

YANI SERVİNİ  
JANETA SERVİNİ

---

# DİJİTAL ELEKTRONİK VE MİKROİŞLEMCİLER

---

II (ikinci) Sınıf – Elektroteknik Mesleđi  
Elektronik ve Telekomüniksyon Elektro Teknisyeni

Manastır, 2011

**Dijital Elektronik ve Mikroişlemciler**

II (İkinci) Sınıf, Elektroteknik Mesleği,  
Elektronik ve Telekomünikasyon Elektro Teknisyeni

<b>Yazarlar:</b>	Yük.Lis.El.Müh. Yani Servini Mr. Janeta Servini, Yük.Lis.El.Müh
<b>Değerlendiriciler:</b>	Prof. Dr. Mirka Popnikolova Radevska Yük.Lis.El.Müh. Petse Petrov Yük.Lis.El.Müh. İlçe Atsevski
<b>Redaksiyon:</b>	Prof. Dr. Arif Ago
<b>Düzeltilici:</b>	Dr. Aktan Ago
<b>Çeviri:</b>	Ervin Salih
<b>Bilgisayar yapımı:</b>	Yani Servini
<b>Çizimler yapımı ve grafik düzenleme:</b>	Yani Servini
<b>Kapak tasarım çözümü:</b>	Yani Servini
<b>Kapak tasarımı:</b>	Nikola Sotirovski
<b>Yayıncı:</b>	M.C. Eğitim ve Bilim Bakanlığı, Mito Hacı-Vasilev Yasmin Caddesi bb, Üsküp
<b>Baskı:</b>	Grafički Tsentar Dooel, Üsküp
<b>Tiraj:</b>	20

Makedonya Cumhuriyeti Eğitim ve Bilim Bakanlığı'nın 15.03.2011 tarihli no: 22-293/3 kararıyla bu ders kitabının kullanımına izin verilmiştir.

CIP - Каталогизација во публикација  
Национална и универзитетска библиотека "Св.Климент Охридски" , Скопје  
АВТОР: Сервини, Јани - автор  
ОДГОВОРНОСТ: Сервини, Жанета - автор  
НАСЛОВ: Дигитална електроника и микропроцесори : II(втора) година - електротехничка струка : електротехничар за електроника и телекомуникации  
ИМПРЕСУМ: Скопје : Министерство за образование и наука на Република Македонија, 2011  
ФИЗИЧКИ ОПИС: 266 стр. : илустр. ; 29 см  
ISBN: 978-608-226-317-5  
УДК: 621.38.049.77(075.3), 004.31(075.3)  
ВИД ГРАЃА: монографска публикација, текстуална граѓа, печатена  
ИЗДАВАЊЕТО СЕ ПРЕДВИДУВА: 08.11.2011  
COBISS.MK-ID: 89132298

*Verilen bir sorunun çözümlenmesinin en önemli kazancı, onun çözümünün bulunması değil, bu çözümün bulunması sırasında elde edilen enerjidir.*

*Anonim*



# İÇİNDEKİLER

Önsöz.....	ix
<b>1 Sayı Sistemleri ve Kodları.....</b>	<b>1</b>
<b>I) DİJİTAL ELEKTRONİĞE GİRİŞ.....</b>	<b>3</b>
1.1 Temel Terimler .....	3
1.2 Bilgi ve Bilginin Kodlanması.....	6
1.3 Bilgi Miktarı için Ölçü Birimleri.....	7
1.4 Bilgi Çeşitleri.....	8
1.5 Dijital Devrelerin ve Ağların Ayrımı.....	9
<b>II) SAYI SİSTEMLERİ VE KODLARI .....</b>	<b>12</b>
1.6 Temel Terimler .....	12
1.7 Sayı Sistemleri .....	13
1.7.1 Sayıların Herhangi Sayı Sisteminden Onlu Sayı Sistemine Dönüşümü .....	14
1.7.2 İkili Sayı Sisteminden On altılı ve Sekizli Sayı Sistemine ve Tersine Dönüşüm .....	16
1.7.3 Onlu Sayı Sisteminden Herhangi Sayı Sistemine Dönüşüm .....	17
1.7.4 İkili Sayı Sistem Aritmetiği .....	18
1.7.5 Pozitif ve Negatif Sayıların İşaretlenmesi .....	20
1.7.6 Çift Tümleyenle İşaretleme .....	22
1.8 İkili Kodlar .....	24
1.8.1 Sayısal Kodlar .....	25
1.8.2 Alfasayısal Kodlar.....	28
1.9 Belirtik ve Örtük Değerler .....	30
Tekrarlama Soruları ve Ödevleri .....	31
<b>2 Boole Cebri .....</b>	<b>35</b>
2.1 Giriş .....	37
2.2 Aksiyomlar ve Mantıksal İşlemler .....	37
2.3 Teoremler ve Kanunlar .....	39
2.4 Anahtarlamalı Fonksiyonlar ve Onların Gösterilmesi .....	41
2.4.1 Tablolu Gösterim .....	42
2.4.2 Analitik Gösterim.....	43
2.4.2.1 Tamamen Verilmiş Fonksiyonlar .....	44
2.4.2.2 Kısmen Verilmiş Fonksiyonlar .....	45
2.4.3 Bir Şekilden Başka Şekile Dönüşüm .....	46

<b>2.5 Standart Mantıksal Fonksiyonlar .....</b>	<b>49</b>
<b>2.6 Anahtarlama Fonksiyonlarının Minimizasyonu.....</b>	<b>50</b>
2.6.1 Analitik Minimizasyon Yöntemi.....	51
2.6.2 Karno Minimizasyon Yöntemi .....	51
2.6.2.1 Karno Yönteminin Kullanımı .....	56
2.6.2.2 ANB/BNB Şeklinde Verilen Fonksiyonların Minimizasyonu.....	60
2.6.2.3 Kısmen Verilmiş Fonksiyonların Minimizasyonu .....	61
<b>2.7 Anahtarlama Ağlar .....</b>	<b>63</b>
2.7.1 Temel Mantıksal Devreler.....	63
2.7.2 Diğer Temel Mantıksal Devreler .....	65
2.7.2.1 Arabellek Devresi .....	66
2.7.2.2 Üç Durumlu Arabellek Devresi .....	66
2.7.2.3 Karşılıklı (İletim) Geçiti .....	69
2.7.3 Anahtarlama Ağlarının Analizi.....	70
2.7.4 Anahtarlama Ağlarının Sentezi.....	74
<b>Tekrarlama Soruları ve Ödevleri .....</b>	<b>79</b>
<b>3 Birleşimsel Ağlar .....</b>	<b>85</b>
<b>I) Aritmetik-Mantıksal İşlemler Gerçekleştiren Devreler.....</b>	<b>87</b>
<b>3.1 Giriş .....</b>	<b>87</b>
<b>3.2 Toplama ve Çıkarma Devreleri.....</b>	<b>87</b>
3.2.1 İkili Toplayıcılar .....	87
3.2.2 Tümlenme Devresi .....	90
3.2.3 Çıkarma Devresi .....	91
<b>3.3 Dijital Karşılaştırıcı .....</b>	<b>92</b>
<b>II) Anahtarlama Matrisler.....</b>	<b>93</b>
<b>3.4 Giriş .....</b>	<b>93</b>
<b>3.5 Kodlayıcılar ve Kod Çözümler .....</b>	<b>94</b>
3.5.1 Kodlayıcı .....	94
3.5.2 Öncelikli Kodlayıcı .....	97
3.5.3 Kod Çözümler.....	98
3.5.4 NBCD-den-7 Bölümlüye Kod Çözümler .....	100
<b>3.6 Çoğullayıcı ve Çoğullama Çözümler .....</b>	<b>101</b>
3.6.1 Çoğullayıcı.....	102
3.6.2 Çoğullama Çözümler .....	104
<b>III) Programlanabilir Mantıksal Yapılar.....</b>	<b>107</b>
<b>3.7 Giriş ve Ayrım .....</b>	<b>107</b>
<b>3.8 PROM Belleği.....</b>	<b>108</b>
<b>Tekrarlama Soruları ve Ödevleri .....</b>	<b>112</b>

<b>4 Flip-Floplar .....</b>	<b>115</b>
4.1 Giriş ve Temel Terimler.....	117
4.2 SR Flip-Flop.....	120
4.2.1 OYA Türünden SR Flip-Flop .....	120
4.2.2 OVE Türünden SR Flip-Flop .....	122
4.2.3 Pals-Sinyalinin Seviyesiyle Palslanmış SR Flip-Flop .....	123
4.2.4 Pals-Sinyalinin Kenarıyla Palslanmış SR Flip-Flop .....	126
4.2.5 Master-Slave Yapılı SR Flip-Flop .....	127
4.3 JK Flip-Flop .....	130
4.4 T Flip-Flop.....	132
4.5 D Flip-Flop .....	133
4.5.1 Kitleme Devresi .....	135
4.5.2 Temel Bellek Hücresi .....	136
4.5.3 D Flip-Flop Mantığının Değişmesi .....	137
4.6 Tümeleşik Flip-Floplar .....	138
Tekrarlama Soruları ve Ödevleri .....	139
<b>5 Yazmaçlar.....</b>	<b>143</b>
5.1. Giriş ve Temel Terimler ve Kavramlar.....	145
5.1. Sabit Yazmaç.....	147
5.2. Ötelemeli Yazmaç.....	150
5.3. Dairesel Yazmaç .....	153
5.4. İkiyönlü Ötelemeli Yazmaç .....	154
5.5. Dizisel Girişli ve Karışık Çıkışlı Ötelemeli Yazmaç .....	155
5.6. Karışık Girişli ve Dizisel Çıkışlı Ötelemeli Yazmaç .....	156
5.7. Evrensel Yazmaç .....	157
Tekrarlama Soruları ve Ödevleri .....	158
<b>6 Sayaçlar .....</b>	<b>161</b>
6.1 Giriş ve Temel Terimler ve Kavramlar.....	163
6.2 Sayaçların Tabanı ve Kapasitesi .....	164
6.3 Sayaçların Ayrımı.....	165
6.4 Asenkron Sayaçlar.....	166
6.4.1 İkili Asenkron Sayaç.....	166
6.4.2 Geriye Asenkron İkili Sayaç.....	169
6.4.3 İkili Asenkron İkiyönlü Sayaç.....	171
6.4.4 Rastgele Tabanlı Asenkron Sayacın Projelenmesi .....	172
6.4.4.1 5 Tabanlı Asenkron Sayaç.....	173
6.5. Senkron Sayaçlar .....	175
6.5.1 İkili Senkron Sayaç .....	175
6.5.2 İkili Senkron Geriye Sayaç .....	177

6.5.3 İkili Senkron İkiyönlü Sayaç .....	177
6.5.4 Rastgele Tabanlı Senkron Sayacın Projelenmesi.....	178
6.5.4.1 10 Tabanlı Senkron Sayacın Sentezi.....	179
<b>6.6 Dairesel Sayaçlar .....</b>	<b>181</b>
6.6.1 5 Tabanlı Dairesel Sayaç .....	182
6.6.2 Onlu Dairesel Sayaç .....	183
<b>Tekrarlama Soruları ve Ödevleri .....</b>	<b>185</b>
<b>7 Bellek Bileşenleri .....</b>	<b>189</b>
7.1 Giriş .....	191
7.2 Bellek Hiyerarşisi .....	191
7.3 Belleğin İç organizasyonu ve Temel Terimler ve Kavramlar.....	192
7.4 Bellek Bileşenlerin Ayrımı.....	201
7.5 ROM Bellek Bileşenleri.....	203
7.6 RAM .....	203
7.6.1 SRAM Bellek Hücreleri .....	204
7.6.2 Asenkron Okuma ve Yazma.....	210
7.6.3 Senkron Okumanın Bellek Döngüsü .....	211
7.6.4 Senkron Yazmanın Bellek Döngüsü .....	213
<b>Tekrarlama Soruları ve Ödevleri .....</b>	<b>214</b>
<b>8 Dijital-Analog ve Analog-Dijital Dönüşüm .....</b>	<b>217</b>
8.1 Giriş .....	219
8.2 Dijital-Analog Dönüşüm .....	220
8.3 Temel Denklemler, Terimler ve İletim Özelliği.....	221
8.4 D/A Dönüştürücüler .....	222
8.4.1 R/2 <sup>n</sup> R Ağırlıklı Direnç Ağılı D/A Dönüştürücü .....	223
8.4.2 R/2R Basamaklı Direnç Ağılı D/A Dönüştürücü.....	225
8.5 Analog-Dijital Dönüşüm .....	227
8.6 Temel Terimler ve Kavramlar .....	228
8.7 Karakteristik Parametreler ve İletim Özelliği.....	231
8.8 A/D Dönüştürücülerin Ayrımı ve Çeşitleri.....	235
8.8.1 Paralel ADC.....	236
8.8.2 D/A Dönüşümüne Dayanan A/D Dönüştürücüler.....	238
8.8.2.1 Sayaç Rampalı ADC .....	239
8.8.2.2 Ardaşık Yaklaşımla ADC .....	241
8.8.3 Tümüleşik Devreye Dayanan ADC .....	243
8.8.3.1 Tek Eğimli A/D Dönüştürücü .....	243
8.8.3.2 Çift Eğimli A/D Dönüştürücü .....	245
8.8.4 Delta-Sigma A/D Dönüştürücüler .....	247
<b>Tekrarlama Soruları ve Ödevleri .....</b>	<b>249</b>
<b>Kaynakça .....</b>	<b>253</b>



## ÖNSÖZ

“Dijital Elektronik ve Mikroişlemciler” ders kitabı, elektronik ve telekomünikasyon elektro teknisyen eğitim profili elektroteknik mesleğinden ikinci yıl sınıflarında incelenen, aynı isimli ders için mevcut eğitim planlarına ve programlarına uyumlu olarak yazılmıştır. Yazı öngörülen öğretim içeriklerini tamamıyla kapsayarak, sekiz tematik bütününe ayrılmıştır. Sunulan malzeme esastır, çünkü elektronik aletlerin, cihazların ve aygıtların temel ve ayrılmaz işlevsel bölümleri olan elemanlar ve bileşenleri olduğu gibi, bilgisayarlara ve mikroişlemcilere dayalı sistemleri inceliyor.

- (1) Birinci konusal birimi olan SAYI SİSTEMLERİ VE KODLARI bilgilerinin dijital şekilde kodlama şekillerinin tanımlanmasıyla, bilgi miktarının ölçülmesiyle ve farklı kriterlere göre dijital devrelerin ve ağların ayrımıyla başlıyor. Devamda dijital elektronik ve bilgisayar tekniği için temel sistemler olan ikili ve on altılı sayı sistemlerine ağırlık verilmiştir. Özellikle, ikili, on altılı ve onlu sayı sistemleri arasında bir sayı sisteminden başka sayı sistemine dönüşüm, ardından, ikili aritmetiğin temel işlemlerinin: toplama, çıkarma, çarpma ve bölmenin gerçekleştirme kuralları, negatif sayıların ikili şekilde işaretlenmesi ve sayısal (nümerik) ve metinsel (alfa-sayısal) veriler için ikili kodlar inceleniyor.
- (2) İkinci konusal biriminde, BOOLE CEBRİ, mantık cebirinin başlangıç aksiyomlarına, mantıksal işlemlere, kanunlara ve teoremlere dikkat verilmiştir. Ardından temel anahtarlama işlevleri, onların analitik, tablololu ve grafiksel şekilleri ve bir şekilden başka bir şekile geçmeleri açıklanmıştır. Bu konunun önemli bölümü, Karno kartları yöntemiyle anahtarlama fonksiyonlarının minimizasyonudur. Bu bölümü daha kolay anlamak için büyük sayıda işlenmiş örnekler verilmiştir. Yukarıda belirlenen konular dışında, bu konusal biriminde standart mantıksal devreler: VE, VEYA, DEĞİL, O-VE (Olumsuz VE), O-VEYA (Olumsuz VEYA), D-VE (Dışlamalı VE), D-VEYA (Dışlamalı VEYA) ve üç durumlu arabellek devresi için sembolleri tanımlanmıştır. Özel bir önem, gerçekleştirdikleri işlevlerin belirlenmesi için daha basit mantıksal diyagramların incelenmesine ve VE-VEYA (O-VE) ve VEYA-VE (O-VEYA) türünden iki seviyeli mantıksal ağların sentezine verilmiştir.
- (3) Üçüncü konusal biriminde, BİRLEŞİMSEL AĞLAR bölümünde, dijital karşılaştırıcının, toplama, tamamlama ve çıkarma devrelerin, anahtarlama matrisler kodlayıcı, kod çözücü, çoğullayıcı ve çoğullama çözücünün mantıksal yapıları ve çalışma prensipleri inceleniyor. Sonunda progamlanabilir mantıksal yapıların ayrımı verilmiş ve PROM bellek yapılarının çalışma şekli açıklanmıştır.
- (4) Dördüncü konusal biriminde, FLİP-FLOPLAR bölümünde, pals sinyalinin seviyesine karşılık veren, standart ve master-slave düzenleştirmeli temel asenkron ve senkron (palslı) ardışık devrelerin çalışması inceleniyor. Daha belirgin olarak SR, JK, T ve D flip-flopları (iki durumlu devreleri) ve aynı zamanda pals sinyalin yükselen veya düşen kenarın meydana gelmesiyle etkinleşen anahtarlama flip floplarının çalışması inceleniyor.

Bu bölümde kilitleme devreleri (latch) ve elementer RAM bellek hücresi de işleniyor. Bu arada doğruluk tabloları, mantıksal denklemler ve zamansal diyagramlar kullanılıyor. Ek olarak flip-flopların (iki durumlu devrelerin) karşılıklı dönüşümüne ve onların daha karmaşık ardışık bileşenlerin gerçekleşmesinde uygulama olanaklarına ilişkin bölüm ayrılmıştır.

- (5) Beşinci konusal biriminde, YAZMAÇLAR bölümünde, yazmaçların mantıksal yapıları, onların temel çalışma prensipleri ve verilerin girildiği (doldurulduğu) ve okunduğu şekline göre yazmaçların ayrımı inceleniyor. Sabit yazmaçlar, ötelemeli yazmaçlar, kombine girişli yazmaçlar, kombine çıkışlı yazmaçlar ve tümel yazmaçlar, işlevlerine ve uygulanmalarına göre birbirinden farklıdır ve dijital sistemlerinde standart bileşenler olarak kullanılıyorlar.
- (6) Altıncı konusal birimi, SAYAÇLAR, bizi sayaçların asenkron ve senkron olarak ayrılmasıyla, sayaçların mantıksal yapıları ve çalışma prensibiyle tanıtıyor. Bu konusal biriminde zamansal diyagramlar aracılığıyla daha detaylı şekilde ikili sayaçların davranmaları, gerisayım sayaçları ve iki yönlü sayaçlar açıklanmıştır. Bu bölümde rastgele sayma tabanı (modülü) ile daha basit asenkron ve senkron sayaçların projelenmesiyle ilgili iki örnek verilmiştir. Sonunda 5 ve 10 tabanı ile çevrim sayaçları (beşli ve onlu sayaç) incelenmiştir.
- (7) Yedinci konusal bütünü, BELLEK BİLEŞENLERİ, bizi bellek bileşenleri ve belleğin organizasyonu ile ilgili temel terimlerle ve kavramlarla tanıtıyor. Belleklerin ayrımı farklı bellek çeşitlerin: ROM, PROM, EPROM, EEPROM, RAM karşılaştırılmasını sağlıyor, bununla beraber ise bellek tümleşik devreler arasındaki benzerliklerin ve farklılıkların da anlaşılması sağlanıyor. Sabit RAM (SRAM) bellek hücresinin mantıksal yapısının, kontrol sinyallerin rollerinin ve işlevsel tablosunu kullanarak çalışma prensibinin açıklanmasına özel dikkat verilmiştir. Bu bölümde belleğin adreslemesi hakkında da bahsedilmiştir, zamanlama diyagramlarının incelenmesiyle ise basit bir şekilde, bellekte okuma ve yazma süreçlerin gerçekleşmesi sunulmuştur.
- (8) Sekizinci konuda, ANALOG-DİJİTAL VE DİJİTAL-ANALOG DÖNÜŞÜMÜ (ADC ve DAC) bölümünde, ADC ve DAC süreçlerine ilişkin temel terimleriyle ve ADC ve DAC'ın farklı yöntemlerinin prensipleriyle tanışıyoruz. Bu son konusal biriminin sonunda, ağırlıklı ve basamaklı direnç ağı ile D/A dönüştürücülerin, paralel A/D dönüştürücünün, saymalı rampalı ve ardışık yaklaşımlı A/D dönüştürücülerin, tümleşik devrelere dayalı A/DK dönüştürücülerin ve delta-sigma A/D dönüştürücülerin çalışma prensibi açıklanmıştır.

Tematik birimlerinde işlenen malzemenin büyük kısmının yazarı, yüksek lisans elektroteknik mühendisi ve Manastır'da "Gyorgi Naumov" orta belediye teknik okul (OBTO) öğretmeni Yani Servini'dir, ders kitabının bir parçası olan tekrarlama soruları ve ödevlerin yazarı ise yüksek lisans elektroteknik mühendisi ve Manastır'ın "Gyorgi Naumov" OBTO öğretmeni Mr. Janeta Servini'dir.

Yazılan malzeme son bir kaç yılda güncel olan mesleki kitaplara dayanarak hazırlanmış. Yazarlar olarak, metinde açıklamaların, öğrencilerin yaşına uygun, ancak bu dersten öğretim programının amaçlarının yerine getirilmesi yönünde açık, anlaşılır, geniş kapsamlı ve ayrıntılı olmasına uğraştık. Bu yönde büyük çaba harcayarak, mesleki açıdan olduğu gibi pedagojik ve metodolojik açıdan, hem genişlikte hem derinlikte, öğretim içeriklerin sunarken, nicelik ve kalite azalmadan, uygun yazma tarzı kullanmaya dikkat ediyorduk.

Devrelerin çalışma şeklini açıklarken, kıvamlı (uyumlu) işretleme sistemleri, uygun doğruluk tabloları ve işlevsel tabloları, mantıksal denklemler ve uygun numaralandırmalı formüller, prensipiyel mantıksal ve elektrik şemaları kullandık. Bu arada, mantıksal devreler ve elektronik el-emanlar için standart semboller uygulayarak, şemaların karakteristik noktalarında gerilimlerin zamanlama diyagramları kullanılmıştır.

İkinci sınıfta mesleki dersin söz konusu olması ve bu kitabın amaçlı olan öğrencilerin matema-tikten temel bilgilerinin olmasını göz önüne alarak, ağırlığı devrelerin çalışma prensibinin daha detaylı ve daha kapsamlı açıklanmasına ve incelenmesine koyulmuş, matematiksel tarafın ise en az derecede olmasına özen gösterdik.

Ders kitabının metninde, işlenen metodolojik birimlerin esasını daha iyi anlamak için, her tematik birimi için ayrı olarak, birçok karakteristik örnekler seçilmiş ve çözülmüştür. Ek, olarak her tematik birimin sonunda, büyük sayıda farklı ağırlıkta tekrarlama için sorular ve ödevler verilmiştir. Onların cevaplanması ve çözmesiyle, öğrenciler alınan bilgiyi kontrol edebilir ve tes-pit edebilir ve aynı zamanda kalite seviyesini büyük ölçüde yükseltilebilir. Bu yüzden, soruların ve ödevlerin bu ders kitabının önemli bölümü olduğunu düşünüyoruz.

İncelenen malzemenin kapsamı ve derinliği bakımından yeterli büyük olduğunu göz önüne alarak, öğretmenler, bu dersten öğretim programın gerçekleşme ihtiyacına ve bu dersin okunduğu sınıflarda öğrencilerin kapasitesine bağlı olarak, belirli öğretim içeriklerin seçilmesine ve daha büyük ağırlık koyması olanağı verilmiştir. Bunun dışında, soruların ve ödevlerin verilmesi öğretmenlere farklı öğretim yöntemlerin uygulamasına yol açarak, öğrencilerin ek olarak kendi yaratıcılığının gelişmesini sağlıyor.

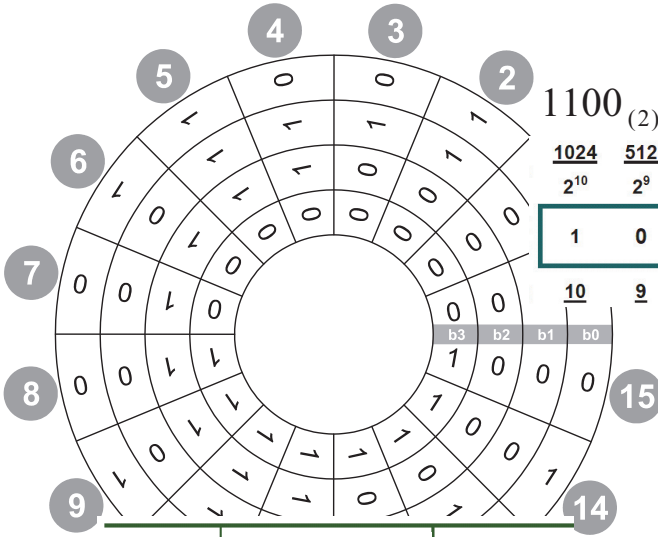
Böyle yaklaşımın, bu dersten ders veren öğretmen meslektaşlarına, uygun bilgi iletimi fonksi-yonunda ve her günlük çalışma görevlerinin yerine getirmesinde öğretim sürecinin kaliteli gerçekleştirilmeleri için büyük ölçüde yardım edeceğini içten ümit ediyoruz.

Sonunda bu ders kitabın final verziyonunun kalitesinin iyileşmesine, yapıcı ve iyi niyetli öner-iler ve notlarla büyük katkı veren değerlendicilere teşekkürlerimizi sunuyoruz.

Manastır, haziran 2010

Yazarlardan





$$1100_{(2)} = 1 \cdot 2^3 + 1 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0 = 8 + 4 + 1 = 12_{(10)}$$

$$\begin{array}{r} \underline{1024} \quad \underline{512} \quad \underline{256} \quad \underline{128} \quad \underline{64} \quad \underline{32} \quad \underline{16} \quad \underline{8} \quad \underline{4} \quad \underline{2} \quad \underline{1} \\ 2^{10} \quad 2^9 \quad 2^8 \quad 2^7 \quad 2^6 \quad 2^5 \quad 2^4 \quad 2^3 \quad 2^2 \quad 2^1 \quad 2^0 \end{array}$$

1	0	0	1	1	0	1	0	0	0	1
10	9	8	7	6	5	4	3	2	1	0

onlu	on altılı	ikili
0	0	0000
1	1	0001
2	2	0010
3	3	0011
4	4	0100
5	5	0101
6	6	0110
7	7	0111
8	8	1000
9	9	1001
10	A	1010
11	B	1011
12	C	1100
13	D	1101
14	E	1110
15	F	1111

# 1. SAYI SİSTEMLERİ VE KODLARI

Bu tematik birimini inceledikten sonra

- ⊕ Sinyallerin dijital iletiminin avantajlarını tanıyacaksınız;
- ⊕ Bilgileri dijital şekilde kodlama yöntemlerini tanıyacaksınız;
- ⊕ Bilgi miktarının ölçülmesini açıklayabileceksiniz;
- ⊕ Dijital devrelerin ve ağların ayırımını bileceksiniz ve farklı kriterlere göre açıklayacaksınız;
- ⊕ Sayı sistemlerini tanıyacaksınız ve ayırt edeceksiniz;
- ⊕ Sayıların bir sayı sisteminden başka sayı sistemine dönüşümünü açıklayacaksınız ve uygulayacaksınız;
- ⊕ İkili aritmetiği uygulayacaksınız;
- ⊕ Negatif sayıların ikili şekilde işaretlenmesini anlayacaksınız ve onunla ilgili ödevler çözeceksiniz;
- ⊕ Sayısal ve alfa-sayısal verilerin ikili kodlarını açıklayacaksınız;



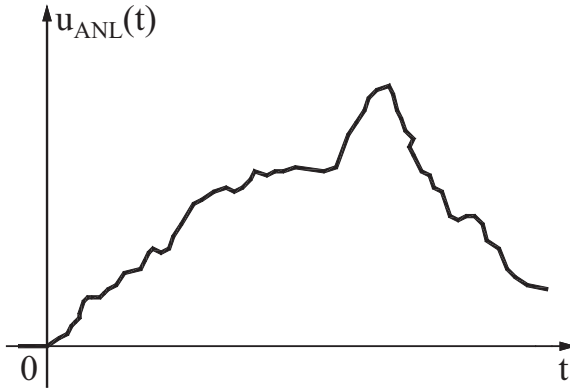
# I) DİJİTAL ELEKTRONİĞE GİRİŞ

## 1.1. TEMEL TERİMLER

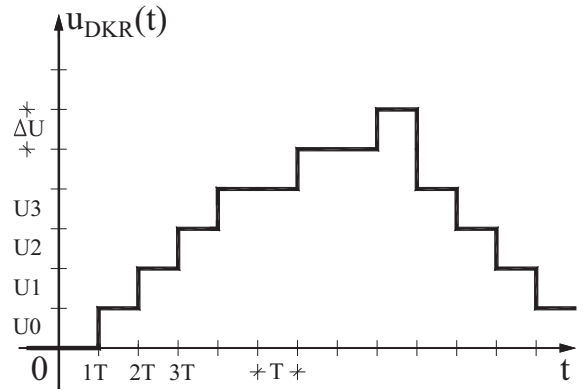
Tüm doğal olaylar ve süreçler zaman geçtikçe değişiyor. Bu yüzden onlar, analitik ya da grafiksel olarak, zamana bağlı fonksiyonlarla ifade ediliyor. İzlenen ve tespit edilen büyüklüğe bağlı olarak bu fonksiyonlar *sürekli* (kesintisiz) ya da *aralıklı* (kesintili) olabilir.

Makroskopik bakış noktasından, yakın tüm doğal olaylar süreklidir, çünkü onların zamanla değişiklikleri ani (hızlı) atlayışsızdır. Böyle fiziksel büyüklükler için çok sayıda örnekler vardır: sıcaklık, basınç, hız, doğal ışık, uzunluk vb. Bu tür olaylar, herhangi bir zaman aralığında sonsuz çok farklı değerler alan zamansal fonksiyonlarla tanımlanıyor. Şek.1-1'de sürekli bir zamansal fonksiyonun grafiği gösterilmiştir. Bu fonksiyonun amplitüdü yavaşça değiştiği açıkça görülüyor, çünkü herhangi bir sonlu (sınırlı) zaman aralığında, onun amplitüdü sonsuz sayıda değer alıyor.

Ancak, doğada aralıklı (kesintili) olaylar da vardır. Bu olaylar için, amplitütte hızlı değişikliklerin meydana gelme olayı karakteristiktir, çünkü bir ya da fazla anda, kesintili olayı ifade eden, zamana bağlı fonksiyon, bir değerden başka bir değere hemen geçiyor.



Şek. 1-1. Sürekli fonksiyonun zamansal diyagramı

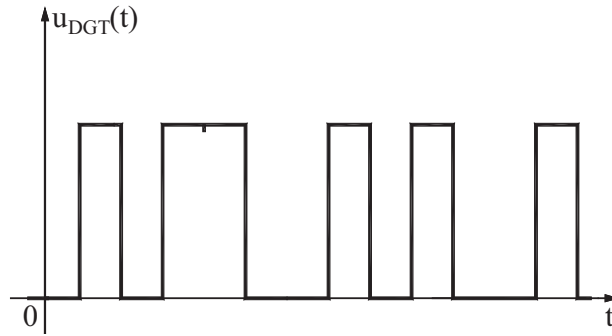


Şek. 1-2. Aralık fonksiyonun zamansal diyagramı

İnsan için özel ilgi alanı, *aralık* özelliği olan olaylardır. Bunlar sınırlı bir kümeden belirli kapsamdan değerler alabilen kesintili fonksiyonlardır. Onlarla örneğin, atomda elektronların enerjisi, konuşulan dillerde alfabenin harfleri, sayı sistemlerinden rakamlar ve benzer değerler ifade edilebilir. Herhangi bir sınırlı zaman aralığında, aralık fonksiyonun numaralandırılabilen sınırlı sayıda değerleri vardır. Her değer benzersiz belirlenmiş sayıyla numaralandırılıyor. Şek.1-2'de böyle bir fonksiyon için örnek verilmiştir. Görüldüğü gibi bu fonksiyonun amplitüdü aralık seviyeler alıyor.

İnsanın olayları tanıma ve kullanma isteği, bu olayları ölçen ve inceleyen farklı cihazların yapılmasına, sonuçları sunmasına ve elde edilen sonuçları iletmesine yol açmış. İnceleyeceğimiz cihazlar elektrikli. Onlar elektrik sinyalleri: akımlar ve gerilimlerle çalışıyor. *Giriş ve çıkış büyüklükleri, doğal sürekli olaylarla analogik (benzerlik) olarak değişen sinyalleri olan teknik cihazlara analog (doğrusal) cihazlar* denir. Ayrık büyüklüğün her seviyesinin belirli sayıyla ifade edilebileceğinden dolayı, *ayrık sinyallerle çalışan cihazlara dijital cihazlar* denir. *Dijital* terimi Latince digitus kelimesinden kaynaklanıyor ve parmak anlamına geliyor ya da daha uygun anlamı “parmaklarla saymak” olur. Bu aslında insan toplumunda sayıları ilk tanımlama şekliymiş, ancak bugün İngilizce *dijit* kelimesiyle bağlanıyor ve rakam ya da sayı anlamına geliyor. Analog ve dijital çalışma şeklini göstermek için örnek olarak zamanı saatla ölçülmesi alınabilir. Şöyle ki, saat ibreleri sürekli hareket ederse, bu zamanın analog gösterme şeklidir, saat böyle ise analog cihazdır. Ancak, saat zamanı her saniye ya da her dakika değişen sayılarla gösterirse, bu zamanın dijital şekilde gösterilmesidir, saat böyle ise dijital cihazdır. Buna benzer olarak, bir alet ya da enstrüman elektrik büyüklüğü ölçülen değeri ibre hareketiyle gösterilirse, o zaman bu alet analog alettir, ölçüler değer sayı şeklinde okunuyorsa o zaman dijital alet söz konusudur.

Dijital cihazda sayıların fiziksel tanımlanması, özel dalgalı şekilleri olan ayrı sinyal türü aracılığıyla yapılıyor. Bu sinyaller *dijital* sinyallerdir, böyle bir sinyal için örnek ise Şek.1-3'te verilmiştir. Şekilden görüldüğü gibi, dijital sinyalin ikili şekli ya da iki seviyesi var: yüksek ile alçak ve aslında gerilimli ya da daha seyrek olarak akımlı dürtüler ve duraklamalar dizisidir. Devamda her bir sayıyı sadece iki rakamın kombinasyonu, ya da 1-ler ve 0-larla tanımlayabileceğimizi (kodlayabileceğimizi) göstereceğiz. Bunun için iki farklı seviyeli sinyal gerekiyor.

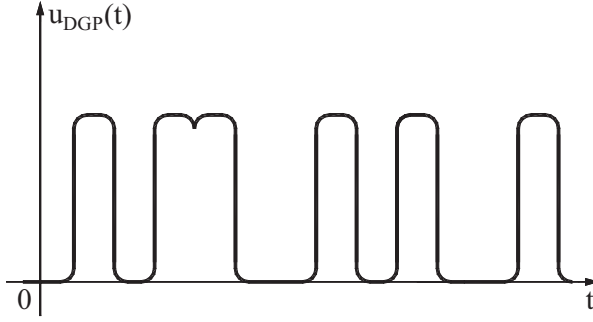


Şek. 1-3. Gerilimli dijital sinyalin zamansal diyagramı

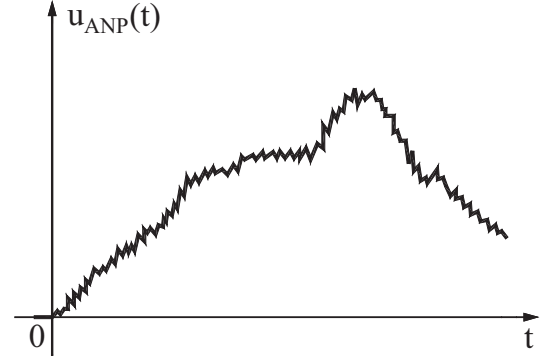
Dijital sinyallerin kullanma nedeni, bu sinyalleri üreten elektronik elemanların ve devrelerin yapılımları daha basit ve hesaplı olmasıdır. Dijital devreler iki olası durumdan birinde olabilir. Bu yüzden, her dijital cihazın temel yapı elemanları iki durumlu elektronik devreleridir. Elektrik sinyallerini dijital şekilde gerçekleştiren dijital devrelerin basit yapıları ve uygun fiyatları dışında, başka bir önemli etken onların gürültülere ve engellere karşı az duyarlı olmaları ya da onlara karşı dirençliğidir. Böylece, onların uzakta iletimi çok daha güvenilirdir. Şek.1-1'de gösterilen analog sinyalin ve Şek.1-3'te gösterilen dijital sinyalin, birbirinden bağımsız olarak, bir alıcıya gönderildiklerini tahmin edelim ve iletim yolunda istenmeyen parazit sinyallerin etkilediğini düşünelim.



Gerçekleşen iletimden sonra alınan sinyaller Şek.1-4'te ve Şek.1-5'te verilmiştir ve şekillerden görüldüğü gibi gürültünün etkisinden dolayı her iki sinyalde biçim bozukluğu meydana gelmiş. Orijinal sinyalin yenilenmesi, Şek.1-4'teki dijital sinyal için, Şek.1-5'teki analog sinyale kıyasen çok daha basittir. Buna göre, alınan analog sinyalde, amplitütte tüm istenmeyen değişikliklerin giderilmesi, alınan dijital sinyalinden dürtü ya da duraklama durumunun belirlenmesinden çok daha zordur.



Şek. 1-4. İletimden sonra elde edilen dijital sinyalin zamansal şekli



Şek. 1-5. İletimden sonra elde edilen analog sinyalin zamansal şekli

Farklı analog cihazlar için örnekler tüm analog ölçme aletleri, analog hesaplama makineleri ve benzer cihazlardır, dijital cihazlar için örnekler ise hesap makineleri, dijital aletler ve en karmaşık cihazlar olarak dijital hesaplama makineleri-bilgisayarlardır.

Analog çalışma şeklinin dijital çalışma şekline daha doğru (tam) olduğu görünmesine rağmen, sürekli sinyallerin isabetliği az durumda tamamiyle kullanılabilir, çünkü ölçmek için kullanılan aletler ve donanım, elde edilen sonuçları çok yüksek doğrulukla ölçmeleri, okumaları, işletmeleri, iletmeleri ya da başka bir şekilde yorumlamaları daha zordur. Diğer taraftan, dijital sinyaller sayısal biçimde ifade ediliyorlar ve bu yüzden elde edilen sonuçların niceliksel şekilde ifade edilmesi, işletilmesi, iletilmesi, korunması ve okunması daha kolaydır. Tek kelimeyle, dijital çalışma şekli insana çok daha yakındır, çünkü onlarla çalışması ve işlemesi daha kolaydır. Bu sonuca destek olarak, pratikte büyük sayıda analog büyüklüklerin sınırlı sayıda ayrık değerlerin toplamı olarak gösterilmesidir, yani ayrık büyüklüklere dönüşüyor ya da ayrıklaşıyor. Bunun için çok basit ve genel bir örnek, ağırlığın ölçmesi sırasında, ağırlığın kilogram, hektogram, dekagram ve gram gibi farklı bireysel ağırlıkların toplamı olarak ifade edilmesidir.

Analog ve dijital cihazların dışında, giderek fazla hem analog hem dijital çalışma şeklinin iyi taraflarını kullanan, yani *hibrid prensibine* göre çalışan cihazlar kullanılıyor. Şöyle ki, gerçek analog giriş sinyalleri, "yapay" ayrık sinyallere dönüşerek gereken işlemler bu sinyaller üzerine gerçekleşiyor. İşlenen dijital sinyaller ardından yeniden analog şekile dönüşüyor. Böyle dönüşümler **analog-dijital dönüştürücüleri** (ADD) ve **dijital-analog dönüştürücüleri** (DAD) olarak adlandırılan kurgular aracılığıyla gerçekleşiyor. Bu kurguların sayesinde, dijital cihazların tüm avantajlarıyla, sinyallerin esasında analog olduğu durumlarda bile çok geniş kullanımı sağlanmıştır.

Ancak, kesintisiz büyüklüklerin ayrık büyüklükler olarak ifade edilmesi sırasında, bilerek daha küçük ya da daha büyük hata yapıyor. Bu hata *ayrıklaşma hatası* ya da *niceleme hatası* olarak adlandırılıyor. Diyelim ki, yukarıda verdiğimiz örnekte gram bölümü ya da daha az değeri için hiçbir ağırlık doğru olarak ölçelmiyecek. Ölçme sırasında daha büyük doğruluğun elde edilmesi için, daha küçük ölçü birimleri kullanılmalıdır, ya da *kuvantumlar* veya sürekli büyüklük için *ayrıklaşma seviyeleri* kullanılmalıdır. Buna göre, **hibrid (karışık) elektronik cihazlarda**, analog büyüklüğün niceleme ya da seviyelere göre ayrıklaşma sorunu büyük önem taşıyor çünkü bununla hata da, yani çalıřma isabetliđi de belirleniyor.

**Dijital elektronik**, *dijital devreleri ve A/D ile D/A dönüşüm devreleri, onların analizi, sentezi, projelenmesi ve gelişimi açısından inceliyor.*

## 1.2. BİLGİ VE BİLGİNİN KODLANMASI

Karşılıklı iletişimde ve farklı cihazların kullanımıyla insanlar etraflarını saran dünya hakkında yeni bilgiler ediniyor, bildiriler ve yenilikler alıyorlar, haberler iletiyorlar ve haber değişimi gerçekleştiriyorlar. **Bilgi** kelimesi hergünlük yaşamda bildirme gibi aynı anlamı var ve bu yüzden bu terim en sıkça olarak iletim sistemlerinde, telekomünikasyon sistemlerinde kullanılmıřmış. Ancak, bilimin ve tekniđin hızlı gelişimi bilgilerin sadece hızlı ve doğru iletimi ve bilginin gönderme gereksinimi değil, onların işlenmesi ve korunması (belleđe yerleşmesi) gereksinimi de ortaya çıkmıř.

İşin kolaylaşması için, insan bilginin hem iletimini hem işletimini otomatik olarak, makineyle yapılmasını istiyormuş. Bilginin iletimi, korunması ve işletimi için basit şekilde ifade edilmesi için farklı yöntemler aranılmıřmış. Bu arada bilgi taşıyıcısı, bir parametrenin değişimini, genelde amplitüt değişimini içeren elektrik sinyalidir. Böylece “soyut bir şekile” sembollerle kaydedilen bilgilerin, “fiziksel” olarak elektrik sinyallerle sunulması sağlanmıřtır. Bununla ilgili yapılan büyük sayıda arařtırmalar bilgilerin dijital elektrik sinyallerle-sadece iki farklı seviyeli sinyallerle tanımlanması hesaplı, güvenilir, en uygun ve en kaliteli çalışma şekli olduğunu göstermiş. Böylece, bilgi ikili şekilde, sadece iki sembolden oluşan dizilerle ifade ediliyor. Buna göre, bilgilerin iletimi, işletmesi ve korunması sırasında, onların tanımlanması insana yakın olan şekilden daha farklı bir şekildedir, yani alfabetik ve sayısal semboller (harfler ve rakamlar) kullanılmıyordur, onların yerine ikili semboller kullanılmıyordur.

Bilgilerin sınırlı bir kümenin elemanları olan semboller yardımıyla tanımlanmasına **bilgilerin kodlanması** denir. *Bilgilerin kodlanması için kullanılabilen tüm semboller kümesine kod alfabeti denir, kod alfabetinden her bireysel sembole ise kod sembolü denir. Bir bilgiyi ya da bilgi kısmını tanımlayan her kod semboller grubuna kod sözcüğü denir.* İkili kodlanmış bilgiler, ikili semboller grubu, yani sadece iki farklı değer alabilen ve B ikili kümesine ait olan semboller grubu tanımlayan kod sözcüklerinden oluşacak. Genelde bu iki sembol için 1 ve 0 işaretlerin kullanılması kabul edilmiştir, öyle ki  $B = \{1,0\}$  geçerlidir. Böylece, ikili kodların ve ikili sayı sisteminin dijital elektronikte ve bilgilerin dijital işletiminde neden o kadar büyük önemi olduğu şimdi açıkça anlaşılıyor.

### 1.3. BİLGİ MİKTARI ÖLÇÜ BİRİMLERİ

Bilgi iletimi söz konusu olunca, bildiriye alan insan tarafından öznel olarak, bilginin içeriği en önemlidir, ya da aldığı bildiride ne kadar sürpriz olacağı önemliyken, bilgi biçimi ve iletim şekli neredeyse hiç önemli değildir. Buna göre, gerçekleşmesi için büyük bir ihtimali olan olayın, gerçekten gerçekleştiği bildirisi, az miktarda bilgi içeriyor. Ters olarak, gerçekleşmesi için çok küçük olanağı olan beklenmeyen olayın meydana geldiği bildirisi, çok daha fazla bilgi miktarı içerir. Buradan bilgi miktarı ve olayların gerçekleşme olanağı arasındaki ilişki ortaya çıkıyor.

Ancak, dijital teknikte bilgi miktarının ölçü birimleri söz konusu olunca, gerçekleşme olanağını doğrudan etkisi yoktur. Bilgi miktarının ölçülmesi için birim olarak bir bit tanımlanıyor. Bir bit  $B = \{1,0\}$  ikili kümeden bir sembolü belirliyor. Bit terim, İngilizce *Binary digit* (**bit**) kavramının kısaltmasından geliyor, **ikili rakam** anlamına geliyor ve **b** ile işaretleniyor. Bitin değeri 1 ya da 0 olabilir, ya da dijital sinyalde bitin temsil ettiği sinyalin yüksek ya da alçak gerilim seviyenin meydana gelmesidir. Bu yaklaşım, dijital elektronikte doğru iletimin, öngörülen işletimin ve ikili sembollerin doğru korunmasının en önemli olduğundan, olayların gerçekleşme ihtimalinin ise önemli olmadığından geliyor. Bilgilerin işletildiği sistemlerde farklı şekilde işletme olanakları ve işletme hızına ile bilgilerin bellekleme kapasitenin büyüklüğüne büyük önem veriliyor. Buna göre, bir mesajın daha büyük sayıda kod sözcüklerden oluştuğu durumda ve her sözcük fazla bittens oluştuğu zaman, mesajın içerdiği bilgi miktarının da daha büyük olması açıkça anlaşılıyor.

Bununla ilgili, bittens daha büyük birim olarak 1 bayt (byte) birimi tanımlanıyor. Bir **bayt** sekiz bit içeriyor ve 1-ler ve/veya 0-ların rastgele sıralamasını belirliyor. Genelde [B] ile işartleniyor. Aşağıdaki şekilde (Şek.1-6) iki bayt örneği gösterilmiştir

$b_7$	$b_6$	$b_5$	$b_4$	$b_3$	$b_2$	$b_1$	$b_0$		$b_7$	$b_6$	$b_5$	$b_4$	$b_3$	$b_2$	$b_1$	$b_0$
1	0	1	0	1	1	1	0		0	1	0	0	0	0	1	1

Şek. 1-6. Bir bayt uzunluğunda iki bellek sözcük örnekleri

Geçmişte bittens daha büyük birim olarak 1 **nibıl** (İng. nibble) kullanılıyormuş, hem de dört bittens oluşan gruplar (diziler) için, ancak bu birim bugün çok az kullanılıyor ya da tetrat terimiyle değiştiriliyor.

Baytın da küçük birim olduğundan dolayı, pratikte bayttan daha büyük birimler tanımlanmıştır, o birimler de şunlardır: 1 **kilobayt**  $2^{10}=1024$  bayttan oluşan birimdir ve [KB] ile işaretleniyor,  $2^{10}$  [KB]= $2^{20}$  [B] içeren 1 **megabayt** [MB],  $2^{10}$  [MB]= $2^{20}$  [KB]= $2^{30}$  [B]'tan oluşan 1 **gigabayt** [GB] ve  $2^{10}$  [GB]= $2^{20}$  [MB]= $2^{30}$  [KB]= $2^{40}$  [B]'tan oluşan 1 **terabayt** [TB]. Açıkladığımızdan görünen şu ki, verilerin dijital işletim dilinde, „kilo” önekin biraz geleneksel olmayan kullanımı vardır, çünkü  $2^{10} = 1024$  birim içeriyor ve bu yüzden [K] ile işaretlenerek, 1000 birimle ilgili olan [k] öneki ile geleneksel işaretlenmeden farklıdır. Buna göre 1 [MB] =  $1024 \times 1024$  [KB], vs. Çarpan şimdiye kadar alıştığımız gibi 1000 değil, 1024 olacak.

Örnek olarak şu basit sorunu çözeceğiz. Herhangi bir metinsel sembolün (harf, rakam ya da noktalama işareti) 1 bayt [B] ile kodlanabileceğini alalım ve bir sayfanın 3.000 sembol içerdiğini tahmin ederek, 200 sayfalık kitap için ne kadar bellek alanın ayırmamız gerektiğini hesaplamalıyız.

Çözüm: 200 sayfa x 3.000 sembol = 200 sayfa x 3.000 [B] = 600.000 [B]. 1[KB] = 2<sup>10</sup> [B] = 1024 [B] ≈ 1000 [B] olduğuna göre, kitabı bellekte korumamız için 600 [MB] kapasiteli bellek bileşenimiz gerekecek.

Burada, bir kod sözcüğünde bit sayısı n'e bağlı olarak, 0-lardan ve 1-lerden meydana gelebilen farklı kombinasyonların toplam sayısını (N), aşağıdaki denklemle hesaplayabileceğimizi not etmemiz çok önemlidir:

$$N = 2^n \quad (1-1)$$

Böylece, örneğin elimizde 1 [B] baytımız olursa, o zaman onunla 2<sub>8</sub> = 256 farklı kombinasyon ifade edilebilir. Buna göre 1 bayt ile 256 farklı bilgi tanımlayabiliriz. Örneğin, 0'dan 255'e kadar tam sayıları ya da İngilizce alfabesinden tüm küçük ve büyük harfler, Makedon alfabesinin küçük ve büyük harfleri vs, ya da toplam 256 farklı sembol.

## 1.4. BİLGİ ÇEŞİTLERİ

Bilgisayarın çalışması sürecinde, bilgisayarın doğru çalışmasının sürekliliğini sürdürmek için önemli olan farklı ve özel bilgiler taşıyan verilerin değişimi ve akımı gerçekleşiyor. Burada farklı **komut** (*emir,yönerge*), **veri** ve **adres** türleri aittir.

**Yönergeler** bilgisayarın neyin yapmasını gereken bilgileri içeriyorlar. Bir mantık birimi oluşturan komutlar dizisi, bilgisayarın çalıştığı programı tanımlıyor.

**Veriler** dış dünyadan belirli olay veya bilgisayar içinde gerçekleşen bazı işlemin sonucu olarak elde edilen değer hakkında bilgi içeriyorlar.Veriler olarak farklı zamansal değişken ayrık veya sürekli büyüklüklerin değerleri sayılabilir. Veriler bilgisayara ekleniyor, programın yönergelerine bağlı olarak işletiliyor ve bu arada yeni veriler üretiliyor.

Veriler ve komutlar dışında, bilgisayarda ayrı bir bilgi türü daha arasında değişim gerçekleşiyor, onlar da adreslerdir. **Adresler**, verilen bir yönergenin uygulandığı verilerin kesin yeri (pozisyonu) nerede olduğunu gösteren bilgiler içeriyorlar.

Bilgisayarın bilgileri işletmesi için, tüm bilgiler ikili şekilde giriliyor. Soyut düzeyde bu bilgiler **0**-lar ve **1**-ler'den oluşan dizilerdir, ancak fiziksel olarak bu bilgiler sıkça olarak gerilim dürtüleri ve duraklamalar grupları ya da diziler şeklinde elektrik sinyallerdir.

Bilgisayarda gerçekleşen belirli programların, giriş verileri olarak sayısal veriler dışında başka programların ve adreslerin de kabul edilebildiğinden dolayı, hem programların, hem verilerin hem adreslerin, **sözün daha geniş anlamında, veriler oldukları** alınabilir.

Sonuç olarak hem yönergeler hem veriler hem adresler, aslında ikili şekilde kodlanmış ve uygun elektrik sinyallerle tanımlanan bilgiler olduklarını diyebiliriz. Bilgisayarın iç organizasyonu ve çalışan programın sayesinde, bilgisayar bu bilgileri birbirinden ayırıyor.

## 1.5. DİJİTAL DEVRELERİN VE AĞLARIN AYIRIMI

Dijital devrelerin sadece iki farklı seviyesi olabilen elektrik sinyallerle çalıştığını belirttik. Böyle sinyaller ikili şekilde kodlanmış belirli veriler tanımlıyor, bu verilerin herbiri kendisinde dış dünya hakkında belirli bilgi içeriyor. Dijital tekniğinde temel, ana yapı elemanları *mantıksal devrelerdir (kapılar, geçitler)*.

Onlar yarı iletken anahtarlamalı elemanların kullanımıyla yapılıyor, her başka dijital bileşen, ağ ya da kurgu ise belirli sayıda mantıksal devrelerin uygun bağlanmasıyla gerçekleşebilir. Gerçekleşen işe bağlı olarak, dijital cihazların yapılı olduğu tüm temel bileşenler iki gruba ayrılabilir: *birleşimsel mantık bileşenleri (mantıksal bileşenler)* ve *bellekssel (ardışık) bileşenler*. Mantıksal kurgular 1 veya 0 mantıksal durumları, yani onların kombinasyonlarını gerçekleştiriyor, ancak onların önceki durumlarını hafıza etmek olanağı yoktur. *Bu özelliklere sahip olan tüm elektrik ağlara birleşimsel (kombinsyonel) mantıksal-anahtarlamalı ağlar denir, ya da kısaca sadece birleşimsel ağlar denir.* Çıkıştan girişe doğru geri bağlantının olmadığından dolayı, onlarda çıkış sinyalleri sadece giriş sinyalleri olduğu zaman vardır. Giriş sinyalleri kaybolursa, o zaman çıkış sinyalleri de kayboluyor. Fakat, verilerin dijital işletimi sırasında, daha sonra yeniden kullanılacak belirli verilerin bellekte korunması gereği meydana gelebilir. Bu amaçla, önceki mantıksal durumların hafıza edilmesi için uygulanan bellek elemanları kullanılıyor. **Flip-flop** olarak adlandırılan iki kararlı (iki durumlu) multi vibratör (titreşken), bir bitli verinin korunması için kullanılan iki kararlı durumlu elektronik devredir. Flip-flopun sadece bir bitlik en küçük bilgi miktarını belleklediğinden dolayı, flip-flop dijital elektroniğinde temel bellek elemanı ya da temel bellek hücresidir. Flip-floplar, pozitif geri bağlantının uygulandığı temel mantıksal devrelerin kullanımıyla gerçekleşiyor.

*Böyle bellekleme özelliğe sahip olan elektrikli ağlara ardışık mantıksal-anahtarlamalı ağlar ya da kısaca sadece ardışık ağlar veya ardışık otomatlar denir.* Bu elemanlarda ve ağlarda, giriş uyarı sinyallerin etkisi bittikten sonra bile çıkış sinyalleri vardır. Bu yüzden, sıradaki çıkış durumları mevcut giriş sinyallere bağlıdır, ancak aynı zamanda giriş sinyallerin sıralamasında, yani devrenin geçtiği önceki durumların sırasına bağlıdır.

Sezgisel bir şekilde, birleşimsel ağların, ardışık ağlarda belleklenmiş ve oradan gelen verilerle farklı işlemlerin yaptıkları anlaşılır. Birleşimsel devrelerin yapılması için mantıksal devreler kullanılıyor, ardışık ağlarda ise temel yapı elemanları flip-floplardır. Ardışık ağlarda flip-floplar dışında farklı mantıksal devre çeşitleri de kullanılıyor. Devamda, karmaşık dijital kurguların yapımında en çok kullanılan birleşimsel ve ardışık bileşenleri inceleyeceğiz.

**Aritmetik-mantık fonksiyonları gerçekleştirme devreleri:** Tüm temel aritmetik işlemler, hata diferansiyel ve integral işlevler bile farklı toplama süreçleriyle gerçekleşebilir. Buradan *ikili toplayıcının* dijital aritmetik cihazlarında çok büyük önemi olduğu sonucuna varabiliriz. İkili toplayıcı birleşimsel ağlar grubuna aittir. Bu grupta, ikili toplayıcı dışında, *çıkarma devresi, tamamlama devreleri, karşılaştırma devreleri* vb. devreler yer alıyor.

Burada özellikle *aritmetik-mantık birimleri* önemli yer alıyorlar. Bu birimler, farklı aritmetik ve mantık işlevler gerçekleştiren tümleşik bileşenlerdir.



**Anahtarlama matrisler:** *Anahtarlama matrisleri* anahtarlama elemanları satırlarda ve sütünlarda yerleşerek, matris yapıları oluşturan bileşik birleşimsel ağlardır. Bu matrislerin her biri farklı şekilde çalışıyor ve onları işlevsel isimlerine göre ayırt ediyoruz:

1. **Kodlayıcı:** Bu mantıksal ağla kodlama süreci gerçekleşiyor. Onun girişlerinde bazı sayı sisteminden rakamlar ya da bazı alfabeden harfler ve özel işaretler (semboller, karakterler) tanımlayan sinyaller giriyor, çıkışta ise ikili sayı sisteminden veya ikili koddan bazı kodlama sözcükleri elde ediliyor.
2. **Kod çözücü:** Verilerin dijital işletimi sırasında, ikili kodlanmış verilerin, örneğin onlu sayı sistemine veya tabanı ikiden farklı başka bir sayı sistemine gibi başka bir şekile dönüşüm gereği ortaya çıkabilir. Bu süreç kodlama sürecinin tersidir ve kod çözücü birleşimsel ağların kullanımıyla gerçekleşiyor.
3. **Çoğullayıcı (seçici):** Çoğullayıcı ağın girişinde var olan büyük sayıda verilerden bir veriyi seçen ve bu veriyi tek çıkışa ileten birleşimsel ağdır. Hangi verinin çıkışa gönderileceği, özel seçici (adres) girişleri yardımıyla belirleniyor. Ayrıca, elde olan birçok giriş gruplarında, bir giriş veri grubunu seçen ve seçtiği giriş grubunu tek çıkış hatlarına aktaran çoğullayıcılar da vardır.
4. **Çoğullama çözücü:** Bu anahtarlama ağı çoğullayıcıdan ters şekilde çalışıyor. Şöyle ki, çoğullama çözücüsü tek girişten gelen veriyi alıp, onu birçok çıkıştan birine iletiyor. Çoğullayıcıda olduğu gibi, bu durumda da uygun çıkışı adreslemek için seçici hatlar vardır. Ayrıca, tek giriş veri grubunu, birçok çıkış hatlarından birine ileten çoğullama çözücülerine de rastlanabilir.

**Programlanabilir mantıksal yapılar:** Bu birleşimsel ağların bileşik matris yapıları vardır, onların en önemli özelliği ise *anahtarlama elemanları arasındaki bağlantıları programlama* olanağı vermesidir. Anahtarlama elemanları kullanıcı tarafından verilen şekilde bağlanıyor. Bu bileşenler dijital sistemlerinde giderek fazla kullanım bulan tümleşik bileşenlerdir.

**Yazmaçlar:** Yazmaçlar dijital cihazlarda, özellikle aritmetik-mantık birimiyle beraber olarak, en sıkça kullanılan elemanlardır. Yazmaçlar belirli sayıda flip-floptan oluşuyorlar. Bilgiden her bir ayrı flip-flopta hafıza ediliyor ve bu yüzden yazmaçlar ardışık ağlar grubuna aittir. Farklı yazmaç çeşitleri vardır, ancak en çok bilinen yazmaçlar sabit ve ötelemeli yazmaçlardır. Sabit yazmaçlarda, onların içeriği olarak, sınırlı uzunlukta ikili veri geçici olarak korunabilir. Örneğin, 1 nibil ya da 1 bayt. Ötelemeli yazmaçlarda, yazmaç içeriğinin, yani içinde bulunan verinin birer birer bit sağa ya da sola kayması gerçekleşebilir.

**Sayaçlar:** Sayaçlar da ardışık ağlar grubunda yer alıyor, çünkü temel yapı elemanı olarak bellek elemanı, flip-flop kullanıyorlar. Sayaç girişinde meydana gelen her dürtü ile, sayaç durumunun ardışık değişmesi meydana geliyor. Sayaçın her durumu birkaç bit dizisi, yani ikili kodlanmış onlu sayı tanımlıyor. Buna göre bu kurgu giriş dürtüleri sayıyor. Belirli dürtü sayısından sonra, sayaç başlangıç durumuna dönüyor ve sayma döngüsel olarak tekrarlanıyor.

**Bellekler:** Bellekler ikili verileri deęişmemiş şekilde koruyor, öyle ki belirli bir süre sonra bu veriler yeniden kullanılabilir. Bellek düzenlenmiş yazmaçlar kümesi olarak tanımlanabilir. Bu arada her yazmaç bir verinin korunabileceęi *bellek yeri* tanımlıyor. Bellek yerine yerleşen veri bu bellek yerin *içeriğidir* ve bu arada bir bellek yerinde bir *bellek sözcüğün* yerleşebileceęi söyleniyor. Bellek sözcüklerin, genelde baytlarla ifade edilen sabit *uzunlukları* var. Her bellek yerinin bellekte kendi pozisyonu var ve bu pozisyon uygun numarayla: *bellek yerin adresiyle* belirleniyor. Adres bellek yerin bellekteki pozisyonu tanımlamak için kullanılıyor. Adresin verilmesiyle (belirlenmesiyle) her bir bellek yerine ulaşılabilir ve onun içeriğininle farklı işlemler yapılabilir, yani belleklenmiş veri: bellekte yerleşmiş veri okunabilir veya o yerde yeni veri yazılabilir. Tüm bellek yerlerin toplam sayısı belleğin *kapasitesini* belirliyor. Bellek kapasitesi kilobaytlarla veya megabaytlarla ifade ediliyor.

Belleklerin kalitesini deęerleyen ölçütler şunlardır: en çok bellek organizasyonuna baęlı olan çalışma hızı, bellek yerlerine orta erişim zamanı, yitim gücü ve fiyatı.

**A/D ve D/A dönüşüm için dijital devreler:** Dijital cihazlarda işletilmesi gereken büyük sayıda sürekli türünden bilgiler vardır, örneğin: sıcaklık, basınç, hız, zaman vb. Gerçek dünyadan sürekli olayların dijital sinyallere dönüşümü, dijital tekniğinin gelişimiyle giderek daha önemli oluyor. Sürekli olaylar, onlara analog olan elektrik sinyallere *sensörler* yardımıyla dönüşüyor. Elde edilen analog sinyalleri, dijital sinyallere dönüşümü, *analog-dijital (A/D) dönüştürücüler* olarak adlandırılan elektronik devrelerde yapılıyor ve dönüştürülen sinyaller bu şekilde dijital cihazlarda işletiliyor. Ters *süreç dijital-analog (D/A) dönüştürücülerle* gerçekleşiyor. D/A dönüştürücünün girişinde dijital sinyal geliyor, çıkışta ise ona uygun analog sinyal elde ediliyor. Bu şekilde elde edilen analog sinyal *deęiştirici* aralığıyla sürekli özellięi olan sinyale dönüşüyor.

Dijital elektronik devrelerin incelenmesi sırasında iki temel sorun ortaya çıkıyor: bir sorun dijital devrelerin *analizini* yapmaktır, dięer sorun ise *sentez* sorunudur, yani önceden verilmiş özelliklerle dijital devrelerin *gelişimi, projelenmesi veya tasarlanması*.

