

175775

**KISMEN BELİRLİ ARDIŞIL MAKİNELER İÇİN YENİ BİR
DURUM İNDİRGEME YÖNTEMİ VE BU YÖNTEMİN
ÇEŞİTLİ LOJİK TASARIM KONULARINA
UYGULANMASI**

DOKTORA TEZİ
Y. Müh. Orhan UÇAR
(504960013012)

Tezin Enstitüye Verildiği Tarih : 30 Haziran 2003
Tezin Savunulduğu Tarih : 23 Aralık 2003

Tez Danışmanı : Prof.Dr. Ahmet DERVİŞOĞLU (2.6.2004)
Diğer Jüri Üyeleri Prof.Dr. Bülent ÖRENCİK (2.6.2004)
Doç.Dr. Ece Olcay GÜNEŞ (2.6.2004)
Prof.Dr. Ertuğrul ERİŞ (Y.T.Ü.) (16.6.2004)
Yrd.Doç.Dr. Arda YURDAKUL (B.Ü.) (11.06.04)

ARALIK 2003

ÖNSÖZ

Bu Tez çalışmasında çok büyük emekleri bulunan Sayın Prof.Dr. Ahmet Dervişoğlu'na ve büyük desteğinden dolayı sevgili eşim Yük.Müh. Hülya Uçar'a teşekkürlerimi sunarım.

Aralık, 2003

Yük.Müh. Orhan Uçar



İÇİNDEKİLER

KISALTMALAR	v
TABLO LİSTESİ	vi
ŞEKİL LİSTESİ	vii
SEMBOL LİSTESİ	ix
ÖZET	x
SUMMARY	xiii
1. GİRİŞ	1
2. KISMEN BELİRLİ ARDIŞIL MAKİNELER İÇİN YENİ BİR DURUM İNDİRGEME YÖNTEMİ	6
2.1. Tanımlar	6
2.2. Diğer Durum İndirgeme Yöntemleri	8
2.3. Bu Çalışmada Geliştirilen İndirgeme Yöntemi	10
2.4. Algoritmanın Adımları	11
2.5. Önerilen Yöntemin Diğer Yöntemlerden Temel Farklılıkları	13
2.6. İndirgeme Örneği 1	14
2.7. İndirgeme Örneği 2	20
2.8. İndirgeme Örneği 3	22
2.9. Test Sonuçları	24
3. DURUM İNDİRGEME İÇİN GELİŞTİRİLEN YÖNTEMLERİN BOOLE FONKSİYONLARININ İNDİRGENMESİNE UYGULANMASI	26
3.1. Geliştirilen Yöntemin Adımları	27
3.1.1. Algoritma 1	27
3.1.2. Algoritma 2	27
3.1.3. Algoritma 3	28
3.2. İndirgeme Örneği 1	28
3.3. İndirgeme Örneği 2	31
3.4. Sonuç	34
4. DURUM İNDİRGEME İÇİN GELİŞTİRİLEN YÖNTEMLERİN ASENKRON ARDIŞIL DEVRELERİN KODLANMASI PROBLEMİNE UYGULANMASI	38
4.1. Geliştirilmiş Olan Durum Kodlama Yöntemi	40
4.1.1. Minimum geçiş zamanlı kodlama için durum kümesine uygulanan bölmeleme	42
4.1.2. Bir sütunun tanımladığı bölmeleme sayısı	43
4.1.3. Durum kodlama yöntemi	44
4.2. Test Sonuçları	48

5. TEK KOŞULLU BİR ÖRTÜ PROBLEMİNİN ÇÖZÜMÜ İÇİN BİR YÖNTEMİN VE BU YÖNTEME DAYANAN BİR BİLGİSAYAR PROGRAMININ GELİŞTİRİLMESİ	49
5.1. Geliştirilen Yöntem	50
5.2. Sonuçlar	52
6. GENEL ÖRTÜ PROBLEMLERİNİN ÇÖZÜMÜNDE KULLANILABİLECEK YENİ BİR YÖNTEM	54
6.1. Tanımlar	55
6.2. Minimal Örtüyü Oluşturan Blokların Sayısının Hesaplanması	55
7. SONUÇLAR VE TARTIŞMA	68
7.1. Durum İndirgeme İçin Yeni Bir Yöntemin Geliştirilmesi	68
7.2. Durum İndirgeme İçin Geliştirilen Yöntemlerin Boole Fonksiyonlarının İndirgenmesine Uygulanması	69
7.3. Durum İndirgeme İçin Geliştirilen Yöntemlerin Ardışıl Devrelerin Kodlanmasına Uygulanması	70
7.4. Tek koşullu Örtü Problemlerinde Kullanılabilecek Yeni Bir Yöntemin Geliştirilmesi	70
7.5. Genel Örtü Problemlerinde Kullanılabilecek Yeni Bir Yöntemin Geliştirilmesi	71
KAYNAKLAR	73
EKLER	76
ÖZGEÇMİŞ	81

KISALTMALAR

BDD	: Binary Decision Diagram
CCP	: Closed Compatible Path
EBAY	: En Büyük Adımlar Yöntemi
FSM	: Finite State Machine
LB	: Lower Bound
MC	: Maximal Clique
MCNC	: Microelectronics Center of North Carolina
MORP	: Multiple Output Reduction Program
OPASKOD	: Optimal Asenkron Kodlama
SCC	: Strongly Connected Component
SRC	: State Reduction and Covering
UB	: Upper Bound



TABLO LİSTESİ

	<u>Sayfa No</u>
Tablo 2.1 SRC'nin bençmark testleri sonuçları	25
Tablo 3.1 Asal bileşen listesi	29
Tablo 3.2 MCNC bençmarkları	36
Tablo 3.3 MÖRP'un bençmark testleri sonuçları	37
Tablo 4.1 OPASKOD bençmark test sonuçları	48
Tablo 6.1 $f_m \neq 0$ olan örtü problemlerinin sayısının eleman sayısına (n) ve m_2 ' ye göre dağılımı	57



ŞEKİL LİSTESİ

	<u>Sayfa No</u>
Şekil 1.1	Lojik devrenin genel gösterilimi 2
Şekil 2.1	Kısmen belirli bir ardışıl makine 7
Şekil 2.2	M_1 makinesine ilişkin durum tablosu 15
Şekil 2.3	M_1 makinesine ilişkin çiftler tablosu 15
Şekil 2.4	M_1 makinesine ilişkin çiftler tablosu ve ağırlıklar 16
Şekil 2.5	M_1 makinesine ilişkin maksimal uyumluların hesabı 17
Şekil 2.6	M_1 makinesine ilişkin uyumlu çiftler grafi 17
Şekil 2.7	M_1 makinesine ilişkin maksimal uyumlular grafi 18
Şekil 2.8	M_1 makinesine ilişkin asal uyumlular grafi 18
Şekil 2.9	M_1 makinesine ilişkin kapalı yollar 19
Şekil 2.10	Minimal kapalı örtü arama ağacı 19
Şekil 2.11	Bençmark ex2'nin durum tablosu 20
Şekil 2.12	Ex2 bençmarkı için minimal kapalı örtüyü arama ağacı 22
Şekil 3.1	Örtü matrisi 29
Şekil 3.2	Karnaugh ile ortak indirgeme 30
Şekil 3.3	Minimale yakın devre 30
Şekil 3.4	Minimal devre 31
Şekil 3.5	Örtü matrisi 31
Şekil 3.6	İkinci adımda örtü matrisi 32
Şekil 3.7	Son adımda örtü matrisi 32
Şekil 3.8	Örtü arama ağacı 33
Şekil 3.9	Örtü arama ağacında son aşama 34
Şekil 4.1	Ardışıl makine A 41
Şekil 4.2	Makina A'nın durumlarının 3 farklı kodlaması 41
Şekil 4.3	Bölmelemeler matrisi 45
Şekil 4.4	Maksimal uyumlular kümesinin elde edilmesi 46
Şekil 4.5	A makinasının durumlarının kodları 46
Şekil 4.6	B makinasının durumlarının tablosu 46
Şekil 4.7	B makinesinin bölmeleme listesi 47
Şekil 4.8	B makinesi için örtü matrisi 47
Şekil 4.9	B makinesinin kritik yarışsız kodlanması 48
Şekil 5.1	8 bloklu ve 23 elemanlı bir örtü problemi 50
Şekil 6.1	$n=5$ için $f_m \neq 0$ olan bir durumda uyumsuzluk grafi 56
Şekil 6.2	$n=5$ değişkenli $f_m=1$ olan bütün FSM'lerin uyumsuzluk grafları 58
Şekil 6.3	$n=6$ değişkenli $f_m=1$ olan bütün FSM'lerin uyumsuzluk grafları 58
Şekil 6.4	$n=6, m_2=3$ için $f_m=1$ olan bir örtü probleminin uyumsuzluk grafi 60

Şekil 6.5	$n=9, m_2=3$ için $f_m=1$ olan bir örtü probleminin uyumsuzluk grafi	60
Şekil 6.6	$n=10, m_2=2$ için $f_m=1$ olan bir örtü probleminin uyumsuzluk grafi	61
Şekil 6.7	$n=10, m_2=4$ ve $f_m=2$ olan bir örtü probleminin uyumsuzluk grafi	61
Şekil 6.8	$n=15, m_2=2$ ve $f_m=1$ olan bir örtü probleminin uyumsuzluk grafi	62
Şekil 6.9	$n=15, m_2=6$ ve $f_m=3$ olan bir örtü probleminin uyumsuzluk grafi	63
Şekil 6.10	$n=7, m_2=2$ için $f_m=1$ olan bir örtü probleminin uyumluluk ve uyumsuzluk grafi	64
Şekil 6.11	$n=9, m_2=2$ için $f_m=1$ olan bir örtü probleminin uyumluluk ve uyumsuzluk grafi	64
Şekil 6.12	$n=14, m_2=2$ için $f_m=1$ olan bir örtü probleminin uyumsuzluk grafi	65
Şekil 6.13	$n=12, m_2=2$ için $f_m=2$ olan bir örtü probleminin uyumsuzluk grafi	66
Şekil 6.14	$n=18, m_2=2$ için $f_m=1$ olan bir örtü probleminin uyumsuzluk grafi	67
Şekil 6.15	$n=102, m_2=2$ için $f_m=1$ olan bir örtü probleminin uyumsuzluk grafi	67



SEMBOL LİSTESİ

B_i	: i'nci uyumlu blok
C_a	: Asal uyumlular sınıfı
C_p	: Uyumlu çiftler kümesi
C_s	: Maksimal uyumlular sınıfı
$\overline{C_s}$	: Maksimal uyumsuzlar sınıfı
$ C_s $	: C_s sınıfındaki blok sayısı
f_d	: $n-f_n$
f_m	: m_0-m_2
f_n	: m_2 tane blok ile kapsanabilecek maksimum değişken sayısı
$I(B_i)$	: B_i bloğunun öz-izleyici sınıfı
m_2	: Uyumsuzlar sınıfındaki en büyük bloğun içerdiği değişken sayısı
m_0	: Minimal örtüdeki blok sayısı
M_2	: $\overline{C_s}$ uyumsuzlar sınıfındaki en büyük blok
M	: Minimal makinenin bloklarının oluşturduğu küme
n	: Değişken sayısı
P_i	: i'nci kapalı yol
P_a	: Asal uyumlulardan oluşan kapalı yollar
P_p	: Uyumlu çiftlerden oluşan kapalı yollar
P_s	: Maksimal uyumlulardan oluşan kapalı yollar
P_u	: Uyumlulardan oluşan kapalı yollar
s_i	: i'nci durum
S_i	: i'nci durumu içeren kapalı yollar kümesi

KISMEN BELİRLİ ARDIŞIL MAKİNELER İÇİN YENİ BİR DURUM İNDİRGEME YÖNTEMİ VE BU YÖNTEMİN ÇEŞİTLİ LOJİK TASARIM KONULARINA UYGULANMASI

ÖZET

VLSI devreleri modern elektronik ürünlerinde yaygın olarak kullanılmaktadır. Planar tümdevrenin 1959 yılında Robert Noyce ve Jack Kelby tarafından bulunmasından sonra tek bir çipe sığdırılabilen tranzistor sayısı her iki yılda bir hemen hemen iki katına çıkmıştır. VLSI devrelerinin karmaşıklığı ve performans gereksinimleri üstel olarak artarken, tasarım süreçlerinde bilgisayar destekli tasarım araçlarının kullanılması zorunluluk haline gelmiştir. Bu Tezde kısmen belirli ardışıl makinelerde durum indirgeme konusu ele alınmıştır.

Durum indirgeme senkron ve asenkron ardışıl devre tasarımında en önemli adımlardan biridir. Durum indirgeme için geliştirilen algoritmalar, durum kodlama ve tek koşullu örtü problemi çözme gibi lojik tasarımın diğer adımlarında da kullanılabilir.

Bu Tezde, verilen bir kısmen belirli ardışıl sonlu durum makinesinin minimal eşdeğerini bulmak için üç ana algoritmadan oluşan yeni bir yöntem geliştirilmiştir. İlk algoritma maksimal uyumlular sınıfı C_s 'yi ve C_s sınıfından minimele yakın bir kapalı örtü elde etmektedir. İkinci algoritma uyumlu durum çiftleri kümesi C_p 'den kapalı yollar kümesi P_p 'yi, maksimal uyumlular sınıfı C_s 'den kapalı yollar kümesi P_s 'yi, uyumlular sınıfı C_u 'dan kapalı yollar kümesi P_u 'yu ve asal uyumlular sınıfı C_a 'dan kapalı yollar kümesi P_a 'yı elde etmektedir. Üçüncü algoritma P_u , P_a ve P_s kapalı yollar kümelerinin herhangi birinden bir minimal kapalı örtü elde edebilmektedir. P_s kümesinden elde edilen kapalı örtü bir üst sınır vermekte, bazen minimal kapalı örtü bu yolla elde edilebilmektedir. Minimal makine P_u ve P_a kümelerinin herhangi birinden elde edilebilmektedir. Kapalı yollar kümesi P_a kullanıldığında minimal makine çok kısa sürede elde edilebilmektedir. P_a kapalı yollar kümesi kavramına ve bu kümeden minimal kapalı örtü elde etmeye dayanan bir yöntemle literatürde rastlanmamıştır. Bu algoritmalara dayanan, SRC (State Reduction and Covering) adı verilen bir bilgisayar programı geliştirilmiştir. Geliştirilen program MCNC ve diğer bençmarklarda Rho yöntemi [1] ve Puri yöntemi [2] ile karşılaştırılmıştır. P_p kapalı yollar kümesinden minimal makine elde etmeye dayanan Rho yöntemi ile mark1 ve ex3 gibi bazı bençmarklarda minimal makine elde edilememiştir. Bu Tezde geliştirilen yöntem ve bu yöntemle dayanan SRC programı ile minimal makine daima elde edilebilmekte ve sonuca hızlı bir şekilde ulaşılabilir. SRC programı başta MCNC bençmarkları olmak üzere pek çok bençmark için bir PC ile test edilmiştir. Kullanılan PC'nin işlemcisi

Pentium 4 1.8GHz ve belleği 256MB'dır. Test sonuçları Bölüm 2' de verilmiştir. Test sonuçlarından da görüldüğü gibi SRC programı özellikle kritik benchmarklarda daha iyi sonuç vermiştir.

Durum indirgeme için geliştirilen algoritmalar, tek koşullu ve iki koşullu örtme problemi içeren başka alanlarda da kullanılabilir. Bu Tezde durum indirgeme için geliştirilen algoritmalar, n değişkenli m tane Boole fonksiyonunun VE/VEYA kombinezonsal devresi ile gerçekleştirilmesi problemine uygulanmıştır. Bu şekilde elde edilen algoritmaya dayanan, MORP (Multiple Output Reduction Program) adı verilen bir bilgisayar programı geliştirilmiştir. MORP ile, verilen Boole fonksiyonlarının iki seviyeli minimal ifadesi elde edilmektedir. Algoritma ile daima bir minimal çözüm veya tüm minimal çözümler elde edilebilmektedir. Bununla birlikte eğer minimal çözümün elde edilmesi çok uzun zaman alıyorsa, algoritma minimale yakın bir çözümü de çok kısa sürede vermektedir.

MORP, MCNC (Microelectronics Center of North Carolina) benchmarklarında test edilmiş ve elde edilen sonuçlar, UC Berkeley üniversitesinde geliştirilen ESPRESSO programının verdiği sonuçlar ile karşılaştırılmıştır. Test sonuçları Bölüm 3'de verilmiştir. Bu sonuçlardan da görüldüğü gibi MORP ile minimal çözümler kabul edilebilir sürelerde elde edilmiş ve 9 tane benchmarkta ESPRESSO programına ilişkin sürelerden daha kısa sürelerde çözümlere ulaşılmıştır.

Durum indirgeme için geliştirilen algoritmalar durum kodlama konusuna da uygulanmış ve OPASKOD (OPTimal ASenkron KODlama) adı verilen bir bilgisayar programı geliştirilmiştir. OPASKOD durum kodlama benchmarkları ile test edilmiş ve minimum geçiş zamanlı kodlamalar kısa sürede elde edilmiştir. Test sonuçları Bölüm 4'te verilmiştir.

Bu Tezde genel örtü problemlerini çözmek için geliştirilmiş olan bir yöntem ve bu yönteme dayalı bir bilgisayar programı UCPS (Unate Covering Problem Solver) ile, tamamen keyfi olarak oluşturulan 1000×1000 mertebesindeki bir minimal örtü problemi saniyeler mertebesinde bir sürede çözülebilmektedir. Program, Boole fonksiyonlarının VE/VEYA devreleri ile minimal gerçekleştirilmesine ilişkin çeşitli benchmarklarda denenmiş ve çözümler kısa sürede belirlenmiştir. Geliştirilen program MCNC benchmarklarından elde edilen örtü matrislerine uygulanmıştır. Test sonuçları Bölüm 5'de verilmiştir. UCPS programı kullanılarak, verilen tek koşullu bir örtü problemi için, bir minimal örtü veya tüm minimal örtüler elde edilebilmektedir.

Bu Tezde ayrıca toplamlar çarpımı biçimindeki bir Petrick fonksiyonunu çarpımlar toplamı biçiminde açarak bir minimal örtüyü veya tüm minimal örtülerini veren bir bilgisayar programı PETS (Petrick Solver) geliştirilmiştir.

Bu Tezde, minimal örtüyü bulmadan, bir minimal örtüdeki blokların sayısını belirlemek için bir yöntem geliştirilmiştir.

Genel olarak $m_2 \leq n_2$ dir; burada m_2 en büyük maksimal uyumsuz bloktaki eleman sayısıdır. Minimal örtünün elde edilmesi pek çok uygulamada önemlidir. Minimal örtü şu şekilde elde edilebilir. En büyük maksimal uyumsuz bloktaki elemanlar için dallanma yöntemi kullanılarak veya Petrick fonksiyonu yazılarak bu elemanlar için tüm kısmi örtüler elde edilir. Eğer bu kısmi örtülerden herhangi birisi aynı zamanda bir örtü ise bu örtü bir minimal örtüdür. Bu çalışmada, en büyük maksimal uyumsuz

bloğun birden fazla sayıda olduđu durumda, eđer bir tanesi ile bir minimal  rt  elde edilebiliyorsa hepsinden bir minimal  rt  elde edilebildiđi g sterilmiřtir. Eđer herhangi bir tanesinden minimal  rt  elde edilemiyorsa, minimal  rt y  elde etmek i in ka  tane blok daha gereklidir? Bu  alıřmada m_2 ve m_8 arasındaki iliřkiler uyumluluk ve uyumsuzluk grafları kullanılarak arařtırılmıřtır. En b y k maksimal uyumsuz bloktaki eleman sayısı m_2 bilindiđinden, $f_m = m_8 - m_2$ belirlenebilirse, aynı zamanda deđerli bir bilgi olan m_8 belirlenebilir.   nk  bazı uygulamalarda m_8 deđerini bilmek yeterlidir. Ayrıca eđer m_8 deđerini bilinirse, bir minimal  rt y  elde etmek kolaylařmaktadır.



A NEW METHOD FOR THE STATE REDUCTION OF INCOMPLETELY SPECIFIED FINITE SEQUENTIAL MACHINES AND APPLICATION OF THE DEVELOPED METHODS TO THE OTHER LOGIC SYNTHESIS STEPS

SUMMARY

VLSI (Very Large Scale Integrated) circuits are widely used in modern electronic products. Since the invention of the planar integrated circuits by Robert Noyce and Jack Kelby in 1959, the number of transistors that can be successfully fabricated on a single chip has doubled almost every two years. As the complexity and performance requirements of VLSI circuits are increasing exponentially, the design process has to be automated by using computer-aided design (CAD) tools. A CAD tool is a computer program that can help an IC designer in the VLSI design process. This dissertation is concerned with the problem of synthesizing a class of logic circuits, more specifically state reduction of of incompletely specified sequential machine.

State reduction is an important step in the design of the synchronous and asynchronous sequential machines. The algorithms which are developed for state reduction can also be used for other topics of the logic synthesis, such as state encoding and covering problems.

In this dissertation, we present a method which includes three main algorithms to find a minimal equivalent of a given incompletely specified finite sequential machine. The first algorithm determines maximal compatible class C_s and a near-minimal closed cover from C_s . The second algorithm determines the closed paths, P_p , from the compatible pairs class C_p , closed paths P_u , from the compatible class C_u , closed paths P_a , from the prime compatible class C_a and closed paths P_s , from the maximal compatible class C_s . The third algorithm determines a minimal closed cover from any one of P_u , P_a and P_s . The closed cover obtained from P_s gives an upper bound and sometimes a minimal closed cover. A minimal machine can be obtained from any one of P_p , P_u and P_a . However a minimal machine is obtained in a very short time through the closed paths P_a . P_a closed paths concept and an algorithm which uses this concept, is not observed in the literature. The reduction algorithms are implemented in an efficient computer program SRC (State Reduction and Covering). In [1] an algorithm based on closed compatible pairs is given. However it can not find a minimal machine for some benchmarks such as mark1 and ex3. SRC always finds a minimal machine and it is very fast. SRC is run and tested using a PC (Pentium 4 1.8GHz with 256MB RAM), on several FSMs including the MCNC FSM benchmarks and the results are given in Chapter 5. From test results, it can be seen that, despite of the limited computing resources, SRC is more efficient on all benchmarks, especially on the critical benchmarks.

Some of the algorithms given in this dissertation, can also be used in other algorithms which deal with unate and binate covering problems.

In this dissertation we also present an algorithm to realize m Boolean functions with n variables by the use of a two level AND/OR combinational circuit. The algorithm is implemented in an efficient computer program MORP (Multiple Output Reduction Program), which gives the minimal expressions of the given Boolean functions. The algorithm always gives a minimal solution or all minimal solutions. However, if it takes too long time to obtain a minimal solution then the algorithm gives a near-minimal solution in a shorter time.

Using MORP, MCNC (Microelectronics Center of North Carolina) benchmarks are tested on a PC and the minimal solutions are obtained in reasonable time duration. Test results are compared with the results obtained with ESPRESSO which is developed in UC Berkeley. Test results are given in Chapter 3. As seen from the results, minimal solutions are obtained by MORP in a reasonable computer duration. Moreover, in 12 benchmarks MORP gives better results than ESPRESSO.

State reduction methods which are developed in this Thesis, are applied to state encoding problem and a computer program, OPASKOD (Race-free State Assignment Program) is developed. OPASKOD is tested on state encoding benchmarks and race-free encodings are obtained in a reasonable computer duration. Test results are given in Chapter 4.

In this Thesis, a method is developed to solve unate covering problems. Based on this method, a computer program UCPS (Unate Covering Problem Solver) is developed. Using UCPS, unate covering problems, expressed by a 1000×1000 matrix are solved in a few seconds. UCPS is tested with covering problems of MCNC benchmarks. Test results are given in Chapter 5. Using UCPS program, a minimal cover or all minimal covers can be obtained. Educational version of the UCPS is used to obtain all minimal covers.

In this Thesis, a program PETS (Petrick Solver) is also developed to obtain a minimal cover or all minimal covers using Petrick function.

In this Thesis, a method is developed to determine the number of blocks, m_b , in a minimal cover.

In general, $m_2 \leq m_b$, where m_2 is the number of elements in a maximal clique. To obtain a minimal cover is important for many applications. A minimal cover may be obtained in the following way. Using branching algorithm or writing Petrick function only for the elements of a maximal clique find all the partial covers of the elements of a maximal clique. If any one of these partial covers is also a cover, then it is a minimal cover. Suppose that there are more than one maximal cliques. We have shown that if one of them gives a minimal cover, then all of them gives a minimal cover. If none of them gives a minimal cover, then how many more blocks do we need to obtain a minimal cover? We have investigated the relation between m_2 and m_b using the compatibility and incompatibility graphs. Since m_2 is known, if we determine $f_m = m_b - m_2$ then m_b also is determined, which is a valuable information. Because, in some applications it is enough to know m_b . Furthermore, if m_b is known, then it becomes easy to find a minimal cover.

1. GİRİŞ

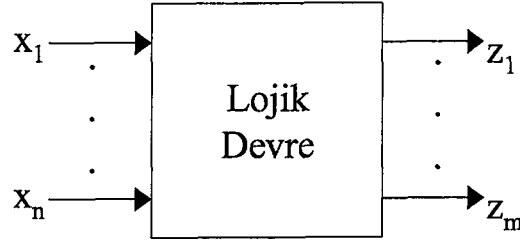
Elektrik-elektronik endüstrisi temel endüstri haline gelmiştir; telekomünikasyon, bilgisayar, uçak, tekstil, otomobil ve hemen her türlü endüstrinin temel öğelerinden biri elektrondur ve analog elektrik devreleri yerlerini dijital devrelere terketmektedir.

Aşağıda, dijital devrelerin başlıca üstünlükleri verilmiştir.

- 1) Tümüleşik devre olarak üretime elverişlidirler.
- 2) Güvenilirlikleri yüksektir.
- 3) Dinamikleri (en büyük işaretin, sıfırdan farklı en küçük işarete oranı) yüksektir.
- 4) İşaret yeniden üretilebilir (rejenerasyon).
- 5) Bellek elemanlarının hacmi küçük, kapasitesi büyük ve erişim süresi kısadır.
- 6) Gürültünün etkisi azdır.
- 7) Kodlama yoluyla hatalı işareti yakalamak ve düzeltmek mümkündür.
- 8) Mikroişlemci v.b. bileşenlerle yüksek marifetli sistemler gerçekleştirilebilir.

VLSI devreleri modern elektronik ürünlerinde yaygın olarak kullanılmaktadır. Tümüleşik devre tasarımı tekniğinin 1959 yılında geliştirilmesinden bu yana tek bir çipe sığdırılabilen transistor sayısı neredeyse her iki yılda bir iki katına çıkmaktadır. VLSI devrelerinin karmaşıklığı ve performans gereksinimleri zaman geçtikçe üstel olarak arttığından VLSI devre tasarımı işleminde CAD (Computer Aided Design) yazılımlarının kullanılması zorunluluk haline gelmiştir. Bu Tezde dijital devrelerin bir alt kümesi olan ardışıl devrelerin sentezinde kullanılan durum indirgeme ile ilgili yeni yöntemler geliştirilmiştir. Geliştirilen yöntemler Boole fonksiyonlarının indirgenmesi ve durum kodlama gibi diğer lojik tasarım konularına da uygulanmıştır.

Lojik devreler (i) Kombinezonsal (combinational), (ii) Ardışıl (sequential) lojik devreler şeklinde ikiye ayrılır. Genel olarak bir lojik devre, aşağıdaki şekilde görüldüğü gibi, n girişli, m çıkışlı bir devredir.



Şekil 1.1 Lojik devrenin genel gösterilimi

Bu şekilde gösterilen $x_1, x_2, \dots, x_n$ 'e serbest girişler denir; bunlar analog devrelerdeki bağımsız kaynaklara karşı düşer. $x_i, z_i \in \{0,1\}$ dir ve belirli bir anda bir giriş takımı uygulanıp karşı düşen çıkışlar alındıktan sonra yeni bir giriş takımı uygulanır ve karşı düşen çıkışlar alınır.

Belirli bir andaki girişler, o andaki çıkışları tamamen belirliyorsa, yani $z_i = f_i(x_1, x_2, \dots, x_n)$, $i=1,2,\dots,m$ ise devreye kombinezonsal (statik, belleksiz) lojik devre denir.

Devrenin belirli bir andaki çıkışları önceki girişlere de bağlı ise devreye ardışıl (dinamik, bellekli) lojik devre denir. Bu durumda $z_i = f_i$ fonksiyonları, şimdiki, bir önceki, ..., p önceki girişlere bağlıdır ve karmaşıktır. İfadeleri basitleştirmek için durum kavramından yararlanılır. Durum, geçmişin geleceği etkileyen özüdür.

Buna göre, $z_i = f_i(x_1, x_2, \dots, x_n, s_j)$, $i=1,2,\dots,m$ dir. Burada, $x_1, x_2, \dots, x_n$ belirli bir andaki giriş, s_j de o andaki durumdur. Lojik devrelerde, durumu genelde flip-flop'ların (bellek elemanlarının) çıkışları oluşturur. Bu çıkışlar $y_1, y_2, \dots, y_s$ ise $z_i = f_i(x_1, x_2, \dots, x_n, y_1, y_2, \dots, y_s)$, $i=1,2,\dots,m$ olur.

Genel olarak, devre tasarımının birinci adımı, problemin sözle (grafik v.b. ile) tanımıdır. Klasik ardışıl lojik devre tasarım adımları da şöyledir:

- 1) Problemin sözle tanımlanması.
- 2) Durum diyagramının belirlenmesi.
- 3) Durum indirgeme.
- 4) Durum kodlama.
- 5) Devrenin belirlenmesi.

Genel olarak lojik devreler tasarlanırken devre ister ardışıl ister kombinezonsal olsun farklı tasarım adımlarında temel algoritmalar kullanılır. Örneğin örtü bulma problemi ardışıl devre tasarımının yukarıda gösterilen 3'üncü ve 4'üncü tasarım adımlarında kullanılabilir. Yani durum indirgeme tasarım adımı geliştirilen teknikler durum kodlama tasarım adımı da kullanılabilir. Bununla birlikte ardışıl devre tasarımında kullanılan bu teknikler kombinezonsal devre tasarımında veya görüntü

işleme gibi farklı alanlarda da kullanılabilir.

Bu Tez çalışması 5 ana kısımdan oluşmaktadır. Bu kısımlar aşağıda verilmiştir.

