

**T.C.  
MİLLÎ EĞİTİM BAKANLIĞI**

**BİLİŞİM TEKNOLOJİLERİ**

**ARİTMETİK DEVRELER**

**Ankara, 2013**

- Bu modül, mesleki ve teknik eğitim okul/kurumlarında uygulanan Çerçeve Öğretim Programlarında yer alan yeterlikleri kazandırmaya yönelik olarak öğrencilere rehberlik etmek amacıyla hazırlanmış bireysel öğrenme materyalidir.
- Millî Eğitim Bakanlığınca ücretsiz olarak verilmiştir.
- **PARA İLE SATILMAZ.**

# İÇİNDEKİLER

AÇIKLAMALAR .....	iii
GİRİŞ .....	1
ÖĞRENME FAALİYETİ - 1 .....	3
1. KARNO HARİTALARI .....	3
1.1. Karno Haritalarının Kuralları.....	3
1.3. Üç Değişkenli Karno Haritaları .....	7
1.4. Dört Değişkenli Karno Haritaları.....	8
1.5. Karno Haritalarında İndirgeme .....	8
1.5.1. Tanımsız İfadelerin Karno Haritalarında Kullanılması .....	14
UYGULAMA FAALİYETİ .....	15
ÖLÇME VE DEĞERLENDİRME .....	16
ÖĞRENME FAALİYETİ - 2.....	18
2. TOPLAYICI DEVRELERİ.....	18
2.1. Yarım Toplayıcı .....	18
2.1.1. Yarım Toplayıcı Devrenin Tasarımı.....	18
2.2. Tam Toplayıcı.....	20
2.2.1. Tam Toplayıcı Devresinin Tasarımı .....	20
2.3. Toplayıcı Entegresi .....	23
2.3.1. Paralel Toplama.....	23
2.3.2. Toplayıcı Entegresi İle İkilik Sistemde Dört Bitlik İki Sayıyı Toplama .....	23
UYGULAMA FAALİYETİ .....	25
ÖLÇME VE DEĞERLENDİRME .....	27
ÖĞRENME FAALİYETİ - 3 .....	28
3. ÇIKARICI DEVRELERİ.....	28
3.1. Yarım Çıkarıcı .....	28
3.1.1. Yarım Çıkışı Devre Tasarımı.....	28
3.2. Tam Çıkarıcı .....	30
3.2.1. Tam Çıkarıcı Devrenin Tasarımı .....	30
3.3. Birin Tümleyenine Göre Çıkarma.....	32
3.3.1. Birin Tümleyenine Göre Dört Bitlik İki Sayıyı Çıkaran Devrenin Tasarımı .....	36
3.3.2. Çıkarıcı Entegresi .....	37
UYGULAMA FAALİYETİ .....	40
ÖLÇME VE DEĞERLENDİRME .....	41
ÖĞRENME FAALİYETİ - 4.....	43
4. ÇARPMA DEVRELERİ.....	43
4.1. Çarpma Devresi (Multiplier).....	43
4.1.1. Bir Bitlik İki Sayıyı Çarpan Devre .....	43
4.1.2. İki Bitlik İki Sayıyı Çarpan Devre.....	44
4.2. Çarpma Entegresi.....	50
UYGULAMA FAALİYETİ .....	52
ÖLÇME VE DEĞERLENDİRME .....	53
ÖĞRENME FAALİYETİ - 5 .....	54
5. KARŞILAŞTIRMA DEVRELERİ .....	54
5.1. Yarım Karşılaştırmacı.....	54

---

5.1.1. Yarım Karşılaştırıcı Devrenin Tasarımı .....	54
5.2. Tam Karşılaştırıcı.....	55
5.2.1. Tam Karşılaştırıcı Devrenin Tasarımı .....	55
5.3. Karşılaştırıcı Entegreleri .....	57
UYGULAMA FAALİYETİ .....	59
ÖLÇME VE DEĞERLENDİRME .....	60
MODÜL DEĞERLENDİRME .....	61
CEVAP ANAHTARLARI .....	64
KAYNAKÇA .....	68

# AÇIKLAMALAR

<b>ALAN</b>	<b>Bilişim Teknolojileri</b>
<b>DAL/MESLEK</b>	<b>Bilgisayar Teknik Servisi</b>
<b>MODÜLÜN ADI</b>	<b>Aritmetik Devreler</b>
<b>MODÜLÜN TANIMI</b>	Lojik devreler ve tümleşik devreler modüllerinin devamı niteliğindedir. Lojik devrelerin Karno Haritası yöntemiyle indirgenmesini ve aritmetik devrelerin kullanılmasını öğreten bir öğrenme materyalidir.
<b>SÜRE</b>	40/32
<b>ÖN KOŞUL</b>	“Tümleşik Devreler” modülünü tamamlamış olmak
<b>YETERLİK</b>	Aritmetik devreler ile çalışma yapmak
<b>MODÜLÜN AMACI</b>	<b>Genel Amaç</b> Aritmetik devre uygulamalarını yapabileceksiniz. <b>Amaçlar</b> <ol style="list-style-type: none"><li>1. Mantık devrelerinde sadeleştirme işlemleri yapabileceksiniz.</li><li>2. Toplayıcı devreleri uygulamaları yapabileceksiniz.</li><li>3. Çıkarıcı devreleri uygulamaları yapabileceksiniz.</li><li>4. Çarpma devreleri uygulamaları yapabileceksiniz.</li><li>5. Karşılaştırma devreleri uygulamaları yapabileceksiniz.</li></ol>
<b>EĞİTİM ÖĞRETİM ORTAMLARI VE DONANIMLARI</b>	<b>Ortam:</b> Elektronik laboratuvarları, dijital elektronik laboratuvarları, mikro işlemci laboratuvarları, bilgisayar laboratuvarları <b>Donanım:</b> Doğru akım güç kaynağı, kablo, el takımları, ttl veya cmos toplayıcı entegresi,ttl veya cmos çarpma entegresi, ttl veya cmos karşılaştırıcı entegresi, ttl veya cmos kapı entegreleri, led, osilaskop, lojik eğitim seti (breadboard, sekiz tane ledve anahtar), bilgisayar ve elektronik devre tasarım yazılımları
<b>ÖLÇME VE DEĞERLENDİRME</b>	Her faaliyet sonrasında o faaliyetle ilgili değerlendirme soruları ile kendi kendinizi değerlendireceksiniz. Modül sonunda uygulanacak ölçme araçları ile modül uygulamalarında kazandığınız bilgi ve beceriler ölçülerek değerlendirilecektir.



# GİRİŞ

**Sevgili Öğrenci,**

Günümüzde yarı iletkenlerin ucuzlaması, üretim tekniklerinin hızlanması; dijital devrelerin, hassas çalışması, az yer kaplaması ve az güç harcaması gibi özellikler dijital elektronik sistemlerin kullanım sahasını oldukça genişletmiştir. Dijital elektronik devrelerle tasarlanmış başta bilgisayarlar olmak üzere, trafik sinyalizasyon sistemleri, cep telefonları, saatler, hesap makineleri, fotoğraf makineleri, kameralar, akıllı aletler vb. cihazlar toplum yaşamını kolaylaştırması açısından büyük önem taşımaktadır.

Bu modülde göreceğiniz temel matematiksel işlemleri yapan aritmetik devrelerin kullanılmasıyla mikro işlemcilerin olmazsa olmaz ünitesi olan aritmetik lojik birimini (AritmeticLogicUnit) tanımış olacaksınız.





# ÖĞRENME FAALİYETİ-1

## AMAÇ

Karno haritalarını kullanarak mantık devrelerinde sadeleştirme yapabileceksiniz.

## ARAŞTIRMA

- Karno haritaları hakkında araştırma yapınız.
- Doğruluk tablosu (truth table), toplam terim (Maxterm), çarpım terim (minterm), çarpımların toplamı (sop), toplamların çarpımı (pos), giriş-çıkış değişkeni, dijital elektronikte tanımsız ifade (fi) terimlerini araştırınız.
- İkilik sistemde toplama işleminin nasıl yapıldığını araştırınız.
- Toplama işlemini gerçekleştirebilecek entegreleri araştırınız.

## 1. KARNO HARİTALARI

Dijital elektronikte en önemli kural az maliyetli devreler oluşturmaktır. Bunu yapabilmek için sadeleştirme yöntemleri kullanılır. Devreler en sade şekilde yani gereksiz devre elemanları kullanılmadan oluşturulmuş olmalıdır. Sadeleştirme işlemleri Boolean Matematiği ve Karno Haritaları yöntemleriyle yapılır. Karno Haritaları (K-maps) yöntemi, Boolean Matematiğine göre daha kolay ve güvenilir olduğundan dijital devre tasarlanırken daha çok tercih edilir.

Karno Haritalarının temel mantığını kavradıktan sonra sadeleştirme işlemleri çok basit olarak yapabileceksiniz.


### 1.1. Karno Haritalarının Kuralları

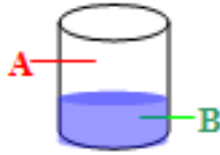
- Karno Haritaları karelerden oluşur. Her bir kare standart çarpım terimi (minterm-mi) ya da standart toplam terimi (maxterm-mi) gösterir. Hangi terimin kullanılacağını devreyi tasarlayacak kişi belirler.
- Tablonuzun kaç kareden oluşacağını tespit etmek için giriş değişken sayısını bilmeniz gerekir.  $N = \text{giriş değişkeni}$ , Kare sayısı  $= 2^N$  formülüyle hesaplanır. Her bir kareye sıfırdan başlayıp soldan sağa doğru numara verilir, satır bitiminde bir alttaki satırın solundan sağa doğru numara vermeye devam edilir.
- Çıkış fonksiyonu genelde aktif lojik 1 seviyesine (çarpım terime) göre belirlenir ama devreyi tasarlayan kişi isterse aktif lojik 0 seviyesine (toplam terime) göre de devreyi tasarlayabilir. Eğer lojik 1 aktif kabul edildiyse, hedef en çok 1'i gruplamaktır. Kutuların içindeki 1'ler dikkate alınır. Boş olan kutular 0 (sıfır)

demektir ve sıfırlar grupta dışında tutulmalıdır. Eğer lojik 0 aktifse; hedef en çok 0'ı gruplamaktır. Kutuların içindeki 0'lar dikkate alınır. Boş olan kutular bir demektir ve birler grupta dışında tutulmalıdır. Modülde göreceğiniz örneklerin çoğu aktif lojik 1 seviyesine göre yapılmıştır.

- Gruplamalarda seçilen kutu sayısı 1,2,4,8,16,...olmalıdır. Bu sayı 3,5,6,7,9,10,15,...olamaz.
- Karno haritaları üzerinde çapraz gruplama yapılamaz. Gruplar alt alta ya da yan yana olmalıdır. Gruplamalar ayrıntılı bir şekilde 1.1.5 numaralı başlık altında anlatılmıştır.
- Karno haritalarının kutuları doldurulurken doğruluk tablolarından; kanonik SOP formdan ve ya kanonik POS formdan yararlanılır.


## 1.2. İki Değişkenli Karno Haritaları

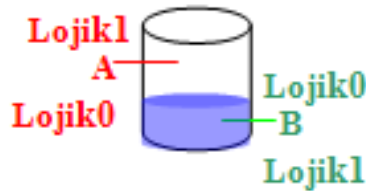
ALARM 



**Örnek 1:** Bir su tankının seviyelerini kontrol eden A ve B olmak üzere iki adet algılayıcı sensör bulunmaktadır. A sensörü su ile temas ettiğinde, B sensörü ise su ile temas etmediğinde alarmı çalıştırmaktadır. Bu devrenin doğruluk tablosunu ve Karno haritasını oluşturun ve çıkış fonksiyonunu yazın.

**Çözüm**

ALARM 



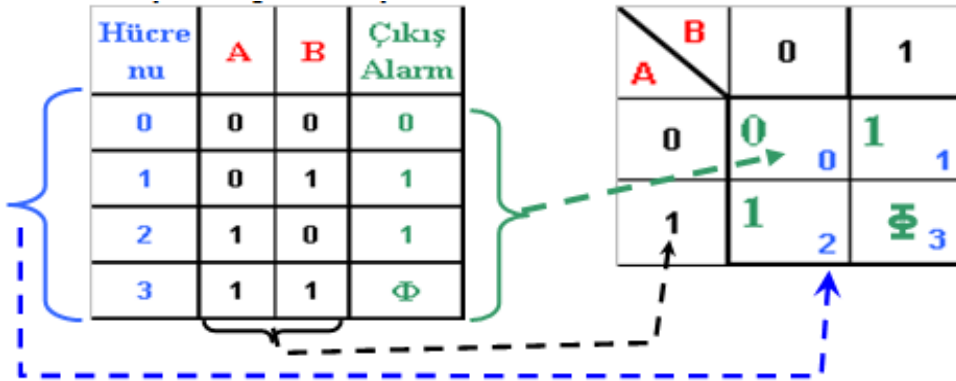
- Alarmın lojik olarak hangi seviyede aktif olacak belirleyeceğiz. Lojik 1 seviyesinde aktif olduğunu kabul edilmiştir.

- Doğruluk tablosunu çizmeden önce devrenin kaç giriş elemanın olduğunu belirleyeceğiz. Bu soruda A ve B olmak üzere iki tane giriş değişkenimiz vardır.
- Doğruluk tablosunu çizeceğiz. Doğruluk tablosunu çizerken su seviyesinin olasılıklarına göre alarmın çalıp çalmayacağı belirlenecek.
  - Birinci olasılıkta su iki sensörün arasında bir yerdeyse yani  $a=0$  ve  $b=0$  iken alarm çalışmaz ( $\text{alarm}=0$ ).
  - İkinci olasılıkta tankta su kalmamışsa yani  $b=1$  ve  $a=0$  iken alarm çalışır ( $\text{alarm}=1$ ).
  - Üçüncü olasılıkta tanktaki su taşmak üzereyse  $a=1$  ve  $b=0$  iken alarm çalışır ( $\text{alarm}=1$ ).
  - Dördüncü ve son olasılıkta tank hem taşmak üzereyedir  $a=1$  hem de tankta su kalmamış  $b=1$  gibi gözükmetedir. Bu son durum yer çekimine göre olamayacağından ötürü bu durum tanımsız ( $f(\Phi)$ ) olarak değerlendirilecektir.
- Doğruluk tablosu çizildikten sonra giriş değişkeni  $N=2$  olduğundan Karno haritası  $2^2=4$  elemanlı olacaktır.

A	B	Çıkış Alarm
0	0	0
0	1	1
1	0	1
1	1	$\Phi$

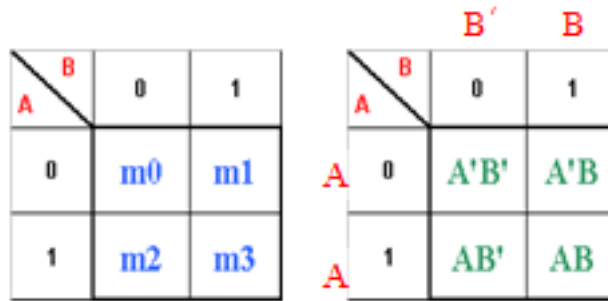
**Şekil 1.1: Örnek 1'in doğruluk tablosu**

- Doğruluk tablosuna göre Karno haritasının oluşturulması aşağıda verilmiştir. Renkler tabloya göre uyumlu verilmiştir.
  - Mavi değerler a ve b ikilik sistemdeki giriş bilgilerinin onluk sistemdeki karşılığını ve aynı zamanda tablo içindeki kare (minterm) numarasını gösterir.
  - Yeşil değerler çıkış fonksiyonunu ve alarmın hangi zamanlarda aktif olduğunu gösteriyor.
  - Siyah olan sayılar giriş değişkenlerinin durumlarını gösteriyor.
  - Karno haritasının karelerini, giriş değişkenleri arasındaki ilişki aşağıdaki şekilde gösterilmiştir.



Şekil 1.2: İki değişkenli doğruluk tablosunu karno haritasına yerleştirme

- Sorunun başında devremizi aktif lojik1 seviyesinde kabul ettiğimiz için Şekil 1.2'deki (yeşil olan sayılarda) alarmın bir olduğu yerleri kullanmamız gerekecek (karno kuralları üçüncü madde). Lojik1 seviyesinde giriş değişkenin bir olması kendisini (tabloda siyahla yazılmış olan sayılar), sıfır olması değişkenin değil anlamına gelir. A'nın bir olduğu yerler değişkenin kendisi (A) , A'nın sıfır olduğu yerler değişkenin değil ( $\bar{A}$  veya  $A'$ ) olarak gösterilir.
- Karno haritalarında mintermleri göstermek amacıyla aşağıdaki şekilde iki değişkenli karno haritaları gösterilmiştir.



Şekil 1.3: İki değişkenli haritanın mintermle gösterimi



Şekil 1.4: Çözümlü sorunun şekil 1.3'e göre düzenlenmiş hali

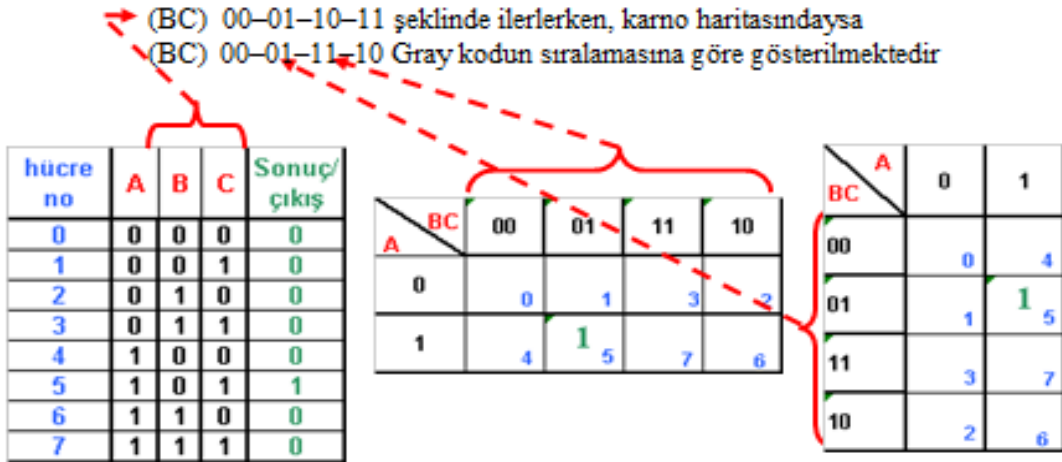
- Şekil 1.4'teki her bir kutunun içindeki çarpım şeklindeki ifadelerin arasına toplam sembolü koyarak çıkış fonksiyonumuzu  $F(\text{Alarm})=AB'+A'B$  olarak elde etmiş oluruz.
- ( $\Phi$ ) ile gösterilen ve normalde istenmeyen ya da gerçekleşmesi mümkün olmayan durumlar, sonucu etkilemeyeceğinden karno haritalarında çıkış fonksiyonuna aktif çıkış olarak dahil edilebilir. Karno haritalarında tanımsız ( $\Phi$ ) ifadelerin çıkış fonksiyonunun en az maliyetle indirgenmesine nasıl yardımcı olabileceği 1.1.5.1. numaralı konuda anlatılacaktır.

### 1.3. Üç Değişkenli Karno Haritaları

Üç tane giriş değişkeni  $N=3$  için  $2^3=8$  kareden oluşmaktadır. İki değişkenli karno haritalarındaki bütün kurallar üç değişkenli içinde geçerlidir. Şekil 1.5'teki gibi üç değişkenli haritalar iki farklı şekilde gösterilebilir.

Şekil 1.5'teki doğruluk tablosundaki aktif lojik 1'e göre rastgele sonuçlar verilmiştir. Sonuçtaki sıfırlar kullanılmayacağından sadece sonuçtaki birler haritaya yerleştirilmiştir.

Üç değişkenli karno haritaları iki değişkenli haritalara benzer şekilde doldurulmaktadır. Bunun yanında dikkat edilmesi gerek önemli bir fark vardır. İkilik sistemde sayılar doğruluk tablosunda:



Şekil 1.5: Üç değişkenli doğruluk tablosu karno haritası

Haritadaki sıralamada bir değerden diğerine geçerken her seferinde aynı anda sadece bir bit'in yerinin değişmiş olması (gray koda göre haritanın düzenlenmesi) gerekmektedir. Bunun nedeni karno haritalarında en doğru ve sade sadeleştirilmenin gerçekleştirilmek istenmesidir.

Verilen tablodaki devrenin çıkışında sadece beş numaralı kare (m5) aktif sonuç vermiştir. Bu karede  $A=1, B=0, C=1$ 'dir. Buna göre bu devrenin fonksiyonu  $F=AB'C$  olur.

## 1.4. Dört Değişkenli Karno Haritaları

Dört giriş değişkenli karno haritalarında iki değişkenli haritalarda ve üç değişkenli haritalarda anlatılan bütün özellikler geçerlidir. Özellikle de üç değişkenli haritada anlatılan aynı anda sadece bir bit'in değerinin değişmesi (gray koda göre sıralama) kuralı!

Dört tane giriş değişkeni  $N=4$  için  $2^4=16$  kareden oluşmaktadır. Şekil 1.6'daki gibi dört değişkenli haritalar da iki farklı şekilde gösterilebilir.

hücre no	A	B	C	D	Sonuç/çıkış
0	0	0	0	0	0
1	0	0	0	1	0
2	0	0	1	0	0
3	0	0	1	1	1
4	0	1	0	0	0
5	0	1	0	1	0
6	0	1	1	0	0
7	0	1	1	1	0
8	1	0	0	0	0
9	1	0	0	1	0
10	1	0	1	0	0
11	1	0	1	1	0
12	1	1	0	0	1
13	1	1	0	1	0
14	1	1	1	0	0
15	1	1	1	1	0

AB \ CD	00	01	11	10
00	0	1	1	2
01	4	5	7	6
11	1	12	13	14
10	8	9	11	10

CD \ AB	00	01	11	10
00	0	4	1	8
01	1	5	13	9
11	1	3	7	15
10	2	6	14	10

Şekil 1.6: Dört değişkenli doğruluk tablosu karno haritası

Şekil 1,6'daki doğruluk tablosundaki aktif lojik1'e göre rastgele sonuçlar verilmiştir. Bu devrenin fonksiyonu  $f= m3+m12$  yani  $F= A'B'CD+ ABC'D'$  olur.

(0011) (1100)

## 1.5. Karno Haritalarında İndirgeme

Karno haritalarının boolean matematiğine göre daha kolay ve güvenilirdir. Buraya kadar karno haritalarının nasıl oluşturulduğunu ve içinin nasıl doldurulduğunu gördünüz.

Sadeleştirme yapmadan önce aktif lojik seviyesine göre kareler kendi arasında gruplandırılır. Devre aktif lojik1'e göre ise haritadaki birler kendi aralarında gruplanır, sıfırlar kullanılmaz. Devre aktif lojik0'a göre ise haritadaki sıfırlar kendi aralarında gruplanır, birler kullanılmaz.