*Analiz* yapısal ve işlevsel açıdan yapılıyor. Analizin amacı, verilen dijital kurgunun yapısını ve çalışma şeklini belirlemektir. Bu arada cihazın mantık, elektrik ya da montaj şeması ve giriş deęişkenleri (uyarma, tahrik) biliniyor, kurgudan gelen yankı ya da çıkışlar belirlenmelidir. Analiz özellikle dijital devrenin kullanımı, tamiri ve bakımı sırasına gereklidir.

*Sentez* sırasında, elde edilmesi gereken kurgunun özellikleri biliniyor, örneğin yapması gereken işlevler ve onların giriş sinyallerinden baęlılığı. İlk olarak kurgunun blok-diyagramı, her bloğun rolü ve görevleri, her bloğun girişleri ve çıkışları ve verilerin ve sonuçların korunması gereęi varsa bellek kapasitesi tanımlanıyor. Ardından cihazın teknik açıdan yapılabilmesi için birleşimsel ve bellek elemanların seçimi yapılıyor. Bu sorun aslında dijital cihazların yapılmasına, daha doğru projelenmesine ve gelişimine kalıyor.

## II) SAYI SİSTEMLERİ VE KODLAR

### 1.6. TEMEL TERİMLER

Dijital teknikte bilgiler sayılar gibi sayısal olarak tanımlanıyor ve işletiliyor. Bu yüzden normal olarak, sayıların dijital kurgular için “anlayabilir” olması için, bu sayıları hangi şekilde en uygun tanıtabileceğimiz sorusu ortaya çıkıyor. *Bu bölümde bilgilerin dijital sistemlerinde gerçek şekilde tanımlanmasına en yakın tanıtım şekillerini inceleyeceğiz.* Bu sorun çok önemlidir ve içinde kodlama sürecinin, özellikle ikili kodların incelemesini ve ikili sayı sisteminin en önemli olduğu farklı sayı sistemlerin analizini içeriyor.

Bilgilerin ikili şekilde belirlenmesi insanın anlayış şeklinin bir kısmı olarak zor düşünebilir çünkü ikili yazılmış veriler aslında *sadece iki sembolden oluşan dizilerdir.* Bu yüzden *ikili notasyon (ikili işaretleme)*, tüm temel devreleri anahtarlamalı olduğu ve çıkışları bildiğimiz gibi sadece iki olası durudan birinde olduğu dijital cihazlar için “doğaldır”. **Kodlama terimi, bilgileri bir kümenin elemanları olan semboller yardımıyla tanımlanmasıdır.** Örneğin, bir alfabenin harfleri ya da bazı sayı sisteminin rakamları. Devamda, dijital cihazların, özellikle dijital hesaplama makinelerin – bilgisayarların çalışmasının analizi sırasında uygun kullanım gören kodlar incelenecek.

Ulusal diller ile bir benzetme yapalım. İnsan toplumunda her ulusal dil, insanlar arasında iletişim kurulması için bir elemandır. Konuşmakla, biz aslında, bilgileri sözcüklere “kodluyoruz” ve bu sözcükler, uygun halkın her üyesi için anlaşılır oluyor. Yazılmış metin, aslında “kodlanmış” terimlerin bir hafıza etmek (bellekleme) türü tanımlıyor. Her terime bir kelime uyar, her kelime seslere ayrılabilir, her ses için ise uygun simge, yani ulusal alfabeden harf vardır. Sözcüklerin yazılması (kodlanması) ya da okunması (kod çözümlenmesi) için verilen ulusal dilin sözdizimi bilinmelidir. Sözdizimi hem okuma hem yazma sırasında kesinlikle uyulması gereken kurallar ve düzenlemeler grubudur.

**Sayı (nümerik) sistemler, isimleri rakam olan semboller sistemleridir ve onlarla sayılar işaretleniyor.** *Ağırlıklı (pozisyonel) sayı sistemleri ve ağırlıksız sayı sistemleri vardır. Biz sadece ağırlıklı sayı sistemlerini inceleyeceğiz. Her ağırlıklı sayı sisteminin b ile işaretlenen kendi tabanı (temeli, kökü) vardır. Taban, aslında sistemde farklı rakamların toplam sayısıdır.* Genel olarak, sayı sisteminin tabanı olarak birden daha büyük ya da eşit olan herhangi sayı alınabilir. *Onlu sayı sistemini önceden biliyoruz ve bu sistemi her günlük yaşamda kullanıyoruz: bu sistemin 10 farklı rakamı var ve tabanı  $b=10$ 'dur. Ancak bu tematik biriminde kısaca, 16 farklı rakamı ve tabanı  $b=16$  olan on altılı sayı sistemini, 8 rakamı ve tabanı  $b=8$  olan sekizli sayı sistemini ve sonunda en önemli sayı sistemi olarak, sadece iki rakamı ve tabanı  $b=2$  olan ikili sayı sistemini inceleyeceğiz.*



## 1.7. SAYI SİSTEMLERİ

Farklı sayı sistemlerin tanıtımıyla sayıların yazılması ve işaretlenmesi belirlenmelidir. Bu amaçla, ilk önce her sayı sistemi için rakamların şeklini belirlememiz gerekecek. Onlu sayı sisteminin on farklı rakamı var **0,1,2,3,4,5,6,7,8,9**. Bu rakamlar on rakamdan daha az olan diğer sayı sistemlerin rakamları için de semboller olarak alınıyor. Böylce, örneğin, sekizli sayı sisteminin sekiz rakamı vardır: **0,1,...,6,7**, ikili sayı sisteminin ise sadece iki rakamı var: **0** ve **1**. Sayı sisteminin tabanı ondan daha büyük ise, o zaman İngilizce alfabetesinin ilk harfleri de edkleniyor: A, B, C. Böylce on altılı sayı sisteminin 16 rakamı var: **0,1,...,8, 9, A, B, C, D, E, F**.

Sayıları o şekilde yazacağız ki önce rakamları kullanarak sayıyı yazıyoruz, ardından parantezde veya indis olarak sayıyı yazıldığı sayı sisteminin tabanını yazıyoruz. Onlu sayı sistemi için 10, DEC ya da D yazıyoruz, on altılı sayılar için 16, HEX ya da sadece X ekleniyor, sekizli sayılar için 8, OCT ya da sadece Q, ikili sayılar için ise 2, BIN veya sadece B ekleniyor.

Sıkça sorun olarak şu soru ortaya çıkıyor: b tabanı olan bir sayı sisteminde n rakam verilmişse kaç farklı sayı N yazılabilir. N sayısı, b sayı sistemin tabanını n üsü ile üsleyerek hesaplanıyor:

$$N = b^n \quad (1-2)$$

Buradan eşit rakam sayısı ise, farklı sayı sistemlerinde, daha büyük tabanı olan sayı sistemde daha fazla sayıların yazılabileceği açıkça görülüyor. Örneğin, onlu sayı sisteminde üç rakamla toplam  $10^3 = 1000$  farklı sayı yazılabilir, on altılı sayı sisteminde toplam yazılabilen farklı sayıların sayısı daha büyüktür ve  $16^3 = 4096$  farklı sayı yazılabilir, sekizli sistemde toplam  $8^3 = 512$  sayı yazılabilir, ikili sayı sisteminde ise sadece  $2^3 = 8$  farklı sayı yazılabilir.

Ağırlıklı sayı sistemlerinde sayıdaki her rakamın belirli *ağırlığı* (*ağırlık değeri*) vardır. Ağırlık sayıdaki rakamın pozisyonel (ayrimsal) noktasına (şimdiye kadar ondalık nokta olarak bildiğimiz) göre pozisyonuna (yerine) bağlıdır. **Ağırlık** *ashında sayı sisteminin b tabanının tam üsüdür* (*potansıdır*). Noktanın solundan birinci rakamın ağırlığı (sıfıncı pozisyon) (sıfıncı sayı pozisyonundanrakamın ağırlık değeri)  $b^0$  değerindedir. İkinci rakamın (birinci pozisyon) ağırlığı  $b^1$ -dir vs. En büyük ağırlığı (tabanın en yüksek üsü, en yüksek seviye) noktanın solunda bulunan rakamın vardır ve bu rakam MSD ile işaretlenir (*İng. Most Significant Digit*), yani bu rakam *en değerli* (*en büyük değeri olan*) rakamdır. Noktanın sağ tarafından birinci rakamın (eksi birinci pozisyon)  $b^{-1}$  ağırlığı vardır, ikinci rakamın  $b^{-2}$  vs. Buna göre, en düşük ağırlığı (tabanın en küçük üsü) sayının en sağında bulunan rakamın olacak. Bu rakam LSD ile (*İng. Last Significant Digit*) ile işaretlenerek, *en değersiz* (*en az değeri olan*) rakamdır.

Tabanı b ve n-tamsayı ve m-kesit-rasyonel rakamı olan herhangi bir sayı sisteminde X sayısının değerini belirlemek için, ağırlıklı formül olarak bilinen aşağıdaki formül kullanılabilir.

$$X = X_{(10)} = \sum_{i=-m}^{i=n-1} c_i t_i = \sum_{i=-m}^{i=n-1} c_i b^i = c_{n-1} b^{n-1} + c_{n-2} b^{n-2} + \dots + c_1 b^1 + c_0 b^0 + c_{-1} b^{-1} + \dots + c_{-m} b^{-m} \quad (1-3)$$

(1-3) formülünde,  $c_i$  ile pozisyonel noktasından sayarak i-nci yerde bulunan rakam işaretlenmiştir. Bu arada 0-ncı yer birinci soldandır, ardından 1-nci, 2-nci vs. rakam geliyor. Noktanın

sağ tarafından birinci yer (-1)-nci rakamdır, ardından (-2)-nci, (-3)-ncü rakam geliyor vs.  $t_i$  ile verilen rakamın pozisyon değeri (ağırlığı) tanımlanmıştır. Doğal ağırlık sistemleri için şu denklem her zaman geçerlidir:

$$t_i = b^i \quad (1-4)$$

$b$  sayı sisteminin tabanını belirleyen sabittir. (1-3) formülüyle alında herhangi sayı sistemin den onlu sayı sistemine dönüşüm yapılıyor.

Farklı sayı sistemlerinde tam sayıların analizi sırasında, onlu tam sayılarda olduğu gibi, ayırma noktası veya virgöl yazılmıyor. Noktanın hemen en sağdaki rakamın yanında, yani en az ağırlıklı rakamın yanında olduğu biliniyor.

Her sayı sistemi için, sayıların *gerçek (doğrudan, varsayılan)* değeri dışında, onların *tümleyeni* de tanımlanıyor. Bir sayının ( $X$ ) tümleyeni  $\bar{X}$  ile işaretleniyor ve aşağıda verilmiş tanıma göre belirleniyor:

$$\bar{X} = K - X \quad (1-5)$$

$K$ , değeri  $b^n$  ya da  $b^n - 1$  olan sabittir. Bu arada  $b$  sayı sisteminin tabanıdır,  $n$  ise verilen sayıda rakamların sayısıdır.

$K = b^n - 1$  olunca, uygulanan sayı sisteminde *en büyük sayıya kadar tümleyen* elde ediliyor. Böylece, örneğin dört rakamlı onlu sayılar için ( $b=10, n=4$ )  $K$  sabitinin değeri  $K = 10^4 - 1 = 9999$  olacak.  $X=1234$  sayısı verilmişse, onun dokuzaya kadar tümleyeni  $\bar{X}^9 = 9999 - 1234 = 8765$  olacak. İkili sayı sisteminde,  $\bar{X}^1, X^1$  ya da sadece  $\bar{X}$  ile belirtilen bire kadar tümleyeni, yani tek (birinci) tümleyeni söz konusu oluyor.

$K$  sabitinin  $K = b^n$  değeri olunca, o zaman sayı sisteminde *sayıların kapsamına kadar tümleyeni* elde ediliyor. Böylece dört rakamlı onlu sayılar için ( $b=10, n=4$ ),  $K$  sabitinin değeri  $K = 10^4 = 10000$  olacak.  $X=1234$  sayısının 10-a kadar tümleyeni  $\bar{X}^{10} = 10000 - 1234 = 10000 - 1234 = 8766$  olacak. İkili sayı sisteminde, tümleyeni ikiye kadardır, yani ikinci (çift) tümleyenidir ve  $\bar{X}^2, X^2$  ile işaretleniyor.

$K$  sabitlerinin,  $K=b^n - 1$  en büyük sayıya kadar tümleyeni ve  $K = b^n$  sayı kapsamına kadar tümleyenler kıyaslayarak, ikinci sabit için  $K = (b^n - 1) + 1$  yazabiliriz. Buna göre sayının ikinci tümleyeni birinci tümleyenden, yani birinci tümleyene 1 ekleyerek elde ediliyor.

Sayıların tümleyen değerlerin, devamda göreceğimiz gibi, verilerin dijital işletiminde çok büyük önemi vardır. Şöyle ki, onlarla ön işaretli sayıların (pozitif ve negatif sayıların) tanıtımı sağlanıyor, bununla ise temel aritmetik işlemlerin, toplama ve çıkarma işlemlerin gerçekleşmesi de sağlanıyor.

### 1.7.1. SAYILARIN HERHANGİ SAYI SİSTEMİNDEN ONLU SAYI SİSTEMİNE DÖNÜŞÜMÜ

İlk önce insanların alıştığı ve her günlük hayatta kullandığı onlu sayı sistemini inceleyeceğiz. Bu sayı sisteminin on farklı rakamı vardır. Bu rakamlar şunlardır:  $c=\{0,1,2,3,4,5,6,7,8,9\}$  ve tabanı  $b=10$ 'dur. Ağırlık formülünün uygulanmasını  $5387_{(10)}$  sayısı ile açıklayacağız.

Formülü uygulayarak şunu elde ediyoruz:

$$5387_{(10)} = 5 \cdot 10^3 + 3 \cdot 10^2 + 8 \cdot 10^1 + 7 \cdot 10^0 = 5000 + 300 + 80 + 7 = 5387_{(10)}$$

Çözümde görüldüğü gibi, onlu sayı sisteminde yazılan sayının ağırlık formülüne (1-3) göre de aynı değeri var, çünkü bu formül zaten sayının insan için anlaşılır onlu sisteminde değerini veriyor.

On altılı sayı sisteminin 16 farklı rakamı var ve onlar şöyle işaretleniyor:  $c=\{0,1,2,3,4,5,6,7,8,9,A,B,C, D,E,F\}$  ve tabanı  $b=16$ -dır. Bu sistemin, onlu sisteminden altı sayının belirtmesi için altı ek rakamı var. A rakamıyla  $10_{(10)}$  sayısı belirtiliyor, B ile  $11_{(10)}$ , C ile  $12_{(10)}$  sayısı, D ile  $13_{(10)}$ , E ile  $14_{(10)}$  ve F rakamıyla  $15_{(10)}$  sayısı ifade ediliyor.

Bir on altılı sayının değerinin hesaplanması ya da buna eşit olan onlu sayı sistemine dönüşümü, aşağıdaki örnekle verilmiştir. Yeniden ağırlık formülü uygulanarak, öyle ki şu elde ediliyor:

$$A2B_{(16)} = A \cdot 16^2 + 2 \cdot 16^1 + B \cdot 16^0 = 10 \cdot 16^2 + 2 \cdot 16^1 + 11 \cdot 16^0 = 2560 + 32 + 11 = 2603_{(10)}$$

Sekizli sayı sisteminin 8 rakamı:  $c=\{0,1,2,3,4,5,6,7\}$  ve  $b=8$  tabanı vardır. Burada her rakamın ağırlığı 8 sayısının üsü olacak, çünkü 8 bu sayı sisteminin tabanıdır. Sekizli sayı sisteminden onlu sayı sistemine dönüşümü örnekle göreceğiz. Örnekte  $157_{(8)}$  sayısının dönüşümü yapılıyor.

$$157_{(8)} = 1 \cdot 8^2 + 5 \cdot 8^1 + 7 \cdot 8^0 = 64 + 40 + 7 = 111_{(10)}$$

Onlu	On altılı	İkili	Sekizli
0	0	0000	0
1	1	0001	1
2	2	0010	2
3	3	0011	3
4	4	0100	4
5	5	0101	5
6	6	0110	6
7	7	0111	7
8	8	1000	10
9	9	1001	11
10	A	1010	12
11	B	1011	13
12	C	1100	14
13	D	1101	15
14	E	1110	16
15	F	1111	17

Tab.1-1. Farklı sayı sistemlerinde kullanılan rakamların gözden geçirmesi

Doğal ikili sayı sisteminin sadece iki rakamı var: 0 ve 1. Bu iki rakamdan herhangi *bit* olarak adlandırılıyor çünkü İngilizce *Binary digit* sözlerinin kısaltmasıdır ve **ikili rakam** anlamına geliyor. İkili sistemde en büyük ağırlıklı rakam *MSB (Most Significant Bit)*, olarak belirtiliyor ve *en değerli* ya da *en çok değerli bit* demektir, en küçük ağırlıklı bit ise *LSB (Last Significant Bit)* ile belirtiliyor ve *en değersiz ya da en az değerli bit* demektir. İkili sayı sisteminden onlu sayı sistemine geçiş yeniden ağırlık formülü (1-3) ile yapılıyor. Bu dönüşüm,  $10010010_{(2)}$  ikili sayının dönüşümüyle gösterilmiştir:

$$10010011_{(2)} = 1 \cdot 2^7 + 0 \cdot 2^6 + 0 \cdot 2^5 + 1 \cdot 2^4 + 0 \cdot 2^3 + 0 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 2^0 = 128 + 16 + 2 + 1 = 147_{(10)}$$

Tab.1-1'de farklı sayı sistemlerinden tüm rakamların simgeleri ve eşdeğerli değerleri verilmiştir.

### 1.7.2. İKİLİ SAYI SİSTEMİNDEN ON ALTILI VE SEKİZLİ SAYI SİSTEMİNE VE TERSİNE DÖNÜŞÜM

On altılı sayı sisteminden ikili sayı sistemine dönüşüm ve tersi çok kolay yapılıyor çünkü  $16 = 2^4$  hem de tablo1-1'in doğrudan uygulanmasıyla. On altılı sistemden ikili sisteme dönüşüm sırasında, aşağıdaki örnekte verilmiş olduğu gibi, her on altılı rakam basit bir şekilde uygun nibil (dört bir grubu) ile değiştiriliyor. En sol pozisyonlarda sıfırlar meydana gelirse, onlar basitçe ihmal ediliyor çünkü hiçbir ağırlıkları yoktur.

$$5C_{(16)} = 0101\ 1100 = 01011100_{(2)} = 1011100_{(2)}$$

Ters durumda, ikili sayı sisteminden on altılı sayı sistemine dönüşüm yapmak gerekince, verilen ikili sayı pozsyon noktasından sola ve sağa dörder bit gruplara ayrılıyor. Bu ayrılma sırasında en solda ve en sağda dört bit elde edilmiyorsa, en sol gruba önden, en sağ gruba ise arkadan, dört bit elde edilene kadar 0-lar ekleniyor. Sadece *tam sayılarla* çalıştığımızdan dolayı soldan sağa dörtlü gruplar oluşturuyoruz. Sonunda *nibil* elde edilmezse, o zaman önden bit dörtlüsü elde etmek için gerektiği kadar 0-lar ekleniyor. Ardından her nibil tablo 1-1'e göre uygun onaltılı rakamla değiştiriliyor. Devamda  $101010_{(2)}$  ikili sayısının on altılı sayıya dönüşümü verilmiştir:

$$101010_{(2)} = 0010 \cdot 1010 = 2A_{(16)}$$

Sekizli sistemden ikili sisteme dönüşüm ve ters yönde dönüşüm süreci, on altılı-ikili dönüşümünde kullanılan süreçle eşittir, sadece bu durumda üç bitli gruplarla çalışılıyor çünkü  $8 = 2^3$ . Bu dönüşüm aşağıdaki örneklerle gösterilmiştir.

$$421_{(8)} = 100 \cdot 010 \cdot 001 = 110010001_{(2)}$$

$$10110011_{(2)} = 010 \cdot 110 \cdot 011 = 263_{(8)}$$

Sekizli sayı sisteminden on altılı sayı sistemine dönüşüm ve tersi, aşağıdaki örneklerde verilmiş olduğu gibi, en kolay olarak, bir sayı sisteminde verilmiş olan sayıyı önce ikili sayı sistemine dönüştürerek, ardından elde edilen sayıyı ikili sayı sisteminden diğer sayı sistemine dönüştürerek yapılıyor.

$$BC_{(16)} = 1011 \cdot 1100 = 10111100_{(2)} = 010 \cdot 111 \cdot 100 = 274_{(8)}$$

$$762_{(8)} = 111 \cdot 110 \cdot 010 = 111110010_{(2)} = 0001 \cdot 1111 \cdot 0010 = 1F2$$

Daha basit dönüşüm için “uzun” ikili sayılar genelde kısaltılmış on altılı yazılışta yazılıyor. Bunun nedeni, bilgisayarda veri değişiminin, bir, iki, dört veya fazla bayt uzunluğuna verilerin arasında gerçekleşmesidir. Kısaca bir baytın sekizli olduğunu, yani sekiz bitlik gruptan, diziden oluştuğunu hatırlayalım. Buna göre, veriler genelde 8, 16, 32, 64 vs. bit uzunluğundadır, bu uzunlukların dört ile bölünür olması, onların ikili sayı sisteminden on altılı sayı sistemine ve ters dönüşümü çok kolaydır.

Sekizli yazılış daha seyrek kullanılıyor. Sekizli yazılış, uzunluğu üç ile bölünebilir bit uzunluğunda veriler kullanan bilgisayarlarda uygulanması uygundur. Örneğin, 12 ya da 24 bit uzunluğunda gruplar. Ancak, bu uzunluklar 4 ile de bölünür, öyle ki bu durumda da on altılı yazılışla çalışılabilir.

### 1.7.3. ONLU SAYI SİSTEMİNDEN HERHANGİ SAYI SİSTEMİNE DÖNÜŞÜM

Onlu sayı sisteminden herhangi bir başka sayı sistemine dönüşüm o şekilde yapılıyor ki, verilen onlu sayıyı, dönüşüm yapmak istediğimiz sayı sisteminin tabanıyla bölünüyor. Bölme sırasında, verilen sayının yanında sağ taraftan yazılan bölüm ve bölüm altında yazılan belirli kalan elde ediliyor. Şimdi bu ilk bölüm tabanla bölünüyor ve yeniden sağ tarafta yazılan ikinci bölüm ve yeniden ikinci bölüm altında yazılan ikinci kalan elde ediliyor. Bu süreç, bölüm 0 elde edilene kadar, bu bölümle elde edilen kalan önemli olmadan devam ediyor.

Sayı rakamları, aslında her ayrı bölmenin kalanlarıdır. Son kalan sayının en değerli rakamı (*MSD*), elde edilen birinci kalan ise en az ağırlıklı rakamdır (*LSD*). Daha basit söyleyişle, sayı rakamları elde edilen kalanların ters yönde, yani sağdan sola yazılmasıyla elde ediliyor.

Söylediklerimizi, onlu sayı sisteminden on altılı, sekizli ve ikili sayı sisteme dönüşüm örneklerinin incelenmesiyle göstereceğiz.  $15797_{(10)}$  on altılı yazılışla  $3DB1_{(16)}$ ,  $3336_{(10)}$  sekizli yazılışla  $6410_{(8)}$  ve  $155_{(10)}$  ikili yazılışla  $100110011_{(2)}$  olacak.

Onlu sayı / 16	11797	987	61	3	0
Kalan:		1	11	13	3

$$\begin{array}{r} B \quad D \\ \hline \end{array}$$

On altılı sayı:                    3                    D                    B                    1

Onlu sayı / 8	3336	417	12	6	0
Kalan:		0	1	4	6

On altılı sayı:                    6                    4                    1                    0

Onlu sayı / 2	155	77	38	19	9	4	2	1	0
Kalan:		1	1	0	1	1	0	0	1

İkili sayı: 1 0 0 1 1 0 1 1

Onlu sistemden ikili sistemine dönüşümü sıkça yapmamız gerektiğinden dolayı, devamda yukarıda verilen örnek üzerine uygulayacağımız daha hızlı süreç göstereceğiz. İlk önce birkaç bit uzunluğunda bellek sözcüğünün elimizde olduğunu ve her bit üstüne uygun ağırlığının yazılmış olduğunu tahmin edeceğiz. Şimdi, verilen onlu sayının yazılan ağırlıklara uyup uymadığını bakıyoruz ya da hangisiden daha küçük olduğuna bakıyoruz. İncelediğimiz örnekte  $155 < 256$  olduğuna göre 128 ağırlığı en büyüktür ve 1 ile işaretliyoruz. Bu ağırlık b7 bitini uyar ve 155 sayısı için ikili şekilde en büyük ağırlığı olacak (MSB). Buna göre, dönüşüm sırasında 10 bitten sadece 8 bit (1 bayt) kullanacağız, daha büyük ağırlıklı diğer bitlere ise gerek yok ve onların hepsi 0 olacak. Şimdi, bu pozisyondan bağlayarak, aşağıya doğru, 128'i sıradaki ağırlıkla topluyoruz.  $(128+64) > 155$  olduğundan dolayı, bu ağırlığı dikkate almıyoruz, yani bu pozisyonda bit 0 yazıyoruz. 32 ağırlıkla devam ediyoruz. Burada da toplamın  $(128+32=)160 > 155$  olduğundan dolayı, bu pozisyonda da 0 yazıyoruz. 16 ile devam ediyoruz.  $(128+16=)144 < 155$  olduğundan dolayı, 16 ağırlığı altında 1 yazıyoruz ve kontrol etmekle devam ediyoruz, ancak şimdi 144 değerimiz var ve sıradaki ağırlık 8'dir.  $(144+8=)152 < 155$  olduğuna göre, bu biti de alıyoruz. Şimdi  $152+4$  kontrol ediyoruz. Bu toplam 155'ten daha büyüktür ve 4 ağırlığı altındaki biti almıyoruz, ancak tam olarak 3 değeri daha gerekiyor ve bu değer için 2 ve 1 ağırlıkları olan son iki biti alıyoruz. Böylece en sonuna  $152 + 2 + 1 = 155$  elde ediyoruz. Önümüzde ikili ağırlıklı tablo olunca, onlu sistemden ikili sisteme dönüşüm, kalanlarla bölmemizin ve aranan ikili sayıyı elde etmek için kalanları tersine döndürmemiz gerektiği önceki yönteme kıyasen çok daha hızlıdır.

Ağırlıklar:	<u>1024</u>	<u>512</u>	<u>256</u>	<u>128</u>	<u>64</u>	<u>32</u>	<u>16</u>	<u>8</u>	<u>4</u>	<u>2</u>	<u>1</u>
	$2^{10}$	$2^9$	$2^8$	$2^7$	$2^6$	$2^5$	$2^4$	$2^3$	$2^2$	$2^1$	$2^0$
İkili sayı:				1	0	0	1	1	0	1	1
Pozisyonlar:	<u>10</u>	<u>9</u>	<u>8</u>	<u>7</u>	<u>6</u>	<u>5</u>	<u>4</u>	<u>3</u>	<u>2</u>	<u>1</u>	<u>0</u>
Bitler:	11	10	9	8	7	6	5	4	3	2	1

### 1.7.4. İKİLİ SAYI SİSTEM ARİTMETİĞİ

İkili aritmetik, ikili sayı sisteminde toplama, çıkarma, çarpma ve bölme işlemlerinin gerçekleştiği kuralları tanımlıyor.

**Toplama.** *Toplamanın* yapıldığı kurallar şunlardır:

1. İki 0'ın toplamı 0 veriyor;
2. 0 ve 1'in, ya da 1 ve 0'ın toplamı 1 veriyor;
3. İki 1'in toplamı sonuç olarak sıfır veriyor, ancak aynı zamanda daha büyük ağırlıklı (daha yüksek seviyeli, sınıflı) bite elde (carry) 1 de veriyor. (1 taşıyor)

Bu, en iyi şekilde aşağıda verilen  $101111_{(2)}$  ve  $111_{(2)}$  sayıların toplamı örneğiyle anlaşılabilir:

Elde (taşıma):		1	1	1	1	
Birinci toplayan:	1	0	1	1	1	1
İkinci toplayan:	0	0	0	1	1	1
Toplam:	1	1	1	0	0	1

**Çıkarma.** İkili sistemde çıkarma kuralları şunlardır:

- 0'dan 0 çıkarılırsa, veya 1'den 1 çıkarılırsa, sonuç 0-dır;
- 1'den 0 çıkarılırsa, sonuç olarak 1 elde ediliyor;
- 0'dan 1 çıkarılırsa, daha yüksek ağırlığı olan bitten 1 ödünç alınıyor. Orada 0 kalacak, daha az ağırlıklı bite ise  $2_{(10)}$ , yani  $10_{(2)}$  taşınıyor. Şimdi daha alçak seviyeye taşınan  $2_{(10)}$ 'den, yani  $10_{(2)}$ 'den, 1'i alınarak, 1 kalıyor ve sonuç olarak 1 elde ediliyor.

Yeniden bir örnek inceleyeceğiz. Bu örnekte  $11_{(2)}$  sayısı,  $10110_{(2)}$ 'den çıkarılıyor.

				10	
Ödünç almalar:			0	θ	10
Çıkarılan:	1	0	±	±	θ
Çıkaran:				1	1
Fark:	1	0	0	1	1

**Çarpma.** İkili çarpmanın gerçekleştiği kurallar şunlardır:

- Çarpanlardan biri 0 olunca sonuç 0'dır;
- Sadece iki çarpan 1 olunca, sonuç 1 olur.

İki ikili sayının çarpılması sırasında onlu çarpmadan şu prensip uygulanıyor: önce çarpanın LSB'i verilen sayıyla (çarpılanla) çarpılıyor ve birinci bölümlü çarpım elde ediliyor. Ardından çarpılan, çarpan LSB-den soldan birinci bite çarpılıyor ve elde edilen ikinci bölümlü çarpım, birinci bölümlü çarpım altında yazılıyor, ancak bir yer için sola kayarak yazılıyor vs. Bu işlem çarpanın MSB'nin verilen sayıyla (çarpılanla) çarpmanın sonucu olarak son bölümlü çarpımın elde edilmesine kadar devam ediyor. Devamda çarpma ile ilgili örnek verilmiştir.

Çarpılan:			1	1	0	1
Çarpan:				1	0	1
Birinci bölümlü çarpım:			1	1	0	1
İkinci kısmen çarpım:			0	0	0	0
Üçüncü kısmen çarpım:		1	1	0	1	
Çarpım:	1	0	0	0	0	0
						1



**Bölme.** Bölme için şu kurallar geçerlidir:

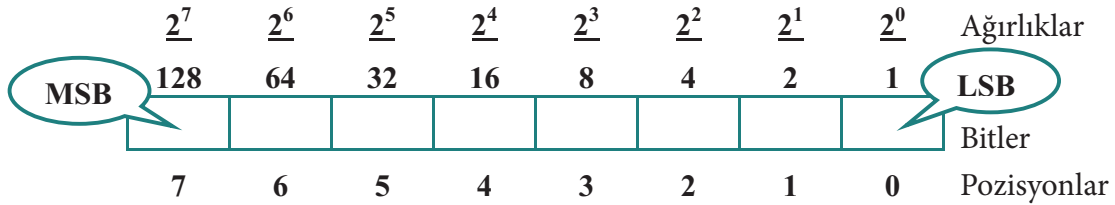
1. 0 ile bölme tanımlanmış değildir,
2. 0, 1 ile bölünürse bölüm 0'dır, ve
3. 1, 1 ile bölünürse 1 elde ediliyor.

İkili bölme, onlu bölüme benzerdir, sadece çarpma ve çıkarma işlemleri daha basittir çünkü ikili sistemde gerçekleşiyor, bölümün rakamları ise çok kolay elde ediliyor çünkü onlar ya 1 ya 0 olabilir. Devamda bölme ile ilgili bir örnek verilmiştir: bölünen 11110'dır, bölen ise 110'dur.

$$\begin{array}{r}
 11\ 110 : 110 = 101 \\
 -110 \\
 \hline
 = = 110 \\
 - 110 \\
 \hline
 = = =
 \end{array}$$

### 1.7.5. POZİTİF VE NEGATİF SAYILARIN İŞARETLENMESİ

Şimdiye kadar incelediğimiz derslerde, bilgisayara girile verilerin sadece pozitif tam sayılar ya da sıfır olabileceğini tahmin etmiştik. Şöyle ki, verilerin Şek 1-6'da gösterilmiş olduğu gibi, uzunluğu 1 bayt (8 bit) olan kelimeler olarak tanıttığımızı tahmin edersek, yedinci bitin en büyük ağırlığı olacak  $-2^7=128$  ve bu bit MSB olacak, sıfırıncı bitin ise en küçük ağırlığı  $2^0 = 1$  olacak ve bu bit LSB olacak. Sözcüklerin tüm 8 biti doğal ikili sayı sisteminde kodlamak için kullanılırsa, o zaman (1-1) denklemini uygulayarak, bu 8 bitle toplam  $2^8=256$  sayı belirtebiliriz. İkili yazılımda en küçük  $00000000_{(2)}$  sayısından en büyük olan  $11111111_{(2)}$  sayısına kadar ya da onlu yazılımda  $0_{(10)}$ 'dan  $255_{(10)}$ 'e kadar sayılar işaretlenilebilir.



Şek.1-7. Doğal ikili sayı sistemine göre 1 bayt uzunluğunda bellek sözcüğü

Ancak, bilgilerin işletimi sırasında, tam pozitif sayılarla çalışma dışında, tabii ki negatif sayıları da göz önüne almalıyız, öyle ki onların değeri ayrı şekilde işaretlenmelidir. Her günlük çalışmada, sıradan aritmetikte, pozitif sayılar, önlerinde “+” işareti koyularak işaretleniyor ya da bu işaret pozitif sayılar için yazılmıyor, ancak her negatif sayı önünde “-” işareti duruyor. Bilgisayarlar sadece ikili şekilde yazılmış sayılarla çalıştığından dolayı, ön işaret için ayrı bit kullanılıyor. Bu bit genelde ikili vektörde en sol pozisyonda bulunuyor. Bu arada “+” işareti “0” bitiyle değiştiriliyor, “-” işareti ise “1” bitiyle değiştiriliyor.

Pozitif sayılar tüm sistemlerde aynı şekilde işaretleniyor. Onlarda ön işaret, doğal ikili sayı sisteminde olduğu gibi, en yüksek pozisyon yerinde 0 yazılarak belirtiliyor.

$$\text{Örn. 1. } 69_{(10)} = 1000101_{(2)} \Rightarrow (+ 69) = 0\ 1000101 = 01000101$$



Negatif sayısal değerler üç farklı şekilde tanımlanabilir: sadece işaret bitin değiştirildiği SM sistem ile (İng. *sign and magnitude*), DC sistem (İng. *digit complement*) ile ya da birinci tümleyen (bire kadar tümleyen, 1's) ile ve RC sistemi (İng. *range complement*) ya da ikinci tümleyen (ikiye kadar tümleyen, 2's) ile.

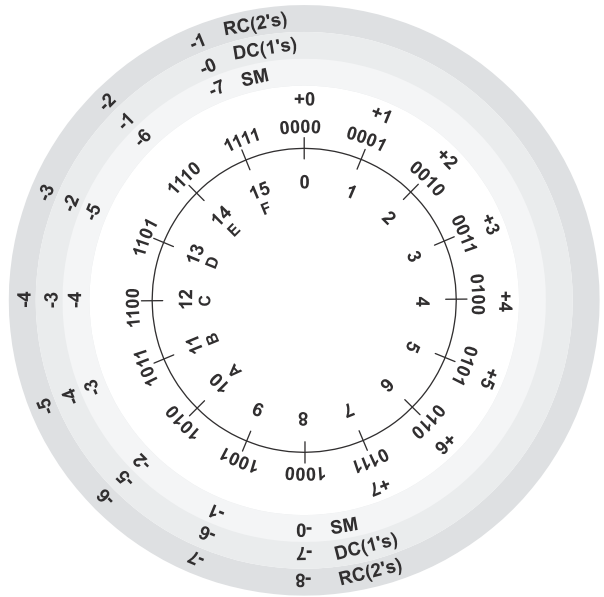
**SM sistem:** *ön işaretli yazılımlı* bu sistemde ya da *işaret ve değer* sisteminde sayının birinci biti sayının işaretini belirliyor, diğer bitler ise değerseldir (ağırlıklıdır). Bu ağırlıklı bitler doğal ikili sayı sisteminde negatif sayının mutlak değerini tanımlıyor. Bu yazılım şeklini en iyi olarak, SM sisteminde sayıların nasıl yazıldığı ile ilgili iki örnek inceleyerek anlaşılabilir:

Örn. 2.  $4_{(10)} = 100_{(2)}$  olduğundan dolayı  $(+4) = 0\ 100 = 0100_{(SM)}$ ,  $(-4) = 1\ 100 = 1100_{(SM)}$ ;

Örn. 3.  $5_{(10)} = 101_{(2)}$  olduğundan dolayı  $(+5) = 0\ 101 = 0101_{(SM)}$ ,  $(-5) = 1\ 101 = 1101_{(SM)}$ ;

Örneklere görüldüğü gibi negatif ön işareti en basitçe 1 ile değiştiriliyor onlu değer ise ikili değerle değiştiriliyor.

**TC sistemleri:** Sayıların negatif değerlerinin ifade edilmesi genelde tümleyen ikili sayıların uygulanmasıyla yapılıyor. Bu tanıma şekli, negatif sayıların böyle tümleyen şekilde işaretmesiyle aritmetik işlemlerin gerçekleştiği sırada ön işaret bitiyle, değersel (ağırlık) bitleriyle çalışıldığı gibi aynı şekilde çalıştığından geliyor. İki tümleyen sistem vardır: DC sistemi ya da birinci tümleyen (1's) ve RC sistemi ya da ikinci tümleyen (2's). Şek.1-8'deki çembersel tanımla ve Tab.1-2 ile dörder bitle kodlanmış sayısal değerlerin gösterme şekilleri verilmiştir.



Şek. 1-8. 4 bitle kodlanmış ön işaretli sayıların farklı sistemlerde tanıtımı

Pozitif sayılar		Negatif sayılar								
SM, DC, RC	Değ	SM		Değ	DC (1's)		Değ	RC (2's)		Değ
0 0 0 0	+0	1	0 0 0	-0	1	0 0 0	-7	1	0 0 0	-8
0 0 0 1	+1	1	0 0 1	-1	1	0 0 1	-6	1	0 0 1	-7
0 0 1 0	+2	1	0 1 0	-2	1	0 1 0	-5	1	0 1 0	-6
0 0 1 1	+3	1	0 1 1	-3	1	0 1 1	-4	1	0 1 1	-5
0 1 0 0	+4	1	1 0 0	-4	1	1 0 0	-3	1	1 0 0	-4
0 1 0 1	+5	1	1 0 1	-5	1	1 0 1	-2	1	1 0 1	-3
0 1 1 0	+6	1	1 1 0	-6	1	1 1 0	-1	1	1 1 0	-2
0 1 1 1	+7	1	1 1 1	-7	1	1 1 1	-0	1	1 1 1	-1

Tab. 1-2. 4 bitle kodlanmış ön işaretli sayıların farklı sistemlerde tanıtımı

Verilen tablodan, pozitif sayıların, her belirtme şeklinde aynı şekilde yazıldığını ve negatif sayılar için en büyük ağırlıklı bitin (MSB), hangi belirtme şeklini kullandığımız önemli olmadan, değeri 1 olduğunu görebiliriz. Ayrıca, tablodan görüldüğü gibi sayıların ön işaretle ya da birinci tümleyenle yazıldığı sırasında, sıfır iki şekilde ifade edilebilir: pozitif ve negatif sıfır olarak. Bu arada tanımlanan negatif ve pozitif sayıların sayısı eşittir. SM ve DC işaretlemeye iki sıfır vardır: pozitif ve negatif sıfır, RC işaretlemesi ise pozitif olarak alınan sadece bir sıfır değeri olduğuna göre farklıdır. Böylece RC işaretlemeye pozitif sayıdan bir negatif sayı fazla vardır.

**DC sistemi:** Örnek olarak  $6_{(10)} = 110_{(2)}$  sayısının negatif değerinin, yani  $-6_{(10)}$  sayısının ifade edilmiş şeklini göreceğiz. DC sistemine göre bu değer tanınması kolaydır, çünkü önce sayının mutlak değeri belirleniyor, ardından ikili sayı olarak yazılıyor ve en sonunda elde edilen iki sayıdan, aşağıdaki örnekte verilmiş olduğu gibi her bir bit birer birer ayırarak tümleşiyor

Verilen negatif sayı: - 6

Mutlak değer: 6

İkili eşdeğer: 0110

Tümleşme:  $1001_{(1's)}$ , yani DC sisteminde kodlanmış -6 sayısı.

Negatif sayıların birinci tümleyenle onlu sistemine dönüşüm aynı şekilde yapılıyor. Önce verilen ikili vektörden her bit ayırarak tümleşiyor, ardından onun yeni elde edilen ikili kombinasyondan onlu değeri belirleniyor ve sonunda işareti ekleniyor. Devamdaki örnekte, DC sistemin söz olduğunu tahmin ederek,  $11010001_{(1's)}$  ikili vektöründen onlu değer belirlenmesi gerekiyor.

Verilen ikili sayı: 11010001

Tümleşme (1's): 00101110

Mutlak değer:  $00101110 = 46$

Onlu sayı: - 46

Birinci tümleyenle negatif sayıların onlu sistemine dönüşümü, daha hızlı ve daha basit şekilde yapılabilir. Şöyle ki, sayıda sıfırların bulunduğu pozisyonların (yerlerin) ağırlıkları toplanıyor ve işaret ekleniyor. Bu yöntemle göre, önceki  $11010001$  sayısının örneği için şunu elde ediyoruz:

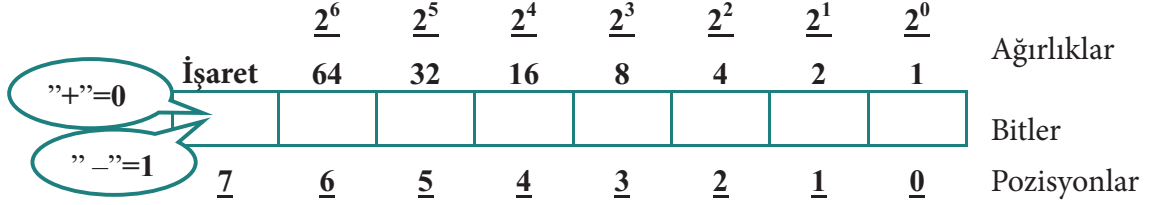
$$2^5 + 2^3 + 2^2 + 2^1 = 32 + 8 + 4 + 2 = 46, \text{ ya da } 11010001_{(1's)} = - 46.$$

Pratikte her üç yöntem kullanılıyor, ancak en önemli yöntem çift tümleyenli işaretlemeye, çünkü bu yöntemle en basit şekilde ön işaretli tam sayıların toplam ve çıkarma aritmetik işlemleri yapılabilir. Bu yüzden bu yöntemle devamda ayrı dikkat verilmiştir.

### 1.7.6. ÇİFT TÜMLEYENLE İŞARETLEME

Bu işaretleme İngilizce orijinal olarak *two's compliment notation* adlandırılıyor, kısaca *2's complement* olarak işaretleniyor ya da sayı yanına indis olarak (2's) ekleniyor. Bu yazılış şekli sıkça olarak işaretli tam sayılarla çalışıldığı zaman kullanılıyor.

Sayıların çift tümleyenle tanımlandığı sırasında da işaret için bitin kullanıldığı en önemlidir. İşaret biti en yüksek yedinci pozisyonda bulunuyor, yani sekizinci bittir. Şek.1-9'dan görüldüğü gibi, bu bitin değeri 0 ise, o zaman sayı pozitifdir, ancak bu bit 1 ise o zaman sayı negatiftir.



Şek. 1-9. Çift tümleyeni tanımlayan bayt

Bu durumda da, denk.  $(1-1)$ 'e göre yine  $2^8 = 256$  farklı sayı yazılabilir, ancak onlardan 128 pozitif olacak, 128 ise negatif olacak. Çift tümleyenle en büyük pozitif sayı 0111111 sayıdır ya da onlu +127 sayıdır. En küçük sayı ise 00000000, ya da onlu yazılıta 0 sayıdır. Buna göre sıfır pozitif sayı olduğu alınıyor. Şimdiye kadar söylenenleri ve bu son açıklamayı göze önüne alarak, sezgisel olarak en büyük negatif sayının -1, en küçük negatif sayının ise -128 olduğu tahmin edilebilir.

Negatif sayıların tanımlanması için devamda söz edeceğiz. Verilen herhangi tam negatif onlu sayının çift tümleyen şekline dönüşümü şu adımları uygulanmasıyla yapılabilir:

1. Sayının mutlak değeri onlu şekilde yazılıyor;
2. Onlu sayıdan ikili sayıya dönüşüm yapılıyor;
3. Her bir ayrıdan tümleşerek, ikili sayının birinci tümleyeni elde ediliyor (1's), ve
4. Sayının elde edilen birinci tümleyen 1 için artıyor (ekleniyor)

Elde edilen sonuç, verilen negatif sayının çift tümleyende (2's) gösterimidir. Devamda, bilgisayarın 1 bayt uzunluğunda verilerle çalıştığını tahmin ederek,  $-6_{(10)}$  negatif sayısının 2's tümleyen şekile dönüşümü gösterilmiştir.

Onlu sayı:	- 6
Mutlak değer:	6
İkili sayı:	00000110
Birinci tümleyen 1's:	11111001
1 için artma:	+ 1
	-----
İkinci tümleyen (2's):	11111010

Ters sorun ortaya çıkarsa ya da çift tümleyende verilmiş negatif sayının onlu değerinde eşdeğeri aranır, o zaman önceden açıklana aynı sürecin gerçekleşmesi gerekiyor:

1. Verilen ikili 2's sayısının birer birer biti tümleşerek onun birinci tümleşen gösterimi elde ediliyor;
2. Sayının elde edilen birinci tümleyeni 1 için büyütülüyor;
3. Elde edilen ikili sayının onlu sayı sistemine dönüşümü yapılıyor ve “-“ (eksi) işareti ekleniyor.

Sıradaki örnekte 11110001(2's) ikili vektörün onlu sayı sistemine dönüşümü yapılıyor.

$$\begin{array}{r}
 \text{Sayının ikinci tümleyeni}_{(2's)}: \quad 11110001 \\
 \text{Birinci tümleyen}_{(1's)}: \quad 00001110 \\
 \text{1 için artma:} \quad \quad \quad \quad + 1 \\
 \text{-----} \\
 \text{Mutlak değer:} \quad \quad \quad \quad 00001111 = 15 \\
 \text{Onlu sayı:} \quad \quad \quad \quad \quad - 15
 \end{array}$$

Çift tümleyenden negatif sayıların onlu sisteme dönüşümü daha basit şekilde de yapılabilir. Şöyle ki, sayıda 0-ların bulunduğu yerlerde ağırlıklar toplanıyor ve elde edilen sonuca 1 ekleniyor ve sonunda işarette yazılıyor. Önceki örnek için, 11110001 için şunu elde ediyoruz:

$$(2^3 + 2^2 + 2^1) + 1 = (8 + 4 + 2) + 1 = 15 = 16, \text{ ya da } 11110000_{(2's)} = - 16.$$

Çift tümleyenle ifade edilen sayılarla toplama işlemi çok kolay gerçekleşiyor, en önemlisi ise çıkarmanın da toplama ile yapılmasıdır. Süreç oldukça basittir:

1. Çift tümleyende tanımlanmış verilen sayılar, sıradan ikili sayılar gibi toplanıyor (işaret biti diğer bitler gibi alınıyor);
2. Bu arada sekizinci bitte (yedinci pozisyonda) taşırma (aşma, overflow) ya da aktarma varsa bu aktarma dikkate alınmıyor, sonucun diğer sekiz biti ise çözümü veriyorlar ve
3. Taşırma yoksa elde edilen toplam aranan çözümdür.

Devamda verilen örnekler, işaretli sayıların toplama ve çıkarma süreçlerini gösteriyorlar. Birinci örnekte 5 ve 3 sayılarının toplanması verilmiştir, ikinci örnekte ise 6'nın 2'den çıkarılması, ya da (+2)+(-6), sonuç olarak -4 elde ediliyor.

$$\begin{array}{r}
 00000101 \quad 5 \\
 + 00000011 \quad + 3 \\
 \text{-----} \\
 00001000 \quad +8
 \end{array}
 \quad
 \begin{array}{r}
 00000010 \quad 2 \\
 + 11111010 \quad - 6 \\
 \text{-----} \\
 11111100 \quad - 4
 \end{array}$$

## 1.8. İKİLİ KODLAR

**Kodlama** terimi altında, bilgilerin bir kümenin elemanları olan semboller yardımıyla tanımlama şeklidir. Dijital sistemler, sadece iki durumda bulunabilen elektronik anahtarlama elemanları içeriyor, öyle ki onlar için en uygun tanımlama şekli *ikili şekildir*. Bunun nedeni çok basittir. Şöyle ki, ikili şekilde yazılmış herhangi bir veri, 0 ve 1 bitlerinden bir dizi tanımlıyor.

Ulusal alfabeye analog olarak, bilgilerin belirli kodla yazılı ifade edilmesi için elinde olan tüm farklı semboller kümesine **kod alfabeti** denir. İkili kodlar için kod alfabeti sadece iki elemanı olan **ikili kümedir**: *mantıksal sıfır (bit 0) ve mantıksal bir (bit 1)*. Bir kavramın ya da bilginin kodlandığı ya da tanımladığı semboller grubuna **kod sözcüğü (kelimesi)** denir.

Kod sözcüğünün kendi **uzunluğu** var, o da yazıldığı sembollerin toplam sayısıdır. Biz genelde, tüm kod sözcüklerin aynı uzunlukları olan kodlar kullanacağız. Böyle kodlara *düzgün kodlar* denir. İkili kodlar genelde uzunluğu 8 bit, yani 1 bayt olan sözcüklerle çalışıyor.

Düzgün kodlardan farklı olarak, kod sözcüklerin farklı uzunluğu olan düzgün olmayan kodlar da vardır. Herhangi bir ulusal dilde yazmak aslında düzgün olmayan kodla koldlanmaktadır. Aynı herhangi bir sayı sisteminde sayıların yazılması için de geçerlidir.

Eğer bir kod, hiçbir bilgi tanımlayan ya da yeni bir bilgi tanımlamayan en azından bir kodu varsa, o zaman bu kod için *artıklı kod* olduğu deniliyor.

Kod sözcüklerin kodlandığı ve kod çözümlenmesi yapıldığı şekil, matematiksel denklemlerle ya da kurallar ve düzenlemeler grubu yardımıyla ifade ediliyor (örneğin, bir ulusal dilin sözdizimi), ancak sıkça kod tabloları uygulanıyor. Kod tablosunun iki sütunu ve fazla satırı vardır. Sol sütunda, satır satır ardışık olarak, kodlanması gereken semboller yazılıyor. Sağ sütun, aynı şekilde satır satır dolduruluyor, ancak bu sütun her kaynak sembolüne tek anlamda uyan kod sözcükle riyle dolduruluyor. İkili kodlama söz konusu olunca, sol tarafta harfler ya da onlu rakamlar, sağ tarafta ise ikili kodlu sözcükler yazılıyor.

İkili kodlar *ağırlıklı* ya da *ardışık* olabilir. *Ağırlıklı kodlarda* bitlerin birleştirilmesi (kombinasyonu) o şekilde yapılıyor ki kod sözcüğünden her bite belirli ağırlık veriliyor. Ağırlıklı kod için örnek olarak, yukarıda açıklanan doğal ikili sayı sistemini alabiliriz. Fakat, kod kombinasyonları ya da sözcükleri başka bazı ağırlıklı kanunlara göre de oluşabilir. Büyük sayıda özel amaçlı ikili kodlar bu şekilde gelişmiştir. Ağırlıklı olmayan tüm kodlar *ardışık kodlar* grubuna aittir. Onlarda bitlerin kod sözcüğünde uygun ağırlıkları yoktur, sadece bitlerin kod sözcüğünde sıralaması önemlidir. Bu kodlarda, onlu sayı ve ikili kod sözcüğü arasındaki bağlantı sıkça olarak özel kod tablosuyla veriliyor.

### 1.8.1. SAYISAL KODLAR

Basit dönüşüm amacıyla, ikili sayılar genelde on altılı gösterimde tanımlanıyor. Diğer taraftan, hesap makinelerinde, dijital aletlerde vs, verilerin onlu şekilde girilmesi için veya sonuçların onlu şekilde elde etme gereksinimi ortaya çıkıyor. Ancak, önceden gördüğümüz gibi, ikili sayı sisteminden onlu sayı sistemine dönüşüm oldukça karmaşıktır. İkili ve onlu sayı sistemlerin iyi tarafların birleşmesi için ve insanların onlu şekilde düşünme alışkanlıklarını karşılamak için, farklı *ikili (BCD) kodları* gelişmiş. Bu kısaltma İngilizce *Binary Coded Decima* ifadesinden geliyor ve ikili kodlanmış onlu sayılar anlamına geliyor. Bu kodlarda, onlu sayıdan her bir rakam tekanlamlı eşsiz belirlenmiş kod sözcüğü kullanarak ayrıdan kodlanıyor. Tüm onlu rakamın kodlanabilemesi için, en az 4 bitin kullanılması gerekiyor, çünkü 3 bit yeterli değildir. Şöyle ki, üç bitle  $2^3 = 8$  kod sözcüğü ya da rakam kodlanabilir ve bu sayı kodlanması gereken 10 rakamdan daha küçüktür. Diğer taraftan, 4 bit ile  $2^4=16$  farklı kombinasyon edilebilir ve bu sayı yeterlidir, çünkü  $16>10$ . Buna göre, her onlu rakam için birer nibil (4 bit grubu, tetrad) kullanılıyor. Bu gruplarda bitlerin sıralaması bazı tabloya veya düzenliliğe göre düzenleniyor. 4 bitle 16 farklı tetrad kodlanabileceğinden dolayı, belli ki 6 kod sözcüğü kodlama için kullanılmamış kalacak. Bu sebepten bu kodların atrıklı kodlar olduğu ortaya çıkıyor. Bu yüzden onlu sayıların farklı ikili kodlarla kodlama olanağı vardır (teoretik olarak  $16!/6!$  ya da yaklaşık  $29 \times 10^9$ ).

Onlu rakam	8421 (NBCD)	2421	Ayken	3_fazlalık	5421	Onlu rakam
0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 1 1	0 0 0 0	0
1	0 0 0 1	0 0 0 1	0 0 0 1	0 1 0 0	0 0 0 1	1
2	0 0 1 0	0 0 1 0	0 0 1 0	0 1 0 1	0 0 1 0	2
3	0 0 1 1	0 0 1 1	0 0 1 1	0 1 1 0	0 0 1 1	3
4	0 1 0 0	0 1 0 0	0 1 0 0	0 1 1 1	0 1 0 0	4
5	0 1 0 1	1 0 1 1	1 0 1 1	1 0 0 0	1 0 0 0	5
6	0 1 1 0	1 1 0 0	1 1 0 0	1 0 0 1	1 0 0 1	6
7	0 1 1 1	1 1 0 1	1 1 0 1	1 0 1 0	1 0 1 0	7
8	1 0 0 0	1 1 1 0	1 1 1 0	1 0 1 1	1 0 1 1	8
9	1 0 0 1	1 1 1 1	1 1 1 1	1 1 0 0	1 1 0 0	9

Tab.1-3. Farklı ikili (BCD) kodların kod tabloları

Tablo 1-3 pratikte sıkça kullanılan birkaç ikili (BCD) kodu gösteriyor. En çok kullanılan kod 8421 BCD kodudur. Bu kod *doğal BCD* ya da *NBCD* kodu olarak da biliniyor. Her onlu rakam bir nibil uzunluğunda, benzersiz belirlenmiş kod sözcüğüyle kodlanıyor. Kodun adının 8421 başlangıç belirlemesi, kod sözcüğünde dört bitin ağırlık değerleriyle ilgilidir.

Devamda verilen örneklerle NBCD-koduna kodlama ve kod çöme şekli gösterilmiştir.

$$\text{Örn.1. } 7694_{(10)} = 0111 \cdot 0110 \cdot 1001 \cdot 0100_{(NBCD)} = 0111011010010100_{(NBCD)}$$

$$\text{Örn. 2. } 001101010010_{(NBCD)} = 0011 \cdot 0101 \cdot 0010_{(NBCD)} = 352_{(10)}$$

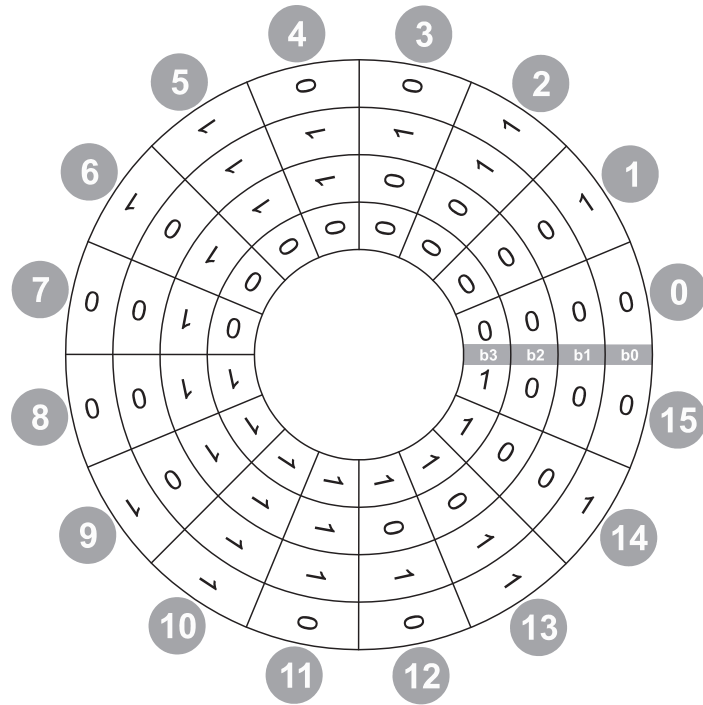
Örneklerden, çalışma prensibinin on altılı sayı sisteminden ikili sayı sistemine ve tersine dönüşümün yapıldığı prensibiyle aynı olduğu görünüyor.

8421 BCD kodu dışında, başka BCD kodları da vardır. Bunlardan daha çok bilinen kodlar şunlardır: *Grey kodu*, *Ayken kodu*, *3\_fazlalık kodu*, *5421*, *ötelemeli kod*, *2421*, vb. Bunlardan bazıları tab.1-3'te verilmiştir. Bu kodlarda kodlama 8421 (NBCD) BCD kodu gibi aynı prensibe göre yapılıyor, sadece kod kombinasyonlarında bitlerin sıralamasında fark vardır.

**Grey kodu.** Bu koda biraz fazla yer vereceğiz, çünkü farklı alanlarda büyük kullanışı var. Örneğin, hataların meydana gelmesi için dijital sinyallerin iletiminde, ardından disklerde olduğu gibi açılışın ikili şekilde gösterilmesi önemli olan cihazlarda, Karno kartlar yöntemiyle mantıksal fonksiyonların küçültülmesinde (bunun için ikinci başta söz edeceğimiz) vb. Grey kodunun en önemli özelliği, komşu kod sözcüklerin birbirinden sadece bir bitte farklı olmalarıdır. Tab.1-4 ve Şek.1-10 Grey koduna göre dört bitle kodlama şekli gösterilmiştir. İki ve üç bit için Grey kodunun kod kombinasyonlarını daha kolay ayırtmak için, tab.1-4 onlar gölgeli belirlenmiştir.

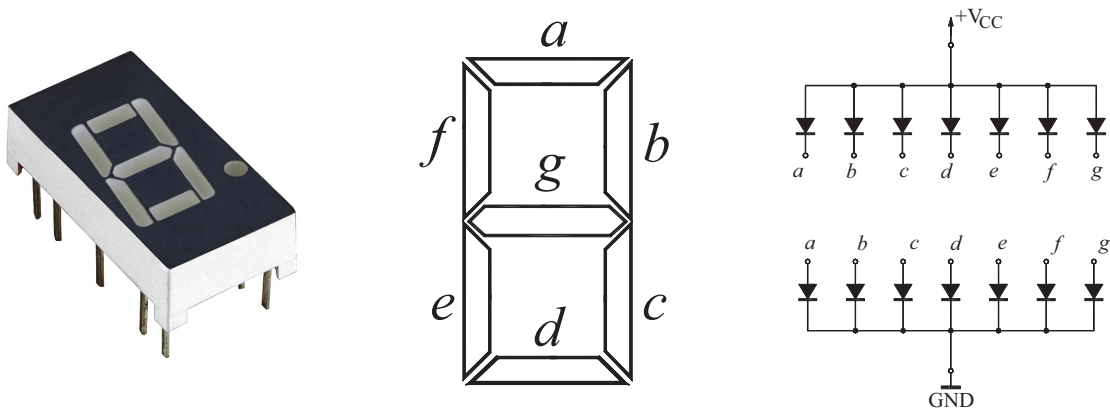
DC	b <sub>3</sub>	b <sub>2</sub>	b <sub>1</sub>	b <sub>0</sub>
0	0	0	0	0
1	0	0	0	1
2	0	0	1	1
3	0	0	1	0
4	0	1	1	0
5	0	1	1	1
6	0	1	0	1
7	0	1	0	0
8	1	1	0	0
9	1	1	0	1
10	1	1	1	1
11	1	1	1	0
12	1	0	1	0
13	1	0	1	1
14	1	0	0	1
15	1	0	0	0

Tab. 1-4.Gery tablosu



Şek. 1-10. Grey kod çemberi

**Yedi bölütlü kod.** En sonunda, sadece pratik amaçla insanın sayısal değerleri daha kolay okuma ihtiyacından dolayı oluşan yedi bölütlü sayısal sistemleri hakkında bahsedeceğiz. Bu kod, sonuçları aydınlatmalı (LED) diyotlu gösterge (küçük ekran) aracılığıyla gösteren dijital cihazlarla ilgilidir. Şek.1-11'de böyle bir ekran gösterilmiştir. Bu ekran a, b, c, d, e, f, g harfleriyle belirtilmiş yedi bölütle oluşuyor. Göstergenin her ayrı bölütü yanabilir veya yanmayabilir. Buna göre her bölüt sadece bir bitle tanımlanabilir. Her onlu rakam aydınlanan ayrı bölütlerin kombinasyonu oluşabilir. Böylece yedi bölütlü sistem yedişer bit on kod sözcüğü kullanılıyor. Bu arada, her ayrı kombinasyon bir onlu rakam tanımlıyor.



Şek. 1-11. Işınlayan (LED) diyotlu gerçek yedi bölütlü gösterge ve onun sembolü

Yedi bölütlü göstergeler ortak anotla (OA, İng.CA) ya da ortak katotla (OK, İng.CC) üretiliyorlar. OK ile göstergelerde tüm katotlar, toprağa bağlanması gereken bir noktada bağlıdır,



anotlar ise birbirinden ayrıdır. Bölütün aydınlanması için uygun anoda yüksek gerilim, mantıksal 1 seviyesi getirilmelidir. OA ekranlarda tüm anotlar ortak bir pinde bağlıdır ve yüksek seviyede olduğu için güçle beslemeye başlanmalıdır, katotlar ise ayrıdır ve herbiri ayrı pinde çıkıyor. Herhangi bir bölüt, uygun katoda alçak gerilim seviyesi, mantıksal sıfır seviyesi ya da topraklama, tablo seviyesine getirildiğinde aydınlanıyor.