- 1) Kısmen belirli ardışıl makinelerde durum indirgeme için yeni bir yöntemin ve bu yönteme dayanan bir bilgisayar programının (SRC) geliştirilmesi.
- 2) Durum indirgeme için geliştirilen tekniklerin Boole fonksiyonlarının indirgenmesine uygulanması ve bu yönteme dayanan bir bilgisayar programının (MORP) geliştirilmesi.
- 3) Durum indirgeme için geliştirilen tekniklerin durum kodlama problemine uygulanması ve bu yönteme dayanan bir bilgisayar programının (OPASKOD) geliştirilmesi.
- 4) Tek koşullu bir örtü problemi için bir minimal örtüyü veya bütün minimal örtüleri elde eden bir yöntemin ve bu yönteme dayanan bir bilgisayar programının (UCPS) geliştirilmesi.
- 5) Minimal örtüdeki blok sayısının belirlenmesine ilişkin yeni bir yöntemin geliştirilmesi.

Bu Tez çalışmasında senkron/asenkron ardışıl devrelerde minimal durumlu eşdeğer devreyi veren yeni bir yöntem geliştirilmiştir. Bu yöntemde kullanılan teknikler Boole fonksiyonlarının indirgenmesine de uygulanmıştır. Bu teknikler tek koşullu (unate covering) veya iki koşullu (binate covering) örtme problemleri ile karşılaşılan diğer alanlarda da kullanılabilir.

Kısmen belirli ardışıl makineler için geliştirilen yöntem Bölüm 2’de verilmiştir. Bu bölümde verilen yönteme dayanan bir durum indirgeme programı olan SRC (State Reduction and Covering) geliştirilmiş ve MCNC FSM bençmarkları üzerinde test edilerek son yıllarda yayınlanan en önemli iki durum indirgeme algoritması Rho [1] ve Puri [2]’nin verdiği sonuçlarla karşılaştırılmıştır. Test sonuçları ilgili bölümlerde tablolarla verilmiştir. Bu tablolardan da görüleceği gibi SRC programı ile diğer yöntemlere göre çok hızlı indirgeme yapılabilir. Ayrıca diğer yöntemlerin indirgeyemediği bençmarklar da çok kısa sürede indirgenmiştir.

Örneğin Rho ile karşılaştırma yapıldığında tüm bençmarklarda SRC’nin çok daha hızlı indirgeme yapabildiği gözlenmiş, Rho yöntemi ile indirgenemeyen ex3 ve mark1 FSM’lerinin SRC tarafından saniyenin yüzde birinden daha kısa sürede indirgendiği gözlenmiştir. Ayrıca indirgenmesi zor bir bençmark olarak bilinen ex2 ile test edildiğinde Rho yöntemi “time-out” ile sonuçlanmaktadır. Bu bençmark için

minimal makine SRC tarafından 102s.'de elde edilmiştir.

Puri ile karşılaştırıldığında SRC'nin yine tüm bençmarklarda daha hızlı indirgeme yaptığı gözlenmiştir; ex2, Puri yöntemi ile test edilmemiştir. Bu bençmark FSM için C_a sınıfındaki blok sayısı 1366'dır. Puri yöntemi C_a sınıfını kullanarak indirgeme yaptığı için, blok sayısı çok fazla olan bu bençmarkta zorlanacağı açıktır.

Bölüm 3' de verilen yöntem n değişkenli m tane Boole fonksiyonunu birlikte indirgeyerek optimal sonucu vermektedir. Bu yöntemden yararlanılarak MORP geliştirilmiştir. Bu program University of California Berkeley'de geliştirilen ESPRESSO [3] programları ile sonuç bölümünde karşılaştırılmıştır.

Durum indirgeme için geliştirilen algoritmalar durum kodlama konusuna da uygulanmış ve OPASKOD adı verilen bir bilgisayar programı geliştirilmiştir. OPASKOD durum kodlama bençmarkları ile test edilmiş ve minimum geçiş zamanlı kodlamalar kısa sürede elde edilmiştir. Test sonuçları Bölüm 4'te verilmiştir.

Bu Tezde genel örtü problemlerini çözmek için geliştirilmiş olan yöntem ve bu yönteme dayalı bilgisayar programı UCPS (Unate Covering Problem Solver) ile, tamamen keyfi olarak oluşturulan 1000x1000 boyutlarında bir matrisle ifade edilen bir örtü problemine ilişkin minimal çözüm saniyeler mertebesinde bir sürede belirlenebilmektedir. Program, Boole fonksiyonlarının VE-VEYA devreleri ile minimal gerçekleştirilmesine ilişkin çeşitli bençmarklarda denenmiş ve çözümler kısa sürede belirlenmiştir. Geliştirilen program MCNC bençmarklarından elde edilen örtü matrislerine uygulanmıştır. Test sonuçları 5'inci bölümde verilmiştir. UCPS programı kullanılarak, verilen tek koşullu bir örtü problemi için, bir minimal örtü veya tüm minimal örtüler elde edilebilmektedir. Bu Tezde ayrıca toplamlar çarpımı biçimindeki bir Petrick fonksiyonunu çarpımlar toplamı biçiminde açarak bir minimal örtüyü veya tüm minimal örtüleri veren bir bilgisayar programı PETS (Petrick Solver) geliştirilmiştir.

Bölüm 6' da bir örtü problemini çözmeden minimal örtünün blok sayısını belirleyen bir yöntem üzerinde çalışılmıştır. Bu tür bir çalışmaya literatürde rastlanmamıştır.

Bugüne kadar minimal bir örtü bulmak için geliştirilen yöntemlerden bazıları minimal örtünün alt sınırı olarak maksimal uyumsuzlar sınıfı $\overline{C_s}$ 'nin en büyük bloğu M_2 'deki eleman sayısı olan m_2 'yi kullanmıştır [1, 2, 3, 4]. Bu alt sınır pek çok örtü probleminde minimal veya minime yakın örtüdeki blok sayısını vermektedir. Ancak bu alt sınırın hangi durumlarda minimal örtüdeki blok sayısını verdiği, hangi durumlarda vermediği, eğer vermiyorsa minimal örtüdeki blok sayısına ne kadar uzak olduğu gibi sorulardan literatürde bahsedilmemiştir. Bu Tez çalışmasında

ortaya atılan bu konular Bölüm 6’da incelenmiştir.

Uyumlular veya uyumsuzlar grafindan beşgen, yedigen, dokuzgenleri bulan etkili bir algoritma geliştirildiği taktirde örtü problemlerinin çözümünde önemli bir adım atılmış olacaktır.



2. KISMEN BELİRLİ ARDIŞIL MAKİNELER İÇİN YENİ BİR DURUM İNDİRGEME YÖNTEMİ

Durum indirgeme senkron ve asenkron ardışıl devre tasarımıdaki önemli adımlardan biridir. Durum indirgeme için geliştirilen algoritmalar durum kodlama, örtü bulma gibi lojik devre tasarımıdaki diğer adımlarda da kullanılabilir. Bu çalışmada geliştirilen teknikler de genel tekniklerdir ve tek koşullu örtme (unate covering) veya iki koşullu örtme (binate covering) problemleri ile karşılaşılan diğer alanlarda da kullanılabilir.

Bu kısımda kısmen belirli ardışıl makineler için geliştirilen yöntem verilmiştir. Bu yöntemle dayanarak geliştirilen durum indirgeme programı SRC (State Reduction and Covering) MCNC FSM benchmarkları üzerinde test edilmiş, elde edilen sonuçlar Rho [1] ve Puri [2]'nin verdiği sonuçlarla karşılaştırılmıştır. Test sonuçları sonuç kısmında verilmiştir. Bu tablodan da görüleceği gibi SRC programı ile diğer yöntemlere göre çok hızlı indirgeme yapılabilmektedir. Ayrıca diğer yöntemlerin indirgeyemediği benchmarklar da çok kısa sürede indirgenmiştir.

2.1 Tanımlar

Bu bölümde, yöntemde kullanılan tanımlamalar verilmiştir.

Tanım 2.1: Bir ardışıl makinede $g(x, d_i)$, yani d_i durumunun 1-izleyicisi veya $f(x, d_i)$ çıkışı tanımlanmamış olabilir. Böyle bir makineye kısmen belirli ardışıl makine denir.

Tanım 2.2: Makine d_i durumunda iken k uzunluklu $I_k = x_1, x_2, \dots, x_k$ giriş dizisi uygulandığında buna karşı düşen $g(I_k, d_i)$ durum dizisi tanımlı ise I_k , d_i 'nin bir uygulanabilir giriş dizisidir denir; buna göre 1-uzunluklu her giriş dizisi uygulanabilir bir giriş dizisidir; d_i 'nin uygulanabilir tüm giriş dizilerinin oluşturduğu küme A_i ile gösterilecektir.

		x	
		0	1
d	A	B,1	-, -
	B	-, 0	C, 0
	C	A, 1	B, 0

Şekil 2.1 Kısmen belirli bir ardışıl makine

Şekil 2.1'deki makine için $g(101, B) = C, A, -$ olduğuna göre $I_3 = 101$ B'nin bir uygulanabilir giriş dizisidir. Şekil 2.1'den, $f(101, B) = 0, 1, -$ olduğu görülmektedir. Öte yandan $g(10, C)$ dizisi tanımlı olmadığından $I_3 = 100$ dizisi C'nin uygulanabilir bir giriş dizisi değildir.

Tanım 2.3: d_i ve d_j durumlarının uygulanabilir giriş dizilerinin oluşturduğu kümelerin arakesiti A_{ij} olsun. Her $I \in A_{ij}$ için $f(I, d_i)$ ve $f(I, d_j)$ çıkış dizilerinin tanımlı olan karşılıklı elemanları eşit ise d_i ve d_j durumları uyumludur denir ve $d_i \cong d_j$ yazılır; karşılıklı elemanları tanımlı ise eşit olan $f(I, d_i)$ ve $f(I, d_j)$ dizilerine uyumlu dizi çifti denilecek ve $f(I, d_i) \cong f(I, d_j)$ yazılacaktır.

$$f(I_4, d_i) = 1010$$

$$f(I_4, d_j) = 10-0$$

$$f(I_4, d_k) = 1-00$$

olsun. Tanım 2 gereğince $d_i \cong d_j$ ve $d_j \cong d_k$ olabilir fakat $d_i \not\cong d_k$ dır. Yani, uyumluluk bağıntısı geçişlilik özelliğine sahip değildir.

Tanım 2.4: Her $I_k \in A_{ij}$ için $f(I_k, d_i)$ ve $f(I_k, d_j)$ çıkış dizileri uyumlu ise d_i ve d_j k-uyumludur denir ve $d_i \overset{k}{\cong} d_j$ yazılır.

Tanım 2.5: $C \subseteq D$ olan C kümesindeki tüm durum çiftleri uyumlu ise C'ye bir uyumlular sınıfı denir; bu sınıfların oluşturduğu küme C_u ile gösterilecektir.

Tanım 2.6: C_i bir uyumlular sınıfı ve $C_i \neq C_j$ olmak üzere $C_i \subseteq C_j$ olan bir C_j uyumlular sınıfı yoksa C_i 'ye bir maksimal uyumlular sınıfı denir. Maksimal uyumlular sınıflarının oluşturduğu kümeye son sınıf denilecek ve C_s ile gösterilecektir.

Tanım 2.7: C_u uyumlular sınıfının bazı elemanlarının oluşturduğu $C_0 = \{C_1, C_2, \dots, C_k\}$ kümesi,

(i) $C_1 \cup C_2 \cup \dots \cup C_k = D$ koşulunu sağlıyorsa C_0 'ye D durum kümesinin bir örtüsü denir.

(ii) D 'nin, blok sayısı k dan az olan bir uyumlular örtüsü yoksa C_0 'ye D nin bir minimal örtüsü denir.

(iii) C_0 'nün her bloğunun her 1-izleyicisi C_0 'nün bir bloğu tarafından içeriliyorsa C_0 'ye D nin kapalı bir örtüsü denir.

Tanım 2.8: C_u 'nun bir elemanının öz-izleyicileri: C_u 'nun bir C_i bloğunun tüm 1-izleyici bloklarının oluşturduğu küme göz önüne alınsın. Bu küme içinden durum sayısı 1 olan ve C_i tarafından içerilen bloklar çıkarılarak elde edilen kümeye, C_i 'nin öz-izleyiciler kümesi denir ve $I(C_i)$ ile gösterilir; $I(C_i)$ 'de $C_j \subseteq C_k$ olan bir C_j varsa C_j de atılır.

Tanım 2.9: Asal uyumlular sınıfı C_a : Aşağıdaki biçimde oluşturulan C_a kümesine asal uyumlular sınıfı denir. (i) C_s nin her elemanı C_a nın da elemanıdır. (ii) C_s nin bir C_i bloğunun bir alt-bloğu C_{ia} olsun.

$I(C_{ia}) \supseteq I(C_i)$ değilse C_{ia} , C_a nın bir elemanıdır.

C_s 'nin blokları ve bu blokların alt blokları C_u 'yu oluşturduğuna göre, C_a 'da bulunmayan bir C_u bloğu C_d , C_s 'nin bir C_i bloğunun alt bloğudur; öyle ki $I(C_d) \supseteq I(C_i)$ dir.

$C_i \neq \emptyset$ ve $I(C_d) \supseteq I(C_i)$ ve $I(C_d) \supseteq I(C_j)$ ise; yani bir C_d bloğunu eleyen birden çok C_i , $C_j, \dots$ varsa ve C_d bir bloğun izleyicisi olarak görülüyorsa, o takdirde C_d bloğu C_a 'dan elenmeyebilir.

2.2 Diğer Durum İndirgeme Yöntemleri

Ardışıl devrelerde durum indirgeme için çok sayıda yöntem geliştirilmiştir. Bunlar arasında Rho [1], Puri [2], Gimpel [5], Graselli [6] en çok bilinen indirgeme yöntemleridir.

Rho yönteminde uyumlu çiftlerden yararlanır. Bu yöntemde uyumlu çiftlerden oluşan kapalı yollar kümesi elde edilerek iki koşullu örtme problemi tek koşullu örtme problemine dönüştürülür ve blokları uyumlu çiftlerden oluşan bir eşdeğer makine elde edilir. Sonra da bu bloklar minimum sayıda C_a bloğuna sıkıştırılarak minimal makine elde edilir. Rho yönteminin diğer yöntemlere göre en önemli üstünlüğü C_a asal uyumlular sınıfının hesaplanmamasıdır; fakat Rho yöntemin bu

üstünlüğünün yanısıra çok önemli sakıncaları bulunmaktadır. Bu sakıncalar şunlardır:

- (i) Uyumlu çiftlerden oluşan bir yolun maliyetinin hesaplanması (uyumlu çiftleri örten minimum sayıdaki kapalı C_a bloklarının sayısı) zaman alıcı bir işlemdir. Üstelik bu işlemin her örtü kombinezonunda yapılması gerekmektedir.
- (ii) Maliyet hesabı açık ve basit değildir. Uyumlu çiftlerden oluşan bir kapalı yoldaki çiftleri örten bloklar yolda bulunmayan bazı uyumlu durum çiftlerini de içerebilir. Bu sorun nedeni ile maliyet hesabı sonucunda elde edilen C_a bloklarının oluşturduğu küme kapalı olmayabilir. Bu sorun yöntemin verildiği makaledeki testlerde bir bençmarkta ortaya çıkmıştır. Bu bençmark için yöntem kapalı olmayan bir örtü vermiştir; yani, bu bençmark için minimal makine elde edilememiştir.
- (iii) Uyumlu çiftlerden oluşan iki kapalı yolun birleştirilmesi ile oluşan yeni kümenin maliyeti, yeni kümeyi oluşturan iki kapalı yolun maliyetlerinin toplamına çoğu zaman eşit değildir. Yani kapalı yolların maliyetlerinin toplanabilmesi imkanı yoktur. Bu durumda her kombinezona karşı düşen her yeni kümede yeni maliyet hesaplarının yapılması gereklidir. Bu işlemler de indirgeme zamanını çok arttırabilir.
- (iv) Uyumlu çiftlerden oluşan kapalı yolların birleştirilmesi ile oluşan kümeler kapalı olmayabilir. Ancak bu yöntemin en önemli özelliği kapalı yollarla işlem yapılması, böylece kapalılık kontrolünün ayrıca yapılmamasıdır. Bu sorun yöntemin en önemli üstünlüğünü bir ölçüde kaybetmesine ve kapalılık kontrolünün yapılmasına neden olmaktadır.

Puri yönteminde ise asal uyumlular sınıfı C_a 'dan yararlanılır. Bu yöntemin sakıncaları şunlardır:

- (i) Asal uyumlular sınıfını hesaplama ve bu sınıf ile minimal kapalı örtü arama, işlem süresini önemli ölçüde arttırılabilir.
- (ii) Bu yöntemin en önemli sakıncası minimal kapalı örtünün C_a 'nın bloklarından elde edilmeye çalışılmasıdır.
- (iii) Minimal kapalı örtü arama algoritması açık ve basit değildir.

- (iv) Bu yöntemde oluşturulan arama ağacının derinliği dolayısıyla arama işleminin süresi çok artabilir. Bunun için bir önlem alınmamıştır.
- (v) Bu yöntemde etkili bir üst sınır kullanılmamıştır.
- (vi) Özellikle çok sayıda C_a bloğu içeren bençmarklarda yöntemin işlem süresi çok artabilir. Bu yöntemin verildiği makalede [2] çok sayıda C_a bloğu içeren ex2 bençmarkı için test sonucu verilmemiştir.

Yukarıda verilen sakıncalara karşılık bu yöntemde maliyet hesabı çok basittir. Maliyet, arama ağacındaki seviye sayısına; yani kullanılan C_a bloğu sayısına eşittir.

Tez çalışmasının bu bölümünde geliştirilen indirgeme yöntemi yukarıda Rho ve Puri yöntemleri için verilen toplam 10 sakıncanın sadece 1'ine sahiptir. Geliştirilen yöntemde C_a bloklarının hesabı yöntemin tek dezavantajını oluşturur. Ancak bu dezavantajın etkisi, minimal kapalı örtünün C_a bloklarından değil de C_a bloklarından oluşan kapalı kümelerden elde edildiği için önemli ölçüde azaltılmıştır.

2.3 Bu Çalışmada Geliştirilen İndirgeme Yöntemi

Bu Tez çalışmasında senkron ve asenkron ardışıl devrelerde durum indirgeme için yeni bir yöntem geliştirilmiştir. Bu yöntem üç algoritmadan oluşmaktadır. Bu algoritmalar aşağıda verilmiştir.

- a) **Algoritma 1:** Maksimal uyumlular sınıfı C_s 'nin ve C_s 'den bir minimal örtünün bulunması.
- b) **Algoritma 2:** Uyumlu Çiftlerden oluşan kapalı yolların (P_p) bulunması, C_u 'nun bloklarından oluşan kapalı yolların (P_u) bulunması, C_a 'nın bloklarından oluşan kapalı yolların (P_a) bulunması ve C_s 'nin bloklarından oluşan kapalı yolların (P_s) bulunması.
- c) **Algoritma 3:** P_u , P_a , P_s kapalı yollar kümelerinden yararlanarak minimal kapalı örtünün elde edilmesi.

İlk algoritmada C_s 'den elde edilen minimal örtüdeki blok sayısı minimal makinedeki durum sayısı için bir alt sınır oluşturur.

İkinci algoritmada P_s kümesinden elde edilen minime yakın kapalı örtüdeki blok sayısı minimal makinedeki durum sayısı için bir üst sınır oluşturur. Birçok durumda bu üst sınır minimal makinedeki durum sayısını vermektedir. Eğer alt sınır üst sınıra

eşit ise minimal makine C_s 'nin bloklarından elde edilmekte ve P_u , P_a kapalı yollar kümeleri oluşturulmamaktadır.

Üçüncü algoritmada bir önceki adımda elde edilen P_u , P_a , P_s kapalı yolları kullanılır. Bu kümelerin herhangi birinden yararlanarak minimal kapalı örtünün elde edilmesi mümkündür; fakat P_a kapalı yollar kümesi kullanılarak çok kısa sürede bir minimal kapalı örtü elde edilebilmektedir. Bu nedenle geliştirilen programda P_a kapalı yollar kümesi kullanılarak minimal kapalı örtü elde edilmektedir.

Bu Tezde kullanılan, C_a veya C_u blokları ile kapalı yollar elde etme kavramına literatürde rastlanmamıştır. Bu Tez çalışmasında P_a ve P_u kapalı yollar kümesi kavramı tanımlanmış ve bu kümelerden bir minimal kapalı örtü elde etmek için yeni bir yöntem geliştirilmiştir. P_a kapalı yollar kümesi kavramı, bunun dışında, başka alanlarda da kullanılabilir. Örneğin, C_a bloklarını eleyerek, blok sayısını azaltmak için bu kavramdan yararlanılabilir. Kapalı yollar kümesi P_a 'daki bir kapalı yolun örtü oluşturması ve bu yola ilişkin alt sınırın üst sınıra eşit veya büyük olması halinde bu kapalı yol elenebilir. Elenen bir kapalı yolu oluşturmak için kullanılan C_a bloğu da C_a kümesinden atılabilir.

2.4 Algoritmanın Adımları

Geliştirilen algoritmayı oluşturan adımlar aşağıda verilmiştir:

Adım 1: Verilen ardışıl devreye ilişkin çiftler tablosundan yararlanarak uyumlu ve uyumsuz durum çiftleri belirle. Eğer bu işlem sonucunda uyumlu durum çifti elde edilemezse verilen ardışıl makine minimaldir. Eğer uyumsuz durum çifti elde edilemezse verilen makinedeki bütün durumlar eşdeğerdir ve minimal makinedeki durum sayısı 1'dir; yani karşı düşen devre kombinezonsaldır.

Adım 2: Uyumlu ve uyumsuz durum çiftlerinden yararlanarak maksimal uyumlular sınıfı C_s 'yi ve Maksimal uyumsuzlar sınıfı $\overline{C_s}$ 'yi elde et. Bunun için geliştirilen yöntem ayrıt 2.6' da örnekle açıklanmıştır.

Adım 3: Maksimal uyumsuzlar sınıfının en büyük bloğu M_2 'deki durum sayısı m_2 'yi bir alt sınır olarak al. C_s sınıfından ağaç yapısını kullanarak bir minimal örtü elde et. Minimal örtüdeki blok sayısını alt sınır olarak al. Bu işlemler sonucunda minimal makinedeki durum sayısı için bir alt sınır (LB) elde edilir. Bu adımda ayrıca durum sayısı ve C_s 'deki blok sayısından küçük olanı ile bir üst sınır (UB) elde edilir. Yani üst sınır $UB = \min\{n, |C_s|\}$ 'dir.

Eğer alt sınır üst sınıra eşit ise minimal makine elde edilmiş olur. Bu durumda üst sınır durum sayısı n 'ye eşit ise minimal makinedeki durum sayısı n 'dir ve minimal makine n tane durumdan oluşur. Eğer alt sınır üst sınıra eşit ve üst sınır C_s 'deki blok sayısına eşit ise minimal makine C_s 'nin bloklarından oluşur.

Adım 4: Rho yönteminde verilen algoritmaya benzer şekilde bir maksimal uyumlular grafi oluştur. Bu graftaki her düğüm bir uyumlu durum çifti yerine C_s kümesinin bir bloğudur. Bu graftan, Tarjan [7] algoritması kullanarak kuvvetli bağlı bileşenler kümesini (SCC) oluştur. Bir bloğun belirli bir giriş için birden çok izleyicisi varsa, ilk karşılaşılan blok izleyici olarak seçilmektedir. Bu nedenle C_s 'den elde edilen kapalı örtü, bir bloğun belirli bir giriş için birden çok izleyicisi yoksa minimal, aksi halde minimal veya minime yakındır.

Adım 5: Kuvvetli bağlı bileşenler algoritmasını kullanarak C_s bloklarından oluşan kapalı yollar kümesini elde et. Bu kümenin elde edilmesi ile minimal kapalı örtü bulma problemi minimal örtü bulma problemine indirgenmiş olur. Bu tez çalışmasında geliştirilen bir örtü bulma algoritması ile minimal örtü çok kısa sürede elde edilir.

Adım 6: Adım 5'de elde edilen minime yakın veya minimal kapalı örtüdeki blok sayısından üst sınırı (UB) elde et. Eğer alt sınır üst sınıra eşit ise minimal makine elde edilmiş olur. Bu durumda Adım 5'de elde edilen minimal kapalı örtünün blokları minimal makinenin bloklarıdır.

Adım 6 sonunda minimal makine elde edilemezse P_u , P_a kapalı yollar kümelerinden herhangi biri kullanılarak minimal makine oluşturulabilir.

Adım 7: P_p kapalı yollar kümesini elde etmek için düğümleri uyumlu çiftler olan uyumlu çiftler grafi oluşturulur. Benzer şekilde P_u kapalı yollar kümesini elde etmek için düğümleri C_u 'nın blokları olan uyumlular grafi, P_a kapalı yollar kümesini elde etmek için düğümleri C_a 'nın blokları olan asal uyumlular grafi oluşturulur.

Bu adımda elde edilen graflar ve Tarjan [7] algoritması kullanılarak P_u , P_a kapalı yollar kümeleri oluşturulur. Bu kümeler, Tezde geliştirilmiş olan bir algoritma ile hızlı bir şekilde oluşturulmaktadır.

Elde edilen kapalı yollar eleme işleminden geçirilir. Eleme işleminde bir kapalı yolun belirlediği alt sınır LB hesaplanır. Eğer $LB \geq UB$ ise bu kapalı yollar elenir. Eleme işlemi aşağıda verilen 4 yol ile yapılabilir.

- 1) Kapalı yolu oluşturan blokların sayısının kullanılması ($\|LB\|_0$)
- 2) Kapalı yoldaki blokların C_a blokları kullanılarak sıkıştırılması, bu yol ile içerilen bloklar atılarak kapalı yoldaki blok sayısının azaltılması ($\|LB\|_1$)
- 3) Kapalı yoldaki blokların C_s blokları ile örtülecek şekilde bir minimal örtünün bulunması ($\|LB\|_2$)
- 4) Kapalı yola, yolun örtmediği durumlar eklenerek elde edilen kapalı yol bloklarının, C_s blokları ile örtülecek şekilde bir minimal örtünün bulunması ($\|LB\|_3$)

Bu Tezde 2'nci eleme yöntemi ile eleme işleminden geçirilen kapalı yollar, algoritmanın bir sonraki adımında kullanılarak (Adım 8), bir üst sınır (UB) elde edilir. 4'üncü eleme yönteminden geçirilen kapalı yollar algoritmanın bir sonraki adımında kullanılarak (Adım 8), bir minimal makine elde edilir.

Adım 8: Bir s_i durumunu içeren kapalı yolların oluşturduğu küme S_i olsun. Bütün durumlar için S_i ($i=1,2,\dots,n$) kümelerini elde et. Her bir S_i kümesinden bir kapalı yol olarak oluşturulan örtünün bir minimal makine veya bir üst sınır oluşturup oluşturmadığını kontrol et. Üst sınır ve alt sınır kullanılarak pek çok kombinezon elenebilir ve böylece Algoritma 3 oldukça hızlandırılabilir. Bu adım P_p kapalı yollar kümesinin kullanılması durumunda çok uzun sürebilir. Bunun iki ana nedeni vardır:

- (i) Uyumlu çiftlerden oluşan bir kapalı yola karşı düşen minimal makinedeki blok sayısının belirlenmesi uzun süren bir işlemdir.
- (ii) Kapalı yollardan oluşan bir kümeye yeni bir kapalı yol eklendiğinde toplam kapalı yol maliyetinin belirlenmesi uzun süren bir işlemdir.

Bu adımda P_u kapalı yollar kümesinin kullanılması durumunda yukarıdaki iki zorlukla karşılaşılmadığından işlem süresi oldukça kısalmır. Bununla birlikte bu adımda P_a kümesinin kullanılması ile işlem süresi daha da azaltılabilir. Bunun nedeni C_a kümesindeki blok sayısının C_u kümesindeki blok sayısından az olması ve C_u kümesinden küçük blokların atılarak C_a 'nın elde edilmiş olmasıdır. Bu nedenlerden dolayı geliştirilen programda minimal makine P_a kapalı yollar kümesinden elde edilmektedir.

Bu yöntemin akış tablosu ek olarak verilmiştir.

2.5 Önerilen Yöntemin Diğer Yöntemlerden Temel Farklılıkları

Bu kısımda önerilen durum indirgeme yönteminin, Rho ve Puri yöntemlerinden en önemli farklarından bir tanesi, çok hızlı bir şekilde minimale yakın bir makinenin elde edilmesidir. Bu sayede, önerilen yöntem ile hem çok kısa sürede bir minimale yakın makine elde edilmekte, hem de iyi bir üst sınır elde edilmektedir. Bu üst sınır kullanılarak minimal makine de hızlı bir şekilde elde edilebilmektedir.

Önerilen yöntemin literatürdeki diğer yöntemlerden temel farklılıklarından ikincisi ise asal uyumlular sınıfından kapalı yolların elde edilmesidir. Asal uyumlular sınıfından kapalı yollar elde edildiğinde, bu yollardan bir örtü oluşturanlar ile yeni bir üst sınır elde edilebilmektedir. Ayrıca asal uyumlu bloklardan oluşan kapalı yolların minimum kaç asal uyumlu blok ile kapsanacağı ve birden fazla kapalı yolun birleştirilmesi ile elde edilen yeni kapalı yolun minimum kaç asal uyumlu blok is kapsanacağının hesabı hızla yapılabilmektedir.

Önerilen yöntemde, elde edilen bir minimal makine adayının kapalılık kontrolünün yapılmaması da yöntemin temel farklılıklarından bir tanesidir. Bu sayede, minimal makineyi elde etmek için ağaç yapısı ile elde edilen kombinezonlarda kapalılık kontrolü yapılmayarak indirgeme işlemi hızlandırılmış olur.

Önerilen yöntemde alt ve üst sınırların etkin bir şekilde kullanılması, bu sayede minimal örtü ağacının budanması, maksimal uyumlu blokların hızla elde edilebilmesi ve bu bloklardan minimale yakın kapalı örtütün hızlı bir şekilde elde edilebilmesi, minimal kapalı örtü bulma ağacı kullanımı diğer temel farklılıkları oluşturur.

