



# LOJİK TASARIM LABORATUVARI

ITU  
Elektronik ve  
Haberleşme  
Mühendisliği  
Bölümü

2021

## 1. Genel Açıklamalar

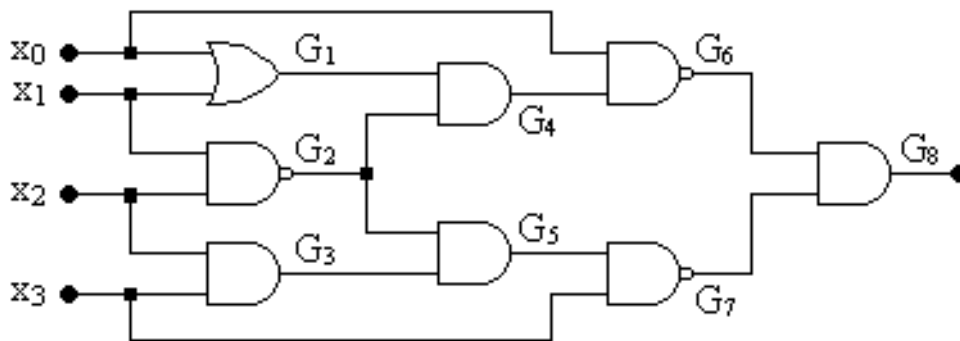
Giriş değişkenleri  $x = x_1, x_2, \dots, x_n$  ve çıkış değişkeni  $z$  olan bir Boolean fonksiyonu; eğer  $z$  çıkışının değeri,  $x$  değerlerinin sadece o anki değerlerine bağlı ve  $z$ 'nin önceki değerlerinden tamamen bağımsız ise bir kombinezonsal fonksiyondur. Bir kombinezonsal fonksiyonun devre elemanları kullanılarak gerçekleştirilmesi sonucu elde edilen devreye, kombinezonsal devre denilir.

Kombinezonsal devrelerin grafları, yönlü ve çevre içermeyen graflardır (DAG) ve geribesleme bağlantıları içermez. Bu yüzden kombinezonsal devrelere çevrimsiz devreler de denilir. Kombinezonsal devrelerin analizi, devre tarafından gerçekleştirilen fonksiyonun belirlenmesini içerir. Sayısal devre içinde kullanılan farklı elamanların Boolean ifadeleri ile devre içindeki her bir bağlantının Boolean ifadeleri belirlenebilir. Sayısal devreler, sayısal elemanların birleşmesinden meydana geldiği için devrenin çıkış fonksiyonu, Boolean cebri kullanılarak elde edilen ifadelerin yardımı ile Boolean ifadeleri cinsinden belirlenebilir. Bunun yanında iki değerli Boolean cebri,  $B_2=\{0,1\}$ , sayısal devrelerin tanımlanmasında kullanılır. Devre içindeki her bir bağlantı, 0 ve 1 değerlerinden birini alabilecek değişken olarak gösterilir ve devrenin fonksiyonu, devre girişlerinin bütün olası kombinasyonlarına karşılık devre çıkışının aldığı değerlerin oluşturduğu doğruluk tablosu ile belirlenebilir. Böylece devre fonksiyonunun belirlenmesi, Boolean ifadeleri veya doğruluk tablosu formunda olabilir. Bir sayısal devre, devrenin kendi fonksiyonunu gerçekleştirip gerçekleştirmediğinin belirlendiği test aşamasında analiz edilir. Devrede bir hata olduğu belirlendikten sonra hata yerinin bulunması ve hata düzeltimi aşamalarına geçilir. Bunun yanında, verilen iki adet devrenin aynı fonksiyonu gerçekleştirip gerçekleştirmediğinin belirlenmesinde devre analizi yöntemleri kullanılır.

## 2. Deneyde Yapılacaklar

### 2.1. Adım 1

Şekil 1'de verilen devrenin her bir kapısına ilişkin Boolean ifadelerini, devrenin giriş değişkenleri cinsinden bulunuz. Bu şekildeki devreyi deney setine kurunuz. Kullanılan bütün tümdevrelerin besleme ve toprak bağlantılarını yapınız. Devrenin girişlerini lojik anahtarlardan alıp, devredeki her bir kapının çıkışını LED'lere (Light Emitting Diode) bağlayınız ve Tablo 1'de verilen doğruluk tablosunu doldurunuz.



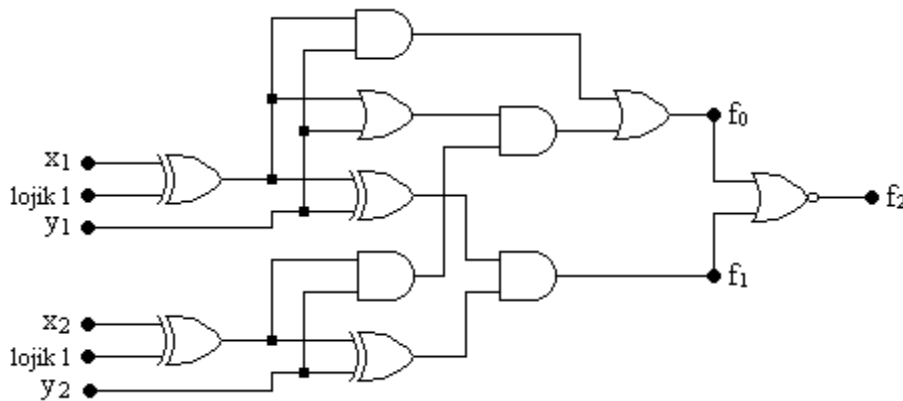
**Şekil 1:** Combinational circuit to be analyzed.

**Tablo 1:** Şekil 1’de verilen devrenin doğruluk tablosu.

x3	x2	x1	x0	G1	G2	G3	G4	G5	G6	G7	G8
0	0	0	0								
0	0	0	1								
0	0	1	0								
0	0	1	1								
0	1	0	0								
0	1	0	1								
0	1	1	0								
0	1	1	1								
1	0	0	0								
1	0	0	1								
1	0	1	0								
1	0	1	1								
1	1	0	0								
1	1	0	1								
1	1	1	0								
1	1	1	1								

**2.2. Adım 2**

Şekil 2’deki devreyi deney setine kurunuz. Kullanılan bütün tümdevrelerin besleme ve toprak bağlantılarını yapınız. Devrenin girişlerini lojik anahtarlardan alıp, devre çıkışlarını LED’lere bağlayınız ve Tablo 2’de verilen doğruluk tablosunu doldurunuz. Elde edilen doğruluk tablosunun yardımıyla devrenizin fonksiyonunu belirleyiniz

**Şekil 2:** Fonksiyonu belirlenecek kombinezonsal devre.

**Tablo 2:** Şekil 2’de verilen devrenin doğruluk tablosu.

x1	x2	y1	y2	f0	f1	f2
0	0	0	0			
0	0	0	1			
0	0	1	0			
0	0	1	1			
0	1	0	0			
0	1	0	1			
0	1	1	0			
0	1	1	1			
1	0	0	0			
1	0	0	1			
1	0	1	0			
1	0	1	1			
1	1	0	0			
1	1	0	1			
1	1	1	0			
1	1	1	1			

**3. Malzeme Listesi**

Malzeme	Adet	Kütüphane
7400 – NAND IC	1	logi7400dip
7402 – NOR IC	1	logi7400dip
7408 – AND IC	1	logi7400dip
7432 – OR IC	1	logi7400dip
7486 – EXOR IC	1	logi7400dip

## 1. Genel Açıklamalar

Bir kombinezonsal devrenin tasarımında ilk olarak sözle tanım ile ifade edilen devre fonksiyonu için devrenin  $x_1, x_2, \dots, x_n$  girişleri ile  $z$  çıkışına karşılık düşen doğruluk tablosu oluşturulur. Doğruluk tablosunda  $x_1, x_2, \dots, x_n$  değişkenlerinin bütün  $2^n$  adet giriş kombinasyonlarının oluşturduğu küme,  $n$ -küp veya  $n$ -boyutlu uzay olarak anılır.  $n$ -boyutlu uzay içinde birer nokta olan  $2^n$  adet giriş kombinasyonunun her biri için  $z$  çıkışının değeri, 1 (1-noktası), 0 (0-noktası) veya belirlenmemiş (keyfi noktası) olur. Bir giriş kombinasyonu için çıkışın belirlenmemiş olması, bu giriş kombinasyonunun asla uygulanmayacağını ve 0 ya da 1 olabileceğini gösterir. Eğer bir devrenin çıkışı, 1-noktasına karşılık gelen bütün girişler için 1 ve 0-noktasına karşılık gelen bütün girişler için 0 ise bu devre, çıkışına ait olan fonksiyonu gerçekler denilir. Kombinezonsal devre sentezinde amaç, verilen bir devre fonksiyonunun gerçekleştirilmesidir. Kombinezonsal lojik devre sentez yöntemleri genel olarak iki grupta toplanabilir. Birinci yöntem, elde edilen doğruluk tablosu yardımıyla Quine-McCluskey veya Karnaugh yöntemlerinin uygulanmasıyla minimal fonksiyonu bulmaktır. Minimal fonksiyonun bulunmasında kullanılacak yöntemin hangisi olacağına fonksiyon içinde bulunan bağımsız değişken sayısının belirlenmesi sonucu karar verilir. Değişken sayısı, 4-5'e kadar olan fonksiyonlarda Karnaugh yönteminin uygulanması daha çabuk sonuca ulaştıracağı için tercih edilir. Minimal fonksiyona karşı düşen devre, iki seviyeli (çarpımlar toplamı ya da toplamalar çarpımı) gerçekleştirilebileceği gibi belirli bir gecikme süresi göz önüne alınarak iki seviyeli devreden daha az devre karmaşıklığına sahip olacak şekilde çok seviyeli olarak da gerçekleştirilebilir. Çıkış sayısı birden fazla olan devrelerde aynı zamanda devre çıkışlarına ait olan fonksiyonlar iki seviyeli olarak birlikte indirgenerek devrenin, PLA devre karmaşıklığının azaltılması amaçlanır. Kombinezonsal devrelerde karmaşıklık, kapı sayısı artı giriş sayısı olarak tanımlanır. Ancak, tüm kapıların giriş sayıları aynı ise devre karmaşıklığı sadece kapı sayısı ile de belirlenebilir. Bir kombinezonsal devrenin seviyesi ise, devrenin her bir girişinden her bir çıkışına uzanan yollarda bulunan maksimum kapı elemanı sayısıdır. Kombinezonsal lojik devre sentezinde ikinci yöntem ise sözle tanımdan bir algoritma çıkararak, bu algoritmaya karşı düşen devreyi gerçeklemektir. Bu yöntem, genellikle değişken sayısı ve/veya keyfi çıkışları fazla olan fonksiyonların gerçekleştirilmesinde oldukça elverişlidir. Örnek olarak, karşılaştırmacı ve kodlayıcı devreleri bu yöntemle gerçekleştirilebilir.

Bu iki kombinezonsal devre sentez yöntemi karşılaştırıldığında, ilk yöntem için değişken sayısı arttığında doğruluk tablosunun üstel biçimde büyüdüğü görülmektedir. İkinci yöntemde ise böyle bir sorunla karşılaşmaz ama sözle tanımdan her zaman bir algoritma çıkarabilmek de mümkün olmamaktadır.

Teorik olarak, kapı elemanları ile devre tasarımı gerçekleştirirken kullanılacak kapı ve kapı giriş sayısının minimalleştirilmesi esas alınır. Ancak, uygulamada tümdevreler kullanıldığı için minimallik kavramı, tümdevre sayısı ile ilişkili olmaktadır. Minimal tümdevre sayısını sağlamak için var olan farklı yöntemlerden biri, aynı tür kapı kullanımı için kullanılan ortak kapı dönüşümleridir.