Tab.1-5'te birbirine göre karşılıklı tümleyici olan iki yedi bölütlü kodu gösterilmiştir. Biri ortak katotlu göstergeye, diğeri ise ortak anotlu ekrana ilişkindir.

Onlu rakam	Ortak katotlu ekran						
	a	b	c	d	e	f	g
0	1	1	1	1	1	1	0
1	0	1	1	0	0	0	0
2	1	1	0	1	1	0	1
3	1	1	1	1	0	0	1
4	0	1	1	0	0	1	1
5	1	0	1	1	0	1	1
6	1	0	1	1	1	1	1
7	1	1	1	0	0	0	0
8	1	1	1	1	1	1	1
9	1	1	1	1	0	1	1

a) Ortak katotlu ekran

Ortak anotlu ekran							Onlu rakam
a	b	c	d	e	f	g	
0	0	0	0	0	0	1	0
1	0	0	1	1	1	1	1
2	0	0	1	0	0	1	0
3	0	0	0	0	1	1	0
4	1	0	0	1	1	0	0
5	0	1	0	0	1	0	0
6	0	1	0	0	0	0	0
7	0	0	0	1	1	1	1
8	0	0	0	0	0	0	0
9	0	0	0	0	1	0	0

b) Ortak anotlu ekran

Tab. 1-5. LED diyodlu yedi bölütlü ekranın kodlar tablosu

### 1.8.2. ALFASAYISAL KODLAR

İnsan ve bilgisayar arasında iletişim monitör (ekran) yardımıyla ya da yazıcı aracılığıyla gerçekleşiyor. Bu arada, normal olarak çeşit alfasayısal (metinsel) semoller kullanılıyor, örneğin alfabe işaretleri yani küçük ve büyük harfler, noktalama işaretleri, belirli sayısal veriler, yani üzerine matematikle işlemlerin gerçekleşmesi için kullanılmayan sayılar, örneğin telefon numaraları ya da adreslerdeki sayılar, bazı özel grafiksel işaretler vs. Büyük sayıda sembollerin sözkonusu olduğu göz önüne alınarak, uzunluğu 4 bitten daha fazla olan kod sözcüklü kodların kullanma gereksinimi ortaya çıkmış. Saydığımız tüm işaretler *alfasayısal kodlar* olarak adlandırılan özel kodlarla kodlanıyor.

Sıkça olarak ASCII belirtmesini taşıyan ve ASKİ olarak okunan alfasayısal kod kullanılıyor. Kısaltma İngilizce *American Standard Code for Information Interchange* ifadesinden geliyor ve Bilgi Değişimi İçin Standart Amerikan Kodu demektir. Başlangıçta bu kod ABD'nde standart olarak kullanılıyormuş, ardından uluslararası standart olarak da kabul edilmiş ve ISO-7 işaretiyle belirleniyor. Standart *ASCII kod* tablosu tab.1- 6'da verilmiştir. Tablodan görüldüğü gibi bu kod şu sembolleri (İng. characters) kapsıyor: bazı özel kontrol semboller, onlu sayılar, İngilizce alfabesinden tüm büyük ve küçük harfler, noktalama işaretleri, bazı özel grafiksel işaretler ve bazı matematiksel işaretler. Standart *ASCII kodu*, kodlamak için 7 bit kullanıyor. Buna göre onunla  $2^7=128$  farklı işaret kodlanabilir. Bu işaretlerin kodları tab.1-6 ASCII tablosunda verilmiştir.

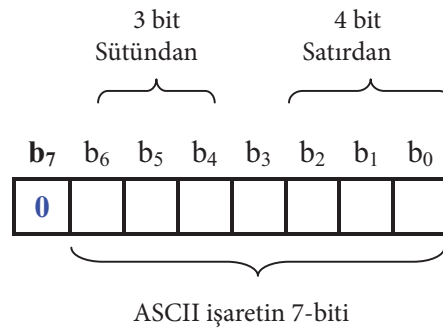


Standart ASCII kodun kıld sözcüklerinin aslen 7 bitleri olmasına rağmen, onların belleklenmesi için 1 bayt, ya da 8 bit uzunluğunda bellek sözcükleri uygulanıyor. Sol taraftan son bit serbesttir, yani kodlama ve kod çözme süreçlerinde bu bitin değeri 0 olduğu alınıyor.

Bitler $b_3b_2b_1b_0$	Bitler $b_6b_5b_4$							
	000	001	010	011	100	101	110	111
0000	NUL	DLE	SP	0	@	P	`	p
0001	SOH	DC1	!	1	A	Q	a	q
0010	STX	DC2	"	2	B	R	b	r
0011	ETX	DC3	#	3	C	S	c	s
0100	EOT	DC4	\$	4	D	T	d	t
0101	ENQ	NAK	%	1	E	U	e	u
0110	ACK	SYN	&	6	F	V	f	v
0111	BEL	ETB	`	7	G	W	g	w
1000	BS	CAN	(	8	H	X	h	x
1001	HT	EM	)	9	I	Y	i	y
1010	LF	SUB	*	:	J	Z	j	z
1011	VT	ESC	+	;	K	[	k	{
1100	FF	FS	,	<	L	\	l	
1101	CR	GS	-	=	M	]	m	}
1110	SO	RS	.	>	N	^	n	~
1111	SI	US	/	?	O	_	o	DEL

Tab. 1-6. Standart ASCII kod tablosu

Her işaretin kod sözcüğü tab.1-6 tablosunu göz önüne alarak ve kod sözcüklerin şek.1-12 gösterilmiş olan oluşma şeklinin uygulanmasıyla elde edilebilir.



Şek. 1-12. ASCII kod sözcüklerin oluşma prensibi

ASCII kodun tab.1-6'da gösterilmiş tablo sadece İngilizce dilinde kullanılan Latin alfabetini kullanıyor. Makedon alfabetinden Ч, ч, К, к, III, ил, Г, г, Ж gibi harfleri veya diğer Avrupa ülkelerinde kullanılan Ä, ü, ö, ř, Ł, é gibi bazı özel harfler içermiyor.

Bu nedenden dolayı, bu tablo sekizinci bitle genişleyerek, yeni 128 sembol için ek yerler kazanılmış yavda tamamıyla 1 bayt kullanılan ASCII-8 kodunda toplam 256 işaret için yer ayrılmış. Ancak bu ek şarteler de farklı ülkelerin alfabelerinin tüm şarteleri için yeterli olmadığı için, farklı ülkelerde farklı tablolar oluşuyor. Böylece, örneğin Makedon alfabelerinden özel şarteler Windows için 1211 işaretli ASCII tablosunda bulunuyor (Cyrillic Code Page 1211 Alphabet). Bu tabloda Rus, Sırp ve Bulgar alfabelerinden harfler de yer alıyor. Slovenya, Hırvatistan, Sırbistan, Çek, Polonya ve diğer Orta Avrupa ülkeleri Windows Code Page Latin 1210 kod tablosunu kullanıyor.

Geçmişte gelişen ve günümüzde de geniş kullanımı olan başka bir alfasayısal kod da vardır, o da *EBCDIC kodudur* (İng. *Extended Binary Coded Decimal Interchange Code*), ya da uzatılmış ikili-kodlanmış ondalık değişim kodu, 8 bit uzunluğunda kod sözcükleri kullanıyor ve tanınık Amerikan bilgisayar üretici şirketi olan IBM tarafından tanıtılmış.

## 1.9. BELİRTİK VE ÖRTÜK DEĞERLER

Bilgilerin ikili vektörler olarak ikili şekilde, yani farklı uzunlukta bitler grubu (dizisi) şeklinde tanımlanmış olduğundan dolayı ve farklı prensibe göre: bir ikili sayı, ön işaretli veya işaretli ya da belli bir ikili koda göre bir kod sözüğü olarak yazıldıklarından dolayı, tek bir terim – **sözcük** (*word*) tanıtılıyor. Bu terimle *belirli uzunlukta herhangi bir grubu* belirtiliyor. Bununla ilgili *sözcüğün (verinin) belirtik ve örtük (açık ve kapalı) değer* terimlerini de tanımlayacağız. Bu değerler, insan için anlaşılır ve verilen ikili vektörün kod çözülmesiyle elde edilen değerlerdir. Şöyle ki, sözcüğün belirtik değeri, doğal ikili sayı sistemine göre yazılmış tüm bitler ağırlıklı bitlerin olduğunu alarak, sözcük bitlerinin dönüşümü ile elde edilen 0-dan büyük ya da eşit olan tam pozitif onlu sayıdır. Diğer taraftan, sözcüğün örtük değeri, sözcük bitlerinin belirli ikili koduna veya ikili sisteme göre dönüştüğü ya da kodun çözüldüğü değerdir.

Bu iki terimi daha kolay anlamamız için iki örnek inceleyeceğiz. Bilgisayarın belleğinde iki bayta şu iki ikili sözcüğün (vektörün): a) 01010100 ve b) 11010100 korunduğunu tahmin edelim.

Örn. 2. (b) 11010100.

Belirtik değer =  $128 + 64 + 16 + 4 = 212$

Örtük değerler:

- SM sistemine göre =  $-(64 + 16 + 4) = -84$ .
- DC sistemine göre =  $-(32 + 8 + 2 + 1) = -43$ .
- RC sistemine göre =  $-[(32 + 8 + 2 + 1) + 1] = -44$
- NBCD koduna göre = 54
- 3-fazlalık koduna göre = 21
- ASCII koduna göre = T.

Örn. 2. (b) 11010100.

Belirtik değer =  $128 + 64 + 16 + 4 = 212$

Örtük değerler:

- SM sistemine göre =  $-(64 + 16 + 4) = -84$ .
- DC sistemine göre =  $-(32 + 8 + 2 + 1) = -43$ .
- RC sistemine göre =  $-[(32 + 8 + 2 + 1) + 1] = -44$
- NBCD koduna göre =  $\text{⌘} 4$
- 3-fazlalık koduna göre =  $\text{⌘} 1$
- ASCII koduna göre =  $\text{⌘}$

Örneklerde “ $\text{⌘}$ ” sembolünün meydana geldiği yerlerde hata belirtiliyor, çünkü öyle bir kod sözcüğü yoktur.

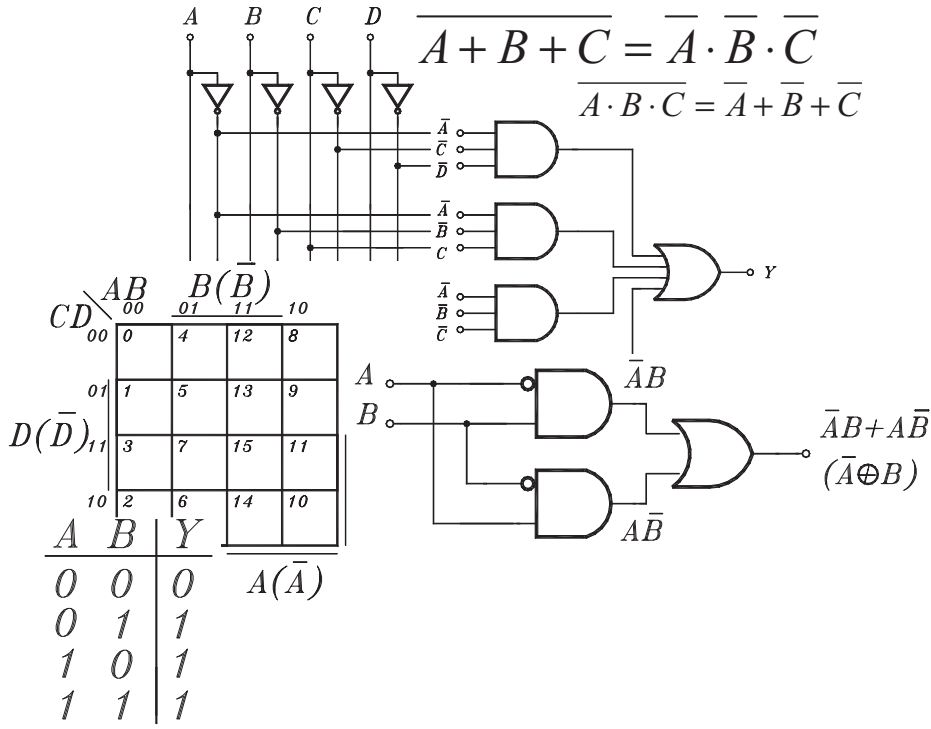
**TEKRARLAMA SORULARI VE ÖDEVLERİ**

- 1-1. Sürekli ve ayırık fonksiyonlar neye göre farklıdır?
- 1-2. Dijital sinyallerin şekli nasıldır?
- 1-3. a)analog b) dijital cihazlar nasıl sinyallerle çalışıyor?
- 1-4. Analog ve dijital çalışma şekillerin avantajlarını ve dezavantajlarını say ve açıkla.
- 1-5. Bilgi terimi altında ne tanımlanıyor?
- 1-6. Soyut seviyede bilgiler nasıl belirleniyor ve gerçek olarak nerede “basılıyorlar”?
- 1-7. Kodama terimi altında ne tanımlanıyor? Kod sözcüğü nedir?
- 1-8. Bilgi miktarının temel ölçü birimi nedir ve dijital elektronik çerçevesinde nasıl tanımlanıyor?
- 1-9. Bit [B] dışında, bilgilerin dijital işletiminde şu daha büyük birimler de kullanılıyor: a) Bir bayt [B]; Bir kilobayt [KB]; c) Bir megabayt [MB]; ç) Bir gigabayt [GB]; d) Bir terabayt [TB]. Baytın uzunluğunu belirle, ardından ise kilobaytın, megabaytın, gigabaytın ve terabaytın kaç bayt içerdiğini cevapla.
- 1-10. Veriler terimi daha geniş anlamda neyi belirtiyor?
- 1-11. a) yönergeler, b) sözün daha dar anlamında veriler, c) adresler nasıl bilgiler taşıyorlar?
- 1-12. Dijital cihazların temel yapı elemanları hangileridir?
- 1-13. Dijital cihazların ayrıldığı iki büyük grup hangileridir?
- 1-14. a) Birleşimsel b) ardışık ağların özellikleri nedir?
- 1-15. İkili toplayıcılar.....gerçekleştiği .....ağlardır.
- 1-16. Anahtarlamalı matrisler, anahtarlama elemanları.....olan .....matrislerdir.
- 1-17. Kodlayıcı.....grubuna aittir ve onun amacı.....
- 1-18. Kod çözücü..... grubuna aittir ve onun amacı.....
- 1-19. Çoğullayıcı .....grubuna aittir ve onun amacı.....
- 1-20. Çoğullama çözücü .....grubuna aittir ve onun amacı.....
- 1-21. Proglamlanabilir mantıksal yapılar, anahtarlamalı elemanları.....yapılar oluşturan ve ..... olanağı olan..... ağlardır.
- 1-22. Flip flop..... tanımlıyor çünkü onda.....
- 1-23. Yazmaçların ve sayaçların temel yapı elemanları nedir?
- 1-24. a) yazmaçların b)sayaçların temel amacı nedir?
- 1-25. Bellekler .....’de dijital bileşenler ve cihazlardır
- 1-26. Bellek .....’nın .....kümesi tanımlıyor.
- 1-27. Herhangi bir bellek konumun yeri ..... belirleniyor.
- 1-28. Her bellek konumun..... uzunluğu var ve onda..... yerleşiyor.
- 1-29. Belleğin kapasitesi.....’dir.
- 1-30. Analog-dijital dönüşüm devreleri..... gerçekleştiriyor.

- 1-31. Dijital-analog dönüşüm devreleri.....gerçekleştiriyor
- 1-32. Dijital cihazların analiz sorunu.....kapsıyor.
- 1-33. Dijital cihazların sentez (projeleme) sorunu.....kapsıyor.
- 1-34. Sayı sistemleri .....'nın sistemleridir.
- 1-35. Her ağırlıklı sayı sistemin kendi..... var, o da .....'dır.
- 1-36. Verilen sayı sistemlerin tabanları kaçtır ve rakamları hangileridir a) onlu; b) on altılı; c) sekizli; ç) ikili.
- 1-37. Sayı sistemin temeli b ve elimizde olan rakamların sayısı n verilmiş olsun. Bu sistemde kaç farklı sayı N yazılabilir?
- 1-39. (a) onlu; (b) on altılı; (c) sekizli; (ç) ikili sayı sistemi sözkonusu olursa ve bu arada elde  $n=4$  rakam olsa, şunları cevapla (1) kaç farklı sayı yazılabilir? (2) en küçük sayı hangisidir? (3) en büyük sayı hangisidir?
- 1-40. Sayıda her rakamın ağırlık değeri nasıl belirleniyor?
- 1-41. Herhangi bir sayıda n tam sayı ve m kesit-rasyonel yeri olan bir X sayının değeri hangi formüle göre hesaplanıyor? Verilen sayı hangi sayı sisteminde tanıtılmıştır?
- 1-42. (1) 7531; (2) 9862; (3) 41onlu sayıların (a) dokuza kadar tümleyen; (b) ona kadar tümleyen belirle
- 1-43. Verilen sayıların: (a)  $EE_{(16)}$ ; (b)  $F0_{(16)}$ ; (c)  $10_{(16)}$ ; (ç)  $CDA_{(16)}$ ; (d)  $10_{(8)}$ ; (e)  $100_{(8)}$ ; (f)  $77_{(8)}$ ; (g)  $1000_{(2)}$ ; (h)  $1111_{(2)}$ ; (i)  $1011_{(2)}$  değerlerini belirle yani onlu sayı sistemine dönüştür.
- 1-44. (a) 123; (b) 69; (c) 127; (ç) 128; (d) 255 onlu sayıları (1) ikili; (2) on altılı; (3) sekizli sayı sistemine dönüştür.
- 1-45. Sayı sistemleri arasında şu dönüşümleri yapın (a) onaltılı ve sekizli sistemden  $14_{(16)}$  ve  $57_{(8)}$  sayılarını ikili sisteme; (b) ikili sistemden  $1010111_{(2)}$  sayısını on altılı sisteme; (c) onaltılı sistemden  $24_{(16)}$  sekizli sisteme; sekizli sistemden  $346_{(8)}$  onaltılı sisteme.
- 1-46. Verilen ikili sayı çiftlerini topla (a) 1111 ve 1011; (b) 1011 ve 1011; (c) 10111011 ve 11110111.
- 1-47. 1101 ikili sayısını (a) 1110; (b) 1011; (c) 1101 sayılarla çarp.
- 1-48. İkili sayı sisteminde şu çıkarma işlemlerini yap (a) 1110 sayısında 1011 sayısını çıkar; (b) 10100000–dan 10001111 çıkar; (c) 10100001–den 10001111 çıkar
- 1-49. Verilen ikili sayı çiftleri arasında bölme işlemlerini yap (a) 1010 ve 100; (b) 10110110 ve 1011; (c) 10011110 ve 1100; (ç) 10000011 ve 1001
- 1-50. Verilen onlu sayıları (a) +37; (b) 0; (c) -37; (ç) -41 ve (d) -99, (1) ön işaretli tam sayılar (SM sistemi); (2) tek tümleyen gösterimli sayılar (DC sistemi, 1's) olarak tanımla
- Verinin 1 bayt (8 bit) uzunluğunda sözcük olarak bellekte korunduğunu ve birinci bitin ön işaret için olduğunu tahmin et.

- 1-51. Verilerin bir biti ön işaret için ayrılmış bayt şeklinde yazılmış olursa, aranan dönüşümleri yap (a) +128, +15, -1, -7 ve -127 onlu sayıları çift tümleyen gösterime (RC sistemi, 2's) ; (b) 01110011, 01011101, 11000101 ve 10111101 çift tümleyenli sayıları onlu sayılara dönüştür
- 1-52. Verilen işlemleri 2's-tümleyen gösterimde hesapla, ardından elde edilen sonuçların doğru olup olmadıklarını kontrol et (a)  $14 + 23$ ; (b)  $9 - 6$ ; (c)  $8 - 1$ ; (ç)  $5 - 7$ ; (d)  $14 - 35$ ; (e)  $-12 + 19$ ; (f)  $-48 - 5$ . Verilen bir biti ön işaret için ayrılmış baytlar şeklin yazılmış olduğunu tahmin ediyoruz.
- 1-53. Kodlama terimi altına ne tanımlanıyor?
- 1-54. Kod alfabesi nedir?
- 1-55. İkili kodların kod alfabesi nedir? Onun sembolleri hangileridir?
- 1-56. Kod sözcüğü nedir? Onun uzunluğu nasıl ifade ediliyor?
- 1-57. (a) düzgün ve (b) düzgün olmayan kodların özellikleri nedir?
- 1-58. Artıklı kodlar .....kod sözcükleri içeriyor.
- 1-59. Kod tablosu ne için kullanılır? Onun görünüşünü betimle.
- 1-60. Ağırlıklı kodlarda kod sözcüğünden her bit.....
- 1-61. Ardışık kodlarda ..... önemlidir.
- 1-62. İkili (BCD) kodların özelliği.....
- 1-63. (a) 18367; (b) 42509 onlu sayıların NBCD şeklini yaz.
- 1-64. Verilen NBCD sözcüklerle a) 10000011; b)10011100 hangi kodlar kodlanmıştır.
- 1-65. 132, 645 ve 7890 onlu sayıları (a) Grey kodunda; (b) Ayken kodunda; (c) 3\_fazlalık kodunda; (ç) BCD 5421 kodunda kodla.
- 1-66. Yedi-bölütlü ekran kodunun uygulanmasıyla kaç farklı sembol elde edilebilir? Onlardan hangileri kullanılıyor?
- 1-67. LED diyotlu ve ortak a)anodlu; b)katotlu ekran için, (1) 7; (2) 4; (3) 9 rakamların gösterilmesi gerekirse, abcdefg kod sözcüğü nasıl olacak?
- 1-68. abcdefg kod sözcüğünün (1) 1001111; (2) 1100011 şekli varsa, ortak (a) katotlu; (b) anotlu LED ekranında ne gösterilecek.
- 1-69. Alfasayısal kodların özellikleri nedir?
- 1-70. (a) KUrbaga; (b) UB40; (c) Çocuk verilerini, standart ASCII kodunda kodla. Her kod sözcüğünü on altılı gösterimde yaz.
- 1-71. Onlu belirtmede şu ASCII kod sözcükleri verilmiştir: (a) 66 73 84 79 71 61; (b) 83 71 79 80 74 69. Kodlanan verileri belirle.
- 1-72. (a) ağırlıklı ve ardışık; (b) düzgün ve düzgün olmayan; (c) artıklı ve artıklı olmayan kodlar arasında fark nedir. (1) 8421(NBCD) kodu; (2) ASCII kodu; (3) Yedi bölütlü kod hangilerine aittir.
- 1-73. Sözcük terimiyle..... belirtiliyor.
- 1-74. (a) 10101011; (b) 11001100; (c) 01010001; (ç) 00111001 verileri verilmiştir. Her veriyi ayrı olarak (1) SM; (2) DC; (3) RC; (4) 8421 NBCD sayı; (5) ASCII kodunda yazılmış veri olarak gözlenirse, bu verilerin (1) belirtik değerini; (2) örtük değerini belirle.





# 2.

## BOOLE CEBRİ

Bu konusal birimini inceledikten sonra

- ⊕ Boole cebrin aksiyomlarını, kanunlarını ve teoremlerini tanıyacaksınız;
- ⊕ Anahtarlama fonksiyonları cebirü, tablo ve grafik şekilde tanımlayabileceksiniz;
- ⊕ Anahtarlama fonksiyonlarının tanımlama şekilleri arasında geçiş ödevleri çözeceksiniz;
- ⊕ Dört değişkenli anahtarlama fonksiyonlarının analitik yoluyla ve Carnot kartları yöntemiyle minimizasyon ödevleri çözeceksiniz;
- ⊕ Standart mantıksal devrelerin sembollerini tanıyacaksınız, yaptıkları işlemlere göre ayıracaksınız ve mantıksal diyagramlarda uygulayacaksınız;
- ⊕ Mantıksal devrelerin analizi ve senteziyle ilgili daha basit ödevler çözeceksiniz;
- ⊕ Daha basit iki seviyelimantıksağ ağlar tanımlayabileceksiniz.





## 2.1. GİRİŞ

Sayıların ikili tanımlanması insanın ifade ediliş şekli olarak zor düşünebilir, çünkü biz onlu şekilde düşünmeye alıştık. Fakat, dijital cihazların temel oluşturucu parçaları, sadece iki durumda olabilme özeliğine sahip olan elektronik devrelerdir, öyle ki onların “doğal dili” ikili belirtmedir (gösterimdir). Bundan dolayı, dijital teknikte ikili sayı sistemi ve ikili sayılarla çalışan uygun cebir uygulanıyor.

*Boole cebri*, köklerini yeni matematik disiplini olarak meydana geldiği XIX. yüzyılın ortalarından çekiyor. Bu disiplinin kurucusu, adını da ona göre kazanan İngiliz matematikçi George Boole’muş. Bu cebirin biçimsel-mantıksal düşünme ve sonuçlandırmaya dayandığından dolayı, *mantıksal cebir* terimi de kullanılıyor. Bu kanunlar sadece doğru ya da yanlış olabilen ifadelere dayanıyor, yani sadece iki değeri olan ifadelere dayanıyor. Bu değerleri ilk olarak büyük Yunan felsefecisi Aristoteles yazmış. George Boole biçimsel-mantıksal neticelendirmenin kanunları cebir ilişkiler ve işlemler ile tanımlanmasını teklif etmiş Böylece biçimsel-mantıksal neticelendime sürecinin, basit şekilde niceliksel şekilde tanıtmayı ve teknik açıdan sadece iki durumu olan bileşenler uygulanmasıyla gerçekleşmesini ve otomatizasyonu sağladır. Anahtarlamalı elemanların ve mantıksal devrelerin bu şekilde davrandıklarından dolayı, bu cebir *anahtarlamalı cebir* olarak da adlandırılıyor.

## 2.2. AKSİYOMLAR VE MANTIKSAL İŞLEMLER

**Boole cebri** sadece iki farklı eleman içeren  $B$  ikili küme tanımlanan tümden gelimli matematiksel sistemidir. Bu iki değer için farklı literatürde farklı sembollere rastlanabilir, ancak biz „1” (*mantıksal bir*) ve „0” (*mantıksal sıfır*) sembollerini kullanacağız. Böylece  $B = \{1, 0\}$ . Buna göre, Boole cebirinde tüm sabitlerin ve değişkenlerin, 1 veya 0 değerlerinden sadece bir değeri olabilir ve bu yüzden onlara *mantıksal ya da anahtarlamalı değişkenler* de denir. Bağımsız değişkenler genelde İngilizce alfabesinden büyük harflerle belirtiliyor:  $A, B, C, D, E, \dots$  ya da  $X_0, X_1, X_2, X_3, \dots$ , bağımlı değişkenler ise, yani *mantıksal, anahtarlamalı ya da komutasyon fonksiyonları* ise  $Y, Y_0, Y_1, Y_2$  veya  $F, F_1, F_2$  olarak işaretleniyor.

$B$  kümesinde iki iç ikili işlem tanımlanıyor „+” ve „.”. Bu işlemler Hantigton aksiyomları ise üç aksiyom olarak bilinen aşağıdaki üç aksiyomu memnun ediyor:

**A.1.** İkili iç işlemler birbirine göre değişimli ve dağıtımlıdır ya da  $\{B\}$  kümesinden herhangi  $A, B, C$  değişkenler için şu geçerlidir:

$$A + B = B + A, A \cdot B = B \cdot A$$

$$A(B + C) = (AB) + (AC), A + (BC) = (A + B)(A + C)$$

**A.2.** İkili iç işlemlerin farklı neutral elemanları, 0 ve 1, vardır. Buna göre herhangi mantıksal değişken  $A$  için, 0 elemanı var öyle ki  $A + 0 = A$  geçerlidir ve eleman 1 var öyle ki  $A \cdot 1 = A$  geçerlidir.

**A.3.** Herhangi mantıksal değişken  $A$  için, tek ters değişkeni  $\bar{A}$  vardır, öyle ki  $A + \bar{A} = 1, A \cdot \bar{A} = 0$  geçerlidir.

Verilen aksiyomlardan doğrudan çıkan önemli bir özellik çiftlik (simetri) prensibidir. Buna göre tüm aksiyomlar çiftlerle gidiyor, hem de ayrıdan „+” işlemi için ve ayrıdan işlemi „.” için. Bu prensibe göre „+” işleminin „.” işlemiyle ve 1 elemanın 0 elemanı ile karşılıklı değişmesi mümkündür, öyle ki „+” işlemi için aksiyomlardan çekerek, „.” işlemi için çiftlik aksiyomları elde ediliyor ve tersi

Boole cebirinde üç temel (elementer) işlemler vardır: iki ya da fazla işlenen ile çalışan iki işlem: mantıksal toplama (+) ve mantıksal çarpma (·) ve bir işlenenle çalışan bir birli işlem: Mantıksal evirme (olumsuzlama) (̄).

Mantıksal toplama **VEYA** (İng. OR) işlemi ve mantıksal ayırtım olarak da adlandırılır, işleci ise „+” işareti dışında „ $\cup$ ” da olabilir. Mantıksal çarpma **VE** (İng. AND ya da mantıksal birleşim olarak da adlandırılıyor, işleci ise „.” işareti dışında, „ $\cap$ ” ve „&” olabilir. Mantıksal ifadelerin yazılması sırasında bu işleç genelde yazılmıyor. Mantıksal evirme aynı zamanda **DEĞİL** (İng. NOT) ya da **TÜMLEME** işlemi olarak da adlandırılıyor, „̄” işareti dışında „é” ya da „~” ile belirtilebilir.

Temel mantıksal işlemleri tab.2-1, tab.2-2 ve tab.2-3'te gösterilmiş şekilde tanımlanıyor.

VEYA (+)
$0 + 0 = 0$
$0 + 1 = 1$
$1 + 0 = 1$
$1 + 1 = 1$

Tab.2-1. VEYA

VE (·)
$0 \cdot 0 = 0$
$0 \cdot 1 = 0$
$1 \cdot 0 = 0$
$1 \cdot 1 = 1$

Tab.2-2.VE

DEĞİL (̄)
$\bar{0} = 1$
$\bar{1} = 0$

Tab.2-3.DEĞİL

Temel mantıksal işlemler

Tablolardan görüldüğü gibi VEYA (mantıksal toplama) işlemi için neutral eleman 0'dır, VE (mantıksal çarpma) işlemi için ise 1'dir. Buna göre şu sonuçlara varabiliriz:

1. İki işlenen toplanırsa, sadece aynı zamanada her iki işlenen 0 değerinde bulunduğu halde sonuç 0'dır, aksi takdirde sonuç 1 olacak, ya da en az bir 1 sonuç olarak 1 verir;
2. İki işlenen çarpılırsa, sonuç, sadece her iki işlene 1 değerinde olunca 1 olur, aksi halde 0 elde ediliyor, yani en az bir 0 sonuç olarak 0 verir;
3. Herhangi bir işlenenin değeri 0 değilse, o zaman değeri 1 olur ve tersi işlenenin değeri sıfır değilse, o zaman değeri 0'dır.

Verilen işlemlerden, en yüksek gerçekleşme seviyesi tümleşme (DEĞİL işlemi, olumsuzluk) işleminin vardır, ardından mantıksal çarpma (VE işlemi) ve en sonunda mantıksal toplama (VEYA işlemi) işlem geliyor. İşlemlerin gerçekleşme sırası parantezlerin kullanımıyla değişebilir.

Temel mantıksal işlemlerin birleşmesiyle başka, biraz daha karmaşık işlemler elde edilebilir: **OVE** ya da **Olumsuz VE**, (İng. NAND) çarpmanın tümleşmesiyle elde ediliyor (VE ve ardında DEĞİL) ve **OYA** ya da **Olumsuz YA** (İng. NOR) toplanmanın tümleşmesiyle elde ediliyor (VEYA ve ardında DEĞİL). Ayrıca şu işlemler de elde edilebilir: **Dışlayıcı VEYA** ya da **ayırcalıklı VEYA**, **D-YA** (İng. XOR), „ $\oplus$ ” işareti ile belirtiliyor ve **Dışlayıcı olumsuz VEYA**, yani **D-OYA** (İng. XNOR) OYA işleminin tümleyeni ile elde ediliyor (ODA, adından DEĞİL)

Yukarıda açıklanan tüm işlemler, tab. 2-4, tab. 2-5, tab. 2-6 ve tab. 2-7'de tanımlanmıştır.

OYA ( $\bar{+}$ )	OVE ( $\bar{\cdot}$ )	D-YA ( $\bar{\oplus}$ )	D-OYA ( $\bar{\oplus}$ )
$\overline{0+0} = 1$	$\overline{0 \cdot 0} = 1$	$0 \oplus 0 = 0$	$\overline{0 \oplus 0} = 1$
$\overline{0+1} = 0$	$\overline{0 \cdot 1} = 1$	$0 \oplus 1 = 1$	$\overline{0 \oplus 1} = 0$
$\overline{1+0} = 0$	$\overline{1 \cdot 0} = 1$	$1 \oplus 0 = 1$	$\overline{1 \oplus 0} = 0$
$\overline{1+1} = 0$	$\overline{1 \cdot 1} = 0$	$1 \oplus 1 = 0$	$\overline{1 \oplus 1} = 1$

Tab.2-4. OYA

Tab.2-5. OVE

Tab.2-6. D-YA

Tab.2-7. D-OYA

Karmaşık mantıksal işlemler

Tanımlamalardan şu sonuçlara varılabilir:

1. OYA işleminin sonucu, sadece her iki işlenen 1 olunca 1 olur, aksi halde 0 elde ediliyor;
2. OVE işleminin sonucu, sadece her iki işlenen 1 olunca 0 olur, tüm diğer durumlarda sonuç 1 olur;
3. D-YA işleminin sonucu işlenenler aynı değerde olursa 0 olur, yani iki 0 sonuç olarak 0 verir, ancak iki 1 de sonuç olarak 0 verir. İşlenenlerin ters değerleri varsa, sonuç olarak 1 elde ediliyor;
4. D-OYA işleminin sonucu D-YA işlemine göre evriktir. Bu işlem aslında işlenenlerin değerlerini kıyaslıyor, karşılaştırıyor. İki işlenen eşitse, o zaman sonuç 1 olur, farklıysa sonuç olarak 0 elde ediliyor.

### 2.3. TEOREMLER VE KANUNLAR

Hantigton aksiyomlarından, Boole cebirinde uygun uygulamaları olan farklı teoremler elde edilebilir. Biz sadece VE, VEYA ve DEĞİL işlemlerini içeren teoremlere dikkat vereceğiz. Bu teoremlerden bazıları Boole cebirin kanunlarını ifade ediyor, tümü beraber olarak ise mantıksal denklemlerin ve mantıksal ifadelerin çözülmesi ve basitleştirilmesi sırasında kurallar olarak kullanılıyor. Teoremler, çiftlilik prensibini uygulayarak simetrik çiftlerle verilmiştir. Teoremlerde yer alan tüm değişkenler mantıksal değişkenler olup, değerleri sadece 0 veya 1 olabilir. Buna göre herhangi bir değişkenin 1 değeri varsa, ya da eğer  $A = 1$  ise, o zaman  $\bar{A} = 0$  ve tersi: eğer  $A = 0$  ise, o zaman  $\bar{A} = 1$ .

Önce, sadece bir değişken içeren teoremleri sayacağız. Bu teoremler şunlardır:

$$\overline{\bar{A}} = A \quad (t. 2-1)$$

$$A + 0 = A \quad A \cdot 1 = A \quad (t. 2-2)$$

$$A + 1 = 1 \quad A \cdot 0 = 0 \quad (t. 2-3)$$

$$A + A = A \quad A \cdot A = A \quad (t. 2-4)$$

$$\bar{A} + \bar{A} = \bar{A} \quad \bar{A} \cdot \bar{A} = \bar{A} \quad (t. 2-5)$$

$$A + \bar{A} = 1 \quad A \cdot \bar{A} = 0 \quad (t. 2-6)$$

Birleşme, değişme ve dağılma kanunlarını ifade eden teoremler devamda verilmiştir:

$$A + (B + C) = (A + B) + C \quad A \cdot (B \cdot C) = (A \cdot B) \cdot C \quad (\text{t. 2-7})$$

$$A + B = B + A \quad A \cdot B = B \cdot A \quad (\text{t. 2-8})$$

$$A (B + C) = A B + A C \quad A + B \cdot C = (A + B) \cdot (A + C) \quad (\text{t. 2-9})$$

Teoremlerin isplanması üç aksiyoma dayanıyor, ancak bizim için bunun esas önemi yoktur. Bu yüzden örnekleme için sadece birini ispatlayacağız. Birleşme ve değişme kanunlarını ifade eden denklemleri ve dağılma kanununun ifade eden birinci denklemi, sezgisel olarak çok kolay anlıyoruz çünkü sıradan cebirde geçerli olan aynı kanunlara çok benzerdir. Ancak, dağılma kanununun ikinci denklemi biraz acayip görünüyor, be bizim anlamamıza uymuyor. Bu yüzden bu denklemi isptalyacağız, hem de iki yöntemle.

(1) Birinci ispatı analitik (cebirsal) şekilde aksiyomları ve yukarıda saydığımız teoremlerin uygulanmasıyla göstereceğiz. Bu arada denklemin sağ tarafından çekerek sol tarafını elde edeceğiz

$$\begin{aligned} (A + B)(A + C) &= AA + AC + AB + BC = A + AC + AB + BC = A + AB + AC + BC = \\ &= A(1 + B) + AC + BC = A + AC + BC = A(1 + C) + BC = A + BC \end{aligned}$$

(2) İkinci ispatlama şeklini *kusursuz tümevarım (endüksiyon) yönteminin* uygulanmasıyla gerçekleştireceğiz. Bu yöntemle göre, teorem o şekilde ispatlanıyor ki denklemin sol tarafının, değişkenlerin alabileceği tüm değerler kombinasyonları için, denklemin sağ tarafındaki ifadeyle aynı değeri olup olmadığı tespit ediliyor. Bu durumda üç değişkenin olduğundan dolayı, toplam  $2^3 = 8$  olası kombinasyon meydana gelebilir. Bu kombinasyonlardan her biri için, eşitlik işaretinin sol tarafındaki mantıksal ifadenin, yani  $(A+B)(A+C)$  ifadesinin değerini ve sağ tarafta bulunan ifadenin, yani  $(A + BC)$  ifadesinin değerini hesaplayacağız, elde edilen sonuçları ise tabloda yazacağız. Değişkenlerin her kombinasyonu için eşit sonuç elde edilmesi, teoremin ispatlanmış olduğu demektir. Tab.2-8 tablosundan  $(A+B)(A+C)$  ifadesinin değeri A, B ve C değişkenlerinin alabileceği her değer kombinasyonu için  $(A+BC)$  ifadesiyle aynı değeri olduğu görülüyor. Böylece ispat tamamlanmıştır.

A	B	C	$(A+B) \cdot (A+C)$	$(A+B \cdot C)$
0	0	0	0	0
0	0	1	0	0
0	1	0	0	0
0	1	1	1	1
1	0	0	1	1
1	0	1	1	1
1	1	0	1	1
1	1	1	1	1

Tab. 2-8. Kusursuz tümevarım yöntemi

Boole cebirinde De-Morgan kanunların ve teoremlerin özellikle önemli yeri vardır. Bu kanunlar şu biçimde yazılabilir:

$$\overline{A + B + C + \dots} = \bar{A} \cdot \bar{B} \cdot \bar{C} \dots, \quad \overline{A \cdot B \cdot C \dots} = \bar{A} + \bar{B} + \bar{C} + \dots \quad (\text{t. 2-10})$$

Verilen ifadelerden şu sonuçlara varabiliriz:

1. Birden fazla değişkenin mantıksal toplamının tümleyeni, her değişkenin ayrıdan tümleyenlerin mantıksal çarpımıyla değiştirilebilir ve

2. Birden fazla değişkenin mantıksal çarpımının tümleyeni, her değişkenin ayrıdan tümyenlerin mantıksal toplamıyla değiştirilebilir.

Şimdiye kadar saydığımız teoremler dışında, şu teoremler de önemlidir:

$$A + AB = A, \quad A(A + B) = A \quad (\text{t. 2-11})$$

$$A + \bar{A}B = A + B, \quad A(\bar{A} + B) = AB \quad (\text{t. 2-12})$$

$$AB + A\bar{B} = A, \quad (A + B)(A + \bar{B}) = A \quad (\text{t. 2-13})$$

$$AB + \bar{A}C = (A + C)(\bar{A}B), \quad (A + B)(\bar{A} + C) = AC + \bar{A}B \quad (\text{t. 2-14})$$

$$AB + \bar{A}C + BC = AB + \bar{A}C, \quad (A + B)(\bar{A} + C)(B + C) = (A + B)(\bar{A} + C) \quad (\text{t. 2-15})$$

$$AB + BC + \bar{B}C = AB + C, \quad (A + B)(B + C)(\bar{B} + C) = (A + B)C \quad (\text{t. 2-16})$$

(t. 2-11), (t. 2-12) ve (t. 2-13) teoremleri *soğurma teoremleri* olarak da biliniyor.

Sonunda *gelişme (açılım) teoremini* de belirteceğiz:

$$Y(A, B, C, \dots) = [A \cdot Y(1, B, C, \dots)] + [\bar{A} \cdot Y(0, B, C, \dots)] \quad (\text{t. 2-17})$$

Daha karmaşık mantıksal ifadelerin basitleştirilmesi ve çözülmesi sırasında, şimdiye kadar saydığımız tüm aksiomlar, kanunlar ve teoremler kullanılıyor. Birkaç örnek görelim:

$$\begin{aligned} \text{Örn. 1. } \quad & ABC + ABC\bar{C} + A\bar{B}C = A(BC + B\bar{C} + \bar{B}C) = A[B(C + \bar{C}) + \bar{B}C] = A(B + \bar{B}C) = \\ & = A(B + C) = AB + AC \end{aligned}$$

$$\begin{aligned} \text{Örn. 2. } \quad & (A + B)(A + \bar{B})(\bar{A} + C) = (AA + A\bar{B} + AB + B\bar{B})(\bar{A} + C) = (A + A\bar{B} + AB)(\bar{A} + C) = \\ & = [A(1 + \bar{B}) + AB](\bar{A} + C) = (A + A\bar{B})(\bar{A} + C) = A(1 + \bar{B})(\bar{A} + C) = AC \end{aligned}$$

$$\begin{aligned} \text{Örn. 3. } \quad & Y(A, B, C) = (A + B)[A(B + C) + AB + AC] = A(1 + B)[1(B + C) + 1B + 1C] + \\ & + A(0 + B)[0(B + C) + 0B + 0C] = A(B + C + B + C) + AB(0 + 0 + 0) = A(B + C) \end{aligned}$$

## 2.4. ANAHTARLAMALI FONKSİYONLAR VE ONLARIN GÖSTERİLMESİ

Değeri diğer mantıksal değişkenlere bağlı olan her mantıksal değişken mantıksal (anahtarlamalı) fonksiyon tanımlıyor. Anahtarlamalı fonksiyonlar üç şekilde gösteriliyor: **kombinasyonel tablolar (doğruluk tablosu)** kullanarak tablolü şekilde; mantıksal denklemler yarımıyla analitik (cebirsal) şekilde ve **mantıksal semboller** (standartlaşmış blok-diyagramlar) kullanarak grafiksel şekilde. Devamda fonksiyonların tablolü ve analitik gösterimi incelenecektir. Ayrıca, grafiksel gösterime de metnin devamında özel dikkat ve yer verilmiştir, çünkü grafiksel gösterimin anahtarlamalı fonksiyonların şematik gösterimine yönlendirdiğinden dolayı büyük önemi vardır.

### 2.4.1. TABLOLU GÖSTERİM

Tablolu (tabelar) gösterim sırasında ilk önce kombinasyon tablosu ya da doğruluk tablosu çiziliyor. Sol tarafta tüm bağımsız değişkenlerin isimleri yazılıyor, fonksiyonun ismi ya da fazla fonksiyon varsa, fonksiyonların isimleri ise tablonun sağ tarafında yazılıyor. Böylece ne kadar toplam bağımsız ve bağımlı değişken varsa, o kadar sütun elde ediliyor. Ondan sonra satırlarda değerlerin bağımsız değişkenlerin alabildiği tüm olası kombinasyonlar yazılıyor, en sonunda her kombinasyon için uygun sütunda her kombinasyon için fonksiyon değerleri giriliyor.

$n$  değişkenden bağlı fonksiyonun verildiğini tahmin edersek, o zaman kombinasyonlar tablosunda bağımsız değişkenler için  $n$  sütun ve fonksiyon için bir sütun olacak. Toplam  $N = 2^n$  olası giriş kombinasyon olduğundan dolayı, doğruluk tablosunda toplam  $N=2^n$  satırın olduğu açıkça görülüyor. Bu satır onlu şekilde uygun indislerle " $i$ " işaretleniyor, öyle ki birinci satıra 0 indisi eşlik ediyor, son satıra ise  $(N-1)$  indisi, yani  $(2^n-1)$  indisi eşlik ediyor. 2, 3 ve 4 değişkenli herhangi bir fonksiyonun kombinasyonel tabloların tab.2-9, tab.2-10 ve tab.2-11 gibi işaretlidir.

$i$	$AB$	$Y$
0	00	
1	01	
2	10	
3	11	

Tab.2-9. İki değişkenli fonksiyon

$i$	$ABC$	$Y$
0	000	
1	001	
2	010	
3	011	
4	100	
5	101	
6	110	
7	111	

Tab.2-10. Üç değişkenli fonksiyon

$i$	$ABCD$	$Y$
0	0000	
1	0001	
2	0010	
3	0011	
4	0100	
5	0101	
6	0110	
7	0111	
8	1000	
9	1001	
10	1010	
11	1011	
12	1100	
13	1101	
14	1110	
15	1111	

Tab.2-11. Dört değişkenli fonksiyon

Kombinasyonel fonksiyonların kombinasyonel tabloları

Elimizde  $n$  bağımsız değişken varsa, ondan elde edilen toplam fonksiyonlar sayısı  $N_F$  şu denklemlerle belirleniyor:

$$N_F = 2^{2^n} \quad (2-18)$$

### 2.4.2. ANALİTİK GÖSTERİM

Analitik yazılım şeklini geleneksel cebirden biliyoruz. Geleneksel cebire benzer olarak, Boole cebirinden de **mantıksal, boole** ya da **anahtarlamalı denklem** olarak adlandırılan belirli denklemler oluşuyor. Şöyle ki, „=“ eşitlik işaretinin sol tarafında fonksiyon (bağımlı değişken) yazılıyor, sağ tarafında ise mantıksal işlemler işaretleriyle bağlı bağımsız değişkenler yazılıyor. Genel olarak, her anahtarlamalı fonksiyon farklı şekillerde yazılabilir, öyle ki bazan daha basit, bazan da daha karmaşık biçim elde ediliyor. Biz mantıksal fonksiyonların normlaştırmış (standart, kanonik) yazılış biçimlerini tanıyacağız. Yapısı tam olarak belirlenmiş böyle biçimlere **normal biçimler (NB)** denir. Burda fonksiyonların, **ayırıcı normal biçimde (ANB-İng.DNF)** ve **bağlayıcı normal biçimde (BNB-İng.CNF)** tanıtımı söz konusudur.

ANB, fonksiyonu toplam şeklinde, yani bağımsız değişkenlerin ( $\Sigma p$ ) çarpımlarının ( $p$ ) toplamı ( $\Sigma$ ) olarak ifade ediyor. Parsiyel (kısmi) çarpımda tüm bağımsız değişkenlerin doğrudan ya da tümleyen şeklinde olup olmaması önemli olmadan yer alırsa, parsiyel çarpıma **miniterm** ( $m$ ) denir (temel çarpım, dolu ya da tam birleşme). ANB'nin toplamında giren tüm çarpımlar miniterm ise, o zaman **kusursuz ANB ( $\Sigma m$ ) (KANB)** söz konusudur. Diğer taraftan BNB, fonksiyonu toplamların çarpımı ( $\Pi$ ) olarak, yani bağımsız değişkenlerin ( $\Pi s$ ) toplamların ( $s$ ) çarpımı ( $\Pi$ ) olarak ifade ediyor. Parsiyel toplam, tüm bağımsız değişkenlerin toplamını tanımlarsa, parsiyel toplama maksterm ( $M$ ) (temel toplam, dolu ya da tam ayrılma) denir. Bu arada parsiyel toplamda yer alan bağımsız değişkenler doğrudan ya da tümleyen şekilde olabilir. BNB'nin çarpımında giren tüm toplamlar maksterm ise, o zaman **kusursuz BNB ( $\Pi M$ ) (KBNB)** elde ediliyor.

Daha iyi açıklamak için üç ve dört değişkene bağlı olan birkaç fonksiyon örneği inceleyeceğiz:  $Y = Y(A, B, C)$ ,  $Z = Z(D, G, H)$ ,  $F = F(X_1, X_2, X_3, X_4)$ . ANB'de altları çizilmiş ifadeler minitermlerdir ( $m$ ), BNF'de ise makstermlerdir ( $M$ ).

$$\begin{array}{ll} \text{KANB } Y = \underline{ABC} + \underline{\overline{A}B\overline{C}} + \underline{\overline{A}\overline{B}C} & \text{ANB } Y = \underline{\overline{A}B\overline{C}} + \underline{\overline{A}BC} + \underline{\overline{A}B\overline{C}} + \underline{\overline{B}C} \\ \text{BNB } Y = (A + \overline{B})(\overline{A} + \overline{B})(\overline{A} + C) & \text{KBNB } Y = (A + \overline{B} + C)(\overline{A} + B + \overline{C}) \\ \text{BNB } Z = (\underline{\overline{D} + \overline{G} + H})(G + \overline{H}) & \text{ANB } Z = \underline{\overline{D}GH} + \underline{D\overline{G}H} + \underline{GH} + \underline{\overline{D}} \\ \text{KANB } \underline{X_1 X_2 \overline{X_3} \overline{X_4}} + \underline{\overline{X_1} X_2 X_3 X_4} + \underline{\overline{X_1} \overline{X_2} X_3 \overline{X_4}} & \\ \text{KBNB } \underline{(X_1 + \overline{X_2} + X_3 + \overline{X_4})(X_1 + \overline{X_2} + \overline{X_3} + X_4)} & \end{array}$$

*Normal biçimler çok hızlı ve basit bir şekilde gerçekleştirilir, ancak bu biçimi kullanmamızın en önemli nedeni, bu biçimde fonksiyonların ikişer seviyeden elde edilmesidir.* Şöyle ki, ANB'de önce değişkenler mantıksal çarpılıyor, ardından tüm çarpımlar mantıksal olarak toplanıyor (VE-VE-YA). BNB'de terstir: birinci seviyede değişkenler mantıksal toplanıyor, ikinci seviyede ise sonuçlar mantıksal çarpılıyor (VEYA-VE). Devamda göreceğimiz gibi, bu çok önemli özelliktir.

Ancak, normal biçimler için, onların genel durumda *artıklı biçimler* olduklarını bilmemiz gerekiyor, yani mantıksal fonksiyonu en basit ve en kısa şekilde gösterecek biçimlerdir, çünkü aynı fonksiyonun tanımlanması için en az sayıda ifadelerden daha fazla ifade içeriyorlar. En az sayıda ifadedeki normal biçimler (toplamlar ve çarpımlar) ve aynı zamanda bu ifadelerin herbiri en az değişken içeren biçimlere minimum normal biçimler olarak tanımlanıyor: **MANB ve MBNB**. Bu biçimler mantıksal fonksiyonu en kısa ve en basit şekilde göstermelidir.



### 2.4.2.1. TAMAMEN VERİLMİŞ FONKSİYONLAR

KANB ve KBNB, indisler kümesi aracılığıyla genelde daha basit onlu gösterimli analitik şekilde belirtiliyorlar. Şöyle ki, temel ifadeler yerine (çarpımlar ve toplamlar) minterm işareti:  $m$  ve maksterm işareti kullanılıyor. Bunların dışında uygun indis  $i_{ij}$  ya da  $i_{ok}$  de yazılıyor. Minterm yanındaki indis,  $i_{ij}$  fonksiyonun değeri 1 olduğu satırların sıra numarasına uyar, maksterm yanındaki indis  $i_{ok}$  ise fonksiyonun değeri 0 olan satırların sıra numarasına uygundur. Böylece,  $j + k = N$  her zaman geçerlidir ( $N = 2^n$ ). Buna göre, her iki indis 0 ile  $N-1$  arasındaki kapsamına aittir:  $[0, 1, 2, \dots, 2^n - 1]$ . Böylece fonksiyonun bu KANB'İ mintermlerin toplamı olarak yazılıyor:  $Y = m_{i_{11}} + m_{i_{12}} + \dots + m_{i_{1j}}$ , fonksiyonun KBNB'i ise makstermlerin çarpımı olarak gösteriliyor:  $Y = M_{i_{01}} + M_{i_{02}} + \dots + M_{i_{0k}}$ . İndisler kümesi aracılığıyla daha kısa tanımlama şu şekilde daha sıkça kullanılıyor. KANB için şunu elde ediyoruz  $Y = \sum m(i_{i_1, i_{12}, \dots, i_{1j}})$  ya da  $f^{(1)} = (i_{11}, i_{12}, \dots, i_{1j})$ , KBNB için ise  $Y = \prod M(i_{i_{01}, i_{02}, \dots, i_{0k}})$  ya da  $f^{(0)} = (i_{01}, i_{02}, \dots, i_{0k})$  elde ediliyor. KANB'de bulunmayan indisler KBNB'de bulunacak ve tersi, çünkü fonksiyonun değeri 1 değilse, değeri 0 olacaktır.

Üç değişkenli herangi bir fonksiyonun kombinasyon tablosu tab. 2-12 olarak verilmiştir. Bu arada tüm mintermler, yani makstermler ayrı sütunlarda işaretlenmiştir.

i	ABC	Y	$m_i$	Mintermler	$M_i$	Makstermler
0	000		$m_0$	$\bar{A} \cdot \bar{B} \cdot \bar{C}$	$M_0$	$A + B + C$
1	001		$m_1$	$\bar{A} \cdot \bar{B} \cdot C$	$M_1$	$A + B + \bar{C}$
2	010		$m_2$	$\bar{A} \cdot B \cdot \bar{C}$	$M_2$	$A + \bar{B} + C$
3	011		$m_3$	$\bar{A} \cdot B \cdot C$	$M_3$	$A + \bar{B} + \bar{C}$
4	100		$m_4$	$A \cdot \bar{B} \cdot \bar{C}$	$M_4$	$\bar{A} + B + C$
5	101		$m_5$	$A \cdot \bar{B} \cdot C$	$M_5$	$\bar{A} + B + \bar{C}$
6	110		$m_6$	$A \cdot B \cdot \bar{C}$	$M_6$	$\bar{A} + \bar{B} + C$
7	111		$m_7$	$A \cdot B \cdot C$	$M_7$	$\bar{A} + \bar{B} + \bar{C}$

Tab. 2-12. Üç değişkenli fonksiyonun mintermleri ve makstermleri

Tablolardan görüldüğü gibi her maksterm uygun mintermin tümleyen değeridir ya da her  $i = (0, 1, 2, \dots, 2^n - 1)$  şu eşitlik eşittir:

$$M_i = \overline{m_i} \quad (2-19)$$

Tab.2-13 olarak belirtilen sıradaki kombinasyon tablosunda, üç aynı değişkene A, B ve C, bağlı olan üç farklı fonksiyon verilmiş:  $Y = Y(A, B, C)$ ,  $Z = Z(A, B, C)$ ,  $W = W(A, B, C)$ . Verilmiş fonksiyonlardan indis kümesiyle tanımlanmış bazıları için farklı normal biçimler verilmiştir.



$i$	$ABC$	$Y$	$Z$	$W$
0	000	1	0	1
1	001	0	0	1
2	010	1	0	0
3	011	0	0	1
4	100	0	1	0
5	101	1	1	0
6	110	0	0	1
7	111	1	1	1

Tab. 2-13 Üç değişkenli Y, Z ve W fonksiyonların kombinasyon tabloları

$$\text{KBNB: } Y = \prod M(1,3,4,6) = (A+B+\bar{C})(A+\bar{B}+\bar{C})(\bar{A}+B+C)(\bar{A}+\bar{B}+C)$$

$$\text{KANB: } Y = \sum m(0,2,5,7) = (\bar{A}\bar{B}\bar{C})(\bar{A}B\bar{C}) + (\bar{A}B\bar{C}) + (ABC)$$

$$\text{KANB: } W = \sum m(0,1,3,6,7) = (\bar{A}\bar{B}\bar{C}) + (\bar{A}\bar{B}C) + (\bar{A}BC) + (ABC) + (ABC)$$

$$\text{KBNB: } Z = \prod M(0,1,2,3,6) = (A+B+C)(A+B+\bar{C})(A+\bar{B}+C)(A+\bar{B}+\bar{C})(\bar{A}+\bar{B}+C)$$

#### 2.4.2.2. KISMEN VERİLMİŞ FONKSİYONLAR

Şimdiye kadar tüm mantıksal fonksiyonları, bağımsız değişkenlerin her kombinasyonu için fonksiyonun değeriyle veriliyordu. Bu arada fonksiyonun 0 veya 1 değeri vardı.

$i$	$ABCD$	$F$
0	0000	x
1	0001	1
2	0010	1
3	0011	x
4	0100	x
5	0101	x
6	0110	0
7	0111	0
8	1000	1
9	1001	0
10	1010	1
11	1011	1
12	1100	0
13	1101	0
14	1110	0
15	1111	x

Ancak, pratikte sıkça *komple olmayan (kısmen) verilmiş (tanımlanmış) fonksiyonlara* rastlanabilir. Bu durumun meydana gelmesinin iki farklı nedeni var, ancak pratikte aynı bir nedene özetlenebilir. İlk olarak, bazan fonksiyonun, giriş değişkenlerinin bir ya da fazla kombinasyon için hangi değeri olacağı önemli değil. Diğer taraftan, bağımsız değişkenlerin bazı kombinasyonları hiçbir zaman ortaya çıkmayabilir.

Her iki durumda, belirli giriş kombinasyonları için fonksiyonun hangi değeri olacağı önemli olmadığı alınabilir. Fonksiyonun bu değerleri “önemsiz” (İng. “don’t care”) olarak adlandırılıyor ve farklı kitaplarda farklı sembollerle işaretleniyor, örneğin: „/” ; „\” ; „-” ; „b” ; „x” veya „X”. Biz devamda „x” sembolünü kullanacağız.

Dört değişkenli, kısmen verilmiş  $F = F(D, C, B, A)$  fonksiyonun doğruluk tablosu tab.2-14’te tanımlanmıştır.

Tab. 2-14. Kısmen tanımlanmış  $F(D, C, B, A)$  fonksiyonun kombinasyon tablosu

Onun indis kümesiyle KANB ve KBNB şeklinde tanımlanması şöyle olacaktır:

$$F = \sum m(1,2,8,10,11) + \sum_{xm} x(0,3,4,5,15) \text{ и } F = \prod M(6,7,9,12,13,14) \prod_{xM} x(0,3,4,5,15).$$

### 2.4.3. BİR ŞEKİLDEN BAŞKA ŞEKİLE DÖNÜŞÜM

Fonksiyonun doğruluk tablosuyla veya analitik şekilde tanıtıldığı önemli olmadan, bir şekilden öteki şekile dönüşüm nispeten kolaydır.

Fonksiyon tablosu verilmiş olduğu zaman, onun analitik tanıtımın her iki biçimi elde edilebilir: Kusursuz ayırıcı normal biçiminde (KANB) ve kusursuz bağlayıcı normal biçiminde (KBNB). KANB tabloda fonksiyonun değeri 1 olduğu kaç satır varsa, o kadar minterm toplamı yazılarak elde ediliyor. Devanda verilmiş olan örnekten görüldüğü gibi, verilen satırda değişkenlerin değeri 1 ise, mintermlerde bağımsız değişkenler doğrudan (nominal, tümlenmemiş) şekilde meydana geliyor, değişkenlerin değeri sıfır ise o zaman bağımsız değişkenler tümlenmiş şekilde meydana geliyor. Örnekte  $Y(A,B,C)$  ve  $Z(A,B,C)$  fonksiyonların kombinasyon tabloları, tab.2-15 a) ve b) – de verilmiş. Buradan onların KANB şekli elde ediliyor:

$$Y = \sum m(0,3,7) = (\overline{A}\overline{B}\overline{C}) + (\overline{A}BC) + (ABC)$$

$$Z = \sum m(1,2,3,4) = (\overline{A}\overline{B}C) + (\overline{A}B\overline{C}) + (\overline{A}BC) + (A\overline{B}\overline{C})$$

$i$	$ABC$	$Y$
0	000	1
1	001	0
2	010	0
3	011	1
4	100	0
5	101	0
6	110	0
7	111	1

a)  $Y = \text{Sm}(0,3,7)$

$i$	$ABC$	$Z$
0	000	0
1	001	1
2	010	1
3	011	1
4	100	1
5	101	0
6	110	0
7	111	0

b)  $Z = \text{Sm}(1,2,3,4)$

Tab. 2-15. Üç değişkenli anahtarlama fonksiyonların kombinasyon tabloları

KBNB, tabloda fonksiyonun değeri 0 olduğu kaç satır varsa, o kadar maksterm çarpımı yazılarak elde ediliyor. Bu durumda makstermde uygun satırda değeri 1 olan değişken tümleşiyor, değeri 0 olan değişken ise doğrudan olarak yazılıyor. KBNB şeklinin elde edilmesi, önceki örnekten, tab.2-15 ile verilmiş,  $Y$  ve  $Z$  fonksiyonlarıyla gösterilmiştir:

$$Y = \prod M(1,2,4,5,6) = (A+B+\overline{C})(A+\overline{B}+C)(\overline{A}+B+C)(\overline{A}+B+\overline{C})(\overline{A}+\overline{B}+C)$$

$$Z = \prod M(0,5,6,7) = (A+B+C)(\overline{A}+B+\overline{C})(\overline{A}+\overline{B}+C)(\overline{A}+\overline{B}+\overline{C})$$

Daha az sayıda minterm, ya da maksterm veren şeklin kullanılması doğrudur, çünkü sonraki basitleştirilmesi daha uygundur.

Fonksiyon analitik şekilde, mantıksal denkleme verilmiş olduğu zaman, tablolu gösterim şekline dönüşüm şu şekilde yapılıyor. Önce kombinasyon tablosu çiziliyor ve sütunlarda bağımsız değişkenler ve fonksiyon belirtiliyor, ardından bağımsız değişkenlerin tüm olası kombinasyonları yazılıyor. Ondan sonra, verilen denklemde, sırasıyla her giriş kombinasyonu değiştiriliyor ve fonksiyonun değeri hesaplanıyor. Bu değer, tabloda fonksiyon sütununun uygun satırında yazılıyor.

Analitik şekilden tablolu şekile dönüşümle ilgili bir örnek inceleyelim. Üç değişkenden, A, B ve C, bağlı olan W fonksiyonu verilmiştir:  $W = \overline{A}BC + ABC + \overline{A}B \cdot ABC = 0$  giriş kombinasyonun başlayarak, ABC = 111 giriş kombinasyonuna kadar her giriş kombinasyonunun değeri aşağıda hesaplanıyor:

Bu fonksiyonun doğruluk tablosu tab. 2-16 ile verilmiştir.

<i>i</i>	<i>ABC</i>	<i>W</i>
0	000	0
1	001	0
2	010	0
3	011	0
4	100	1
5	101	1
6	110	0
7	111	1

A = 0, B = 0, C = 0 olunca, W=010+000+01=0;

A = 0, B = 0, C = 1 olunca, W=011+001+01=0;

A = 0, B = 1, C = 0 olunca, W=000+010+00=0;

A = 0, B = 1, C = 1 olunca, W=001+011+00=0;

A = 1, B = 0, C = 0 olunca, W=110+100+11=1;

A = 1, B = 0, C = 1 olunca, W=111+101+11=1;

A = 1, B = 1, C = 0 olunca, W= 100+110+10=0;

A = 1, B = 1, C = 1 olunca, W=101+111+10=1.