Gruplamadaki en önemli ikinci nokta en fazla grubu bulup değişken ve kapı sayısını azaltmaktır.

- İki kare grup oluşturmuşsa bir değişken yok olur.
- Dört kare grup oluşturmuşsa iki değişken yok olur.
- Sekiz kare grup oluşturmuşsa üç değişken yok olur.
- 16 kare grup oluşturmuşsa dört değişken yok olur.
- Bir eleman hiçbir gruba katılmıyorsa tek olarak işleme tabi tutulur.

Vereceğimiz örneklerde boolean ifadeleri kullanarak karno haritasının oluşturmayı da görmüş olacaksınız.

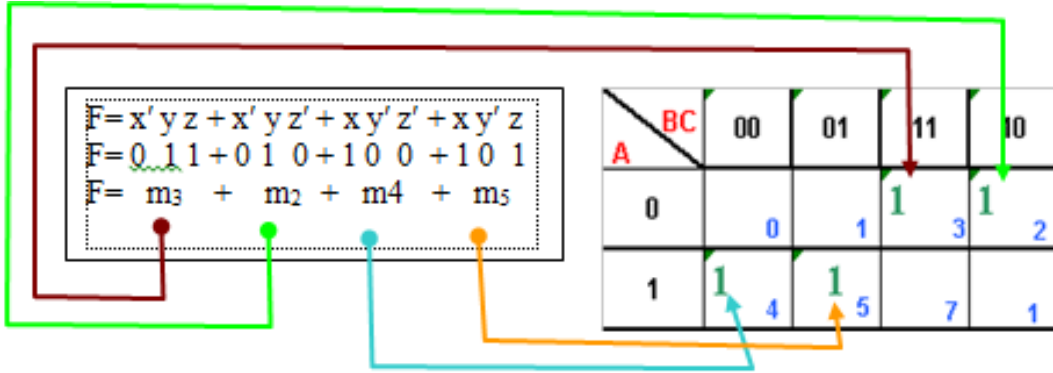
**Örnek 2:**  $F = x'yz + x'yz' + xy'z' + xyz$  boolean fonksiyonunu karno haritası kullanarak sadeleştirin.

- Birinci adım olarak karno haritasının kaç değişkenli olduğu bulunmalıdır. Soruya bakıldığında x, y ve z haricinde başka giriş değişkeni olmadığına göre bu devrede  $N=3$ 'tür diyebiliriz. Yani çizeceğimiz karno haritası  $2^3=8$  kareden oluşacaktır.
- İkinci adım üç değişkenli karno haritası çizilir.

A \ BC	00	01	11	10
0	0	1	3	2
1	4	5	7	6

**Şekil 1.7: Üç değişkenli karno haritası**

- Üçüncü adım devremizin aktif lojik 1 mi yoksa lojik 0 mı olacağını bilmemiz gerekmektedir. Verilen ifade çarpımların toplamı (kanonik sop form) olduğundan bu devrenin lojik 1'de aktif olduğu anlaşılır.
- Dördüncü adım karno haritasının aktif çıkışlara göre iinin doldurulmasıdır. Bunun için verilen her bir ifadenin kaç numaralı (kare) ya da çarpım terim (minterm) olduğunu bulacağız. Örneğin  $x'yz$  ifadesine karşılık gelen ikili sayı 011'dir ve 011 karesindeki m3 minterminin çıkışta aktif yani 1 olduğunu temsil eder.
- Haritada yan yana olan birler kendi aralarında ikili gruplamaya örnektirler. Aşağıda haritanın gruplandırılmış hali bulunmaktadır.



Şekil 1.8: Boolean ifadenin minterme çevrilmesi ve karno haritasına aktarma

- Soruda bu fonksiyon  $F = \sum m(2,3,4,5)$  şeklinde de verilebilirdi. (Kanonik SOP formun kalıp gösterimi  $F_{CSOP} = \sum(m(\dots)) + \Phi(\dots)$ ).  $\sum$  (sigma) işareti matematikte birden fazla toplamı temsil etmek (hesaplatmak) için kullanılan bir semboldür.  $\sum m(2,3,4,5) = m_2 + m_3 + m_4 + m_5$ 'dir.
- Grupların tanımları çıkarılırken, grubun kapsadığı kutularda değişiklik göstermeyen değişkenler alınır. Değişiklik gösteren değişkenler etkisiz sayılır örneğin Şekil 1.8'deki  $m_3$  ve  $m_2$  mintermlerine bakıldığında: A değişkeni ikisinde de sıfırdır. İki terimde de farklılık gösterilmediği ve ikisinde de sıfır olduğu için A değişkeninin değili kullanılacaktır. B değişkenine baktığımızda iki terimde de bir olduğu görülür. İki karede de aynı ve bir olduğundan B değişkenin kendisi kullanılacaktır. C değişkenine bakıldığında C'nin ilk terimde bir ikinci terimde ise sıfır olduğu görülmektedir. Bu yüzden C değişkeni kullanılmayacaktır. Bu gruptan çıkan sonuç  $A'B$ 'dir. İkinci grupta da aynı şekilde bakıldığında sonucu  $AB'$  çıkar. Bulunan sonuçlar birleştirildiğinde sadeleştirilmiş çıkış fonksiyonu  $F = A'B + AB'$  bulunur.

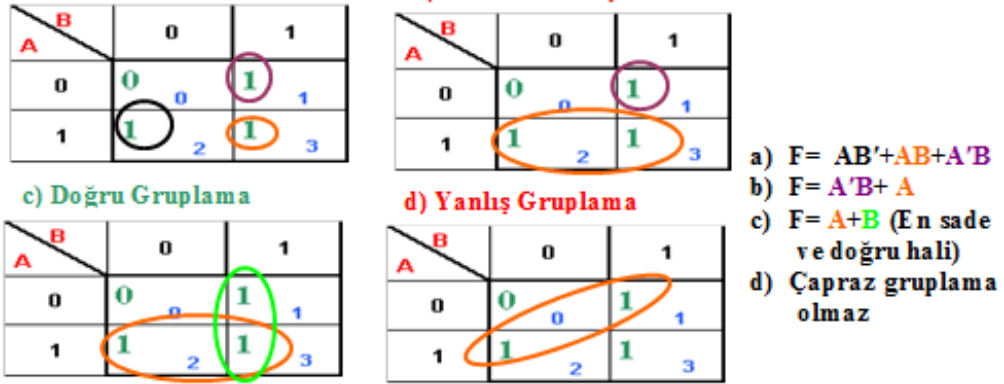


Şekil 1.9: Gruplanmış ve sadeleştirilmiş karno haritası

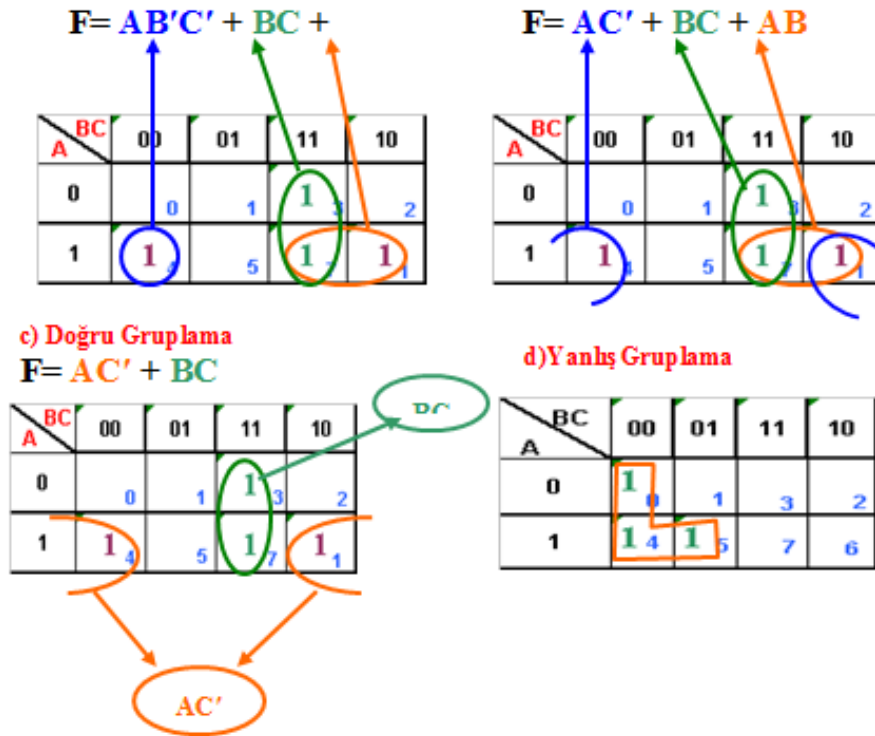


Aşağıda çeşitli gruplama örnekleri verilmiştir.

➤ İkili gruplamalara örnek



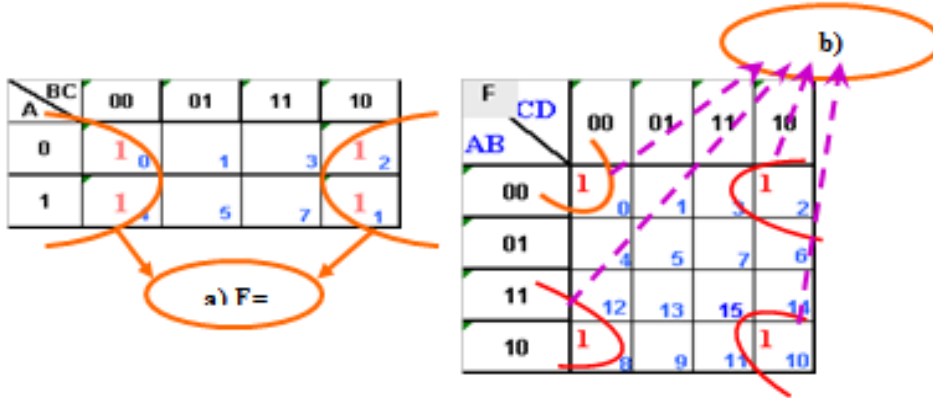
Şekil 1.10: İkili gruplama örnekleri 1



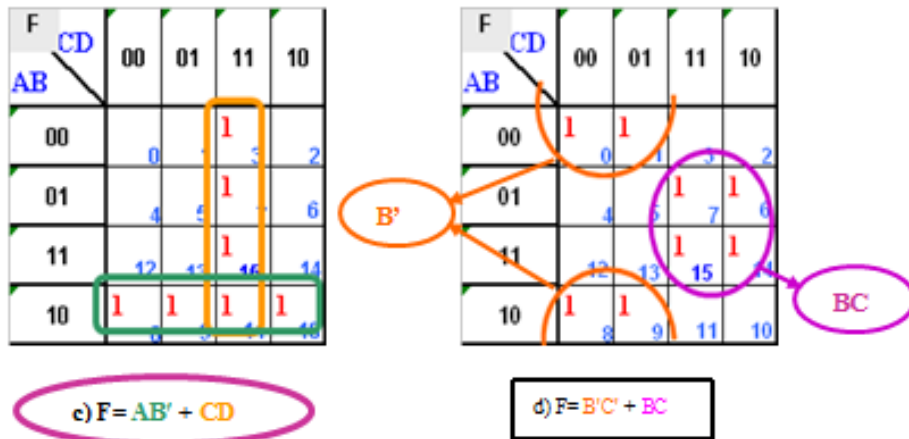
Şekil 1.11: İkili gruplama örnekleri 2

Şekil 1.11'de görüldüğü gibi gruplama en sade sonucu verecek şekilde olmalıdır.

➤ Dörtlü gruplamalara örnek

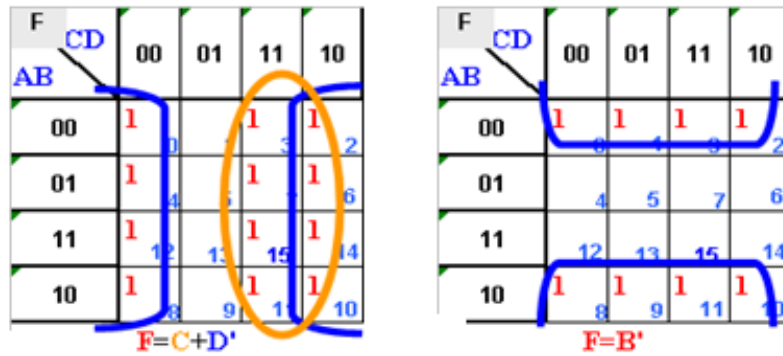


Şekil 1.12: Dörtlü gruplama örnekleri 1

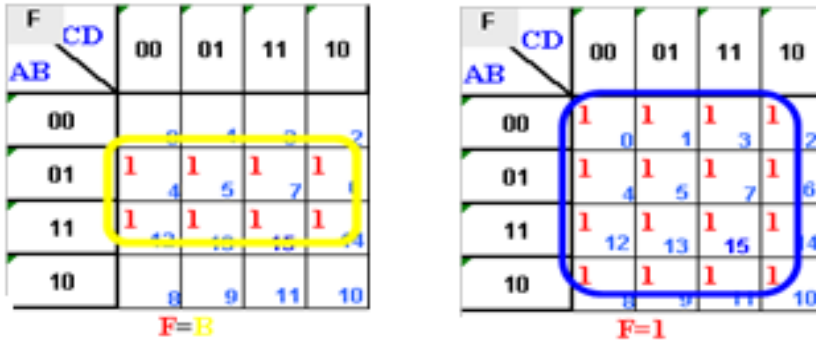


Şekil 1.13: Dörtlü gruplama örnekleri 2

➤ 8'li ve 16'h gruplama örnekleri

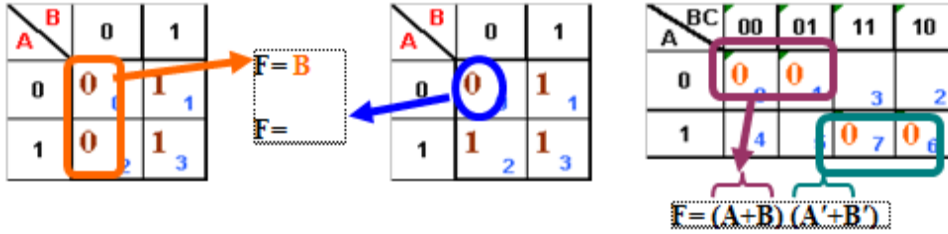


Şekil 1.14: 8'li gruplama örnekleri 1



Şekil 1.15: 8'li ve 16'lı gruplama örnekleri 2

- Devre tasarımında lojik0 aktif seçildiğinde karno haritasının indirgenmesinde toplam terimler (maxterm) kullanılır ve sıfırlar gruplanır. Sıfıra göre yapılan indirgemelerde değişkenin kendisi sıfırla, değil birle temsil edilir. Çarpım terimler (Maxterm) toplamların çarpımını veren terimlerdir. Şekil 1.16'da gösterilmiştir.



Şekil 1.16(a): Karno haritalarında sıfıra göre ikili gruplama

**Örnek 3:**  $f(A,B,C)=\Pi(1,6,7,3)$  fonksiyonunun karno haritasını çizin.

- Kanonik POS formun kalıp gösterimi  $F_{CPOS}=\Pi(M(.....) * \Phi(.....))$
- Giriş değişkenleri a,b,c olmak üzere üç tanedir.
- $\Pi$  (pi) terimi verilmiş ifadelerin çarpımlarını bulan matematiksel bir terimdir. Birden fazla çarpımı göstermek için kullanılır.  $\Pi M(1,3,6,7)=M1 * M3 * M6 * M7$  'dir.  $M_i=mi$ 'dir.
- $\Pi$  ile gösterilen fonksiyonlarda lojik0 aktif seviyedir. Buna göre karno haritasında  $M1, M3, M6$  ve  $M7$ 'yle gösterilen karelere sıfır verdikten sonra sıfırlara göre sadeleştirme yapmamız gerekmektedir.

		bc			
		00	01	11	10
a	0	0	1	0	
	1	4	5	6	

Şekil 1.16(b): Karno haritalarında sıfıra göre indirgeme örnek soru

- İndirgeme sonucu  $F = (A+C) * (A'+B')$  bulunur.

### 1.5.1. Tanımsız İfadelerin Karno Haritalarında Kullanılması

1.2. başlık numaralı konuda verilen bir numaralı örneğin çözümünde gerçekleşmesi imkansız olan bir olay vardı. Bu tür olması mümkün olmayan olayları doğruluk tablosunda gösterirken tanımsız (fi ( $\Phi$ )) ifadeleri kullanılır.

Tanımsız ifadelerin çıkışta lojik aktif olduğu düşünülerek oluşturulan karno haritasındaki gruplama sayısının, tanımsız olarak bırakılarak oluşturulmuş karno haritasındaki gruplamaya göre artış gösterdiği (maliyetin azalacağı) durumlarda (devrenin çıkışı daha sade hale gelecekse) tanımsız ifadeler lojik aktif sayılabilir.

1.1.2. başlık numaralı konuda verilen 1 numaralı örneğin çözümünü tekrar inceleyelim.

Hücre nu	A	B	Çıkış Alarm
0	0	0	0
1	0	1	1
2	1	0	1
3	1	1	$\Phi$

	A	B	
A	0	1	
0	0	1	1
1	1	0	1

$ALARM = AB' + A'B$

	A	B	
A	0	1	
0	0	1	1
1	1	0	1

$ALARM = A+B$

Şekil 1.17: Tanımsız ifadelerin karno haritalarında kullanılması

Doğruluk tablosu ve karno haritaları yukarıda verilmiştir. Bu devre:

Alarm =  $\sum m(1,2) + \Phi(3)$  şeklinde de gösterilebilir. Birden fazla tanımsız ifade olursa araya virgül konarak gösterilir. ÖRN:  $F = \sum m(1,2,7,9) + \Phi(3,8)$

Doğruluk tablosundaki tanımsız ifadenin tanımsız sayıldığı karno haritasında gruplanan devrenin çıkış fonksiyonu  $ALARM = AB' + A'B$  (beş kapı ve iki giriş ile tasarlanır) olarak bulunurken, doğruluk tablosundaki tanımsız ifadenin aktif (lojik1) sayıldığı ikinci karno haritasında gruplanan devrenin çıkış fonksiyonu  $ALARM = A+B$  (bir kapı iki giriş ile tasarlanır) olarak bulunur. Bu örnekte tanımsız ifadenin aktif sayılması devrenin maliyetini düşürücü yöndedir ve devre için en ucuz tasarım olan A+B kullanılmalıdır.

## UYGULAMA FAALİYETİ

$F = x'y'z + x'yz' + x'y'z' + xy'z$  boolean fonksiyonunu karno haritası kullanarak sadeleştirin.

İşlem Basamakları	Öneriler
➤ Karno haritasının kaç değişkenli olduğunu belirleyiniz.	➤ Fonksiyonda ki giriş değişkenlerini inceleyiniz.
➤ Karno haritasını çiziniz.	➤ Değişken sayısına dikkat ediniz.
➤ Devrenin lojik 1 de mi yoksa lojik sıfırda mı aktif olacağını belirleyiniz?	➤ Verilen ifade incelenir.
➤ Karno haritasının içini doldurunuz.	➤ Karno haritasını doldurma işlemi aktif çıkışlara göre yapılmalıdır.
➤ Karno haritası üzerinde guruplama işlemi yapınız.	
➤ Çıkış fonksiyonunu sadeleştiriniz.	➤ Gruplandırılmış çıkışlar sadeleştirilir.

## ÖLÇME VE DEĞERLENDİRME

Aşağıdaki soruları dikkatlice okuyunuz ve doğru seçeneği işaretleyiniz.

- Aşağıdakilerden hangisi karno haritalarında kullanılan indirgeme (gruplandırma) yöntemlerinden biri değildir?  
A) 2 B) 4 C) 6 D) 16
- Giriş değişken sayısı  $N=4$  olan karno haritası kaç kareden oluşmaktadır.  
A) 64 B) 4 C) 16 D) 8
- $(3+5)_{10} = ( \_ \_ \_ \_ )_2$  toplam işleminin ikilik sistemdeki sonucu aşağıdakilerden hangisidir.  
A) 0008 B) 1000 C) 1010 D) 0100
- Aşağıdaki karno haritasının çıkış fonksiyonunun en sade hali hangi boolean ifadesinde verilmiştir.  
A)  $(B+D). (C+D')$   
B)  $B' D' + C' D + A' B' C'$   
C)  $(B'+D'). (C'+D). (A' B' C')$   
D)  $B' D' + C' D$

CD	00	01	11	10
AB				
00	1 0	1 1	3 1	2
01	4	1 5	7	6
11	12	1 13	15	14
10	1 8	1 9	11	1 10

- Aşağıdaki karno haritasının çıkış fonksiyonunun en sade hali hangi Boolean ifadesinde verilmiştir.  
A)  $F = (A+B) * (A'+B')$   
B)  $F = (A'+B) * (A+B')$   
C)  $F = AB + A'B'$   
D)  $F = A' B + AB'$

BC	00	01	11	10
A				
0	0 0	0 1	3	2
1	4	5	0 7	0 6

- Aşağıdaki doğruluk tablosunun karno haritası aşağıdakilerden hangisidir.

sıra	X	Y	Z	F
0	0	0	0	0
1	0	0	1	1
2	0	1	0	1
3	0	1	1	1
4	1	0	0	0
5	1	0	1	0
6	1	1	0	0
7	1	1	1	1

A)

F	YZ	00	01	11	10
X					
0	1 0	1 0	1 0		
1	1 0	1 0			

B)

F	YZ	00	01	11	10
X					
0	0 0	0 1	1 0		
1	0 1	1 1	1 1		

C)

F	YZ	00	01	11	10
X					
0	1 0	1 1	1 1		
1	1 0	0 0			

D)

F	YZ	00	01	11	10
X					
0	0 1	1 1	1 1		
1	0 0	1 0			

**Aşağıdaki cümlelerin başında boş bırakılan parantezlere, cümlelerde verilen bilgiler doğru ise D, yanlış ise Y yazınız.**

7. ( ) Devrelerin sadeleştirilmesinde kullanılan yöntemlerden karno haritası yöntemi, Boolean Cebrine göre daha güvenilir ve kolay bir yöntemdir.
8. ( ) Doğruluk tablolarında bulunan tanımsız durumlar karno haritalarında aktif çıkış olarak kullanılabilirler.
9. ( ) Karno haritalarında sadece bire göre indirgeme yapılabilir.
10. ( ) Üç tane giriş değişkeni için faydalanılacak karno haritası 6 kareden oluşmaktadır.

### **DEĞERLENDİRME**

Cevaplarınızı cevap anahtarıyla karşılaştırınız. Yanlış cevap verdiğiniz ya da cevap verirken tereddüt ettiğiniz sorularla ilgili konuları faaliyete geri dönerek tekrarlayınız. Cevaplarınızın tümü doğru ise bir sonraki öğrenme faaliyetine geçiniz.

