

57220



**PC BİLGİSAYARLARINDA
SAYISAL ELEKTRONİK DENEY SETİ
SİMÜLASYONU**

Yalçın İŞLER

**Yüksek Lisans Tezi
ELEKTRONİK-HABERLEŞME MÜH. A.B.D.
ISPARTA - 1996**

T.C.
SÜLEYMAN DEMÜREL ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

PC BİLGİSAYARLARINDA
SAYISAL ELEKTRONİK DENEY SETİ SİMÜLASYONU

F.C. YÜKSEK LİSANS TEZİ
DOKÜMAN

Yalçın İŞLER

YÜKSEK LİSANS TEZİ

ELEKTRONİK - HABERLEŞME MÜHENDİSLİĞİ ANABİLİM DALI

M. Merdan
Prof.Dr. Mustafa MERDAN
(Danışman)

Ali Kökçe
Prof. Dr. Ali KÖKÇE
(Üye)

Ali Omral
Yrd. Doç. Dr. Ali OMRAL
(Üye)

ÖZET

Projenin amacı, üniversitelerin elektronik bölümlerinde verilmekte olan Sayısal Elektronik derslerinde öğrencilere uygulamaların yaptırıldığı deney setlerine benzer eğitim araçlarının yaptığı işi bilgisayar ortamında simülasyon yolu ile daha ucuza ve daha pratik olarak yapılabileceğini ortaya koymaktır.

Projenin gerçekleştirilmesi sırasında Turbo Pascal programlama dili kullanılmıştır. Bu programlama dilinin seçiminde sunduğu grafik ve nesne tabanlı yazılım geliştirme kolaylıkları etkili olmuştur.

Deney setleri oldukça kapsamlı olduğu için, sadece temel teşkil edecek olan deneylerin bilgisayarda simülasyonlarının yapılması hedef olarak belirlenmiştir. Bu deneylerin dışında kalan diğer deneyler için, ayrı bir bilgisayar programı yazılması gereklidir. Bu deneylerde temel kapı devrelerinin ve çeşitli flip-flop' ların kullanımı, sayıcı, kodlayıcı ve kod çözücü devrelerinin tasarımı ile 4 - bitlik ikilik sayıcı ve 4 - bitlik tam toplayıcı entegrelerinin kullanımı yer almaktadır.

Programlama tekniği olarak, gelecekte yapılabilecek değişiklikler göz önüne alınarak oldukça esnek teknikler kullanılmıştır. Buna rağmen, hazırlanan bu simülasyon programının kullanımı ve anlaşılması çok kolaydır.

ABSTRACT

This project was prepared to provide Digital Logic Experiments System likely was used by electronic departments of universities.

It was aimed that the computer software of a experiment system simulator that can easily provide some advantages. These advantages : it is more cheaper and practical than real word experiments system.

Turbo Pascal programming language was used in preparing this project. Because, it has some advantages for graphical object oriented programming techniques.

Because of the experiments systems were fairly complex, only some common experiments were consisted into this project. It must be written as another computer simulating program if the other experiments are becoming extremely necessary. In these experiments that defined in this project, there were using some common logic gates' circuits and various flip-flops, designing counter, encoder and decoder circuits, and using counter and full adder integrated circuits.

More flexible programming techniques are selected for thinking about changings in the future. Because of the project was designed hardly and strongly, using the simulating program is very easy.

ÖNSÖZ ve TEŞEKKÜR

Bilgisayar destekli olarak yapılan deney setlerinde, öğrencilerin kurdukları deneylerin sonuçlarını gözlemleyebilmek için ihtiyaç duydukları dijital gösterge panelleri ve osiloskop, sinyal üretici gibi pahalı cihazlara olan ihtiyaçları ortadan kalkacaktır. Ayrıca, deneyler yapılırken öğrencilerin malzemelerle direkt temasları söz konusu olmadığı için, öğrencilerin malzemeleri bozmaları riski de ortadan kalkacaktır. Böylece deney setleri satın almak yerine daha ucuz olan bilgisayarda simülasyon programlarını satın almak, okullara maddi yönden de kazanç sağlayacaktır.

Elektronik devre tasarımı devrenin çalışmasını göstermek için doğru yöntemleri gerektirir. Modern entegre devrelerin karmaşıklığı nedeniyle, bilgisayar destekli tasarım yapmak zorunlu hale gelmiştir. Aynı zamanda, bilgisayar destekli devre analizi ile devre hakkındaki bilgilerden laboratuvar şartlarında yapılan ölçümlerle saptanması hemen hemen mümkün olmayanlar da saptanabilir.

Bu araştırmanın konusunu öneren ve araştırmanın her safhasında büyük bir ilgi ile yapıcı eleştirileri ve önerilerini esirgemeyen danışman hocam Prof. Dr. Mustafa MERDAN' a şükranlarımı sunarım. Çalışmalarım sırasında aynı yardımda bulunan değerli arkadaşım Elektronik Mühendisi Abdullah ARISAN' a, ayrıca bu çalışmanın her safhasında manevi katkı, emek ve anlayış gösteren sevgili eşim Özlem İŞLER' e teşekkürü bir borç bilirim.

İÇİNDEKİLER

ÖZET	i
ABSTRACT	ii
ÖNSÖZ ve TEŞEKKÜR	iii
ŞEKİLLER LİSTESİ	vi
TABLolar LİSTESİ	vii
1. GİRİŞ	1
2. PROGRAMLAMA DİLİ	2
3. SAYISAL ELEKTRONİK DENEY SETİ	5
3.1. Deneylerde kullanılan temel malzemeler	6
3.1.1. Sayısal mantık kapı entegreleri	6
3.1.1.1. VE kapısı	6
3.1.1.2. VEYA kapısı	7
3.1.1.3. DEĞİL kapısı	7
3.1.1.4. ÖZEL VEYA kapısı	8
3.1.1.5. VE DEĞİL kapısı	8
3.1.1.6. VEYA DEĞİL kapısı	9
3.1.2. Sayısal mantık hafıza entegreleri	9
3.1.2.1. Pozitif kenar tetiklemeli JK flip-flop	10
3.1.2.2. Negatif kenar tetiklemeli JK flip-flop	11
3.1.2.3. Pozitif kenar tetiklemeli RS flip-flop	12
3.1.2.4. Negatif kenar tetiklemeli RS flip-flop	13
3.1.2.5. Pozitif kenar tetiklemeli D tipi flip-flop	14
3.1.2.6. Negatif kenar tetiklemeli D tipi flip-flop	14
3.1.3. Sayısal mantıkta kullanılan bazı kompleks entegreler	15
3.1.3.1. Sayıcı entegresi : 7493	15
3.1.3.2. Toplayıcı entegresi : 7483	16
3.2. Deneyler	16
3.2.1. Bazı lojik kapıların çalışmasının incelenmesi	18
3.2.2. Temel sequential devrelerin çalışmasının incelenmesi - 1	19
3.2.3. Temel sequential devrelerin çalışmasının incelenmesi - 2	21
3.2.4. Temel sequential devrelerin çalışmasının incelenmesi - 3	22
3.2.5. Asenkron 4 bit binary sayıcı	24
3.2.6. Asenkron 4 bit BCD sayıcı	26
3.2.7. Senkron 4 bit binary sayıcı	28
3.2.8. 7493 ile 4 bit binary sayıcı	30
3.2.9. Seri giriş paralel çıkışlı shift register	33
3.2.10. Paralel giriş seri çıkışlı shift register	35
3.2.11. Kodlayıcı devresinin çalışmasının incelenmesi	38
3.2.12. Kod çözücü devresinin çalışmasının incelenmesi	40
3.2.13. 4 bitlik iki sayının toplama devresi	42
3.2.14. 4 bitlik iki sayının çıkarma devresi	45
3.2.15. 8 bitlik iki sayının toplama ve çıkarma devresi	48
4. DENEYLERİN BİLGİSAYARA TANITIMI	52
5. DENEYLERİN YAPILMASI	72



ŞEKİLLER LİSTESİ

Şekil 3.01. VE kapısının şekli	6
Şekil 3.02. VEYA kapısının şekli	7
Şekil 3.03. DEĞİL kapısının şekli	7
Şekil 3.04. ÖZEL VEYA kapısının şekli	8
Şekil 3.05. VE DEĞİL kapısının şekli	8
Şekil 3.06. VEYA DEĞİL kapısının şekli	9
Şekil 3.07. Pozitif kenar tetiklemeli JK flip-flop' un şekli	10
Şekil 3.08. Negatif kenar tetiklemeli JK flip-flop' un şekli	11
Şekil 3.09. Pozitif kenar tetiklemeli RS flip-flop' un şekli	12
Şekil 3.10. Negatif kenar tetiklemeli RS flip-flop' un şekli	13
Şekil 3.11. Pozitif kenar tetiklemeli D tipi flip-flop' un şekli	14
Şekil 3.12. Negatif kenar tetiklemeli D tipi flip-flop' un şekli	14
Şekil 3.13. 7493 sayıcı entegresinin şekli	15
Şekil 3.14. 7483 toplayıcı entegresinin şekli	16
Şekil 3.15. Deney - 1' in ekran görüntüsü	18
Şekil 3.16. Deney - 2' nin ekran görüntüsü	19
Şekil 3.17. Deney - 3' ün ekran görüntüsü	21
Şekil 3.18. Deney - 4' ün ekran görüntüsü	23
Şekil 3.19. Deney - 5' in ekran görüntüsü	24
Şekil 3.20. Deney - 6' in ekran görüntüsü	26
Şekil 3.21. Deney - 7' nin ekran görüntüsü	28
Şekil 3.22. Deney - 8' in ekran görüntüsü	30
Şekil 3.23. Deney - 9' un ekran görüntüsü	33
Şekil 3.24. Deney - 10' un ekran görüntüsü	35
Şekil 3.25. Deney - 11' in ekran görüntüsü	38
Şekil 3.26. Deney - 12' nin ekran görüntüsü	40
Şekil 3.27. Deney - 13' ün ekran görüntüsü	42
Şekil 3.28. Deney - 14' ün ekran görüntüsü	45
Şekil 3.29. Deney - 15' in ekran görüntüsü	48
Şekil 5.01. Programın ilk ekran görüntüsü	72

TABLOLAR LİSTESİ

Tablo 3.01. VE kapısının doğruluk tablosu	7
Tablo 3.02. VEYA kapısının doğruluk tablosu	7
Tablo 3.03. DEĞİL kapısının doğruluk tablosu	8
Tablo 3.04. ÖZEL VEYA kapısının doğruluk tablosu	8
Tablo 3.05. VE DEĞİL kapısının doğruluk tablosu	9
Tablo 3.06. VEYA DEĞİL kapısının doğruluk tablosu	9
Tablo 3.07. Pozitif kenar tetiklemeli JK flip-flop' un doğruluk tablosu	11
Tablo 3.08. Negatif kenar tetiklemeli JK flip-flop' un doğruluk tablosu	11
Tablo 3.09. Pozitif kenar tetiklemeli RS flip-flop' un doğruluk tablosu	12
Tablo 3.10. Negatif kenar tetiklemeli RS flip-flop' un doğruluk tablosu	13
Tablo 3.11. Pozitif kenar tetiklemeli D tipi flip-flop' un doğruluk tablosu	14
Tablo 3.12. Negatif kenar tetiklemeli D tipi flip-flop' un doğruluk tablosu	15
Tablo 3.13. Deney - 1' deki çıkış ifadelerinin fonksiyonları	18
Tablo 3.14. Deney - 1' in sonucunda elde edilen doğruluk tablosu	19
Tablo 3.15. Deney - 2' deki giriş ve çıkışların anlamları	20
Tablo 3.16. Deney - 2' nin sonucunda elde edilen doğruluk tablosu	20
Tablo 3.17. Deney - 3' deki giriş ve çıkışların anlamları	21
Tablo 3.18. Deney - 3' ün sonucunda elde edilen doğruluk tablosu	22
Tablo 3.19. Deney - 4' deki giriş ve çıkışların anlamları	22
Tablo 3.20. Deney - 4' ün sonucunda elde edilen doğruluk tablosu	24
Tablo 3.21. Deney - 5' deki giriş ve çıkışların anlamları	25
Tablo 3.22. Deney - 5' in sonucunda elde edilen doğruluk tablosu	25
Tablo 3.23. Deney - 6' daki giriş ve çıkışların anlamları	27
Tablo 3.24. Deney - 6' nın sonucunda elde edilen doğruluk tablosu	27
Tablo 3.25. Deney - 7' deki giriş ve çıkışların anlamları	29
Tablo 3.26. Deney - 7' nin sonucunda elde edilen doğruluk tablosu	29
Tablo 3.27. Deney - 8' deki giriş ve çıkışların anlamları	31
Tablo 3.28. Deney - 8' in sonucunda elde edilen doğruluk tablosu	31
Tablo 3.29. Deney - 9' daki giriş ve çıkışların anlamları	33
Tablo 3.30. Deney - 9' un sonucunda elde edilen doğruluk tablosu	34
Tablo 3.31. Deney - 10' daki giriş ve çıkışların anlamları	36
Tablo 3.32. Deney - 10' un sonucunda elde edilen doğruluk tablosu	36
Tablo 3.33. Deney - 11' deki giriş ve çıkışların anlamları	38
Tablo 3.34. Deney - 11' in sonucunda elde edilen doğruluk tablosu	39
Tablo 3.35. Deney - 12' deki giriş ve çıkışların anlamları	40
Tablo 3.36. Deney - 12' nin sonucunda elde edilen doğruluk tablosu	41
Tablo 3.37. Deney - 13' de öğrenciye verilecek boş doğruluk tablosu	43
Tablo 3.38. Deney - 13' deki giriş ve çıkışların anlamları	43
Tablo 3.39. Deney - 13' ün sonucunda elde edilen doğruluk tablosu	44
Tablo 3.40. Deney - 14' deki giriş ve çıkışların anlamları	46
Tablo 3.41. Deney - 14' de öğrenciye verilecek boş doğruluk tablosu	46
Tablo 3.42. Deney - 14' ün sonucunda elde edilen doğruluk tablosu	47
Tablo 3.43. Deney - 15' deki giriş ve çıkışların anlamları	49
Tablo 3.44. Deney - 15' de öğrenciye verilecek boş doğruluk tablosu	50

Tablo 3.45. Deney - 15' in sonucunda elde edilen doğruluk tablosu	51
Tablo 3.46. Özel veya ile sinyal işleme	51
Tablo 4.01. Programlarda kullanılan tip isimleri	53
Tablo 6.01. MYLOGICS.PAS programı içinde tanımlanan elemanlar ve tip isimleri	75



1. GİRİŞ

Bu araştırmanın amacı, teknik okulların elektronik bölümlerinde ihtiyaç duyulan Sayısal Elektronik Deney Seti' nin piyasadaki diğer deney setlerinden daha ucuz ve daha pratik olarak elde edilebilmesini sağlamaktır.

Rashid' e (1989) göre, elektronik devre tasarımı devrenin çalışmasını göstermek için doğru yöntemleri gerektirir. Modern entegre devrelerin karmaşıklığı nedeniyle, bilgisayar destekli tasarım yapmak zorunlu hale gelmiştir. Aynı zamanda, bilgisayar destekli devre analizi ile devre hakkındaki bilgilerden laboratuvar şartlarında yapılan ölçümlerle saptanması hemen hemen mümkün olmayanlar da saptanabilir.

Deney seti tasarımı işlemi için bilgisayar ortamının tercih edilmesinin yukarıdaki neden dışında üç temel nedeni daha vardır. Bunlardan birincisi, bilgisayarda kullanılan bir deney seti hemen her okulda mevcut olan bilgisayar laboratuvarını kullanacağından ekstra bir fiziki alan gereksinimi oluşturmamasıdır. İkinci nedeni ise, deney seti ile birlikte kullanılması zorunlu olan ve deney sonuçlarını gözlemlemeye yarayan bazı pahalı cihazlara olan bağımlılığı ortadan kaldırmasıdır. Son nedeni ise, kullanıcının yapabileceği bir hatadan dolayı meydana gelebilecek fiziki bozunumları önlemesidir. Bilgisayar ile yapılan deneylerde, her hangi bir malzemenin bozulması veya yanması söz konusu değildir.

Deneylerin seçilmesi aşamasında, S.D.Ü. Burdur Meslek Yüksek Okulu Elektronik Programı öğretim görevlileri, S.D.Ü. Mühendislik-Mimarlık Fakültesi Elektronik-Haberleşme Mühendisliği Bölümü Başkanı ve danışman hocam Prof.Dr. Mustafa MERDAN ve yüksek lisans eğitimini beraber almakta olduğumuz Elektronik Mühendisi Abdullah ARISAN ile yapılan ikili görüşmeler etkili olmuştur.

Bilgisayar programının yazılacağı programlama dilinin seçiminde ise, eldeki teknik imkanlar göz önüne alınmış olup ikinci bölümde ayrıntılı bir şekilde bu dilin tanıtımı yapılmıştır.

2. PROGRAMLAMA DİLİ

Bilgisayar programı yazılırken Turbo Pascal programlama dili kullanılmıştır. Bunun en büyük nedeni, Turbo Pascal programlama dilinin, diğer programlama dillerinden farklı olarak programlama kavramının ve programcılığın öğretilmesi amacıyla tasarlanmış olmasıdır. Bu yüzden ilk zamanlarda daha çok akademik çalışmalarda FORTRAN' a karşı iyi bir alternatif olarak tercih edilmiştir. Bununla birlikte günümüzde, çok güçlü olan C programlama dilinin bir çok özelliğini daha basit olarak sunduğu için, her tip profesyonel amaçlı programın yazılabildiği bir dil haline gelmiştir.

Altan' a (1993) göre, Turbo Pascal programlama dili, yüksek seviyeli diller arasında yer alır. Bu özelliğiyle öğrenilmesi kolay bir programlama dilidir.

Turbo Pascal programlama dili yapısal bir programlama dilidir. Yapısal programlama dillerinin özelliği, bu diller kullanılarak yazılan programların mantıksal bütünlük gösteren bloklara ayrılabilmesidir. Bu yolla, uzun ve karmaşık programların bloklara ayrılarak daha kolay biçimde yazılabilmeleri mümkün olmaktadır.

Modern programlama dilleri yapısal olma eğilimindedir. Günümüzde, yapısal programlama dilleri kullanılarak yazılan programların hem yazılımlarının hem de analizlerinin daha kolay olduğu kabul edilmektedir. Böylece akademik çalışma için yazılan bir bilgisayar programı, daha sonra bu programı geliştirmek isteyen kişilere kolaylıklar sağlamaktadır.

Erdun' a (1993) göre, Turbo Pascal, grafik modunda da zengin bir komut kütüphanesine sahiptir. Buna rağmen, grafiklerle ilgili komutların çalışabilmesi için BGI uzantılı grafik sürücü programlarının, aktif sürücüde ve aktif alt dizinde bulunması gerekmektedir. Mesela, VGA ekrana sahip bir bilgisayar için EGAVGA.BGI dosyasının ve Hercules ekrana sahip bir bilgisayar için HERC.BGI dosyasının aktif alt dizinde bulunması gerekmektedir. Aksi takdirde, bilgisayar programı grafik çizim komutlarını tanımayacağından, program çalıştırılamayacaktır.

Schumann' a (1992) göre, son yıllarda profesyonel programlar arasında kullanım kolaylığı yönünden bir rekabet göze çarpıyor. Buna göre hangi program daha kolay kullanılabiliriyorsa veya hangi programda ekran düzenlemesi daha iyi ise o program öne çıkıyor. Bu durum Turbo Pascal programlama dilinde Turbo Vision' u gündeme getirmiştir.

Bir benzetme yapmak gerekirse programdaki hesaplama, veri tabanı gibi işlemleri programın etini, ekran menüleri, hata mesajları, edit işlemleri ise programın iskeletini oluşturur. Turbo Vision ekranında, tam profesyonel bir menü düzenlemesi ve edit işlemleri yardımı sağlıyor.

Projede, yukarıda belirtilen avantajlarından dolayı Turbo Pascal programlama dilinin Turbo Vision' u tercih edilmiştir. Yazılan bilgisayar programlarında, Turbo Vision' un Objects (Temel Nesne Tanımlamaları) ünitesinden yararlanılmıştır.

En önemli etken ise, Turbo Pascal programlama dilinin 5.5 versiyonundan itibaren sunduğu Nesne Tabanlı Programlama (OOP : Object Oriented Programming) tekniği ile ilgili desteğidir.

Banger' e (1993) göre, Nesne (Object), belleğin değiştirilebilen değerler ve / veya fonksiyonları (metodları) yerine getiren, isimlendirilmiş bir bölgedir. Yani nesneler, yapı olarak değişen ve fonksiyon içeren yapısal değişkenlerdir. Bu özelliğe *Nesnenin Paketleme Özelliği* adı verilir.

Bir sistem içerisinde, değişik karakterde bir çok nesne yer alır. Bu nesneler tamamen birbirlerinden farklı özellikler taşıyabildikleri gibi aynı veya benzer özellikleri de taşıyabilirler.

Örneğin: Bir büroda yazı makinası, telefon, fax, sekreter, memur ve müdür gibi nesneler yer alır. Esas itibarıyla bunların hepsi birer nesnedir. Her birinin işlevi farklıdır. Fakat ortak özellikleri de yok değildir. Sekreter, memur ve müdür birer insandır. Bu insan olma özelliğidir. Hepsı büroda, farklı miktarlarda olmakla beraber, bir ücret karşılığı çalışır. İşlerini yapmalarını istediğimizde hepsi bir iş yapar, fakat yaptıkları işler farklıdır. İstek aynı, davranışlar farklıdır. İşlerini yaparken kullandıkları bilgiler de aynı derecede birbirlerinden farklıdır.

Bunun yanı sıra ortaya çıkan diğer bir özellik de, paketlenen fonksiyonların nasıl bir işlev göstereceği belirtilmeksizin, sadece nasıl kullanılacağını belirtilmesi yapılabilmelidir. Buna da *Nesnenin Soyutlama Özelliği* adı verilir. Paketleme ve Soyutlama, bir nesneyi ayırt etmek için yeterli iki özelliktir.

Nesneler için söylenebilecek özelliklerden biri de türeme özelliğidir. Bir nesne tanımlanırken daha önceden belirlenmiş bir başka nesneyi kendisine taban olarak seçebilir. Bu durum, yeni tanımlanan nesnenin, kendisine taban seçtiği özelliklerini kullanma

hakkına karşılık gelir. Aynı zamanda yeni nesnenin taban nesne ile aynı özellikleri taşımasına sebep olur.

Yeni nesne kendisine yeni özellikler ekleyebileceği gibi devraldığı özellikleri de geliştirebilir, değiştirebilir. Bu özelliğe *Nesnenin Türemeye Özelliği*, özellikleri devralmaya da *Miras Alma* veya *Kalıtım* adı verilir.

Şu şekilde örneklenebilir :

Bir oto tamircisi, oto boyama (fırça ile), kaynak yapma (ark kaynağı ile) ve karbüratör ayarlama konusunda çalışıyor olsun. Bu bir oto tamircisi nesnesidir. Bu oto tamircisinin üç oğlu olduğunu varsayalım ve bunların her birini yetiştirdikten sonra, birer konuda uzmanlaşmalarını sağlamış olsun : Oto boyacısı (havalı fırça ile boyama), Oto kaynakçısı (oksijen ile kaynak yapma), Oto elektrikçisi (elektrik işleri).