Literatürde verilen yöntemlerin kısmen belirli ardışıl makinelerde durum indirgeme için harcadıkları süreler uyumlu durum çifti sayısına, maksimal uyumlu blokların sayısına ve özellikle asal uyumlu blokların sayısına çok bağlıdır. Örneğin Puri yöntemi ile, asal uyumlu blok sayısı birkaç yüz mertebesindeki makinelerde dahi minimal makine elde edilemeyebilir. Rho yönteminde de benzer durum söz konusudur. Rho yönteminde ayrıca uyumlu durum çiftlerinin izleyicilerinin sayısı da indirgeme süresini çok arttırabilmektedir. Önerilen yöntemde, asal uyumlu bloklardan kuvvetli bağlı bileşenler ve bu bileşenlerden kapalı yollar elde edilerek asal uyumlu blok sayısının yöntemde harcanan süredeki etkisi oldukça azaltılmaktadır. Bu sayede, diğer yöntemler ile 200-300 mertebesindeki asal uyumlu blok oluşturan makinelere ilişkin minimal makine elde edilemezken, önerilen yöntemde 1366 asal uyumlu bloğu bulunan ex2 benchmarkına ilişkin minimal makine hızlı bir şekilde elde edilmiştir.

2.6 İndirgeme Örneği 1

İndirgenecek olan sonlu durum makinesine ilişkin durum tablosu Şekil 2.2’de verilmiştir. Bu makinenin 6 durumu, 2 giriş değişkeni ve 1 çıkış değişkeni vardır.

		Girişler			
		00	01	11	10
Durumlar	1	3,0	1,-	-, -	-, -
	2	6,0	2,0	1,-	-, -
	3	-, 1	-, -	4,0	-, -
	4	1,0	-, -	-, -	5,1
	5	-, -	5,-	2,1	1,1
	6	-, -	2,1	6,-	4,1

Şekil 2.2 M_1 makinesine ilişkin durum tablosu

M_1 makinesine ilişkin çiftler tablosu oluşturulduğunda makinenin uyumlu ve uyumsuz çiftleri Şekil 2.3’de gösterildiği gibi elde edilir. Bu şekilde ‘X’ işaretine karşı düşen satır ve sütundaki durum çiftleri uyumsuz, ‘√’ işaretine karşı düşen satır ve sütundaki durum çiftleri uyumludur.

2	√				
3	X	X			
4	X	√	X		
5	√	√	X	√	
6	√	X	√	√	X
	1	2	3	4	5

Şekil 2.3 M_1 makinesine ilişkin çiftler tablosu

Yukarıda gösterilen çiftler tablosundan yararlanarak maksimal uyumlular sınıfı hesaplanmaktadır. Maksimal uyumlu sınıfların hesaplanması için bu Tezde geliştirilen algorithmada önce bütün durumları içeren bir blok yazılır. Çiftler

Aşağıda Şekil 2.4’de çiftler tablosundan yararlanılarak durumların ağırlıklarının hesaplanması gösterilmiştir; bir durumun, uyumsuz olduğu durumların sayısına, o durumun ağırlığı denilecektir.

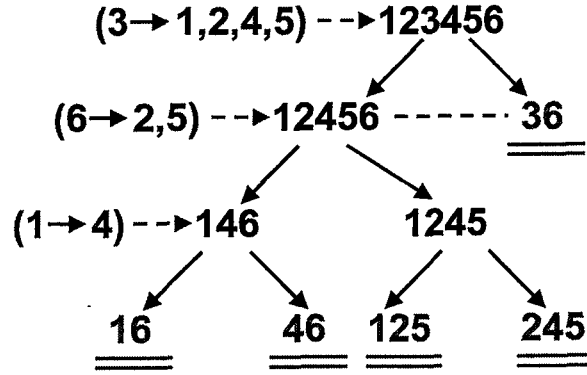
0	2	✓				
2	3	X	X			
2	4	X	✓	X		
1	5	✓	✓	X	✓	
2	6	✓	X	✓	✓	X
		1	2	3	4	5
		2	2	2	0	2

Durumlar	1	2	3	4	5	6
Ağırlık	2	2	4	2	2	2

Bu şekilden görüldüğü gibi 3 durumunun ağırlığı en büyüktür. Bu nedenle tüm durumları içeren bloktan dallanma işlemine 3 durumundan başlanılmalıdır. Dallanma işlemi ile maksimal uyumlu blokların hesabı Şekil 2.5’de gösterilmiştir.

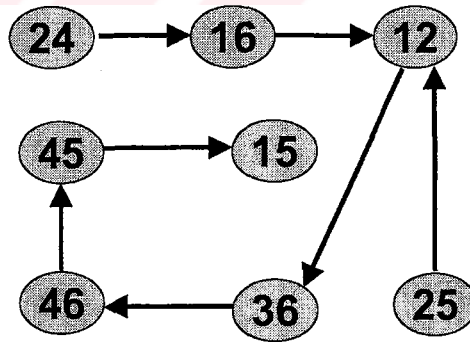
M_1 makinesine ilişkin uyumlu çiftler grafi Şekil 2.6'da gösterilmiştir. Buna benzer

şekilde düğümleri maksimal uyumlular olan maksimal uyumlular grafi Şekil 2.7’de gösterilmiştir; bir bloğun belirli bir giriş için birden çok izleyicisi varsa, sadece ilk izleyici alınmıştır. Tarjan algoritması ile maksimal uyumlular grafinin kuvvetli bağlı bileşenleri hesaplandığında, Şekil 2.7’de gösterilen tek düğümlü graf elde edilir. Buna göre maksimal uyumlu blokların tamamı bir kapalı yol oluşturmaktadır.



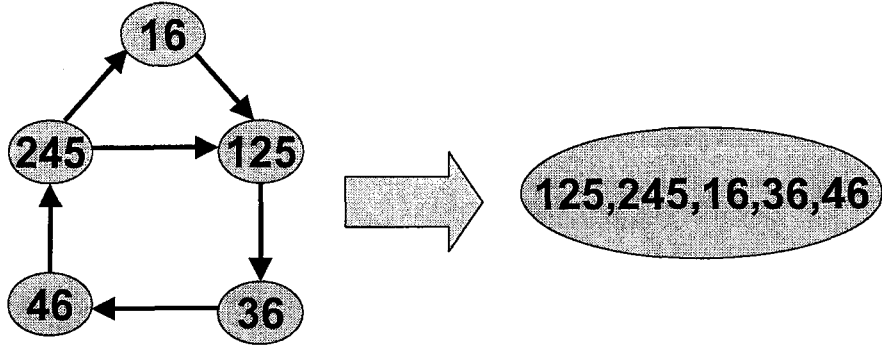
Şekil 2.5 M_1 makinesine ilişkin maksimal uyumluların hesabı

C_s sınıfından elde edilen kuvvetli bağlı bileşenler grafi sadece bir düğüm içerdiğinden, C_s sınıfından bir minimal makine kolayca elde edilmiş olur; bu makine C_s 'nin bloklarından oluşur. Bu durumda minimal makinedeki blok sayısının üst sınırı C_s sınıfındaki maksimal uyumlu blok sayısına eşit olduğundan, üst sınır 5 olarak elde edilir.



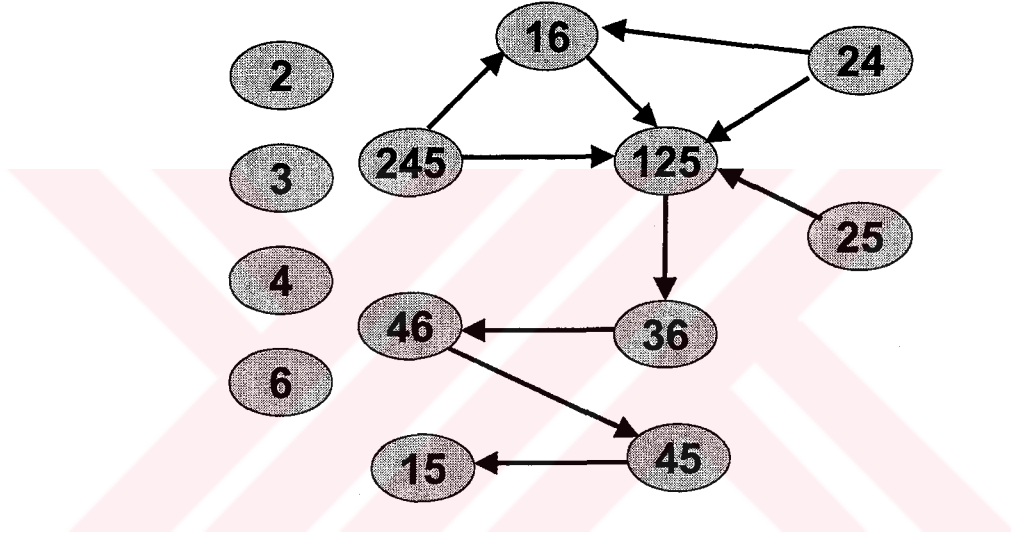
Şekil 2.6 M_1 makinesine ilişkin uyumlu çiftler grafi

Şekil 2.8’de düğümleri C_a sınıfının blokları olan graf gösterilmiştir. Bu grafın kuvvetli bağlı bileşenleri hesaplandığında, graftaki her bir düğümün aynı zamanda kuvvetli bağlı bileşen olduğu belirlenir. Bu durumda kuvvetli bağlı bileşenler grafi asal uyumlular grafi ile aynı graftır.



Şekil 2.7 M_1 makinesine ilişkin maksimal uyumlular grafi

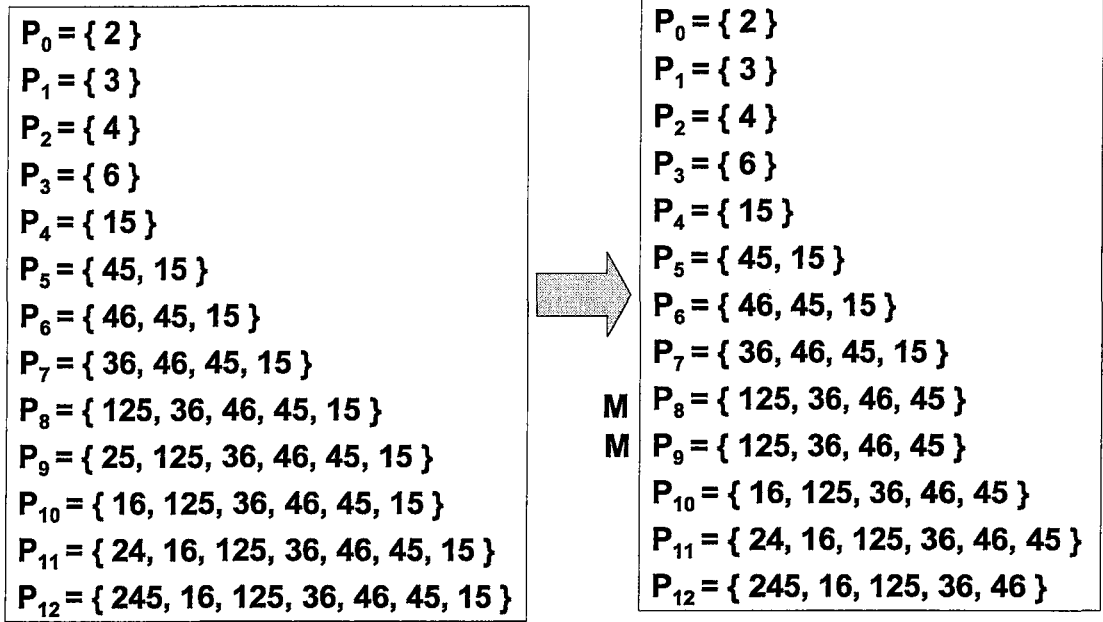
Şekil 2.8’de verilen asal uyumlular grafindan yararlanarak kapalı yollar hesaplanır. Elde edilen kapalı yollar Şekil 2.9’da verilmiştir.



Şekil 2.8 M_1 makinesine ilişkin asal uyumlular grafi

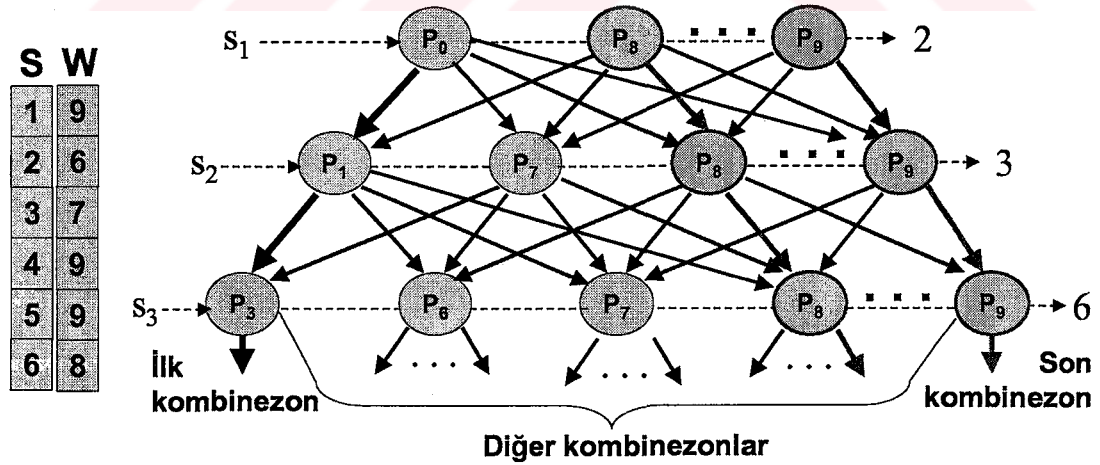
Şekil 2.9’dan da görüldüğü gibi kapalı yol sayısı 12’dir. Bu şekilde sol tarafta verilen kapalı yolların içerdiği fazlalık blokların elenmesi ile elde edilen kapalı yollar sağ tarafta verilmiştir.

Şekil 2.9’da verilen kapalı yollardan yararlanarak minimal kapalı örtü arama ağacı elde edilir. Bunun için önce durumların ağırlıkları hesaplanır. Bir durumun ağırlığı, kendisini içeren kapalı yolların sayısıdır. Şekil 2.10’da durumların ağırlıkları gösterilmiştir. Bu şekilden de görüldüğü gibi 2 durumunun ağırlığı en küçüktür. Bu durum 6 tane kapalı yol tarafından içerilmektedir. Makinenin durumları ağırlıklarına göre küçükten büyüğe doğru sıralandığında minimal kapalı örtü ağacının seviyeleri elde edilir. Buna göre ağacın ilk 3 seviyesini Şekil 2.10’da görüldüğü gibi 2, 3 ve 6 durumları oluşturur.



Şekil 2.9 M_1 makinesine ilişkin kapalı yollar

Minimal kapalı örtü arama ağacının durum sayısı kadar seviye içermesine gerek yoktur. Eğer minimal makinenin üst sınırı UB ise, bu ağaç için UB-1 tane seviye oluşturulmalıdır. Eğer bu ağaçtan bir minimal kapalı örtü elde edilemez ise UB'ye karşı düşen makine minimaldir. Öte yandan minimal kapalı örtü elde etme ağacı en az minimal makinenin alt sınırı olan LB tane seviye içermelidir.



verilen ağaç yapısında bir kapalı yolun birden çok seviyede görünmesini engelleyerek yöntemi hızlandırmak için, i. seviyede s_i durumunu içeren kapalı yollar yerine, s_i durumunu kapalı yolu oluşturan C_a bloğunda içeren kapalı yollar yazılabilir.

2.7 İndirgeme Örneği 2

İndirgenmesi en zor MCNC bençmarklarından olan ex2'nin durum tablosu Şekil 2.11'de verilmiştir. Bu bençmarkın 19 durumu, 2 giriş değişkeni ve 2 çıkış değişkeni vardır.

	00	01	10	11
0	-, --	-, --	-, --	-, --
1	2, --	4, --	0, --	3, --
2	6, --	9, --	0, --	0, 11
3	0, --	0, --	7, --	8, --
4	2, --	1, --	6, --	5, --
5	0, --	0, --	0, --	6, --
6	1, 00	0, --	2, --	0, 11
7	5, 11	2, 00	0, --	0, --
8	5, --	0, --	0, --	1, 00
9	5, --	3, 11	0, --	0, --
10	11, --	13, --	0, --	12, --
11	15, --	18, --	0, --	0, --
12	0, --	0, --	16, --	17, --
13	11, --	10, 00	15, --	14, --
14	0, --	0, --	0, --	15, --
15	10, 00	0, --	11, --	0, 11
16	14, 11	11, --	0, --	0, --
17	14, --	0, --	0, --	10, 00
18	14, --	12, --	0, 11	0, --

Şekil 2.11 Bençmark ex2'nin durum tablosu

Bu durum tablosundan elde edilen çiftler tablosu kullanılarak maksimal uyumlular sınıfı C_s aşağıdaki gibi elde edilir:

$C_s = \{ [0,3,8,9,11,12,16,17,18], [0,3,7,8,11,12,16,17,18], [0,4,5,13,14,16,17, 18], [0,5,9,11,14,16,17,18], [0,5,7,11,14,16,17,18], [0,4,5,9,14,16,17,18], [0,4, 5,7,14,16,17,18], [0,4,5,8,13,16,17,18], [0,5,8,9,11,16,17,18], [0,5,7,8,11,16, 17,18], [0,4,5,8,9,16,17,18], [0,4,5,7,8,16,17,18], [0,2,5,9,11,14,16,18], [0,2,5, 7,11,14,16,18], [0,2,3,9,11,12,16,18], [0,2,3,7,11,12,16,18], [0,2,5,9,11,14, 15,18], [0,2,5,6,9,14,15,18], [0,2,3,9,11,12,15,18], [0,2,3,6,9,12,15,18], [0,1, 3,8,10,11,12,17], [0,5,6,13,14,15,18], [0,4,5,6,13,14,18], [0,4,5,6,9,14,18],$

[0,1,5,10,11,14,17], [0,1,5,8,10,11,17], [0,5,10,11,14,15], [0,5,6,10,14,15],
 [0,3,10,11,12,15], [0,3,6,10,12,15], [0,1,2,5,11,14], [0,1,5,6,10,14], [0,1,2,5, 6,14],
 [0,1,2,3,11,12], [0,1,3,6,10,12], [0,1,2,3,6,12] }

Çiftler tablosu kullanılarak maksimal uyumsuzlar sınıfı $\overline{C_s}$ aşağıdaki gibi elde edilir:

$\overline{C_s} = \{ [7,9,10,13], [1,7,9,13], [1,15,16], [1,7,15], [1,4,15], [2,10,13], [2,4,10], [10,18], [1,18], [15,17], [6,17], [2,17], [10,16], [6,16], [8,15], [12,14], [8,14], [3,14], [12,13], [11,13], [3,13], [5,12], [4,12], [6,11], [4,11], [6,8], [2,8], [6,7], [3,5], [3,4], [0] \}$

$\overline{C_s}$ 'nin en büyük bloğu [7,9,10,13]'dir. Bu nedenle $m_2=4$ ve $LB=4$ 'tür. C_s 'den elde edilen minimale yakın kapalı örtüdeki blok sayısı 17'dir. $LB \neq UB$ olduğundan minimal makine Adım 8 sonunda elde edilir.

C_a kümesindeki asal uyumlu sınıfı sayısı 1366'dır. Tarjan [7] algoritması kullanılarak 1314 kuvvetli bağlı bileşen elde edilir. Bu nedenle C_a sınıfından elde edilen kapalı yollar kümesi P_a 'da 1314 tane kapalı yol bulunur. Bu kapalı yollardan bazıları aşağıda gösterilmiştir:

$P_0 = \{ [0,9,18], [0,8,17], [0,7,16], [0,6,15], [0,5,14], [0,4,13], [0,3,12], [0,2,11], [0,1,10] \}$

$P_{32} = \{ [0,15] \}$

$P_{57} = \{ [0,4,5,9,14,16,17,18], [0,5,6,10,15], [0,2,5,11,14], [0,1,3,11,12], [0,4,13,18], [0,3,12,17], [0,8,10,17], [0,3,8,17], [0,2,11,15], [0,1,10,12], [0,1,10,11], [0,1,8,10], [0,3,18], [0,7,16], [0,2,16], [0,6,14], [0,6,12], [0,9,11] \}$

$P_{98} = \{ [0,2,3,9,11,12,16,18], [0,5,6,14,15], [0,8,17], [0,7,16], [0,4,13], [0,1,10] \}$

$P_{111} = \{ [0,12,18], [0,17], [0,16], [0,14] \}$

$P_{155} = \{ [0,7,8,11,16,17,18], [0,2,11,12,18], [0,5,6,14,15], [0,9,12,18], [0,4,13], [0,3,12], [0,1,10] \}$

$P_{256} = \{ [0,2,5,9,11,14,18], [0,3,9,12,18], [0,5,6,14,15], [0,8,17], [0,7,16], [0,4,13], [0,1,10] \}$

$P_{501} = \{ [0,2,3,7,11,18], [0,2,9,12,18], [0,5,6,14,15], [0,3,9,12], [0,8,17], [0,7,16], [0,4,13], [0,1,10] \}$

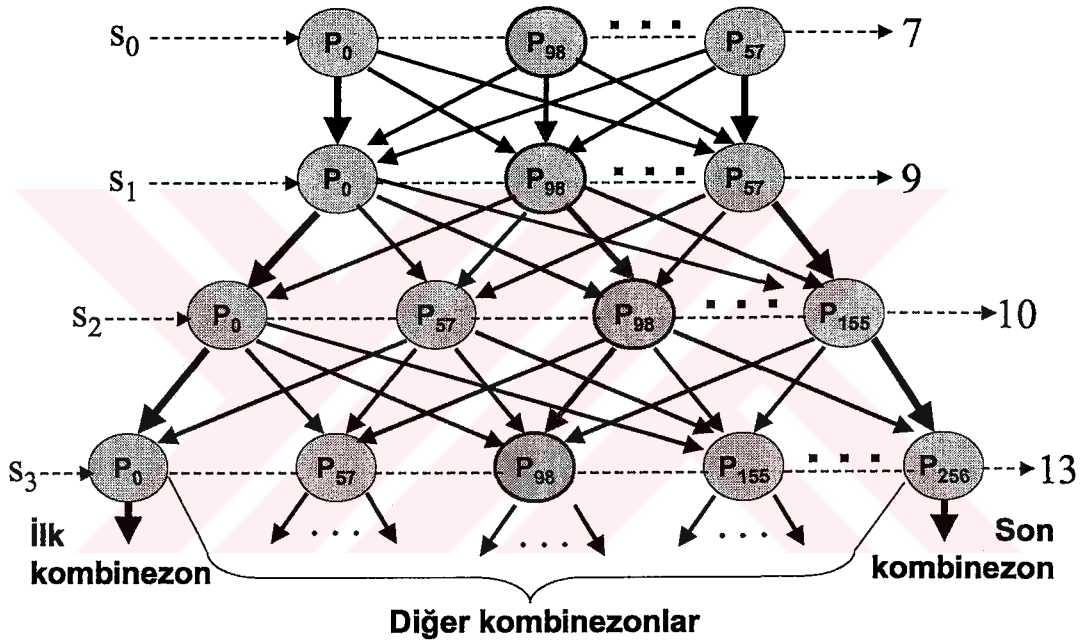
$P_{1042} = \{ [0,16,17,18], [0,11,12], [0,15], [0,10] \}$

$$P_{1252} = \{ [0,2,3,9], [0,5,6], [0,8], [0,7], [0,1] \}$$

$$P_{1313} = \{ [0,2,3], [0,9], [0,8], [0,7], [0,6] \}$$

Etkili budama (prunning) algoritmaları kullanılarak kapalı yol sayısı oldukça azaltılabilir. Bu tez çalışmasında geliştirilen budama algoritmaları ile ex2 için kapalı yol sayısı 1314'ten 34'e indirilmiştir.

Şekil 2.12'de kapalı yollardan yararlanılarak oluşturulan minimal kapalı örtü arama ağacı gösterilmiştir.



Şekil 2.12 Ex2 benchmarkı için minimal kapalı örtüyü arama ağacı

Şekil 2.12'deki kapalı yolların oluşturduğu kombinezonlardan minimal makineyi veren kapalı yol kombinezonu P_{98} 'dir. Bu kombinezona karşı düşen minimal makine aşağıda verilmiştir:

$$M = \{ [0,2,3,9,11,12,16,18], [0,7,8,16,17], [0,5,6,14,15], [0,4,13], [0,1,10] \}.$$

2.8 İndirgeme Örneği 3

MCNC benchmarklarından olan mark1'in standart KISS formatı kullanılarak durum geçiş tablosu aşağıda verilmiştir. Bu benchmarkın 15 durumu, 5 giriş değişkeni ve 16 çıkış değişkeni vardır. KISS formatında i; giriş değişkeni sayısını, o; çıkış değişkeni

sayısını, p; durum geçiş satırı sayısını, s; durum sayısını gösterir. Durum geçiş satırlarında sırası ile giriş, ilk durum, bir sonraki durum ve çıkış verilir. Örneğin ilk durum geçiş satırının anlamı şudur:

Makine 0 numaralı durumda iken 0---- girişi geldiğinde 1 durumuna geçmektedir ve çıkış -11---1-00----- olmaktadır. Burada '-' keyfi anlamı taşır, yani '-' 0 veya 1 olabilir.

```
.i 5
.o 16
.p 36
.s 15
0---- st0   st1   -11---1-00-----
0---- st1   st1   -11---1-00-----
0---- st2   st1   -11---1-00-----
0---- st3   st1   -11---1-00-----
0---- st4   st1   -11---1-00-----
0---- st5   st1   -11---1-00-----
0---- st6   st1   -11---1-00-----
0---- st7   st1   -11---1-00-----
0---- st8   st1   -11---1-00-----
0---- st9   st1   -11---1-00-----
0---- st10  st1   -11---1-00-----
0---- st11  st1   -11---1-00-----
0---- st12  st1   -11---1-00-----
0---- st13  st1   -11---1-00-----
0---- st14  st1   -11---1-00-----
1---- st1   st3   -11---1-00-----
1---- st2   st0   -11---1-00-----
1---- st3   st4   101---1-01-----
1-111 st4   st13  -11---1-00-----
1-110 st4   st10  -11---1-00-----
1-10- st4   st9   -11---1-00-----
1-011 st4   st8   -11---1-00-----
1-010 st4   st7   -11---1-00-----
1-001 st4   st6   -11---1-00-----
1-000 st4   st5   -11---1-00-----
1---- st5   st14  0011--1-00-----
1---- st6   st14  00100-0-00000011
1---- st7   st14  001---1100-----
1---- st8   st14  010---1-00-----
1---- st9   st14  001---1010000101
1---- st10  st11  -11---1-00100000
10--- st11  st13  -11---1-00-----
11--- st11  st12  -11---1-00-----
1---- st12  st13  -110110-00-----
1---- st13  st14  -11---1-00-----
1---- st14  st3   -110110-00-----
```

Bu ifade ile elde edilen çiftler tablosu kullanılarak maksimal uyumlular sınıfı C_s aşağıdaki gibi elde edilir:

$$C_s = \{ [0,1,2], [0,2,10], [0,2,11], [0,2,4], [0,5,7], [0,2,13], [0,9], [0,3], [0,14], [0,12], [0,6], [0,8] \}$$

Çiftler tablosu kullanılarak maksimal uyumsuzlar sınıfı $\overline{C_s}$ aşağıdaki gibi elde edilir:

$$\overline{C_s} = \{ [1,3,4,5,6,8,9,10,11,12,13,14], [1,3,4,6,7,8,9,10,11,12,13,14], [2,3,5,6,8,9,12,14], [2,3,6,7,8,9,12,14], [0] \}$$

$\overline{C_s}$ 'deki en büyük blok 12 durumudur. Böylece $m_2=12$ ve $LB=12$ olarak elde edilir. Öte yandan $UB=12$ ($\min\{n, |C_s|\}$) olarak elde edilir. Bu bençmark için $UB=LB$ olduğundan minimal makine Adım 3'de elde edilmiş olur. Rho yöntemin ile bu bençmark için minimal makine elde edilememiştir.

2.9 Test Sonuçları

Bu bölümde verilen yönteme dayanan bir durum indirgeme programı olan SRC (State Reduction and Covering) MCNC FSM bençmarkları üzerinde test edilerek elde edilen sonuçlar, son yıllarda yayınlanan en önemli iki durum indirgeme algoritması Rho[1] ve Puri [2] sonuçları ile karşılaştırılmıştır. Test sonuçları aşağıdaki tabloda verilmiştir. Bu tablodan da görüldüğü gibi SRC programı ile diğer yöntemlere göre daha iyi sonuçlar elde edilmiştir. Diğer yöntemlerin indirgeyemediği bençmarklar kısa sürede indirgenmiştir. SRC programının testi için Pentium 4 1.8GHz işlemcili 256MB belleği olan bir PC kullanılmıştır..

Rho yöntemi ile karşılaştırıldığında tüm bençmarklarda SRC'nin çok daha hızlı indirgeme yapabildiği gözlenmiş, Rho yöntemi ile indirgenemeyen ex3 ve mark1 FSM'lerinin SRC tarafından saniyenin yüzde birinden daha kısa sürede indirgendiği gözlenmiştir. Ayrıca indirgenmesi zor bir bençmark olarak bilinen ex2 ile denendiğinde Rho yöntemi "time-out" ile sonuçlanmıştır. Bu bençmark için minimal sonuç SRC tarafından 102s.'de elde edilmiştir. Rho yöntemi için verilen indirgeme süreleri DEC Station 5000 platformunda elde edilmiştir.

Puri yöntemi ile karşılaştırıldığında SRC'nin yine tüm bençmarklarda daha hızlı indirgeme yaptığı gözlenmiştir. Ayrıca ex2, Puri yöntemi ile indirgenememiştir. Bu bençmark için C_a sınıfındaki blok sayısı 1366'dır. Puri yöntemi C_a sınıfını kullanarak indirgeme yaptığı için yöntemin bu bençmarkta zorlanacağı açıktır. Puri yöntemi için verilen indirgeme süreleri SUN SPARC 1+ Workstation platformunda elde edilmiştir.