**Sözle Tanım 1 :** BCD sayıları, (84-2-1) koduna dönüştüren kombinezonsal devrenin tasarlanması istenmektedir. BCD'den (84-2-1)'e kod dönüştürücü devresi, dört adet bağımsız giriş,  $x_3x_2x_1x_0$ , ( $x_3$  : En anlamlı bit - Most Significant Bit: MSB) ve dört adet çıkıştan, 84-2-1, (8 : MSB) oluşmaktadır. Bu devrenin giriş değişkenleri, on tabanındaki sayıların ikili kodlanmış halinde iken, çıkışları bu sayıların (84-2-1) kodundaki karşılığıdır. Bir giriş kombinasyonunun (84-2-1) kod karşılığı, çıkışlarından her biri 0 veya 1 değerine sahip olan ve bu değerler ile çıkışa ait kodun katsayıları (8 4 -2 -1) ile çarpılıp toplandığında elde edilen sonuç, bu giriş kombinasyonunun on tabanındaki karşılığı olan değerler kombinasyonudur. On tabanındaki sayılar (0-9) haricinde geriye kalan altı giriş kombinasyonu için çıkışlar, keyfi değerlerini alır. Tablo 1'de bu kod dönüştürücünün doğruluk tablosu verilmiştir.

**Tablo 1 :** BCD – (84-2-1) kod dönüştürücü doğruluk tablosu.

$x_3$	$x_2$	$x_1$	$x_0$	$f_8$	$f_4$	$f_{-2}$	$f_{-1}$
0	0	0	0	0	0	0	0
0	0	0	1	0	1	1	1
0	0	1	0	0	1	1	0
0	0	1	1	0	1	0	1
0	1	0	0	0	1	0	0
0	1	0	1	1	0	1	1
0	1	1	0	1	0	1	0
0	1	1	1	1	0	0	1
1	0	0	0	1	0	0	0
1	0	0	1	1	1	1	1
10- 15				Keyfi			

Elde edilen doğruluk tablosundaki her bir çıkış için elde edilen Boolean fonksiyonları;

$$f_8 : x_0x_2 + x_1x_2 + x_1'x_3$$

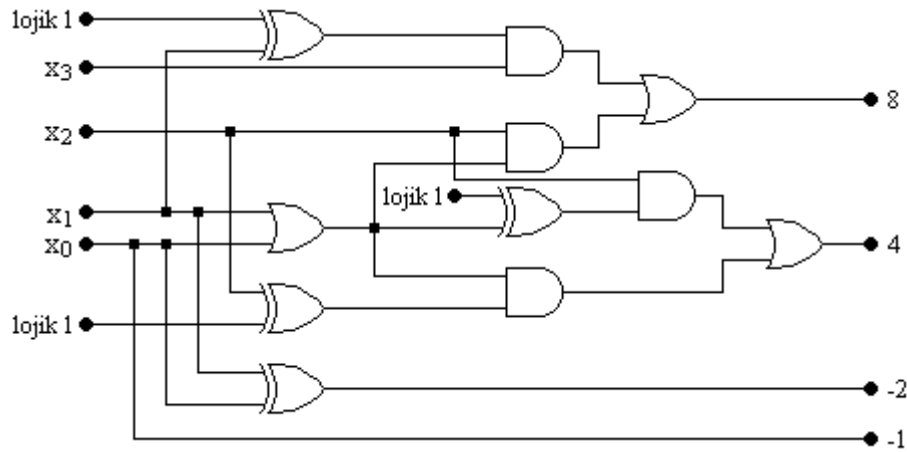
$$f_4 : x_0x_2' + x_1x_2' + x_0'x_1'x_2$$

$$f_{-2} : x_0x_1' + x_0'x_1$$

$$f_{-1} : x_0$$

(1)

şeklindedir. Bu çıkış fonksiyonlarının iki seviyeli çarpımlar toplamı şeklinde AND, OR ve NOT kapıları kullanılarak gerçekleştirilmesi istendiğinde 7 adet iki girişli AND kapısı, 1 adet üç girişli AND kapısı, 2 adet üç girişli OR kapısı, 1 adet iki girişli OR kapısı ve 3 adet NOT kapısı ile toplam 6 adet tümdevre gerekmektedir. Bu fonksiyonların ortak bileşenlerinin bulunmasına ve aynı tür kapıların kullanılmasına dikkat ederek 3 adet tümdevre ile 4 seviyeli olarak tasarlanan devre, Şekil 1'de verilmiştir.



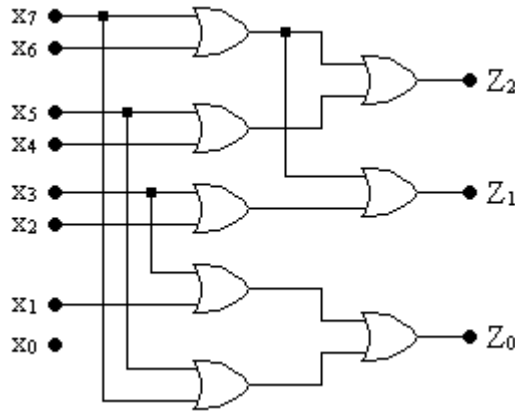
Şekil 1: BCD'den 84-2-1'e kod dönüştürücü devresi

**Sözle Tanım 2 :** Sekiz girişli,  $x_7x_6x_5x_4x_3x_2x_1x_0$ , ve üç çıkışlı,  $z_2z_1z_0$ , olan bir indis kodlayıcı devresi tasarlanmak istenmektedir. İndis, devre girişlerinde yer alan değişkenlerden birinin girişi lojik 1 değerine ve diğerlerinin lojik 0 değerine sahip olması ile belirlenir ve devrenin çıkış değerleri ( $z_2$  : MSB), bu girişin sahip olduğu indis değerinin ikili kodlanmış halidir. Öncelik kodlayıcısının doğruluk tablosu, Tablo 2'de verilmiştir.

Tablo 2: İndis kodlayıcı doğruluk tablosu

$x_7$	$x_6$	$x_5$	$x_4$	$x_3$	$x_2$	$x_1$	$x_0$	$z_2$	$z_1$	$z_0$
0	0	0	0	0	0	0	1	0	0	0
0	0	0	0	0	0	1	0	0	0	1
0	0	0	0	0	1	0	0	0	1	0
0	0	0	0	1	0	0	0	0	1	1
0	0	0	1	0	0	0	0	1	0	0
0	0	1	0	0	0	0	0	1	0	1
0	1	0	0	0	0	0	0	1	1	0
1	0	0	0	0	0	0	0	1	1	1
Diğer giriş kombinasyonları								Keyfi		

İndis kodlayıcı devresinin tasarımı için 8 değişkenli 3 adet indirgenmiş fonksiyonu Quine-McCluskey veya Karnaugh yöntemi ile bulmak yerine her bir çıkış fonksiyon değerinin 1 olması için hangi giriş değişkenlerinin 1 olması gerektiği belirlenerek gerçekleştirilen devre, Şekil 2'de verilmiştir.



**Şekil 2:** İndis kodlayıcı devresi.

Bir Boolean fonksiyonu SSI kapı elemanları (Small Scale Integrated circuits) ile gerçekleştirilebildiği gibi aynı zamanda MSI (Medium Scale Integrated circuits), LSI (Large Scale Integrated circuits) ve VLSI (Very Large Scale Integrated circuits) ailesinden PLD (Programmable Logic Devices) ve PLD'lerin bir uzantısı olan FPGA (Field Programmable Gate Arrays) ve PLC (Programmable Logic Circuits) gibi elemanlar ile de gerçekleştirilebilir. SSI, MSI, LSI ve VLSI tümdevreler sırasıyla 1-10, 10-100, 100-1000 ve 1000-... arasında kapı elemanı içeren tümdevrelerdir.

## 2. Deney Sırasında Yapılacaklar

### 2.1. Adım 1

Şekil 1'de verilen devreyi deney setine kurunuz. Kullanılan bütün tümdevrelerin besleme ve toprak bağlantılarını yapınız. Devrenin girişlerini lojik anahtarlardan alıp devrenin çıkışlarını LED'lere bağlayınız. Doğruluk tablosunda verilen giriş kombinasyonlarını uygulayarak devrenizin istenilen fonksiyonu gerçekleştirip gerçekleştirmediğini saptayıp keyfi giriş kombinasyonlarına karşılık gelen çıkışların değerlerini belirleyiniz.

### 2.2. Adım 2

Şekil 2'de verilen devreyi deney setine kurunuz. Kullanılan bütün tümdevrelerin besleme ve toprak bağlantılarını yapınız. Devrenin girişlerini lojik anahtarlardan alıp devrenin çıkışlarını LED'lere bağlayınız. Doğruluk tablosunda yer alan giriş kombinasyonlarını uygulayarak devrenizin istenilen fonksiyonu gerçekleştirip gerçekleştirmediğini saptayınız.

## 3. Malzeme Listesi

Malzeme	Adet	Kütüphane
7408 – AND tümdevresi	1	logi7400dip
7432 – OR tümdevresi	2	logi7400dip
7486 – EXOR tümdevresi	1	logi7400dip



### 1. Genel Açıklamalar

Kod çözücüler (decoder), genellikle  $n$  girişli  $2^n$  çıkışlı MSI tümdevrelerdir. Yine de, ikili kodlanmış on tabanındaki sayılar (Binary Coded Decimal (BCD) : 0-9) için 4 girişli 10 çıkışlı,  $4 \times 10$ , kod çözücüler de mevcuttur. Kod çözücü girişlerinin her bir değer kombinasyonu için bu kombinasyonun on tabanındaki karşılığı olan çıkış, aktif olurken diğer çıkışlar aktif değildir. Buna göre, aktif-0 ve aktif-1 çıkışlı olarak iki tür kod çözücü vardır. Aktif-0 çıkışlı bir kod çözücü tümdevresinde, uygulanabilecek her bir giriş kombinasyonuna karşılık ilişkili çıkış lojik 0 değerine sahip iken, diğer çıkışlar lojik 1 değerine sahip olur. (Benzer şekilde aktif-1 çıkışlı kod çözücü tümdevresinde ilişkili çıkış lojik 1 değerini alırken, diğer çıkışlar lojik 0 değerini alır.) Böylece, her bir çıkış, kod çözücünün giriş kombinasyonuna ilişkin maksterimini (veya minterimini) oluşturur. Bundan dolayı, kod çözücü tümdevreleri ile herhangi bir Boolean fonksiyonu gerçekleştirilebilir. Belirli giriş ve çıkış sayısına sahip olan kod çözücü tümdevreleri ile giriş ve çıkış sayıları artırılmış kod çözücü yapıları elde edilebilir.

Çoğullayıcılar (multiplexer), seçilen bir girişteki veriyi, veri hattına aktarırlar. Çoğullama işlemi, çok sayıda bilginin daha az sayıda kanal veya hat üzerinden iletimidir. Böylece, birden fazla veri, istenen sırada tek bir veri hattından iletilebilir. Bundan dolayı, çoğullayıcılar veri toplayıcı olarak da adlandırılır. Veri hattının diğer tarafında veri dağıtıcı (demultiplexer) kullanılarak birden fazla veri tek bir hat üzerinden zamanda çoğullama yapılarak iletilebilir. Çoğullayıcılar,  $n$  adet kontrol girişi ve  $2^n$  adet veri girişi olmak üzere toplam  $n+2^n$  adet girişe sahiptir. Çoğullayıcılarda veri aktarımı,  $n$  adet kontrol girişinin yardımıyla  $2^n$  adet girişteki verinin çıkışa aktarılması ile sağlanır. Çıkışa aktarılacak olan verinin bulunduğu giriş, indisi kontrol giriş kombinasyonunun on tabanındaki karşılığı olan giriştir. Böylece,  $2^n \times 1$ 'lik bir çoğullayıcı kullanılarak  $n$  değişkenli bir Boolean fonksiyonu, fonksiyonun aldığı değerler veri girişlerine, değişkenleri ise kontrol girişlerine bağlanarak her bir giriş kombinasyonu için ilgili kombinasyona ait lojik değer çıkışa aktarılması ile gerçekleştirilebilir. Belirli giriş sayısına sahip olan çoğullayıcı tümdevreleri ile giriş sayıları artırılmış çoğullayıcı yapıları elde edilebilir.

