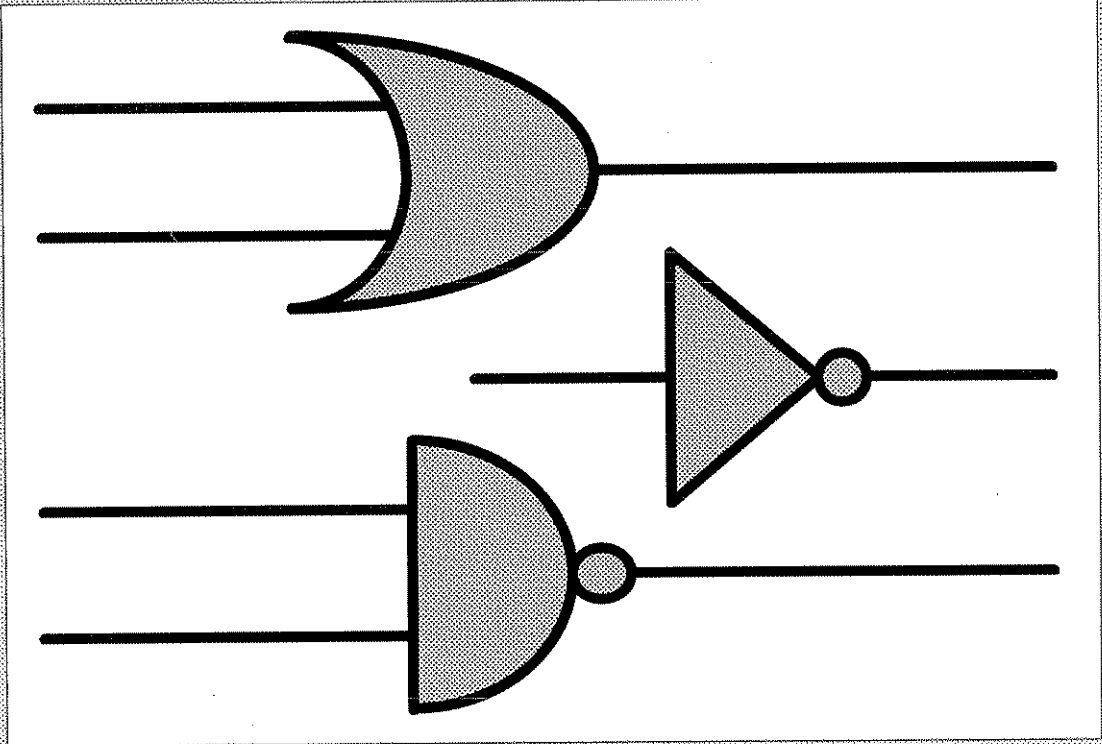




ONDOKUZ MAYIS ÜNİVERSİTESİ
MÜHENDİSLİK FAKÜLTESİ
ELEKTRİK - ELEKTRONİK MÜHENDİSLİĞİ BÖLÜMÜ



SAYISAL ELEKTRONİK LABORATUVARI DENEYLERİ

Yrd.Doç.Dr. Hatice SEZGİN

ELEKTRİK-ELEKTRONİK MÜHENDİSLİĞİ BÖLÜMÜ SAYISAL ELEKTRONİK LABORATUVARI

Deneyin Adı: YOL SEÇİCİLER(MULTİPLEXERS)

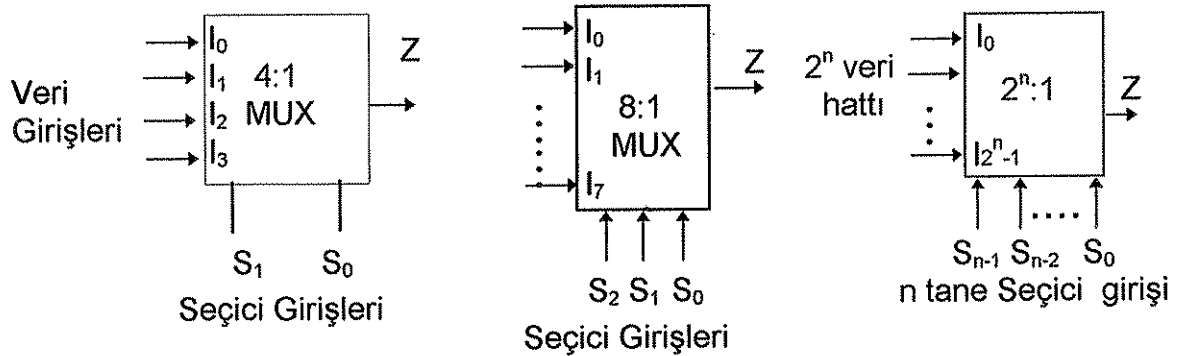
Amaç:

Yol seçiler(MUX)' in işlevlerinin anlaşılabilmesi için, doğruluk tablosunu oluşturup lojik kapılarla bir MUX tasarlanacak ve bu devre MUX tümdevresi ile karşılaştırılacaktır. MUX' ların sayısal elektronikteki kullanım alanları kimi örneklerle anlatılacaktır.

Açıklayıcı Bilgiler:

YOL SEÇİCİLER(MUX)

Yol seçici bir grup veri girişi, bir grup seçici(veya adres) girişi ve bir çıkışa sahiptir. Seçici girişleri, data girişlerinden birini seçerek çıkış ucuna bağlar. Şekil 1'de 4:1 MUX, 8:1 MUX ve $2^n : 1$ MUX gösterilmiştir.



Şekil 1. Multiplexerler

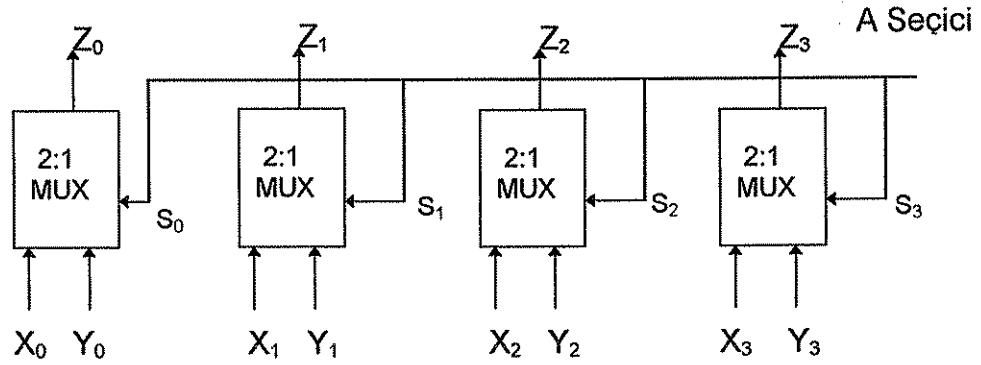
4:1 MUX' un doğruluk çizelgesi aşağıda verilmiştir. Eğer seçici girişleri $AB=00$, çıkış= I_0 benzer şekilde seçici girişleri 01, 10 ve 11 için çıkışlar I_1, I_2 , ve I_3 olarak elde edilir.

$S_1 S_0$	Z
00	I_0
01	I_1
10	I_2
11	I_3

Yol seçicilerin kullanım alanları:

1. Yol seçiciler sayısal dizge tasarımında, yüklenecek veya işlenecek verileri seçmede kullanılır.

Şekil 2 de iki ayrı 4 bitlik bilgiden istenilen bir tanesini seçmeye yarayan MUX' lu devre verilmiştir. Veriler $X_0X_1X_2X_3$ ve $Y_0Y_1Y_2Y_3$ olsun.

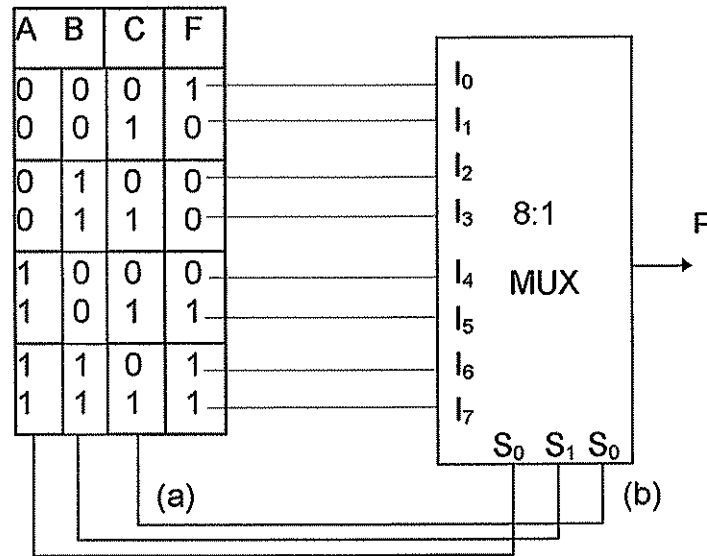


Şekil 2. Quadruple MUX

Eğer seçici girişi $A=0$ ise, $X_3X_2X_1X_0$ verisi çıkışa aktarılır; eğer $A=1$ ise, $Y_3Y_2Y_1Y_0$ verisi çıkışa aktarılır.

2. Seçiciler aynı zamanda lojik mantık işlevleri gerçeklemede kullanılır. n seçici 2^n veri girişli, yani $2^n:1$ MUX kullanarak n değişkenli mantık işlevleri doğrudan gerçekleştirilebilir. Bunun için, n tane seçici(adres) ucuna n değişken bağlanmalı, işlevi oluşturan minterimlere karşılık düşen veri girişleri mantık 1, diğer girişler mantık 0 yapılmalıdır.

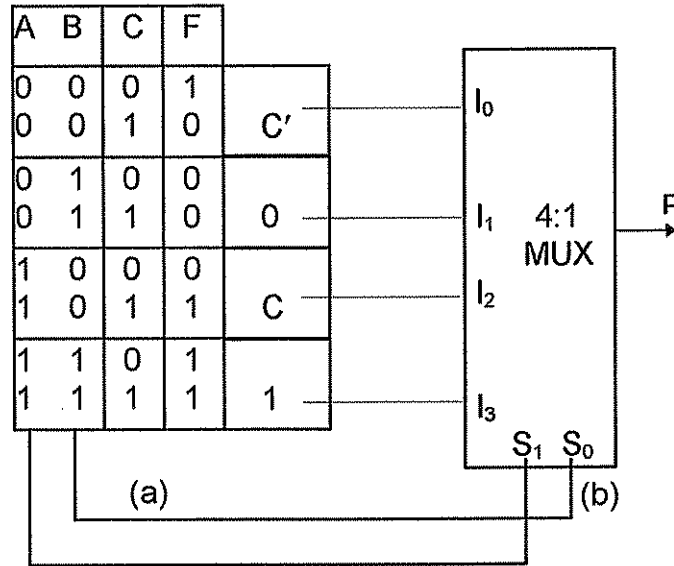
Örneğin 3 değişkenli (A,B,C) $F=\Sigma(0,5,6,7)$ işlevin 8:1 Mux kullanılarak gerçekleştirilmesi şekil 3 te verilmiştir.



Şekil 3. (a) İşlevin doğruluk çizelgesi, (b) Bağlantılanması

n deęişkenli işlev daha da küçük boyutlu MUX kullanarak gerçekenmek istenirse, işleve uygun olarak ek mantık kullanımını gerektirir. Bu durumda deęişkenlerden bazıları seçici uçlarına kalan deęişkenler ise işleve uygun biçimde ek mantık kullanılarak çıkışları veri girişlerine gelebilir.

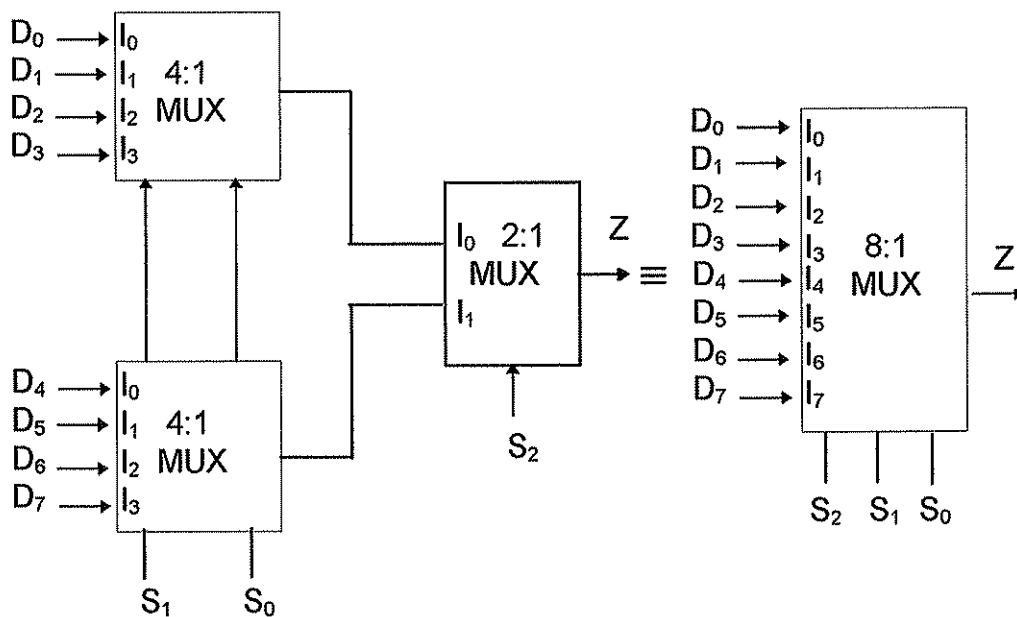
$F = \Sigma(0,5,6,7)$ işlevin 4:1 Mux kullanılarak gerçekenmesi şekil 4 te verilmiştir.



Şekil 4. (a) İşlevin doğruluk çizelgesi, (b) Bağlantılanması

3. Küçük boyutlu farklı MUX' ların uygun biçimde bağlantılanmasıyla daha büyük boyutlu MUX'lar elde edilebilir.

Şekil 5 de 2 tane 4:1 MUX ve 2:1 MUX kullanılarak 8:1 MUX' un gerçekenmesi gösterilmiştir.



Şekil 5. 4-1 ve 2-1 MUX kullanılarak gerçeken 8-1 MUX

Hazırlık Soruları:

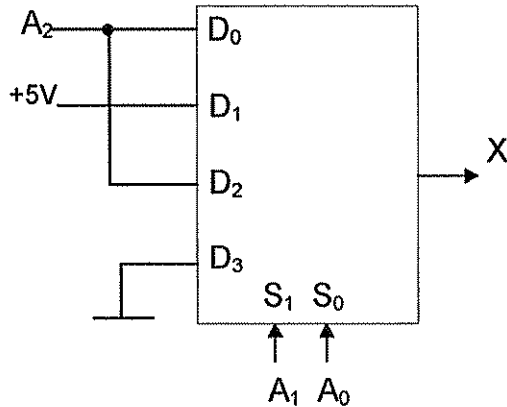
1. 2:1 MUX' un doğruluk tablosunu oluşturun ve lojik kapılarla gerçekleştiriniz.
2. 8:1 MUX kullanarak 64:1 MUX' u elde ediniz.
3. 16:1 ve 4:1 MUX kullanarak 64:1 MUX' u elde ediniz ve 8:1 ile gerçekleştirilen 64:1 MUX ile arasındaki farkı belirtiniz.
4. 8:1 MUX ve 4:1 MUX kullanarak 32:1 MUX, 16:1 MUX ve 2:1 MUX kullanarak 32:1 MUX elde ediniz ve hangi devre daha kullanışlıdır.
5. 4 bitlik 4 tane veriden yalnızca birini seçen devreyi MUX larla gerçekleştiriniz.
6. $f = \sum(0, 3, 5, 6, 9, 10, 12, 15)$ mantık işlevini,
 - a. 4:1 MUX ile gerçekleştiriniz.
 - b. 8:1 MUX ile gerçekleştiriniz.
 - c. (a), (b) şıklarından hangisi daha kullanışlıdır.
7. 74151, 74157 entegrelerin iç yapısını araştırınız.
8. Deneyin yapılışı bölümünde tasarlanması gereken devreleri tasarlayınız.

Deneyin Yapılışı

1. 8:1 MUX' u 74151 tüm devresi ile gerçekleştiriniz ve doğruluk tablosunu oluşturunuz.
2. 74151 ve 74157 kullanarak 16:1 MUX gerçekleştiriniz.
3. Dışarıdan girilen 4 bitlik 2 veriden birini seçen devreyi kurunuz ve çeşitli değerler için devreyi kontrol ediniz.
4. 4 bitlik ABCD biçiminde verilen bilgi için tek-eşlik biti üretilecektir(4 bitlik datada 1' lerin sayısı çift ise çıkış=1, tek ise çıkış=0 üretecek). 8:1 MUX kullanarak devreyi tasarlayınız ve kurarak deneyiniz.
5. $A = A_2A_1$ ve $B = B_2B_1$ biçiminde iki bitlik veri karşılaştırılmak isteniyor. Yalnızca $A \geq B$ olduğunda mantık 1, üretecek devreyi 8:1 MUX kullanarak tasarlayınız ve kurarak deneyiniz.

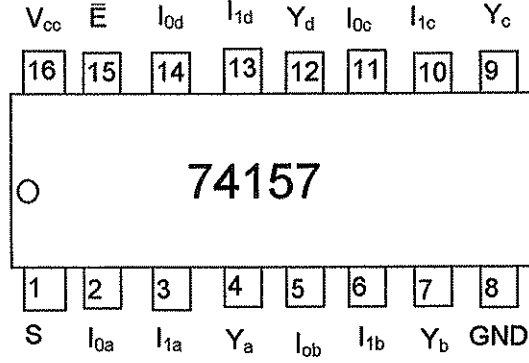
İstenenler:

1. Hazırlık sorularının yanıtlarını düzgün biçimde çizerek kısaca açıklayınız.
2. Deney sonuçlarınızı yorumlayınız.
3. 74151 kullanarak geçersiz BCD kod dedektörü tasarlayınız.
4. Aşağıda mantık devrede gerçekleştirilen Boole işlevini ifade ediniz.

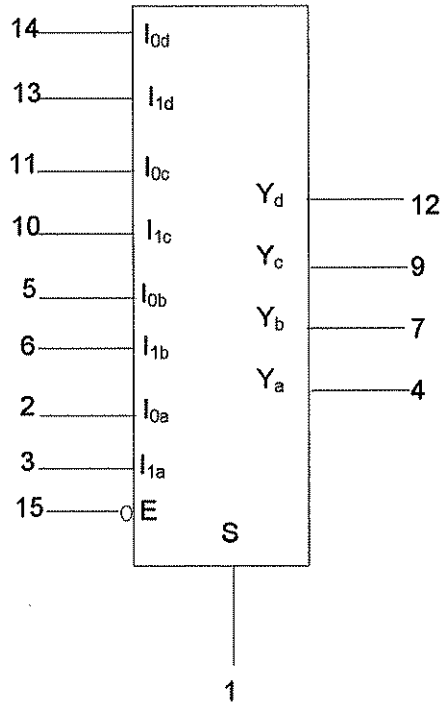


EK: 74157 TM DEVRESİ(4 TANE İKİ GİRİŞLİ MUX)

Ortak seçici giriş ve yetkilendirme girişı kullanılarak, iki ayrı kaynaktan gelen 4 bitlik veriyi seçebilir. Çıkışı, seçilen veriyi aynen gösterir(Şekil 2 deki yapıda).



(a) 74157 tmleřik devre st grnř



(b) lojik sembol

Doğruluk Tablosu

\bar{E}	S	I_0	I_1	Y
H	X	X	X	L
L	H	X	L	L
L	H	X	H	H
L	L	L	X	L
L	L	H	X	H

S: Ortak seçici giriş

\bar{E} : Yetkilendirme(Active Low)

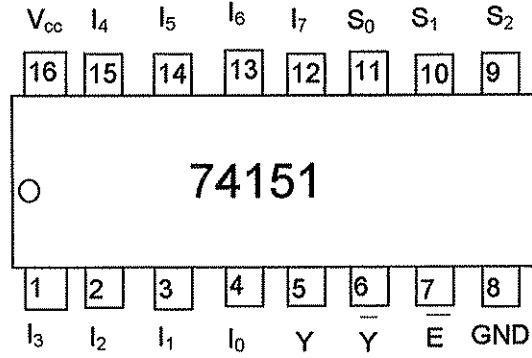
I_{0a} -, I_{0d} : İlk kaynaktan gelen veri girişleri

I_{1a} -, I_{1d} : İkinci kaynaktan gelen veri girişleri

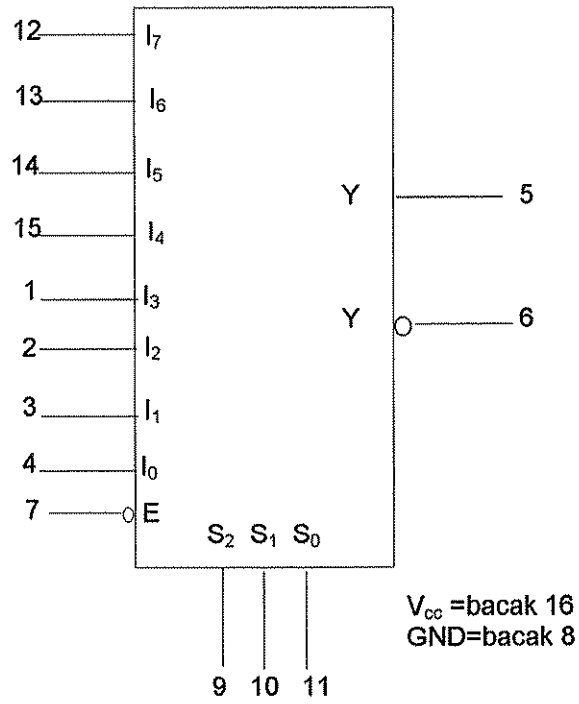
Y_a - Y_d : Çıkışlar

H:Mantık 1, L:Mantık 0, X : 0 veya 1

74151 TM DEVRESİ (8:1 MUX)



(a) 74151 tmleřik devre st grnř



S_2, S_1, S_0 : Seęici giriřler

$I_0, I_1, I_2, I_3, I_4, I_5, I_6, I_7$: Veri giriřleri

\bar{E} : Yetkilendirme (Active Low)

Y: ıkıř

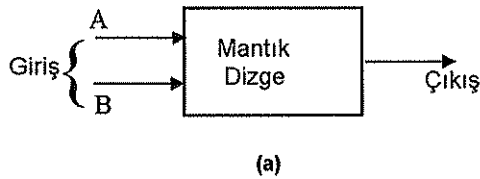
(b) lojik sembol

ONDOKUZ MAYIS ÜNİVERSİTESİ
MÜHENDİSLİK FAKÜLTESİ
ELEKTRİK-ELEKTRONİK MÜHENDİSLİĞİ BÖLÜMÜ

Deney Adı : ARİTMETİK MANTIK BİRİMİ (ALU)

Teorik Bilgi : Bir aritmetik mantık birimi bir MSI (Middle Scale Integration) devredir ve iki değişken üzerinde 16 mümkün mantık işlemi gerçekleştirir.