Tablo 2.1 Bençmark Testlerinin Sonuçları

No	Bençmark	n	i	o	SRC		Rho [1]		Puri [2]	
					n _r	süre(s.)	n _r	süre(s.)	n _r	süre(s.)
1	bbara	10	4	2	7	0.00	7	0.15	7	0.00
2	bbsse	16	7	7	13	0.00	13	0.23	13	0.02
3	bbtas	6	2	2	6	0.00	6	0.05	6	0.00
4	beecount	7	3	4	4	0.00	4	0.07	4	0.01
5	cse	16	7	7	16	0.00	-	-	16	0.01
6	donfile	24	2	1	1	0.00	-	-	1	0.00
7	ex1	20	9	19	18	0.00	18	0.37	18	0.01
8	ex2	19	2	2	5	102	-	Zaman aşımı	-	-
9	ex3	10	2	2	4	0.00	bulunamadı	-	4	2.51
10	ex4	14	6	9	14	0.00	-	-	14	0.01
11	ex5	9	2	2	3	0.00	3	5.17	3	0.23
12	ex6	8	5	8	8	0.00	-	-	8	0.01
13	ex7	10	2	2	3	0.00	3	12.03	3	0.26
14	keyb	19	7	2	19	0.00	-	-	19	0.02
15	lion	4	2	1	4	0.00	-	-	4	0.01
16	lion9	9	2	1	4	0.00	4	0.00	4	0.01
17	mark1	15	5	16	12	0.00	bulunamadı	-	12	0.02
18	modulo12	12	1	1	1	0.00	-	-	1	0.01
19	opus	10	5	6	9	0.00	9	0.05	9	0.01
20	planet	48	7	19	48	0.00	-	-	48	0.11
21	planet1	48	7	19	48	0.00	-	-	48	0.11
22	s1	20	8	6	20	0.00	-	-	20	0.02
23	s1a	20	8	6	1	0.00	-	-	1	0.01
24	s8	5	4	1	1	0.00	-	-	1	0.01
25	sand	32	11	9	32	0.00	-	-	32	0.02
26	scf	121	27	56	97	0.20	97	6.77	97	0.99
27	shiftreg	8	1	1	8	0.00	-	-	8	0.02
28	sse	16	7	7	13	0.00	13	0.20	13	0.02
29	styr	30	9	10	30	0.00	-	-	30	0.04
30	tav	4	4	4	4	0.00	-	-	4	0.01
31	tbk	32	6	3	16	0.71	16	21.60	16	1.64
32	train4	4	2	1	4	0.00	-	-	4	0.01
33	train11	11	2	1	4	0.00	4	0.38	4	0.01

n: durum sayısı i: giriş sayısı o: çıkış sayısı n_r: indirgenen makinenin durum sayısı

3. DURUM İNDİRGEME İÇİN GELİŞTİRİLEN YÖNTEMLERİN BOOLE FONKSİYONLARININ İNDİRGENMESİNE UYGULANMASI

Bu çalışmada durum indirgeme için geliştirilen ve bir önceki bölümde anlatılan teknikler iki seviyeli kombinezonsal devrelerin tasarımına uygulanmıştır. İki seviyeli kombinezonsal devre tasarımı kombinezonsal devre tasarımında önemli bir adım olduğu gibi ardışıl devre tasarımında da önemli bir adımdır; çünkü ardışıl devrelerin sentezi kombinezonsal devrelerin sentezine indirgenebilmektedir. Ayrıca çoğu zaman çok-seviyeli kombinezonsal devreler iki seviyeli kombinezonsal devreler yardımı ile tasarlanmaktadır.

İki seviyeli kombinezonsal devre tasarımında problemin boyutu arttığında problemin çözüm süresi ve kullanılan bellek miktarı çok arttığından minimal sonuç veren klasik yöntemler bazı benchmarklarda yetersiz kalmaktadır. Bu tip yöntemlerin bilinen bir örneği Quine-McCluskey [8] yöntemidir. Bu tip yöntemlerin yetersiz kalmasından dolayı işlem süresini ve kullanılan bellek miktarını azaltan fakat optimale yakın sonuç veren bazı sezgisel (buluşsal) yöntemler geliştirilmiştir. Bu yöntemler arasında MINI [9], AREVALO [10] ve ESPRESSO [3] bulunmaktadır. Bu yöntemlerin yanında optimal sonuç veren ESPRESSO-EXACT [3] ve SCHERZO [11] programları bulunmaktadır.

Bu çalışmada n değişkenli m tane Boole fonksiyonunu iki seviyeli minimal VE/VEYA kombinezonsal lojik devre ile gerçekleyebilmek için bir algoritma geliştirilmiştir. Bu algorithmadan yararlanılarak MORP (Multiple Output Reduction Program) isimli hızlı ve az bellek kullanan bir bilgisayar programı geliştirilmiştir. MORP kısa sürede minimal veya minime oldukça yakın bir çözüm vermektedir. İndirgeme işlemine devam edildiğinde, MORP ile ilk aşamada elde edilen çözüm sürekli olarak iyileştirilerek minimal örtü de elde edilmektedir. Minimal örtünün birden fazla olması durumunda da bütün çözümler MORP ile elde edilebilmektedir.

MORP, Pentium 4 1.8GHz işlemcili 256MB belleğe sahip bir PC üzerinde çalıştırılarak MCNC benchmarkları üzerinde denenmiştir. Bu benchmarklarda minimal örtüler kısa sürede hesaplanmıştır. MORP'un, MCNC benchmarklarındaki başarısı UCB (University of California Berkeley) üniversitesinde geliştirilen ESPRESSO programının verdiği sonuçlarla, sonuç bölümünde karşılaştırılmıştır. Tablo 3.3'den

görüldüğü gibi MORP, MCNC bençmarklarının büyük çoğunluğunda ESPRESSO programına üstünlük sağlamıştır.

3.1 Geliştirilen Yöntemin Adımları

Bu çalışmada geliştirilen yöntem n değişkenli m tane Boole fonksiyonunu birlikte indirgeyerek optimal sonucu vermektedir. Bu yöntemden yararlanılarak MORP geliştirilmiştir. Bu program University of California Berkeley’de geliştirilen ESPRESSO [3] programlarının sonuçları ile karşılaştırılmıştır. Yöntem 3 algoritmadan oluşmaktadır. Bu algoritmalar aşağıda, kısımlar halinde verilmiştir.

3.1.1 Algoritma 1

Bu algoritmada asal bileşenler elde edilmektedir. Bu algoritma için klasik Quine-McCluskey [8] metodu kullanılmıştır.

3.1.2 Algoritma 2

Bu algoritmada minimale yakın bir çözüm elde edilmektedir. Bunun için bir örtü matrisi oluşturularak temel asal bileşenler elde edilmektedir. Temel asal bileşenler çözüme eklenerek bu asal bileşenlere karşı düşen satır ve sütunlar örtü matrisinden çıkarılmaktadır. Bir sonraki adımda bu çalışmada geliştirilen iyileştirilmiş En Büyük Adımlar Yöntemi (EBAY) (greedy algorithm [3]) örtü matrisine uygulanır. Bu nedenle algoritma bu adımda minimal değil minimale yakın bir sonuç vermektedir. EBAY [3] algoritması uygulandıktan sonra tekrar temel asal bileşen arama adımına geri dönlür. Bu çevrim örtü matrisinde satır kalmayıncaya kadar devam eder.

İyileştirilmiş EBAY yönteminde, EBAY [3] yöntemi ile elde edilmiş olan her bloğun örtüye girme nedeni yani o blok tarafından içerilen fakat diğer bloklar tarafından içerilmeyen elemanlar hesaplanır. Minimale yakın örtüdeki iki bloğun örtüye girme nedenleri eğer bir blok tarafından içeriliyor ise, bu iki blok atılıp yerlerine yeni blok minimal örtüye dahil edildiğinde minimale yakın örtüdeki blok sayısı 1 azaltılmış olur. Bu sayede eğer k tane bloğun minimal örtüye girme nedenleri bir blok tarafından içeriliyor ise minimale yakın örtüdeki blok sayısı $k-1$ azaltılır. Bu durum Bölüm 5’de bir örnek ile de açıklanmıştır.

Bu algoritmanın sonunda n değişkenli m tane Boole fonksiyonu için minimal veya minimale yakın bir çözüm kısa sürede elde edilmektedir. Bu çözümdeki asal bileşen sayısı bir üst sınır olarak kullanılmaktadır.

3.1.3 Algoritma 3

Bu algoritmada bir minimal örtü veya bütün minimal örtüler elde edilir. Algoritma 2’de elde edilen örtü matrisi bu adımda da kullanılır. Ancak temel asal bileşenlere ilişkin satır ve sütunlar ile örtülen satır ve sütunlar matristen çıkarılmıştır. Dallanma yöntemi ile örtü problemi daha küçük örtü problemleri elde edilecek şekilde ayrıştırma işlemi yapılır. Elde edilen örtü problemleri çevrimsel hale getirilerek alt sınırları hesaplanır. Eğer alt sınır, üst sınırdan büyük veya eşit ise ilgili örtü problemi budanmış olur. Algoritma 2’de elde edilen üst sınır arama ağacının budanmasında kullanılır. Algoritma 3 sonucunda elde edilen asal bileşenler temel asal bileşenlere eklenerek minimal çözüm elde edilir. Eğer bütün minimal örtülerin elde edilmesi isteniyorsa, satır örtme, sütun örtme gibi elemeler ve alt sınırın üst sınıra eşit olması durumundaki elemeler yapılmaz.

Algoritma-2 ve Algoritma-3 ile verilen algoritmalarda, UCPS’den (Unate Covering Problem Solver) yararlanılmıştır. UCPS algoritmaları, verilen herhangi bir örtü matrisi için bir minimal örtüyü veya bütün minimal örtülerini elde edebilmektedir. Geliştirilen UCPS algoritmaları Bölüm 5’de, algoritmaların akış diyagramları ise EK B’de verilmiştir.

3.2 İndirgeme Örneği 1

Aşağıda verilen üç fonksiyonun birlikte indirgenmesi işlemleri adım adım incelenmiştir.

$$f_1=(7,9,11,13,14,15)+k(5)$$

$$f_2=(0,1,2,3,6,7,15)$$

$$f_3=(0,1,2,6,7,8,9,14,15)+k(3,4)$$

Bu fonksiyonların Algoritma 1 ile elde edilen asal bileşenleri 10 tanedir ve listesi aşağıda verilmiştir.

Tablo 3.1 Asal Bileşen Listesi

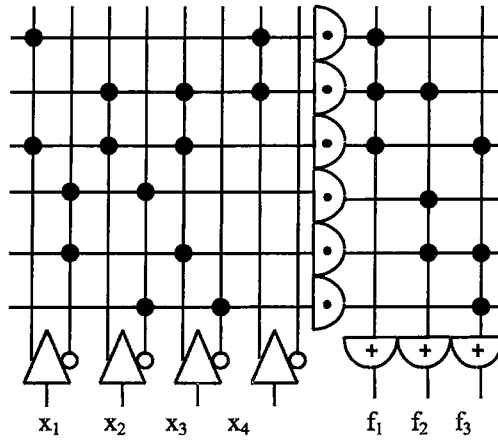
AB No	Asal Bileşen	Fonksiyon
PI ₁	-11- (X ₂ X ₃)	f[3]
PI ₂	-1-1 (X ₂ X ₄)	f[1]
PI ₃	1—1 (X ₁ X ₄)	f[1]
PI ₄	0-1- (X ₁ ' X ₃)	f[2,3]
PI ₅	0--0 (X ₁ ' X ₄ ')	f[3]
PI ₆	00-- (X ₁ ' X ₂ ')	f[2,3]
PI ₇	-00- (X ₂ ' X ₃ ')	f[3]
PI ₈	-111 (X ₂ X ₃ X ₄)	f[1,2,3]
PI ₉	111- (X ₁ X ₂ X ₃)	f[1,3]
PI ₁₀	1001 (X ₁ X ₂ ' X ₃ ' X ₄)	f[1,3]

Yukarıdaki tablodan yararlanarak elde edilen örtü matrisi aşağıdaki şekilde verilmiştir.

	f ₁						f ₂						f ₃									
	7	9	11	13	14	15	0	1	2	3	6	7	15	0	1	2	6	7	8	9	14	15
PI ₁																	X	X			X	X
PI ₂	X			X		X																
PI ₃		X	X	X		X																
PI ₄									X	X	X	X					X	X	X			
PI ₅														X			X	X				
PI ₆							X	X	X	X				X	X	X						
PI ₇														X	X				X	X		
PI ₈	X					X						X	X					X				X
PI ₉					X	X															X	X
PI ₁₀		X																		X		

Şekil 3.1 Örtü Matrisi

Yukarıdaki şekilden de görüldüğü gibi minimal örtüyü temel asal bileşenler belirler. Minimal örtü $M=\{PI_3, PI_4, PI_6, PI_7, PI_8, PI_9\}$ olarak belirlenir. Aşağıdaki şekilde fonksiyonların Karnaugh diyagramı ile indirgenmesi gösterilmiştir.



Şekil 3.4 Minimal devre

Şekil 3.3 ve Şekil 3.4'den görüldüğü gibi, ayrı ayrı indirgemeye karşı düşen devrede 11 kapı, 27 giriş ve 8 çarpım terimi, birlikte indirgemeye karşı düşen devrede ise 9 kapı, 23 giriş ve 6 çarpım terimi bulunmaktadır.

3.3 İndirgeme Örneği 2

Aşağıda örtü matrisi verilen iki fonksiyonun birlikte indirgenmesi işlemleri adım adım incelenmiştir.

	f_1			f_2		
	m_0	m_1	m_2	m_3	m_4	m_5
PI_1	X					X
PI_2			X		X	
PI_3				X		X
PI_4	X	X		X	X	
PI_5		X	X			

Şekil 3.5 Örtü matrisi

Yukarıdaki şekilden de görüldüğü gibi örtü matrisinde temel asal bileşen, örtülen satır veya sütun bulunmamaktadır. Bu durumda en çok minterm içeren PI_4 asal bileşeni seçilir. PI_4 asal bileşenine ilişkin satır ve sütunlar atıldığında aşağıdaki şekilde verilen örtü matrisi elde edilir.

	f_1	f_2
	m_2	m_5
PI_1		X
PI_2	X	
PI_3		X
PI_5	X	

Şekil 3.6 İkinci adımda örtü matrisi

Bu şekilde temel asal bileşen bulunmamaktadır. Ancak PI_1 asal bileşeni PI_3 asal bileşenini, PI_2 asal bileşeni ise PI_5 asal bileşenini kapsar. Bu satırlar elendiğinde aşağıdaki şekil elde edilir.

	f_1	f_2
	m_2	m_5
PI_1		X
PI_2	X	

Şekil 3.7 Son adımda örtü matrisi

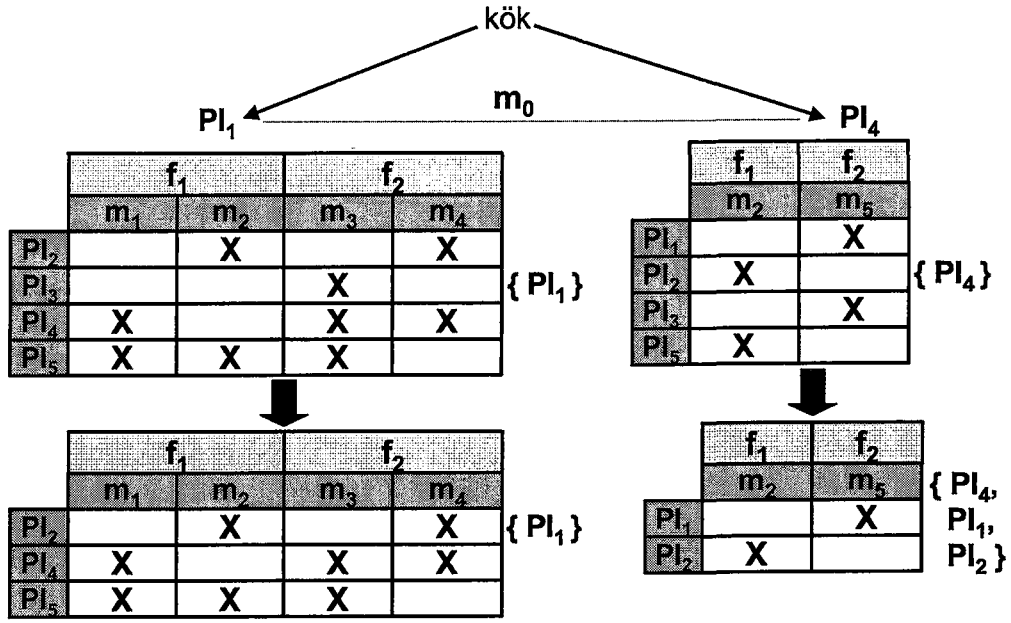
Yukarıdaki şekilden minimale yakın çözüm aşağıdaki gibi elde edilir:

$$f_1 = PI_4 + PI_2$$

$$f_2 = PI_4 + PI_1$$

Yukarıdaki çözümde 3 tane asal bileşen bulunduğundan üst sınır 3 olarak belirlenir. Örtü matrisinden mintermlerin ağırlıkları, yani tarafından kapsandıkları asal bileşen sayıları hesaplandığında tüm ağırlıkların 2 olduğu görülür.

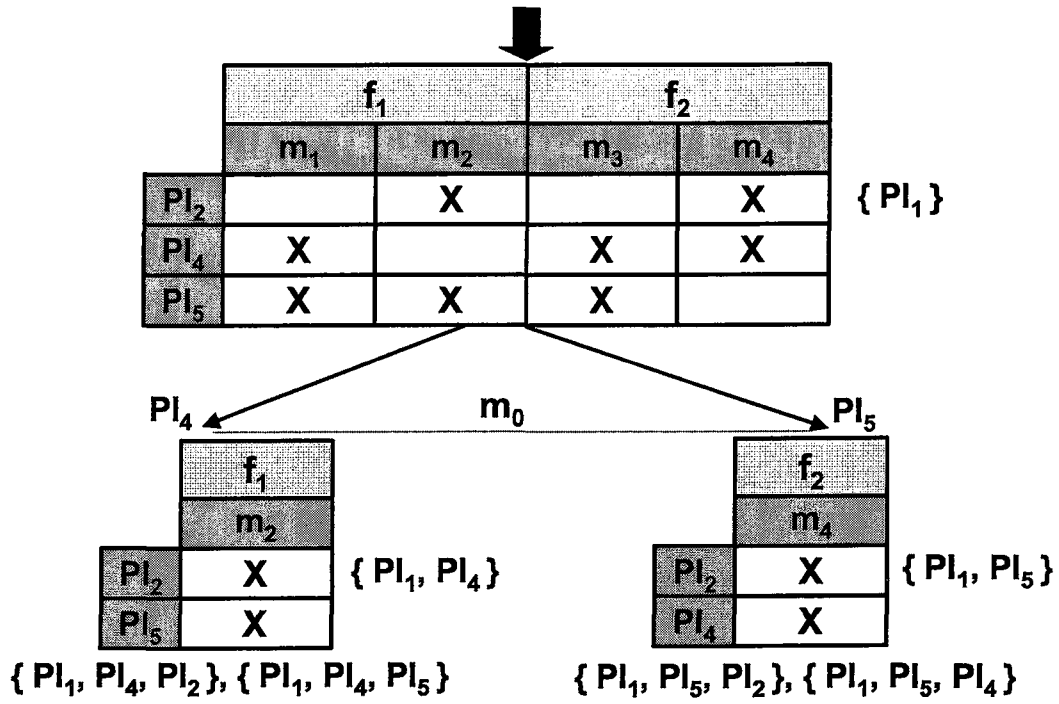
Ağırlıklarına göre dizilen mintermler örtü arama ağacının seviyelerini belirler.



Şekil 3.8 Örtü arama ağacı

Yukarıdaki şekilden de görüldüğü gibi sağ dalda $\{ PI_1, PI_2, PI_4 \}$ çözümü elde edilir. Bu ağaçtaki sol dalda elde edilen örtü probleminde tekrar dallanma yapıldığında aşağıda verilen çözümler elde edilir.

- 1.Çözüm: $\{ PI_1, PI_4, PI_2 \}$
- 2.Çözüm: $\{ PI_1, PI_4, PI_5 \}$
- 3.Çözüm: $\{ PI_1, PI_5, PI_2 \}$
- 4.Çözüm: $\{ PI_1, PI_5, PI_4 \}$



Şekil 3.9 Örtü arama ağacının son aşaması

Yukarıda verilen çözümlerin tamamı 3 asal bileşen içermektedir. Bu nedenle Algoritma 2 ile elde edilen minimale yakın örtü, minimaldir.

3.4 Sonuç

Bu Tezde geliştirilen MORP ve MORP-EBAY programlarının MCNC bençmarkları ile testlerinin sonuçları Tablo 3.3'de verilmiştir. MORP algoritmasının ilk adımında MORP-EBAY algoritması kullanılır. Bu şekilde elde edilen minimale yakın örtüdeki blok sayısı bir üst sınır olarak kullanılır. MORP algoritması ile bu üst sınır ve minimal örtü arama ağacı kullanılarak minimal örtü elde edilir.

Tablo 3.3'de NM minterm sayısını, NPI asal bileşen sayısını, MC minimal çözümdeki terim sayısını göstermektedir. Bu tablodan da görüldüğü gibi MORP'un iki algoritması (MORP ve MORP-EBAY) ESPRESSO ve ESPRESSO-EXACT programlarına üstünlük sağlamaktadır. MORP-EBAY, 12 bençmarkta ESPRESSO'dan daha iyi çözüm verirken, ESPRESSO sadece 3 bençmarkta MORP-EBAY'dan daha iyi çözüm vermektedir. Bu çözümlerdeki terim sayısı MORP-EBAY'dan 3 eksiktir. Ayrıca MORP-EBAY'ın üstün olduğu bençmarklardaki terim sayısı ESPRESSO'ya göre oldukça düşük olabilmektedir. Örneğin addm4 isimli bençmarkta MORP-EBAY ile 0.4 s.'de 190 terimli bir çözüm elde edilirken ESPRESSO ile 1s.'de 200 terimli bir çözüm elde edilmektedir. Bu bençmark için

MORP-EBAY dahi minimal çözümü verirken ESPRESSO-EXACT ile bu benchmark indirgenememiştir. Aynı şekilde apex4 isimli benchmarkta MORP-EBAY ile 4 s.'de 430 terimli bir çözüm elde edilirken ESPRESSO ile 6s.'de 436 terimli bir çözüm elde edilmektedir. MORP-EBAY optimale yakın sonuç veren bir algoritma olmasına karşın 30 benchmarkın 18'inde optimal, 3'ünde optimale 1 terim uzak ve geriye kalanların çoğunda 2 veya birkaç terim optimale uzak sonuç vermiştir. Bu sonuçlar MORP-EBAY'ın genelde optimal, aksi halde optimale oldukça yakın sonuç verdiğini göstermektedir. Tablo 3.3'den 8 benchmarkta ESPRESSO-EXACT'ın çözüm vermediği ve m4 ile mlp4 benchmarklarında ise hata mesajı verdiği görülmektedir. MORP, 30 benchmarkın 26'sında optimal, sadece 4 tanesinde optimale yakın sonuç vermiştir. Tablo 3.3'den ESPRESSO-EXACT ile indirgenemeyen benchmarkların önemli bir kısmının MORP ile indirgenebildiği görülmektedir.

Tablo 3.3'den görüldüğü gibi MORP ve MORP-EBAY yöntemleri alu2, alu3 ve prom1 benchmarklarında ESPRESSO ve ESPRESSO-EXACT yöntemlerinden daha yavaş sonuç vermiştir. Bu benchmarkların tamamında minimal örtüyü asal bileşenler oluşturmaktadır. MORP ve MORP-EBAY yöntemlerinde, durum indirgeme için geliştirilen örtü algoritmalarının Boole fonksiyonlarının indirgenmesine uygulanması amaçlandığından ve bu nedenle asal bileşenleri elde etmek için klasik ve yavaş olan Quine-McCluskey yöntemi kullanıldığından, bu benchmarklarda hızlı sonuç elde edilememiştir. MORP ve MORP-EBAY yöntemlerinde hızlı bir asal bileşen bulma algoritması kullanıldığında bu benchmarklardaki indirgeme süresi de azaltılabilir.

Tablo 3.3'den görüldüğü gibi prom2 benchmarkı için MORP ve MORP-EBAY yöntemleri ile ESPRESSO yöntemine göre minimalden daha uzak örtüler elde edilmiştir. Bu durumun temel sebebi, ESPRESSO-EXACT'ın bazı sezgisel algoritmalar kullanması, bu yöntemin minimal örtüyü elde etme amacının olmaması nedeniyle bütün asal bileşenleri hesaplamadan verilen örtüdeki bileşenleri azaltmaya çalışmasıdır. Bu sezgisel algoritmalar ile, bazı örtü problemlerinde bu nedenlerden dolayı hızlı bir şekilde minime yakın örtü elde edilebilmektedir. Ancak bütün benchmarklar gözönüne alındığında, bu algoritmalar ile genel olarak önerilen yöntemden daha büyük örtüler elde edildiği gözükmemektedir. Ayrıca MORP yöntemi kullanılarak, yeterli bellek varsa ve yeterli süre beklenirse minimal makine elde edilebilecektir.

Tablo 3.2 MCNC ben markları

No	Ben�mark	NİxNO	NM	NPI	"Cyclic Core" Matrisi	MC
1	5xp1	7x10	784	390	220x389	63
2	9sym	9x1	696	1680	1680x696	84
3	addm4	9x8	1310	1122	409x427	189
4	alu2	10x8	1380	434	0x0	68
5	alu3	10x8	1076	540	0x0	64
6	apex4	9x19	2970	2336	992x1089	428
7	apla	10x12	157	201	201x157	25
8	dekoder	4x7	49	26	7x16	9
9	dist	8x5	591	401	85x89	120
10	ex5	8x63	7620	2532	1941x4597	65
11	f51m	8x8	1024	561	21x50	76
12	lin.rom	7x36	2306	1087	996x2107	128
13	luc	8x27	2246	190	0x0	26
14	m1	6x12	218	59	0x0	19
15	m2	8x16	831	243	109x298	47
16	m3	8x16	1105	344	195x574	62
17	m4	8x16	2134	670	414x861	101
18	max46	9x1	62	49	0x0	46
19	max128	7x24	1616	469	390x1292	78
20	max512	9x6	1616	535	417x826	133
21	max1024	10x6	3232	1278	1164x2468	259
22	mlp4	8x8	678	606	438x463	121
23	newtpla1	10x2	14	6	0x0	4
24	newtpla2	10x4	608	23	0x0	9
25	pope	6x48	1614	593	356x981	59
26	prom1	9x40	8306	9326	0x0	472
27	prom2	9x21	3027	2635	2501x2854	287
28	rd53	5x3	48	51	0x0	31
29	rd73	7x3	428	211	0x0	127
30	rd84	8x4	411	633	0x0	255
31	risc	8x31	844	46	0x0	28
32	sao2	10x4	747	184	0x0	58
33	sqr6	6x12	259	205	181x203	47
34	sym10	10x1	837	3150	3150x837	210
35	tms	8x16	790	162	10x36	30
36	z5xp1	7x10	576	390	220x298	63
37	z9sym	9x1	420	1680	1680x420	84

Bu  alıřmada geliřtirilmiř olan bilgisayar programı hem arařtırma amacı ile hem de eēitim amacı ile kullanılmaya elveriřlidir. Program, pratikte karřılařılabilecek problemleri kısa s rede indirgeyebilecek hıza sahiptir.

Tablo 3.3 MORP'un Test Sonuçları

No	Benchmark	NIxNO	MORP	MORP -EBAY	Espresso -Heuristic	Espresso - Exact
1	5xp1	7x10	63 / 4.7 s.	63 / 1.6 s.	65 / 0 s.	63 / 1 s.
2	9sym	9x1	84 / 31 s.	85 / 1.5 s.	87 / 0 s.	Error
3	addm4	9x8	189 / 11.3 s.	190 / 0.4 s.	200 / 1 s.	Error
4	alu2	10x8	68 / 43 s.	68 / 43 s.	68 / 0 s.	68 / 1 s.
5	alu3	10x8	64 / 39 s.	64 / 39 s.	66 / 0 s.	64 / 1 s.
6	apex4	9x19	428 / 125 s.	430 / 4 s.	436 / 6 s.	-
7	apla	10x12	25 / 94 s.	25 / 94 s.	25 / 0 s.	25 / 1 s.
8	dekoder	4x7	9 / 0 s.	9 / 0 s.	9 / 0 s.	9 / 0 s.
9	dist	8x5	120 / 0.3 s.	120 / 0.3 s.	121 / 0 s.	120 / 1 s.
10	ex5	8x63	72/145 s	72/145 s.	74 / 2 s.	Error
11	f51m	8x8	76 / 1.26 s.	76 / 1.26 s.	77 / 1 s.	76 / 2 s.
12	lin.rom	7x36	128 / 446 s.	145 / 18 s.	128 / 6 s.	Error
13	luc	8x27	26 / 1.4 s.	26 / 1.4 s.	26 / 1 s.	26 / 1 s.
14	m1	6x12	19 / 0 s.	19 / 0 s.	19 / 0 s.	19 / 1 s.
15	m2	8x16	47 / 0.5 s.	47 / 0.3 s.	47 / 0 s.	47 / 1 s.
16	m3	8x16	62 / 5.7 s.	65 / 0.6 s.	65 / 1 s.	62 / s.
17	m4	8x16	102 / 1032	107 / 3.2 s.	107 / 1 s.	Error
18	max46	9x1	46 / 0 s.	46 / 0 s.	46 / 0 s.	46 / 0 s.
19	max128	7x24	78 / 3.5 s.	85 / 0.3 s.	82 / 1 s.	78 / 2 s.
20	max512	9x6	133 / 2123	136 / 4s.	142 / 2 s.	133 / 7 s.
21	max1024	10x6	267 / 410 s.	270 / 60 s.	274 / 7 s.	Error
22	mlp4	8x8	121 / 20 s.	125 / 0.7 s.	128 / 1 s.	Error
23	newtpla1	10x2	4 / 0 s.	4 / 0 s.	4 / 0 s.	4 / 0 s.
24	newtpla2	10x4	9 / 3 s.	9 / 0 s.	9 / 0 s.	9 / 0 s.
25	pope	6x48	59 / 32 s.	62 / 1.6 s.	62 / 1 s.	59 / 5 s.
26	prom1	9x40	472 / 61 s.	472 / 61 s.	472 / 13 s.	Error
27	prom2 ^(*)	9x21	287/0 s.	287 / 0 s.	287 / 23 s.	Error
28	rd53	5x3	31 / 0.1 s.	31 / 0.1 s.	31 / 0 s.	31 / 0 s.
29	rd73	7x3	127 / 0.8 s.	127 / 0.8 s.	127 / 1 s.	127 / 1 s.
30	rd84	8x4	255 / 0.33 s.	255 / 0.33 s.	255 / 0 s.	255 / 1 s.
31	risc	8x31	28 / 0.66 s.	28 / 0.66 s.	28 / 0 s.	28 / 0 s.
32	sao2	10x4	58 / 4.9 s.	58 / 4.9 s.	58 / 0 s.	58 / 1 s.
33	sqr6	6x12	47 / 173 s.	48 / 0.1 s.	49 / 0 s.	47 / 1 s.
34	sym10	10x1	210 / 66 s.	210 / 43 s.	210 / 2 s.	Error
35	tms	8x16	30 / 0.2 s.	30 / 0.2 s.	30 / 0 s.	30 / 1 s.
36	z5xp1	7x10	63 / 4.78 s.	63 / 0.3 s.	63 / 1 s.	63 / 1 s.
37	z9sym	9x1	84 / 892 s.	88 / 1 s.	85 / 0 s.	Error

(*) Bu benchmark için programlara verilen yarı indirgenmiş ifade bir minimal örtüdür. Bu nedenle ESPRESSO programı, programa verilen yarı indirgenmiş ifadedeki terim sayısını azaltamamakta ve giriş olarak verilen yarı indirgenmiş ifadeyi sonuç olarak vermektedir. MORP programı, bu benchmarkta giriş olarak verilen yarı indirgenmiş ifadeyi bir üst sınır olarak kullanmaktadır. Ancak bellek yetersizliğinden dolayı, MORP programı ile bu benchmark için 287 terimden daha küçük bir örtü bulunmadığı gösterilememiştir. Yeteri kadar bellek kullanılabildiği takdirde MORP programı ile 287 terimden daha küçük bir örtü olmadığı gösterilebilir.