Kodlayıcılar (encoder),  $2^n$  adet girişe,  $n$  adet çıkışa sahiptir. Bu açıdan kod çözücünün yapısına göre ters bir yapıya sahiptir. Girişlerinden yalnızca bir tanesi aktif (Aktif-0 girişli kodlayıcı için sadece bir tane giriş lojik 0 ve diğer girişler lojik 1 değerini alır. Aktif-1 girişli kodlayıcı için sadece bir giriş lojik 1 değerini alırken, diğer girişler lojik 0 değerini alır.) olduğunda, kodlayıcının çıkışı, aktif girişin indisinin iki tabanındaki karşılığıdır. Örneğin aktif-1  $8 \times 3$  kodlayıcısına  $x_7x_6x_5x_4x_3x_2x_1x_0$  : 00010000 girişi uygulandığında, çıkış değeri olarak  $(3)_{10} = (011)_2$  ikili kodu elde edilir. Birden fazla aktif giriş varsa, çıkış tanımsızdır. Öncelik kodlayıcılar, girişlerinden birden fazlasının aktif olmasına izin verirken çıkışlarında öncelikli olan aktif girişe ait olan değeri üretirler ve diğer aktif girişleri önemsemezler. Öncelik sıralaması, genellikle en büyük giriş indisinden en küçük giriş indisine doğrudur. Öncelikli kodlayıcılar, genellikle mikroislemcili sistemlerde kesme (interrupt) kontrolü için kullanılırlar.

## 2. Deney Sırasında Yapılacaklar

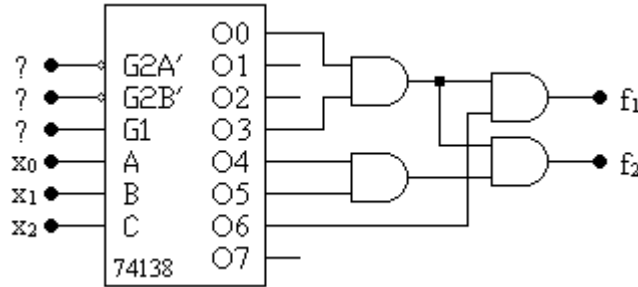
### 2.1. Adım 1

Tablo 1’de doğruluk tablosu verilen  $f_1$  ve  $f_2$  fonksiyonlarının kod çözücü ve AND kapıları ile tasarlanması istenmektedir. Buna göre tasarlanan devre, Şekil 1’de verilmiştir.

**Tablo 1 :** Kod çözücü ve çoğullayıcılar ile gerçekleştirilecek  $f_1$  ve  $f_2$  fonksiyonlarının doğruluk tablosu.

$x_2$	$x_1$	$x_0$	$f_1$	$f_2$
0	0	0	0	0
0	0	1	1	1
0	1	0	1	1
0	1	1	0	0
1	0	0	1	0
1	0	1	1	0
1	1	0	0	1
1	1	1	1	1

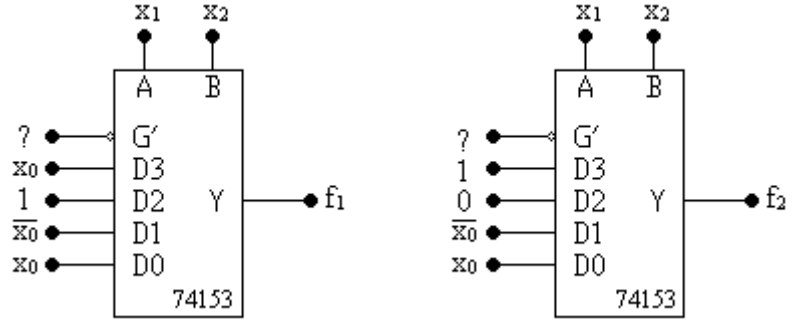
Şekil 1’deki devreyi deney setine kurunuz. Bütün tümleşik elemanlara besleme ve toprak bağlantısını yapınız. Kod çözücünün kontrol girişlerine uygun lojik değerler bağlayınız. Bunun için kod çözücünün katalog bilgisinden yararlanınız. Devre girişlerini lojik anahtarlardan alıp çıkışlarını LED’lerden gözleyerek doğruluk tablosunu sağlayıp sağlamadığını gösteriniz.



**Şekil 1:** Tablo 1’de verilen  $f_1$  ve  $f_2$  fonksiyonlarının 74138 kod çözücüsü ile tasarımı.

### 2.2. Adım 2

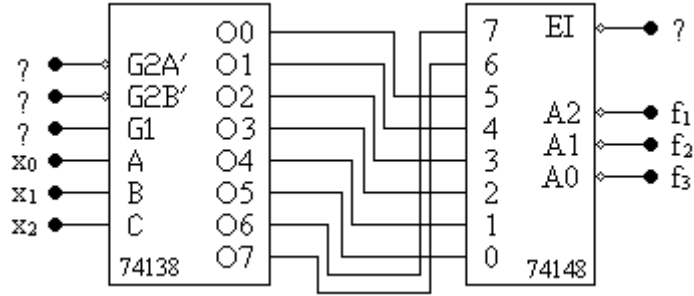
Tablo 1’de doğruluk tablosu verilen  $f_1$  ve  $f_2$  fonksiyonlarının çoğullayıcılar ile tasarlanması istenmektedir. Buna göre tasarlanan devre, Şekil 2’de verilmiştir. Şekil 2’de verilen devreyi deney setine kurunuz. Bütün tümleşik elemanlara besleme ve toprak bağlantısını yapınız. Çoğullayıcının kontrol girişlerine uygun lojik değerler bağlayınız. Bunun için çoğullayıcının katalog bilgisinden yararlanınız. Devrenin girişlerini lojik anahtarlardan alıp, çıkışlarını LED’lerden gözleyerek doğruluk tablosunu sağlayıp sağlamadığını belirleyiniz.



**Şekil 2:** Tablo 1’de verilen  $f_1$  ve  $f_2$  fonksiyonlarının 74153 çoğullayıcısı ile tasarımı.

### 2.3. Adım 3

Şekil 3’te aktif-0 çıkışlı 74153 kod çözücü ve aktif-0 girişli ve çıkışlı 74148 öncelikli kodlayıcı tümdevrelerini içeren devrenin analiz edilmesi istenmektedir. Bunun için Şekil 3’teki devreyi deney setine kurunuz. Bütün tümleşik elemanların besleme ve toprak bağlantılarını yapınız. Devre girişlerini lojik anahtarlardan alıp, çıkışlarını LED’lere bağlayınız. Tablo 2’de verilen doğruluk tablosunu doldurarak devrenin fonksiyonunu belirleyiniz.



**Şekil 3:** Fonksiyonu belirlenecek devre.

**Tablo 2:** Şekil 3’te verilen devrenin doğruluk tablosu.

$x_2$	$x_1$	$x_0$	$f_1$	$f_2$	$f_3$
0	0	0			
0	0	1			
0	1	0			
0	1	1			
1	0	0			
1	0	1			
1	1	0			
1	1	1			

**3. Malzeme Listesi**

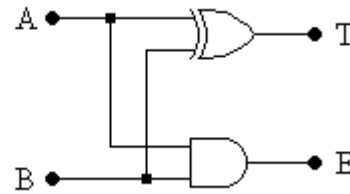
Malzeme	Adet	Kütüphane
7404 – NOT tümdevresi	1	logi7400dip
7408 – AND tümdevresi	1	logi7400dip
74138 – kod çözücü tümdevre	1	logi7400dip
74148 – öncelikli kodlayıcı tümdevre	1	<del>74148-11b</del> 74148_edited2021.circ
74153 – çoğullayıcı tümdevre	1	logi7400dip

### 1. Genel Açıklamalar

Bazı çok değişkenli fonksiyonların doğal yapılarından dolayı, bu fonksiyonların blok yapılar (iterative networks) kullanılarak gerçekleştirilmesi daha uygun olur. Buna göre, eş bloklar birbirleri ile uyumlu bir biçimde bağlanarak istenilen giriş ve çıkış sayısında devreler oluşturulabilir. Bu devreler, bu tür fonksiyonların diğer yöntemlerle gerçekleştirilmesine oranla büyük kolaylık sağlar. Bilindiği gibi giriş sayısı arttıkça fonksiyonun alacağı değer-nokta sayısı da üstel olarak artmaktadır. Blok yapılarda ise, sadece birim modül tasarımı yapılır ve bu modüller birbirlerine bağlanarak geniş ölçekli devreler tasarlanabilir. Bağlantıların ve yapıların basitliği nedeniyle bu blok yapılar, özellikle VLSI devrelerde kullanılmaktadır. Örnek olarak toplama, çıkarma, çarpma, karşılaştırma ve benzeri devreler, blok yapılar kullanılarak tasarlanır.

Bir toplama devresinin en basit blok yapısı, yarı toplayıcı devresidir. Yarı toplayıcı devresinin iki adet girişi, A ve B, ve iki adet çıkışı, E ve T, vardır. A ve B toplanacak iki biti gösterirken T çıkışı, toplamı, E çıkışı ise bu toplam sonucunda oluşan eldeyi gösterir. Şekil 1'de yarı toplayıcı devresinin doğruluk tablosu ve kapı elemanları ile tasarımı verilmiştir.

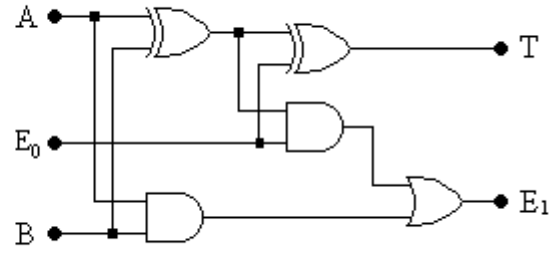
A	B	E	T
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0



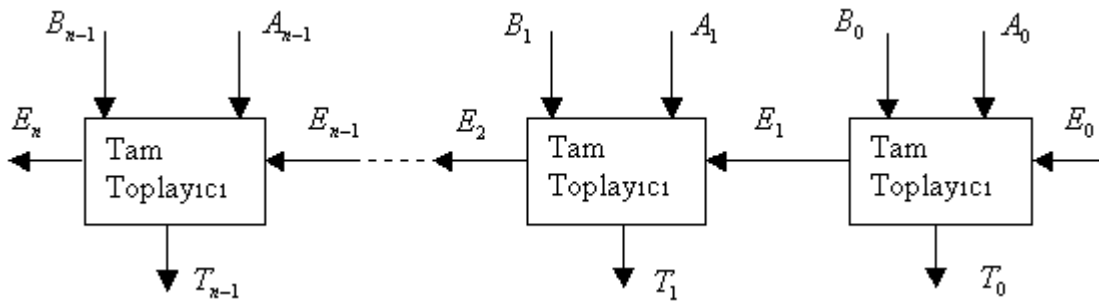
**Şekil 1:** Yarı toplayıcı doğruluk tablosu ve devresi.

Bir bitten daha fazla bit içeren sayıların toplama işleminde, bitlerin toplamında ortaya çıkan elde bitlerini göz önüne almak gerekir. Yarı toplayıcılar kullanılarak tasarlanan tam toplayıcı devresi ve doğruluk tablosu, Şekil 2'de verilmiştir. Tam toplayıcılar, 3 girişli 2 çıkışlı bloklardır. Yarı toplayıcılardan farklı olarak elde girişinin de blok yapısına katılmasıyla aritmetik toplama işlemini gerçekleştirecek modüller elde edilmektedir. Elde çıkışlarının düşük anlamlı bitlerden yüksek anlamlı bitlere Şekil 3'teki gibi aktarılmasıyla  $n$ -bitlik paralel toplayıcı elde edilir.  $n$ -bitlik paralel toplama devresinde, toplam sonucunun oluşması için  $n$ . tam toplayıcı bloğu,  $n-1$  adet tam toplayıcı bloğunun oluşturduğu elde bitini beklediğinden dolayı yavaş çalışır. İstenmeyen bu durumu engellemek için paralel toplama devreleri, öngörülü elde üretici (look ahead carry) toplama devreleri ile tasarlanır.