Burada oto boyacısı, oto kaynakçısı ve oto elektrikçisi türeyen nesnelerdir. Bu türeyen nesneler taban nesnenin (yani, oto tamircisi babanın) özelliklerini göstereceklerdir. Oto boyayacaklar, kaynak yapacaklar ve karbüratör ayarlayacaklardır.

Oto boyacısından boya yapmasını istenildiğinde, havalı fırça ile boyama yapacaktır. Oysa oto tamircisi sadece fırça ile boyama yapmayı bilmekteydi. İkisinin de yaptığı iş boyama işidir. Fakat oto boyacısı, oto tamircisinden tüm özellikleri aynen alıp boyama işini kendisine özgü olarak değiştirdi.

Aynı şekilde oto kaynakçısı da oksijen kaynağı ile kaynak yapmaktadır. Oto tamircisi ise sadece ark kaynağı yapmayı bilmekteydi. Oto kaynakçısı aynı zamanda boya yapma yeteneğine de sahiptir. Bu yeteneği oto tamircisinden miras yolu ile gelmiştir ve fırça ile boya yapma yeteneğine sahiptir.

Üçüncü kolu oluşturan oto elektrikçisi ise farklı bir durum oluşturmaktadır. Oto tamircisi elektrik işleri yapamamasına karşın oto elektrikçisi bu işleri yapabilmektedir. Oto elektrikçisi aynı zamanda boyama, kaynak ve karbüratör ayarlama işlerini de oto tamircisinden öğrendiği kadarı ile yapabilmektedir. Tabii eğer istenirse oto elektrikçisinin örneğin, boyama yeteneği iptal edilerek, bu isteğin oluşması halinde hiç bir iş yapmaması da sağlanabilir. Öyle ya, her bilgi miras yolu ile babadan oğula aktarılamayabilir.

3. SAYISAL ELEKTRONİK DENEY SETİ

Teknik okulların elektronik ve bilgisayar ile ilgili bölümlerinde temel ders olarak Sayısal Elektronik dersleri okutulmaktadır. Bu dersin konusu, sadece elektrik var veya yok olmasına göre mantık kurularak elektronik devrelerin tasarlanmasıdır. Bu yüzden, diğer teknik derslerin çoğunda olduğu gibi, bu konuların öğrenciye aktarılabilmesi ancak iyi düzeyde yapılabilecek uygulamalarla sağlanır.

Teknik uygulamaların yapılabilmesi için piyasada çeşitli deney setleri mevcuttur. Fakat, bu setlerin pahalı olması ve kullanılırken pahalı yardımcı cihazlara ihtiyaç duymaları yüzünden daha ucuz yöntemlerin geliştirilmesine ihtiyaç duyulmuştur.

Günümüzde bilgisayar, tekniğin her kolunda uygulama alanı bulduğundan, teknik okulların bir çoğunda yetersiz de olsa bilgisayar laboratuvarları mevcuttur. Bu yüzden, teknik deneylerin bilgisayar üzerinde yapılabilmesi halinde ayrı ayrı laboratuvarlar ve oldukça yüklü sayılabilecek mali yüklerden teknik okullarımız kurtulabileceklerdir.

Bu düşüncelerin ışığında proje oluşturulmaya başlandı. İlk aşamada hangi uygulamaların yaptırılacağı saptandı. Bu saptamada eğitim - öğretime devam eden lise ve üniversite düzeyinde eğitim veren değerli hocalarımızdan fikir alındı. Bu fikirlerin ışığında, ilk aşamada İ.T.Ü. Elektrik - Elektronik Fakültesi Devreler ve Sistemler Ana Bilim Dalı tarafından yayınlamış olan deney kitabı ile S.D.Ü. Mühendislik - Mimarlık Fakültesi Elektronik Haberleşme Mühendisliği Bölümü' ne satın alınan deney setinin kitabında verilen deneyler içinden aşağıdaki deneylerin yaptırılabilmesi uygun görüldü :

1. Bazı lojik kapıların çalışmasının incelenmesi
2. Temel sequential devrelerin çalışmasının incelenmesi - 1 (JK Flip-Floplar)
3. Temel sequential devrelerin çalışmasının incelenmesi - 2 (RS Flip-Floplar)
4. Temel sequential devrelerin çalışmasının incelenmesi - 3 (D Flip-Floplar)
5. Asenkron 4 bit binary sayıcı
6. Asenkron 4 bit BCD sayıcı
7. Senkron 4 bit binary sayıcı
8. 7493 ile 4 bit binary sayıcı
9. Seri giriş paralel çıkışlı shift register
10. Paralel giriş seri çıkışlı shift register

11. Kodlayıcı devresinin çalışmasının incelenmesi
12. Kod çözücü devresinin çalışmasının incelenmesi
13. 4 bitlik iki sayının toplama devresi
14. 4 bitlik iki sayının çıkarma devresi
15. 8 bitlik iki sayının toplama ve çıkarma devresi

3.1. Deneylerde kullanılan temel malzemeler

Bu bölümde deneylerde kullanılacak olan elektronik elemanlarının tanıtımı yapılacaktır. Bu elemanların sayısal elektronikte nasıl kullanıldıkları, sembolleri ve doğruluk tabloları verilecektir.

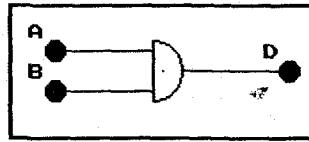
3.1.1. Sayısal mantık kapı entegreleri

Bu bölümde sayısal elektroniğin temelini teşkil eden malzemeler tanıtılacaktır. Bunlar sırası ile, VE, VEYA, DEĞİL, ÖZEL-VEYA, VE-DEĞİL, VEYA-DEĞİL kapı entegreleridir. Aşağıda sırası ile deney setinde bulunan bu elemanların şemaları ve doğruluk tabloları verilmiştir.

Bu bölümdeki tüm şekiller deney setinin kitabından ve doğruluk tabloları ise Barkana ve Taşlıca (1994) ile Yarcı' dan (1992) alınmıştır.

3.1.1.1. VE (AND) KAPISI :

Bu devre elemanı sadece girişlerinin hepsinden 1 geldiği zaman 1 çıkışı verir. Diğer durumlarda 0 çıkışı verir.



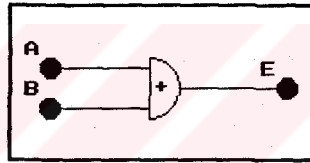
Şekil 3.1. AND kapısının şekli

Tablo 3.1. AND kapısının doğruluk tablosu.

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

3.1.1.2. VEYA (OR) KAPISI :

Bu devre elemanı sadece girişlerinin hepsinden 0 geldiği zaman 0 çıkışı verir. Diğer durumlarda 1 çıkışı verir.



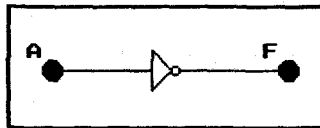
Şekil 3.2. OR kapısının şekli

Tablo 3.2. OR kapısının doğruluk tablosu.

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

3.1.1.3. DEĞİL (NOT) KAPISI :

Bu devre elemanı girişinde 1 varsa 0, 0 varsa 1 çıkışı verir.



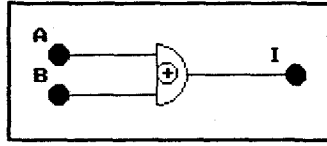
Şekil 3.3. NOT kapısının şekli

Tablo 3.3. NOT kapısının doğruluk tablosu.

A	F
1	0
0	1

3.1.1.4. ÖZEL-VEYA (XOR) KAPISI :

Bu devre elemanı girişleri aynı olduğu zaman 0, farklı olduğu zaman 1 çıkışını verir.



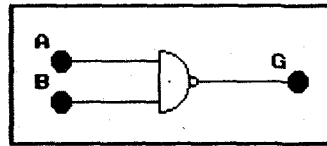
Şekil 3.4. XOR kapısının şekli

Tablo 3.4. XOR kapısının doğruluk tablosu.

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

3.1.1.5. VE-DEĞİL (NAND) KAPISI :

Bu devre elemanı sadece girişlerinin hepsinden 1 geldiği zaman 0 çıkışını verir. Diğer durumlarda 1 çıkışını verir.



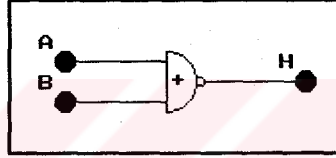
Şekil 3.5. NAND kapısının şekli

Tablo 3.5. NAND kapısının doğruluk tablosu.

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

3.1.1.6. VEYA-DEĞİL (NOR) KAPISI :

Bu devre elemanı sadece girişlerinin hepsinden 0 geldiği zaman 1 çıkışını verir. Diğer durumlarda 0 çıkışını verir.



Şekil 3.6. NOR kapısının şekli

Tablo 3.6. NOR kapısının doğruluk tablosu.

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

3.1.2. Sayısal mantık hafıza entegreleri

Bu bölümde sayısal elektroniğin temelinden bir adım sonrasını teşkil eden malzemeler tanıtılacaktır. Bunlar sırası ile, negatif kenar tetiklemeli JK flip-flop, pozitif kenar tetiklemeli JK flip-flop, negatif kenar tetiklemeli RS flip-flop, pozitif kenar tetiklemeli RS flip-flop, negatif kenar tetiklemeli D tipi flip-flop, ve pozitif kenar tetiklemeli D tipi flip-flop entegreleridir. Bu malzemelerin hepsi de AND-OR-NOT gibi

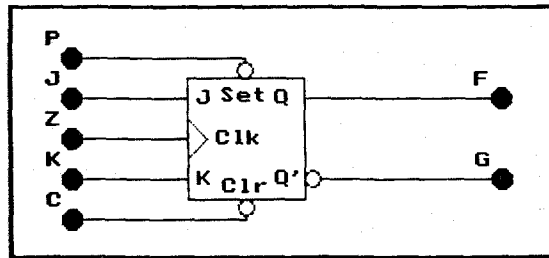
kapı entegreleri ile tasarlanabilen ve girişlerine uygulanan bilgiyi saklayabilirler (bir çeşit hafıza).

Bu malzemeler, genellikle pozitif ve negatif kenar tetiklemeli olurlar. Bunun anlamı, bu malzemelerin her birinde girişlerini kabul etmeleri için izin girişleri bulunur. Bu izin girişleri 1 iken 0 olduğu anda senkron girişlerini kontrol edenlerine negatif kenar tetiklemeli ve 0 iken 1 olduğu anda senkron girişlerini kontrol edenlerine ise pozitif kenar tetiklemeli denir. Asenkron girişler olan Clr ve Set ise, 0 yapıldıkları anda flip-flop' un içeriğini sırası ile 0 veya 1 yapmaktadırlar. Aşağıdaki doğruluk tablolarında belirtilen Q_{n+1} ve Q'_{n+1} çıkışların oluşacak olan değerlerini ve Q_n ve Q'_n çıkışların şimdiki durumlarını göstermektedir. Bunun yanı sıra, x hangi değerde olduğu önemli değil (Don't Care), 0-1 0' dan 1' e geçiş anını ve 1-0 ise 1' den 0' a geçiş anını temsil etmektedir. Aşağıda sırası ile deney setinde bulunan flip-flop' ların şemaları ve doğruluk tabloları verilmiştir.

Bu bölümde yer alan malzemelerin şekilleri ve doğruluk tabloları Barkana ve Taşlıca' dan (1994) alınmıştır.

3.1.2.1. Pozitif kenar tetiklemeli JK flip-flop :

Sadece tetikleme girişi olarak uygulanan Clk (Z) girişi 0 seviyesinden 1 seviyesine geçerken senkron girişleri olan J ve K' yı değerlendirmektedir. J' den 1 gelince çıkış 1, K' dan 1 gelince çıkış 0, her ikisinden de 0 gelince çıkışlarını korumakta ve her ikisinden de 1 gelince çıkışlarının tersini çıkış olarak vermektedir.



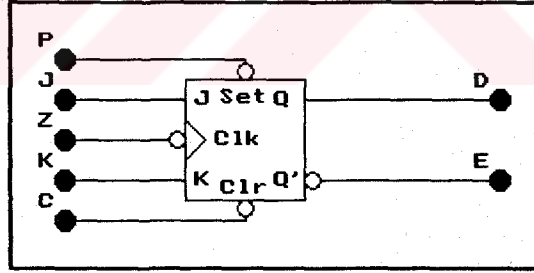
Şekil 3.7. Pozitif kenar tetiklemeli JK flip-flop' un şekli

Tablo 3.7. Pozitif kenar tetiklemeli JK flip-flop' un doğruluk tablosu.

Clk	J	K	Clr	Set	Q_{n+1}	Q'_{n+1}
x	x	x	0	1	0	1
x	x	x	1	0	1	0
x	x	x	0	0	1	1
x	0	0	1	1	Q_n	Q'_n
0-1	1	0	1	1	1	0
0-1	0	1	1	1	0	1
0-1	1	1	1	1	Q'_n	Q_n
0-1 hariç	x	x	1	1	Q_n	Q'_n

3.1.2.2. Negatif kenar tetiklemeli JK flip-flop :

Sadece tetikleme girişi olarak uygulanan Clk (Z) girişi 1 seviyesinden 0 seviyesine geçerken senkron girişleri olan J ve K' yı değerlendirmektedir. J' den 1 gelince çıkış 1, K' dan 1 gelince çıkış 0, her ikisinden de 0 gelince çıkışlarını korumakta ve her ikisinden de 1 gelince çıkışlarının tersini çıkış olarak vermektedir.



Şekil 3.8. Negatif kenar tetiklemeli JK flip-flop' un şekli

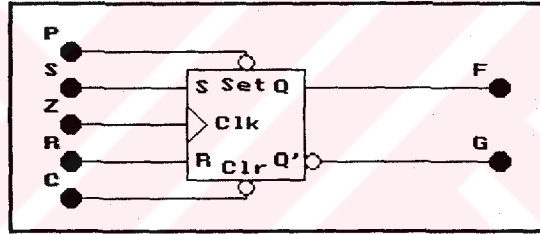
Tablo 3.8. Negatif kenar tetiklemeli JK flip-flop' un doğruluk tablosu.

Clk	J	K	Clr	Set	Q_{n+1}	Q'_{n+1}
x	x	x	0	1	0	1
x	x	x	1	0	1	0
x	x	x	0	0	1	1
x	0	0	1	1	Q_n	Q'_n

1-0	1	0	1	1	1	0
1-0	0	1	1	1	0	1
1-0	1	1	1	1	Q'_n	Q_n
1-0 hariç	x	x	1	1	Q_n	Q'_n

3.1.2.3. Pozitif kenar tetiklemeli RS flip-flop :

Sadece tetikleme girişi olarak uygulanan Clk (Z) girişi 0 seviyesinden 1 seviyesine geçerken senkron girişleri olan R ve S' yi değerlendirmektedir. S' den 1 gelince çıkış 1, R' den 1 gelince çıkış 0, her ikisinden de 0 gelince çıkışlarını korumakta ve her ikisinden de 1 gelince çıkışlarının ikisi birden 1 olmaktadır. Bu bir hatadır. Çünkü, Q çıkışındaki değerin tersi Q' çıkışından verilmelidir. Bu yüzden, RS flip-floplar kullanılırken her iki girişinin de 1 olmaması sağlanmalıdır. Aksi takdirde, istenmeyen sonuçlar ile karşılaşılabilir ve kurulan devreler yanlış çalışabilir.



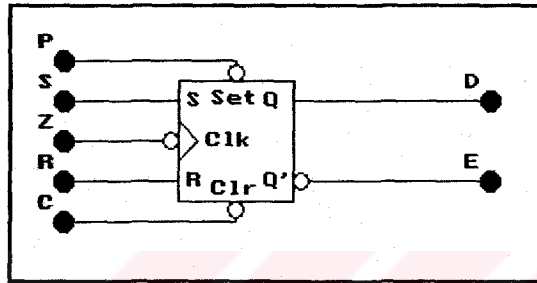
Şekil 3.9. Pozitif kenar tetiklemeli RS flip-flop' un şekli

Tablo 3.9. Pozitif kenar tetiklemeli RS flip-flop' un doğruluk tablosu.

Clk	R	S	Clr	Set	Q_{n+1}	Q'_{n+1}
x	x	x	0	1	0	1
x	x	x	1	0	1	0
x	x	x	0	0	1	1
x	0	0	1	1	Q_n	Q'_n
0-1	1	0	1	1	1	0
0-1	0	1	1	1	0	1
0-1	1	1	1	1	1	1
0-1 hariç	x	x	1	1	Q_n	Q'_n

3.1.2.4. Negatif kenar tetiklemeli RS flip-flop :

Sadece tetikleme girişi olarak uygulanan Clk (Z) girişi 1 seviyesinden 0 seviyesine geçerken senkron girişleri olan R ve S' yi değerlendirmektedir. S' den 1 gelince çıkış 1, R' den 1 gelince çıkış 0, her ikisinden de 0 gelince çıkışlarını korumakta ve her ikisinden de 1 gelince çıkışlarının ikisi birden 1 olmaktadır. Bu bir hatadır. Çünkü, Q çıkışındaki değerin tersi Q' çıkışından verilmelidir. Bu yüzden, RS flip-floplar kullanılırken her iki girişinin de 1 olmaması sağlanmalıdır. Aksi takdirde, istenmeyen sonuçlar ile karşılaşılabilir ve kurulan devreler yanlış çalışabilir.



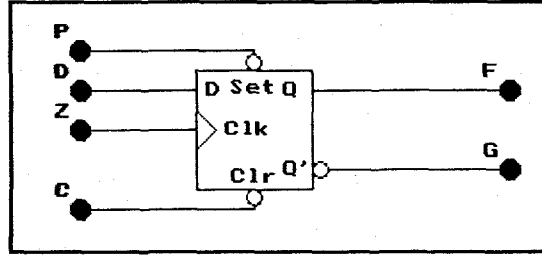
Şekil 3.10. Negatif kenar tetiklemeli RS flip-flop' un şekli

Tablo 3.10. Negatif kenar tetiklemeli RS flip-flop' un doğruluk tablosu.

Clk	R	S	Clr	Set	Q_{n+1}	Q'_{n+1}
x	x	x	0	1	0	1
x	x	x	1	0	1	0
x	x	x	0	0	1	1
x	0	0	1	1	Q_n	Q'_n
1-0	1	0	1	1	1	0
1-0	0	1	1	1	0	1
1-0	1	1	1	1	1	1
1-0 hariç	x	x	1	1	Q_n	Q'_n

3.1.2.5. Pozitif kenar tetiklemeli D tipi flip-flop :

Sadece tetikleme girişi olarak uygulanan Clk (Z) girişi 0 seviyesinden 1 seviyesine geçerken senkron girişi olan D' den gelen değeri aynen Q çıkışına vermektedir.



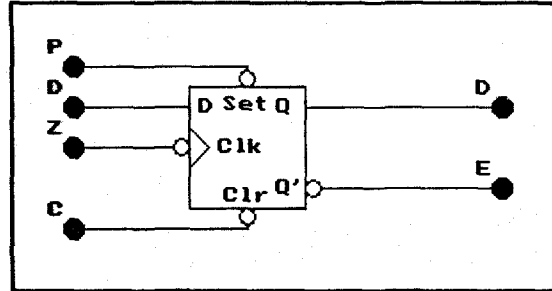
Şekil 3.11. Pozitif kenar tetiklemeli D tipi flip-flop' un şekli

Tablo 3.11. Pozitif kenar tetiklemeli D tipi flip-flop' un doğruluk tablosu.

Clk	D	Clr	Set	Q_{n+1}	Q'_{n+1}
x	x	0	1	0	1
x	x	1	0	1	0
x	x	0	0	1	1
x	0	1	1	Q_n	Q'_n
0-1	1	1	1	1	0
0-1	0	1	1	0	1
0-1 hariç	x	1	1	Q_n	Q'_n

3.1.2.6. Negatif kenar tetiklemeli D tipi flip-flop :

Sadece tetikleme girişi olarak uygulanan Clk (Z) girişi 1 seviyesinden 0 seviyesine geçerken senkron girişi olan D' den gelen değeri aynen Q çıkışına vermektedir.



Şekil 3.12. Negatif kenar tetiklemeli D tipi flip-flop' un şekli

Tablo 3.12. Negatif kenar tetiklemeli D tipi flip-flop' un doğruluk tablosu.

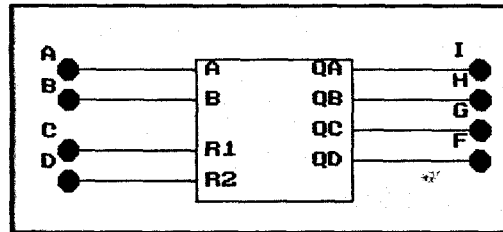
Clk	D	Clr	Set	Q_{n+1}	Q'_{n+1}
x	x	0	1	0	1
x	x	1	0	1	0
x	x	0	0	1	1
x	0	1	1	Q_n	Q'_n
1-0	1	1	1	1	0
1-0	0	1	1	0	1
1-0 hariç	x	1	1	Q_n	Q'_n

3.1.3. Sayısal mantıkta kullanılan bazı kompleks entegreler

Bu bölümde şu ana kadar kullanılan malzemeler kullanılarak yapılabilecek daha karmaşık (kompleks) devrelerin tek bir malzeme haline getirilmiş olanlarına iki örnek verilecektir. Bunlar 4 bitlik yukarı sayıcı ve 4 bitlik iki sayının toplamasını yapabilen entegre devre elemanlarıdır.

3.1.3.1. Sayıcı entegresi : 7493

İ.T.Ü. Elektrik - Elektronik Fakültesi Devreler ve Sistemler Ana Bilim Dalı' nın deney kitabına göre, bu entegre devre elemanı TTL olarak gelen bir sinyali 4 bit binary olarak sayabilir. Bu malzeme maksimum 16 megahertze kadar olan TTL sinyalleri sayabilir. Lojik 0 olarak en fazla 0.7 volt ve lojik 1 olarak en az 2 volt uygulamayı gerektirir. Teközgen'e (1992) göre bu entegrenin şekli aşağıdaki gibidir.



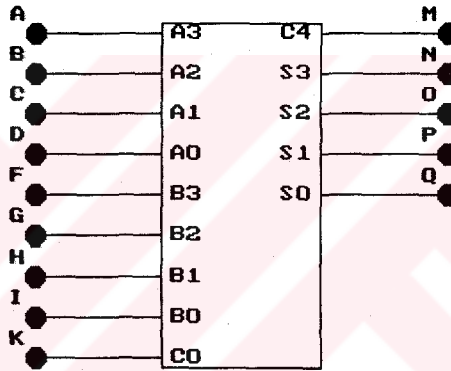
Şekil 3.13. 7493 sayıcı entegresinin şekli

7493 entegresi iki kısımdan oluşur. Bunlardan birincisi A girişindeki sinyali Mod-2 olarak sayar ve QA çıkışından verir. İkincisi ise, B girişindeki sinyali Mod-8 olarak sayar

ve QD-QC-QB çıkışlarından verir. Eğer, Mod-8' den (0-7 arası) daha büyük sayıcı yapılmak isteniyorsa sinyal A' dan uygulanmalı ve QB çıkışı ile A girişi kısa devre edilmelidir. Buradaki R1 ve R2 girişlerinin her ikisinden birden 1 geldiği anda, sayıcı çıkışı sıfırlanmaktadır.

