik sayısal devre tasarımı modeli, bilgisayar ortamına taşınarak bir program geliştirilmiş ve bu programla birlikte belli bir maksimum değişken sayısına sahip lojik fonksiyona karşılık gelen statik sayısal devreyi şematik olarak elde etmek mümkün olmuştur. Statik sayısal devre tasarımında, yarıiletken elektronik devre elemanı olarak, gerilim kontrollü anahtar elemanı özelliğine sahip geçiş kapıları (TG) kullanılmıştır. TG'ler, çoğullayıcı yapısı oluşturacak şekilde organize edilmişlerdir. Geliştirilen program, sadece, sayısal devreyi değil, bu sayısal devreye model teşkil eden BDD yapısını da çizdirecek şekilde geliştirilmiştir. Fiziksel olarak gerçekleşmesi planlanan sayısal devreye karşılık gelen lojik fonksiyonun alabileceği maksimum değişken sayısı 14

(a,b,.....,m,n) olarak seçilmiş ve bu sayede VLSI tasarımı standartlarına uygunluk sağlanmıştır. Çalışma, birtakım lojik fonksiyonların, geliştirilen programa uygulanması, elde edilen devrelerin simüle edilmesi ve sonuçlarının sunulması ile tamamlanmıştır.



## 2. GEÇİŞ TRANSİSTÖR LOJİK HAKKINDA GENEL BİLGİLER

Geçiş transistör lojiği, en genel haliyle, giriş işaretleri ve tümleyenleri ve NMOS veya CMOS yapıların oluşturduğu lojik stili olarak adlandırılabilir. Giriş işaretlerinin tümleyenlerini elde etmek için evirici yapısı kullanılmaktadır. Şekil 2.1’de geçiş transistör XOR-NXOR lojik kapısı gösterilmiştir.



Şekil 2.1 Geçiş transistör lojik xor kapısı (Munteanu, 2001)

Literatürde, geçiş transistör lojiği ailesi içerisinde bir çok alt gruplandırma yapılmıştır. Gruplandırmalar, statik ve dinamik lojik devreler olarak yapıldığı gibi, devre işleyiş ve yapısı, kullanılan yarı iletken elemanlar (salt NMOS veya CMOS) gibi kriterler göz önünde tutularak yapılmış gruplandırmalarla da karşılaşmak mümkündür. Geçiş transistör lojik stillerine örnek olarak, CMOS (with pass gates), DPL (Double Pass Transistor Logic), SPL (Single Rail Pass Transistor Logic), CPL (Complementary Pass Transistor Logic), EEPL, SRPL (Swing Restored Pass Transistor Logic) ve PPL (Push Pull Pass Transistor Logic) verilebilir. Zimmermann ve Fichtner (1997) çalışmalarında, isimleri geçen geçiş transistör stillerini, CMOS lojik alternatifleri ile geniş bir şekilde değerlendirmiştir.

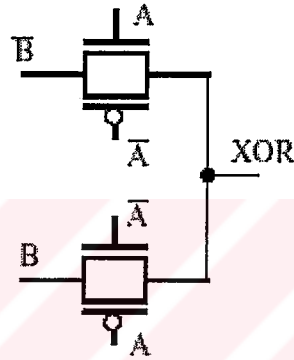
Yapıların, PMOS ve NMOS birlikte içirme durumları veya sadece NMOS içirme durumları göz önüne alınarak bir gruplandırma yapılırsa, CPL-TG'nin, DPL'nin ve DVL'nin, PMOS ve NMOS birlikte içeren yapılar ve CPL'nin ve SPL'nin ise sadece NMOS içeren yapılar oldukları görülür. NMOS ve PMOS birlikte içeren komplement yapıların çıkışlarında 0 volt ve Vdd gerilim dalgalanması elde edilebilirken, sadece NMOS içeren yapıların çıkışlarındaki gerilim seviyesi, bir NMOS için, eşik seviyesi kadar zayıflamış olarak elde edilir. Diğer bir ifade ile, NMOS ve PMOS' lardan ibaret komplement geçiş transistörleri çıkışlarında 0 - Vdd dalga şekli oluştururken, bu durum, yapılarında sadece NMOS içeren geçiş transistör lojik modellerinde 0 - (Vdd - Vth) şeklindedir. Burada, salt NMOS'lardan ibaret geçiş transistör lojik türlerinin, komplement geçiş transistörlerine göre daha az silikon alanı kapladıklarını

belirtmek gerekir. Devam eden bölümde, geçiş transistörleri kısaca incelenecektir.

## 2.1 Geçiş Transistör Çeşitleri

### 2.1.1 CPL-TG

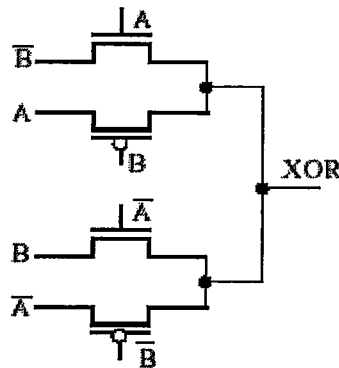
Şekil 2.2, CPL-TG yapısına basit bir örnektir. CPL-TG yapısını, CMOS geçiş kapıları (TG) oluşturur. CPL-TG, geçiş kapılarının, çoğullayıcı yapısında organize edilmeleri ile birlikte oluşturulurlar. Şekil 2.2’de, xor lojik kapısı bu şekilde elde edilmiştir. Abu ve Khater (96), çalışmalarında, CPL-TG ile gerçeklediği tam toplayıcının, geleneksel CMOS tam toplayıcı devresine göre yarı yarıya daha az enerji harcamasını not etmiştir.



Şekil 2.2 CPL-TG ile xor lojik kapısı tasarımı (Munteanu, 2001)

### 2.1.2 DPL

Suzuki (93) yaptığı çalışma ile, CPL-TG yapısının geliştirmiştir ve DPL’yi tanıtmıştır. CPL-TG’ye benzer olarak, DPL de, NMOS ve PMOS yapıları birlikte kullanır.



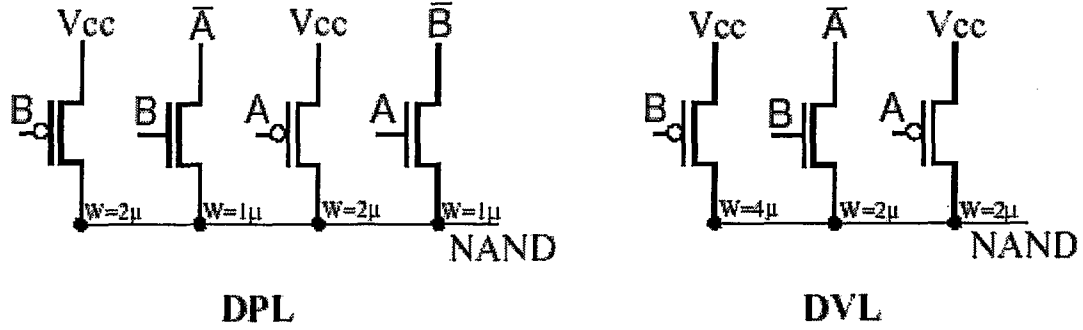
Şekil 2.3 DPL ile xor lojik kapısı tasarımı (Munteanu, 2001)

Girişlerine uygulanan işaretlerin ve kontrol değişkenlerinin organizasyonu, DPL’yi, CPL-

TG'den ayıran temel noktadır. Şekil 2.3 DPL'yi geçiş transistör lojik modeli olarak kullanan bir xor lojik kapısına örnek olarak verilmiştir.

### 2.1.3 DVL

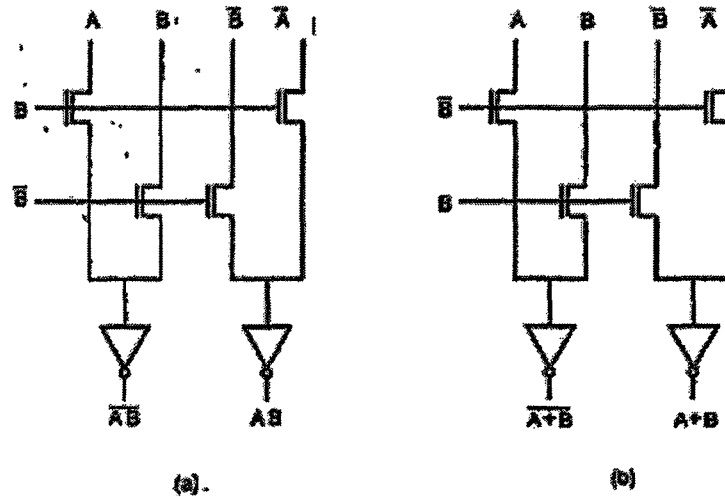
DPL'in bir versiyonudur. DPL'de fazlalık bağlantıların iptal edilmesi ve geçiş ve kontrol değişkenlerinin tekrar organizasyonu ile elde edilir. Şekil 2.4'de, DPL ve DVL ile lojik nand kapılarına örnek verilmiştir.



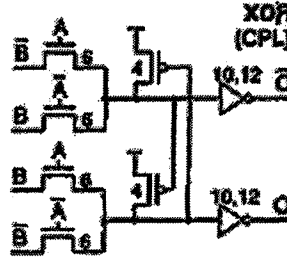
Şekil 2.4 DPL ve DVL ile lojik nand kapısı (Munteanu, 2001)

### 2.1.4 CPL

CPL, en çok bilinen ve en yaygın geçiş transistör lojik modelidir. CPL devreleri, birbirinin tümleyeni iki adet çıkış oluşturacak şekilde oluşturulurlar. Bu sayede, çıkış elde edilirken, çıkışın tümleyeni de beraberinde elde edilmiş olur. Bu durum, eğer çıkışın, devre yapısı içerisinde, hem kendisine hem de tümleyenine ihtiyaç duyuluyor ise, avantaj sağlayabilir.



Şekil 2.5 (a) CPL lojik nand2 kapısı , (b) CPL lojik nor2 kapısı (Munteanu, 2001)

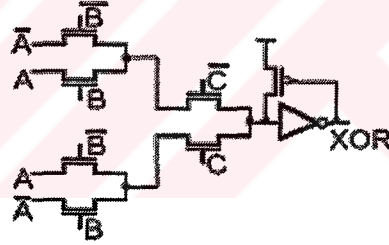


Şekil 2.6 CPL lojik xor kapısı (Munteanu, 2001)

Yapı, sadece NMOS transistörleri ihtiva ettiğinden, önceki sayfalarda bahsettiğimiz, bir NMOS için, çıkışın  $0 - (V_{dd} - V_{th})$  gerilim düşümünü giderebilmek için en azından ikinci seviyeden sonra, seviye restorasyonuna ihtiyaç duyar.

#### 2.1.5 SPL

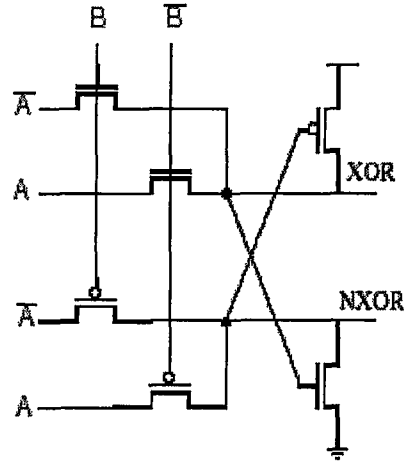
CPL'nin versiyonu olan SPL, CPL gibi, NMOS'lardan ibaret bir yapıdır. Farklı yanı, çıkışında tek bir işaret üretir. CPL'de olan, çıkış ve beraberinde çıkışın tümleyeni, SPL için geçerli değildir. CPL'den tek farkı bu olan SPL yapısının devre şekli, Şekil 2.7'de verilmiştir.



Şekil 2.7 SPL ile lojik xor kapısı (Munteanu, 2001)

#### 2.1.6 PPL

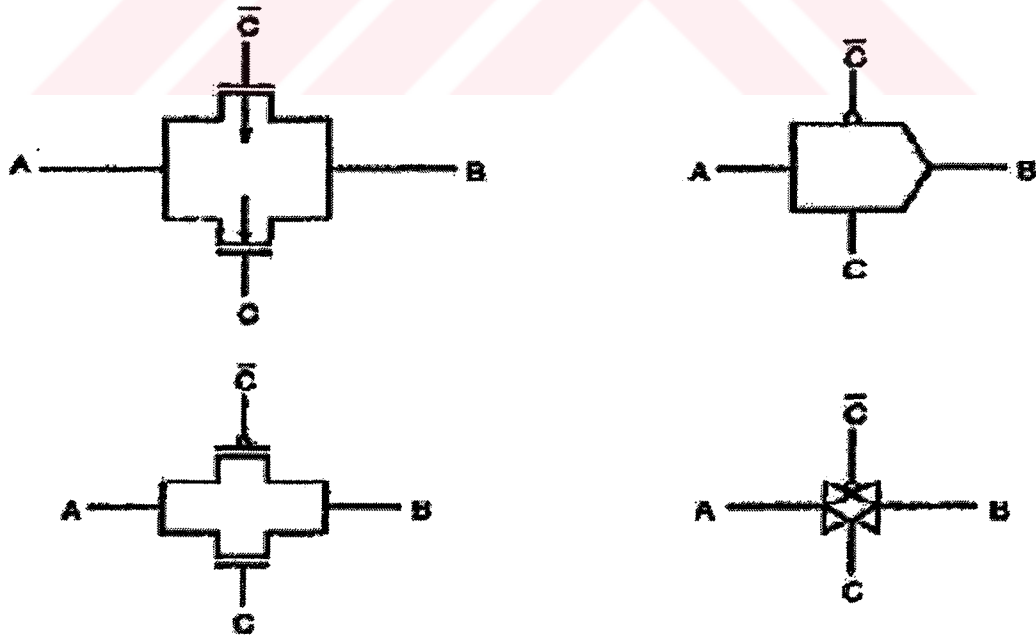
PPL, bahsedilen diğer yapılardan farklı olarak, salt NMOS yapılardan ibaret olabileceği gibi, NMOS ve PMOS birlikte kullanılarak da gerçekleştirilebilir. PPL, PMOS ve NMOS'un gate uçlarına toprak gerilimi uygulandığında, PMOS iletimde, NMOS kesimde, PMOS ve NMOS'un gate uçlarına  $V_{dd}$  gerilimi uygulandığında PMOS kesimde, NMOS iletimde durumlarının avantajını kullanır. CPL de olduğu gibi, çıkışında, çıkış işareti ile birlikte, tümleyenini de oluşturur.



Şekil 2.8 PPL ile lojik xor ve nxor kapıları (Munteanu, 2001)

## 2.2 CMOS Transmisyon Kapıları (Geçiş Kapıları)

Bu başlık altında, PTL ailesinin bazı türleri içerisinde ve yapılan çalışmada kullanılan CPL-TG modelinin temel yapı taşı olan CMOS transmisyon kapıları tanıtılacaktır. Kısaca TG olarak isimlendirilen transmisyon kapıları, Şekil 2.9'da görülebileceği üzere, bir NMOS ve bir PMOS'un paralel bağlanması ile meydana gelen yapının ismidir.

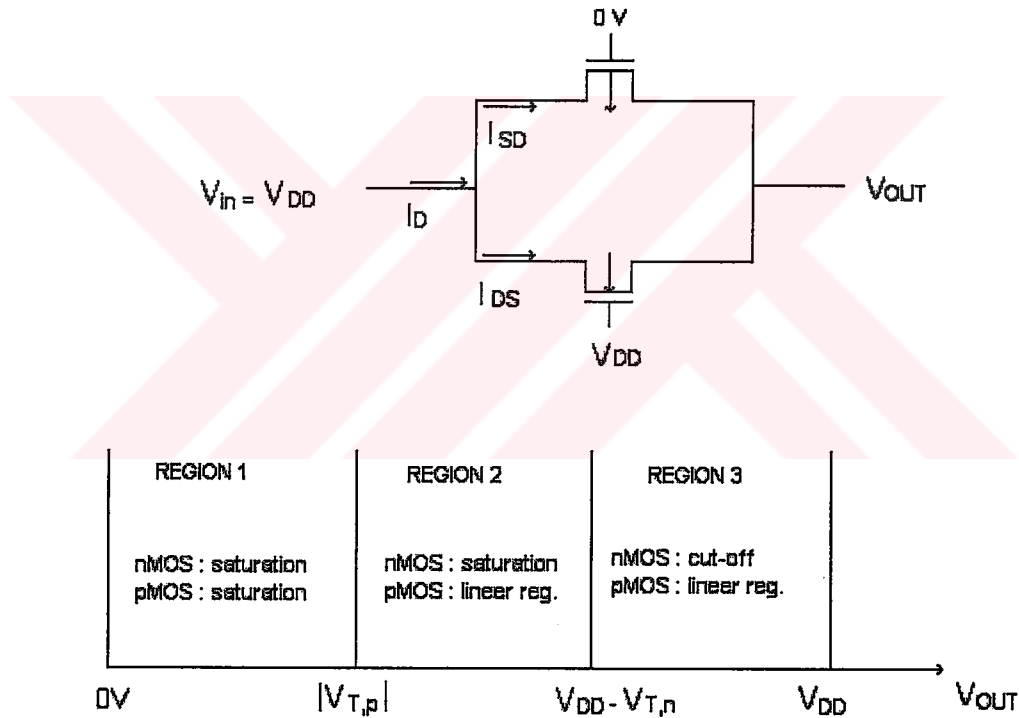


Şekil 2.9 CMOS transmisyon kapısı gösterim şekilleri (Leblebici, 1996)

Birbirlerine paralel bağlı her iki transistörün gate bacaklarına bağlanan işaretler birbirlerinin

tümleyenleri olacak şekilde organize edilir. Bu sayede, TG, A ve B düğümleri arasında, iki yönlü, C işareti tarafından kontrol edilen bir anahtar görevi görür. C işareti, lojik 1 seviyesinde ( $V_{DD}$ ) iken, her iki transistör de kapanır ve A ile B düğümleri arasında düşük değerli bir direnç bölgesi oluşarak, bu iki düğüm arasında akım akması sağlanır. Kontrol işareti olan C, eğer lojik 0 değerine çekilirse, her iki transistör de kesime giderek devreyi açar ve A ile B düğümleri arası açık devre olur. Yani transistörler bu durumda yüksek empedans oluştururlar.

Detaylı bir DC analiz yapabilmek için, Şekil 2.10'da verilen şartları inceleyeceğiz. Giriş düğümü A ve kontrol işareti,  $V_{DD}$  gerilimine bağlanarak, transistörlerin iletimde olmaları sağlanmıştır. Şu noktadan sonra Şekil 2.10'da verilen bias şartları ile TG'nin giriş çıkış akım-gerilim karakteristiği,  $V_{OUT}$ 'un bir fonksiyonu olarak elde edilmeye çalışılacaktır.



Şekil 2.10 TG'nin bias koşulları ve karşılık düşen çalışma bölgeleri (Leblebici, 1996)

NMOS için  $V_{DS}$  ve  $V_{GS}$  gerilimleri, (2.1) ve (2.2)'deki gibidir.

$$V_{DS,n} = V_{DD} - V_{OUT} \quad (2.1)$$

$$V_{GS,n} = V_{DD} - V_{OUT} \quad (2.2)$$

NMOS transistör,  $V_{OUT} > V_{DD} - V_{T,n}$  şartı için kapalı olacak iken,  $V_{OUT} < V_{DD} - V_{T,n}$  durumunda saturasyon modunda çalışacaktır. PMOS için  $V_{DS}$  ve  $V_{GS}$  gerilimlerini yazalım.

$$V_{DS,p} = V_{OUT} - V_{DD} \quad (2.3)$$

$$V_{GS,p} = -V_{DD} \quad (2.4)$$

PMOS için,  $V_{OUT} < |V_{T,p}|$  durumunda transistör saturasyonda,  $V_{OUT} > |V_{T,p}|$  durumunda ise lineer çalışma bölgesinde olduğunu söyleyebiliriz. Bu analiz ile birlikte, Şekil 2.10'da da gösterdiğimiz gibi, TG için,  $V_{OUT}$  çıkış gerilimine bağlı üç çalışma bölgesi tanımlar. TG üzerinden akan toplam akım (2.5)'de gösterilmiştir.

$$I_D = I_{DS,n} + I_{SD,p} \quad (2.5)$$

Bu noktada, her iki transistör için eşdeğer direnç tanımlayalım.

$$R_{eq,n} = \frac{V_{DD} - V_{OUT}}{I_{DS,n}} \quad (2.6)$$

$$R_{eq,p} = \frac{V_{DD} - V_{OUT}}{I_{SD,p}} \quad (2.7)$$

(2.6) ve (2.7)'de ayrı ayrı verilen NMOS ve PMOS eşdeğer direnç ifadeleridir. Diğer taraftan, TG toplam direnci, NMOS ( $R_{eq,n}$ ) ve PMOS ( $R_{eq,p}$ ) dirençlerinin paralel eşdeğeridir. Şimdi, Şekil 2.10'da gösterilen üç değişik çalışma bölgesine karşılık düşen direnç değerlerini araştıralım.

Birinci bölgede (*region 1*), çıkış gerilimi, PMOS eşik gerilim değerinden küçüktür,  $V_{OUT} < |V_{T,p}|$ . Şekil 2.10'a göre, her iki transistör de saturasyon modunda çalışmaktadır. Aşağıda, birinci bölge için tanımlanan eşdeğer direnç değerleri verilmiştir.

$$R_{eq,n} = \frac{2(V_{DD} - V_{OUT})}{k_n(V_{DD} - V_{OUT} - V_{T,n})^2} \quad (2.8)$$

$$R_{eq,p} = \frac{2(V_{DD} - V_{OUT})}{k_p(V_{DD} - |V_{T,p}|)^2} \quad (2.9)$$

İkinci bölgede (*region 2*),  $|V_{T,p}| < V_{OUT} < (V_{DD} - V_{T,n})$ 'dir. PMOS, lineer bölgede çalışırken, NMOS saturasyon modunda çalışmaya devam eder. Yine ikinci bölge için PMOS ve NMOS direnç değerlerini yazarsak;

$$R_{eq,n} = \frac{2(V_{DD} - V_{OUT})}{k_n(V_{DD} - V_{OUT} - V_{T,n})^2} \quad (2.10)$$

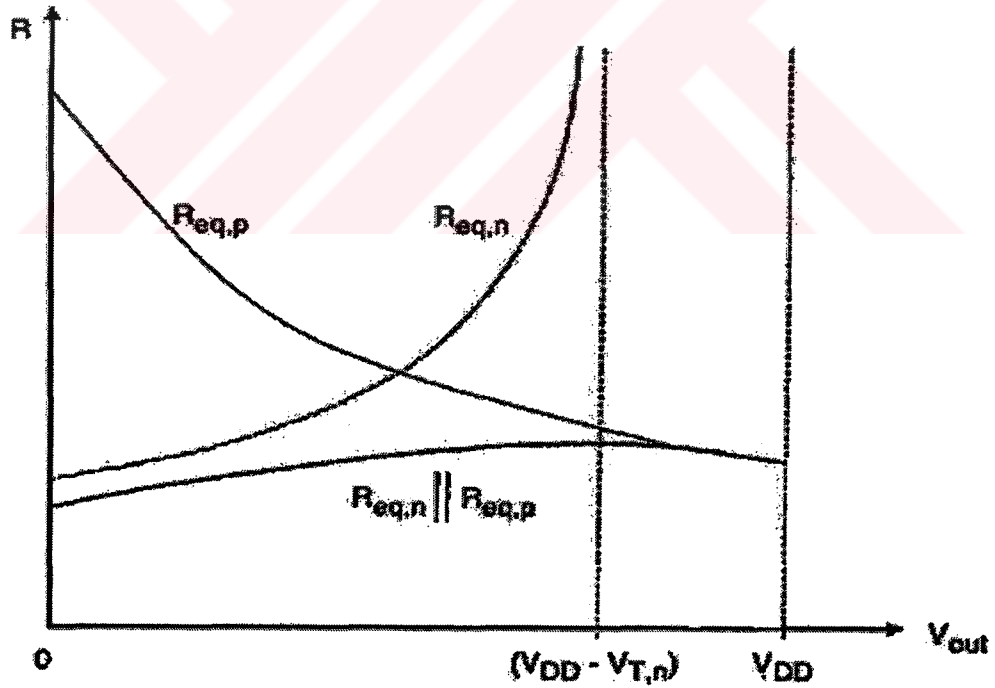
$$\begin{aligned}
R_{eq,p} &= \frac{2(V_{DD} - V_{OUT})}{k_p[2(V_{DD} - |V_{T,p}|)(V_{DD} - V_{OUT}) - (V_{DD} - V_{OUT})^2]} \\
&= \frac{2}{k_p[2(V_{DD} - |V_{T,p}|) - (V_{DD} - V_{OUT})]}
\end{aligned} \tag{2.11}$$

(2.10) ve (2.11) eşitliklerini elde ederiz.

Üçüncü bölgede (*region 3*),  $V_{OUT} > (V_{DD} - V_{T,n})$ 'dir. NMOS kesimde, PMOS ise lineer bölgede çalışmaya devam edecektir. NMOS kesimde olduğundan, açık devre olarak düşünmek mümkündür. Bu sebepten, bu bölge için NMOS direnç eşdeğeri tanımlanmayacak, sadece PMOS eşdeğer direnç ifadesi verilecektir.

$$R_{eq,p} = \frac{2}{k_p[2(V_{DD} - |V_{T,p}|) - (V_{DD} - V_{OUT})]} \tag{2.12}$$

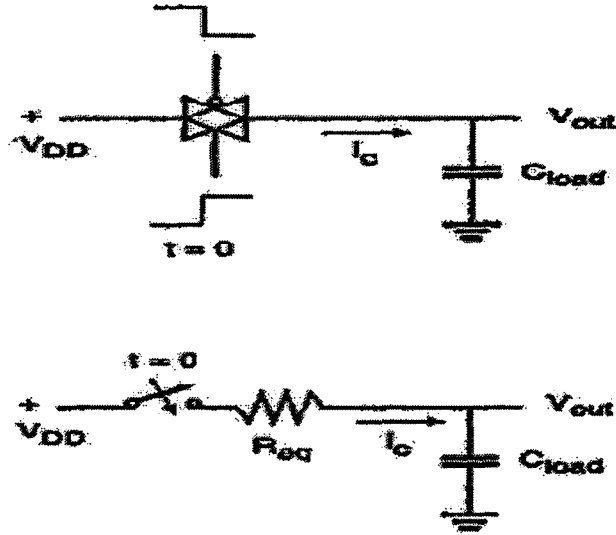
Üç çalışma bölgesi için tanımlanan eşdeğer direnç değerleri birleştirilir ve çizdirilir ise Şekil 2.11'deki  $f(R, V_{out})$  eğrisi elde edilir.