$E_0$	A	B	T	$E_1$
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1



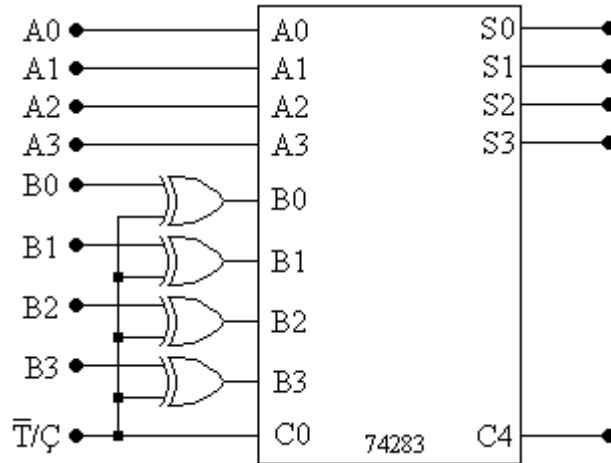
Şekil 2: Tam toplayıcı doğruluk tablosu ve devresi.

Şekil 3:  $n$ -bitlik paralel toplayıcı yapısı.

Toplama devreleri ile tabana veya tabanın 1 eksiğine tümleme yöntemleri kullanılarak çıkarma işlemi gerçekleştirilebilir. Sayısal sistemlerde genellikle taban olarak 2 kullanıldığından ikiye veya bire tümleme kullanılarak çıkarma işlemi gerçekleştirilir. İkiye tümleme, 2 tabanında, basamak sayısı  $n$  olan bir  $B$  sayısının  $B_2 = 2^n - B$  şeklindeki ifadesidir. İki sayı birbirinden çıkarılacağı zaman çıkarılan sayının ikiye tümleneni ile eksilen sayı toplanır. Böylece,  $T = A + B_2$  toplamı,  $A + 2^n - B = 2^n + (A - B)$  ifadesine eşit olur. Buna göre,

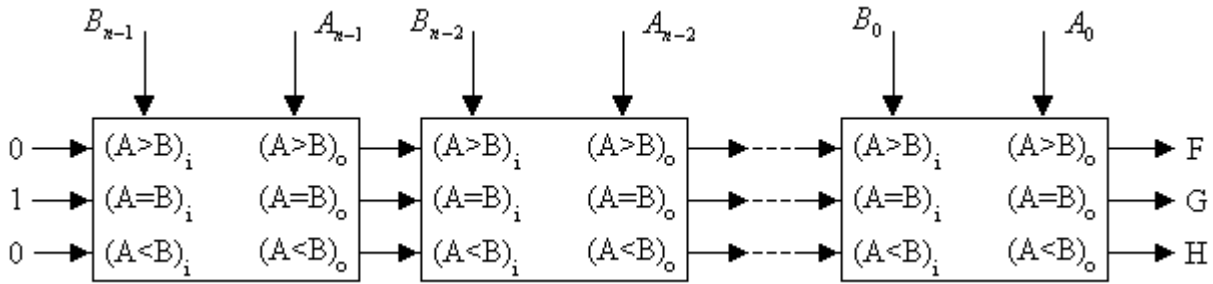
i)  $A \geq B$  ise  $T = 1XXX...X$  biçiminde  $n+1$  hanelidir ve 1 atıldığında ( $2^n$  ifadesi toplamdan çıkarılıyor)  $A-B$  elde edilir.

ii)  $A < B$  ise  $T = 2^n + (A - B) = 2^n - (B - A)$  toplamı  $n$  hanelidir (sonuç negatiftir), bu durumda toplam,  $(B - A)$  sayısının 2 tabanına tümlenmiş olur. Bir sayının 2'ye tümlenmesinin 2'ye tümlenmesi, bu sayının kendisine eşit olacağından dolayı  $T$  toplamının 2 tabanına tümlenmesini ( $T_2$ ) alarak  $(B - A)$  sayısı elde edilmiş olur.  $(A - B)$  farkı ise elde edilen sayının negatiftir. Benzer şekilde bire tümleme ile çıkarma işlemi gerçekleştirilebilir. Şekil 4'te 4-bitlik paralel toplayıcı tümdevresi, 74283, kullanılarak topla ve çıkar kontrol girişleri ile toplama ve ikiye tümleme ile çıkarma işlemini gerçekleyen devre verilmiştir.



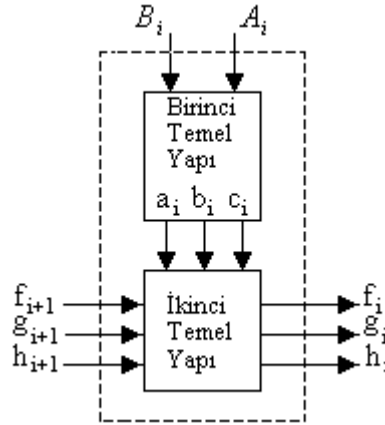
**Şekil 4:** 74283 tümdevresi ile tasarlanan toplama/çıkarma devresi.

İki sayının birbirine göre büyük, küçük veya eşit olduğunu gösteren devrelere karşılaştırma devreleri denir. Karşılaştırma işlemi en yüksek anlamlı bittten veya en düşük anlamlı bittten başlanarak tekrarlamalı olarak yapılabilir. Şekil 5'te verildiği gibi bir bitlik karşılaştırıcı birim modüllerinin birbirlerine kaskad bağlanması ile  $n$ -bitlik karşılaştırıcı devresi gerçekleştirilebilir.



**Şekil 5:** En anlamlı bittten başlanarak  $n$ -bitlik iki sayının karşılaştırılması için kullanılan tekrarlamalı yapı

Karşılaştırma işlemi  $n$ . karşılaştırma birim modülüne  $(A>B)_i = 0$ ,  $(A=B)_i = 1$  ve  $(A<B)_i = 0$  'ın uygulanması ile başlar. Karşılaştırma devresinin birim modülü, iki temel yapıdan oluşur. Birinci temel yapı, iki girişine,  $A_i$  ve  $B_i$ , gelen bitleri karşılaştırarak  $a_i : A_i > B_i$ ,  $b_i : A_i < B_i$  ve  $c_i : A_i = B_i$  çıkışlarını üreten yapıdır. İkinci temel yapı ise A ve B sayılarının  $(n-1)$ . bittten  $(i+1)$ . bite kadar olan bitlerin karşılaştırılması sonucunda elde edilen  $f_{i+1}$ ,  $g_{i+1}$ ,  $h_{i+1}$  girişleri ve birinci temel yapıdan gelen  $a_i$ ,  $b_i$ ,  $c_i$  girişleri ile bir sonraki birim modüle A ve B sayılarının  $(n-1)$ . bittten  $i$ . bite kadar olan bitlerin karşılaştırma sonucunu,  $f_i$ ,  $g_i$ ,  $h_i$ , veren yapıdır. Şekil 6'da karşılaştırma birim modülünün yapıları gösterilmiştir.



**Şekil 6:** Karşılaştırma devresi birim modülünün bloklarla gösterilimi.

Birinci temel yapıda  $a_i$ ,  $b_i$  ve  $c_i$  çıkışlarına ilişkin ifadeler şu şekildedir :

$$a_i = \begin{cases} 1 & \text{eger } A_i > B_i \\ 0 & \text{aksi halde} \end{cases} \quad b_i = \begin{cases} 1 & \text{eger } A_i < B_i \\ 0 & \text{aksi halde} \end{cases} \quad c_i = \begin{cases} 1 & \text{eger } A_i = B_i \\ 0 & \text{aksi halde} \end{cases}$$

Eğer  $A_i > B_i$  ise  $A_i = 1$ ,  $B_i = 0$ 'dır. Yani  $a_i = A_i \cdot \overline{B_i}$  olur.  $i = 0, 1, 2, \dots, (n-1)$

Eğer  $A_i < B_i$  ise  $A_i = 0$ ,  $B_i = 1$ 'dir. Yani  $b_i = \overline{A_i} \cdot B_i$  olur.  $i = 0, 1, 2, \dots, (n-1)$

Eğer  $A_i = B_i$  ise  $A_i = 0$ ,  $B_i = 0$ 'dır veya  $A_i = 1$ ,  $B_i = 1$ 'dir.

$$\text{Yani } c_i = \overline{A_i} \cdot \overline{B_i} + A_i \cdot B_i = \overline{A_i \oplus B_i} = \overline{a_i + b_i} \text{ olur. } i = 0, 1, 2, \dots, (n-1)$$

İkinci yapıda  $f_i$ ,  $g_i$  ve  $h_i$  çıkışlarına ilişkin ifadeler ise şu şekildedir :

$f_i$ , (n-1). bitten i. bite kadar olan bitlerin gösterdiği sayılardan  $A_{n-1} A_{n-2} \dots A_i > B_{n-1} B_{n-2} \dots B_i$  ise 1,  
 $g_i$ , (n-1). bitten i. bite kadar olan bitlerin gösterdiği sayılardan  $A_{n-1} A_{n-2} \dots A_i = B_{n-1} B_{n-2} \dots B_i$  ise 1,  
 $h_i$ , (n-1). bitten i. bite kadar olan bitlerin gösterdiği sayılardan  $A_{n-1} A_{n-2} \dots A_i < B_{n-1} B_{n-2} \dots B_i$  ise 1 değerini almaktadır.

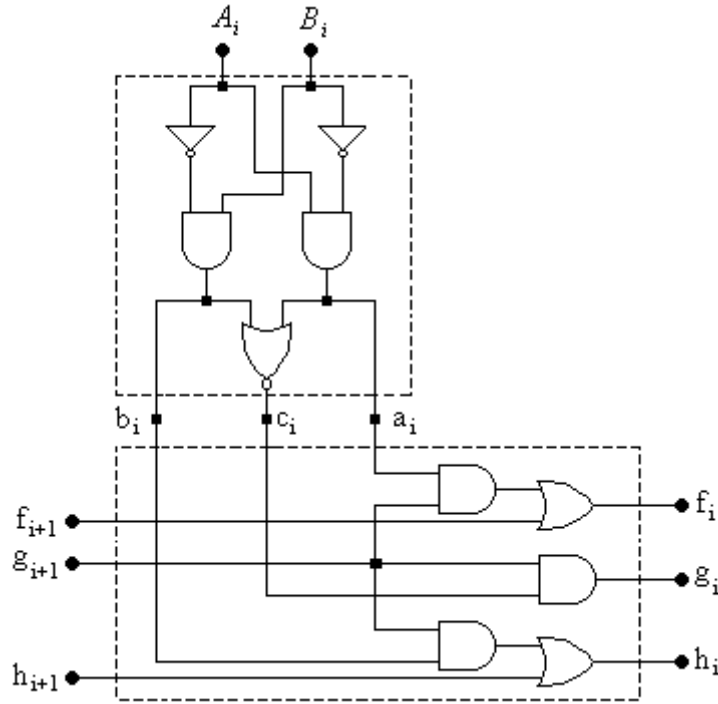
$$f_i = 1 \text{ olması için } f_{i+1} = 1 \text{ veya } g_{i+1} = 1 \text{ ve } a_i = 1 \text{ olması gerekir. } \Rightarrow f_i = f_{i+1} + g_{i+1} \cdot a_i$$

$$g_i = 1 \text{ olması için } g_{i+1} = 1 \text{ ve } c_i = 1 \text{ olması gerekir. } \Rightarrow g_i = g_{i+1} \cdot c_i$$

$$h_i = 1 \text{ olması için } h_{i+1} = 1 \text{ veya } g_{i+1} = 1 \text{ ve } b_i = 1 \text{ olması gerekir. } \Rightarrow h_i = h_{i+1} + g_{i+1} \cdot b_i \text{ olur.}$$

Yukarıda verilen ifadeler ile karşılaştırma devresinin birim modülünün kapı elemanları kullanılarak tasarımı, Şekil 7'de verilmiştir. Ayrıca dört bitlik karşılaştırma devresi olarak 7485 tümdevresi bulunmaktadır.