# ÖĞRENME FAALİYETİ-2

## AMAÇ

Toplayıcı devreleri uygulamaları yapabileceksiniz.

## ARAŞTIRMA

- İkilik sistemde toplama işleminin nasıl yapıldığını araştırınız.
- Toplayıcı entegresinin bacak bağlantılarını ve toplama işlemini gerçekleştirebilecek başka entegreleri araştırınız.

## 2. TOPLAYICI DEVRELERİ

Bilgisayarların gerçekleştirdikleri işlemlerden birisi aritmetik işlemlerdir. Aritmetik işlemlerin en temeli ikilik sayı sistemindeki iki bit'in toplanmasıdır.

Tümleşik devrelerde kullanılan iki temel toplayıcı tipi vardır. İki bitin toplamasını yapan devreye yarım toplayıcı, üç bitin toplamasını yapan devreye tam toplayıcı devresi adı verilir.

### 2.1. Yarım Toplayıcı

Yarım toplayıcı devresi ikilik sistemdeki birer bitlik iki sayının toplanmasıyla olur. Bu toplama işlemi dört olasılık üzerine kuruludur.

$$A=0 \text{ ve } B=0 \text{ olması } 0 + 0 = 0$$

$$A=0 \text{ ve } B=1 \text{ olması } 0 + 1 = 1$$

$$A=1 \text{ ve } B=0 \text{ olması } 1 + 0 = 1$$

$A=1$  ve  $B=1$  olması  $1 + 1 = 10$  onluk sistemdeki ikinin karşılığı ikilik sistemde "10" (bir sıfır)'dır. Bu sonuç iki bit çıktığından taşma olmuştur. Böyle durumlarda taşma olmaması için çıkış fonksiyon sayısı, taşması tahmin edilen bit sayısına göre hesaplanır. Yarım toplayıcı devresinde bir bit taşma olduğundan iki tane çıkış fonksiyonu kullanılır. Bunlar toplama (Sum (S)) ve elde (Carry (C)) fonksiyonlarıdır. Elde biti taşma olup olmadığını göstermek için kullanılır.

#### 2.1.1. Yarım Toplayıcı Devrenin Tasarımı

Yarım toplayıcı devresini çizebilmemiz için lojik devre tasarlama adımlarını takip etmemiz gerekmektedir. Takip edeceğimiz adımlar:



- Devrenin doğruluk tablosunu hazırlayıp doğruluk tablosuna göre her iki çıkışın karno haritalarını çizeceğiz,

kare nu	A	B	C elde	S toplam
0	0	0	0	0
1	0	1	0	1
2	1	0	0	1
3	1	1	1	0

C	B	0	1
A	0	0 <sub>0</sub>	0 <sub>1</sub>
1	0 <sub>2</sub>	1 <sub>3</sub>	

S	B	0	1
A	0	0 <sub>0</sub>	1 <sub>1</sub>
1	1 <sub>2</sub>	0 <sub>3</sub>	

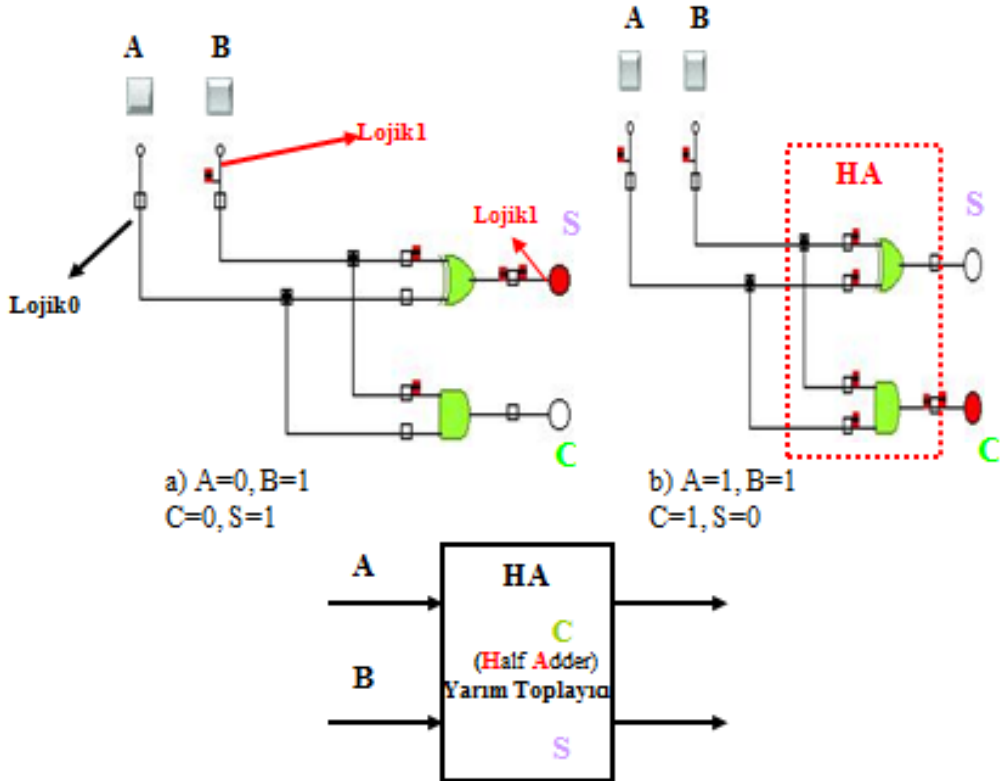
Şekil 2.1: Yarım toplayıcı devresinin doğruluk tablosu ve çıkış fonksiyonlarının karno haritaları

- Karno haritalarının sadeleştirmelerini yapıp çıkış fonksiyonlarını bulacağız,

Elde çıkışının fonksiyonu  $C = AB$

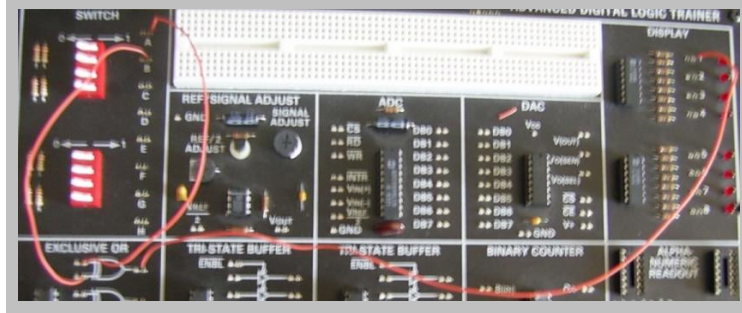
Toplam çıkışının fonksiyonu  $S = AB' + A'B$  toplam fonksiyonunu iki and ve bir or kapısı olmak üzere üç kapı yerine EX-OR kapısıyla sadece tek bir kapıyla yapabiliriz. Böylece bütün devre iki kapıyla yapılmış olur.  $S = AB' + A'B = A \oplus B$

- Çıkış fonksiyonlarına göre devrenin tasarımını çizeceğiz ve çizdiğimiz devreyi bilgisayar üzerinde simülasyon programlarında deneyeceğiz.



Şekil 2.2: Yarım toplayıcı devrenin simülasyon (crocodilephysics) programındaki uygulaması ve sembolik entegre gösterimi

- Simülasyon programında kurduğumuz devre doğru çalıştıysa devremizi kuracağız. Devrenizi kurmadan önce uygulama faaliyetindeki işlem basamakları ve önerileri okuyunuz.



Resim 2.1: Yarım toplayıcı devrenin toplam çıkışının eğitim setinde kurulması

## 2.2. Tam Toplayıcı

Bir bitlik iki sayıyı toplayan devrelerin yarım toplayıcı olduğunu gördünüz. Bir bitlik üç sayıyı toplayan devrede tam toplayıcı (fulladder) devresidir. Tam toplayıcı devresinin üç girişi vardır. İlk iki giriş toplama yapılması istenen iki bir bitlik sayıyı, üçüncü düşük değerlikli bitte bir önceki toplamadan ( $A+B$ ) gelen elde (carry input) bitidir. Üç girişin maksimum alabileceği değer birdir ve üçünün bir olması durumunda çıkış  $1+1+1 = (11)_2$ 'dir. Bu yüzden devrenin iki çıkış fonksiyonu vardır. Bunlardan biri toplam (sum) diğeri elde (carryout) çıkışıdır.  $S=A+B+C_i$  şeklinde formül haline getirilebilir.

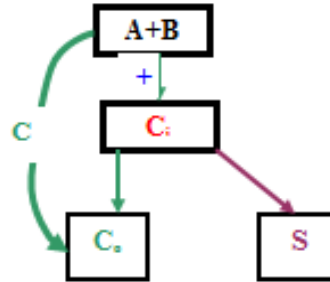
### 2.2.1. Tam Toplayıcı Devresinin Tasarımı

Tam toplayıcı devresini çizebilmemiz için lojik devre tasarlama adımlarını takip etmemiz gerekmektedir. Takip edeceğimiz adımlar:

- Devrenin doğruluk tablosunu hazırlayacağız,

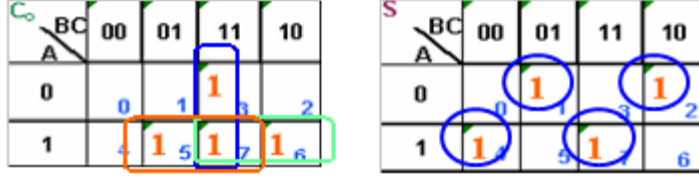
$C_o$ ,  $S=A+B+C_i$  işleminden sonra oluşan eldeyi temsil eder. Girişteki elde ( $C_i$ )  $A+B$  işleminden gelen toplam değeriyle toplanır.  $A+B$  işleminde elde oluşursa varsa direkt  $C_o$  bir olur.

hücre no	A	B	$C_i$	elde çıkışı $C_o$	toplam S
0	0	0	0	0	0
1	0	0	1	0	1
2	0	1	0	0	1
3	0	1	1	1	0
4	1	0	0	0	1
5	1	0	1	1	0
6	1	1	0	1	0
7	1	1	1	1	1



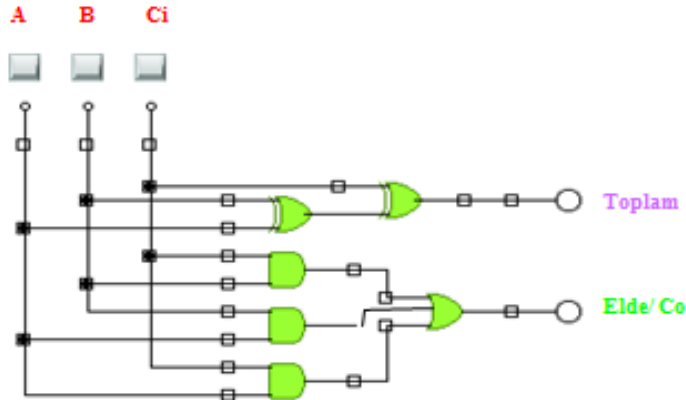
Şekil 2.3: Tam toplayıcı devresinin doğruluk tablosu ve tam toplama işlem basamakları

- Her iki çıkışın karno haritalarını çizeceğiz,



Şekil 2.4: Tam toplayıcı devresinin çıkış fonksiyonlarının karno haritaları

- Karno haritalarının sadeleştirmelerini yapıp çıkış fonksiyonlarını bulacağız,
- Tam toplayıcı devresinin sadeleştirilmiş toplam çıkış fonksiyonu;
  - $S = AB'C_i + ABC_i + A'B'C_i + A'BC_i'$  (daha az kapı kullanabilmek için ayrı ayrı  $C_i$  ve  $C_i'$  ortak parantezlerine aldık)
  - $S = C_i(AB + A'B') + C_i'(AB' + A'B)$  (Birinci parantez EX-NOR = (EX-OR)' ikinci parantez EX-OR kapısının formülünü göstermektedir. Bu yüzden aşağıdaki gibi yazılabilir.)
  - $S = C_i(A \oplus B)' + C_i'(A \oplus B)$  bu ifadeye  $(A \oplus B)$  gördüğümüz yere X yazdığımızı varsayalım
  - $S = C_iX' + C_i'X$  denklemini buluruz bu da bir EX-OR kapısıdır.  $S = C_i \oplus X$  Son olarak X gördüğümüz yere  $(A \oplus B)$  'yi tekrar geri yazdığımızda,  $S = C_i \oplus (A \oplus B) = (A \oplus B) \oplus C_i$  toplama fonksiyonu bulunmuş oldu. (Demorgan kurallarından değişme özelliği)
  - Tam toplayıcı devresinin sadeleştirilmiş elde çıkışı  $C_o = AC + AB + BC_i$  olarak bulunur.
- Çıkış fonksiyonlarına göre devrenin tasarımını çizeceğiz, ve çizdiğimiz devreyi bilgisayar üzerinde simülasyon programlarında deneyeceğiz,



Şekil 2.5: Tam toplayıcı devresi

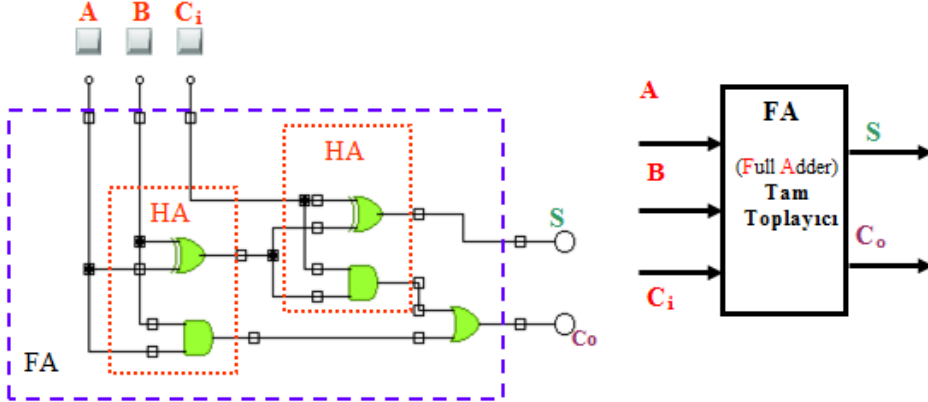
Bu haliyle devremiz toplam yedi kapıyla tasarlanıyor. Lojik devre tasarlanmanın en önemli kuralı az maliyetli olması yani az kapı kullanılmasıydı. Bu devre  $C_o$  çıkışının karno haritasını grupladığımızda normalde altı kapı ile yapılırken  $C_o$  çıkışına daha farklı bir gruplama yapılırsa beş kapı ile de yapılabilir. Karno haritasına göre anlamsız bir sadeleştirme olmasına rağmen bu devrede daha az kapı sağladığından anlamlı hale gelmiş olur. Özel bir durum olduğundan gruplama aşağıdaki gibi yapılabilir.

C	BC	00	01	11	10
A	0	0	1	1	2
1	1	1	1	1	6

Şekil 2.6: Tam toplayıcı devrenin karno haritasının özel durum gruplaması

Karno haritasından bulunan indirgemeye göre;

- $C_o = AB'C_i + A'BC_i + AB$  bu sonuç  $C_i$  parantezine alınır.
- $C_o = C_i(AB' + A'B) + AB$  çıkar.
- Buradan da  $C_o = C_i(A \oplus B) + AB$  çıkar.  $(A \oplus B)$  toplam fonksiyonunda zaten kullanılacağından burada  $(A \oplus B)$ 'yi tekrar kullanmayarak bir kapıdan tasarruf edilmesi sağlanmış olmaktadır. Yeni devre aşağıdaki gibidir.



Şekil 2.7: İki yarım toplayıcının birleştirilmesiyle oluşturulmuş tam toplayıcı devresi ve sembolik entegre gösterimi

Şekil 2.7'den de anlaşılacağı gibi tam toplayıcı devresi iki yarım toplayıcının kullanılmasıyla da elde edilebilir.

- Simülasyon programında kurduğumuz devre doğru çalıştıysa devremizi kurabiliriz.

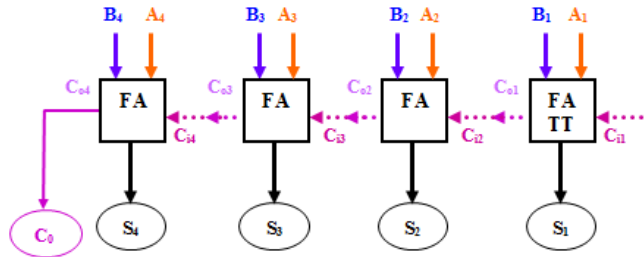
## 2.3. Toplayıcı Entegresi

### 2.3.1. Paralel Toplama

Paralel toplamada her bir iki biti toplamak ve oluşan eldeyi bir sonraki iki bite aktarmayı sağlamak için tam toplayıcılar kullanılır. İki dört bitlik sayının toplanması için dört tane tam toplayıcı kullanılması gerekmektedir.

$$\begin{array}{r} A_4 A_3 A_2 A_1 \\ + B_4 B_3 B_2 B_1 \\ \hline C_0 S_4 S_3 S_2 S_1 \end{array} \quad \begin{array}{l} S_1 = A_1 + B_1 + C_{i1} \\ S_2 = A_2 + B_2 + C_{i2} \\ S_3 = A_3 + B_3 + C_{i3} \\ S_4 = A_4 + B_4 + C_{i4} \end{array}$$

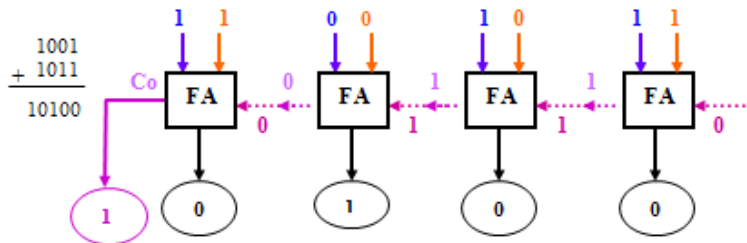
Şekil 2.8: Toplama işlemi genel formatı



Şekil 2.9: Dört bitlik paralel toplama devresi

Paralel toplamada ilk elde girişi  $C_i = 0$  olmalıdır çünkü  $A_1 + B_1$  ilk işlemidir ve daha öncesinde elde doğuracak bir işlem yapılmamıştır. Yukarıdaki devrede birinci toplamının elde çıkışı kendinden sonraki toplamının elde girişini oluşturur ( $C_{o1} = C_{i2}$ ).

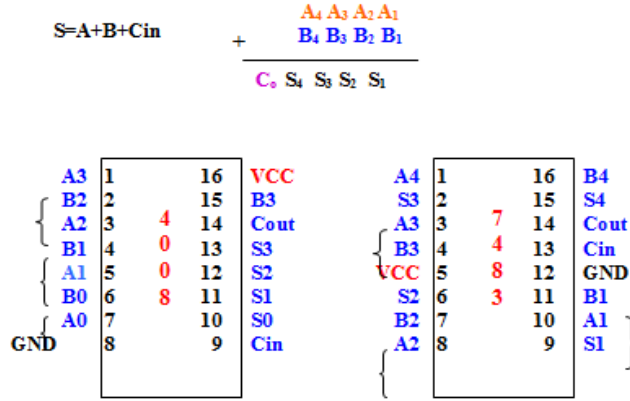
**Örnek 4:** Aşağıdaki toplama işlemini şekil 2.8'deki devreye yerleştirip sonucunu bulacağız.



Şekil 2.10: Örnek dört bitlik paralel toplama devresi

### 2.3.2. Toplayıcı Entegresi İle İkilik Sistemde Dört Bitlik İki Sayıyı Toplama

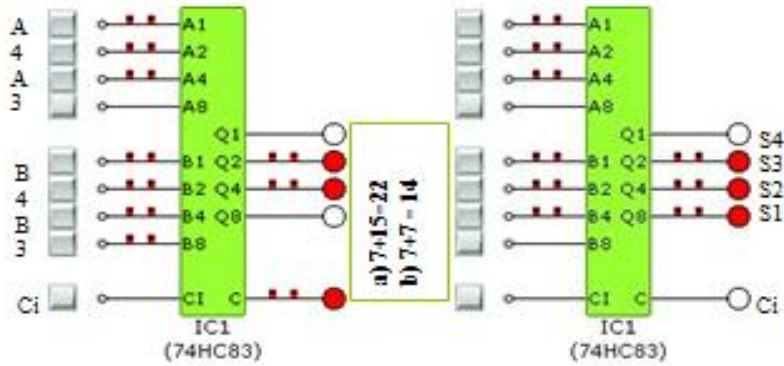
Toplayıcı entegresinde yapılan toplama işlemi paralel toplama mantığına göre yapılmaktadır. Bu entegre ikilik sistemdeki dört bitlik iki sayının toplanmasını yapar. Toplamının doğru olması için  $C_{in}$  girişi lojik0 olmalıdır. Toplam (S) ve elde (Cout) çıkışlarına led bağlanacaktır.



**Şekil 2.11: Dört bitlik iki sayıyı toplayan-çıkaran entegrelerin bacak bağlantıları**

7483 entegresini kullanarak iki tane dört bitlik sayıyı toplayan devreyi oluşturmak aşağıdaki adımlar takip edilir.

- 0 ve 1 (+5V) bilgilerini elde ederek çeşitli sayıları elde etmek için, A3,A2,A1,A0, B3,B2,B1,B0 giriş değişkenlerinin her biri bir anahtara,
- GND ucu toprağa,
- Devredeki VCC ucu +5 Voltluk Dc güç kaynağına,
- S3,S2,S1,S0 toplam ve Cout elde çıkışları her biri birer lambaya (led) bağlanır.



**Şekil 2.12: Dört bitlik iki sayıyı toplayan devrenin 7483 entegresiyle simülasyon (crocodilephysics) programda yapılması**

- Simülasyon programında kurduğumuz devre doğru çalıştıysa devremizi kuracağız.

Devre Şekil 2.11'deki bacak bağlantılarını dikkate alarak kurulmalıdır simülasyon programdakine göre değil

## UYGULAMA FAALİYETİ

Dört bitlik bir sayıya bir ekleyen devreyi tasarlayınız.