Şekil 2.11 TG eşdeğer direnç eğrisi (Leblebici, 1996)

Şekil 2.11'den görüleceği gibi,  $V_{OUT}$  değişimi ile PMOS ve NMOS eşdeğer dirençleri değişirlerken, TG direnci ( $R_{eq,n} || R_{eq,p}$ ) neredeyse sabittir. TG bu özelliği ile oldukça ilgi uyandırıcıdır.

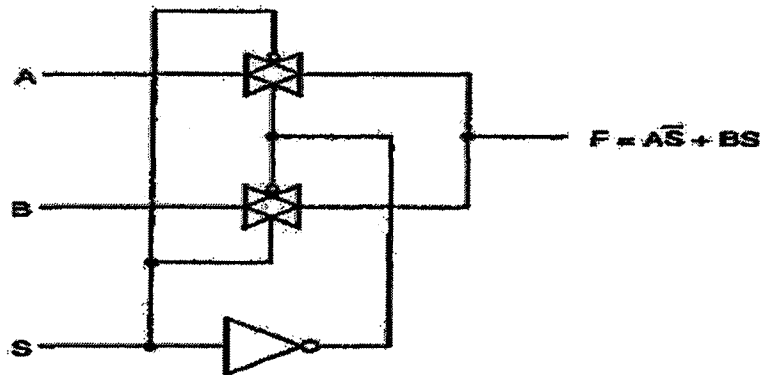
TG, bir direnç ve seri anahtar elemanı olarak modellenenebilir (Şekil 2.12).



Şekil 2.12 TG'nin eşdeğer modeli (Leblebici, 1996)

TG'ler ile gerçekleştirilen devrelerde, kontrol işaretleri, tümleyenleri ile beraber bulunmak zorundadır. Çünkü, kontrol işareti, kendisi NMOS'a uygulanmış ise, tümleyeni PMOS'a uygulanmalı veya kontrol işareti PMOS'a uygulanmış ise, tümleyeni NMOS'a uygulanmalıdır.

TG'lerin sıklıkla kullanıldığı alanların başında, çoğullayıcı gerçeklemeleri gelir. Şekil 2.13'de iki girişli bir çoğullayıcı yapısı TG'ler kullanılarak gerçekleştirilmiştir.



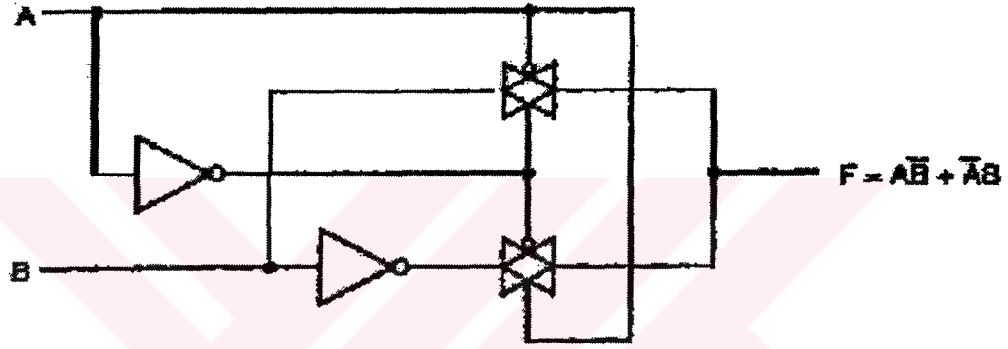
Şekil 2.13 TG'ler ile gerçekleştirilen iki girişli çoğullayıcı devresi (Leblebici, 1996)

Çalışma prensibi oldukça basit olan, şekli verilen çoğullayıcı yapısı, kontrol değişkeni olan S işaretinin lojik 1 olması ile altta yer alan TG elemanı akım akıtacağı, devresinin kapanacağı ve sonuçta çıkışın B'ye eşit olacağı, S işaretinin lojik 0 olması durumunda, üstte yer alan TG

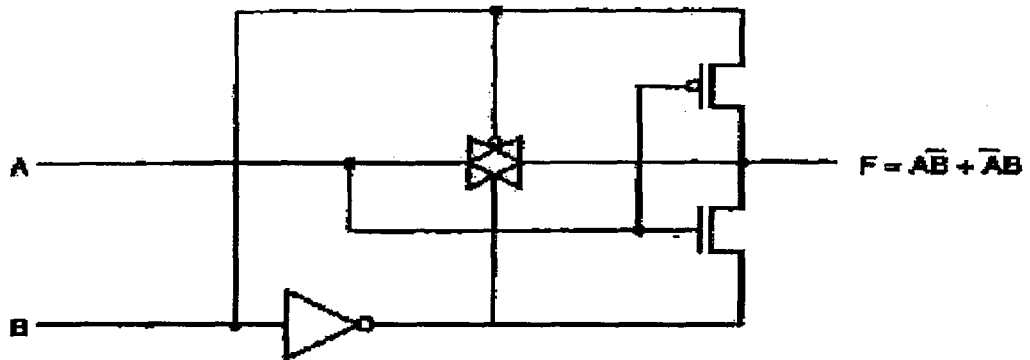
elemanının akım akıtacağı, devresinin kapanacağı ve çıkışın A'ya eşit olacağı şeklinde açıklanabilir. Dikkat edilmesi gereken husus, devrede S işaretinin iki durumu için de sadece bir kolun akım akıtacağı gerçeğidir. Kapalı olan yolun dışında kalan yol veya yollar açık devre olacak şekilde S işareti veya işaretleri organize edilerek kontrolü yapılır.

Diğer bir ilgi çekici konu ise, işaret geçişleri göz ardı edilirse, herhangi bir t anında devrenin sadece bir kolu üzerinden akım akacağından güç harcaması sadece o t anında üzerinden akım geçen yol ile ilişkilendirilir.

TG kullanarak gerçekleştirilen, sekiz ve altı transistör kullanan, lojik xor fonksiyonu gerçeklemeleri Şekil 2.14 ve Şekil 2.15'de verilmiştir.



Şekil 2.14 TG yapıları kullanan ve sekiz transistör ile gerçekleştirilen lojik xor kapısı (Leblebici, 1996)

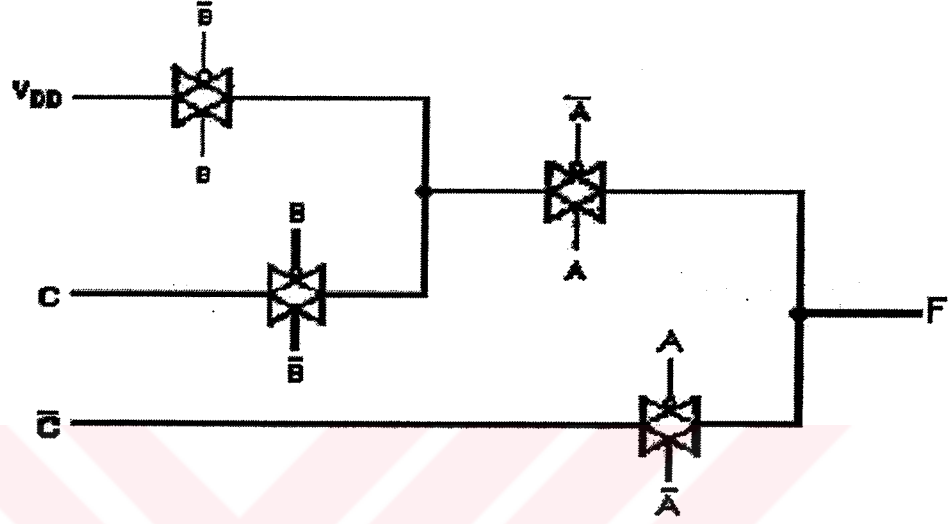


Şekil 2.15 TG yapısı kullanan altı transistör ile gerçekleştirilen lojik xor kapısı (Leblebici, 1996)

Şekil 2.14 tamamen TG'lerden ibaret olmasına karşın, Şekil 2.15 TG ve ilave NMOS ve PMOS içerir. TG'ler ile gerçeklemede, her kontrol işaretinin kendisi ve tümleyeni bulunmalıdır, bu sebepten ötürü kaynaklar ve kaynakların tersini (inverter yapısı) alan

devreler gereklidir. TG'ler kullanarak gereklenen  $F=AB+A'C'+AB'C$  lojik fonksiyonu, Őekil 2.16'da verilmiřtir.

Bu blmde TG'ler ele alınarak, devre davranıřları verilmeye alıřılmıřtır. TG'ler, tezimde alıřmasını yaptığım CPL-TG (Őekil 2.2) geiř transistr lojik modelinin yapı tařını oluřturmaktadır.



Őekil 2.16  $F=AB+A'C'+AB'C$  lojik fonksiyonu gereklemesi (Leblebici, 1996)

### 3. SAYISAL DEVRE TASARIMININ MODELLENMESİ

Her tasarım metodu, arka planında matematiksel bir gerçeğe dayandırılarak ortaya sunulmaktadır. Fiziksel devreler, bir matematiksel fonksiyonu gerçeklerler. Analog tasarımlar, analog işaretlere karşılık gelen matematiksel fonksiyonları ve sayısal tasarımlar da, lojik işaretlere karşılık gelen lojik fonksiyonları gerçeklerler.

Bir tasarımı, efektif kılmanın, ekonomik bir şekilde gerçekleştirmenin ve hedeflenen kriterlere uygunluğunun belirlenmesinin ardında birtakım temel felsefeler yatmaktadır. Üstteki paragrafta bahsedilen, her tasarımın arka planında, bir matematiksel gösterimi gerçeklediği gerçeğini düşünürsek, gerçekleştirilecek olan matematiksel fonksiyonun, minimize edilmiş eşleniğini bulmak veya bu matematiksel fonksiyonun yerine daha efektif fonksiyonlar tanımlamak ve yeni şekliyle fiziksel gerçekleştirme yapmak, bizlerin daha efektif ve ekonomik çözümler ortaya koymamızı sağlar. Diğer bir nokta, mevcut fonksiyonları gerçekleyen devre elemanlarının iyileştirilmesi olabilir. Böylece, daha az güç harcayan, daha az yer kaplayan ve daha hızlı çalışan çözümleri ortaya koymak mümkün olabilecektir.

Bir tasarımı efektif kılmanın diğer bir yolu ise, o fonksiyonu gerçekleyecek fiziksel yapıyı oluşturmak için harcanan zamandır. Günümüzde özellikle büyük sayıda transistör ihtiva eden devre tasarımları bilgisayar ortamında geliştirilmekte ve analiz edilebilmektedir. Bilgisayarların, devre analizi ve sentezi konularında kullanılması ile birlikte, her ölçekte analog ve sayısal devrelerin analizleri ve özellikle sayısal devrelerin sentezleri kolay bir şekilde yapılabilmekte ve bu sayede üretim öncesi tasarımcılara, geri besleme sağlanabilmektedir.

Sayısal devrelerin, gürültüye karşı daha az hassas ve iki durumlu ifade edilebiliyor olmaları, bilgisayar ortamında sentezlenebilmelerini kolaylaştıran temel etkenlerdir. Bu durum ayrıca, sayısal devreler için geniş VLSI araçlarının tanımlanmalarını ve kullanılmalarını beraberinde getirmiştir. Bu sayede, büyük sayıda transistör içeren sayısal devreler bilgisayar ortamında tanımlanabilmekte ve simüle edilebilmektedir.

Ardışıl olmayan sayısal devrelere karşılık gelen lojik fonksiyonların, matematiksel olarak ifade edilmelerinde, kanonik gösterim, standart gösterim veya tanımlı diğer lojik operandları kullanan gösterimler kullanılır. Kanonik gösterim, minterim toplamları veya makterim çarpımları şeklinde ifade edilir. Yine lojik fonksiyonları, tablo ile ifade etmek de mümkündür. Doğruluk tablosu olarak adlandırılan tabloda, fonksiyonun kapsadığı değişken uzayında, tüm mümkün girişler ve çıkışlar tanımlanmıştır. Bir lojik fonksiyona ait doğruluk tablosu ve

kanonik gösterim, birbiri arasında kolayca geçiş yapılabilen iki farklı gösterim şeklidir.

Lojik fonksiyonların, lojik kapılar ile veya transistör seviyesinde gerçekleşmesinde, kullanılacak olan lojik kapı veya transistör sayısı fonksiyonun karmaşıklığı ile ilgilidir. Bu sebepten, lojik fonksiyonların, doğruluk tablosunu etkilemeyecek şekilde minimize edilerek, fonksiyonun karmaşıklığının azaltılması önemlidir. Bu sayede daha az sayıda, lojik kapı veya transistör kullanılarak devre gerçekleştirilebilir. Lojik fonksiyonların indirgenmesinde kullanılan iki temel yöntem olarak, Karnaugh haritası ve McCluskey minimizasyon yöntemi verilebilir. Bu yöntemlerin uygulanması ile özellikle lojik kapılar seviyesinde minimize edilmiş lojik devre tasarımları gerçekleştirilir. Bahsedilen gösterim ve minimizasyon yöntemlerinin dışında, lojik fonksiyonların gösterim ve minimizasyonlarını bir paket halinde sunan bir yöntem olarak BDD örnek verilebilir. BDD'lerin, özellikle geçiş transistör lojiği kullanan sayısal devre sentezlerinde, devre mimarisi ile olan benzerlik ve uyumu, tasarımcı ve araştırmacıların ilgisini toplamıştır.

### 3.1 İkili Karar Diyagramları (BDD's)

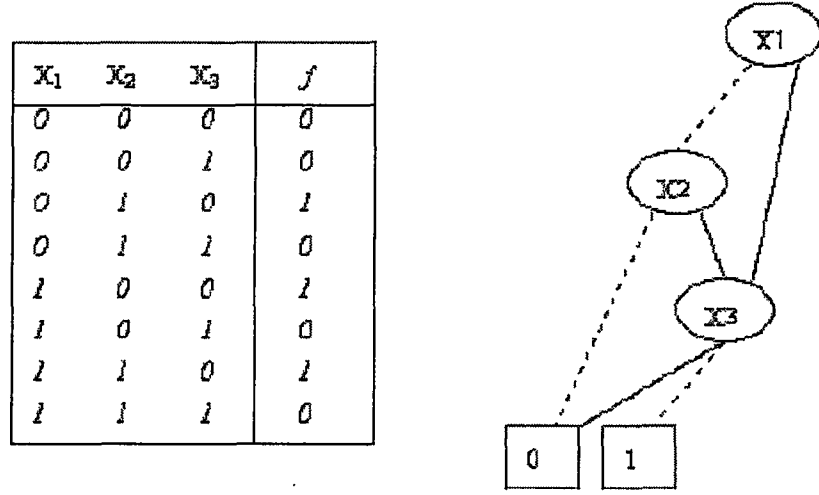
Özellikle son on beş senedir, lojik fonksiyonları grafiksel olarak sembolize eden ve transistör seviyesinde sayısal devre tasarımına olan uygunluğu ile dikkati çeken BDD gösterim şekli, araştırmacı ve tasarımların ilgisini BDD'lerin ve BDD'leri kullanan tasarım metotları üzerinde toplamayı başarmıştır.

BDD, Lee (1959) ve Akers (1978) tarafından tanıtılmış ve Bryant (92), sıralama ve indirgeme sınırlamaları tanımlayarak, OBDD (sıralanmış BDD) ve ROBDD (indirgenmiş ve sıralanmış BDD) tanımlamaları yapmıştır. BDD'nin grafiksel özelliğini kullanan ancak ikil durumun, haliyle lojik fonksiyon yapısının dışında kalarak, aritmetik işlemler için uyarlanmış (MTBDD ve EVBDD) BDD türleri de tanımlanmışlardır.

#### 3.1.1 BDD Genel Yapısı

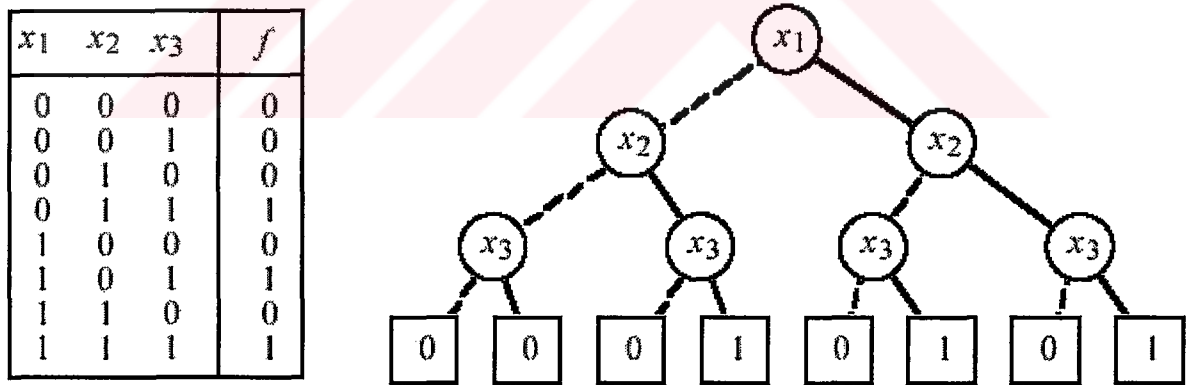
BDD'lerin, lojik fonksiyonları sembolize eden, grafiksel gösterimler olduklarını söylemiştik. Şekil 3.1 incelenecek olursa, doğruluk tablosu şeklin sol tarafında verilen bir lojik fonksiyona ilişkin BDD gösterimi şeklin sağ tarafında görülebilir. BDD, grafiksel bir gösterimdir. Düğümler ve düğümleri bağlayan bağlantılardan oluşur. Düğümler iki çeşittir; birincisi terminal olmayan düğümler, Şekil 3.1'de elipslerle çizilmiş olanlar ve diğer çeşit ise terminal düğümleridir ki, Şekil 3.1'de karelerle gösterilmişlerdir. Şekilde, noktalı çizgiler  $f(x) |_{x=0}$  ve sürekli çizgiler ise,  $f(x) |_{x=1}$ 'e karşılık gelmektedir. BDD oluşturulmasında, temel olarak,

Shannon açılımı kullanılır. BDD, ikil karar ağaçlarının (BDT), sıralanması ve indirgenmesinden sonra elde edilen yapının adıdır.



Şekil 3.1 Doğruluk tablosu ile verilen lojik fonksiyon ve ona ilişkin ve BDD gösterimi [2]

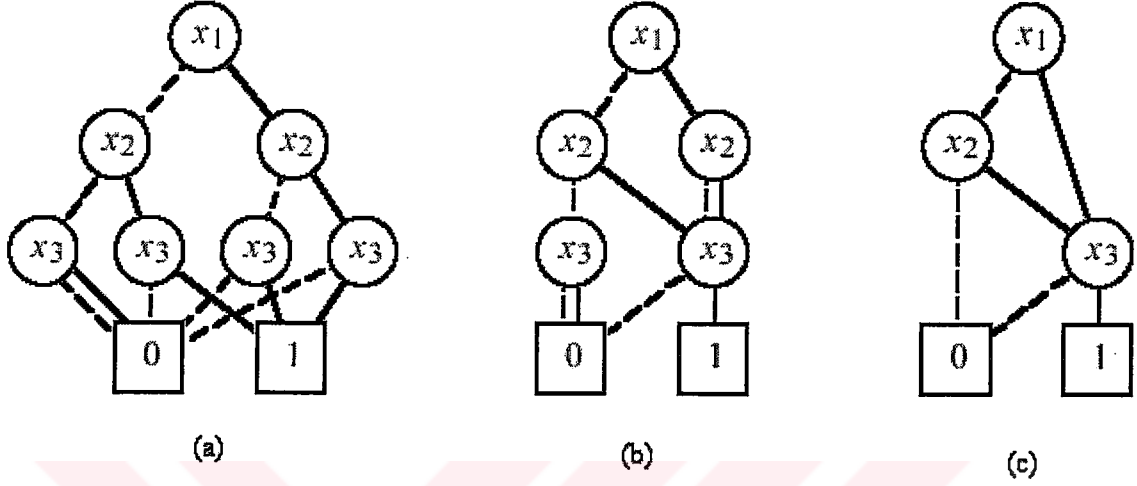
BDT'ler, en genel halde bir lojik fonksiyonun hiçbir indirgeme olmaksızın gösterildiği grafiksel form yapısıdır. Şekil 3.2'de gösterilmiştir.



Şekil 3.2 Doğruluk tablosu verilen bir lojik fonksiyon ve ilişkin BDT (Bryant, 1992)

Her ne kadar, bu yapıyla BDT, lojik fonksiyonu ifade ediyor olsa bile, gerekli sıralama ve indirgemelerden sonra elde edilen BDD, aynı doğruluk tablosuna ilişkin lojik fonksiyonu daha az düğüm ve bağlantı ile ifade edecektir. Şekil 3.3'de, Şekil 3.2'de verilen BDT üzerinde sıralama ve indirgeme yapılarak elde edilen BDD gösterilmiştir. BDT'ler içerisinde yer alan terminal olmayan düğümler, iki adet çocuk düğüm oluşturur ve çocuk düğümlerine 0 (kesikli

çizgi) ve 1 (düz çizgi) bağlantıları ile bağlanırlar. Çocuk düğüm oluşturan düğümlere ise, o çocuk düğümlerin babası olarak isim verilir. BDT'den BDD'ye geçişte, indirgemeler olabileceğinden, bir babaya ait çocuk düğümlerden birisi veya ikisi indirgeme sonucunda kaybolabilir. Şekil 3.3'de, Şekil 3.2'deki BDT'nin indirgenme aşamaları ve birtakım çocuk düğümlerin nasıl kayboldukları görülebilir.



Şekil 3.3 (a)Terminal düğümlerinin indirgenmeleri, (b)Terminal olmayan düğümlerin indirgenmeleri, (c)Terminal olmayan gereksiz düğümlerin yok edilmesi (Bryant, 1992)

### 3.1.2 Shannon Açılımı

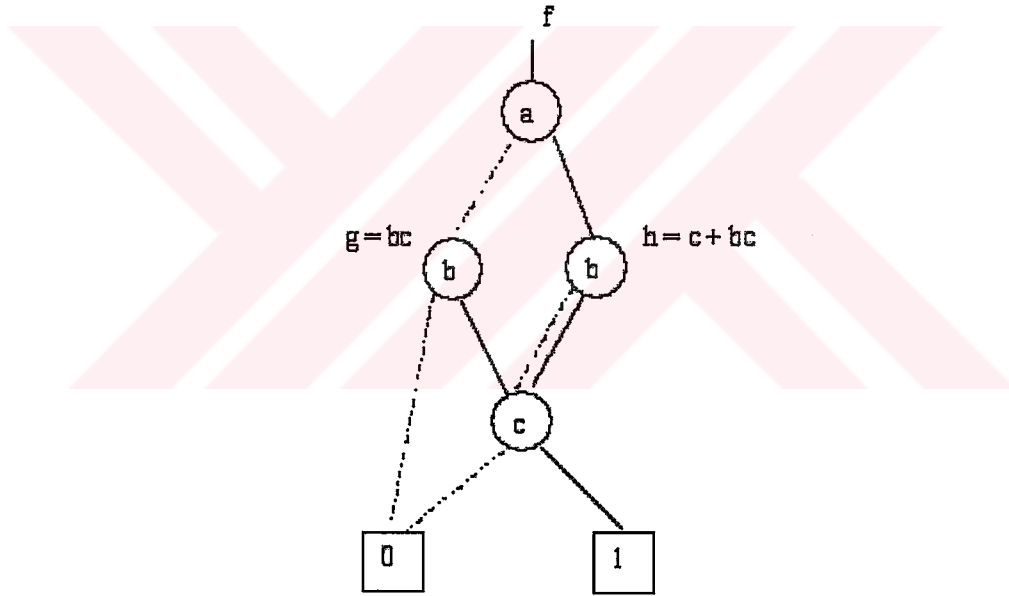
BDD oluşturulmasında, yinelemeli Shannon açılımından faydalanılır. (3.1) eşitliği ile lojik fonksiyonun  $x_i=0$  ve  $x_i=1$  için fonksiyon parçalarına ayrılarak seviyeler oluşturulur.

$$F = xf_x + x'f_{x'} \quad (3.1)$$

Örnek olarak,  $f=ac+bc$  fonksiyonuna ait BDD gösterimini, Shannon açılımı kullanarak elde edelim.

$$\left. \begin{aligned} f_{a'} &= f(a=0) = bc \\ f_a &= f(a=1) = c + bc \\ g_{b'} &= g(bc) |_{b=0} = 0 \\ g_b &= g(bc) |_{b=1} = c \\ h_{b'} &= h(c+bc) |_{b=0} = c \\ h_b &= h(c+bc) |_{b=1} = c \end{aligned} \right\} \quad (3.2)$$

(3.1)'de verilen Shannon açılımının, en son değişken seviyesine kadar yinelenerek kullanılması ile (3.2)'de gösterilen durum fonksiyonlarına ulaşılır. (3.2)'deki eşitlikler belli bir sıra çerçevesinde hesaplanır. Şekil (3.4)'de verilen  $f=ac+bc$  lojik fonksiyonuna ilişkin BDD gösterimlerinde, düğümler arası bağlantılar, kesikli ve düz çizgiler olmak üzere iki tipte bağlantı olarak ifade edilirler. Düz çizgiler, Shannon açılımında, ilgili değişkenin lojik 1 değerini alması sonucunda elde edilen fonksiyonu, kesikli çizgiler ise, Shannon açılımında, ilgili değişkenin lojik 0 değerini alması sonucunda elde edilen fonksiyonu çocuk düğümlere aktaracak şekilde bağlantıları oluşturur. BDD gösterimlerinde, bahsedilen çizgiler, ya terminal olmayan düğümler arasında bağlantılar kurarlar veya terminal olmayan düğümlerden, son durum olan terminal düğümleri arasında bağlantılar oluştururlar. En alt seviye olan, terminal düğümlerin yer aldığı seviye, BDD'nin tamamlandığı, diğer ifade ile Shannon açılımının, yinelenen yapısının sona erdirildiği noktadır.