**Şekil 7:** Karşılaştırma devresi birim modülünün lojik kapılarla gerçekleştirilmesi.

## 2. Deney Sırasında Yapılacaklar

### 2.1. Adım 1

Deney öncesi hazırladığınız 2-bitlik paralel toplayıcı devrenizi deney setine kurunuz. Devrenizde bulunan bütün tümdevrelerin besleme ve toprak bağlantısını yapınız. Devrenizin girişlerini lojik anahtarlardan alıp çıkışlarını LED'lere bağlayarak devrenizin istenilen işlevi gerçekleştirip gerçekleştirmediğini saptayınız.

### 2.2. Adım 2

Şekil 4'te verilen devreyi deney setine kurunuz. Devrede bulunan bütün tümdevrelerin besleme ve toprak bağlantısını yapınız. Devrenin girişlerini lojik anahtarlardan alıp çıkışlarını LED'lere bağlayarak Tablo 1'i doldurunuz.

**Tablo 1:** Toplama ve çıkarma devresi sonuç tablosu.

T/Ç	A	B	$A_3A_2A_1A_0$	$B_3B_2B_1B_0$	$C_4$	$S_3S_2S_1S_0$
0	8	7				
0	11	12				
0	3	4				
1	1	5				
1	6	6				
1	14	9				

**2.3. Adım 3**

7485 tümdevresini deney setine yerleştiriniz. Tümdevrenin girişlerini lojik anahtarlardan alıp gerekli tüm bağlantıları yaptıktan sonra çıkışları LED'lere bağlayınız ve Tablo 2'yi doldurunuz.

**Tablo 2:** Karşılaştırma devresi sonuç tablosu.

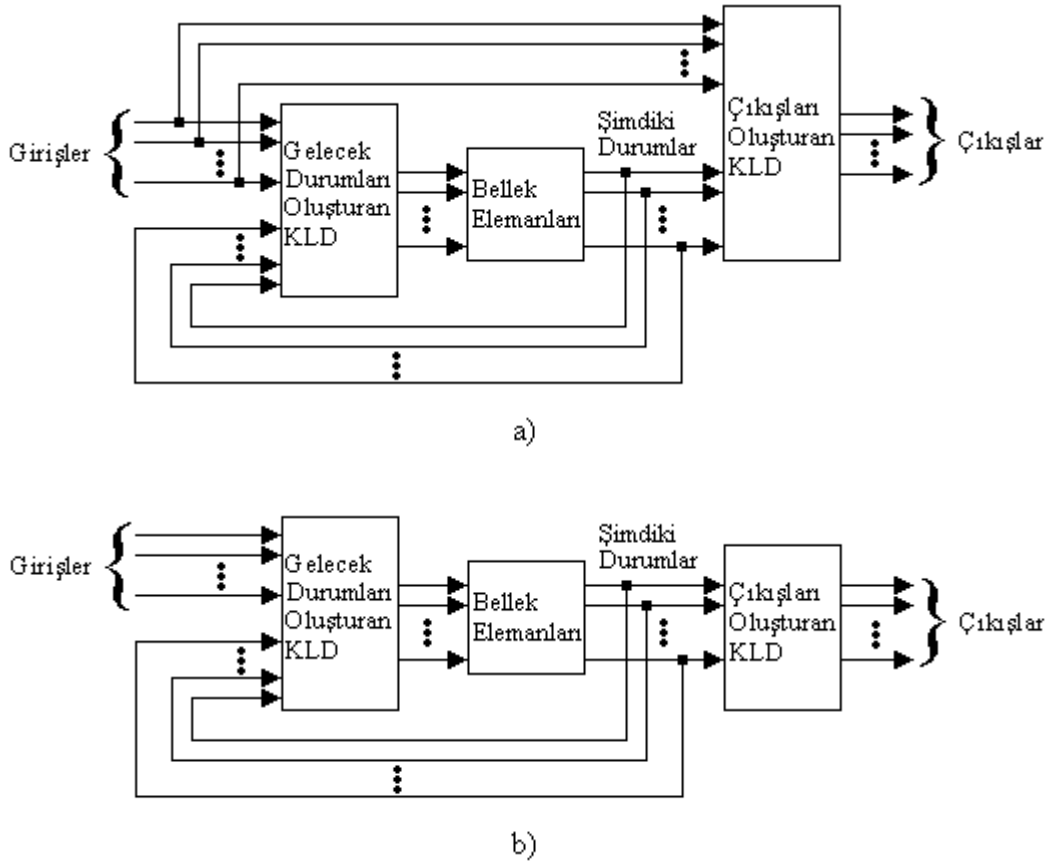
A	B	$A_3A_2A_1A_0$	$B_3B_2B_1B_0$	$A=B$	$A>B$	$A<B$
8	3					
4	4					
12	15					
2	13					
11	11					
10	0					

**3. Malzeme Listesi**

Malzeme	Adet	Kütüphane
7408 – AND tümdevresi	1	logi7400dip
7432 – OR tümdevresi	1	logi7400dip
7485 – 4-bit karşılaştırıcı tümdevre	1	logi7400dip
7486 – EXOR tümdevresi	1	logi7400dip
74283 – 4-bit paralel toplayıcı tümdevre	1	logi7400dip

### 1. Genel Açıklamalar

Kombinezonsal devrelerin çıkışları, sadece o andaki giriş değerlerine bağlı iken ardışıl devrelerin çıkışları, o andaki giriş değerlerine ve durumlara bağlıdır. Dolayısıyla, ardışıl devreler (makine), kombinezonsal devrelerden farklı olarak geçmiş durumları saklayan bellek elemanları içerirler. Çıkış türlerine göre Mealy ve Moore olmak üzere iki tip ardışıl devre vardır. Mealy tipi ardışıl devrede çıkışlar, o andaki girişlere ve durumlara bağlıdır. Moore tipi ardışıl devrede ise çıkışlar yalnızca o andaki durumlara bağlıdır. Bu durum sırasıyla Şekil 1’de gösterilmiştir.



**Şekil 1:** a) Mealy tipi ardışıl devre modeli, b) Moore tipi ardışıl devre modeli.

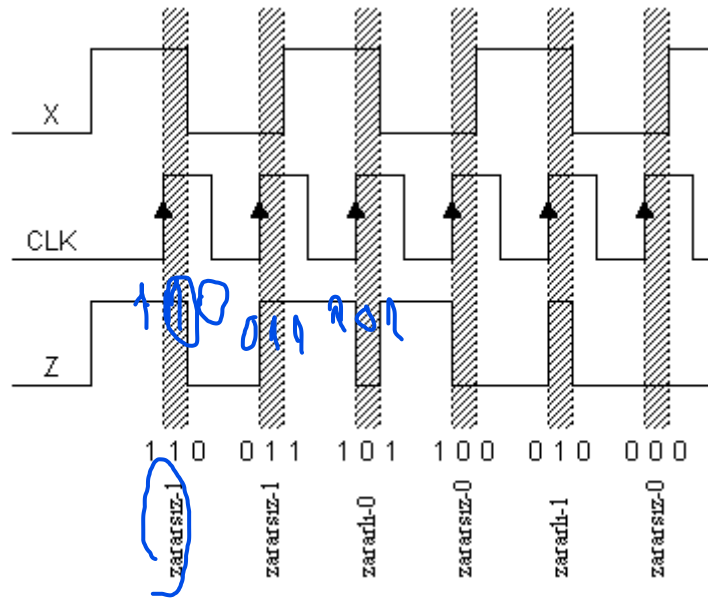
Ardışıl devreler, osilatörlü olup olmamalarına göre asenkron ve senkron olmak üzere ikiye ayrılırlar. Asenkron ardışıl devrelerde merkezi saat yoktur ve durum geçişleri, giriş değerlerinin değişmesi ile sağlanır. Senkron ardışıl devrelerde ise, periyodik saat darbeleri üreten bir merkezi saat vardır ve bu merkezi saat, bütün bellek elemanlarının saat girişlerine bağlanmıştır. Devre, sadece saat tarafından tetiklendiğinde durumunu değiştirir ve yeni durum, devrenin tetiklendiği andaki girişlere ve duruma bağlıdır. Diğer saat darbesi gelene kadar devre durumunu korur. Eğer senkron ardışıl devrede kullanılmayan durumlar varsa ve devre bu durumlardan birine gittiğinde kullanılan durumlara geri dönemiyorsa, devrenin kilitlenen türden olduğu belirtilir.

Bir senkron ardışıl devrenin analiz aşamaları genel olarak şu şekildedir:

- Verilen devre yardımıyla bellek elemanlarının giriş ve ardışıl devrenin çıkış fonksiyonları, o anki durum ve ardışıl devrenin giriş değişkenleri cinsinden belirlenir.
- Bellek elemanlarının giriş fonksiyonları ve tanım bağıntıları kullanılarak devrenin durum denklemleri elde edilir. Durum denklemleri, bir sonraki durumları belirleyen ifadelerdir. Bellek elemanlarının tanım bağıntıları,  $Q_+ = JQ' + K'Q$ ,  $Q_+ = D$ ,  $Q_+ = S + R'Q$  ve  $Q_+ = TQ' + T'Q$  şeklindedir.
- Elde edilen durum denklemleri ve ardışıl devrenin çıkış fonksiyonları ile durum tablosu veya durum diyagramı oluşturulur.

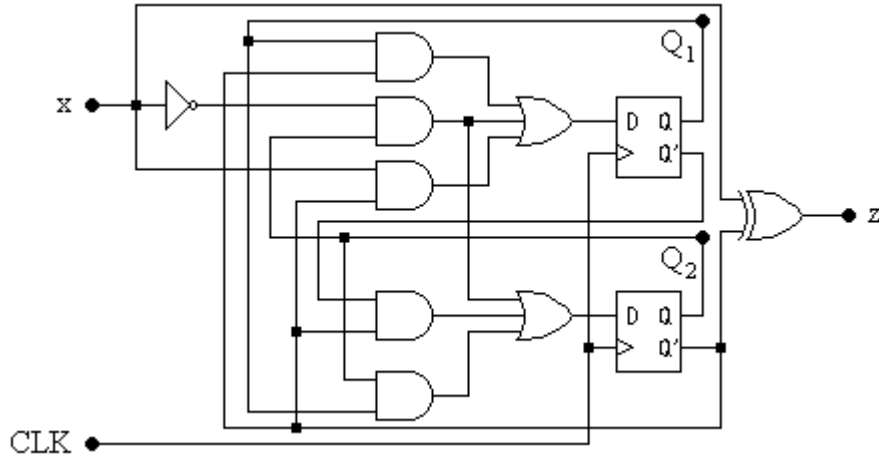
Senkron ardışıl devrelerde, başlangıç durumu ve giriş dizisi verildiğinde, bir sonraki durumların ve çıkışların zamana göre değişimini gösteren diyagramlara zaman diyagramı denir. Fiziksel olarak bir senkron ardışıl devrenin girişleri saat işaretinin tetiklenmesiyle aynı anda değiştirilemeyeceği için girişler, saat işaretinin tetiklenmesinden, yani istenen durumların sağlanmasından ancak bir süre sonra değiştirilebilir. Mealy tipi devrede çıkışlar, o andaki girişlere de bağlı olduğu için bu süreç içerisinde devre çıkışlarında istenmeyen değerler görülebilir. Bu sürece, kritik zaman aralığı denir. Bu zaman aralığında devrenin çıkışları hatalıdır. Hatalı çıkışlar, zararlı veya zararsız olarak ikiye ayrılır. Hatalı çıkışlar aynı zamanda kendi içlerinde, hatalı çıkışın 0 veya 1 değerine sahip olmasına göre ikiye ayrılır. Kritik zaman aralığı öncesi, kritik zaman aralığı ve kritik zaman aralığı sonrasında çıkışta sürekli bir değişim, 010 veya 101 değişimi varsa, bu çıkışın sırasıyla hatalı zararlı 1 veya 0 değerine sahip olduğu belirtilir. Diğer durumlarda çıkış, hatalı zararsızdır. Moore tipi makinelerde ise bu tür hatalı çıkışlar söz konusu değildir. Mealy tipi makinelerde hatalı çıkışlardan kurtulmak için çeşitli yöntemler vardır. Bu yöntemlerden biri, Mealy tipi makinenin gerçeklediği fonksiyonu, durum sayısının artmasını göz önüne alarak Moore makineleri ile gerçeklemektir. Şekil 2'de zararlı ve zararsız çıkışlar gösterilmektedir ve durum geçişlerinin saat işaretinin yükselen kenarında olduğu kabul edilmiştir.