4. DURUM İNDİRGEME İÇİN GELİŞTİRİLEN YÖNTEMLERİN ASENKRON ARDIŞIL DEVRELERİN KODLANMASI PROBLEMİNE UYGULANMASI

Bu çalışmada, durum indirgeme yöntemlerinde geliştirilen, 2'nci ve 5'inci bölümde anlatılan teknikler ardışıl devrelerin kodlanmasına uygulanmış, Tracey [12] yöntemine dayanan yeni bir yöntem ve bu yönteme dayanarak OPASKOD programı geliştirilmiştir.

Ardışıl devreler senkron veya asenkron olarak sınıflandırılırlar. Senkron ardışıl devrelerde durum değişimi saat işaretiyle sağlanır. Eğer bir durum değişimi birden çok flip-flop'un durum değiştirmesini gerektiriyorsa, bütün flip-flop'lar aynı anda durum değiştirir; çünkü bütün flip-flop'lar saat işaretiyle senkronize edilmektedir. Senkron ardışıl devrelerde giriş işareti değiştiği zaman devre hemen durum değiştirmez; bir sonraki duruma, girişten sonra saat işaretinin alınmasıyla geçilir.

Senkron devrelerin kullanılmalarının uygun olmadığı başlıca durumlar şunlardır :

1. Devre herhangi bir zamanda değişen ve saat işareti ile senkronize edilemeyen girişlere sahiptir.
2. Devre yeterince büyük ve lojik elemanların cevabı yeterince hızlıysa, elektriksel işaretin iletim hatlarındaki ulaşım zamanı önemlidir. Bu durumda düzgün senkron çalışmayı sağlamak için gerekli saat işaretinin bütün flip-flop'lara aynı anda varışını sağlamak çok zor olabilir.
3. Devrenin mümkün olduğunca hızlı çalışması istenir. Oysa senkron ardışıl devrelerde durum değişikliği için, giriş işareti değiştikten sonra saat işaretinin değişmesi için de beklemek gerekir.

Bu durumlarda, asenkron ardışıl devrelerin kullanılması gerekebilir. Asenkron ardışıl devreler, bir saat işareti ile senkronize edilmemektedir. Girişlerde bir değişiklik olduğunda, bir saat işareti beklemeden durum değişikliği oluşur. Eğer çeşitli flip-flop'ların durumları değişiyorsa, bu flip-flop'lar aynı anda durum değiştiremeyebilirler.

Zamanlama problemleri nedeniyle asenkron devrelerin tasarımı senkron devrelerden

daha zordur. Uygun tasarlanmış bir senkron devrede zamanlama problemleri ortadan kaldırılmıştır; çünkü saat işareti uygulanmadan önce bütün flip-flop girişlerinin kararlı bir değere ulaşması için giriş değişikliklerinden sonra yeterince beklenmektedir. Oysa, bir asenkron devredeki çeşitli yollar boyunca eşit olmayan gecikmelerden kaynaklanan zamanlama problemlerini ortadan kaldırmak için özel tasarım teknikleri gereklidir.

Bu bölümde asenkron ardışıl devre tasarım adımları da incelenmiştir. Bu adımlar aşağıda verilmiştir.

1. Problemin sözle tanımlanmasından giriş ve çıkış arasındaki ilişki belirlenir. Bir satırda bir tane kararlı durum olacak şekilde ilkel durum tablosu oluşturulur ve her kararlı duruma ilişkin çıkış belirlenir.
2. İlkel durum tablosu indirgenerek minimal makinaya ilişkin durum tablosu belirlenir.
3. Durumlar arasında kritik yarışlar olmayacak şekilde bir durum kodlaması yapılır.
4. Durum tablosunda herbir durum için kabul edilen kodlar yazılarak bir kod geçiş tablosu oluşturulur. Eğer Mealy çıkış tablosu kullanılırsa, çıkıştaki geçişleri önlemek için gerekli, daha önce belirlenmemiş olan durum çıkışları doldurulur. Kod geçiş tablosundan yararlanarak, çıkışları ve bir sonraki durumları veren Boole fonksiyonları bulunur.
5. Aksama olmayacak şekilde gerçeklemeler bulunur.
6. Tasarım labaratuarda veya bilgisayar simülasyonu ile test edilir.

Kritik yarışsız durum kodlama asenkron ardışıl devre tasarımındaki en zor adımlardan biridir. Kritik yarışlar dikkate alınmadan yapılan durum kodlama devrenin hatalı çalışmasına neden olabilir. Bu nedenle kritik yarışların önlenmesi gereklidir. Kritik yarışsız durum kodlama yapabilmek için çeşitli yöntemler geliştirilmiştir.

J.H.Tracey [12] tarafından temel moddaki normal durum tablolarında kritik yarışsız kodlama yapabilmek için bir yöntem geliştirilmiştir. Bu yöntemde bütün durum değişimleri doğrudan geçiş özelliği göstermektedir. Böylece daha hızlı çalışma sağlanmaktadır. Yöntem, her giriş için bölmelemelerin yazılması ve bundan yararlanarak her giriş için ayrı ayrı kritik yarışların önlenmesi esasına dayanmaktadır. Bu nedenle, Tracey [12] yöntemi ile kodlanan makinedeki durum

değişimleri hızlıdır. Bu üstünlüğünün yanında bu yöntem, kodlama sonucundaki durum değişkeni sayısının artabilmesi, bu yöntemle dayanan programların çalışma süresinin fazla olması ve çok bellek kullanımı gerektirmesi sakıncalarına sahiptir. Tracey, işlem zamanını kısaltabilmek için geliştirdiği yöntemde bazı değişiklikler yapan iki yöntem de önermiştir. Ancak bu iki yöntem kullanıldığında kodlama için gerekli durum değişkeni sayısı artmaktadır.

Robert J. Smith [28], Tracey yöntemini temel alan yeni bir yöntem geliştirmiştir. Bu yöntem kullanılarak daha büyük durum tabloları için kodlama yapılabilir, ancak durum değişkeni sayısı en az Tracey yöntemindeki kadardır; daha da fazla olabilir.

P. David Fisher ve Sheng-Fu Wu [29] büyük durum tablolarını kodlayan bir yöntem geliştirmişlerdir. Bu yöntemde durum geçişleri doğrudan geçişler olmayabilir. Durum geçişlerinin doğrudan geçişler olmaması, karşı düşen devrenin hızını azaltabilmektedir. Bu yöntem kullanılarak, iki durum arasında geçiş varsa, bu durumlara ilişkin Hamming uzaklığı 1 olacak şekilde kodlama yapılır; bu mümkün değilse ara durumlar kullanılarak her seferinde bir tane durum değişkeninin değişmesi sağlanır.

Bu kısımda, Tracey yöntemine dayanarak geliştirilen yeni bir yöntem verilmiştir. Bu yöntem ile kritik yarışsız durum kodlama ve durumlar arasında doğrudan geçiş olması şartlarını sağlayan, minimum sayıda durum değişkeni elde edilir. Bu yöntemle dayanarak geliştirilen OPASKOD (Optimal Asenkron Kodlama) programı, çeşitli benchmarklar ile test edilmiştir. Test sonuçları Tablo 4.1’de verilmiştir.

4.1 Geliştirilmiş Olan Durum Kodlama Yöntemi

Geliştirilen algoritmayı incelemeden önce aşağıdaki tanımların verilmesi gerekmektedir.

Bir d_j durumu i girişi uygulandığında d_k durumuna gitsin.

Tanım 4.1: d_j durumu ile d_k durumunun ikili tabandaki kodları aynı ise d_j o giriş altında kararlıdır denir.

Tanım 4.2: Eğer d_j ve d_k durumlarının ikili tabandaki kodları bir bit fark ediyorsa devre geçiş durumundadır denir.

Tanım 4.3: d_k durumunun ikili tabandaki kodu d_j durumunun ikili tabandaki kodundan iki veya daha fazla bit fark ediyorsa, devre şimdiki durumdan bir sonraki

duruma yarışıyor denir.

Tanım 4.4: Eğer yarış durumu varsa ve eşit olmayan iletim gecikmelerinin devreyi istenenden farklı bir kararlı duruma ulaştırması ihtimali varsa yarış kritik yarış olarak adlandırılır, aksi halde yarışa kritik olmayan yarış denir.

Tanım 4.5: d_i durumundan d_j durumuna doğrudan geçiş $[d_i, d_j]$ olarak yazılır ve d_i durumundaki bütün durum değişkenlerinin aynı anda değişmesiyle elde edilir.

Tanım 4.6: Eğer eşit olmayan iletim gecikmelerinin bu iki geçişin ortak bir durumu paylaşmalarına neden olması ihtimali varsa, $[d_i, d_j]$ doğrudan geçişi $[d_k, d_l]$ doğrudan geçişi ile kritik yarış yapar.

Tanım 4.7: Minimum geçiş zamanlı durum kodlamada bütün geçişler doğrudan geçişlerdir.

Örnek 4.2: Şekil 4.1' deki A makinası ele alınsın.

	00	01	11	10
1	①	2	4	3
2	3	②	②	②
3	③	4	2	③
4	1	④	④	2

Şekil 4.1 Ardışıl makina A

	y_1y_2	$y_1y_2y_3y_4$	$y_1y_2y_3$
1	00	0000	000
2	01	0011	011
3	10	0101	110
4	11	1001	101

Şekil 4.2 Makina A'nın durumlarının 3 farklı kodlaması

Şekil 4.1'deki makina A için yapılan Şekil 4.2'nin 1. sütunundaki kodlama hatalı çalışmaya neden olur. $I1=00$ girişi altında $2 \rightarrow 3$ geçişi sağlanırken y_1 ve y_2 aynı anda değişmezse ve y_2 , y_1 den önce değişirse $01 \rightarrow 00$ kod geçişi olur. 00 kararlı a durumu olduğundan başka bir değişiklik olmaz ve makina 10 ile kodlanan 3 durumuna gitmesi gerekirken hatalı çalışarak 1 durumuna gider; kritik yarış vardır.

İkinci sütunda yapılan kodlama iki doğrudan geçişin birbiriyle kritik yarışmasını gösterir. $I2=01$ girişi altında $1 \rightarrow 2$ ve $3 \rightarrow 4$ geçişleri 0001 durumunda geçici olarak bulunabilir. Bu durumda 1 den 2 ye geçiş, 3 den 4 e geçiş ile kritik yarışıyor denir. Bu kodlama, bütün geçişler doğrudan olacak sınırlaması olmazsa doğru çalışmayı sağlayabilir. $1 \rightarrow 2$ geçişi $0000 \rightarrow 0010 \rightarrow 0011$ durumuna geçişle sağlanabilir. $3 \rightarrow 4$ geçişi de $0101 \rightarrow 1101 \rightarrow 1001$ durumuna geçişle sağlanabilir. Araya başka durumların girmesiyle ve her seferinde sadece bir bit değişmesiyle ardışıl devrelerin kritik yarışsız kodlanması D.Fisher [21] tarafından incelenmiştir.

Şekil 4.2'deki tablonun üçüncü sütununda verilen kodlama minimum geçiş zamanlı kodlama örneğidir. Durumlar arasındaki bütün geçişler doğrudan geçişler olarak sınıflandırılabilir ve hepsi yaklaşık T zamanında olur.

4.1.1 Minimum Geçiş Zamanlı Kodlama için Durum Kümesine Uygulanan Bölmeleme

Minimum geçiş zamanlı kodlama yapılmasında durum kümesi için bölmeleme tanımı kullanılmaktadır. Durum kümesi bölmelemelerinin minimum geçiş zamanlı kodlamada kullanılmasında aşağıda verilmiş olan tanımlardan yararlanılacaktır.

Tanım 4.8: Bir D durumlar kümesine ilişkin π bölmelemesi, çift çift kesişimleri boş olan D'nin alt kümelerinin oluşturduğu kümedir. Ayrık alt kümeler π 'nin blokları olarak adlandırılır. Eğer bu alt kümelerin birleşim kümesi D ise, bölmeleme tamamen belirlidir; aksi halde bölmeleme kısmen belirlidir. π içinde görünmeyen D'nin elemanlarına o bölmelemeye göre belirli olmayan veya seçmeli elemanlar denir.

Şekil 4.2'de üçüncü sütunda verilen kodlamada y_1 değişkeni π_1 bölmelemesini tanımlar: $\pi_1 = \{1,2 ; 3,4\}$. π_1 in blokları 1,2 ve 3,4' dür. Kodlama π_1 , π_2 ve π_3 bölmelemeleri ile elde edilir. Bu bölmelemeler sırasıyla y_1 , y_2 ve y_3 ile elde edilir. İki bloklu bölmelemenin kodlanmasında, hangi bloğun 0 ile hangisinin 1 ile kodlandığının önemi yoktur. Benzer şekilde kodlamadaki bölmelemelerin sırası da önemsizdir.

Tanım 4.9: Her bir kararsız durumun doğrudan kararlı bir duruma gittiği akış

tablosuna normal akış tablosu denir. Bu şekilde bir akış tablosuna karşı düşen asenkron makinaya da normal makina denir.

4.1.2 Bir Sütunun Tanımladığı Bölmeleme Sayısı

Gözönüne alınan sütundaki tekrarlı kararlı durumlar $1,2,3,...,k$ olsun ve tekrar sayıları $n_1, n_2, \dots, n_k$ olsun. Öte yandan aynı sütundaki tekrarsız kararlı durum sayısı p olsun. Buna göre i 'nci tekrarlı durumun tanımladığı geçiş sayısı n_i 'dir. Bir tekrarsız durum ile bu tekrarlı durumun tanımladığı bölmeleme sayısı n_i 'dir; bu tekrarlı durum ile tekrar sayısı n_j olan durumun tanımladığı bölmeleme sayısı $n_i.n_j$ 'dir. Buna göre gözönüne alınan sütunun tanımladığı bölmeleme sayısı aşağıdaki teorem ile ifade edilebilir.

Teorem 4.1: Bir sütundaki tekrarlı kararlı durumlar $1,2,3,...,k$ olsun ve tekrar sayıları $n_1, n_2, \dots, n_k$ olsun. Öte yandan aynı sütundaki tekrarsız kararlı durum sayısı p olsun. O zaman bu sütunun tanımladığı bölmeleme sayısı aşağıdaki gibidir.

$$p.(n_1+n_2+\dots+n_k) + \sum_{i=1}^{k-1} \sum_{j=i+1}^k n_i.n_j$$

Yukarıdaki formülden de görüleceği gibi bir sütunda tekrarlı kararlı durum yoksa, o sütun bölmeleme tanımlamaz; yani o sütunun kritik yarış oluşturmaması için önlem almak gerekmez.

Teorem 4.2: Bir sütunun tanımladığı bölmelemeler $\pi_1, \pi_2, \dots, \pi_k$ olsun; π_i bölmelemesinin blokları y_i durum değişkeni ile kodlansın ve durum kodlaması buna uygun olarak tamamlansın. Kodlama bu şekilde yapıldığı takdirde kritik yarış oluşamaz.

İspat:

π bölmelemesi ya $\{d_i, d_j; d_m, d_n\}$ şeklinde ya da $\{d_i, d_j; d_m\}$ şeklindedir.

$\pi = \{d_i, d_j; d_m, d_n\}$ olsun ve bloklar 0 ve 1 şeklinde kodlansın. Buna göre $[d_i, d_j]$ geçişi yerine $[d_m, d_n]$ geçişi oluşamaz. Çünkü $[d_i, d_j]$ geçişi sırasında y durum değişkeni 0 olarak kalmaktadır. Dolayısıyla değişen durum değişkenleri sonucunda d_m veya d_n durumlarının kodları oluşamaz. Çünkü bu durumların y durum değişkeni 1'dir. Benzer şekilde $[d_m, d_n]$ geçişi sırasında d_i ve d_j durumlarının kodları oluşamaz. Öte yandan bölmeme $\{d_i, d_j; d_m\}$ şeklinde ise ve d_i, d_j ve d_m bloklarının kodları farklı olduğundan $[d_i, d_j]$ geçişi esnasında d_m durumunun kodu oluşamaz.

4.1.3 Durum Kodlama Yöntemi

Bir akış tablosu için önce bölmelemeler listesi oluşturulur. Daha sonra bölmelemeler listesindeki bütün bölmelemeleri kapsayan yeni bölmelemeler oluşturulmaya çalışılır. Yeni bölmelemelerin herbiri bir durum değişkeni ile ifade edileceğinden, en az sayıda durum değişkeni olacak şekilde bölmelemeler oluşturulmalıdır. $\pi_1=\{ab,cf\}$, $\pi_2=\{ae,cf\}$ olduğunda $\pi_A=\{abe,cf\}$ bölmelemesi, π_1 ve π_2 bölmelemeleri ile değiştirilebilir. Herhangi bir bölmelemeler çifti tek bir bölmeleme ile değiştirilemeyecek durumuna gelindiğinde, kodları oluşturacak bölmelemeler bulunmuş olur.

Bütün bölmelemeleri kapsayan en az sayıdaki bölmelemeyi sistematik bir şekilde bulmak için önce bölmelemeler listesi “0”, “1” ve “-” lerden oluşan bir matrise dönüştürülür. Bu matrise bölmelemeler matrisi denir. Bölmelemeler matrisini oluşturmak için bölmelemelerin bir bloğu “0” ve diğer bloğu “1” ile kodlanır. Bu matriste bölmelemeler satırları, durumlar sütunları oluşturur. Bir satıra ilişkin bölmelemedeki blokların kodlarının karşı düştükleri durumlara yazılmasıyla bölmelemeler matrisi elde edilir. Bölmelemede olmayan durumlar keyfidir ve “-” ile gösterilir. Bölmelemeler matrisinin herhangi bir satırı yerine tümleyeni kullanılabilir.

Bütün bölmelemeleri kapsayan en az sayıdaki bölmelemeyi elde edebilmek için bölmelemeler matrisi indirgenir. İndirgenmiş matristeki bölmeleme satırları bütün bölmelemeleri kapsamaktadır. Bu durumda indirgenmiş matrisin sütunlarında makinanın durumlarına karşı düşen kodlar elde edilir. İndirgenmiş matrisin satır sayısı kullanılacak durum değişkeni sayısına eşittir. Böylece minimum durum değişkeni ile kodlama, bölmelemeler matrisindeki bölmelemelerin hepsini kapsayan minimum satırlı indirgenmiş matristen elde edilmiş olur.

Tanım 4.10: Bölmelemeler matrisinin iki satırı a ve b olsun. a ve b satırları her ikisinin de belirli oldukları yerlerde aynı iseler uyumludurlar. a ve b uyumlu çiftine karşı düşen satır, a ve b’ nin belirli oldukları yerlerde a ve b ile aynı, diğer yerlerde de keyfidir.

Bölmelemeler matrisini indirgemek için durum indirgeme için geliştirilen ve genelleştirilmiş şekli Bölüm 5’de verilen örtü algoritmalarından yararlanılmıştır.

Bölmelemeler matrisi indirgenirken, bir satırın diğer bütün satırlarla ve satırların tümleyenleri ile uyumlu olup olmadığı kontrol edilir. Bütün uyumlu satır çiftleri yazıldıktan sonra, Bölüm 2’de verilen ağaç yapısı ile maksimal uyumlular elde edilir.

4.1.3.1 Maksimal Uyumlu Bölmelemelerin Bulunması

Tümleyenler de dahil olmak üzere bütün bölmelemelerin oluşturduğu bloktan uyumsuz olanlar ağaç yapısı ile ayrılır. Ayırma işlemi bittiğinde diğerleri tarafından kapsanan uyumlu bölmelemeler elenir. Geriye kalan bölmelemeler maksimal uyumlu bölmelemelerdir. Bulunan maksimal uyumlu bölmelemeler arasından, bütün bölmelemeleri kapsan en az sayıda maksimal uyumlu seçilir. Bu amaçla satırları maksimal uyumlu bölmelemeler, sütunları ise bölmelemeler olan bir örtü matrisi yazılır. Bu örtü matrisi Bölüm 5’de verilen UCPS (Unate Covering Problem Solver) yöntemi ile çözülür.

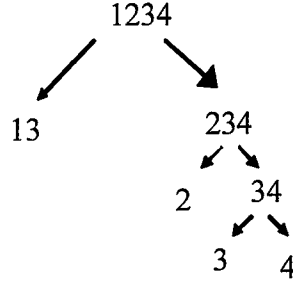
Örnek 4.2: Şekil 4.1’deki A makinasının, kritik yarış olmayacak şekilde kodlanma adımları aşağıda verilmiştir.

Şekil 4.1’den $\pi_1=\{14,23\}$, $\pi_2=\{12,34\}$, $\pi_3=\{14,23\}$, $\pi_4=\{13,24\}$ bölmelemeleri elde edilir. Bölmelemeler matrisi aşağıdaki verilmiştir.

	I1	I2	I3	I4
1	0	1	1	0
2	0	0	1	1
3	0	1	1	0
4	0	1	0	1

Şekil 4.3 Bölmelemeler matrisi

Bu matristen uyumlu çiftler $\{(1,3)\}$ ve uyumsuz çiftler $\{(1,2),(1,4),(2,3), (2,4), (3,4)\}$ elde edilir. Maksimal uyumlular $\{13, 2, 4\}$ olarak elde edilir. Maksimal uyumsuzlar ise $\{124, 234\}$ olarak elde edilir. Maksimal uyumlular kümesinin elde edilmesi Şekil 4.4’de gösterilmiştir.



Şekil 4.4 Maksimal uyumlular kümesinin elde edilmesi

Bu kümelerden yararlanarak minimal örtüye girecek maksimal uyumlular $\{13, 2, 4\}$ olarak bulunur. Bu bloklar ile oluşturulan kodlama Şekil 4.5' de verilmiştir.

	1	2	3	4
13	0	1	1	0
2	0	0	1	1
4	0	1	0	1

Şekil 4.5 A makinasının durumlarının kodları

Şekil 4.5' den 1 durumunun kodu 000, 2 durumunun kodu 101, 3 durumunun kodu 110, 4 durumunun kodu 011 olarak elde edilir. Farklı iki durumun kodları da farklı olduğundan kodlama tamamlanmıştır; yani durumlar bu şekilde kodlanırsa kritik yarış oluşmaz. Görüldüğü gibi durum sayısı 4 olduğu halde, durum değişkeni sayısı 2 değil, 3'dür; yani kodlama minimal değildir. Öte yandan, elde edilen kodlama, Tracey yöntemi ile elde edilebilecek minimal bir kodlamadır.

Örnek 4.2: Şekil 4.6 ile verilen B makinasının, kritik yarış oluşmayacak şekilde kodlanmasına ilişkin adımlar aşağıda verilmiştir.

	I0	I1	I2	I3
1	1,0	3,0	4,0	3,0
2	1,0	6,0	3,0	2,0
3	6,0	3,0	3,0	3,0
4	-,0	4,0	4,0	2,0
5	1,0	4,0	3,0	5,0
6	6,0	6,0	-,0	5,0

Şekil 4.6 B makinasının durumlarının tablosu

Şekil 4.6' da verilen B makinesinin bölmeleme listesi Şekil 4.7'de verilmiştir.

π_0	{24,13}	0101--
π_1	{56,13}	0-0-11
π_2	{56,24}	-0-011
π_3	{14,23}	1001--
π_4	{35,14}	0-101-
π_5	{45,13}	0-011-
π_6	{26,13}	010--1
π_7	{26,45}	-1-001
π_8	{36,12}	001--1
π_9	{36,15}	0-1-01

Şekil 4.7 B makinesinin bölmeleme listesi

Uyumlu bölmelemeler hesaplandığında $\{\pi_0, \pi_2\}$, $\{\pi_3, \pi_4\}$, $\{\pi_3, \pi_7\}$, $\{\pi_3, \pi_9\}$ ve $\{\pi_5, \pi_7\}$ olarak bulunur. Tümleyenler ise $\{\pi_0, \pi_3, \pi_5\}$ olarak seçilir.

Maksimal uyumlular kümesi $C_s = \{(\pi_0, \pi_1, \pi_5, \pi_6), (\pi_7, \pi_9, \pi_3'), (\pi_2, \pi_4, \pi_8), (\pi_7, \pi_5'), (\pi_2, \pi_0'), (\pi_4, \pi_3'), (\pi_8, \pi_9), (\pi_6, \pi_7), (\pi_1, \pi_2), (\pi_3)\}$ olarak elde edilir.

Satırları maksimal uyumlular, sütunları bölmelemeler olan örtü matrisi Şekil 4.8'de verilmiştir.

Şekil 4.8'de minimal örtü $\{(\pi_0, \pi_1, \pi_5, \pi_6), (\pi_7, \pi_9, \pi_3'), (\pi_2, \pi_4, \pi_8)\}$ olarak elde edilir. Bu örtüye ilişkin kodlama Şekil 4.9' da verilmiştir.

	π_0	π_1	π_2	π_3	π_4	π_5	π_6	π_7	π_8	π_9
$(\pi_0, \pi_1, \pi_5, \pi_6)$	1	1	0	0	0	1	1	0	0	0
(π_7, π_9, π_3')	0	0	0	1	0	0	0	1	0	1
(π_2, π_4, π_8)	0	0	1	0	1	0	0	0	1	0
(π_7, π_5')	0	0	0	0	0	1	0	1	0	0
(π_2, π_0')	1	0	1	0	0	0	0	0	0	0
(π_4, π_3')	0	0	0	1	1	0	0	0	0	0
(π_8, π_9)	0	0	0	0	0	0	0	0	1	1
(π_6, π_7)	0	0	0	0	0	0	1	1	0	0
(π_1, π_2)	0	1	1	0	0	0	0	0	0	0
(π_3)	0	0	0	1	0	0	0	0	0	0

Şekil 4.8 B makinesi için örtü matrisi

Aşağıdaki şekilden de görüldüğü gibi B makinesinin minimum geçiş zamanlı ve

kritik yarış olmadan kodlanabilmesi için 3 durum değişkenine ihtiyaç vardır ve durum değişkeni sayısı minimaldir.

Durum	Kod
1	000
2	110
3	011
4	100
5	101
6	111

Şekil 4.9 B makinesinin kritik yarışsız kodlanması

4.2 Test Sonuçları

Bu çalışmada durum indirgeme için geliştirilen, 2'nci ve 5'inci bölümde anlatılan teknikler, asenkron ardışıl devrelerin kodlanmasına uygulanmış ve Tracey [12] yöntemine dayanan yeni bir yöntem ve bu yöntemle dayanarak OPASKOD programı geliştirilmiştir. OPASKOD programı çeşitli benchmarklar ile test edilmiştir. Test sonuçları Tablo 4.1'de verilmiştir. Bu tablodan da görüldüğü gibi, Tracey yöntemi ile daha önce kodlanamayan TraceyD ve SmithM4 benchmarkları, OPASKOD ile kodlanabilmiştir. Bu benchmarkları Tracey tabanlı bir yöntemle, yani elde edilen makine minimum geçiş zamanlı olacak şekilde kodlayan başka bir yöntemle literatürde rastlanmamıştır.

Tablo 4.1 OPASKOD benchmark test sonuçları

Benchmark	NS	NIxNO	NP	NCP	NSV	Süre (s.)
TraceyA	4	2x1	3	0	3	0 s.
TraceyB	5	2x1	7	3	3	0 s.
TraceyC	6	2x1	10	3	3	0 s.
TraceyD	12	2x1	52	38	6	701 s.
SmithM4	12	2x1	59	30	5	44 s.
SmithM6	6	2x2	10	2	4	0.14 s.

NS: Durum Sayısı

NP: Bölmeleme Sayısı

NSV: Durum Değişkeni Sayısı

NI: Giriş Sayısı

NCP: Tümleyenli Bölmeleme Sayısı

NO: Çıkış Sayısı

TEK KOŞULLU BİR ÖRTÜ PROBLEMİNİN ÇÖZÜMÜ İÇİN BİR YÖNTEMİN VE BU YÖNTEME DAYANAN BİR BİLGİSAYAR PROGRAMININ GELİŞTİRİLMESİ

Örtü problemi (covering problem) evrensel bir problemidir. Ardışıl devre tasarımında durum indirgeme ve durum kodlama, kombinezonsal devre tasarımında Boole fonksiyonlarının indirgenmesi başta olmak üzere pek çok alanda bu problemle karşılaşilmektedir. Bu nedenlerle bu problemin kısa sürede az bellek kullanılarak çözümü önem kazanmaktadır.

Bu Tezde genel örtü problemlerini çözmek için geliştirilmiş olan yöntem ve bu yönteme dayalı bilgisayar programı UCPS (Unate Covering Problem Solver) ile, tamamen keyfi olarak oluşturulan 1000x1000 boyutlarında bir matrisle ifade edilen bir örtü problemine ilişkin minimal çözüm saniyeler mertebesinde bir sürede belirlenebilmektedir. Program, Boole fonksiyonlarının VE-VEYA devreleri ile minimal gerçekleştirilmesine ilişkin çeşitli bençmarklarda denenmiş ve çözümler kısa sürede belirlenmiştir. Geliştirilen program MCNC bençmarklarından elde edilen örtü matrislerine uygulanmıştır.

Örtü problemi, bir Boole fonksiyonunun indirgenmesi ise, satırlar asal bileşenlere, sütunlar da mintermlere karşı düşer. Ardışıl makinalarda durum indirgeme halinde, satırlar uyumlu durumlara ilişkin bloklara, sütunlar ise durumlara karşı düşer. Matristeki satır ve sütun sayısının büyük olması halinde minimal bir örtüyü makul bir makine süresinde belirlemek oldukça zordur.