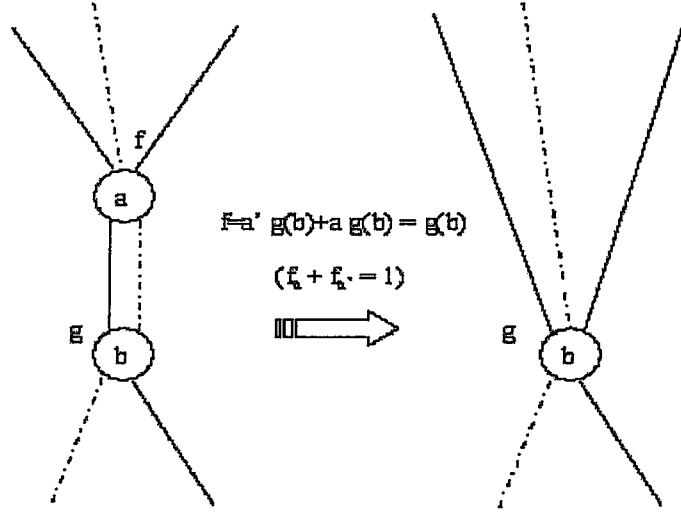
Şekil 3.4  $f = ac + bc$  fonksiyonuna ait BDD haritalaması

### 3.1.3 BDD İndirgeme Kuralları

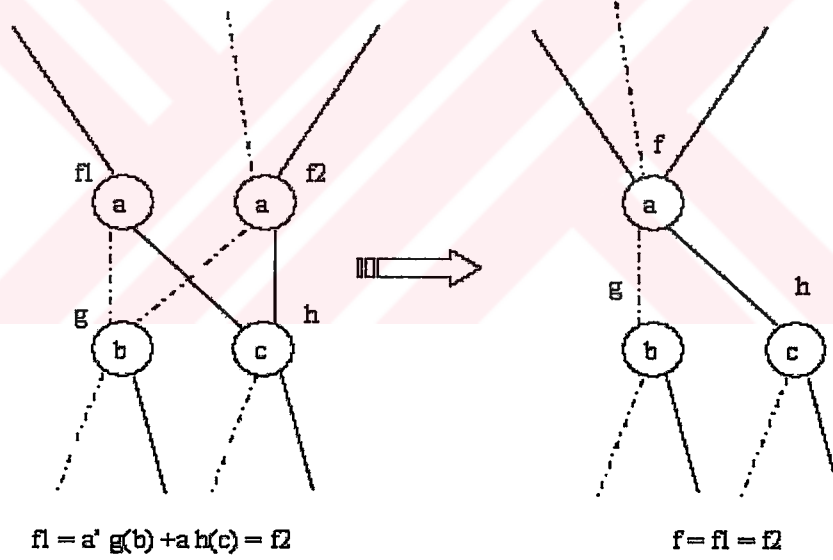
Temel olarak iki başlık altında toplamak mümkündür.

- 1) Fazlalık düğümlerin yok edilmesidir. (Terminal olmayan bir düğümün iki çocuk düğümünün eşlenik olması durumunda karşılaşılr.)
- 2) Aynı çocuk düğümlere simetrik bağlı ve aynı seviyede olan iki veya daha fazla sayıda düğümden birisini bırakacak şekilde diğerlerinin yok edilmeleridir.

Şekil 3.5 ve Şekil 3.6'da, sırasıyla birinci indirgeme kuralının şematik gösterimleri yer almaktadır.

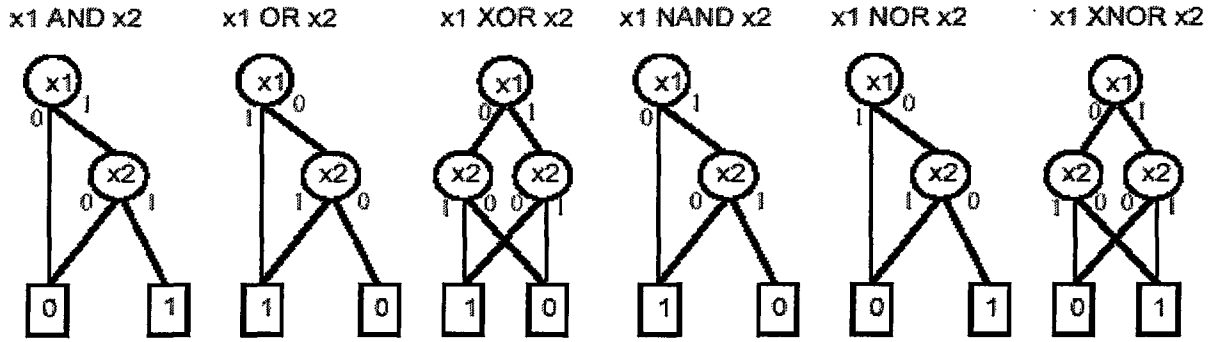


Şekil 3.5 Birinci indirgeme kuralının gösterimi



Şekil 3.6 İkinci indirgeme kuralının gösterimi

Şekil 3.7'de, birtakım temel lojik kapıların BDD gösterimlerine yer verilmiştir.



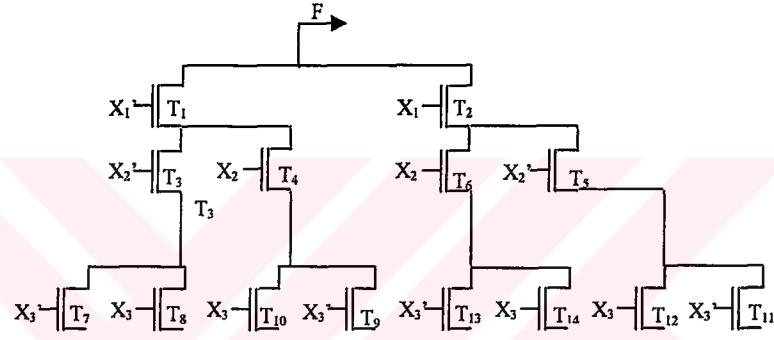
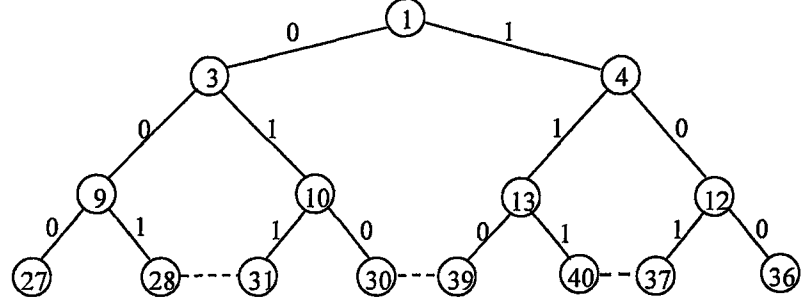
Şekil 3.7 Temel lojik kapılarının BDD gösterimleri [1]

### 3.2 BDD Kullanarak Sayısal Devre Modellenmesi

Lojik fonksiyonların, indirgenmesi ve modellenmesini birlikte ele alan BDD'ler, bu haliyle geçiş transistörü tasarımlarına olan uyumluluğu, yayımlanan birçok makale ve bildiride üzerinde durulmuştur. Çalışmaların birçoğu sadece, verilen bir BDD'yi ele alarak, bölüm 2'de bahsedilen, değişik geçiş transistörü modelleri ile nasıl modellenebileceği üzerinde durmuş ancak bir lojik fonksiyondan indirgenmiş karar diyagramının nasıl elde edileceği konusuna açıklık kazandırmamıştır. Bununla beraber, üzerinde durdukları karar diyagramları ve haliyle karşılık gelen devre yapısı, küçük yapılar olup, genelleştirmeler yapılmamıştır. Jaekel, Bandyopadhyay ve Jullien (1998), yaptıkları çalışmada, BDD'nin elde edilmesi konusunda her ne kadar doyurucu bilgiler vermiş olsalar da, BDD'yi oluşturan lojik fonksiyonun kodlanması hakkında açık bilgilere yer vermemişlerdir. Hazır bir kod kullanılarak, BDD'nin indirgenmesi ve yarı iletken elektronik devre elemanlarından oluşan devre yapısına nasıl model olabileceği verilmiştir (Şekil 3.8). Kodlama, BDD kullanarak indirgeme ve elektronik devre elemanları ile modelleme konularını bir bütün halinde ele alan bir çalışma Avcı, Yıldırım (2003) tarafından ele alınmış ve konu en başından sonuna kadar netlik kazanmıştır. Özellikle, son bahsettiğim çalışma, tez çalışmamın iskeletini oluşturmuştur. Jaekel, Bandyopadhyay ve Jullien (1998), yaptıkları çalışmada, oluşturdıkları BDD'dan yola çıkarak, dinamik sayısal devre tasarımına örnek vermiş, Avcı, Yıldırım (2003), çalışmalarında statik sayısal devre tasarım metodu önermişlerdir.

Lojik fonksiyonların, sadece Shannon açılımının tekrarlanması ile elde edilen, karakterler ile ifadesi, bilgisayar ortamında işlenmesi açısından çok elverişli değildir. Bu sebepten dolayı, Shannon açılımını bünyesinde barındıran bir kodlamanın ve bu kodlama vasıtası ile lojik fonksiyonları temsil etmenin gerekliliği açıklık kazanmaktadır. Avcı, Yıldırım (2003),

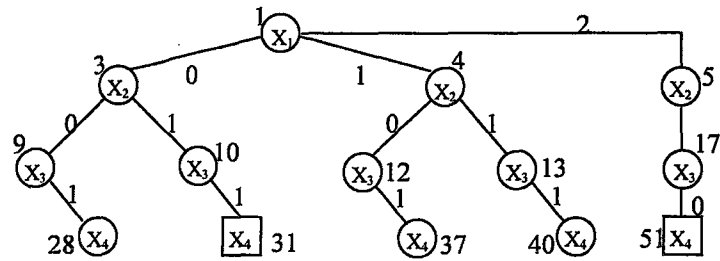
çalışmalarında, Shannon açılımını, bünyesinde gömülü olarak barındıran bir kodlama yapısını tanıtmışlardır. Kodlamanın, genel şekli, lojik fonksiyonu nasıl temsil ettiği ilerleyen bölümlerde detaylı bir şekilde incelenecektir.



Şekil 3.8 123 karar diyagramı ve geçiş transistörü ile gerçekleşmesi (Jaekel, 1998)

Şu noktadan itibaren, bu tez çalışması ile olan paralellikleri itibariyle, yukarıda bahsedilen iki çalışmanın, detaylı bir şekilde portresi çizilecektir.

### 3.2.1 123 Karar Diyagramı



Şekil 3.9 123 karar diyagramı gösterimi (Jaekel, 1998)

123 karar diyagramı, Şekil 3.9'da verilen grafiksel gösterim örneğinde olduğu gibi, grafiksel bir gösterim ve bir de, düğümlerin birbirleri ile olan statülerini ve yan yana olma durumlarını

tuttuğu bir liste olmak üzere iki parçadan oluştuğunu söyleyebiliriz. Grafikselsel gösterim, düğümlerin, düğümler transistörlere model olacağına göre, sonuçta transistörlerin birbirleri ile nasıl bağlandıkları bilgisini bizlere grafikselsel olarak verirken, liste ise, transistörlerin, yerleştirilmeleri öncesinde, transistör yerleşimine model olacak düğümlerin yerleşim planları üzerinde mümkün ve iyileştirme amaçlı oynamalar yapmak için tanımlanmıştır. Grafikselsel gösterim üzerinde yer alan elemanlar sırasıyla, terminal düğümleri, terminal olmayan düğümler ve düğümleri bağlayan bağlantılardır. 123 karar diyagramı içerisinde, terminal olmayan düğümler, maksimum üç kenar oluşturacak şekilde düşünülmüşlerdir. Bunlar kenarlar sırası ile 0,1 ve 2 olacak şekilde numaralandırılmışlardır. 0 ve 1 numaralı kenarlar, genel BDD yaklaşımının tıpatıp aynısıdır. 2 numaralı kenar ve haliyle çocuk düğüme olan bağlantı, 123 karar diyagramının getirdiği bir yeniliktir. Aslında, sadece yerleşimde keyfiyet sağlamak ve transistör devresinin tasarımına karar diyagramı içerisinde bir adım daha yaklaşabilmek adına tanımlanan üçüncü bir kenardır. Bu kenar, bizlere yabancı ve genel BDD yapısı içerisinde tanımlanmamış kenar, birazdan liste yapısı içerisinde açıklık kazanacaktır.

Terminal olmayan düğümlerinin, grafikselsel gösterimde daireler olarak sembolize edildiğinin de altını çizdikten sonra, diğersel bir yapı olan, terminal düğümleri açıklayalım. Terminal düğümleri, kareler şeklinde gösterilen yapılar, üzerlerinde etiketlenmiş geçiş değişkenini gösterirler. Geçiş değişkeni, lojik 0 referans, lojik 1 referans, ilgili seviye kaynak veya ilgili seviye kaynağın tümleyenini olabilir. Bu dört alternatiften hangisinin konumlandırılacağı, indirgeme sonrası, terminal düğüm durumu ile yakından ilgilidir. Her düğüm bünyesinde bir kod veya kod parçasını barındırır. Eğer terminal düğümü, kendisine ait kod parçası tüm 0'lerden ibaret ise, üzerinde göstereceği etiket, yani geçiş değişkeni, lojik 0 referans olacaktır. Tüm 1'lerden ibaret bir koda sahip olsaydı, bu yeni durumda, o terminal düğümünün üzerindeki etiketin lojik 1 referans olacağını söylememiz gerekirdi.

Grafikselsel gösterim, terminal olmayan düğümlerin, geleneksel BDD yaklaşımına göre, fazladan bir tane komşuluklarının olması dışında temel olarak oldukça örtüşmektedir. Bu noktada, terminal olmayan düğümler için, üç komşuluğunun olması bir şart değildir veya her terminal olmayan düğümün üç adet komşuluğı olacaktır demek doğru değildir. Terminal olmayan düğümler, sadece bir kenar da oluşturabilirler, bu rakam iki veya üç adet de olabilir. Bu tamamen ilgili düğümün barındırdığı kod parçası ve diğersel düğümler ve onların barındırdıkları kod parçaları ile olan ilişkilerine bağlı bir durum olarak karşımıza çıkar.

123 karar diyagramında, i, bulunulan seviyeyi ve X ise, o seviyeye karşılık gelen kaynağı sembolize edecek şekilde, 123 karar diyagramında, terminal ve terminal olmayan düğümler,

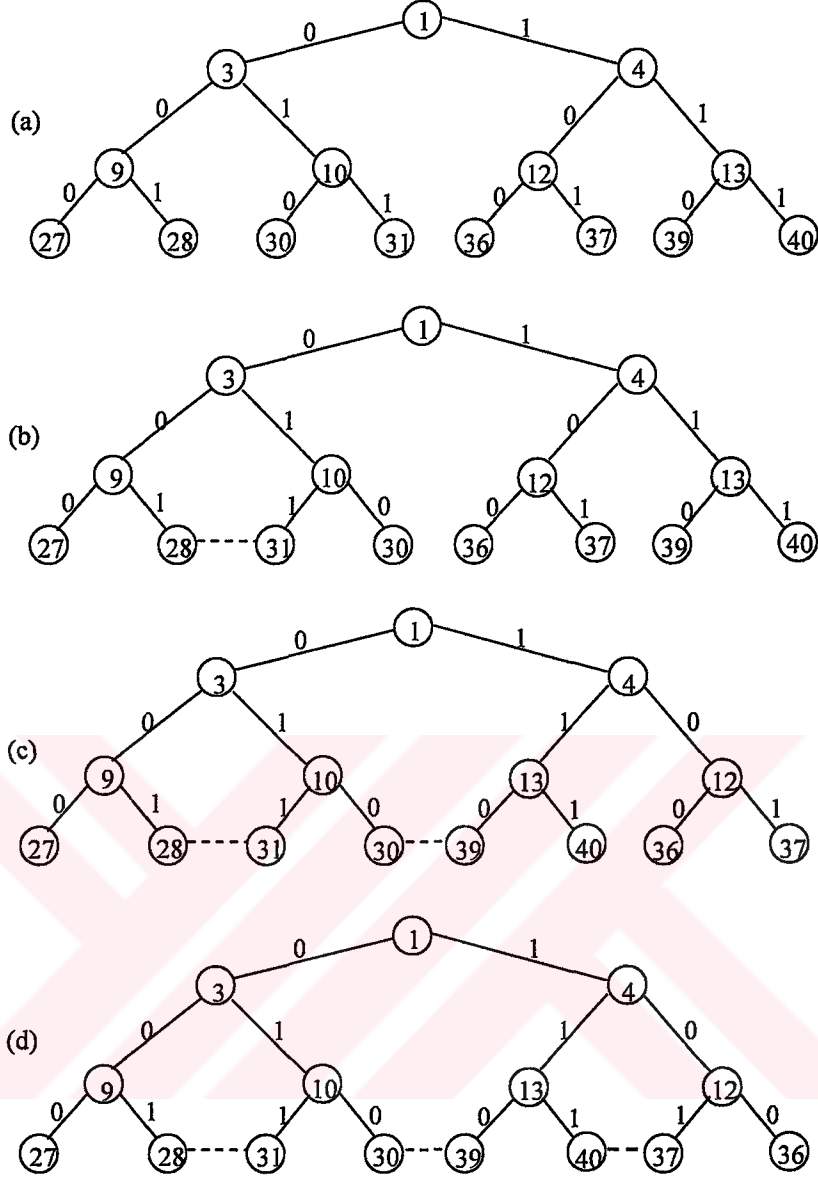
Xi'ler şeklinde etiketlenmişlerdir. Düğümler içerisinde etiketlenmiş bu Xi değerlerinin, Şekil 3.8'de nasıl transistörlerin gate uçlarına bağlandıklarını ve bu sayede transistörleri açıp kapamak suretiyle kontrol ettiklerini görebiliriz. Bu durum terminal olmayan düğümler için geçerlidir. Yukarıda da bahsedildiği gibi, eğer etiketlenen düğüm bir terminal düğümü ise, bu durumda, Xi'ler referans belirteceklerdir. 123 karar diyagramları bünyesinde bulunan düğümler, Xi etiketleri haricinde, her bir düğüm için farklı değerlerde bir düğüm numarası ile gösterilirler. Bunun basit bir formülü verilebilir. Birinci seviye olan en üst seviyede, ilk düğüm her zaman düğüm numarası olarak 1 değerini almak koşulu ile, kenarları e ile sembolize eder ve düğüm numarası da n ile gösterirsek, çocuk düğümlerin düğüm numaraları (3.3) eşitliği yardımı ile kolaylıkla hesaplayabiliriz.

$$(n+1) = 3*n + e \quad (3.3)$$

(3.3)'de, n+1 ile ifade edilen eşitliğin sol tarafı, çocuk düğümleri göstermektedir. Yine, e'nin 0 ve 1 değerlerini aldıklarını, bu sayede iki adet çocuk düğüme karşılık gelen iki farklı düğüm numarası elde edildiğini de vurgulamak gerekir.

Bundan sonraki bölümde, grafiksel olarak tanıtilan 123 karar diyagramının, liste yaklaşımından bahsedilecektir. 123 karar diyagramını, geleneksel yaklaşımdan ayıran ve farklı kılan en önemli husus, getirdiği listeleme yaklaşımıdır. Bu sayede, sayısal devre tasarımına son derece yakın bir karar diyagramını ortaya koymaktadır.

123 karar diyagramı listeleme yaklaşımı, esas olarak, bir i seviyesine ilişkin düğümlerin listelenmesidir. Bunu, ileride bahsedilecek olan indirgeme işlemleri esnasında, düğümlerin pozisyonlarını da mümkün mertebe sayısal devre mimarisine uygun hale getirebilmek ve transistörlerin yerleşimine ilişkin bilgileri depolamak için yapar. Şekil 3.8'de, 123 karar diyagramında, alt seviyede yer alan düğümlerin, birbirlerine kesikli çizgiler ile bağlandıkları görülmektedir. Bunun anlamı, bu düğümlerin, kendi altlarında oluşturacakları, karar diyagramlarının birbirinin aynısı olması, karşılık gelen fonksiyonların aynı lojik fonksiyonlar olması veya kesikli çizgiler ile birleştirilen düğümlere ait kodların örtüşmesidir. Bu düğümler, ileride indirgenecek olan düğümlerdendir. N adet aynı seviyede, aynı koda sahip düğüm var ise, bir tanesi dışında diğerlerinin yok edilmesi ile, birbirlerini tekrarlayan blokların gereksiz yere yer işgal etmelerinin önüne geçilmiş olacaktır. İki tip listeden bahsedilmektedir. İlki, köşeli parantez içerisinde, diğeri ise küme parantezi içerisinde olacak şekilde gösterilmiştir. İlk tipte, listeleniş sırası önemlidir. İkinci tipte, sıralanış açısından tamamen keyfiyet söz konusudur. Liste, gerek birinci tip liste, gerekse ikinci tip liste olsun, listelenen elemanlar,



Şekil 3.10 Liste gösterimi akışı (Jaekel, 1998)

bir liste, mevcut bir düğüm veya hayali bir düğüm olabilir (hayali düğüm, literatürde dummy node olarak verilmiştir). Hayali düğüm için yer ayrılmasının nedeni, transistör tasarımına maksimum uygunluk için, yeni yerleşim planlanmasında önceden düğümler için boşluklar tanımlamasıdır. Tekrar listelere dönmek gerekirse, ilk listeleme tipine örnek olarak, listemiz  $L=[3,[4,5],6]$  şeklinde olsun. Bu listelemeye ait geçerli yerleşim pozisyonları şu şekilde oluşur.

- 3 4 5 6    6 5 4 3    3 5 4 6    6 4 5 3

Bu listeleme şeklinde kural olarak, listeler ya verilen sıralama ile veya verilen sıralamanın

tam tersi sıra ile sıralanırlar. Bu tanımlama ile 4 ve 5 numaralı düğümlerin niçin hep ortada kaldıkları, en sola veya sağa kaymadıkları görülmektedir. Her ne kadar, 4 ve 5 kendi içerisinde düz sıralı veya ters sıralı olarak bulunabiliyorsa da birinci tip liste, örnekte verilen listeye göre, 4 ve 5'i tek bir eleman olarak kabul eder ve tanımı itibariyle ne en sola, ne de en sağa yerleştirir. Benzer şekilde, 3 ve 6'da birinci liste modeline ve verilen örneğe göre, hiçbir zaman ortada bulunamaz. Ya en solda veya en sağda olmak zorundadırlar.

İkinci tip liste ise, elemanların tamamen serbest bir şekilde sıralanabildiği liste yapısıdır. İkinci tip listeye ilişkin örneğimiz ise,  $L=\{3,[4,5],6\}$  olsun. İkinci tip listelemeye göre oluşabilecek muhtemel yerleşimler ise şu şekildedir.

- 3 4 5 6    3 6 4 5    4 5 3 6    4 5 6 3    6 4 5 3  
           3 5 4 6    3 6 5 4    5 4 3 6    6 3 5 4    6 5 4 3

İkinci tip listelemede, yine birinci tip listeleme ile listelenmiş, köşeli parantez içerisine alınmış yapı, tek bir eleman olarak değerlendirilmiş, ancak bu sefer,  $[4,5]$  tek bir eleman göstermek ve kendi içerisinde sıralı ve ters sıralı olacak şekilde sıralanabilecekken, bir grup halinde, listenin en sağ, en solu veya ortasında bulunabilmektedir. Şu aşamadan sonra, Şekil 3.10'da verilen 123 karar diyagramlarına ait, listeleri oluşturalım. İlk grafik, Şekil 3.10(a)'ya ilişkin düğümler işlenmemiş geçici bir liste şeklini oluştururlar (3.4). d alt indisi ile belirtilen düğümler, şekilde yer almayan düğümler olmasına karşın, adresleri diğer bir ifade ile kullandıkları alan sonra kullanılabilme ihtimaline karşı tutulmaktadır. Her bir düğüm, beraberinde üç adet düğümü alacak şekilde paketlenmiştir. Şekil 3.10(b)'de, kesikli çizgiler ile gösterilen 28 ve 31 numaralı düğümlerin birleşeceklerini ifade eder. Bu halde yeni durum için yeni bir liste hazırlanmalıdır. Oluşacak yeni liste, (3.5)'de verilmiştir.

$$L1=\{\{27,28,29_d\},\{30,31,32_d\},\{33_d,34_d,35_d\},\{36,37,38_d\},\{39,40,41_d\},\{42_d,43_d,44_d\},\{45_d,46_d,47_d\},\{48_d,49_d,50_d\},\{51_d,52_d,53_d\}\} \quad (3.4)$$

$$L2=\{[\{27,29_d\},28,31,\{30,32_d\}]\{33_d,34_d,35_d\},\{36,37,38_d\},\{39,40,41_d\},\{42_d,43_d,44_d\},\{45_d,46_d,47_d\},\{48_d,49_d,50_d\},\{51_d,52_d,53_d\}\} \quad (3.5)$$

(3.5)'de liste1 kuralına bağlı kalarak, 28 ve 31 numaralı düğümlerin listede daima komşu kalacaklarını söylemeliyiz. Şekil 3.10(c)'de 30 ve 39 numaralı düğümleri de birleştirirsek, oluşacak liste (3.6)'daki gibi olur.

$$L3=\{[\{33_d,34_d,35_d\},\{27,29_d\},28,31,32_d,30,39,\{40,41_d\}]\{36,37,38_d\},\{42_d,43_d,44_d\}\},\{45_d,46_d,47_d\},\{48_d,49_d,50_d\},\{51_d,52_d,53_d\}\} \quad (3.6)$$

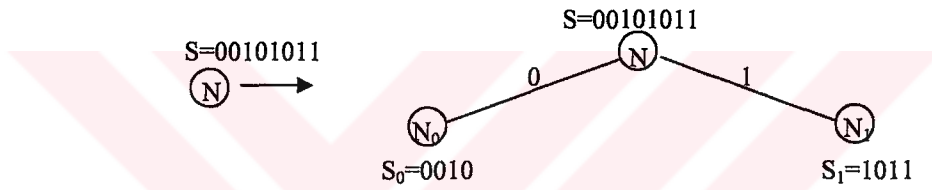
37 ve 40 numaralı düğümleri de organize ederek yerleşimlerini ayarlayabiliriz ancak, 36 ve 27 numaralı düğümlerin, komşuluk oluşturmaları, (3.7)'den sonra mümkün olamayacaktır.

$$L4 = \{ \{ \{ 33_d, 34_d, 35_d \}, \{ 27, 29_d \}, 28, 31, 32_d, 30, 39, 41_d, 40, 37, \{ 36, 38_d \}, \{ 42_d, 43_d, 44_d \} \}, \{ \{ 45_d, 46_d, 47_d \}, \{ 48_d, 49_d, 50_d \}, \{ 51_d, 52_d, 53_d \} \} \} \quad (3.7)$$

Bu dört listeleme sonrası oluşan karar diyagramı, Şekil 3.10(d)'de verilmiştir.