A ve B gibi iki mantık değişkenini , F çıkışına sahip mantık devresine giriş olarak düşünelim (şekil-1) İçteki mantık dizgede sadece üç temel mantık işlevi (AND , OR , NOT) kullanıldığında , çıkışta kullanılabilecek 16 olası mantık işlemi elde edilebilir.



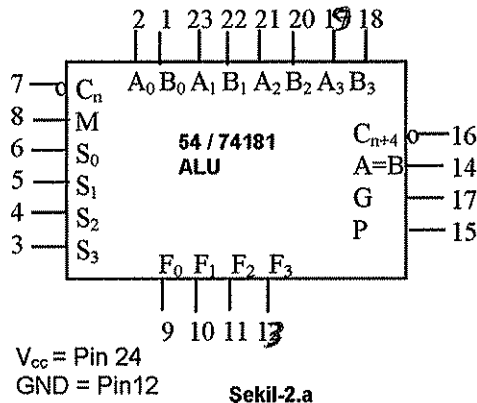
İşlem No İşlem Adı

1	A
2	\bar{A}
3	B
4	\bar{B}
5	$A \cdot \bar{A} = 0$
6	$A + \bar{A} = 1$
7	$A \cdot B$
8	$\overline{A \cdot B}$
9	$A + B$
10	$\overline{A + B}$
11	$A \cdot \bar{B}$
12	$\overline{A \cdot \bar{B}}$
13	$\bar{A} + B$
14	$\overline{A + B}$
15	$A \oplus B$
16	$A \odot B$

Şekil-1

(b)

Bu deneyde incelenecek olan MSI '181 ALU , şekil-2 'de verilmiştir. Giriş değişkeni A , 4-bitlik bir sözcüktür ($A_3A_2A_1A_0$). Ve bu yüzden diğer değişken B de 4-bitlik bir sözcüktür ($B_3B_2B_1B_0$) . F çıkışı da 4-bitlik bir sözcük olacaktır ve 4 işlev seçme girişi ($S_3S_2S_1S_0$) şekil-1.b'de verilen işlemlerden hangisinin gerçekleşeceğini belirler. Çizelge-1'de tüm olası işlevlerin nasıl seçileceği verilmiştir.



Şekil-2.a

DENEY:

1. Çizelge biçiminde verilen mantıksal işlemleri gerçekleştirmek amacıyla , 74181 ALU tüm devresinin çıkış uçlarını ledlere bağlayınız.
2. Girişleri $A=1111$, $B=01010$ seçerek çizelgedeki tüm işlevleri gözleyerek F çıkışlarını kaydediniz.
3. Eğer $S_0S_1S_2S_3 = 1001$, $M=LOW$, $C_n = LOW$, $A= 1100$ ve $B =0101$ ise , F ve C_{n+4} değerleri ne olmalıdır ?

ÇİZELGE-1

54/74181 ALU İŞLEV ÇİZELGESİ

Selection $S_3 S_2 S_1 S_0$				Active-HIGH Data		
				$M = H$ Logic Functions	$M = L$; Arithmetic Operations	
					$C_n = H$ (no carry)	$C_n = L$ (with carry)
L	L	L	L	$F = \bar{A}$	$F = A$	$F = A \text{ PLUS } 1$
L	L	L	H	$F = \overline{A + B}$	$F = A + B$	$F = (A + B) \text{ PLUS } 1$
L	L	H	L	$F = \bar{A}B$	$F = A + \bar{B}$	$F = (A + \bar{B}) \text{ PLUS } 1$
L	L	H	H	$F = 0$	$F \text{ MINUS } 1 \text{ (2's COMP.)}$	$F = \text{ZERO}$
L	H	L	L	$F = \overline{AB}$	$F = A \text{ PLUS } \bar{A}\bar{B}$	$F = A \text{ PLUS } \bar{A}\bar{B} \text{ PLUS } 1$
L	H	L	H	$F = \bar{B}$	$F = (A + B) \text{ PLUS } \bar{A}\bar{B}$	$F = (A + B) \text{ PLUS } \bar{A}\bar{B} \text{ PLUS } 1$
L	H	H	L	$F = A \oplus B$	$F = A \text{ MINUS } B \text{ MINUS } 1$	$F = A \text{ MINUS } B$
L	H	H	H	$F = \bar{A}\bar{B}$	$F = \bar{A}\bar{B} \text{ MINUS } 1$	$F = \bar{A}\bar{B}$
H	L	L	L	$F = \bar{A} + B$	$F = A \text{ PLUS } AB$	$F = A \text{ PLUS } AB \text{ PLUS } 1$
H	L	L	H	$F = \overline{A \oplus B}$	$F = A \text{ PLUS } B$	$F = A \text{ PLUS } B \text{ PLUS } 1$
H	L	H	L	$F = B$	$F = (A + \bar{B}) \text{ PLUS } AB$	$F = (A + \bar{B}) \text{ PLUS } AB \text{ PLUS } 1$
H	L	H	H	$F = AB$	$F = AB \text{ MINUS } 1$	$F = AB$
H	H	L	L	$F = 1$	$F = A \text{ PLUS } A^*$	$F = A \text{ PLUS } A \text{ PLUS } 1$
H	H	L	H	$F = A + \bar{B}$	$F = (A + B) \text{ PLUS } A$	$F = (A + B) \text{ PLUS } A \text{ PLUS } 1$
H	H	H	L	$F = A + B$	$F = (A + \bar{B}) \text{ PLUS } A$	$F = (A + \bar{B}) \text{ PLUS } A \text{ PLUS } 1$
H	H	H	H	$F = A$	$F = A \text{ MINUS } 1$	$F = A$

Selection $S_3 S_2 S_1 S_0$				Active-LOW Data		
				$M = H$ Logic Functions	$M = L$; Arithmetic Operations	
					$C_n = L$ (no carry)	$C_n = H$ (with carry)
L	L	L	L	$F = \bar{A}$	$F = A \text{ MINUS } 1$	$F = A$
L	L	L	H	$F = \overline{AB}$	$F = AB \text{ MINUS } 1$	$F = AB$
L	L	H	L	$F = \bar{A} + B$	$F = \bar{A}\bar{B} \text{ MINUS } 1$	$F = \bar{A}\bar{B}$
L	L	H	H	$F = 1$	$F = \text{MINUS } 1 \text{ (2's COMP.)}$	$F = \text{ZERO}$
L	H	L	L	$F = \overline{A + B}$	$F = A \text{ PLUS } (A + \bar{B})$	$F = A \text{ PLUS } (A + \bar{B}) \text{ PLUS } 1$
L	H	L	H	$F = \bar{B}$	$F = AB \text{ PLUS } (A + \bar{B})$	$F = AB \text{ PLUS } (A + \bar{B}) \text{ PLUS } 1$
L	H	H	L	$F = \overline{A \oplus B}$	$F = A \text{ MINUS } B \text{ MINUS } 1$	$F = A \text{ MINUS } B$
L	H	H	H	$F = A + \bar{B}$	$F = A + \bar{B}$	$F = (A + \bar{B}) \text{ PLUS } 1$
H	L	L	L	$F = \bar{A}\bar{B}$	$F = A \text{ PLUS } (A + B)$	$F = A \text{ PLUS } (A + B) \text{ PLUS } 1$
H	L	L	H	$F = A \oplus B$	$F = A \text{ PLUS } B$	$F = A \text{ PLUS } B \text{ PLUS } 1$
H	L	H	L	$F = B$	$F = \bar{A}\bar{B} \text{ PLUS } (A + B)$	$F = \bar{A}\bar{B} \text{ PLUS } (A + B) \text{ PLUS } 1$
H	L	H	H	$F = A + B$	$F = (A + B)$	$F = (A + B) \text{ PLUS } 1$
H	H	L	L	$F = 0$	$F = A \text{ PLUS } A^*$	$F = A \text{ PLUS } A \text{ PLUS } 1$
H	H	L	H	$F = \bar{A}\bar{B}$	$F = AB \text{ PLUS } A$	$F = AB \text{ PLUS } A \text{ PLUS } 1$
H	H	H	L	$F = AB$	$F = \bar{A}\bar{B} \text{ PLUS } A$	$F = \bar{A}\bar{B} \text{ PLUS } A \text{ PLUS } 1$
H	H	H	H	$F = A$	$F = A$	$F = A \text{ PLUS } 1$

RAPORDA İSTENENLER :

1. Deneyde elde ettiğiniz sonuçları vererek yorumlayınız.
2. A=1101 ve B=0111 verildiğinde (M=L , aktif high giriş) çıkış işareti 1100 oluyorsa mode seçici girişler nasıl seçilmiştir? Aynı giriş işaretleri için M=L ve Cn=L olsaydı aktif low giriş durumunda bulduğunuz mode seçici girişler için hangi çıkışı elde ederdiniz?
3. 8 bitlik A ve B giriş sözcükleri için 74181 kullanılarak ÇİZELGE-1 de verilen işlemler gerçekleştirilebilir mi? Nasıl? Sizerek anlatınız.

ONDOKUZ MAYIS ÜNİVERSİTESİ
Elektrik - Elektronik Müh. Bölümü
Sayısal Elektronik Lab.

Deney Adı : TOPLAYICI

Amaç : Tam toplayıcılar kullanılarak toplama devreleri kurulması.

1.AÇIKLAYICI BİLGİLER :

İşaretili sayılar :

İşaretili sayılar, çoğu bilgisayarda Çizelge 1 de gösterildiği gibi saklanır (4 bit için). Pozitif sayılar gerçek biçiminde, negatif sayılar ikiye tümleyen biçimde saklanır. Sayılar için 4 bitlik yer ayrıldığında toplama sonucu elde edilen sayı için de aynı uzunlukta yer ayrılacaktır. Eğer aynı işaretili sayılar toplanırsa yanıt çok daha geniş olabilir ve ayrılan bitler yetmeyebilir. Buna üst taşma (overflow) denir. Çünkü fazladan bir bite daha taşmak gerekir ve sonucun işaret biti yanlış olur. Farklı işaretili sayıların toplanmasında üst taşma olmaz, sonuç daima doğrudur.

Çizelge 1. 4 - bit işaretili sayıların gösterilimi

10 tabanına göre sayı	Bilgisayardaki gösterilimi
+ 7	0111
+ 6	0110
+ 5	0101
+ 4	0100
+ 3	0011
+ 2	0010
+ 1	0001
0	0000
- 1	1111
- 2	1110
- 3	1101
- 4	1100
- 5	1011
- 6	1010
- 7	1001
- 8	1000

Çizelge 1' deki + 7 den + 1 e kadar olan sayılar gerçek biçimdeki sayılardır, - 1 den - 8 e kadar olan sayılar ikiye tümleyen biçimdeki sayılardır.

Bilgisayar gösterilimindeki soldan ilk bit (MSB) işaret bitidir ve pozitif sayılar için ' 0 ' , negatif sayılar için ' 1 ' olmaktadır.

4 bit işaretli sayıların toplanmasında meydana gelen üst taşma :

a) Zıt işaretli sayılar : İşaret konumunda üst taşma oluşamaz.

$$\begin{array}{r} 0111 = + 7 \\ + 1110 = - 2 \\ \hline 10101 = + 5 \\ \downarrow \\ \downarrow \\ \text{İşaret biti} \end{array}$$

Dışarıda elde var : Yanıtın pozitif olduğunu gösterir.

$$\begin{array}{r} 1001 = - 7 \\ + 0010 = + 2 \\ \hline 01011 = - 5 \\ \downarrow \\ \downarrow \\ \text{İşaret biti} \end{array}$$

Dışarıda elde yok : Yanıtın negatif ve ikiye tümleyen biçimde olduğunu gösterir.

b) Aynı işaretli sayılar : İşaret konumunda üst taşma meydana gelebilir.

$$\begin{array}{r} 1001 = - 7 \\ + 1110 = - 2 \\ \hline 10111 = \text{HATA!} \\ \downarrow \\ \text{İşaret biti değişmiş} \end{array}$$

üst taşma hatası

(sonuç 5 bit olarak - 9 (doğru))

$$\begin{array}{r} 0111 = + 7 \\ + 0010 = + 2 \\ \hline 01001 = \text{HATA!} \\ \downarrow \\ \text{İşaret biti değişmiş} \end{array}$$

üst taşma hatası

(sonuç 5 bit olarak + 9 (doğru))

Bu dört durum için de elde edilen toplama sonucunun doğruluğundan emin olmak için yapılan toplama işleminin üst taşma yanığı taşıyıp taşımadığının denetlenmesi gerekir. Bunu denetleyecek mantık devre, toplama işlemi

$$\begin{array}{r} A_4 A_3 A_2 A_1 \\ B_4 B_3 B_2 B_1 \\ \hline C_4 \Sigma_4 \Sigma_3 \Sigma_2 \Sigma_1 \end{array}$$

biçiminde olmak üzere çizelge 2 ' deki gibi doğruluk çizelgesinden yararlanarak $F = \bar{A}_4 \bar{B}_4 \Sigma_4 + A_4 B_4 \bar{\Sigma}_4$ mantık işlevini gerçekleyecek biçimde tasarlanabilir.

Çizelge 2. Üst taşma yanığı için doğruluk çizelgesi

İşaret Bitleri			Yanığı
A_4	B_4	Σ_4	Y
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	0

Toplama sonucunda üsttaşma yanılığısı yoksa, zıt işaretli sayılar toplanıyor demektir ve sonuç dört bit $\Sigma_4 \Sigma_3 \Sigma_2 \Sigma_1$ olarak doğrudur. Σ_4 sonucun işaret bitidir. 5. bit C_4 'ün dikkate alınmaması gerekir.

Toplama sonucunda üst taşma yanılığısı varsa, aynı işaretli sayılar toplanıyor demektir. Sonuç 5 bittir ve $C_4 \Sigma_4 \Sigma_3 \Sigma_2 \Sigma_1$ olarak doğrudur. C_4 sonucun işaret bitidir.

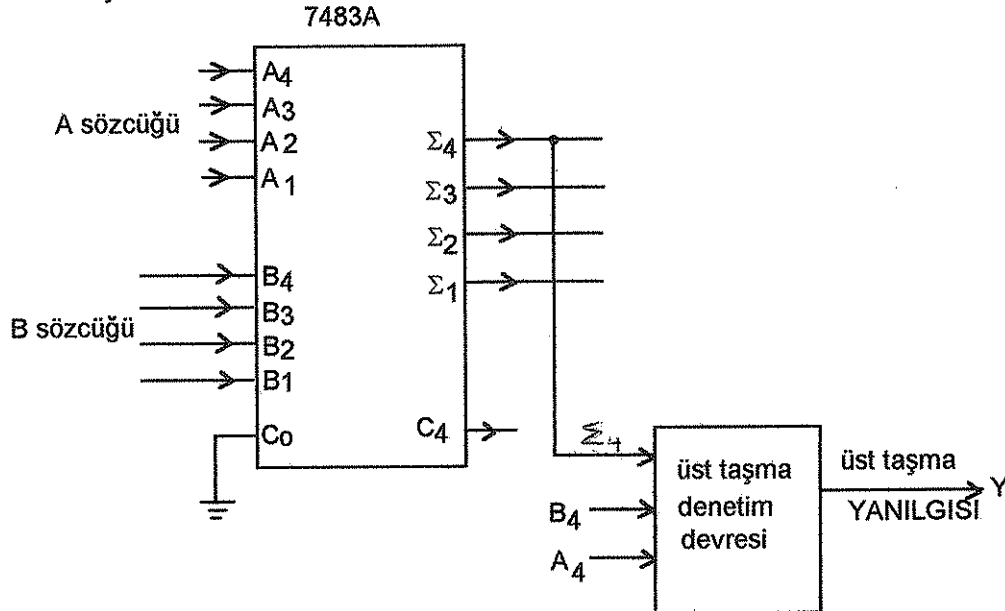
2.HAZIRLIK SORULARI :

1. Şekil 1 deki devrede kullanılacak üst taşma denetim devresini tasarlayınız.
2. Ek 2 'de 7447 BCD - 7 - segment kod çözücü / sürücü tümdevresi tanıtılmıştır. Bu tümdevre 4 - bitlik BCD kodu çözerek 7 - segment göstergede görmemizi sağlar. Buna göre deneyin yapılışı (2) ve (3) teki soruların yanıtını düşününüz.
3. İki tane 7483 tümdevresi kullanarak 8 bit sayıların nasıl toplanabileceğini tasarlayınız.

3. DENEYİN YAPILIŞI :

7483 tümdevrenin iç yapısı Ek 1 'de verilmiştir.

1. Şekil 1 ' deki devreyi kurarak çıkış uçlarını LED ' lere bağlayınız. 4 bitlik işaretli $A_4 A_3 A_2 A_1$ ve $B_4 B_3 B_2 B_1$ sayılarını a) 7 ve 2 b) 4 ve 3 seçerek işaretlerinin olası tüm durumları için üst taşma yanılığısının ortaya çıkıp çıkmadığını denetleyiniz.
2. $Y = 0$ olduğu durumlar için toplama sonucunu LED ' lere bakarak yazınız. Bu durumda BCD kodu onlu sayıya çeviren kod çözücü çıkışındaki 7 - segment göstergelyi kullanarak toplama sonucunu görebilir miyiz, nasıl?
3. $Y = 1$ durumu için (2) dekileri tekrarlayınız.
4. İki tane 7483 kullanarak 8 bitlik işaretli iki sayının toplanması için tasarladığınız devreyi kurunuz ve deneyiniz.



Şekil 1

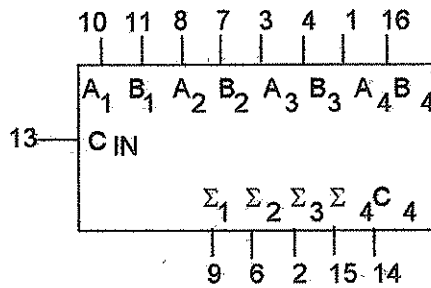
İSTENENLER :

1. Tasarladığınız üsttaşma denetim devresinin mantık çizgesini veriniz.
2. Deney sonuçlarını yorumlarıyla birlikte kısaca veriniz.

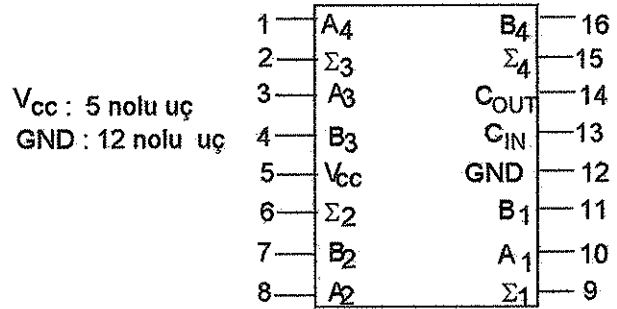
3. Deney 2 ve 3 için , toplama sonucunu 7 - segment göstergede işareti ile birlikte gösterecek mantık devreyi tasarlayınız.
4. Deney 4 te kullandığınız devre öbek çizgesini ve deney sonuçlarını veriniz.

Ek - 1

7483 Tümdevresi : 4 bit ikili tam - toplayıcı (giriş eldesi ile) ($A_1 - A_4, B_1 - B_4$) gibi dört bit iki sözcüğü ve giriş eldesi C_{IN} ' i toplayarak ($\Sigma_1 - \Sigma_4$) ikili toplam çıkışını ve elde çıkışı C_{OUT} ' u (en anlamlı bit olarak) verir.