3.1.3.2. Toplayıcı entegresi : 7483

Bu entegre devre elemanı TTL olarak gelen iki tane 4 bitlik binary sayıyı toplayıp sonucu 4 bitlik binary olarak vermeye yarar. Entegrenin özelliği, bir önceki basamaktan elde girişinin (C0) olması ve toplamın sonucunda oluşacak eldeyi çok hızlı hesaplayıp elde çıkışından (C4) verebilmesidir. Teközgen' e (1992) göre bu entegrenin şekli aşağıdaki gibidir.



Şekil 3.14. 7483 toplayıcı entegresinin şekli

Bu entegrenin ABCD girişlerine uygulanan birinci sayı ile FGHI girişlerine uygulanan ikinci sayı ve K elde girişi toplanarak NOPQ sonuç çıkışından verilir. Eğer sonuçta elde oluşmuşsa, M çıkışından verilir.

$MNOPQ = ABCD + FGHI + K$ ifadesi entegrenin genel çıkış tanımlama ifadesidir.

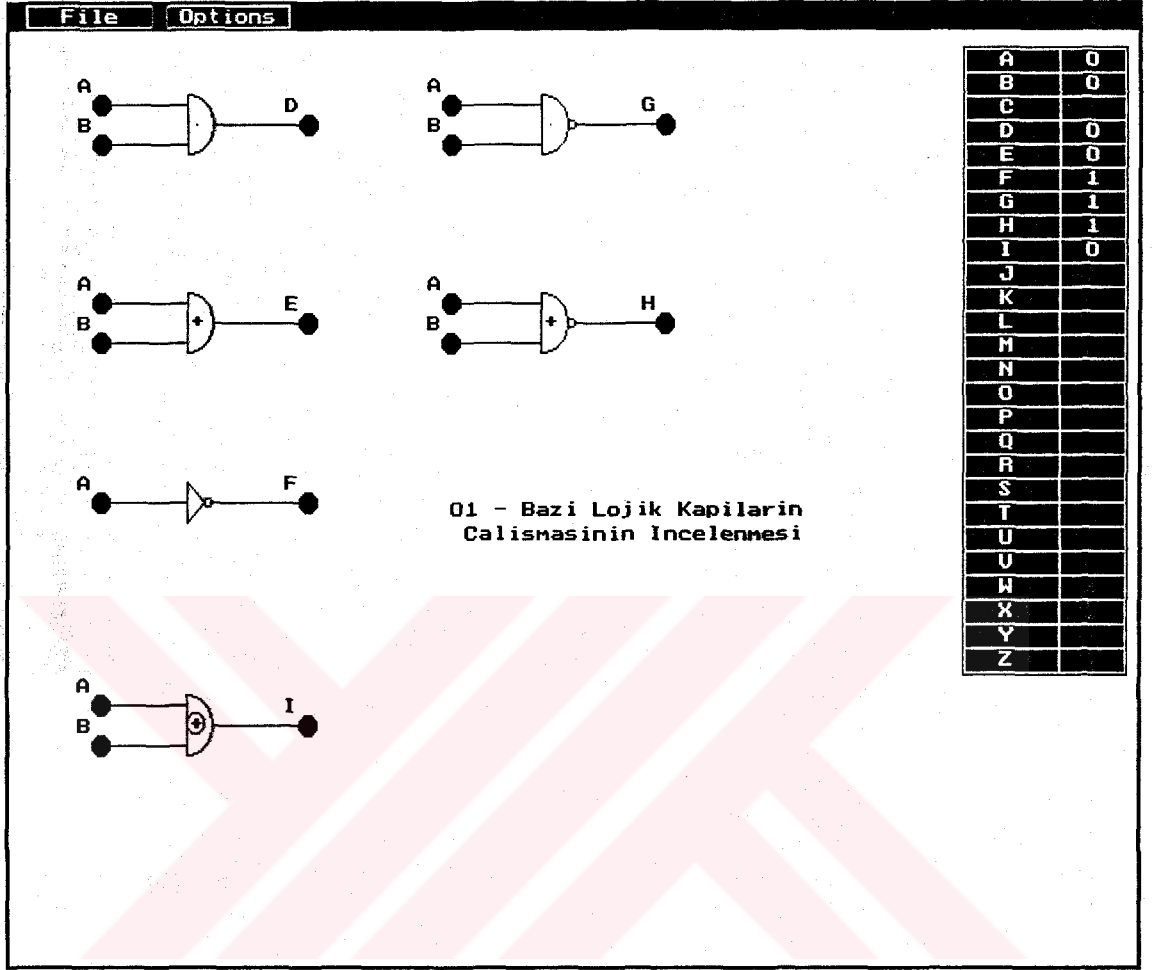
3.2. Deneyler

Bu bölümde, tasarlanan deney setinde yer alan deneylerin tanıtımları yapılacaktır. Bu deneyler deney numaralarına göre sıralıdır :

1. Bazı lojik kapıların çalışmasının incelenmesi
2. Temel sequential devrelerin çalışmasının incelenmesi - 1 (JK Flip-Floplar)
3. Temel sequential devrelerin çalışmasının incelenmesi - 2 (RS Flip-Floplar)
4. Temel sequential devrelerin çalışmasının incelenmesi - 3 (D Flip-Floplar)
5. Asenkron 4 bit binary sayıcı
6. Asenkron 4 bit BCD sayıcı
7. Senkron 4 bit binary sayıcı
8. 7493 ile 4 bit binary sayıcı
9. Seri giriş paralel çıkışlı shift register
10. Paralel giriş seri çıkışlı shift register
11. Kodlayıcı devresinin çalışmasının incelenmesi
12. Kod çözücü devresinin çalışmasının incelenmesi
13. 4 bitlik iki sayının toplama devresi
14. 4 bitlik iki sayının çıkarma devresi
15. 8 bitlik iki sayının toplama ve çıkarma devresi

3.2.1. Bazı lojik kapıların çalışmasının incelenmesi

1 nolu deneyi seçince, aşağıdaki gibi bir ekran görüntüsü elde edilir :



Şekil 3.15. Deney - 1' in ekran görüntüsü

Bu deneyde öğrenciye temel lojik kapı devrelerinin tanıtımı yapılacaktır. Bu deneyde AND, OR, NOT, XOR, NAND ve NOR kapı devreleri bulunmaktadır. A ve B genel girişler olarak belirlenmiştir. Burada belirlenen çıkışlar ve ifadeleri ise aşağıdaki gibidir :

Tablo 3.13. Deney - 1' deki çıkış ifadelerinin fonksiyonları

ÇIKIŞLAR	FONKSİYONLARI	LOJİK GÖSTERİMİ
D	A AND B	$A \cdot B$
E	A OR B	$A + B$
F	NOT A	A'
G	A NAND B	$(A \cdot B)'$
H	A NOR B	$(A + B)'$
I	A XOR B	$A' \cdot B + A \cdot B'$

Bu deneyde öğrenciye JK flip-flopların tanıtımı yapılacaktır. Bu deneyde negatif ve pozitif kenar tetiklemeli JK flip-flop' lar bulunmaktadır. C, J, K, P ve Z genel girişler olarak belirlenmiştir.

Öğrenciye girişlere verilen çeşitli değerlere göre (hangisinin değeri değiştirilecek ise, o değişkeni gösteren harfe basılmalıdır) D, E, F, G çıkış değişkenlerinin alacakları değerleri yorumlayarak doğruluk tablosu şeklinde sunması istenebilir.

Burada belirlenen girişler, çıkışlar ve anlamları ise aşağıdaki gibidir :

Tablo 3.15. Deney - 2' deki giriş ve çıkışların anlamları

İFADELER	ANLAMLARI
P	flip-flop' ların asıl çıkışlarının 1 olmasını sağlayan asenkron giriş
C	flip-flop' ların asıl çıkışlarının 0 olmasını sağlayan asenkron giriş
J	flip-flop' ların asıl çıkışlarının 1 olmasını sağlayan senkron giriş
K	flip-flop' ların asıl çıkışlarının 0 olmasını sağlayan senkron giriş
Z	flip-flop' ların girişlerindeki senkronluğu sağlayan giriş
D	negatif kenar tetiklemeli flip-flop' un asıl çıkışı (Q)
E	negatif kenar tetiklemeli flip-flop' un ters çıkışı (Q')
F	pozitif kenar tetiklemeli flip-flop' un asıl çıkışı (Q)
G	pozitif kenar tetiklemeli flip-flop' un ters çıkışı (Q')

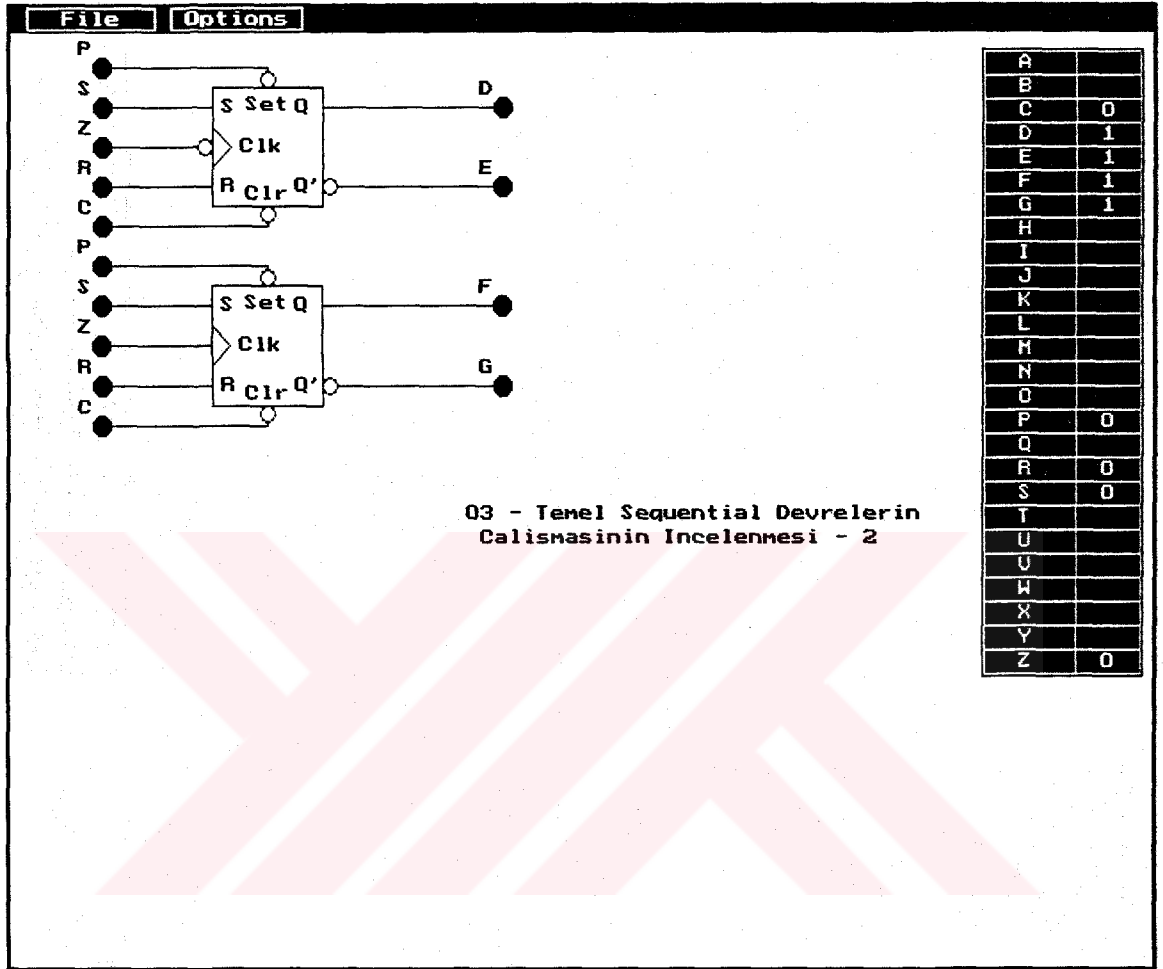
Bu doğruluk tablosu aşağıdaki tabloda verildiği gibi olmalıdır.

Tablo 3.16. Deney - 2' nin sonucunda elde edilen doğruluk tablosu.

Z	P	C	J	K	D	E	F	G
x	0	0	x	x	1	1	1	1
x	1	0	x	x	0	1	0	1
x	0	1	x	x	1	0	1	0
1-0	1	1	0	0	D	E	F	G
1-0	1	1	0	1	0	1	F	G
1-0	1	1	1	0	1	0	F	G
1-0	1	1	1	1	E	D	F	G
0-1	1	1	0	0	D	E	F	G
0-1	1	1	0	1	D	E	0	1
0-1	1	1	1	0	D	E	1	0
0-1	1	1	1	1	D	E	G	F
diğer	1	1	x	x	D	E	F	G

3.2.3. Temel sequential devrelerin çalışmasının incelenmesi - 2 (RS Flip-Floplar)

3 nolu deneyi seçince, aşağıdaki gibi bir ekran görüntüsü elde edilir :



Şekil 3.17. Deney - 3' ün ekran görüntüsü

Bu deneyde öğrenciye RS flip-flop'ların tanıtımı yapılacaktır. Bu deneyde negatif ve pozitif kenar tetiklemeli RS flip-flop' lar bulunmaktadır. C, S, R, P ve Z genel girişler olarak belirlenmiştir.

Burada belirlenen girişler, çıkışlar ve anlamları ise aşağıdaki gibidir :

Tablo 3.17. Deney - 3' deki giriş ve çıkışların anlamları

İFADELER	ANLAMLARI
P	flip-flop' ların asıl çıkışlarının 1 olmasını sağlayan asenkron giriş
C	flip-flop' ların asıl çıkışlarının 0 olmasını sağlayan asenkron giriş
S	flip-flop' ların asıl çıkışlarının 1 olmasını sağlayan senkron giriş

R	flip-flop' ların asıl çıkışlarının 0 olmasını sağlayan senkron giriş
Z	flip-flop' ların girişlerindeki senkronluğu sağlayan giriş
D	negatif kenar tetiklemeli flip-flop' un asıl çıkışı (Q)
E	negatif kenar tetiklemeli flip-flop' un ters çıkışı (Q')
F	pozitif kenar tetiklemeli flip-flop' un asıl çıkışı (Q)
G	pozitif kenar tetiklemeli flip-flop' un ters çıkışı (Q')

Öğrenciye girişlere verilen çeşitli değerlere göre (hangisinin değeri değiştirilecek ise, o değişkeni gösteren harfe basılmalıdır) D, E, F, G çıkış değişkenlerinin alacakları değerleri yorumlayarak doğruluk tablosu şeklinde sunması istenebilir.

Bu doğruluk tablosu aşağıdaki tabloda verildiği gibi olmalıdır.

Tablo 3.18. Deney - 3' ün sonucunda elde edilen doğruluk tablosu.

Z	P	C	S	R	D	E	F	G
x	0	0	x	x	1	1	1	1
x	1	0	x	x	0	1	0	1
x	0	1	x	x	1	0	1	0
1-0	1	1	0	0	D	E	F	G
1-0	1	1	0	1	0	1	F	G
1-0	1	1	1	0	1	0	F	G
1-0	1	1	1	1	E	D	F	G
0-1	1	1	0	0	D	E	F	G
0-1	1	1	0	1	D	E	0	1
0-1	1	1	1	0	D	E	1	0
0-1	1	1	1	1	D	E	G	F
diğer	1	1	x	x	D	E	F	G

3.2.4. Temel sequential devrelerin çalışmasının incelenmesi - 3 (D Tipi Flip-Floplar)

Bu deneyde öğrenciye D tipi flip-flopların tanıtımı yapılacaktır. Bu deneyde negatif ve pozitif kenar tetiklemeli D tipi flip-flop' lar bulunmaktadır. C, A, P ve Z genel girişler olarak belirlenmiştir.

Burada belirlenen girişler, çıkışlar ve anlamları ise aşağıdaki gibidir :

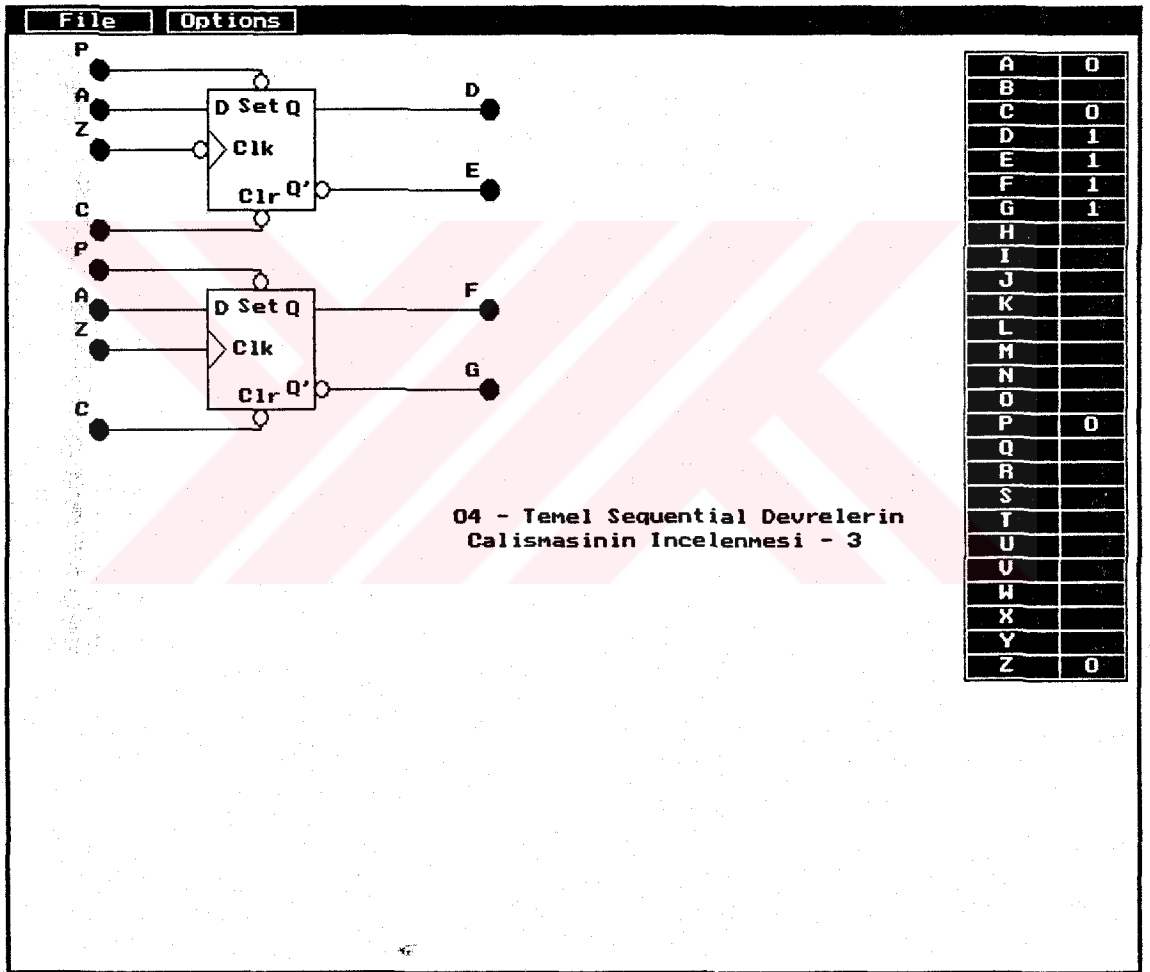
Tablo 3.19. Deney - 4' deki giriş ve çıkışların anlamları

İFADELER	ANLAMLARI
P	flip-flop' ların asıl çıkışlarının 1 olmasını sağlayan asenkron giriş
C	flip-flop' ların asıl çıkışlarının 0 olmasını sağlayan asenkron giriş
A	flip-flop' ların asıl çıkışlarına aktarılacak değer girilmesini sağlayan senkron giriş
Z	flip-flop' ların girişlerindeki senkronluğu sağlayan giriş

D	negatif kenar tetiklemeli flip-flop' un asıl çıkışı (Q)
E	negatif kenar tetiklemeli flip-flop' un ters çıkışı (Q')
F	pozitif kenar tetiklemeli flip-flop' un asıl çıkışı (Q)
G	pozitif kenar tetiklemeli flip-flop' un ters çıkışı (Q')

Öğrenciye girişlere verilen çeşitli değerlere göre (hangisinin değeri değiştirilecek ise, o değişkeni gösteren harfe basılmalıdır) D, E, F, G çıkış değişkenlerinin alacakları değerleri yorumlayarak doğruluk tablosu şeklinde sunması istenebilir.

4 nolu deneyi seçince, aşağıdaki gibi bir ekran görüntüsü elde edilir :



Şekil 3.18. Deney - 4' ün ekran görüntüsü

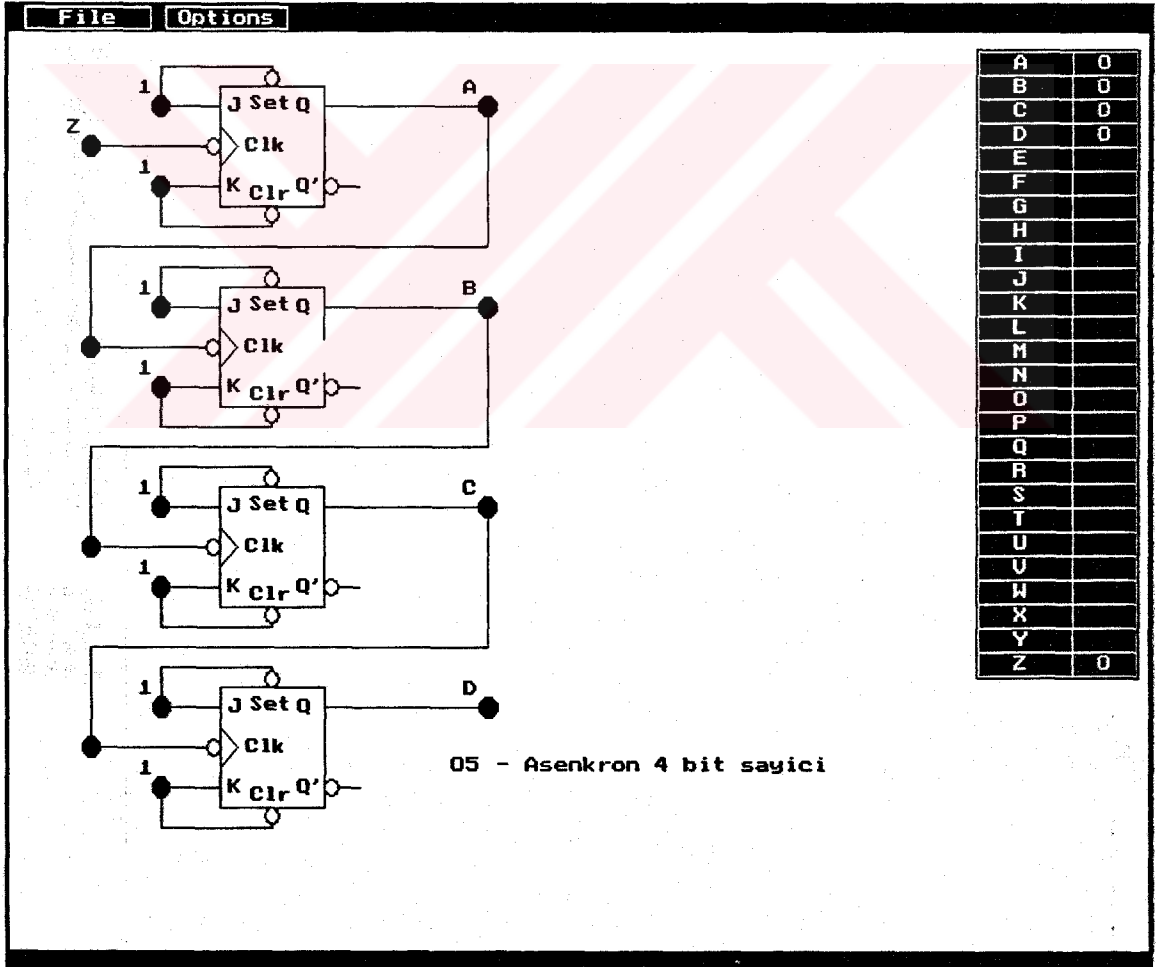
Bu doğruluk tablosu aşağıdaki tabloda verildiği gibi olmalıdır.