123 karar diyagramlarının düğümlerinin açılım ve indirgenmelerine ilişkin kurallar tanımlanmıştır. Bu kurallar aşağıda sıralanarak açıklanmıştır.

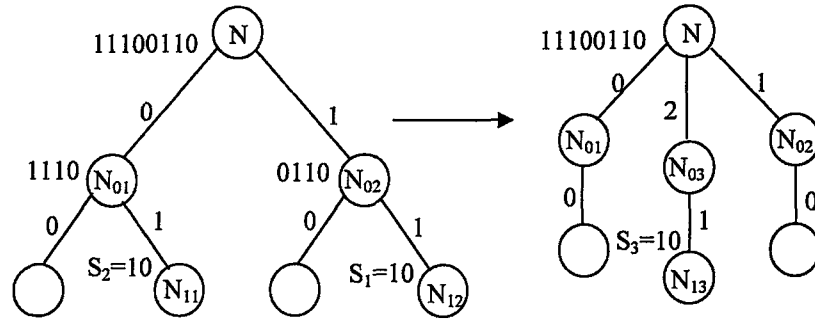
- Çocuk düğüm oluşturma : Terminal olmayan, i. Seviyede bulunan ve 2m uzunluklu bir koda sahip bir düğüm, iki adet, (i+1). seviyede, m uzunluklu koda sahip çocuk oluşturur. Çocuk düğüm oluşturacak olan düğüm, çocuk düğümler ile 0(1) ile etiketlenen yol üzerinden bağlanır (Şekil 3.11).



Şekil 3.11 Çocuk düğüm oluşturma (Jaekel, 1998)

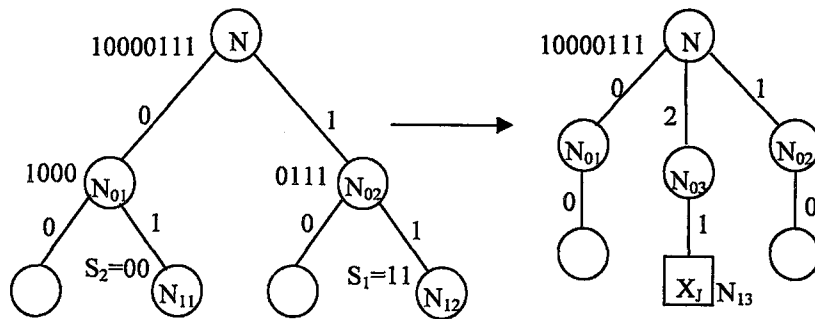
- Yolların birleştirilmesi 1 :  $p_1 = [N - e_{01} - N_{01} - e_{11} - N_{11} - \dots - e_{k1} - N_{k1}]$  ve  $p_2 = [N - e_{02} - N_{02} - e_{12} - N_{12} - \dots - e_{k2} - N_{k2}]$  şeklinde, N düğümünden, N düğümünden farklı ama kendileri aynı seviyede olan  $N_{k1}$  ve  $N_{k2}$  düğümlerine iki yol var ise ve bu yollar ve üzerlerindeki düğümler aşağıda tanımlanan kriterlere uyuyorlar ise, yollar birleştirilebilir.  $N_{k1}$  ve  $N_{k2}$ , sırasıyla,  $S_1$  ve  $S_2$  uzunluklu kodlara sahip iki düğüm ise;
  - $e_{01}(e_{02})$ , etiketleri 0(1);
  - tüm i'ler,  $1 < i \leq k$  için,  $e_{i1}$ 'in etiketi =  $e_{i2}$ 'nin etiketi ise;
  - $S_1 = S_2$  ise;
  - $p_1$  yolu üzerindeki her düğüm,  $N_{01}, N_{11}, \dots, N_{k1}$  ve  $p_2$  yolu üzerindeki her düğüm,  $N_{02}, N_{12}, \dots, N_{k2}$  sadece bir baba düğüme bağlı iseler;
    - $e_{03}$  etiketi=2;
    - tüm i'ler,  $1 < i \leq k$  için,  $e_{i3} = e_{i2}$ ;
    - $S_3 = S_2$ ;
  - $p_1$  ve  $p_2$  yolları birleştirilerek,  $p_3 = [N - e_{03} - N_{03} - e_{13} - N_{13} - \dots - e_{k3} - N_{k3}]$  elde

edilir (Şekil 3.12).



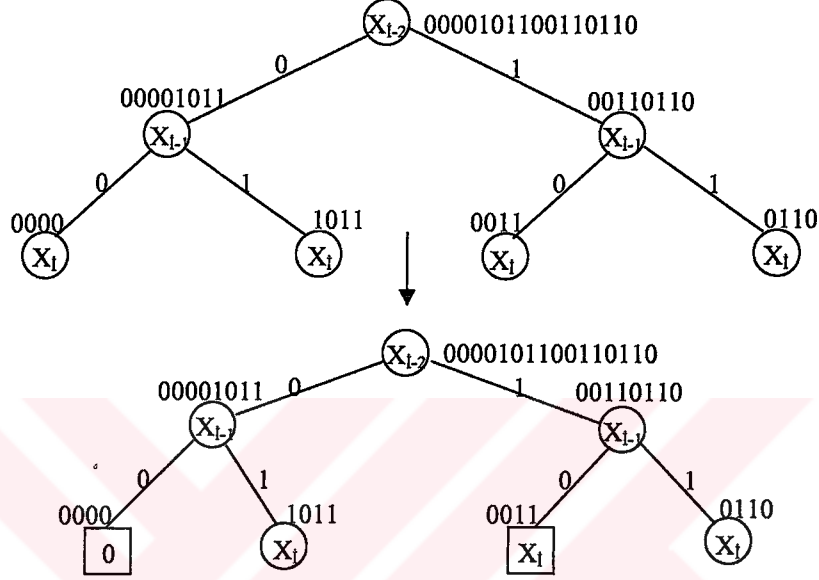
Şekil 3.12 Yolların birleştirilmesi 1 (Jaekel, 1998)

- Yolların birleştirilmesi 2 :  $p_1 = [N - e_{01} - N_{01} - e_{11} - N_{11} - \dots - e_{k1} - N_{k1}]$  ve  $p_2 = [N - e_{02} - N_{02} - e_{12} - N_{12} - \dots - e_{k2} - N_{k2}]$  şeklinde, N düğümünden farklı, j. seviyede olan,  $X_j$  ile etiketli ve  $S_1$  ve  $S_2$  kodlarına sahip,  $N_{k1}$  ve  $N_{k2}$  düğümlerine iki yol var ve aşağıda sıralanan özellikleri sağlıyor ise;
  - $e_{01}(e_{02})$ , etiketleri 0(1);
  - tüm i'ler,  $1 < i \leq k$  için,  $e_{i1}$ 'in etiketi =  $e_{i2}$ 'nin etiketi ise;
  - $S_1$  tüm sıfır(tüm bir) ve  $S_2$  tüm bir(tüm sıfır)'lardan oluşuyor ise;
  - $p_1$  yolu üzerindeki her düğüm,  $N_{01}, N_{11}, \dots, N_{k1}$  ve  $p_2$  yolu üzerindeki her düğüm,  $N_{02}, N_{12}, \dots, N_{k2}$  sadece bir baba düğüme bağlı iseler;
  - $e_{03}$  etiketi=2;
  - tüm i'ler,  $1 < i \leq k$  için,  $e_{i3}=e_{i2}$ ;
  - $N_{k3}, X_j(X_j)$  terminal düğüme dönüşecek şekilde,
  - $p_1$  ve  $p_2$  yolları birleştirilerek,  $p_3 = [N - e_{03} - N_{03} - e_{13} - N_{13} - \dots - e_{k3} - N_{k3}]$  elde edilir (Şekil 3.13).



Şekil 3.13 Yolların birleştirilmesi 2 (Jaekel, 1998)

- Terminal olmayan düğümün, terminal düğüme dönüştürülmesi :
  - Eğer, terminal olmayan bir düğüme ait kod, tüm sıfırlardan (tüm birlerden) ibaret ise, o düğüm terminal düğümü olarak değiştirilir ve sıfır (bir) olarak etiketlenir.
  - Eğer, terminal olmayan bir düğüme ait  $2m$  uzunluklu bir kod, ilk  $m$  uzunluklu kısmı, tüm sıfır (tüm bir) ve takip eden ikinci  $m$  uzunluklu parçası, tüm bir (tüm sıfır) şeklinde ise, düğüm terminal düğümü olarak değiştirilir ve  $X_i(X_i')$  olarak etiketlenir (Şekil 3.14).

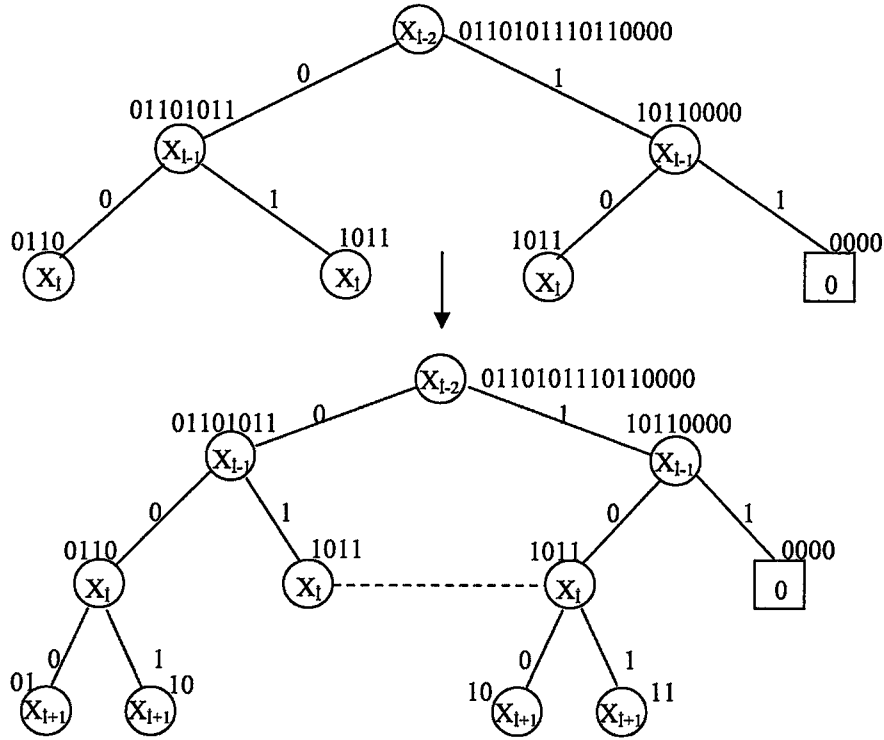


Şekil 3.14 Terminal olmayan düğümün, terminal düğüme dönüştürülmesi (Jaekel, 1998)

- Alt kümelerin birleştirilmesi : Eğer, herhangi bir  $i$  seviyesindeki düğüme ait kod, aynı seviyedeki başka bir düğüme ait kod ile aynı ise, o düğüm ve tüm çocukları indirgenir, o düğüm, aynı seviyedeki ve aynı koda sahip diğer düğüm ile birleştirilir (Şekil 3.15).

Yukarıdaki bölümde, grafiksel gösterimi, liste yapısı gösterimi ve indirgeme kuralları ile birlikte verilen 123 karar diyagramları, aşağıda tüm bir örnek ile sonlandırılacaktır.

Şekil 3.16'da verilen, oluşturma ve indirgeme aşamalarının gösterildiği 123 karar diyagramını, beş değişken içeren bir lojik fonksiyona ilişkin, 32 uzunluklu bir kod oluşturmaktadır.



Şekil 3.15 Alt kümelerin birleştirilmesi (Jaekel, 1998)

İlk düğüm, 1 numarası ile etiketlenmiş ve tüm, 32 uzunluklu kodu bünyesinde tutmaktadır (Şekil 3.16(a)). 1 numaralı düğüm, (3.3) bağıntısına bağlı kalarak, 3 ve 4 numaraları ile etiketlenen iki adet çocuk düğüm oluşturur (Şekil 3.16(b)). (3.3) kullanılarak, ilk dört seviye, benzer şekilde oluşturulur (Şekil 3.16(c)). Her seviyedeki düğümler, bu örnekte 32 uzunluklu olan kodu,  $2^{(seviye-1)}$  ile bölerek paylaşırlar. Buna göre, Şekil 3.16(c)'de, 30 numaralı düğüme ait kod parçası, tüm kodun sekiz eşit parçaya ayrıldıktan sonra elde edilen sekiz eşit uzunluklu kod parçasının üçüncü parçası olacağı açıktır. Ayrıca, herhangi bir seviyede tüm düğümlere ait kodlar, arka arkaya eklendiklerinde, 32 uzunluklu tüm kodun elde edildiğine dikkat edilmelidir. İlk dört seviye açılımından sonra yapılan listeleme gösterimi aşağıdaki gibidir.

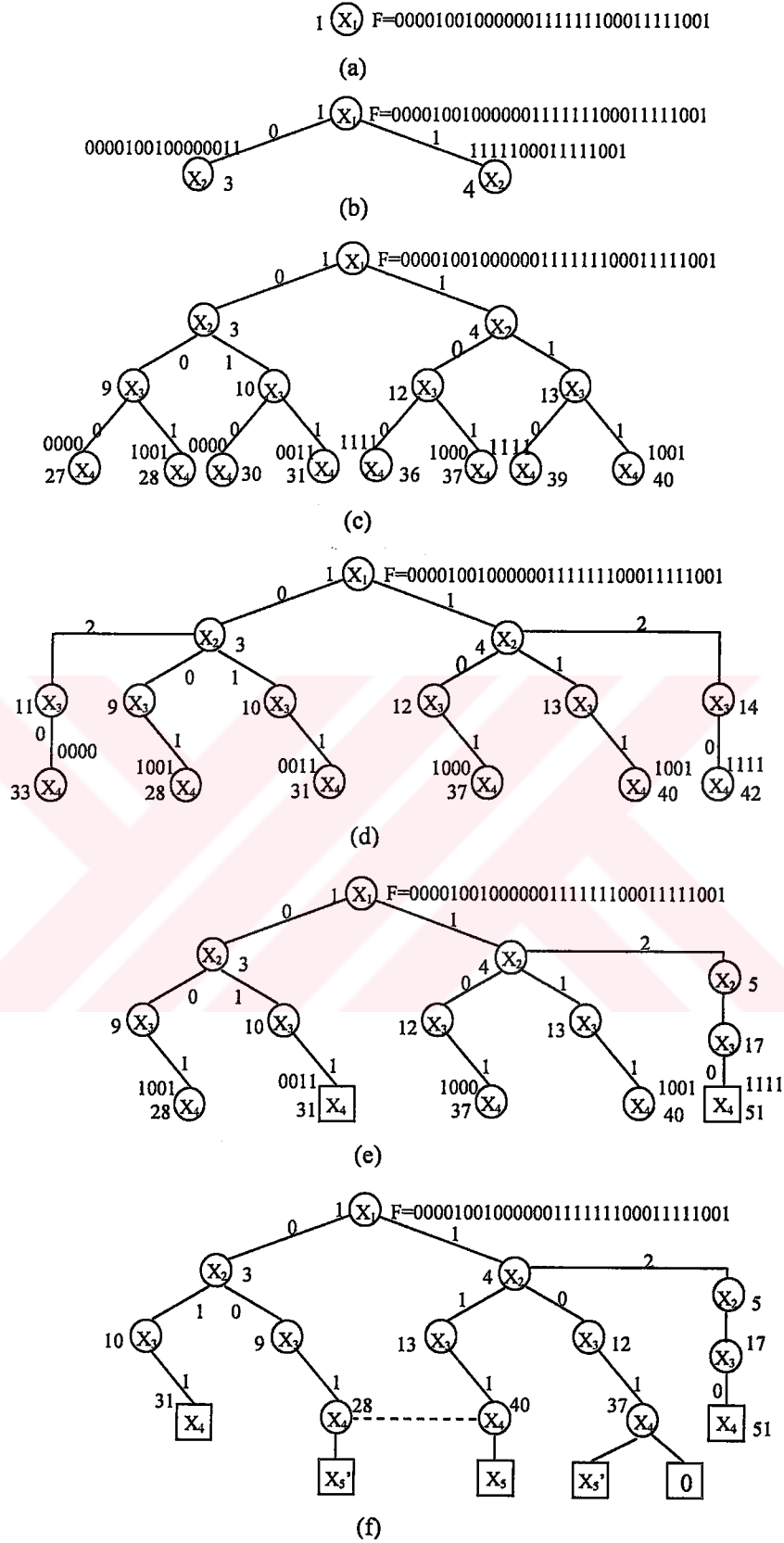
$$PL1 = \{ \{ \{ 27, 28, 29_d \}, \{ 30, 31, 32_d \}, \{ 33_d, 34_d, 35_d \}, \{ 36, 37, 38_d \}, \{ 39, 40, 41_d \}, \{ 42_d, 43_d, 44_d \} \}, \{ \{ 45_d, 46_d, 47_d \}, \{ 48_d, 49_d, 50_d \}, \{ 51_d, 52_d, 53_d \} \} \} \text{ (Jaekel, 1998).}$$

Şekil 3.16(c)'de, 27, 30 ve 36, 39 düğümlerinin eşit kodlara sahip olduklarını gösmekteyiz. O halde, yolların birleştirilmesi 1 uygulanarak, 27, 30 ve 36, 39 indirgenerek, 33 ve 42 numaralı düğümler tanımlanarak, bu düğümlere atamaları yapılabilir (Şekil 3.16(d)). 33 ve 42 numaralı düğümler, şu aşamaya kadar, liste yapısından da görülebileceği üzere, d ile indislenmiş hayali düğümler idi ve şekilde görünmemektedirler. Bu andan itibaren 33 numaralı düğüm, 27 ve 30 numaralı düğümleri, 42 numaralı düğüm ise, 36 ve 39 numaralı düğümleri tek başlarına

sembolize gerçeklemektedirler. Hiç şüphe yokki bu durumda, iki düğüm arası bağlantıya denk gelen transistörlerden, iki adet kazanmış oluyoruz. Şekil 3.16(d)'de, 2 numaralı kenar ve bağlantı, metal bağlantıya denk gelmektedir. Bu aşamadan sonra, yolların birleştirilmesi 2, 42 ve 33 numaralı düğümlere uygulanırsa, 33 ve 42 numaralı düğümleri, tek başına gerçekleyebilecek 51 nolu düğüm tanımlanır ve 33 ve 42 kaybolur. 51 liste yapısında hayali bir düğüm idi, şu aşamadan sonra, şekildeki yerini almıştır (Şekil 3.16(e)). Şu durumda, 51 numaralı düğümün, 33 ve 42 nolu düğümlerin yerini, dolayısıyla, 27,30,36 ve 39 numaralı düğümlerin yerini aldığını ve tek bir düğümlle gösterdiğini görmekteyiz. Şimdide, 28 ve 40 numaralı düğümlere ve sahip oldukları kodlara bir bakalım. Aynı koda sahip oldukları görülmekte olan bu iki düğüme, alt kümelerin birleştirilmesi uygulanırsa, Şekil 3.16(f) elde edilir. Şekil 3.16(e)'de, birbirlerine komşu olmayan 28 ve 40 numaralı düğümler, bir üst seviyeden, 3 ve 4 numaralı düğümlerin, 180 derece eksenleri etrafında döndürülmesi ile komşulukları sağlanabilir (Şekil 3.16(f)). En son hali ile, Şekil 3.16(f)'deki yapıya ilişkin liste aşağıda verilmiştir.

$PL2 = [ \{ \{31,32_d\}, \{34_d,35_d\} \}, 29_d, 28, 40, 41_d, \{ \{37,38_d\}, \{43_d,44_d\} \}, \{ \{45_d,46_d,47_d\}, \{48_d,49_d,50_d\} \}, \{51,52_d,53_d\} \} ]$  (Jaekel, 1998).

Sonuç olarak, Jaekel, Bandyopadhyay ve Jullien (1998) yaptıkları çalışma ile BDD ile büyük benzerlikler içeren, 123 karar diyagramını tanıtmışlar, beraberinde liste gösterimini de eklemişlerdir. Bir noktasının açıklık kazanmaması dışında bu çalışma oldukça başarılı bir şekilde geçiş transistörü ile sayısal devre dizaynına model oluşturmaktadır. Açıklık kazanmayan bu nokta, yukarıda bahsedildiği gibi, kodlamanın nasıl yapıldığına yer verilmemiş olmasıdır. Bundan sonraki bölüm içerisinde, kodlamanın nasıl yapıldığına da atıfta bulunan, Avcı ve Yıldırım (2003)'ün çalışmaları ve önerdikleri karar modeli incelenecektir.



Şekil 3.16 Beş değişkenli, 00001001000000111111100011111001 ile kodlanmış lojik fonksiyona ilişkin, 123 karar diyagramı oluşturma ve indirgeme basamakları (Jaekel, 1998)

### 3.2.2 CPL-TG Yapısı İçin Genelleştirilmiş Karar Diyagramı Modeli

Avcı ve Yıldırım (2003) tarafından önerilen bu yaklaşım, kodlamayı da bünyesinde bulunduran, geçiş transistörü sayısal devre tasarımına model olan karar diyagramını ve bu karar diyagramından, CPL-TG yapısına geçişi içermektedir. CPL-TG yapısı içerisinde, bölüm 2’de de bahsedildiği gibi, çoğullayıcı yapısı ile ardışıl olmayan sayısal devre sentezlenmesine atıfta bulunulmuştur.

Bahsedilen bu çalışma, bir bütün olarak, bu tez çalışmasına şeklini veren model olmuştur. Bu çalışmanın detaylarına geçmeden hemen önce, bazı tanımlamaların verilmesinde fayda vardır.

- Geçiş değişkeni : Geçiş değişkeni olarak isimlendirilen değişken, bir lojik fonksiyona ilişkin değişkenler içinde, en yüksek ASCII koduna karşılık gelen değişkendir. Mesela, değişkenleri  $a, b$  ve  $c$  şeklinde ve minterim toplamları şeklinde verilen bir lojik fonksiyona ilişkin geçiş değişkeninin  $c$  olduğunu söyleyebiliriz. Geçiş değişkenleri çoğullayıcı yapısı içerisinde, referans belirtirler, yani, toprak gerilim seviyesi olsun, Vdd gerilimi olsun, nasıl referans belirtiyorlarsa, geçiş değişkeni ve tümleyeni de referans belirtirler.
- Kontrol değişkeni : Bir lojik fonksiyon içerisinde, geçiş değişkeni dışında kalan tüm değişkenler, kontrol değişkenleridir. Yukarıda verilen örnek için,  $a$  ve  $b$  kontrol değişkenleridir. Kontrol değişkenleri, TG’lerin gate uçlarına bağlantılar yaparak, TG’lerin iletimde veya kesimde olmalarını kontrol ederler. Kontrol ve geçiş değişkenleri, işaretlerin kendileri bağımsız bir kaynaktan ve tümleyenleri ise, evirici çıkışından elde edilirler.

Verilen bir lojik fonksiyondan, geçiş kapılarından (TG) oluşan devre yapısına geçiş ve doğrulanması sırasıyla, kodlama, karar diyagramı ve indirgeme, gerçekleştirme adımlarından ibarettir.