Şekil 2. 7483 tümdevresi mantık simgesi

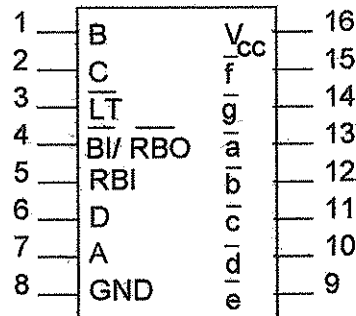
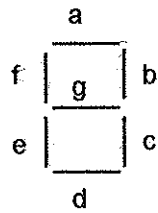


Şekil 3. 7483 tümdevresi üst görünüşü

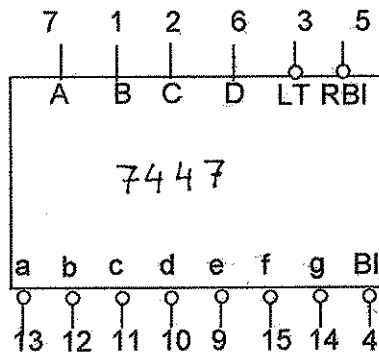
$A_1 - A_4$: A giriş sözcüğü
 $B_1 - B_4$: B giriş sözcüğü
 C_{IN} : Giriş eldesi
 $\Sigma_1 - \Sigma_4$: Toplama çıkışları
 C_{OUT} : Elde çıkışı

Ek - 2

7447 Tümdevresi : BCD yi 7 - segment gösterge için çözer ve sürer.



Üstten görünüşü



Mantık simgesi

A, B, C, D : BCD girişler
 $\bar{a} - \bar{g}$: çıkışlar

ONDOKUZ MAYIS ÜNİVERSİTESİ
MÜHENDİSLİK FAKÜLTESİ
ELEKTRİK-ELEKTRONİK MÜHENDİSLİĞİ BÖLÜMÜ

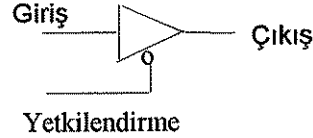
Deney Adı : KAYDEDİCİLER ARASI VERİ AKTARIMI

Amaç : Kaydediciler arasında veri aktarımı için BUS sisteminin ve üç durumlu tamponların tanıtılması.

Açıklayıcı Bilgi:

Üç durumlu Tamponlar : Yetkilendirme veya denetim ucu ile denetlenerek, çıkışları mantık 1 (High), mantık 0 (Low) veya açık (yüksek empedans) durumlarından birini alabilen sayısal devrelerde geniş kullanıma sahiptirler. Üç durumlu kapının mantık simgesi ve doğruluk çizelgesi Şekil-1'de verilmiştir.

Kontrol	Giriş	Çıkış
0	0	0
0	1	1
1	0	Yüksek Emp.
1	1	Yüksek Emp.

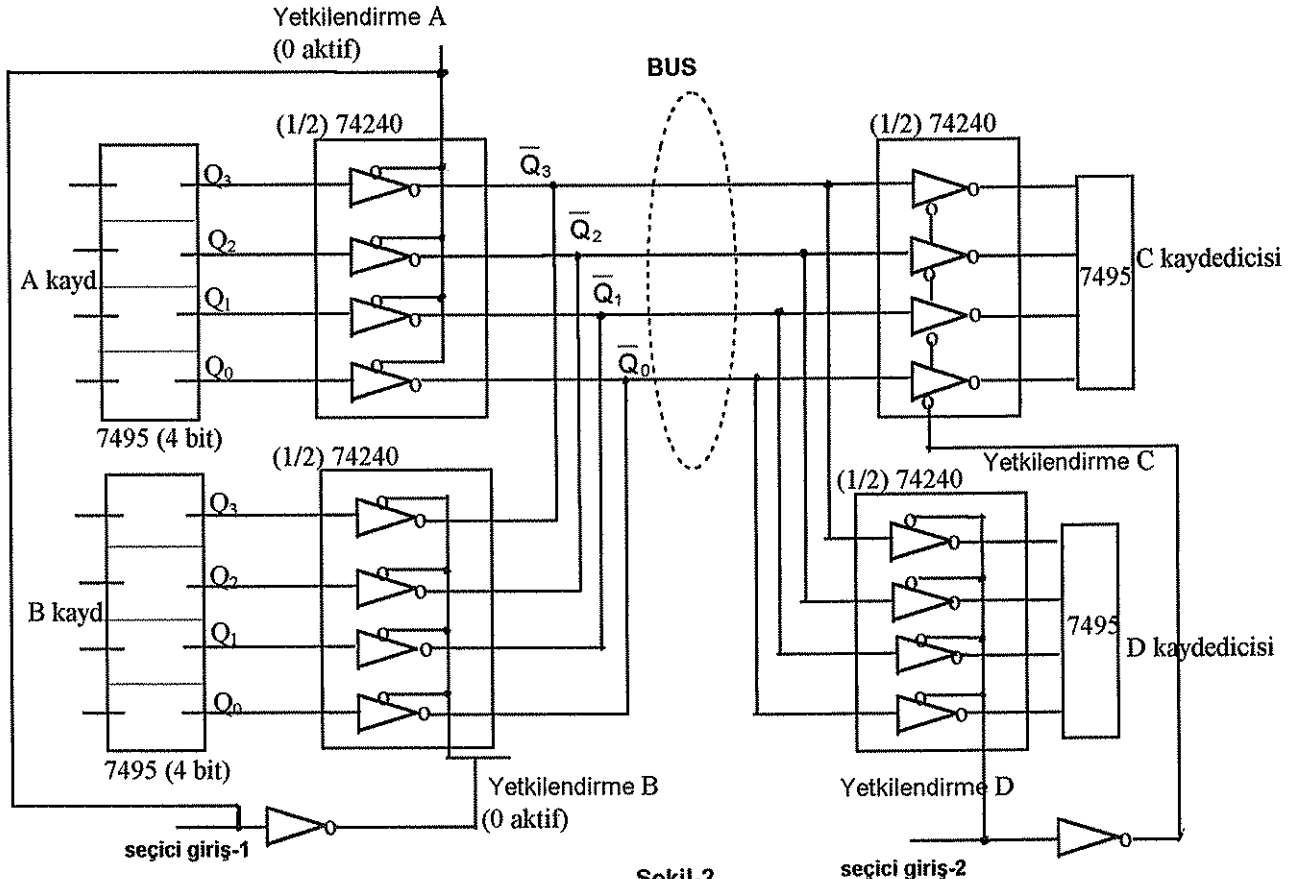


Şekil-1

Bus : Bilgisayar gibi çok sayıda kaydedici içeren sayısal bir dizgede herhangi bir kaydedici ile bir başkası arasında paralel veri aktarımını sağlayabilmek için çok fazla sayıda hat gereklidir. Bunun çözümü için BUS (taşıt, bara) denilen, her biri bir bit aktarabilen tellerden oluşan ortak yol kullanılır. BUS 'ın bir ucu verici, diğer ucu alıcı durumundadır. verici tarafındaki kaydedicilerden yalnızca bir tanesindeki veri, alıcı taraftaki kaydedicilerden yalnızca bir tanesine aktarılması istenir. Bu nedenle kaydedicilerle BUS arasında üç durumlu tamponlar bulunur. Ve bunlardan yalnızca birer tanesi yetkilendirilir. (alıcıdan bir tane ve vericiden bir tane) Bu nedenle BUS 'ın verici tarafındaki (MUX) düzenek, alıcı tarafta yol çoklayıcı (DEMUX) düzenek bulunur.

Deneyin Yapılışı :

Deneyde kullanılacak üç durumlu tampon için 74240 tümdevresi, 4-bitlik kaydediciler için 7495 tümdevresi kullanılacaktır. Bu tümdevreler Ek-1 ve 2'de verilmiştir. Şekil-2'deki devrede 4-bitlik bara kullanılarak bilgi aktarımı incelenecektir. Bilgi aktarım verici taraftaki A, B kaydedicilerinden alıcı taraftaki C, D kaydedicilerine doğru paralel biçimde olacaktır. her iki taraftaki kümede yalnızca ikiser kaydedici bulunduğu için MUX-DEMUX kullanımına gerek kalmadan TÖMLEYEN kapısı kullanılmıştır.



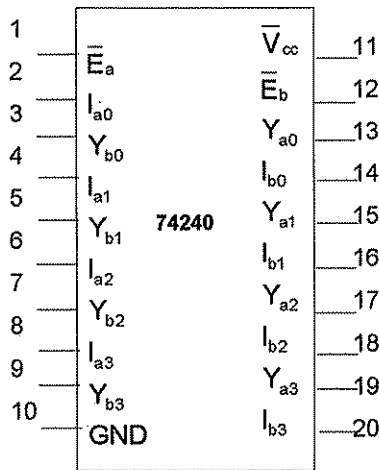
Şekil-2

- Şekil-2 'deki devreyi kurunuz. 1.seçici giriş 0 verilirse A vericisi , 1 verilirse B vericisi seçilmiş olacaktır. 2. seçici giriş 0 verildiğinde D , 1 verildiğinde ise C seçilmiş olur.
- A kaydedicisine 1010 ve B kaydedicisine 1101 bilgisini paralel olarak yükleyiniz.
 - A 'daki bilgiyi C kaydedicisine aktarınız.Bu sırada BUS 'daki bilgi ne olur?
 - A'daki bilgiyi D'ye aktarınız.
 - B'deki bilgiyi D'ye aktarınız.
- A kaydedicisine 1111 vre B kaydedicisine 0001 bilgisini kaydediniz.B'deki bilgiyi C'ye aktarınız.

RAPORDA İSTENENLER :

- Deney sonuçlarını vererek yorumlayınız.
- Verici ve alıcı taraflarda 8'er tane 8-bitlik kaydedici bulunan bir BUS sistemi (bir anda yalnızca iki kaydedici arasında veri aktarımına izin verilebilecek) öbek çizgesi vererek kısaca çalışmasını anlatınız.Verici taraftaki 3.kaydedicilerdeki 10111010 bilgisini , alıcı taraftaki 7.kaydediciye nasıl aktaracağını anlatınız.

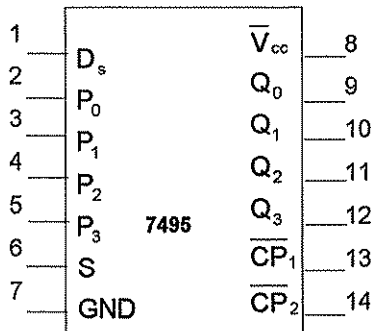
Ek-1 . 74240 üç durumlu tampon. 8 tane 3 durumlu tampon , 4erli gruplar halinde yetkilendiriliyor. $I_{a0} - I_{a3}$ girişlerine karşılık $Y_{a0} - Y_{a3}$ çıkışlarına sahip ve bu grup E_a ucu ile yetkilendiriliyor. $I_{b0} - I_{b3}$ girişlerine karşılık $Y_{b0} - Y_{b3}$ çıkışlarına sahip grubun yetkilendirme ucu E_b .



İŞLEV ÇİZELGESİ					
GİRİŞLER				ÇIKIŞLAR	
\bar{E}_a	I_a	\bar{E}_b	I_b	\bar{Y}_a	\bar{Y}_b
L	L	L	L	H	H
L	H	L	H	L	L
H	X	H	X	(Z)	(Z)

H : High
L : Low
X : Dont carry
(Z) : Yüksek empedans

Ek-2. 7495 4-bit Shift Register : Seri sağa kaydırma ve paralel yükleme , aynı saat girişiyle aktif hale getirilmektedir. Seri veya paralel olarak yüklenen bilgi senkron (eşzamanlı) olarak saat girişinin düşen kenarında çıkışlara iletilir.



üst görünüşü

İŞLEV ÇİZELGESİ									
İŞLEM TÜRÜ	GİRİŞLER					ÇIKIŞLAR			
	S	\overline{CP}_1	\overline{CP}_2	D_s	P_n	Q_0	Q_1	Q_2	Q_3
Paralel Yükleme	H	X	↓	X	P_n	P_0	P_1	P_2	P_3
Kaydırma	L	↓	X	L	X	L	Q_0	Q_1	Q_2
	L	↓	X	H	X	H	Q_0	Q_1	Q_2

S : Seçme girişi \overline{CP}_1 : seri saat girişi (negatif kenar tetiklemeli)
 D_s : Seri bilgi girişi \overline{CP}_2 : Paralel saat girişi (" " ")
 P_0-P_3 : Paralel bilgi girişi
 Q_0-Q_3 : Paralel çıkışlar

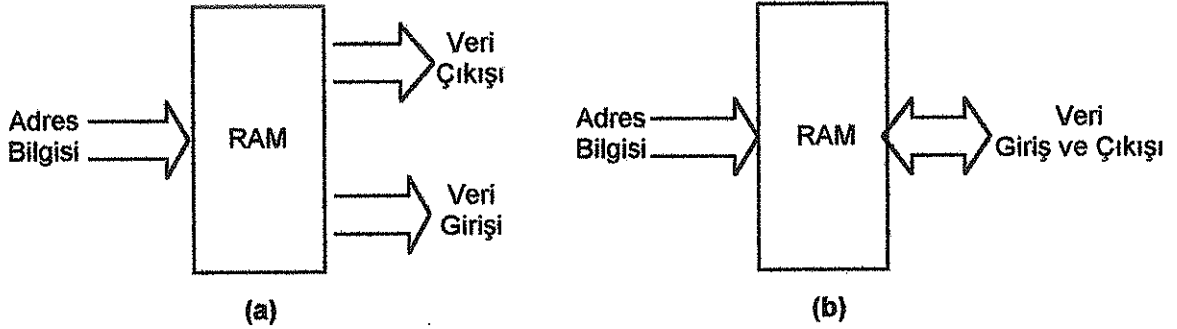
**ONDOKUZ MAYIS ÜNİVERSİTESİ
ELEKTRİK-ELEKTRONİK MÜHENDİSLİĞİ BÖLÜMÜ
SAYISAL ELEKTRONİK LABORATUVARI**

BELLEKLER VE PROGRAMLANABİLİR MANTIK YAPILAR (RAM)

Deneğin Amacı : Bu deneyde öncelikle bellekler hakkında genel bir bilgi edinilecek, RAM belleklerin akınması, yazılması ve adreslenmeleri ile ilgili uygulamalar yapılacaktır.

1. Açıklayıcı Bilgiler

RAM (Random Access Memory - Rastgele Erişilebilir Bellek) bellekler hem okumaya hem de yazılmaya elverişli belleklerdir. Veriler belleklere binary (ikili) işaretler olarak yazılır ya da okunur. Bir verinin belleğe yazılması veya okunabilmesi için söz konusu verinin bellekte kaydedileceği ya da okunacağı bellek alanının belirlenmesi gerekir. Bu adres bilgisi yardımıyla belleğin o alanı (location) okunabilir ya da yazılabilir. Şekil-1.a ve b ' de bir RAM belleğin giriş çıkış gösterimleri verilmiştir. Şekil-1.a' da gösterilen RAM bellekte giriş - çıkış veri yolu (Data bus) ayrıdır. Şekil-1.b' de gösterilen bellekte ise aynı veri yolunda hem giriş hemde çıkış yapılabilmektedir.



Şekil-1. RAM belleklerin bilgi giriş-çıkışları

Bir belleğin bellek kapasitesi sahip olduğu adres hatlarının ve data hatlarının sayısı ile belirlenir.

n : Adres hattı sayısı

m : Veri hattı sayısı olarak alınır ise, bir belleğin kapasitesi ;

Toplam bitlerin sayısı = $2^n \times m$ eşitliğinden elde edilir.

Örneğin 4-bit adres hattı ve 4-bit veri hattı olan bir belleğin toplam bitleri sayısı

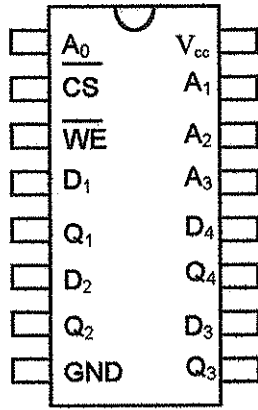
$$2^4 \times 4 = 64 \text{ bit olarak bulunur.}$$

Adres hattı sayısı 16-bit, data hattı sayısı 8-bit olan bir belleğin toplam bitleri sayısı ;

$$216 \times 8 = 64\ 000 \times 8 = 512\ 000 \text{ bit ya da } 64\text{Kbyte olarak bulunur.}$$

2. 74189 RAM Belleğinin Yapısı

74189 RAM belleğinin bacak bağlantıları, açıklamaları Şekil-2' de gösterilmiştir.



D_1, D_2, D_3, D_4 → Veri yolu (giriş)

Q_1, Q_2, Q_3, Q_4 → Veri yolu (çıkış)

A_0, A_1, A_2, A_3 → Adres Yolu

\overline{CS} : Aygıt yetkilendirme (chip select)

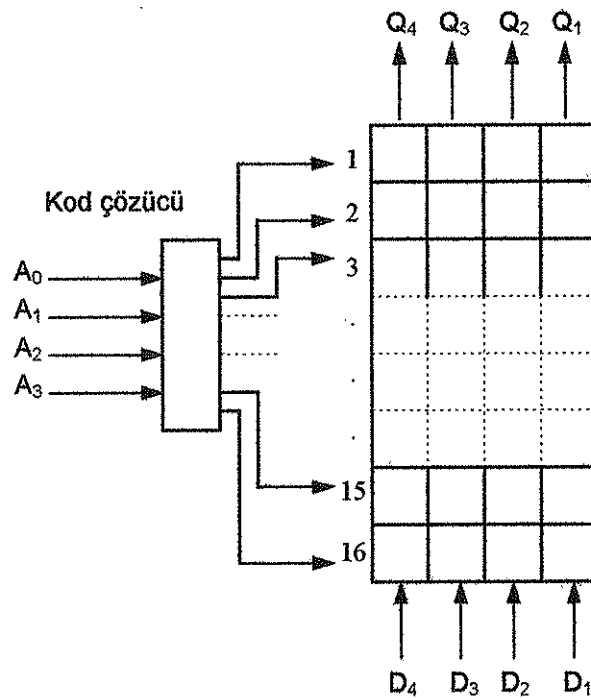
\overline{WE} : Yazma işlevi aktiveleme (write enable)

V_{CC} : + 5V (Besleme gerilimi)

GND : 0V (ground)

Şekil-2. 74189 RAM belleğinin bacak bağlantıları

Şekil-2' den de görüldüğü gibi 74189 4-bit adres ucuna sahiptir. Toplam bellek alanı 16 x 4 bittir. Şekil-3' de 74189 aygıtının bellek haritası gösterilmiştir.



Şekil-3. 74189 bellek haritası

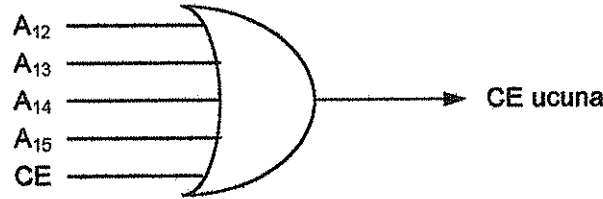
74189 RAM belleğinde bulunan kontrol işaretlerinin işlevleri Tablo-1' de verilmiştir.

\overline{CS}	\overline{WE}	İŞLEV	ÇIKIŞLARIN DURUMU
L	L	Yazma	Yüksek-Z
L	H	Okuma	Terslenmiş Çıkış
H	X	Seçilmemiş	Yüksek-Z

Tablo-1. 74189 RAM belleği kontrol işaretleri işlev tablosu

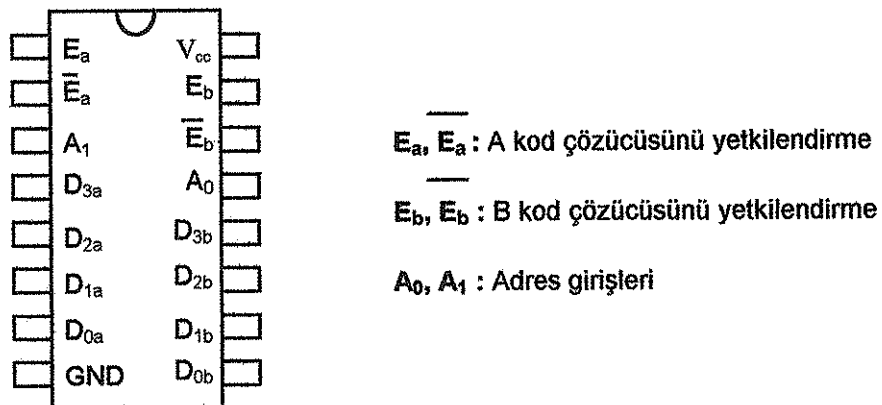
3. Adresleme :

Bellekler genellikle adreslenerek kullanılırlar. Yani belleğin aktif hale gelmesi için bir denetim işareti kullanılır. Bu denetim işareti aktifse bellekte aktif hale gelir. 74189 aygıtını aktif hale getirmek için kullanılan denetim işaretinin CE ucuna uygulanması gerekir. Bir belleğin toplam adres alanı içerisinde örnek olarak 0000-FFFF uzunluğunda bir bellek alanı içinde hangi aralıkta bulunacağı adres kod çözücülerle belirlenir. Adresleme işleminde genellikle adres hattının boşta kalan bit leri kullanılır. Örnek olarak ; eğer bir 74179 RAM belleğini 16-bit adres hattı bulunan bir sistemde XXXF adresine yerleştirmek için Şekil-4' deki donanım kullanılır.



Şekil-3. 74189 RAM belleğinin XXXF adresine yerleştirilmesi

Adresleme için genellikle bu iş için imal edilmiş kod çözücü entegre devreler kullanılır. Bu deneyde adresle için 74155 aygıtı kullanılacaktır. Bu aygıt iki'den dört'e bir kod çözücüdür. 74155' in içinde A ve B olmak üzere adres girişleri ortak fakat yetkilendirme uçları farklı iki adet kod çözücü vardır. Şekil-4' de bacak bağlantısı, Tablo-2' de ise giriş-çıkış fonksiyon işlevi gösterilmiştir.