Tablo 3.20. Deney - 4' ün sonucunda elde edilen doğruluk tablosu.

Z	P	C	A	D	E	F	G
x	0	0	x	1	1	1	1
x	1	0	x	0	1	0	1
x	0	1	x	1	0	1	0
1-0	1	1	0	0	1	F	G
1-0	1	1	1	1	0	F	G
0-1	1	1	0	D	E	0	1
0-1	1	1	1	D	E	1	0
diğer	1	1	x	D	E	F	G

3.2.5. Asenkron 4 bit binary sayıcı

5 nolu deneyi seçince, aşağıdaki gibi bir ekran görüntüsü elde edilir :



Şekil 3.19. Deney - 5' in ekran görüntüsü

Bu deneyde öğrenciye negatif kenar tetiklemeli flip-floplar ile tasarımı yapılmış ve kurulmuş dört bitlik ikilik asenkron sayıcının tanıtımı yapılacaktır. Z genel girişinden gelen darbeler sayılarak DCBA çıkışlarından verilmektedir.

Burada belirlenen girişler, çıkışlar ve anlamları ise aşağıdaki gibidir :

Tablo 3.21. Deney - 5' deki giriş ve çıkışların anlamları

İFADELER	ANLAMLARI
D	sayıcı çıkışının en yüksek değerlikli biti (MSB)
C	sayıcı çıkışının en yüksek değerlikli bitinden sonra gelen biti
B	sayıcı çıkışının en düşük değerlikli bitinden önce gelen biti
A	sayıcı çıkışının en düşük değerlikli biti (LSB)
Z	sayılacak darbelerin girilmesini sağlayan giriş

Öğrenciye girişten verilen darbelerin sayısına (0 - 19 arasında) göre (Z değişkeninin değeri elle değiştirilecek ise Z harfine basılmalıdır veya menüden Automatic komutu verilmelidir) D, C, B, A çıkış değişkenlerinin alacakları değerleri yorumlayarak doğruluk tablosu şeklinde sunması istenebilir. Menüden *Automatic* komutu verilmiş ise, Z değişkeninin değeri otomatik olarak değiştirileceğinden, DCBA çıkışlarının aldığı değerler sıra ile klavyeden komut vermeye gerek kalmadan görülebilir.

Bu doğruluk tablosu aşağıdaki tabloda verildiği gibi olmalıdır.

Tablo 3.22. Deney - 5' in sonucunda elde edilen doğruluk tablosu.

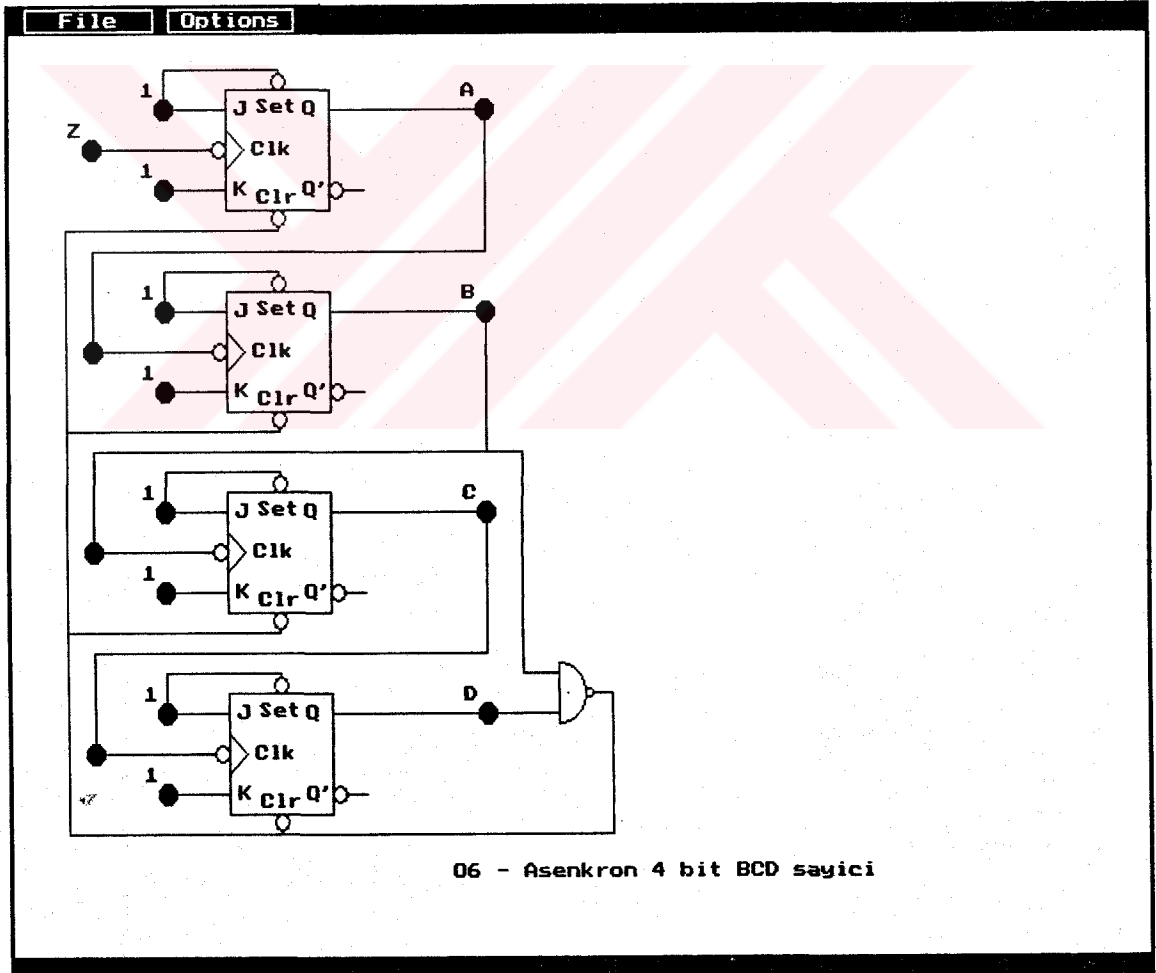
Z	D	C	B	A
0	0	0	0	0
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0
5	0	1	0	1
6	0	1	1	0
7	0	1	1	1
8	1	0	0	0
9	1	0	0	1
10	1	0	1	0
11	1	0	1	1
12	1	1	0	0
13	1	1	0	1
14	1	1	1	0

15	1	1	1	1
16	0	0	0	0
17	0	0	0	1
18	0	0	1	0
19	0	0	1	1

Burada dikkat edilecek nokta, sayıcı MOD 16 SAYICI olarak çalışmaktadır. Gelen darbedenin mod 16' ya göre elde karşılığı 4 bit olan ikilik karşılığı DCBA sırasında verilir (Yani, 16 = 0, 19 = 3 gibi).

3.2.6. Asenkron 4 bit BCD sayıcı

6 nolu deneyi seçince, aşağıdaki gibi bir ekran görüntüsü elde edilir :



Şekil 3.20. Deney - 6' nın ekran görüntüsü

Bu deneyde öğrenciye negatif kenar tetiklemeli JK flip-floplar ile tasarımı yapılmış ve kurulmuş dört bitlik ikilik asenkron sayıcının tanıtımı yapılacaktır. Z genel girişinden gelen darbeler sayılarak BCD (İkilik Kodlanmış Onluk) olarak DCBA çıkışlarından verilmektedir.

Burada belirlenen girişler, çıkışlar ve anlamları ise aşağıdaki gibidir :

Tablo 3.23. Deney - 6' daki giriş ve çıkışların anlamları

İFADELER	ANLAMLARI
D	sayıcı çıkışının en yüksek değerlikli biti (MSB)
C	sayıcı çıkışının en yüksek değerlikli bitinden sonra gelen biti
B	sayıcı çıkışının en düşük değerlikli bitinden önce gelen biti
A	sayıcı çıkışının en düşük değerlikli biti (LSB)
Z	sayılacak darbelerin girilmesini sağlayan giriş

Öğrenciye girişten verilen darbelerin sayısına (0 - 19 arasında) göre (Z değişkeninin değeri elle değiştirilecek ise Z harfine basılmalıdır veya menüden Automatic komutu verilmelidir) D, C, B, A çıkış değişkenlerinin alacakları değerleri yorumlayarak doğruluk tablosu şeklinde sunması istenebilir. Menüden *Automatic* komutu verilmiş ise, Z değişkeninin değeri otomatik olarak değiştirileceğinden, DCBA çıkışlarının aldığı değerler sıra ile klavyeden komut vermeye gerek kalmadan görülebilir.

Bu doğruluk tablosu aşağıdaki tabloda verildiği gibi olmalıdır.

Tablo 3.24. Deney - 6' nın sonucunda elde edilen doğruluk tablosu.

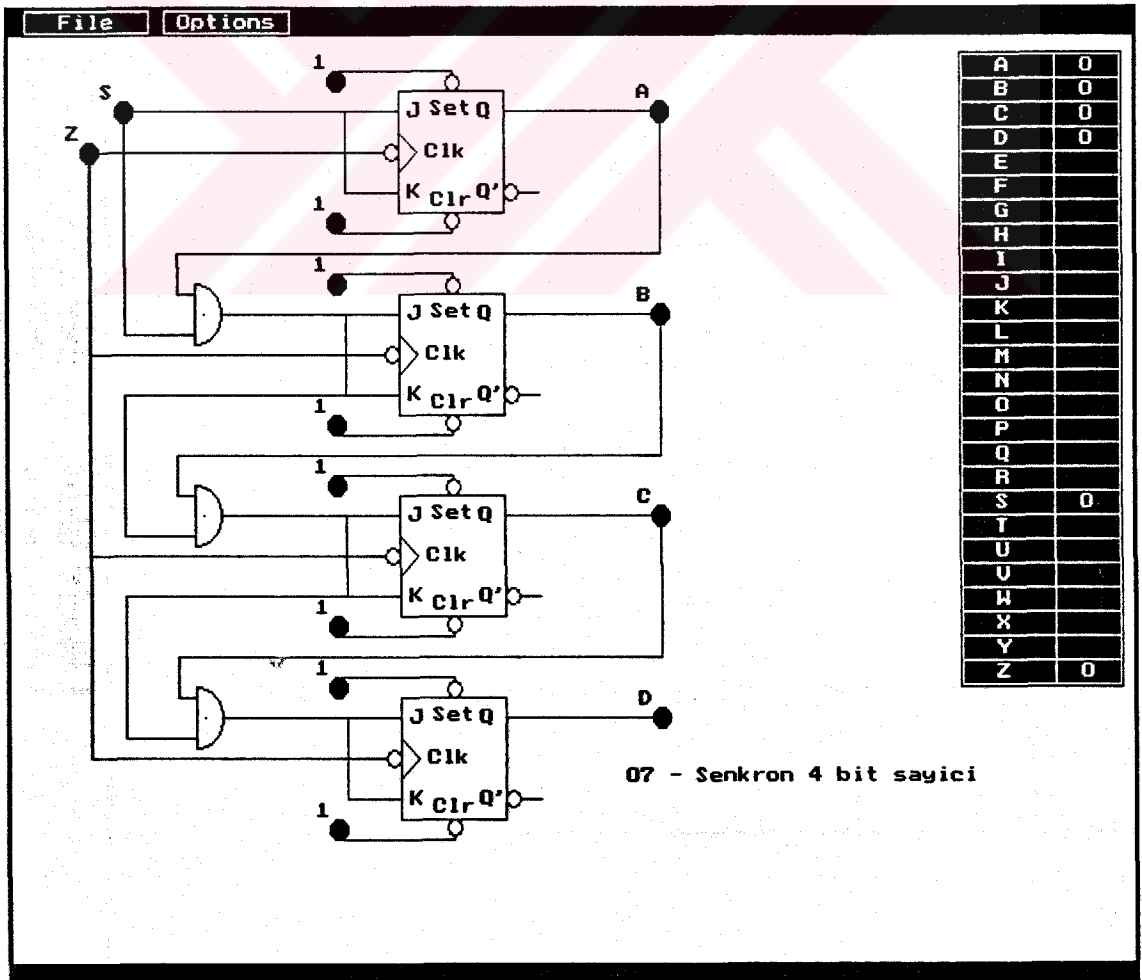
Z	D	C	B	A
0	0	0	0	0
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0
5	0	1	0	1
6	0	1	1	0
7	0	1	1	1
8	1	0	0	0
9	1	0	0	1
10	0	0	0	0
11	0	0	0	1
12	0	0	1	0

13	0	0	1	1
14	0	1	0	0
15	0	1	0	1
16	0	1	1	0
17	0	1	1	1
18	1	0	0	0
19	1	0	0	1

Burada dikkat edilecek nokta, sayıcı MOD 10 SAYICI olarak çalışmaktadır. Gelen darbedenin mod 10' a göre elde karşılığı 4 bit olan ikilik karşılığı DCBA sırasında verilir (Yani, 10 = 0, 19 = 9 gibi). Burada çıkış 0 ile 9 arasında ikilik kodlanmış olan çıkışı DCBA' dan verilir.

3.2.7. Senkron 4 bit binary sayıcı

7 nolu deneyi seçince, aşağıdaki gibi bir ekran görüntüsü elde edilir :



Şekil 3.21. Deney - 7' nin ekran görüntüsü

Bu deneyde öğrenciye negatif kenar tetiklemeli JK flip-floplar ile tasarımı yapılmış ve kurulmuş dört bitlik ikilik senkron sayıcının tanıtımı yapılacaktır. Z genel girişinden gelen darbeler sayılarak ikilik olarak DCBA çıkışlarından verilmektedir.

Burada belirlenen girişler, çıkışlar ve anlamları ise aşağıdaki gibidir :

Tablo 3.25. Deney - 7' deki giriş ve çıkışların anlamları

İFADELER	ANLAMLARI
D	sayıcı çıkışının en yüksek değerlikli biti (MSB)
C	sayıcı çıkışının en yüksek değerlikli bitinden sonra gelen biti
B	sayıcı çıkışının en düşük değerlikli bitinden önce gelen biti
A	sayıcı çıkışının en düşük değerlikli biti (LSB)
Z	sayılacak darbelerin girilmesini sağlayan giriş

Öğrenciye girişten verilen darbelerin sayısına (0 - 19 arasında) göre (Z değişkeninin değeri elle değiştirilecek ise Z harfine basılmalıdır veya menüden Automatic komutu verilmelidir) D, C, B, A çıkış değişkenlerinin alacakları değerleri yorumlayarak doğruluk tablosu şeklinde sunması istenebilir. Menüden *Automatic* komutu verilmiş ise, Z değişkeninin değeri otomatik olarak değiştirileceğinden, DCBA çıkışlarının aldığı değerler sıra ile klavyeden komut vermeye gerek kalmadan görülebilir.

Bu doğruluk tablosu aşağıdaki tabloda verildiği gibi olmalıdır.

Tablo 3.26. Deney - 7' nin sonucunda elde edilen doğruluk tablosu.

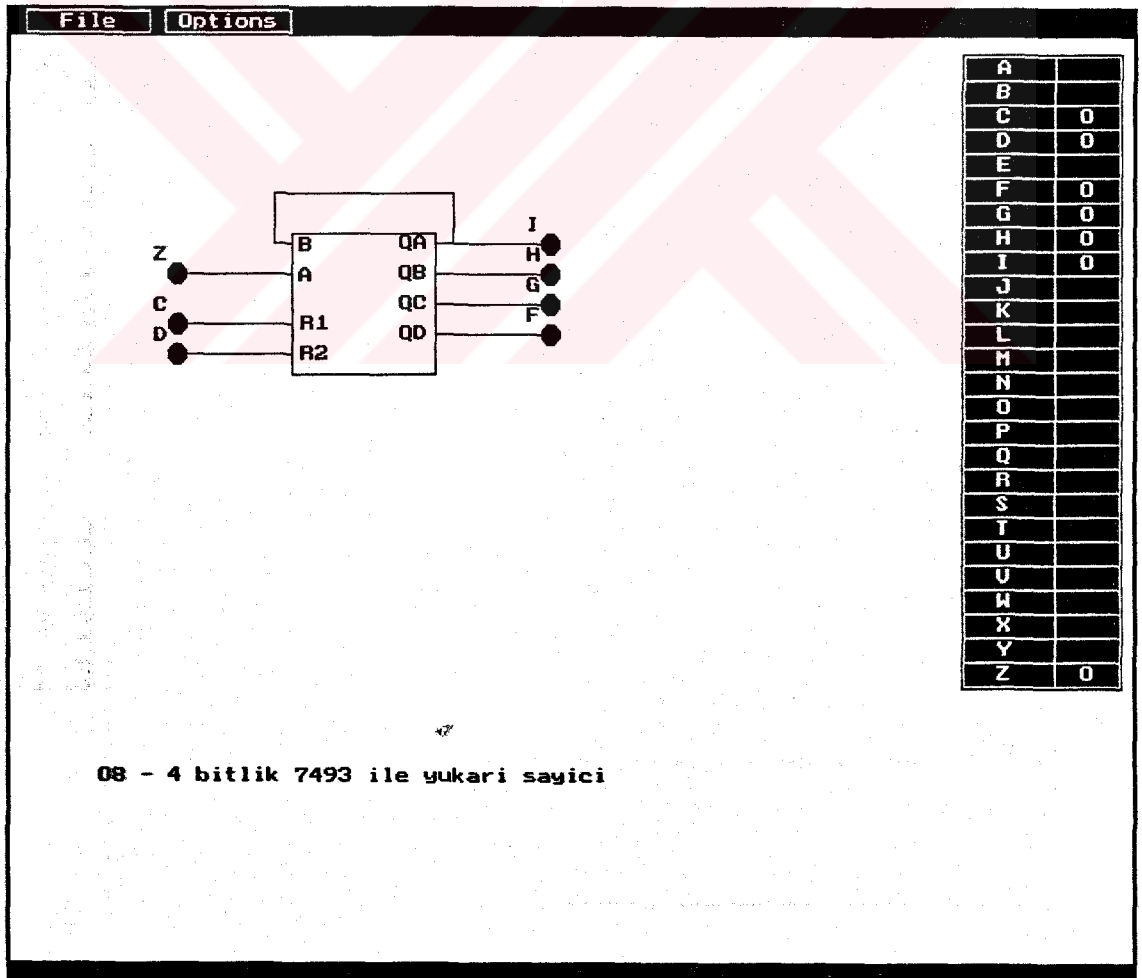
Z	D	C	B	A
0	0	0	0	0
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0
5	0	1	0	1
6	0	1	1	0
7	0	1	1	1
8	1	0	0	0
9	1	0	0	1
10	1	0	1	0
11	1	0	1	1

12	1	1	0	0
13	1	1	0	1
14	1	1	1	0
15	1	1	1	1
16	0	0	0	0
17	0	0	0	1
18	0	0	1	0
19	0	0	1	1

Burada dikkat edilecek nokta, sayıcı MOD 16 SAYICI olarak çalışmaktadır. Gelen darbedenin mod 16' ya göre elde karşılığı 4 bit olan ikilik karşılığı DCBA sırasında verilir (Yani, 16 = 0, 19 = 3 gibi). Burada çıkış 0 ile 15 arasında ikilik kodlanmış olan çıkışı DCBA' dan verilir.

3.2.8. 7493 ile 4 bit binary sayıcı

8 nolu deneyi seçince, aşağıdaki gibi bir ekran görüntüsü elde edilir :



Şekil 3.22. Deney - 8' in ekran görüntüsü

Bu deneyde öğrenciye 7493 entegresi ile gerçekleştirilmiş dört bitlik ikilik asenkron sayıcının tanıtımı yapılacaktır. Z genel girişinden gelen darbeler sayılarak ikilik kodlanmış olarak DCBA çıkışlarından verilmektedir.

Burada belirlenen girişler, çıkışlar ve anlamları ise aşağıdaki gibidir :

Tablo 3.27. Deney - 8' deki giriş ve çıkışların anlamları

İFADELER	ANLAMLARI
D	sayıcı çıkışının en yüksek değerlikli biti (MSB)
C	sayıcı çıkışının en yüksek değerlikli bitinden sonra gelen biti
B	sayıcı çıkışının en düşük değerlikli bitinden önce gelen biti
A	sayıcı çıkışının en düşük değerlikli biti (LSB)
Z	sayılacak darbelerin girilmesini sağlayan giriş

Öğrenciye girişten verilen darbelerin sayısına (0 - 19 arasında) göre (Z değişkeninin değeri elle değiştirilecek ise Z harfine basılmalıdır veya menüden Automatic komutu verilmelidir) D, C, B, A çıkış değişkenlerinin alacakları değerleri yorumlayarak doğruluk tablosu şeklinde sunması istenebilir. Menüden *Automatic* komutu verilmiş ise, Z değişkeninin değeri otomatik olarak değiştirileceğinden, DCBA çıkışlarının aldığı değerler sıra ile klavyeden komut vermeye gerek kalmadan görülebilir.

Bu doğruluk tablosu aşağıdaki tabloda verildiği gibi olmalıdır.

Tablo 3.28. Deney - 8' in sonucunda elde edilen doğruluk tablosu.