- Lojik fonksiyonların kodlanması : Lojik fonksiyonlardan, BDD yapısı içerisine uygulanacak kodun elde edilmesinde ilk adım, ilgili lojik fonksiyonun incelenerek, geçiş değişkeni ve kontrol değişkeninin (değişkenlerinin) belirlenmesidir. Geçiş değişkeni belirlenen lojik fonksiyon, geçiş değişkeninin çıkarılarak, yerine referansları sembolize edecek  $k_n$ ’lerin belli bir sıra ile yerleştirilmesi ile yeniden organize edilir. (3.8) ve (3.9) bu geçişlerin nasıl gerçekleştirildiğini göstermektedir.  $k$  ile gösterilen ve 1’den 4’e kadar indislenmiş değişkenlerin belli bir sıra ile verildiği, Şekil 3.17 ve (3.9)’dan görülmektedir.

$$F = a'b'c + ab' + abc' \quad (3.8)$$

$$F = a'b'k_1 + a'bk_2 + ab'k_3 + abk_4 \quad (3.9)$$

Kodlamanın yapılabilmesi için, bir doğruluk tablosundan faydalanılır. (3.8)’e ilişkin doğruluk tablosu, Şekil 3.17’de gösterilmiştir. Kodlama, doğruluk tablosunun ilk satırından

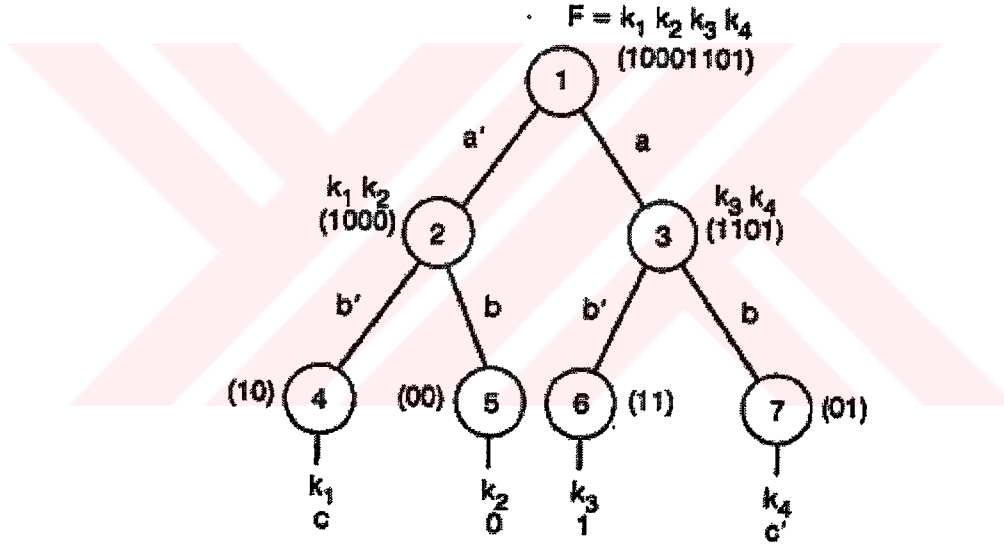
başlanılarak ve lojik fonksiyonun yapısı da beraber düşünülerek yapılır. Tek tek bahsetmek gerekirse, çarpılar toplamı şeklinde gösterilen F fonksiyonu (3.9), bize sırası ile (3.8) yapısı içerisinde hangi terimleri aramamız gerektiğini ifade etmektedir. Buna göre, araştırılması gereken ilk yapı,  $a'b'k_1$ 'dir. Bu yapıya ilişkin, (3.8) içerisindeki teriminin,  $a'b'c$  olduğu görülmektedir. Burada  $k_1$ 'e karşılık gelen değişken c değişkenidir. O halde tabloya gidilerek c'ye ilişkin kod araştırılmalıdır. Tabloya bakıldığında (Şekil 3.17) ilk terime ait kod parçası, 10 olduğu görülmektedir. Bu işlemi üç kez daha tekrarlamamız gerekir. Bu işlemin kaç defa yapılacağı tamamen lojik fonksiyonun değişken sayısı ile ilişkilidir. N değişkenli bir lojik fonksiyon için bu sayı  $2^{(N-1)}$  olarak verilebilir. (3.9)'a tekrar dönerek, ikinci araştırmamız gereken terimin  $a'bk_2$  olduğunu görebiliriz. (3.8) yapısına baktığımızda, incelememiz gereken terim,  $a'b$  yapısı olmalıdır. Ancak, (3.8)'e baktığımızda, böyle bir terimin olmadığını görmekteyiz. Bunun anlamı,  $a'b$  teriminin 0 referansını işaret etmesidir. Diğer bir deyişle bu terim  $a'b0$  şeklinde gizli bir şekilde, fonksiyon içerisinde vardır. O halde, bu terim için referans 0'dır ve tablodan 0 için tanımlanan kod bu terimin kodu olur ve  $a'b$  için kod 00 olarak yazılır. İlk iki adım sonunda elde edilen kod, birinci terimin kodu ve ikinci terimin kodu ardarda eklenecek şekilde verilebilir. Yani ilk iki adım sonunda kod 1000 şeklindedir. Üçüncü adımda ele almamız gereken terimi, yine (3.9)'a bakarak öğrenebiliriz. Buna göre, (3.8) içerisinde aramamız gereken terim,  $ab'k_3$  yapısı olmalıdır. (3.8) içerisinde bu formattaki yapı,  $ab'$  olarak verilmiştir ve buradan,  $k_3$ 'ün 1 olduğunu anlamak mümkündür. O halde tekrar tabloya giderek, 1 için tanımlanan kod ilk iki adımda oluşturulan koda eklenir. İlk iki adımda elde edilen kod, 1000 ve üçüncü adıma ilişkin kod ise 1'e karşılık gelen kod 11'dir. O halde, üç adım sonrasında toplam kod 100011 olacaktır. Son adımda, yine (3.9) yapısına bakarsak, incelenmesi (3.8) içerisinde aranması gereken terimin  $abk_4$  yapısı olduğunu görebiliriz. Fonksiyon içerisinde (3.8) bu yapıya uygun terim  $abc'$  şeklinde yer almaktadır. O halde tekrar tabloya gidilerek (Şekil 3.17) c' ile ilişkilendirilmiş kod parçasının 01 olduğunu görürüz. Bu son kod parçasını, ilk üç basamakta elde ettiğimiz kod ile birleştirerek, toplam kodu 10001101 şeklinde elde ederiz. Fonksiyon üç değişkenli olması nedeniyle, yukarıdaki işlemler dört kez tekrarlanarak, N değişken sayısını belirtmek üzere,  $2^N$  uzunluklu bir kod oluşturulur. N bizim ele aldığımız örnek için 3 olduğuna göre, kodun uzunluğu 8 olacaktır. Bu kod, tamamen lojik fonksiyonu temsil eder ve farklı lojik fonksiyonlar için farklı kodlar elde edilir. Kodlama ile birlikte, F gibi çarpımlar toplamı şeklinde ifade edilen bir lojik fonksiyonu ve tümleyenini (F') tek bir dizi yapısı içerisinde tutmak mümkündür (F+F'). Tersten düşünerek, verilen bir kod ile birlikte, lojik fonksiyonun ve deęilinin kanonik

gösterimine ulaşılabilir.

| $a$ | $b$ | $F$   | $F$  | Code |
|-----|-----|-------|------|------|
| 0   | 0   | $k_1$ | 0    | 00   |
| 0   | 1   | $k_2$ | $c'$ | 01   |
| 1   | 0   | $k_3$ | $c$  | 10   |
| 1   | 1   | $k_4$ | 1    | 11   |

Şekil 3.17  $F=a'b'c+ab'+abc'$  lojik fonksiyonuna ait doğruluk tablosu ve kodlama stratejisi (Avcı ve Yıldırım, 2003)

- BDD gösterilimi ve indirgeme kuralları : Şekil 3.18'de, kodu elde edilmiş olan lojik fonksiyona (3.8) ilişkin, ikil karar diyagramı (BDD) verilmiştir.



Şekil 3.18 (3.8)'de verilen lojik fonksiyonuna ve fonksiyona ilişkin elde edilen 10001101 kodu ve ikil karar diyagramı yapısı (Avcı ve Yıldırım, 2003)

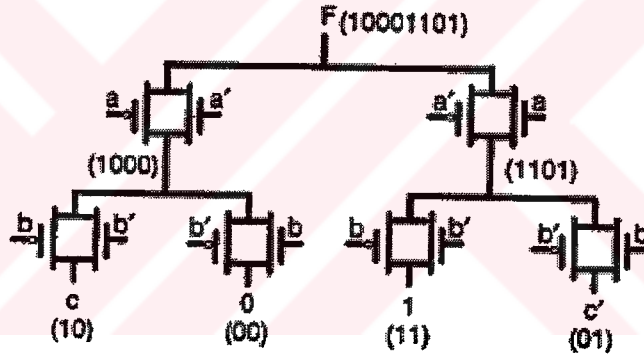
Bir önceki başlık içerisinde tanıtılan ve ayrıntıları ile verilen 123 karar diyagramı yapısından farklı olarak, bu çalışmada, kenarlar, değişkenler ve değişkenlerin tümleyenleri ile tanımlanmışlardır (Şekil 3.18). Diğer bir farklılık ise çocuk düğümlerin yerleşim stratejisidir. Bu çalışmada, bir düğüme ait çocuk düğümler (3.10) eşitliğinden faydalanılarak numaralandırılırlar.

$$(n+1) = 2*n + e \quad (3.10)$$

(3.10) içerisinde,  $e$ , kenarlar, değişkenin tümleyenini ise 0, değişkenin kendisi ise 1 değerlerini alacak şekilde,  $n$  ise çocuk düğümleri oluşturacak baba düğümün, düğüm

numarasını ifade etmektedir.  $(n+1)$  ile ifade edilen, bir alt seviyede bulunacak olan çocuk düğümlerin düğüm numaralarıdır. Yine bu gösterim şeklinde, terminal düğümlere yer verilmemiş sadece terminal olmayan düğümler, BDD gösterimi içerisinde yer almıştır. Aşağıdaki bölümde, indirgeme kuralları verilmiştir.

- **Kural 1 :** Eğer, iki düğüm aynı kodlara sahiplerse, bu iki düğüm birleştirilir.
- **Kural 2 :** Eğer, herhangi bir düğüme ilişkin kod, tüm sıfırlardan (tüm birlerden) oluşuyorsa, ilgili düğüm toprağa ( $V_{dd}$ 'ye) bağlanır.
- **Fiziksel gerçekleştirme :** Fiziksel gerçekleştirme aşamasında, karar diyagramının model olduğuna değinilmişti. Tüm aşamaları tamamlanan ve grafiksel olarak elde edilen BDD, son haliyle, BDD düğümleri, TG'lerden oluşan devre yapısında, transistörler arası düğümlere, BDD gösterimindeki kenarlar, yani düğümler arasındaki bağlantılar ise TG'leri temsil edecek şekilde devre oluşturulur. (3.8)'deki lojik fonksiyondan elde edilen BDD, Şekil 3.18'de verilmişti. Şekil 3.19'da ise, Şekil 3.18'de BDD yapısı verilen lojik fonksiyonun CPL-TG geçiş transistörü lojigi ile gerçekleştirilmesine ilişkin şekil verilmiştir.



Şekil 3.19  $F=a'b'c+ab'+abc$  lojik fonksiyonunun CPL-TG devre yapısı

Şekil 3.19'da verilen CPL-TG modelinde, TG'ler, çoğullayıcı yapısında yerleştirilmişlerdir. Bu konunun, bölüm 2'de de üzerinde durulmuştu. Gerek BDD, gerekse CPL-TG yapısı içerisinde, kontrol değişkenlerinin dağılımının belli bir stratejisinin olduğu söylenmelidir. Öyleki, BDD'nin oluşturulmasında, Shannon açılımının saklı olduğunu görmekteyiz. Geleneksel gösterimde, kesikli çizgiler ile, 123 karar diyagramında 0 ile temsil edilen kenar etiketlerinin, bu yeni yöntemde, kontrol değişkeninin tümleyeni ile, yine, geleneksel gösterimde düz çizgi ile, 123 karar diyagramında 1 ile gösterilen kenar etiketlerinin, kontrol değişkeni ile etiketlendirildiklerini görmekteyiz.

Şekil 3.19'dan da görülebileceği üzere, TG'ler ( $p_{mos}||n_{mos}$ ) şeklinde, aynı yapıda yanyana dizilecek şekilde organize edilmişler, ancak gate uçlarına bağlanan kaynak bağlantıları,

BDD'yi ve sonuçta lojik fonksiyonu gerçeklemek için birbirinin tersi olacak şekilde organize edilmişlerdir. Bu sayede BDD'nin kenar komşulukları, kontrol değişkeni (kontrol değişkeninin tümleyeni) sağlanmış olmaktadır. Örnek vermek gerekirse, Şekil 3.18'de, a' değişkenine karşılık gelen yapı, Şekil 3.19'da sol ve üstte yer alan, PMOS gate ucuna a kontrol değişkeninin, NMOS gate ucuna a' kontrol değişkenin bağlandığı TG'dir. Bu yapıyı düşündüğümüzde, niçin Şekil 3.18'deki 1. ve 2. düğümleri birbirlerine bağlayan a' kenarına karşılık geldiğini anlayabiliriz. Diğer düğümler arası bağlantılar (kenarlar) için de durum aynı şekilde tezahür eder.

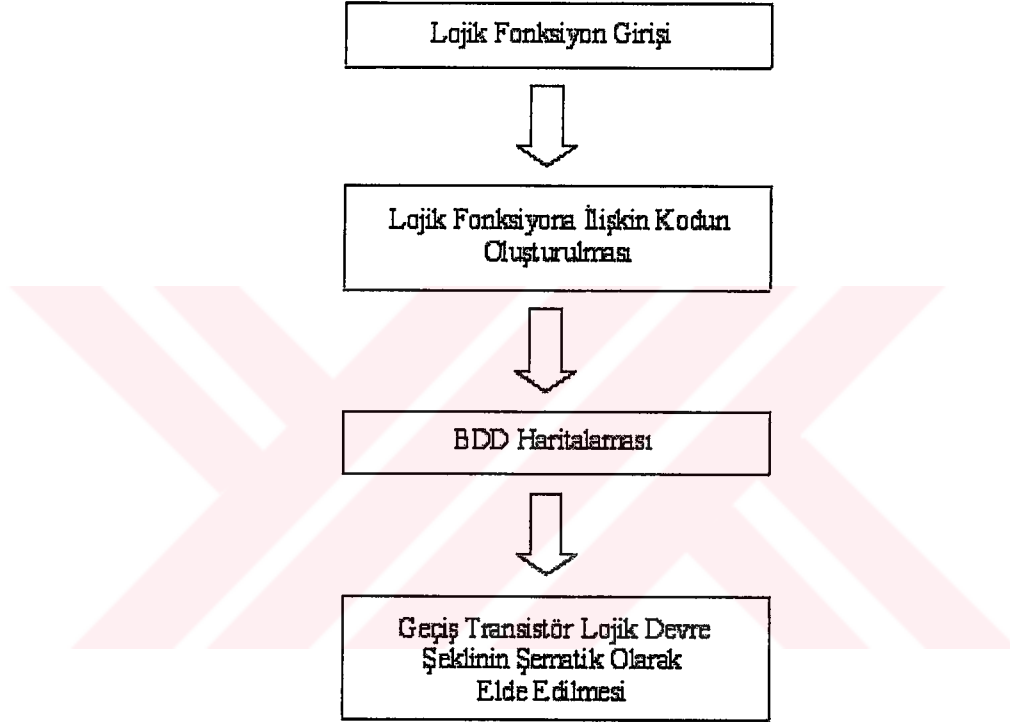


#### 4. PROGRAMIN GELİŞTİRİLMESİ VE GENEL YAPISI

Program, C++ Builder 6 görsel geliştirme ortamında, görsel bir program olarak tasarlanmıştır. Geliştirilen program, üç ana bölümden ibarettir. Bunlar;

- Çarpımlar toplamı şeklinde girilen bir lojik fonksiyona ilişkin kodun elde edilmesi,
- BDD çizdirilmesi,
- Geçiş transistör devresinin çizdirilmesidir.

Geliştirilen programa ait genel akış diyagramı, Şekil 4.1’de verilmiştir.



Şekil 4.1 Program Genel Akış Diyagramı

Program, programlama stratejileri içerisinde, seri çalışan bir program olarak düşünülebilir. Seri programlama, birçok açıdan pipeline mekanizmasına benzer. Bir blok işlemini tamamlar ve diğer blok bir önceki bloktan kendisine sunulan işlenmiş datayı giriş olarak kabul eder. Bu durum, en son çıkışın üretilmesi aşamasına kadar bu şekilde devam eder. Programın, oldukça gelişmiş olan hazır görsel araçları, geliştirme esnasında yoğun bir şekilde kullanılmıştır.

İlk bölüm olan, lojik fonksiyonun, ikili koda dönüştürülmesi aşamasında, çarpım toplamaları şeklinde girilen lojik fonksiyona ilişkin geçiş ve kontrol değişkenlerinin tesbiti yer almakta ve kaç adet kontrol değişkeninin var olduğunun tespiti yer almaktadır. Bir sonraki aşamada, kontrol değişkeni sayısına bağlı kalarak sıralı bir biçimde,  $2^{\text{kontrol değişkeni sayısı}}$  adet ikil giriş

fonksiyona uygulanmakta ve fonksiyonun herbir girişe karşılık gelen cevabı üretilmektedir. Bu cevaplar, dinamik olarak oluşturulan kod tablosundan faydalanılarak tek tek ikili ikil kodlara dönüştürülmekte ve bu ikili ikil kodlar, ardarda eklenerek toplam kodu oluşturmaktadır. Girilen lojik fonksiyon için, ingilizce karakterler olmak üzere, ilk 14 adet karakter kullanılacak şekilde düşünülmüştür. Yani çarpım toplamaları şeklinde girilecek olan lojik fonksiyonun, değerlendirebileceği değişkenler  $a, b, \dots, m, n$  aralığında tanımlıdır. Bu tanımlama ile, maksimum 13 seviye BDD ve geçiş transistör lojik seviyesi tanımlanmıştır.

İkinci bölümde, BDD 'nin çizdirilmesi tamamlanmıştır. Genel BDD yapısı model alınarak tasarlanan çizim platformu, değişken sayısının fazla olduğu örneklerde, şekli dinamik olarak çizecek şekilde tasarlanmıştır. Bu durumda, düğüm sayısının fazla olması, bir monitör alanına sığmaması sebebiyle, şekiller arası geçişler çeşitli sembollerin şekil üzerinde fare tıklanması olayına aktive edilmesi ile sağlanmıştır. İndirgeme kuralları, bir yapı dizisi içerisinde tutulan düğümlerin, kodları tek tek karşılaştırılarak uygulanmıştır. Düğümlere ait kayıtlar tutulmuş ve sonuçta hangi düğümlerin şekil içerisinde yerlerini alacaklarına, hangi düğümlerin ise yok edileceklerine karar verilmiştir. Yapı dizisi ile birlikte, düğüm isimli bir değişken tanımlanmış, düğümün numarası, düğümün tuttuğu kod, düğümün diğer düğümlerle olan ilişkileri gibi farklı türdeki dokuz adet değişkenin, tek bir değişken olarak tutulması sağlanmıştır ve uygun yerlerde bu bilgilerin geri çağırılarak düğümün durumu ve nasıl davranacağı hakkında değerlendirmeler yapmak mümkün olmuştur. Aşağıda .h olarak program içerisine uygulanan node olarak isimlendirilmiş yapı görülmektedir.

```
#ifndef NODE_H

#define NODE_H

struct NODE

{

    int ID;

    int status;

    int statusjoint;

    int connection;

    AnsiString sid;
```

```

AnsiString var;

AnsiString varptl1;

AnsiString varptl2;

float x1,y1,x2,y2,x3,y3,x4,y4,x5,y5,x6,y6;

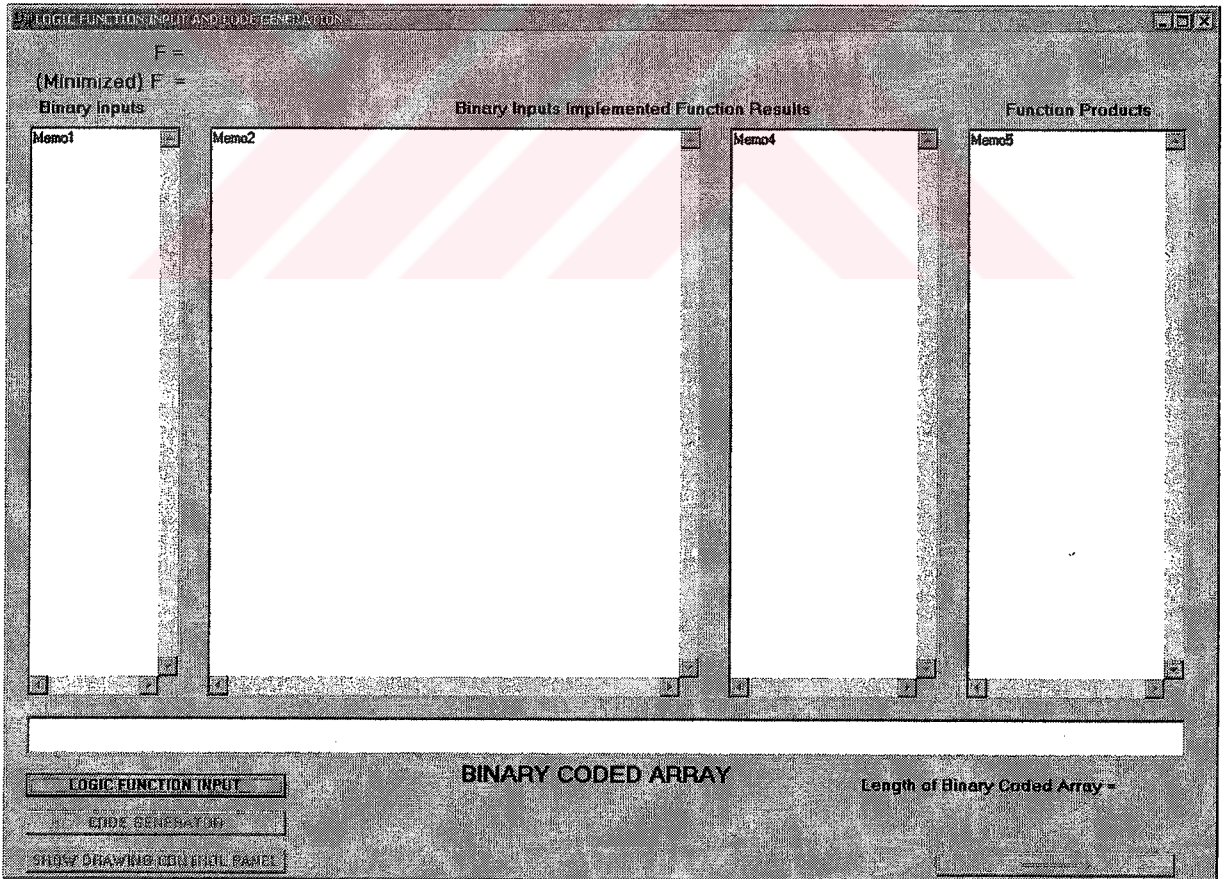
};

#endif

```

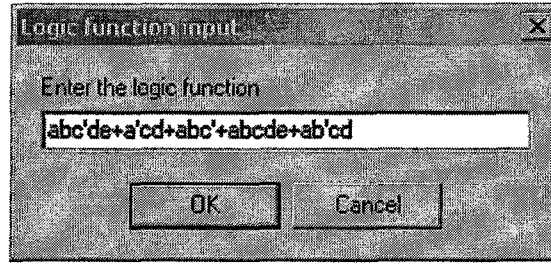
Üçüncü ve son bölümde, geçiş transistörlerinden oluşan devre şeklini veren şematik gösterimin çizimi tamamlanmıştır. Bu bölümde, ikinci bölümde olduğu gibi yapı dizisinden, dolayısıyla düğüm değişkenleri ve tuttukları verilerin birbirleri ile ilişkisel durumlarından yararlanılmıştır.

Aşağıda, lojik bir fonksiyonun, programa girilmesi aşamasından, son aşamaya kadar birtakım ekran görüntülerine yer verilecektir.



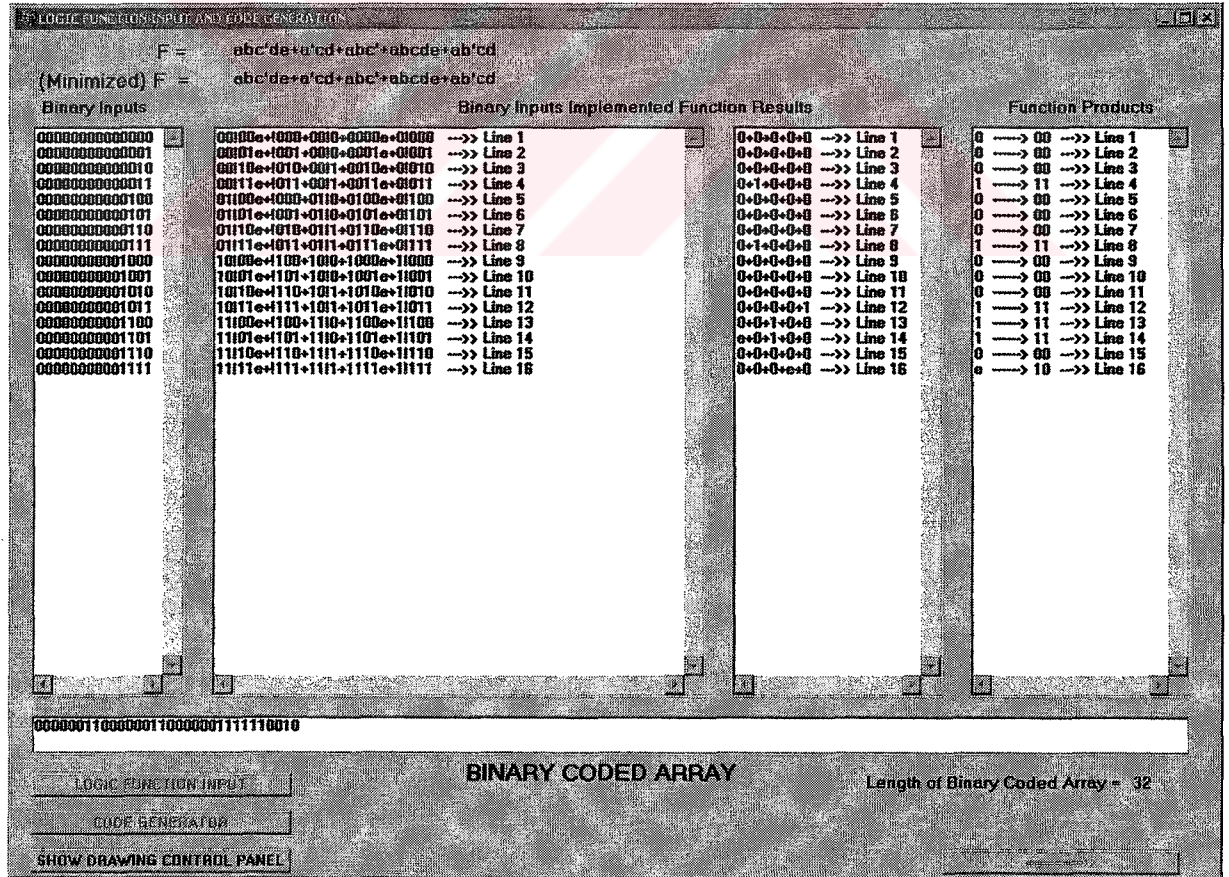
Şekil 4.2 Fonksiyon girişi ve kod oluşturma paneli

Şekil 4.2’de gösterilen panel, programın çalıştırılması ile karşılaşılan ilk ekrandır. Lojik fonksiyon giriş düğmesine tıklanması ile Şekil 4.3’de gösterilen ekran görüntüsüne ulaşılır.