Tab. 2-16. Üç değişkenli W(A,B,C) anahtarlamalı fonksiyonun kombinasyon tablosu

Bir analitik şekilde öteki analit şekile dönüşüm farklı yöntemlerle yapılabilir. Uygulanacak yöntem, fonksiyonun son şekli olması gereken, başlangıç (verilen) şekile bağlıdır.

NB'den SNB'ye nasıl dönüşeceğine ilişkin iki örnek vereceğiz.  $Z = Z(A,B,C)$  fonksiyon şu iki şekilde verilmiş olsun:  $Y = \overline{A}B + C$ ,  $Z = (A + \overline{B} + C)B$ . Birinci fonksiyon ANB'de verilmiş ve ondan KANB elde edilmesi gerekiyor, BNB şeklinde verilen ikinci fonksiyondan ise KBNB elde edilmesi gerekiyor.

Örn. 1. 
$$Y = \overline{A}B + C = \overline{A}B1 + 11C = \overline{A}B(C + \overline{C}) + 1(B + \overline{B})C = \overline{A}BC + \overline{A}B\overline{C} + 1(BC + \overline{B}C) = \overline{A}BC + \overline{A}B\overline{C} + (A + \overline{A})(BC + \overline{B}C) = \overline{A}BC + \overline{A}B\overline{C} + ABC + \overline{A}BC + \overline{A}BC + \overline{A}B\overline{C}$$

Örn. 2. 
$$Z = (A + \overline{B} + C)B = (A + \overline{B} + C)(0 + B + 0) = (A + \overline{B} + C)(A\overline{A} + B + 0) = (A + \overline{B} + C)[(A + B)(\overline{A} + B) + 0] = (A + \overline{B} + C)[(A + B)(\overline{A} + B) + C\overline{C}] = (A + \overline{B} + C)\{[(A + B)(\overline{A} + B) + C][(A + B)(\overline{A} + B) + C]\} = (A + \overline{B} + C)(A + B + C)(\overline{A} + B + C)(A + B + \overline{C})(\overline{A} + B + \overline{C})$$

Diğer yönde, KNB'den NB'e dönüşüm, aslında verilen fonksiyonun belirli basitleştirilmesidir ve önceden saydığımız teoremlerin uygulanmasıyla yapılabilir. Devamdaki örnekler, KANB şeklinde verilmiş  $U(X, Y, Z)$  fonksiyonunun ve KBNB şeklinde verilmiş  $V = V(X, Y, Z)$  fonksiyonunun nasıl basitleştirilebileceğini gösteriyor.

$$\begin{aligned} \text{Örn. 3. } U(X, Y, Z) &= X\bar{Y}Z + X\bar{Y}\bar{Z} + XYZ + \bar{X}YZ + \bar{X}\bar{Y}Z = X\bar{Y}(Z + \bar{Z}) + YZ(X + \bar{X}) + \bar{X}\bar{Y}Z = \\ &= X\bar{Y} + YZ + \bar{X}\bar{Y}Z = X\bar{Y} + Z(Y + \bar{X}\bar{Y}) = X\bar{Y} + Z(\bar{X}Y) \end{aligned}$$

$$\begin{aligned} \text{Örn. 4. } V(X, Y, Z) &= (X + \bar{Y} + Z)(X + Y + Z)(X + Y + \bar{Z})(\bar{X} + Y + \bar{Z}) = [(X + Z) + Y\bar{Y}] \\ &[(Y + \bar{Z})X + \bar{X}] = (X + Z)(Y + \bar{Z}) \end{aligned}$$

Örneklerden, elde edilen sonuçların minimal normal biçimde (MNB) oldukları açıkça görülmüyor. Bununla anahtarlamalı fonksiyonların minimizasyon sorununa giriliyor. Bu karmaşık bir sorundur ve devamda daha detaylı bir şekilde açıklanacaktır.

Bir NB'den diğer NB'e geçiş dağılım kanununun uygulanmasıyla gerçekleştirilebilir. Ancak bu yöntem çok zor olabilir ve bu yüzden indis kümesinin üzerinden geçiş yöntemini kullanacağız.

ANB'den BNB'e ya da ters yönde geçiş, sırayla, KNB'den geçerek yapılıyor. Böylece, BNB'de verilmiş fonksiyon, önce analitik KBNB şekline genişleniyor, ardından bu KBNB şekli indis kümesi biçiminde yazılıyor. Ondan sonra indis kümesinin yardımıyla fonksiyonun KANB elde ediliyor. Fonksiyon KANB'de analitik şekline yazılarak, uygun teoremlerin uygulanmasıyla basitleştiriliyor. ANB biçiminden BNB biçimine geçiş için ters süreç uygulanıyor, yani ANB'den KANB'ne geçişle başlanıyor, ardından KBNB şekli belirleniyor ve sonunda BNB elde ediliyor. Devamda her iki durum için birer örnek verilmiştir.

$$\begin{aligned} \text{Örn. 5. } F_1(A, B, C) &= A + B\bar{C} + \bar{A}\bar{B}C = A11 + 1B\bar{C} + \bar{A}\bar{B}C = A(B + \bar{B})(C + \bar{C}) + \\ &+ (A + \bar{A})B\bar{C} + \bar{A}\bar{B}C = (AB + \bar{A}\bar{B})(C + \bar{C}) + AB\bar{C} + \bar{A}\bar{B}C + \bar{A}\bar{B}C = ABC + AB\bar{C} + \\ &+ \bar{A}\bar{B}C + \bar{A}\bar{B}\bar{C} + ABC + \bar{A}\bar{B}\bar{C} + \bar{A}\bar{B}C = \sum m(7, 6, 5, 4, 6, 2, 1) = \sum m(1, 2, 4, 5, 6, 7) \end{aligned}$$

$$\begin{aligned} \text{Örn. 6. } F_2(A, B, C) &= \prod M(0, 3) = (A + B + C)(A + \bar{B} + \bar{C}) = A + (B + C)(\bar{B} + \bar{C}) = \\ &= A + B\bar{B} + B\bar{C} + \bar{B}C + C\bar{C} = A + B\bar{C} + \bar{B}C \end{aligned}$$

$$\begin{aligned} \text{Örn. 7. } F_3(A, B, C) &= A(\bar{B} + C)(A + B + \bar{C}) = (A + 0 + 0)(0 + \bar{B} + C)(A + B + \bar{C}) = \\ &= (A + B\bar{B} + C\bar{C}) + (A\bar{A} + \bar{B} + C)(A + B + \bar{C}) = [(A + B)(A + \bar{B}) + C\bar{C}](A + \bar{B} + C) \\ &(\bar{A} + \bar{B} + C)(A + B + \bar{C}) = (A + B + C)(A + \bar{B} + \bar{C})(A + \bar{B} + C)(\bar{A} + \bar{B} + C)(A + B + \bar{C}) = \\ &= \prod M(0, 3, 2, 6, 1) = \prod M(0, 1, 2, 3, 6) \end{aligned}$$

$$\text{Örn. 8. } F_4(A, B, C) = \sum m(4, 5, 7) = \bar{A}\bar{B}\bar{C} + \bar{A}\bar{B}C + ABC = \bar{A}\bar{B}1 + ABC = \bar{A}\bar{B} + ABC$$

## 2.5. STANDART MANTIKSAL FONKSİYONLAR

Temel mantıksal işlemleri: VE, VEYA ve DEĞİL (tümleme, evirme), OVE ve OYA işlemlerini gerçekleştiren fonksiyonlar, ile D-YA ve D-OYA, büyük önem taşıyorlar ve bu yüzden tab. 2-17 a), b), c), ç), d), e), f)-de bir kez daha onların doğruluk tabloları ve analitik şekilleri verilmiş.

A	B	$A + B$
0	0	0
0	1	1
1	0	1
1	1	1

a) VEYA

A	B	$A \cdot B$
0	0	0
0	1	0
1	0	0
1	1	1

b) VE

A	$\bar{A}$
0	1
1	0

c) DEĞİL

A	B	$\overline{A + B}$
0	0	1
0	1	0
1	0	0
1	1	0

ç) OYA

A	B	$\overline{A \cdot B}$
0	0	1
0	1	1
1	0	1
1	1	0

d) OVE

A	B	$A \oplus B$
0	0	0
0	1	1
1	0	1
1	1	0

e) D-YA

A	B	$\overline{A \oplus B}$
0	0	1
0	1	0
1	0	0
1	1	1

f) D-OYA

Tab. 2-17. Standart mantıksal fonksiyonlar

D-YA ve D-OYA fonksiyonların tablolarına bakarsak, bu iki mantıksal fonksiyonun, eşitsizlik ya da eşitlik tespit etmek için kullanılabileceklerini sonuca varabiliriz. Şöyle ki, D-YA fonksiyonun, sadece A ve B değişkenleri birbirinden farklı olduğu zaman, değeri 1 olur, yani A=0 ve B=1 ya da A=1, B=0 olduğu zaman. A ve B aynı (eşit) oldukları zaman D-YA fonksiyonu sonuç olarak 0 veriyor. Diğer taraftan, D-OYA fonksiyonu bundan ters çalışıyor, çünkü sadece A ve B değişkenleri birbirine eşitse, yani A=0 ve B=0 ya da A=1 ve B=1 olunca, D-OYA fonksiyonun değeri 1 olur. A ve B farklıysa, D-OYA'nın değeri 0'dır. Bunun dışında, D-YA fonksiyonun tablosundan, bu fonksiyonun ikili sayı sisteminde aritmetik toplama için kullanılabileceği görülebilir, çünkü birer birer bitin toplama kurallarını yerine getiriyor. Temel mantıksal fonksiyonları, yani VEYA, VE ve DEĞİL (tümleme) fonksiyonları uygulayarak, D-YA ve D-OYA fonksiyonları mantıksal denklemlerle analitik şekilde tanımlanabilir.

$$Y_{D-YA} = A \oplus B = A \cdot \bar{B} + \bar{A} \cdot B \quad (2-20)$$

$$Y_{D-OYA} = \overline{A \oplus B} = A \cdot B + \bar{A} \cdot \bar{B} \quad (2-21)$$

*Kombinasyonlarıyla, herhangi bir karmaşık fonksiyonun elde edilebileceği anahtarlamalı fonksiyonların kümesine işlevsel olarak tam mantıksal fonksiyonlar sistemi denir.*

Böyle bir sistem, örneğin VE, VEYA ve DEĞİL temel fonksiyonlar kümesidir, çünkü onlarla herhangi karmaşık anahtarlama fonksiyon ifade edilebilir. Devamdaki bölümde DEĞİL, VE ve VEYA fonksiyonlarının O-VE fonksiyonun yardımıyla nasıl ifade edilebileceği gösterilmiştir:

$$\bar{A} = \bar{A} + \bar{A} = (\overline{A \cdot A}) \quad (2-22)$$

$$AB = (\overline{\overline{A \cdot B}}) \quad (2-23)$$

$$A + B = (\overline{\overline{A + B}}) = (\overline{\overline{A \cdot B}}) \quad (2-24)$$

Benzer şekilde, temel fonksiyonların sadece O-YA fonksiyonunla ifade edilebileceğini gösterceğiz:

$$\bar{A} = \bar{A} \cdot \bar{A} = (\overline{A + A}) \quad (2-25)$$

$$AB = (\overline{\overline{A \cdot B}}) = (\overline{\overline{A + B}}) \quad (2-26)$$

$$A + B = (\overline{\overline{A + B}}) \quad (2-27)$$

Buna göre, her karmaşık anahtarlama fonksiyon sadece OVE fonksiyonun uygulanmasıyla ya da sadece OYA fonksiyonun uygulanmasıyla gerçekleştirilebilir. Demek ki, hem OVE fonksiyonu da, yani DEĞİL ve VE fonksiyonları da işlevsel açıdan komple sistem oluşturuyorlar. Aynı OYA fonksiyonu için de geçerlidir, yani DEĞİL ve VEYA fonksiyonları için de geçerlidir. İşlevsel açıdan komple sistem oluşturan fonksiyonlara **genel (evrensel) fonksiyonlar** denir.

*Sonunda çok önemli bir sonuç çıkarabiliriz: genel mantıksal fonksiyonların: VE, VEYA ve DEĞİL ya da sadece OVE ya da sadece OYA fonksiyonların uygulanmasıyla herhangi bir anahtarlama fonksiyon ifade edilebilir.* Bu çıkarım, pratikte farklı mantıksal fonksiyonlar gerçekleştiren elektronik elemanların ve birleşenlerin üretiminde önemli rol oynadı.

## 2.6. ANAHTARLAMALI FONKSİYONLARIN MİNİMİZASYONU

Aynı bir anahtarlama fonksiyonun farklı *normal biçimleri* (NB) olduğunu görmüştük. Bu NB eşit sayıda değişkenler ve işlemler içermiyor ve doğal olarak en az sayıda elemanlar (toplamlar ya da çarpımlar) ve eleman başına en az sayıda değişken içeren NB seçiliyor. Verilen fonksiyonun minimum biçime getiren sürece anahtarlama fonksiyonların minimizasyonu denir. Tabii ki, minimizasyon sonucu olarak verilen fonksiyonun MANB, ya da MBNB şekli elde edilmelidir.

*Minimizasyonun yapıldığı bir yöntem, cebir dönüşümlerin doğrudan uygulanmasıyla analitik yöntemidir.* Bu arada mantıksal ifadelerin basitleştirilebilen Boole cebri kuralları kullanılıyor. Anahtarlama fonksiyonların böyle minimizasyon yöntemi sırasında aslında bir analitik şekilde başka analitik şekile geçiliyor. Minimizasyon sürecinde fonksiyonun genelde SNB olan bir NB'den çekerek, MNB'in elde edilmesi gerekiyor. Ancak, bu yöntem çok karmaşıktır ve anahtarlama fonksiyonun minimum şekline getireceğine güvenli yol yoktur.

Analitik minimizasyon dışında mantıksal fonksiyonların minimizasyonu için başka yöntemler de var. Burada fonksiyonları grafiksel yoluyla minimize eden Karno kartları yöntemini ancağız. Bu yöntem en sıkça olarak en çok beş değişkene bağlı olan fonksiyonların minimizasyonu için kullanılıyor, genelde ise üç ya da dört değişkenli fonksiyonlar için kullanılıyor.

Belirli sayıda değişkenlere bağlı olan ve minimizasyonun uygulanması gereken fonksiyonlar için *Kvayn Mek Klaski (Quine McCluskey) yöntemi* denilen başka bir tablolü yöntem uygulanıyor. Bu yöntem daha büyük sayıda değişkenli fonksiyonların minimizasyonu sırasında kullanılıyor ve özellikle minimizasyon sürecini, bilgisayar kullanarak otomatikleştirmek istediğimiz zaman uygundur.

### 2.6.1. ANALİTİK MİNİMİZASYON YÖNTEMİ

*Anahtalamalı fonksiyonların analitik (cebirsal) minimizasyon yöntemi aslında verilen mantıksal ifadenin basitleştirilmesi demektir.* Basitleştirmek hakkında önceki bölümlerde bahsedilmişti. Bununla ilgili olarak, mantıksal fonksiyonların minimizasyon sorununu en iyi olarak, daha birkaç örneğin incelenmesiyle anlayacağız. Sadece, bu yöntemin uygulanması sırasında en önemli Boole cebri teoremlerini bilmemiz gerektiğini hatırlayalım. Bu teoremler burada minimizasyon sırasında basitleştirme kuralları olarak kullanılıyor.

$$\begin{aligned}
 \text{Örn. 1. } F_1(A, B, C, D) &= \overline{A+B+C+D} + ABC + \overline{A}BD + A\overline{C} + ABC + AB\overline{D} + \overline{A}C = \\
 &= \overline{A}B\overline{C}\overline{D} + \overline{A}B\overline{D} + \overline{A}C + ABC + AB\overline{D} + \overline{A}C = \overline{A}(\overline{B}\overline{C}\overline{D} + C) + \overline{A}B\overline{D} + \\
 &+ A(BC + \overline{C}) + AB\overline{D} = \overline{A}B\overline{D} + \overline{A}B\overline{D} + \overline{A}C + \overline{A}B + A\overline{C} + AB\overline{D} = \overline{A}B(D + \overline{D}) + \\
 &+ \overline{A}C + AB(1 + \overline{D}) + A\overline{C} = \overline{A}B + \overline{A}C + AB + A\overline{C} = \overline{A}(\overline{B} + C) + A(B + \overline{C})
 \end{aligned}$$

$$\begin{aligned}
 \text{Örn. 2. } F_2(A, B, C) &= AB + \overline{A}C + BC = AB + \overline{A}C + (A + \overline{A})BC = AB(1 + C) + \overline{A}C(1 + B) = \\
 &= AB + \overline{A}C
 \end{aligned}$$

İkinci örnekteki fonksiyon  $F_2(A, B, C, D)$  genişleme teoremiyle de minimize edilebilir:

$$\begin{aligned}
 \text{Örn. 3. } F_2(A, B, C) &= AB + \overline{A}C + BC = A(1B + 1\overline{C} + BC) + \overline{A}(0B + 0\overline{C} + BC) = \\
 &= A(B + 0 + BC) + \overline{A}(0 + C + BC) = AB(1 + C) + \overline{A}C(1 + B) = AB + \overline{A}C
 \end{aligned}$$

Yapılan minimizasyonla aslında (t.2-15) teoremi ispatlanıyor.

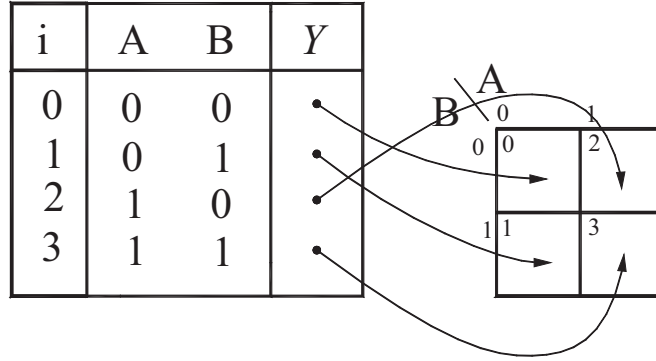
### 2.6.2. KARNO MİNİMİZASYON YÖNTEMİ

Bu yöntem çok basittir ve yeterince pratiktir ve bu yüzden çok sıkça kullanılıyor. *Minimizasyon grafik yoluyla gerçekleşiyor, öyle ki fonksiyon Karno kartı (KK) olarak adlandırılan özel bir tablo yardımıyla tanımlanıyor.*

Çalışmayla başlanması ve fonksiyon için fonksiyonun doğruluk tablosundan elde edilen uygun KK'nın oluşması gerekiyor. KK belirli sayıda alanlardan (karelerden) oluşan poligondur. Verilen anahtarlama fonksiyonunun kombinasyonel tablosundan herhangi bir satırın tanımlanması için, KK'dan sadece bir alan yeterlidir. *Buna göre sadece bir kare ile verilen fonksiyondan bir min-term ya da bir maksterm tanımlanabilir.* KK mantıksal fonksiyonların çok basit bir şekilde basitleştirilmesini sağlıyor: KK'nın görsel incelenmesiyle.

KK ve doğruluk tablosu arasındaki bağlantının açıklanması için, önce en basit durumu göstereceğiz, o da Şek.2-1'de verilmiş iki değişkenli fonksiyonu  $Y(A,B)$  için KK'nın görünüşüdür ve onun doğruluk tablosudur.





Res. 2-1. İki değişkenli fonksiyonun Karno kartı ve doğruluk tablosu

Fonksiyon değerlerinin girilmesi gereken sütunun doldurulmuş olmadığını tahmin edelim, çünkü şimdi KK'nın nasıl görüneceğini açıklamak istiyoruz. Şekilden görüldüğü gibi iki değişkenli fonksiyon için KK'nın toplam  $2^2 = 4$  alanı vardır. Ardından ondalık gösterimde, indisler aracılığıyla, doğruluk tablosundan her satır ve gösterilen KK'nın her alanı işaretleniyor. Bu şekilde belirtilmiş KK kombinasyon tablosunun yerine kullanılabilir.

Şimdi KK alanları fonksiyon değerleriyle doldurulmalıdır. Alanların doldurulması iki farklı şekilde yapılıyor:

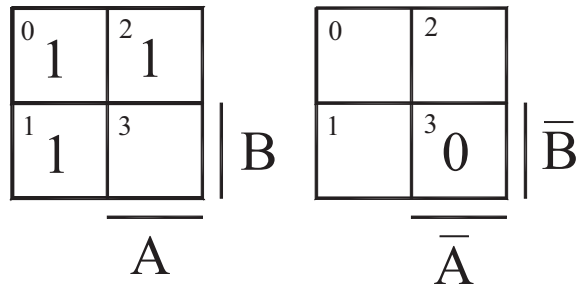
1. Çizilmiş KK'da 1 ile sadece fonksiyon değerlerinin uygun satırlarına uyan alanlar doldurulabilir. Böylece fonksiyon KANB şekline, yani mintermler toplamı ile gösterilmiş olacak ve

2. Fonksiyon KBNB şeklinde, yani makstermlerin çarpımı şekline de gösterilebilir. Bunun için KK'da sadece fonksiyonun değeri 0 olduğu satırlara uygun alanlarda 0 yazılıyor.

Örnek olarak, tablo 2-18 OVE fonksiyonunun kombinasyon tablosunu tanımlıyor, Şek.2-2 a) ise, alanların 1 değerleriyle doldurulmuş olduğuna göre bu fonksiyonun KANB'de KK'nı gösteriyor. Şek.2-2 b) OVE fonksiyonunu KBNB şekline gösteriyor, çünkü bu defa KK'da sadece değeri 0 olan alanlar doldurulmuştur.

i	A	B	Y
0	0	0	1
1	0	1	1
2	1	0	1
3	1	1	0

Tab.2-18. OVE fonksiyonunun doğruluk tablosu



a) KANB şekli

b) KBNB şekli

Şek.2-2. OVE fonksiyonunun Karno kartları

Üç değişkenli fonksiyon için KK'nın  $2^3 = 8$  alanı vardır, görünüşü ise Şek 2-3'te gösterilmiştir, dört değişkenli fonksiyonun KK'ı Şek 2-4'te verilmiştir ve  $2^4 = 16$  alanı olduğu görünüyor.



		AB		B( $\bar{B}$ )	
		00	01	11	10
C	0	0	2	6	4
	1	1	3	7	5
		A( $\bar{A}$ )			

$C(\bar{C})$

Şek.2-3. Üç değişkenli fonksiyonun Karno kartı

		AB		B( $\bar{B}$ )	
		00	01	11	10
CD	00	0	4	12	8
	01	1	5	13	9
D( $\bar{D}$ )	11	3	7	15	11
	10	2	6	14	10
		A( $\bar{A}$ )			

$C(\bar{C})$

Şek.2-4. Dört değişkenli fonksiyonun Karno kartı

Her KK için her alanın indisi ezberlenmelidir, çünkü bu indis verilen fonksiyonun doğruluk tablosundan uygun satıra uyar, ancak işaretlemek oldukça basittir. Şöyle ki, KK fonksiyonun KANB şekline ilgili ise, KK 1-lerle dolduruluyor ve o zaman KK kenarları yanında değişkenleri doğrudan şekilde belirtiyoruz. Alanların işaretlenmesi sağ alanda birinci değişkenden başlıyor, ikinci değişken yukarıda, üçüncü sağda ve dördüncüsü solda yazılıyor. Her kenar için, tüm alanlardan bir yarısını KK kenarı yanında çizgi ile işaretlenmiş doğrudan şekilde belirli değişken örtüyor, alanların diğer yarısı ise değişkenin tümleyen şekliyle örtülüyor. KK'da fonksiyonun 0-ları yazılırsa, o zaman bu fonksiyon KBNB şeklinde görünüyor ve o zaman çizgiler yanında değişkenler tümleyen şekilde yazılıyor, örtülmemiş alanlar için ise onların şekli doğrudandır. Bu işaretlendirmeyi metnin devamında en çok kullanacağız.

		B		
		0	4	12
D	1	1	1	1
	3	1	1	1
	2	1	1	1
	1	1	1	1
		A		

$C$

Şek. 2-5. Dört değişkenli fonksiyon  $Y(A, B, C, D) = \Sigma m(1,2,3,8,12)$  için KK örneği

Şek.2-5'ten, her iki mintermin  $m_8 = \bar{A}\bar{B}\bar{C}\bar{D}$ , ve  $m_{12} = \bar{A}B\bar{C}\bar{D}$  bağlı alanlarda buldukları görülüyor. Mintermler birbirinden, B değişkeninin bir elemenda tümleşik diğer elemenda ise doğrudan, yani tümleşik olmayan şekilde olduğuna göre farklıdır. Bu iki mintermi toplarsak şunu elde edeceğiz:  $\bar{A}\bar{B}\bar{C}\bar{D} + \bar{A}B\bar{C}\bar{D} = \bar{A}\bar{C}\bar{D}(B + \bar{B} = \bar{A}\bar{C}\bar{D})$ .

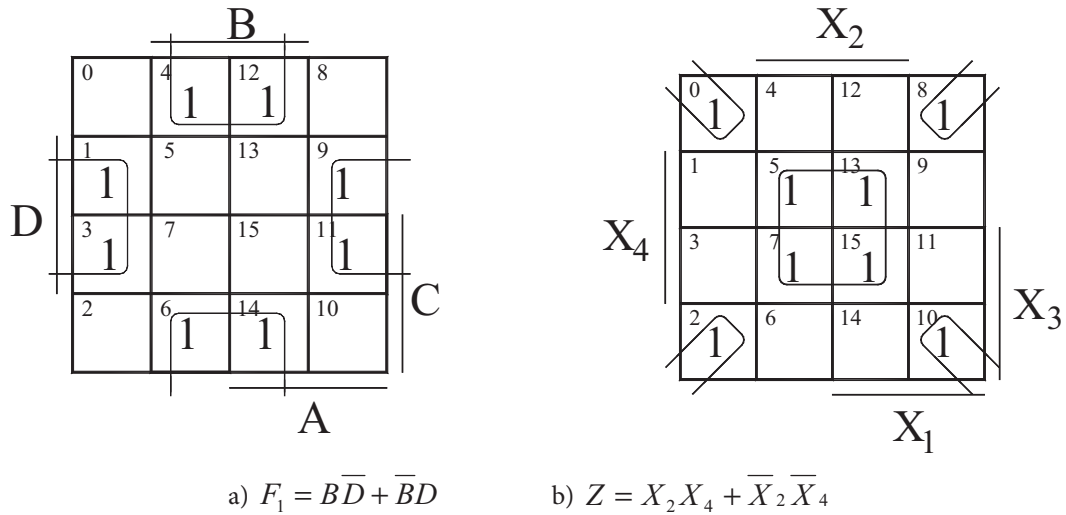
KK'ların en önemli özelliği birbiriyle yatay veya dikey temas eden alanların, Grey koduna dayanarak, sadece bir değişkene göre farklı olan minterme, ya da maksterme uymalarıdır. Bu değişken bir alana uyan elemenda doğrudan ortaya çıkıyor, tümleyen şekilde ise diğer alana uyan elemenda ortaya çıkıyor. Bu alanlar bağlanabilecek (birleşebilecek) ve bu yüzden onları **bağlı alanlar** olarak adlandıracağız. Bu özelliğin sunduğu basitleştirmeyi görmek için, KK'ı Şek.2-5 te verilmiş  $Y = Y(A, B, C, D)$  fonksiyonu için bir örnek inceleyeceğiz.

Dörder değişken içeren her iki mintermin, B değişkeni elenerek sadece üç değişken içeren tek bir elemanla değiştirilebileceğini görüyoruz. Bu prensip 1 ile doldurulmuş ve bağlı olan (yatayda veya dikeyde yan yana bulunan) herhangi iki diğer alan için uygulanabilir.

Buna göre KK, geometrik görüntüleme yardımıyla daha basit ifadelerle değiştirilebilen min-term kombinasyonların kolay tanınmasını sağlıyor. Genel olarak, her bağlı minterm çifti, min-termlerden birdeğişken daha az değişken içeren tek bir elemana kombine edilebilir. Bu eleman, bağlı alanları tanımlayan mintermlerden, bir mintermde tümlenmemiş, diğer mintermde ise tümlenmiş olan değişken elenerek elde ediliyor.

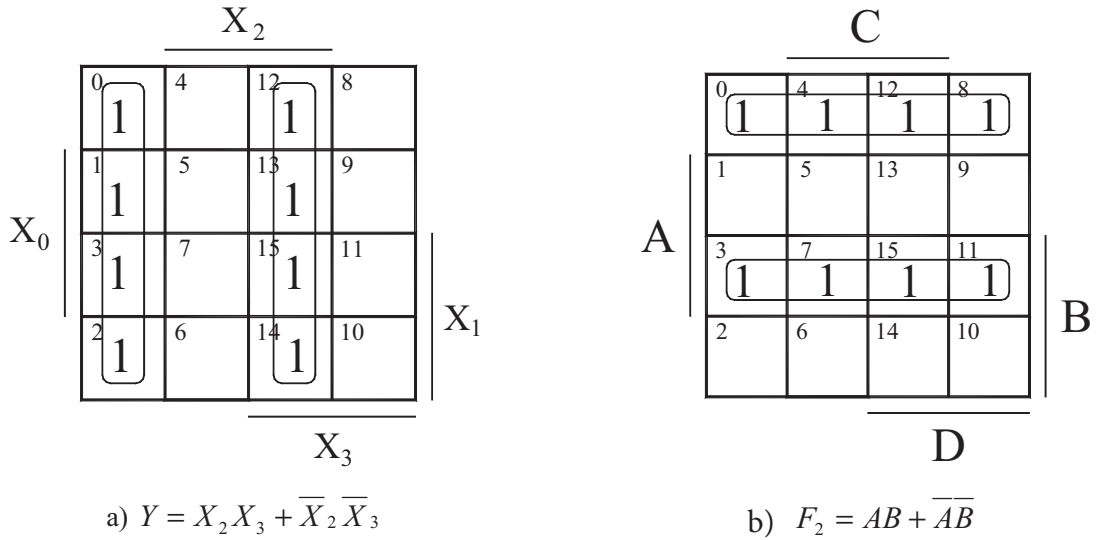
Bağlı alanlar sadece geometriksel olarak birbirine temas eden alanlar değil, bir değişkenin ortaya çıkma şekline (doğrudan ya da tümleşik) göre farklı olan mintermleri temsil eden alanlar da bağlı alanların olduğunu vurgulamalıyız. Buna dayanarak, herhangi satırın en sol sütununda bulunan her alanın, aynı satırda en sağ sütunda bulunan alanla bağlı olduğu ve herhangi sütunun en üst satırında bulunan alanın, aynı sütunda en alt satırda bulunan alanla bağlı olduğu ortaya çıkıyor. Kısaca, KK'nin üst kenarı alt kenarla temastadır, sol kenar ise sağ kenarla temastadır. Son köşegen alanları (dört köşe) bağlı alanlardır, ancak her köşegenin iki tarafındaki son alanları, bağlı alanlar değildir.

KK'ından iki bağlı alanın, bir değişkenin elendiği elemanın elde edildiğini gördük. Benzer şekilde her  $N=2^n$  bağlı alan ortaya çıkınca, onlardan bir kez tümleşik, bir kez doğrudan (nominal) şekilde bulunan  $n$  değişkenin elenmiş olduğu, tek bir elemanın elde edilmesini göstereceğiz. Şek.2-6 a), b), c) ve ç)'de dört bağlı alanın farklı gruplaşma şekilleri gösterilmiştir.

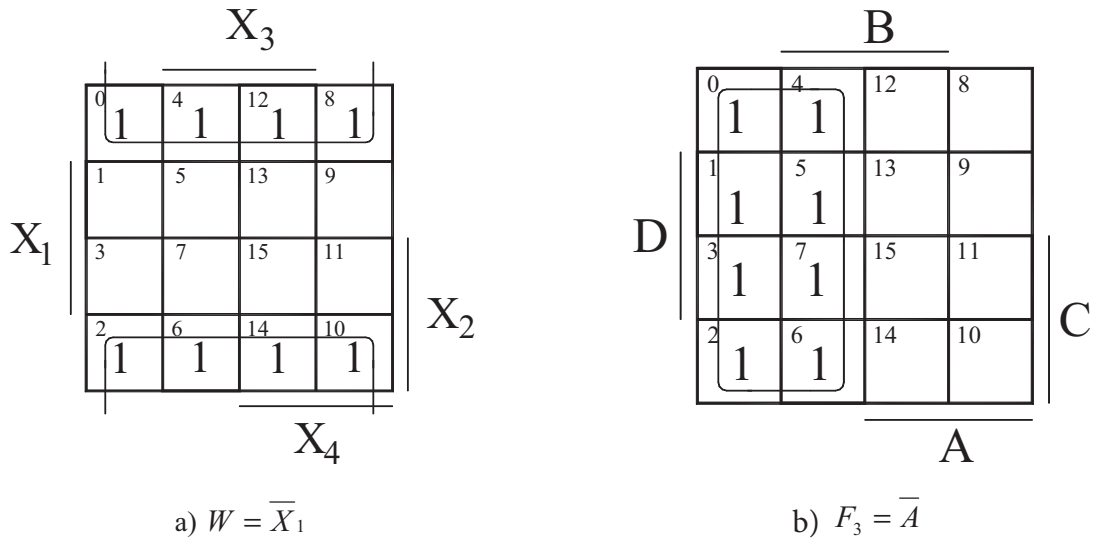


Şek.2-6. Karnı kartları

Şek.2-6 a)'da  $F_1 = F_1(A, B, C, D)$  fonksiyonun, Şek. 2-6 b)'de  $Z = Z(X_1, X_2, X_3, X_4)$  fonksiyonun, Şek. 2-7 a)'da  $Y = Y(X_3, X_2, X_1, X_0)$  fonksiyonun ve Şek. 2-7 b)'de  $F_2 = F_2(D, C, B, A)$  fonksiyonun KK'ı verilmiştir. Şek. 2-8 a) ve b)'de sırasıyla sekiz alanın bağlandığı  $W = W(X_4, X_3, X_2, X_1)$  ve  $F_3 = F_3(A, B, C, D)$  fonksiyonların KK'ları tanımlanmıştır. Her çevreleme yanında, uygun alan grubunu tanımlayan çarpım yazılmıştır.



Şek. 2-7. Dört değişkenli fonksiyonlar için KK'nda dört alanın bağlanma örnekleri



Şek. 2-8. Dört değişkenli fonksiyonlar için KK'nda sekiz alanın bağlanma örneği

Daha büyük sayıda değişkene bağlı olan fonksiyonlara gelince, görünürlüğün çok daha az olacağı bilinmelidir. Böylece, örneğin 5 değişkenli fonksiyonlar için KK'nın  $2^5 = 32$  karesi olur, ancak bu durumda da kullanılabilir. Bununla ilgili olarak, 5 değişenden KK'nın gösterilmesi için iki yöntem vardır. Bir yöntemde göre yan yana 16'şer alanlı iki KK bağlanıyor, diğer yöntemde göre 16 alanlı iki KK yan yana bulunuyor, ancak birbirinden ayrı olarak, bir bir üstüne oldukları düşünülüyor. 6 değişkenli fonksiyon için  $2^6 = 64$  alanlı KK gerekecek. Bu durumda KK'nın çok büyük olacağı açıkça görülüyor ve KK ile çalışma o kadar zor olacak ki aslında kullanışsız olacak.

### 2.6.2.1. KARNO YÖNTEMİNİN KULLANIMI

İlerleyen bölümlerde, verilen anahtarlamalı fonksiyonun minimizasyonu için Karno yönteminin hangi şekilde uygulanması gerektiğini açıklayacağız. Fonksiyon doğruluk tablosuyla ya da bir KNB'nde: KANB veya KBNB şeklinde verilebilir. Saydığımız herhangi bir şekilden, verilen fonksiyon için KK oluşabilir doldurulabilir. Şimdiye kadar açıkladıklarımız KANB şeklinde verilmiş fonksiyonlarla ilgiliydi. Bu yüzden KANB şeklinde verilmiş fonksiyonların KK'ın, yani 1 ile doldurulmuş KK'ın kullanımının açıklamasıyla devam edeceğiz.

**KANB'de verilmiş fonksiyonların minimizasyonu.** Şimdiye kadar açıkladıklarımızdan, sezgisel olarak verilen fonksiyonun minimizasyonunun hangi şekilde yapılacağı sonucu ortaya çıkıyor. Şöyle ki, 1'lerle doldurulmuş tüm alanlar kapsanmalıdır çünkü her öyle bir alan fonksiyonun bir mintermini belirtiyor. Bu arada, bağlanmış alanların edebildiği kadar az, yüksek erimli ( $r$ -range) geçerli yüzeyler grupların oluşması gerekiyor.  $r$  erimden geçerli yüzey,  $(n-r)$  ortak değişkeni olan,  $2^r$  sayıda alanın gruplaşmasıyla oluşuyor. Bu arada,  $r$  pozitif tam sayıdır ve onun için  $k < n$  geçerlidir,  $n$  ise verilen fonksiyonun bağlı oldupu bağımsız değişkenlerin toplam sayısıdır. Böylece en az çarpım elde edilecektir ve her çarpımın en az sayıda değişkeni olacaktır.

Buna göre, verilen fonksiyonun KK yardımıyla basitleştirilmesi sırasında şu prensplerin yerine getirilmesi gerekiyor:

1. Seçilen alanların kombinasyonları o şekilde olmalıdır ki her alan gruplarda en az bir kere yer almalıdır, ya da en az bir alan grubunda bulunmalıdır. Bu arada bir alan birkaç grupta yer alabilir;
2. Ayrı gruplar o şekilde seçilmelidir ki her grupta edebildiği kadar çok sayıda alanın girmesi sağlanmalıdır, ancak aynı zamanda edebildiği kadar az sayıda farklı grupların elde edilmesi gerekiyor.

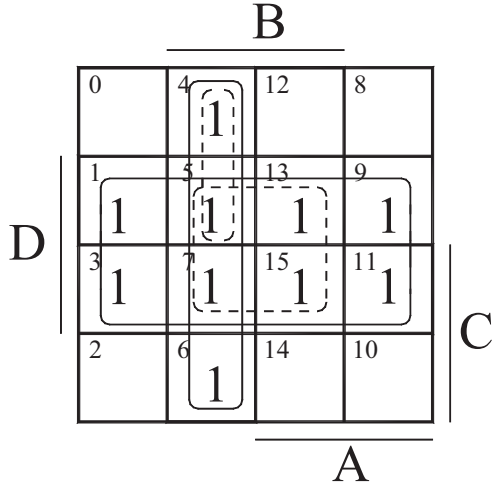
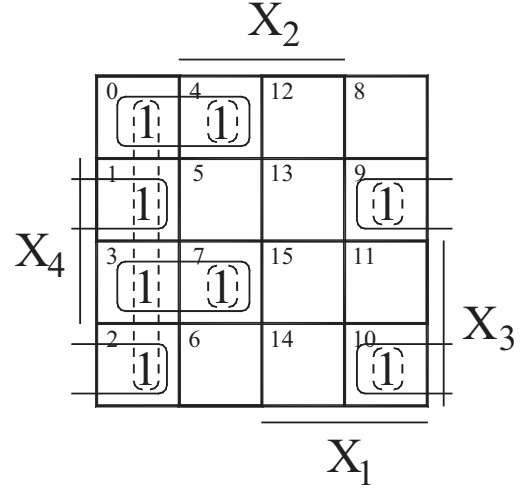
Değişkenlerin herhangi bir çarpımına implikant denir. İmplikant bir grupta bir, iki, üç, dört vs. alanın çevrelenmesiyle elde ediliyor. **Birincil implikant (BİK)** başka bir implikantta tamamiyle girmeyen implikanttır, yani başka bir çevrelenmiş grupta tamamiyle kapsamamış çevrelenmiş alanlar grubudur. Genel olarak, bir fonksiyonun tanımlanması için tüm BİK'lerin kullanılması şart değildir. Bahsettiklerimizi daha iyi aşağıdaki tipik örnekle anlayacağız:

	B			
	0	4	12	8
			1	1
	1	5	13	9
D	3	7	15	11
	1			
	2	6	14	10
				1
	A			

Şek.  $Y_1 = Y_1(A, B, C, D)$   
fonksiyonunun KK'ı

Burada KK'ı Şek.2-9'da gösterilmiş  $Y_1 = Y_1(A, B, C, D)$  fonksiyonu söz konusudur. Bu örnekte BİK'ler şunlardır:  $p_1 = m_2 + m_3$ ,  $p_2 = m_8 + m_{12, p_3} = m_2 + m_{10, p_4} = m_8 + m_{10}$ . Fonksiyon iki şekilde tanımlanabilir:  $Y_1 = p_1 + p_2 + p_3$  olarak ya da  $Y_1 = p_1 + p_2 + p_4$  olarak. Her iki gösterimde  $p_1$  BİK'in kullanılması gerekiyor, çünkü aksi takdirde  $m_3$  mintermi göz önüne alınmayacak. Bu yüzden,  $p_1$  implikantına **temel (esas) implikant (TİK)** denir. Buna göre TİK en azından bir minterm (1 ile doldurulmuş) alan kapsayan ve hiçbir başka TİK ile örtülmemiş BİK'tir.  $p_2$  BİK'i de aynı öyle TİK'tir, çünkü onuz  $m_2$ -mintermi olmayacak.  $p_3$  ve  $p_4$  BİK-leri TİK değildir.

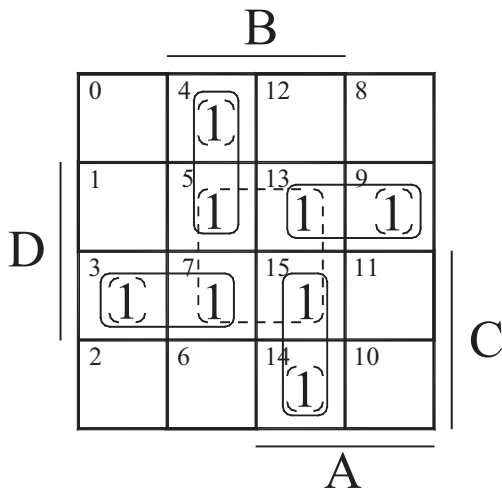
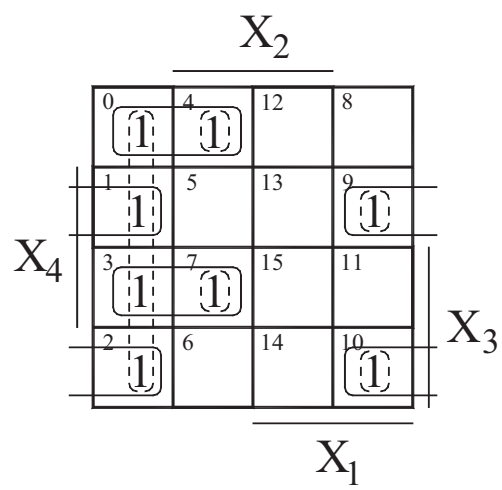
MANB şeklinde tanıtmak için tüm TİK-lerin ve fonksiyonun tüm diğer 1-lerinin örtülemsi için gereken BİK-lerin göz önüne alınması gerekiyor. Bu arada, minimizasyonun ekonomik olarak yapıldığını diyoruz, eğer edebildiği kadar az PİK elde edilirse ve her BİK'in edebildiği kadar çok alanı varsa.

Şek. 2-10.  $Y_2 = Y_2(A, B, C, D)$  fonksiyonun KK'ıŞek. 2-11.  $Y_3 = Y_3(X_1, X_2, X_3, X_4)$  fonksiyonun KK'ı

Şek.2-10'da  $Y_2 = Y_2(A, B, C, D)$  fonksiyonun KK'ı gösterilmiştir. Kesik çizgilere iki implikant işaretlenmiştir, dolu çizgiyle iki BİK işaretlenmiştir. Bu örnekte BİK-ler aynı zamanda hem TİK'tir. Şek.2-11'de  $Y_3 = Y_3(X_1, X_2, X_3, X_4)$  fonksiyonun KK'ı ile ilgili bir örnek gösterilmiştir. Kesik çizgilerle BİK-ler, dolu çizgilerle ise TİK-ler işaretlenmiştir.

Daha büyük sayıda alan içeren BİK'in başlangıçta arama çabası fonksiyonun minimum şekli olmayan sonuç verebilir. Örneğin aşağıdaki iki örneğe bakalım.

$Z_1 = Z_1(A, B, C, D)$  fonksiyonun Şek.2-12'de verilmiş KK'ı var,  $Z_2 = Z_2(X_1, X_2, X_3, X_4)$  fonksiyonun KK'ı ise Şek.2-13'te gösterilmiştir.

Şek 2-12.  $Z_1 = Z_1(A, B, C, D)$  fonksiyonuŞek. 2-13.  $Z_2 = Z_2(X_1, X_2, X_3, X_4)$  fonksiyonun KK'ı

Yukarıda açıklanan prensibi takip ederek her iki şekilde alanları kesik çizgilerle çevrelememiz gerekir. Ancak, böyle çevreleme en kolay ve en basit çözümü vermediği görünüyor. Gerçekten öyle olduğunu diğer şekilde dolu çizgilerle çevreleme şekli gösteriyor. Böyle çevreleme fonksiyonlar hakkında daha basit tanımlama sunuyor.

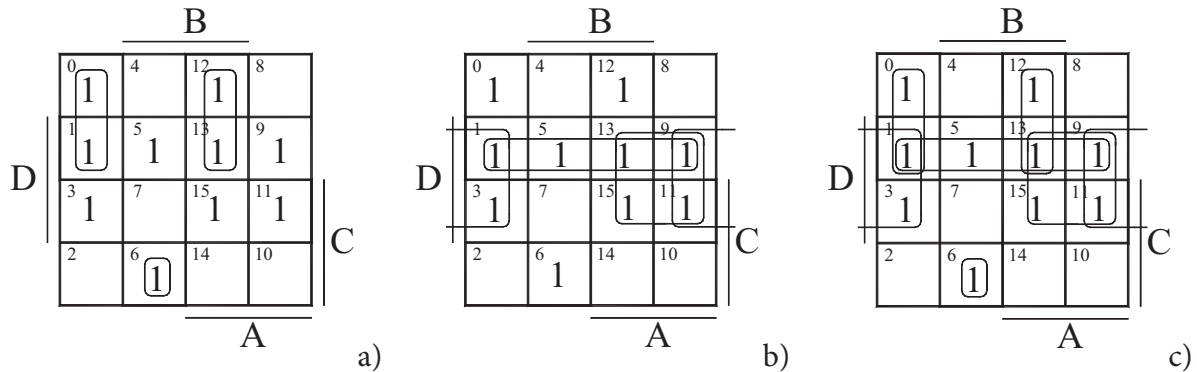
Fonksiyonun minimum şeklini elde etmek için ve minimizasyon sırasında istenmeyen hatadan kaçınılması için şu algoritmayı uygulamamız gerekir:

1. TİK olarak, hiçbir başka alanla birleşemeyen alanlar alınıyor ve çevreleniyor;
2. Tek bir şekilde başka bir alanla yan yana olan alanlar tanıtılsın ve çiftlere gruplaşsın;
3. Diğer üç alanla gruplaşabilen, tek şekilde dörtlük yapabilen 1-lerle alanlar tanıtılsın;
4. Bu süreç 8 alanlı gruplar için de devam ediyor;
5. İlk 4 adımın tamamlanmasından sonra örtünmeyen alanlar kalırsa, onlar edebildiği kadar az ancak daha büyük 1'ler gruplara gruplaştırılabilir.

Başka bir deyişle, KK'dan tüm birler birer birer kontrol ediliyor. En önce daha büyük grup oluşturamayabildiğinden dolayı yalnız kalan tüm 1-ler alınıyor. Ardından çevrelenmemiş kalan her 1-li için, en az bir çevrelenmemiş ve kontrol edilen alan dışında başka alanla örtünemeyen 1-li'nin olup olmadığı kontrol ediliyor. Böyle çiftler varsa, onlar da çevreleniyor. Bu işlem her kalan ve örtünmemiş 1'linin, diğer örtünmemiş alanlarla dörtlük oluşturmayan en az bir alanla dörtlük oluşturup oluşturamadığı tespit ediliyor. Böyle dörder 1-li gruplar varsa onlar da çevreleniyor. Devamda sekizer 1-li grupların olup olmadığı tespit ediliyor vs. Kapsanmamış 1-ler kalırsa, onlar edebildiği kadar fazla 1-ler, özellikle örtülmemiş 1'ler kapsayan daha az sayıda gruplara katılıyor.

Bu algoritma ile ilk olarak tüm TİK-ler belirleniyor, kalanlardan ise tüm 1-leri ve fonksiyonu en azından bir kez kapayan daha büyük ve edebildiği kadar az PİK seçiliyor.

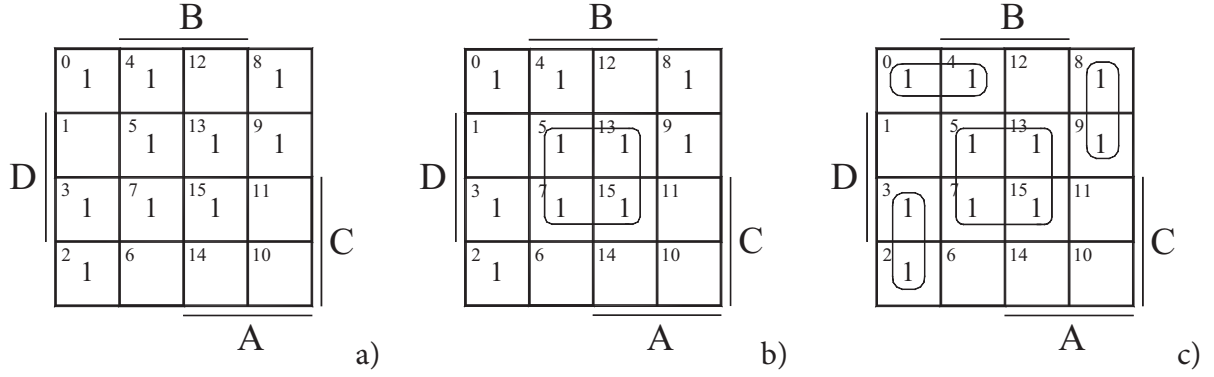
Verilen algoritma, KK'ı Şek.2-15 a), b) ve c)'de verilmiş  $F_1 = F_1(A, B, C, D)$  fonksiyonun ve KK'ı Şek.2-15 a), b) ve c)'de verilmiş  $F_2 = F_2(A, B, C, D)$  fonksiyonun aşağıda verilen çözüm örnekleri üzerine tamamıyla uygulanmış. İkinci örnek biraz daha özeldir, çünkü algoritmanın beşinci adımını da gerçekleştiriyor. Kalan örtünmemiş alanların hangi şekilde gruplaştığını kendimiz düşünmeliyiz ve karar vermeliyiz.



Şek. 2-14.  $F_1 = F_1(A, B, C, D)$  fonksiyonun KK'ı

Birinci örnekte  $F_1$  fonksiyonu için şu minimum şekli elde ediliyor:

$$F_1 = ABCD + \overline{ABC} + \overline{ABC} + AD + BD.$$

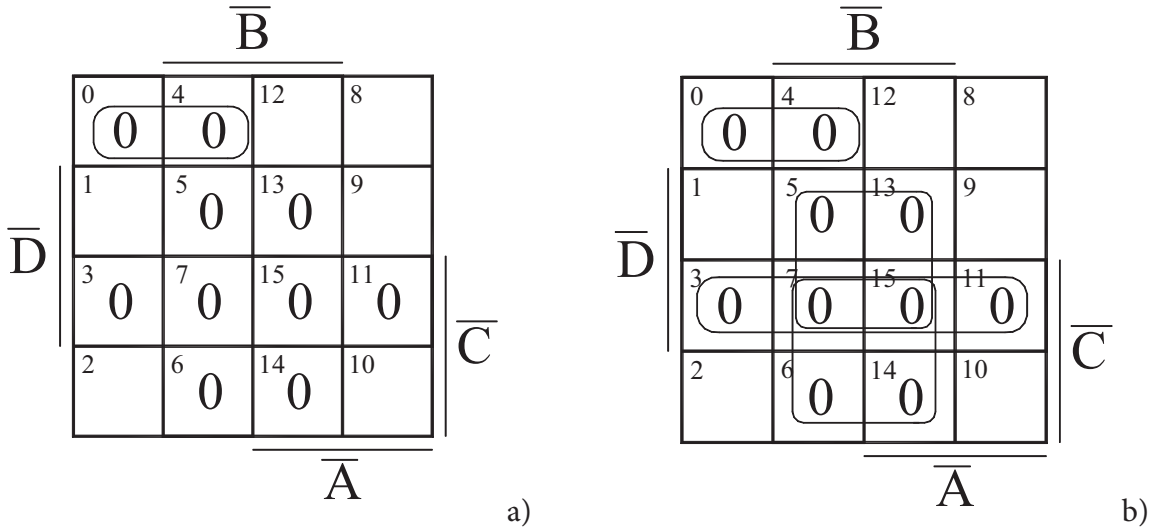
Şek. 2-15.  $F_2 = F_2(A, B, C, D)$  fonksiyonun KK'ı

İkinci örnekte  $F_2$  fonksiyonunun minimum şekli şudur:

$$F_2 = BD + \overline{ABC} + \overline{ACD} + \overline{ABC}.$$

**KBNB şeklinde verilmiş fonksiyonların minimizasyonu.** Mantıksal fonksiyonun KBNB şekline de verilebildiğine göre, bu sefer de 1'lerle değil 0'larla doldurulmuş KK'nın gözetleneceği doğaldır. Bu şekilde verilmiş fonksiyonların minimizasyonu prensipte fonksiyonun KANB'de verilince 1 ile doldurulmuş KK'nın durumunda olduğu gibi aynı olacak, sadece şimdi gruplaşma ve çevreleme 0-larla doldurulmuş alanlar arasında yapılacak. Sonunda fonksiyon MANB şeklinde değil MBNB şeklinde elde ediliyor.

Burada **implisent** terimini tanımlayacağız. Her implisent toplam çarpımdan bir elemanı, herhangi toplamı tanımlıyor. **Birincil implisent (BİS)** başka bir implisente tamamıyla girmeyen implisenttir. **Temel implisent (TİS)** en azından bir maksterm (0 ile alan) kapsayan BİS'tir. Çalışma prensibi önceki durumla aynıdır, KBNB şeklinde fonksiyonun minimizasyonun gösterilmesi için KK'ı Şek.2-16 a) ve b)de verilmiş olan  $Z = Z(A, B, C, D)$  fonksiyon örneğini çözeceğiz.

Şek. 2-16.  $Z = Z(A, B, C, D)$  fonksiyonun KK'ı

Son örnek için MBNB şeklinde minimize edilmiş  $Z$  fonksiyonu için şu elde ediliyor:

$$Z = (A + C + D)(\overline{B} + \overline{C})(\overline{B} + \overline{D})(\overline{C} + \overline{D}).$$

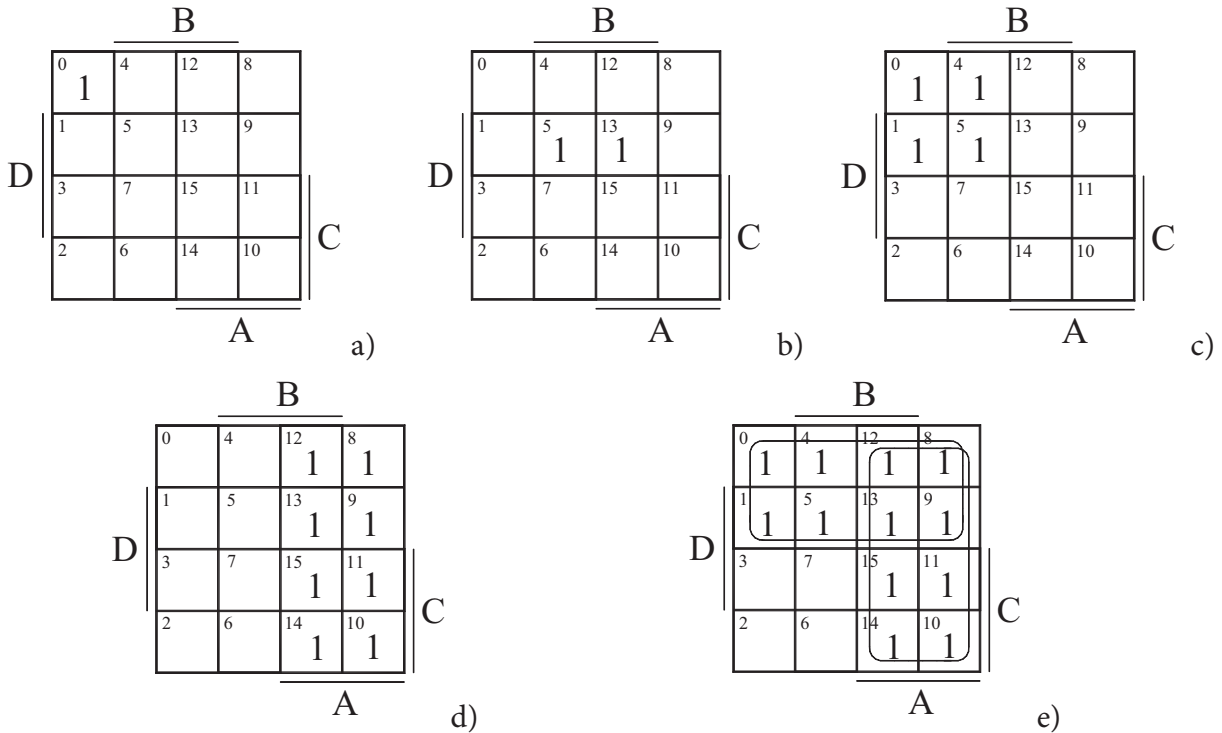


### 2.6.2.2. ANB/BNB ŞEKLİNDE VERİLEN FONKSİYONUN MİNİMİZASYONU

Yukarıdaki bölümlerde KK yöntemin kullanıldığı şeklini açıkladık, ancak bunu fonksiyonun doğruluk tablosunun, yani onun KANB veya KBNB şekli verildiği takdirde yaptık. Ancak, bazan ANB veya BNB şeklinde verilen fonksiyonlara rastlanabilir. Bu durumlarda ANB veya BNB cebirsel şekilde KANB, yani KBNB biçimine nasıl genişletildiğini göstermiştir. Ardından fonksiyonun doğruluk tablosunun nasıl oluştuğunu ve ondan KK'ı nasıl oluştuğunu gösterdik. Böyle analitik genişlemenin çok karmaşık ve zor olabileceğinden dolayı ve çözülmesi sırasında bir adımın daha gerektiğinden, yani kombinasyon tablosunun oluşması gerektiğinden dolayı, ANB veya BNB şeklinde verilmiş fonksiyondan KK'ın doğrudan nasıl doldurulduğunu göstereceğiz. Verilen örnek, ANB şeklinde  $Y = \overline{A}BCD + B\overline{C}D + A\overline{C} + A$  ile verilmiş  $Y = Y(A, B, C, D)$  fonksiyonuyla ilgilidir.

KK alanlarında 1'in yazdırılması adım adım yapılıyor ve birinci çarpımdan başlayarak son çarpıma kadar devam ediyor. Birinci çarpım,  $\overline{A}BCD$  mintermdir ve o fonksiyonun KK'ında  $m_0$  olarak doğrudan yazılabilir. İkinci çarpım  $B\overline{C}D$  'dir ve bu çarpım KK'nda  $B=1, C=0, D=1$  geçerli olduğu yere uygundur ve A değişkeninin değerine bağlı değildir, yani A'nın iki değeri için geçerlidir. Buna göre bu çarpımla  $A=0, B=1, C=0, D=1$  ( $m_0$ ) ve  $A=1, B=1, C=0, D=1$  ( $m_{13}$ ) geçerli olduğu iki alan doldurulacaktır. Benzer şekilde üçüncü çarpım, AC için dört alan doldurulacaktır: B değişkeninin iki değeri için ve D değişkeninin iki değeri için ve  $A=0, C=0$  geçerli olduğu alanlar doldurulacaktır. Son olarak A elemanı için sekiz alan doldurulacaktır.  $A=1$  ve kalan üç değişkenin, B, C ve D'nin her iki değeri için ve  $A=1$  geçerli olduğu alanlar doldurulacaktır.

Tüm bu işlem Şek 2-17'de (a), b), c), ç) ve d) adım adım gösterilmiştir ve ayrı ayrı doldurulmanın kombinasyonu olan KK'ın son görünüşü verilmiştir.



Şek. 2-17. ANB şeklinde verilen fonksiyonun minimizasyon süreci

Gerçeklene işlemde görüldüğü gibi birkaç minterm:  $m_0, m_5, m_{13}$  1-lerle birden fazla kez doldurulmuştur. Ancak böyle bir durum sorun yaratmıyor çünkü minterm kendi kendisi toplanacağı zaman aynı minterm bir kez olarak elde ediliyor. Son resimde minimizasyon yapılmıştır ve buradan fonksiyonun MANB'i,  $Y = A + \overline{C}$  şeklindedir.



### 2.6.2.3. KISMEN VERİLMİŞ FONKSİYONLARIN MİNİMİZASYONU

Şimdiye kadar, tam olarak tanımlanmış mantıksal fonksiyonların nasıl minimize edildiğini gördük. Böyle fonksiyonun değeri, bağımsız değişkenlerin her kombinasyonu için her zaman 0 veya 1'di. Bu şekilde verilen fonksiyonla, mintermlerin ve makstermlerin uyduğu KK alanları hemen doldurulabiliyordu ve normal olarak ardından fonksiyon minimum şekilde elde ediliyordu. Ancak, pratikte kısmen verilmiş fonksiyonlara da rastlanabilir.

Kısmen tanımlanmış fonksiyon minimizasyonun gerçekleştiği süreci göstermek için,

$$Y = \sum m(1,2,5,6,9) + \sum_{xm} m(10,11,12,13,14,15).$$

ANB şeklinde verilmiş olan  $Y = Y(A, B, C, D)$  fonksiyonuyla ilgilidir.

Verilen ANB şekline göre,  $m_1, m_2, m_5, m_6, m_9$  mintermleri için fonksiyonun değeri  $Y = 1$  olacak,  $m_{10}, m_{11}, m_{12}, m_{13}, m_{14}, m_{15}$  mintermlerin uyduğu bağımsız değişken kombinasyonları için fonksiyonun değeri önemli değildir. Bu yüzden bu mintermlere uyan alanlarda „x” sembolünü yazacağız.