101  
010  
110



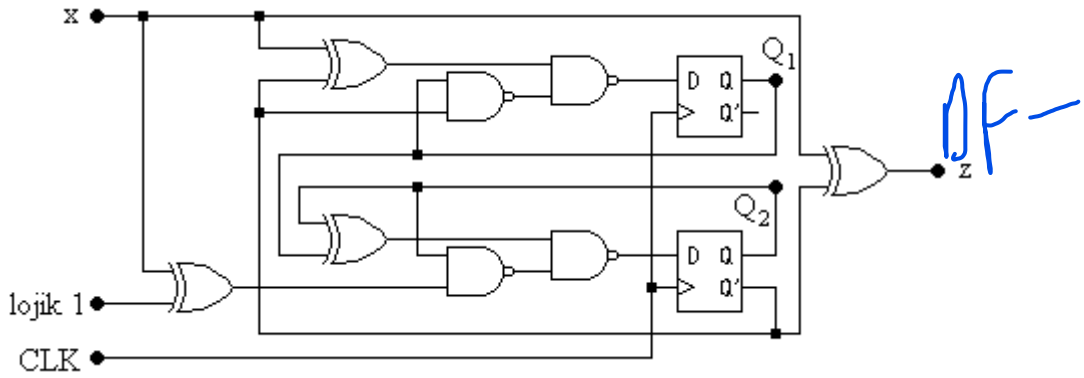
**Şekil 2:** Zararlı ve zararsız hatalı çıkışların zaman diyagramı üzerinde gösterimi.

Bu deneyde analizi yapılacak Mealy makinesine ilişkin devre şeması, Şekil 3'te verilmiştir.



**Şekil 3:** Analizi yapılacak senkron ardışıl devre.

Teorik olarak, SSI elemanları ile sentez yaparken kullanılacak kapı sayısı ve kapı giriş yelpaze sayısı ile bellek elemanı sayısının minimalleştirilmesi esas alınır. Ancak uygulamada, tümleşik devreler kullanıldığından minimalite kavramı, tümleşik devre sayısı ile ilişkili olmaktadır. Şekil 3'te verilen devre için toplam 6 adet tümleşik devre gerekirken ortak bileşenler için ortak yapılar kullanarak ve aynı tür kapı dönüşümü yaparak NAND ve EXOR kapıları ile tasarlanan Şekil 4'teki devre için toplam 3 adet tümleşik devre gerekmektedir.



Şekil 4: Şekil 3'te verilen devrenin NAND ve EXOR kapıları ile tasarımı.

Şekil 3'teki devre yardımıyla bellek elemanlarının giriş fonksiyonları ve çıkış fonksiyonu

$$D_1 = Q_1.Q_2' + X'.Q_2 + X.Q_2' \quad D_2 = X'.Q_2 + Q_1'.Q_2' + Q_1.Q_2 \quad z = x \oplus Q_2' \quad (1)$$

olarak belirlenir. D tipi bellek elemanının tanım bağıntısını kullanarak durum denklemleri;

$$Q_{+1} = Q_1.Q_2' + X'.Q_2 + X.Q_2' \quad Q_{+2} = X'.Q_2 + Q_1'.Q_2' + Q_1.Q_2 \quad (2)$$

olarak bulunur. Tüm giriş ve durum kombinasyonları için bir sonraki durumlar ve devrenin çıkışını içeren durum tablosu Tablo 1'de verilmiştir.

Tablo 1: State transition table for the circuit in Figure 4.

x	Q <sub>1</sub>	Q <sub>2</sub>	Q <sub>+1</sub>	Q <sub>+2</sub>	z
0	0	0	0	1	1
0	0	1	1	1	0
0	1	0	1	0	1
0	1	1	1	1	0
1	0	0	1	1	0
1	0	1	0	0	1
1	1	0	1	0	0
1	1	1	0	1	1

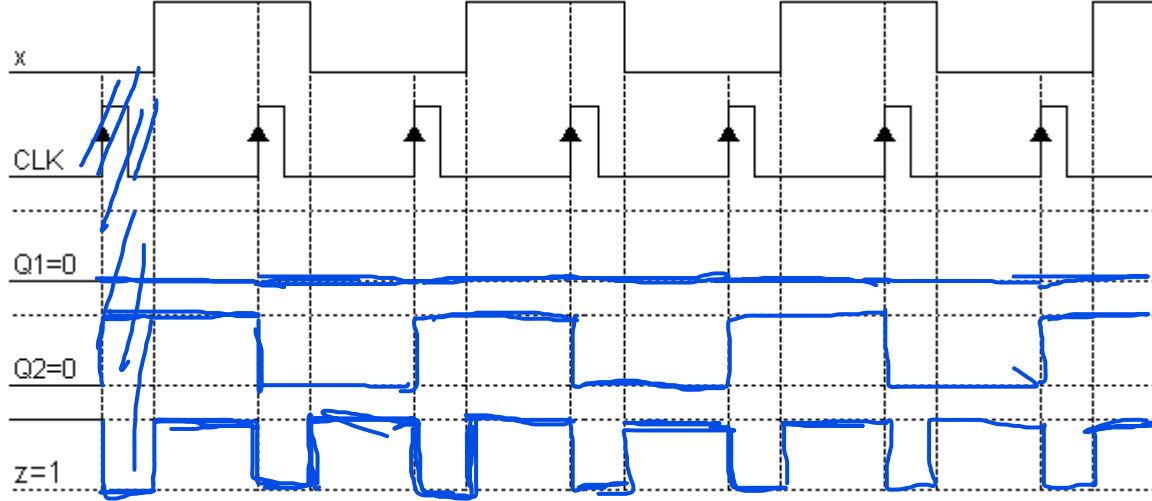
## 2. Deney Sırasında Yapılacaklar

### 2.1. Adım 1

Şekil 4'te verilen devreyi deney setine kurunuz. Bütün tümleşik elemanlara besleme ve toprak bağlantılarını yapınız. Devre girişini, lojik anahtardan aldıktan sonra, istediğiniz başlangıç durumunu elde edebilmek için bellek elemanlarının preset ve clear girişlerini lojik anahtarlara bağlayarak bellek elemanlarının saat girişlerini, ortak debounce pushbutton'dan alınız. Bellek elemanlarının ve devrenin çıkışlarını LED'lere bağlayınız. Kurduğunuz devrenin durum diyagramına göre çalışıp çalışmadığını, bellek elemanlarının clear ve preset girişlerini kullanarak ve bellek elemanlarının ve devrenin çıkışlarını LED'lerden gözleyerek saptayınız.

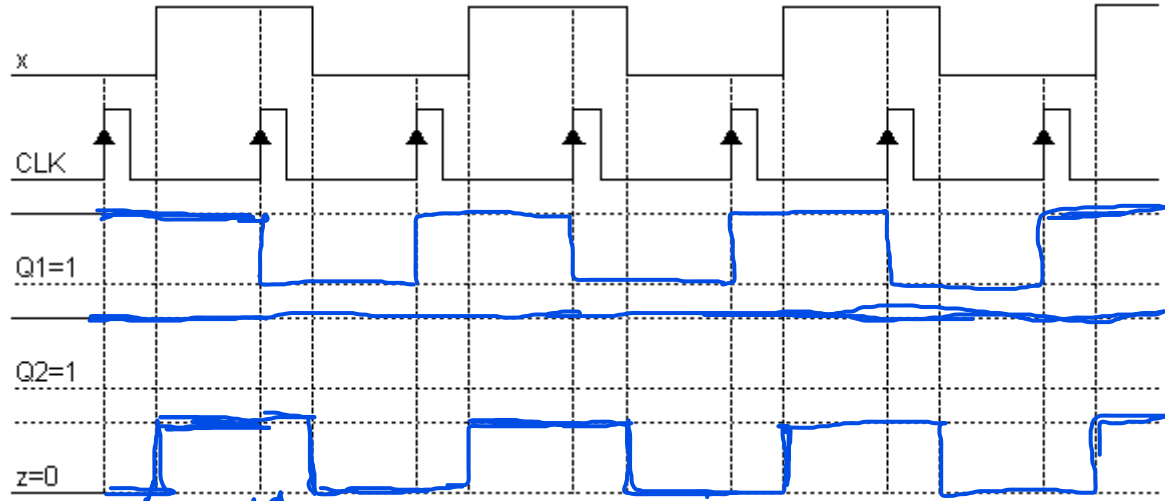
## 2.2. Adım 2

$Q_1Q_2 = 00$  başlangıç durumu için zaman diyagramlarını tamamlayınız. Oluşacak hatalı çıkışların türlerini belirleyiniz.



## 2.3. Adım 3

3.  $Q_1Q_2 = 11$  başlangıç durumu için zaman diyagramlarını tamamlayınız. Oluşacak hatalı çıkışların türlerini belirleyiniz.



## 4. Malzeme Listesi

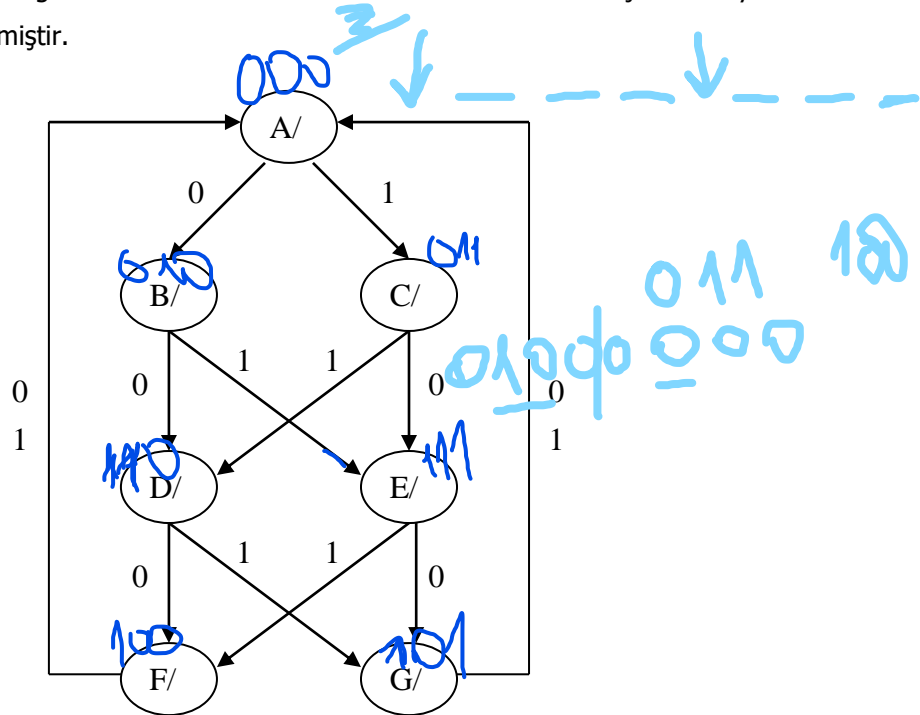
Malzeme	Adet	Kütüphane
7400 – NAND Tümdevresi	1	logi7400dip
7474 – D bellek tümdevresiw	1	logi7400dip
7486 – XOR Tümdevresi	1	logi7400dip

## 1. Genel Açıklamalar

Ardışıl devrelerin tasarımı için çeşitli yöntemler vardır. Aşağıda bu yöntemlerden birinin aşamaları verilmiştir. Buna göre;

1. Ardışıl fonksiyonun sözle tanımından durum diyagramının elde edilmesi.
2. Elde edilen durum diyagramının indirgenmesi (state reduction).
3. Durum kodlaması (state assignment).
4. Durum tablosunun oluşturulması.
5. Ardışıl devrenin tasarımında kullanılacak bellek elemanlarının seçimi.
6. Ardışıl devrenin uyarma tablosunun oluşturulması ve bellek elemanlarının ters tanım bağıntıları ile bellek elemanlarının giriş ve ardışıl devrenin çıkış fonksiyonlarının elde edilmesi.
7. Ardışıl devrenin gerçekleştirilmesi.