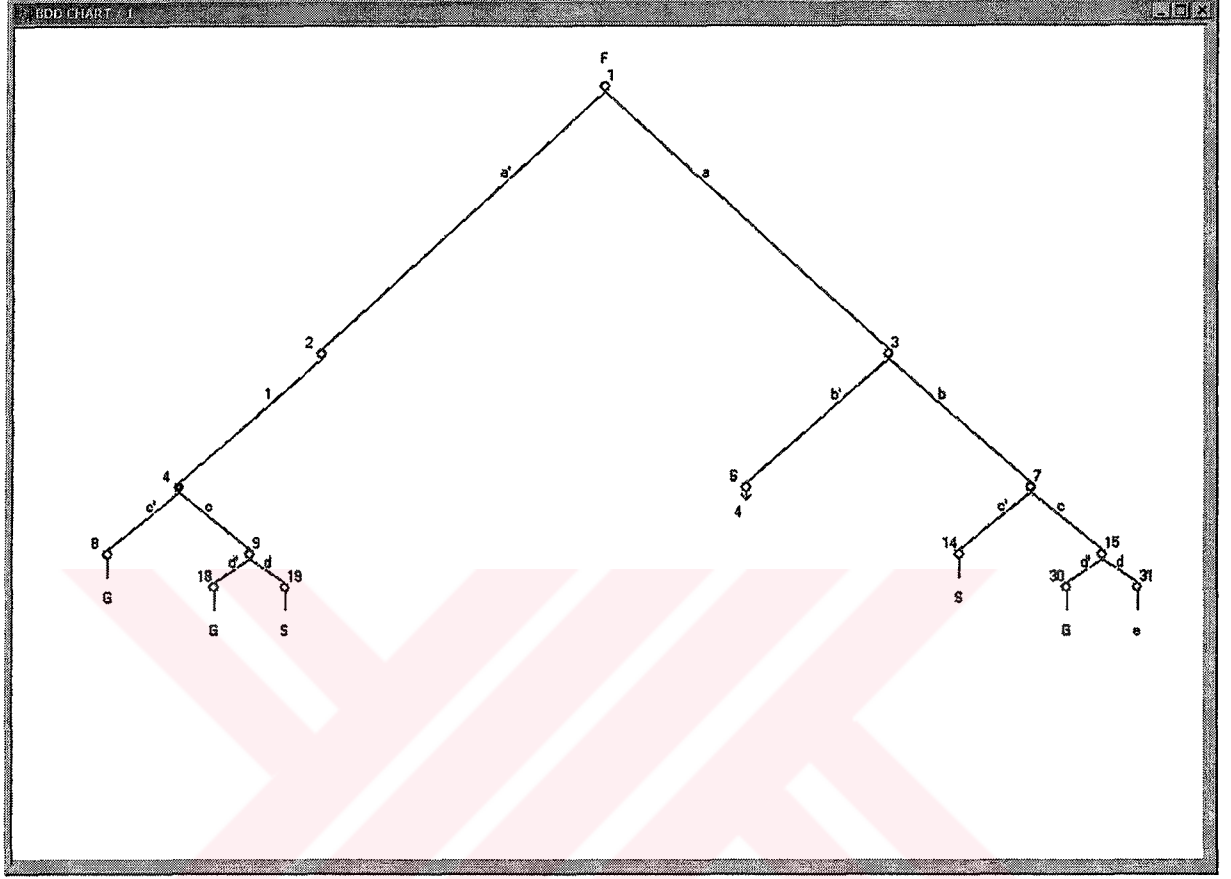
Şekil 4.3 Lojik Fonksiyon Giriş Paneli

Şekil 4.3’de gösterilen panel içerisine, BDD şeklinin ve geçiş transistör devre modelinin oluşturulması istenilen lojik fonksiyon girilmektedir. Bu örnek için girilen lojik fonksiyon, Şekil 4.3’den de görülebileceği üzere,  $abc'de+a'cd+abc'+abcde+ab'cd$  şeklindedir. Şekil 4.4’de, girilen fonksiyona ilişkin kodun oluşturulması aşamasını gösteren ekran görüntüsüne yer verilmiştir.



Şekil 4.4 Lojik fonksiyona ait kodun oluşturulmasını gösteren ekran çıktısı

Kodun elde edilmesinden sonraki aşama, kod bilgisinden faydalanılarak BDD haritalamasının oluşturulmasıdır. Şekil 4.5’de, BDD haritalamasını gösteren ekran görüntüsüne yer verilmiştir.

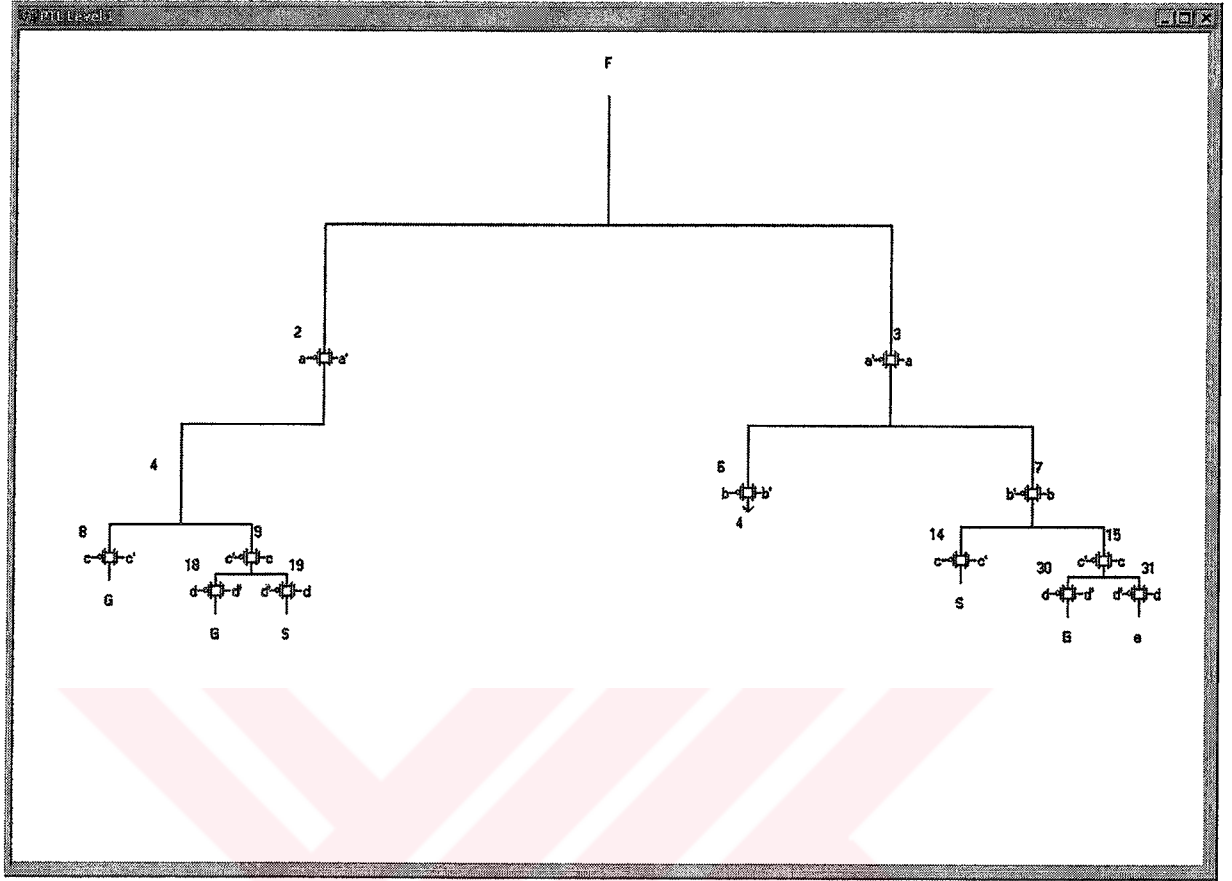


Şekil 4.5 BDD haritalamasını gösteren ekran çıktısı

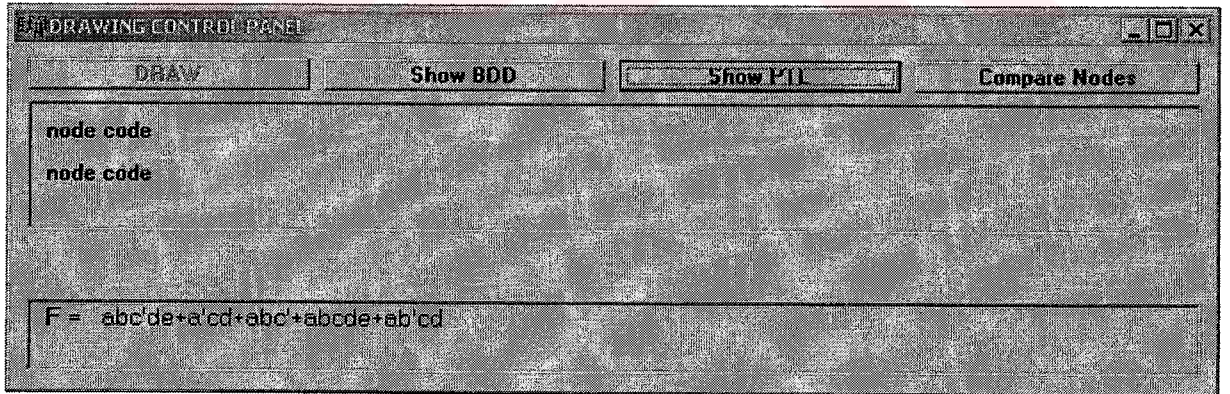
Son aşama ise, ilgili fonksiyona ait geçiş transistör lojik devre modelinin elde edilmesi ve gösterilmesidir. Şekil 4.6’da ilgili fonksiyona ait geçiş transistör devre modeli gösterilmiştir. Bir diğer panel de, programın çalışma esnasında, ekran görüntüleri arasında geçişi sağlamak amacıyla yerleştirilmiştir (Şekil 4.7).

Son olarak, şekillerde verilen ekran çıktıları üzerinde yer alan düğme kontrollerinin üstlendikleri görevlerden kısaca bahsedilecektir. Şekil 4.2’de yer alan ilk düğme kontrolü olan “logic function input”, kullanıcının modellenmesini istediği lojik fonksiyonu girebilmesini mümkün kılar. Şekil 4.2’de yer alan “code generator” isimli diğer kontrol düğmesi, girilen lojik fonksiyona karşılık gelecek olan kodun üretilmesi görevini üstlenmiştir. Şekil 4.2’de mevcut bulunan diğer bir kontrol ise “show drawing control panel” isimli düğmedir. Bu düğme kontrolü, Şekil 4.7’de yer alan, çizim düğme kontrollerinin bulunduğu panele ulaşımı

sağlar.



Şekil 4.6 Geçiş transistör devre modelini gösteren ekran çıktısı



Şekil 4.7 Çizim kumanda paneli

Şekil 4.7 içerisinde yer alan ilk düğme kontrolü olan “draw”, çizim işlemini başlatır. Bu düğme kontrolüne basılması, aynı panel içerisinde bulunan diğer üç düğme kontrolünü aktive eder. Şekil 4.7 içerisinde yer alan diğer bir düğme kontrolü olan “show bdd” ise, girilen fonksiyona karşılık gelen BDD’yi kullanıcının karşısına çıkarır. Benzer şekilde, “show ptl”

düğme kontrolü de, girilen fonksiyona karşılık gelen geçiş transistör lojik devre modelini kullanıcının karşısına çıkarır. Son kontrol olan “compare nodes” ise, kontrol ve denetleme amaçlı konulmuştur. Kullanıcı, şekil üzerinde yer alan veya yer almayan düğümlere ilişkin kod parçalarını incelemek isterse, bu düğme kontrolünü kullanabilecektir.



## 5. UYGULAMALAR

Bu bölümde, bazı lojik kapılara ilişkin lojik fonksiyonlara, keyfi bir lojik fonksiyona, Avcı ve Yıldırım (2003)'ün çalışmalarında yer verdiği lojik fonksiyona ve iki bit iki sayının toplamına ait lojik fonksiyonlara karşılık gelen devre yapılarının, geliştirilen program ile elde edilmelerine yer verilecektir. Ayrıca, devrelere ilişkin simülasyon sonuçları, devre yapıları ile birlikte sunulacaktır.

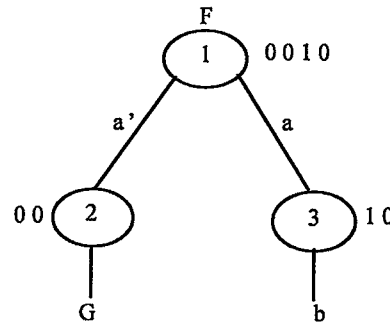
### 5.1 Lojik Ve Kapısının Gerçeklenmesi

Program girişine,  $F=ab$  uygulanmış ve programın ürettiği BDD ve geçiş transistör lojik devresi elde edilmiştir. Şekil 5.1'de verilen doğruluk ve kod tablosu vasıtası ile,  $F= ab$  fonksiyonuna ilişkin kodun, 0010 şeklinde olacağını görebiliriz. İki değişkenli yapı için genel ifade, (5.1)'de verilmiştir.  $F=ab$  lojik fonksiyonuna ait kod, Şekil 5.1'deki tablo ve (5.1) kullanılarak, 0010 şeklinde elde edilir.

$$F=a'k_1+ak_2 \quad (5.1)$$

| a | F     | F    | code |
|---|-------|------|------|
| 0 | $k_1$ | 0    | 00   |
| 1 | $k_2$ | $b'$ | 01   |
| 0 | $k_1$ | b    | 10   |
| 1 | $k_2$ | 1    | 11   |

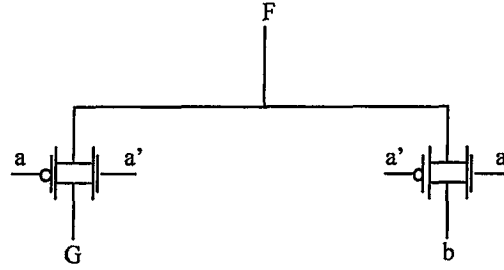
Şekil 5.1  $F=ab$  (ve) fonksiyonuna ilişkin kod tablosu



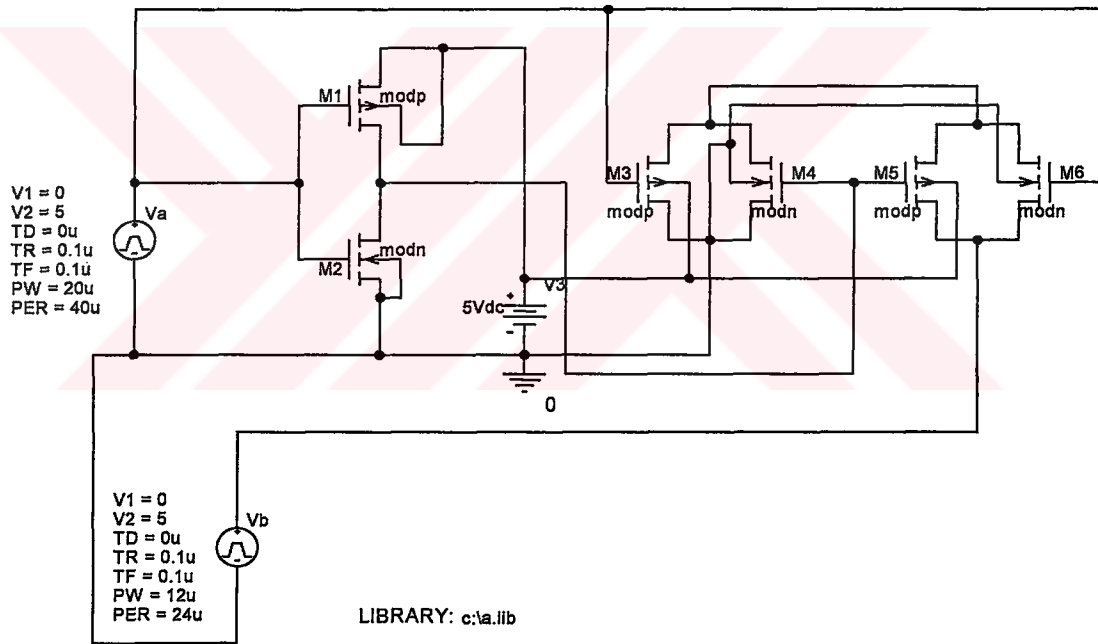
Şekil 5.2  $F=ab$  (ve) fonksiyonuna ilişkin BDD gösterimi

Şekil 5.2'de BDD yapısını görebileceğimiz,  $F=ab$  fonksiyonuna ilişkin geçiş transistörü

modeli Şekil 5.3'de verilmiştir. Şekil 5.3'de de görüldüğü gibi, devrenin,  $V_{dd}$  dışında,  $a$ ,  $a'$  ve  $b$  gibi üç adet kaynağa ihtiyacı vardır. Bu kaynaklardan,  $a$  ve  $b$  bağımsız kaynaklar olarak düşünülürken,  $a'$ 'nin tümleyeni ise, bir eviriciye  $a$ 'nın uygulanmasıyla elde edilir. Devrenin simülasyonu, PSPICE programı vasıtasıyla gerçekleştirilmiştir. Şekil 5.4'de, devrenin tüm yapısını görmek mümkündür.

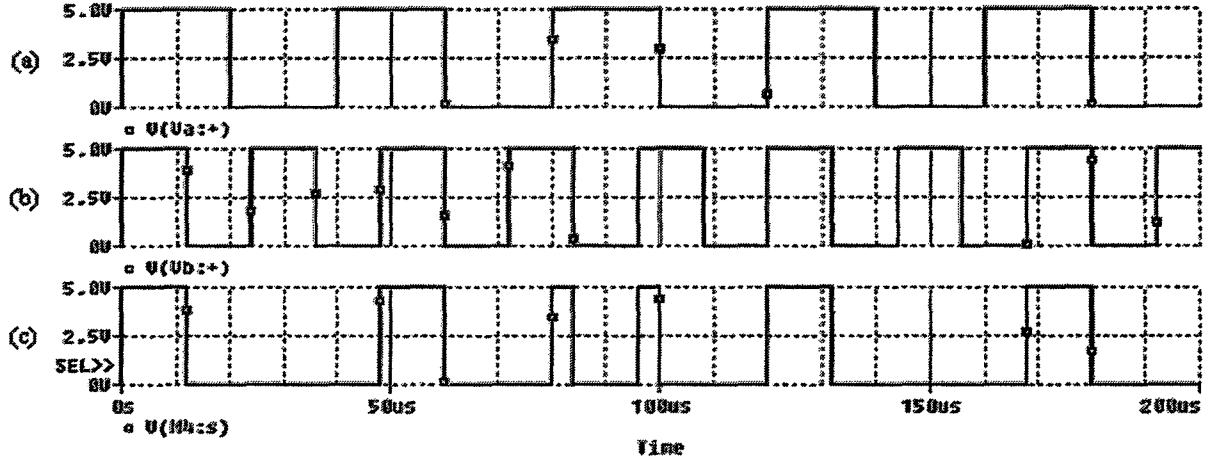


Şekil 5.3  $F=ab$  (ve) fonksiyonuna ilişkin geçiş transistör lojiği devre yapısı



Şekil 5.4  $F=ab$  (ve) fonksiyonuna ilişkin, kaynakları da içeren devre yapısı

Şekil 5.5'de, Şekil 5.3 kullanılarak elde edilen Şekil 5.4'deki devreye ilişkin simülasyon sonuçlarına yer verilmiştir. Sırasıyla, giriş 1(a), giriş 2(b) ve çıkış(c) alt alta gelecek şekilde verilmiştir.



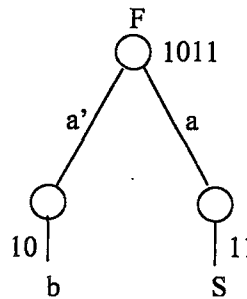
Şekil 5.5 Şekil 5.4'deki devreye ilişkin simülasyon sonucu, (a) giriş 1, (b) giriş 2, (c) çıkış

## 5.2 Lojik Veya Kapısının Gerçeklenmesi

Program girişine,  $F=a'b+ab'+ab$  uygulanmış ve programın ürettiği BDD ve geçiş transistör lojik devresi elde edilmiştir. Şekil 5.1'de verilen doğruluk ve kod tablosu vasıtası ile,  $F=ab$  fonksiyonuna ilişkin kodun, 0010 şeklinde olacağını görebiliriz. Ve kapısında kodun elde edilmesi ile aynı şekilde,  $F=a'b+ab'+ab$  fonksiyonuna ait kod, 1011 şeklinde elde edilir.

| a | F     | F    | code |
|---|-------|------|------|
| 0 | $k_1$ | 0    | 00   |
| 1 | $k_2$ | $b'$ | 01   |
| 0 | $k_1$ | b    | 10   |
| 1 | $k_2$ | 1    | 11   |

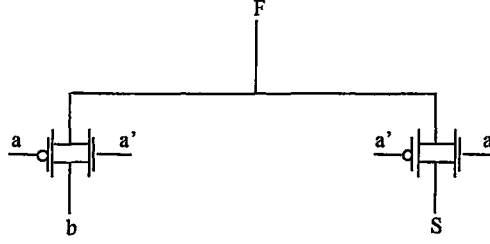
Şekil 5.6  $F=a'b+ab'+ab$  (veya) fonksiyonuna ilişkin kod tablosu



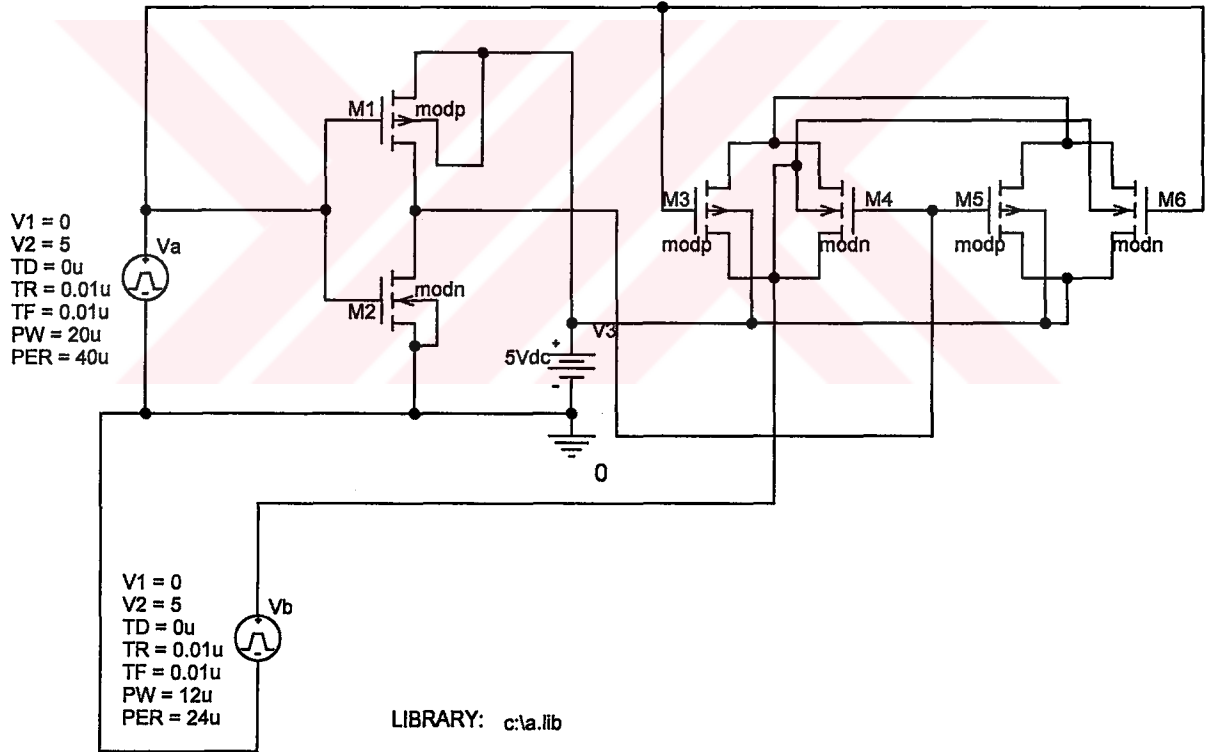
Şekil 5.7  $F=a'b+ab'+ab$  (veya) fonksiyonuna ilişkin BDD gösterimi

Şekil 5.7'de BDD yapısını görebileceğimiz,  $F=a'b+ab'+ab$  fonksiyonuna ilişkin geçiş

transistörü modeli Şekil 5.8’de verilmiştir. Şekil 5.8’de de görüldüğü gibi, devrenin, Vdd dışında, a, a’ ve b gibi üç adet kaynağa ihtiyacı vardır. Bu kaynaklardan, a ve b bağımsız kaynaklar olarak düşünülürken, a’nın tümleyeni ise, bir eviriciye a’nın uygulanmasıyla elde edilir. Devrenin simülasyonu, PSPICE programı vasıtasıyla gerçekleştirilmiştir. Şekil 5.9’de, devrenin tüm yapısını görmek mümkündür.