Bir boyutlu örtü problemi (unate covering problem) bir matris ile ifade edilebilir. $C[i,j]_{m \times n}$ örtü matrisinin satırları örten bloklara, sütunları da örtülen elemanlara karşı düşer. Matrisin i . satırı B_i bloğuna, j . sütunu da v_j elemanına karşı düşsün. Matrisin c_{ij} elemanı 1 ise, B_i bloğu v_j elemanını örtüyor, c_{ij} elemanı 0 ise B_i bloğu v_j elemanını örtmüyor demektir. Minimal örtü problemi şöyle tanımlanır: Minimum sayıda blok belirle öyle ki tüm elemanlar örtülsün. Aşağıda 8 bloklu ve 23 elemanlı bir örtü problemine ilişkin örtü matrisi verilmiştir.

B_1 10000100100101010001010
 B_2 01101101010000110110001
 B_3 10000100100110001000010
 B_4 00101010111010011001101
 B_5 01000111001001010110001
 B_6 11010100100111010100010
 B_7 00100010101010101000101
 B_8 01010101011000010101000

Şekil 5.1 8 bloklu ve 23 elemanlı bir örtü problemi

Genel örtü problemi bir matrisle ifade edilebileceği gibi, Petrick fonksiyonu adı verilen ve toplamlar-çarpımı biçimindeki bir Boole fonksiyonu ile de ifade edilebilir. Matrisle ifade etmede her satır bir bloğa karşı düşerken, Petrick fonksiyonunda her çarpan bir elemana karşı düşmektedir ve o elemanı örten blokların toplamına eşittir. Kolayca görülebilir ki minimum sayıda değişkene 1 değeri verildiğinde fonksiyonun değeri 1'e eşit oluyorsa bu minimum sayıdaki blok bir minimal örtü oluşturur. Böyle bir değişken kümesi belirlemenin bir yolu, Petrick fonksiyonunu çarpımlar-toplamı biçimine dönüştürmektir; bu açınımdaki minimum sayıda çarpan içeren her terim bir minimal örtü verir.

Bu çalışmada, toplamlar çarpımı biçimindeki bir Petrick fonksiyonunu, çarpımlar-toplamı biçimine dönüştüren, dolayısıyla tüm minimal örtüleri veren bir bilgisayar programı da verilmiştir. Bu program, büyük örtü problemlerinin çözümü için elverişli değildir; fakat eğitim amaçlı örtü problemleri için yeteri kadar hızlıdır ve dallanma yönteminin anlaşılmasını kolaylaştırmaktadır. Örneğin $P=(A+B+C)(D+E+F) (A+H+I) (C+D+I)$ fonksiyonu 81 dal içermektedir, yani, çarpımlar toplamı açılımının 81 terimi vardır. Bu örtü probleminin minimal örtüsü, Petrick fonksiyonu açıldığında AD olarak elde edilir.

4.3 Geliştirilen Yöntem

Minimale yakın bir örtüyü kısa sürede bulabilmek için şöyle bir yol izlenebilir: En çok sayıda elemanı örten (örtü matrisi C'nin en çok 1 içeren satırına karşı düşen) bir blok seç. Seçilen blok B_1 olsun. B_1 bloğuna ilişkin satırı ve bu satırın örttüğü sütunları örtü matrisi C'den sil. Elde edilen matriste yine aynı kritere göre B_2 satırını seç. B_2 satırını ve örttüğü sütunları örtü matrisi C'den sil. Bu şekilde C örtü matrisinde sütun kalmayınca kadar devam et. Seçilen blokların kümesi $B=\{B_1, B_2, \dots, B_k\}$ olsun. Bu kümede $B_i \subseteq B_j$ olan bloklar var ise B_i 'yi kümeden at. Bu şekilde elde edilen B kümesi minimale yakın bir örtü verir. Bu yöntemle EBAY (En

Büyük Adımlar Yöntemi) denilecektir.

Bu çalışmada EBAY yöntemi iyileştirilerek kısa sürede minimale yakın bir örtü elde edilmekte, elde edilen sonuç minimal çözüm için bir üst sınır (UB) alınarak dallanma yöntemiyle minimal örtü (MC) elde edilmektedir. Eğer UB değeri MC' deki blok sayısına yakın ise, MC kısa sürede elde edilmektedir. Öte yandan, MC'deki blok sayısı m_0 olsun; MC için bir alt sınır (LB) belirlenmekte ve alt sınırın m_0 'ye yakın olması durumunda çözüme hızla ulaşılabilir.

Bu çalışmada geliştirilen iyileştirilmiş EBAY yöntemi ve iyileştirilmiş dallanma yöntemi ile önce m_0 'ye yakın üst sınırı belirlenmekte sonra da bundan yararlanılarak minimal örtü elde edilmektedir.

Yukarıda verilen 8 bloklu ve 23 elemanlı örtü problemine ilişkin minimal örtü $MC=\{B_2, B_4, B_6\}$ olarak elde edilir. Bu problem için $m_0=3$ olmakta ve aynı çözüm literatürde verilen EBAY (Greedy, [3]) yöntemi ile de elde edilmektedir. Ancak EBAY yöntemi blok ve eleman sayısı çok az olabilen örtü problemlerinde dahi minimal örtüyü elde edemeyebilir. Bu durum büyük örtü problemlerinin çözülmesini gerektiren benchmarklarda daha belirgin hale gelir. Literatürde verilen EBAY yöntemi bazı benchmarklarda minimalden oldukça uzak örtüler elde etmektedir.

Aşağıda EBAY yönteminin minimal örtüyü elde edemediği bir örtü problemi ve iyileştirilmiş EBAY yöntemi ile minimal örtünün elde edilme adımları verilmiştir.

Tek koşullu bir örtü problemi matris şeklinde ifade edildiği gibi bir küme ile de ifade edilebilir. Örneğin $\{123, 134, 235, 56, 67, 27\}$ kümesi 6 bloklu ve 7 elemanlı bir örtü problemine ifade etmektedir. Bu örtü probleminde EBAY yöntemi kullanılarak sırası ile 123,56,134,67 blokları örtüye dahil edilmek üzere seçilir. Bu yöntem ile elde edilen minimale yakın örtüde 4 blok bulunmakta ve birbirini içeren bloklar bulunmamaktadır. İyileştirilmiş EBAY yönteminde, EBAY yöntemi ile elde edilmiş olan her bloğun örtüye girme nedeni yani o blok tarafından içerilen fakat diğer bloklar tarafından içerilmeyen elemanlar hesaplanır. Minimale yakın örtüdeki iki bloğun örtüye girme nedenleri eğer bir blok tarafından içeriliyor ise, bu iki blok atılıp yerlerine yeni blok minimal örtüye dahil edildiğinde minimale yakın örtüdeki blok sayısı 1 azaltılmış olur. Örneğin yukarıdaki çözümde 123 ve 56 blokları minimale yakın örtüden atılıp yerlerine 235 bloğu eklendiğinde minimale yakın olan 4 bloklu çözümdeki blok sayısı 1 azaltılır ve minimal örtü elde edilir. Aynı şekilde EBAY yöntemi ile elde edilen minimale yakın örtüden 123 ve 67 blokları atılıp yerlerine 27 bloğu eklendiğinde yine 3 bloklu minimal örtü elde edilir.

Geliştirilmiş olan yöntem ile eğer k tane bloğun minimal örtüye girme nedenleri bir blok tarafından içeriliyor ise minimele yakın örtüdeki blok sayısı $k-1$ azaltılır.

Eğitim amaçlı uygulamalarda, örtü probleminin birden çok çözümü varsa, tümünün elde edilmesi istenmekte, ayrıca programın görsel ve kullanıcı dostu olması arzu edilmektedir. Bu özelliklerin gerçekleştirilmesi halinde program büyük bençmarklarda yavaşlamaktadır. Eğitim amaçlı problemler büyük olmadığından, bir örtü probleminin tüm çözümleri saniyeler mertebesindeki bir makine süresinde belirlenebilmektedir. Bu nedenle geliştirilen yöntemle ilişkin bilgisayar programının iki versiyonu hazırlanmıştır.

4.4 Sonuçlar

Bu Tezde geliştirilmiş olan iyileştirilmiş EBAY, iyileştirilmiş dallanma yöntemleri ve bu yöntemlere dayalı bilgisayar programı ile tamamen keyfi olarak oluşturulan 1000×1000 boyutlarında bir matrisle ifade edilen bir örtü problemine ilişkin minimal çözüm saniyeler mertebesinde bir sürede belirlenebilmektedir. Program, Boole fonksiyonlarının VE-VEYA devreleri ile minimal gerçekleştirilmesine ilişkin çeşitli bençmarklarda denenmiş ve çözümler kısa sürede belirlenmiştir. Geliştirilen algoritmaları kullanan MORP'un verdiği sonuçlar, University of California Berkeley'de geliştirilen ESPRESSO programı ile elde edilen sonuçlarla 3'üncü bölümde Tablo 3.3'de karşılaştırılmıştır. Bu tablodan da görüldüğü gibi UCPS algoritmasını kullanan MORP, pek çok bençmarkta optimal, geriye kalan bençmarklarda ise optimele oldukça yakın sonuç vermiştir. Oysa ESPRESSO programı genelde optimal çözümü değil, optimele yakın bir çözüm vermektedir.

EBAY yöntemi, max1024 isimli bençmarkta, 298 bloklu minimele yakın bir örtü elde etmektedir. Minimal örtüde ise 259 blok bulunmaktadır. Yani EBAY yöntemi bu bençmarkta 39 blok fazlası ile (%15) minimele yakın örtüyü elde edebilmektedir. İyileştirilmiş EBAY yönteminde ise 298'li çözüm iyileştirilerek 270 bloklu çözüm elde edilmiştir. Bu durumda iyileştirilmiş EBAY yöntemi ile %11'lik bir iyileştirme sağlanmış, minimal örtüden %4 fazla bloklu çözüm elde edilmiştir. max512 isimli bençmarkta ise EBAY yöntemi ile 153 bloklu minimele yakın bir örtü elde edilmektedir. Minimal örtüde ise 133 blok bulunmaktadır. Yani EBAY yöntemi bu bençmarkta 20 blok fazlası ile (%15) minimele yakın örtüyü elde edebilmiştir. İyileştirilmiş EBAY yönteminde ise 153'lü çözüm iyileştirilerek 137 bloklu çözüm elde edilmiştir. Bu durumda iyileştirilmiş EBAY yöntemi ile %12'lik bir iyileştirme sağlanmış, minimal örtüden %3 fazla bloklu çözüm elde edilmiştir. Diğer pek çok bençmarkta ise (5xp1, sqr6, m2, gibi) iyileştirilmiş EBAY yöntemi ile tam bir

iyileştirme sağlanmış ve minimal örtü elde edilmiştir. Minimal örtünün elde edilemediği bençmarklarda, iyileştirilmiş dallanma yöntemi ile minimal örtü elde edilmiştir.

UCPS, kısmen belirli ardışıl makinelerde durum indirgeme problemine de uygulanmıştır. UCPS algoritmasını kullanan SRC (State Reduction and Covering) programı ile elde edilen sonuçlar, Puri [2] ve Rho [1] tarafından geliştirilmiş olan programların verdiği sonuçlarla 2'nci bölümde Tablo 2.1'de karşılaştırılmıştır. Bu tablodan da görüldüğü gibi SRC programı ile diğer yöntemlere göre hızlı indirgeme yapılabilmektedir.

UCPS algoritmalarında kullanılan teknikler durum kodlama problemine de uygulanmıştır. Tracey minimum geçiş zamanlı durum kodlama algoritması büyük örtü problemlerinin çözümünü gerektirmektedir. Tracey kodlama yöntemini temel alan yeni bir durum kodlama yöntemi, UCPS algoritması kullanılarak geliştirilmiştir. Geliştirilen program literatürdeki bençmarklarda denenmiş, kritik yarışsız kodlama işlemi hızlı bir şekilde gerçekleştirilmiştir. Test sonuçları Tablo 4.1'de verilmiştir.

5. GENEL ÖRTÜ PROBLEMLERİNİN ÇÖZÜMÜNDE KULLANILABİLECEK YENİ BİR YÖNTEM

Örtü problemi üzerinde yıllardan beri çalışmalar yapılmaktadır; fakat bu problemin çözümü için polinomsal dereceden bir yöntem geliştirilememiştir. Geliştirilmiş olan algoritmaların karmaşıklık derecesi NP-complete (Non-deterministic Polynomial-complete)'dir. Problemin boyutu arttığında çözüm süresi ve kullanılan bellek miktarı çok artmaktadır. Sürekli olarak gelişen tümdevre teknolojisinde gün geçtikçe daha karmaşık örtü problemlerinin çözülmesine ihtiyaç duyulmaktadır.

Örtü probleminde örtülen elemanlar ve örten bloklar vardır. Örten bloklardan minimum sayıda kullanarak tüm elemanları örtme problemi örtü problemi olarak tanımlanır. Bu problemin çözümününe minimal örtü denir.

Geliştirilmiş minimal örtü bulma yöntemlerinde genel uygulama örtüyü oluşturabilecek blokların hesaplanması (Boole fonksiyonları için asal bileşenlerin, durum indirgeme için C_a 'nın bloklarının hesaplanması) ve bu bloklardan minimal örtünün oluşturulmasıdır. Minimal örtünün belirlenmesinde bazı yöntemler en büyük maksimal uyumsuz blok M_2 'deki (Maximal Clique) eleman sayısı m_2 'yi kullanmaktadır [1,2,3,4]. Bu yöntemlerde M_2 için bir Petrick fonksiyonu yazılmakta ve bu fonksiyon açılarak M_2 'nin elemanları için tüm kısmi örtüler bulunmakta, bunların içinde örtü oluşturan bir kombinezon varsa bir minimal örtü elde edilmektedir. Kısmi örtüler bir çok durumda bir minimal örtü içermekte, bazen de hiçbir kısmi örtü, örtü oluşturmamaktadır. Hiç bir kısmi kısmi örtünün bir minimal örtü oluşturmaması halinde yeni blokların kısmi örtüye eklenmesi ile örtülmemiş elemanların da örtülmesi gerekir. M_2 için yazılan Petrick fonksiyonu ile elde edilen kısmi örtülerin ne zaman bir minimal örtü oluşturduğu, ne zaman bir minimal örtü oluşturmadığı konularında bir çalışmaya literatürde rastlanmamıştır. Bu Tezde bir minimal örtüdeki blokların sayısını belirlemek için yeni bir yöntem geliştirilmiştir.

Bu tezde verilen bir algoritma ile m_2 kolayca belirlenebilmektedir. Eğer $f_m = m_6 - m_2$ ile tanımlanan f_m de makul bir sürede belirlenebilirse, minimal örtüdeki blok sayısı m_6 belirlenmiş olur. Bazı uygulamalarda m_6 değerini bilmek yeterlidir. Ayrıca eğer m_6 değeri biliniirse, bir minimal örtüyü elde etmek kolay hale gelmektedir.

5.1 Tanımlar

Tanım 6.1: Bir örtü probleminde örtülecek eleman sayısı (Boole fonksiyonlarını indirgeme problemi için minterm sayısı, durum indirgeme için durum sayısı) n olsun. m_2 tane blok ile kapsanabilecek maksimum eleman sayısı ise f_n olsun. Bu durumda $n-f_n$ sayısı f_d ile gösterilecektir.

Tanım 6.2: Bir örtü probleminde m_2 tane blok ile f_n tane eleman kapsandığında kapsanmamış olarak geriye kalan f_d tane elemanı kapsayabilmek için minimal örtüye dahil edilmesi gereken minimum blok sayısı f_m ile gösterilecektir; minimal örtüdeki blok sayısı m_0 ise $f_m = m_0 - m_2$ 'dir.

5.2 Minimal Örtüyü Oluşturan Blokların Sayısının Hesaplanması

Maksimal uyumsuzların en büyük bloğundaki eleman sayısı m_2 bir alt sınırdır; yani $m_2 \leq m_0$ 'dür. Bu alt sınır minimal örtüdeki blok sayısına eşit değilse $f_m \neq 0$ 'dır. Aşağıda $f_m \neq 0$ olması için bir teorem verilmiştir.

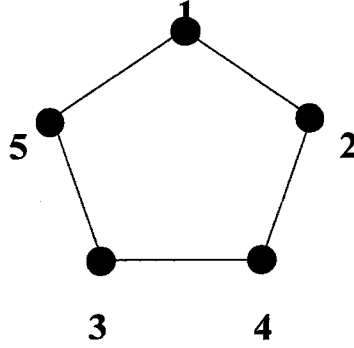
Teorem 6.1: Eleman sayısı n olan bir örtü probleminde $f_m \neq 0$ olabilmesi için eleman sayısı n en az 5 olmalıdır.

İspat: Bu teoremin ispatı aşağıda adımlar halinde verilmiştir:

- 1) $m_2=1$ ise bütün elemanlar birbirleri ile uyumlu olduğundan blok sayısı 1'dir ve bu blok tüm elemanları örter; $f_m=0$ 'dır. O halde $f_m \neq 0$ olması için $m_2 > 1$ olmalıdır. Yani m_2 sayısının en küçük değeri 2'dir. Genelliği bozmadan $m_2=2$ seçilsin ve M_2 bloğunu oluşturan elemanlar 1 ve 2 olsun.
- 2) $n=3$ ise üçüncü eleman ya 1 ve 2 ile uyumsuzdur; yani, ya 2 ile ya da 1 ile uyumludur. Aksi halde $m_2=3$ olmaktadır. Buna göre uyumlular sınıfı $C_u=\{1,3,2\}$ veya $C_u=\{1,2,3\}$ olacağından $m_0=2$ ve $f_m=0$ olur.
- 3) $n=4$ ise 3 ve 4 elemanı hem 1 hem de 2 ile aynı anda uyumsuz olamaz. Çünkü o durumda m_2 'nin artması gerekir. O halde 3 ve 4 elemanları, 1 ve 2 elemanlarından en az bir tanesi ile uyumludur. Eğer 3 ve 4 elemanları farklı elemanlar ile uyumlu ise (bir tanesi 1 elemanı ile uyumlu, diğeri 2 elemanı ile uyumlu) $f_m \neq 0$ olamaz. Çünkü bu durumda bu iki blok bir örtü oluşturur; $m_0=2$ ve $f_m=0$ olur. Öte yandan 1 ve 2 elemanları aynı eleman ile uyumlu ise (ikisi de 1 ile uyumlu veya ikisi de 2 ile uyumlu) $f_m \neq 0$ olması için 3 ve 4 elemanlarının birbirleri ile uyumsuz olması gerekir. Fakat 3 ve 4 elemanları birbirleri ile uyumsuz ise, 3 ve 4 elemanlarının her ikisi de 1 veya 2 ile uyumsuz olduğundan

m_2 'nin artması gerekir (Bu durumda $M_2=134$ veya $M_2=234$ olması gerekir).
Sonuç olarak $n=4$ için $f_m \neq 0$ olamaz.

- 4) $n=5$ ise 3 ,4 elemanlarını 1 ile uyumlu, 5 elemanını 2 ile uyumlu ve 3 ile 4 elemanlarını birbirleri ile uyumsuz olarak seçtiğimizde $m_2=2$ olarak kalır ve $f_m \neq 0$ durumu oluşur. Böyle bir durumda $f_m=1$ 'dir. Bu durum için uyumsuzluk grafi Şekil 6.1'de verilmiştir.



Şekil 6.1 $n=5$ için $f_m \neq 0$ olan bir durumda uyumsuzluk grafi

Teorem 6.2: Eleman sayısı $n=5$ veya $n=6$ olan bir örtü probleminde f_m 'nin maksimum değeri 1'dir ve $f_m=1$ olabilmesi için uyumluluk ve uyumsuzluk graflarının beşgen içermesi gerekir.

İspat: Bu teoremin ispatı aşağıda $n=5$ için ve $n=6$ için ayrı ayrı verilmiştir.

Eleman sayısı $n=5$ olması halinde tanımlanabilecek $f_m \neq 0$ olan bütün örtü problemlerine ilişkin uyumsuzluk grafları Şekil 6.2'de verilmiştir. Bu şekilden de görüldüğü gibi $n=5$ için $f_m \neq 0$ olan örtülerin uyumsuzluk grafi bu şekildeki 12 tane uyumsuzluk grafindan bir tanesidir. Bu nedenle f_m 'nin maksimum değeri 1'dir ve bu değeri alabilmesi için uyumsuzluk grafinin beşgen içermesi gerekir.

Eleman sayısı $n=6$ olması halinde tanımlanabilecek $f_m \neq 0$ olan bütün örtü problemleri için uyumsuzluk graflarının birbirlerinden farklı olanları Şekil 6.3'de verilmiştir. Bu şekilden de görüldüğü gibi $n=6$ için $f_m \neq 0$ olan örtülerin uyumsuzluk grafi bu şekildeki 4 tip uyumsuzluk grafindan bir tanesidir. Bu şekildeki 4 tip uyumsuzluk grafi 684 farklı çiftler tablosuna karşı düşen bütün örtü problemleri için geçerlidir. Bu nedenle f_m 'nin maksimum değeri 1'dir ve bu değeri alabilmesi için uyumsuzluk grafinin beşgen içermesi gerekir.

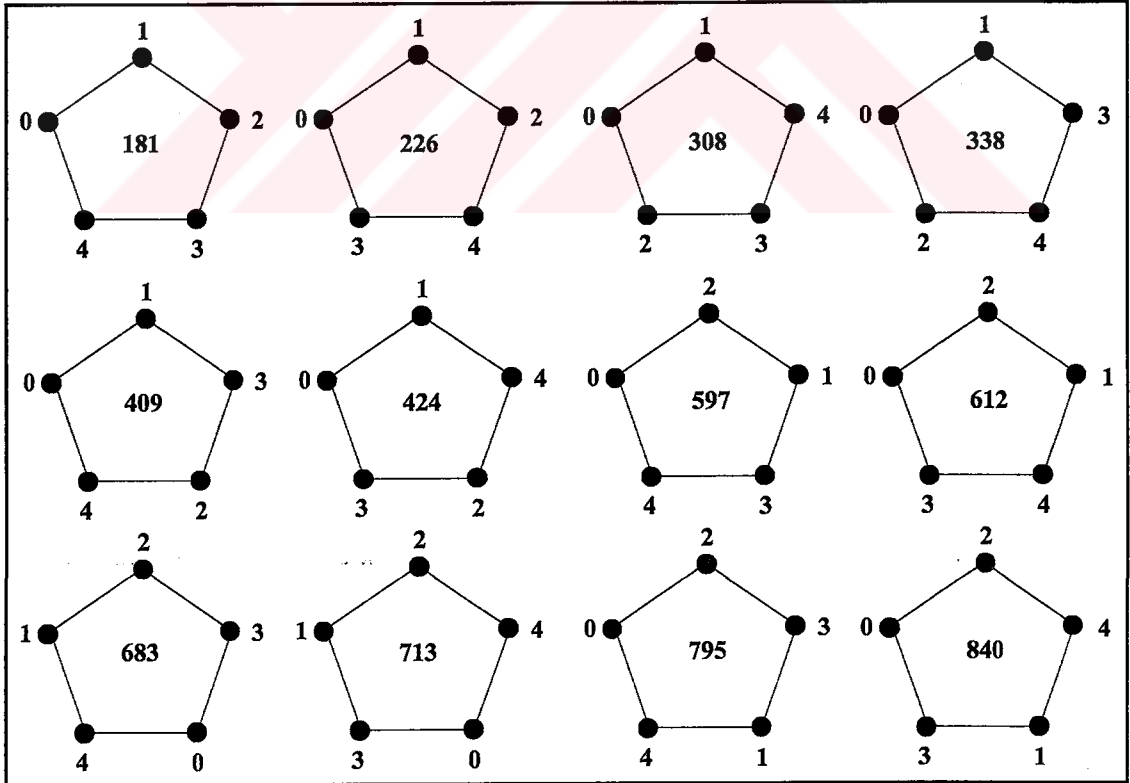
Aşağıdaki tabloda n sayısının 1'den 7'ye kadar olan değerleri için, geliştirilen bir

bilgisayar programı yardımıyla $f_m \neq 0$ olan bütün örtü problemlerinin sayısı gösterilmiştir. Eleman sayısı n artırıldığında işlem süresi çok fazla arttığından tabloyu n sayısının daha büyük değerleri için oluşturmak mümkün olmamıştır.

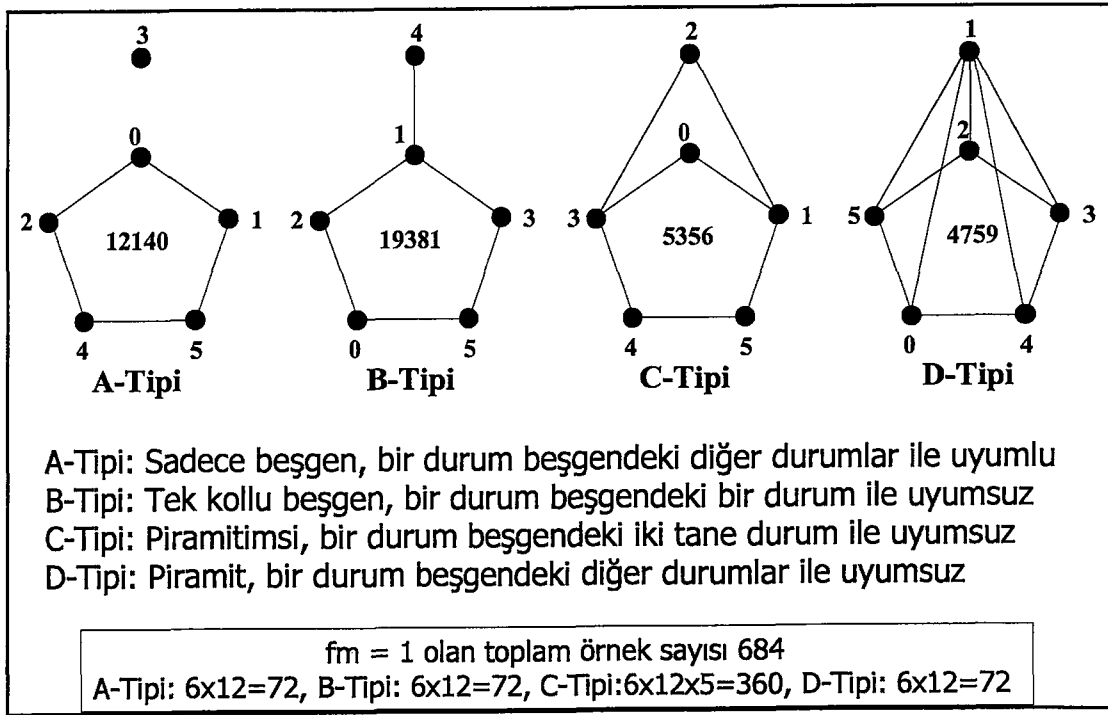
Tablo 6.1 $f_m \neq 0$ olan örtü problemlerinin sayısının eleman sayısına (n) ve m_2 'ye göre dağılımı

n	$f_m=1$	$m_2=2$	$m_2=3$	$m_2=4$	$m_2=5$	$m_2=6$	$m_2=7$
1	0	-	-	-	-	-	-
2	0	0	-	-	-	-	-
3	0	0	0	-	-	-	-
4	0	0	0	0	-	-	-
5	12	12	0	0	0	-	-
6	684	612	72	0	0	0	-
7*	12.226	2.581	9.501	144	0	0	0

* $n=7$ için verilen değerler çiftler tablosunun ilk 500.000 örtü problemi kombinezonu denenerek elde edilen değerlerdir.



Şekil 6.2 $n=5$ elemanlı $f_m=1$ olan bütün örtü problemlerinin birbirinden farklı olan uyumsuzluk grafları



Şekil 6.3 $n=6$ elemanlı $f_m=1$ olan bütün örtü problemlerinin birbirinden farklı olan uyumsuzluk grafları

Eleman sayısı n olan bir örtü probleminin uyumsuzluk grafinin beşgen, yedigen veya dokuzgen,... içermesi halinde $f_m \neq 0$ olmaktadır. Bu özellikten yararlanılarak eleman sayısı n 'nin herhangi bir değeri için $f_m \neq 0$ olan bir örtü problemi oluşturulabilmektedir.

Eleman sayısı n olan bir örtü probleminin uyumsuzluk grafinin x tane beşgen (yedigen, dokuzgen, ...) içermesi ve bu beşgenlerdeki (yedigenlerdeki, dokuzgenlerdeki,...) bir örtülen elemanın diğer beşgenlerdeki (yedigenlerdeki, dokuzgenlerdeki,...) bütün örtülen elemanlar ile uyumsuz olması halinde $f_m=x$ olmaktadır. Bu özellikten yararlanılarak eleman sayısı n 'nin herhangi bir değeri için $f_m=x$ olan bir örtü problemi oluşturulabilmektedir.

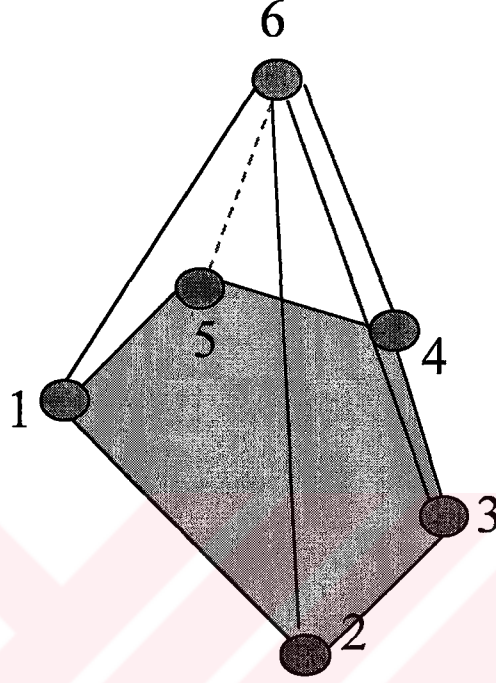
Yukarıdaki özellikler aşağıda verilen uyumsuzluk grafları ile doğrulanmıştır.

Şekil 6.4 ve Şekil 6.5'de eleman sayısı $n=6$ için ve $n=9$ için uyumsuzluk grafları verilmiştir. Bu şekillerden de görüldüğü gibi eleman sayısı n ve m_2 artsa bile $f_m=1$ olarak kalmıştır.