Şekil-4. 74155 kod çözücüsünün bacak bağlantıları

A_0	A_1	E_x	\overline{E}_x	D_{0x}	D_{1x}	D_{2x}	D_{3x}
X	X	L	X	H	H	H	H
X	X	X	H	H	H	H	H
L	L	H	L	L	H	H	H
H	L	H	L	H	L	H	H
L	H	H	L	H	H	L	H
H	H	H	L	H	H	H	L

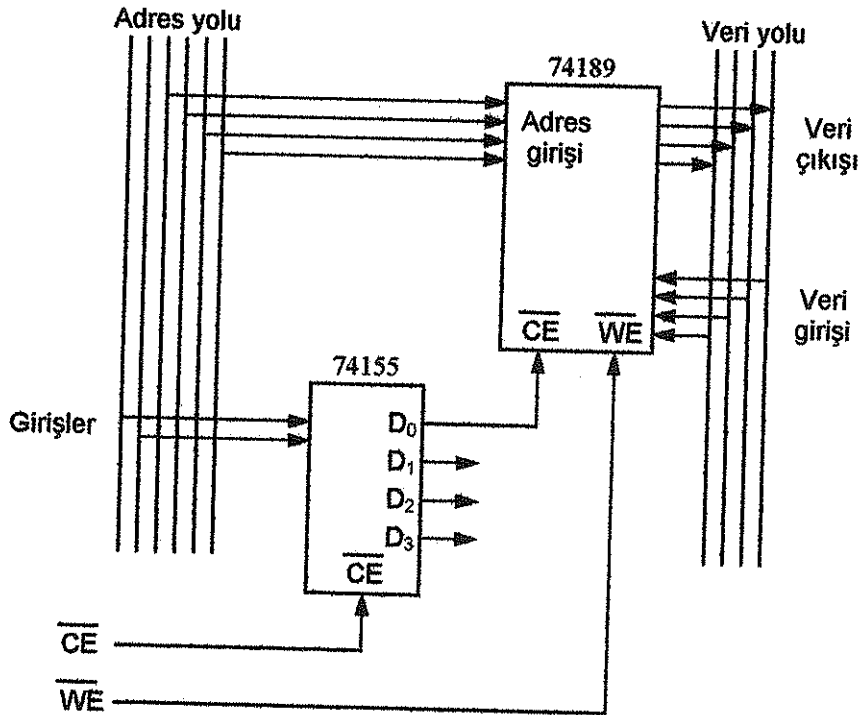
Tablo-2. 74155 kod çözücüsünün fonksiyon tablosu

x indisi ile a ya da b seçeneklerinden birisi belirtilmektedir.

4. 74189 RAM belleği kullanılarak Gerçeklenen Uygulamalar

4.1. 74189 RAM Belleğinin 4-bit'lik Veri Yoluna Bağlanması

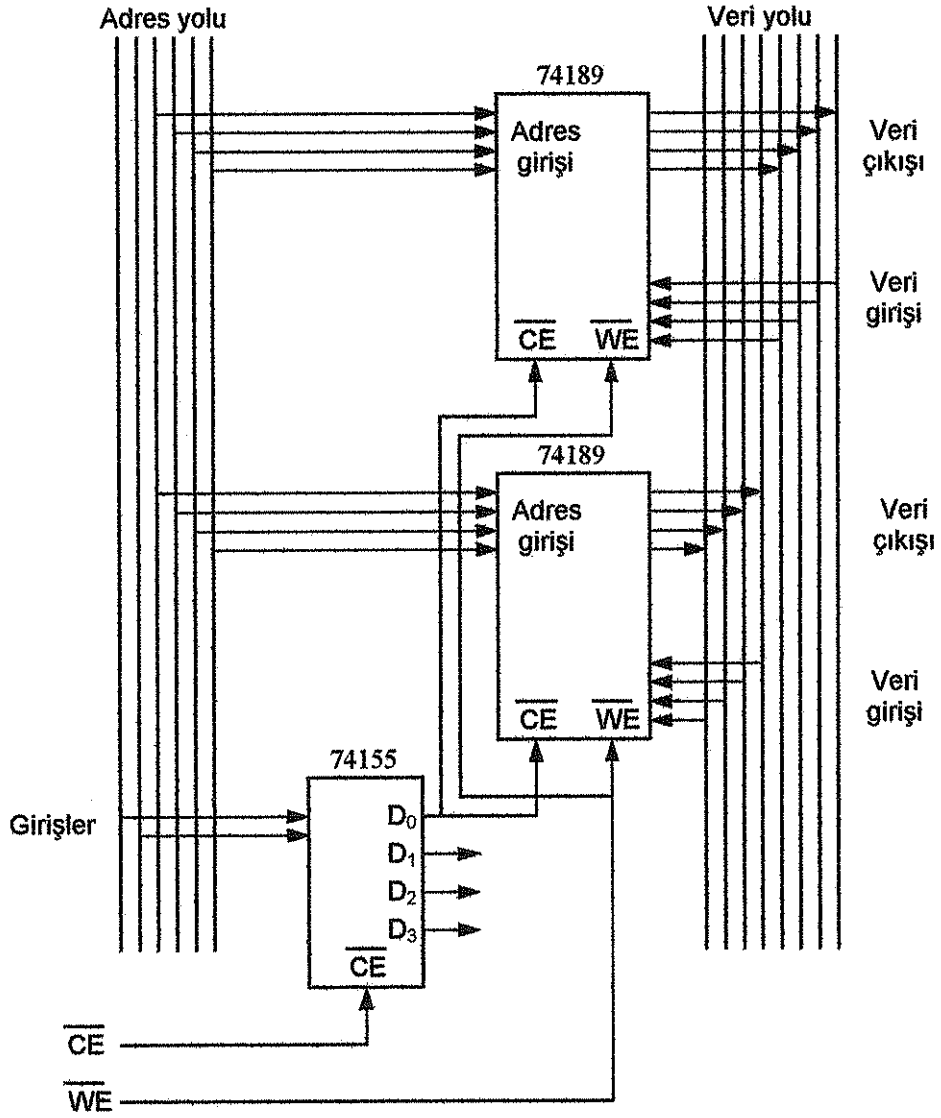
Şekil-5' de 74189 RAM belleğinin 74155 kod çözücü aygıtı yardımıyla 4-bit'lik veri yoluna bağlanması gösterilmiştir. 74155 aygıtının boşa kalan çıkışları diğer aygıtları yetkilendirmek için kullanılabilir.



Şekil-5. RAM belleğinin 4-bit'lik veri yoluna bağlanması

4.2. 74189 RAM bellekleri kullanılarak 8-bit' lik Veri Yolunun Elde Edilişi

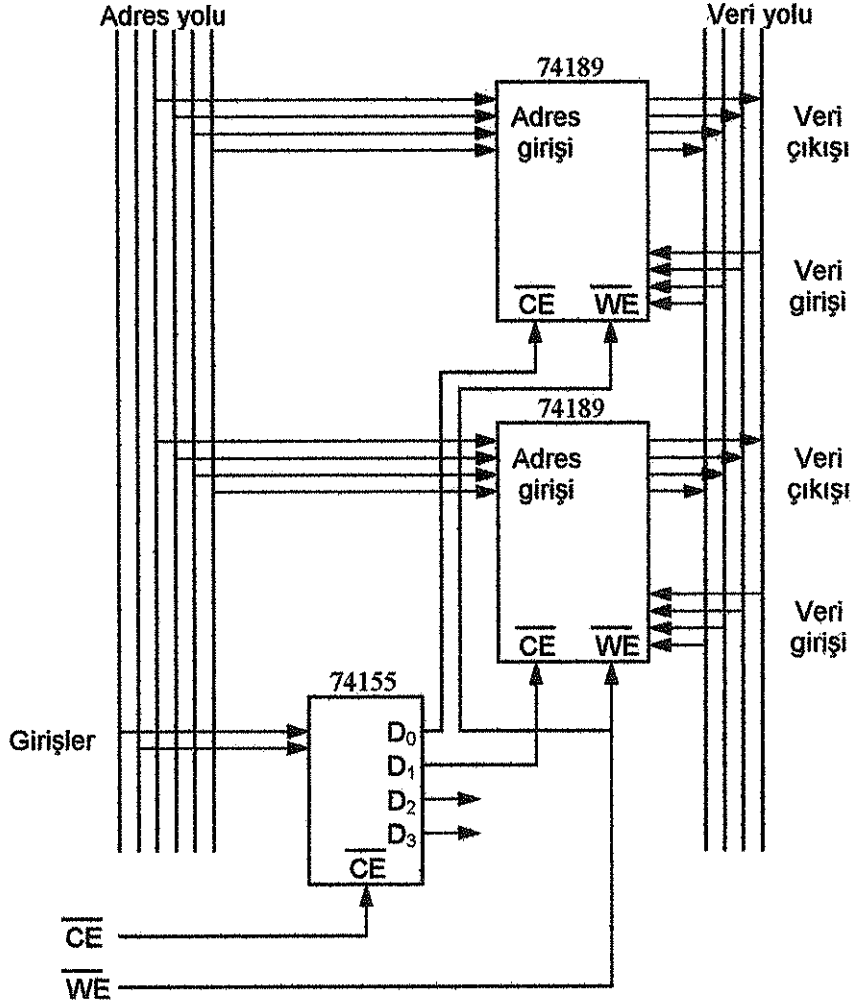
8-bit ' lik bir veri yolu elde edebilmek için, aygıt yetkilendirme uçları ve yazma yetkilendirme uçları ortak iki adet 74189 aygıtı kullanmak gereklidir. Bu devrenin bağlantısı Şekil-6' de gösterilmiştir.



Şekil-6. İki adet 74189 RAM kullanılarak 8-bit' lik veri yolunun elde edilmesi

4.3. Birden Fazla RAM Belleğin Aynı Veri Yoluna Bağlanması

Bir çok uygulamada, aynı veri yolu üzerinde birden fazla bellek bulunur. Bunlar RAM ya da EPROM vb. gibi bellekler olabilir. Burada yalnızca iki adet 74189 RAM belleğin 4-bitlik bir veri yoluna bağlanması incelenecektir. Şekil-7' de bu bağlantı gösterilmiştir.



Şekil-7. Birden fazla RAM belleğin aynı veri yoluna bağlanması

Şekilden de görüldüğü gibi her bir RAM bellek için ayrı bir yetkilendirme ucu kullanılmıştır. Bu sayede istenilen bellek aktif hale getirilebilmektedir. WE uçları ise ortaktır. Fakat ayrı ayrı da kontrol edilebilir.

5. Hazırlık Soruları

- 5.1. Belleklerin işlevleri, türleri ve kullanım alanları hakkında bilgi ediniz.
- 5.2. Bir belleğin kapasitesi nelere bağlıdır? Araştırınız.

6. Denevin yapılışı

- 6.1. Şekil-5' deki devreyi kurunuz. RAM belleğin herhangi bir adres alanına 4-bit uzunluğunda bir bilgi yazınız ve daha sonra yazdığınız bilgiyi okuyarak yaptığınız işlemin doğruluğunu kontrol ediniz.
- 6.2. Şekil-62 daki devreyi kurarak herhangi bir adrese 8-bit uzunluğunda bir bilgi yazınız. Yazdığınız bilgiyi okuyarak kontrol ediniz.
- 6.3. a) Şekil-7' deki devreyi kurunuz. Her iki belleğin aynı adres alanına 4-bit uzunluğunda veriler yazınız. Daha sonra yazdığınız verileri okuyarak kontrol ediniz.
b) Birinci belleğin herhangi bir bellek alanına 4-bit uzunluğunda bir bilgi yazınız. Daha sonra birinci belleği yazma, ikinci belleği okuma durumuna getirerek, veriyi ikinci belleğe aktarınız. İkinci belleği okuyarak işlemin doğruluğunu kontrol ediniz.

ELEKTRİK-ELEKTRONİK MÜHENDİSLİĞİ BÖLÜMÜ

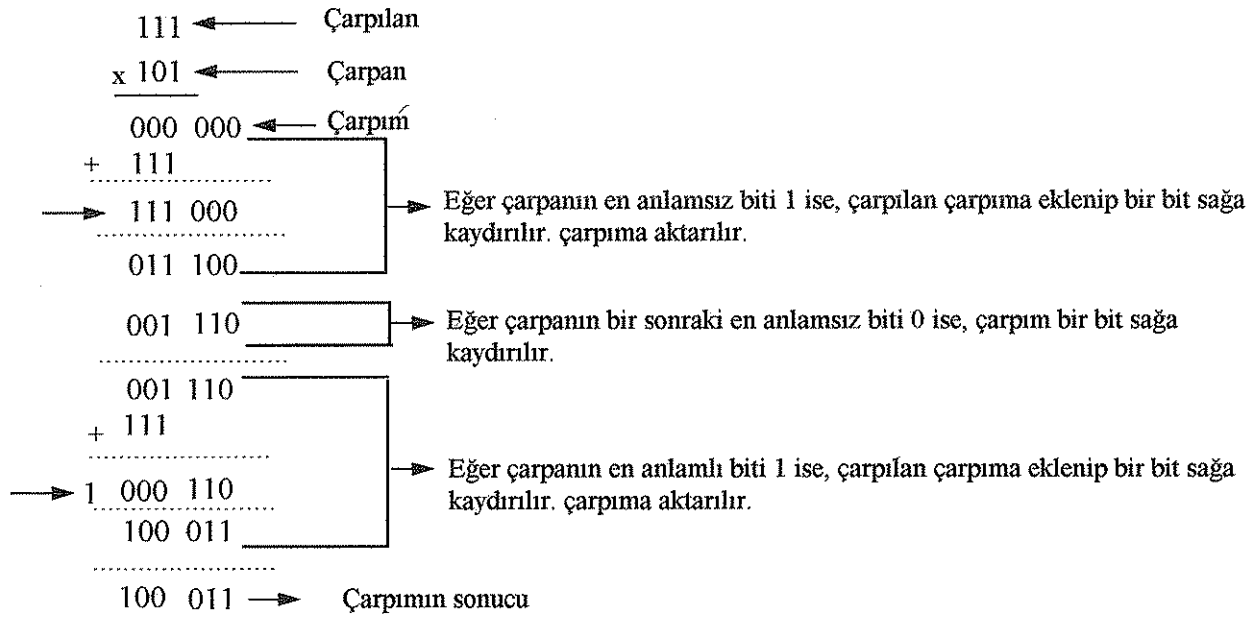
SAYISAL ELEKTRONİK LABORATUVARI

Deneyin Adı: İKİLİ ÇARPMA

Amaç: Bilgisayarlar yaptığı temel işlevler arasında en fazla zaman gerektiren işlemlerden biri de çarpmadır. Günümüzdeki teknolojik imkanlara göre değişik çarpma yöntemleri geliştirilmiştir. Bu çarpma işleminin nasıl gerçekleştiği aşağıda anlatılmaktadır.

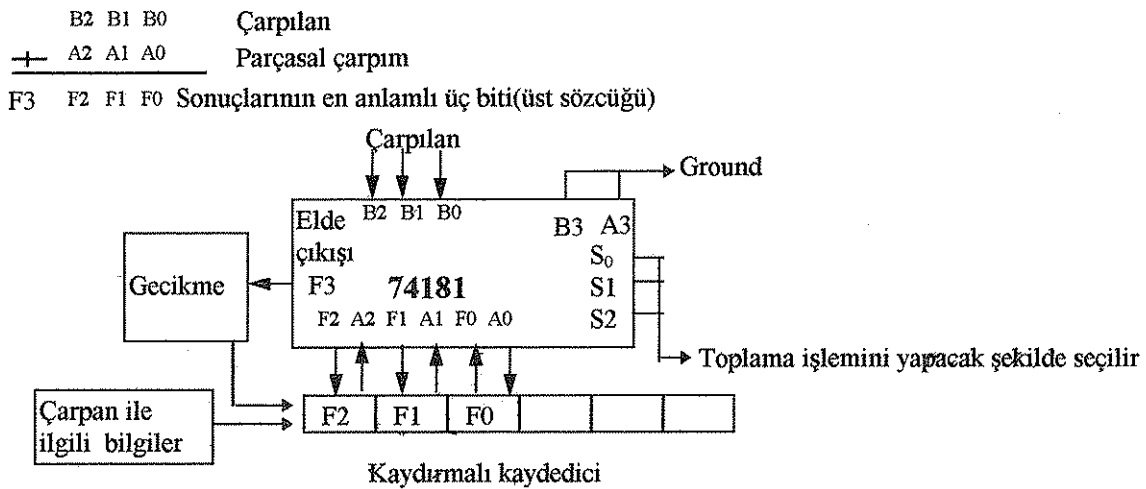
Çarpma işlemi

Üç bitlik iki tane iki sayının çarpımı aşağıdaki gibi yapılmaktadır.



Çarpıcının Öbek Çizgesi

Çarpan hiçbir yere kayıtlı değildir. Üç bitlik iki sayının çarpımın sonucu en çok 6 bit olacağından çarpımın ifade edilebilmesi için 6 bitlik yer ayrılmalıdır.



Şekil.1

Başlangıçta tüm kaydediciler sıfırlanır. Çarpılan sayı tam toplayıcının B0-B2 girişlerine uygulanır ve çarpanın en az anlamlı biti test edilir. Eğer çarpanın en az anlamlı biti 1 ise çarpılan çarpıma eklenip, elde edilen sonuç çarpıma yazılır ve çarpım bir bit sağa kaydırılır. Çarpanın bir sonraki en az anlamlı biti sıfırsa, çarpım sonucu bir bit sağa kaydırılır. Eğer çarpanın en anlamlı biti 1 ise çarpılan çarpıma eklenip, elde edilen sonuç çarpıma yazılır ve çarpım bir bit sağa kaydırılır.

Şekil.1' de öbek çizgesi verilen devrenin öğeler ile gerçekleştirilmesi şekil.2' de verilmiştir. Çarpan sayıya yer alan bitlerin değerlerine göre gerekli kaydırma ve toplama işlemleri, SW1, SW2, SW3 anahtarlarıyla sağlanmaktadır.

Veri anahtar uçları:

SW1: Saat(\uparrow) hem toplama hemde kaydırma işleminde kullanılmaktadır.

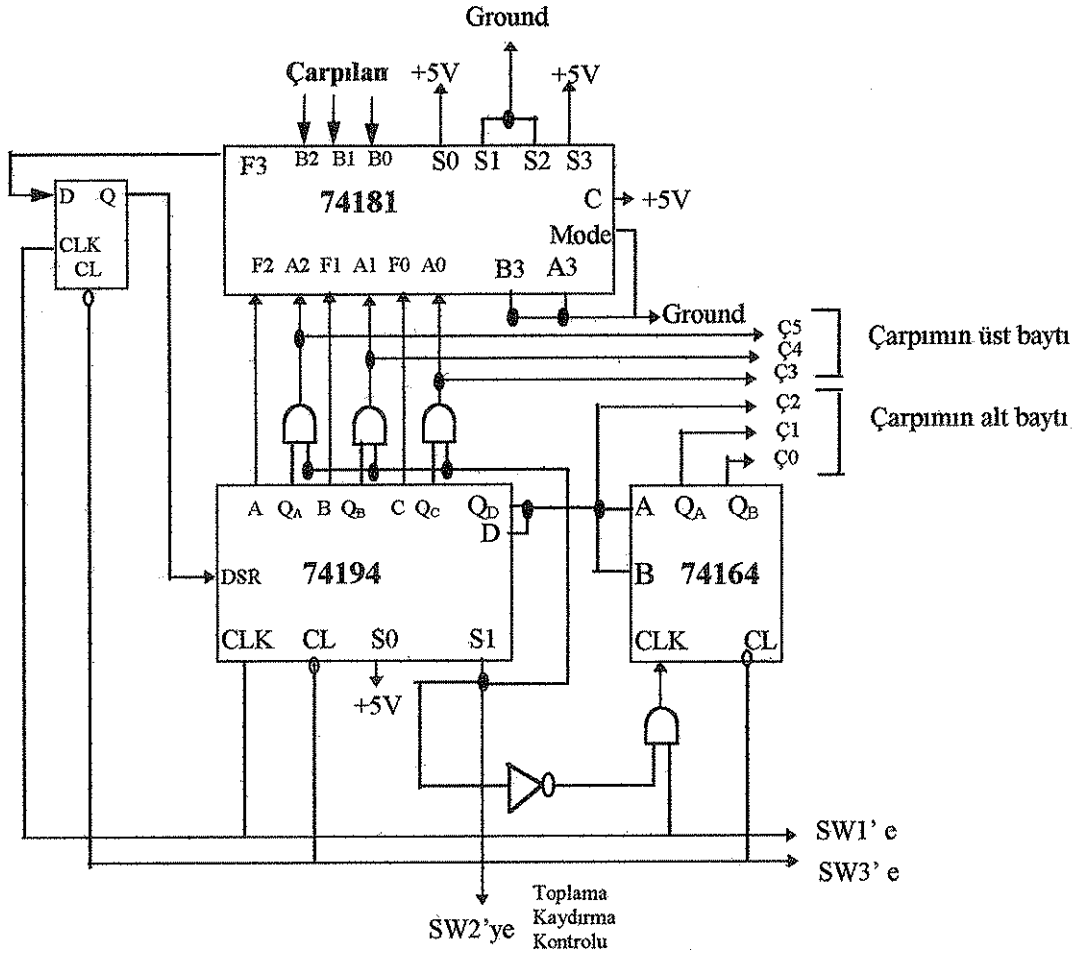
SW2 : Toplama/Sağa kaydırma kontrolü

SW2 mantık 1 = Toplama

SW2 mantık 0 = Sağa kaydırma

Burada SW2 anahtarı çarpanın biti olarak görev yapmaktadır. Eğer SW2 mantık 0, çarpanın o biti sıfır. SW2 mantık 1 ise çarpanın o biti mantık 1 anlamına gelmektedir.