KK'nın görünüşü Şek. 2-18'de verilmiştir. İlerleyen çalışmada „x”leri isteğimize göre yorumlayabilmemiz en önemlidir: *minimizasyonu basitleştirirse 1 veya 0 olarak yorumlayabiliriz, ya da fonksiyonun basitleştirilmesinde katkıları yoksa onları ihmal edebiliriz.*

				B							
		0	4	12	8						
D	1	1	1	X	1						
	3			X	X						
	2	1	1	X	X						
											C
										A	

Şek.2-18.  $Y = Y(A, B, C, D)$  fonksiyonunun KK'ı

				B							
		0	4	12	8						
D	1	(1)	(1)	X	(1)						
	3			X	X						
	2	(1)	(1)	X	X						
											C
										A	

Şek.2-19. „x”-siz minimizasyon

				B							
		0	4	12	8						
D	1	(1)	(1)	X	(1)						
	3			X	X						
	2	(1)	(1)	X	X						
											C
										A	

Şek.2-20. „x” ile minimizasyon

Şek.2-19'da, „x” ile doldurulmuş hiçbir alanın göz önüne alınmadığı halde minimizasyon yapılmıştır. Bu durumda fonksiyon için şu ifade elde ediliyor:

$$Y = (m_1 + m_2) + (m_1 + m_9) + (m_2 + m_6) = \overline{A}\overline{C}D + \overline{B}\overline{C}D + \overline{A}C\overline{D}.$$

Şek.2-20'de,  $m_{13}, m_{14}, m_{10}$  bulunan „x”-leri 1 olarak yorumlayarak, fonksiyon için önceki durumdaki daha basit biçim elde ediyoruz:

$$Y = (m_1 + m_5 + m_9 + m_{13}) + (m_2 + m_6 + m_{10} + m_{14}) = C\overline{D} + \overline{C}D.$$

$m_{11}, m_{12}, m_{15}$  alanlarında bulunan diğer „x”ler fonksiyonda elemanların sayısının düşürülmesi, ya da elemanlarda değişkenler sayısının azalması için yardımcı olamaz, öyle ki bu „x”leri 0 olarak alıyoruz. Kısmen tanımlanmış fonksiyonların Karno kartlar yöntemiyle minimizasyon sürecini ekolarak daha iyi açıklamak için Şek.2-21'de a), b), c), ç), d), e)'de dört değişkene bağlı fonksiyonlar örnekleri verilmiştir.

	<b>B</b>				
	0	4	12	8	
			1	1	
<b>D</b>	1	5	13	9	
	1	X	1	1	
	3	7	15	11	
	X	1			
	2	6	14	10	
	X	1			
	<b>A</b>				

$$a) Y_1 = AB + \bar{C}D + \bar{A}C$$

	<b>B</b>				
	0	4	12	8	
	X	X		1	
<b>D</b>	1	5	13	9	
	1	X			
	3	7	15	11	
	X		X	1	
	2	6	14	10	
	1			1	
	<b>A</b>				

$$b) Y_2 = \bar{B}\bar{D} + \bar{A}\bar{B} + \bar{B}C$$

	<b><math>\bar{C}</math></b>				
	0	4	12	8	
	0			0	
<b><math>\bar{A}</math></b>	1	5	13	9	
		0	0	0	
	3	7	15	11	
	X	X			
	2	6	14	10	
	X			X	
	<b><math>\bar{D}</math></b>				
	<b><math>\bar{B}</math></b>				

$$c) Y_3 = (C + A)(\bar{C} + \bar{A})(\bar{D} + B + \bar{A})$$

	<b><math>X_2</math></b>				
	0	4	12	8	
	X		X	0	
<b><math>X_4</math></b>	1	5	13	9	
			X	0	
	3	7	15	11	
		0			
	2	6	14	10	
	0	0			
	<b><math>X_1</math></b>				
	<b><math>X_3</math></b>				

$$ç) Y_4 = (X_1 + \bar{X}_3)(\bar{X}_1 + X_2 + X_3)(\bar{X}_1 + X_3 + \bar{X}_4)$$

	<b><math>\bar{C}</math></b>				
	0	4	12	8	
		0	0		
<b><math>\bar{A}</math></b>	1	5	13	9	
	0	0	X	0	
	3	7	15	11	
	0		X	0	
	2	6	14	10	
	0			X	
	<b><math>\bar{D}</math></b>				
	<b><math>\bar{B}</math></b>				

$$d) Y_5 = (\bar{C} + B)(C + \bar{A})(C + \bar{B})$$

	<b>B</b>				
	0	4	12	8	
				1	
<b>D</b>	1	5	13	9	
			1	1	
	3	7	15	11	
		1	1		
	2	6	14	10	
		1	1	1	
	<b>A</b>				
	<b>C</b>				

$$e) Y_6 = BD + BC + \bar{A}\bar{B}\bar{C} + \bar{A}B\bar{D}$$

Şek. 2-21. Karno kartları yönteminin uygulanmasıyla dört değişkenli fonksiyonların minimizasyonu

## 2.7. ANAHTARLAMALI AĞLAR

Mantıksal fonksiyonlar pratik olarak farklı teknik çözümler uygulayarak farklı şekilde gerçekleştiriyorlar, ancak her zaman iki durumu olan elemanlar ve/veya bileşenler kullanıyorlar: mekanik anahtarlar, röleler, yarı iletken anahtarlar vb. Nasıl olsa, genelde bu elemanların uygun semalar ve semboller yardımıyla grafiksel gösterilme gereksinimi ortaya çıkıyor. Amacımız mantıksal fonksiyonları elektronik anahtarlar ile gerçekleşmesi olduğunu göz önüne alarak, devamında dikkatimizi bu konunun açıklamasına, yani mantıksal fonksiyonların mantıksal devreler ve sembolik işaretleri yardımıyla grafiksel gösterime yönlendireceğiz.

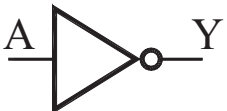
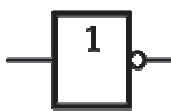
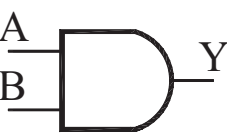
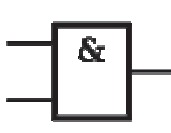
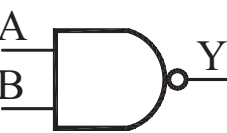
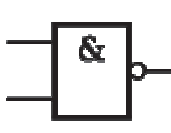

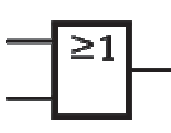
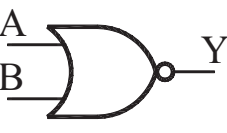
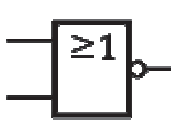
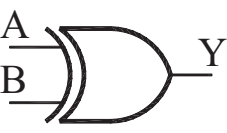
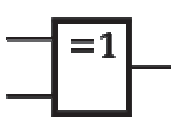
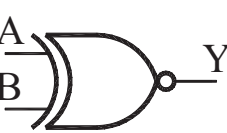
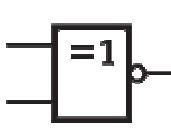
*Belirli sayıda farklı mantıksal devrelerin uygun bağlanmasıyla elde edilen ve anahtarlama fonksiyonu gerçekleştiren her yapıya mantıksal, birleşimsel, komütasyon veya anahtarlama ağ denir.* Anahtarlama ağ uygun blok şemayla (mantıksal blok-diyagramla) gösterilebilir. Blok-diyagram grafiksel tanımlama şeklidir ve mantıksal devreler sembollerinin uygulanmasıyla elde ediliyor. Birleşimsel (kombinasyonel) ağlarda bir mantıksal devreden ağın herhangi bir girişime geri bağlantı yoktur. Bu yüzden ağın her bir girişi sadece giriş değişkenlerin mevcut değerlerine bağlıdır.

### 2.7.1. TEMEL MANTIKSAL DEVRELER

Şimdiye kadar incelediğimiz tüm standart anahtarlama fonksiyonlar teknik açıdan **mantıksal devreler, kapılar ya da geçitler** olarak adlandırılan özel bileşenlerin yardımıyla yapılıyorlar. Her mantıksal devrenin, gerçekleştirdiği fonksiyona bağlı bir çıkışı ve çıkıştaki fonksiyonun bağlı olduğu değişkenlerin getirildiği bir ya da fazla girişi vardır.

Her mantıksal devre için, uygun grafiksel sembolü-elementer blok-diyagramı ya da **mantıksal sembolü** vardır. Bu mantıksal sembol yardımıyla bu devre mantıksal diyagramda gösteriliyor. Farklı mesleki kitaplarda mantıksal devreler için farklı sembollere rastlanabilir. IEEE (elektroteknik mesleğinden mühendisler enstitüsü), IEC (Uluslararası elektroteknik komisyonu) tarafından teklif edilen mantıksal devrelerin işaretlenmesini kabul etmiş ve standartlandırmıştır. Bu standarda göre iki temel sembol türü vardır. Bir grupta, mantıksal fonksiyona bağlı olarak farklı biçimleri olan semboller yer alıyor, ikinci grupta ise dikdörtgen biçimde olan semboller kullanılıyor. Amerikan Ulusal Standartlar Enstitüsü (ANSİ), farklı biçimde olan sembolleri uyguluyor. Bu semboller genel olarak dünya çapında da eğitim gereksinimleri için kullanılıyor. Diğer taraftan, dikdörtgen biçimli semboller dijital bileşenler, cihazlar ve aygıtlar üreticileri tarafından belgelerin korunması için kullanılıyor. Tüm bunları göz önüne alarak, devamdaki bölümlerde ANSİ standardına göre işaretlemeyi, yani farklı biçimleri olan semboller kullanacağız. Buna rağmen, tab. 2-19 tablosunda, VE, VEYA ve DEĞİL temel (elementer) mantıksal devreler sembollerinin tam ve kıyaslama gösterimi verilmiştir. Ayrıca eviricinin, genel OVE ve OYA devrelerinin her iki standarda göre işaretlenmesi verilmiştir. Ek olarak her mantıksal devre yanında, doğruluk tablosu ve kısa açıklamayla analitik şekilde gerçekleştiği fonksiyon belirtilmiştir.

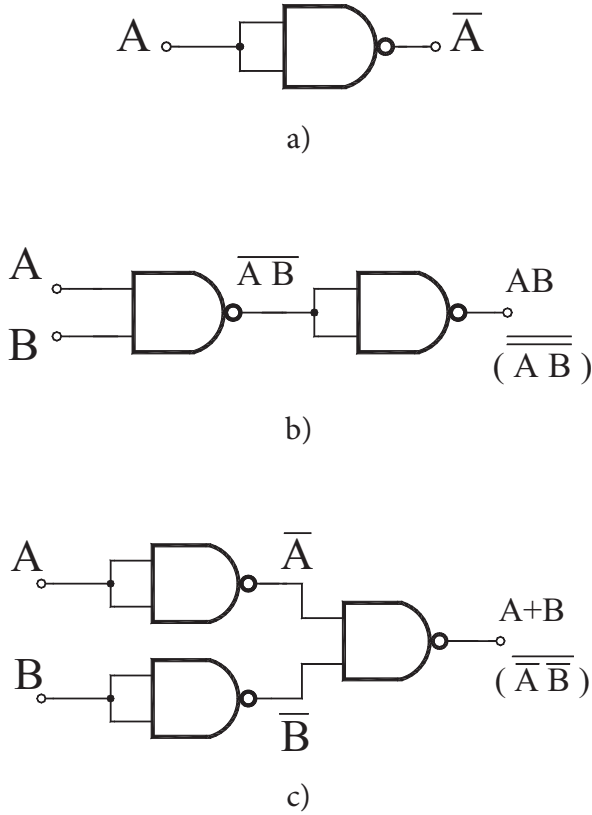
EVİRİCİ (DEĞİL), OVE ve OYA mantıksal fonksiyonların sembollerinden, değişkenin tümlemesi küçük çemberle (o) işaretlendiği görülüyor.

Mantıksal Devre	Mantıksal sembol		Mantıksal denklem	Doğruluk tablosu	Fonksiyonun açıklaması															
	ANSI	IEC/IEEE																		
Evirici (DEĞİL)			$Y = \bar{A}$	<table border="1"> <tr><td>A</td><td>Y</td></tr> <tr><td>0</td><td>1</td></tr> <tr><td>1</td><td>0</td></tr> </table>	A	Y	0	1	1	0	Tümleme (evirme)									
A	Y																			
0	1																			
1	0																			
VE			$Y = A \cdot B$	<table border="1"> <tr><td>A</td><td>B</td><td>Y</td></tr> <tr><td>0</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>1</td><td>0</td></tr> <tr><td>1</td><td>0</td><td>0</td></tr> <tr><td>1</td><td>1</td><td>1</td></tr> </table>	A	B	Y	0	0	0	0	1	0	1	0	0	1	1	1	Mantıksal çarpma
A	B	Y																		
0	0	0																		
0	1	0																		
1	0	0																		
1	1	1																		
OVE			$Y = \overline{A \cdot B}$	<table border="1"> <tr><td>A</td><td>B</td><td>Y</td></tr> <tr><td>0</td><td>0</td><td>1</td></tr> <tr><td>0</td><td>1</td><td>1</td></tr> <tr><td>1</td><td>0</td><td>1</td></tr> <tr><td>1</td><td>1</td><td>0</td></tr> </table>	A	B	Y	0	0	1	0	1	1	1	0	1	1	1	0	Mantıksal çarpmanın tümleyeni
A	B	Y																		
0	0	1																		
0	1	1																		
1	0	1																		
1	1	0																		
VEYA			$Y = A + B$	<table border="1"> <tr><td>A</td><td>B</td><td>Y</td></tr> <tr><td>0</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>1</td><td>1</td></tr> <tr><td>1</td><td>0</td><td>1</td></tr> <tr><td>1</td><td>1</td><td>1</td></tr> </table>	A	B	Y	0	0	0	0	1	1	1	0	1	1	1	1	Mantıksal toplama
A	B	Y																		
0	0	0																		
0	1	1																		
1	0	1																		
1	1	1																		
OYA			$Y = \overline{A + B}$	<table border="1"> <tr><td>A</td><td>B</td><td>Y</td></tr> <tr><td>0</td><td>0</td><td>1</td></tr> <tr><td>0</td><td>1</td><td>0</td></tr> <tr><td>1</td><td>0</td><td>0</td></tr> <tr><td>1</td><td>1</td><td>0</td></tr> </table>	A	B	Y	0	0	1	0	1	0	1	0	0	1	1	0	Mantıksal toplamının tümleyeni
A	B	Y																		
0	0	1																		
0	1	0																		
1	0	0																		
1	1	0																		
D-YA			$Y = A \oplus B$ $Y = A\bar{B} + \bar{A}B$	<table border="1"> <tr><td>A</td><td>B</td><td>Y</td></tr> <tr><td>0</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>1</td><td>1</td></tr> <tr><td>1</td><td>0</td><td>1</td></tr> <tr><td>1</td><td>1</td><td>0</td></tr> </table>	A	B	Y	0	0	0	0	1	1	1	0	1	1	1	0	Dışlamalı mantıksal toplama
A	B	Y																		
0	0	0																		
0	1	1																		
1	0	1																		
1	1	0																		
D-OYA			$Y = \overline{A \oplus B}$ $Y = AB + \bar{A}\bar{B}$	<table border="1"> <tr><td>A</td><td>B</td><td>Y</td></tr> <tr><td>0</td><td>0</td><td>1</td></tr> <tr><td>0</td><td>1</td><td>0</td></tr> <tr><td>1</td><td>0</td><td>0</td></tr> <tr><td>1</td><td>1</td><td>1</td></tr> </table>	A	B	Y	0	0	1	0	1	0	1	0	0	1	1	1	Dışlamalı mantıksal toplamının tümleyeni
A	B	Y																		
0	0	1																		
0	1	0																		
1	0	0																		
1	1	1																		

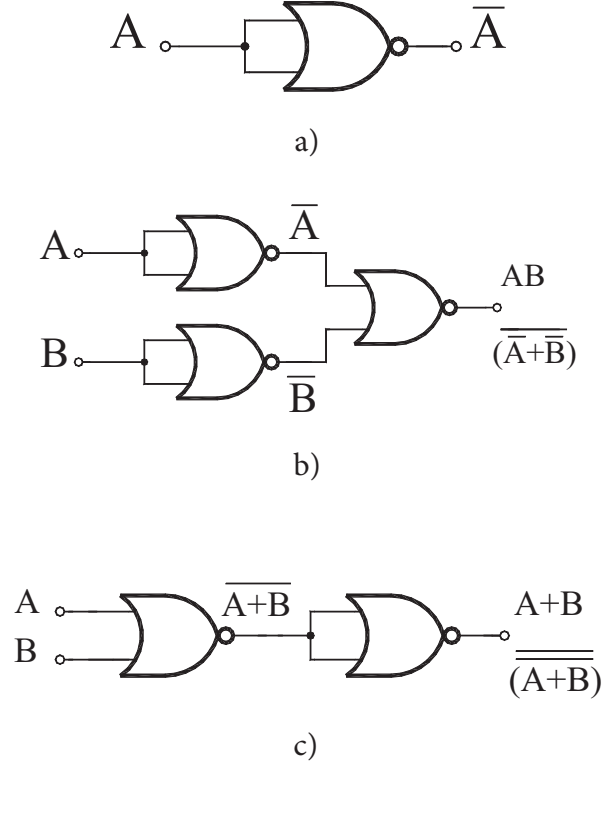
Tab.2-19. Standart mantıksal devreler sembollerinin anlamları

Tablo 2-19'un son iki satırında, DIŞLAMALI VEYA ve OYA, yani D-YA ve D-OYA mantıksal fonksiyonları gerçekleştiren mantıksal devreleri verilmiştir, çünkü bu iki mantıksal devrele hem teoride hem pratikte sıkça rastlanıyor.

Özellikle önemli mantıksal devreler olarak OVE ve OYA mantıksal devrelerini vurgulamak gerekiyor, çünkü onların yardımıyla her diğer anahtarlamalı fonksiyon gerçekleştirilebilir. DEĞİL, VE ve VEYA temel fonksiyonların OVE fonksiyonu aracılığıyla ifade edildiği (2-22, 2-23, 2-24) denklemlerden hareket ederek, Şek.2.22'de a), b) ve c)'de DEĞİL, VE ve VEYA temel mantıksal devrelerin sadece OVE devreleriyle gerçekleştirme şekli gösterilmiştir. Diğer taraftan (2-25, 2-26 ve 2-27) denklemlerini uygulayarak, Şek.2.23'te a), b) ve c)'de temel mantıksal devrelerin OYA devrelerin uygun bağlanmasıyla tanımlanmıştır.



Şek. 2-22. Sadece OVE geçitle



Şek. 2-23. Sadece OYA geçitle

DEĞİL (Evirici), VE ve VEYA temel mantıksal devrelerin gerçekleştirilmesi

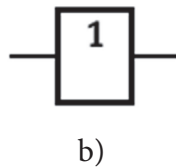
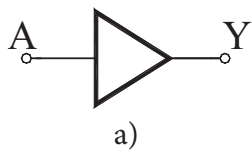
## 2.7.2. DİĞER TEMEL MANTIKSAL DEVRELER

Temel ve evrimsel mantıksal devreler dışında, pratikte sıkça özel amaçları olan diğer temel mantıksal devreler de vardır. Bunlar da arabellek devresi, üç durumlu devre ve iki yönlü geçitidir. Bu devreleri devamda inceleyeceğiz, çünkü bu devrelerin gerçek dijital bileşenlerin projelendirilmesinde ve yapılmasında çok önemli rolleri vardır, bununla beraber de büyük pratik uygulamaları vardır.

Bu devrelerin tanımlanması pratik çalışmada gerçek sorunlardan geliyor. Daha doğrusu, bir mantıksal devreden çıkışın, bu devre gerçekten sağlayabildiği akımdan daha büyük akım sağlayan başka bir elemanla veya bileşenle bağlanma gereksiniminden ve farklı mantıksal devreden fazla çıkışın bir noktada bağlanma ihtiyacından geliyor.

### 2.7.2.1. ARABELLEK DEVRESİ

Önce uyumlama devresini tanıtacağız (arabellek, İng. *buffer*). Bu devrenin evirici gibi bir girişi ve bir çıkışı vardır, sadece arabellek devresinde çıkış girişin mantıksal değerini izliyor, yani çıkışta girişin bulunduğu yani mantıksal seviye elde ediliyor. Şek.2-24 a) ve b)'de her iki standarda göre arabelleğin mantıksal sembolleri gösterilmiştir, ANSİ ve IEC/IEEE. Arabelleğin davranışını göz önüne alarak, mantıksal denklemi  $Y = A$  olacak, doğruluk tablosu ise tab.2-20 tabosunun şekline uygun olacak.



$$Y = A.$$

A	Y
0	0
1	1

Şek.2-24. Sembolik işaretleme

Mantıksal denklem

Tab.2-20. Kombinasyon tablosu

Arabellek devresi (arabellek)

Arabellek devrenin en önemli özelliği onun aynı mantıksal seviye sırasında daha yüksek çıkış akımının verme olanağıdır. Bu özellikten dolayı, arabellek, alçak dirençli harcıyıcıya doğrudan bağlanamayan mantıksal devrenin çıkışında bağlanıyor, çünkü harcıyıcının mantıksal devreden fazla enerji alma ve bu şekilde aşırı yüklenme ve hasar tehlikesi vardır. Arabellek, uyum sağlama ve verilen mantıksal derenin izin verilen en yüksek akımdan daha büyük akım çekebilene harcıyıcıyla bağlanma kurgusu olarak kullanılıyor. Bu devreler, bazı mantıksal devrenin verebildiği güçten daha yüksek güç gereken yerlerde kullanılıyor ve bu yüzden arabellek devresine yürütücü (sürücü) mantıksal devre ya da uyarma devresi, güçlü çıkışlı mantıksal devre veya sürücü (ing. *driver*) de denir.

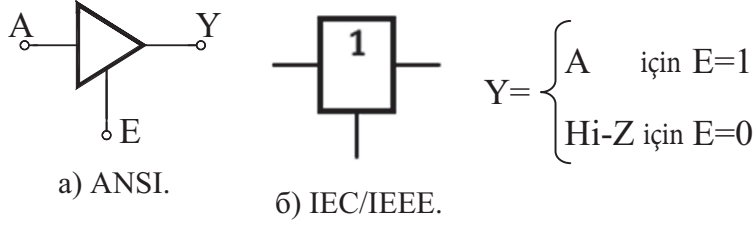
### 2.7.2.2. ÜÇ DURUMLU ARABELLEK DEVRESİ

Bu devreye üç-statikli arabellek de denir. Adı İngilizce terminolojiden geliyor ve three-state ya da tri-state buffer terimi altında rastlanabilir, çünkü iki genel mantıksal durumu dışında: 1 veya 0, Y devrenin çıkışı *üçüncü durumda* veya *yüksek empedans durumunda* olabilir. Bu durum genelde HiZ, Hi-Z veya sadece Z ile işaretleniyor. Devre üçüncü durumda bulunduğu zaman, hiç enerji tüketmiyor ( $I_Y \rightarrow 0$ ). Bu, üçüncü durumda Y çıkışı kesildiğinden ve sonsuz büyük değerli rezistör olarak davrandığından kaynaklanıyor ( $Z_Y \rightarrow \infty$ ,  $R_Y \rightarrow \infty$ ).

Devrenin normal çalışması E (İng. Enable) işaretlenmiş yeni eklenmiş kontrol girişinin mantıksal seviyesi üzerinden sağlanıyor. E kontrol girişinde 1 getirilirse ( $E = 1$ ), o zaman devre normal arabellek olarak davranıyor, yani girişin mantıksal durumu çıkışa aktarılıyor ( $Y=A$ ). Ancak, E girişine 0 getirilirse ( $E=0$ ),  $R_Y \rightarrow \infty$  ve  $I_Y \rightarrow 0$  olunca devrenin çıkışı üçüncü duruma geçerek ( $Y = \text{HiZ}$ ), devre aslında kapanıyor. Buna göre E kontrol sinyalinin aktif seviyesi 1'dir, çünkü sadece  $E=1$  olunca, arabellek normal şekilde çalışıyor.

Üç-statiklikli arabellek devrenin davranışı su musluğuna andırıyor.

Şek.2-25 a) ve b)de üç durumlu arabelleğin her iki standarda göre: ANSI ve IEC/IEEE, mantıksal sembolleri gösterilmiştir, tab.2-21 doğruluk tablosu ise bu devrenin çalışma şeklini daha fazla yansıtıyor.



E	A	Y
0	x	Hi-Z
1	0	0
	1	1

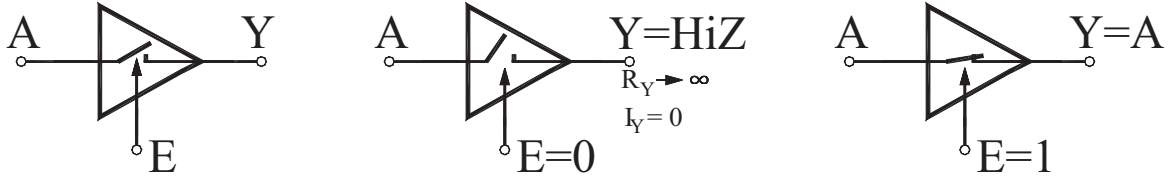
Şek.2-25. Sembolik işaretler

Mantıksal denklem

Tab. 2.21. Kombinasyon tablosu

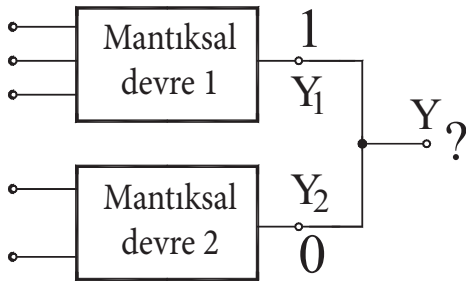
Üç durumlu arabellek devresi (üç statiklikli arabellek) ve 1'de aktif olan kontrol sinyali

Üç durumlu devrenin çalışma prensibini açıklamak için, onu dışardan kontrol edilen mekanik anahtar olarak göstereceğiz (2-26).

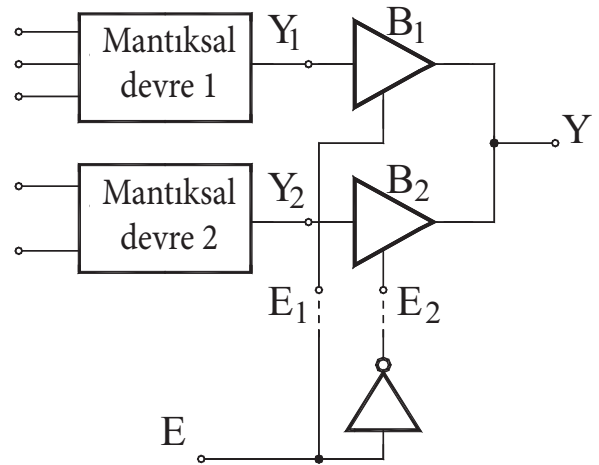


Şek. 2-26. Bir yönde kontrol edilen mekanik anahtar olarak arabellek devresi

Üç durumlu arabellek pratik nedenlerden meydana gelmiş, hem de Şek.2-27'de gösterilmiş olduğu gibi bir noktada farklı mantıksal devrelerden en az iki çıkışın bağlanma gereksiniminden. Kapılardan çıkışların basitçe bağlanması sorun yaratabilir çünkü ortak noktadaki durum kontrol edilemiyor. Böyle durum, devrelerin çıkışlarındaki durumların birbirinden farklı olduğu zaman meydana geliyor. Bu durumda bağlantı noktasının mantıksal durumu belirlenemiyor ve sonuç olarak çakışma (çarpılma) meydana gelebilir.



Şek. 2-27. Mantıksal devreler çıkışlarının doğrudan bağlanması ve çakışmanın meydana gelmesi



Şek. 2-28. Üç durumlu arabelleğin aracılığıyla mantıksal devre çıkışlarının bağlanması

Çakışma, Şek.2-28'e göre üç durumlu arabelleğin kullanımıyla kaçınılabilir. Şekilden görüldüğü gibi,  $B_1$  ve  $B_2$  arabelleklerin var olmasından dolayı mantıksal devrelerin çıkışları gereğe göre bağlantıdan ayrılabilir. Şöyle ki, Y ortak bağlantı noktasındaki mantıksal durum sadece geçirilen arabelleğe, yani aktif olana arabelleğe bağlı olacak. Bu arada ikinci devrenin arabelleği pasif durumda tutulmalı ve onun çıkışı üçüncü duruma gitmesi zorlanak ortak noktadan koparak ayrılacak. Bu durum, ortak noktanın mantıksal durumun belirlenmesi gereken devre arabelleğinin kontrol çıkışına 1 getirilerek, pasif devrenin arabelleğindeki kontrol girişine 0 getirilerek sağlanıyor, örneğin  $E_1 = 1$ ,  $E_2 = 0$ . Buradan, ortak noktadaki mantıksal durumun sadece birinci devredeki çıkışın mantıksal seviyesinden belirleneceği açıkça görünüyor ya da  $Y = Y_1$  geçerlidir. Tab.2.22 işlevsel tablo, açıklanan çalışma şeklini daha detaylı açıklıyor.

E1	E2	Y
0	0	Hi-Z
0	1	$Y_2$
1	0	$Y_1$
1	1	?

Tab. 2-22. İki kontrol girişle yönetim

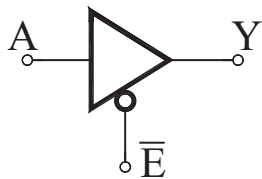
E	E1	E2	Y
0	1	0	$Y_1$
1	0	1	$Y_2$

Tab. 2-23. Bir kontrol girişle ve evirici ile yönetim

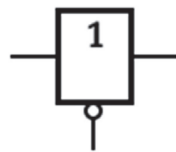
Hatanın meydana gelmemesi için, her iki kontrol girişi tek bir kontrol girişi E'ye bağlanabilir. Bu arada bir arabelleğin kontrol girişi bu ortak girişe doğrudan bağlanacak, ikinci arabelleğin kontrol girişi ise, Şek.2.27'de kesik çizgilerle gösterilmiş olduğu gibi evirici üzerinden gidecek. Bu durumda 2-23 kombinasyon tablosu geçerli olacak. Bu tabloya göre  $E=0$  olunca o zaman  $Y = Y_1$ , ancak  $E=1$  ise o zaman  $Y = Y_2$ .

Ortak noktada bağlanma, birden fazla mantıksal devrenin çıkışları için de yapılabilir. Bu arada, kontrol girişlerle yönetime dikkat edilmelidir. Şöyle ki, her zaman mantıksal devrelerden sadece biri, yani çıkışları bağlı olan devreler aktif olmalıdır, tüm diğer devrelerin pasif olmaları gerekiyor.

Pratikte sıkça, kontrol sinyalin aktif seviyesi alçak olan üç durumlu devreler de vardır. Bu durumda, devrenin sembolünde, Şek.2-29 a) ve b) şekillerine göre, kontrol girişte küçük çember ekleniyor. Böyle üç-statikli devre, tab.2-24 doğruluk tablosuna uygun olarak,  $E=0$  olunca aktif olacak,  $E=1$  olunca ise pasif olacak ve çıkışta üçüncü durum elde edilecek.



a) ANSI.



b) IEC/IEEE.

$$Y = \begin{cases} A & \text{için } E=0 \\ \text{Hi-Z} & \text{için } E=1 \end{cases}$$

Mantıksal denklem

E	A	Y
0	0	0
0	1	1
1	x	Hi-Z

Tab. 2-24. Kombinasyon tablosu

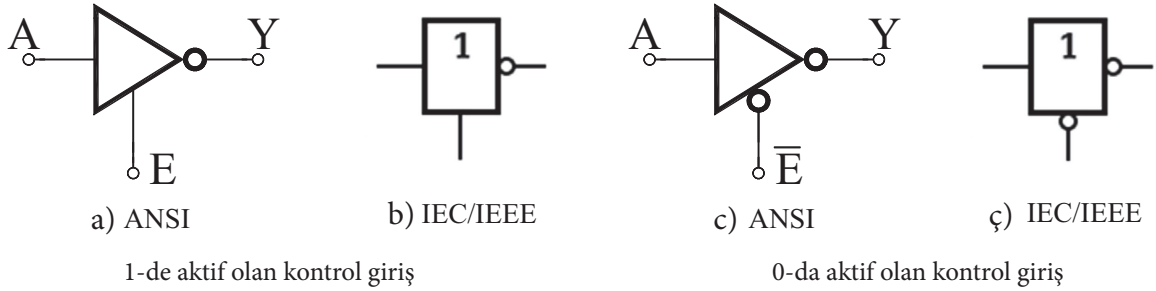
Şek.2-29. Sembolik işaretler

Üç durumlu arabellek devresi (üç statikli arabellek) ve 0'da aktif olan kontrol girişi



Çalışmayı **imkansızlaştırmak** için D (İng.disable) kontrol girişli üç durumlu arabelleklerin yapımları da vardır. Bu durumda, kontrol girişine aktif seviyenin getirilmesiyle, D çıkışı üçüncü duruma gidiyor. D'de pasif mantıksal seviye getirilirse, çıkış normal çalışıyor ve onda girişin durumu aktarılıyor.

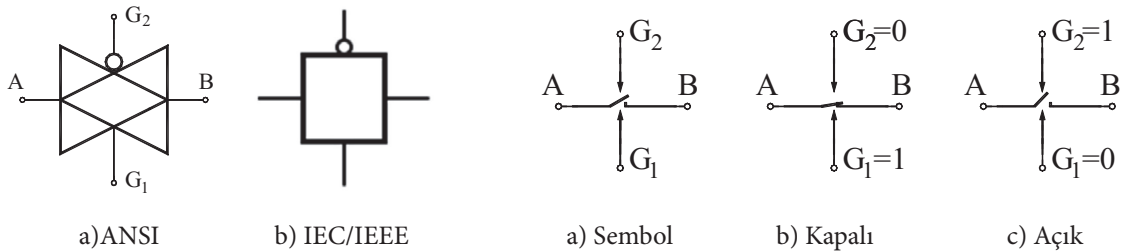
Sıkça rastlanan diğer arabellek türü, arabellek-eviricilerdir. Bu arabellek türleri, prensipte yukarıda saydığımız arabellek devreleri gibi aynı şekilde çalışıyorlar, sadece arabellek-eviriciler,ek olarak giriş sinyalinin evirmesini gerçekleştiriyorlar. Onlarda, kontrol girişinde uygun aktif sinyal getirilirse, çıkışta giriş değişkenin tümleyen değeri elde ediliyor ( $Y = \bar{A}$ ). Bu yüzden, devrenin mantıksal sembolünün çıkışında, Şek.2-30 a), b), c), ç) göre küçük çember ekleniyor. Ters durumda, kontrol girişi pasif ise, arabellek üçüncü duruma gidiyor.



Şek. 2-30. Üç durumlu arabellek-eviricinin mantıksal sembolleri

### 2.7.2.3. KARŞILIKLI (İLETİM) GEÇİTİ

Pratikte, mantıksal sembolü Şek.2-31'de gösterilmiş devrelere de özel ilgi gösteriliyor. Şek.2-31 a)'da sembolün ANSİ standardına göre görünüşü, Şek.2-31 b)'de ise sembolün IEC/IEEE standardına göre tanımlanması verilmiştir. Burada, iki yönde, ya da karşılıklı giriş ve çıkış arasındaki bağlantıyı imkansızlaştıran veya kesen devre söz konusudur. Bu yüzden bu devre, Şek.2-32'ye göre dışardan kontrol edilen mekanik anahtar olarak da gösterilebilir.



Şek.2-31.Karşılıklı anahtarın Şek.2-32. sembolik işaretleri

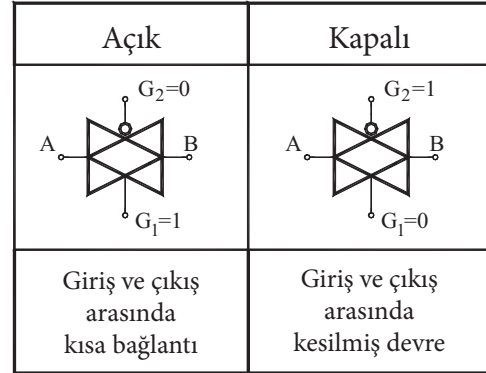
Dışardan kontrol edilen mekanik anahtar olarak iletim geçiti

Devrenin davranışı kontrol girişlerin mantıksal durumları üzerinden yönetiliyor. Şöyle ki, iletim geçiti  $G_1$  kontrol girişine 1 getirilince açılıyor, ikinci kontrol girişinde  $G_2$ 'de ise 0 getiriliyor. Böylece giriş ve çıkış arasında bağlantı kuruluyor ve aralarında iki yönde iletim sağlanıyor: A'dan B'ye ya da B'den A'ya. Aksi durumda,  $G_1$  yönetim girişine 0 getirilirse ve aynı zamanda  $G_2$  ikinci kontrol girişine 1 getirilerek, iletim kapısı kapanıyor çünkü A girişi/çıkışı ve B girişi/çıkışı arasında bağlantı kurulamıyor. Bu yüzden çıkış girişten kesilmiş (çıkarılmış, kapanmış) durumdadır ve sonsuz büyük (yüksek) dirençlik durumunda, ya da üçüncü durumda bulunuyor. İki yönlü geçidin çalışma prensibi tab. 2-25 tablosunda ve Şek.2-33 ile gösterilmiştir.

Çalışma şekli sayesinde bu devrenin elektroniğin farklı dallarında büyük kullanımı var ve bu yüzden birkaç farklı ismi vardır. Bu devre iki yönlü geçit, ikili anahtar, analog anahtar ya da zamansal seçici olarak da adlandırılıyor.

G1	G2	Giriş – çıkış iletişimi
1	0	Kısa bağlantı ( $R_{A-B} \rightarrow 0$ )
0	1	Kesinti (Hi-Z) ( $R_{A-B} \rightarrow \infty$ )

Tab. 2-25. İletim geçidin fonksiyonel tablosu



Şek.2-33. İletim geçidin çalışma şeklinin açıklaması

İkili anahtarın çalışmasının kontrolü sadece tek bir kontrol giriş hatıyla yapılabilir. Bu durumda bir kontrol girişin her mantıksal seviyesi doğrudan biçimde olmalıdır, diğer kontrol girişin mantıksal seviyesi ise evirici aracılığıyla tümleyen şekilde olmalıdır. Bağlanma prensipte, Şek.2.28'de gösterilmiş ve E1 ve E2 kontrol girişlerin E tek kontrol hatında bağlı olduğu üç durumlu arabellek yönetimi ile aynıdır.

### 2.7.3. ANAHTARLAMALI DEVRELERİN ANALİZİ

Anahtarlama ağının analizi onun işlevsel planıyla ilgilidir, çünkü ağın yapısı onun mantıksal diyagramından biliniyor. *Analizin görevi, verilen bireşimsel ağın ayrı ayrı mantıksal fonksiyonlarını açıklamaktır, amaç ise ağın ayrı noktalarında, giriş değişkenlerinin her kombinasyonu için mantıksal durumların belirlenmesidir.* Şimdiye kadar dikkat edilenden, analiz sorununun dijital cihazların kullanımı ve bakımı sırasında özellikle önemli olduğu ortaya çıkıyor.

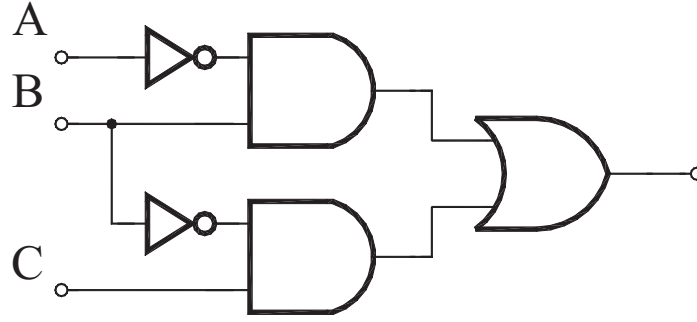
Ağın mantıksal diyagramı uygulanan mantıksal devrelerin uygun mantıksal sembollerin uygulanmasıyla çizilmelidir ve her mantıksal devrede girişlerin ve çıkışların arasında var olan ilişkiler belirtilmelidir. Her anahtarlama ağları için tüm giriş değişkenlerinin, yani doğrudan (gerçek, nominal) ya da tümleyen şekilde meydana gelebilen bağımsız değişkenlerinin isimleri ve çıkış değişkenlerinin, yani fonksiyonlarının tüm isimleri biliniyor.

Verilen ağın yapılan analizinden sonra fonksiyonun analitik şeklinin, yani ağın gerçekleştiği fonksiyonların elde edilmesi gerekiyor. Bu şekil mümkün olduğu kadar basitleştirilmeli ya da daha doğru minimize edilmelidir. Sonunda, gerekirse, elde edilen denklemden anahtarlama ağının doğruluk tablosu da yapılabilir.

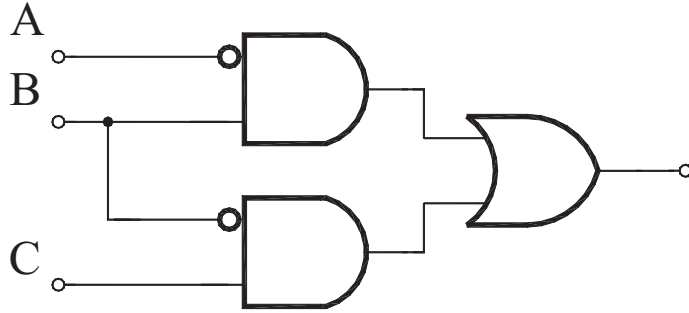
*Analiz ağın girişinden çıkışına doğru başlıyor ve mantıksal devrelerin tüm çıkışları belirtiyor ve her çıkış için mantıksal devrenin ne türden olduğuna bağlı olan uygun denklem yazılıyor.* Demek ki, tüm mantıksal devrelerin çıkışlarının anahtarlama fonksiyonlarının analitik şekilleri adım adım, ağın girişinden çıkışa doğru belirleniyor. Bu süreç ağın herhangi mantıksal devrelerin her çıkışı için ifade (her fonksiyonun) elde edilene kadar devam ediyor.

Ancak bunun ardından fonksiyonun basitleştirilmesiyle devam ediliyor. Son denklemden, yani fazla basitleştirilmeyebilen denklemden, aranırsa, ağın kombinasyon tablosu oluşturuluyor.

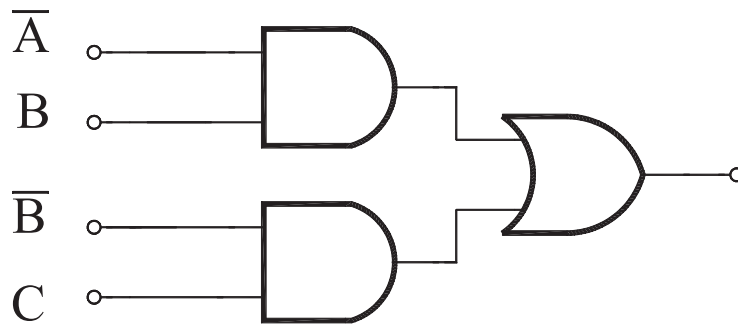
Şek.2-34 a), b) ve c)'de ilk bakıştan birbirinden farklı üç mantıksal diyagram gösterilmiştir. Fakat, bu diyagramları inceleyerek, her üç diyagramın  $Y(A, B, C) = \overline{A}B + \overline{B}C$  fonksiyonu gerçekleştiren anahtarlama ağının tanıttığını sonuca varılıyor. Fark sadece tümleşmenin gösteriş şeklinde vardır, öyle ki her diyagram geçerlidir ve elde edilen mantıksal fonksiyona göre, diğer iki diyagram gibi eşit olarak kullanılabilir. Bununla ilgili olarak, Şek.2-34 a)'da tümleme evirici ile tanıtılmıştır. Şek.2-34 b) biraz daha basittir çünkü küçük çemberle "o" işaretlenen tümleşmiş girişli devreler kullanılıyor. Şek. 2-34 c)'de işaretleme en basittir çünkü değerlerin tümleyenleri önceden elde edildiği tahmin ediliyor ve bu değişkenlerin üzerine çizgi " - " ile gösteriliyor.



a) eviricilerle mantıksal olumsuzluk (tümleme)



b) mantıksal devrelerin girişleriyle mantıksal olumsuzluk (tümleme)

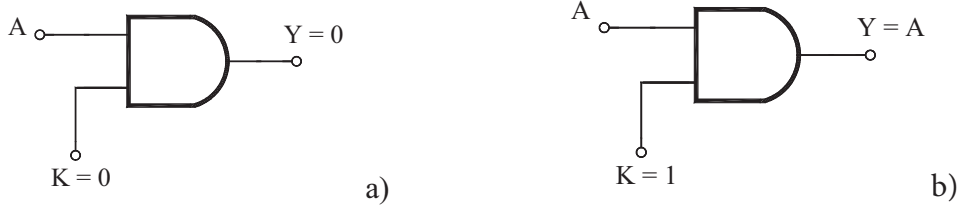


c) tümleşmiş giriş değişkenli mantıksal olumsuzluk (tümleme)

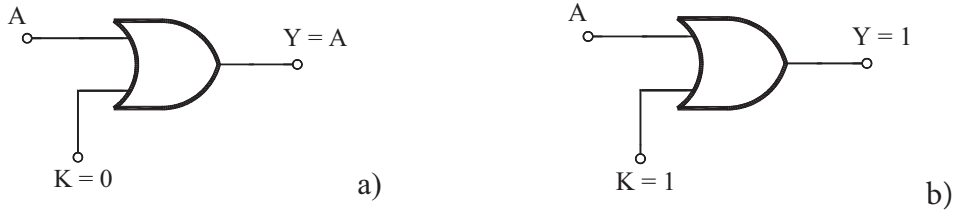
Şek.2-34.  $Y(A, B, C) = \overline{A}B + \overline{B}C$  fonksiyonunun mantıksal diyagramları

Burada, pratik gerçekleştirme bakış açısından her mantıksal diyagram farklı çözümlerle tanımlanacağını vurgulamak gerekiyor. Buna göre, Şek.2-34 a)'daki diyagramda iki evirici, iki VE devresi ve bir VEYA devresi kullanılıyor. Şek.2-34 b)'deki diyagram sadece üç mantıksal devre kullanıyor. Bunlardan birer giriş evirebilen iki OVE devresi ve bir VEYA devresi. Şek.2-34 c)'deki diyagram üç devre kullanıyor çünkü değişkenlerin tümlenmesi başka bir mantıksal yapıyla önceden yapılmıştır.

Sıradaki iki şekilde, Şek.2-35 ve Şek.2-36, devamda sıkça rastlayacağımız, sırasıyla VE ve VEYA devrelerinin en basit ikişer uygulama örnekleri tanımlanmıştır.



Şek. 2-35. Mantıksal VE devrenin analizi

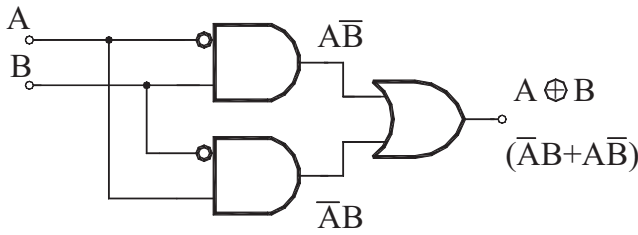


Şek. 2-36. Mantıksal VEYA devrenin analizi

Şek.2-37 ve Şek.2-38'de sunulan mantıksal diyagramların analizi ile şu iki mantıksal denklem elde ediliyor:

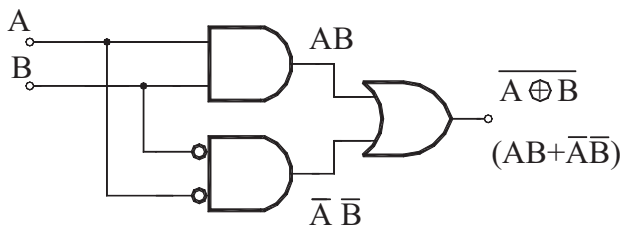
$$Y = \overline{A}B + A\overline{B} \quad (2-28)$$

$$Y = AB + \overline{A}\overline{B} \quad (2-29)$$



A	B	$\overline{A}B + A\overline{B}$
0	0	1
0	1	0
1	0	0
1	1	0

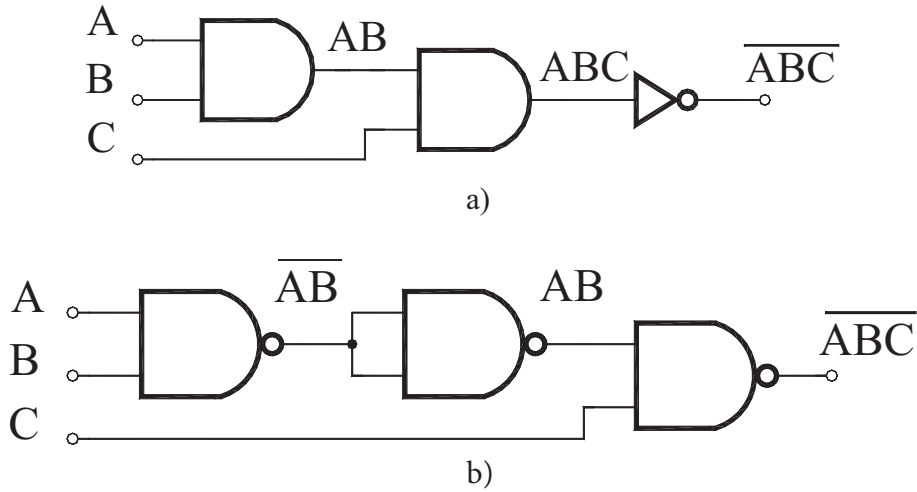
Şek. 2-37. D-YA mantıksal devrenin gerçekleştirimi



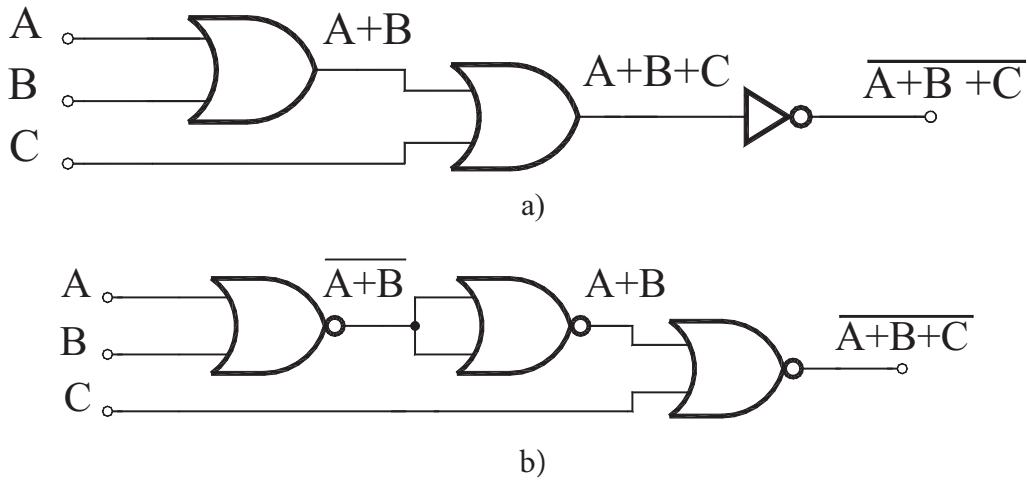
A	B	$AB + \overline{A}\overline{B}$
0	0	1
0	1	0
1	0	0
1	1	0

Şek. 2-38. D-OYA mantıksal devrenin gerçekleştirimi

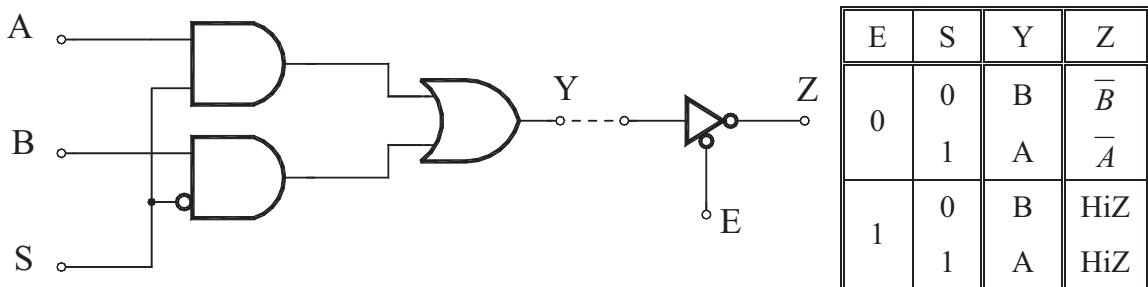
Bunların her biri için uygun kombinasyon tabloları doldurulursa ve bu tablolar D-YA ve D-OYA mantıksal devrelerin doğruluk tablolarıyla kıyaslanırsa, Şek.2-37 ve Şek.2.38'de verilmiş olan mantıksal diyagramlar aslında D-YA ve D-OYA fonksiyonlarını gerçekleştirdikleri kanıtlanıyor. Şek.2-39 a) , b)'de ve Şek. 2-40 a), b)'de üç giriş değişkenden bağlı olan OVE ve OYA mantıksal fonksiyonların gerçekleştiği dört basit ancak karakteristik mantıksal diyagram örneği daha verilmiştir. Şek.2-41 üç durumlu arabellek ile mantıksal diyagramın analiz örneği tanımlıyor.



Şek. 2-39. Üç girişli OVE mantıksal devrenin yapılımları



Şek. 2-40. Üç girişli OYA mantıksal devrenin yapılımları



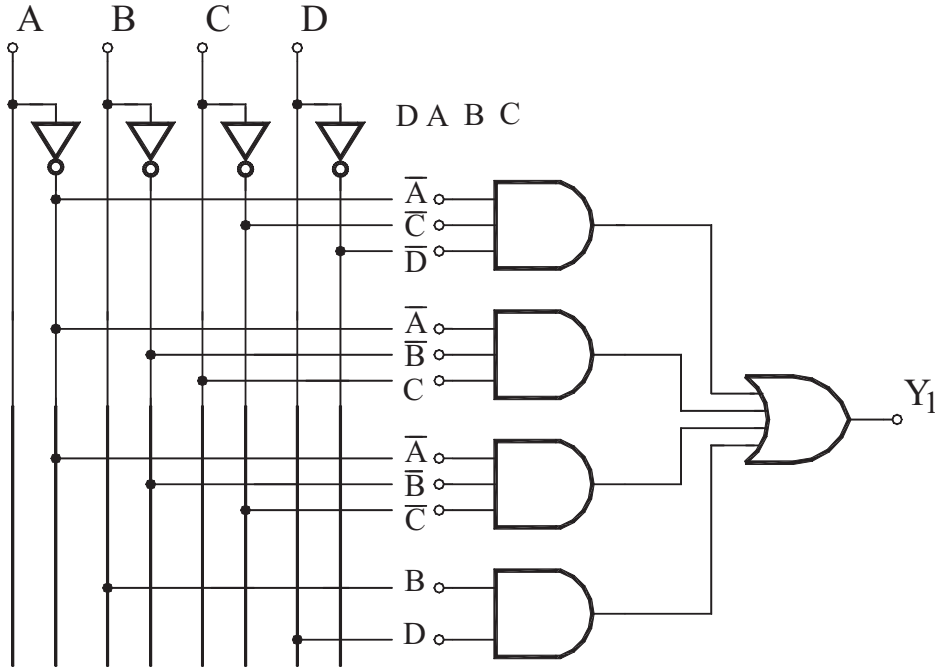
Şek. 2-41. Üçstatikli arabellek-evirici çıkışlı mantıksal diyagramın analizi

### 2.7.4. ANAHTARLAMALI AĞLARIN SENTEZİ

Bir anahtarlama ağının sentezi yapılması için, ağın belirtildiği anahtarlama fonksiyonunun bilinmesi gerekiyor. Buna göre, verilmiş bir normal biçimden ya da mantıksal fonksiyonun doğruluk tablosundan ağ fiziksel olarak gerçekleştiren mantıksal diyagramın elde edilmesi gerekiyor. Ağın oluşumu sırasında çok önemli ölçüt tabii ki kullanılan mantıksal devrelerin toplam sayısıdır. Tabii ki anahtarlama ağının edebildiğimiz kadar daha az mantıksal devreyle ve mantıksal devre başına edebildiği kadar daha az giriş sayısı ile gerçekleştirmeye çaba gösteriyoruz, böyle ki sentez sürecinde yapmamız gereken birinci şey verilen anahtarlama fonksiyonunun minimizasyonunu yapmaktır ve fonksiyonu MANB veya MBBN şeklinde getirmektir.

Bir mantıksal fonksiyon MANB şeklinde (en az çarpımlar toplamı) tanımlanmışsa, o zaman bu fonksiyonu gerçekleştiren anahtarlama ağ, bir VEYA devresine giriş olarak bağlanan, belirli sayıda VE devreden oluşuyor. VEYA devrenin çıkışından aranan fonksiyon elde ediliyor. Her giriş değişkeni önce birinci seviyeyi tanımlayan VE devresine gönderiliyor, ardından ise ikinci seviyeyi tanımlayan VEYA devresi aracılığıyla iletiliyor. Böyle yapı iki seviyeli VE-VEYA mantıksal sistemi olarak adlandırılıyor.

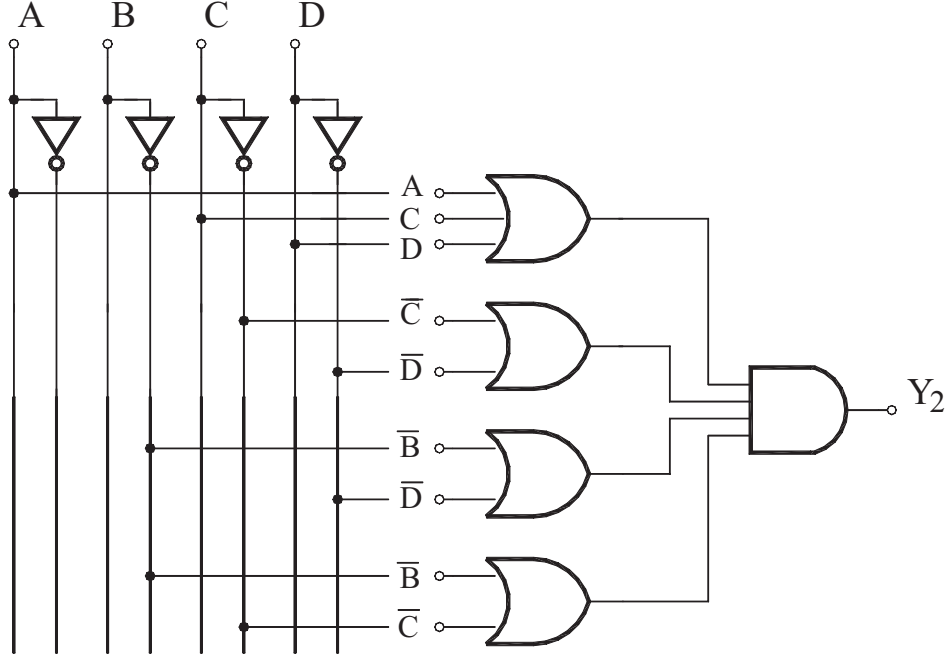
Şek.2-42'de  $Y_1 = \bar{A} \bar{C} \bar{D} + \bar{A} \bar{B} C + A \bar{B} \bar{C} + BD$  fonksiyonunu gerçekleştiren iki seviyeli VE-VEYA ağ örneği verilmiştir.



Şek.2-42. İki seviyeli VE-VEYA mantıksal yapı ile dört değişkenli fonksiyonun sentezi

Mantıksal bir fonksiyon MBBN (en az sayıda toplamlar çarpımı) şeklinde verilmişse, o zaman yine iki seviyeli mantıksal yapı elde edilecek, fakat bu yapı VEYA-VE türünden olacak. Buna göre, giriş devreleri VEYA devreleri olacak ve onlar birinci seviyeyi tanımlıyor, ikinci seviyede ise çıkış VE devresi olacak ve ondan aranan fonksiyon elde ediliyor.

$Y_2 = (A + C + D)(\bar{C} + \bar{D})(\bar{B} + \bar{D})(\bar{B} + \bar{C})$  fonksiyonu gerçekleştiren iki seviyeli VEYA-VE ağ örneği Şek.2-43'te verilmiştir.



Şek.2-43. İki seviyeli VEYA-VE mantıksal yapı ile dört değişkenli fonksiyonun sentezi

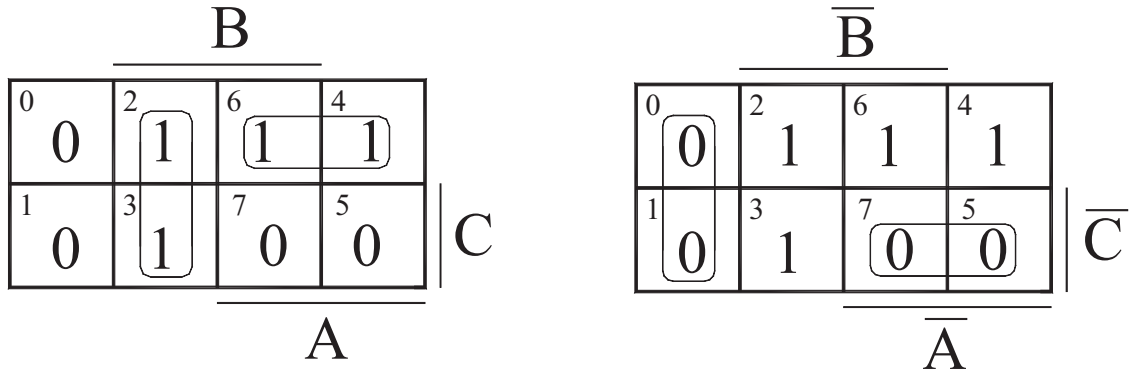
Önceki bölümlerde her anahtarlamalı fonksiyonun sadece OVE devrelerin ya da sadece OYA devrelerin uygulanmasıyla tanımlayabileceğini açıklamıştık. Bu iki fonksiyon teknik uygulaması için de daha kolaydır, öyle ki pratik nedenlerden dolayı da OVE ve OYA mantıksal devrelerin kullanımı tavsiye edilir. Sıradaki örneklerle, sadece OVE devrelerin uygulanmasıyla ya da sadece OYA devrelerin uygulanmasıyla mantıksal fonksiyonun fiziksel gerçekleşmesinin nasıl elde edildiğini açıklayacağız.

Üç değişkenli Z anahtarlamalı fonksiyonu inceleyelim:  $Y = Y(A, B, C) = \sum m(2, 3, 4, 6) = \prod M(0, 1, 5, 7)$ . Bu fonksiyonun kombinasyon tablosu Tab.2-26'da verilmiştir, KANB ve KBNB şekilde KK ise Şek.2-44 a) ve b)'de tanımlanmış. Fonksiyonun minimizasyonundan sonra, MANB şeklinde  $Y = \bar{A}B + A\bar{C}$ , olarak yazılabilir, ya da MBNB şeklinde  $Y = (A + B)(\bar{A} + \bar{C})$  olarak yazılabilir. Şek.2-45 a)'da gösterilen ikiseviyeli VE-VEYA yapısı, fonksiyonu MANB şeklinde gerçekleştiriyor, MBNB şekline göre ise Şek.2-45 b)'de gösterilen iki seviyeli VEYA-VE ağı elde ediliyor.

$i$	$ABC$	$Y$
0	000	0
1	001	0
2	010	1
3	011	1
4	100	1
5	101	0
6	110	1
7	111	0

Uygulamanın sadece OVE ve sadece OYA geçitleriyle nasıl elde edilebileceğini göstermek için, fonksiyonun MANB şekline ve MBNB şekline ayrıdan çift tümlenme gerçekleştireceğiz. Bununla aslında hiçbir şey değişmiyor, ancak bundan sonra De Morgan teoremini uyguladıktan sonra neyin elde edileceğini göreceğiz.