Z	D	C	B	A
0	0	0	0	0
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0
5	0	1	0	1
6	0	1	1	0
7	0	1	1	1
8	1	0	0	0
9	1	0	0	1
10	1	0	1	0
11	1	0	1	1
12	1	1	0	0
13	1	1	0	1
14	1	1	1	0

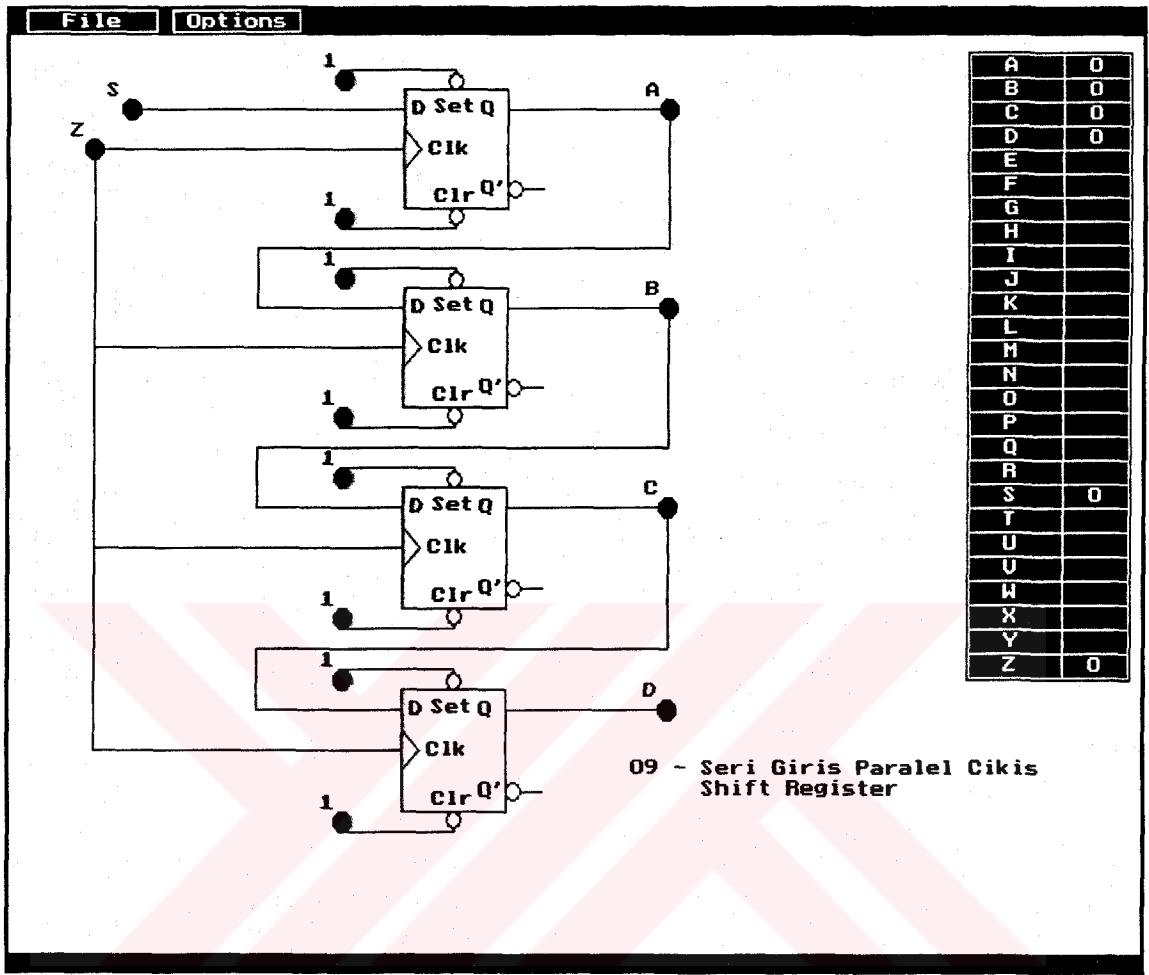
15	1	1	1	1
16	0	0	0	0
17	0	0	0	1
18	0	0	1	0
19	0	0	1	1

Burada dikkat edilecek nokta, sayıcı MOD 16 SAYICI olarak çalışmaktadır. Gelen darbedenin mod 16' ya göre elde karşılığı 4 bit olan ikilik karşılığı DCBA sırasında verilir (Yani, 16 = 0, 19 = 3 gibi). Burada çıkış 0 ile 15 arasında ikilik kodlanmış olan çıkışı DCBA' dan verilir. Bu entegre devre ile kurulmuş devrenin, negatif kenar tetiklemeli JK flip-flop' lar ile tasarlanmış hali 6 nolu deneyde verildiği gibidir. Öğrenci bu deneyden, entegre olarak tasarlanmış ve bir amaca yönelik iş yapan devre elemanların kullanımını öğrenmeye başlamaktadır.



3.2.9. Seri giriş paralel çıkış shift register

9 nolu deneyi seçince, aşağıdaki gibi bir ekran görüntüsü elde edilir :



Şekil 3.23. Deney - 9' un ekran görüntüsü

Bu deneyde öğrenciye pozitif kenar tetiklemeli D tipi flip-floplar ile tasarımı yapılmış ve kurulmuş dört bitlik ikilik seri giriş paralel çıkış sola kaydırmalı shift register' in tanıtımı yapılacaktır. Z genel girişinden gelen darbeler sayılarak BCD (İkilik Kodlanmış Onluk) olarak DCBA çıkışlarından verilmektedir.

Burada belirlenen girişler, çıkışlar ve anlamları ise aşağıdaki gibidir :

Tablo 3.29. Deney - 9' daki giriş ve çıkışların anlamları

İFADELER	ANLAMLARI
D	paralel çıkışının en yüksek değerlikli biti (MSB)
C	paralel çıkışının en yüksek değerlikli bitinden sonra gelen biti
B	paralel çıkışının en düşük değerlikli bitinden önce gelen biti

A	paralel çıkışının en düşük değerlikli biti (LSB)
S	registerin seri giriş ucu
Z	sıradaki seri bilginin alınmasını sağlayan tetikleme girişi

Öğrenciye girişten verilen seri bilgi ve tetikleme girişlerine göre (S değişkeninin değeri elle değiştirilecek ise S harfine basılmalıdır ve Z değişkeninin değeri elle değiştirilecek ise Z harfine basılmalıdır veya menüden Automatic komutu verilmelidir) D, C, B, A çıkış değişkenlerinin alacakları değerleri yorumlayarak doğruluk tablosu şeklinde sunması istenebilir. Menüden *Automatic* komutu verilmiş ise, Z değişkeninin değeri otomatik olarak değiştirileceğinden, aşağıdaki S girişlerine göre DCBA çıkışlarının aldığı değerler sıra ile görülebilir.

Bu doğruluk tablosu aşağıdaki tabloda verildiği gibi olmalıdır.

Tablo 3.30. Deney - 9' un sonucunda elde edilen doğruluk tablosu.

S	D	C	B	A
0	0	0	0	0
1	0	0	0	1
0	0	0	1	0
0	0	1	0	0
0	1	0	0	0
1	0	0	0	1
1	0	0	1	1
0	0	1	1	0
1	1	1	0	1
0	1	0	1	0
0	0	1	0	0
1	1	0	0	1
1	0	0	1	1
1	0	1	1	1
1	1	1	1	1
0	1	1	1	0
1	1	1	0	1
1	1	0	1	1
0	0	1	1	0
1	1	1	0	1

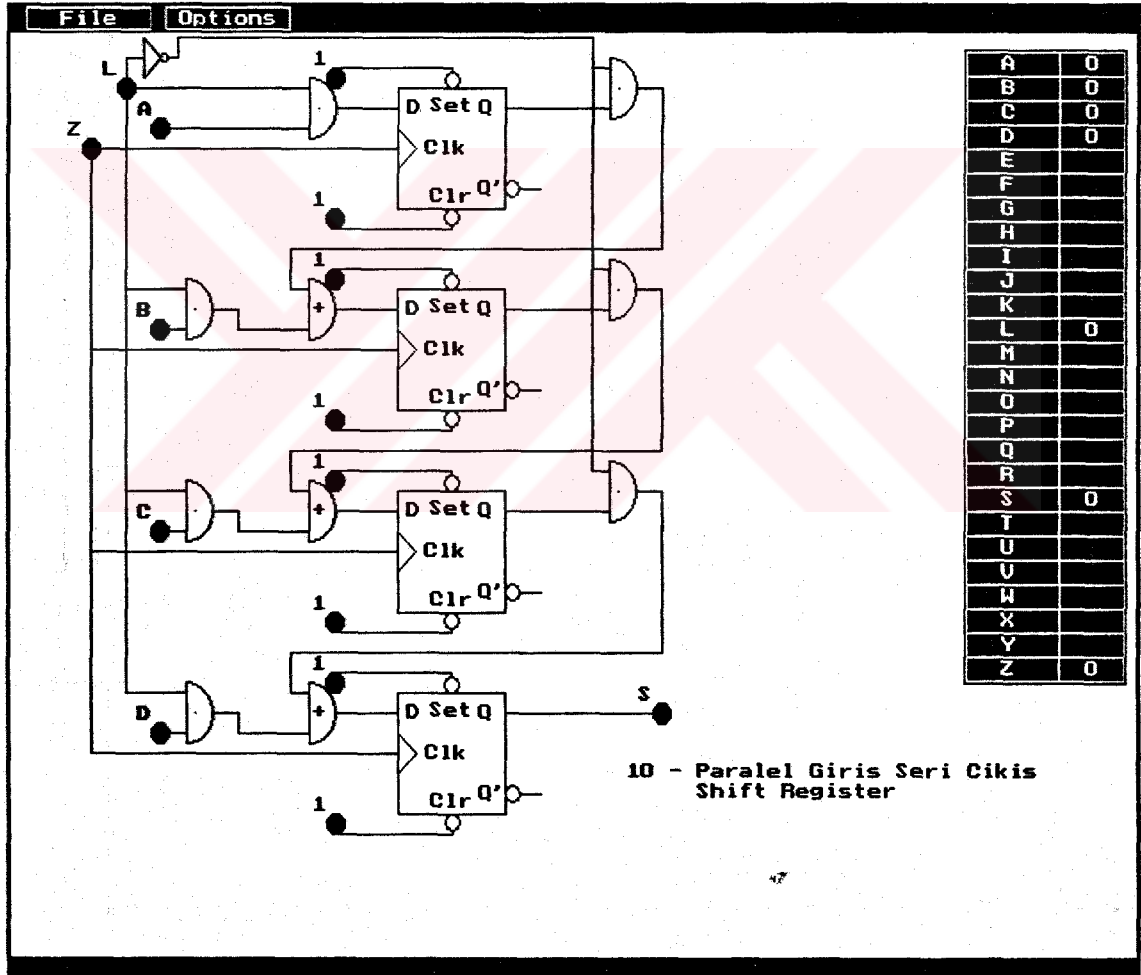
Yukarıda verilen doğruluk tablosunda, registerde bulunan ilk bilginin 0000 olduğu ve S girişinden sırası ile 0, 1, 0, 0, 0, 1, 1, 0, 1, 0, 0, 1, 1, 1, 0, 1, 1, 0, 1 bilgilerin

geldiği varsayılmıştır. Her S seri bilgisi geldiğinde, DCBA çıkışlarında oluşacak son durum hemen yanında verilmiştir.

Burada dikkat edilecek nokta, register seri giriş sola kaydırmalı paralel çıkış olarak çalışmaktadır. Gelen her Z tetiklemesinde, C' deki bilgi D' ye, B' deki bilgi C' ye, A' daki bilgi B' ye ve seri S bilgisi ise A' ya aktarılır (Yani, bir sağdaki bilgi bir sola aktarılmaktadır). Burada bilginin alınması işlemi, Z girişinin 0' dan bire geçişlerinde olmaktadır (Pozitif kenarlarda).

3.2.10. Paralel giriş seri çıkışlı shift register

10 nolu deneyi seçince, aşağıdaki gibi bir ekran görüntüsü elde edilir :



Şekil 3.24. Deney - 10' un ekran görüntüsü

Bu deneyde öğrenciye pozitif kenar tetiklemeli D tipi flip-floplar ile tasarımı yapılmış ve kurulmuş dört bitlik ikilik paralel giriş sola kaydırmalı seri çıkış shift register' in tanıtımı yapılacaktır.

Burada belirlenen girişler, çıkışlar ve anlamları ise aşağıdaki gibidir :

Tablo 3.31. Deney - 10' daki giriş ve çıkışların anlamları

İFADELER	ANLAMLARI
D	paralel girişinin en yüksek değerlikli biti (MSB)
C	paralel girişinin en yüksek değerlikli bitinden sonra gelen biti
B	paralel girişinin en düşük değerlikli bitinden önce gelen biti
A	paralel girişinin en düşük değerlikli biti (LSB)
S	registerin seri çıkış ucu
Z	sıradaki seri bilginin verilmesini sağlayan tetikleme girişi
L	paralel girişlerin registere yüklenmesini sağlayan kontrol giriş

Öğrenciye girişten verilen paralel bilgi ve tetikleme girişlerine göre (D, C, B, A, L değişkenlerinin değerleri elle değiştirilecek ise D, C, B, A harfine basılmalıdır ve Z değişkeninin değeri elle değiştirilecek ise Z harfine basılmalıdır veya menüden Automatic komutu verilmelidir) S çıkış değişkeninin alacağı değerleri yorumlayarak doğruluk tablosu şeklinde sunması istenebilir. Menüden *Automatic* komutu verilmiş ise, Z değişkeninin değeri otomatik olarak değiştirileceğinden, aşağıdaki L ve D, C, B, A girişlerine göre S çıkışının aldığı değerler sıra ile görülebilir.

Bu doğruluk tablosu aşağıdaki tabloda verildiği gibi olmalıdır.

Tablo 3.32. Deney - 10' un sonucunda elde edilen doğruluk tablosu.

L	Z	DCBA	S
1	x	dcba	d
0	0	xxxx	d
0	1	xxxx	c
0	2	xxxx	b
0	3	xxxx	a
0	4	xxxx	a
0	5	xxxx	a
0	6	xxxx	a
0	7	xxxx	a
1	x	hgfe	h
0	0	xxxx	h

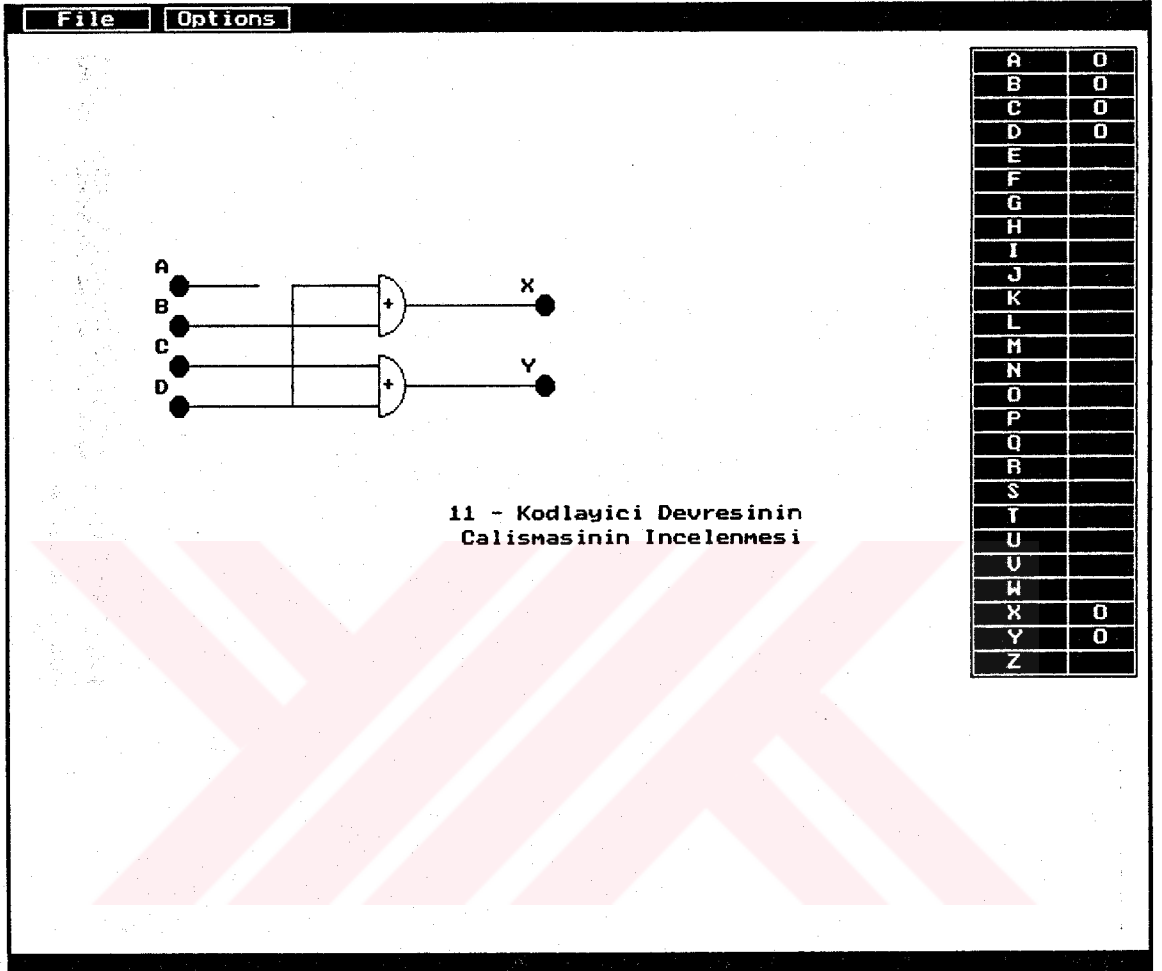
0	1	xxxx	g
0	2	xxxx	f
1	x	lkji	l
0	0	xxxx	l
0	1	xxxx	k
0	2	xxxx	j
0	3	xxxx	i
0	4	xxxx	i
0	5	xxxx	i

Yukarıda verilen doğruluk tablosunda, x ile gösterilen değerin 0 veya 1 olmasının önemli olmadığı belirtilmektedir. Girişlerin ilk olarak DCBA = dcba ve L = 1 olduğu varsayılmaktadır. Daha sonra L = 0 olarak Z' den 7 defa tetikleme gelmektedir. Bu sırada S çıkışı sırası ile d, c, b, a, a, a, a, a olmaktadır. Burada tekrar L = 1 ve DCBA = hgfe olduktan sonra tekrar L = 0 olmaktadır. Bu sırada Z' den iki defa tetikleme gelerek S çıkışının sırası ile h, g, f olmasını sağlamaktadır. Burada tekrar L = 1 ve DCBA = lkji olduktan sonra tekrar L = 0 olmaktadır. Bu sırada Z' den 5 defa tetikleme gelerek S' den sırası ile l, k, j, i, i, i çıkışlarının verilmesini sağlamaktadır. Her S seri bilgisi çıkışı için, DCBA ve L girişlerinin aldıkları değerler ile Z' ye gelen darbe sayısı hemen yanında olacak şekilde yukarıdaki tabloda verilmiştir.

Burada dikkat edilecek nokta, register paralel giriş sola kaydırmalı seri çıkış olarak çalışmaktadır. Gelen her Z tetiklemesinde, D' deki bilgi S' ye, C' deki bilgi D' ye, B' deki bilgi C' ye ve A' daki bilgi ise B' ye aktarılır. Burada bilginin S' den verilmesi işlemi, Z girişinin 0' dan bire geçişlerinde olmaktadır (Pozitif kenarlarda). Eğer L girişinden 1 verilirse paralel girişler registre yüklenir. Eğer L girişinden 0 verilirse register seri çıkış işlevini gerçekleştirir. 'D' flip-flop' unun değeri S seri çıkışından verilir.

3.2.11. Kodlayıcı devresinin çalışmasının incelenmesi

11 nolu deneyi seçince, aşağıdaki gibi bir ekran görüntüsü elde edilir :



Şekil 3.25. Deney - 11' in ekran görüntüsü

Bu deneyde öğrenciye *VEYA* lojik kapıları ile tasarımı yapılmış ve kurulmuş dörtten ikiye kodlayıcı devresinin tanıtımı yapılacaktır. Bu tür devrelerde A, B, C ve D girişlerinden aynı anda sadece bir tanesinin lojik "1" olacağı kabul edilmektedir.

Burada belirlenen girişler, çıkışlar ve anlamları ise aşağıdaki gibidir :

Tablo 3.33. Deney - 11' deki giriş ve çıkışların anlamları

İFADELER	ANLAMLARI
D	3 nolu giriş ucu
C	2 nolu giriş ucu
B	1 nolu giriş ucu

A	0 nolu giriş ucu
Y	çıkış bilgisinin yüksek değerlikli biti
X	çıkış bilgisinin düşük değerlikli biti

Öğrenciye girişlerden sırasıyla uygulanacak lojik “1” lere göre (D, C, B, A değişkenlerinin değerleri değiştirilecek ise D, C, B, A harfine basılmalıdır) Y ve X çıkış değişkenlerinin alacağı değerleri yorumlayarak doğruluk tablosu şeklinde sunması istenebilir. Aşağıdaki doğruluk tablosunda D, C, B ve A girişlerine göre Y ve X çıkışlarının aldığı değerler sıra ile görülebilir.

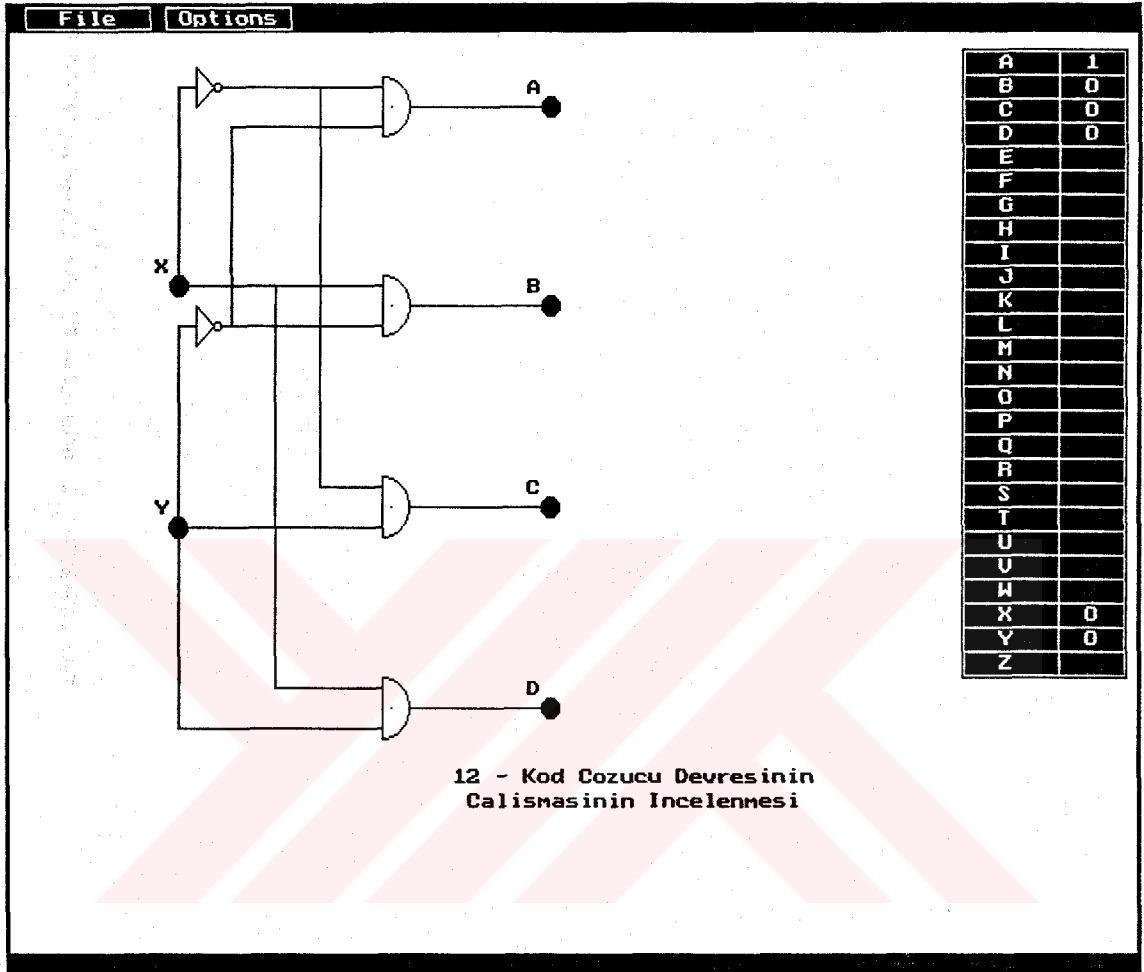
Tablo 3.34. Deney - 11’ in sonucunda elde edilen doğruluk tablosu.