SW3 : Clear(mantık 0)



Şekil.2.

İstenenler:

1. 74181(Alu),74164(8bit seri-giriş, paralel -çıkış kaydırmalı kaydedici), 74194(Universal kaydırmalı kaydedici) entegrelerine ilişkin bilgilerinizi tekrarlayınız ve gerekli kaynakları yanınızda bulundurunuz.

2.Şekil.2' deki gibi düzenek kullanılarak en çok kaç bitlik sayılarla çarpma yapılabilir. Bu duruma göre, gerekli değişiklikleri yaparak öbek çizgeyi çiziniz.

Deneyin Yapılışı:

1. 74181 4 bit aritmetik mantık birimi ile 3 bitlik iki bilgiyi toplayan devreyi kurunuz ve çeşitli değerler için toplamın sonucunu elde ile birlikte tablo halinde veriniz.(Deney düzeneğini bozmayınız.)

2. 74194 kaydırmalı kaydedici entegresini kullanarak yükleme ve daha sonra kaydırma işlemini gerçekleyip gözleyiniz. (Deney düzeneğini bozmayınız.)

3. 74164 8 bit seri giriş, paralel çıkış kaydırmalı kaydedici entegresini kullanarak yükleme ve daha sonra kaydırma işlemini gerçekleyip gözleyiniz. (Deney düzeneğini bozmayınız.)

4. Şekil-2' deki devreyi kurunuz.

5. 111×101 sayılarını çarpımı.

Çarpılan 111 olarak B2-B0 uçlarına uygulanır.Çarpanın en anlamsız biti 1 SW2 mantık 1 (topla)(clock), SW2 mantık 0 (kaydır)(clock).

Çarpanın ikinci biti sıfır olduğundan SW2 mantık 0 (kaydır)(clock).

Çarpanın en anlamlı biti 1 SW2 mantık 1 (topla)(clock), SW2 mantık 0 (kaydır)(clock).

Çarpım sonucunu görmek için SW2 mantık 1 seviyesine çekilir.(Neden?)

ONDOKUZ MAYIS ÜNİVERSİTESİ
Elektrik- Elektronik Müh. Bölümü
Sayısal Elektronik Lab.

Deney Adı: ARDIŞIL DİZGE TASARIMI - TRAFİK IŞIĞI DENETLEYİCİSİ

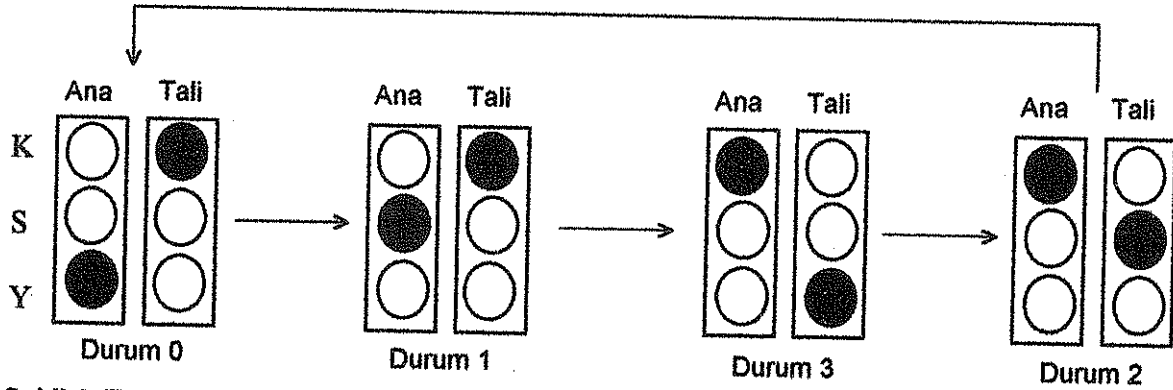
Amaç : Ardışıl mantık dizge örneği olarak trafik ışığı denetleyicisi tasarımının yapılması ve tek kararlı devrenin kullanımı.

1. Trafik ışığı denetleyicisi tasarımı:

Bir ana yol ve bu ana yola bağlı bir tali (yan) yolun trafik denetimini sağlamak için trafik ışığı denetleyicisi tasarımı yapılması istenmektedir.

Trafik ışığı denetleyicisinden istenenler:

Ana yoldaki yeşil ışık en az 25 s yanacak ve yan yolda bir taşıt olana kadar yanmaya devam edecek. Yan yoldaki yeşil ışık, yan yolda bir taşıt kalmayınca kadar veya en çok 25 s yanacak. Hem ana ve hem yan yoldaki yeşilden kırmızıya geçiş aralarındaki sarı uyarı ışığı 4 s yanacak. Bu olası 4 durum, tekrarlama sırasıyla şekil 1 de gösterilmiştir.



Şekil 1. Trafik ışıklarının istenen yanma sırasıyla verilmiş durumları.

Küçük bir sayıcı kullanılarak bu dört 'durum' un her biri bununla gösterilebilir.

Trafik ışığı denetleyicisinde, sayıcının durumları 3 giriş değişkeni ve 2 durum değişkeniyle belirlenir. Bu değişkenlerin bazı koşulları sağlaması halinde sayıcı, sonraki (gelecek) duruma ilerler.(veya aynı durumda kalır). Bu üç değişken şunlardır:

V_S : Yan yoldaki araç algılayıcısı
 T_L : Uzun zamanlayıcı (25 s)
 T_S : Kısa zamanlayıcı (4 s)

Ana ve yan yoldaki trafik ışıklarının durumları sıra ile aşağıdaki gibi verilebilir :

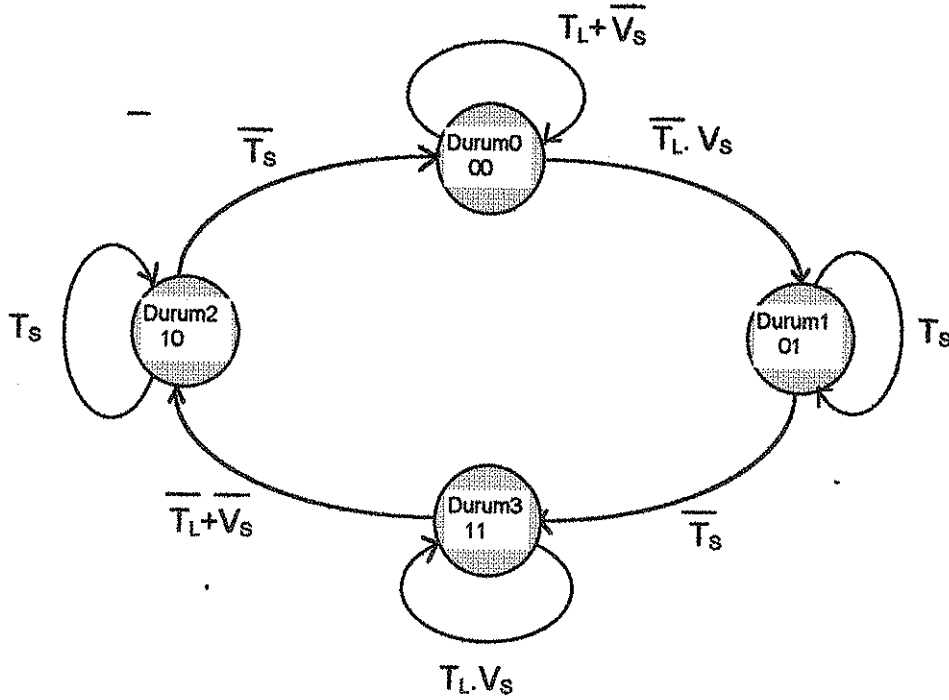
Durum 0: Sayıcı 00 durumunda . Eğer uzun zamanlayıcı (T_L)'on' ise veya yan sokak üzerinde hiçbir araç (V_S) yoksa, 00 durumunda kalacak. Eğer uzun zamanlayıcı (T_L) 'off' ise ve yan sokakta bir araba varsa (V_S) 01 durumuna gidecek.

Durum 1: Sayıcı 01 durumundadır. Eğer kısa zamanlayıcı (T_S) 'on' ise 01 durumunda kalacak. T_S 'off' ise 11 durumuna gidecek.

Durum 3: Sayıcı 11 durumundadır. Eğer T_L 'on' ise ve yan cadde üzerinde bir araç varsa, 11 durumunda kalacak. T_L 'off' ise veya yan sokakta hiçbir taşıt yoksa 10 durumuna gidecek.

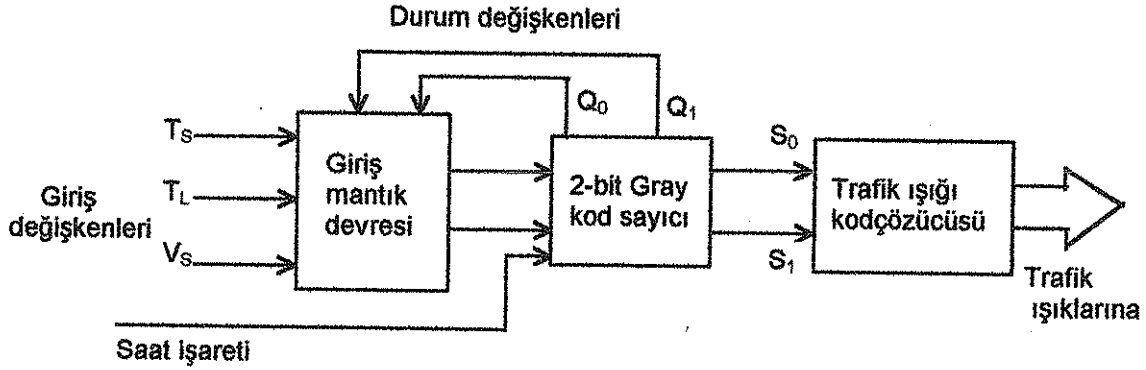
Durum 2: Sayıcı 10 durumundadır. T_S 'on' ise 10 durumunda kalacak. T_S 'off' ise 00 durumuna gidecek.

Durum geçişlerini gösteren durum çizgesi şekil 2 de verilmiştir.



Şekil 2. Durum çizgesi (diyagramı).

Şekil 3 deki öbek çizgeyle ardışıl mantık dizge tanımı yapılabilir. Giriş mantık öbeği, üç giriş değişkeni (V_s, T_L ve T_s) nin flip-floplara geçişini sağlamak için iki tane data seçicisi içerir. Data seçiciler (MUX1 ve MUX2) 74153 tümdevresidir. Sayıcının flip-flop çıkışları, seçici girişlere bağlı olduğundan seçilecek girişi 'şimdiki durum' belirler (I_0-I_3 arasında). Şimdiki durum gelecek durum çizelgesi çizelge 1 de verilmiştir.



Şekil 3. Ardışıl mantık dizge öbek çizgesi.

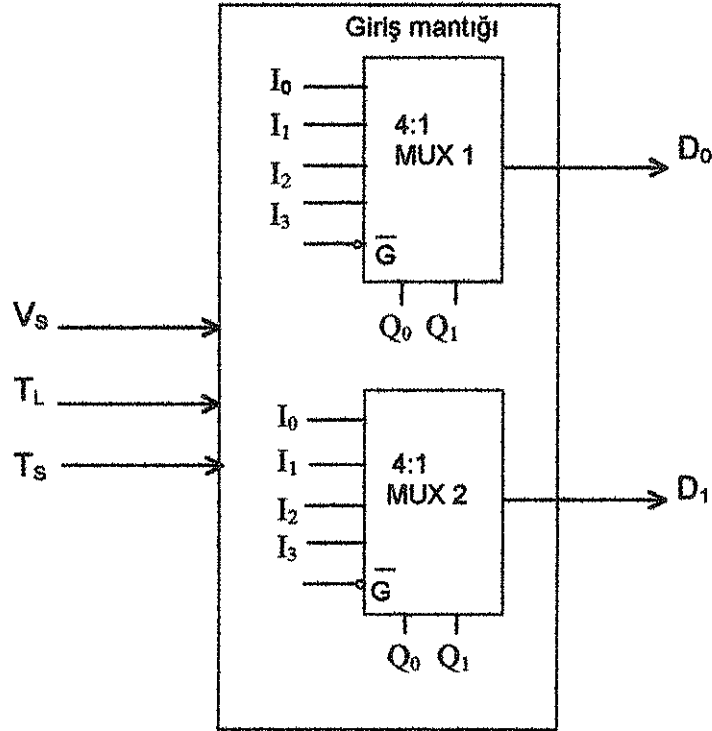
Çizelge 1.

Şimdiki durum		Gelecek durum		Giriş şartları
Q_0	Q_1	Q_0	Q_1	
0	0	0	0	$T_L + \overline{V_s}$
0	0	0	1	$\overline{T_L} V_s$
0	1	0	1	T_s
0	1	1	1	$\overline{T_s}$
1	1	1	1	$T_L V_s$
1	1	1	0	$\overline{T_L} + \overline{V_s}$
1	0	1	0	T_s
1	0	0	0	$\overline{T_s}$

Çizelge 1 deki tablo incelenirse, her satır çiftinde; sayıcı, varsayılan iki olası durumu gösterir. Örneğin ilk satır çiftinde sayıcı 00 durumundadır; 'uzun zamanlayıcı ve araç algılayıcı' girişlere bağlı olarak ya 00 durumunda kalacak veya 01 durumuna gidecektir.

MUX1 ve MUX2 nin çıkış uçları D f/f ların girişleri olduklarından, f/f çıkışlarının gelecek durumları her saat darbesinde girişler ile aynı olacaktır. Ve durumların geçiş sıraları hep 00-01-11-10-00... biçiminde isteğe uygun olacaktır.f/f çıkışları bir durum kodçözücüsünden geçip istenen biçimde trafik lambalarının yanmasını sağlanacaktır.

1.1) Giriş mantık devresi



Şekil 4. Giriş mantık devresi.

Şimdiki durum - gelecek durum çizelgesinden görüleceği gibi durumlar arası geçişlerde yalnızca bir durum değişkeni değişmekte, diğeri aynı kalmaktadır. Bu, 'giriş mantığı'ndaki öğelerin nasıl seçileceğini de belirleyecektir. Örneğin $Q_0Q_1 = 00$ dan $Q_0Q_1 = 01$ geçişinde belirleyici girişler neler olmalı? Q_0 daima 'sıfır' kalmaktadır, Q_1 ise $0 \rightarrow 1$ geçişi yapmaktadır. Bu geçişi sağlayan koşul $\bar{T}_L \cdot V_S$ nin 'mantık 1' olması koşuldur. O halde, giriş mantığında yer alan MUX ların seçici girişleri Q_0 ve Q_1 olmalı, MUX1 in I_0 girişi '0', MUX2 nin I_0 girişi ise $\bar{T}_L \cdot V_S$ olmalıdır. Bu koşul olmazsa durum değişmeyecek (00 kalacak), koşul olursa 01 durumuna geçilecektir.

$Q_0Q_1 = 01$ den $Q_0Q_1 = 11$ geçişinde : Q_1 daima '1' kalmakta, Q_0 ise $\bar{T}_S = 1$ koşulu ile $0 \rightarrow 1$ geçişi yapmaktadır. Öyleyse MUX1 in I_1 girişi \bar{T}_S , MUX2 nin I_1 girişi '1' olmalıdır.

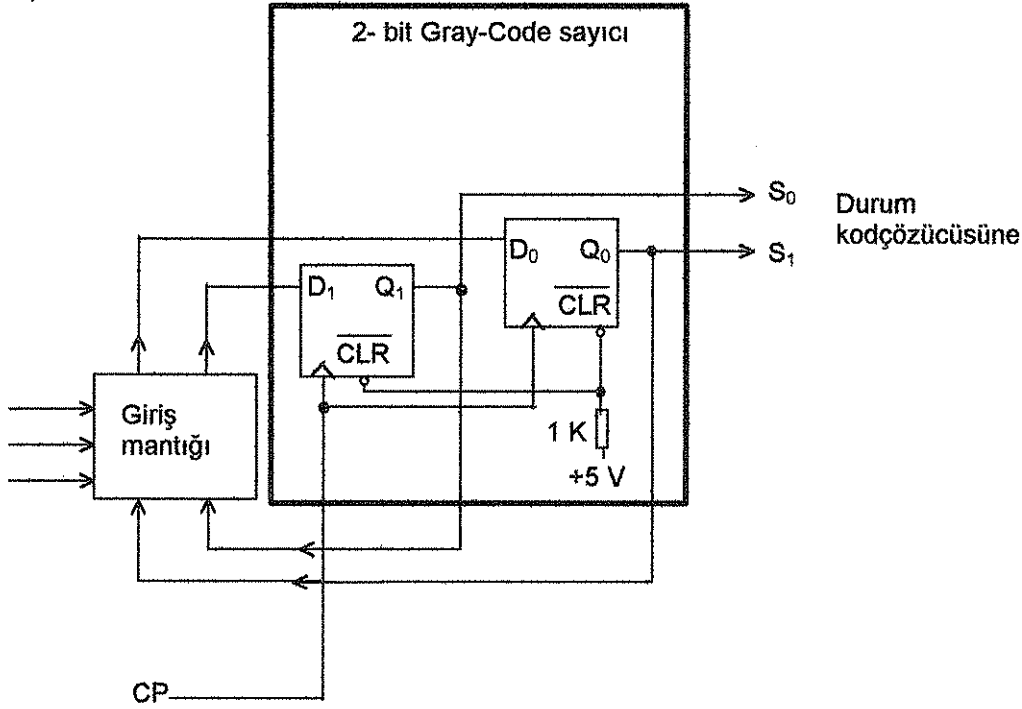
$Q_0Q_1 = 11$ den $Q_0Q_1 = 10$ geçişinde : Q_0 daima '1' kalmakta, Q_1 ise $\bar{T}_L + \bar{V}_S = 0$ koşulu ile $1 \rightarrow 0$ geçişi yapmaktadır. Öyleyse MUX1 in I_3 girişi '1', MUX2 nin I_3 girişi $\bar{T}_L \cdot V_S$ olmalıdır.

$Q_0Q_1 = 10$ dan $Q_0Q_1 = 00$ geçişinde : Q_1 daima '0' kalmakta, Q_0 ise $\bar{T}_S = 1$ koşulu ile $1 \rightarrow 0$ geçişi yapmaktadır. Öyleyse MUX1 in I_2 girişi \bar{T}_S olmalı, MUX2 nin I_2 girişi '0' olmalıdır.

1.2) 2-bit Gray - Code Sayıcı

Giriş mantık devresi çıkışları Gray -Code sayıcının girişlerini oluşturur. İki tane D f/f tan oluşan eşzamanlı (senkron) sayıcıdır. Çıkışları 'durumlar'dır. Bunlar durum kodçözücüsüne gider ve

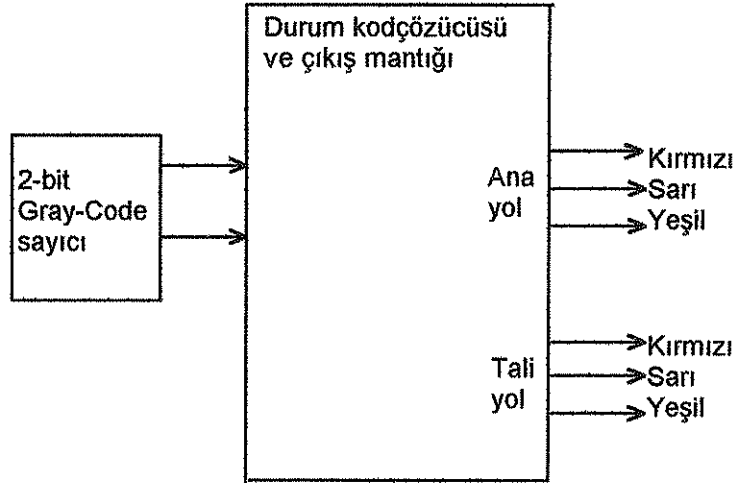
aynı zamanda gelecek durumun belirlenmesi için giriş mantık devresinin girişlerini oluştururlar(şekil 5 te).



Şekil 5.