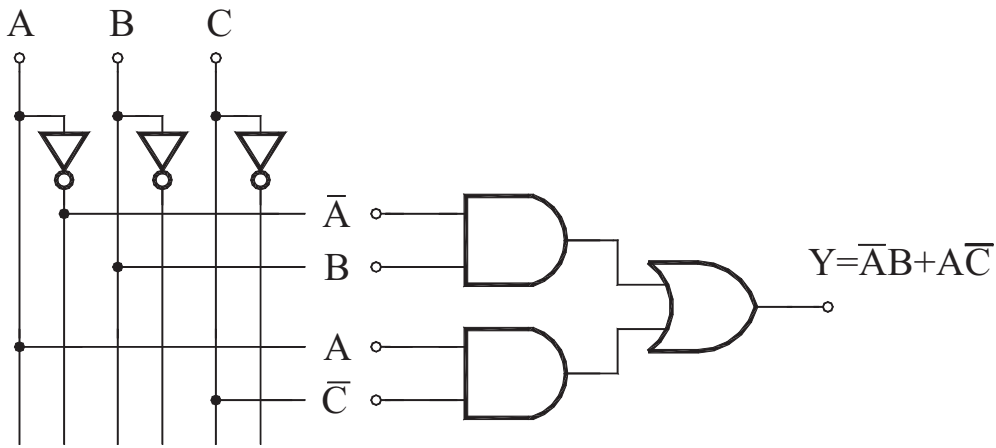
Tab. 2-26.  $Y(A, B, C) = \sum m(2, 3, 4, 6) = \prod M(0, 1, 5, 7)$  fonksiyonun doğruluk tablosu



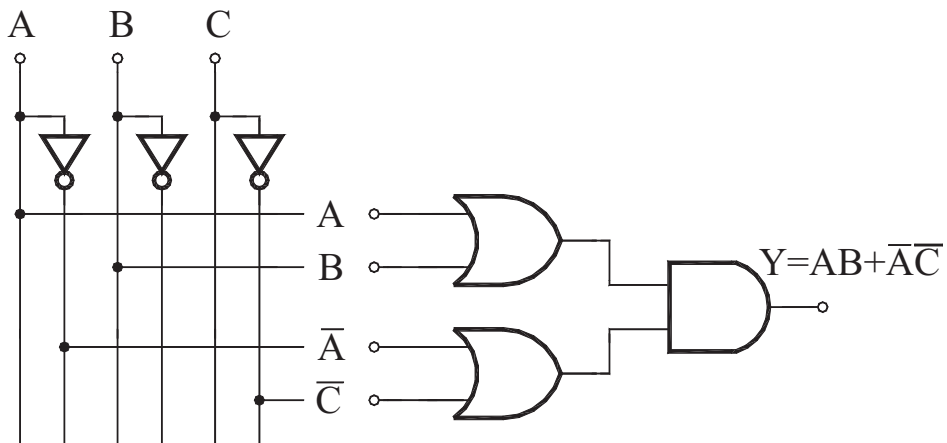
a) KANB şeklinde minimizasyon

b) KBNB şeklinde minimizasyon

Şek.2-44.  $Y(A, B, C) = \sum m(2,3,4,6) = \prod M(0,1,5,7)$  fonksiyonun Karno kartları



a) İki seviyeli VE-VEYA birleşimsel ağın gerçekleştirilmesi

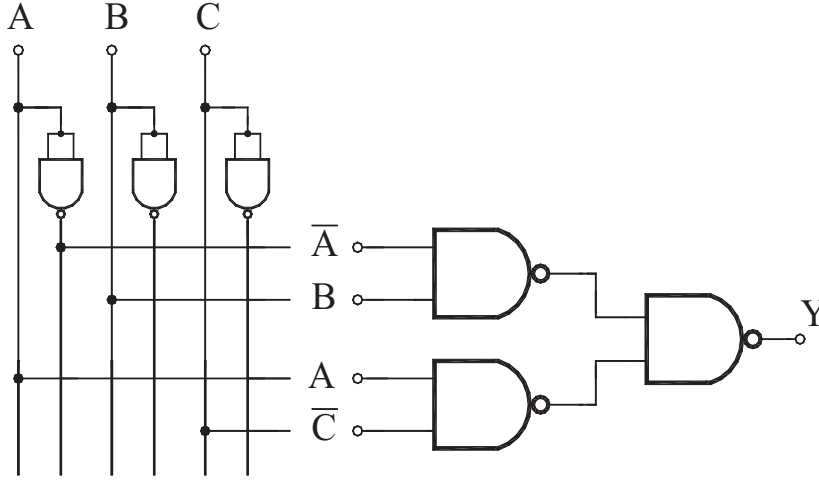


b) İki seviyeli VE-VEYA birleşimsel ağın gerçekleştirilmesi

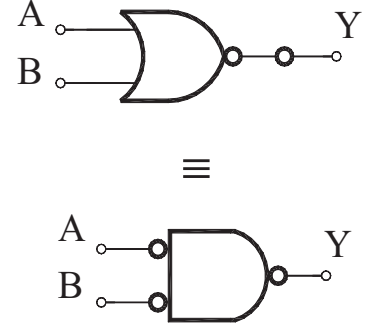
Şek. 2-45.  $Z(A, B, C) = \sum m(2,3,4,6) = \prod M(0,1,5,7)$  fonksiyonun iki seviyede sentezi



MANB'den çekerek  $Y = \bar{Y} = \overline{AB + AC} = (\overline{AB}) \cdot (\overline{AC})$  elde ediliyor. Şek. 2-46'daki dönüşümü uygulayarak, fonksiyonun elde edilen MANB şekli, Şek. 2-47'de gösterilmiş olduğu gibi sadece OVE devrelerin kullanımıyla gerçekleştirilebilir. Bu ağın yapılandırılması, Şek. 2-45 a)'da gösterilmiş olan yapıya aynı olduğu kolayca görülebilir, sadece her mantıksal devre OVE devreyle değiştiriliyor.

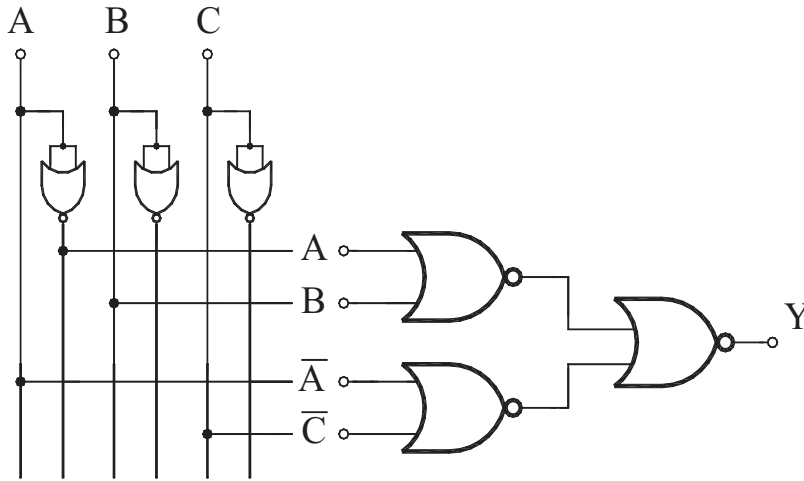


Şek. 2-47.  $Y = \sum m(2,3,4,6) (= \prod M(0,1,5,7))$  fonksiyonunun sentezi

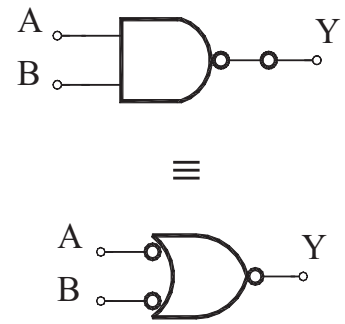


Şek. 2-46.

Benzer şekilde, fonksiyonun MBNB (Mantık Akıllı) şeklinden çekerek  $Y = \bar{Y} = \overline{(A + B) \cdot (\bar{A} + \bar{C})} = \overline{(A + B)} + \overline{(\bar{A} + \bar{C})}$  elde ediliyor. Fonksiyonun bu şekli, Şek. 2-49'dan görüldüğü gibi, sadece OYA (Or-Ya) geçitlerin kullanımıyla gerçekleştirilmiştir. Bu arada Şek. 2-48'deki dönüşüm kullanılmıştır. Bu durumda da elde edilen son yapılm ve Şek. 2-45 b)'deki mantıksal diyagramın, sadece ilk diyagramdaki tüm mantıksal devrelerin ikinci diyagramda OYA geçitleriyle değiştirildiğine göre farklıdır.



Şek. 2-49.  $Z = Z = \prod M(0,1,5,7) (= \sum m(2,3,4,6))$  fonksiyonunun sentezi



Şek. 2-48.

Yukarıdaki şekillerden görüldüğü gibi (2-22) denkleme uygun olarak,  $\bar{A} = \overline{A \cdot A}$ , Şek 2-47'de eviriciler, girişleri bağlı olan iki girişli OVE devreleri ile değiştirilmiştir. Benzer, (2-25) denklemini göz önüne alınarak,  $A = A + A$ , Şek.2-49'daki eviriciler girişleri bağlı olan iki girişli OYA devreleriyle değiştirilmiştir.

Tüm söylediklerimizden şöyle bir genel sonuca varabiliriz: sadece *OVE devrelerden oluşan ağın elde edilmesi* için, verilen fonksiyonun MANB'de ifade edilerek başlanıyor. Ardından fonksiyonun bu şekli için uygun iki seviyeli VE-VEYA mantıksal ağın yapılıp çizilmelidir. En sonunda tüm mantıksal devreler OVE devreleriyle değiştirilmelidir.

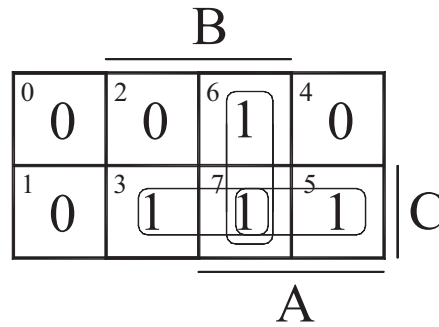
Buna benzer, *mantıksal ağın OYA yapılıp elde edilmesi* için, önce fonksiyon MBNB şekline tanımlanmalıdır. Ardından uygun iki seviyeli VEYA-VE ağ çiziliyor ve sonunda tüm mantıksal devreler OYA devreleriyle değiştiriliyor.

**Anahtalamalı ağların projelenmesi:** Bu bölümün sonunda, basit bir sorun çözen anahtalamalı ağın projelendiği süreci sunacağız. Üç üyeli jüri, en iyi yetenek yarışmasında şarkı söyleyen aday için önündeki düğmeye basarak bu adayın sonraki TV yayında devam edeceği ya da etmeyeceği kararı veriyor. Aday jüriden en az iki oy alırsa yeşil ışık yanıyor ve ışığın yanması bu adayın devam edeceğini gösteriyor. Ancak kırmızı ışık yanarsa, adayın tatmin edici kalite göstermediğini ve yarışmadan düştüğünü gösteriyor.

Önce tab.2-27 doğruluk tablosunu oluşturuyoruz. Jürinin bastırıldığı üç düğmeyi bağımsız (giriş) ikili değişken A, B ve C olarak göstereceğiz. Bu değişkenlerin 0 değeri düğmenin bastırılmadığını belirtiyor, 1 ise düğmenin bastırıldığını belirtiyor. Fonksiyon olarak, yani bağımsız olmayan (çıkış) değişkenleri olarak iki ışığı alacağız: yeşil IY ve kırmızı IK. Bu değişkenler için 1 değeri ışığın yandığını belirtiyor, 0 değer ise ışığın yanmadığını belirtiyor. Elde edilen tablodan, ışıkların karşılıklı tümlşik oldukları görülüyor ( $IK = \overline{IY}$ ). Buna göre bir fonksiyonun çözülmesi yeterlidir çünkü diğeri ilkin tümlenmesiyle elde ediliyor. Gereken hedefe ulaşmak için fonksiyonlardan birinin minimizasyonu ile devam ediyoruz, örneğin Şek. 2-50'den KK'ın uygulanmasıyla yeşil ışığı IY aktifleştiren fonksiyonu. Bu fonksiyonun minimum çözümünün şu MANB şekli vardır:  $SZ=AC+AB+BC$ . Sonunda birleşimsel (kombinasyon) ağını çizerek, verilen sorunu çözüyoruz.

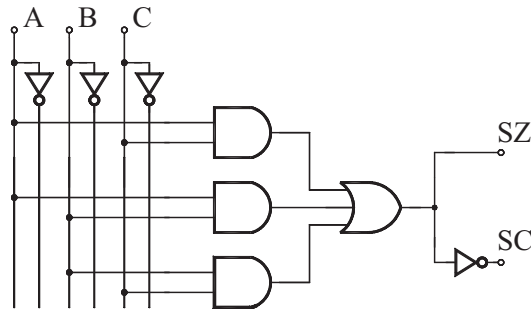
<i>i</i>	<i>ABC</i>	<i>SZ</i>	<i>SC</i>
0	000	0	1
1	001	0	1
2	010	0	1
3	011	1	0
4	100	0	1
5	101	1	0
6	110	1	0
7	111	1	0

Tab. 2-27. Doğruluk tablosu



Şek. 2-50. Minimizasyon

Şek. 2-51. Mantıksal diyagram



## TEKRARLAMA SORULARI VE ÖDEVLERİ

- 2-1. Boole cebri nedir?
- 2-2. Hantigton aksiomlarını say!
- 2-3. İkililik prensibi nasıl uygulanıyor?
- 2-4. Temel mantıksal işlemlerini say.
- 2-5. Mantıksal ifadelerin çözülmesi sırasında mantıksal işlemler hangi önceliğe göre hesaplanıyor?
- 2-6. Verilen mantıksal fonksiyonları doğruluk tablolarıyla ve mantıksal denklemlerle analitik şekilde tanımla: VE, VEYA, DEĞİL (tümleme), OVE, OYA, D-YA ve D-OYA.
- 2-7. (t. 2-12), (t. 2-14) ve (t. 2-15) teoremlerini şöyle ispatla (a) analitik yoluyla; (b) kusursuz endüksiyon (tümevarım) yöntemiyle; (c) genişleme teoremin uygulanmasıyla.
- 2-8. Verilen mantıksal ifadeleri analitik yoluyla basitleştir: (a)  $1 + \overline{AB} + \overline{ABC} + \overline{BC}$   
(b)  $0 + \overline{ABD} + \overline{BD} + C$ ; (B)  $1(BC + \overline{BC})$ ; (r)  $0(\overline{ABC} + \overline{AB} + BC + \overline{ABC} + \overline{AC})$ .
- 2-9. Şu mantıksal ifadeleri analitik yoluyla basitleştir: (a)  $\overline{(A + B + C)}(A + \overline{B})C$ ; (b)  $\overline{(A + B + C)}(A + \overline{B})$ ; (c)  $\overline{(\overline{AB})} + (\overline{ABC})$ ; (d)  $\overline{(\overline{AB})} + (\overline{ABC}) + C$ .
- 2-10. Herhangi bir anahtarlamalı fonksiyonun verilebildiği şekilleri say.
- 2-11.  $n$  değişkenli her  $Y$  mantıksal fonksiyonu için kombinasyon tablosunun görünüşünü detaylı açıkla.
- Satırların ve sütunların sayısı kaçtır? Onlarda ne giriliyor? İndis nedir ve hangi kapsam içinde değişiyor? Tablo satırları indislerle hangi prensibe göre numaralandırılıyor?
- 2-12. Her mantıksal fonksiyonun analitik şekilde verilebildiği normlaşmış (standartlaşmış) biçimleri say.
- 2-13. Fonksiyonun ANB şekli nasıldır? Bu biçim neyi tanımlıyor?
- 2-14. İmplikant nedir? Minterm nedir? KANB ne tanımlıyor?
- 2-15. Fonksiyonun BNB şekli nasıldır? Bu biçim neyi tanımlıyor?
- 2-16. İmplicent nedir? Maksterm nedir? KBNB ne tanımlıyor?

$i$	$ABC$	$F_1$	$F_2$	$F_3$
0	000	0	1	0
1	001	1	0	1
2	010	0	1	0
3	011	1	x	1
4	100	0	0	x
5	101	1	x	x
6	110	1	0	1
7	111	0	0	x

2-17. Tab.2-28 doğruluk tablosuyla üç değişkenli üç fonksiyon tanıtılmıştır:  $F_1(A,B,C)$ ,  $F_2(A,B,C)$  ve  $F_3(A,B,C)$ . Her fonksiyon için indisler kümesi yardımıyla onların KANB ve KBNB şekillerini yaz.

Tab.2-28. Ödev 2-17'den  $F_1$ ,  $F_2$ ,  $F_3$  mantıksal fonksiyonların kombinasyon tabloları.

- 2-18. Verilen fonksiyonlardan her biri için tüm mintermlerin ya da makstermlerin altını çiz, ondan sonra hangi fonksiyonların ANB, KANB, BNB ya da KBNB şeklinde verildiğini cevapla: (a)  $F_1(A, B, C) = ABC + \overline{A}\overline{B}\overline{C}$ ; (b)  $F_2(A, B, C) = \overline{A}\overline{B}C + \overline{A}B\overline{C} + \overline{C}\overline{B}$ ; (c)  $Y(A, B, C) = (A + B + \overline{C})(\overline{A} + \overline{B} + C)(\overline{A} + \overline{B} + \overline{C})$ ; (ç)  $Z(A, B, C) = (A + C)(\overline{A} + \overline{B})(B + \overline{C})$ .
- 2-19. Analitik ANB ve BNB şeklinde verilmiş şu fonksiyonlar için : (a)  $F_1(X_1, X_2, X_3) = X_1X_2\overline{X}_3 + \overline{X}_1\overline{X}_2$ ; (b)  $F_2(X_1, X_2, X_3) = (X_1 + \overline{X}_2 + X_3)(\overline{X}_2 + \overline{X}_3)$ ; (c)  $(A, B, C) = AB + \overline{A}C$ ; (ç)  $Z(A, B, C) = (A + C)(\overline{A} + B)$  (1) doğruluk tablolarını belirle; (2) indisler kümesi yardımıyla göster; (3) KANB ve KBNB şeklinde göster.
- 2-20. KANB ve KBNB şeklinde verilen fonksiyonları analitik yoluyla basitleştir: (a)  $Y(A, B, C) = \overline{A}\overline{B}\overline{C} + \overline{A}B\overline{C}$ ; (b)  $Z(A, B, C) = (A + B + C)(A + \overline{B} + C)(\overline{A} + \overline{B} + C)$ .
- 2-21. Aşağıdaki fonksiyonları analitik yoluyla ANB şeklinden KBNB şekline geçiti yap: (a)  $Y(A, B, C, D) = B\overline{C} + \overline{A}\overline{B}D$ ; (b)  $Y(A, B, C, D) = \overline{A}BD + CD$ .
- 2-22. Aşağıdaki fonksiyonları analitik yoluyla BNB şeklinden KANB şekline geçitini yap: (a)  $Z(A, B, C, D) = (\overline{B} + D)(A + B + \overline{D})$ ; (b)  $Z(A, B, C, D) = (A + B + D)(\overline{C} + \overline{D})$ .
- 2-23. İşlevsel tam mantıksal fonksiyonlar sistemi kümesinin içeriğinde hangi mantıksal fonksiyonlar giriyor?
- 2-24. Genel (evrensel) fonksiyonları say.
- 2-25. DEĞİL, VE ve VEYA temel fonksiyonlarını (a) OVE; (b) OYA fonksiyonlarla ifade et.
- 2-26. Fonksiyonların MANB ve MBNB minimum şekilleri için hangi özellikler geçerlidir?
- 2-27. Mantıksal fonksiyonların minimizasyonu hangi yöntemlerle gerçekleştirilebilir?
- 2-28. Verilen fonksiyonlar analitik (cebirsal) yoluyla minimize edilsin: (a)  $Y(A, B, C) = (A + B)(B + C)(\overline{B} + C)(\overline{B} + \overline{C})$ ; (b)  $Z(A, B, C) = AB + \overline{A}C + BC + \overline{B}\overline{C}$ .
- 2-29. (a) analitik yoluyla; (b) Karno yönteminin uygulanmasıyla şu fonksiyonlar minimize edilsin: (a)  $F_1(A, B, C, D) = \overline{A}BCD + \overline{A}\overline{B}\overline{C}D + \overline{A}BC\overline{D} + \overline{A}B\overline{C}\overline{D} + \overline{A}\overline{B}C\overline{D}$ ; (b)  $F_2(A, B, C, D) = (A + B + C + D)(\overline{A} + \overline{B} + C + D)(A + B + \overline{C} + \overline{D})(\overline{A} + \overline{B} + \overline{C} + D)(A + B + C + \overline{D})$
- 2-30. Karno kartlar yönteminin uygulanmasıyla, indisler kümesi ile verilen şu fonksiyonlar minimize edilsin:
- (a)  $Y(A, B, C) = \prod M(0, 1, 2, 4, 5)$
- (b)  $Y(A, B, C) = \prod M(0, 1, 4, 7)$
- (c)  $Y(A, B, C, D) = \prod M(0, 1, 2, 4, 5, 7, 8, 12, 13, 14, 15)$
- (ç)  $Y(A, B, C, D) = \prod M(0, 2, 4, 5, 6, 8, 10, 11, 14, 15)$
- (d)  $Y(A, B, C, D) = \prod M(0, 1, 2, 3, 7, 8, 10, 11, 12, 14)$
- (e)  $Y(A, B, C) = \sum m(0, 2, 6, 7)$
- (f)  $Y(A, B, C) = \sum m(0, 1, 3, 5, 6)$

$$(g) Y(A, B, C, D) = \sum m(0,3,4,6,7,8,11,12,13,15)$$

$$(h) Y(A, B, C, D) = \sum m(4,5,6,7,8,9,10,12,13,15)$$

$$(i) Y(A, B, C, D) = \sum m(0,2,4,5,6,7,9,11,13,15)$$

$$(j) Y(A, B, C, D) = \sum m(0,2,3,4,6,8,9,10,11,15)$$

2-31. Karno kartla yönteminin uygulanmasıyla, ANB ve BNB şeklinde verilmiş aşağıdaki fonksiyonlar minimize edilsin:

$$(a) F(A, B, C) = A\bar{B} + \bar{A}C$$

$$(b) F(A, B, C, D) = ABC\bar{D} + \bar{A}\bar{B}\bar{C}$$

$$(c) F(A, B, C, D) = \bar{A}BD + \bar{B}\bar{C}$$

$$(ç) F(A, B, C, D) = BCD + \bar{A}CD$$

$$(d) F(A, B, C, D) = (A + \bar{B} + C + \bar{D})(\bar{A} + \bar{C} + \bar{D})$$

$$(e) F(A, B, C, D) = (B + C + D)(\bar{A} + D)$$

$$(f) F(A, B, C, D) = (\bar{B} + \bar{C} + \bar{D})(\bar{A} + B + \bar{D})$$

2-32. Karno kartları yöntemini uygulayarak, aşağıdaki indisler kümesiyle verilmiş kısmen tanımlı fonksiyonlar minimize edilsin:

$$(a) Y(A, B, C) = \prod_{xM} M(0,5,7) \prod_{xM} M(1,4,6)$$

$$(b) Y(A, B, C, D) = \prod M(1,3,4,5,6,8,12,14) \prod_{xM} M(7,10,15)$$

$$(c) Y(A, B, C, D) = \prod M(1,2,8,10,11) \prod_{xM} M(0,3,4,5,15)$$

$$(ç) Y(A, B, C, D) = \prod M(0,2,4,5,6,11,15) \prod_{xM} M(8,10,14)$$

$$(d) Y(A, B, C) = \sum m(2,3,7) + \sum_{xm} m(5,6)$$

$$(e) Y(A, B, C, D) = \sum m(0,4,5,6,8,12,14,15) + \sum_{xm} m(1,2,10)$$

$$(f) Y(A, B, C, D) = \sum m(1,2,3,4,5,9,11,12) + \sum_{xm} m(10,13,15)$$

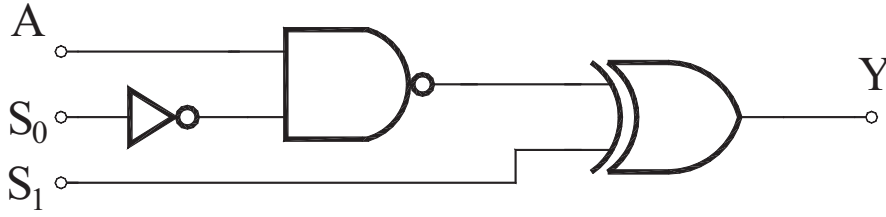
2-33. Şu mantıksal devrelerin mantıksal sembollerini çiz: (a) Üç girişli VE devresi; (b) Evirici; (c) İki girişli OVE devresi; (ç) Üç girişli OYA devresi; (d) İki girişli D-YA devresi; (e) İki girişli D-OYA devresi; (f) Arabellek devresi; (g) Üç durumlu arabellek; (i) Üç durumlu arabellek-evirici; (i) İkili (iletim) geçidi.

2-34. Üç durumlu arabellek ve ikili (iletim) geçidi arasındaki fark nedir?

2-35. Anahtarlamalı ağ nasıl oluşuyor?

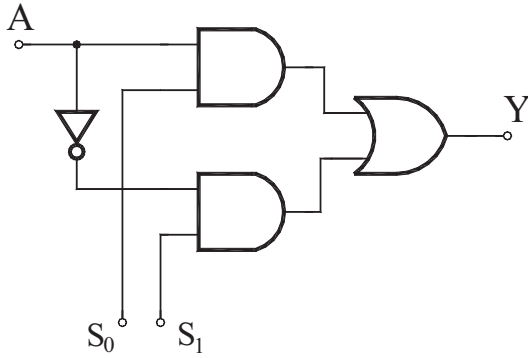
2-36. Birleşimsel ve ardışık anahtarlamalı ağ arasındaki fark nedir?

- 2-37. Şek.2.-52'de bir birleşimsel (kombinasyonel) ağ gösterilmiştir.  $S_1$  ve  $S_0$  girişlerin tüm kombinasyonları için doğruluk tablosunu çiz ve  $Y$  çıkış fonksiyonunu belirle. Elde edilen sonuçtan ağın davranışını yorumla.

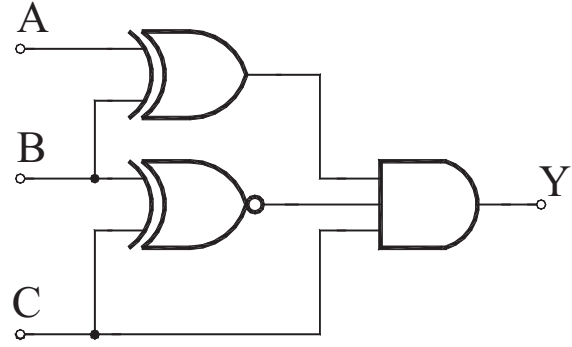


Ödev 2-37 şekli

- 2-38. Şek.2.53'teki birleşimsel ağ için,  $S1$  ve  $S2$  giriş değişkenler,  $Y$  çıkış değişkeni olmak üzere doğruluk tablosunun oluşması ve doldurulması gerekiyor. Doldurulmuş tabloya dayanarak ağın çalışmasını açıkla.
- 2-39.  $A, B$  ve  $C$  giriş değişkenlerin hangi kombinasyonu için, Şek.2-54'te gösterilen mantıksal ağın  $Y$  çıkışının değeri 1 olacaktır?

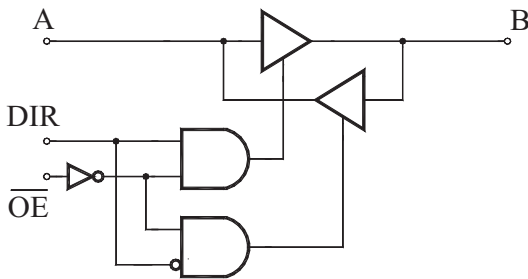


Ödev 2-38 şekli



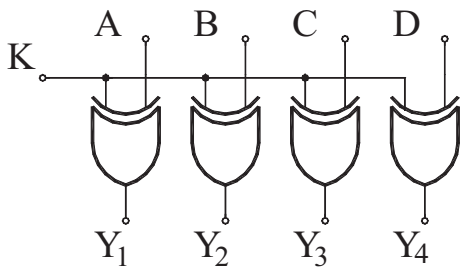
Ödev 2-39 şekli

- 2-40. Şek.2-56'da gösterilmiş birleşimsel ağ için eşliğinde olan kombinasyon tablosunun doldurulması gerekiyor, ardından ise bu ağın fonksiyonu ve pratik uygulanması belirtilsin ve açıklansın.



Kontrol sinyaller		Çıkışlar	
		A	B
0	0		
	1		
1	x		

Ödev 2-40 şekli



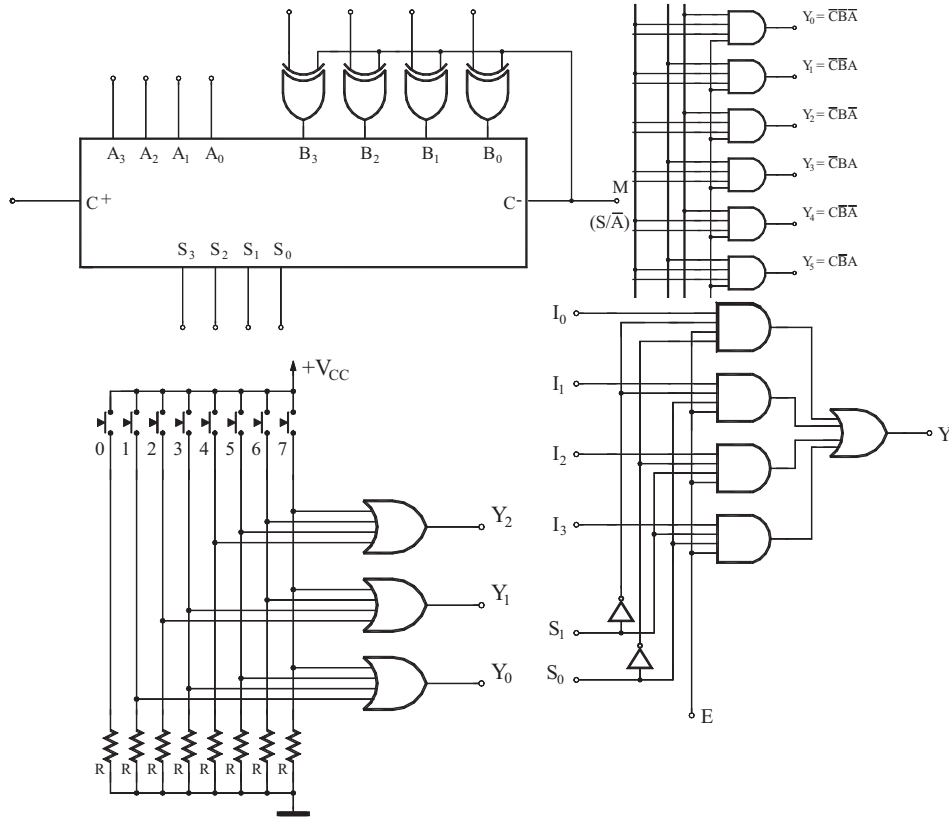
- 2-41. Şek.2-55'te gösterilen mantıksal ağ için,  $K$  değişkeninin bir kez 0 bir kez 1 değeri olduğunu tahmin edelim ve her durum için çıkış sinyallerin mantıksal seviyelerini belirle.  $K$  değişkeninin rolü nedir? Açıkla!

Ödev 2-41 şekli

- 2-42. İki girişli evirici, VE ve VEYA mantıksal devreleri sadece iki girişli (a) OVE; (b) OYA devrelerle gerçekleştirir.
- 2-43. İki seviyeli VE-VEYA ağın ve ardından OVE devrelerin yardımıyla şu fonksiyonlar gerçekleştirilsin:  
 (a)  $Y(A, B, C, D) = \bar{A}B + \bar{B}\bar{C}D + \bar{D}$ ; (b)  $F(A, B, C, D) = A\bar{B} + B\bar{C}\bar{D} + C$ .
- 2-44. İki seviyeli VEYA-VE ağın uygulanmasıyla ve ardından OYA devrelerle şu fonksiyonlar elde edilsin (a)  $Z(A, B, C, D) = (A + \bar{C})(\bar{A} + B + \bar{D})D$ ; (b)  $F(A, B, C, D) = (\bar{A} + B)(\bar{B} + C + D)\bar{D}$ .
- 2-45.  $Y(X_1, X_2, X_3, X_4) = X_1X_2 + X_1X_3X_4 + X_1X_2X_3X_4$  fonksiyonun sadece OVE mantıksal devrelerin kullanımıyla gerçekleştirilmesi gerekiyor, (\*) bu arada her devrenin sadece ikişer girişi olmalıdır.
- 2-46.  $Y(X_1, X_2, X_3, X_4) = (X_1 + X_2)(X_1 + X_2 + X_3)(X_1 + X_2 + X_3 + X_4)$  fonksiyonun OYA geçitlerin kullanımıyla gerçekleştirilmesi gerekiyor, (\*) bu arada her geçitin sadece 2'er girişi olmalıdır.
- 2-47. (\*)  $Y(X_3, X_2, X_1, X_0) = \sum m(0, 1, 2, 5, 7, 12, 13, 14)$  fonksiyonu KBNB şekline ifade edilsin, ardından minimize edilsin ve (a) OYA; (b) OVE mantıksal devrelerle gerçekleştirilsin.







# 3. BİRLEŞİMSSEL AĞLAR

Bu konusal birimini öğrendikten sonra

- ⊕ Aşağıdaki birleşimsel ağların mantıksal yapısını anlatacaksınız;
  - ⊕ toplama devresi ve çıkaram devresi;
  - ⊕ kodlayıcı, kod çözücü;
  - ⊕ çoğullayıcı, çoğullama çözücü;
- ⊕ Daha basit birleşimsel ağları analize edebileceksiniz;
- ⊕ Daha basit birleşimsel ağların çalışma şeklini açıklayabileceksiniz;
- ⊕ Daha karmaşık birleşimsel ağlar oluşturabileceksiniz;
- ⊕ Birleşimsel ağlarla ödevler çözebileceksiniz;



## I) ARİTMETİK-MANTIKSAL İŞLEMLER GERÇEKLEŞTİREN DEVRELER

### 3.1. GİRİŞ

Dijital teknikte karmaşık işlemlerin büyük bölümü dijital teknikte birkaç mantıksal devrenin uygun bağlanmasıyla gerçekleştirilebilir. *Ağda çıkış sinyallerin sadece giriş sinyallerin değerlerinin mevcut kombinasyonuna bağlı durumda elde edilen dijital ağlar birleşimsel (kombinasyonel) ağlar tanımlıyorlar.* Ağda çıkışların önceki durumların (değerlerin) onların sonraki durumlara etkilemiyor. Buna göre, birleşimsel ağların bilgileri koruma özelliği yoktur.

Birleşimsel ağların çok geniş kullanımları var ve neredeyse her bir dijital cihazda rastlanabilir. Bilgisayar sistemlerinde bu ağlar aritmetik veya mantıksal hesaplama işlemlerinde, belirli sayısal değerlerin üretiminde, bilgilerin kodlanması ve kod çözümlenmesi için, belleklerde adreslenmiş konumların bulunması için, belirli bağlantıların seçimi için ve benzer işlemlerde kullanılıyor. Bu ağların uygulama çeşitliliğinden dolayı, genelde gerçekleştirdikleri işleve göre adlandırılıyorlar. Örneğin: toplayıcı, çıkarıcı, tümleyici, karşılaştırıcı, kodlayıcı, kod çözücü, çoğullayıcı (seçici), çoğullama çözücü (dağıtıcı) vs. Birleşimsel ağların bazıları, bellek hücreleri içermediklerine rağmen, pratikte içeriği sadece okunabilen bellek bileşenleri olarak kullanılabilir şekilde düzenleniyor ve gerçekleştiriliyor.

### 3.2. TOPLAMA VE ÇIKARMA DEVRELERİ

Hesap makineleri ve bilgisayarlar gibi dijital cihazlarda, bilgilerin işletme süreci aritmetik ve/veya mantıksal işlemler gerçekleştiren özel devrelerin kullanımıyla yapılıyor.

Devamda iki temel hesaplama işlemleri: toplamayı ve çıkarmayı gerçekleştiren temel birleşimsel devreleri tanıtacağız. Çıkarmanın gerçekleşmesi sırasında sayıların tümleyen değerlere gerek olduğundan dolayı, mantıksal tümeleme işlemini gerçekleştiren devreyi de inceleyeceğiz.

#### 3.2.1. İKİLİ TOPLAYICILAR

Toplama devrenin temel aritmetik işlemlerin gerçekleşmesinde temel rolü vardır. Ayrıca dijital hesaplama makinenin, aritmetik birim rolü oynayan sadece bir toplama devreyle yapılabildiğini vurgulayalım. Diğer üç işlem programlama ile yapılabilir, yani işlemciye toplayıcının diğer üç işlemin gerçekleşmesi için kullandığı yönergelerin verilmesiyle yapılabilir. Önemi göz önüne alarak, önce toplayıcının gerçekleştirilmesine dikkat vereceğiz.

**Yarı Toplayıcı:** Mantıksal aritmetik bileşenlerde kullanılan temel modül *yarı toplayıcıdır* (İng. *half adder, HA*). Yarı toplayıcının rolü iki biti toplamaktır ve sonuç olarak onların toplamını vermektir, ancak aynı zamanda, tab.3-1 tablosunda verilen toplama kurallarına göre elde bitinin de üretimidir. Yarı toplayıcının doğruluk tablosu tab.3-2'de gösterilmiştir, mantıksal sembolü ise Şek.3-1'de verilmiştir.

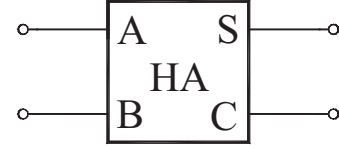
Şek.3-1'den yarı toplayıcının birer bit için iki girişi: A ve B ve iki çıkışı: bir çıkış toplam (tutar) biti için S ve bir çıkış elde biti için (İng. *carry*) C olduğu görülüyor.

0+0=0
0+1=1
1+0=1
1+1=0 ve elde 1.

Tab. 3-1. Aritmetik toplama kuralları

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

Tab.3.2. Yarı toplayıcının doğruluk tablosu

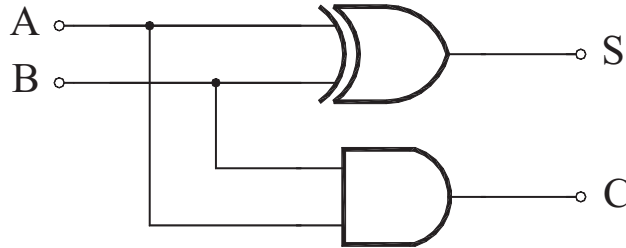


Şek. 3-1. Yarı toplayıcının mantıksal sembolü

A ve B bitleri birbirine eşitse, toplam biti ( $S = 0$ ) olacak: ya ikisi aynı zamanda 0 olursa ya da ikisi 1 olursa. Birinci durumda elde biti 0 olacak ( $C = 0$ ), diğer durumda ise elde biti meydana gelecek ( $C = 1$ ). Toplamın, sadece A ve B'nin karşılıklı tümleyen değeri olduğu zaman, değeri ( $S=1$ ) olacak. Bu arada her iki durumda elde olmayacak ( $C = 0$ ). Bu ilişkiler analitik şekilde şu mantıksal denklemlerle yazılabilir:

$$S = \overline{A}B + A\overline{B} \quad \text{ve} \quad C = AB \quad (3-1)$$

Bu denklemlerden, Şek.3-2'de gösterilen yarı toplayıcının uygulaması da ortaya çıkıyor. Toplam biti S, girişleri A ve B olan D-YA devresinden elde ediliyor, elde biti C ise aynı giriş bitleri A ve B ile VE devresinden oluşuyor.



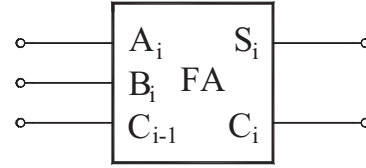
Şek. 3-2. Yarı toplayıcının mantıksal diyagramı

**Tam (komple) toplayıcı:** Yarı toplayıcı ile sadece bir bitli ikili sayılar toplanabilir. Çünkü onda giriş olarak önceki seviyeden, ya da daha az ağırlıklı pozisyonlardan toplama elde bitleri giriş olarak yoktur. Bu sorunu çözen devreye *tam (komple) toplayıcı* denir (İng. *full adder*, FA), mantıksal sembolü ise Şek.3-3'te gösterilmiştir. Şekilden görüldüğü gibi, tam toplayıcıda önceki ağırlık seviyesinden elde taşıyan ek girişi vardır. İşaretleme, tam toplayıcının her ikili sayıda belirli i-nci yerinde bulunan bitleri topladığını tahmin ettiğimizden belirtiliyor. Tam toplayıcının doğruluk tablosu Tab.3-3'te tanıtılmıştır.

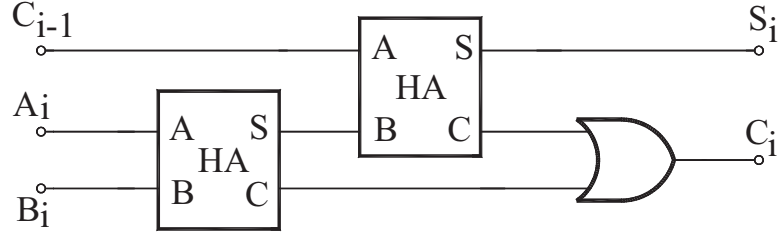
Tam toplayıcı, Şek. 3-4'te gösterilmiş olduğu gibi iki yarı toplayıcının ve bir VEYA devrenin kullanımıyla yapılabilir. Bu uygulama en ekonomik değildir ve bu yüzden tam toplayıcı, Tab. 3-3'te verilmiş olan doğruluk tablosunun doğrudan incelenmesiyle yapılarak, farklı pratik çözümler ortaya çıkıyor.

$A_i$	$B_i$	$C_{i-1}$	$S_i$	$C_i$
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

Tab.3-3. Doğruluk tablosu



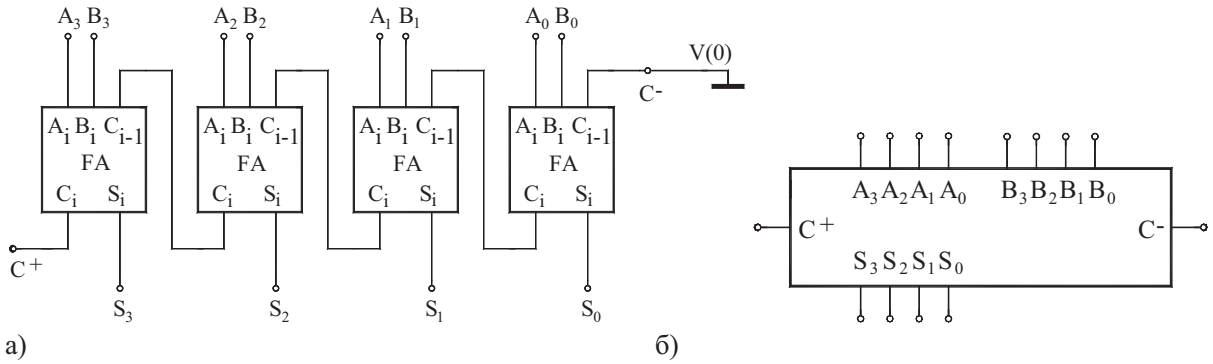
Şek. 3-3. Tam toplayıcının mantıksal sembolü



Şek. 3-4. Tam ikili toplayıcının mantıksal diyagramı

**İkili paralel toplayıcı:** Fazla tam toplayıcının uygulanmasıyla çok bitli sayılar toplayıcısı projelenebilir. Bir tam toplayıcıda sadece iki bitin sığabileceğini bilerek, kullanılan tam toplayıcıların sayısı toplanan sayıların uzunluğuna bağlı olduğu açıkça görünüyor.

Şek. 3-5'te 4 bitli pozitif sayılar (bir *nibble* uzunluğunda) için bir *paralel toplayıcı* gösterilmiştir. Bu uygulama 4 tam toplayıcı kullanıyor ve bu arada iki sayının en az değerli bitlerin (LSB-bitleri) toplandığı tam toplayıcının  $C_{i-1}$  girişinde mantıksal 0 değeri var. Bu bitlerin en alçak pozisyon seviyesinde bulunduğu nedeniyle, bu bitlerde önceki seviyeden elde taşınmaz çünkü öyle bir seviye yoktur. Bu iki bitin toplanması yarı toplayıcının kullanımıyla gerçekleşemez çünkü böyle durumda iki ya da fazla nibil uzunluğunda sayıların toplanmasının gerçekleşmesi için fazla böyle toplayıcının bağlanması mümkün değildir.



Şek.3-5. Paralel 4 bitli toplayıcının mantıksal yapısı ve sembolik işareti

İki sayının toplanması sırasında *taşma* (İng. *overflow*) olayının meydana gelmesi karakteristiktir. Taşma aslında iki en değerli bitin (MSB-bitlerin) toplanması sırasında meydana gelen eldedir. Üç biti toplayan tam toplayıcının (son tam toplayıcının)  $C_i$  çıkışında 1 meydana gelirse, elde edilen toplamın sadece dört bitle ifade edilemeyen sayı olduğu demektir, yani 4 bitle gösterilebilen sayı olan  $1111_{(2)} = 15_{(10)}$  sayısından daha büyük sayının elde edildiği demektir. Bu yüzden bu hat *taşmanın algılama hattı* (C+) olarak işaretleniyor.

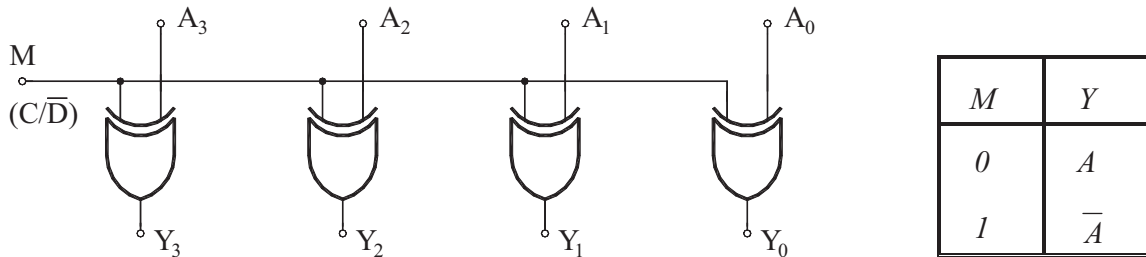
### 3.2.2. TÜMLEME DEVRESİ

Hesaplama işlemleri dijital cihazlarda sıkça olarak doğal ikili sayı sisteminde ya da ön işaretli sayıların gösterilmesi için kullanılan ikili sistemlerden birinde gerçekleşiyor. Buna göre, tümleme işlemi de hangi sistemin kullanıldığına bağlı olarak farklı mantıksal devrelerle ve yapıyla uygulanıyor. Prensipde, tümlenmiş değerler her iki sistemde, verilen bir A sayısının birinci ve ikinci tümleyeni için  $\bar{A} = K - A$  denklemin uygulanmasıyla elde ediliyor. Birinci tümleyenin sayı sisteminde en büyük sayıya kadar, ikinci tümleyenin ise sayı sisteminin sayılar kapsamına kadar tamamlanmasıyla ilgili olduğunu hatırlatalım. Buna göre ikili sayı sisteminde bire kadar tümleyen (tek tümleyen, 1's) ve ikiye kadar tümleyeni (çift tümleyen, 2's) vardır. Bunun dışında, sayıların tümleyen değerlerin ikili sayı sisteminde özellikle önemli olduklarını bir kez daha vurgulayalım, çünkü onlarla negatif sayılar tanımlanıyor.

*Tek tümleyen* sayının her biti onun tümleyen değeri ile değiştirilerek elde ediliyor. İkili sayının  $A=A_3A_2A_1A_0$  şeklinde verildiğini tahmin edelim ve örnek olarak göstermek için  $A = 1011$  sayısını alalım. Bu sayının birinci tümleyeni  $A_{(1s)} = 0100$  olacak. Birinci tümleyenin elde edilmesi basitçe evirici (DEĞİL) devrelerle gerçekleştirildiği açıkça görünüyor, çünkü  $A_{(1s)} = \bar{A}_3\bar{A}_2\bar{A}_1\bar{A}_0$ .

Bu işlemin dizisel şekilde gerçekleşmesi uygulandığı zaman, verilen sayının tümleyeni sadece bir eviriciyle elde edilebilir. Şöyle ki, eviricinin girişinde ardışık olarak sayının  $A_0, A_1, A_2, A_3$ , bitleri ayrıdan getiriliyor, eviricinin çıkışında ise onların tümleyen değerleri  $\bar{A}_0, \bar{A}_1, \bar{A}_2, \bar{A}_3$  elde ediliyor. Dizisel çalışma şekli oldukça yavaş olduğundan dolayı genelde verinin paralel işletmesi uygulanıyor.

Paralel süreç sırasında evirici devrelerin sayısı sözcüğün uzunluğuyla eşit olmalıdır, yani her ayrı verinin ifade edildiği toplam bit sayısı ile eşit olmalıdır. Tek tümleyenin elde edilmesi basitçe, Şek. 3-6'da gösterilen mantıksal diyagrama göre D-YA mantıksal devrelerin kullanımıyla gerçekleşiyor.



Şek. 3-6. Tümleme devrenin mantıksal diyagramı ve işlevsel tablosu

D-YA devreden her çıkış için şu denklem geçerlidir:

$$Y_i = M\bar{A}_i + \bar{M}A_i \quad (3-2)$$

Bu mantıksal fonksiyondan,  $Y_i$  tümleyicinin çıkışlarında  $M = 1$  için girişinde getirilen sayının her bitin ayrı tümleyen değerlerin elde edildiği ( $Y_i = \bar{A}_i$ ), bununla beraber de verilen A sayısının tümleyeni de  $Y = \bar{A} = A_{(1s)}$  elde edildiği ortaya çıkıyor.

Diğer taraftan, M kontrol hattına 0 getirilirse ( $M = 0$ ) tümleyicinin çıkışlarında A giriş sayısının bitleri elde ediliyor ( $Y_i = A_i$ ), bununla ise doğrudan (gerçek) şekilde sayı elde ediliyor ( $Y = A$ ).

Verilen ikili sayının *ikinci tümleyeni*, birinci tümleyene 1 ekleyince elde ediliyor. Böylece örneğin  $A = 0101$  sayısının birinci tümleyeni  $\bar{A} = A_{(1s)} = 1010$ , ikinci tümleyeni ise  $A_{(2s)} = 1011$  olacak. Buna göre, ikinci tümleyeni uygulayan ağın, birinci tümleyenin üretildiği D-YA devresi yanı sıra birinci tümleyeni 1 mantıksal sabitiyle toplamak için toplayıcının da kullanılması gerekecek.

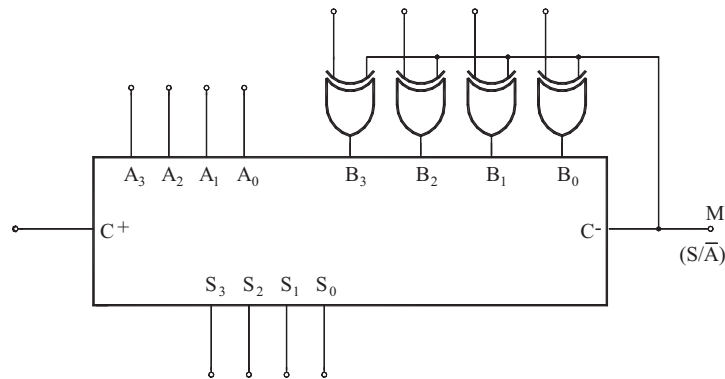
### 3.2.3. ÇIKARMA DEVRESİ

Ön işaretli ikili sayılarda uygulanması gereken aritmetik işlemler, genelde tümleştirici aritmetiğin uygulanmasıyla gerçekleşiyor. Bu arada sayıların doğrudan (gerçek) değerleri ve onların tümleyen değerleri uygulanıyor. Bu çalışma şeklinin belirli avantajları var, özellikle negatif sayılarla işlemler uygulandığı zaman. Negatif sayıların üç şekilde ifade edilebildiğini önceki derslerden biliyoruz: SM sisteminde ön işaretli, DC sisteminde birinci tümleyenle ya da RC sisteminde ikinci tümleyenle. Her üç tanımlama şeklinde en değerli – MSB bitin değeri 1-dir. Böylece, en değerli bitin bu değeri makine dilinde herhangi ikili sayının negatif değerini belirtiyor. *Negatif sayıların tümleyen şeklinde ifade edilmesi, çıkarma işleminin toplama yardımıyla gerçekleşmesini sağlayarak, işaret biti değersel bit gibi aynı şekilde işleniyor.*

Sayıların SM sisteminde doğrudan değerleri ve ön işaret biti ile ifade edildiği durumda, sayılar bir bütün olarak sadece aynı ön işaretleri olduğu zaman toplanabilir. Bu açıklama negatif sayıların tümleyenleri yardımıyla tanımlanmasının daha doğru olduğu yönündedir.

DC sisteminde birinci tümleyenle toplama ve çıkarma işlemleri oldukça basittir, çünkü sayının birinci tümleyeni kolayca elde ediliyor, ancak eldenin meydana gelmesi sırasında, elde edilen sonuçun elde 1 ile toplanması gerekiyor, öyle ki anılan işlemlerin uygulanması sırasında zorluklarla karşılaşabiliriz..

Diğer taraftan, RC sisteminde ikinci tümleyenin uygulanmasıyla toplama ve çıkarma işlemleri, önceki duruma kıyasen daha basittir, çünkü elde 1-in eklenmesi sonuca yapılmıyor, öyle ki ikinci tümleyenle çalışma çok daha yaygındır. Şöyle ki, pozitif sayıların çıkarılması azalan (çıkartılan) ve azaltanın (çıkartanın) ikinci tümleyeninin toplanması ile dolaylı olarak gerçekleşebilir. Dolaylı çıkarma süreci Şek. 3-7'de gösterilmiştir. Burada Şek. 3-5'teki paralel dört bitli toplayıcı ve Şek. 3-6'daki dijital tümleyici uygulanmıştır.



Şek. 3-7. Toplama ve çıkarma ağı mantıksal diyagramı

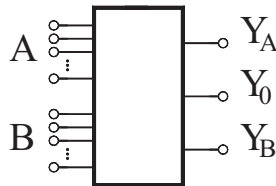
Bu yapılandırma,  $M (S/\bar{A})$  kontrol girişine alçak seviye getirilirse ( $M=0$ ) toplama olanağı sunuyor,  $M=1$  durumunda ise çıkarma olanağı veriyor. İlk dört bitli sayı  $A_i$  girişlerine doğrudan getiriliyor, ikinci sayı ise tümlene devresinin D-YA devreleri aracılığıyla  $B_i$  girişlerine götürülüyor. Gösterilen mantıksal yapı,  $M$  kontrol hattına alçak mantıksal seviye getirilirse ( $M=0$ ) toplayıcı olarak çalışacak, öyle ki D-YA devreleri dört bitli  $B$  toplayanı toplayıcının girişine değişmemiş-doğrudan şekilde iletiyor.

Fakat,  $M$  kontrol hattında yüksek mantıksal seviye getirilirse ( $M=1$ ), ağ çıkarma devresine dönüşüyor. Bu durumda, D-YA devrelerin çıkışlarında  $B$  azaltanın birinci tümleyeni oluşuyor, ancak  $M=1$  olduğundan dolayı toplayıcı bu 1'i ekleyerek  $B$  sayısını ikinci tümleyende ifade ediyor. Böylece tümleşik bileşenin girişlerinde  $A$  azalanı ve  $B$  azaltanın ikinci tümleyeni  $B_{(2s)}$  yani  $B$  sayısının negatif değeri bulunuyor. Böylece  $S_i, i=1,2,3,4$  çıkışlarında elde edilen ve  $S$  olarak işaretlenen toplama sonucu aslında  $A$  ve  $B$  sayıların farkıdır.

### 3.3. DİJİTAL KARŞILAŞTIRICI

Dijital cihazlardan sıkça iki ikili sayının karşılaştırılmasını yapmaları aranıyor, öyle ki sonuç olarak bir sayının diğer sayıdan daha büyük, eşit ya da daha küçük olduğu bilgisi elde ediliyor. Bu fonksiyonu uygulayan mantıksal ağa **dijital** veya **ikili karşılaştırıcı** denir, blok-diyagramı ise Şek. 3-8'de tanıtılmıştır.

Karşılaştırılan ikili sayıların, mantıksal bileşenin girişlerini tanımlayan eşit sayıda bitleri ( $n$ ) vardır. Sayılardan hangisinin daha büyük olduğu ya da eşit oldukları bilgisi işlevsel tab. 3-4'e uygun olarak üç çıkış hatı üzerinden elde ediliyor. Tablodan, iki sayının ilişkisine uygun olarak, çıkış hatlardan sadece birisinde matıksal 1'in meydan geleceği görülüyor.



Şek. 3-8. Dijital karşılaştırıcının mantıksal diyagramı

$A$	$B$	$Y_0$	$Y_A$	$Y_B$
$A = B$		1	0	0
$A > B$		0	1	0
$A < B$		0	0	1

Tab. 3-4. İşlevsel tablo

Sorunun esasını anlamak için, en basit durumu tahmin edeceğiz:  $A$  ve  $B$  bir bitli ikili sayıların karşılaştırılması. İşlevsel tabloya (tab.3-4) dayanarak, bu mantıksal bileşenin tab.3-5 olarak verilen doğruluk tablosu elde edilebilir.

$A$  bitini  $B$  bitiyle karşılaştırmayı gerçekleştiren bir bitli karşılaştırıcının mantıksal diyagramı Şek.3-9'da gösterilmiştir. Şekilde bir D-OYA mantıksal devresi dışında, iki evirici ve iki VE mantıksal devre de uygulanmıştır.

Mantıksal diyagramın ortasında en önemli olarak D-OYA devresi yer alıyor, çünkü bu devre aynı giriş kombinasyonlarını tanıyabiliyor. D-OYA devresinin bu özeliği doğruluk tablosundan (tab.3-5) açıkça görünüyor ve buradan D-OYA devresinin eşitlik algılayıcı rolü olduğunu kolayca sonuç olarak çıkarabiliriz, çünkü sadece her iki giriş biti birbirine eşit olduğu durumda devrenin çıkışı 1 olacak. Diğer taraftan, sadece bitlerin birbirinden farklı olduğu durumda çıkış 0 olacak.



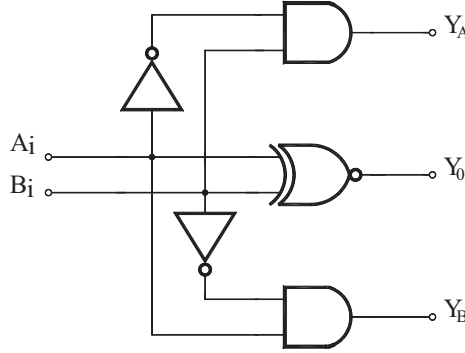
Açıklanan koşul şu mantıksal denklemle yazılabilir:

$$Y_0 = \overline{(A \oplus B)} = AB + \overline{AB} = \begin{cases} 1 \text{ eğer } A=B, \\ 0 \text{ eğer } A \neq B. \end{cases} \quad (3-3)$$

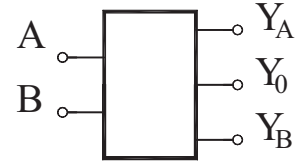
$A > B$  koşulu  $Y_A = A\overline{B}$  denklemiyle yerine getirilecek, çünkü sadece  $A=1$  ve  $B=0$  olunca  $Y_A=1$  elde ediliyor. Benzer şekilde,  $A < B$  koşulu  $Y_B = \overline{A}B$  denklemine uygun olarak yerine getirilecek. Bu durumda,  $Y_B=1$  sadece ve sadece  $A=0$  ve  $B=1$  olursa olacak.

A	B	$Y_0$	$Y_A$	$Y_B$
0	0	1	0	0
0	1	0	0	1
1	0	0	1	0
1	1	1	0	0

Tab. 3-5. Doğruluk tablosu



a)



b)

Şek. 3-9. Bir bitli karşılaştırıcının mantıksal diyagramı ve sembolü

Sunulan analiz ve mantık prensibi iki çok bitli sayının karşılaştırılma devrelerin gerçekleşmesi için gelişebilir ve genişlenebilir.

A ve B iki dört bitli ikili sayının karşılaştırıcısı olarak, Şek.3-7'deki toplama ve çıkarma dijital ağı kullanılabilir. Şöyle ki,  $A=B$  ise, tüm çıkışlar  $S_i$ ,  $i=1,2,3,4$ , alçak mantıksal seviyede bulunacak, yani 0 olacaklar ( $S_i=0$ ). Ancak, eğer  $A > B$  ise en yüksek seviyeden iletimde, yani (C+) çıkışında mantıksal 1 meydana gelecek,  $A < B$  sırasında ise bu çıkış hattında mantıksal 0 meydana gelecek.

## II) ANAHTARLAMALI MATRİSLER

### 3.4. GİRİŞ

Anahtarlamalı matrisler büyük sayıda girişli ve çıkışlı karmaşık birleşimsel mantıksal-anahtarlamalı ağlardır. Onların en önemli özelliği ağın herhangi çıkışındaki mantıksal durumun girişlerin mevcut mantıksal durumlardan bağlı olmalarıdır. **Anahtarlamalı matrisler dizilerde, satırlarda ve sütunlarda matris yapıları oluşturarak sıralanmış anahtarlamalı elemanlardan oluşuyor.** Bunun dışında, bu ağlarda gerçekleşen anahtarlamalı fonksiyonlar da matris şeklinde verilmiştir. Ancak, bu mantıksal ağlara çok daha sıkça işlevsel adlarıyla rastlanıyor. Bunlar arasında daha çok bilinenlerden kodlayıcıları, kod çözücülerini, çoğullayıcıları ve çoğullama çözücülerini anacağız

Anahtarlamalı matrisler mantıksal devrelerin uygulanmasıyla gerçekleşiyor, ancak çok daha sıkça olarak tümleşik devreler tekniğinde yapılmış komple çözümler de bulunabilir.

### 3.5. KODLAYICILAR VE KOD ÇÖZÜCÜLER

**Kodlayıcı** (İng. encoder) bir sayısal bilgiyi birincil olan bir sayı sisteminden başka bir ikincil sayı sistemine ya da koda kodlama (dönüştürme) gerçekleştiren mantıksal ağıdır. Birincil sayı sistemi olarak onlu sistem alınıyor, ikincil sistem ise genelde doğal ikili sayı sistemidir, ya da bazı ikili kodudur, örneğin doğal BCD kodu (NBCD), ya da 8421-kodu veya başka bazı kod.

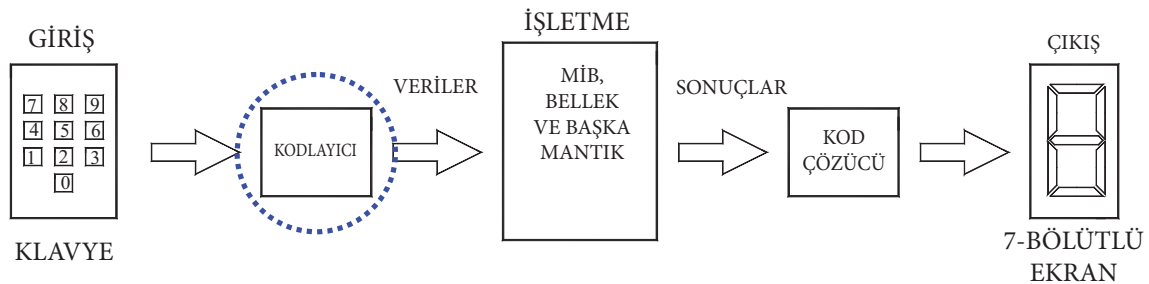
**Kod çözücüsü** de dönüştürme gerçekleştiriyor, ancak dönüştürmeyi ters yönde yapıyor. *Kod çözücüsü dijital bilgiyi ikincil sayı sisteminden birincil sayı sistemine çözüyor, yani ikili şekilde yazılmış olan veriyi onlu şekile dönüştürüyor.*

Bu iki birleşimsel ağ türü dışında, **kod çevirici** olarak adlandırılan mantıksal ağlar da vardır. Onlar bilgiyi bir ikincil sistemden başka bir ikincil sisteme ya da koda çeviriyor.