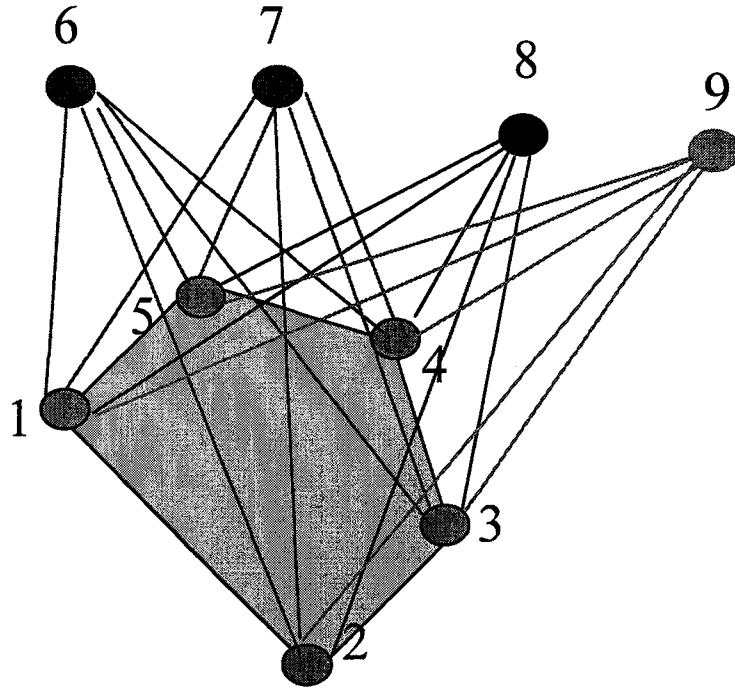
Şekil 6.6' da $n=10$ ve $m_2=2$ için $f_m=1$ olan bir örtü probleminin uyumsuzluk grafi verilmiştir. Bu şekilde gösterilen kafesteki bir beşgendeki tüm elemanlar diğer beşgendeki tüm elemanlar ile uyumsuz değildir. Örneğin 1 ve 8, 2 ve 10, 3 ve 6

elemanları birbirleri ile uyumludur.

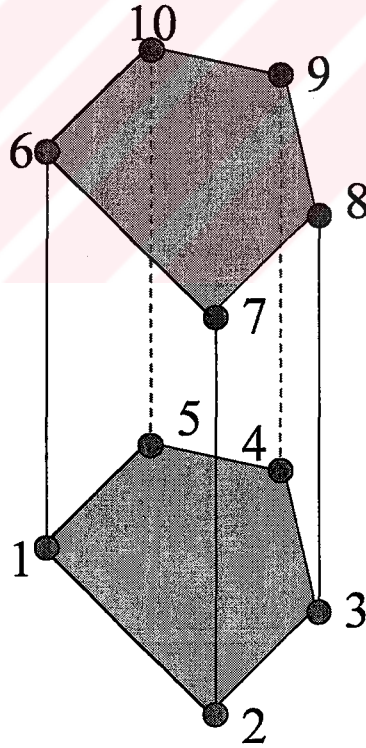
Şekil 6.7' de $n=10$ ve $m_2=4$ için $f_m=2$ olan bir örtü probleminin uyumsuzluk grafi verilmiştir. Bu şekilde gösterilen kafesteki bir beşgendeki tüm elemanlar diğer beşgendeki tüm elemanlar ile uyumsuzdur.



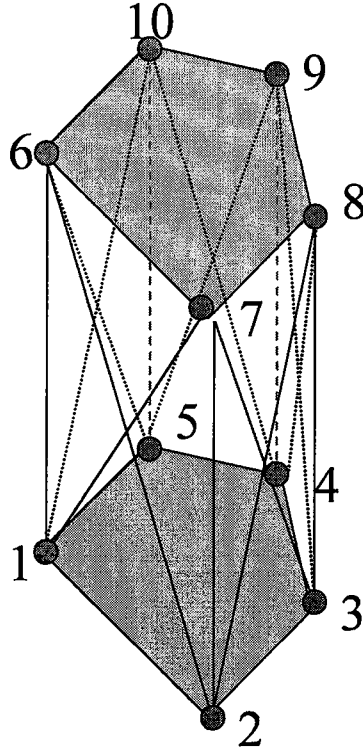
Şekil 6.4 $n=6$ ve $m_2=3$ için $f_m=1$ olan bir örtü probleminin uyumsuzluk grafi



Şekil 6.5 $n=9$ ve $m_2=3$ için $f_m=1$ olan bir örtü probleminin uyumsuzluk grafi



Şekil 6.6 $n=10$, $m_2=2$ için $f_m=1$ olan bir örtü probleminin uyumsuzluk grafi

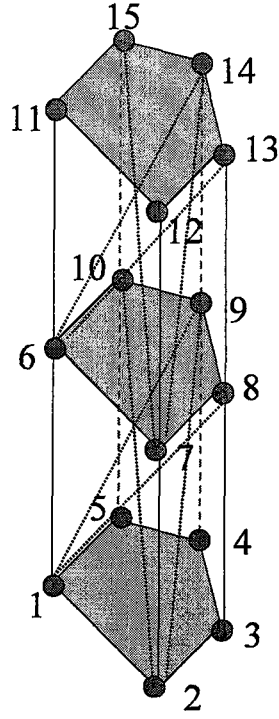


Şekil 6.7 $n=10$, $m_2=4$ ve $f_m=2$ olan bir örtü probleminin uyumsuzluk grafi

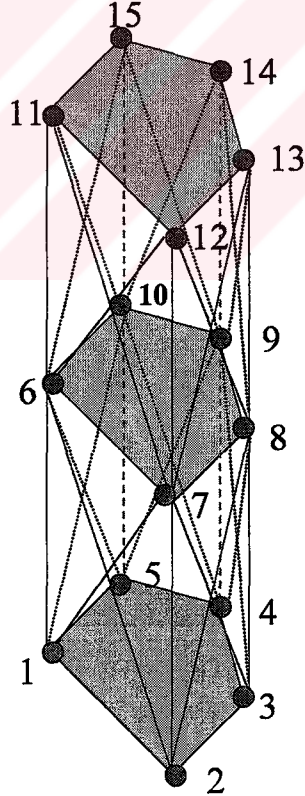
Yukarıdaki şekillerden kolayca görüldüğü gibi, f_m sayısı, kafes yapısına yeni bir beşgen ekleyerek ve bu beşgendeki bütün elemanları diğer beşgenlerdeki bütün elemanlar ile uyumsuz yaparak arttırılabilir. Buna Şekil 6.8 ve Şekil 6.9'da uyumsuzluk grafları verilen örtü problemleri de örnek olarak verilebilir.

Şekil 6.8'de $n=15$ ve $m_2=2$ için $f_m=1$ olan bir örtü probleminin uyumsuzluk grafi verilmiştir. Bu şekilde gösterilen kafesteki bir beşgendeki tüm elemanlar diğer beşgendeki tüm elemanlar ile uyumsuz değildir. Örneğin 1 ve 10, 2 ve 6, 3 ve 9 elemanları birbirleri ile uyumludur. Bu şekilde görüldüğü gibi eleman sayısı $n=15$ ve uyumsuzluk grafi 3 tane beşgen içerse bile $f_m=1$ olarak kalmaktadır.

Şekil 6.9'da $n=15$ ve $m_2=6$ için $f_m=3$ olan bir örtü probleminin uyumsuzluk grafi verilmiştir. Bu şekilde gösterilen kafesteki bir beşgendeki tüm elemanlar diğer beşgendeki tüm elemanlar ile uyumsuzdur. Bu şekilden de görüldüğü gibi Şekil 6.9'daki kafes yapısına yeni bir beşgen katı eklenerek ve yeni beşgendeki bütün elemanları kafesteki diğer beşgenlerdeki elemanlar ile uyumsuz yapılarak f_m sayısının 1 artması sağlanmıştır.



Şekil 6.8 $n=15$, $m_2=2$ için $f_m=1$ olan bir örtü probleminin uyumsuzluk grafi

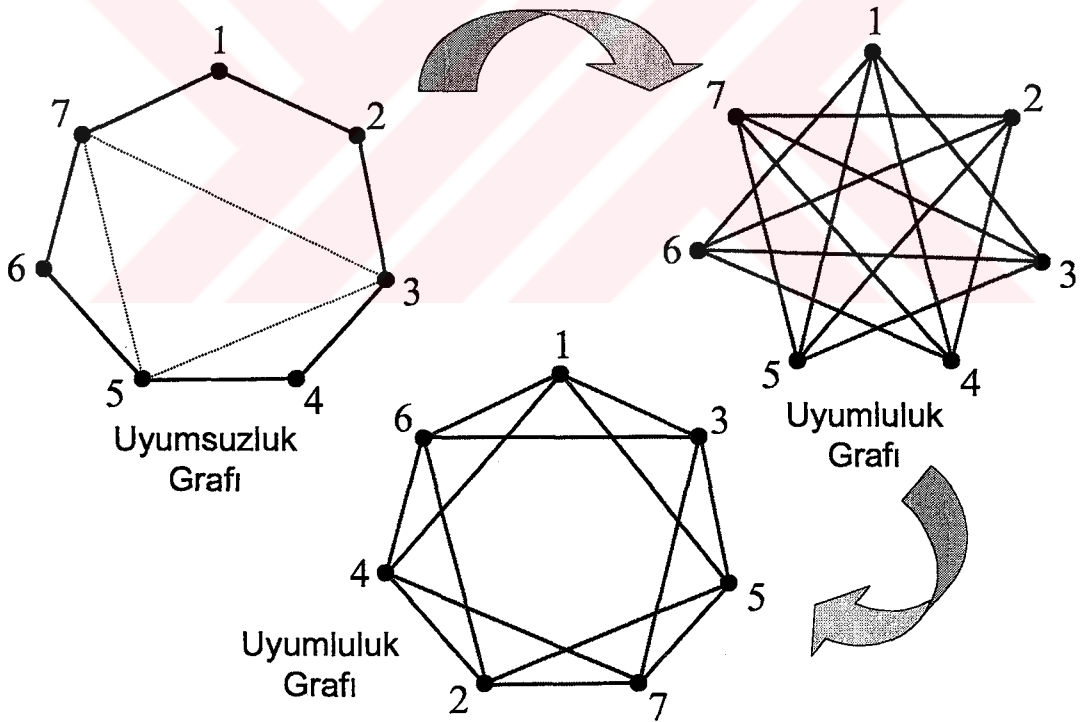


Şekil 6.9 $n=15$, $m_2=6$ için $f_m=3$ olan bir örtü probleminin uyumsuzluk grafi

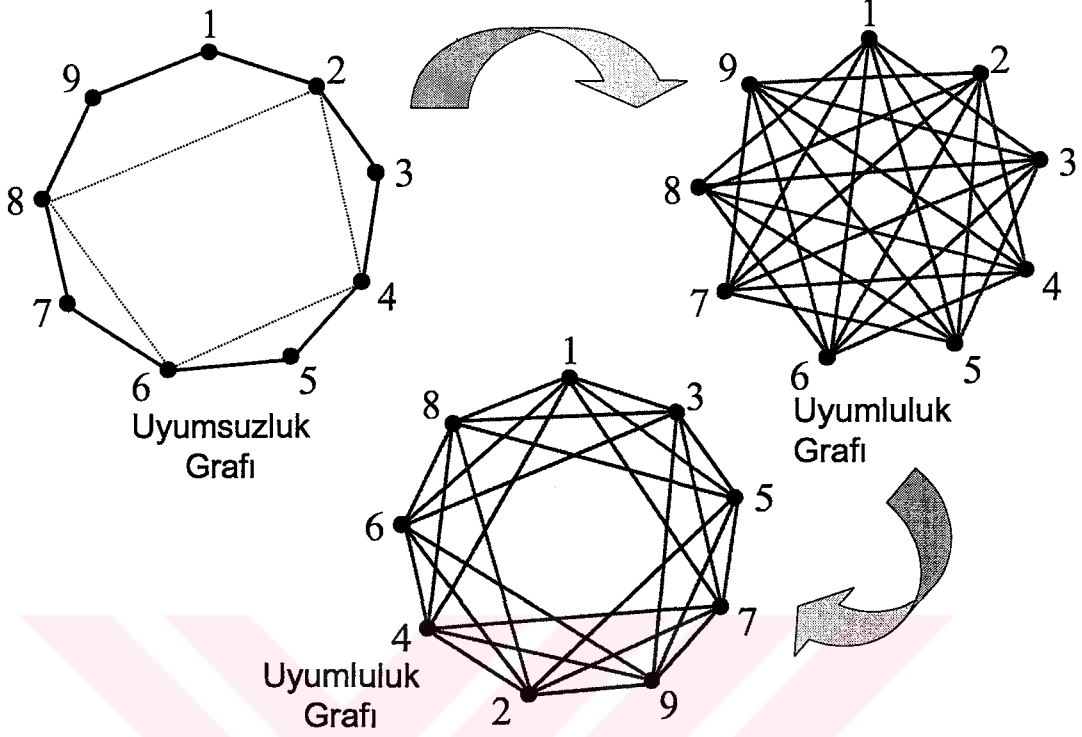
Şekil 6.10 ve Şekil 6.11'de $n=7$ ve $n=9$ için $m_2=2$ ve $f_m=1$ olan örtü problemlerinin uyumluluk grafi ve uyumsuzluk grafi verilmiştir.

Yukarıda verilen şekillerden anlaşılacağı gibi verilen bir örtü probleminin uyumsuzluk grafına bakılarak f_m sayısının dolayısıyla problemin minimal örtüsünde kaç blok olacağının belirlenmesi mümkün olabilir. Minimal örtüdeki blok sayısı $m_0 = m_2 + f_m$ olarak belirlenir.

Bir örtü probleminde f_m sayısının üst sınırının $(n \bmod 5)$ olduğu tahmin edilmektedir. Bu durumda minimal örtüdeki blok sayısının üst sınırı $(m_2 + f_m)$ olarak kolayca belirlenebilir. Daha iyi bir üst sınırın ise örtü probleminin uyumsuzluk grafindaki beşgen sayısı bulunarak elde edileceği tahmin edilmektedir. Bundan daha iyi bir üst sınırın ise örtü probleminin uyumsuzluk grafindaki birbirine dokunmayan beşgen sayısı bulunması ile elde edilebileceği tahmin edilmektedir. Bu sayede birbirlerine dokunmayan beşgenler elde edildiğinde, bu beşgenlerdeki elemanların birbirleri ile uyumsuzlukları kolayca kontrol edilerek, üst sınır yerine m_0 sayısı yani minimal örtüdeki blok sayısı bulunabileceği tahmin edilmektedir.



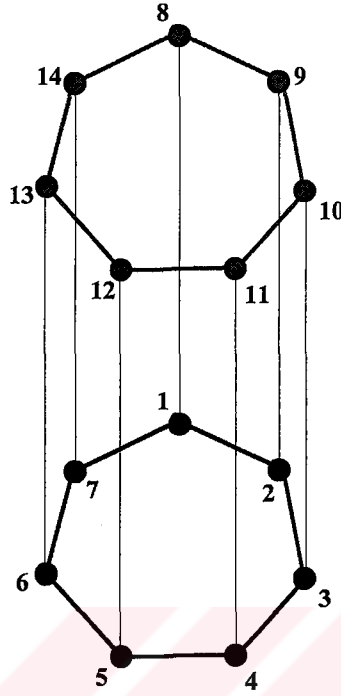
Şekil 6.10 $n=7$, $m_2=2$ için $f_m=1$ olan bir örtü probleminin uyumluluk ve uyumsuzluk grafi



Şekil 6.11 $n=9$, $m_2=2$ için $f_m=1$ olan bir örtü probleminin uyumluluk ve uyumsuzluk grafi

Şekil 6.12’de $n=14$, $m_2=2$ ve $f_m=1$ olan bir örtü probleminin uyumsuzluk grafi verilmiştir. Bu graf iki yedigen alt-graftan oluşmaktadır. Alt ve üst yedigen alt-graflar için $m_2=2$ ve $f_m=1$ ’dir. Bu durumda alt ve üst alt-graflar için ayrı ayrı yazılan kısmi Petrick fonksiyonlarında birer durum kapsanamamaktadır. Örneğin alt grafta kısmi Petrick fonksiyonu ile kapsanamayan durum 1, üst grafta kısmi Petrick fonksiyonu ile kapsanamayan durum ise 9 olsun. Bu alt grafların birleşimi ile elde edilen graf için yazılan kısmi Petrick fonksiyonunda 1 ve 9 durumları kapsanamamaktadır. Ancak 1 ve 9 durumları uyumlu olduğundan tek bir blok tarafından kapsanabilmekte ve bunun sonucunda $f_d=2$ olmasına rağmen $f_m=1$ olarak kalmaktadır. Eğer 1 ve 9 durumları uyumsuz yapılırsa, yani grafta 1 ve 9 durumları arasında bir graf elemanı tanımlanırsa 1 ve 9 durumlarının tek bir blok tarafından kapsanması engellenmiş olur. Ancak bu durumda da alt ve üst alt-graflar için yazılan kısmi Petrick fonksiyonlarında başka iki durumları içermeyecek şekilde blok seçimi yapılabilir. Yani üst graf için yazılan kısmi Petrick fonksiyonunda 14 durumunun, alt graf için yazılan kısmi Petrick fonksiyonunda ise 5 durumunun kapsanmaması

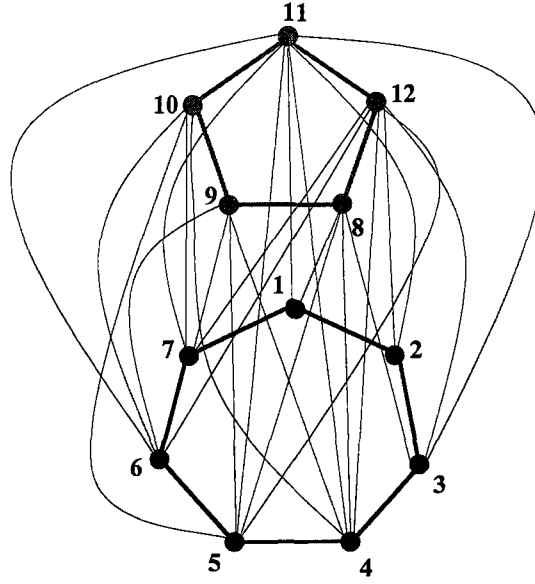
sağlanabilir.



Şekil 6.12 $n=14$, $m_2=2$ için $f_m=1$ olan bir örtü probleminin uyumsuzluk grafi

Bu durumda f_m 'nin değerini 1'den 2'ye çıkarmak için bu durumların da uyumsuz yapılarak tek bir blok ile kapsanması engellenmelidir. Bu işlemin alt ve üst alt grafların bütün durumları için yapılması gerekir.

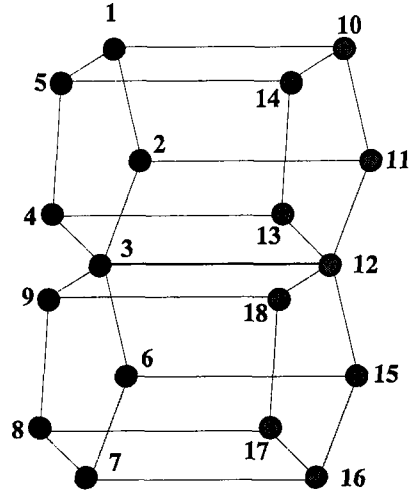
Şekil 6.13'de $n=12$, $m_2=4$ ve $f_m=2$ olan bir örtü probleminin uyumsuzluk grafi verilmiştir. Bu graf yedigen ve beşgen alt graflardan oluşmaktadır. Yedigen ve beşgen alt graflar için $m_2=2$ ve $f_m=1$ 'dir. Bu durumda alt ve üst alt graflar için ayrı ayrı yazılan kısmi Petrick fonksiyonlarında birer durum kapsanamamaktadır. Ancak daha önce bahsedildiği gibi bu alt graflardaki durumların tamamı birbirleri ile uyumsuz yapıldığından $f_m=2$ olabilmektedir.



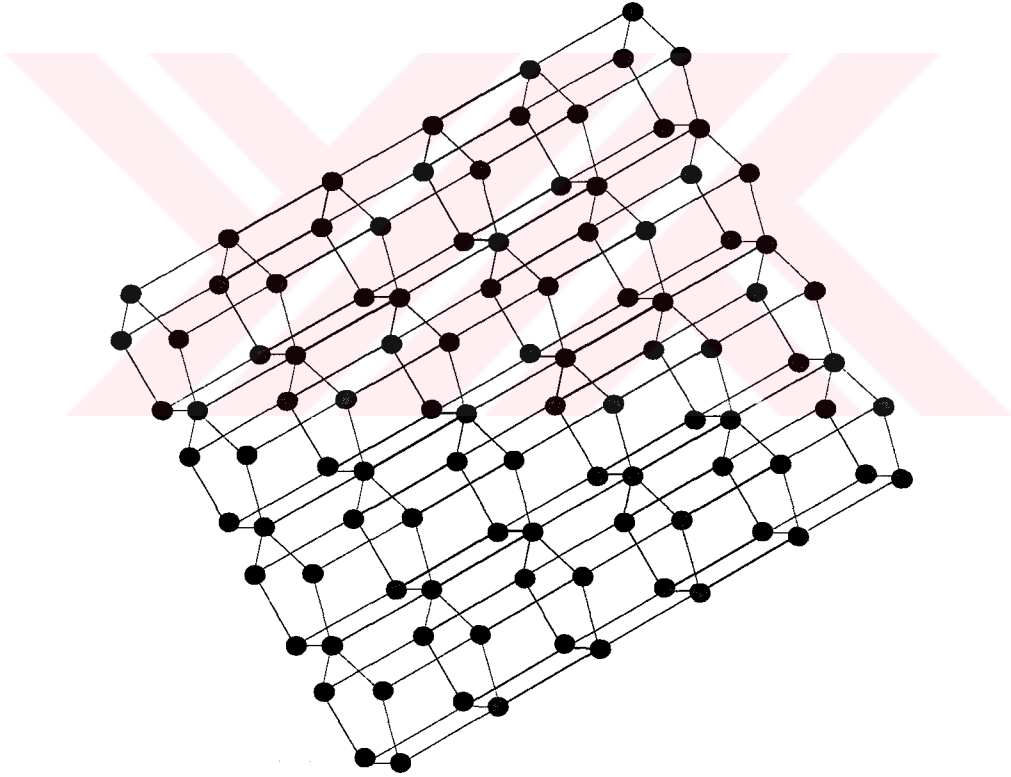
Şekil 6.13 $n=12$, $m_2=2$ için $f_m=2$ olan bir örtü probleminin uyumsuzluk grafi

Şekil 6.14’de $n=18$, $m_2=2$ ve $f_m=1$ olan bir örtü probleminin uyumsuzluk grafi verilmiştir. Bu graf dört tane beşgen alt grafdan veya bir kenarları ortak olan iki ayrı beşgen prizmasından oluşmaktadır. Bütün alt graflar için $m_2=2$ ve $f_m=1$ ’dir. Bu durumda bütün alt graflar için ayrı ayrı yazılan kısmi Petrick fonksiyonlarında birer durum kapsanamamaktadır. Ancak bu alt graflardaki durumların tamamı birbirleri ile uyumsuz yapılmadığından $f_m=1$ olabilmektedir. Bu sonuca $m_2=2$ olmasından da varılabilmektedir. f_m değerinin 1’den büyük olabilmesi için m_2 değerinin 3’den büyük olması gerekmektedir.

Şekil 6.15’de $n=102$, $m_2=2$ ve $f_m=1$ olan bir örtü probleminin uyumsuzluk grafi verilmiştir. Bu graf 24 tane beşgen alt graf içermesine rağmen $m_2=2$ olması nedeni ile f_m değeri 1’den büyük olamamaktadır.



Şekil 6.14 $n=18$, $m_2=2$ için $f_m=1$ olan bir örtü probleminin uyumsuzluk grafi



Şekil 6.15 $n=102$, $m_2=2$ için $f_m=1$ olan bir örtü probleminin uyumsuzluk grafi

6. SONUÇLAR VE TARTIŞMA

Bu Tezde kısmen belirli ardışıl makinelerde durum indirgeme konusu ele alınmıştır.

Durum indirgeme senkron ve asenkron ardışıl devre tasarımında en önemli adımlardan biridir. Durum indirgeme için geliştirilen algoritmalar, durum kodlama ve tek koşullu örtü problemi çözme gibi lojik tasarımın diğer adımlarında da kullanılabilmektedir. Bu Tezde kısmen belirli ardışıl makinelerde durum indirgeme için yeni bir yöntem geliştirilmiş; geliştirilen yöntemler lojik devre tasarımının diğer konularında da uygulanmıştır.

Bu Tezde yapılan çalışmaların ana başlıkları aşağıda verilmiştir.

- 1) Durum İndirgeme İçin Yeni Bir Yöntemin Geliştirilmesi
- 2) Durum İndirgeme İçin Geliştirilen Yöntemin Boole Fonksiyonlarının İndirgenmesine Uygulanması
- 3) Durum İndirgeme İçin Geliştirilen Yöntemin Ardışıl Devrelerin Kodlanmasına Uygulanması
- 4) Tek koşullu Örtü Problemlerinde Kullanılabilecek Yeni Bir Yöntemin Geliştirilmesi
- 5) Genel Örtü Problemlerinde Kullanılabilecek Yeni Bir Yöntemin Geliştirilmesi

Yukarıda başlıkları verilen çalışmalar ile ilgili sonuçların ayrıntıları aşağıda verilmiştir.

6.1 Durum İndirgeme İçin Yeni Bir Yöntemin Geliştirilmesi

Bu Tezde, verilen bir senkron/asenkron ardışıl makineyi indirgeyerek minimal makineyi veren yeni bir yöntem geliştirilmiştir. Bu yöntemde kullanılan teknikler Boole fonksiyonlarının indirgenmesine de uygulanmıştır. Bu teknikler tek koşullu (unate covering) veya iki koşullu (binate covering) örtü problemleri ile karşılaşılan diğer alanlarda da kullanılabilir.

Kısmen belirli ardışıl makineler için geliştirilen yöntem 2'nci bölümde verilmiştir. Bu bölümde verilen yöntemeye dayanan bir durum indirgeme programı olan SRC (State Reduction and Covering) [13] geliştirilmiş ve MCNC FSM bençmarkları üzerinde test edilerek son yıllarda yayınlanan en önemli iki durum indirgeme algoritması olan Puri ve Rho yöntemleri ile karşılaştırılmıştır. Test sonuçları Tablo 2.1'de verilmiştir. SRC programının testi için Pentium 4 1.8GHz işlemcili 256MB belleği olan bir PC kullanılmıştır.

Rho yöntemi ile karşılaştırıldığında tüm bençmarklarda SRC'nin çok daha hızlı indirgeme yapabildiği gözlenmiş, Rho yöntemi ile indirgenemeyen ex3 ve mark1 bençmarkları SRC tarafından saniyenin yüzde birinden daha kısa sürede indirgenmiştir. Ayrıca indirgenmesi zor bir bençmark olarak bilinen ex2 bençmarkında Rho yöntemi "time-out" ile sonuçlanmaktadır. Bu bençmark için minimal sonuç SRC tarafından 102s.'de elde edilmiştir. Rho yöntemi için verilen indirgeme süreleri DEC Station 5000 platformunda elde edilmiştir.

Puri yöntemi ile karşılaştırıldığında SRC'nin yine tüm bençmarklarda daha hızlı indirgeme yaptığı gözlenmiştir. Ayrıca ex2 Puri yöntemi ile test edilmemiştir. Bu bençmark için C_a kümesindeki blok sayısı 1366'dır. Puri yöntemi C_a kümesini kullanarak indirgeme yaptığı ve C_a 'daki blok sayısı yüksek olduğu için yöntemin bu bençmarkta zorlanacağı açıktır. Puri yöntemi için verilen indirgeme süreleri SUN SPARC 1+ Workstation platformunda elde edilmiştir.

6.2 Durum İndirgeme İçin Geliştirilen Yöntemlerin Boole Fonksiyonlarının İndirgenmesine Uygulanması

Durum indirgeme için geliştirilen teknikler iki seviyeli kombinezon sal devrelerin tasarımına uygulanmıştır. Tez çalışmasının bu bölümünde geliştirilen algoritma ile n değişkenli m tane Boole fonksiyonu birlikte indirgenerek minimal örtü elde edilmektedir. Bu aloritmadan yararlanarak MORP [14] geliştirilmiştir. Bu programın verdiği sonuçlar, University of California Berkeley'de geliştirilen ESPRESSO [3] programının verdiği sonuçlarla karşılaştırılmıştır.

Geliştirilen programın MCNC bençmarkları ile test sonuçları Tablo 3.3'de verilmiştir. Bu tablodan da görüldüğü gibi MORP'un iki algoritması (MORP ve MORP-EBAY), ESPRESSO ve ESPRESSO-EXACT programlarına üstünlük sağlamaktadır. MORP-EBAY, 12 bençmarkta ESPRESSO'dan daha iyi çözüm verirken, ESPRESSO sadece 3 bençmarkta MORP-EBAY'dan daha iyi çözüm vermektedir.

6.3 Durum İndirgeme İçin Geliştirilen Yöntemlerin Ardışıl Devrelerin Kodlanmasına Uygulanması

Durum indirgeme ve Boole fonksiyonlarını ortak indirgeme algoritmalarında kullanılan teknikler durum kodlama konusuna da uygulanmıştır. Tracey [12] tarafından geliştirilen minimum geçiş zamanlı durum kodlama algoritması büyük örtü problemlerinin çözülmesine ihtiyaç duymaktadır. Tracey kodlama yöntemini temel alan yeni bir durum kodlama yöntemi, durum indirgeme için geliştirilen teknikler kullanılarak oluşturulmuştur. Geliştirilen yönteme dayanan OPASKOD programı literatürde verilen bençmarklarda denenmiş, bençmarklar kritik yarış oluşmayacak şekilde ve minimum geçiş zamanlı olarak kodlanmıştır.

OPASKOD programının test sonuçları Tablo 4.1’de verilmiştir. Bu tablodan da görüldüğü gibi, Tracey yöntemi ile daha önce kodlanamayan TraceyD ve SmithM4 bençmarkları, yöntemde yapılan iyileştirmeler sayesinde kodlanabilmiştir. Bu bençmarkları Tracey tabanlı bir yöntemle, yani elde edilen makine minimum geçiş zamanlı olacak şekilde kodlayan başka bir yönteme literatürde rastlanmamıştır.

6.4 Tek koşullu Örtü Problemlerinde Kullanılabilecek Yeni Bir Yöntemin Geliştirilmesi

Bu Tezde geliştirilmiş olan iyileştirilmiş EBAY, iyileştirilmiş dallanma yöntemleri ve bu yöntemlere dayalı bilgisayar programı ile tamamen keyfi olarak oluşturulan, 1000x1000 mertebesinde bir matris ile ifade edilen bir minimal örtü problemi saniyeler mertebesinde bir sürede çözülebilmektedir. Program, Boole fonksiyonlarının VE-VEYA devreleri ile minimal gerçekleştirilmesine ilişkin çeşitli bençmarklarda denenmiş ve çözümler kısa sürede belirlenmiştir. Geliştirilen algoritmaları kullanan MORP’un verdiği sonuçlar, University of California Berkeley’de geliştirilen ESPRESSO programı ile elde edilen sonuçlarla 3’üncü bölümde Tablo 3.3’de karşılaştırılmıştır. Bu tablodan da görüldüğü gibi UCPS algoritmasını kullanan MORP, pek çok bençmarkta optimal, geriye kalan bençmarklarda ise optimale oldukça yakın sonuç vermiştir. Oysa ESPRESSO programı genelde optimal çözümü değil, optimale yakın bir çözüm vermektedir.