1.3)Trafik Işığı Kodçözücüsü

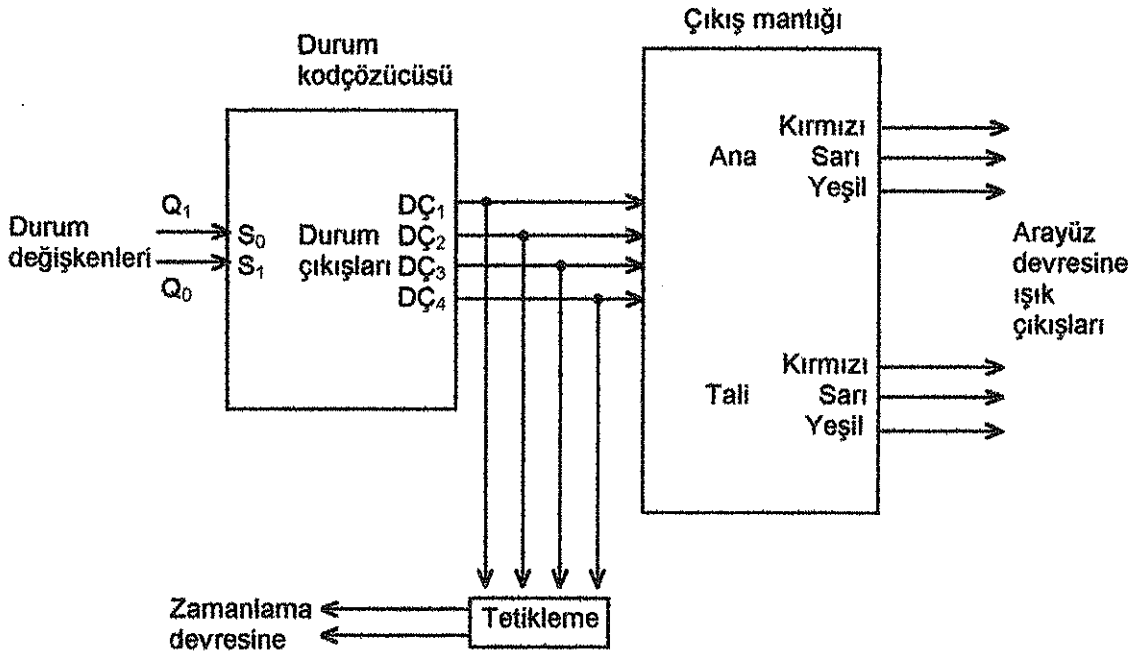
Durum geçişleri istenen sıra ile, 2-bit gray-code sayıcıda sağlandıktan sonra, durum değişkenleri ile trafik ışıkları arasındaki geçişi sağlayan trafik ışığı kodçözücüsüne gerek vardır (şekil6).



Şekil 6.

Devre, dört durum için ve trafik ışığını denetleyen dizgenin çıkış mantık devresi için durum kodçözücüsü görevi yapmaktadır. Ana ve tali yola ilişkin birer kırmızı, sarı ve yeşil trafik ışıklarını gösteren altı tane çıkış vardır. Işıkların istenen sırayla yanmasını sağlayan çıkışlar çizelge 2 deki doğruluk çizelgesinde gösterilmiştir. Mantık '0' , LED'i 'on' konumuna getirmektedir (yakmaktadır). Örneğin; 00 durumu (doğruluk çizelgesindeki ilk satır), ana yoldaki yeşil ışığı 'on', tali yoldaki kırmızı ışığı 'on' yapacaktır.

Önce bir durum kodçözücüsü kullanmak gerekir. Böylece iki durum değişkeniyle belirlenecek 4 durum ayrı ayrı değerlendirilebilecektir. Bu 4 çıkışla uyarılan çıkış mantık devresi kullanılacaktır. Çıkış mantık devresi çıkışları trafik ışıkları aradevresine gidecektir (şekil 7).



Şekil 7.

Çizelge 2.

Durum kodu		Durum çıkışları				Lamba çıkışları					
S ₁	S ₀	DÇ ₁	DÇ ₂	DÇ ₃	DÇ ₄	AK	AS	AY	TK	TS	TY
0	0	0	1	1	1	1	1	0	0	1	1
0	1	1	0	1	1	1	0	1	0	1	1
1	0	1	1	0	1	0	1	1	1	1	0
1	1	1	1	1	0	0	1	1	1	0	1

DÇ: Durum çıkışı, AK: Ana yoldaki kırmızı, TK: Tali yoldaki kırmızı.

1.4) Zamanlayıcı Devre

Kısa ve uzun zamanlayıcı devreler, birer monostable devrelerdir.

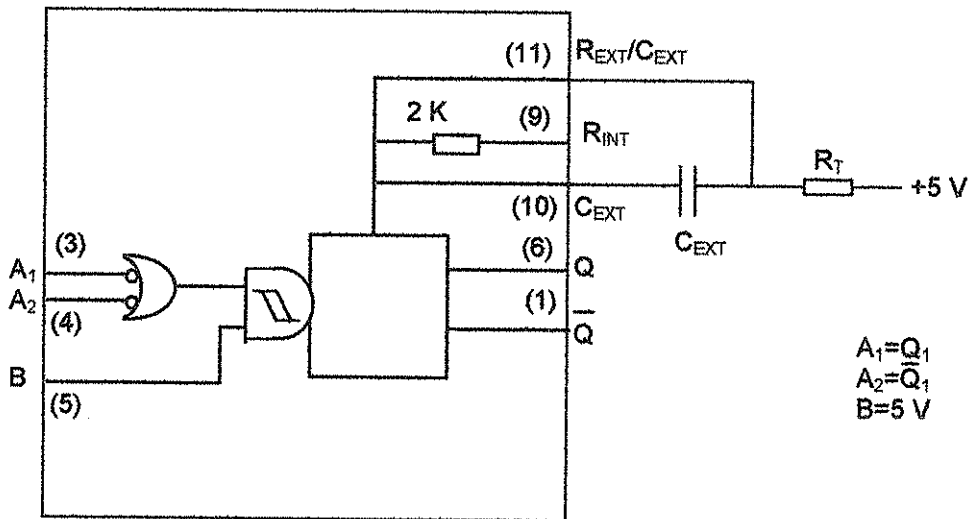
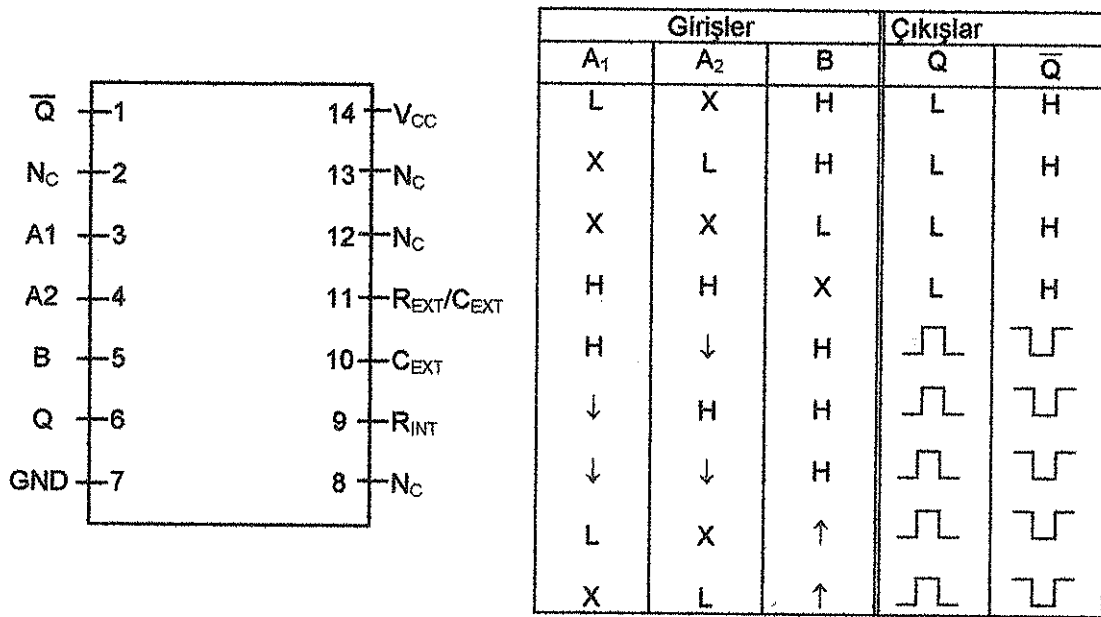
74121 kullanarak monostable multivibratör yapımı

Kısa zamanlayıcı 74121 tümdevresiyle gerçekleştirilecektir. Tablo 1 den görüldüğü gibi gelecek durum tablosundan, kısa zamanlayıcının durum 1 ve 2 de 'on' olduğu görülmektedir. Gelecek durum tablosundan ayrıca, kısa zamanlayıcının tetiklenmesinin, sayıcının Q₁ çıkışı değiştiği da görülür (hem yükselen hem düşen kenarda). 74121 in işlev tablosuna bakılırsa (şekil 8) hem yükselen hem düşen kenar tetiklemesi ek bir mantık kullanmaksızın aynı anda kullanılamaz. Bununla birlikte A₁ ve A₂ girişlerinin herbiri düşen-kenar tetiklemeyi kabul edebilirler. Flip-floplardan her iki geçiş için bir düşen kenar tetiklemesi elde etmek için, basit bir hile gerekir: \bar{Q} çıkışını alarak yükselen kenarı ters çevirmek. Bu, şekil 8 te gösterilmiştir.

74121, 2 kohmluk iç zamanlama direnci içerir. R_{INT} i V_{CC} ye bağlayarak, zamanlama direnci yerine iç direnci seçilebilir (2 kohmu) veya bir dış direnç seçilebilir. Dış zamanlama direnci kullanmak için R_{INT} ucu (9 nolu uç) açık bırakılır (C_{EXT}, R_{EXT} seçilmiş olur). Sığaç dış elemandır ama çok kısa darbeler için yok edilebilir (kaldırılabilir).

$$t_w = 0.7 C_{EXT} R_T$$

eşitliği, yaklaşık darbe genişliği t_w yi verir. Buradan R_T: uygun zamanlama direncidir (ya iç ya dış direnç seçimiyle), C_{EXT} pF cinsinden, R_T kohm cinsinden, t_w ns cinsindedir.



Şekil 8. 74121 (Schmitt Trigger girişli)

(R_{INT} ile 35 ns tipik t_w, R_{EXT} / C_{EXT} ile 40ns - 28s arası t_w elde edilebilir).

2. Deneyin Yapılışı:

1. Şekil 8 deki kısa zamanlayıcı (monostable) devreyi kurunuz (R ve C değerlerini önceden hesaplayınız). Q₁ ve \bar{Q}_1 girişlerini dışarıdan darbe vererek uyarınız. Çıkışı (T_S) osiloskopta gözleyerek, kısa darbe uzunluğu (t_w) nun istenen değerde olup olmadığını denetleyiniz.

2. Uzun zamanlayıcı (T_L) ve araç algılayıcı (V_S) devreler yerine birer anahtar kullanınız. Kısa zamanlayıcı devre çıkışı T_S olarak kullanınız ve şekil 3 de öbek çizgesi verilen devreyi kurunuz. CP için 10 kHz lik pulse generatör kullanınız. Çıkışta arayüz devresi kullanmadan, trafik lambaları yerine LED ler kullanınız.

3. Şekil 2 deki durum çizgesi, devrenin ardışıl durumları için gerekli girişleri belirlemede yol gösterecektir. Önce uzun zamanlayıcıyı 'HIGH', araç algılayıcıyı 'LOW' konuma getiriniz (T_L=1, V_S=0). Clear girişini kullanarak sayıcıyı 00 durumuna getiriniz. Bu, başlangıç durumudur. Bu sırada kısa zamanlayıcı çıkışı 'LOW' dur (sayıcı çıkışından dolayı), tali yolda bir araç olmadığı

varsayılmaktadır ($V_S = 0$ alınmıştı). 00 durumu devam eder. Tali yolda bir araç olduğunu varsayınız, yani $V_S = \text{'HIGH'}$ alınız. Uzun zamanlayıcı henüz 25 s yi doldurmadığı için durum değişmeyecektir. 25 s geçtiğini varsayarak uzun zamanlayıcı anahtarı kapatınız ($T_L = 0$). Devre hemen 01 durumuna geçecek ve 4 s lik kısa zamanlayıcı devreye girecek. Kısa zamanlayıcı 'ON' iken, uzun zamanlayıcı anahtarı tekrar açın ($T_L = 1$). Devre 4 s sonra kendi kendine 11 durumuna geçecektir. Bu durumları zamanlama çizgesi (timing diagram) çizerek; CP, T_L , T_S , V_S değişkenleri için gösteriniz.

4.3 durumunda (11 durumu) kalmak için hangi koşullar gerekir? 11 durumundan tekrar 00 durumuna geçmek için hangi koşullar gerekir? Bu koşulların tümünü sağlayarak devre çıkışını gözleyiniz. Zamanlama çizgesinde de çizerek gösteriniz.

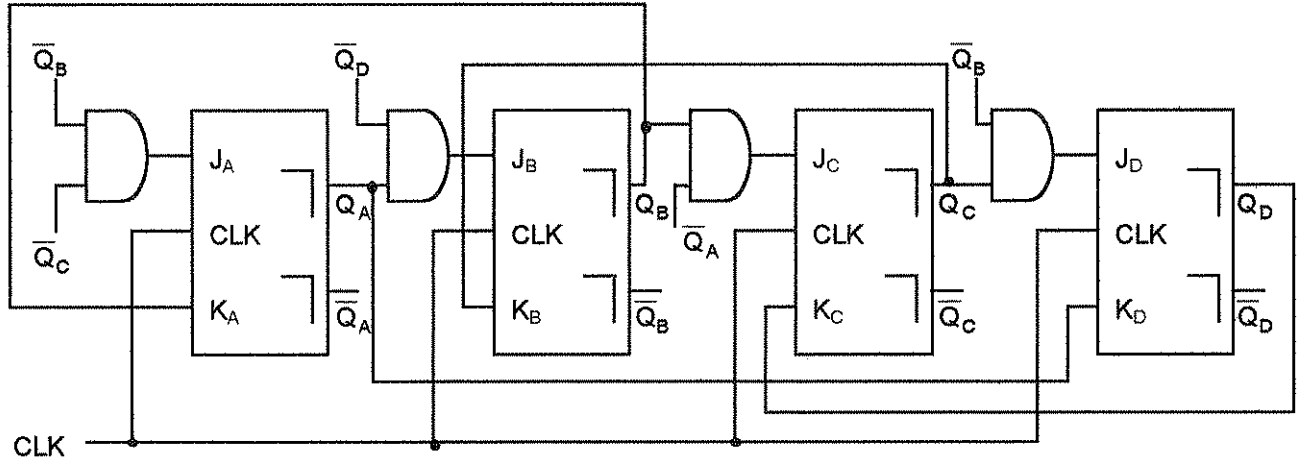
ONDOKUZ MAYIS ÜNİVERSİTESİ
ELEKTRİK-ELEKTRONİK MÜHENDİSLİĞİ BÖLÜMÜ
SAYISAL ELEKTRONİK LABORATUVARI

EŞZAMANLI (SENKRON) SAYICILARIN KOD ÇÖZÜMÜ YÖNTEMİYLE ANALİZİ

Giriş:

Eşzamanlı sayıcıların, tüm flip-flopların aynı zamanda durum değiştirmesini sağlayan ortak saat girişi bağlantıları vardır. Bu yüzden, saat darbesinden bir sonraki sayıya kadar geçecek zaman ripple sayıcılara göre çok daha hızlıdır. Bu hız, çıkıştaki eşzamanlı olmayan durum geçişlerinden kaynaklanan istenmeyen işaretleri azaltır. Bununla birlikte, istenmeyen işaretler her zaman yok edilemez. Bunu önlemenin bir yolu, gray kodu sayımlı ardışıklıktır yani her saat darbesinde sadece bir flip-flopun durum değiştirmesidir.

Kod çözümü, özel bir sayının kontrolüdür. Kodu çözülmüş çıkış, iş üreten bir mantığı gerçekleştirmek için kullanılır. Kod çözücülü sayıcılar, düzensiz sayım üretmek için de kullanılırlar.



Şekil 1

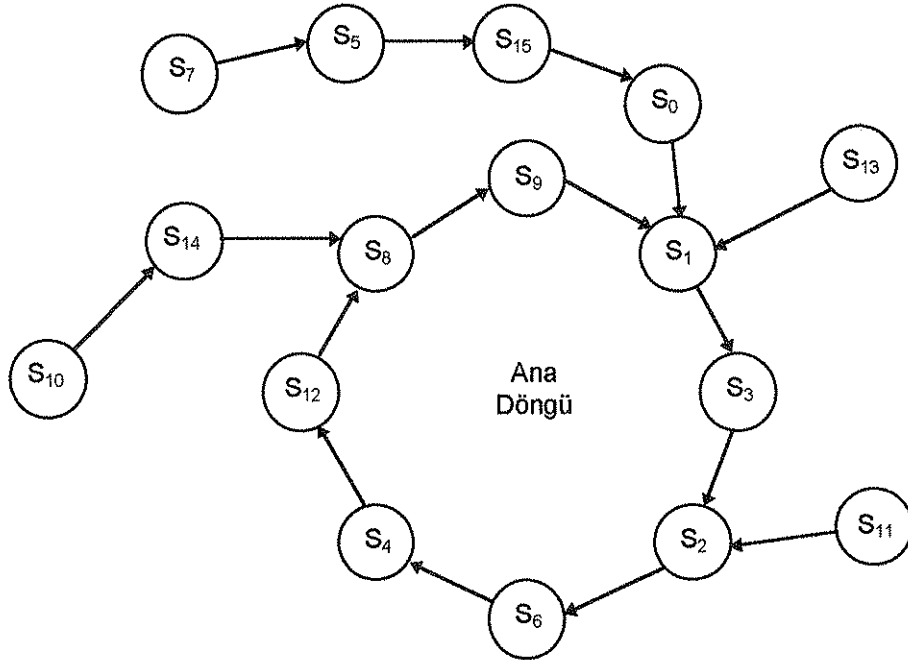
Eşzamanlı Sayıcıların Analizi:

Eşzamanlı sayıcıların analiz yöntemlerinden bir tanesi de durum geçiş çizelgesi yöntemidir. Şekil 1'de gösterilen sayıcıyı ele alacak olursak, Tablo 1'de sayıcıdaki tüm flip-flopların durumları 0000_2 başlangıç değerinden itibaren 2^N tane olacak biçimde (N: bit sayısı) sırasıyla incelenmiştir. Tüm Q çıkışları için o anki J ve K giriş değerleri hesaplanmış ve tabloya yerleştirilmiştir. Ancak bu girişlerin çıkışa yansması için mutlaka saat darbesinin alınması gerekir. Elimizdeki sayıcının 0000_2 durumu için, D, C, ve B girişleri (hem J hem de K) 0'dır. A girişinde ise $J = 1$, K ise 0'dır. Yani bir sonraki saat darbesinde A flip-flop'u set konumuna ($Q_A = 1$) gelecektir. O halde bir sonraki durum 0001_2 olacaktır.

Yukarıdaki işlem tüm (2^N) durum için tekrarlanmalıdır. Bu durumda ana döngünün dışında kalan durumlar için de hesaplamalar yapılmalıdır. Çizelge 1'den yola çıkarak, Şekil 2'deki durum geçiş çizelgesi hazırlanır.

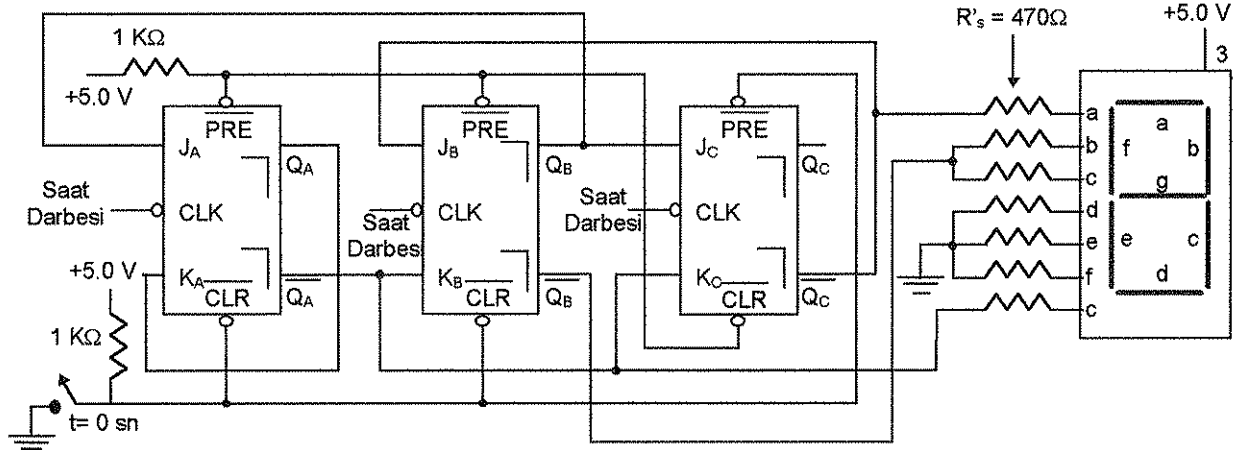
Çizelge 1

Çıkışlar				Girişler								
Q_D	Q_C	Q_B	Q_A	$J_D = \bar{Q}_B \cdot Q_C$	$K_D = Q_A$	$J_C = \bar{Q}_A \cdot Q_B$	$K_C = Q_D$	$J_B = Q_A \cdot \bar{Q}_D$	$K_B = Q_C$	$J_A = \bar{Q}_B \cdot \bar{Q}_C$	$K_A = Q_B$	
0	0	0	0	0	0	0	0	0	0	1	0	
0	0	0	1	0	1	0	0	1	0	1	0	
0	0	1	1	0	1	0	0	1	0	0	1	
0	0	1	0	0	0	1	0	0	0	0	1	
0	1	1	0	0	0	1	0	0	1	0	1	
0	1	0	0	1	0	0	0	0	1	0	0	
1	1	0	0	1	0	0	1	0	1	0	0	
1	0	0	0	0	0	0	1	0	0	1	0	
1	0	0	1	0	1	0	1	0	0	1	0	
0	0	0	1	Bu adımdan itibaren yukarıdaki biçim tekrarlanır.								
1	1	0	1	1	1	0	1	0	1	0	0	
0	0	0	1	Ana döngüye girer.								
0	1	0	1	1	1	0	0	1	1	0	0	
1	1	1	1	0	1	0	1	0	1	0	1	
0	0	0	0	(0000) ile başlayan duruma döner.								
0	1	1	1	0	1	0	0	1	1	0	1	
0	1	0	1	(0101) ile başlayan duruma döner.								
1	0	1	0	0	0	1	1	0	0	0	1	
1	1	1	0	0	0	1	1	0	1	0	1	
1	0	0	0	Ana döngüye girer.								
1	0	1	1	0	1	1	1	0	0	0	1	
0	0	1	0	Ana döngüye girer.								