D	C	B	A	Y	X
0	0	0	0	0	0
0	0	0	1	0	1
0	0	1	0	1	0
0	0	1	1	1	1
0	1	0	0	1	0
0	1	0	1	1	0
0	1	1	0	1	1
0	1	1	1	1	1
1	0	0	0	1	1
1	0	0	1	1	1
1	0	1	0	1	1
1	0	1	1	1	1
1	1	0	0	1	1
1	1	0	1	1	1
1	1	1	0	1	1
1	1	1	1	1	1

Yukarıda verilen doğruluk tablosunda italik (*eğik*) gösterilen satırlarda devre amacının dışında kullanılmış ve yanlış çıkışlar elde edilmiştir. Bu devrenin amacı, dört girişinden birisinden “1” geldiği zaman o girişin ikilik kodunu YX çıkışlarından vermektir. Yani, devrede girişlerden sadece bir tanesinin lojik “1” olması esastır.

3.2.12. Kod çözücü devresinin çalışmasının incelenmesi

12 nolu deneyi seçince, aşağıdaki gibi bir ekran görüntüsü elde edilir :



Şekil 3.26. Deney - 12' nin ekran görüntüsü

Bu deneyde öğrenciye *VE* ve *NOT* lojik kapıları ile tasarımı yapılmış ve kurulmuş ikiden dörde kod çözücü devresinin tanıtımı yapılacaktır. Bu tür devrelerde Y ve X girişlerinden uygulanan koda ait A, B, C ve D çıkış uçlarından aynı anda sadece bir tanesi lojik "1" yapılır.

Burada belirlenen girişler, çıkışlar ve anlamları ise aşağıdaki gibidir :

Tablo 3.35. Deney - 12' deki giriş ve çıkışların anlamları

İFADELER	ANLAMLARI
D	3 nolu çıkış ucu
C	2 nolu çıkış ucu

B	1 nolu çıkış ucu
A	0 nolu çıkış ucu
Y	giriş bilgisinin yüksek değerlikli biti
X	giriş bilgisinin düşük değerlikli biti

Öğrenciye girişlerden sırasıyla uygulanacak ikilik koda göre (Y ve X değişkenlerinin değerleri değiştirilecek ise Y ve X harfine basılmalıdır) D, C, B ve A çıkış değişkenlerinin alacağı değerleri yorumlayarak doğruluk tablosu şeklinde sunması istenebilir. Aşağıdaki doğruluk tablosunda Y ve X girişlerine göre D, C, B ve A çıkışlarının aldığı değerler sıra ile görülebilir.

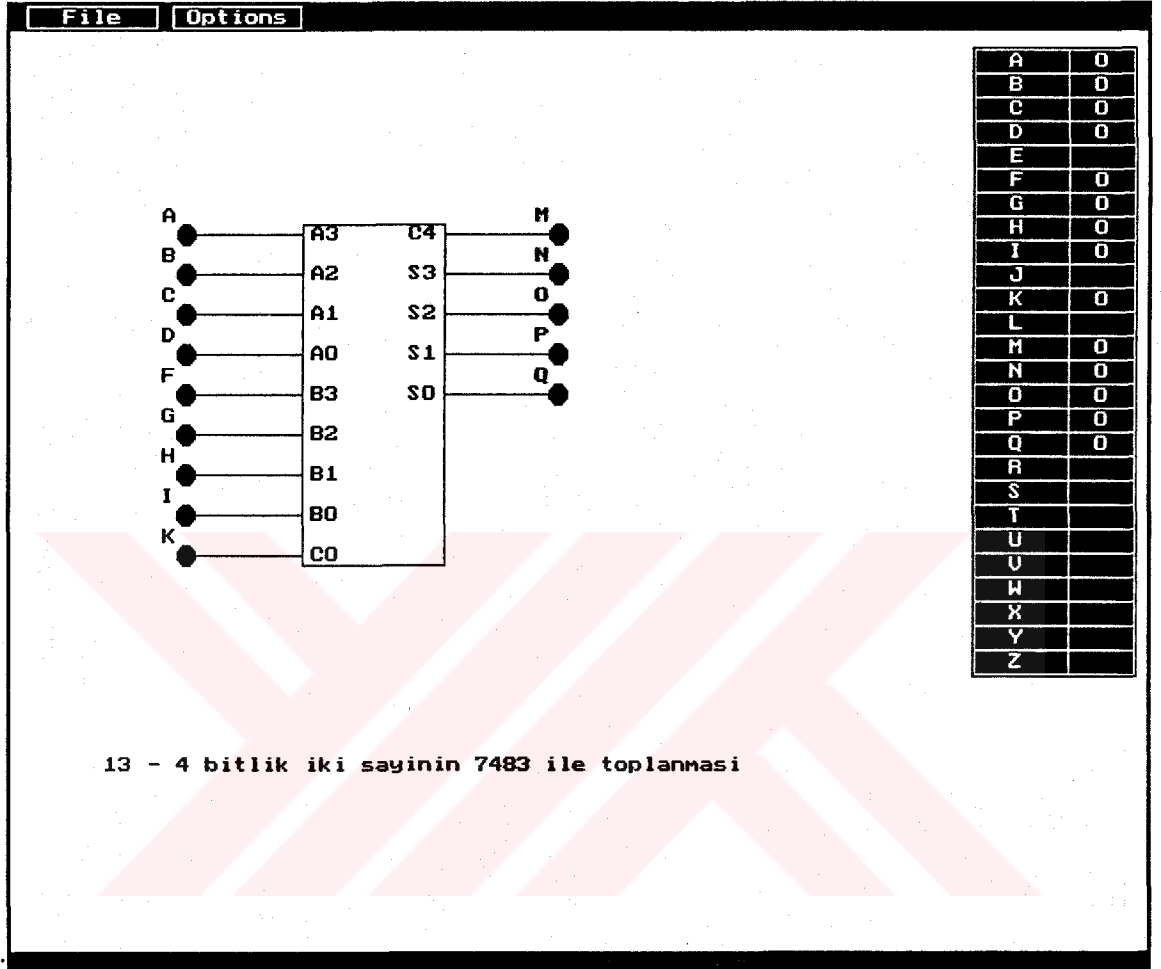
Tablo 3.36. Deney - 12' nin sonucunda elde edilen doğruluk tablosu.

Y	X	D	C	B	A
0	0	0	0	0	1
0	1	0	0	1	0
1	0	0	1	0	0
1	1	1	0	0	0

Bu devrenin amacı, iki girişinden uygulanan ikilik koda ait D (11), C (10), B (01) ve A (00) çıkışını lojik "1" yapmaktadır. Yani devre, bir önceki devrenin karşılığı olarak tasarlanmıştır.

3.2.13. 4 bitlik iki sayının toplama devresi

13 nolu deneyi seçince, aşağıdaki gibi bir ekran görüntüsü elde edilir



Şekil 3.27. Deney - 13' ün ekran görüntüsü

Bu deneyde öğrenciye 7483 entegresi kullanılarak tasarımı yapılmış ve kurulmuş iki tane dört bitlik sayının ($A_3A_2A_1A_0$ ve $B_3B_2B_1B_0$) elde girişi (C_0) ile birlikte toplanarak sonucun ikilik olarak ($S_3S_2S_1S_0$) verilmesini ve sonuç dört bite sığmıyorsa elde çıkışı (C_4) vermesini sağlayan toplama devresinin tanıtımı yapılacaktır.

Öğrenciye girişlerden aşağıdaki tablo 37' den sırasıyla uygulanacak ikilik koda göre (A' dan K' ya kadar olan değişkenlerin değerleri değiştirilecek ise ilgili harfine basılmalıdır) M' den Q' ya kadar olan çıkış değişkenlerinin alacakları değerleri yorumlayarak doğruluk tablosu şeklinde sunması istenebilir.

Bu devrenin amacı, iki tane 4 bitlik sayının tam toplama devresini yapmaktır. Yani devre, iki tane dört bitlik sayıyı varsa bir önceki basamaktan kalan eldesi ile toplamak için tasarlanmıştır.

Tablo 3.37. Deney - 13' de öğrenciye doldurması için verilecek boş doğruluk tablosu.

K	A	B	C	D	F	G	H	I	M	N	O	P	Q
0	0	0	0	0	0	0	0	0					
0	0	0	1	1	0	1	0	1					
0	0	1	0	1	0	0	1	1					
0	1	0	0	0	1	0	0	0					
0	1	0	0	0	0	0	0	1					
0	1	0	0	0	0	1	1	1					
0	0	1	1	1	0	1	1	1					
0	1	0	0	0	1	1	0	0					
0	1	0	0	0	1	1	1	1					
0	1	1	0	0	1	0	1	0					
0	1	1	0	1	1	1	0	0					
0	1	1	1	1	1	1	1	1					
1	0	0	0	0	0	0	0	0					
1	0	0	1	1	0	1	0	1					
1	1	0	0	0	0	0	0	1					
1	1	0	0	0	0	1	1	1					
1	0	1	1	1	0	1	1	1					
1	1	0	0	0	1	1	0	0					
1	1	0	0	0	1	1	1	1					
1	1	1	0	0	1	0	1	0					
1	1	1	0	1	1	1	0	0					
1	1	1	1	1	1	1	1	1					

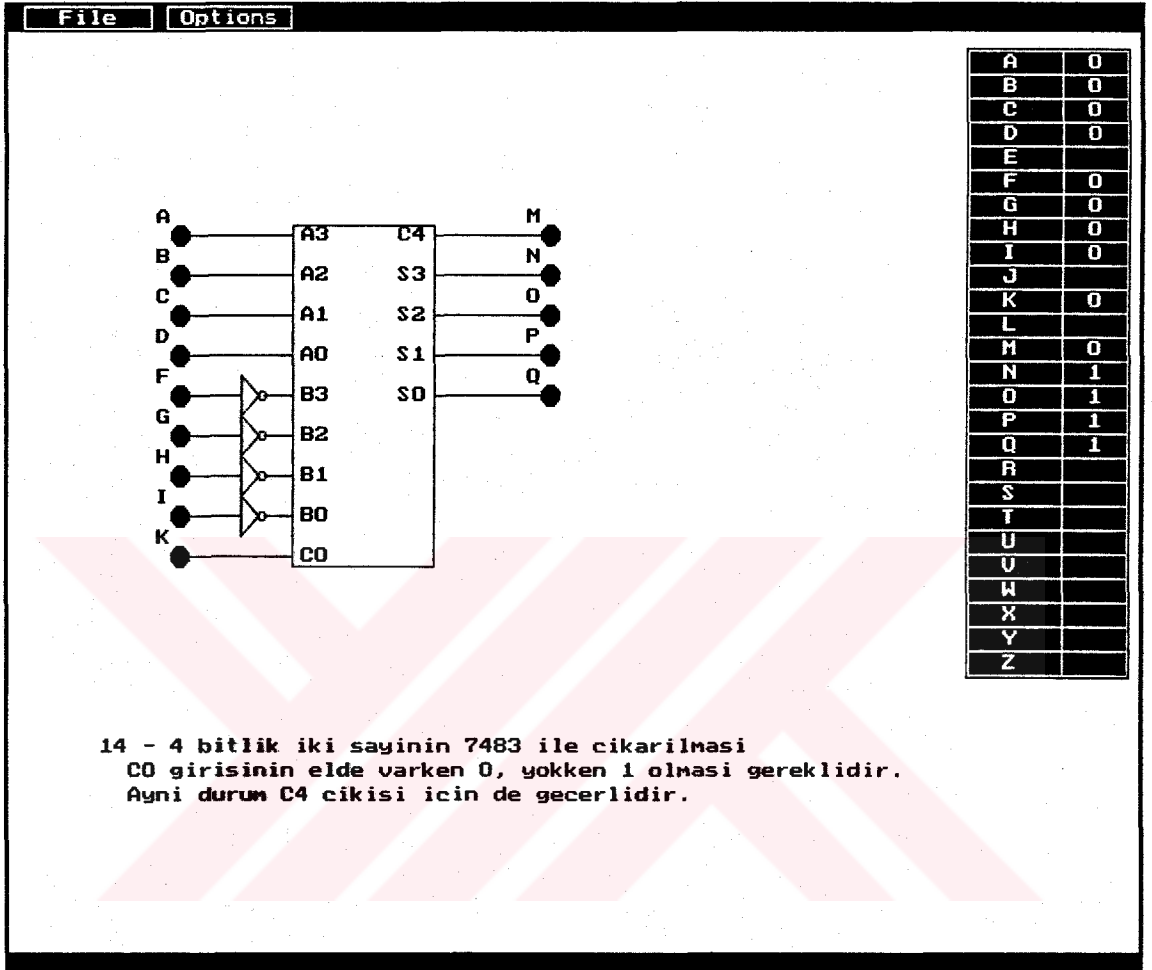
Burada belirlenen girişler, çıkışlar ve anlamları ise aşağıdaki gibidir :

Tablo 3.38. Deney - 13' deki giriş ve çıkışların anlamları

İFADELER	ANLAMLARI
A	ilk sayının en yüksek değerlikli biti (MSB - 3.bit)
B	ilk sayının ikinci biti
C	ilk sayının birinci biti
D	ilk sayının en düşük değerlikli biti (LSB - 0. bit)
F	ikinci sayının en yüksek değerlikli biti (MSB - 3. bit)
G	ikinci sayının ikinci biti
H	ikinci sayının birinci biti
I	ikinci sayının en düşük değerlikli biti (LSB - 0. bit)

3.2.14. 4 bitlik iki sayının çıkarma devresi

14 nolu deneyi seçince, aşağıdaki gibi bir ekran görüntüsü elde edilir :



Şekil 3.28. Deney - 14' ün ekran görüntüsü

Bu deneyde öğrenciye 7483 entegresi ve *DEĞİL* kapı devre elemanları kullanılarak tasarımı yapılmış ve kurulmuş iki tane dört bitlik sayının ($A_3A_2A_1A_0$ ve $B_3B_2B_1B_0$) elde girişi (C_0) ile birlikte farkını alarak sonucun ikilik olarak ($S_3S_2S_1S_0$) verilmesini ve sonuç dört bite sığmıyorsa elde çıkışı (C_4) vermesini sağlayan toplama devresinin tanıtımı yapılacaktır.

Öğrenciye girişlerden aşağıdaki tablo 41' den sırasıyla uygulanacak ikilik koda göre (A' dan K' ya kadar olan değişkenlerin değerleri değiştirilecek ise ilgili harfine basılmalıdır) M' den Q' ya kadar olan çıkış değişkenlerinin alacakları değerleri yorumlayarak doğruluk tablosu şeklinde sunması istenebilir.

Burada belirlenen girişler, çıkışlar ve anlamları ise aşağıdaki gibidir :

Tablo 3.40. Deney - 14' deki giriş ve çıkışların anlamları

İFADELER	ANLAMLARI
A	ilk sayının en yüksek değerlikli biti (MSB - 3.bit)
B	ilk sayının ikinci biti
C	ilk sayının birinci biti
D	ilk sayının en düşük değerlikli biti (LSB - 0. bit)
F	ikinci sayının en yüksek değerlikli biti (MSB - 3. bit)
G	ikinci sayının ikinci biti
H	ikinci sayının birinci biti
I	ikinci sayının en düşük değerlikli biti (LSB - 0. bit)
K	elde girişi (C_0 veya C_{in})
M	elde çıkışı (C_4 veya C_{out})
N	sonucun en yüksek değerlikli biti (MSB - 3. bit)
O	sonucun ikinci biti
P	sonucun birinci biti
Q	sonucun en düşük değerlikli biti (LSB - 0. bit)

Tablo 3.41. Deney - 14' de öğrenciye doldurması için verilecek boş doğruluk tablosu.

K	A	B	C	D	F	G	H	I	M	N	O	P	Q
1	0	0	0	0	0	0	0	0					
1	0	0	1	1	0	1	0	1					
1	0	1	0	1	0	0	1	1					
1	1	0	0	0	1	0	0	0					
1	1	0	0	0	0	0	0	1					
1	1	0	0	0	0	1	1	1					
1	0	1	1	1	0	1	1	1					
1	1	0	0	0	1	1	0	0					
1	1	0	0	0	1	1	1	1					
1	1	1	0	0	1	0	1	0					
1	1	1	0	1	1	1	0	0					
1	1	1	1	1	1	1	1	1					
0	0	0	0	0	0	0	0	0					
0	0	0	1	1	0	1	0	1					
0	0	1	0	1	0	0	1	1					
0	1	0	0	0	1	0	0	0					
0	1	0	0	0	0	0	0	1					
0	1	0	0	0	0	1	1	1					
0	0	1	1	1	0	1	1	1					
0	1	0	0	0	1	1	0	0					

0	1	0	0	0	1	1	1	1					
0	1	1	0	0	1	0	1	0					
0	1	1	0	1	1	1	0	0					
0	1	1	1	1	1	1	1	1					

Aşağıdaki tablo 42' de elde edilecek doğruluk tablosu görülmektedir.

Tablo 3.42. Deney - 14' ün sonucunda elde edilen doğruluk tablosu.

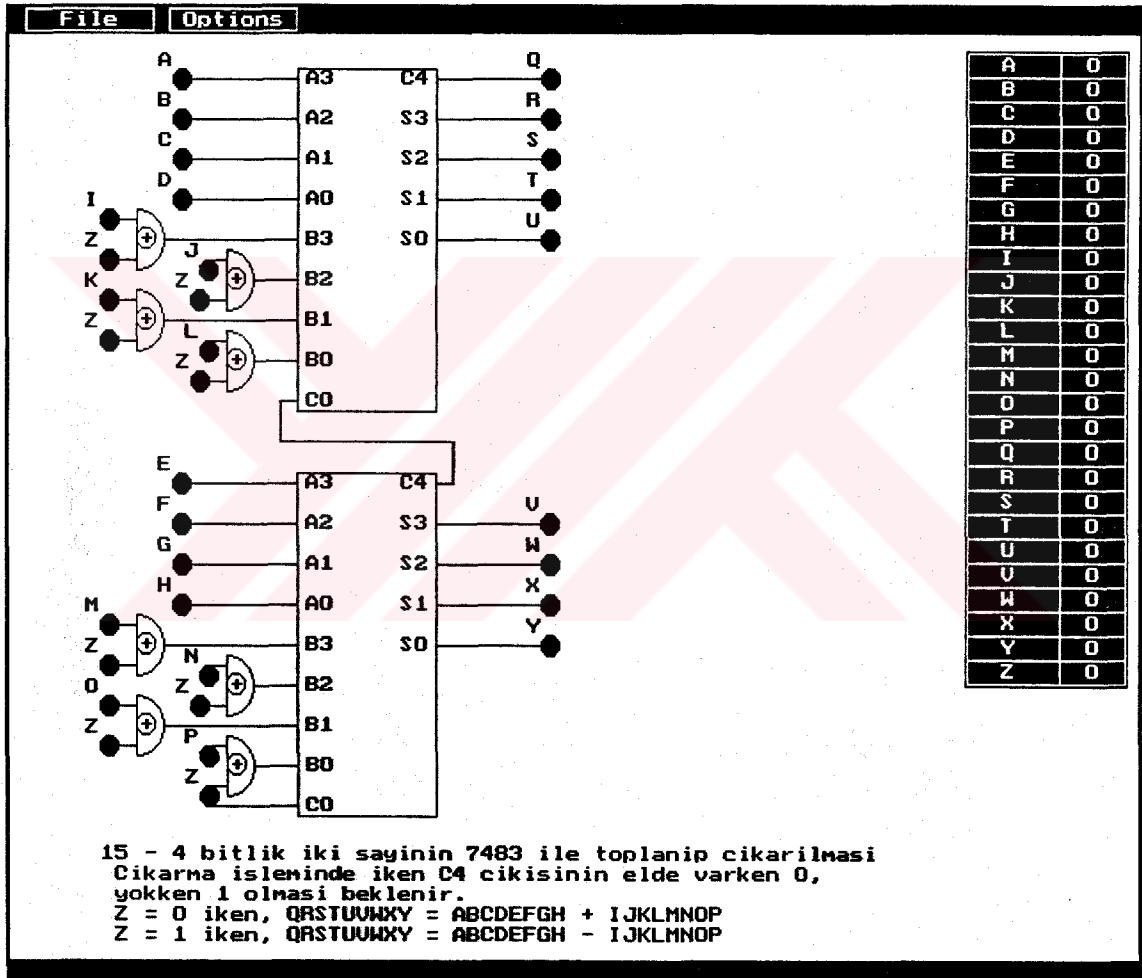
K	A	B	C	D	F	G	H	I	M	N	O	P	Q
1	0	0	0	0	0	0	0	0	1	0	0	0	0
1	0	0	1	1	0	1	0	1	0	1	1	1	0
1	0	1	0	1	0	0	1	1	1	0	0	1	0
1	1	0	0	0	1	0	0	0	1	0	0	0	0
1	1	0	0	0	0	0	0	1	1	0	1	1	1
1	1	0	0	0	0	1	1	1	1	0	0	0	1
1	0	1	1	1	0	1	1	1	1	0	0	0	0
1	1	0	0	0	1	1	0	0	0	1	1	0	0
1	1	0	0	0	1	1	1	1	0	1	0	0	1
1	1	1	0	0	1	0	1	0	1	0	0	1	0
1	1	1	0	1	1	1	0	0	1	0	0	0	1
1	1	1	1	1	1	1	1	1	1	0	0	0	0
0	0	0	0	0	0	0	0	0	0	1	1	1	1
0	0	0	1	1	0	1	0	1	0	1	1	0	1
0	0	1	0	1	0	0	1	1	1	0	0	0	1
0	1	0	0	0	1	0	0	0	0	1	1	1	1
0	1	0	0	0	0	0	0	1	1	0	1	1	0
0	1	0	0	0	0	1	1	1	1	0	0	0	0
0	0	1	1	1	0	1	1	1	0	1	1	1	1
0	1	0	0	0	1	1	0	0	0	1	0	1	1
0	1	0	0	0	1	1	1	1	0	1	0	0	0
0	1	1	0	0	1	0	1	0	1	0	0	0	1
0	1	1	0	1	1	1	0	0	1	0	0	0	0
0	1	1	1	1	1	1	1	1	0	1	1	1	1

Bu devrenin amacı, iki tane 4 bitlik sayının tam çıkarma devresini yapmaktır. Yani devre, iki tane dört bitlik sayıyı varsa bir önceki basamaktan kalan eldesi ile çıkarmak için tasarlanmıştır. Burada çıkarma işleminin yapılabilmesi için öncelikle ikinci sayının tersinin alınması zorunludur. Bu yüzden, B sayısı olarak belirtilen sayı önce *DEĞİL (NOT)* işlemine tabi tutularak 7483 tam toplayıcı entegresine uygulanmıştır. İkilik bir sayının negatifi alınırken sayıya bir eklenmesi gerekli olduğu için, ödünç alma olmadığı zaman

elde girişinden “1” ve elde olduğu zaman ise “0” uygulanması gereklidir. Eğer, isteniyorsa ilk basamağın elde girişi ile son basamağın elde çıkışlarının tersi alınarak devreye de uygulanabilir. Böylece, toplama ve çıkarma devrelerinde eldelerin farklı farklı görevleri varmış gibi kafa karıştırmayı engellenmiş olur.