EBAY yöntemi, max1024 isimli bençmarkta, 298 bloklu minimale yakın bir örtü elde etmektedir. Minimal örtüde ise 259 blok bulunmaktadır. Yani EBAY yöntemi bu bençmarkta 39 blok fazlası ile (%15) minimale yakın örtüyü elde edebilmektedir. İyileştirilmiş EBAY yönteminde ise 298’li çözüm iyileştirilerek 270 bloklu çözüm elde edilmiştir. Bu durumda iyileştirilmiş EBAY yöntemi ile %11’lik bir iyileştirme

sağlanmış, minimal örtüden %4 fazla bloklu çözüm elde edilmiştir. max512 isimli bençmarkta ise EBAY yöntemi ile 153 bloklu minimele yakın bir örtü elde edilmektedir. Minimal örtüde ise 133 blok bulunmaktadır. Yani EBAY yöntemi bu bençmarkta 20 blok fazlası ile (%15) minimele yakın örtüyü elde edebilmiştir. İyileştirilmiş EBAY yönteminde ise 153'lü çözüm iyileştirilerek 137 bloklu çözüm elde edilmiştir. Bu durumda iyileştirilmiş EBAY yöntemi ile %12'lik bir iyileştirme sağlanmış, minimal örtüden %3 fazla bloklu çözüm elde edilmiştir. Diğer pek çok bençmarkta ise (5xp1, sqr6, m2, gibi) iyileştirilmiş EBAY yöntemi ile tam bir iyileştirme sağlanmış ve minimal örtü elde edilmiştir. Minimal örtünün elde edilemediği bençmarklarda, iyileştirilmiş dallanma yöntemi ile minimal örtü elde edilmiştir.

6.5 Genel Örtü Problemlerinde Kullanılabilecek Yeni Bir Yöntemin Geliştirilmesi

Tez çalışmasının bu bölümünde bir örtü problemini çözmeden minimal örtüdeki blok sayısını belirleyen bir yöntem üzerinde çalışılmıştır. Bu tür bir çalışmaya literatürde rastlanmamıştır.

Bugüne kadar geliştirilen yöntemlerden bazıları minimal örtünün alt sınırı olarak maksimal uyumsuzlar sınıfı $\overline{C_s}$ 'nin en büyük bloğu M_2 'deki eleman sayısı olan m_2 'yi kullanmıştır [1, 2, 3, 4]. Bu alt sınır pek çok örtü probleminde minimal veya minimele oldukça yakın örtüdeki blok sayısını vermektedir. Ancak bu alt sınırın hangi durumlarda minimal çözümdeki blok sayısını verdiği, hangi durumlarda vermediği, eğer vermiyorsa minimal örtüdeki blok sayısına ne kadar uzak olduğu gibi sorulara literatürde rastlanmamıştır.

Bu Tezde bir minimal örtüdeki blokların sayısını belirlemek için bir yöntem geliştirilmiştir. Minimal örtü şu şekilde elde edilebilir. En büyük maksimal uyumsuz bloktaki elemanlar için dallanma yöntemi kullanılarak veya Petrick fonksiyonu yazılarak bu elemanlar için tüm kısmi örtüler elde edilir. Eğer bu kısmi örtülerden herhangi birisi aynı zamanda bir örtü ise bu örtü bir minimal örtüdür. Bu çalışmada, en büyük maksimal uyumsuz bloğun birden fazla sayıda olduğu durumda, eğer bir tanesi ile bir minimal örtü elde edilebiliyorsa hepsinden bir minimal örtü elde edilebildiği gösterilmiştir. Eğer herhangi bir tanesinden minimal örtü elde edilemiyorsa, eksik blok sayısı nedir? Bu çalışmada m_2 ve m_6 arasındaki ilişkiler uyumluluk ve uyumsuzluk grafları kullanılarak araştırılmıştır. En büyük maksimal uyumsuz bloktaki eleman sayısı m_2 bilindiğinden, $f_m = m_6 - m_2$ belirlenebilirse, m_6 de belirlenebilir. Bazı uygulamalarda m_6 değerini bilmek yeterlidir. Ayrıca eğer m_6

deęeri bilinirse, bir minimal rty elde etmek de kolay kolaylařmaktadıř.



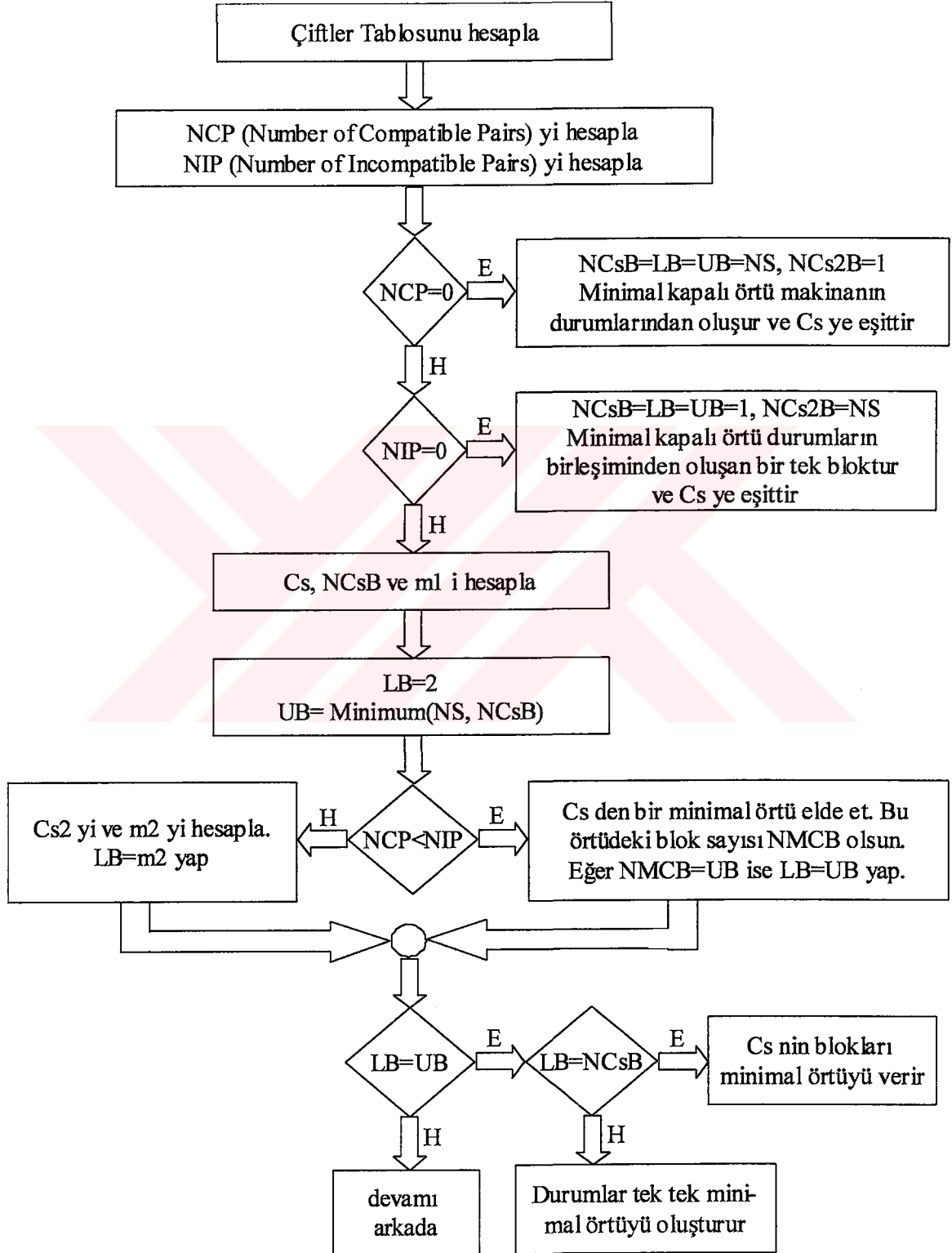
KAYNAKLAR

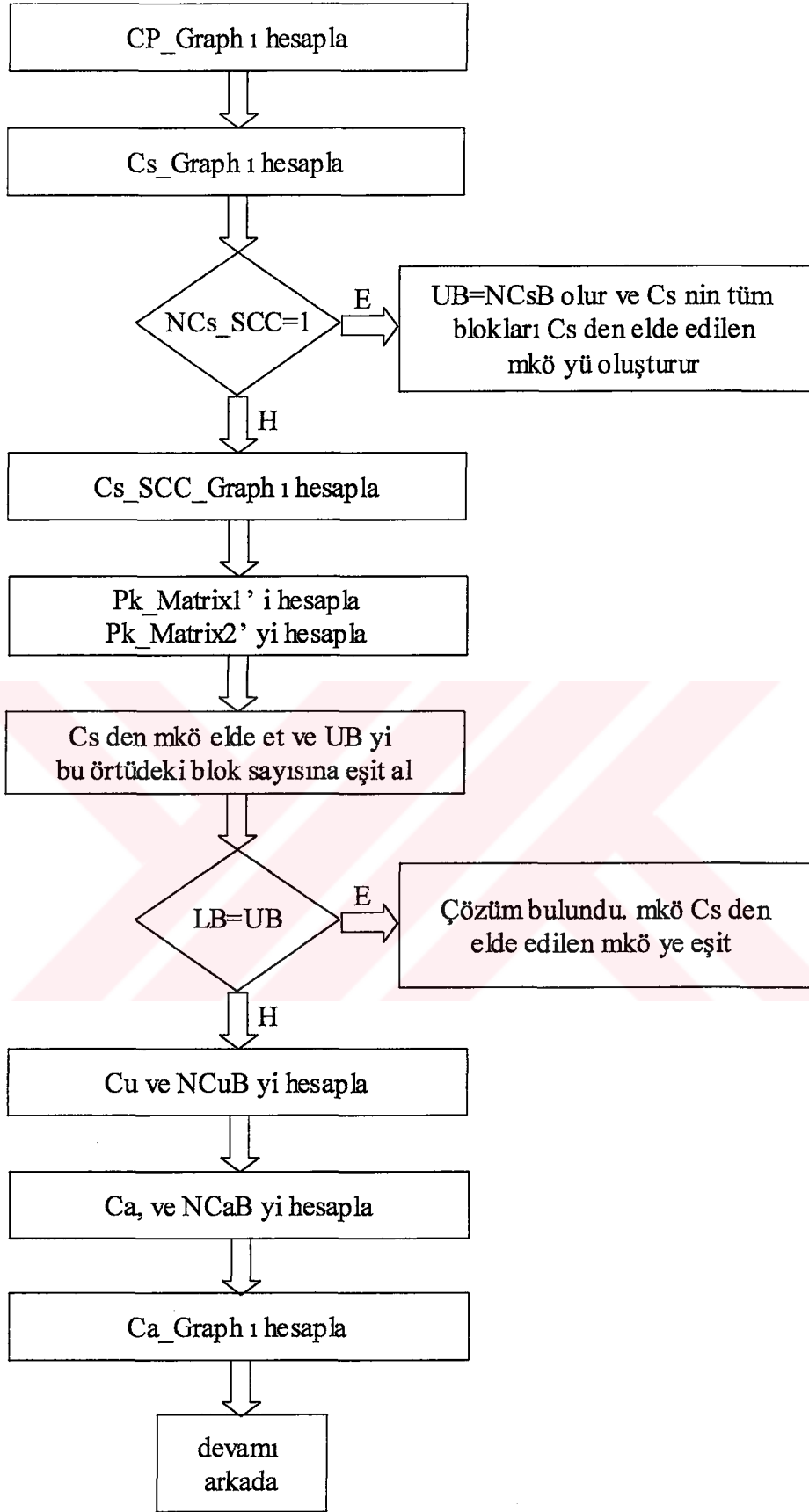
- [1] **Rho J.K., Hachtel G.D., Somenzi F. and Jacoby R.M.**,1994. Exact and Heuristic Algorithms for the Minimization of Incompletely Specified State Machines, *IEEE Transactions on CAD of Integrated Circuits and Systems*, **13**(2), 167-176.
- [2] **Puri R. and Gu J.**, 1993. An Efficient Algorithm to Search for Minimal Closed Covers in Sequential Machines, *IEEE Transactions on CAD of Integrated Circuits and Systems*, **12**(6), 737-745.
- [3] **Brayton R.K., Hachtel G.D., McMullen C.T. and Sangiovanni-Vincentelli A.L.**, 1984, Logic Minimization Algorithms for VLSI Synthesis, Kluwer Academic Publishers, Dordrecht.
- [4] **Coudert O., Madre J.C. and H. Fraisse**, 1993. A New Viewpoint on Two-Level Logic Minimization, *Proc.30th Design Automation Conference*, Dallas, TX, USA, 625-630.
- [5] **Gimpel J.F.**, 1965. A Reduction Technique for Prime Implicant Tables, *IEEE Transactions on Electronics Computers*, **EC-14**, 535-541.
- [6] **Grasselli A. and Luccio F.**, 1965. A method for minimizing the number of internal states in incompletely specified sequential networks, *IRE Transactions on Electronic Computers*, **14**(3), 350-359.
- [7] **Tarjan R.**, 1972. Depth-First Search and Linear Graph Algorithms, *SIAM J.COMPUT.*, **1**(2), 146-160.
- [8] **McCluskey E.J.**, 1986. Logic Design Principles, Prentice-Hall Inc., Englewood Cliffs, N.J.
- [9] **Hong S.J., Cain R.G. and Ostapko D.L.**, 1974. MINI: A Heuristic Approach for Logic Minimization, *IBM Journal R&D*, pp. 443-458.
- [10] **Arevalo Z. and Bredeson J.G.**, 1978. A Method of Simplify a Boolean Function into a Near Minimal Sum-of-Products for Programmable Logic Arrays, *IEEE Transactions on Computers*, Vol. C-27, No.11, pp.1028-1039.
- [11] **Coudert O., Madre J.C.**, 1992. Implicit and Incremental Computation of Primes and Essential Primes of Boolean Functions, *Proc. 29th Design Automation Conference*, Anaheim, CA, USA, June.
- [12] **Tracey J.H.**, 1966. Internal State Assignments for Asynchronous Sequential Machines, *Transactions on Electronic Computers*, **EC-18**(4).

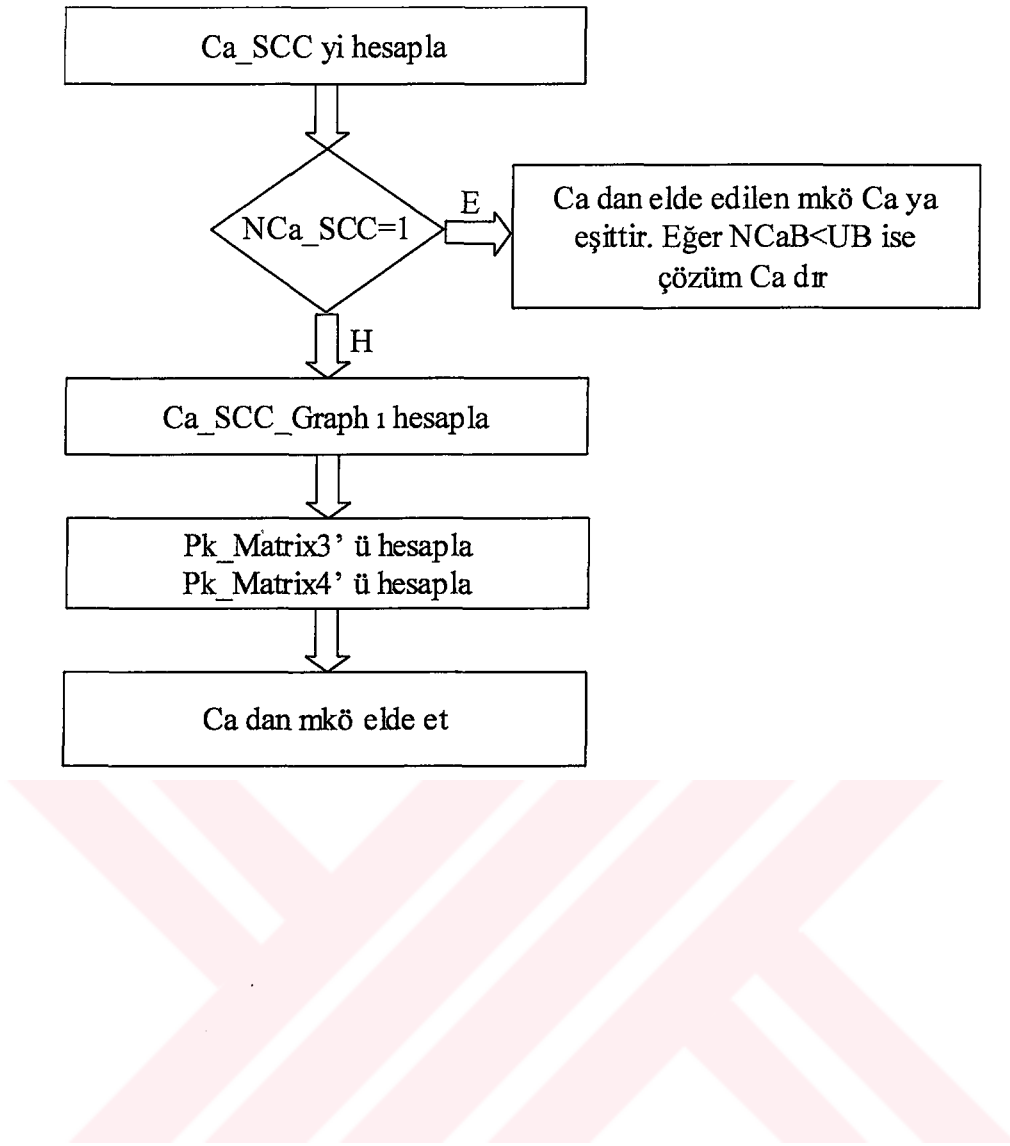
- [13] **Uçar O. and Dervişoğlu A.**, 1999. State Reduction of Incompletely Specified Finite Sequential Machines by the Use of Closed Compatible Pairs, *Proceedings ECCTD'99*, Stresa, Italy, September, 1375-1378.
- [14] **Uçar O. and Dervişoğlu A.**, 1999. Minimal Realization of a Multiple Output AND/OR Combinational Circuit, *International Conference on Electrical and Electronics Engineering (ELECO'99)*, Proceedings, Bursa, December.
- [15] **Rudell R.**, 1989. Logic synthesis for VLSI design, *PhD Thesis*, University of California, Berkeley.
- [16] **Damiani M.**, 1994. Synthesis and Optimization of Synchronous Logic Circuits, *PhD Thesis*, Stanford University.
- [17] **Slavik P.**, 1988. Approximation Algorithms for Set Cover And Related Problems, *PhD Thesis*, State University of Newyork.
- [18] **Villa T.**, 1995. Encoding Problems in Logic Synthesis, *PhD Thesis*, University of California, Berkeley.
- [19] **Kam T.**, 1995. State Minimization of Finite State Machines using Implicit Techniques, *PhD Thesis*, Electrical Engineering and Computer Sciences of the University of California at Berkeley.
- [20] **Tarjan R.**, 1977. Complexity of Combinatorial Algorithms, *Technical Report STAN-CS-77-609*, Computer Science Department, Stanford University.
- [21] **Fisher P.D. and WU S.W.**, 1993. Race-Free State Assignments for Synthesizing Large-Scale Asynchronous Sequential Logic Circuits, *IEEE Transactions on Computers*, 42(9), 1025-1034.
- [22] **Friedman A.D.**, 1986. Fundamentals of Logic Design and Switching Theory, Computer Science Press.
- [23] **Dervişoğlu A., Hacıoğlu H. and Uçar O.**, 1997. A New Method for State Assignment in Incompletely Specified Asynchronous Sequential Machines, *Proceedings ECCTD'97*, Budapest, September, 690-694.
- [24] **Dervişoğlu A.**, 1999. Lojik Devre Ders Notları, İstanbul Teknik Üniversitesi, Electric and Electronics Faculty, Istanbul.
- [25] **Lavagno L. and Sangiovanni-Vincentelli A.**, 1993. Algorithms for Synthesis and Testing of Asynchronous Circuits, Kluwer Academic Publishers.
- [26] **Kam T., Villa T., Brayton R., and Sangiovanni-Vincentelli A.**, 1994. A fully implicit algorithm for exact state minimization, *In Proceedings of the Design Automation Conference*, June, 684-690.
- [27] **Roth C.H.**, 1989. Fundamentals of Logic Design, West Publication.

- [28] **Smith R.J.**, 1974. Generation of Internal State Assignments for Large Asynchronous Sequential Machines, *Transactions on Computers*, C-23(9), 924-932.
- [29] **Wu S.F. and Fisher P.D.**, 1991. Automating the Design of Asynchronous Sequential Logic Circuits, *IEEE Journal of Solid-State Circuits*, 26(3).
- [30] **Tarjan R.**, 1976. Finding A Maximum Independent Set, *Technical Report STAN-CS-77-550, Computer Science Department, Stanford University*.
- [31] **Tarjan R.**, Finding A Maximum Clique, 1972. *Technical Report TR 72-123, Computer Science Department, Cornell University*.
- [32] **Slavik P.**, 1995. A Tight Analysis of the Greedy Algorithm for Set Cover, *Technical Report, Department of Mathematics, State University of Newyork, Newyork*.
- [33] **Brayton R.K., McGeer P.C., Sanghavi J. and Sangiovanni-Vincentelli A.L.**, 1993. A New Exact Minimizer for Two-level Logic Synthesis, *in: Logic Synthesis and Optimization*, T. Sasao (Ed.) Kluwer Academic Publishers, Dordrecht.
- [34] **Yang S.**, 1991. Logic Synthesis and Optimization Benchmarks User Guide, Microelectronics Center of North Carolina.
- [35] **Kam T., Villa T., Brayton R., and Sangiovanni-Vincentelli A.**, 1993. Implicit generation of compatibles for exact state minimization, *Technical Report No. UCB/ERL M93/60*.
- [36] **Dervişoğlu A. ve Uçar O.**, 1993. Birden Çok Boole Fonksiyonunun Birlikte İndirgenmesine İlişkin Optimale Yakın Sonuç Veren Bir Bilgisayar Programının Geliştirilmesi, *Elek.Müh. 5.Ulusal Kongresi*, Bildiriler Kitabı, Trabzon, Eylül, s. 290-295.

EK A SRC PROGRAMININ AKIŞ DİYAGRAMI

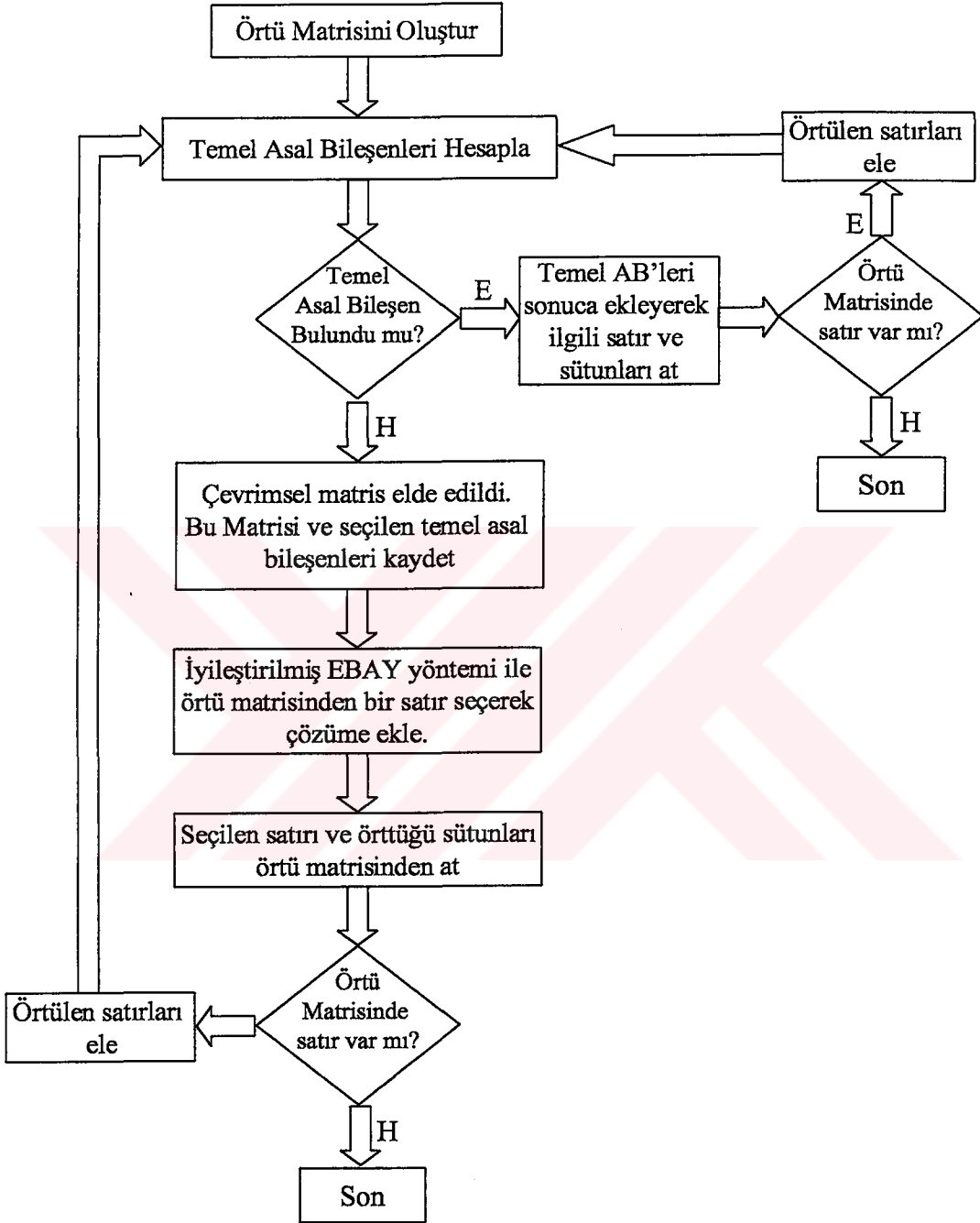




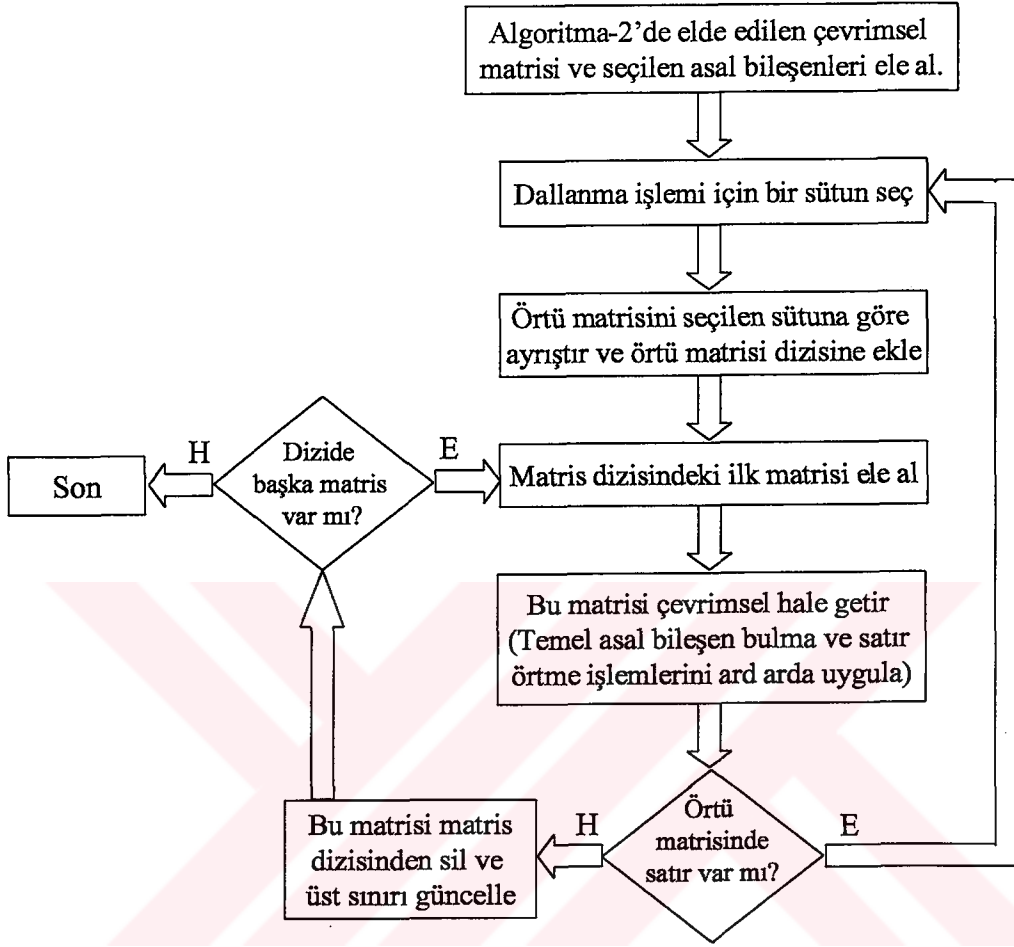


EK B MORP'UN AKIŞ DİYAGRAMI

Algoritma-2: Minimale Yakın Örtünün Hesaplanması



Algoritma-3: Minimal Örtünün Hesaplanması



ÖZGEÇMİŞ

Orhan Uçar, 05.04.1971 yılında Üsküdar'da doğdu. 1992 yılında İ.T.Ü. Elektrik-Elektronik Fakültesi, Elektronik ve Haberleşme Mühendisliği eğitimini tamamladı. 1993 ve 1997 yılları arasında aynı bölümde Devreler ve Sistemler Ana Bilim Dalında araştırma görevlisi olarak çalıştı ve 1996 yılında bu bölümde yüksek lisans eğitimini tamamladı. 1996 yılında İ.T.Ü. Fen Bilimler Enstitüsü, Elektronik ve Haberleşme Bölümünde doktora öğrenimine başladı. 1997 yılı başında Nortel Networks Netaş Araştırma ve Geliştirme bölümünde Sistem Planlama mühendisi olarak çalışmaya başladı. Halen aynı bölümdeki görevine devam etmektedir. Evli ve bir çocuk babasıdır.