Sözle Tanım: Üç bitlik çift eşlenik bit üreticisinin (even parity generator) ardışıl devre elemanları ile tasarlanması istenmektedir. Ardışıl devrenin bir adet seri  $x$  girişi ve bu girişten üç bit alındığında, bu bitlerin içindeki bir sayısı tek ise 1, çift ise 0 değerine ve ara durumlarda 0 değerine sahip olan bir adet  $Z$  çıkışı vardır. Üç bit alındıktan ve buna göre çıkış üretildikten sonra tekrar yeni giriş dizisinin başlangıç bitinin beklendiği duruma geri dönecektir. Verilen sözle tanım ile ardışıl fonksiyonun durum diyagramı, Şekil 1'de verilmiştir.



**Şekil 1:** Üç bitlik çift eşlenik üreticisi durum diyagramı.



Durum indirgeme aşamasında eş durumlara rastlanmaz. Durum kodlaması aşamasında ise Şekil 8.1'de verilen durum diyagramındaki durumlar, A : 000, B : 010, C : 011, D : 110, E : 111, F : 100 ve G : 101 olarak kodlanmıştır. Bu kodlamaya göre oluşan durum ve uyarma tabloları, Şekil 2'de verilmiştir.

M1 100

$Q_1 Q_2 Q_3$ \ $x$	0	1
A : 000	010,0	011,0
B : 010	110,0	111,0
C : 011	111,0	110,0
D : 110	100,0	101,0
E : 111	101,0	100,0
F : 100	000,0	000,0
G : 101	000,1	000,1

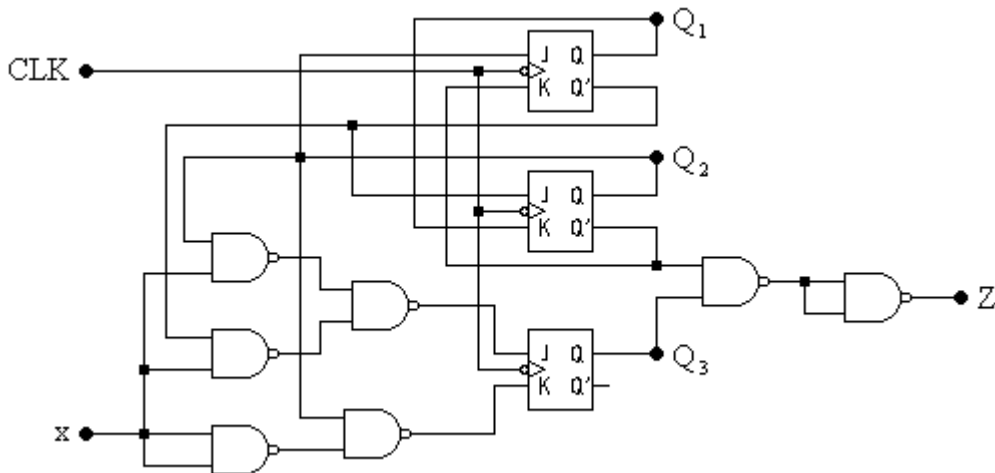
$Q_{+1} Q_{+2} Q_{+3}, Z$

$x$	$Q_1$	$Q_2$	$Q_3$	$Q_{+1}$	$Q_{+2}$	$Q_{+3}$	$Z$
0	0	0	0	0	1	0	0
0	0	0	1	K	K	K	K
0	0	1	0	1	1	0	0
0	0	1	1	1	1	1	0
0	1	0	0	0	0	0	0
0	1	0	1	0	0	0	1
0	1	1	0	1	0	0	0
0	1	1	1	1	0	1	0
1	0	0	0	0	1	1	0
1	0	0	1	K	K	K	K
1	0	1	0	1	1	1	0
1	0	1	1	1	1	0	0
1	1	0	0	0	0	0	0
1	1	0	1	0	0	0	1
1	1	1	0	1	0	1	0
1	1	1	1	1	0	0	0

K : Keyfi

Şekil 2: Senkron ardışıl devrenin durum ve uyarma tabloları.

Ardışıl devrenin tasarımında kullanılacak bellek elemanları için JK bellek elemanı seçilmiştir. JK bellek elemanının ters tanım bağıntısı ve devrenin uyarma tablosu yardımıyla bellek elemanlarının giriş fonksiyonları,  $J_1 : Q_2$ ,  $K_1 : Q_2'$ ,  $J_2 : Q_1'$ ,  $K_2 : Q_1$ ,  $J_3 : x.Q_1' + x.Q_2$ ,  $K_3 : x + Q_2'$  ve devrenin çıkış fonksiyonu,  $Z : Q_2'.Q_3$  şeklinde belirlenir. Bu fonksiyonların NAND kapıları ile gerçekleştirilmesi sonucu oluşan devre, Şekil 3'te verilmiştir.



Şekil 3: NAND kapıları ve JK bellek elemanları ile çift eşlenik bit üreticisi.

## 2. Deney Sırasında Yapılacaklar

### 2.1. Adım 1

Şekil 3'te verilen devreyi deney setine kurunuz. Kullanılan bütün tümdevrelerin besleme ve toprak bağlantısını yapınız. Devrenizin  $x$  girişini lojik anahtardan alırken CLK girişini debounce pushbutton'dan alınız. Bellek elemanlarının Preset ve Clear girişlerini başlangıç durumlarını ayarlayabilmek için lojik anahtarlara bağlayınız. Devrenizin çıkışını ve bellek elemanlarının çıkışlarını LED'lere bağlayarak devrenizin durum tablosunu sağlayıp sağlamadığını gösteriniz.

### 2.2. Adım 2

000 durumundan başlayarak her saat darbesinin düşen kenarından önce bir bit gönderecek şekilde  $x$  girişine 0101000111010100101 dizisini (ilk bit 1, yani en sağdan başlayarak) uygulayıp çıkış dizisini belirleyiniz.

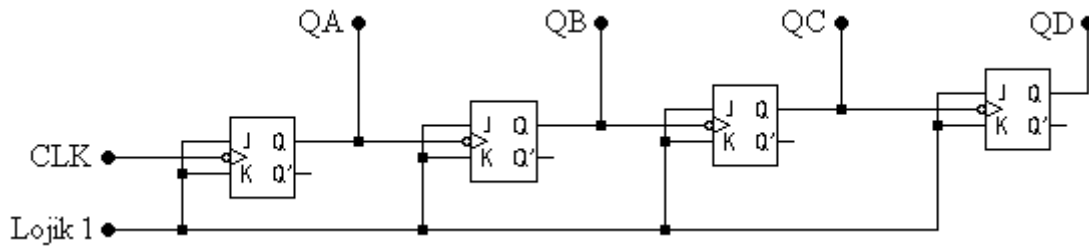
10010000100010101000

## 3. Malzeme Listesi

Malzeme	Adet	Kütüphane
7400 – NAND tümdevresi	2	logi7400dip
7476 – JK bellek tümdevresi	2	logi7400dip

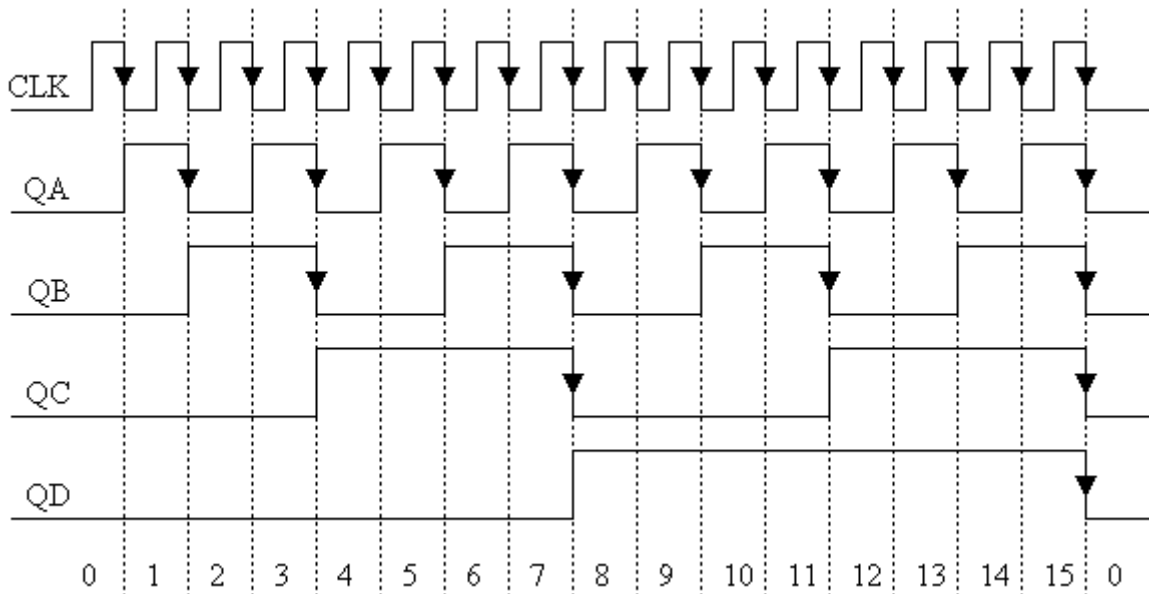
## 1. Genel Açıklamalar

En az bir durum dizisini tekrar eden ardışıl devrelere sayıcı denir. Sayıcılar, çeşitli durum dizilerini tekrar edebilir. Buna göre sayıcılar, ileri, geri, ileri/geri, programlanabilir, ikili kodda, BCD, Gray gibi çeşitli türlerde olabilir. Sayıcılar, bellek elemanlarının tetiklenmesine göre asenkron ve senkron olarak ikiye ayrılır. Asenkron sayıcılarda, bir bellek elemanı kendinden daha düşük anlamlı ilk bellek elemanının çıkışı ile tetiklenir. En düşük anlamlı bellek elemanı ise uygulanan darbeler (veya saat) ile tetiklenir. Şekil 1’de modülo 16 ( $2^4$ ) asenkron ileri sayıcısı verilmiştir.



**Şekil 1:** 4-bitlik asenkron ileri sayıcı devresi.

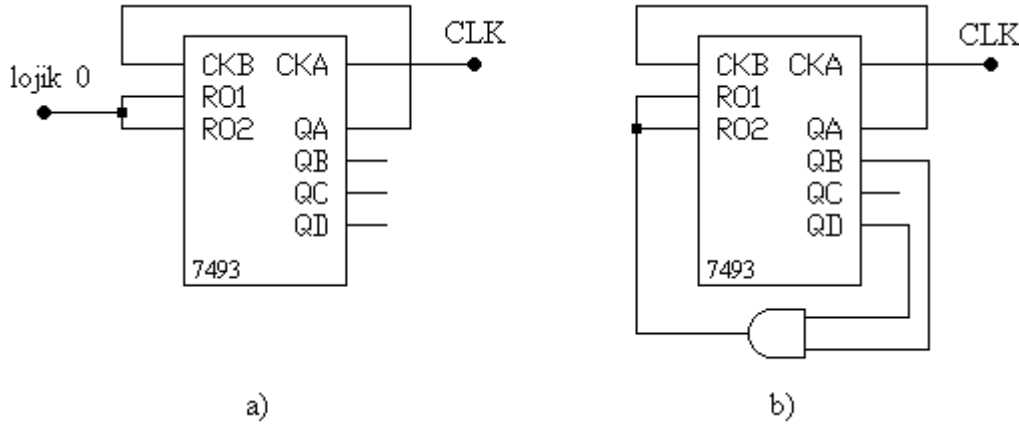
Şekil 1’de verilen devredeki J-K bellek elemanları düşen kenar tetiklemelidir ve tüm J ve K girişlerine lojik 1 değeri atanmıştır. Böylece J-K bellek eleman çıkışlarının, (Q) bir önceki çıkışlarının tümleyeni olması sağlanmıştır. Asenkron sayıcının zaman diyagramı, Şekil 2’de verilmiştir.