3.2.15. 8 bitlik iki sayının toplama ve çıkarma devresi

15 nolu deneyi seçince, aşağıdaki gibi bir ekran görüntüsü elde edilir :



Şekil 3.29. Deney - 15' in ekran görüntüsü

Bu deneyde öğrenciye iki tane 7483 entegresi ve **ÖZEL VEYA** kapı devre elemanları kullanılarak tasarımı yapılmış ve kurulmuş iki tane sekiz bitlik sayının ($A_7A_6A_5A_4A_3A_2A_1A_0$ ve $B_7B_6B_5B_4B_3B_2B_1B_0$) elde girişi (C_0) ile birlikte Z kontrol girişi “0” iken toplamını ve Z kontrol girişi “1” iken farkını alarak sonucun ikilik olarak

($S_7S_6S_5S_4S_3S_2S_1S_0$) verilmesini ve sonuç sekiz bite sığmıyorsa elde çıkışı (C_4) vermesini sağlayan toplama devresinin tanıtımı yapılacaktır. B girişleri Z ile özel veya işlemine tutularak toplamada aynen, çıkarmada tersi alınarak 7483 entegrelerine uygulanması sağlanmaktadır.

Bu deneyde istenirse, Z kontrol girişi otomatik değişme moduna getirilebilir. Fakat, devre girişlerine göre bir toplama bir çıkarma yaparak sonuçları sürekli yazacaktır. Bu da rahatsızlık verir. Bu yüzden Z kontrol girişinin değeri de elle değiştirilmelidir.

Öğrenciye girişlerden aşağıdaki tablo 44' den sırasıyla uygulanacak ikilik koda göre (A' dan P' ye kadar olan değişkenlerin değerleri değiştirilecek ise ilgili harfine basılmalıdır) Q' den Y' ye kadar olan çıkış değişkenlerinin alacakları değerleri yorumlayarak doğruluk tablosu şeklinde sunması istenebilir.

Burada belirlenen girişler, çıkışlar ve anlamları ise aşağıdaki gibidir :

Tablo 3.43. Deney - 15' deki giriş ve çıkışların anlamları

İFADELER	ANLAMLARI
A	ilk sayının en yüksek değerlikli biti (MSB - 7.bit)
B	ilk sayının altıncı biti
C	ilk sayının beşinci biti
D	ilk sayının dördüncü biti
E	ilk sayının üçüncü biti
F	ilk sayının ikinci biti
G	ilk sayının birinci biti
H	ilk sayının en düşük değerlikli biti (LSB - 0. bit)
I	ikinci sayının en yüksek değerlikli biti (MSB - 7. bit)
J	ikinci sayının altıncı biti
K	ikinci sayının beşinci biti
L	ikinci sayının dördüncü biti
M	ikinci sayının üçüncü biti
N	ikinci sayının ikinci biti
O	ikinci sayının birinci biti
P	ikinci sayının en düşük değerlikli biti (LSB - 0. bit)
Q	elde çıkışı (C_4 veya C_{out})
R	sonucun en yüksek değerlikli biti (MSB - 7. bit)
S	sonucun altıncı biti
T	sonucun beşinci biti
U	sonucun dördüncü biti
V	sonucun üçüncü biti

W	sonucun ikinci biti
X	sonucun birinci biti
Y	sonucun en düşük değerlikli biti (LSB - 0. bit)
Z	hangi işlemin yapılacağını seçmeye yarayan kontrol girişi

Bu devrenin amacı, iki tane 8 bitlik sayının tam toplama ve çıkarma devresini yapmaktır. Burada çıkarma işleminin yapılabilmesi için öncelikle ikinci sayının tersinin alınması zorunludur. Bu yüzden, B sayısı olarak belirtilen sayı önce Z kontrol girişi ile *ÖZEL VEYA* işlemine tabi tutularak 7483 tam toplayıcı entegresine uygulanmıştır. Çıkarma işlemi yapılacağı zaman Z' nin değeri "1" olmaktadır. Böylece gelen diğer sinyalin tersi alınmış olmaktadır. İkilik bir sayının negatifi alınırken sayıya bir eklenmesi gerekli olduğu için, Z kontrol girişi direkt olarak elde girişinden uygulanması gereklidir. Tablo 46' dan özel veya işleminin nasıl olup ta, kontrol işlemi için kullanıldığını görebilirsiniz.

Tablo 3.44. Deney - 15' de öğrenciye doldurması için verilecek boş doğruluk tablosu.

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0										0
0	0	0	0	1	0	0	0	0	0	0	0	0	1	1	1										0
0	0	0	0	1	1	1	1	0	0	0	0	0	1	1	0										0
0	0	0	0	0	1	1	0	0	0	0	0	1	1	1	1										0
0	0	1	1	0	0	0	1	0	0	1	1	0	0	1	0										0
1	1	1	1	1	0	1	0	1	1	0	1	1	1	0	0										0
0	1	1	0	0	1	0	0	1	1	1	1	1	1	1	1										0
1	1	0	0	1	0	0	0	0	1	1	0	0	1	0	0										0
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1										0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0										1
0	0	0	0	1	0	0	0	0	0	0	0	0	1	1	1										1
0	0	0	0	1	1	1	1	0	0	0	0	0	1	1	0										1
0	0	0	0	0	1	1	0	0	0	0	0	1	1	1	1										1
0	0	1	1	0	0	0	1	0	0	1	1	0	0	1	0										1
1	1	1	1	1	1	0	1	0	1	1	0	1	1	1	0										1
0	1	1	0	0	1	0	0	1	1	1	1	1	1	1	1										1
1	1	0	0	1	0	0	0	0	1	1	0	0	1	0	0										1
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1										1

Aşağıdaki tablo 45' de elde edilecek doğruluk tablosu görülmektedir.

Tablo 3.45. Deney - 15' in sonucunda elde edilen doğruluk tablosu.

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	1	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0	0	1	1	1	1	1
0	0	0	0	1	1	1	1	0	0	0	0	0	1	1	0	0	0	0	0	1	0	1	0	1	1
0	0	0	0	0	1	1	0	0	0	0	0	1	1	1	1	0	0	0	0	1	0	1	0	1	1
0	0	1	1	0	0	0	1	0	0	1	1	0	0	1	0	0	0	1	1	0	0	0	1	1	1
1	1	1	1	1	0	1	0	1	1	0	1	1	1	0	0	1	1	1	0	1	0	1	1	0	0
0	1	1	0	0	1	0	0	1	1	1	1	1	1	1	1	1	0	1	1	0	0	0	1	1	1
1	1	0	0	1	0	0	0	0	1	1	0	0	1	0	0	1	0	0	1	0	1	1	0	0	0
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
0	0	0	0	1	0	0	0	0	0	0	0	0	1	1	1	1	0	0	0	0	0	0	0	1	1
0	0	0	0	1	1	1	1	0	0	0	0	0	1	1	0	1	0	0	0	0	1	0	0	1	1
0	0	0	0	0	1	1	0	0	0	0	0	1	1	1	1	0	1	1	1	1	0	1	1	1	1
0	0	1	1	0	0	0	1	0	0	1	1	0	0	1	0	0	1	1	1	1	1	1	1	1	1
1	1	1	1	1	0	1	0	1	1	0	1	1	1	0	0	1	0	0	0	1	1	1	1	0	1
0	1	1	0	0	1	0	0	1	1	1	1	1	1	1	1	0	0	1	1	0	0	1	0	1	1
1	1	0	0	1	0	0	0	0	1	1	0	0	1	0	0	1	0	1	1	0	0	1	0	0	1
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	1

Tablo 3.46. Özel veya ile sinyal işleme

Z	S	S özel veya Z işlemi
0	A	A
1	A	not A

4. DENEYLERİN BİLGİSAYARA TANITIMI

Bu deneylerin bilgisayara tanıtılabilmesi için MAKEnn.PAS isimli bilgisayar programları yazılmıştır. Bu programların çalıştırılması sonucunda, DENEYnn.STM isimli dosyalar oluşturulur. Eğer bu dosyalar mevcut değil ise, bu deney dosyalarını oluşturmaya yarayan ilgili MAKEnn.EXE program dosyalarını çalıştırmanız gereklidir. Bu programları çalıştırdığınızda, ekrana şekli çizer ve her bir elemanın başlangıç değerlerinin atamasını otomatik olarak yapar. Bu programlar ekranda çalışırken kullanıcının sadece iki komutunu dikkate alır. Bunlar, ESC tuşu ile programdan çıkış ve S tuşu ile deney dosyasının diske kaydının yapılmasıdır. Burada, nn yerine 01 ile 15 arasında bir sayı yazılmalıdır.

Yeni bir deneyi tanıtılabilmesi için eski deneyin silinmesi gerekmektedir. Eğer var olan bir deney üzerinde değişikliklerin yapılması gerekirse, eski deney dosyasında tablo 4.1' de verilen tip isimlerine dikkat edilerek Init metodlarının kullanıldığı satırları eklenmesi veya çıkarılması gerekmektedir.

AND, OR, NOT gibi entegre devre elemanlarının eklenebilmesi için aşağıdaki gibi bir satırın ilavesi gerekmektedir. Bu satır ile, sol üst köşesi 100, 100 koordinatında olan bir VE kapısı eklenmiş olur.

ObjectList^.Insert(New(PAndLogic, Init(100, 100)));

Düğüm noktalarının tanıtımda ise, tüm noktaların giriş veya çıkış olarak kullanıldıkları gibi değerlerin girilmesi gerekmektedir. Böyle bir durumda aşağıdaki satırlara benzer satırların ilavesi yapılırsa, 100,100 koordinatına sahip bir düğüm noktası eklenmiş olur.

E.PointType := logInput + logOutput;

Yani noktaya hem değer girişi hem de
değer çıkışı olacak.

A.PointType := logInput;

B := A;

C.PointType := logOutput;

D := C;

Yani düğüm noktasının sol - üst birleşim noktalarından bilgi girişi ve sağ - alt birleşim noktalarından ise bilgi çıkışı olacak.

ObjectList^.Insert(New(PGrPoint, Init(100, 100)));

Her hangi bir kısa devre elemanının tanıtımına ise, hangi uçtan hangi uca bilgi akışı olduğu yine PointType tanımlamaları ile verilmelidir.

A.PointType := logInput;

A.X := 100;

A.Y := 100;

B.PointType := logOutput;

B.X := 200;

B.Y := 200;

ObjectList^.Insert(New(PLineLogic, Init(A, B)));

100, 100' den 200, 200' e doğru bilgi akışını sağlayan kısa devre elemanının tanıtımı yapıldı.

Bu bilgisayar programlarında kullanılan tip isimleri MYGROBJ.PAS ve MYLOGICS.PAS ünite dosyalarında tanımlanmışlardır. Bu tanımlamalar ve niçin kullanıldıkları tablo 4.1' de verilmiştir.

Tablo 4.1. Programlarda kullanılan tip isimleri

ELEMAN	PROGRAMDA KULLANILAN TYPE İSMİ
DEĞERİ OLAN HER HANGİ BİR NOKTA	TPOINTVALUE Konumu ve -32768 ile 32767 arasında değeri olabilen her hangi bir noktayı temsil etmek için kullanılır.
TEMEL GRAFİK NESNESİ	TGRAPHOBJECT Konumu, bir harften oluşan ismi, ve X-Y hariç A-Z arası ismi olan 24 tane TPOINTVALUE' dan oluşan temel nesne. Bu nesnede üzerine öykünülen asenkron ve senkron çalışma alt yordamları ile çizim, ilk değer atama gibi işlemleri yapabilen çeşitli alt yordamları mevcuttur.
DÜĞÜM NOKTASI	TGRPOINT Konumu, A-D arası dört tane TPOINTVALUE' dan oluşan tek başına bir düğüm noktasıdır.
ÇİZGİ (KISA DEVRE ELEMANI)	TLINELOGIC İki ucu birleştirmeye yarayan ve A-B gibi iki tane TPOINTVALUE' su bulunan bir grafik nesnesidir.
LOJİK "0" NOKTASI. (GND - TOPRAK)	TGND Değeri sabit olarak lojik "0" olan düğüm noktası nesnesidir.

LOJİK "1" NOKTASI (Vcc - BESLEME)	TVCC Değeri sabit olarak lojik "1" olan düğüm noktası nesnesidir.
AND KAPISI	TANDLOGIC A ve B gibi iki tane TPOINTVALUE' dan gelen değeri lojik AND işlemine uygulayıp sonucu E' den veren grafik nesnesidir.
OR KAPISI	TORLOGIC A ve B gibi iki tane TPOINTVALUE' dan gelen değeri lojik OR işlemine uygulayıp sonucu E' den veren grafik nesnesidir.
NOT KAPISI	TNOTLOGIC A gelen değeri lojik NOT işlemine uygulayıp sonucu E' den veren grafik nesnesidir.
XOR KAPISI	TXORLOGIC A ve B gibi iki tane TPOINTVALUE' dan gelen değeri lojik XOR işlemine uygulayıp sonucu E' den veren grafik nesnesidir.
NAND KAPISI	TNANDLOGIC A ve B gibi iki tane TPOINTVALUE' dan gelen değeri lojik NAND işlemine uygulayıp sonucu E' den veren grafik nesnesidir.
NOR KAPISI	TNORLOGIC A ve B gibi iki tane TPOINTVALUE' dan gelen değeri lojik NOR işlemine uygulayıp sonucu E' den veren grafik nesnesidir.
POZİTİF KENAR TETİKLEMELİ JK - FF	TPETJKFF A noktası J' ye, B noktası K' ya, C noktası tetiklemeye, D noktası normal çıkışa, E noktası terslenmiş çıkışa, F noktası Preset'e ve G noktası ise Reset (Clear)' a bağlanmıştır.
NEGATİF KENAR TETİKLEMELİ JK - FF	TNETJKFF A noktası J' ye, B noktası K' ya, C noktası tetiklemeye, D noktası normal çıkışa, E noktası terslenmiş çıkışa, F noktası Preset'e ve G noktası ise Reset (Clear)' a bağlanmıştır.
POZİTİF KENAR TETİKLEMELİ RS - FF	TPETRSFF A noktası S' ye, B noktası R' ye, C noktası tetiklemeye, D noktası normal çıkışa, E noktası terslenmiş çıkışa, F noktası Preset'e ve G noktası ise Reset (Clear)' a bağlanmıştır.
NEGATİF KENAR TETİKLEMELİ JK - FF	TNETRSFF A noktası S' ye, B noktası R' ye, C noktası tetiklemeye, D noktası normal çıkışa, E noktası terslenmiş çıkışa, F noktası Preset'e ve G noktası ise Reset (Clear)' a bağlanmıştır.

POZİTİF KENAR TETİKLEMELİ D - FF	TPETDFF A noktası veriye, C noktası tetiklemeye, D noktası normal çıkışa, E noktası terslenmiş çıkışa, F noktası Preset'e ve G noktası ise Reset (Clear)' a bağlanmıştır.
NEGATİF KENAR TETİKLEMELİ D - FF	TNETDFF A noktası veriye, C noktası tetiklemeye, D noktası normal çıkışa, E noktası terslenmiş çıkışa, F noktası Preset'e ve G noktası ise Reset (Clear)' a bağlanmıştır.
7493 ENTEGRE DEVRE ELEMANI	TTL7493 A noktası entegrenin A ucuna, B noktası entegrenin B ucuna, C noktası 1 bolu reset ucuna, D noktası 2 nolu reset ucuna , E-F-G-H noktaları ise sırası ile A-B-C-D çıkışlarına bağlanmıştır.
7483 ENTEGRE DEVRE ELEMANI	TTL7483 A-B-C-D noktaları sırası ile A3-A2-A1-A0 girişlerine, E-F-G-H noktaları sırası ile B3-B2-B1-B0 girişlerine, I noktası C0 elde girişine, J noktası C4 elde çıkışına ve K-L-M-N ise sırası ile S3-S2-S1-S0 sonuç çıkışlarına bağlanmıştır.

Örnek bir devre tanıtımı aşağıdaki gibidir. Burada devre üzerinde yapılan açıklamalar italik olarak ve asıl program ise düz yazı olarak verilmiştir. Bu programın yazılıp çalıştırılabilmesi için Turbo Pascal 6.0 veya daha ileri uyarlamasının olması gereklidir.

program Make15;

{ This program was written to form experiment file 'DENEY15.STM'.

This file contains '4 bit binary two number full adder and subtracter circuit' experiment. And this file is used in main program. }

Program ile ilgili giriş ve açıklama kısımları yazıldı.

uses MyLogics, MyGrObj, Objects, Graph, MyEvents;

Programın kullandığı temel tip isimleri ve değişkenlerin tanımlarının yer aldığı üniteler programa ilave edildi.

var dr,m:integer;

ObjectList: Pcollection;

Programdaki nesnelerin sıra ile yerleştirildikleri depolama nesnesi.

A,B,C,D,E: TPointValue;

Ev: TEvent;

StrY: Integer;

i,j: Integer;

P1,P2:PGraphObject;

Programdaki nesnelerin yer değiştirebilmesi için kullanılan temel grafik nesneleri.

S1, S: String;

GraphicsStream: TBufStream;

*Programda çizimi tamamlanan nesnelerin diske kayıtları
için kullanılan ara birim nesnesinin tanıtımı.*

Programın kullandığı genel değişkenler tanımlandı.

procedure Calistir;

Programda yerleştirilen nesnelerin Run metodlarını çağıran alt program.

procedure CallRun(P: PGraphObject); far;

begin

P^.Run;

end;

Belirtilen nesnenin Run metodunu çalıştıran alt program.

procedure CallDone(P: PGraphObject); far;

begin

P^.Done;

end;

Belirtilen nesnenin Done metodunu çalıştıran alt program.

begin

Ev.KeyCode := kbNoKey;

while Ev.KeyCode <> kbEsc do begin

ESC tuşuna basılıncaya kadar devam.

ClearEvent;

```

GetEvent(Ev);
StrY := 50;
if Ev.What = evKeyDown then
  case Ev.CharCode Of
    'O': with GraphicsStream do

```

'O' tuşuna basılınca,

```

begin
  Init('DENEY15.STM', stOpen, 1024);  { Open stream }
  Dispose(ObjectList, Done);
  ObjectList := PCollection(Get);      { Load collection }
  Done;                                { Shut down stream }
  if Status <> 0 then System.write(#7); { Check for error }

```

Diske kayıtlı olan DENEY15.STM isimli nesne dosyasını yükle.

```
end;
```

```

'S': with GraphicsStream do begin

```

'S' tuşuna basılınca,

```

  Init('DENEY15.STM', stCreate, 1024); { Open stream }
  Put(ObjectList);                      { Output collection }
  Done;                                { Shut down stream }
  if Status <> 0 then System.write(#7); { Check for error }

```

Nesne dosyasını DENEY15.STM ismiyle diske kayıt et.

```
end;
```

```

'L': begin

```

'L' tuşuna basınca,

```

  SetFillStyle(SolidFill, GetBkColor);
  Bar (500, 0, GetMaxX, GetMaxY);
  for i := 0 to ObjectList^.Count - 1 do begin
    P1 := ObjectList^.At(i);
    Str(i, S);

```

```

Str(P1^.A.PointType, S1);
Inc(StrY, TextHeight('Q'));
OutTextXY(505, StrY, S + '.A : ' + S1);
Str(P1^.B.PointType, S1);
Inc(StrY, TextHeight('Q'));
OutTextXY(505, StrY, S + '.B : ' + S1);
Str(P1^.C.PointType, S1);
Inc(StrY, TextHeight('Q'));
OutTextXY(505, StrY, S + '.C : ' + S1);
end;

```

*Tüm değişkenlerin A, B, ve C noktalarının
değerlerini tablo şeklinde yaz.*

```

end;
  '1': begin
P1 := ObjectList^.At(1);
if P1^.A.Value = -1 then P1^.A.Value := 0
else P1^.A.Value := -1;
ObjectList^.AtPut(1, P1);
end;
  '3': begin
P1 := ObjectList^.At(3);
end;
  '0': begin
P1 := ObjectList^.At(0);
if P1^.A.Value = -1 then P1^.A.Value := 0
else P1^.A.Value := -1;
ObjectList^.AtPut(0, P1);
end;
  '5': begin
P1 := ObjectList^.At(5);
end;

```

```

    '2': begin
P1 := ObjectList^.At(2);
end;
    '4': begin
P1 := ObjectList^.At(4);
end;
end;
end;
end;

```

```

procedure Ciz;

```

*Tüm nesnelerin ekrana nasıl çizileceklerini belirten
Draw metodlarını çalıştırmaya yarayan alt program.*

```

procedure CallDraw(P: PGraphObject); far;

```

Belirtilen nesnenin Draw metodunu çağırır.

```

begin
    P^.Draw;
end;
begin
    ObjectList^.ForEach(@CallDraw);
end;
begin
    detectgraph(dr,m);
    initgraph(dr,m,'c:\tp\bgi');

```

Grafik modunu başlat.

```

RegisterObjects;
RegisterGrObj;
RegisterLogics;

```

Tüm nesnelerin programa tanıtımını yap.

ObjectList := New(PCollection, Init(10, 5));

Nesnelerin konulacağı sıralı liste için hafızada yer aç.

ObjectList^.Insert(New(PExpName, Init(50, 415,
'15 - 4 bitlik iki sayının 7483 ile toplanıp çıkarılması')));

ObjectList^.Insert(New(PExpName, Init(50, 425,
'Çıkarma işleminde iken C4 çıkışının elde varken 0,')));

ObjectList^.Insert(New(PExpName, Init(50, 435,
' yokken 1 olması beklenir.')));

ObjectList^.Insert(New(PExpName, Init(50, 445,
'Z = 0 iken, QRSTUVWXY = ABCDEFGH + IJKLMNOP')));

ObjectList^.Insert(New(PExpName, Init(50, 455,
'Z = 1 iken, QRSTUVWXY = ABCDEFGH - IJKLMNOP')));

Belirtilen satır ve sütunlara 4 satırlık açıklama yaz.