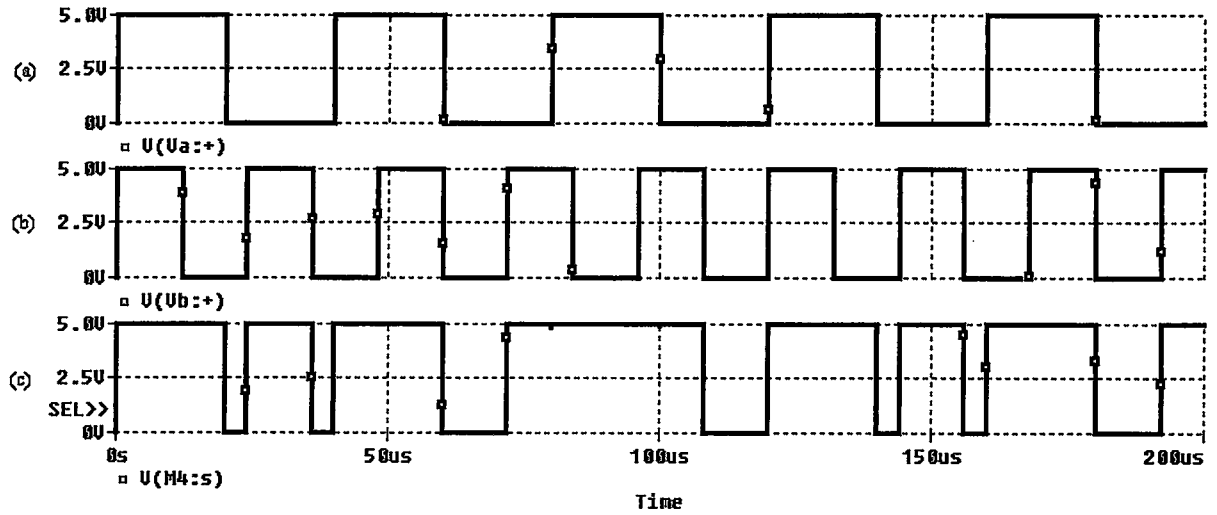


Şekil 5.8  $F = a'b + ab' + ab$  (veya) fonksiyonuna ilişkin geçiş transistör lojigi devre yapısı



Şekil 5.9  $F = a'b + ab' + ab$  (veya) fonksiyonuna ilişkin, kaynakları da içeren devre yapısı

Şekil 5.10’da, Şekil 5.8 kullanılarak elde edilen Şekil 5.9’daki devreye ilişkin simülasyon sonuçlarına yer verilmiştir. Sırasıyla, giriş 1(a), giriş 2(b) ve çıkış(c) alt alta gelecek şekilde verilmiştir.



Şekil 5.10 Şekil 5.9'daki devreye ilişkin simülasyon sonucu, (a) giriş 1, (b) giriş 2, (c) çıkış

Yukarıda, örnekleri verilen, ve (and) ve veya (or) lojik kapılarının, geçiş transistörleri ile gerçekleştirilmesine benzer şekilde, diğer lojik kapılar olan nand, nor, xor ve nxor lojik kapılarının geçiş transistör lojik devre yapıları elde edilebilir.

### 5.3 Keyfi Bir Fonksiyon Gerçeklenmesi

Bu başlık altında, (5.2)'de verilen, keyfi bir lojik fonksiyona ilişkin, tasarım aşamaları yer alacaktır. Sırasıyla, fonksiyona ait kodun oluşturulması, BDD haritalamasının elde edilmesi, geçiş transistör lojik devre modelinin elde edilmesi, kaynakları içeren devre yapısının oluşturulması ve ardından simülasyonunun yapılması yer alacaktır. İlk olarak, kodu elde edelim. Bunun için, n değişkenli lojik fonksiyon için, n-1 değişken ve k sabitlerinden oluşan yapıyı (5.3) ve kod tablosunu (Çizelge 5.1) beraber kullanmamız ve gerçekleştirilmesini istediğimiz lojik fonksiyonu, kanonik formda düşünmemiz gerekir.

$$F=abc'de+a'cd+abc'+abcde+ab'cd \quad (5.2)$$

$$F=a'b'c'd'k_1+a'b'c'dk_2+a'b'cd'k_3+a'b'cdk_4+a'bc'd'k_5+a'bc'dk_6+a'bcd'k_7+a'bcdk_8+ab'c'd'k_9+ab'c'dk_{10}+ab'cd'k_{11}+ab'cdk_{12}+abc'd'k_{13}+abc'dk_{14}+abcd'k_{15}+abcdk_{16} \quad (5.3)$$

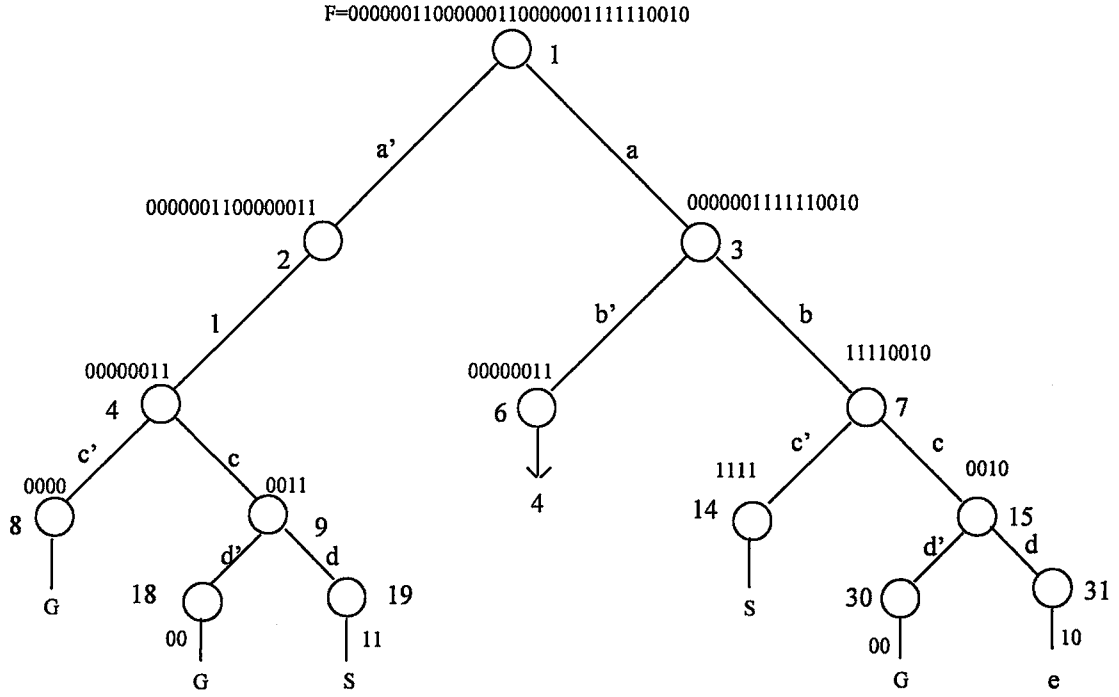
(5.2) içerisindeki, her bir terimi, (5.3) içerisinde aramalı ve bulduğumuz taktirde, ilgili k değerine karşılık gelen değeri, (verilen fonksiyon için 0, 1, e veya e' olabilir) bulmalıyız. Bulunan bu değere karşılık gelen kod parçasını, kod tablosundan elde ederek ve bu işlemi tüm mümkün kombinasyonlar için tekrar etmeliyiz. Biraz daha somutlaştırmak gerekirse, (5.2) içerisinde ilk aramamız gereken yapı, a'b'c'd'k<sub>1</sub> yapısıdır. Böyle bir minterime, (5.2) içinde

rastlamıyoruz. O halde, aradığımız minterime ait  $k_1$  değerinin 0 olduğunu söyleriz. Çizelge 5.1 de verilen kod tablosundan faydalanarak, bu örnek için 32 uzunluklu olacak olan kodun, ilk iki haneden oluşan bölümünü ekde ederiz. Benzer şekilde devam eder ve kalan 15 adet  $k_n$  değerlerini elde eder ve 16 adet  $k$  değerine karşılık gelen 2 uzunluklu kod dizilerini arka arkaya eklersek, tüm kodu elde etmiş oluruz.

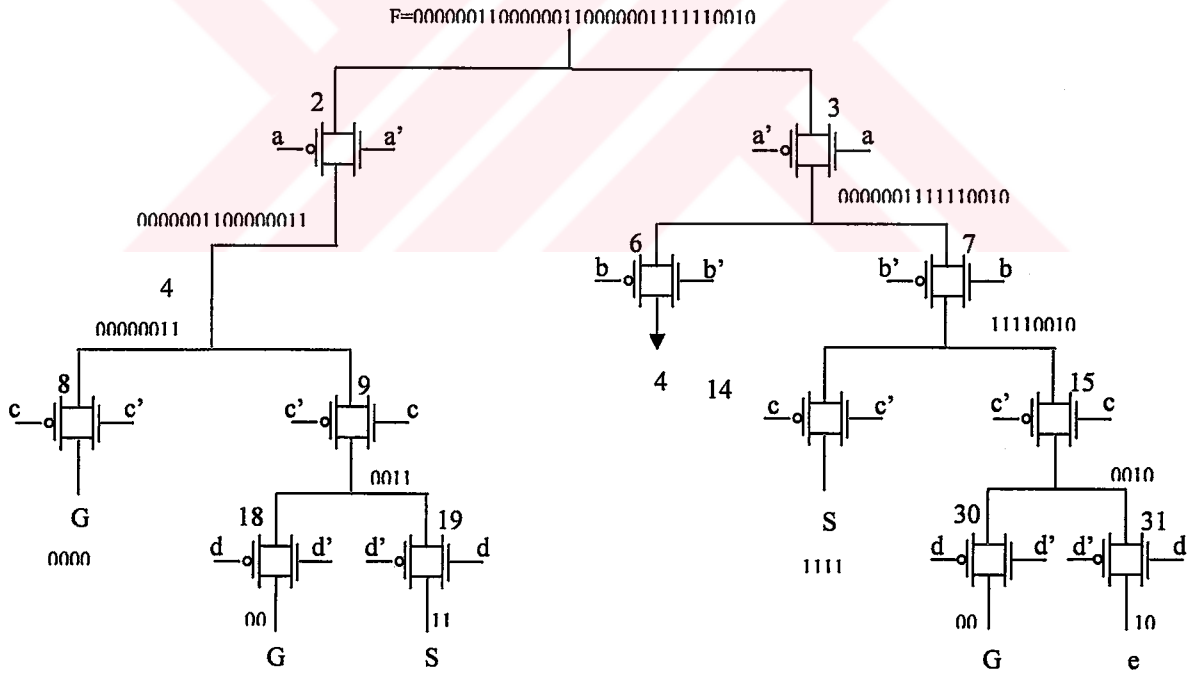
Çizelge 5.1 Örneği verilen lojik fonksiyona ilişkin kod tablosu

| a | b | c | d | F        | F    | code |
|---|---|---|---|----------|------|------|
| 0 | 0 | 0 | 0 | $k_1$    | 0    | 00   |
| 0 | 0 | 0 | 1 | $k_2$    | $e'$ | 01   |
| 0 | 0 | 1 | 0 | $k_3$    | e    | 10   |
| 0 | 0 | 1 | 1 | $k_4$    | 1    | 11   |
| 0 | 1 | 0 | 0 | $k_5$    | 0    | 00   |
| 0 | 1 | 0 | 1 | $k_6$    | $e'$ | 01   |
| 0 | 1 | 1 | 0 | $k_7$    | e    | 10   |
| 0 | 1 | 1 | 1 | $k_8$    | 1    | 11   |
| 1 | 0 | 0 | 0 | $k_9$    | 0    | 00   |
| 1 | 0 | 0 | 1 | $k_{10}$ | $e'$ | 01   |
| 1 | 0 | 1 | 0 | $k_{11}$ | e    | 10   |
| 1 | 0 | 1 | 1 | $k_{12}$ | 1    | 11   |
| 1 | 1 | 0 | 0 | $k_{13}$ | 0    | 00   |
| 1 | 1 | 0 | 1 | $k_{14}$ | $e'$ | 01   |
| 1 | 1 | 1 | 0 | $k_{15}$ | e    | 10   |
| 1 | 1 | 1 | 1 | $k_{16}$ | 1    | 11   |

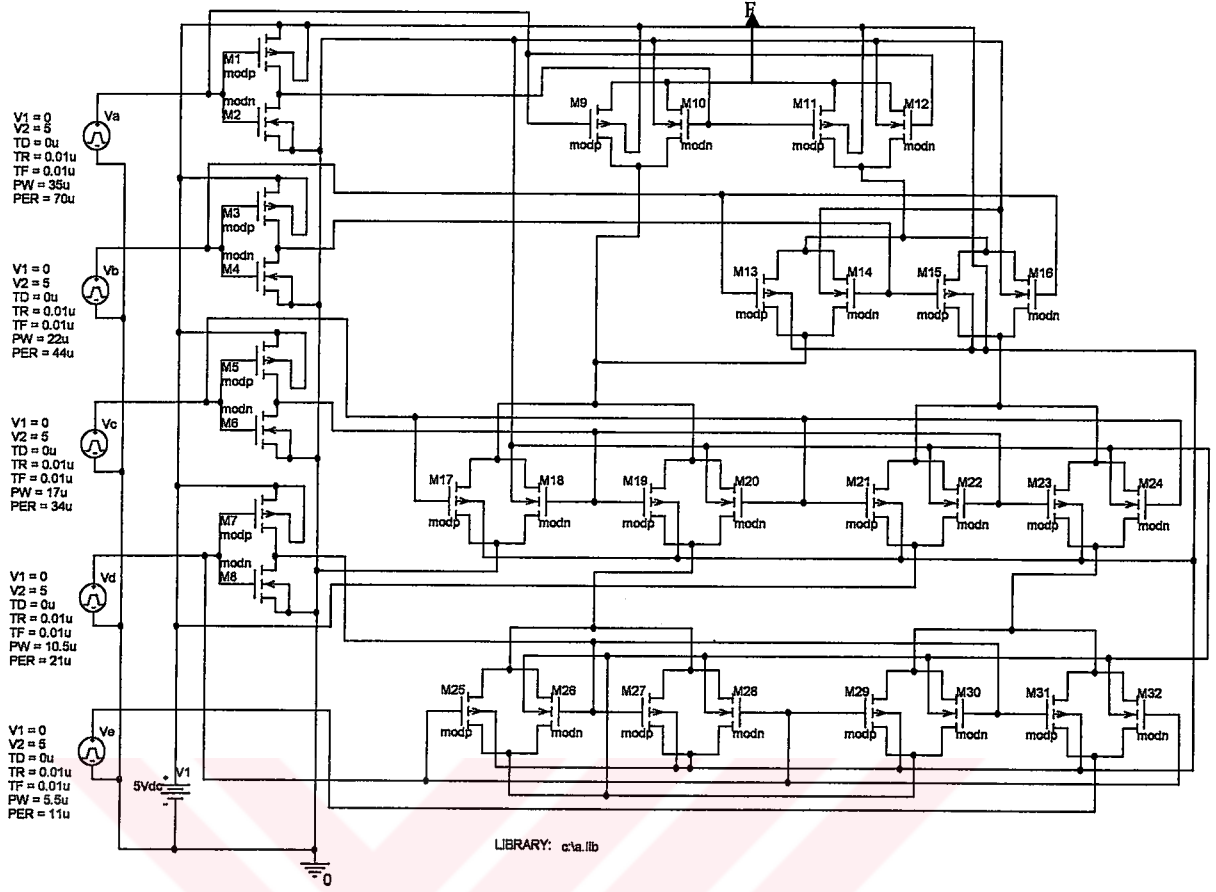
Kodu elde edilen lojik fonksiyona ait BDD gösterimi, Şekil 5.11'de verilmiştir. Şekil 5.11'de, bir numaralı düğüm üzerinde, ilgili lojik fonksiyona ilişkin elde edilen kod görülebilir. BDD gösterimi verilen bu lojik fonksiyona karşılık gelen geçiş transistör lojik devre modeli Şekil 5.12'de verilmiştir. Elde edilen bu modelden faydalanılarak, devre simülasyon platformuna taşınmış (Şekil 5.13) ve simülasyon sonuçları elde edilmiştir (Şekil 5.14).



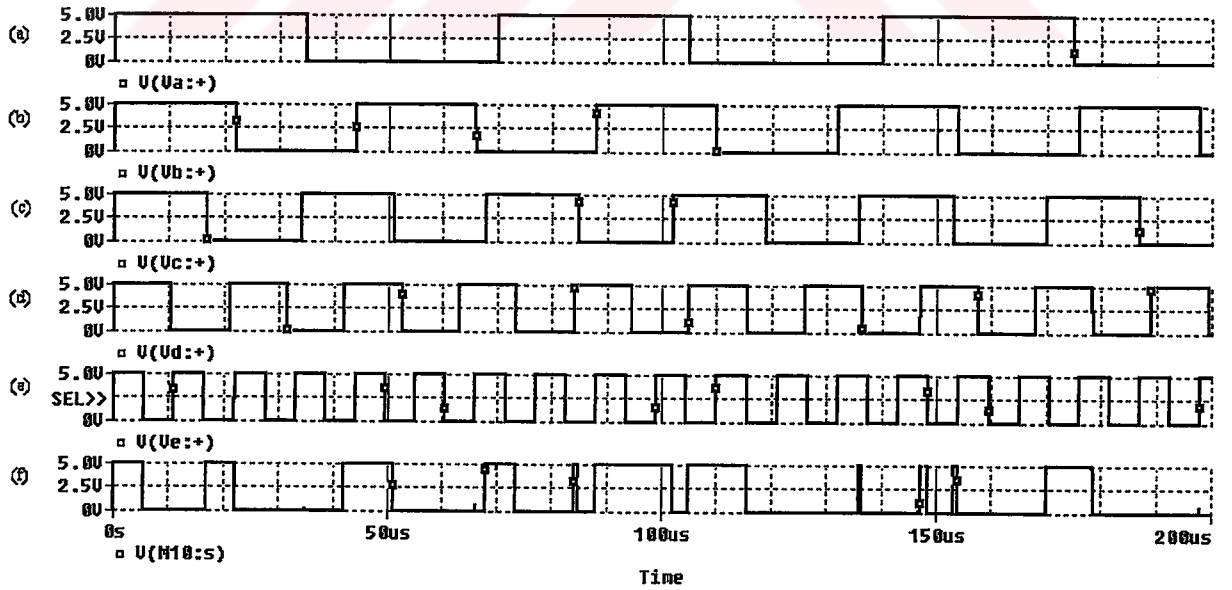
Şekil 5.11 (5.2)'de verilen lojik fonksiyona ait BDD gösterimi



Şekil 5.12 (5.2)'de verilen lojik fonksiyona ait geçiş transistör lojik devre modeli



Şekil 5.13 (5.2)'de verilen fonksiyona ilişkin geçiş transistör lojik devresi



Şekil 5.14 Şekil 5.13'deki devreye ait simülasyon sonuçları, (a), (b), (c), (d) ve (e) sırasıyla girişleri ve (f) ise çıkışı göstermektedir.

Bu bölüm içerisinde, keyfi bir fonksiyonun fiziksel gerçekleşmesi adım adım anlatılmıştır.

Devam eden bölümde, iki bit toplayıcı tasarımı sunularak, çalışmaya son nokta konulacaktır. Bu bölümdekine benzer biçimde, toplayıcı tasarımında da, kodlama, karar diyagramı oluşturulması, geçiş transistör lojik modelinin elde edilmesi, devrenin simülasyon ortamına taşınması ve simülasyon sonuçlarının elde edilmesi adımları gerçekleştirilecektir.

#### 5.4 İki Bit Toplayıcı

Aşağıda, iki bit, iki sayıyı toplayan devreye ilişkin, doğruluk tablosu (Çizelge 5.2), kod tablosu (Çizelge 5.3), BDD haritalaması (Şekil 5.15, Şekil 5.16, Şekil 5.17), geçiş transistör lojik devre modeli (Şekil 5.18, Şekil 5.19, Şekil 5.20), kaynakları içeren devre yapısı (Şekil 5.21) ve simülasyon sonuçları beraber sunulacaktır (Şekil 5.22).

Çizelge 5.2 İki bit toplayıcıya ilişkin doğruluk tablosu

| a | b | c | d | Y <sub>0</sub> | Y <sub>1</sub> | Y <sub>2</sub> | Ondalık |
|---|---|---|---|----------------|----------------|----------------|---------|
| 0 | 0 | 0 | 0 | 0              | 0              | 0              | 0       |
| 0 | 0 | 0 | 1 | 0              | 0              | 1              | 1       |
| 0 | 0 | 1 | 0 | 0              | 1              | 0              | 2       |
| 0 | 0 | 1 | 1 | 0              | 1              | 1              | 3       |
| 0 | 1 | 0 | 0 | 0              | 0              | 1              | 1       |
| 0 | 1 | 0 | 1 | 0              | 1              | 0              | 2       |
| 0 | 1 | 1 | 0 | 0              | 1              | 1              | 3       |
| 0 | 1 | 1 | 1 | 1              | 0              | 0              | 4       |
| 1 | 0 | 0 | 0 | 0              | 1              | 0              | 2       |
| 1 | 0 | 0 | 1 | 0              | 1              | 1              | 3       |
| 1 | 0 | 1 | 0 | 1              | 0              | 0              | 4       |
| 1 | 0 | 1 | 1 | 1              | 0              | 1              | 5       |
| 1 | 1 | 0 | 0 | 0              | 1              | 1              | 3       |
| 1 | 1 | 0 | 1 | 1              | 0              | 0              | 4       |
| 1 | 1 | 1 | 0 | 1              | 0              | 1              | 5       |
| 1 | 1 | 1 | 1 | 1              | 1              | 0              | 6       |

Yukarıdaki tablodan da anlaşılabileceği üzere, ab, ilk iki bitlik sayıyı, cd ise ikinci iki bitlik sayıyı temsil etmektedirler. Verilen doğruluk tablosuna göre, çıkış üç lojik fonksiyonla ifade edilmektedir. Aşağıda bu lojik fonksiyonlar verilmiştir (5.3), (5.4) ve (5.5).

$$Y_0 = a'bcd + ab'cd' + ab'cd + abc'd + abcd' + abcd \quad (5.4)$$

$$Y_1 = a'b'cd' + a'b'cd + a'bc'd + a'bcd' + ab'c'd' + ab'c'd + abc'd' + abcd \quad (5.5)$$

$$Y_2 = a'b'c'd + a'b'cd + a'bc'd' + a'bcd' + ab'c'd + ab'cd + abc'd' + abcd' \quad (5.6)$$

Devam eden bölümde, bu fonksiyonlara karşılık gelecek olan, minimize BDD haritalamaları

verilecektir. Kodlama, bir önceki bölümdekine benzer bir şekilde gerçekleştirilir. (5.7)'de, dört değişkenli bir lojik fonksiyona ilişkin genel açılım verilmiştir.

$$F = a'b'c'k_1 + a'b'ck_2 + a'bc'k_3 + a'bck_4 + ab'c'k_5 + ab'ck_6 + abc'k_7 + abck_8 \quad (5.7)$$

(5.7)'nin belli bir düzen içerisinde verilmiş olduğu, bu dizilişin, sonuçta kodu oluştururken, iki haneli kod parçalarının arka arkaya eklenme sırasıyla ve BDD, dolayısıyla geçiş transistör lojik devre modeli sıralanışı ile çok yakından ilgisinin olduğu vurgulanmalıdır. Şu aşamada, gerçeklememiz gereken üç adet fonksiyon var ise (5.4), (5.5) ve (5.6), her bir fonksiyon için ayrı kodlar oluşturmamızdır.