Zamanla ve dijital sistemlerin gelişmesiyle onlu sistemin her zaman birincil sistem olmadığı duruma gelmiş, öyle ki aynı bir mantıksal ağ bir sistem için kodlayıcıdır, bir sistem için kod çözücüsüdür, üçüncü sistem için ise kod çeviricidir. Bu yüzden ortak bir ismin kullanılması daha pratiktir: **kod çeviricisi** ya da **çevirmeni**. Fakat, pratikte yine de ilk tanıttığımız kodlayıcı veya kod çözücü klasik işlevsel isimler kullanılmaya devam ediliyor. Devamdaki bölümlerde kodlayıcının aslında VEYA mantıksal yapı olduğunu, kod çözücünün ise VE devreli mantıksal yapı olduğunu göreceğiz.

#### 3.5.1. KODLAYICI

Kodlayıcının incelenmesiyle başlamadan önce, dikkatimizi kısaca hesap makinesinin çok basit bir blok-diyagramla tanımlanmış olduğu Şek. 3-10'a yönlendireceğiz. Bu dijital sistemde klavyeden onlu girişin işlemcide giriş verisi olarak işletilmesi için ikili şekilde dönüştürülmesi gerekiyor.



Şek.3.10. Hesap makinesinin en basit blok-diyagramında kodlayıcının yeri

Onlu rakamların ifade edilmesi için sıkça kullanılan süreç iyi bilinen BNCD kodunda (8421 veya doğal ikili kod) kodlamadır, kod tablosu ise tab. 3-6 ile belirtilmiştir. Bu amaç için kullanılan dijital bileşen onlu-dan-NBCD-ye kodlayıcı (DEC/NBCD) olarak biliniyor. Bu kodlayıcının on girişi olmalıdır, her onlu rakam için birer giriş ve NBCD kod sözcüğünde D, C, B ve A dört bitin elde edildiği dört çıkışı olmalıdır.

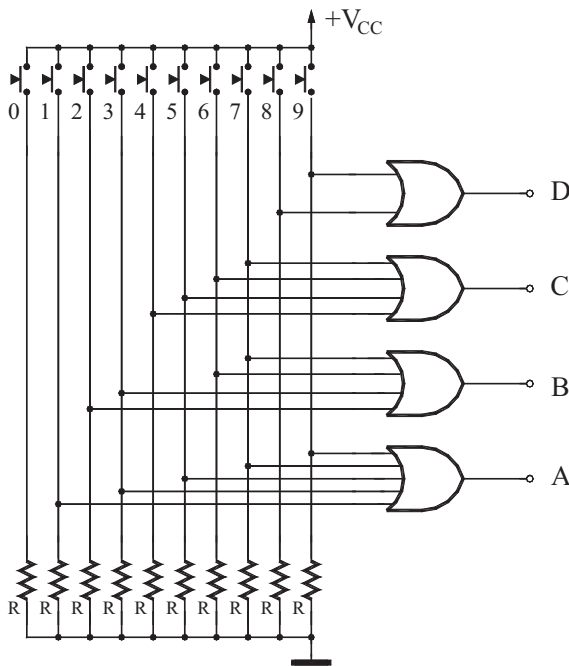
Kod matrisinin oluşmasında anahtarlamalı elemanların seçimini ve bağlanmasını NBCD kod sözcüğünün üçüncü bitin (B) elde edilme örneğini açıklayacağız. Tab. 3-6'dan B bitin değeri 1 (B=1) şu durumlarda elde ediliyor: 2 rakamın girişi, 3 rakamın aktifleştirildiği zaman ya da 6 rakamın meydana geldiği zaman ya da 7 rakamının hattı etkinleştiği zaman.

İncelemeden B çıkışının, 2, 3, 6 ve 7 hatları için dört girişi olduğu VEYA mantıksal devreyle gerçekleştiği açıkça görülüyor. Diğer üç bit: D, C ve A bitlerin elde edilmesi için de benzer mantık kullanılıyor ve sonuç getiriliyor

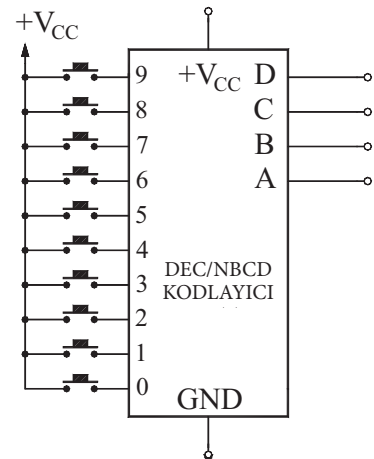
Onlu rakam	8421 (NBCD)			
	D	C	B	A
0	0	0	0	0
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0
5	0	1	0	1
6	0	1	1	0
7	0	1	1	1
8	1	0	0	0
9	1	0	0	1

Tab. 3-6. 8421 (NBCD) kodun kod tablosu

Şek. 3-11'de DEC/NBCD kodlayıcının bir uygulaması gösterilmiştir. Bu uygulama VEYA mantıksal devrelerden oluşan bir seviyeli matris yapının kullanımıyla yapılmıştır. Şekilden görüldüğü gibi, bu kodlayıcıda girişler yüksek seviyede aktiftir, çünkü tuşlardan birine bastırılınca karşılıklı giriş hattında mantıksal 1, yani  $+V_{CC}$  besleme gerilimi getiriliyor. Böylece belirli tuşun bastırılmasıyla giriş hatlardan bazılarında uygun VEYA devreleri etkinleşiyor ve onların çıkışlarında yüksek mantıksal seviye (0) elde ediliyor, VEYA devrenin etkinleştirilmemiş (uyandırılmamış) diğer çıkışlarda ise alçak mantıksal seviye (0) meydana geliyor. Bu şekilde, aktifleştirilen tuşta bulunan onlu rakama uyan NBCD kod sözcüğü elde ediliyor. Şek. 3-12'de bu şekilde uygulanan DEC/NBCD kodlayıcı ayrı mantıksal bileşeni olarak gösterilmiştir.



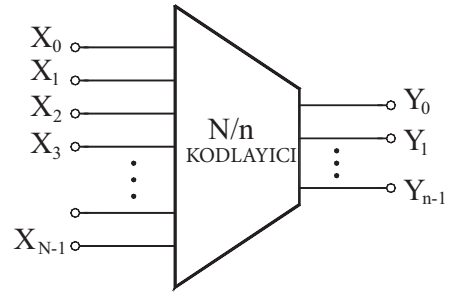
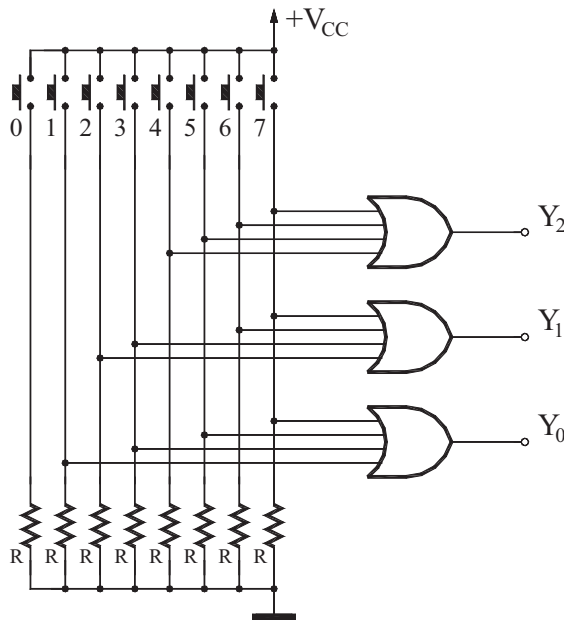
Şek.3-11. DEC/NBCD kodlayıcının mantıksal diyagramı



Şek.3.12 DEC/NBCD kodlayıcının blok diyagramı

Yukarıdaki şekillerden DEC/NBCD kodlayıcının 10 girişi ve 4 çıkışı olduğu görülüyor. 4 bit ile  $2^4=16$  farklı kombinasyonun kodlanabileceğini bilerek, DEC/NBCD kodlayıcıda 16 olası çıkış kombinasyonundan (durumdan) 6'sı kullanılmamış kaldığını ve onların ağ çıkışında hiçbir zaman meydana gelemeyecekleri söylenebilir.

Bunları göz önüne alarak, genel olarak  $N=2^n$  girişli kodlayıcının en çok  $n$  çıkışı olabileceği sonucuna varabiliriz. Bununla ilgili olarak, yukarıdaki mantıksal diyagrama on altılı sayı sistemin tablosuna göre altı giriş daha eklenebilir ve bu şekilde on altılı-dan-ikili-ye kodlayıcı (HEX/BIN) elde edilir. Benzer şekilde, sekizli-den-ikili-ye (OCT/BIN) kodlayıcı söz konusu olursa, sekizli sayı sisteminde ikili sayı sistemine dönüşüm tablosu kullanılabilir ve böylece Şek. 3-13'te yapısı gösterilen ve sembolik işaretinin Şek. 3-14'te verildiği mantıksal ağı elde edilebilir. Tab. 3-7 kombinasyonel doğruluk tablosu böyle bir kodlayıcının çalışma prensibini daha iyi açıklıyor.



Şek.3-13 OCT/BIN kodlayıcının mantıksal yapısı

Şek.3-14. OCT/BIN kodlayıcının sembolik işareti

Girişler								Çıkışlar			İndis
I <sub>0</sub>	I <sub>1</sub>	I <sub>2</sub>	I <sub>3</sub>	I <sub>4</sub>	I <sub>5</sub>	I <sub>6</sub>	I <sub>7</sub>	Y <sub>2</sub>	Y <sub>1</sub>	Y <sub>0</sub>	i
1	0	0	0	0	0	0	0	0	0	0	0
0	1	0	0	0	0	0	0	0	0	1	1
0	0	1	0	0	0	0	0	0	1	0	2
0	0	0	1	0	0	0	0	0	1	1	3
0	0	0	0	1	0	0	0	1	0	0	4
0	0	0	0	0	1	0	0	1	0	1	5
0	0	0	0	0	0	1	0	1	1	0	6
0	0	0	0	0	0	0	1	1	1	1	7

Tab.3-7. OCT/BIN kodlayıcının kombinasyon tablosu

Sekizliden-ikiliye kodlayıcının, her hattın birer sekizli rakamın tanımladığı  $I_0$ 'dan  $I_7$ 'ye kadar sekiz giriş hattı ve onlara eşdeğerli üç bitli ikili değerlerin elde edildiği  $Y_0$ ,  $Y_1$  ve  $Y_2$  olmak üzere üç çıkış hattı vardır. Kodlayıcının çalışma şeklini açıklayan mantıksal denklemler şunlardır:

$$Y_0 = I_1 + I_3 + I_5 + I_7; \quad Y_1 = I_2 + I_3 + I_6 + I_7; \quad Y_2 = I_4 + I_5 + I_6 + I_7; \quad (3-4)$$

Genel olarak, kodlayıcı için not edilmesi gereken şudur ki  $N=2^n$  giriş hattının herbiri için,  $n$  çıkış hattında kullanılan kod tablosuna göre tek bir ikili kod sözcüğün elde edilmesi gerekiyor.

Şek. 3-10 ve Şek. 3-12'ye göre uygulanan kodlayıcılarda sorun olarak, 0 rakamı için giriş hattının aktif olmuş gibi, çıkışta tüm sıfırlar elde ediliyor, halbuki hiçbir giriş hattı aktifleştirilmiş değildir. 0'ncı hattaki tuşun gerçekten aktifleştirildiği zaman da çıkış sonucu aynı olacak. Bu sorunun çözülmesi için kodlayıcıya çalışıyor ya da çalışmaz olmasını kontrol edebilen bit giriş hattı daha eklenebilir. Bunun dışında, çıkışta da bir hat daha eklenebilir ve giriş tuşlardan herhangi biri bastırılmış olursa bu ek çıkış hattının değeri 1 olacak, girişlerden hiçbiri aktif olmadığı zaman ise değeri 0 olacak.

Şimdiye kadar yaptığımız analizleri göz önüne alarak, sunulan kodlayıcıların uygulanması sırasında, verilen anda giriş hatlarında sadece birinin yüksek seviyede, ya da mantıksal 1'de bulunabileceğini tahmin ettiğimizi söyleyebiliriz. Bu yüzden, Şek.3-10 ve Şek.3-12'deki kodlayıcılarda iki ya da fazla tuşun aynı anda bastırılması, kodlayıcının çalışmasında hataya yol açacaktı ve geçerli olmayan (yanlış) çıkış kombinasyonların elde edilmesine neden olabilir.

### 3.5.2. ÖNCELİKLİ KODLAYICI

Yukarıda incelediğimiz kodlayıcıların iki ya da fazla girişin aktifleştirilmesiyle hataya yol açabileceği olumsuz tarafları, *öncelikli kodlayıcılar* ile giderilebilir. Pratikte, böyle mantıksal bileşenler sıkça olarak tümleşik teknikte gerçekleşiyor. Öncelikli kodlayıcılarda aynı zamanda iki ya da fazla tuş bastırılırsa, çıkışta bastırılan tuşlardan daha büyük sayısal değeri olan tuşuna uygun kod kombinasyonu elde edilecek. Buna göre, öncelikli kodlayıcıların daha değişik doğruluk tabloları vardır. Örneğin, OCT/BIN (sekizliden-ikiliye) öncelikli kodlayıcının tablosu, tab. 3-8'de verilen tablo şeklinde olacak.

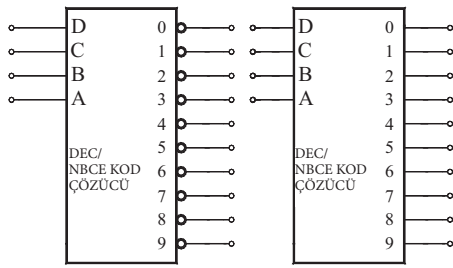
Girişler								Çıkışlar			
$I_0$	$I_1$	$I_2$	$I_3$	$I_4$	$I_5$	$I_6$	$I_7$	$Y_2$	$Y_1$	$Y_0$	V
0	0	0	0	0	0	0	0	x	x	x	0
1	0	0	0	0	0	0	0	0	0	0	1
x	1	0	0	0	0	0	0	0	0	1	1
x	x	1	0	0	0	0	0	0	1	0	1
x	x	x	1	0	0	0	0	0	1	1	1
x	x	x	x	1	0	0	0	1	0	0	1
x	x	x	x	x	1	0	0	1	0	1	1
x	x	x	x	x	x	1	0	1	1	0	1
x	x	x	x	x	x	x	1	1	1	1	1

Tab. 3-8. OCT/BIN öncelikli kodlayıcının kombinasyon tablosu

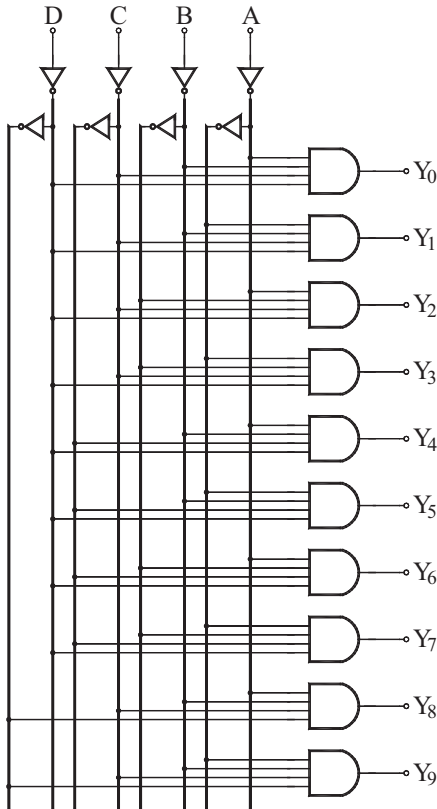
Bu tabloya baktığımızda, diğer girişlerin mantıksal durumları ne olursa olsun, çıkışta her zaman aktifleştirilmiş hatlardan en büyük sayısal değeri olan hatına uygun kod sözcüğünün elde edildiğini görebiliriz. Verilen doğruluk tablosunda V ile işaretlenmiş bir çıkış hattı da bulunuyor. Bu çıkış hattı giriş hatlardan bazısı aktif olup olmadığını gösteriyor. Bu çıkışta bulunan mantıksal değer sıfırdan giriş hattının aktifleştirilmiş (0-ncı tuşun bastırılmış olması) ve hiçbir girişin aktif olmadığı durumları arasında fark yapmayı sağlıyor.

### 3.5.3.KOD ÇÖZÜCÜ

Dijital işletimde tüm bilgilerin ikili şekilde ifade edildiğinden dolayı, bu verilerin insan tarafından kullanılması gerektiğinde verilerin onlu sayı sisteminde dönüşmeler gereksinimi anlaşılıyor. *İkili kodların, daha sonraki kullanımı için uygun şekile dönüşüm sürecine kod çözümlemesi denir, bu süreci uygulayan dijital bileşene ise kod çözücü denir.*



a) Sembolik işaret

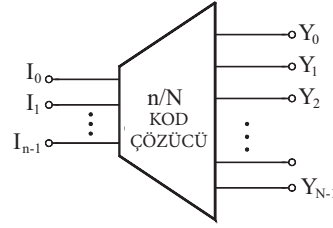


b) Mantıksal diyagram (yapı)  
Şek.3-15.NBCD/DEC kod çözücü

İkili verilerin onlu sayılara çözümlemesi kodlama sürecine ters bir süreçtir. Bu yüzden, Şek.3-15'te verilmiş olan NBCD-den-onlu-ya (NBCD/DEC) kod çözücünün blok-diyagramının on çıkışı ve dört girişi olacak. Çıkışlar  $Y_0$ 'dan  $Y_9$ 'a kadar şeklinde işaretlenmiştir, girişler ise D, C, B, A olarak işaretlenmiştir. Çıkışlar yüksek seviyede aktiftir. Buna göre mantıksal 1, sadece giriş NBCD kod sözcüğü verilen çıkışın ikili şeklini temsil eden sayısal değer olarak çıkışta elde edilecek.

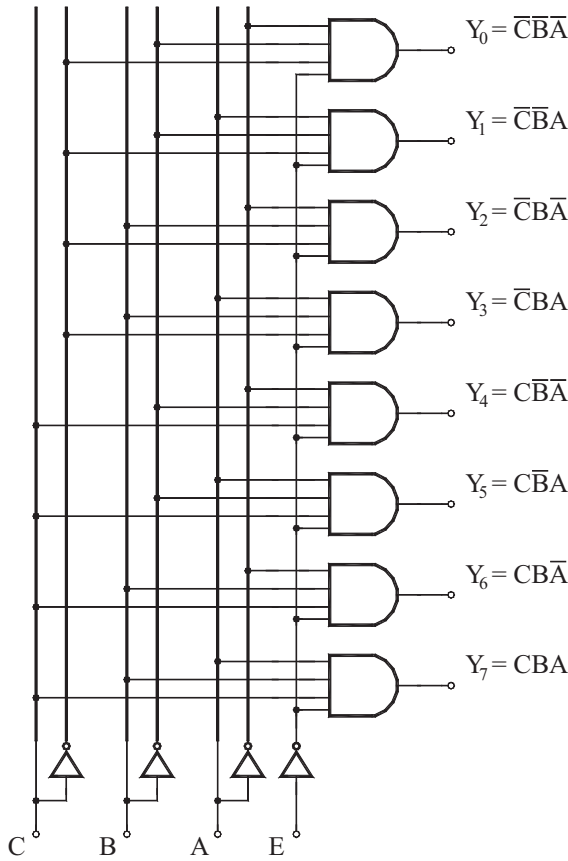
NBCD/DEC kod çözücünün bir uygulaması Şek.3-15 b)'de gösterilmiştir. Kod çözümleme matrisi şu analiz ile elde edilmiştir. Aktif, ya da mantıksal 1 yüksek seviyede, kod çözücünün girişinde getirilen NBCD olarak kodlanmış onlu rakama uygun çıkışın olması gerekiyor. Örneğin  $Y_6$  çıkışında 1 değeri, sadece girişte DCBA=0110 kombinasyonunun meydana geldiği, yani  $\bar{D}=1$  ve  $C=1$  ve  $B=1$  ve  $\bar{A}=1$  olduğu durumda olacaktır ( $Y_6 = \bar{D}CBA$ ). Buna göre, çıkış mantıksal devresi dört girişli VE devresi olacaktır ve bu arada C ve B doğrudan götürülüyor, D ve A ise tümleşik şekilde götürülüyor. Diğer dokuz rakamın bağlanması analog mantıkla elde edilerek sonuç olarak bir seviyeli VE matris yapısı elde ediliyor.

Şek. 3-15 b'deki kod çözücü yanlış bilgileri dışlıyor çünkü girişte, 1010, 1011, 1100, 1101, 1110 veya 1111 gibi geçerli olmayan kombinasyon, yani NBCD rakam tanımlamayan kod sözcüğü getirilirse, o zaman hiçbir çıkış hat aktif olmayacak. Pratikte yanlış kombinasyonu dışlamayan kod çözücü uygulamalara da rastlanabilir.

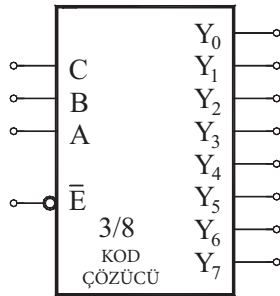


Şek. 3-16. Kod çözücünün mantıksal sembolü

4 bitle  $2^4=16$  farklı kombinasyonun kodlanabileceğini bilerek, NBCD/DEC kod çözücünde 16 olası çıkış kombinasyonundan (durumdan) 6'sı kullanılmamış kaldığı görünüyor. *Bununla ilgili, genel olarak n girişli kod çözücünün en çok  $N=2^n$  çıkışı olabileceğini söyleyebiliriz.*



Şek. 3-17. Yüksek seviyede aktif çıkışlı BIN/OCT kod çözücünün mantıksal diyagramı



Şek. 3-18. Yüksek seviyede aktif çıkışlı BIN/OCT kod çözücünün blok diyagramı

Böylece, örneğin önceki mantıksal yapıya on altılı sayı sistemin tablosundan altı giriş daha eklenebilir ve ikiliden-on altılıya (BIN/HEX) kod çözücü elde ediliyor. Benzer şekilde, ikiliden-sekizliye (BIN/OCT) kod çözücü söz konusu olunca, sekizli sayı sisteminden ikili sayı sistemine dönüşüm tablosu uygulanarak, Şek. 3-17'de verilen mantıksal yapı elde edilebilir, Şek. 3-18'de ise bu yapının blok-diyagramı verilmiştir. Tab. 3-9 kombinasyon tablosu böyle kod çözücünün çalışma prensibini daha detaylı açıklıyor.

İkiliden-sekizliye (BIN/OCT) kod çözücünün üç giriş hattı: C, B, A ve  $Y_0$ 'dan  $Y_7$ 'ye kadar sekiz çıkış hattı vardır. Kod çözücünün çalışma prensibinin açıklandığı mantıksal denklemler her çıkış hattı yanında verilmiştir. Her giriş kombinasyonu için tek olarak belirlenmiş (benzersiz) çıkış hattı vardır, öyle ki girişte belli bir kod sözcüğü gelirse, çıkışta giriş ikili eşdeğere uyan olan sayısal değer çıkış hattı aktifleştiriliyor. Böylece olası  $2^n$  giriş ikili kombinasyonlarda her biri, sekizli sayı sistemin tablosuna göre farklı çıkış hattının aktifleştirilmesine yol açıyor.

Pratikte gerçekleşen kod çözücüler genelde kod çözücünü çalışmasını sağlayan bir giriş hattı daha içeriyorlar. Bu, her VE çıkış devresi çalışmasına izin veren giriş hattına bağlanan birer giriş eklenerek yapılıyor.



Girişler				Çıkışlar							
$\bar{E}$	C	B	A	Y <sub>0</sub>	Y <sub>1</sub>	Y <sub>2</sub>	Y <sub>3</sub>	Y <sub>4</sub>	Y <sub>5</sub>	Y <sub>6</sub>	Y <sub>7</sub>
1	x	x	x	0	0	0	0	0	0	0	0
0	0	0	0	1	0	0	0	0	0	0	0
0	0	0	1	0	1	0	0	0	0	0	0
0	0	1	0	0	0	1	0	0	0	0	0
0	0	1	1	0	0	0	1	0	0	0	0
0	1	0	0	0	0	0	0	1	0	0	0
0	1	0	1	0	0	0	0	0	1	0	0
0	1	1	0	0	0	0	0	0	0	1	0
0	1	1	1	0	0	0	0	0	0	0	1

Tab. 3-9. BIN/OCT kod çözücünün kombinasyon tablosu

İzin (olanak) verme girişine  $\bar{E}$  (İng.enable) mantıksal 0 getirilince E aktiftir ve bu durumda kod çözücü normal çalışıyor. Ancak, bu girişe ( $\bar{E}$ ) yüksek mantıksal seviye, ya da 1 getirilince, hiçbir çıkış hattı, onunla beraber de hiçbir minterm seçilmiş olmayacak çünkü tüm çıkışlar 0 olacak.  $\bar{E}$  çalışma olanağı girişi olan kod çözücüler daha büyük sayıda girişli ve çıkışlı kod çözücülerin elde edilmesi amacıyla bağlanabilir.

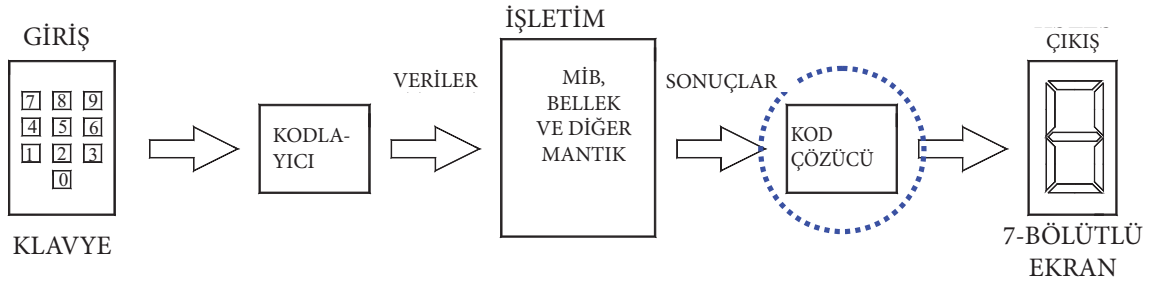
Çıkışların alçak seviyede aktif olmaları gerekirse, o zaman VE devrelerin OVE devrelerle değişmeleri gerekiyor, ya da her çıkışa birer evirici eklenerek her çıkış mintermi tümleşik şekilde verilecek. Böyle durumda, alçak seviyede (mantıksal 0) sadece giriş ikili kombinasyona uyan sayısal değeri olan çıkış olacak, tüm diğer çıkışlar ise 1 olacak.

Kod çözücülerini için söylenmesi gereken önemli özellik, mintermler toplamı olarak ANB şeklinde verilmiş her n-değişkenli mantıksal fonksiyonun, mintermleri üreten  $n$ -den- $2^n$  kod çözücünün ve onların toplamını oluşturan bir VEYA mantıksal devrenin uygulanmasıyla gerçekleştirilmesidir. Fonksiyonun mintermlerine uyan kod çözücünün çıkış hatları, VEYA devresinde giriş olarak kullanılıyor, devrenin çıkışında ise aranan mantıksal fonksiyon elde edilecek. Ayrıca, n-girişli ve m-çıkışlı olan her birleşimsel ağ, bir  $n$ -den- $2^n$  kod çözücüyle ve m VEYA mantıksal devreyle gerçekleştirilebilir. İlerleyen bölümlerde göreceğimiz gibi,  $n$ -den- $2^n$  mantıksal yapılı kod çözücülerin çok büyük kullanımları vardır ve ROM ve RAM bellek bileşenlerin yapılımlarında ayrılmaz parçadır.

### 3.5.4. NBCD-DEN-7 BÖLÜTLÜYE KOD ÇÖZÜCÜ

Pratikte *NBCD-den-7 bölütlü (NBCD/7S)* kod çözücüyle sıkça rastlanıyor. Bunun nedeni dijital cihazların en büyük kısmının sonuçları aydınlatılmış diyodlu ya da sıvı kristalli (ZED veya LCD) 7-bölütlü ekranlarda göstermelerinden kaynaklanıyor. Söylediğimizi hesap makinenin gerçekleştirdiği işletmenin elde edildiği sonucun gösterdiği şekilden basitçe görebiliriz. Hesap makinenin Şek. 3-19'daki blok-diyagramında, işlemciden NBCD kodlanmış çıkış verilerin çıkış 7-bölütlü ekranda her onlu rakamın gösterilebilecek şekilde çözülmesi gerektiği görülüyor.





Şek.3-19.Hesap makinesinin en basit blok-diyagramında kod çözücünün yeri

Ekranın ortak anodlu olduğunu tahmin edersek, o zaman NBCD/7S kod çözücünün girişleri alçak seviyede buldukları zaman aktif olmalıdır. 3-10 işlevsel tablosu kod çözücünün mantıksal ağının sentezinin temelidir ve D,C B ve A giriş değişkenleri olarak NBCD kod sözcüğün dört bitine bağlı olan yedi bölütle her birinin  $a, b, c, d, e, f, g$  fonksiyonların minimizasyonu ile elde edilebilir.

BCD 8421	Onlu rakam	LED ekran	Ortak anodlu ekran							
			a	b	c	d	e	f	g	
0000	0		0	0	0	0	0	0	0	1
0001	1		1	0	0	1	1	1	1	1
0010	2		0	0	1	0	0	1	1	0
0011	3		0	0	0	0	1	1	1	0
0100	4		1	0	0	1	1	0	0	0
0101	5		0	1	0	0	1	0	0	0
0110	6		0	1	0	0	0	0	0	0
0111	7		0	0	0	1	1	1	1	1
1000	8		0	0	0	0	0	0	0	0
1001	9		0	0	0	0	1	0	0	0

Tab. 3-10. LED diyodlu ve ortak anodlu yedibölütü ekranın kodlar tablosu

7-bölütlü ekranın ortak katotla bağlanması durumunda, NBCD/7S kod çözücünün çıkışları mantıksal 1 yüksek seviyede aktif olmaları gerekecek.

### 3.6. ÇOĞULLAYICI VE ÇOĞULLAMA ÇÖZÜCÜ

Çoğullayıcılar ve çoğullama çözücüler verilerin dizisel şekilden paralel şekile ve ters yönde dönüşümün yapıldığı cihazlarda kullanılıyor.

**Çoğullayıcının** dijital elektronikte çok durumlu anahtar rolü vardır. Şöyle ki, onun ikili veriler için çok sayıda girişleri ve sadece bir veri çıkışı vardır. Veriler herhangi bir girişe gelebilir, çıkışta ise onlardan sadece biri seçiliyor. Çıkışta hangi sinyalin meydana geleceği, çoğullayıcıda adres (seçici) girişlerin hangi durumda olduklarına bağlıdır. Çalışma şekline göre çoğullayıcılara **seçiciler** de denir.

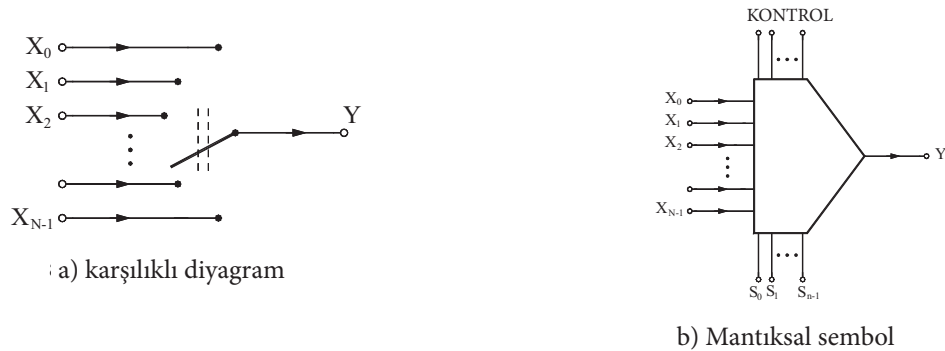
**Çoğullama çözücünün** ters görevi vardır: tek girişten gelen ikili veriyi, çok sayıda olan çıkışlardan birine aktarmalıdır. Bu durumda da verinin hangi çıkış hattında meydana geleceğini seçen adres hatları vardır. Çoğullama çözücünün çalışma şekline göre onlara **dağıtıcılar** da denir.

### 3.6.1. ÇOĞULLAYICI

Genel olarak, çoğullayıcının birçok giriş hatlarından ( $N$ ) birini seçen ve o hatta bulunan biti tek çıkışa ( $Y$ ) ileten anahtarlamalı ağ olduğunu söyleyebiliriz. Veriler için giriş hatları dışında, çoğullayıcıda giriş seçici hatlar ( $n$ ) olarak adlandırılan hatlar da vardır. Bu hat ile *mantıksal durumu çıkışa iletilmesi gereken giriş hattının seçimini yapıyor*. Verilerin giriş hatları sayısının ( $N$ ) ve seçme hatlarının sayısı ( $n$ ) arasındaki ilişki şu denklemlerle verilmiştir:

$$N = 2^n \quad (3-5)$$

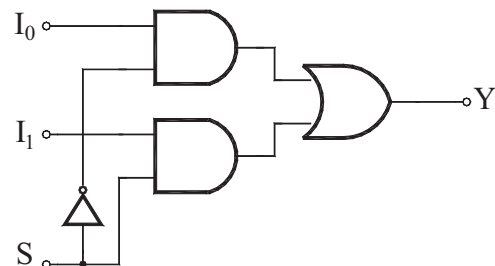
Bu şekilde, çoğullayıcının  $n$  adres hattı varsa, o zaman onlarla  $N = 2^n$  girişten birinin seçilmesini gerçekleştirebilir. Şek. 3-20 a)'da çoğullayıcının kontrollü anahtar olarak eşdeğerli diyagramı gösterilmiştir, Şek. 3-20 b)'de ise sembolü verilmiştir.



Temel mantıksal devrelerin kullanımıyla en basit çoğullayıcı uygulaması 2-den-1-e (iki durumlu) çoğullayıcıdır. Bu çoğullayıcının mantıksal diyagramı Şek. 3-21'de verilmiştir. Çıkış VEYA devresi iki sinyalden birini iletiyor, hangi sinyalin geçirileceği giriş VE devrelerini kontrol eden  $S$  girişinin durumu ile seçiliyor. Eğer  $S = 0$  ise,  $Y$  çıkışında  $I_0$  girişinden gelen sinyal geçirilecek, eğer  $S = 1$  ise, o zaman  $I_1$  girişinden sinyal geçirilecek.

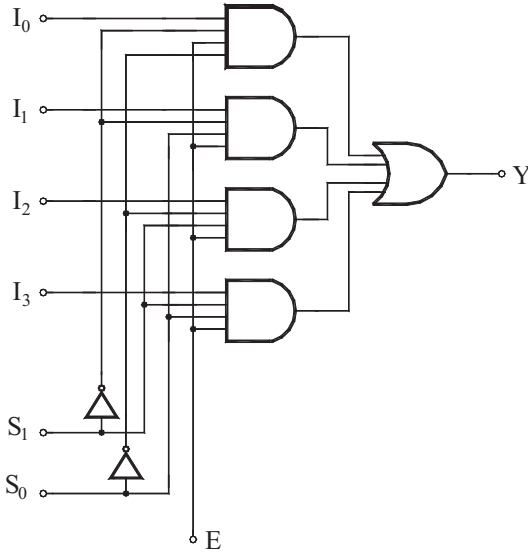
S	Y
0	$I_0$
1	$I_1$

Tab. 3.11. 2-den-1-e çoğullayıcının mantıksal tablosu



Şek. 3-21. 2-den-1-e çoğullayıcının mantıksal diyagramı

Şek. 3-22'de 4-ten-1-e (dört durumlu) çoğullayıcının mantıksal yapısı verilmiştir. Onun veriler için dört girişi, bir çıkışı ve iki adres hattı vardır. Çalışma prensibi tab. 3-12 doğruluk tablosuyla gösterilebilir.



$E$	$S_1$	$S_0$	$Y$
0	x	x	0
1	0	0	$I_0$
	0	1	$I_1$
	1	0	$I_2$
	1	1	$I_3$

Şek.3-22. 4-ten-1-e çoğullayıcının mantıksal diyagramı

Tab.3-12. 4-ten-1-e çoğullayıcının doğruluk tablosu

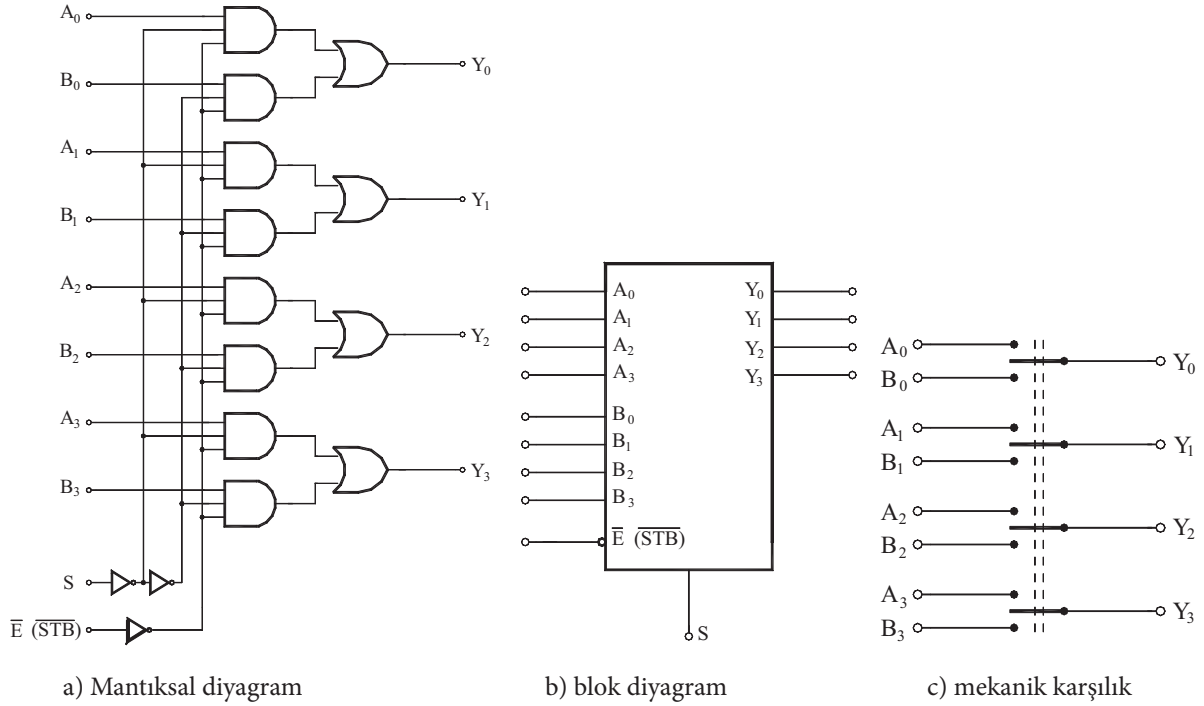
Gösterilen mantıksal diyagramda bir giriş sinyalinin de olduğu görünüyor. Bu sinyal  $E$  ile işaretlenmiş çalışma için olanak (izin) kontrol sinyalidir. Bu yönetim sinyali, monolitik tümleşik devreler olarak gerçekleşen çoğullayıcılarda vardır. Kontrol hattıyla aslında tümleşik devre açılıyor ya da kapanıyor. Şöyle ki, çoğullayıcı aktif olarak sadece  $E = 1$  olduğu zaman çalışıyor. Bu durumda çoğullayıcı çalışma izni alarak tab.3-12'deki tabloya göre normal olarak çalışıyor. Böylece, çoğullayıcının davranması seçici girişlerde getirilen giriş kombinasyonlarına bağlıdır:

- ⊕ Eğer  $S_1S_0 = 00$  ( $0_{DEC}$ ) ise,  $Y$  çıkışında  $I_0$  girişinde bulunan mantıksal seviye meydana gelecektir;
- ⊕ Eğer  $S_1S_0 = 01$  ( $1_{DEC}$ ) ise,  $Y$  çıkışında  $I_1$  girişinde bulunan mantıksal seviye meydana gelecektir;
- ⊕ Eğer  $S_1S_0 = 10$  ( $2_{DEC}$ ) ise,  $Y$  çıkışında  $I_2$  girişinde bulunan mantıksal seviye meydana gelecektir;
- ⊕ Eğer  $S_1S_0 = 11$  ( $3_{DEC}$ ) ise,  $Y$  çıkışında  $I_3$  girişinde bulunan mantıksal seviye meydana gelecektir;

Ancak,  $E$  yönetim girişine alçak seviye getirilirse ( $E = 0$ ), çıkış her zaman sıfırdır ( $Y = 0$ ). Ayrıca,  $E$  kontrol hattının, daha büyük sayıda girişli çoğullayıcılar oluşturma amacıyla birçok aynı yonganın bağlanmasını gerçekleştirebildiğinden dolayı, önemli rolü vardır.

Şimdiye kadar saydığımız çoğullayıcılar elde olan birçok veri girişinden birinin seçimini yapıyorlardı. Ancak, böyle çoğullayıcılar dışında, tümleşik teknikte giriş olarak meydana gelebilen çok sayıda giriş veri hatları grupları arasından bir veri hatları grubu seçen çoğullayıcılar da yapılıyor. Bu çoğullayıcıların, giriş gruplarından herbirinin içerdiği hatları sayısı kadar çıkışı var. Böyle bir çoğullayıcının mantıksal blok-diyagramı Şek. 3-12 a)'da gösterilmiştir, sembolü Şek. 3-23 b)'de verilmiştir, onun kontrollü çok durumlu mekanik anahtar olarak karşılığı ise Şek.3-23 c)'de tanıtılmıştır. Bu seçici matris dörtlü iki durumlu çoğullayıcı tanımlıyor, çünkü çıkışta dörder bilgi hattı içeren iki giriş grubundan birini veriyor.

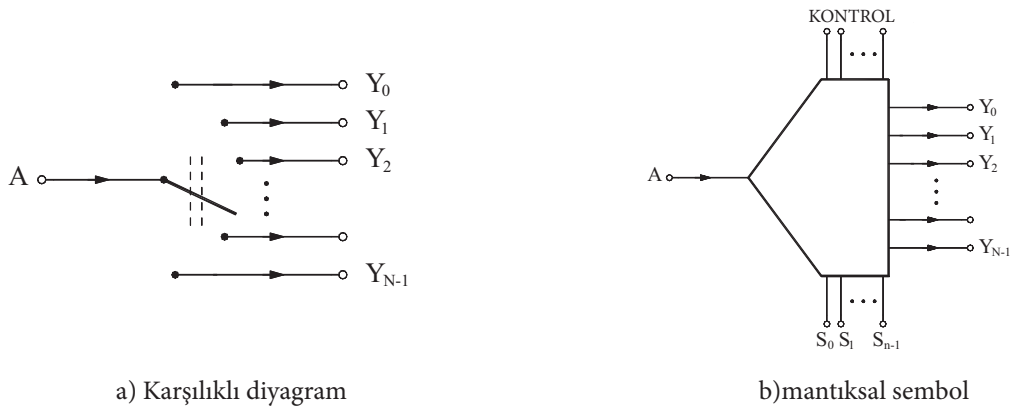
Devrenin sekiz girişi, dört çıkışı, bir seçici girişi ve bir kontrol girişi vardır. S girişi  $S=0$  ise, o zaman  $Y_0, Y_1, Y_2, Y_3$  çıkışlarında,  $A_0, A_1, A_2, A_3$  girişlerine bulunan veriler elde edilecek,  $S = 1$  olunca ise çıkışlarda  $B_0, B_1, B_2, B_3$  meydana gelecek. Tabii ki, bu arada  $\bar{E}$  veya  $\overline{STB}$  olarak işaretlenen kontrol girişi alçak seviyede olmalıdır ve  $\bar{E} = 0$  veya  $\overline{STB} = 0$  geçerli olmalıdır.



Şek. 3-23. Dörtlü iki durumlu çoğullayıcı

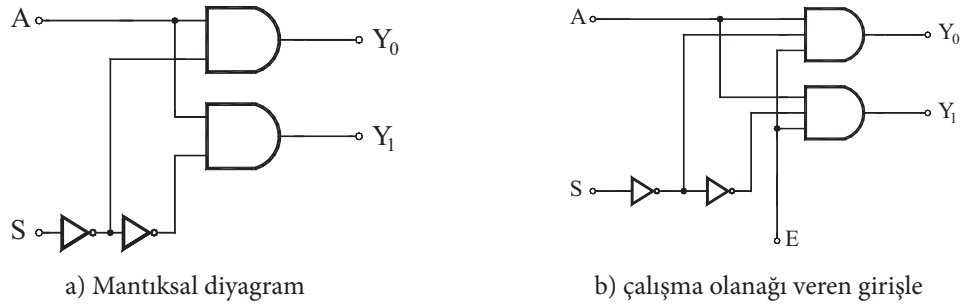
### 3.6.2. ÇOĞULLAMA ÇÖZÜCÜ

Çoğullama çözücüsü veriyi tek bir girişten alıp, varolan çok sayıda çıkışlardan birine iletir. Giriş bilgisini alması gereken çıkış, çoğullayıcıya benzer olarak seçim hatları yardımıyla seçiliyor. Çoğullama çözücü için de (3-5) denklemi geçerlidir, sadece burada  $n$  adres hattı ile  $N$  giriş hattından değil  $N$  ( $N = 2n$ ) çıkış hattından biri seçiliyor. Şek. 3-24 a)'da verilen şekil çoğullama çözücünün mekanik karşılığını gösteriyor, Şek. 3-24 b)'de ise çoğullama çözücünün sembolik işareti verilmiştir.



Şek. 3-24. Çoğullama çözücü

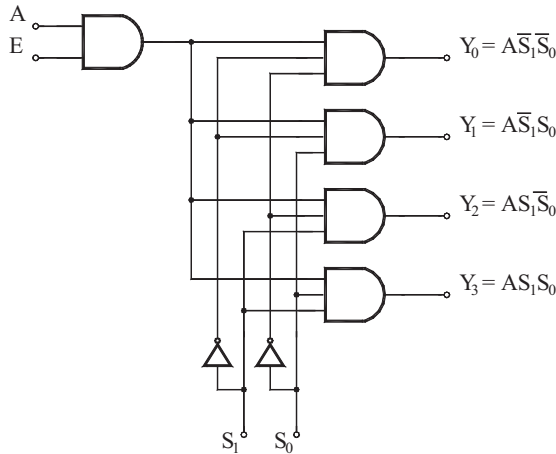
Çoğullama çözücünün en basit uygulaması, Şek. 3-25 a)'da gösterilmiş , o da 1-den-2-ye (iki durumlu) çoğullama çözücüsüdür. Seçici girişte alçak seviye varsa, yani  $S = 0$  ise, o zaman A giriş sinyali  $Y_0$  çıkışına aktarılacak. Ancak seçim hattına yüksek seviye getirirsek ( $S = 1$ ), o zaman giriş verisi  $Y_1$  çıkışında meydana gelecek. Verilen şekilden çıkış VE devrelerin sayısı çoğullama çözücünün çıkışlar sayısına eşit olduğu ve çıkışlardan birine aktarılması gereken giriş verisinin her anda herbir VE devrenin birer girişine getirildiği görülüyor. Mevcut VE devrenin açılması, bir VE devrenin girişine doğrudan, diğerinde ise evirici aracılığıyla tümleyen şekilde giren seçici hattın mantıksal durumu ile kontrol ediliyor. Bu şekilde aranan amaca ulaşılmıştır: verilen anda seçim girişi (S) uyarılınca iki devreden sadece birini açık olması, o da A girişinde bulunan bitin yönelik olduğu devrenin açık olması.



Şek.3-25. 1-den-2-ye çoğullama çözücü

Şek. 3-25 b)'de aynı çoğullama çözücünün, yüksek seviyede aktif olan E çalışma izinli (olanaklı) girişle uygulaması gösterilmiştir. E=1 olunca çoğullama çözücüsü normal çalışıyor, ancak E = 0 olduğu durumda tüm çıkışlar mantıksal 0 durumundadır.

Bu prensibi takip ederek, Şek. 3-26'da gösterilmiş olan 1-den-4-e (dört durumlu) çoğullama çözücünün mantıksal yapısı elde edilebilir. Bu bileşenin çalışma şekli en kolay olarak 3-13 doğruluk tablosu üzerinden açıklanabilir.



Şek.3-26. 1-den-4-e çoğullama çözücünün mantıksal yapısı

E	S <sub>0</sub>	S <sub>1</sub>	Y <sub>0</sub>	Y <sub>1</sub>	Y <sub>2</sub>	Y <sub>3</sub>
0	x	x	0	0	0	0
1	0	0	A	0	0	0
	0	1	0	A	0	0
	1	0	0	0	A	0
	1	1	0	0	0	A

Tab.3-13. 1-den-4-e çoğullama çözücünün kombinasyon tablosu

Bu çoğullama çözücünün de çalışma olanağı için girişi (E) var, sadece bir fark vardır ki burada çıkış VE devrelerin kontrolü dolaylı gerçekleşiyor. Bu çözümde iki girişli VE giriş devresi eklenmiştir: biri A verisi için, diğeri ise E izin girişi için. E=1 olduğu durumda, veri VE giriş geçidini geçiyor ve çıkış VE devrelerine aktarılarak normal çalışma sağlanıyor. Ancak E=0 olunca, giriş VE devresinin çıkışı 0'dır ve bu durumda çıkışları sıfır olan tüm çıkış VE devrelerini aynı anda etkinleştiriyor.

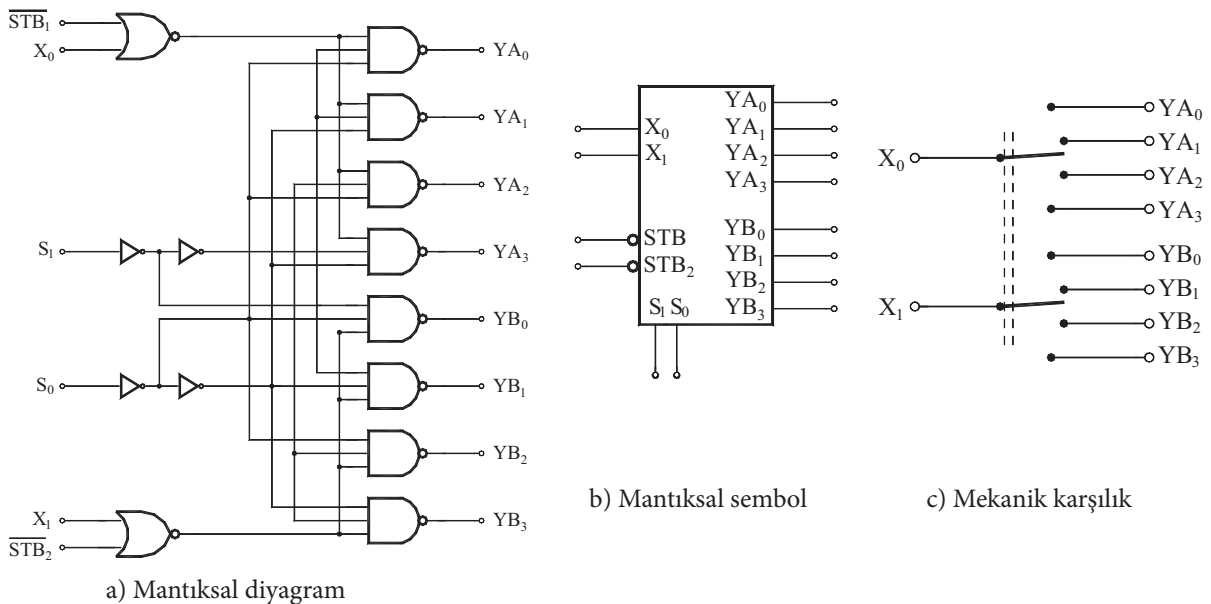
Kod çözücünün ve çoğullama çözücünün davranışını kıyaslayarak, çoğullama çözücüsünü girişi hattında sabit mantıksal 1 seviyesi getirilirse kod çözücü olarak kullanılabilir. Özel olarak, her 1-den- $2^n$  çoğullama çözücü kod çözücü olarak kullanılabilir. Eğer çoğullama çözücünün giriş hattında sabit 1 değeri eklenirse, n seçim hattı ise kod çözücünün n giriş hattı olarak kullanılıyor ve  $2^n$  çıkış hattı ise kod çözücünün çıkış hatları olarak kullanılıyor.

Buna benzer dönüşüm kod çözücünden çoğullama çözücüsüne yapılabilir. Bunun için kod çözücüsünün çalışma olanağı için girişi olmalıdır. Böylece, örneğin 2-den-4-e kod çözücüsü 1-den-4-e çoğullama çözücüsü olarak kullanılabilir. Kod çözücünün izin (olanak) girişi çoğullama çözücüsünde veri girişidir, kod çözücünün iki giriş hattı çoğullama çözücünü seçim hatları olarak alınıyor, dört çıkış ise her iki bileşen için ayrıdır.

Tümleşik teknike yapılan çoğullama çözücüleri, veri ve adres hatları dışında, kontrol hatları da içeriyor. Kontrol hatları bileşeni çalışma için devreye koyabilir ya da daha büyük sayıda çıkışlı bileşenlerin elde edilmesi için aynı çoğullama çözücüleriyle bağlayabilir

Sayıdığımız tüm çoğullama çözücüleri tek girişten elde olan birçok çıkışa iletim gerçekleştiriyordu. Ancak, böyle çoğullama çözücüleri dışında tümleşik teknikte bir giriş veri hatları grubunu, olası çıkış grupları olarak ortaya çıkan birkaç böyle gruptan birine dağıtım yapan çoğullama çözücüleri de vardır. Bu çoğullama çözücüleri ayrı çıkış grubun içerdiği hatlar kadar girişleri vardır.

Böyle bir çoğullama çözücünün mantıksal blok-diyagramı Şek.3-27 a)'da, sembolü Şek.3-27 b)'de, anahtar olarak mekanik karşılığı ise Şek.3-27 c)'de verilmiştir. Verilen örnek çift dört durumlu çoğullama çözücüdür: dört çıkış grubundan her birinin ikişer hattı vardır. Tümleşik devrenin iki veri girişi, sekiz çıkışı, yüksek uyum seviyesi sağlayan iki adres hattı ve iki kontrol hattı vardır. Şöyle ki, bir kontrol hattıyla veriden birinci bitin iletimi sağlanıyor, ikinci kontrol hattıyla ise ikinci bitin iletimi sağlanıyor. İki kontrol hattın uygulanması bir taraftan yonganın iki ayrı 1-den-4-e çoğullama çözücü olarak kullanma olanağı veriyor, diğer taraftan ise, iki kontrol girişi tek bir kontrol girişinde bağlanarak çift dört durumlu çoğullama çözücü elde ediliyor. Seçim hatlarıyla iki giriş hattında bulunan sinyallerin, dört çift çıkış hattın hangi iki çıkış hattın çıkacağı seçiliyor.

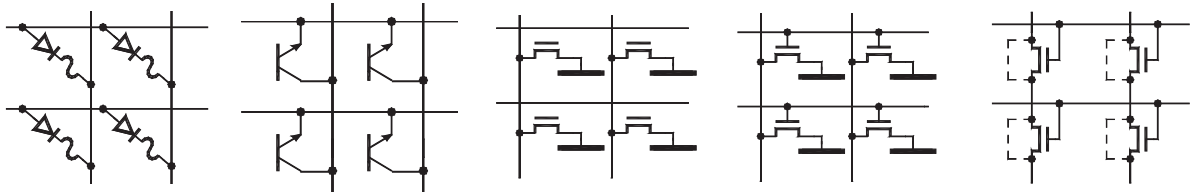


Şek. 3-27. Çift dört durumlu çoğullama çözücüsü

### III) PROGRAMLANABİLİR MANTIK YAPILARI

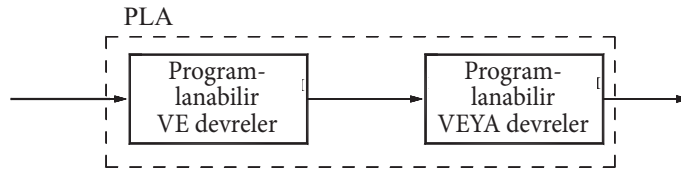
#### 3.7. GİRİŞ VE AYRIM

**Programlanabilir mantık yapılar** (PMY, İng. Programmable Logic Devices, PLD), *programlama yoluyla farklı birleşimsel ağların gerçekleşmesi için kullanılan tümleşik bileşenlerdir. Programlama* sürecinde, aslında ağdaki bağlantıları programlanması söz konusudur, daha doğrusu tümleşik bileşenin içeriğinde giren mantıksal devreler arasında bileşenin içeriğinde bağlanmanın kurulması ve kesilmesi programlanıyor. Bu ağların prensipte, şimdiye kadar incelediğimiz ağlara benzer matris yapıları var, ancak iki seviyeden oluşuyor. Tek fark, PMY'nin programlanmamış durumda, ağın dikey ve yatay hatların arasındaki kesişme noktalarında bulunan sigortalı bağlantı elemanları olarak anahtarlamalı (kesici) elemanların (diyotlar ya da iki kutuplu veya bir kutuplu transistörlerin) yerleşmiş olmasıdır. Aşağıdaki şekilde (Şek. 3-28) bağlantıların gerçekleştiği birkaç alçak ohmlu (kısa bağlantılar) veya yüksek ohmlu (kesilmiş devreler) anahtarlamalı eleman türü gösterilmiştir.

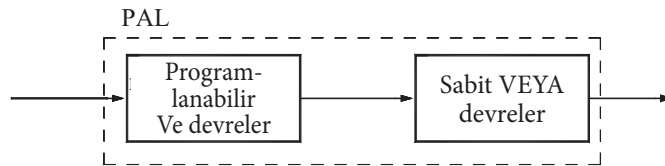


Şek.3-28. Diyodlu ve iki kutuplu transistör ile bağlantı elemanlı matris yapılar

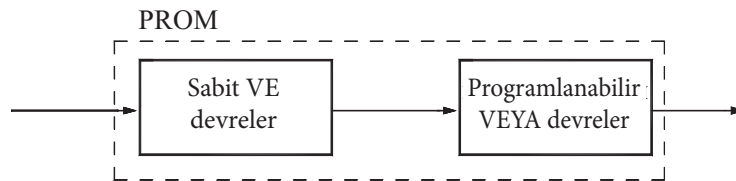
Şek.3-29,30 ve 31'de programlanabilir bileşenlerin temel yapıları belirtilmiştir. *Her yapı iki dizisel bağlanmış ağ içeriyor: biri VE mantıksal devrelerdir, diğeri ise VEYA geçitlerdir.* Buna göre, yukarıda söz ettiğimiz tek aşamalı matris yapılardan farklı olarak, burada *iki seviyeli VE-VEYA ağı mantıksal fonksiyonların uygulanması için iki aşamalı matris yapılardan bahsediliyor.*



Şek. 3-29. PLA programlanabilir bileşenin blok-diyagramı



Şek. 3-30. PAL programlanabilir bileşenin blok-diyagramı



Şek. 3-31. PROM programlanabilir bileşenin blok-diyagramı



İlk iki programlanabilir yapı, amacı çok girişli ve çıkışlı bileşik mantıksal ağların gerçekleştirilmesi olan tümleşik bileşenlerle ilgilidir. Şek. 3.29'a göre yapılan bileşenlerde hem VE devrelerin hem VEYA devrelerin programlanması mümkündür. Bu mantıksal ağların yapısı, **programlanabilir mantık matrisi** anlamına gelen PLA (İng. *Programmable Logic Array*) adıyla biliniyor. Diğer taraftan, Şek. 3-30'a göre yapılan tümleşik bileşenler kavramı *sadece VE devreleri için programlama olanağı verirken, VEYA devreler sabit olarak yapılmıştır*. Bu yapı PAL (Programmable Array Logic) kısaltılmış adıyla biliniyor ve **programlanabilir matris mantığı** anlamına geliyor.

En sonunda, Şek.3-31 *progamlamanın sadece VEYA devresinde yapılabildiği, VE devreleri ise sabit yapılmış* tümleşik bileşenlerle ilgilidir. Bu yapılar **sadece okunabilen programlanabilir bellekler veya kısaca PROM (Programmable Read Only Memory)** olarak biliniyor

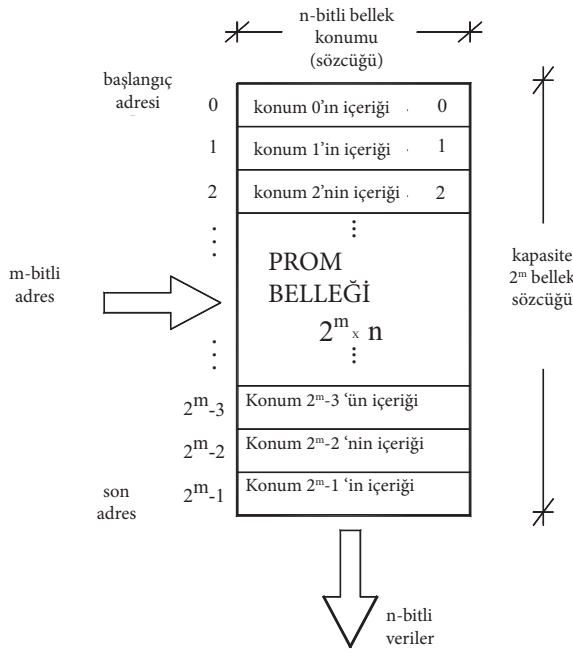
Programlanabilir mantıksal yapıların, bazı özel uygulama alanlarında belirli avantajlarla ve dezavantajlarla, prensipte evrensel özellikleri vardır. Bileşik mantıksal fonksiyonların gerçekleştiği PLA ve PAL matrislerden farklı olarak, PROM yapıların özel amaçları vardır, o da değişmesi gerekmeyen bilgilerin korunmasıdır. Devamda dikattimizi PROM belleklerin çalışmasına yönlendireceğiz.

### 3.8. PROM BELLEĞİ

PROM pratikte sıkça özel yarı iletken bellek türü olarak uygulanan *tümleşik teknikte yapılan dijital bileşendir*. Çalışma sürecinde içeriğinin sadece okunabileceğinden dolayı, PROM tahribatsız (bozucu olmayan) bellekler grubuna aittir. PROM-da korunmuş içerik, yerleşmiş olduğu cihazın elektrikle beslemesinin kesilmesinden sonra bile kaybolmuyor ve bu yüzden PROM belleği bilgisayar sistemlerin başlatılması sırasında kullanılıyor. Böylece, PROM örneğinin bilgisayarın içeriğinde bulunan grafik kartı, diski gibi bileşenlerin doğru çalışıp çalışmadığını kontrol eden ve ardından işletim sistemini, örneğin Windows'u başlatan bilgisayarın sistem yazılımında (BIOS-ta) yerleşiyor.