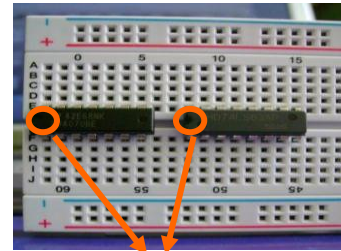
İşlem Basamakları	Öneriler
➤ Devre çıkışı lojik1 de mi yoksa Lojik0 da mı aktif olacak belirleyiniz.	➤ Çıkışın Lojik1 de aktif olması sizin soruyu daha kolay çözmenizi sağlayacaktır.
➤ Devrenin girişinin ve çıkışının kaç bitten oluşacağını belirleyiniz.	➤ Hata olmaması için çıkışın bit sayısının taşma olmayacak şekilde düşülmesi gereklidir.
➤ Devrenin doğruluk tablosunu hazırlayınız.	➤ Giriş değişken sayılarına bakarak karno haritasının kaç kareden oluşacağını 2N formülüne göre hesaplayabilirsiniz. ➤ Karno haritasının kalıbını hazırlarken haritadaki karelerin numaralarının sıralamalarının Gray koda göre verildiğini unutmayınız. ➤ Her bir çıkış için ayrı karno haritası hazırlayınız.
➤ Doğruluk tablosuna bakarak karno haritasının içeri doldurunuz.	
➤ Karno haritasına göre indirgeme yapıp her bir çıkışın fonksiyonunu bulunuz.	➤ Karno haritalarında gruplama kurallarına uygun şekilde gruplama yaparsanız en doğru indirgemeyi yapmış ve en az maliyetli çıkışı bulmuş olursunuz. İndirgemenizin doğruluğunu kontrol etmek için karno haritalarının indirgemesini yapan programları internetten bulabilirsiniz.
➤ Çıkış fonksiyonuna bakarak devre için gerekli elemanları belirleyiniz.	➤ Dijital elektronik devre elemanları 5V ile çalışır. Kullanılan entegrelerin bacak bağlantılarının bulunduğu bir belgeyi yanınızda bulundurun.
➤ Çıkış fonksiyonunuzu verecek devreyi bilgisayardaki simülator programlarda kurup, devreyi çalıştırınız.	➤ Simülator program olarak; CrocodileTechnology, EWB gibi programlardan yararlanabilirsiniz.
➤ Bilgisayardaki devreniz doğru sonuçlar veriyorsa devrede kullanılan lojik elemanları son olarak tekrar belirleyip, devrenizi plakete (breadboard) veya lojik eğitim setinize kurunuz.	➤ Plaket (breadboard) kullanmayı hatırlamıyorsanız devreyi yanlış kurmamak için uygulama öğretmeninizden ya da laboratuvar eşinizden nasıl kullanıldığını hatırlatmasını isteyiniz. ➤ Kullanacağınız entegrelerin 74*** serisine mi ait yoksa 40** serisine mi ait

olduđuna dikkat edin. Devrenizde kullanacađınız bütn kapıların bulabilerseniz aynı seriye ait olmasına dikkat ediniz. Devrenizde kullanacađınız entegrelerin kullanılmayacak duruma gelebilme ihtimalini düşünerek yanınızda yedeklerini bulundurunuz.



**Entegrenin 1 numaralı bacağı**

- Entegreleri takarken bacak bağlantıları ve numaraları çok önemlidir. Entegreyi yukarıdaki gibi yazısı bize göre düz olacak şekilde yerleştirildiđinde kırmızı ile çerçeveye alınmış kısmın (çentiđin) altındaki ilk bacak entegrenin 1 numaralı bacağıdır. Bacak numaraları kırmızı çerçevenin altından başlar alt sıra bitince kırmızı çerçevenin üstünden devam eder.
- Özellikle karmaşık devrelerde hata yaparsanız; kablolarınızı tek tek kontrol etmek yerine devreyi baştan tekrar kurarsanız doğru sonuca daha kolay ulaşabilirsiniz.
- Devre elemanlarını plakete (breadboard) aşğıdaki gibi yerleştirildikten sonra kablo bağlantılarını yapmaya başlarsanız devrenizi daha kolay kurabilirsiniz.



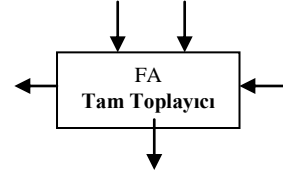
**Entegre başnoktaları**



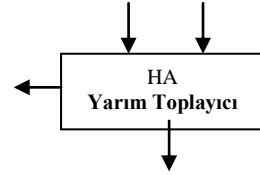
## ÖLÇME VE DEĞERLENDİRME

Aşağıdaki soruları dikkatlice okuyunuz ve doğru seçeneği işaretleyiniz.

1. Aşağıdakilerden hangisi tam toplayıcı devresinde bulunan giriş veya çıkışlardan biri değildir?
- A) Elde Girişi (CarryInput)  
B) Elde Çıkışı (Carry Output)  
C) Borç girişi (Borrow input)  
D) Toplam (sum)



2. 4 bitlik toplam yapan entegre aşağıdakilerden hangisidir.  
A) 74285      B) 7483      C) 7485      D) 74284
3. Aşağıdakilerden hangisi yarım toplayıcı devresinde bulunan giriş veya çıkışlardan biri değildir?
- A) Elde Girişi (CarryInput)  
B) Elde Çıkışı (Carry Output)  
C) Birinci sayı  
D) Toplam (sum)



4. Bir bitlik iki sayının toplamını bulan devrenin adı nedir?
- A) Tam toplayıcı  
B) Yarım toplayıcı  
C) Yarım çıkarıcı  
D) Karno haritası

Aşağıdaki cümlelerin başında boş bırakılan parantezlere, cümlelerde verilen bilgiler doğru ise **D**, yanlış ise **Y** yazınız.

5. ( ) Tam toplayıcı devre birer bitlik iki sayının toplamını veren devredir.
6. ( ) Dört bitlik bir sayıyı sabit iki sayısıyla toplayan devrenin çıkışı en az 5 bit olur.

## DEĞERLENDİRME

Cevaplarınızı cevap anahtarıyla karşılaştırmız. Yanlış cevap verdiğiniz ya da cevap verirken tereddüt ettiğiniz sorularla ilgili konuları faaliyete geri dönerek tekrarlayınız. Cevaplarınızın tümü doğru ise bir sonraki öğrenme faaliyetine geçiniz.

# ÖĞRENME FAALİYETİ-3

## AMAÇ

Çıkarıcı tüm devre elemanını gerektiğinde geliştirdiği devrelerde kullanabileceksiniz.

## ARAŞTIRMA

- Dört Toplayıcı-Çıkarıcı entegrelerinin bacak bağlantılarını araştırınız.
- Bire göre tümlenme (birin komplementi), ikiye göre tümlenme nasıl yapıldığını ve tümlenme alamaya neden ihtiyaç duyulduğunu araştırınız.
- İkilik sistemde çıkarma işleminin nasıl yapıldığını araştırınız.
- İşaret biti terimini araştırınız.

## 3. ÇIKARICI DEVRELERİ

Çıkarma devreleri toplama devresine benzer. Toplayıcı devresinde toplam çıkışı ile fark (difference) çıkışı; elde çıkışı ile de borç (borrow) çıkışı yer değiştirince yarım çıkarıcı devresi elde edilmiş olur.

### 3.1. Yarım Çıkarıcı

Yarım çıkarıcı devresi ikilik sistemdeki birer bitlik iki sayının çıkartılmasıyla olur. Çıkarma işlemleri de toplama işlemi gibi dört olasılık üzerine kuruludur. Bu devrenin doğruluk tablosu basit çıkarma kurallarına göre düzenlenir.

A=0 ve B=0 olması	$0 - 0 =$	0
A=1 ve B=0 olması	$1 - 0 =$	1
A=1 ve B=1 olması	$1 - 1 =$	0
A=0 ve B=1 olması	$0 - 1 =$	1 (borç 1)

Sıfırdan bir çıkmayacağı için bir sonraki bitten bir borç alınır ve ikilik tabanındaki 10 sayısından bir çıkarılır. Alınan borcun doğruluk tablosunda gösterilmesi gerekir bunun için alınacak borçlar içinde bir çıkış fonksiyonu oluşturulur.

#### 3.1.1. Yarım Çıkışı Devre Tasarımı

Yarım çıkarıcı devresini çizemememiz için lojik devre tasarlama adımlarını takip etmemiz gerekmektedir. Takip edeceğimiz adımlar:

- Devrenin doğruluk tablosunu hazırlayıp doğruluk tablosuna göre her iki çıkışın karno haritalarını çizeceğiz,

kare nu	X	Y	Fark difference	borç borrow
0	0	0	0	0
1	0	1	1	1
2	1	0	1	0
3	1	1	0	0

F	Y	0	1
X	0	0	1
1	0	1	0

B	Y	0	1
X	0	0	1
1	0	0	0

Şekil 3.1: Yarım çıkarıcı deresinin doğruluk tablosu ve fark ve borç çıkışlarının karno haritaları

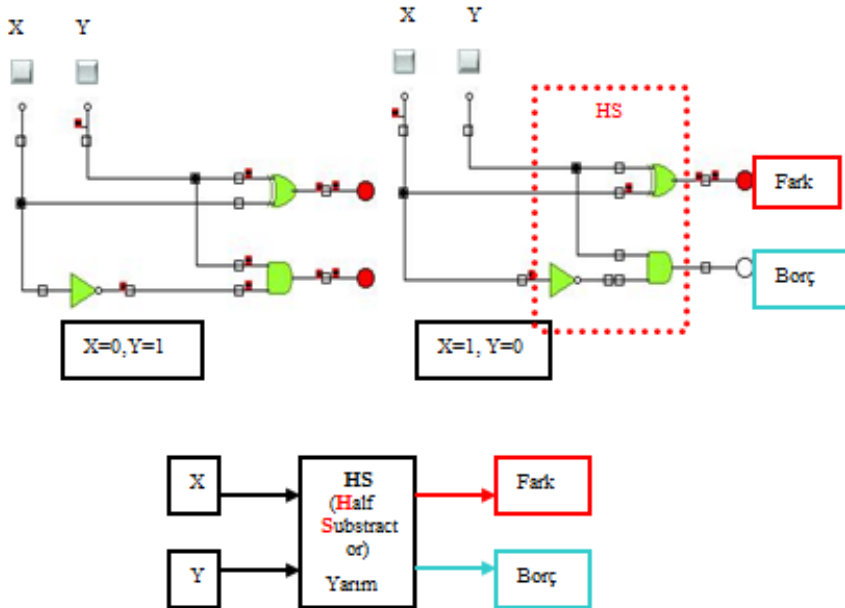
- Karno haritalarının sadeleştirmelerini yapıp çıkış fonksiyonlarını bulacağız,

Fark çıkışına baktığımızda doğruluk tablosuna gerek olmadan da artık sonucu görebilirsiniz. Çıkışı 0-1-1-0 olan kapı EX-OR kapısıdır. Eğer kapıları iyi bilmiyorsanız yarım toplayıcı devresinde olduğu gibi karno haritasından gerekli sadeleştirmeleri yaparak da doğru sonucu bulabilirsiniz.

Fark çıkışı (Difference)  $\text{Fark} = XY' + X'Y = X \oplus Y$

Borç çıkışı (Borrow)  $\text{Borç} = X'Y$  bulunur.

- Çıkış fonksiyonlarına göre devrenin tasarımını çizeceğiz, ve çizdiğimiz devreyi bilgisayar üzerinde simülör programlarda deneyeceğiz,



Şekil 3.2: Yarım çıkarıcı devresinin simülör (crocodilephysics) programındaki uygulaması ve sembolik entegre gösterimi

- Simülör programda kurduğumuz devre doğru çalıştıysa devremizi kuracağız.

## 3.2. Tam Çıkarıcı

Daha düşük değerlikli kademedен bir borç alınmış olabileceği dikkate alınarak iki biti birbirinden çıkararak tümleşik (kombinyonel) bir devredir ve üç giriş ile iki çıkıştan oluşur. Bu devre için b değişkeni ile borç çıkışının karıştırılmaması için giriş değişkenleri olarak X, Y, B<sub>i</sub> çıkış değişkenleri olarak ta F (fark) ve B<sub>o</sub>(borç çıkışı) kullanılacaktır. X çıkarılan sayı, Y çıkan sayı, B<sub>i</sub>'de daha önce alınan borcu gösterecektir.

### 3.2.1. Tam Çıkarıcı Devrenin Tasarımı

Yarım çıkarıcı devresini çizebilmemiz için lojik devre tasarlama adımlarını takip etmemiz gerekmektedir. Takip edeceğimiz adımlar:

- Devrenin doğruluk tablosunu hazırlayıp doğruluk tablosuna göre her iki çıkışın karno haritalarını çizeceğiz,

Sadece bu çıkarma devresi için özel bir durum olarak tam çıkarıcı devrenin doğruluk tablosunu onluk tabanda çıkarma yapıyormuş gibi düşünüp çıkışları (X-Y)-B<sub>i</sub> formülüne göre doldurabilirsiniz.

İşlem sırasındaki sırayla gerçekleşen çıkarma işlemlerinden herhangi bir tanesinden sonra sonucun negatif çıkması borç alınmış, yani borç çıkışının bir olması demek olacaktır. Tablodaki X=0, Y=1, B<sub>i</sub> =1 seçeneğini inceleyelim. İlk önce (0-1)'den parantez içinden negatif bir değer çıkar bu yüzden borç çıkışı B<sub>o</sub>=1 olacaktır. Bir sonraki adım olan (-1+1)işleminin sonucu da bize farkı (F=0) verecektir. X=1, Y=0, B<sub>i</sub> =1 seçeneğinde (1-0) parantez içindeki işlemden sonuç pozitif (+1), daha sonra ki işlemde de 1-1 sonuç sıfır çıkacaktır. Negatif bir sonuç çıkmadığı için borç çıkış fonksiyonu sıfır B<sub>o</sub>=0, fark çıkışı da işlemin sonucunda sıfır çıktığı için sıfır F=0 olacaktır.

hücre no	X	Y	B	Fark difference	borç borrow
0	0	0	0	0	0
1	0	0	1	1	1
2	0	1	0	1	1
3	0	1	1	0	1
4	1	0	0	1	0
5	1	0	1	0	0
6	1	1	0	0	0
7	1	1	1	1	1

X \ YB <sub>i</sub>	B			
	00	01	11	10
0	0	1	1	1
1	1	1	0	0

X \ YB <sub>i</sub>	F			
	00	01	11	10
0	0	1	0	1
1	1	0	1	0

$$B_o = YB_i + X'Y + X'Y'$$

$$F = XY'B_i + X'YB_i + XYB_i + X'Y'$$

$$B_i$$

Şekil 3. 3: Tam çıkarıcı devresinin doğruluk tablosu ve çıkış fonksiyonlarının karno haritaları

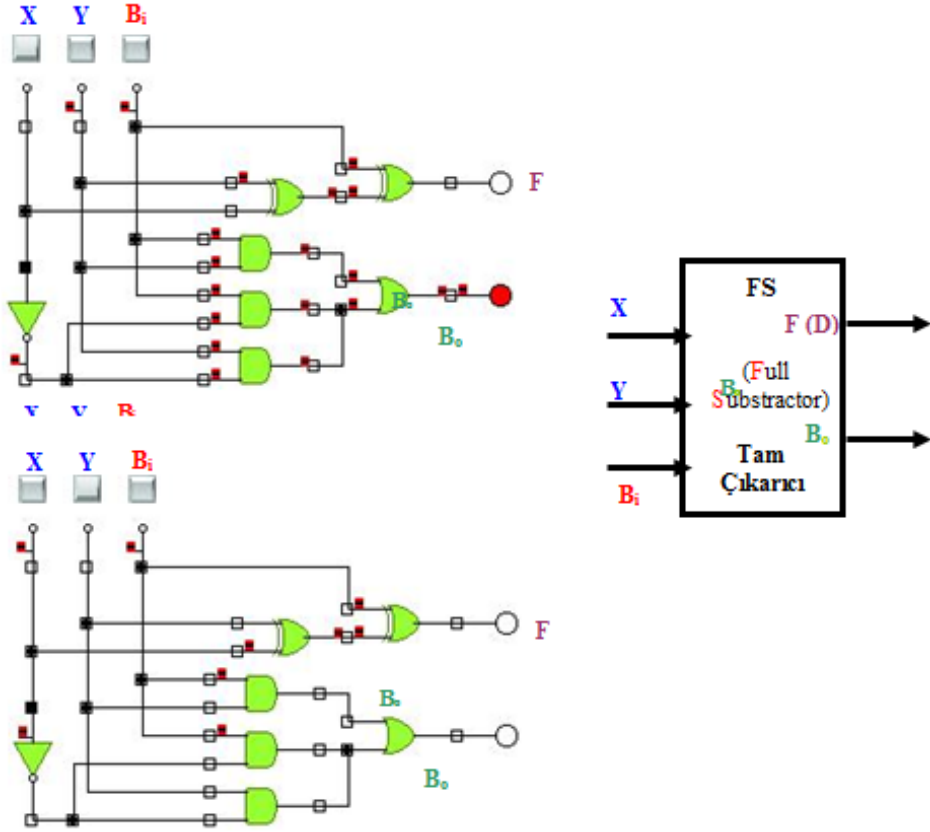
- Karno haritalarının sadeleştirmelerini yapıp çıkış fonksiyonlarını bulacağız.

Toplayıcı devresinde yaptığımız işlemleri fark çıkışı içinde yapacağız. Şekil 2.3'teki sadeleştirilmiş çıkışlara göre bu devre için toplam 12 kapı gerekmektedir. F çıkışı EX-OR

kapılarıyla yapılırsa devre yedi kapıyla yapılabilecek. Bu yüzden F çıkışını en sade hale getireceğiz ve devreyi daha az kapılı haliyle tasarlayacağız.

- $F = XY'B_i' + X'YB_i' + XYB_i + X'Y'B_iB_i$  ve  $B_i'$  ortak parantezlerine alacağız.
- $F = B_i (XY + X'Y') + B_i' (X'Y + XY')$  ilk parantez iki girişli EX-NOR kapısı, ikinci parantezde iki girişli EX-OR kapısıdır.
- $F = B_i(X \oplus Y)' + B_i' (X \oplus Y)$
- **$F = X \oplus Y \oplus B_i$**  bulunur.

➤ Çıkış fonksiyonlarına göre devrenin tasarımını çizeceğiz, ve çizdiğimiz devreyi bilgisayar üzerinde simülasyon programlarında deneyeceğiz,



Şekil 3. 4: Tam çıkarıcı devresinin simülasyon programındaki uygulaması ve sembolik entegre gösterimi

Devrenin Bo çıkışında kullanılan OR kapısı üç girişli olmalıdır. Simülasyon programında üç girişli OR olmadığından devreye bu şekilde bağlanmıştır fakat bu tarz bağlama OR kapısının özelliğinden dolayı sonucu etkilememektedir.

Devreyi iki yarım çıkarıcının birleştirilmesiyle tam çıkarıcı elde ederek tasarlayacağız. Bunun için borç çıkışının karno haritasındaki indirgemeyi tam toplayıcıda yaptığımız gibi özel bir durum olarak tekrar ele alacağız.

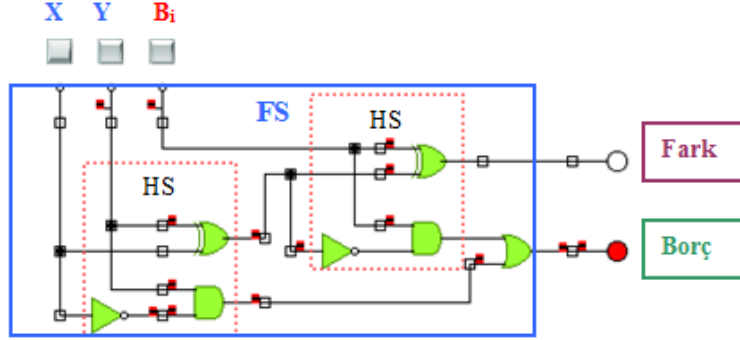
$B_0$	$B_1$	00	01	11	10
X					
0		0	1	1	1
1		4	5	1	6

$$B_o = XYB_i + X'Y + X'Y'B_i$$

$$B_o = B_i(X'Y + XY) + X'Y$$

$$B_o = B_i(X \oplus Y) + X'Y$$

Şekil 3.5: İki yarım çıkarıcının birleştirilmesiyle tam çıkarıcı elde etmek düzenlenen yeni borç çıkışı fonksiyonu



Şekil 3.6: İki yarım çıkarıcının birleştirilmesiyle oluşan tam çıkarıcı

- Simülasyon programında kurduğumuz devre doğru çalıştıysa devremizi kuracağız.

### 3.3. Birin Tümleyenine Göre Çıkarma

İkili sistemde yapılan çıkarma işlemlerinin esasında toplama devreleri yatar. Çıkarma işlemi pozitif bir sayının negatif bir sayıyla toplanmasıdır ( $A + (-B)$ ).

Onluk sistemde sayıların önüne “+” ve ya “-” işaretleri konular. Mikro işlemcili sistemler genelde bir işaret biti içerirler ve çoğunlukla bu bit en büyük değerlikli bitten sonra gelir. İşaret (signed) biti birse sayı negatif, sıfırsa sayı pozitiftir. Üç şekilde kullanılabilir.

- - **0011** ikilik sistemdeki sayının işaretli gösterimi **10011** olarak,
- - **0011** ikilik sistemdeki sayının işaretli gösterimi **11100** bire tümleme alınarak,
- - **0011** ikilik sistemdeki sayının işaretli gösterimi **11101** ikiye tümleme alınarak gösterilebilir.

Tüm mikro işlemcili sistemlerde negatif sayılar, tümleyeni alınarak saklanır, pozitif sayılarınsa tümleyeni alınmaz.

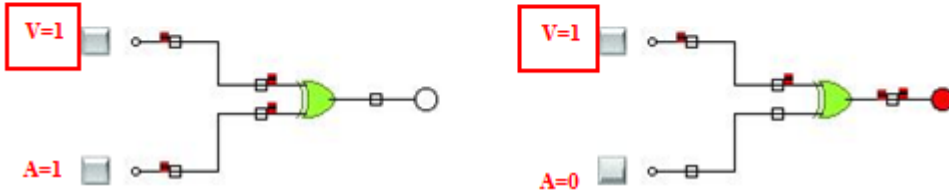
İkilik sistemde tümleyenini bulmak için iki yöntem vardır. Bunlar ikiye ve bire (birin komplementine) göre tümleyenini alma yöntemleridir.

Bire göre tümleme, sıfır yerine bir, bir yerine sıfır konarak yapılır. İkiye göre tümleme ise sıfır yerine bir, bir yerine sıfır konulduktan sonra çıkan yeni sayıya bir eklenerek yapılır.

Bire ve ikiye tümleme işlemlerinin avantajları hem pozitif hem de negatif sayıların kolayca toplanabilmesi ve çıkarılabilmesidir.