Hazırlık Soruları:

Şekil.3'te gösterilen devrenin LED göstergelerinde üreteceği çıkış nedir? Durum geçiş çizelgesini çıkararak 7-segment display'de 8sn süresinde neler gözleneceğini bulunuz. (Saat darbesi: 1 Hz, anahtar t=0'dan itibaren açık)



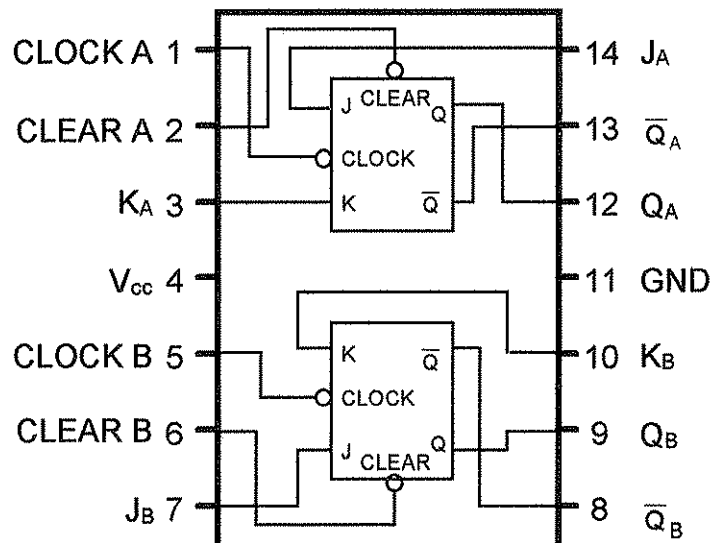
Şekil 3

Deneyin Yapılışı:

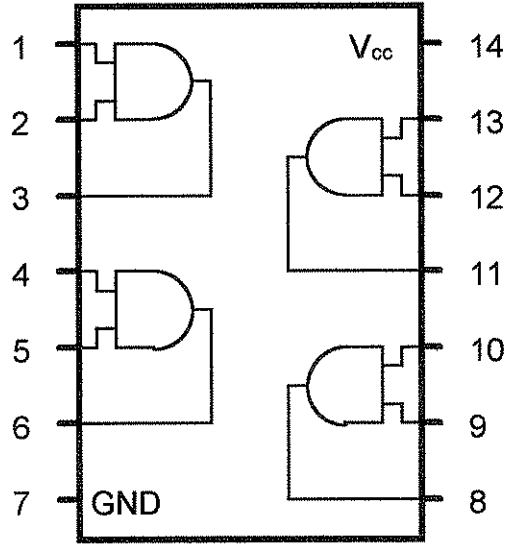
1. Şekil 1'de verilen sayıcı devresini kurunuz. J-K flip-flopları için 7473, VE kapıları için 7408 tümdevreleri kullanılacaktır. Bacak bağlantıları Şekil 4 ve Şekil 5'te verilmiştir.
2. Flip-flop'ların CLEAR girişlerini 0 yaparak sayıcıyı (0000) konumuna getiriniz. Daha sonra saat darbesi uygulayarak çıkışı gözleyiniz.
3. J ve K girişlerine döngünün dışına itici girişler uygulayarak döngü dışı çıkışlar elde ediniz ve saat darbesi uygulayarak çıkıştaki durumları gözleyiniz.
4. Şekil 4'teki devreyi kurarak çalıştırınız. J-K flip-flopları için 7476 tümdevresi kullanılacaktır. Bacak bağlantıları Şekil 6'da verilmiştir.

Raporda İstenenler:

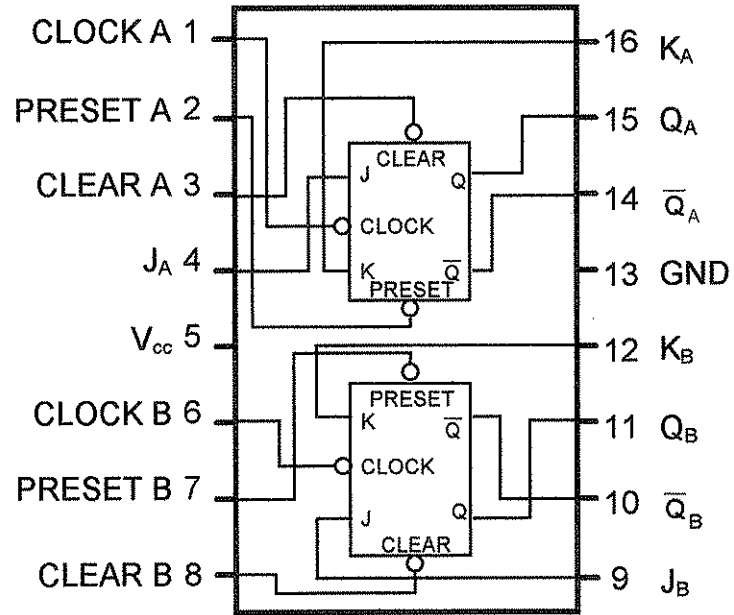
1. Deney sonuçlarını yazarak yorumlayınız.
2. Şekil 1'deki eşzamanlı sayıcının tasarım aşamasını gerçekleştiriniz.



Şekil 4 7473 tümdevresinin yapısı



Şekil 5 7408 tümdevresinin yapısı



Şekil 6 7476 tümdevresinin yapısı

Ondokuzmayıs Üniversitesi
Elektrik-Elektronik Mühendisliği Bölümü
Sayısal Elektronik Laboratuvarı

DENEY ADI: EPROM

DENEYİN AMACI: Bu deneyde öncelikle bellekler hakkında genel bir bilgi edinilecek daha sonra EPROM ların okunması ve adreslenmesiyle ilgili uygulamalar yapılacaktır.

AÇIKLAYICI BİLGİLER:

Litaretürde ROM (Read Only Memory-Yalnız Okunabilir Bellek) olarak adı geçen belleklerin içindeki veriler imalat aşamasında yazılır ve bir daha da silinip elektriksel olarak yazılamazlar. Sadece okunabilirler. RAM (Read Access Memory-Rastgele Erişilebilir Bellek) olarak adı geçen bellekler ise hem okumaya hem de yazılmaya elverişli belleklerdir.

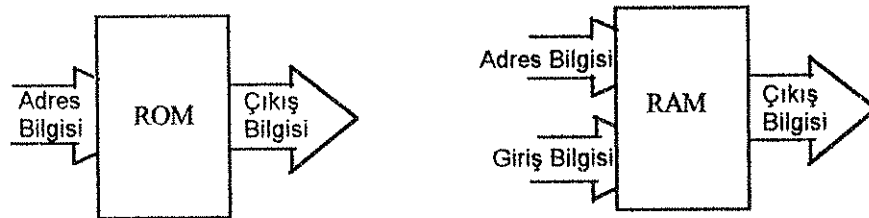
Bu belleklere veriler binary (ikili) işaretler olarak kaydedilir ve okunur. Bir verinin bir belleğe yazılması ve ya bellekteki bir verinin okunabilmesi için söz konusu verinin bellekte kaydedileceği yerin ve ya okunacak verinin bellekteki yerinin adres bilgisinin belirlenmesi gerekmektedir. Bu bilgi yardımıyla o adres noktasındaki veri okunabilir ve ya o adres noktasına (Location) veri yazılabilir.

ROM ve RAM bellekler giriş çıkış tanımları ile şekil-1 deki gibi ifade edilebilirler. Şekilden de anlaşılacağı gibi ROM belleklerde veri girişi uçları yoktur. ROM belleklerin elektriksel olarak yazılabilen çeşitleri de vardır. Bunlar PROM, EPROM, EEPROM olarak isimlendirilirler.

a) PROM (Programmable ROM) : Bu bellekler imalat çıkışı olarak tamamen boştur ve kullanıcı isteği doğrultusunda programlar. Daha sonra PROM un içinde hep bu program olur ve tekrar programlanamaz.

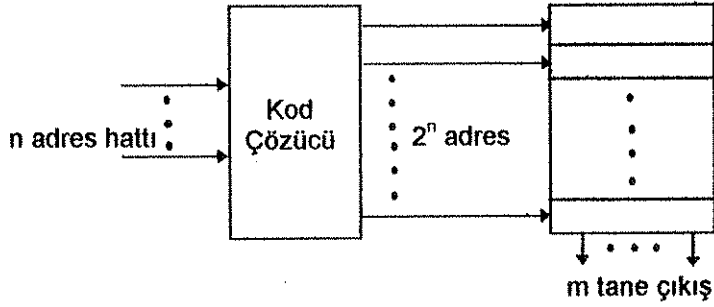
b) EPROM (Erasable Programmable ROM) : Bu tip ROM lar ultraviyole ışınlarla silinip, elektriksel olarak yeniden yazılabilirler.

c) EEPROM (Electrically Erasable Programmable ROM) : Elektriksel olarak silinip yazılabilirler. Bunların özel bir silme uçları vardır.

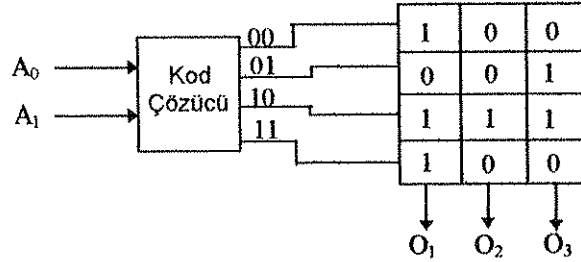


Şekil-1: Belleklerin Bilgi Giriş-Çıkış Durumları

Bellekleri kapasiteleri birbirinden farklıdır. Şekil 2 de $2^n \times m$ boyutlu ROM modeli verilmiştir. Bunun anlamı 2^n tane m bitlik veriyi saklayabilmektir. n, adres bilgisinin bit sayısını belirtmektedir. Bazı bellekler sadece 8 bit bilgiyi saklayabilirken bazıları ise 256000 bit ya da daha fazlasını saklayabilirler. Bir belleği satırlardan oluşan ve her bir satırında bir veri bulunan ve her verinin m sayıda bitten oluştuğu bir dizi gibi düşünebiliriz. Adres bilgisi satır sayısını belirlemekte, Çıkış bilgisi ise her bir satırda ki bit sayısının belirlemekte kullanılır. Bütün adresler 0 olduğunda belleğin ilk satırına, bütün adresler 1 olduğunda ise belleğin son satırına ulaşılır. Bütün bu satırlardaki veriler ise Çıkış Bilgisi hattından görülebilir.

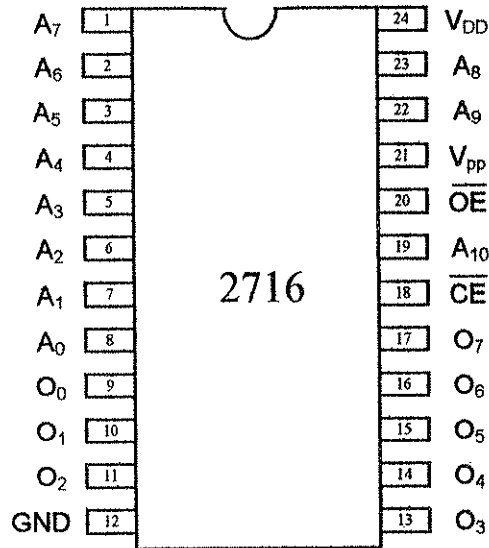
Şekil 2: $2^n \times m$ Boyutlu ROM modeli

Örneğin, 4×3 lük bir ROM un yapısını inceleyelim. 4 ün anlamı bu bellekte 4 satır olduğudur. 3 ün anlamı ise her satırda 3 bitlik veri bulunabileceğidir. $4 (2^2)$ satır adres için 2 tane adres hattı yeterlidir. A_0 ve A_1 . her ikisinde 0 olduğunda 00 adresine ulaşılır. $A_1 = 0, A_0 = 1$ olduğunda 01 adresine, $A_1 = 1, A_0 = 0$ olduğunda 10 adresine ve her ikisi de 1 olduğunda 11 adresine ulaşılır. Bu adreslere veriler imalat aşamasında yazılmıştır ve değiştirilmesi mümkün değildir. Sadece bu verilerin ne olduğu Çıkış Bilgisi Hattından görülebilir. (Eğer bu bir EPROM olsaydı verilerin silinip yeniden yazılabilme olanağı olurdu).

Şekil-3 : Kabaca 4×3 ROM un iç yapısı

2716 EPROM' un YAPISI

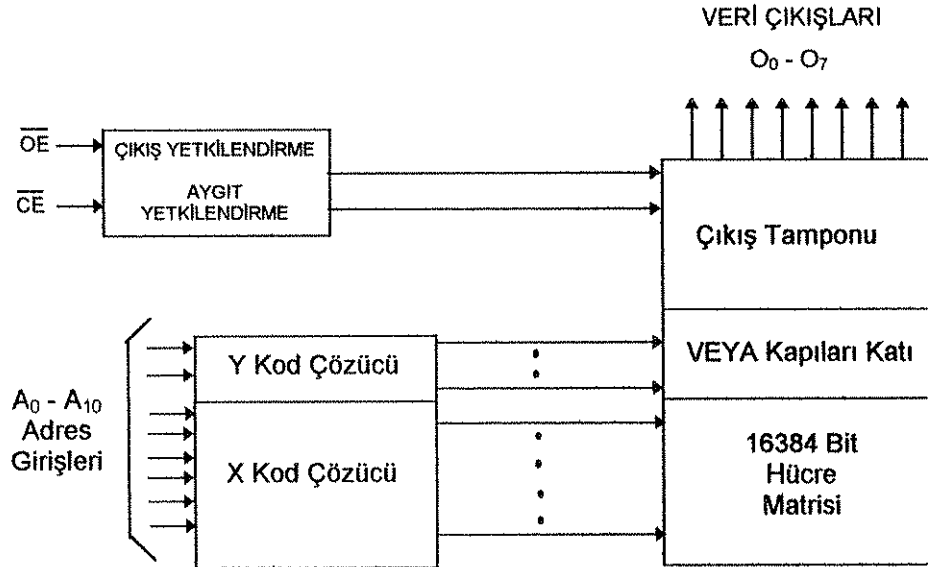
2716 EPROM isminden de anlaşılacağı gibi elektrikselsel olarak programlanabilir. Silmek için entegrenin üzerinde görülen camlı bölgeye UV ışınları uygulamak yeterlidir. İçine tekrar program yazdıktan sonra bu bölge ışık geçirmeyen bir nesne ile kapatılmalıdır ki içindeki program zamanla zarar görmesin. Şekil-4 de bir 2716 EPROM un dıştan görüntüsü ve her bir ucunun işlevleri gösterilmiştir.



Şekil 4: 2716 EPROM un dıştan görüntüsü ve her ucunun işlevi

2716 EPROM aynı anda 16384 biti saklayabilecek kapasitededir. Bu bitleri 8'er tane olarak 2048 satırda saklar. Bu nedenle 2716 2Kx8'lik bir EPROM'dur denilebilir. A_0 dan A_{10} a kadar olan adresler her bir satırı adreslemek için kullanılırlar. Bu adreslerin hepsi 0 olduğu zaman EPROM'un ilk satırına ulaşılır ve buraya 8 bitlik bir veri saklanabilir. Adreslerin hepsi 1 olduğunda ise son satıra ulaşılır. O_0 dan O_7 ye kadar olan veri hatlarından ise her satırdaki 8 bitlik veriler çıkar. 2716'nın bu şekildeki simgesel gösterimi Şekil-5'deki gibidir.

2716 EPROM'un aktif olabilmesi için \overline{CE} (Chip Enable - Aygıt Yetkilendirme) ucunun aktif yani mantık 0 olması gereklidir. Çıktıların alınabilmesi için de \overline{OE} (Output Enable - Çıkış Yetkilendirme) ucunun mantık 0 olması gereklidir.



Şekil 5: 2716'nın blok gösterimi

EPROM'un OKUNMASI:

Daha önce içinde program olan bir EPROM'un içeriğini okuyabilmek için öncelikle \overline{CE} ve \overline{OE} pinlerinin aktif olması gereklidir. Bunun için bu uçlara mantık 0 uygulanmalıdır. Besleme gerilimlerini de uyguladıktan sonra okumak istenilen adres, adres hatlarından girilir ve bu adresin içeriğinde ne olduğu $O_0 - O_7$ hattından dışarıya verildiğinden bu hatlara bakılarak içerik kolayca görülebilir.

EPROM'un ADRESLENMESİ:

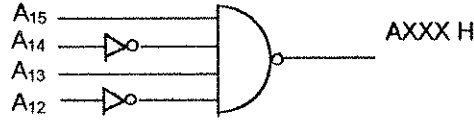
EPROM'lar genellikle mikroişlemci setlerinde kullanılırlar. Bir mikroişlemci setinde birden fazla EPROM vardır ve her birinin ayrı adreslenmesi gereklidir. Bu adresleme için ihtiyaç duyulan adres bilgisi mikroişlemcinin adres hattından gelir. Genelde mikroişlemcilerde 16, daha gelişmiş olanlarında 20 tane adres hattı ($A_0 - A_{15}$, $A_0 - A_{19}$) bulunur ve çoğu mikroişlemci ilk çalıştığında adreslerin tümü 0'dır. Mikroişlemci içindeki Program Counter yardımıyla teker teker artarak EPROM'un her satırına ulaşılır. Ama program içinde gidilmesi gerekli başka adresler varsa, Program Counter buralara da ulaşacaktır. 16 bit adres hattıyla 2^{16} ayrı adres üretilebilir. Bu da 65536 (64K) satır demektir. Ama bu büyüklükte bir EPROM olmadığını düşünürsek bir mikroişlemcinin bütün bellek kapasitesini kullanmak için $64K / 2K = 32$ tane 2716'ya ihtiyacımız vardır. Bunların hepsinin ayrı bir adresi olmalıdır ki devre çalıştığında bunlar sırayla aktif olmalıdır. Fakat her defasında bu kadar geniş bir belleğe ihtiyacımız olmayabilir.

Adresler kolaylık olması açısından daima Hexadecimal olarak gösterilirler. Şimdi aşağıdaki örneği inceleyelim. $A_{15} A_{14} A_{13} A_{12}$ hatlarını sabit değerlerde tutup diğerlerini EPROM'a bağladığımızı düşünelim. Bu durumda adresler,

A_{15} A_{14} A_{13} A_{12} A_{11} A_{10} A_9 A_8 A_7 A_6 A_5 A_4 A_3 A_2 A_1 A_0
 1 0 1 0 X X X X X X X X X X X XAXXX H
 Ya da
 1 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0A000 H den
 1 0 1 0 1 1 1 1 1 1 1 1 1 1 1 1AFFF H e kadar

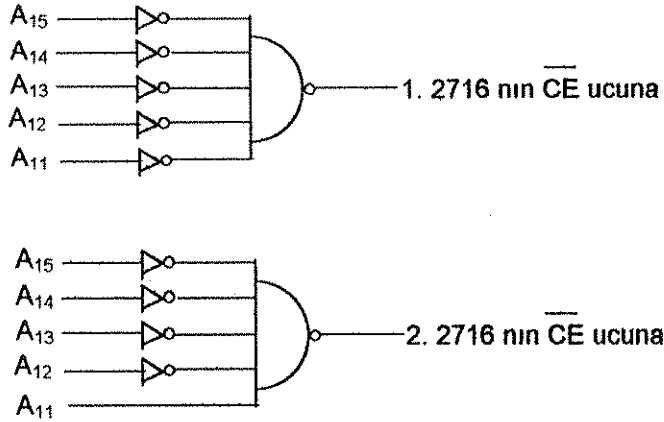
değişecektir.

Bunu gerçeklemek için Şekil 6 deki devreyi kurup, çıkışını 4 K byte lık bir EPROM un \overline{CE} pinine bağlarsak EPROM sadece $A_{15} = 1$, $A_{14} = 0$, $A_{13} = 1$ ve $A_{12} = 0$ olduğu zamanlar çalışacak ve adresleri A000 H ile AFFF H arasında değişecektir.