A.Value := 0;

B := A;

C := A;

D := A;

E := A;

A.PointType := logInput;

B.PointType := logOutput;

E.PointType := logMustInput;

ObjectList^.Insert(New(PGrPoint,
Init(95, 35, B, B, B, B, E, 'A')));

95, 35' e 'A' noktasını,

```
ObjectList^.Insert(New(PGrPoint,
  Init(95, 55, B, B, B, B, E, 'B')));
```

95, 55' e 'B' noktasını,

```
ObjectList^.Insert(New(PGrPoint,
  Init(95, 75, B, B, B, B, E, 'C')));
```

95, 75' e 'C' noktasını,

```
ObjectList^.Insert(New(PGrPoint,
  Init(95, 95, B, B, B, B, E, 'D')));
```

95, 95' e 'D' noktasını,

```
ObjectList^.Insert(New(PGrPoint,
  Init(95, 235, B, B, B, B, E, 'E')));
```

95, 235' e 'E' noktasını,

```
ObjectList^.Insert(New(PGrPoint,
  Init(95, 255, B, B, B, B, E, 'F')));
```

95, 255' e 'F' noktasını,

```
ObjectList^.Insert(New(PGrPoint,
  Init(95, 275, B, B, B, B, E, 'G')));
```

95, 275' e 'G' noktasını,

```
ObjectList^.Insert(New(PGrPoint,
  Init(95, 295, B, B, B, B, E, 'H')));
```

95, 295' e 'H' noktasını,

```
ObjectList^.Insert(New(PGrPoint,
```

Init(55, 105, B, B, B, B, E, 'T')));

55, 105' e 'T' noktasını,

ObjectList^.Insert(New(PGrPoint,
Init(110, 130, B, B, B, B, E, 'J')));

55, 130' e 'J' noktasını,

ObjectList^.Insert(New(PGrPoint,
Init(55, 145, B, B, B, B, E, 'K')));

55, 145' e 'K' noktasını,

ObjectList^.Insert(New(PGrPoint,
Init(110, 170, B, B, B, B, E, 'L')));

110, 170' e 'L' noktasını,

ObjectList^.Insert(New(PGrPoint,
Init(55, 305, B, B, B, B, E, 'M')));

55, 305' e 'M' noktasını,

ObjectList^.Insert(New(PGrPoint,
Init(110, 330, B, B, B, B, E, 'N')));

110, 330' e 'N' noktasını,

ObjectList^.Insert(New(PGrPoint,
Init(55, 345, B, B, B, B, E, 'O')));

55, 345' e 'O' noktasını,

ObjectList^.Insert(New(PGrPoint,
Init(110, 370, B, B, B, B, E, 'P')));

110, 370' e 'P' noktasını,

```
ObjectList^.Insert(New(PGrPoint,
  Init(55, 125, B, B, B, B, E, 'Z')));
```

55, 125' e 'Z' noktasını,

```
ObjectList^.Insert(New(PGrPoint,
  Init(105, 145, B, B, B, B, E, 'Z')));
```

105, 145' e 'Z' noktasını,

```
ObjectList^.Insert(New(PGrPoint,
  Init(55, 165, B, B, B, B, E, 'Z')));
```

55, 165' e 'Z' noktasını,

```
ObjectList^.Insert(New(PGrPoint,
  Init(105, 185, B, B, B, B, E, 'Z')));
```

105, 185' e 'Z' noktasını,

```
ObjectList^.Insert(New(PGrPoint,
  Init(55, 325, B, B, B, B, E, 'Z')));
```

55, 325' e 'Z' noktasını,

```
ObjectList^.Insert(New(PGrPoint,
  Init(105, 345, B, B, B, B, E, 'Z')));
```

105, 345' e 'Z' noktasını,

```
ObjectList^.Insert(New(PGrPoint,
  Init(55, 365, B, B, B, B, E, 'Z')));
```

55, 365' e 'Z' noktasını,

```
ObjectList^.Insert(New(PGrPoint,
  Init(110, 390, B, B, B, B, E, 'Z')));
```

110, 390' e 'Z' noktasını,

```
E.PointType := logOutput + logInput;
ObjectList^.Insert(New(PGrPoint,
  Init(305, 35, A, A, A, A, E, 'Q')));
```

305, 35' e 'Q' noktasını,

```
ObjectList^.Insert(New(PGrPoint,
  Init(305, 55, A, A, A, A, E, 'R')));
```

305, 55' e 'R' noktasını,

```
ObjectList^.Insert(New(PGrPoint,
  Init(305, 75, A, A, A, A, E, 'S')));
```

305, 75' e 'S' noktasını,

```
ObjectList^.Insert(New(PGrPoint,
  Init(305, 95, A, A, A, A, E, 'T')));
```

305, 95' e 'T' noktasını,

```
ObjectList^.Insert(New(PGrPoint,
  Init(305, 115, A, A, A, A, E, 'U')));
```

305, 115' e 'U' noktasını koy.

```
A.PointType := logInput;
```

```
B.PointType := logOutput;
```

```
A.X := 100;
```

```
A.Y := 35;
```

```
B.X := 150;
```

```
B.Y := 35;
```

```
ObjectList^.Insert(New(PLineLogic, Init(A, B)));
```

(100, 35) - (150, 35) arasını kısa devre yap.

A.Y := 55;

B.Y := 55;

ObjectList^.Insert(New(PLineLogic, Init(A, B)));

(100, 55) - (150, 55) arasını kısa devre yap.

A.Y := 75;

B.Y := 75;

ObjectList^.Insert(New(PLineLogic, Init(A, B)));

(100, 75) - (150, 75) arasını kısa devre yap.

A.Y := 95;

B.Y := 95;

ObjectList^.Insert(New(PLineLogic, Init(A, B)));

(100, 95) - (150, 95) arasını kısa devre yap.

A.Y := 115;

B.Y := 115;

ObjectList^.Insert(New(PLineLogic, Init(A, B)));

(100, 115) - (150, 115) arasını kısa devre yap.

A.Y := 155;

B.Y := 155;

ObjectList^.Insert(New(PLineLogic, Init(A, B)));

(100, 155) - (150, 155) arasını kısa devre yap.

A.Y := 235;

B.Y := 235;

ObjectList^.Insert(New(PLineLogic, Init(A, B)));

(100, 235) - (150, 235) arasını kısa devre yap.

A.Y := 255;

B.Y := 255;

ObjectList^.Insert(New(PLineLogic, Init(A, B)));

(100, 255) - (150, 255) arasını kısa devre yap.

A.Y := 275;

B.Y := 275;

ObjectList^.Insert(New(PLineLogic, Init(A, B)));

(100, 275) - (150, 275) arasını kısa devre yap.

A.Y := 295;

B.Y := 295;

ObjectList^.Insert(New(PLineLogic, Init(A, B)));

(100, 295) - (150, 295) arasını kısa devre yap.

A.Y := 315;

B.Y := 315;

ObjectList^.Insert(New(PLineLogic, Init(A, B)));

(100, 315) - (150, 315) arasını kısa devre yap.

A.Y := 355;

B.Y := 355;

ObjectList^.Insert(New(PLineLogic, Init(A, B)));

(100, 355) - (150, 355) arasını kısa devre yap.

ObjectList^.Insert(New(PXorLogic, Init(60, 90)));

Sol üst köşesi 60, 90' da bulunan bir XOR kapısı koy.

ObjectList^.Insert(New(PXorLogic, Init(110, 110)));

Sol üst köşesi 110, 110' da bulunan bir XOR kapısı koy.

ObjectList^.Insert(New(PXorLogic, Init(60, 130)));

Sol üst köşesi 60, 130' da bulunan bir XOR kapısı koy.

ObjectList^.Insert(New(PXorLogic, Init(110, 150)));

Sol üst köşesi 110, 150' de bulunan bir XOR kapısı koy.

A.X := 250;

A.Y := 35;

B.X := 300;

B.Y := 35;

ObjectList^.Insert(New(PLineLogic, Init(A, B)));

(250, 35) - (300, 35) arasını kısa devre yap.

A.Y := 55;

B.Y := 55;

ObjectList^.Insert(New(PLineLogic, Init(A, B)));

(250, 55) - (300, 55) arasını kısa devre yap.

A.Y := 75;

B.Y := 75;

ObjectList^.Insert(New(PLineLogic, Init(A, B)));

(250, 75) - (300, 75) arasını kısa devre yap.

A.Y := 95;

B.Y := 95;

ObjectList^.Insert(New(PLineLogic, Init(A, B)));

(250, 95) - (300, 95) arasını kısa devre yap.

A.Y := 115;

B.Y := 115;

ObjectList^.Insert(New(PLineLogic, Init(A, B)));

(250, 115) - (300, 115) arasını kısa devre yap.

ObjectList^.Insert(New(PTL7483, Init(150, 20)));

Sol üst köşesi 150, 20 ' de bulunan bir 7483 koy.

A.X := 110;

A.Y := 395;

B.X := 150;

B.Y := 395;

ObjectList^.Insert(New(PLineLogic, Init(A, B)));

(110, 395) - (150, 395) arasını kısa devre yap.

E.PointType := logInput + logOutput;

ObjectList^.Insert(New(PGrPoint,
Init(305, 255, A, A, A, A, E, 'V')));

(305, 255) ' e 'V' noktasını,

ObjectList^.Insert(New(PGrPoint,
Init(305, 275, A, A, A, A, E, 'W')));

(305, 275) ' e 'W' noktasını,

ObjectList^.Insert(New(PGrPoint,
Init(305, 295, A, A, A, A, E, 'X')));

(305, 295) ' e 'X' noktasını,

ObjectList^.Insert(New(PGrPoint,
Init(305, 315, A, A, A, A, E, 'Y')));

(305, 315) ' e 'Y' noktasını,

A.X := 250;

A.Y := 235;

B.X := 250;

B.Y := 215;

ObjectList^.Insert(New(PLineLogic, Init(A, B)));

(250, 235) - (250, 215) arasını kısa devre yap.

A.X := 250;

A.Y := 215;

B.X := 150;

B.Y := 215;

ObjectList^.Insert(New(PLineLogic, Init(A, B)));

(250, 215) - (150, 215) arasını kısa devre yap.

A.X := 150;

A.Y := 215;

B.X := 150;

B.Y := 195;

ObjectList^.Insert(New(PLineLogic, Init(A, B)));

(150, 215) - (150, 195) arasını kısa devre yap.

A.X := 250;

A.Y := 255;

B.X := 300;

B.Y := 255;

ObjectList^.Insert(New(PLineLogic, Init(A, B)));

(250, 255) - (300, 255) arasını kısa devre yap.

A.Y := 275;

B.Y := 275;

ObjectList^.Insert(New(PLineLogic, Init(A, B)));

(250, 275) - (300, 275) arasını kısa devre yap.

A.Y := 295;

B.Y := 295;

ObjectList^.Insert(New(PLineLogic, Init(A, B)));

(250, 295) - (300, 295) arasını kısa devre yap.

A.Y := 315;

B.Y := 315;

ObjectList^.Insert(New(PLineLogic, Init(A, B)));

(250, 315) - (300, 315) arasını kısa devre yap.

ObjectList^.Insert(New(PXorLogic, Init(60, 290)));

Sol üst köşesi (60, 290) ' da bulunan bir Xor kapısı koy.

ObjectList^.Insert(New(PXorLogic, Init(110, 310)));

Sol üst köşesi (110, 310) ' da bulunan bir Xor kapısı koy.

ObjectList^.Insert(New(PXorLogic, Init(60, 330)));

Sol üst köşesi (60, 330) ' da bulunan bir Xor kapısı koy.

ObjectList^.Insert(New(PXorLogic, Init(110, 350)));

Sol üst köşesi (110, 350) ' de bulunan bir Xor kapısı koy.

ObjectList^.Insert(New(PTL7483, Init(150, 220)));

Sol üst köşesi (150, 220) ' de bulunan bir Xor kapısı koy.

Ciz;

Devreyi çiz.

OutTextXY(50, GetMaxY - 60,

'Press << ESC >> to quit');

OutTextXY(50, GetMaxY - 20,

'Press << S >> to save this experiment as DENEY15.STM');

Ekrana gerekli açıklamaları yaz.

Calistir;

Programın çalışması için gerekli alt programlara kontrolü devret.

ESC tuşuna basınca da grafiği kapatarak programı sona erdir.

```
ObjectList^.Done;  
closegraph;  
end.
```

Bu programda noktalı zemine yazılan ve devrenin elemanlarının programa tanıtıldığı kısıma istenen tanımlamalar yazılıpü programda iken 'S' tuşuna basılırsa devre DENEY15.STM olarak kayıt edilir. Böylece 15 nolu deney istenilen devre olacak şekilde değiştirilmiş olur. Aynı işlem diğer deney dosyaları üzerinde de yapılabilir. Yazılan bu bilgisayar programı Turbo Pascal 6.0 ile derlenerek çalıştırılmalıdır.



5. DENEYLERİN YAPILMASI

Bu deneylerin yapılabilmesi bilgisayarda bilgisayar programının saklı bulunduğu alt dizine gidilerek MYAPP.EXE programı çalıştırılmalıdır. Örneğin, bu programlar DENEY alt dizininin içinde saklı ise aşağıdaki komutlar bilgisayara verilmelidir.

CD\DENEY

MYAPP

Yukarıdaki komutlar sırası ile verildikten sonra Sayısal elektronik Deney Seti Simülasyonu Programı çalışacaktır. Böylece aşağıdaki gibi bir ekran görüntüsü elde edilecektir.



Şekil 5.1. Programın ilk ekran görüntüsü

Programda hem klavye, hem de mouse kontrolü bulunmaktadır. Bu sayede, hem F10 tuşuna basılarak hem de mouse ile imleç hareket ettirilerek menüdeki komutlardan birisi çalıştırılabilir. Menüden verilebilecek komutlar şunlardır :

1. File

- Save : Çalışılan deney dosyasını kayıt etmeyi sağlar.
- Load : Çalışılacak deney dosyasının hafızaya okunmasını sağlar.
- Quit : Programdan çıkmayı sağlar.

Bu menüdeki Save ve Load komutları, ileriki vadede yapılabilecek olan ilaveler için eklendi. Aslında bu program için, şu anda bir işe yaramamaktadır.

2. Options

- Get Command : Programa komut satırından bir komut verebilmeyi sağlar.

Bu komutlardan birisi DENEY n komutudur. Burada n yerine 1 ile 15 arası bir sayı olmalıdır. Diğer bir komut ise, a ile z arası bir harftir. Bu komut aynı zamanda deneyde iken direkt de çalıştırılabilir. Bu komut sayesinde ekrandaki değişken tablosundaki ilgili giriş değişkeninin değerini değiştirebilmeyi sağlamaktadır. Örneğin, A komutu verilirse, deneyde A olarak isimlendirilen giriş noktasının değeri değiştirilir.

- Change Clock Time (n) : Otomatik darbe modunda iken, Z değişkeninin değerinin hangi sıklıkla otomatik değiştirileceği bu kısma yazılır. Eğer, hiç bir değer yazılmadan ENTER tuşuna basılırsa, parantez içindeki n değeri korunur. Sayı ne kadar küçülürse, program o kadar hızlı bir şekilde Z değişkeninin değerini değiştirecektir.

- Change Clock Mode : Z değişkeninin değeri otomatik veya elle değiştirilecekse bu menüdeki komutlar seçilmelidir. Program çalıştırıldığında, elle değiştirilecek (Manual) komutunu kendisi çalıştırmış durumda olacaktır.

- Automatic : Z değişkeninin değeri otomatik olarak program tarafından belirli zaman aralıklarında değiştirilecek ise, bu komut seçilmelidir.

- Manual : Z değişkeninin değeri otomatik olarak program tarafından belirli zaman aralıklarında değiştirilmeyecek ise, bu komut seçilmelidir.

- Change Experiment : Ekranda çalışılan deneyin değiştirilerek istenilen deneye getirilebilmesi için bu komut çalıştırılarak 01 ile 15 arasındaki deney numaralarından birisi seçilmelidir. Bu deneyleri sıra ile yaptırtmak faydalı olacaktır.

Deneyleri sıra ile yapabilmek için, öncelikle *Change Experiment* tan 01 nolu deneyi seçtirin. Deney ile yapılacak işlemler tamamlandıktan sonra, her seferinde *Change Experiment* komutu ile bir sonraki deneye geçilir. Program ile ilgili işlemlerinizi tamamlayınca Quit komutu ile MS-DOS işletim sistemine geri dönebilirsiniz.

Her deneyde, eğer giriş olarak tanımlanan bir düğüm noktasının değerini ters çevirebilmek için (yani, 1 ise 0 ve 0 ise 1 yapmak), o düğüm noktasının sol üst köşesinde yazan harfi klavyeden tuşlamanız yeterli olacaktır. Bu giriş değişikliğine göre elde edilen çıkış değerleri derhal ekranın sağ tarafındaki tabloda belirtilen 26 tane çıkıştan (A - Z arası) devrede kullanılanların değerlerini son haliyle yazacaktır.

Eğer deneyde, Z girişi kullanılıyorsa devrede otomatik tetikleme girişi olarak kullanılıyordur. Bu Z girişinin değeri ister otomatik ister elle değiştirilmesi sağlanabilir. Program ilk çalıştırıldığında elle (*Manual*) çalışma konumundadır. Eğer Z' nin değerinin otomatik olarak değişmesi isteniyorsa, *Automatic* komutu verilmelidir.

Eğer otomatik çalışma modunda iken değerlerin değişme hızı değiştirilmek istenirse, *Change Clock Time* komutu verilerek burada sıfırdan büyük olmak koşulu ile bir sayı girilmelidir. Burada parantez içinde bir sayı yazılıdır. Bu sayı devrede kabul edilen bekleme süresidir. Burada devrenin çalışması yavaş ise, bu sayıdan küçük bir sayı ve hızlı ise büyük bir sayı girilmelidir. Bu sayı devrenin her bir değerdeki bekleme süresidir.

Deneyler arasındaki geçişlerde her seferinde elle tetikleme komutunu vermek faydalıdır.



6. SONUÇLAR VE ÖNERİLER

Bu projede, hedeflenen işlem gerçekleştirilebilmiştir. Bilgisayar programı kusursuz olarak çalışmaktadır. Elbette ki bu haliyle ticari gayeden oldukça uzaktır. Daha profesyonel bir projenin geliştirilebilmesi, ancak 2 - 3 yıllık bir çalışma sonucunda veya 2 - 3 proje uygulayıcı öğrenci tarafından ele alınmasına bağlıdır.

Projenin sonucunda elde edilen programın en kötü tarafı kendisine ait bir editörünün olmamasıdır. Bu yüzden, bir deney üzerinde değişiklik yapılacağı zaman, ilgili deney numarasına ait MAKEnn.PAS programı üzerinde bilgisayar programcısı olarak değişiklik yapılması zorunlu hale gelmiştir. Buna rağmen, proje ileri vadedeki geliştirme çalışmaları düşünülerek editör ilavesine müsait olacak şekilde tasarlanmıştır.

Projenin diğer bir aksaklığı ise, deneylerde kullanılan malzemeler üzerinde değişiklik yapılacağı zaman da bilgisayar programcılığı düzeyinde müdahale edilebilmesinin zorunlu olmasıdır. Proje profesyonel amaca uygun hale getirilirken, eleman editörü bölümünün de eklenmesi ve makro elemanların tanımlanabilmesi de gereklidir. Eğer, projede kullanılan malzemelerin şekilleri ve çalışma prensipleri konusunda bu haliyle değişiklik yapılmak isteniyorsa MYLOGICS.PAS isimli tanımlama dosyasında programcılık düzeyinde uğraşılması gereklidir. Bu dosyada tanımlanan elemanlar ve TYPE isimleri aşağıda verilmiştir :

Tablo 6.1. MYLOGICS.PAS programı içinde tanımlanan elemanlar ve tip isimleri

ELEMAN	PROGRAMDA KULLANILAN TYPE İSMİ
DÜĞÜM NOKTASI	TGRPOINT
ÇİZGİ (KISA DEVRE ELEMANI)	TLNELOGIC
LOJİK "0" NOKTASI. (GND)	TGND
LOJİK "1" NOKTASI (Vcc)	TVCC
AND KAPISI	TANDLOGIC
OR KAPISI	TORLOGIC
NOT KAPISI	TNOTLOGIC
XOR KAPISI	TXORLOGIC
NAND KAPISI	TNANDLOGIC
NOR KAPISI	TNORLOGIC
POZİTİF KENAR TETİKLEMELİ JK - FF	TPETJKFF
NEGATİF KENAR TETİKLEMELİ JK - FF	TNETJKFF

POZİTİF KENAR TETİKLEMELİ RS - FF	TPETRSFF
NEGATİF KENAR TETİKLEMELİ JK - FF	TNTRSFF
POZİTİF KENAR TETİKLEMELİ D - FF	TPETDFF
NEGATİF KENAR TETİKLEMELİ D - FF	TNETDFF
7493 ENTEGRE DEVRE ELEMANI	TTL7493
7483 ENTEGRE DEVRE ELEMANI	TTL7483

Proje, bu haliyle 21 ayrı dosyada toplam 8500 satırdan (174 dosya kağıdı) oluşmaktadır. Bu dosyalar birleştirilerek tek bir dosya halinde program haline getirilirse, proje biraz daha basitlik kazanacaktır. Bunun için, öncelikle yüksek hafızayı kullanmayı sağlayan alt programların projeye eklenmesi gerekmektedir.

KAYNAKLAR

- Altan, Naci, 1993. " Turbo Pascal 7.0 ". ALFA, 975-7368-31-8, 472s, İstanbul.
- Banger, Gürcan, 1993. " Turbo Pascal 7 ". Bilim Teknik Yayınevi, 975-540-025-7, 756s, İstanbul.
- Barkana, Atila ve Taşlıca, Ali Osman, 1994. " Mikroişlemciler ve Sayısal Sistemler", Milli Eğitim Bakanlığı, 975-11-0858-6, 887s, Eskişehir.
- Erdun, Hakan, 1993. " Turbo ve Borland C & Pascal ile Grafik ". BETA, 975-486-307-5, 752s, İstanbul.
- İTÜ Elektrik - Elektronik Fakültesi Devreler ve Sistemler Anabilim Dalı, 1987. " Lojik Devreler Laboratuvarı Deneyleri ", İTÜ Ofset Baskı Atölyesi, 70s, İstanbul.
- King Instrument Electronics Co. " KL - 300 Digital Logic Lab Experiment Manual ". Sayısal Elektronik Deney Seti' nin deney kitabı, 364s, İngiltere.
- Rashid, Muhammed H., 1989. " SPICE For Circuits and Electronics Using Pspice ". Prentice Hall, 237s, USA.
- Schumann, Michael, 1992. " Turbo Vision Toolbox ". Abacus, 1-557550-147-2, 412s, USA.
- Teközgen, Erdoğan, 1992. " Dijital Elektronik Deneyleri ve Projeleri ". Bizim Büro Basımevi, No:10598, 243s, Ankara.
- Yarcı, Kemal, 1992. " Dijital Elektronik ". Yüce Yayınları, 183s, İstanbul.