Bir bit için bire göre tümleme işlemi lojik EX-OR kapılarıyla yapılabilir. V kontrol girişi olmak üzere; V lojik1 olduğunda çıkış, A girişinin bire göre tümlemesi olur. V kontrol girişi lojik0 olduğunda çıkış, A girişin kendisi olur.

kare nu	V	A	XOR çıkış
0	0	0	0
1	0	1	1
2	1	0	1
3	1	1	0



Şekil 3.7: Bir bitlik ve bire göre tümleme yapan devre

Çıkarma işleminin bire göre tümleneni yöntemine göre yapılması;

$$\begin{array}{r} X \\ - Y \\ \hline B F \end{array} \quad \text{işleminde Y sayısı negatif olan sayıdır. X} \quad \begin{array}{r} + (-Y) \\ \hline Co S \end{array}$$

- İlk işlem olarak Y sayısının bire göre tümleneni alınır. Y pozitif hale gelir.
- $$\begin{array}{r} X \\ + (Y') \\ \hline Co S \end{array}$$

Negatif bir sayıyla pozitif bir sayı toplandığında toplama, pozitif sayı büyükse pozitif, negatif sayı büyükse negatif olacaktır. Çıkan sonucun işaret biti birse sonuç negatif, sıfırsa sonuç pozitiftir. Taşan bitlerle işaret bitini karıştırmayınız. İşaretsiz gösterimlerde işaret biti (unsigned) bulunmadan da işlemler yapılır ve sonuç doğru çıkar.

**Örnek 5:** 3-6= ..... ?

➤ Kırmızılar işaret biti için, mavilerse taşma biti için kullanılmıştır.

$+(3)_{10} = (00011)_2$  (Dört bite göre yapılacak)  $-(6)_{10} = (0110)_2$   $10110$ 'ın bire göre tümleyeni  $11001$ 'dir.

$$\begin{array}{r} 00011 \\ + 11001 \\ \hline \end{array}$$

$010100$   $C_0=0$  taşma yoktur. Bu yüzden sonucun tümleyenini alacağız.  $10011$  sonucu bulduk işaret biti bir olduğundan sonuç  $(1011)_2 = -(3)_{10}$  çıkar.

İşaret Bitsiz Yapılımı:

$$\begin{array}{r} 0011 \\ + 1001 \\ \hline \end{array}$$

$00100$   $C_0=0$  taşma yoktur. Bu yüzden sonucun tümleyenini alacağız.  $0011$  sonucu bulduk.

**Örnek 6 :**  $3-2= \dots ?$

➤ Kırmızılar işaret biti için, mavilerse taşma biti için kullanılmıştır.

$(3)_{10} = (00011)_2$  (Dört bite göre yapılacak)  $-(2)_{10} = (10010)_2$   $10010$ 'ın bire göre tümleyeni  $11101$ 'dir.

$$\begin{array}{r} 00011 \\ + 11101 \\ \hline \end{array}$$

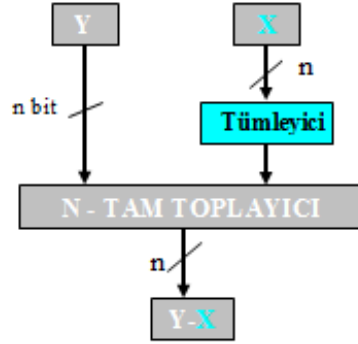
$100000$   $C_0=1$  taşma vardır. Bu yüzden sonucun en sağdaki bitine taşan elde bitini ekleyeceğiz.  $0001$  sonucu bulduk işaret biti sıfır olduğundan sonuç  $(00001)_2 = +(1)_{10}$  çıkar.

İşaret Bitsiz Yapılımı:

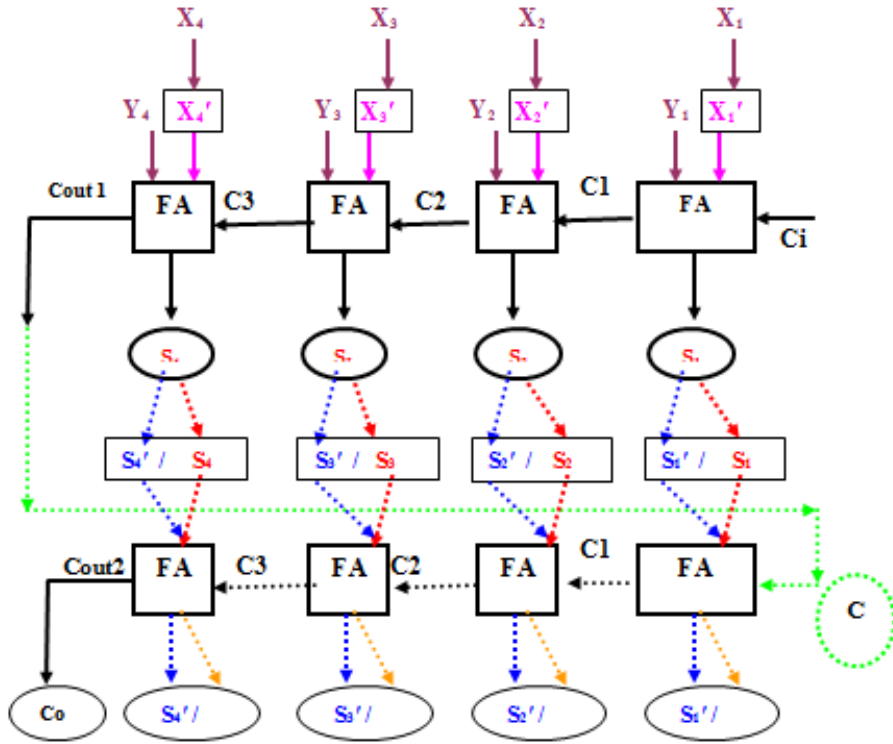
$$\begin{array}{r} 0011 \\ + 1101 \\ \hline \end{array}$$

$10000$   $C_0=1$  taşma vardır. Bu yüzden sonucun en sağdaki bitine taşan elde bitini ekleyeceğiz.  $0001$  sonucu bulduk. İşaret biti olmadığı sayının negatif mi pozitif mi olduğunu taşma bitine göre söyleyeceğiz. Çıkarma işlemlerinde taşma olduğu ( $C_0=1$ ) zaman sayının pozitif olduğu taşma oluşmadığında ise sayının negatif ( $C_0=0$ ) olduğu görülmektedir.





Şekil 3.8: N bitlik paralel çıkarma devresi



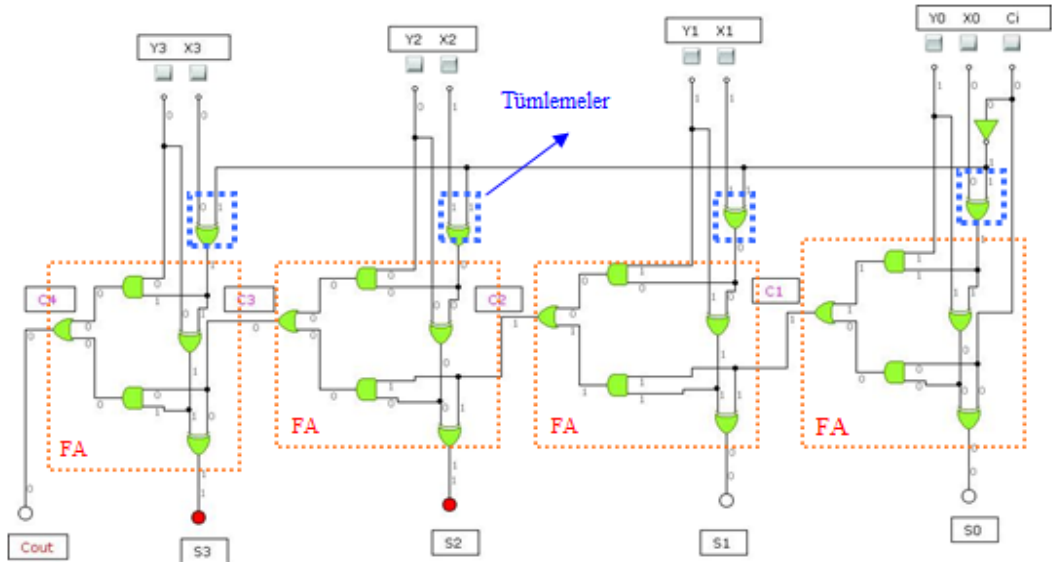
Şekil 3.9: Dört bitlik paralel çıkarma devresi

Şekil 3c.9'da işaret biti (unsigned) kullanılmadan (işaretsiz) dört bit paralel toplama ve bire tümleme yöntemleri kullanılarak çıkarma işleminin yapılması gösterilmiştir. Toplama işlemi sonunda çıkarma işleminin tam sonucunun bulunabilmesi için sonucun pozitif veya negatif olmasına göre (Cout1'e bakarak) Cout1,S4,S3,S2 ve S1'in işlemine devam edilmiştir.

- Toplama işlemi sonunda elde (taşma) yoksa ( $C_o=0$ ) hesaplanan S'nin tümleyeni alınır doğru sonuç elde edilmiş olur. (Elde çıkışının sıfır olması  $|X| > |Y|$  olduğu ve sonucun negatif olduğu anlamına gelir.)
- Elde varsa ( $C_o=1$ ), eldenin S'nin en sağdaki bitine (LSD'ye) eklenmesiyle doğru sonuç elde edilmiş olur. (Elde çıkışının bir olması  $|X| < |Y|$  olduğu ve sonucun pozitif olduğu anlamına gelir.)
- Birinci ve ikinci maddede anlatılan çevirmelerin devreye nasıl bağlandığını çıkarma entegresi konusunda göreceksiniz.

### 3.3.1. Birin Tümleyenine Göre Dört Bitlik İki Sayıyı Çıkaran Devrenin Tasarımı

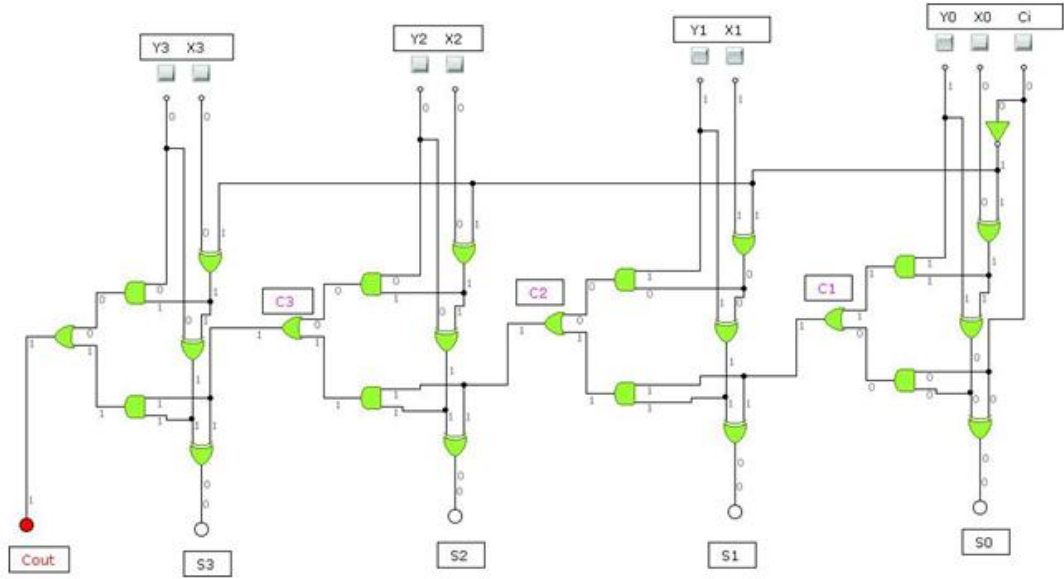
Aşağıdaki şekilde mavi çerçeveli kısımlar tümleme işlemini, kırmızı çerçeveli kısımlarda tam toplayıcıları (Full Adder) göstermektedir.



Şekil 3.10: Dört bitlik iki sayıdan birinden diğerini paralel çıkaran devre ( $3+(-6)$ )

Bu devre, Şekil 3.8'in devre olarak simülasyon programında kurulmuş halidir. Örnek 4'te  $Y=3$ ,  $X=6$ 'dır.  $C_{out}=0$  çıkmıştır.  $C_{out}$  sıfır olduğunda çıkan sayının tümleyeni alınması gerekliydi ve çıkan sayıdan negatif olarak bahsedilmesi gerekiyordu.  $(1100)_2$  sayısının tümleyeni  $(0011)_2$ 'dir.  $(0011)_2 = (3)_{10}$  çıkan sonucun negatif olduğunu da ( $C_{out}=0$ 'dan) biliyoruz o zaman çıkan sonuç  $-3$ 'tür diyebiliriz.

Şekil 2.10'da aynı devrenin Örnek 5'e göre uygulanması vardır.  $C_{out}=1$  çıkmıştır.  $C_{out}=1$  olduğunda sadece  $C_{out}$ 'ta oluşan biri çıkan sayının en küçük değerlikli bitine ekliyorduk. Bu soruda sonuç  $(0000)_2$  çıkmıştır. Buna  $C_{out}$ 'ta bulunan 1 eklendiğinde yeni sonuç  $(0001)_2 = (1)_{10}$  olur. Çıkarma işlemi sonunda sonucun pozitif olduğu zamanlarda  $C_{out}=1$  çıktığından bu işlemin sonucu  $+1$ 'dir.



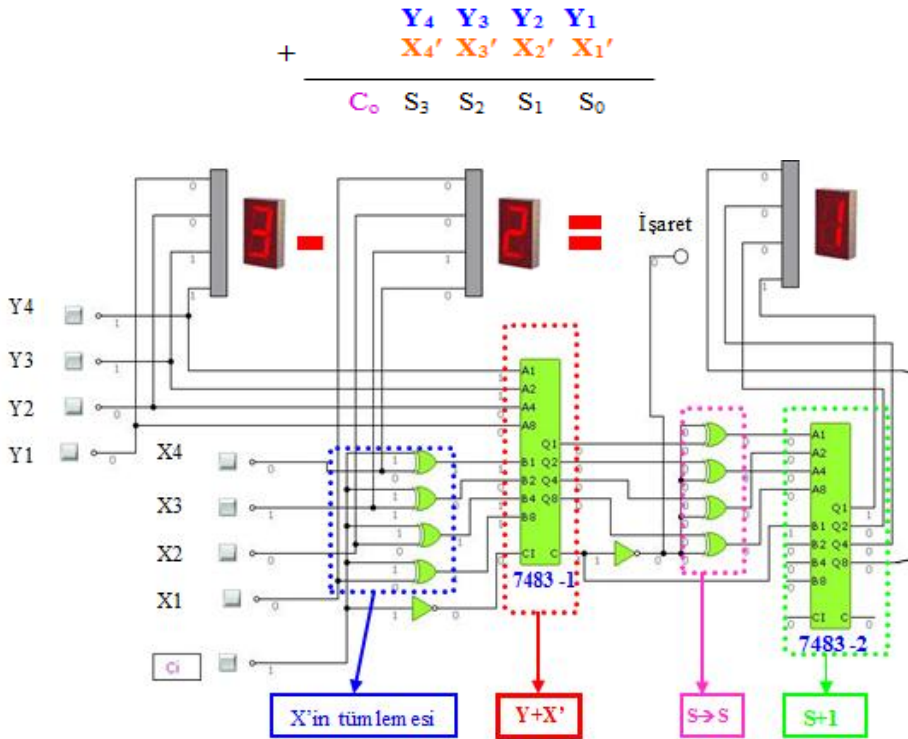
Şekil 3.11: Dört bitlik iki sayıdan birinden diğerini paralel çıkaran devre ( $3+(-2)$ )

### 3.3.2. Çıkarıcı Entegresi

Aritmetik çıkarma işlemi de toplam işleminde oldu gibi 7483 ve toplama yapan diğer entegreler tarafından (4008, 74283) yapılır.

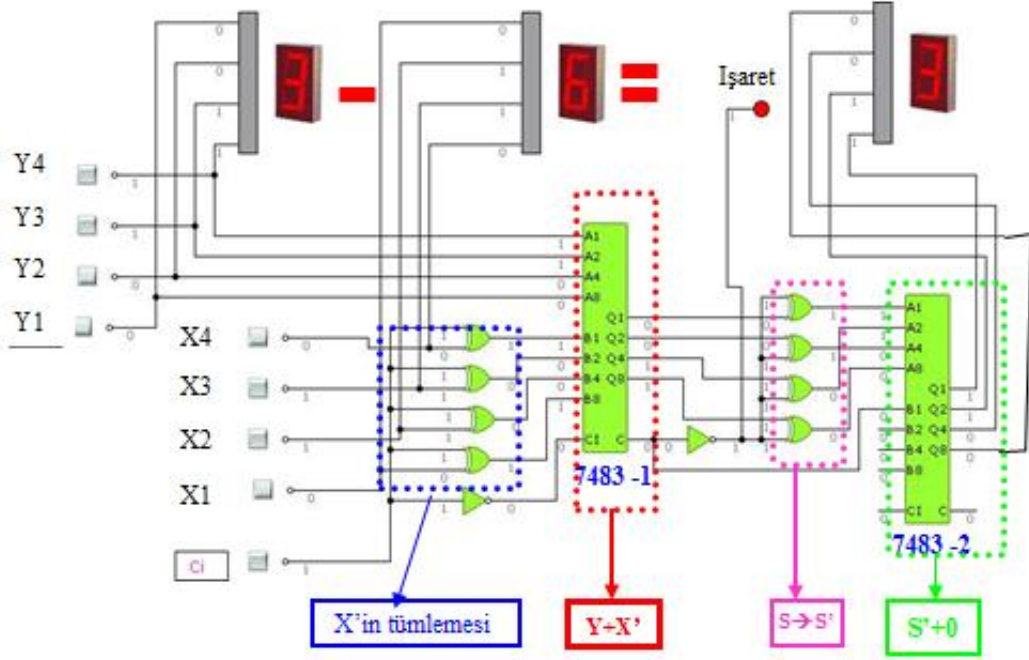
- Göstergeler (Display) ve LED sadece bu devre için ve görsellik amaçlı kullanılmıştır. Devrenin çalışmasına etkileri yoktur. LED yanarsa çıkarma sonucu negatiftir.
- Şekil 3.10 ve 3.11 çıkarma işlemi devresi bire tümlene yöntemine göre kurulmuştur.
- Entegreli çıkarma devresinde ikinci bir toplayıcıya; bir numaralı toplayıcıdan  $C_{out}=1$  çıktığında bu biri birinci toplayıcıdan çıkan sonucun en küçük bitine eklemek için ihtiyaç duyulmuştur. Sıfırın toplamada etkisiz eleman olduğu unutulmamalıdır.
- $Y_1, Y_2, Y_3, Y_4$  giriş bitlerinin her biri sırayla  $A_1, A_2, A_3, A_4$  bacaklarıyla bir numaralı toplayıcı entegresine bağlanacak.
- $X_1, X_2, X_3, X_4$  giriş bitlerinin her biri birer EX-OR kapısının bir bacağına bağlandıktan sonra EX-OR kapılarının çıkışları sırayla  $B_1, B_2, B_3, B_4$  bacaklarıyla bir numaralı toplayıcı entegresine bağlanacak.
- Bir numaralı toplayıcı entegresinin çıkışları  $Q_4, Q_3, Q_2, Q_1$  bitlerinin her biri birer EX-OR kapısının bir bacağına bağlandıktan sonra iki numaralı toplayıcı entegresinin  $A_1, A_2, A_3, A_4$  girişlerine bağlanacak.

- Entegrede  $C_i$  girişi lojik1 ( $C_i=1$ ) olması,  $X_1, X_2, X_3, X_3$  girişlerine bağlı olan EX-OR kapılarının  $X_1, X_2, X_3, X_3$  girişlerinin bire göre tümlenmesi sağlar.  $C_i$  girişinin bir numaralı entegreye bağlanmadan önce not kapısıyla değilinin alınmasının (lojik0 olmasının) sebebi bu devrenin bire göre tümlenmeye yöntemine uygun yapılmaya çalışılmasıdır. (Bir numaralı entegrenin  $C_i$  girişinde lojik1 olması istenirse devrenin baştan ikiye göre tümlenme yöntemi uygun düzenlenmesi gerekir).
- Bu devrenin doğru çalışması için gerekli kilit noktaları, birinci entegrenin elde girişi ( $C_i$ ) ve elde ( $C_{out}$ ) çıkışıdır.
- Birinci toplayıcı entegrenin  $Q_4, Q_3, Q_2, Q_1$  çıkışlarına hangi işlemlerin uygulanacağı birinci entegrenin elde çıkışına ( $C_{out}$ ) göre değişmektedir. Örnek 5 ve Örnek 6 bakıldığında bunun nedenini hatırlayabilirsiniz.
- Şekil 1.12'deki  $3-2=1$  işleminde birinci toplayıcı entegrenin  $C_{out}=0$  olduğundan tam sonuç için  $Q_4, Q_3, Q_2, Q_1$  çıkışlarının bire göre tümlenmesinin alınması gerekmektedir. Tümlenme almak için birinci entegrenin çıkış bitlerine EX-OR kapıları bağlanmıştır. (EX-OR kapısının bir bacağı sabit lojik1 olduğundaysa çıkış fonksiyonu, girişin tersi olur).  $C_{out}$  sıfır olduğundan ikinci toplama entegresiyle yapılan toplama sonucu bozmaz.



Şekil 3.12: Dört bitlik bir sayıdan, dört bitlik bir sayıyı çıkaran devrenin 7483 entegresiyle ve bire göre tümlenme yöntemi kullanılarak yapılmasının simülasyon programında uygulanması

Şekil 3.12'deki 6-3=-3 örneğinde birinci toplama entegresinin Cout çıkışında lojik1 bilgisi vardır. Cout=1 olduğunda Q4,Q3,Q2,Q1 çıkışları değiştirilmeden en küçük değerlikli bitine (LSD) Cout'un eklenmesi gerekmektedir. Çıkışın değiştirilmeden entegreye gitmesi için EX-OR kapısının bir bacağı sabit lojik0 olması gereklidir (EX-OR kapısının bir bacağı sabit lojik0 olduğunda çıkış fonksiyonu, girişe eşit olur). Çıkışa bir (C<sub>out</sub>'u) eklemek içinde iki numaralı toplama entegresini kullandık. İkinci toplama entegresinin kullanmayacağımız ve sonuca etkisi olmayacak bacakları boşta (lojik0) bırakılır.



Şekil 3.13: Dört bitlik bir sayıdan, dört bitlik bir sayıyı çıkaran devrenin 7483 entegresiyle ve bire göre tütleme yöntemi kullanılarak yapılmasının simülasyon programında uygulanması

## UYGULAMA FAALİYETİ

4 bitlik iki sayıdan birinden diğerini çıkaran devreyi 2'ye tümleme yöntemini ve 7483 entegresini kullanarak kurunuz.

İşlem Basamakları	Öneriler
➤ Devrenin kağıt üzerinde çizimini gerçekleştiriniz.	➤ İkilik sistemde 2'ye göre tümlemenin nasıl yapıldığını öğrenmeniz bu soruyu daha kolay çözmenizi sağlayacaktır.
➤ Kağıt üzerinde tasarladığınız devreyi simülatör programda uygulayınız.	➤ Devrenizi çizmeden önce öğrenme faaliyeti tekrar etmeniz devrenizi kurmanızda kolaylık sağlayacaktır.
➤ Devreniz doğru çalıştıysa devrede kullanacağınız elemanları belirleyiniz.	➤ Kullanacağınız elemanları öğretmeninizden isteyiniz.
➤ Devrenizi elektronik ortamda kurunuz.	