Şekil 6: AXXH şeklinde adresleme için kurulan devre

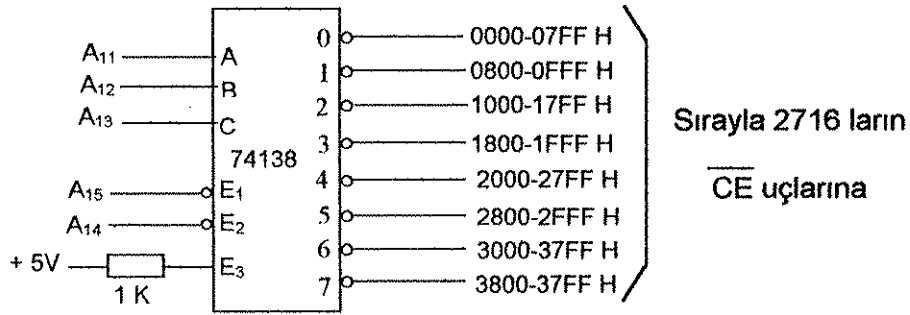
Örneğin, 0000 H ile 0FFF H adresleri arasına 2716 EPROM bağlamak isteniyor. Bu adresler arasında 4 Kbyte lık bir bölge var. 2716 2 Kbyte lık bir EPROM olduğundan dolayı bu bölgeye 2 tane EPROM bağlanabilir. Bir tanesi 0000 H - 07FF H adresleri arasına, diğeri 0800 H - 0FFF H adresleri arasına bağlanacaktır. Bunu gerçekleyebilmek için 2 tane VE kapısı yeterli olacaktır (Şekil 7).



Şekil 7: 0000 H - 0FFF H arasındaki bölgeye 2716 EPROM bağlantısını gösteren devre

Yukarıdaki ilk devrede $A_{15} = A_{14} = A_{13} = A_{12} = A_{11} = 0$ olduğunda 1. 2716 nın \overline{CE} ucuna 0 gidecek ve aktif olacaktır. Diğer durumlarda çalışmayacaktır. $A_{10} - A_0$ adresleri da 2716 nın adres hattına bağlanırsa 1. 2716 nın adresi 0000 H - 07FF H arasında olacaktır. Aynı şekilde 2. 2716 için düşünülürse onun adresi de 0800 H - 0FFF H arasında olur.

Daha fazla sayıda bellek için her biri için VE kapısı kullanmak yerine tek bir Kod Çözücü (DECODER) kullanılabilir. Örneğin, Şekil 8 deki devrede 0000 H - 37FF H adresleri arasına 8 tane 2716 EPROM un bağlanması gösterilmiştir.



Şekil 8: Kod Çözücü kullanarak adresleme

Hazırlık Soruları:

- 1) 2716 ve 2764A EPROM ların yapısı hakkında bilgi edininiz.
- 2) 74138 decoder hakkında bilgi edininiz.
- 3) 2716 ve 2764A EPROM ların kapasiteleri arasındaki farkı bulunuz.
- 4) 5000 H adresinden 5FFF H adresine kadar olan bellek bölgesine EPROM yerleştirilmek isteniyor. Bu bölgeye kaç tane 2716 EPROM sığabileceğini bulunuz. Bu EPROM ları adresleyebilmek için gerekli devreyi tasarlayınız.
- 5) E000 H adresinden F7FF H adresine kadar olan bellek bölgesinin genişliği ne kadardır. Bu bölgeye kaç tane 2716 ve kaç tane 2764A EPROM sığar. Gerekli devreyi 74138 Kod Çözücü kullanarak tasarlayınız.

DENEYİN YAPILIŞI:

- 1) Daha önceden içerisinde veri olan 2716 EPROM u okumak için gerekli devreyi kurarak, çeşitli adresler vererek buralarda hangi bilgilerin saklı olduğunu bulunuz.
- 2) Hazırlık sorusu 3 de tasarladığınız devreyi kurunuz, çeşitli adreslerdeki bilgileri okuyunuz.
- 3) Hazırlık sorusu 4 de tasarladığınız devreyi kurunuz, çeşitli adreslerdeki bilgileri okuyunuz.

RAPORDA İSTENENLER:

- 1) Tasarladığınız devreleri çiziniz.
- 2) Deneyde elde ettiğiniz verileri yazınız.

SAYISAL ELEKTRONİK LABORATUVARI

A/D VE D/A DÖNÜŞTÜRÜCÜLER

1. Deneğin Amacı

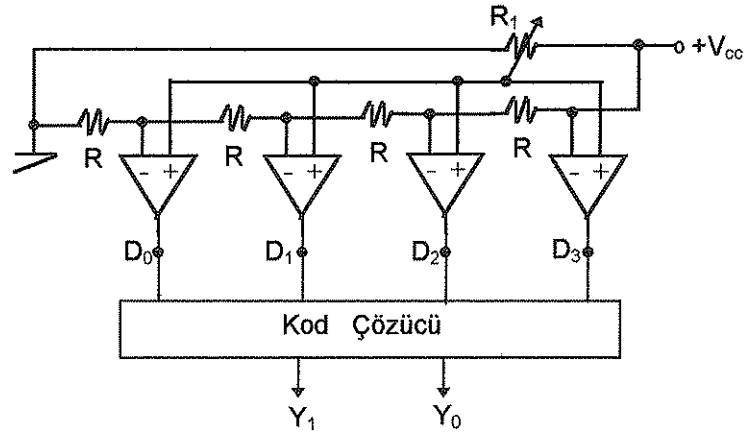
Sayısal Elektronik sistemlerin endüstriyel uygulamalarda kullanılmaya başlamasıyla birlikte karşılaşılan en önemli sorun, çeşitli analog büyüklüklerin (gerilim, akım, basınç, debi, derinlik, ses vb.) sayısal bilgiye dönüştürülebilmesi olmuştur. Analog bir büyüklüğü sayısal büyüklüğe dönüştürmede iki önemli aşama vardır. Birincisi analog büyüklüğün yarattığı fiziksel etkinin algılanması, ikincisi ise algılanan bu büyüklüğün sayısal olarak işlenebilecek duruma getirilmesidir. Bunun yanında bir çok uygulamada, sayısal olarak işlenen biginin tekrar analog hale dönüştürülmesi gerekmektedir. Biz bu deneyde algılanmış bir analog büyüklüğün sayısal bir büyüklüğe dönüştürülmesi ve elde edilen bu sayısal bilginin tekrar analog hale nasıl getirileceği konularını inceleyeceğiz. Bu inceleme sırasında A/D (Analog- Dijital) ve D/A (Dijital- Analog) dönüştürücülerin çalışma ilkeleri, kullandıkları uygulama alanları ve bu tür dönüştürücüler üzerinde ne tür ayarlar yapılabileceği gözlenecektir.

2. Açıklayıcı Bilgiler

Sayısal elektronik sistemlerde A/D ve D/A dönüştürme yapmak için özel tüm devreler kullanılır. Genellikle kullanılan 8-bit A/D ve 8-bit D/A dönüştürücüler piyasada bol miktarda bulunmaktadır. Bu deney yapılırken de A/D ve D/A dönüştürücüler, hazır tümdevre elemanlar üzerinde incelenecektir.

2.1. Analog- Dijital Dönüştürücü (A/D Transducer)

Doğada bulunan fiziksel büyüklükler, algılandıktan sonra, bir A/D dönüştürücü yoluyla, sayısal bilgiye dönüştürülürler. Elde edilen bu bilgi bilgisayarlar tarafından işlenerek, gerekli çözümlerinin yapılması için kullanılır. Şekil-1' de 2-bit' lik en temel A/D dönüştürücü yapısı verilmiştir.



Şekil-1. 2-bit' lik A/D dönüştürücü

Şekil-1' de verilen OPAMP' ların "+" uclarına bağlı oldevrede giriş gerilimi R_1 direnci sayesinde 0 ile $+V_{cc}$ aralığında değiştirilebilmektedir. $+V_{cc}$ gerilimi R dirençleri kullanılarak dörde bölünmüş ve OPAMP' ların "-" uclarına bağlanmıştır. Çıkışların değişimi Tablo1' de verilmiştir.

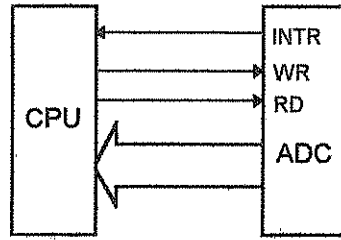
Tablo-1. 2-bit' lik A/D dönüştürücü doğruluk tablosu

V_{gir}	D_3	D_2	D_1	D_0	Y_1	Y_0
$0 < V_{gir} < V_{cc}/4$	0	0	0	0	0	0
$V_{cc}/4 < V_{gir} < 2V_{cc}/4$	0	0	0	V_{cc}	0	V_{cc}
$2V_{cc}/4 < V_{gir} < 3V_{cc}/4$	0	0	V_{cc}	V_{cc}	V_{cc}	0
$3V_{cc}/4 < V_{gir} < 4V_{cc}/4$	0	V_{cc}	V_{cc}	V_{cc}	V_{cc}	V_{cc}
$V_{gir} = V_{cc}$	0	V_{cc}	V_{cc}	V_{cc}	V_{cc}	V_{cc}

Çeşitli uygulamalarda A/D dönüştürme işlevi, bu iş için imal edilmiş tümdevreler kullanılarak gerçekleştirilir.

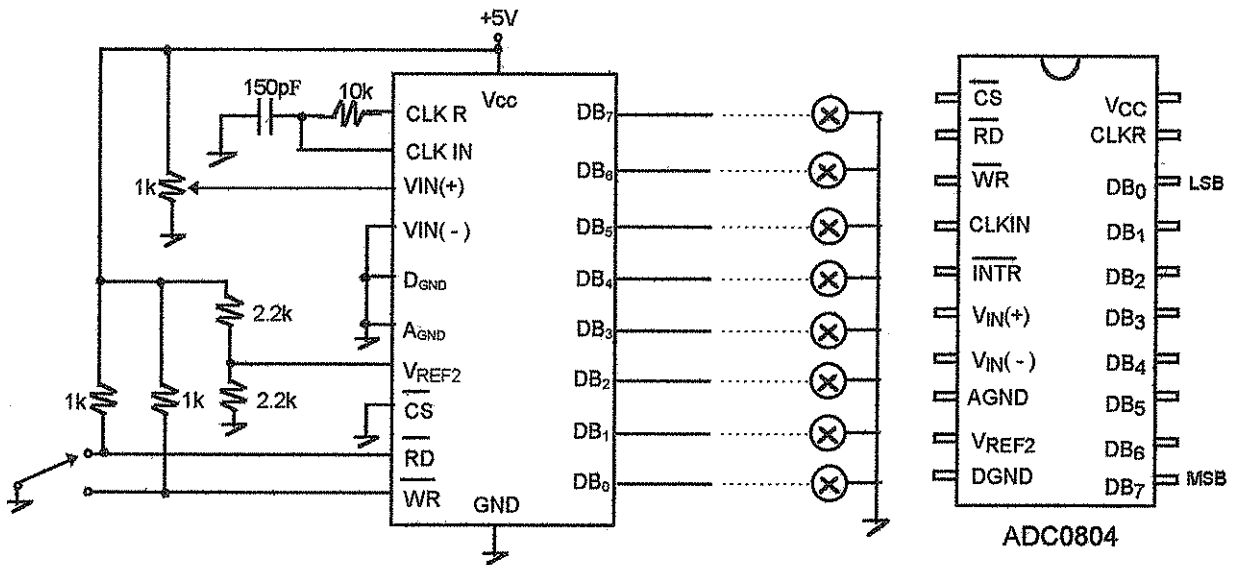
2.2. ADC0804 Tümdevresi

A/D dönüştürücü ADC0804 tümdevresi kullanılarak incelenecektir. ADC0804 tümdevresi 8-bitlik ve microprocessor uyumlu bir A/D dönüştürücüdür. Şekil-2' de bu ilkesel yapı gösterilmiştir.



Şekil-2. ADC - CPU bağlantısı

CPU, RD ve WR denetim işaretlerini kullanarak ADC' den (A/D dönüştürücü) okuma yapabilir. ADC0804 tüm devresi istenilen bir referans analog giriş değeri için interrupt üretebilmektedir. Üretilen bu interrupt INTR ucu kullanılarak CPU 'ya iletilir.Şekil-3' de ADC0804 tümdevresi kullanılarak analog bir işaretin 8-bit' lik sayısal veriye dönüştürülmesi için gerekli devre bağlantısı verilmiştir.



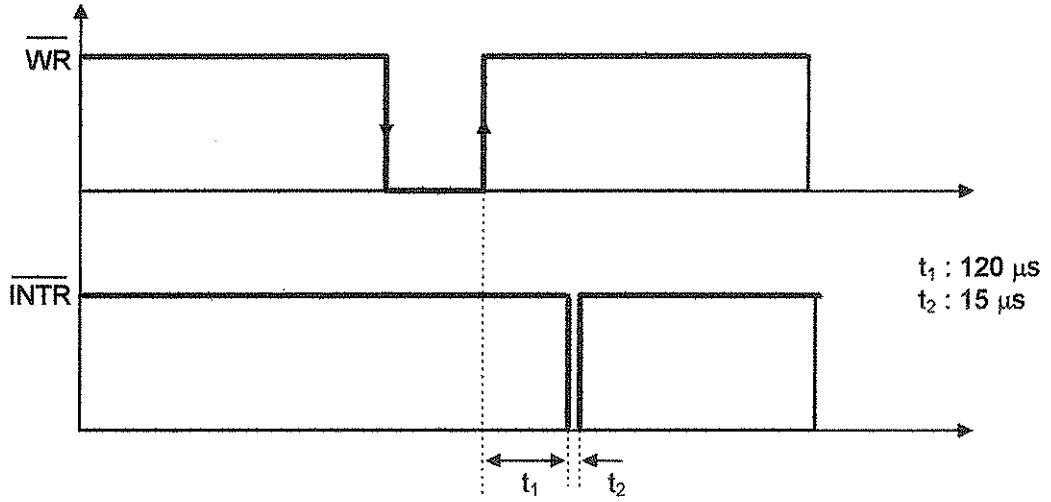
Şekil-3. ADC0804 tümdevresi kullanılarak gerçekleştirilen A/D dönüştürücü

Tablo-2' de \overline{WR} ve \overline{RD} denetim işaretlerinin fonksiyonları gösterilmiştir.

Tablo-2. \overline{WR} ve \overline{RD} denetim işaretlerinin fonksiyonları

\overline{WR}	\overline{RD}	DB ₀ - DB ₇ (Sayısal çıkışlar)	VIN(+) (Analog giriş)
1	1	Yüksek-Z	pasif
↓	1	Yüksek-Z	örnekle
↓	0	aktif	örnekle
1	0	aktif	pasif

ADC0804 tümdevresi interrupt üretebilen bir denetim işaretine de sahiptir. Bu denetim işareti mikroişlemci uygulamalarında kullanılır. Şekil-4' de INTR denetim işaretinin üretilmesi gösterilmiştir.

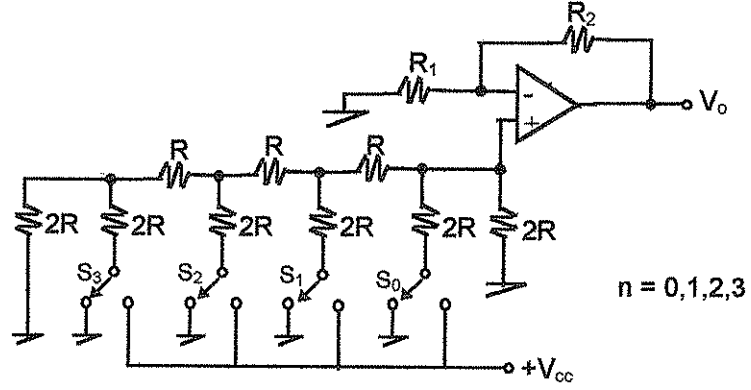


Şekil-4. \overline{INTR} denetim işaretinin üretilmesi

\overline{WR} denetim ucu alçak seviyeye çekildiğinde düşen kenarda analog giriş işareti bir kez örneklenir. \overline{WR} denetim işaretinin yükselen kenarından $120\mu s$ sonra \overline{INTR} ucu $15\mu s$ uzunluğunda bir interrupt üretir.

2.3. Dijital - Analog Dönüştürücü (D / A Transducer)

Dijital bir bilgi tekrar analog fiziksel bir büyüklüğe dönüştürülmek istendiğinde D/A dönüştürücü kullanılır. Şekil-5' de merdiven tipi 4-bit'lik bir dijital analog dönüştürücü gösterilmiştir.



Şekil-5. 4-bit D/A dönüştürücü

Şekil-2' deki devrede çıkış gerilimi V_0 ,

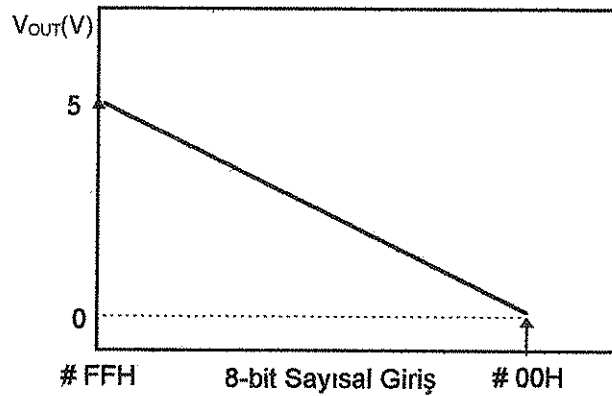
$$\sum_{k=0}^{n-1} \frac{V_{cc}}{2^k(n-1)} \left(\frac{R_2 + R_1}{R_1} \right) \quad (1)$$

n : toplam bitlerin sayısı

eşitliğinden elde edilir.

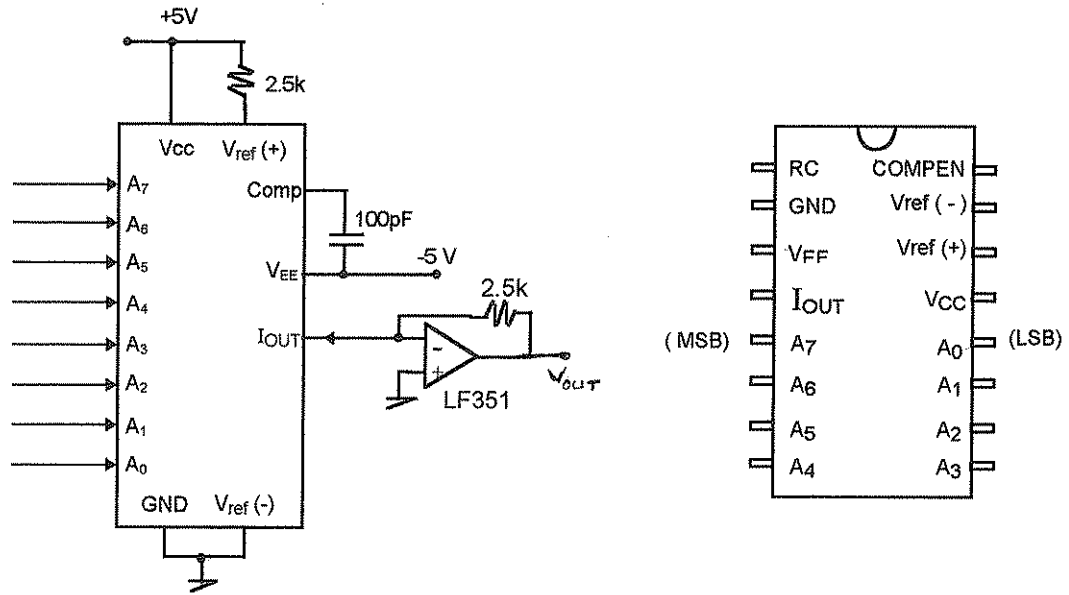
2.4. DAC1408 Tümdevresi

DAC1408 tümdevresi 8-bit' sayısal veri girişi olan bir D/A dönüştürücüdür. Bu tür tümdevreler girişlerindeki sayısal bilgi ile orantılı bir akım üretirler. Şekil-7' deki devrede $5k\Omega$ yük direnci için geçiş karakteristiği Şekil-6' deki gibidir.



Şekil-6. DAC 1408 tümdevresinin geçiş karakteristiği

Şekil-7'da 8-bit'lik sayısal veri girişini analog işarete dönüştürmek için gerekli bağlantı gösterilmiştir.



Şekil-7. DAC1408 tüm devresi kullanarak gerçekleştirilen D/A dönüştürücü

3. Deneğin Yapılışı

3.1. Şekil-3' deki devreyi kurun, AD0804 tümdevresinin sayısal çıkışlarını board üzerinde bulunan led lambalara bağlayın. Daha sonra Tablo-3' de belirtilen ölçümleri yapın.

Tablo-3

V_{in} (V)	0.0	0.1	0.2	0.5	1.0	1.5	2.0	2.5	3.0	3.5	4.0	4.5	5.0
Data Out (H)													

3.2. Şekil-7' daki devreyi kurun, DAC1408 tümdevresinin sayısal girişlerini board üzerinde bulunan anahtarlara bağlayın ve Tablo-4' de belirtilen ölçümleri yapın.

Tablo-4

Data In (H)	#00	#02	#04	#08	#10	#20	#40	#80	#8F	#9F	#BF	#DF	#FF
V_{OUT} (V)													

3.3. ADC çıkışını DAC girişine uygulayarak iki devreyi birleştiriniz. ADC girişine sinüsoidal işaret veriniz DAC çıkışını osiloskopta gözleyiniz.

4. Hazırlık Soruları

4.1. D/A ve A/D dönüştürücülerin kullanım alanlarını araştırınız.

4.2. Bir elektrik motorunun dönme hızı ışık şiddetine bağlı olarak değiştirilmek isteniyor. Böyle bir uygulamayı gerçekleştirmek için A/D ve D/A dönüştürücülerden nasıl yararlanılabilir? Önerdiğiniz devre bağlantısını çizerek yorumlayınız.