**Şekil 2:** 4-bitlik asenkron ileri sayıcının zaman diyagramı.

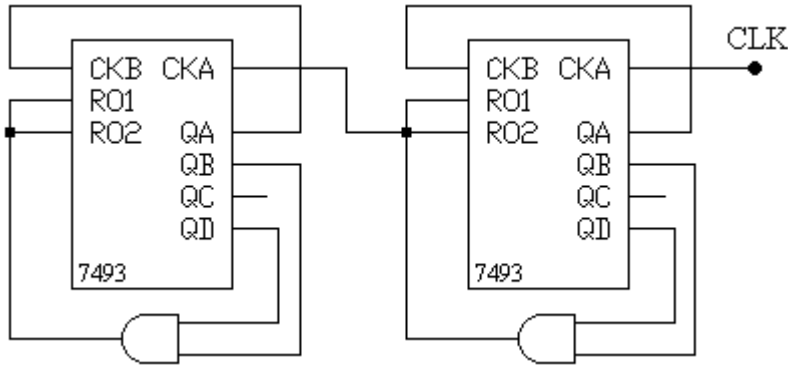
Asenkron sayıcı tümdevresi olarak 7493 tümdevresi örnek gösterilebilir. 7493 tümdevresi, 4-bitlik bir asenkron sayıcıdır. Bu tümdevrenin kontrol girişi olarak 2 adet reset girişi, RO1, RO2, iki adet saat girişi, CKA, CKB ve dört adet çıkışı, QD, QC, QB ve QA (MSB : QD) vardır. RO1 ve RO2 reset girişlerinin

her ikisi de lojik 1 değerinde aktiftir ve bu iki kontrol girişi lojik 1 değerine sahip olduğunda çıkışlar, lojik 0 değerini alırlar, yani sayıcı sıfırlanır. Reset girişleri aktif değil iken QA çıkışı, CKB'ye bağlandığında 7493 tümdevresi modülo 16 sayar. Bu durum, Şekil 3a'da gösterilmiştir. Şekil 3b'de, 7493 tümdevresi ve bir AND kapısı ile tasarlanan BCD sayıcısı verilmiştir. Buna göre BCD sayıcı gerçekleşirken çıkışlar 1010 (QD ve QB lojik 1) olduğunda AND kapısının çıkışı, lojik 1 olacak ve reset girişleri, tümdevrenin çıkışlarını sıfırlayacak ve sayma işlemi devam edecektir.



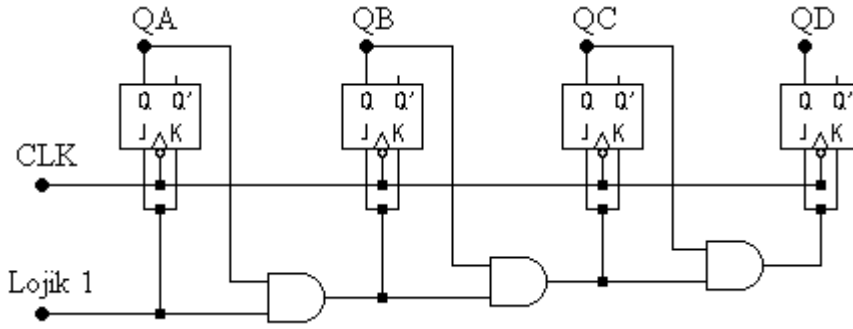
**Şekil 3:** a) 7493 tümdevresi ile modülo 16 sayıcı, b) 7493 tümdevresi ile BCD sayıcı.

Şekil 3b'de verilen BCD sayıcı birim bloğu ile elde edilen 0-99 sayıcısı Şekil 4'te verilmiştir.



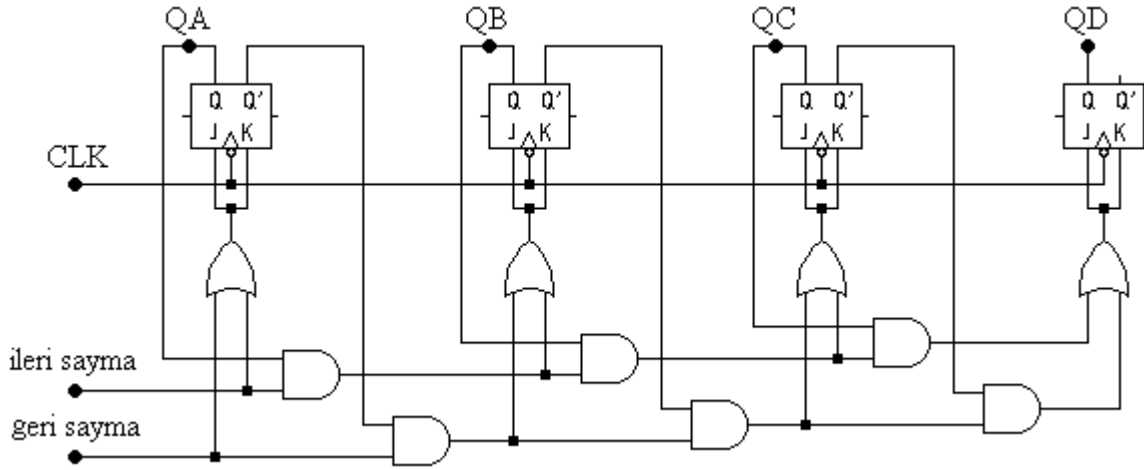
**Şekil 4:** Şekil 9.3b'de verilen BCD sayıcı yardımıyla gerçekleştirilen 0-99 sayıcı.

Senkron sayıcılarda ise merkezi bir saat bütün bellek elemanlarını aynı anda tetikler. Bu yüzden senkron sayıcılar, asenkron sayıcılara göre daha hızlıdır. Şekil 5'te 4-bitlik senkron ileri sayıcı devresi verilmiştir.



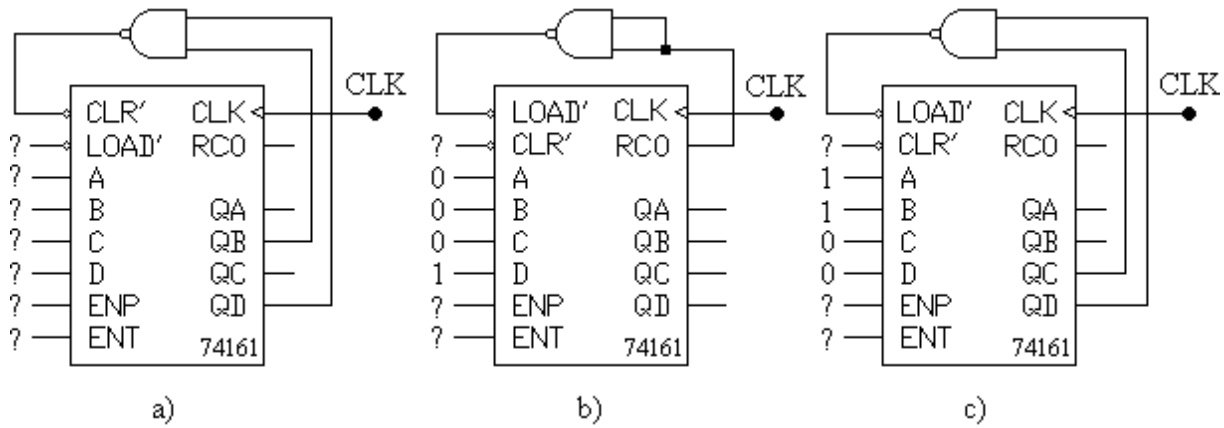
**Şekil 5:** 4-bitlik senkron ileri sayıcı.

Şekil 5'te verilen devre yardımıyla tasarlanan 4-bitlik ileri/geri sayıcı devresi Şekil 6'da gösterilmiştir.



**Şekil 6:** 4-bitlik senkron ileri/geri sayıcı.

İleri/geri sayma yetenekleri yanında senkron sayıcılara paralel yükleme yeteneği kazandırılarak sayıcının istenilen sayıdan başlayarak sayması sağlanabilir. 74161, paralel yüklemeli bir sayıcı tümdevresidir. ENableP (ENP), ENableT (ENT), Load ve Clear olmak üzere dört adet kontrol girişi, saat girişi ve dört bitlik paralel girişi ile dört bitlik paralel çıkışı ve bir bitlik elde çıkışı vardır. Clear ve Load lojik 0'da, ENP ve ENT ise lojik 1'de aktiftir. Clear girişi, tümdevrenin paralel çıkışlarına lojik 0 değerini yüklemek, Load girişi ise paralel girişteki değeri yüklemek için kullanılırken ENP ve ENT kontrol girişleri sayma işlemini durdurmak veya devam ettirmek amacıyla kullanılır. Şekil 7'de 74161 tümdevresi ile tasarlanan sayıcı örnekleri verilmiştir.



**Şekil 7:** a) BCD sayıcı, b) 8-15 arası sayıcı, c) 3-12 arası sayıcı.

Sayıclar genelde, bir olayın gerekleşme sayısının saptanmasında veya sayısal bir sistemde işlemleri denetlemekte kullanılan zamanlama işaretlerinin elde edilmesinde kullanılır. Bu uygulamalar, frekans bölme, bilgi saklama, darbe sayma gibi uygulamalar olabilir.

## 2. Deney Sırasında Yapılacaklar

### 2.1. Adım 1

Şekil 1’de verilen devreyi deney setine kurunuz. Tümdevrelerin gerekli bütün bağlantılarını yapınız. Bellek elemanlarının saat girişini 1Hz’lik TTL dalga işaretinden alınız. Bellek elemanlarının girişlerine uygun değerleri verdikten sonra çıkışları LED’lerden gözleyerek devrenizin istenilen işlevi gerekleyip gereklemediğini saptayınız.

### 2.2. Adım 2

Şekil 3b’deki devreyi deney setine kurunuz. Tümdevrenin gerekli bütün bağlantılarını yapınız. Saat işaret girişini, 1Hz’lik TTL dalga işaretinden alınız. Tümdevrenin çıkışlarını LED’lerden gözleyerek devrenizin istenilen işlevi gerekleyip gereklemediğini saptayınız.

### 2.3. Adım 3

Şekil 5’te verilen devreyi deney setine kurunuz. Tümdevrelerin gerekli bütün bağlantılarını yapınız. Bellek elemanlarının saat girişini 1Hz’lik TTL dalga işaretinden alınız. Bellek elemanlarının çıkışlarını LED’lerden gözleyerek devrenizin istenilen işlevi gerekleyip gereklemediğini saptayınız.

### 2.4. Adım 4

Şekil 7a,b,c’de verilen devreleri deney setine sırasıyla kurunuz. Tümdevrelerin gerekli bütün bağlantılarını yapınız. Saat işareti girişini 1Hz’lik TTL dalga işaretinden alınız. Tümdevrenin çıkışlarını LED’lerden gözleyerek devrenizin istenilen işlevi gerekleyip gereklemediğini saptayınız.

## 3. Malzeme Listesi

Malzeme	Adet	Kütüphane
7400 – NAND tümdevresi	1	logi7400dip
7408 – AND tümdevresi	1	logi7400dip
7476 – JK bellek tümdevresi	2	logi7400dip
7493 – asenkron sayıcı tümdevresi	2	<del>logi7400ic**</del> 7493_edited.circ
74161 – senkron sayıcı tümdevresi	1	logi7400dip