## ÖLÇME VE DEĞERLENDİRME

Aşağıdaki soruları dikkatlice okuyunuz ve doğru seçeneği işaretleyiniz.

1. Onluk sistemdeki  $3-5= \dots$  İşleminin ikilik sistemde gösterimi ve sonucu aşağıdakilerden hangisidir? Sonuç tümlene alınmadan önceki haliyle verilmiştir.

A) 
$$\begin{array}{r} 0011 \\ +0101 \\ \hline 1000 \end{array}$$
 B) 
$$\begin{array}{r} 0011 \\ +1010 \\ \hline 1101 \end{array}$$
 C) 
$$\begin{array}{r} 1100 \\ +1010 \\ \hline 10110 \end{array}$$
 D) 
$$\begin{array}{r} 1100 \\ +0101 \\ \hline 10001 \end{array}$$

2. Onluk sistemdeki 9 sayısının ikilik sistemdeki bire göre tümlenmesi aşağıdakilerden hangisidir?

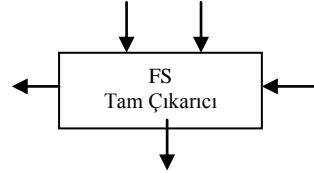
A) 0110 B) 0111 C) 1000 D) 1001

3. Bir bitlik iki sayıyı birbirinden çıkaran devre aşağıdakilerden hangisidir?

A) Tam çıkarıcı  
B) Yarım çıkarıcı  
C) Tam toplayıcı  
D) Çarpıcı

4. Aşağıdakilerden hangisi tam çıkarıcı devresinde bulunan giriş veya çıkışlardan biri değildir.

A) Elde girişi (CarryInput)  
B) Borç çıkışı (Borrow Output)  
C) Borç girişi (Borrow input)  
D) Fark (difference)



5. X kontrol girişinin durumuna göre girişin bire göre tümlenmesini alabilen veya girişin aynısını çıkışa yansıtan kapı hangisidir?

A) OR B) NOR C) AND D) EX-OR

6. Onluk sistemdeki -6 sayısının ikilik sistemdeki bire göre tümlenmesi aşağıdakilerden hangisidir? İşaret biti de gösterilecektir.

A) 0110 B) 0111 C) 1000 D) 1001

7. Çıkarma işlemlerinde kullanılan 74--- serisi entegre aşağıdakilerden hangisidir?

A) 7485 B) 4070 C) 7483 D) 74284

8. Onluk sistemdeki  $9-7 = \dots$  İşleminin ikilik sistemde gösterimi ve sonucu aşağıdakilerden hangisidir? Sonuç en son işlemi yapılmamış elde çıkışıyla olan haliyle verilmiştir.

A) 11001 B) 00111 C) 10001 D) 01110

9. Birin komplementine göre dört bitlik paralel çıkarma işlemi yapan devrede hangi kalıp sembolik tümleşik devre kullanılır?

- A) Tam toplayıcı
- B) Yarım toplayıcı
- C) Yarım çıkarıcı
- D) Not kapısı

10. İki bitlik bir sayıdan sabit iki sayısını çıkaran devrenin doğruluk tablosu aşağıdakilerden hangisidir?

Hücre nu	$X_1$	$X_0$	borç	fark
0	0	0	1	1
1	0	1	0	0
2	1	0	1	0
3	1	1	1	0

A)

Hücre nu	$X_1$	$X_0$	borç	fark
0	0	0	0	1
1	0	1	0	0
2	1	0	0	1
3	1	1	1	0

B)

Hücre nu	$X_1$	$X_0$	borç	fark
0	0	0	0	0
1	0	1	1	1
2	1	0	0	1
3	1	1	1	0

C)

Hücre nu	$X_1$	$X_0$	borç	fark
0	0	0	1	1
1	0	1	0	0
2	1	0	0	1
3	1	1	1	0

D)

## DEĞERLENDİRME

Cevaplarınızı cevap anahtarıyla karşılaştırınız. Yanlış cevap verdiğiniz ya da cevap verirken tereddüt ettiğiniz sorularla ilgili konuları faaliyete geri dönerek tekrarlayınız. Cevaplarınızın tümü doğru ise bir sonraki öğrenme faaliyetine geçiniz.



# ÖĞRENME FAALİYETİ-4

## AMAÇ

Çarpma tüm devre elemanını gerektiğinde geliştirdiği devrelerde kullanabileceksiniz.

## ARAŞTIRMA

- İkilik sistemde çarpma işlemi yapılırken nelere dikkat edildiğini araştırınız.
- Çarpma sembolü ile gösterilen lojik kapıyı araştırınız.
- Çarpma işlemi yapan entegrelerinin bacak bağlantılarını yapan entegreleri araştırınız.

## 4. ÇARPMA DEVRELERİ

### 4.1. Çarpma Devresi (Multiplier)

Çarpma devreleri aşağıda sıralanmıştır.

#### 4.1.1. Bir Bitlik İki Sayıyı Çarpan Devre

Bir bitlik iki sayının çarpılmasında hatırlamamız gereken en önemli nokta çarpmada sıfırın yutan eleman, birinde etkisiz eleman olduğudur. İki tane bir bitlik sayının dört olasılığı vardır bunlar;

$$0*0=0$$

$$0*1=0$$

$$1*0=0$$

$1*1=1$  görüldüğü üzere çıkış bir bit olacaktır. Ne borç ne de elde çıkışı gibi bir çıkışa ihtiyaç vardır.

Lojik devre tasarım sırasına göre sorunun cevabını düşündükten sonraki ilk işlem olarak tablosunu daha sonrada karno haritasını çizip sadeleştirmeyi yapacağız.

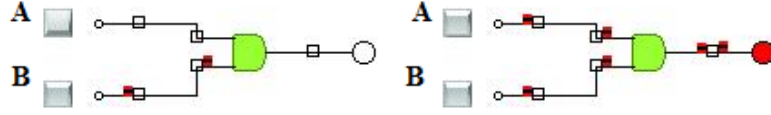
kare nu	A	B	A*B
0	0	0	0
1	0	1	0
2	1	0	0
3	1	1	1

C	B	0	1
A	0	0	0
1	0	0	1
1	1	0	1

Çarpma=AB

Şekil 4.1: Bir bitlik iki sayıyı çarpan devrenin doğruluk tablosu ve karno haritası

Doğruluk tablosuna bakıldığında karno haritası çizilmeden de çarpma işleminin çıkışının and kapısı çıkışını verdiği görülmektedir.



Şekil 4.2(a): Bir bitlik iki sayıyı çarpan devre

#### 4.1.2. İki Bitlik İki Sayıyı Çarpan Devre

- Devrenin doğruluk tablosunu hazırlayacağız:

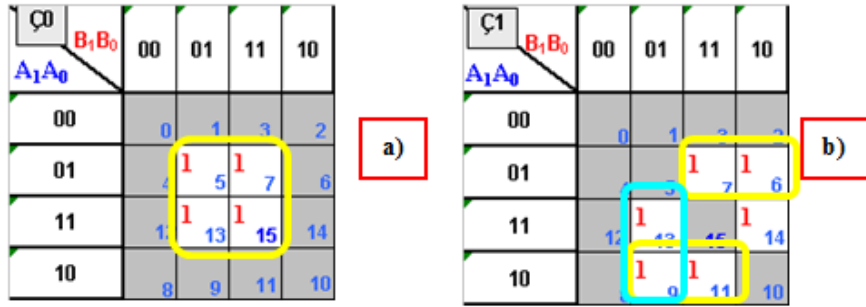
İşaretsiz iki bitlik iki sayıyı çarpan devresinin dört girişi vardır. Doğruluk tablosunu çizebilmemiz için bu devrenin çıkış sayısını hesaplamamız gerekmektedir. Çıkışın kaç bit olduğunu bulmak için iki bitlik en büyük sayının kaç olduğu bulmamız ve bu sayıyı kendisiyle çarpmamız yeterli olacaktır. İkilik sistemdeki iki bitlik en büyük sayı 11'dir bu da onluk sistemdeki 3 sayısına eşittir. ( $3 \times 3 = 9$ ) Dokuz sayısı iki bitlik iki sayının çarpılmasında çıkabilecek en büyük sayıdır ve ikilik sistemdeki karşılığı dört bitle gösterilir (1001). Buna göre iki bitlik iki sayının çarpılması devresinin dört tane çıkış fonksiyonu olacaktır.

hücre no	(A) <sub>10</sub>	(B) <sub>10</sub>	A <sub>1</sub>	A <sub>0</sub>	B <sub>1</sub>	B <sub>0</sub>	Ç <sub>3</sub>	Ç <sub>2</sub>	Ç <sub>1</sub>	Ç <sub>0</sub>	(çarpma) <sub>10</sub>
0	0	0	0	0	0	0	0	0	0	0	0
1	0	1	0	0	0	1	0	0	0	0	0
2	0	2	0	0	1	0	0	0	0	0	0
3	0	3	0	0	1	1	0	0	0	0	0
4	1	0	0	1	0	0	0	0	0	0	0
5	1	1	0	1	0	1	0	0	0	1	1
6	1	2	0	1	1	0	0	0	1	0	2
7	1	3	0	1	1	1	0	0	1	1	3
8	2	0	1	0	0	0	0	0	0	0	0
9	2	1	1	0	0	1	0	0	1	0	2
10	2	2	1	0	1	0	0	1	0	0	4
11	2	3	1	0	1	1	0	1	1	0	6
12	3	0	1	1	0	0	0	0	0	0	0
13	3	1	1	1	0	1	0	0	1	1	3
14	3	2	1	1	1	0	0	1	1	0	6
15	3	3	1	1	1	1	1	0	0	1	9

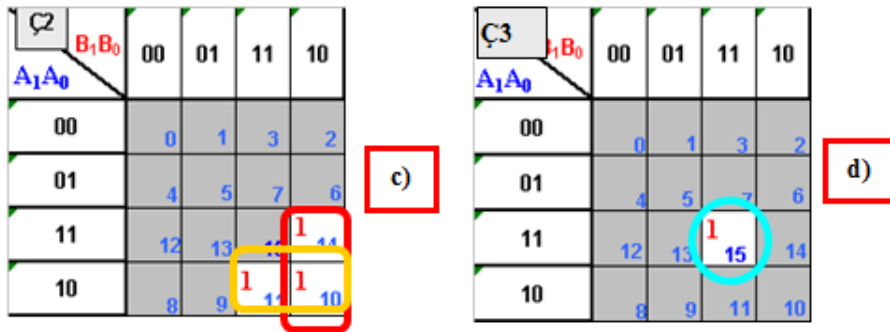
Şekil 4.2(b): İşaretsiz iki bitlik iki sayıyı çarpan devrenin doğruluk tablosu

Doğruluk tablosunun çıkışları iki sayının onluk sistemdeki karşılıklarının birbiriyle çarpıldıktan sonra çıkan sayının ikilik sisteme çevrilmesiyle bulunur. Tabloyu dikkatle inceleyiniz.

- Doğruluk tablosuna bakarak bütün çıkışların karno haritalarının çizeceğiz,



Şekil 4.2(c): İki bitlik iki sayıyı çarpan devrenin çıkışlarının karno haritaları

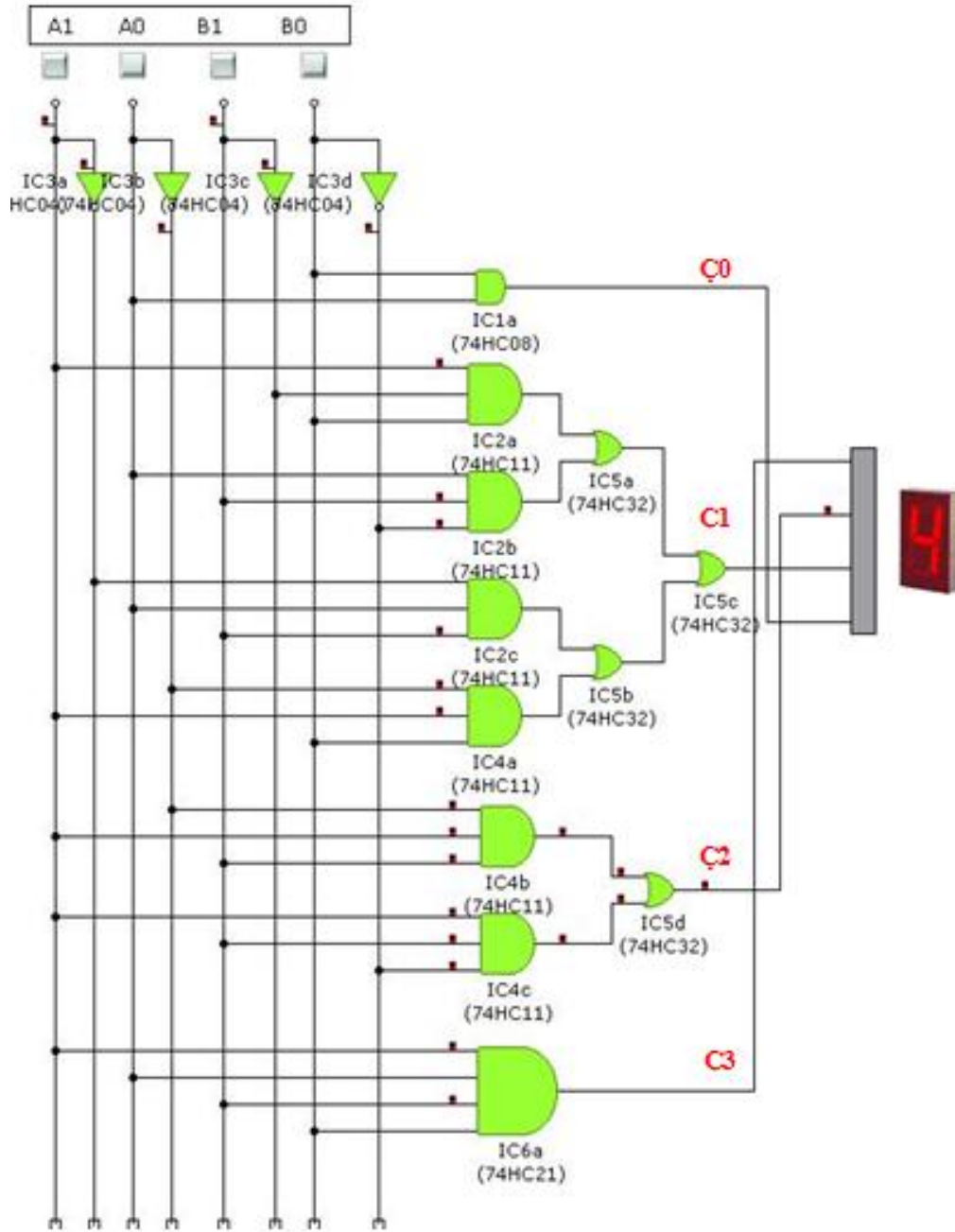


Şekil 4.2(d): İki bitlik iki sayıyı çarpan devrenin çıkışlarının karno haritaları

- Karno haritalarının sadeleştirmelerini yapıp çıkış fonksiyonlarını bulacağız,

Karno haritalarında sadeleştirilmelerinden sonra elde edilen fonksiyon çıkış fonksiyonları:

- $\text{Ç0} = A_0B_0$
  - $\text{Ç1} = A_1B_1'B_0 + A_0B_1B_0' + A_1'A_0B_1 + A_1A_0'B_0$
  - $\text{Ç2} = A_1A_0'B_1 + A_1B_1B_0'$
  - $\text{Ç3} = A_1A_0B_1B_0$
- Çıkış fonksiyonlarına göre devrenin tasarımını çizeceğiz, ve çizdiğimiz devreyi bilgisayar üzerinde simülasyon programlarında deneyeceğiz,

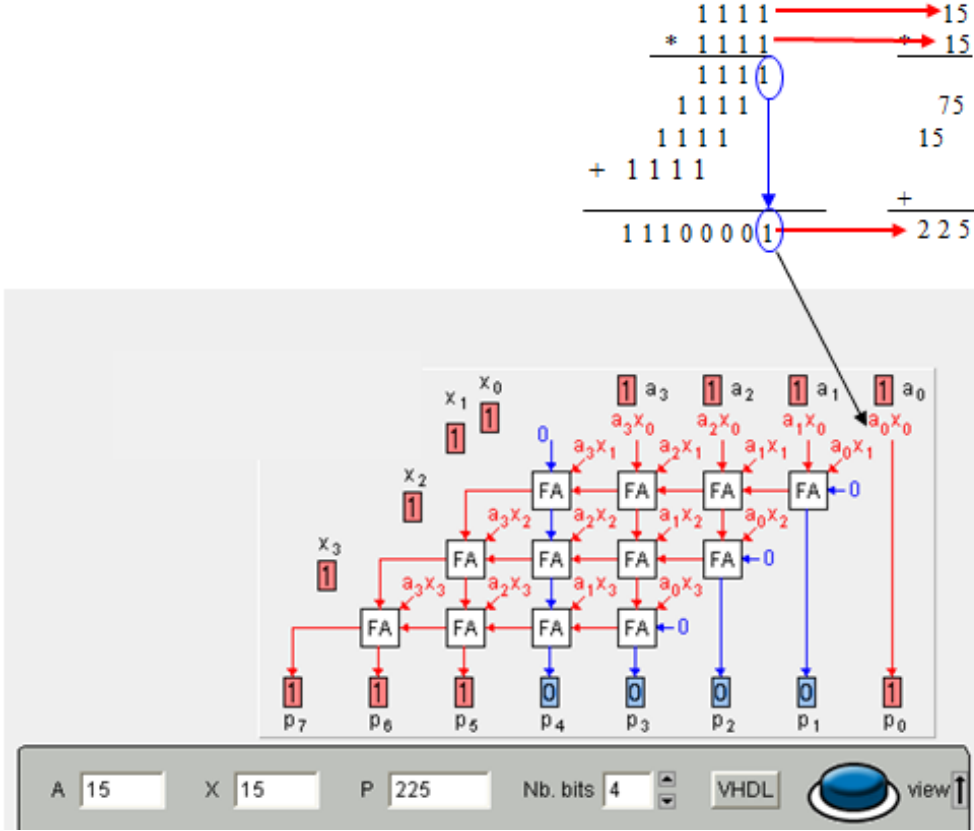


Şekil 4.3: İşaretsiz iki bitlik iki sayıyı çarpan devrenin simülör programda (crocodiletechnology) uygulanması

- Simülör programda kurduğumuz devre doğru çalıştıysa devremizi kuracağız.

Çarpılacak sayıların pozitif mi, negatif mi olduğudur. Sayıların negatif olması çarpma işlem sıralarını etkilemez. İşaret bitlerine bakılarak yapılabilir.

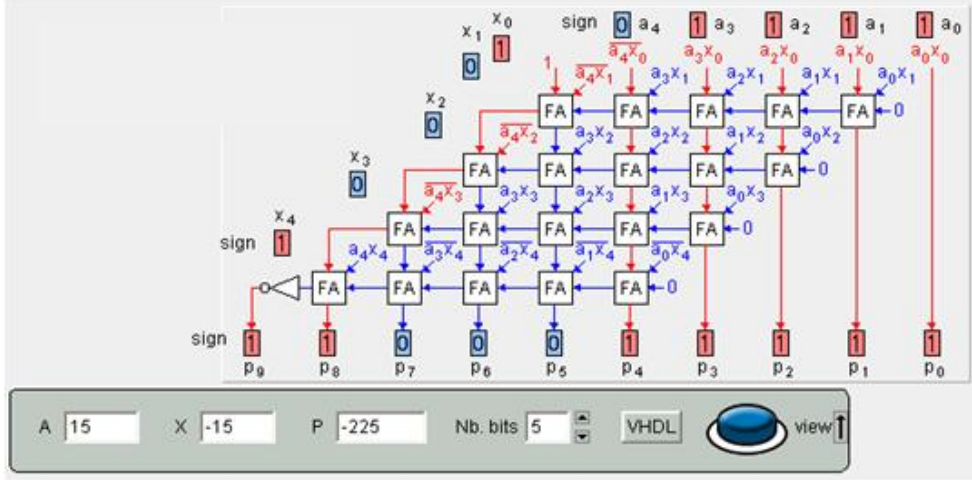
Çarpma işlemi paralel bağlı tam toplayıcılarla gerçekleştirilebilir.



Şekil 4.4: Tam toplayıcılar kullanılarak yapılan işaretsiz iki pozitif sayıyı çarpma işlemi (15\*15)

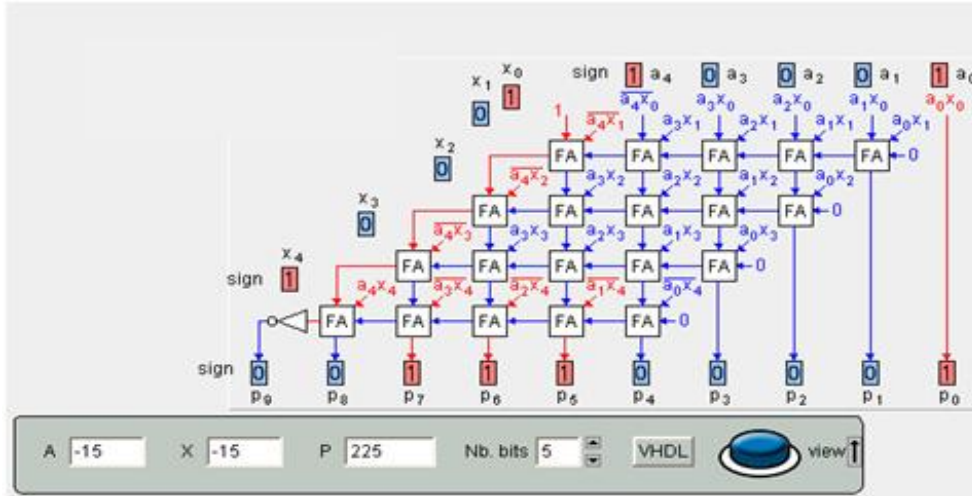
- Devrede A sayısı çarpılan sayı X sayısı çarpan sayıdır. A sayısı her bir satırda X sayısının sabit bir bitiyle (Alt satıra geçerken X sayısının kullanılan biti de X sayısının bir sonraki bitiyle değişir) çarpılır. Bu işlemlerden çıkan sonuçlar tam toplayıcıların girişlerini oluşturur.
- $A_0 * X_0$ 'tan çıkan sonuç matematik kurallarına uygun olarak direkt aşağıya indirilmiştir.
- Birinci sıradaki en sağdaki tam toplayıcının birinci giriş değeri olarak  $A_1 * X_0$  işleminin sonucu, ikinci giriş değeri olarak  $A_0 * X_1$  işleminin sonucu kullanılmış. İlk toplamada elde olmadığından elde girişinden sıfır verisi verilmiştir.
- İkinci satırının en sağdaki tam toplayıcının birinci giriş değerini  $A_0 * X_2$  çarpımının sonucu ikinci giriş değerini de bir üstündeki tam toplayıcının toplam çıkışından gelen sonuç oluşturmaktadır. Her bir toplayıcının toplam çıkışı kendine sonra gelen toplayıcının girişlerinden bir tanesini oluşturmaktadır. En

soldaki toplayıcılarda farklı olarak giriş değeri bir üstteki en soldaki toplayıcının taşıma çıkışından alınmaktadır. En sağda bulunan toplayıcıların toplam çıkışları da direk en aşağı indirilir. Bunun nedeni çarpma işlemi kurallarına bağlı olarak bir alt satıra geçince bir bit sola kayarak yazılmaya başlanmasıdır.