Çizelge 5.3 İki bit toplayıcıya ilişkin kod tablosu

| a | b | c | F     | F    | code |
|---|---|---|-------|------|------|
| 0 | 0 | 0 | $k_1$ | 0    | 00   |
| 0 | 0 | 1 | $k_2$ | $d'$ | 01   |
| 0 | 1 | 0 | $k_3$ | $d$  | 10   |
| 0 | 1 | 1 | $k_4$ | 1    | 11   |
| 1 | 0 | 0 | $k_5$ | 0    | 00   |
| 1 | 0 | 1 | $k_6$ | $d'$ | 01   |
| 1 | 1 | 0 | $k_7$ | $d$  | 10   |
| 1 | 1 | 1 | $k_8$ | 1    | 11   |

(5.7)'de verilen genel ifade ve kod tablosu beraber kullanılarak lojik fonksiyona ait kodlara ulaşıldığını söylemiştik. Kısaca tekrarlamak gerekirse,  $Y_0$  fonksiyonu için, ilk araştırmamız gereken yapı,  $a'b'c'k_1$  yapısıdır. Bu yapıda bir minterime rastlamadığımıza göre,  $k_1$  için 0'dır deriz ve kod tablosunda (Çizelge 5.3) 0'a karşılık gelen kod parçasını, dört değişkenli bu lojik fonksiyon için 16 uzunluklu olacağını bildiğimiz kodun ilk iki hanesini 00 olarak elde ederiz. Bir sonraki aşamada, aramamız gereken yapı,  $a'b'ck_2$  yapısıdır. Yine,  $Y_0$  (5.4) içerisinde araştırarak,  $k_2$ 'ye karşılık gelecek değeri elde ederiz. Bu araştırma ile ikinci iki bitlik kod parçası elde edilir. Bu işlem, sekiz adet  $k_n$  için tekrarlanarak, toplam kod, ikili kod parçalarının sıralı bir biçimde arka arkaya eklenmesi ile elde edilir.  $K_n$  değerlerini elde etmenin, diğer ve daha pratik bir yolu ise, ilgili fonksiyon içerisinde,  $k_n$ 'lere karşılık gelen girişlerin uygulanmasıdır. Örnek vermek gerekirse,  $a'b'c'k_1$  minterimini, fonksiyon içerisinde araştırmak yerine,  $k_1$ 'e karşılık gelen girişler  $a(0)$ ,  $b(0)$  ve  $c(0)$  (Çizelge 5.3) fonksina uygulanarak  $k_1$  elde edilebilir. Özellikle, fonksiyonların kanonik halde verilmedikleri durumda bu yol oldukça pratiktir. Aşağıda, toplayıcıya ilişkin  $Y_0$ 'a,  $Y_1$ 'e ve  $Y_2$ 'ye ait elde

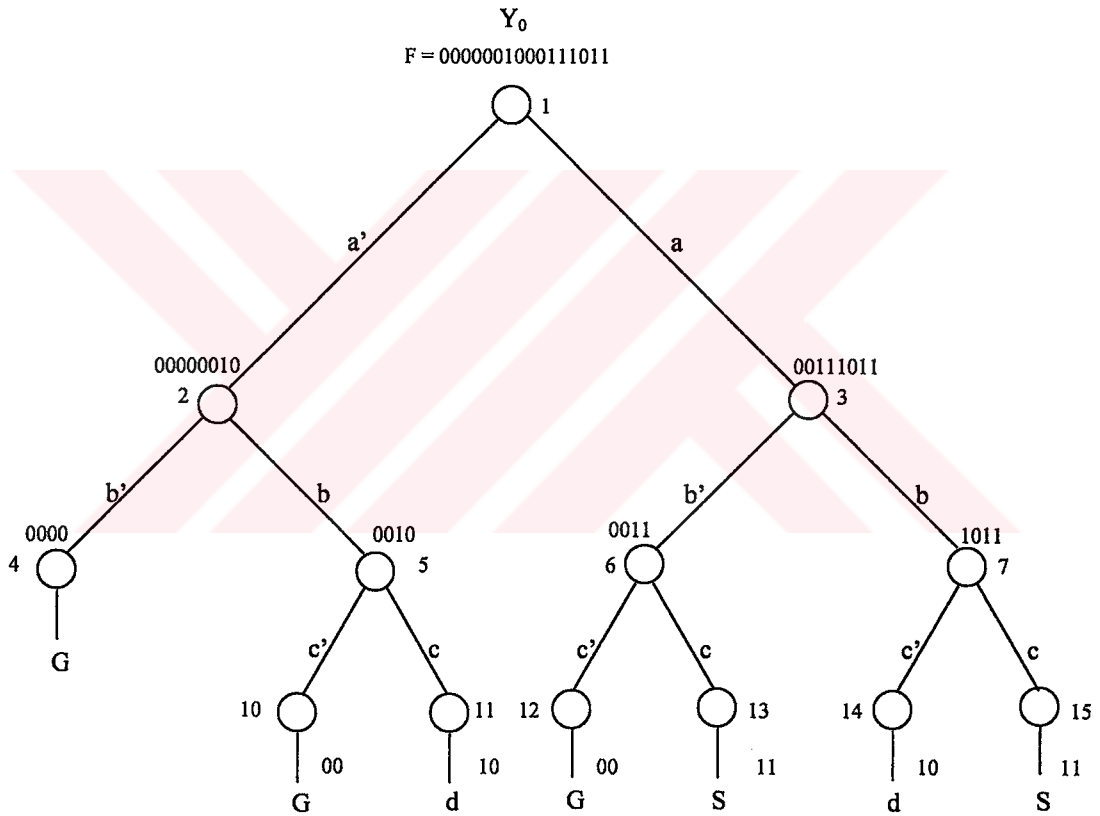
edilen kodlar verilmiştir.

$$Y_0 = 0000001000111011 \quad (5.8)$$

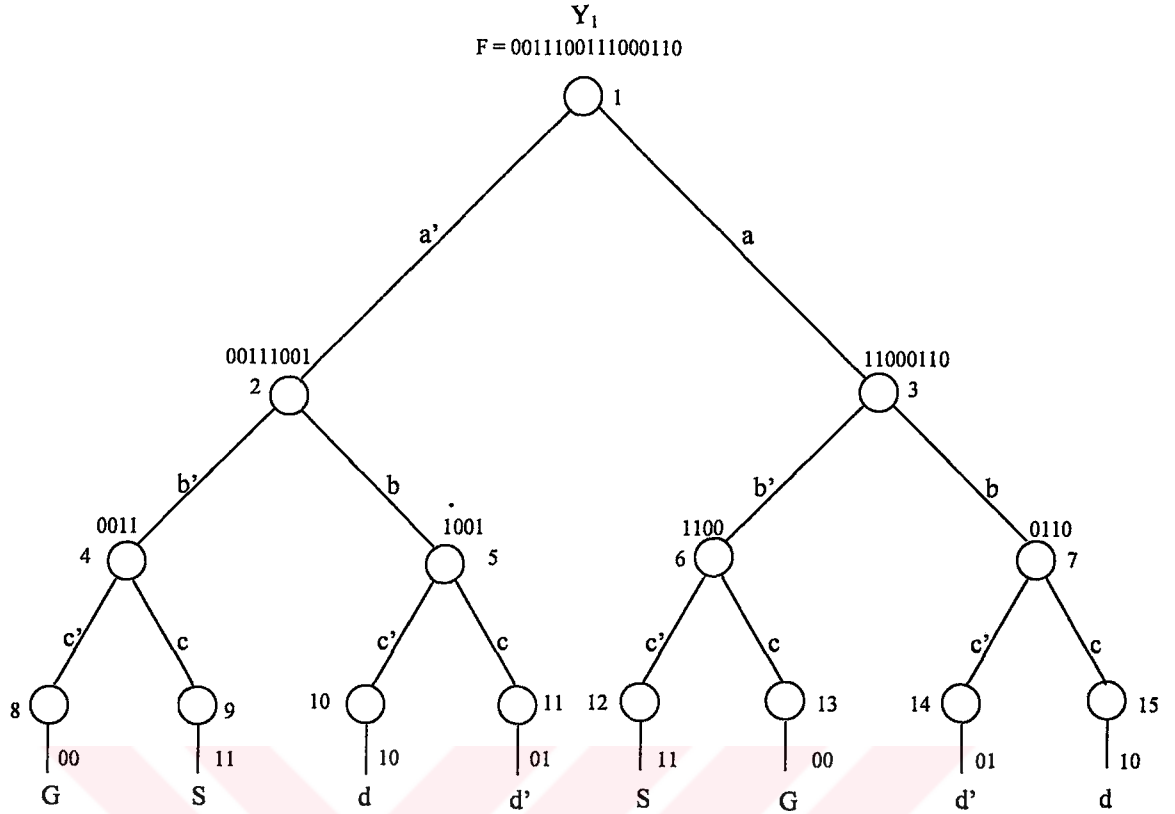
$$Y_1 = 0011100111000110 \quad (5.9)$$

$$Y_2 = 1010010110100101 \quad (5.10)$$

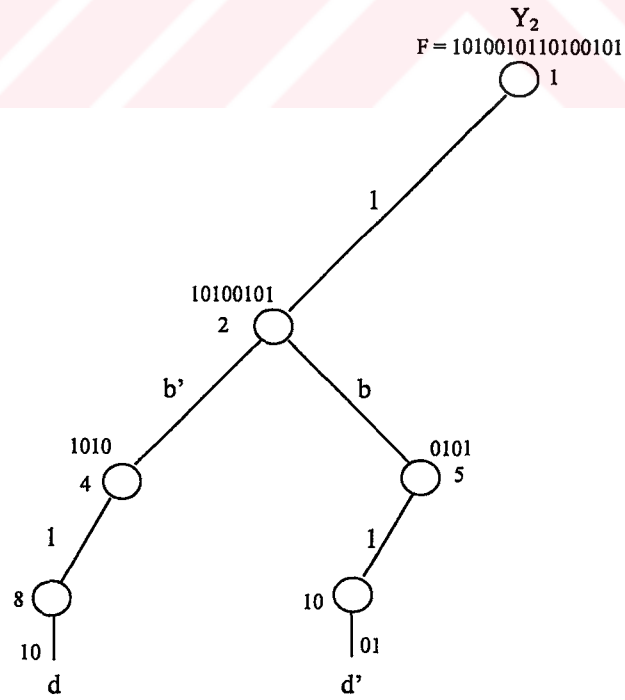
Kodları elde edilen fonksiyonlara ilişkin karar diyagramları, sırasıyla Şekil 5.15, Şekil 5.16 ve Şekil 5.17'de verilmiştir. Şekil 5.18, Şekil 5.19 ve Şekil 5.20'de ise, ilgili fonksiyonlara ait geçiş transistör devre modelleri verilmiştir.



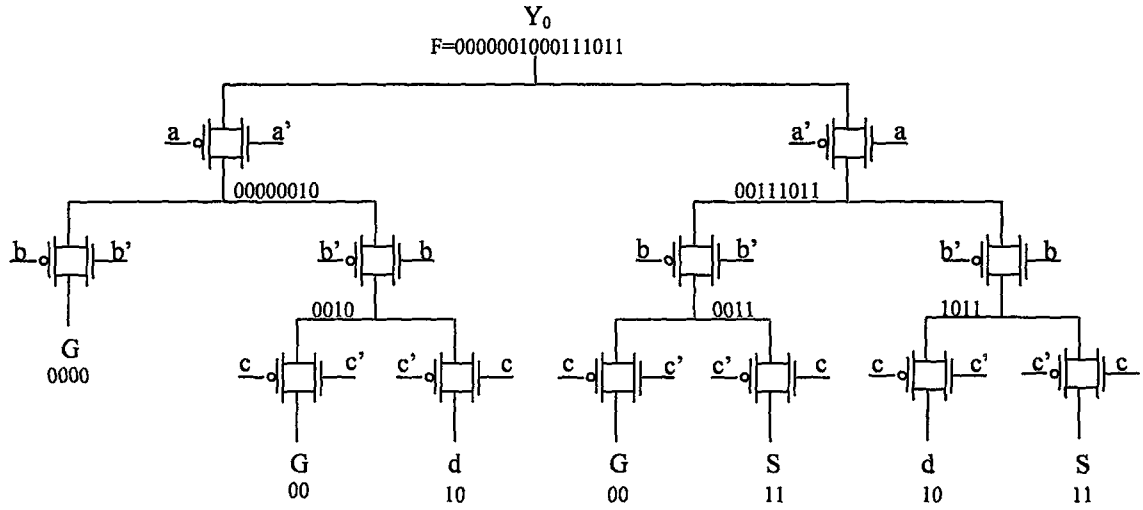
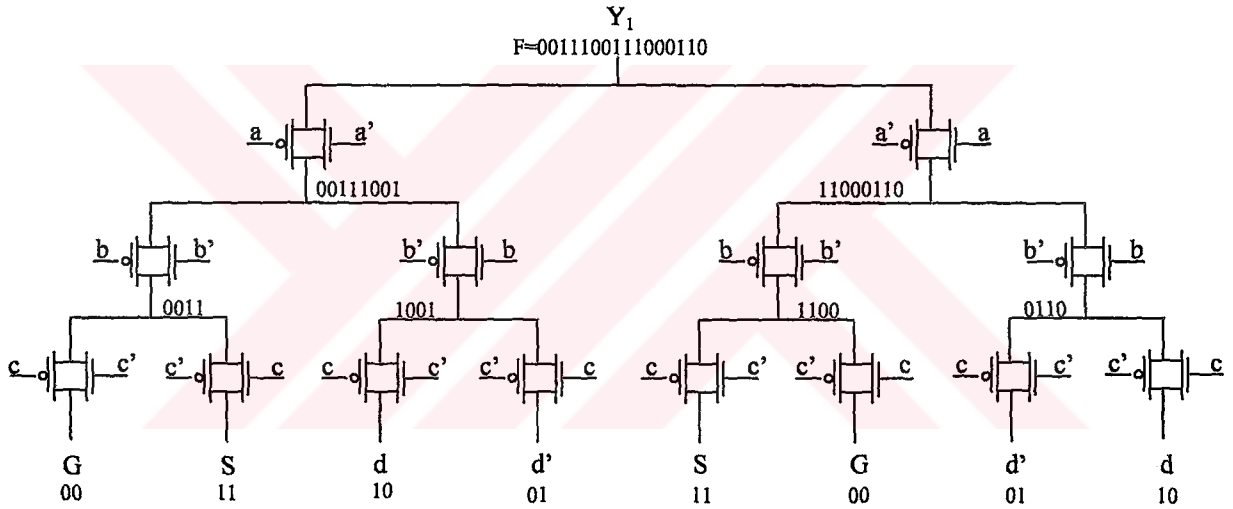
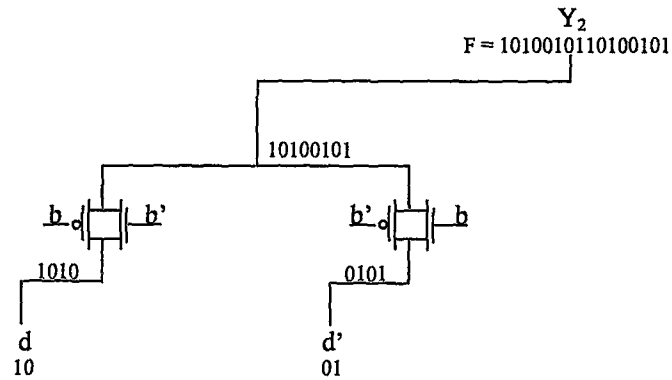
Şekil 5.15  $Y_0$  lojik fonksiyonuna ait karar diyagramı gösterimi



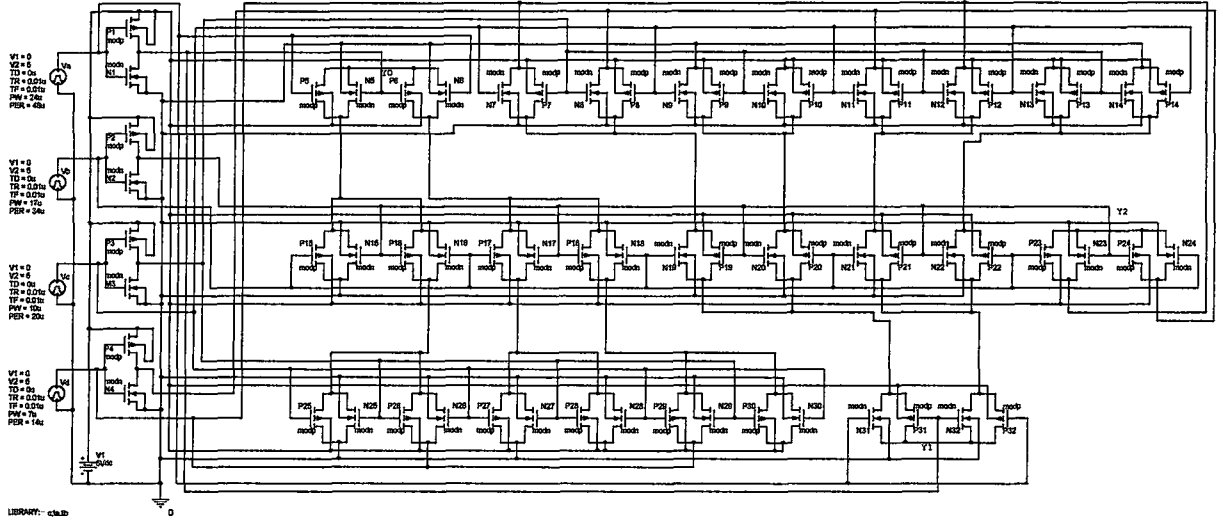
Şekil 5.16  $Y_1$  lojik fonksiyonuna ait karar diyagramı gösterimi



Şekil 5.17  $Y_2$  lojik fonksiyonuna ait karar diyagramı gösterimi

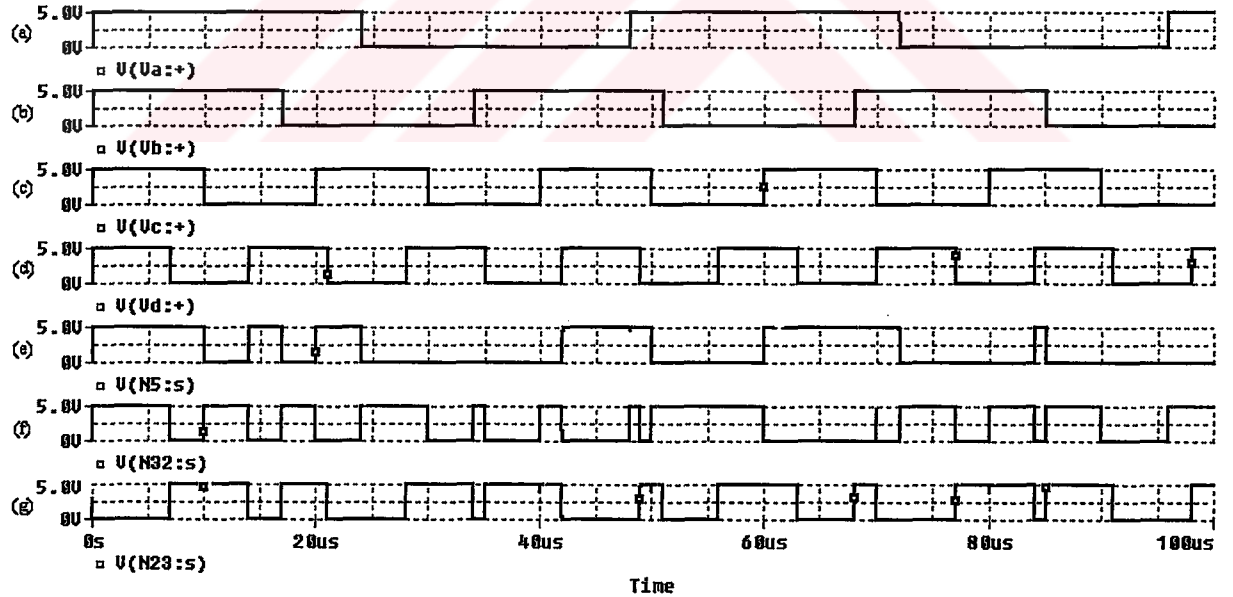
Şekil 5.18  $Y_0$  lojik fonksiyonuna ait geçiş transistör lojik devre modeliŞekil 5.19  $Y_1$  lojik fonksiyonuna ait geçiş transistör lojik devre modeliŞekil 5.20  $Y_2$  lojik fonksiyonuna ait geçiş transistör lojik devre modeli

İlgili lojik fonksiyonlarına ait geçiş transistör devre modelleri elde edildikten sonra, bu modelleri birleştirerek tüm devreyi Şekil 5.21’de gösterildiği şekilde oluştururuz.



Şekil 5.21 İki bit toplayıcıya ilişkin geçiş transistör lojik devresi

Şekil 5.22’de, iki bit toplayıcı devresine ait simülasyon sonuçları yer almaktadır.



Şekil 5.22 İki bit toplayıcıya ilişkin simülasyon sonuçları, (a), (b), (c) ve (d) kaynakları, (e), (f) ve (g) sırasıyla  $Y_0$ ,  $Y_1$  ve  $Y_2$  çıkışlarını göstermektedir.

## 6. SONUÇ

Bu çalışmaya, Avcı ve Yıldırım'ın (2003) önerdikleri, genelleştirilmiş geçiş transistör lojik devre sentez modeli, genel şeklini vermiştir. Ayrıca, Jaekel, Bandyopadhyay ve Jullien'in (1998) üzerinde çalıştıkları 123 karar diyagramları da, ele aldığı konunun paralelliği itibariyle önemli bir kaynak olmuştur.

Özellikle son on senedir, geçiş transistör lojik devre modelleri üzerine yapılan çalışmaların ışığında, karar diyagramlarının, sayısal devre sentezi için, oldukça başarılı bir model olduğu görülmüştür. Karar diyagramlarını ele alan çalışmalar, karar diyagramlarının, bilgisayar ortamında yazılım geliştirmeye olan uyumuna dikkati çekmişler ve genel veya özel amaçlı birtakım yazılımlar geliştirdiklerinden bahsetmişlerdir. Bu iki noktası itibariyle karar diyagramları, sayısal devreler ve tasarımcılar arasında mükemmel bir arayüz oluşturmaktadır.

Yapılan bu çalışmada, karar diyagramlarının bu iki özelliği birleştirilerek, görsel bir yazılım geliştirilmiştir. Geliştirilen yazılım ile bir lojik fonksiyona ait karar diyagramını ve geçiş transistör lojik devre modelini elde etmek mümkün olmaktadır. Bu sayede, bir lojik fonksiyonu, fiziksel olarak gerçeklemek oldukça hızlı bir şekilde mümkün olmaktadır.

Yine çalışmada birtakım uygulamalara yer verilmiş ve bu uygulamalar vasıtası ile yapılan çalışma ana hatları ile daha da somut bir şekil almış ve geliştirilen yazılımın doğruluğu gösterilmiştir.

**KAYNAKLAR**

- Abu-Khater, I.S., Bellaouar, A. ve Elmasry, M.I., (1996), "Circuit Techniques for CMOS Low-Power High-Performance Multipliers", *Solid-State Circuits, IEEE Journal of*, Volume: 31, Issue:10, Pages:1535-1546.
- Akers, S.B., (1978), "Representation of Switching Circuits by Binary-Decision Programs", *The Bell System Technical Journal*, Vol.38, pp. 985-999.
- Avci, M. ve Yildirim, T., (2003), "General Design Method for Complementary Pass Transistor Logic Circuits", *Electronics Letters*, Volume: 39, Issue: 1, Pages: 46-48.
- Avci, M. ve Yildirim, T., (2003), "A Coding Method for 123 Decision Diagram Pass Transistor Logic Circuit Synthesis", *Circuits and Systems*, 2003. ISCAS'03. Proceeding of 2003 International Symposium on, Volume: 5, Pages: V-189-V-192 vol.5.
- Bertacco, V., Minato, S., Verpleatze, P., Benini, L. Ve Micheli, G.D., (1997), "Decision Diagrams and Pass Transistor Logic Synthesis", *Int. Workshop on Logic Synth.*
- Brace, K.S., Rudell, R.L. ve Bryant, R.E., (1990), "Efficient Implementation of a BDD package", in *Proc.DAC*, pages 40-45.
- Bryant, R.E., (1995), "Binary Decision Diagrams and Beyond: Enabling Technologies for Formal Verification", *Computer-Aided Design. ICCAD-95. Digest of Technical Papers. 1995 IEEE/ACM International Conference on*, Pages: 236-243.
- Bryant, R.E., (1992), "Symbolic Boolean Manipulation with Ordered Binary-Decision Diagrams", *ACM Computing Surveys*, 24(3):293-318.
- Bryant, R.E., (1991), "On the Complexity of VLSI Implementations and Graph Representations of Boolean Functions with Application", *IEEE Trans. On Comp.*, 40: 205-213.
- Bryant, R.E., (1986), "Graph-Based Algorithms for Boolean Function Manipulation", *IEEE Trans. on Computers* 35, 677-691.
- Jeakel, A., Bandyopadhyay, S. ve Jullien, G.A., (1998), "Design of Dynamic Pass-Transistor Logic Circuits Using 123 Decision Diagrams", *IEEE Trans. on Circuits and Systems-I*, vol.45, no.11, pp.1172-1181.
- Leblebici, Y. ve Kang, S. (1996). *CMOS Digital Integrated Circuits: Analysis and Design*, Mc-Graw Hill Publishing Co., New York.
- Lee, C.Y., (1959), "Representation of Switching Circuits by Binary-Decision Programs", *The Bell System Technical Journal*, Vol.38, pp.985-999.
- Minato, S., Ishiura, N. ve Yajima, S., (1990), "Shared Binary Decision Diagram with Attributed Edges for Efficient Boolean Function Manipulation", *Design Automation Conference Proceedings. 27th ACM/IEEE*, Pages 52-57.
- Munteanu, M., Bogdan, I., Ivey, P., Powell, N., Psilogorgopoulos, M., Seed, L. ve Chuang, T.S., (1999), "Single-Ended Pass Transistor Logic for Low-Power Design", *Signals, Systems and Computers. Conference Record of Thirty-Third Asilomar Conference on*, Volume: 1, Pages: 364-368 vol.1.
- Munteanu, M., Bogdan, I., Ivey, P. ve Seed, L., (2001), *Single-ended Pass-transistor Logic*

(SPL)- a Design Handbook, Electronic System Group, The Department of Electronic and Electrical Engineering, The University of Sheffield.

Scholl, C. ve Becker, B., (2000), "On the Generation of Multiplexer Circuits for Pass Transistor Logic", Design, Automation and Test in Europe Conference and Exhibition 2000. Proceedings, Pages: 372-378.

Suzuki, M., Ohkubo, N., Yamanaka, T., Shimizu, A. ve Sasaki, K., (1993), "A 1.5 ns 32 b CMOS ALU in Double Pass-Transistor Logic", Solid-State Circuits Conference, 1993. Digest of Technical Papers. 40th ISSCC, 1993. IEEE International, 24-26 Feb. 1993, Pages 90-91, 267.

Yang, C., Singhal, V. ve Ciesielski, M., (1999), "BDD Decomposition for Efficient Logic Synthesis", in International Conference on Computer Design, pp. 626-631.

Yang, C. ve Ciesielski, M., (1999), "Synthesis for Mixed CMOS/PTL Logic: Preliminary Results", in International Workshop on Logic Synthesis.

Yano, C., Sasaki, Y., Rikino, K. ve Seki, K., (1996), "Top-Down Pass Transistor Logic Design", IEEE J. Solid-State Circuits, vol.31, no.6, pp.792-803.

Zimmermann, R. ve Fichtner, W., (1997), "Low-Power Logic Styles: CMOS versus Pass Transistor Logic", IEEE Journal of Solid State Circuits, 32(7): 1079-1090.

#### **İNTERNET KAYNAKLARI**

[1] [http://www.cic.unb.br/docentes/jacobi/pesquisa/tese/MBD\\_Definition.pdf](http://www.cic.unb.br/docentes/jacobi/pesquisa/tese/MBD_Definition.pdf)

[2] <http://www.cs.sunysb.edu/~wenxin/bdd.pdf>

**ÖZGEÇMİŞ**

Doğum Tarihi 23.04.1976

Doğum Yeri İstanbul

Lise 1991-1994 Trabzon Yomra Fen Lisesi

Lisans 1995-2000 Yıldız Teknik Üniversitesi Elektrik-Elektronik Fak.  
Elektronik ve Haberleşme Müh. Bölümü

Yüksek Lisans 2000-2004 Yıldız Teknik Üniversitesi Fen Bilimleri Enstitüsü  
Elektronik ve Haberleşme Müh. Anabilim Dalı  
Elektronik Programı

**Çalıştığı kurum**

2003-Devam ediyor. YTÜ Bilgisayar Müh. Araştırma Görevlisi