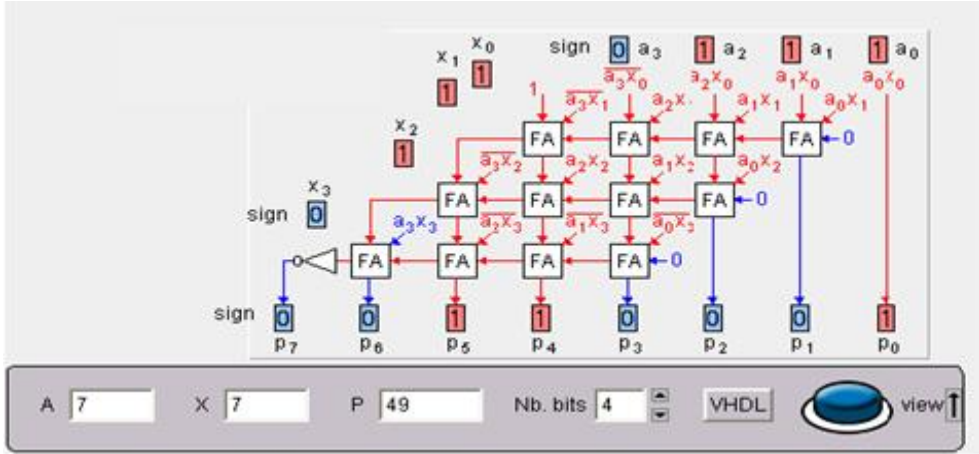


**Şekil 4.5: Tam toplayıcılar kullanılarak yapılan işaretli bir negatif bir pozitif sayıyı çarpma işlemi (15\* (-15))**

Şekil 4.5'teki işaretli çarpma işleminde çarpacağımız sayılardan negatif olanının çarpma yapmadan önce 2'ye göre tümleyeni alınmış. Daha sonra çarpma işlemi yapılmıştır. Çıkan sonuç negatif olduğundan tekrar 2'ye göre tümleyeni alınırsa sonuç 11110001 çıkacaktır. Çarpma işleminde negatif \* pozitif= negatiftir. Sonuçtaki işaret bitinin negatif olması için 1 olması gerekmektedir. Bunun için en son toplam işleminin çıkışının değili alınmıştır.



**Şekil 4.6: Tam toplayıcılar kullanılarak yapılan işaretli iki negatif sayıyı çarpma işlemi ((-15) \* (-15))**



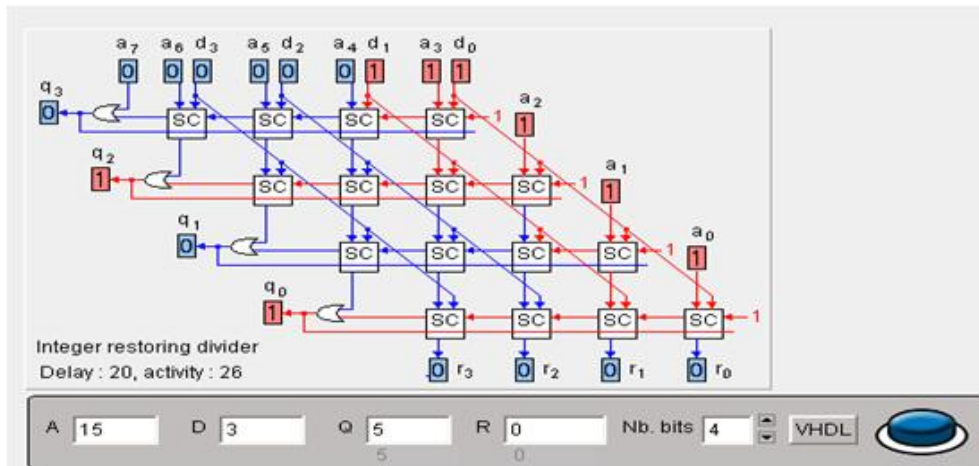
Şekil 4.7: Tam toplayıcılar kullanılarak yapılan işaretli iki pozitif sayıyı çarpma işlemi (7 \*7)

$$\begin{array}{r}
 011 \\
 * 110 \\
 \hline
 00 \\
 + 11 \\
 \hline
 1110
 \end{array}$$

$3 * (-2) = 6$

İşaretili çarpma işlemleri varsa negatif sayıların tümleyenini almadan sayıları çarpıp en son sonucun başına işaret biti getirilerek de yapılabilir.

Bölme (divider) işlemi de çarpma işlemine benzer şekilde yapılır. Aralarındaki fark ise tam toplama devreleri yerine tam çıkarma devreleri kullanılarak yapılmasıdır. Şekil 4.8’de görülen tam toplayıcılar yerine tam çıkarıcılar bulunur.



Şekil 4.8: Tam çıkarıcılar kullanılarak yapılan işaretsiz bölme işlemi (15/3)

A→ bölüneni, D→ böleni, Q→bölümü, D→ kalanı göstermektedir.

Çıkarma işlemleri de negatif sayıların toplamı olduğundan gerekli düzenlemelerle tam toplayıcılar kullanılarak da bölme işlemi yapılabilir.

## 4.2. Çarpma Entegresi

İkilik sistemdeki çarpma işlemlerini en hızlı gerçekleştiren devreler paralel çarpma devreleridir.

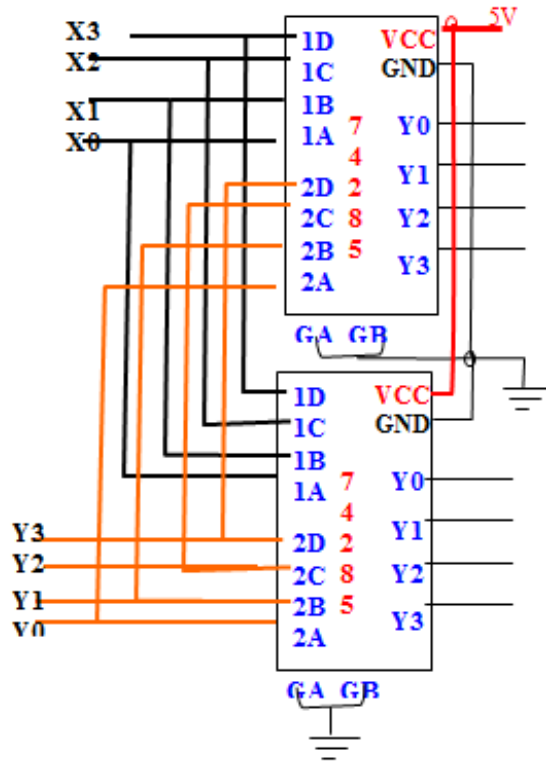
74285 entegresi kullanarak yapacağımız çarpma devresinde dikkat etmemiz gereken önemli bir nokta, bu devrenin çıkış sayısını bilmemizdir. Dört bitlik en büyük iki sayıyı çarptıktan sonra çıkan sayı sekiz bittir. Buna göre bu devrenin çıkışı sekiz bit olmalıdır. 74285 entegresinin bacak bağlantılarına bakıldığında dört çıkışı olduğu görülmektedir. Sekiz çıkışı elde etmek için iki adet entegreyi birbirine paralel bağlamak gerekmektedir.

2C	1	16	VCC
2B	2	15	2D
2A	3	7	/GA
1D	4	4	/GB
1A	5	2	Y0
1B	6	8	Y1
1C	7	5	Y2
GND	8	9	Y3

Şekil 4.9: Dört bitlik iki sayıyı çarpan entegrenin (74285) bacak bağlantıları

- Sekiz bitlik sonuç için iki adet 74285 kullanmamız gerekecektir.
- Birinci ve ikinci entegredeki 1A, 1B, 1C, 1D girişlerine dört bitlik birinci sayının girişleri bağlanır.
- Birinci ve ikinci entegredeki 2A, 2B, 2C, 2D girişlerine dört bitlik ikinci sayının girişleri bağlanır.
- Paralel bağlı iki entegrede de girişler aynı olacağından sonuç etkilenmeyecektir.
- Her iki entegrenin Vcc uçlarına 5V'a, GND uçlarına da toprağa bağlanacaktır.





Şekil 4.10: Dört bitlik iki sayıyı çarpan devrenin 74285 entegresiyle yapılmış hali

## UYGULAMA FAALİYETİ

3 bitlik A sayısı tek olduğunda  $A*2$ , çift olduğunda  $A/2$  işlemlerini yaptıran devreyi tasarlayınız.

İşlem Basamakları	Öneriler
➤ Devre çıkışı lojik1'de mi yoksa Lojik0'da mı aktif olacak belirleyiniz.	➤ Çıkışın Lojik1 de aktif olması sizin soruyu daha kolay çözmenizi sağlayacaktır.
➤ Devrenin girişinin ve çıkışının kaç bitten oluşacağını belirleyiniz.	➤ Hata olmaması için çıkışın bit sayısının taşma olmayacak şekilde düşülmesi gereklidir.
➤ Devrenin doğruluk tablosunu hazırlayınız.	
➤ Doğruluk tablosuna bakarak karno haritasının içeri doldurunuz.	➤ Giriş değişken sayılarına bakarak karno haritasının kaç kareden oluşacağını 2N formülüne göre hesaplayabilirsiniz. Karno haritasının kalıbını hazırlarken haritadaki karelerin numaralarının sıralamalarının Gray koda göre verildiğini unutmayınız. Her bir çıkış için ayrı karno haritası hazırlayınız.
➤ Karno haritasına göre indirgeme yapıp her bir çıkışın fonksiyonunu bulunuz.	
➤ Çıkış fonksiyonuna bakarak devre için gerekli elemanları belirleyiniz.	➤ Dijital elektronik devre elemanları 5V ile çalışır. Kullanılan entegrelerin bacak bağlantılarının bulunduğu bir belgeyi yanınızda bulundurun.
➤ Devrenin kağıt üzerinde çizimini gerçekleştiriniz.	
➤ Çıkış fonksiyonunuzu verecek devreyi bilgisayardaki simülasyon programlarında kurup, devreyi çalıştırınız.	➤ Plaket (breadboard) kullanmayı hatırlamıyorsanız devreyi yanlış kurmamak için uygulama öğretmeninizden ya da laboratuvar eşinizden nasıl kullanıldığını hatırlatmasını isteyiniz.
➤ Bilgisayardaki devreniz doğru sonuçlar veriyorsa devrede kullanılan lojik elemanları son olarak tekrar belirleyip, devrenizi plakete (breadboard) veya lojik eğitim setinize kurunuz.	
➤ Devreniz doğru çalıştıysa devrede kullanacağınız elemanları belirleyiniz.	
➤ Devrenizi elektronik ortamda kurunuz.	

## ÖLÇME VE DEĞERLENDİRME

Aşağıdaki soruları dikkatlice okuyunuz ve doğru seçeneği işaretleyiniz.

1.  $1 \times 1$  işleminin sonucu kaçtır?  
A) 0                      B) 1                      C) 10                      D) 01
2.  $11 \times 01$  işleminin sonucu kaç bittir?  
A) 4                      B) 3                      C) 2                      D) 1
3. 4 bitlik iki sayının sonucunu hesaplayan entegre bloğunun çıkış kaç bit olmalıdır?  
A) 4                      B) 8                      C) 16                      D) 32
4. 74285 entegresi ile 8 bit çıkış elde etmek için entegre bağlantı şekilleri nasıl olmalıdır?  
A) Paralel                      B) Seri                      C) Karmaşık                      D) Hiçbiri
5. Çarpma işlemi yapan kapı aşağıdakilerden hangisidir?  
A) AND                      B) OR                      C) NOT                      D) EX-OR

## DEĞERLENDİRME

Cevaplarınızı cevap anahtarıyla karşılaştırınız. Yanlış cevap verdiğiniz ya da cevap verirken tereddüt ettiğiniz sorularla ilgili konuları faaliyete geri dönerek tekrarlayınız. Cevaplarınızın tümü doğru ise bir sonraki öğrenme faaliyetine geçiniz.

# ÖĞRENME FAALİYETİ-5

## AMAÇ

Karşılaştırma tüm devre elemanını gerektiğinde geliştirdiği devrelerde kullanabileceksiniz.

## ARAŞTIRMA

- 7485 karşılaştırma entegresinin bacak bağlantılarını ve çarpma işlemini yapan başka entegreleri araştırınız.
- Karşılaştırma işlemlerinin kullanıldığı sistemleri ve devreleri araştırınız.

## 5. KARŞILAŞTIRMA DEVRELERİ

Karşılaştırma işlemi, girişlerden birinin (A) diğerine (B) göre durumunun çıkışı oluşturmasıdır. Karşılaştırma devresinin üç çıkışı vardır. Bir girişin diğerinden büyük ( $A > B$ ), küçük ( $A < B$ ) ya da iki girişin birbirine eşit ( $A = B$ ) olduğu durumları gösterir.

Aritmetik devrelerde önemli rol oynar. Toplama ve çıkartma yapan devrelerde, girişleri aynı ve ya farklı iken çıkış vermesi beklenen ve buna göre başka bir sistemi kontrol eden devrelerde, ikilik sistemde adres karşılaştırması yapan devrelerde kullanılabilir.

### 5.1. Yarım Karşılaştırıcı

Yarım karşılaştırma işlemi bir bitlik iki sayıyı birbiriyle karşılaştıran devredir. Bu devrenin çıkışı sayıların birbirine eşit mi yoksa değilmi olmasına göre belirlenir. Bunlara göre bu devrenin iki girişi (A,B) iki de çıkışı ( $A = B$  ,  $A \neq B$ ) bulunmaktadır.

#### 5.1.1. Yarım Karşılaştırıcı Devrenin Tasarımı

- Devrenin doğruluk tablosunu hazırlayıp doğruluk tablosuna göre her iki çıkışın karno haritalarını çizeceğiz,

Hücre nu	A	B	$A=B$	$A \neq B$
0	0	0	1	0
1	0	1	0	1
2	1	0	0	1
3	1	1	1	0

$A \neq B$	B	0	1
A	0	0 <sub>0</sub>	1 <sub>1</sub>
1	0	1 <sub>0</sub>	0 <sub>3</sub>

$A = B$	B	0	1
A	0	1 <sub>0</sub>	0 <sub>1</sub>
1	0	0 <sub>2</sub>	1 <sub>3</sub>

Şekil 5.1: Yarım karşılaştırıcı devrenin doğruluk tablosu ve karno haritaları

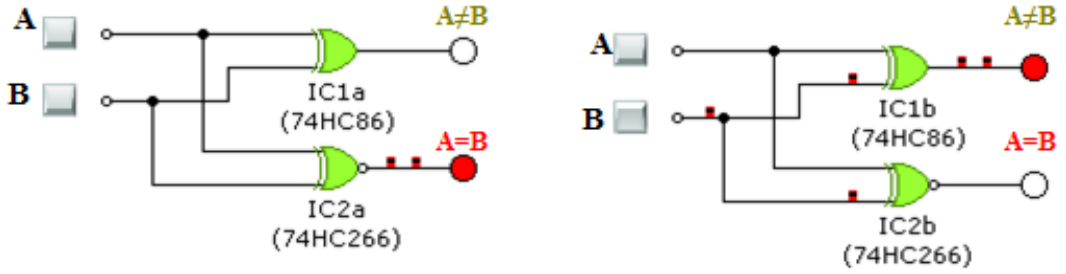
- Karno haritalarının sadeleştirmelerini yapıp çıkış fonksiyonlarını bulacağız,

$$(A \neq B) = A'B + B'A = A \oplus B$$

$$(A = B) = A'B' + AB = (A \oplus B)' = A \odot B$$

En basit karşılaştırmayı yapan kapının EX-OR kapısı olduğu görülmektedir.

- Çıkış fonksiyonlarına göre devrenin tasarımını çizeceğiz, ve çizdiğimiz devreyi bilgisayar üzerinde simülasyon programlarında deneyeceğiz,



Şekil 5.2: Yarım karşılaştırıcı devresinin simülasyon programında (crocodiletechnology) uygulanması

- Simülasyon programında kurduğumuz devre doğru çalıştıysa devremizi kuracağız.

## 5.2. Tam Karşılaştırıcı

Tam karşılaştırıcı devrelerde bir bitlik iki giriş (A,B) ve üç çıkış ( $A > B$ ,  $A < B$ ,  $A = B$ ) bulunur.

### 5.2.1. Tam Karşılaştırıcı Devrenin Tasarımı

- Devrenin doğruluk tablosunu hazırlayacağız.

Hücre nu	A	B	A=B	A>B	A<B
0	0	0	1	0	0
1	0	1	0	1	0
2	1	0	0	0	1
3	1	1	1	0	0

Şekil 5.3: Tam karşılaştırıcı devrenin doğruluk tablosu

- Doğruluk tablosuna göre her iki çıkışın karno haritalarını çizeceğiz.

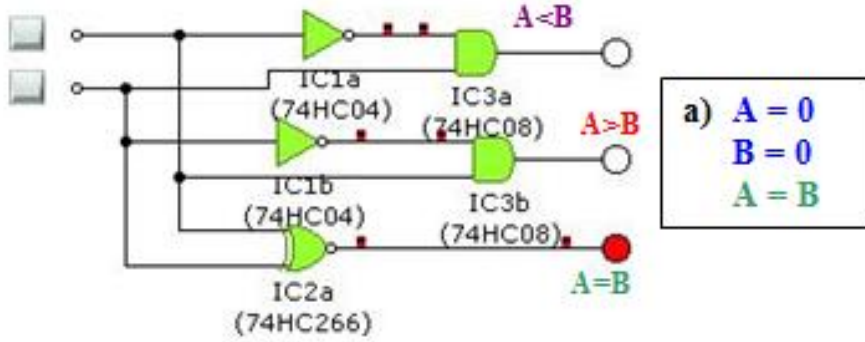
A=B	B	0	1
A \			
0		1 <sub>0</sub>	0 <sub>1</sub>
1		0 <sub>2</sub>	1 <sub>3</sub>

A>B	B	0	1
A \			
0		0 <sub>0</sub>	0 <sub>1</sub>
1		1 <sub>2</sub>	0 <sub>3</sub>

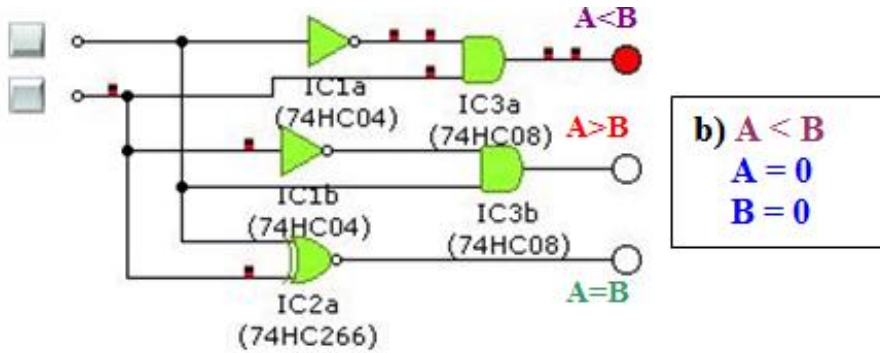
A<B	B	0	1
A \			
0		0 <sub>0</sub>	1 <sub>1</sub>
1		0 <sub>2</sub>	0 <sub>3</sub>

Şekil 5.4: Tam karşılaştırıcı devrenin karno haritası

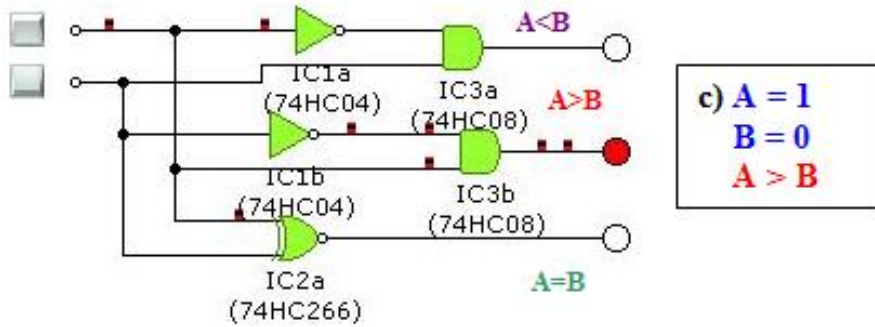
- Karno haritalarının sadeleştirmelerini yapıp çıkış fonksiyonlarını bulacağız,
  - $(A>B) = AB'$
  - $(A=B) = A'B' + AB = (A \oplus B)' = A \odot B$
  - $(A<B) = A'B$
- Çıkış fonksiyonlarına göre devrenin tasarımını çizeceğiz, ve çizdiğimiz devreyi bilgisayar üzerinde simülör programlarda deneyeceğiz,



Şekil 5.5(a): Tam karşılaştırıcı devresinin simülör programda uygulanması ve çalıştırılması



Şekil 5.5(b): Tam karşılaştırıcı devresinin simülör programda uygulanması ve çalıştırılması



Şekil 5. 5(c): Tam karşılaştırıcı devresinin simülasyon programda uygulanması ve çalıştırılması

- Simülasyon programda kurduğumuz devre doğru çalıştıysa devremizi kuracağız.

### 5.3. Karşılaştırıcı Entegreleri

7485 entegresi dört bitlik iki sayıyı karşılaştırmak için kullanılır.

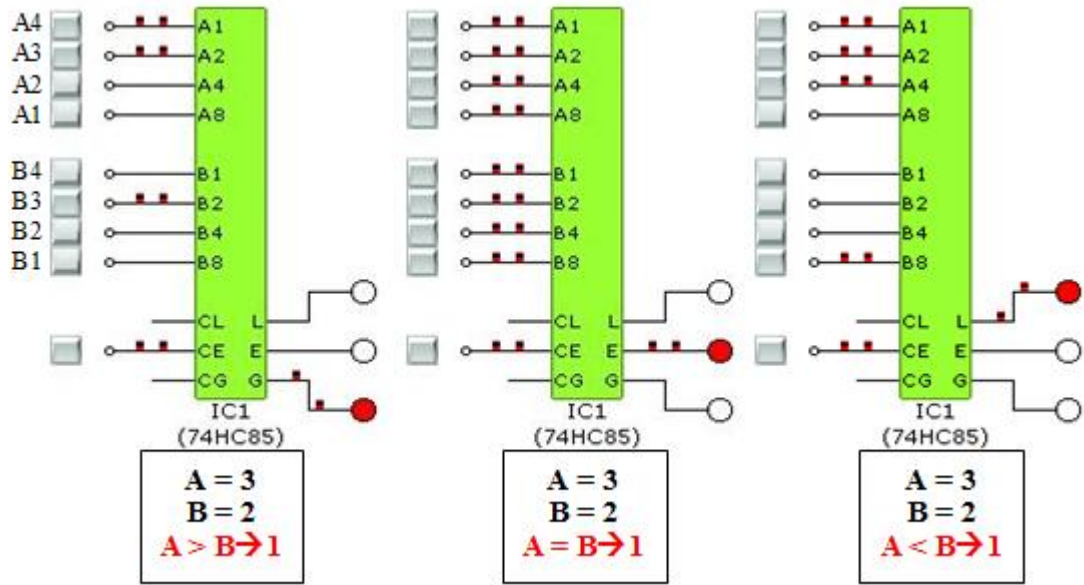
B3	1	16	VCC
IA<B	2	15	A3
IA=B	3	14	B2
IA>B	4	13	A2
OA<B	5	12	A1
OA=B	6	11	B1
OA>B	7	10	A0
GND	8	9	B0

Şekil 5.6: Dört bitlik iki sayıyı karşılaştıran entegrenin (7485) bacak bağlantıları

Şekil 5.6'da entegre bacak bağlantılarına bakıldığında karşılaştırılma yapılan iki kısım dikkati çekmektedir. Sadece bir tane entegreyle karşılaştırma yapacaksanız OA<B, OA=B, OA>B'yi çıkış olarak kullanmalısınız. Karşılaştırıcımızdan önce bir karşılaştırıcı entegre daha bağlıysa ve birbirleriyle bağlantılı bilgilere göre çalışacaklarsa IA<B, IA=B, IA>B'ye bir önceki karşılaştırıcının çıkışları bağlanır.

Simülasyon programda entegrenin çalışması;

- A0, A1, A2 ve A3'e birinci sayıyı temsil edecek girişler bağlanır,
- B0, B1, B2 ve B3'e ikinci sayıyı temsil edecek girişler bağlanır,
- Daha önce bir karşılaştırma entegresinden sonra gelmediğinden ve entegrenin doğru çalışması için IA<B ve IA>B (input IA>B) girişleri toprağa (lojik0), IA=B girişleri ise VCC'ye (lojik1) bağlanır.
- OA<B, OA=B ve OA>B (output A>B)'ye çıkış elemanları (led) bağlanır.
- CE= lojik 1 olması çok önemlidir.



Şekil 5.7: Dört bitlik iki sayı karşılaştıran devrenin 7485 entegresiyle yapılmış halinin simülasyon programında (crocodiletechnology) uygulanması ve çalıştırılması



## UYGULAMA FAALİYETİ

İki bitlik iki sayıyı karşılaştıran devreyi tasarlayınız.

İşlem Basamakları	Öneriler
➤ Devre çıkışı lojik1’de mi yoksa Lojik0’da mı aktif olacak belirleyiniz.	➤ Çıkışın Lojik1’de aktif olması sizin soruyu daha kolay çözmenizi sağlayacaktır.
➤ Devrenin girişinin ve çıkışının kaç bitten oluşacağını belirleyiniz.	➤ Hata olmaması için çıkışın bit sayısının taşma olmayacak şekilde düşülmesi gereklidir.
➤ Devrenin doğruluk tablosunu hazırlayınız.	➤ Giriş değişken sayılarına bakarak karno haritasının kaç kareden oluşacağını 2N formülüne göre hesaplayabilirsiniz.
➤ Doğruluk tablosuna bakarak karno haritasının içini doldurunuz.	➤ Karno haritasının kalıbını hazırlarken haritadaki karelerin numaralarının sıralamalarının Gray koda göre verildiğini unutmayınız. ➤ Her bir çıkış için ayrı karno haritası hazırlayınız.
➤ Karno haritasına göre indirgeme yapıp her bir çıkışın fonksiyonunu bulunuz.	
➤ Çıkış fonksiyonuna bakarak devre için gerekli elemanları belirleyiniz.	
➤ Devrenin kağıt üzerinde çizimini gerçekleştiriniz.	
➤ Çıkış fonksiyonunuzu verecek devreyi bilgisayardaki simülâtör programlarda kurup, devreyi çalıştırınız.	➤ Dijital elektronik devre elemanları 5V ile çalışır. Kullanılan entegrelerin bacak bağlantılarının bulunduğu bir belgeyi yanınızda bulundurun. ➤ Simülâtör yazılımlarından faydalanabilirsiniz.
➤ Bilgisayardaki devreniz doğru sonuçlar veriyorsa devrede kullanılan lojik elemanları son olarak tekrar belirleyip, devrenizi plakete (breadboard) ve ya lojik eğitim setinize kurunuz.	➤ Plaket (breadboar) kullanmayı hatırlamıyorsanız devreyi yanlış kurmamak için uygulama öğretmeninizden ya da laboratuvar eşinizden nasıl kullanıldığını hatırlatmasını isteyiniz. ➤ Kullanacağınız entegrelerin 74*** serisine mi ait yoksa 40** serisine mi ait olduğuna dikkat edin. ➤ Devrenizde kullanacağınız bütün kapıların bulabilerseniz aynı seriye ait olmasına dikkat ediniz.
➤ Devreniz doğru çalıştıysa devrede kullanacağınız elemanları belirleyiniz.	
➤ Devrenizi elektronik ortamda kurunuz.	

## ÖLÇME VE DEĞERLENDİRME

Aşağıdaki soruları dikkatlice okuyunuz ve doğru seçeneği işaretleyiniz.

1. Yarım bitlik karşılaştırıcıda birbirinden bağımsız kaç bit karşılaştırılabilir?  
A) 0                      B) 1                      C) 2                      D) 3
2. En basit karşılaştırma işlemini yapan kapı aşağıdakilerden hangisidir?  
A) AND                      B) OR                      C) NOT                      D) EX-OR
3. Karşılaştırma işleminin sonucu aşağıdakilerden hangisi olamaz?  
A)  $A > B$                       B)  $A < B$                       C)  $A = B$                       D)  $A \times B$
4. Tam toplayıcı elde etmek için kaç adet yarım toplayıcı kullanılması gerekir?  
A) 2                      B) 4                      C) 8                      D) 16
5. Devrenin çalışması için aşağıdakilerden hangisi önemli değildir?  
A) Besleme geriliminin verilmiş olması  
B) CE'nin aktif olması  
C) Çıkışta led'lerin olması  
D) Giriş bitlerinin uygun olması

## DEĞERLENDİRME

Cevaplarınızı cevap anahtarıyla karşılaştırınız. Yanlış cevap verdiğiniz ya da cevap verirken tereddüt ettiğiniz sorularla ilgili konuları faaliyete geri dönerek tekrarlayınız. Cevaplarınızın tümü doğru ise "Modül Değerlendirme"ye geçiniz.

# MODÜL DEĞERLENDİRME

Aşağıdaki cümlelerin başında boş bırakılan parantezlere, cümlelerde verilen bilgiler doğru ise D, yanlış ise Y yazınız.

1. ( ) Çarpma devrelerini tam toplayıcılar kullanılarak yapılabiliriz.
2. ( ) Karşılaştırma devrelerini sistem kontrollerinde kullanabiliriz.
3. ( ) Eksik giriş- çıkış sayısına göre tasarlanmış devreyi en baştan tekrar tasarlamamız gerekir.
4. ( ) Devrelerin sadeleştirilmesinde kullanılan yöntemlerden Boolean yöntemi, karno haritaları yöntemine göre daha güvenilir ve kolay bir yöntemdir.
5. ( ) Karno haritasını oluşturan karelerin sayısı N giriş değişkeni olmak üzere  $2^N$  formülüyle bulunur.
6. ( ) 74\*\*\* serisinden 7485 entegresi karşılaştırma işlemi yapmak için kullanılır.
7. ( ) 74\*\*\* serisinden 7483 entegresi çıkarma işlemi yapmak için kullanılır.
8. ( ) Mikro işlemcilerin içinde toplama, çarpma, çıkarma, karşılaştırma işlemlerinin yapıldığı bir bölüm vardır.
9. ( ) Doğruluk tablosunda tanımsız ifadeler çıkarsa devre yanlış sayılır.
10. ( ) Karno haritalarında 6'lı, 12'li, 16'lı gruplamalar yapılabilir.
11. ( ) Doğruluk tablosundaki sıra numaraları, karno haritasına gray koda göre yerleştirilir.
12. ( ) Karno haritalarında sıfırlara ve birlere göre gruplama yapılabilir.
13. ( ) Kontrol değişkeni olmadan bire göre tülemeyi not kapısıyla da yapabiliriz.
14. ( ) Her bit için bir EX-OR kapısıyla ikiye göre tüleme alabiliriz.
15. ( ) Aritmetik işlemler yapan entegreler sadece 74\*\*\* serisinde vardır.
16. ( ) Karno haritalarında toplam (Maxterm) ve çarpım (minterm) terimler gösterilir.
17. ( ) Karno haritalarında bire göre gruplama yapılırken sıfırlar kullanılmaz.
18. ( ) Paralel toplamada her bir iki biti toplamak ve oluşan eldeyi bir sonraki iki bite aktarmayı sağlamak için tam toplayıcılar kullanılır.

19. ( ) Tam çıkarıcı daha düşük değerlikli kademedan bir borç alınmış olabileceği dikkate alınarak iki biti birbirinden çıkararak tümleşik (kombinasyonel) bir devredir.
20. ( ) Tam çıkarıcı üç giriş ile iki çıkıştan oluşur.
21. ( ) Yarım çıkarıcı üç giriş ile iki çıkıştan oluşur.
22. ( ) Negatif sayılar, mikroişlemcili sistemlerde tümlemeleri alınarak saklanır.
23. ( ) Pozitif sayılar, mikroişlemcili sistemlerde tümlemeleri alınarak saklanır.
24. ( ) Lojik devrelerde çıkarma işlemi negatif bir sayının pozitif bir sayıyla toplanmasıdır.
25. ( ) İşaret biti en düşük değerlikli bitin (LSD) solunda bulunur.
26. ( ) En basit çarpma işlemi OR kapısıyla yapılır.
27. ( ) İki bitlik iki sayıyı çarpan devrenin iki çıkışı bulunur.
28. ( ) İki sayının birbirlerine göre durumlarını çarpma devreleri bulur.
29. ( ) Karşılaştırma devreleri toplama ve çıkarma devrelerinde bulunabilir.
30. ( ) Tam karşılaştırma devrelerinde üç çıkış bulunur.

**Aşağıdaki soruları yönergeler doğrultusunda çözünüz.**

31. Birer bitlik a,b,c sayılarını çarpan devreyi çiziniz.
32.  $T = AB'C' + ABC + A'B'C + A'BC'$  sadeleştirilmiş ifadenin devresini sadece üç giriş ve iki kapı kullanarak gösteriniz.
33. Yandaki Karno haritasının en sade halini bulunuz.

		yz	00	01	11	10
F	x					
	0		1 <sub>0</sub>		1 <sub>3</sub>	1 <sub>2</sub>
	1		1 <sub>4</sub>	1 <sub>5</sub>		Φ <sub>6</sub>

34. Yandaki karno haritasının en sade halini bulunuz.

		yz	00	01	11	10
F	x					
	0		0 <sub>0</sub>		1 <sub>3</sub>	0 <sub>2</sub>
	1		0 <sub>4</sub>	0 <sub>5</sub>		Φ <sub>6</sub>

35. Çıkış =  $A1B1' B0 + A0B1B0'$  devresini çiziniz.
36. 15 katlı bir binada 2 asansör bulunmaktadır. Birinci asansör tek numaralı katlara, ikinci asansör çift numaralı katlara ve 15. kata uğramaktadır. Devrenin doğruluk tablosunu çiziniz.
37. 37 numaralı sorunun karno haritasını çizip çıkışların indirgemelerini yapınız.
38. 38 numaralı soruda bulunan çıkışın devresini çiziniz.

## DEĞERLENDİRME

Cevaplarınızı cevap anahtarıyla karşılaştırınız. Yanlış cevap verdiğiniz ya da cevap verirken tereddüt ettiğiniz sorularla ilgili konuları faaliyete geri dönerek tekrarlayınız. Cevaplarınızın tümü doğru ise bir sonraki modüle geçmek için öğretmeninize başvurunuz.

# CEVAP ANAHTARLARI

## ÖĞRENME FAALİYETİ – 1'İN CEVAP ANAHTARI

1	C
2	C
3	B
4	D
5	A
6	D
7	Doğru
8	Doğru
9	Yanlış
10	Yanlış

## ÖĞRENME FAALİYETİ –2'NİN CEVAP ANAHTARI

1	B
2	A
3	B
4	A
5	Doğru
6	Doğru

## ÖĞRENME FAALİYETİ – 3'ÜN CEVAP ANAHTARI

1	B
2	A
3	B
4	A
5	D
6	D
7	C
8	C
9	A
10	B

### ÖĞRENME FAALİYETİ – 4'ÜN CEVAP ANAHTARI

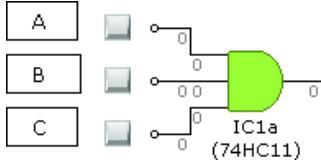
1	B
2	C
3	B
4	A
5	D

### ÖĞRENME FAALİYETİ – 5'İN CEVAP ANAHTARI

1	B
2	D
3	D
4	A
5	C

### MODÜL DEĞERLENDİRMENİN CEVAP ANAHTARI

1	Doğru	16	Doğru
2	Doğru	17	Doğru
3	Doğru	18	Doğru
4	Yanlış	19	Doğru
5	Doğru	20	Doğru
6	Doğru	21	Yanlış
7	Doğru	22	Doğru
8	Doğru	23	Yanlış
9	Yanlış	24	Doğru
10	Yanlış	25	Yanlış
11	Doğru	26	Yanlış
12	Doğru	27	Yanlış
13	Doğru	28	Yanlış
14	Yanlış	29	Doğru
15	Yanlış	30	Doğru

31	
32	$A \oplus B \oplus C$
33	$Z' + XY'$

34	$Z * (X' + Y)$																																																																																																																							
35	<div style="border: 1px solid black; padding: 5px; margin-bottom: 5px; display: inline-block;">             A1 A0 B1 B0           </div> <div style="display: flex; justify-content: space-around; margin-top: 5px;"> <div style="width: 20px; height: 20px; background-color: #ccc; border: 1px solid #ccc;"></div> <div style="width: 20px; height: 20px; background-color: #ccc; border: 1px solid #ccc;"></div> <div style="width: 20px; height: 20px; background-color: #ccc; border: 1px solid #ccc;"></div> <div style="width: 20px; height: 20px; background-color: #ccc; border: 1px solid #ccc;"></div> </div>																																																																																																																							
36	<table border="1" style="border-collapse: collapse; text-align: center;"> <thead> <tr style="background-color: #e0f7fa;"> <th style="color: #ffc107;">hücre no</th> <th>A</th> <th>B</th> <th>C</th> <th>D</th> <th style="color: #dc3545;">TEK</th> <th style="color: #17a2b8;">ÇİFT</th> </tr> </thead> <tbody> <tr><td style="background-color: #e0e0e0;">0</td><td>0</td><td>0</td><td>0</td><td>0</td><td style="color: #dc3545;">0</td><td style="color: #17a2b8;">1</td></tr> <tr><td style="background-color: #e0e0e0;">1</td><td>0</td><td>0</td><td>0</td><td>1</td><td style="color: #17a2b8;">1</td><td style="color: #dc3545;">0</td></tr> <tr><td style="background-color: #e0e0e0;">2</td><td>0</td><td>0</td><td>1</td><td>0</td><td style="color: #dc3545;">0</td><td style="color: #17a2b8;">1</td></tr> <tr><td style="background-color: #e0e0e0;">3</td><td>0</td><td>0</td><td>1</td><td>1</td><td style="color: #17a2b8;">1</td><td style="color: #dc3545;">0</td></tr> <tr><td style="background-color: #e0e0e0;">4</td><td>0</td><td>1</td><td>0</td><td>0</td><td style="color: #dc3545;">0</td><td style="color: #17a2b8;">1</td></tr> <tr><td style="background-color: #e0e0e0;">5</td><td>0</td><td>1</td><td>0</td><td>1</td><td style="color: #17a2b8;">1</td><td style="color: #dc3545;">0</td></tr> <tr><td style="background-color: #e0e0e0;">6</td><td>0</td><td>1</td><td>1</td><td>0</td><td style="color: #dc3545;">0</td><td style="color: #17a2b8;">1</td></tr> <tr><td style="background-color: #e0e0e0;">7</td><td>0</td><td>1</td><td>1</td><td>1</td><td style="color: #17a2b8;">1</td><td style="color: #dc3545;">0</td></tr> <tr><td style="background-color: #e0e0e0;">8</td><td>1</td><td>0</td><td>0</td><td>0</td><td style="color: #dc3545;">0</td><td style="color: #17a2b8;">1</td></tr> <tr><td style="background-color: #e0e0e0;">9</td><td>1</td><td>0</td><td>0</td><td>1</td><td style="color: #17a2b8;">1</td><td style="color: #dc3545;">0</td></tr> <tr><td style="background-color: #e0e0e0;">10</td><td>1</td><td>0</td><td>1</td><td>0</td><td style="color: #dc3545;">0</td><td style="color: #17a2b8;">1</td></tr> <tr><td style="background-color: #e0e0e0;">11</td><td>1</td><td>0</td><td>1</td><td>1</td><td style="color: #17a2b8;">1</td><td style="color: #dc3545;">0</td></tr> <tr><td style="background-color: #e0e0e0;">12</td><td>1</td><td>1</td><td>0</td><td>0</td><td style="color: #dc3545;">0</td><td style="color: #17a2b8;">1</td></tr> <tr><td style="background-color: #e0e0e0;">13</td><td>1</td><td>1</td><td>0</td><td>1</td><td style="color: #17a2b8;">1</td><td style="color: #dc3545;">0</td></tr> <tr><td style="background-color: #e0e0e0;">14</td><td>1</td><td>1</td><td>1</td><td>0</td><td style="color: #dc3545;">0</td><td style="color: #17a2b8;">1</td></tr> <tr><td style="background-color: #e0e0e0;">15</td><td>1</td><td>1</td><td>1</td><td>1</td><td style="color: #17a2b8;">1</td><td style="color: #17a2b8;">1</td></tr> </tbody> </table>	hücre no	A	B	C	D	TEK	ÇİFT	0	0	0	0	0	0	1	1	0	0	0	1	1	0	2	0	0	1	0	0	1	3	0	0	1	1	1	0	4	0	1	0	0	0	1	5	0	1	0	1	1	0	6	0	1	1	0	0	1	7	0	1	1	1	1	0	8	1	0	0	0	0	1	9	1	0	0	1	1	0	10	1	0	1	0	0	1	11	1	0	1	1	1	0	12	1	1	0	0	0	1	13	1	1	0	1	1	0	14	1	1	1	0	0	1	15	1	1	1	1	1	1
hücre no	A	B	C	D	TEK	ÇİFT																																																																																																																		
0	0	0	0	0	0	1																																																																																																																		
1	0	0	0	1	1	0																																																																																																																		
2	0	0	1	0	0	1																																																																																																																		
3	0	0	1	1	1	0																																																																																																																		
4	0	1	0	0	0	1																																																																																																																		
5	0	1	0	1	1	0																																																																																																																		
6	0	1	1	0	0	1																																																																																																																		
7	0	1	1	1	1	0																																																																																																																		
8	1	0	0	0	0	1																																																																																																																		
9	1	0	0	1	1	0																																																																																																																		
10	1	0	1	0	0	1																																																																																																																		
11	1	0	1	1	1	0																																																																																																																		
12	1	1	0	0	0	1																																																																																																																		
13	1	1	0	1	1	0																																																																																																																		
14	1	1	1	0	0	1																																																																																																																		
15	1	1	1	1	1	1																																																																																																																		



37

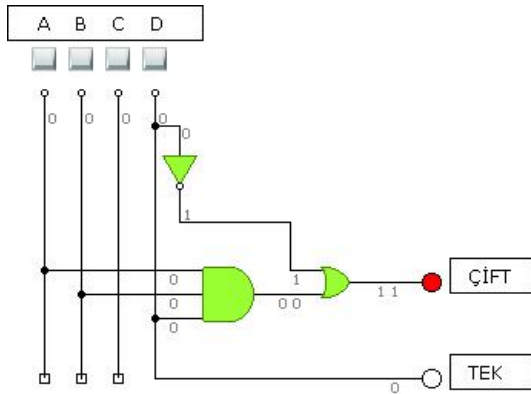
TEK \ CD	00	01	11	10
00	0	1	1	2
01	4	1	1	6
11	12	1	1	14
10	8	1	1	10

TEK = D

ÇİFT \ CD	00	01	11	10
00	1	0	3	1
01	1	4	5	6
11	1	12	13	14
10	1	8	9	10

ÇİFT = D' ABC

38



## KAYNAKÇA

- MANO, Morris, "**Sayısal Tasarım**", Milli Eğitim Basımevi, 2000.
- YAĞIMLI, Mustafa, Fevzi AKAR , "**Dijital Elektronik**", Beta Basım AŞ, Mart 1999.
- <http://www.eng.auburn.edu/~kirkih/ELEC2010/LabManual>
- [http://www.etu.edu.tr/~oergin/bil361/birinci\\_sinav.pdf](http://www.etu.edu.tr/~oergin/bil361/birinci_sinav.pdf)
- <http://www.etu.edu.tr/~oergin/bil361/boz/aritmetik4.pdf>
- <http://www.iticu.edu.tr/ogr/salsan/LojikDevreler.htm>
- <http://www.spsu.edu/cs/faculty/bbrown/papers/karnaugh.pdf>
- [www.uzak.mersin.edu.tr/UZAK/TP/EndElo/Eln-123/sy-06.pdf](http://www.uzak.mersin.edu.tr/UZAK/TP/EndElo/Eln-123/sy-06.pdf)
- <http://www2.itu.edu.tr/~acarser/SSTU2006-2007/proje4.html>
- [http://www2.itu.edu.tr/~orssi/thesis/2006/Bitirme\\_son.doc](http://www2.itu.edu.tr/~orssi/thesis/2006/Bitirme_son.doc)