PROM'un Şek.3-31'deki blok-diyagramını göz önüne alarak, girişinde önceden oluşmuş sabit bağlantılı VE matrisinin olduğunu görebiliriz. Bu aslında m-girişli ve  $2^m$  iç çıkışlı kod çözücüdür. Bu çıkışlar, m VEYA mantıksal devresi içeren çıkış matrisin girişleri olarak kullanılabilir. VEYA devresinin her çıkışı m-giriş mantıksal değişkenli rastgele fonksiyon olabilir. Buna göre, organizasyon açısından, PROM'a genelde baytla  $1[B]=8[b]$  ifade edilen n-bitlik sabit uzunluğunda  $N = 2^m$  bellek sözcüğü içeren bellek bileşeni olarak bakabiliriz. Korunmuş (belleklenmiş)  $N=2^m$  sözcüğün her biri kullanıcının gereksinimine göre n-bit uzunluğunda belirli kombinasyondur. Bellekte her sözcüğün yerleşmesi için uygun yeri vardır. Buna göre *bellek Şek.3-32'ye göre büyük sayıda bellek yeri içeren bir sonlu ve düzenlenmiş kümenin olduğunu söyleyebiliriz*. Bellekte her yer (konum) belirli sayıda anahtarlamalı elemandan oluşmuştur. Anahtarlamalı elemanların sayısı bellek sözcüğünün uzunluğuyla eşittir. Herhangi bir yerde yerleşmiş sözcük, yani bilgi o yerin *içeriğini* tanımlıyor. Konumlara erişim belirli konumu benzersiz şekilde belirleyen sayının verilmesiyle (belirlenmesiyle) gerçekleşiyor. Bu sayıya *konumun adresi* denir. Bellek yerinin adresi biliniyorsa, aslında onun bellekteki yeri belirlenmiş oluyor. Böylece, PROM'un herhangi bir bellek konumuna erişim, adresin belirlenmesiyle doğrudandır. Bu yüzden bellekte her korunmuş veriye erişim zamanı eşittir.





Şek. 3-32. PROM bellek programlanabilir bileşenin organizasyon yapısı

$i$	$A_2$	$A_1$	$A_0$	$D_3$	$D_2$	$D_1$	$D_0$
0	0	0	0	1	0	0	0
1	0	0	1	0	0	0	1
2	0	1	0	0	0	1	1
3	0	1	1	0	0	1	0
4	1	0	0	0	1	1	0
5	1	0	1	0	1	1	1
6	1	1	0	0	1	0	1
7	1	1	1	1	1	0	0

Tab. 3-14. PROM'da korunması gereken sekiz dört bitli veri

Bununla ilgili, PROM'un kapasitesi olarak adreslerin toplam sayısı tanımlanıyor, yani bellek sözcüğün uzunluğuyla çarpılmış PROM'da korunabilen toplam sözcük sayısı. PROM'un büyük sayıda konumları olduğundan dolayı, kapasite genelde kilobaytlarla ya da megabaytlarla ifade ediliyor.  $1[MB] = 2^{20}[B] = 2^{10}[KB] = 1024[KB]$ .

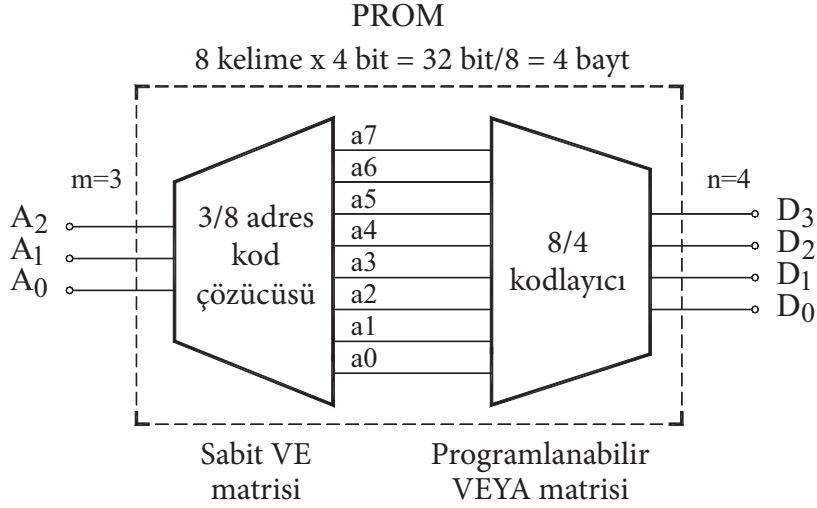
Tipik bir PROM bellek bileşeninde, tüm bağlantılar kurulmuş olup içeriği hep 1-lerle doldurulmuştur. Programlama sürecinde PROM programlayıcı olarak adlandırılan özel bir cihaz kullanılıyor. Bu cihazla, sadece 0-ların yazılması gereken yerlerde anahtarlamalı elemanlar yanıyor. Elemanların yanması, PROM'un amacına bağlı normal çalışması sırasında bağlanmış olduğu gerilimden çok daha yüksek değerli gerilim dürtülerin salınmasıyla yapılıyor.

PROM belleğinin çalışmasını daha iyi anlamak için şu örneği inceleyeceğiz. Örnekte PROM-bellek bileşeninde tab.3-14'e göre sekiz dört bitli verinin kalıcı korunması söz konusudur.

Tablodan görüldüğü gibi 000 adresli birinci konumun içeriği 1000 verisi olmalıdır, adresi 001 olan ikinci konumda 0001 verisi yerleşmiştir vs, içeriği 1100 olan 111 adresli son sekizinci bellek konumuna kadar.

$N=8$  verinin korunması gerektiğinden dolayı,  $8=2^m$  koşulunu yerine getirilecek şekilde belirli sayıda adres hatlı ( $m$ ) PROM-belleği seçeceğiz.  $8=2^m$  'den  $m=3$  olduğunu elde ediyoruz. Diğer taraftan, bellek yongasında girilen her verinin 4 bit uzunluğu var. Buradan çıkış veri hatlarının sayısı- $n$ 'in 4 olduğu elde ediliyor ( $n=4$ ). Aslında, her veri, adreslenen sekiz bellek konumdan herbirinin içeriğini tanımlıyor. Şek.3-33'te verilen örneğin çözülmesi için uygulanacak PROM bileşenin blok-diyagramı gösterilmiştir.

Adres kod çözücüsü pratik olarak sabit, matrisli VE yapıyla gerçekleşiyor, bilgilerin girmesi ise VEYA matrisin programlanması ile yapılıyor. VEYA matrisi aslında, adres kod çözücünün çıkışı olduğu kadar girişi olan ve bellek sözcüğünün uzunluğu olduğu kadar çıkışı olan kodlayıcıdır



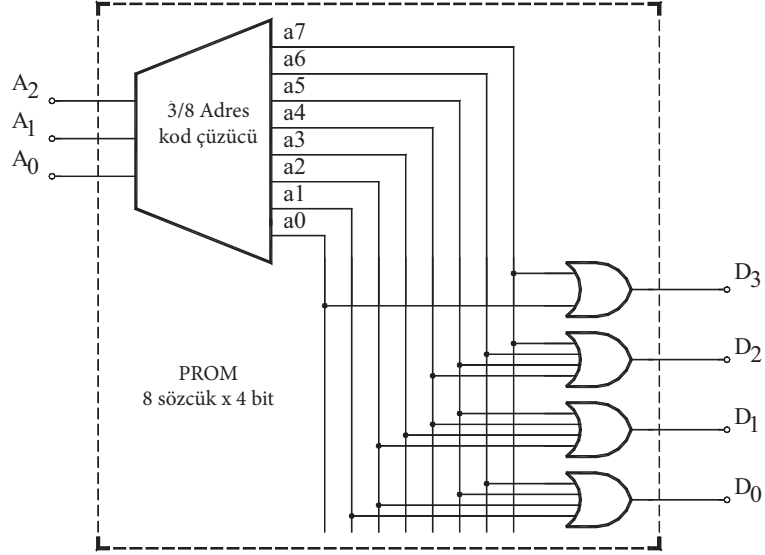
Tablonun uygulama şekli, Şek.3-34'te tanımlanmış PROM bileşenin mantıksal diyagramıyla gösterilmiştir. Verinin, yani D bellek sözcüğünün her biti,  $A_2, A_1$  ve  $A_0$  olarak belirtilen üç girişli değişkenin fonksiyonu olacak. Bu fonksiyon, adres kod çözücünün sekiz çıkışından -  $a_0, a_1, \dots, a_7$ , hangilerinin verilen kod tablosuna uygun VEYA devresinin girişleri olacağına bağlı olarak programlamayla elde edilebilir. Bununla ilgili olarak,  $D_3, D_2, D_1$  ve  $D_0$  çıkış bitleri, aşağıdaki KANB şekillerinden elde ediliyor.

$$\begin{aligned} D_3 &= a_0 + a_7 = A_2 A_1 A_0 + \bar{A}_2 \bar{A}_1 \bar{A}_0 \\ D_2 &= a_4 + a_5 + a_6 + a_7 = \dots \\ D_1 &= a_2 + a_3 + a_4 + a_5 = \dots \\ D_0 &= a_1 + a_2 + a_5 + a_6 = \dots \end{aligned}$$

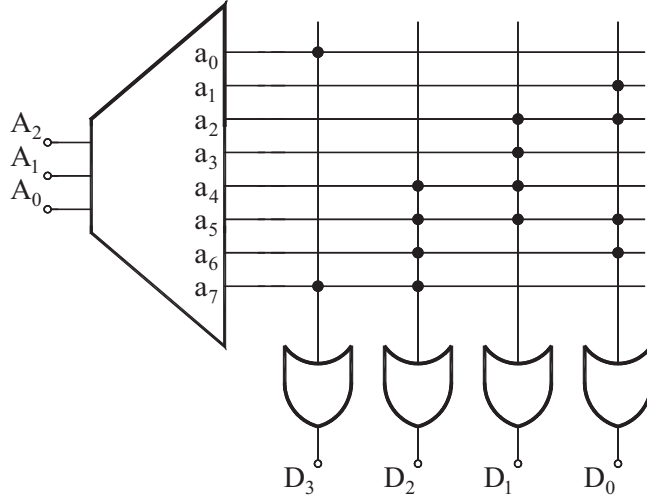
Mantıksal denklemler tablodan elde ediliyor. Tablodan  $D_3$  çıkışının elde edilmesi için sıfırıncı veya yedinci adres hattının ( $a_0$  veya  $a_7$ ) aktif olması gerektiği görülüyor, bu yüzden sadece onlarla birinci VEYA devresinde giriş olarak bağlantılar oluşuyor. Diğer altı girişin bağlı olması gerekmiyor. Benzer şekilde,  $D_2$  çıkışının elde edilmesi için, sadece  $a_4, a_5, a_6$  ve  $a_7$  ile bağlantılar programlanıyor,  $D_1$  için sadece  $a_2, a_3, a_4$  ve  $a_5$  ile bağlanma gerçekleşiyor,  $D_0$ 'ın elde edilmesi için ise uygun VEYA devresine sadece  $a_1, a_2, a_5$  ve  $a_6$  girişleri bağlanıyor.

Bileşen üreticisinin yonga çerçevesinde adres kod çözücüsünü yaptığından dolayı, kullanıcı konumların içeriklerini tek başına programlayıcı yardımıyla giriyor. Adres girişleri aracılığıyla bellek konumunun adresi veriliyor, veri hatları aracılığıyla ise belirli bellek konumunda, onun içeriği olarak uygun bellek sözcüğü giriliyor. Bellek sözcüğün girilmesi adres kod çözücünün çıkışları ile çıkış matrisinden her VEYA devresinin girişleri arasında bağlantıların kurulması veya elenmesi yoluyla yapılıyor.

Daha iyi görünürlük amacıyla, Şek. 3-34'teki bağlanma şekli Şek. 3-35 olarak gösterilmiştir. Bu şekilde kurulan bağlantılar, adres kod çözücünün çıkış hatları ile VEYA devresinin girişlerinin kesişiminde bağlantı noktasının eklenmesiyle belirlenmiştir.



Şek. 3-34. 4'er bitlik 8 sözcük kapasiteli PROM belleğin mantıksal diyagramı



Şek.3-35. 4'er bitlik 8 sözcük kapasiteli PROM belleğin mantıksal diyagramı

Programlamanın tamamlanmasından sonra, yani PROM'da gereken içeriklerin girilmesinden sonra, PROM'un tasarlanmış olduğu cihaza eklenerek normal çalışma düzeninde kullanılabilir.

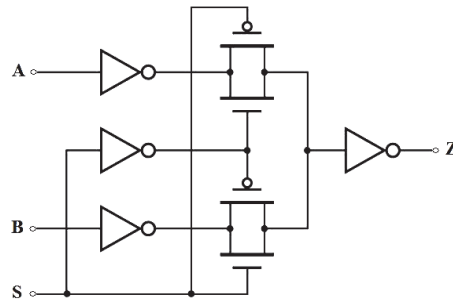
PROM'un içeriğinin sadece okunabildiğinden dolayı, bu bellek programlanabilir ROM bellekler (İng. *Programmable Read Only Memory*) grubuna aittir. PROM bellek yongaların olumsuz tarafı onların sadece bir kez programlanabilmesi ve girilen içeriğin artık değişmeyeceği olmasıdır. Pratik uygulamalarda, birkaç kez programlanabilen EPROM bellek bileşenleri de kullanılıyor. Bu bileşenlerin birkaç kez programlanabilme olanağı içeriklerinin değişme imkânını sağlıyor (İng. *Erasable Programmable ROM*). EPROM'da eski içeriklerin silinmesi mor-ötesi (UV) ışınımın ekspozisyonu ile yapılıyor ve bu yüzden EPROM-lar UVEPROM olarak da adlandırılıyor. EEPROM ya da E PROM bellek yongaları (İng. *Electrically Erasable Programmable ROM*) fazla kez programlanabilir, ancak içeriklerin silinmesi elektrik dürtüleriyle gerçekleşiyor. Son yıllarda pratik Flash bellek yongaları da giderek fazla kullanılıyor. Flash bellek yongaları, USB Flash bileşenlerinde ve bellek kartlarında kullanılan özel EPROM bellek türüdür.

## TEKRARLAMA SORULARI VE ÖDEVLERİ

- 3-1. Birleşimsel ağların en önemli özellikleri nedir?
- 3-2. Birleşimsel ağlar ile hangi işlemler gerçekleşiyor?
- 3-3. İkili toplayıcının rolü nedir?
- 3-4. Yarı toplayıcının rolü nedir?
- 3-5. Yarı toplayıcının doğruluk tablosunu, mantıksal diyagramını ve mantıksal sembolünü çiz.
- 3-6. Tam toplayıcının rolü nedir?
- 3-7. Tam toplayıcının doğruluk tablosunu, mantıksal diyagramını ve mantıksal sembolünü çiz.
- 3-8. Tam toplayıcılar kullanarak iki iki bitli sayı için paralel toplayıcının mantıksal diyagramını çiz. Hangi giriş kombinasyonları için ya da giriş sayılarının hangi değerleri için elde çıkış hattı aktif olacak ( $C+=1$ ) ve neden?
- 3-9. Karşılaştırıcının rolü nedir?
- 3-10. Bir bitli karşılaştırıcının doğruluk tablosunu ve mantıksal diyagramını çiz.
- 3-11. (\*) Giriş bağlantıları olarak meydana gelen  $A=A_1A_0$  ve  $B=B_1B_0$  iki iki bitli sayıları için dijital (ikili) karşılaştırıcı projeye. Bileşenin şu koşulları yerine getirmesi gerekiyor:
- a) Tab. 3-4'e uygun üç çıkışı:  $Y_A$ ,  $Y_B$  ve  $Y_0$  olmalıdır
  - b) mantıksal durumları karşılaştırılan iki sayının arasındaki ilişkiye bağlı olan iki çıkış  $G$  ve  $L$  olmalıdır. Bu arada:
    - ⊕ Eğer  $A > B$  ise o zaman  $G = 1$  ve  $L = 0$ ;
    - ⊕ Eğer  $A < B$  ise o zaman  $G = 0$  ve  $L = 1$ ;
    - ⊕ Eğer  $A = B$  ise o zaman  $G = 0$  ve  $L = 0$ .
- 3-12. (Tek) tümleme devresinin rolü nedir?
- 3-13. İki bitli sayılar için (tek) tümleyicinin mantıksal diyagramını çiz.
- 3-14. (\*) İki bitli sayıların toplanması için paralel devre ve iki bitli sayıların tek tümleme devresini uygulayarak, iki bitli sayıların çift tümelemesini gerçekleştiren mantıksal devre çiz.
- 3-15. İki iki bitli sayının toplama/çıkarma devresinin mantıksal diyagramını çiz.
- 3-16. Anahtarlamalı matrisler nasıl birleşimsel ağlardır?
- 3-17. Kodlayıcının nasıl işlevi olduğunu açıkla!
- 3-18. DEC/NBCD kodlayıcının doğruluk tablosunu çiz. NBCD kod sözcüğünden a) D; b) C; c) B; ç) A bitinin hangi giriş kombinasyonu için 1 değeri var?
- 3-19. Girişleri yüksek seviyede aktif olan bir seviyeli VEYA matris yapısı olarak DEC/NBCD kodlayıcının uygulamasını ve onun mantıksal sembolünü çiz.
- 3-20. (\*) OVE mantıksal devreleri uygulayarak girişleri alçak seviyede aktif olan DEC/NBCD kodlayıcının gerçekleştirmesini ve onun mantıksal sembolünü çiz.
- 3-21. OCT/BİN kodlayıcının girişleri yüksek seviyede aktif olan bir seviyeli VEYA matris yapısı olarak uygulamasını ve onun mantıksal sembolünü çiz. (\*) Çalışma izni için giriş ekle!

- 3-22. Standart kodlayıcı ile öncelikli kodlayıcı arasındaki fark nedir?
- 3-23. Elimizde iki kodlayıcının olduğunu tahmin edersek: biri öncelikli OCT/BİN ve biri öncelikli OCT/BİN ve ikisinde aynı anda [2], [3] ve [4] tuşlarına basarsak iki kodlayıcının çıkışında hangi durumun elde edileceğini cevapla. Çıkışları farklı mıdır? Açıklayın!
- 3-24. (\*) HEX/BİN kodlayıcının uygulamasını çiz.
- 3-25. Kodlayıcının girişinde N farklı sembol gelirse, tüm giriş sembollerin kodlanabilmesi için çıkışların sayısı, yani kod sözcüğün uzunluğu n hangi koşulu yerine getirmelidir?
- 3-26. Kod çözücünün rolü nedir?
- 3-27. Çıkışları a) alçak seviyede b) yüksek seviyede aktif olan NBCD/DEC kod çözücüsü için doğruluk tablolarını çiz.
- 3-28. Çıkışları yüksek seviyede aktif olan NBCD/DEC kod çözücüsü için VE mantıksal devreleri uygulayarak bir seviyeli VE matris ağı olarak gerçekleşmesinin çiz. Çıkışların alçak seviyede aktif olmaları için ağa ne tür değişimler yapılmalıdır?
- 3-29. Hangi giriş kombinasyonu için (a) 0, (b) 1, (c) 2, (ç) 3, (d) 4, (e) 5, (f) 6, (g) 7, (h) 8, (ı) 9 NBCD/DEC kod çözücünün çıkışı aktif olacak.
- 3-30. NBCD/7S kod çözücüsü için doğruluk tablosunu çiz. (\*) Elimizde rastgele sayıda girişleri olan gereken tüm mantıksal devrelerin olduğunu tahmin ederek bu mantıksal ağı projeye.
- 3-31. Kod çözücünün girişinde n bit uzunluğunda kod sözcükleri gelirse, çıkışların en yüksek sayısı N hangi koşulu yerine getirmelidir?
- 3-32. (\*) 4/16 (BIN/HEX) kod çözücünün birleşimsel ağını şunları uygulayarak gerçekleştir: a) VE devrelerle b) OVE devrelerle ve 1) alçak seviyede 2) yüksek seviyede aktif olan çalışma izini için E girişinin uygulanmasıyla.
- 3-33. (\*) Elinde E çalışma izni girişli 2-den-4-e kod çözücü varsa 3-ten-8-e kod çözücü elde etmek için onların bağlanmasını gerçekleştir.
- 3-34. Çoğullayıcının işlevi nedir?
- 3-35. (a) 2-den-1-e, (b) 4-ten-1-e, (c) 8-den-1-e ve (\*) 1) alçak seviyede 2) yüksek seviyede aktif olan çalışma olanağı için girişli çoğullayıcının kombinasyon tablosunu, mantıksal diyagramı ve mantıksal sembolünü çiz.

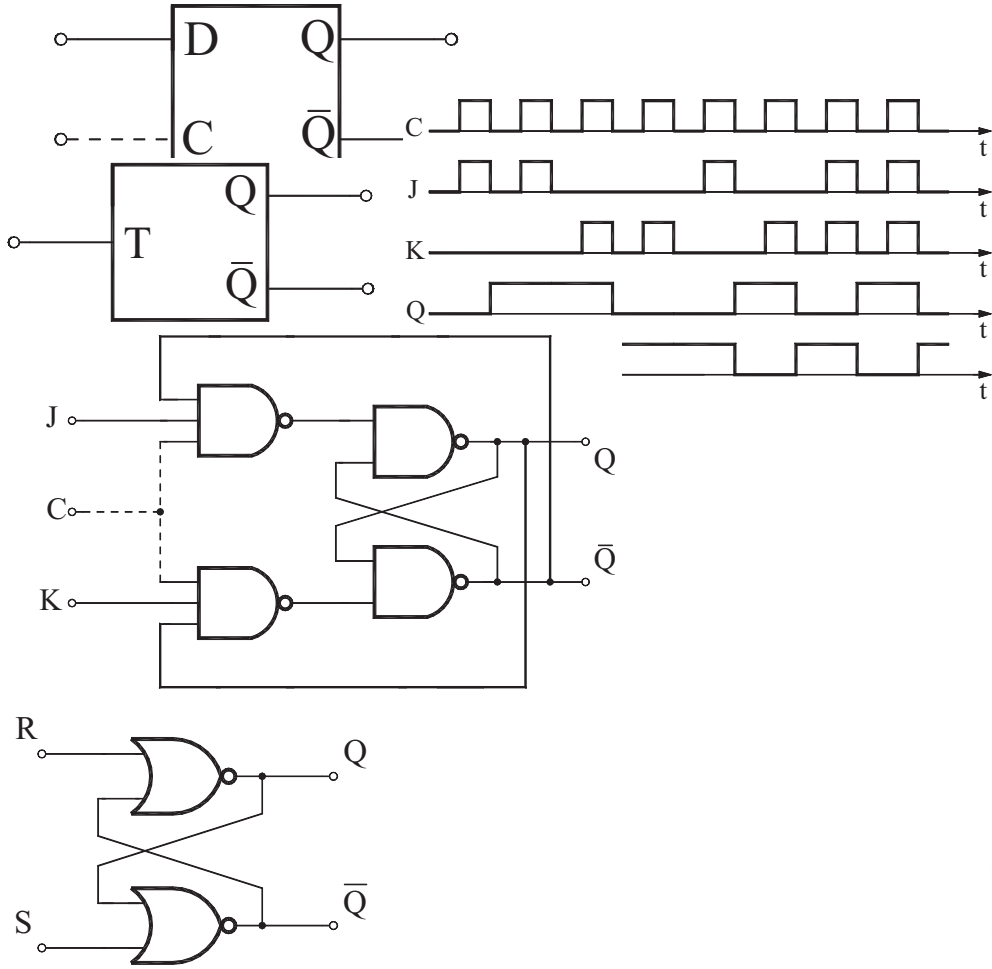
3-36. (\*) Aşağıdaki şekilde eviriciler dışında iletim geçitleri de içeren mantıksal ağ gösterilmiştir. Ağ için doğruluk tablosunu oluştur ve doldur ve onun davranışını, sembolik işaretini ve olası uygulamasını açıkla.



Ödev 3-36 şekli

- 3-37. dörtlü-ikidurumlu (4-ten-2-ye) çoğullayıcının mantıksal diyagramını çiz.
- 3-38. (\*) (a) çift-ikidurumlu (2-den-2-ye), (b) çift-dördürümlü (2-den-4-e) çoğullayıcının mantıksal diyagramını çiz.
- 3-39. Çoğullama çözücü ne işlev gerçekleştiriyor?
- 3-40. (a) 1-den-2-ye, (b) 1-den-4-e, (1-den-8-e) (\*) ve 1) alçak seviyede, 2) yüksek seviyede aktif olan çalışma olanağı için girişli çoğullama çözücünün kombinasyon tablosu, mantıksal diyagramı ve mantıksal sembolü çizilsin.

- 3-41. Elinde 1-den-4-e çoğullama çözücüsü varsa onun 2-den-4-e kod çözücüsüne dönüşümünü yap.
- 3-42. Elinde 2-den-4-e kod çözücü ve çalışma izni için özel giriş varsa, onun 1-den-4-e çoğullama çözücüye dönüşümünü yap.
- 3-43. Alçak seviyede aktif seçim bileşeni için özel girişli çift-dört durumlu (2-den-4-e) çoğullama çözücünün mantıksal diyagramını çiz.
- 3-44. Alçak seviyede aktif olan seçim bileşeni için özel girişli (a) çift-ikidurumlu (2-de-2), (b) dörtlü-ikidurumlu (4-te-2) çoğullama çözücünün mantıksal diyagramını çiz.
- 3-45. Programlanabilir mantık yapıları nedir?
- 3-46. Programlanabilir mantık yapıların programlanması nasıl gerçekleşiyor?
- 3-47. Programlanabilir mantık yapılarda bağlantı noktalarının kurulması için kullanılan (a) alçak ohmlu, (b) yüksek ohmlu elektronik elemanları için iki örnek göster.
- 3-48. PLA nasıl bir yapıdır ve onun ana uygulaması nedir?
- 3-49. PAL bileşenin blok-diyagramını çiz.
- 3-50. PAL nasıl bir yapıdır ve onun ana uygulaması nedir?
- 3-51. PROM bileşenin blok-diyagramını çiz.
- 3-52. PROM nasıl bir yapıdır ve ana uygulaması nedir?
- 3-53. Bellek nasıl düzenlenmiştir?
- 3-54. Bellekte her bilgi nasıl biçimde korunuyor ve nerede yerleşiyor?
- 3-55. Bellek konumunun adresi nedir?
- 3-56. Bellek kapasitesi nedir ve nasıl ifade ediliyor?
- 3-57. PROM'da yeni içeriğin girilmesi nasıl yapılıyor?
- 3-58. Hangi PROM bellek çeşitleri vardır ve aralarındaki fark nedir?
- 3-59. (a) 16 x 4 (4b'lik 16 sözcük), (b) 128 x 8 (8b'lik 128 sözcük) PROM'un blok-diyagramını çiz.
- 3-60. 64K x 16 organizasyonlu PROM verilmiştir. bu bellek bileşenin (a) kaç adresi giriş hattı var, b) kaç veri çıkış hattı vardır, c) PROM'un a) sözcüklerle, b) baytlarla ifade edilmiş kapasitesi ne kadardır.
- 3-61.(\*) Uygun kapasiteli ve bellek sözcüklerin uzunluğu ile PROM'un kombinasyon tablosu çizilsin, öyle ki 0000 en küçük adresli konuma en büyük dört bitli sayı 1111 yerleşiyor vs. adreslerin artmasıyla onlardan yerleşen sayılar azalıyor ve böylece 1111 adresi olan son bellek konumunda 0000 verisi yerleşiyor. (\*\*) Çıkış verinin her biti için (a) adres kod çözücünün çıkışlarında, (b) girişlerinden fonksiyonu uygun kusursuz normal biçiminde yaz.



# 4. FLİP- FLOPLAR

Bu konusal birimini öğrendikten sonra:

- ⊕ Aşağıdaki flip-flopların standart ve master-slave yapımlı temel ardışık devreler olarak çalışma prensibini anlayacaksınız:
  - ⊕ SR Flip-flop;
  - ⊕ JK Flip-flop;
  - ⊕ T Flip-Flop;
  - ⊕ D Flip-flop;
- ⊕ Flip-flopların farklı dönüşümlerini gösterebileceksiniz;
- ⊕ Flip-flopların daha kompleks ardışık bileşenlerin gerçekleşmesinde flip flopların uygulanmasını tanıyacaksınız:
  - ⊕ Kilitleme devresi;
  - ⊕ RAM bellek hücresi.
- ⊕ Daha kompleks ardışık ağların yapısında flip-flopların uygulanmasını anlayacaksınız





#### 4.1. GİRİŞ VE TEMEL TERİMLER

**Flip-flop (iki durumlu, kapan devresi)** olarak adlandırılan **iki kararlı multi vibratör**, dijital sistemlerde verilerin depolanmasını sağlayan temel elemandır. Flip-flop elementer bellek hücresi tanımıyor, çünkü bir bitlik veri ya da en küçük bilgi miktarı hafıza edebilir. Hatırlama yeteneği, flip-flopun iki kararlı durumdan birinde bulunabilmesinin sonucudur.

Flip-flop yenileyici devreler grubuna aittir, çünkü uygulama sırasında pozitif geri bağlantının olması gerekiyor. Yapılımına ve çalışma şekline bağlı olarak farklı flip-flop çeşitleri vardır. Ayrık teknikte, flip floplar iki kuvvetlendirici aşamın, en basitçe iki transistörle bağlanmasıyla uygulanıyor, ardından iki mantıksal devrenin uygun bağlanmasıyla uygulanabilir, bu arada en sıkça olarak birbirine bağlı OVE veya OYA devreleri kullanılıyor ya da tülmeşik teknikte ayrı dijital bileşen olarak yapılıyor.

Her flip-flopun birbirine bütünleyici iki çıkışı ve bir ya da fazla girişi vardır. Uluslararası sözleşmeyle, flip-flopun durumu Q ile işaretlenmiş çıkışın (nominal, doğrudan çıkışın) değeriyle ifade edilmesi kabul edilmiştir, böylece aslında  $\bar{Q}$  ile işaretlenmiş çıkışın (tümleyici çıkışın) da değeri belirleniyor. Tab. 4-1'den görüldüğü gibi eğer Q=1 ( $\bar{Q}$  =0) ise flip-flop için **düzenlenmiş** ya da **ayarlanmış** olduğunu diyoruz, ya da eğer Q=1 ( $\bar{Q}$  =0) **silinmiş** ya da **sıfırlanmış** olduğunu diyoruz.

Flip-flopun durumu	Çıkışlar	
	Q	$\bar{Q}$
Ayarlanmış (Düzenlenmiş)	1	0
Sıfırlanmış (Silinmiş)	0	1

Tab. 4-1. Flip-flopun durumları

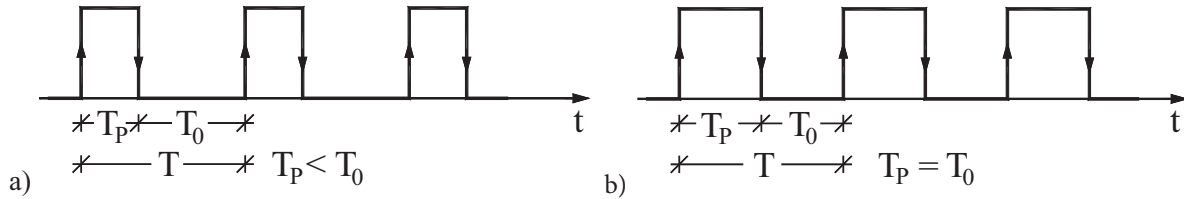
Girişler aracılığıyla flip-flopun çalışması yönetiliyor ve/veya onda saklanacak veriler getiriliyor. Q,  $\bar{Q}$  çıkışların durumları, teoretik olarak sonsuz uzun süre korunabilir, pratikte ise girişlerden bazısı, çıkışlara efektif etkisi olan dürtünün getirilmesine kadar, veya çıkışların durumunu değiştiren dürtünün getirilmesine kadar korunabilir. Girişlerin, çıkışlar üzerine doğrudan (dolaysız) ya da endirekt (dolaylı) etkisi olduğuna bağlı olarak, asenkron (eşzamansız) flip-floplar ve senkron (eşzamanlı) flip-floplar vardır.

**Asenkron** flip-floplarda çıkışların durumu sadece bilgi (veri) girişlerinde meydana gelen uyarma sinyallerin seviyesine bağlıdır. Bu girişlerin durumu, çıkış mantıksal seviyelerin değişip değişmeyeceğini belirliyor.

**Senkron** flip-floplarda veri girişleri dışında bir ayrı giriş daha var, o da CLK, CK, CL, C,  $C_p$  veya  $T_1$  ile işaretlenebilen **pals** (*clock*) sinyal girişidir. Bu sinyal dörtgen gerilim dürtü şeklinde ya da daha sıkça pratik gereksinimlerden dolayı, karesel dalga şeklindedir.

Palsın, yüksek seviyede, yani 1'de  $T_p$  dürtüsünde ve alçak seviyede, ya da 0'da  $T_0$  duraklamasında, tam olarak belirlenmiş zaman aralıkları var. Böylece pals sinyalinin  $T$  periyodu da belirleniyor,  $T = T_p + T_0$ . Pals sinyali flip-flopların çalışmasının zamansal yönetimi ve uyumlaşma (senkronize) sinyalidir ve bir şeyin meydana geleceği anı (zamanı) belirliyor. Bu yüzden bu flip-floplara palslı flip-floplar da denir. Şöyle ki, flip-flopun durumu giriş sinyallerin kombinasyonuna bağlı olarak değişebilir, ancak bu değişiklik sadece pals-sinyalinde aktif (anahtarlama) seviye meydana geldiği anda olabilir. Palsın aktif seviyesi, flip-flopun çıkışlarına girişlerin efektif etkisi olduğu süre seviyesidir ("çıkışlarda bir şeyin olmasına" yol açabilen durum). Genel olarak, aktif seviye 0'ın seviyesinde veya 1'in seviyesinde olabilir, ancak genellikle aktif seviye olarak yüksek seviye alınıyor. Aktif seviyenin alçak seviye olduğu flip-floplar yapımlarında, bu durum özellikle vurgulanacaktır.

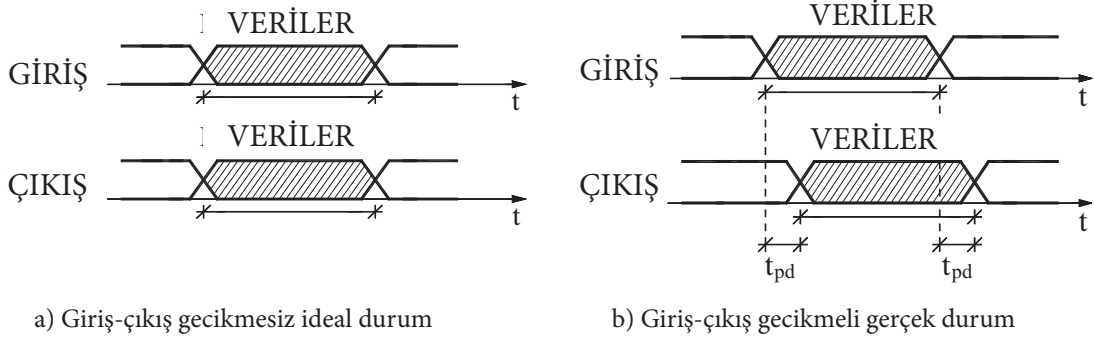
Şek. 4-1 a) ve b)'de dikdörtgen ve kare pals-sinyalin zamansal diyagramları gösterilmiştir. Şekilde palsın iki en önemli bölümleri belirtilmiştir: ön (pozitif, yükselen) kenarı ve arka (negatif, düşen) kenarı. Bunun vurgulanması önemlidir, çünkü kullanımda olan flip-flopların en büyük bölümü pals-sinyalin yükselen ya da düşen kenarın meydana gelmesiyle aktifleşiyor. **Aktif kenara tetikleyici (anahtarlama) ya da efektif kenar da denir (İng. triggering), çünkü flip-flopların çıkışında değişiklik sadece o an yaşanabilir.** Pals-sinyalin **döngüsü (aralığı)** bir aktif kenardan diğer aktif kenara kadar süreyi tanımlıyor. Bu süre aslında pals-sinyalin  $T$  periyoduyla eşittir.



Şek. 4-1. Dikdörtgen ve kare pals sinyallerin zamansal diyagramları

Flip-flopun hangi kenarda aktifleştiği önemli olmadan, flip-flop çıkışının dürtüye belirli bir süre sonra karşılık verdiğiğine değineceğiz. Bu süre çok kısadır, ancak sinyalin flip-floptan geçince gecikme sonucu olarak vardır. Bu süreye gecikme zamanı (süresi) denir ve  $t_{pd}$ ,  $t_d$ , veya  $\Delta t$  ile işaretleniyor. Gecikme zamanı flip-flopun üretildiği teknolojiye bağlı olarak değişiyor ve birkaç  $[ns]$  ile birkaç onluk  $[ns]$  arasında değişebiliyor. Flip-flopların çalışma prensibini daha kolay anlamak için, zamansal diyagramların çizildiği sırasında, ideal durumları gözetleyeceğiz, yani gecikme zamanının dikkate almayacağız ( $t_{pd}=0$ ), öyle ki çıkış gecikmeden, girişe göre aynı zamanda meydana geldiğini alacağız. Gerçek durumlar ya da çıkış ve giriş arasında zamansal farkı ( $t_{pd} > 0$ ), sadece flip-flopun çalışması sırasında pratik uygulamalarda belirli sorunların meydana geldiğini vurgulamak istediğimiz zaman dikkate alacağız.

Şek. 4-2 a)'da gecikemenin ihmal edilmiş flip-flop örneği olarak giriş sinyallerin meydana gelmesi ve onlara karşı gösterilen tepki, yani çıkış sinyaller gösterilmiştir. Şek. 4-2 b)'de dürtüden verilen karşılığa kadar belirli gecikmenin olduğu gerçek durum gösterilmiştir.

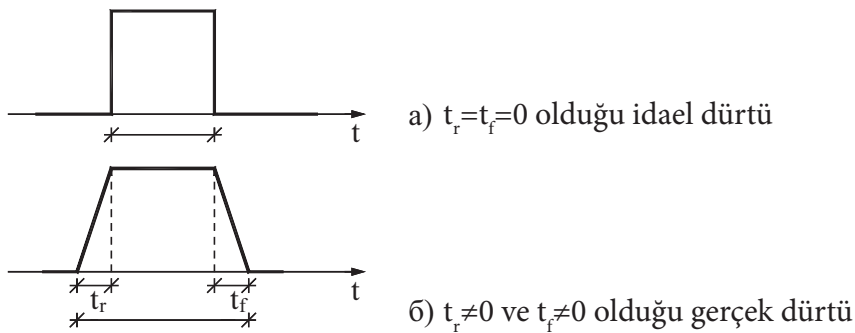


Şek.4-2. Flip-flop'ta giriş ve çıkış sinyallerinin zamansal diyagramları

Flip-flop çıkışı, onunla beraber çalışma şekli de analitik olarak belirli mantıksal denklemlerle belirlenebilir. Bu denklem her flip-flop için özeldir ve bu yüzden bu denkleme *karakteristik denklem* veya geçiş fonksiyonu denir. Bu cebirsel şekil dışında, tablo gösterimi de var. Tablo gösterimi iki tablo türü ile uygulanabilir: *karakteristik tablo* veya *geçiş ve çıkış tablosu*, diğeri ise *uyarma* ya da *eksitasyon tablosudur*. Geçiş tablosu ardışık ağların incelenmesi sırasında kullanılıyor, uyarma tablosu ise ardışık ağların gerçekleşmesi sırasında kullanılıyor. Çalışmanın gösterilmesi için flip flopun girişlerinde ve çıkışlarında gerilimlerin zamansal diyagramlarının grafiksel gösteriminin de kullanıldığını önceden vurgulamıştık.

Flip-flop çıkışı, girişlerinde getirilen sinyallerin fonksiyonu, palslı flip-flop durumunda pals seviyesinin fonksiyonu ve flip-flop çıkışlarının önceki durumunun fonksiyonu olabilir. Demek ki, zamansal sıralamasından, ya da flip-flopun geçtiği mantıksal durumlardan bağıllık ortaya çıkıyor. Buna göre, flip-flop elementer ardışık bileşenin olduğu da ortaya çıkıyor. Flip-flop çıkışının tanımlandığı sırasında, birbirini izleyen farklı zamansal aralıklarında mantıksal durumların yer aldığından dolayı, devrenin durumu hakkında önceki (şimdiki) zaman aralığına:  $t$  veta  $T$ , ya da sıradaki zaman aralığına  $t+T$ ,  $t+1$ ,  $t_n+1$ , veya  $t^+$ , olduğuna ilişkin işaretleri kullanılıyor. Bu doğrultuda flip-flop çıkışının durumuyla ilgili şu işaretler kullanılıyor  $Q(t)$  ve  $Q(t+T)$ , veya  $Q(t)$  ve  $Q(t+1)$ , veya  $Q(t_n)$  ve  $Q(t_n+1)$ , veya  $Q_n$  ve  $Q_{n+1}$ , veya  $Q(t)$  ve  $Q(t^+)$ , veya en basit olarak  $Q$  ve  $Q^+$ :

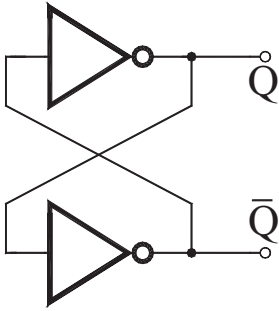
Devamda, gerilim sinyallerinin, Şek.4-3 a)'ya göre ideal dikdörtgen şekilde olduklarını tahmin ederek, Şek. 4-3 b)'de verilen gerçek dürtü sırasında meydana gelen sinyalin yüksek ya da alçak seviyeye ulaşma zamanını ihmal edeceğiz.



Şek.4-3. Flip-flop girişinde veya çıkışında gerilimi dürtüsü

## 4.2. SR FLİP-FLOP

Prensipite, flip-flop iki evirici devrenin çapraz bağlantısıyla uygulanarak, Şek. 4-4'te görüldüğü gibi bir devrenin çıkışı ikinci devrenin girişine bağlanıyor, ikinci devrenin çıkışı ise birinci devreye gidiyor.



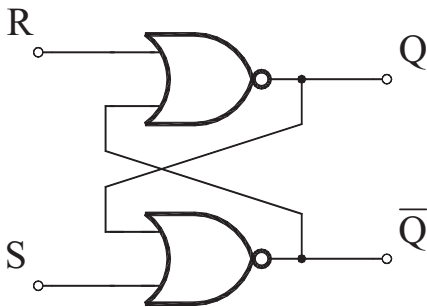
Şek. 4-4. İki evirici devrenin çapraz bağlantısı

Böyle bağlanma flip-flop çıkışlarının kararlı (sabit) ve birbirine tümleşik olmaları sağlanıyor. Kesişimle bir kararlı durumdan başka kararlı duruma geçiş süresi ihmal edilebilir kadar kısa, sıfıra hemen eşit olduğu yenileme sürecin oluşması için, pozitif geri bağlantı gerçekleşiyor. Fakat, bu şekilde uygulanan flip-flopun belirlenemeyen ve tanımlanamayan rastgele durumu olacak: durum ya  $Q=1$  ( $\bar{Q}=0$ ), ya da  $Q=0$  ( $\bar{Q}=1$ ) olacak. Ayrıca, çıkışların durumları ile yönetim olanağı yok çünkü girişler de yoktur.

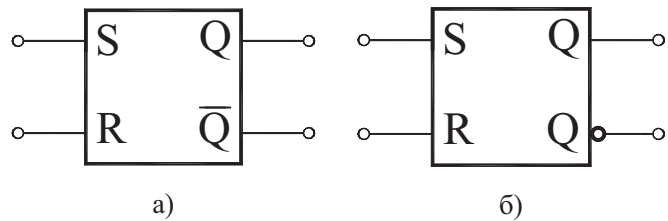
Bu nedenlerden dolayı, fazla girişli devrelerin farklı şekillerde kesişmesi uygulanıyor ve bunun hakkında devamında söz edeceğiz.

### 4.2.1. OYA TÜRÜNDEN SR FLİP-FLOP

İkişer girişli iki OYA devrenin kesişmesiyle elde edilen flip-flopa popüler olarak **SR** ya da **RS flip-flopu** denir. İngilizce terminolojide böyle devre SR Latch olarak yani kilitleme veya tutma(koruma) devresi olarak da biliniyor. İlerdeki bölümlerde leç terimini uygulayarak çevirisiz kullanacağız. **SR flip-flopun** mantıksal yapısı Şek.4-5'te gösterilmiştir, sembolik işaretleri ise Şek.4-6 a) ve b)'de verilmiştir



Şek.4-5. SR flip-flopun mantıksal diyagramı



Şek.4-6. SR flip-flopun sembolik işaretleri

S (SET) girişi flip flopun ayarlanması (düzenlenmesi) için kullanılıyor ( $Q = 1$ ). Ayarlama girişi 1 getirilerek ( $S=1$ ), R ise 0'da tutularak yapılıyor. S girişi 1 seviyesine ulaşınca, çıkışta  $Q = 1$  ( $\bar{Q} = 0$ ) durumu meydana geliyor: çıkış önceden 1 ise, 1'de kalıyor, ancak çıkış bundan önce 0 ise, seviyesi değişecek ve 1'e gidecek.

R (RESET) ile işaretlenen ikinci girişle, önceki şekilden ters şekilde flip-flop siliniyor, sıfırlanmış duruma getiriliyor ( $Q=0$ ): şimdi R girişine 1 getiriliyor, S çıkışı ise 0'a ayarlanıyor. R girişi 1 değerine ulaştınca, çıkışta 0 meydana geliyor, ya da  $Q=0$  ( $\bar{Q}=1$ ). Çıkışın bundan önce değeri 0 ise 0 kalacak, ancak 1 ise o zaman 0'a geçişecek. S ve R girişleri aynı zamanda 0 ise ( $S=R=0$ ), flip-flop önceki durumunu koruyor.

S ve R girişlerine aynı anda 1 getirilirse ( $S=R=1$ ), o zaman flip-flopun çıkış durumu hem Q hem  $\bar{Q}$ , 0'da olacak şekile bulunacak. Demek ki, baştan tahmin ettiğimiz flip flopun çıkışları birbirine göre tümleşik olmaları durumu geçerli olmayacak. Bunun dışında flip-flopun bir sonraki durumu belirlenmiş olabilir. Şöyle ki,  $S=1$  ve  $R=1$  uyarılmasından sonra, her iki giriş alçak seviyeye gideceğini  $S=0$  ve  $R=0$  tahmin edersek, o zaman çıkış daha uzun süre 1'de kalan sinyale bağlı olacak. Çıkış seviyesi daha geç (ikinci olarak) alçak seviyeye düşen giriş sinyalinin durumundan belirleniyor. İki sinyalin aynı anda değiştiğini aldığımızı göre, Q ve  $\bar{Q}$  nun çıkış seviyesi kesin olarak belirlenemiyor. Bu yüzden  $S=1$  ve  $R=1$  giriş kombinasyonuna SR flip-flopunda izin verilmiyor.

Açıkladıklarımızdan flip flop üzerine efektif etkinin, yani çıkış değişmesinin sadece S veya R girişlerinden bazısında 1'in bulunması meydana geleceğine sonuç çıkarabiliriz. Bu yüzden, S ve R girişlerin 1'de, ya da yüksek seviyede aktif olduklarını diyoruz.

SR flip-flopun çalışması tab. 4-2 olarak işaretlenmiş geçiş ve çıkış tablosundan tamamen belirlenebilir, onun uyarma tablosu ise tab. 4-3 olarak verilmiştir. Tab. 4-1 geçiş tablosundan hareket ederek, SR flip-flopların çalışma prensibi analitik yoluyla da, aşağıdaki karakteristik denklemle açıklanabilir:

$$Q^+ = S + \bar{R} Q, \text{ ose } Q^+ = (S + Q) \bar{R}, \text{ ku } SR = 0 \quad (4-1)$$

S	R	$Q^+$
0	0	Q
0	1	0
1	0	1
1	1	?

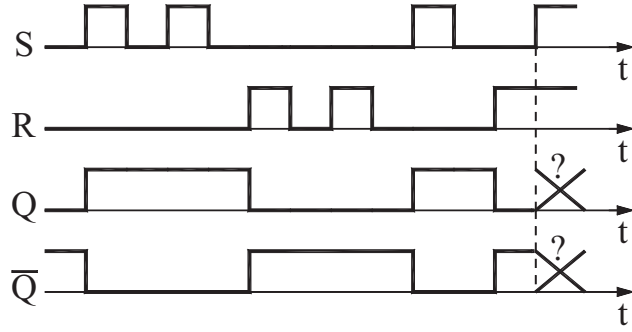
Tab. 4-2. SR flip-flopun geçiş ve çıkış tablosu

Q	$Q^+$	S	R
0	0	0	x
0	1	1	0
1	0	0	1
1	1	x	0

Tab. 4-3. SR flip flopun uyarma tablosu

4-1 denkleminde  $SR=0$  koşulunun geçerli olması gerekiyor.  $SR=0$  sınırlandırılması, yukarıda S ve R girişlerin aynı anda aktif olma yasağını ifade ediyor, yani ikisi de aynı zamanda mantıksal 1 durumunda olamaz.

SR flip-flopun çalışması, Şek.4-7'de verilen zamansal diyagramlarla gösterilmiştir. Bu arada başlangıç durumunun  $Q=0$  ( $\bar{Q}=1$ ) olduğu tahmin ediliyor, yani başlangıçta flip flop sıfırlanmış durumdadır.

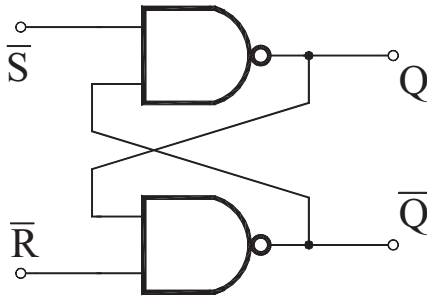


Şek. 4-7. SR flip-flopunda giriş ve çıkış katakteristik noktaların zamansal diyagramları

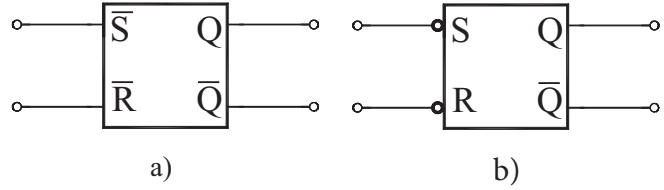
Şek.4-5'te gösterilen SR flip-flop yapılıminın leç (mandal) adlandırdığımızı önceki bölümde söylemiştir. Mandal için saydamlık karakteristik olaydır, çünkü bu devrede girişlerde mantıksal seviyelerin her değişikliği, çıkış seviyelerin değişmesine yol açıyor (yansıyor). Daha basit deyişle, çıkış giriş tarafından “görünebiliyor” ve giriş çıkış tarafından “görünebiliyor”.

#### 4.2.2.OVE TÜRÜNDEN $\bar{S} \bar{R}$ FLİP-FLOP

OYA türünden SR flip-flop dışında, Şek. 4-8'e göre ikişer girişli iki OVE devrenin çapraz bağlanmasıyla gerçekleşen OVE türünden SR flip-floplar da vardır. 1'de aktif olan SR OYA flip-floptan farklı olarak, SR OVE flip-flop 0 seviyesinde aktiftir, çünkü onun çıkışı iki girişten birinde 0'ın meydana gelmesine karşılık veriyor. Bu yüzden böyle flip-flop bazı kitaplarda genelde  $\bar{S} \bar{R}$  ile işaretleniyor, ya da S ve R girişlerine küçük çember işareti ekleniyor (o).  $\bar{S} \bar{R}$  flip-flopun mantıksal sembolleri Şek. 4-9 a) ve b)'de gösterilmiştir.



Şek. 4-8. S R flip flopun mantıksal diyagram

Şek. 4-9.  $\bar{S} \bar{R}$  flip-flopun sembolleri

$\bar{S} \bar{R}$  flip flopun geçiş ve çıkış tablosu tab. 4-4 ile işaretlenmiştir, uyarma (eksitasyon) tablosu ise tab.4-5 olarak verilmiştir.

$\bar{S}$	$\bar{R}$	$Q^+$
0	0	?
0	1	1
1	0	0
1	1	Q

Tab. 4-4.  $\bar{S} \bar{R}$  flip-flopun geçiş ve çıkış tablosu

Q	$Q^+$	$\bar{S}$	$\bar{R}$
0	0	0	x
0	1	0	1
1	0	1	0
1	1	x	1

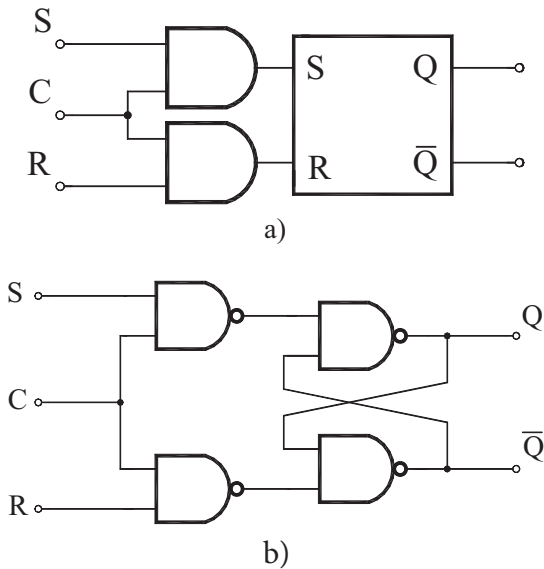
Tab.4-5.  $\bar{S} \bar{R}$  flip-flopun uyarma tablosu

Verilen kombinasyon tablosundan hareket ederek,  $\bar{S} \bar{R}$  flip-flopun çalışması kolayca açıklanabilir. Şöyle ki, girişte  $\bar{S} = 0$  ve  $\bar{R} = 1$  kombinasyonu getirilirse, o zaman flip-flop ayarlanmış durumda olacak, ya da  $Q=1$ , ( $\bar{Q}=0$ ) olacak. Diğer taraftan girişler  $\bar{S}=0$  ve  $\bar{S}=1$  ile uyarılırsa, o zaman flip-flop sıfırlanmış olacak, yani  $Q=0$ , ( $\bar{Q}=1$ ) olacak. Her iki giriş mantıksal 1 durumunda oldukları zaman, flip-flop çıkışının sonraki durumu önceki durum gibi aynı kalacak. Bu flip-flopta iki girişte 0-ların getirilmesi yasaktır. Bu durumda, her iki giriş aynı zamanda 0'sa ( $\bar{S}=\bar{R}=0$ ), flip-flopun çıkışları  $Q$  ve  $\bar{Q}$  birbirine göre tümleşik olmayacak, her ikisi 1 seviyesinde olacak ve bu durum kolayca belirli olmayan çıkışın meydana gelmesine neden olabilir.

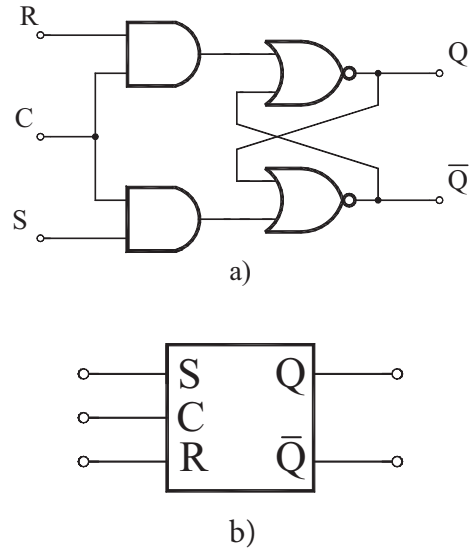
OVE mantıksal devreli  $\bar{S} \bar{R}$  flip-flopun Şek.4-8'deki mantıksal diyagramı, giriş sinyalleri OVE geçitlerine evirici devreler üzerinden getirilerek OYA türünden SR flip-flopa dönüşebilir ve çalışabilir.

#### 4.2.3. PALS-SİYALİN SEVİYESİYLE PALSLANAMIŞ SR FLİP-FLOP

Bu flip-flop S ve R girişlerin mantıksal seviyesiyle yönetilmelidir, ancak bunu Şek. 4-10 a)'da gösterilen bağlanma şekli uygulanarak C pals-sinyalinin yüksek mantıksal seviyenin ( $C=1$ ) meydana geldiği anda gerçekleşebilir. OYA devrelerle gerçekleşen **palslı SR flip-flopun** mantıksal yapısı Şek.4-10 b)'de gösterilmiştir, aynıısının OVE devrelerle uygulaması Şek. 4-11 a'da gösterilmiştir, sembolü ise Şek. 4-11 b)'de verilmiştir. S ve R girişlerinde uygun mantıksal gerilim seviyeleri getiriliyor, çıkışı ise aynı zamanda iki VE devreye giriş olarak getirilen C palsının mantıksal durumuyla kontrol edilen iki VE devrenin yer almasından dolayı, flip-flopun durumu üzerine efektif etki, sadece pals-sinyali için ön (yükselen, pozitif) kenarın meydana geldiği zaman gerçekleşiyor. Bu devre de mandaldır çünkü bunda da saydamlık olayı karakteristiktir. Ancak bu özellik sadece pals sinyalin yüksek olduğu ( $C=1$ ) durumda geçerlidir. Bu yüzden bu sinyale çalışma izni (olanağı) için E sinyali olarak bakabiliriz.



Şek.4-10. Palslı SR flip-flopun mantıksal yapısı



Şek.4-11. Palslı SR flip-flopun sembolü

S ve R girişleri sadece pals-sinyalle senkronize oldukları zaman flip-flopun durumuna etkileyebildiklerinden dolayı, bu girişlere senkron girişler denir, flip-flopa ise **senkronize** veya **palslı SR flip-flop** denir.



Palslı SR flip-flopun çalışması Tab. 4-6'da gösterilen geçiş ve çıkış tablosuyla ve Tab. 4-7'de verilmiş eksitasyon tablosuyla gösterilebilir. Onlar aslında asenkron SR flip flop için geçiş ve çıkış tablosuyla aynıdır, ancak bu tablolar sadece palsın alçak mantıksal seviyeden (0) yüksek seviyeye (1) geçtiği zaman geçerlidir ve mantıksal devrelerin çalışmasını sağlayan palsın ön kenarın ortaya çıkmasıyla meydana gelen kararlı yüksek seviyesi var. Bu durum, daha net olarak onun C pals girişli karakteristik denkleminde görülebilir:

$$Q^+ = (S + \bar{R} Q) C + Q \bar{C}, \text{ ya da } Q^+ = [(S + Q) \bar{R}] C + Q \bar{C} \quad (4-2)$$

Bu arada SR=0 koşulunun geçerli olması gerekiyor.

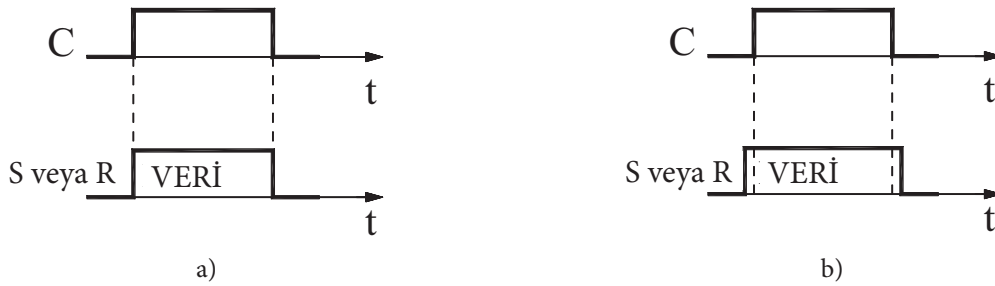
S	R	Q <sup>+</sup>
0	0	Q
0	1	0
1	0	1
1	1	?

Tab.4-6. SR flip flopun geçiş ve çıkış tablosu

Q	Q <sup>+</sup>	S	R
0	0	0	x
0	1	1	0
1	0	0	1
1	1	x	0

Tab. SR flip-flopun uyarma tablosu

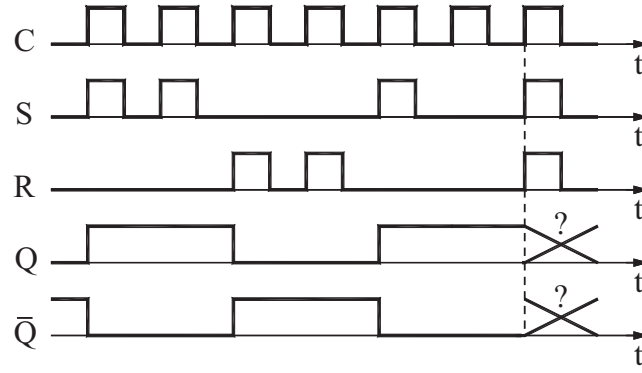
Palslı flip-flopun güvenilir çalışması için, C pals-dürtünün sürdüğü sürece S ve R girişlerin seviyeleri kararlı olmalıdır daha doğrusu, Şek. 4-12 a)'da gösterilmiş olduğu gibi S ve R girişlerinde veriler palsın ön kenarından önce getirilmelidir ve arka kenarından hemen sonra tamamlanmalıdır. Böyle flip-flop için pals sinyalinin seviyesiyle yönetildiği, yani bir seviyeden başka seviyeye geçtiği söyleniyor (İng. Level-triggered ya da pulse-triggered). Biz genelde Şek.4-12 b) ile açıklanan ideal durumları gözetleyeceğiz. Bu koşul geçerli değilse, yani S veya R girişlerden biri mantıksal seviyesini değiştirirse, bu değişiklik açık olan giriş devrelerinden geçecek, çünkü pals yüksek seviyededir. Böylece pals-dürtü sırasında çıkışta değişiklik meydana gelecek.



Şek. 4-12. Pals seviyesiyle yönetilen SR flip-flopta giriş sinyallerin uyumluluğu

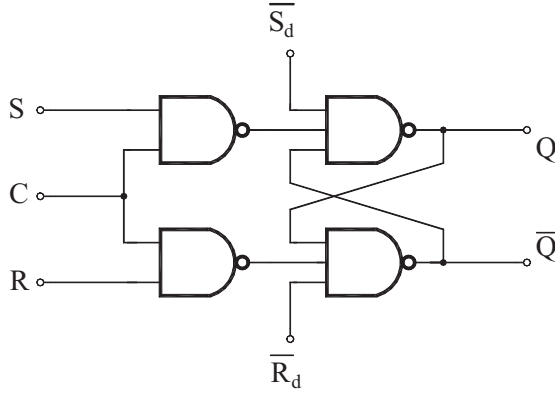
Şek.4-13'teki örnekle bu şekilde palstanmış SR flip-flopun çalışma prensibi gösterilmiştir. Bu arada başlangıç durumunun sıfırlanmış olduğu tahmin edilmiştir:  $Q=0$ ,  $(\bar{Q}=1)$ . Sunulan zaman-sal diyagramlardan saydamlığın sadece  $C=1$  olunca meydana geldiği görülüyor çünkü sadece bu zaman aralığında girişlerde meydana gelen değişiklikler çıkışlara yansıyor (çünkü sadece o zaman birbirini "görebiliyorlar").



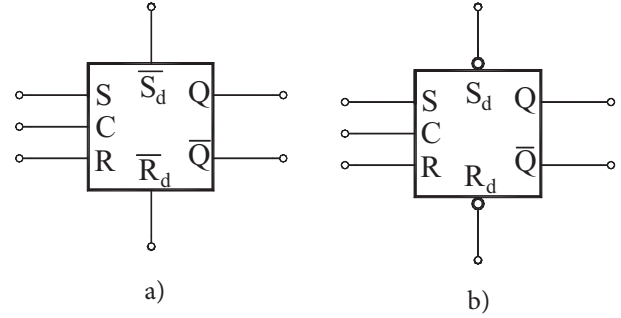


Şek. 4-13. Palslı SR flip-flopun çalışma şeklinin zamansal diyagramı

Palslı flip-floplarda, Şek.4-14'te gösterilmiş olduğu gibi genelde iki giriş daha ekleniyor. Böyle bir flip-flopun mantıksal sembolü Şek.4-15 a) ve b)'de verilmiştir. Yeni girişler alçak seviyede aktiftir ve  $\overline{S_d}$  veya  $\overline{PRS}$  ile (İng. PRESET, başlangıç ayarı) ve  $\overline{R_d}$  veya  $\overline{CLR}$  ile (İng. CLEAR, silme, başlangıç sıfırlama) işaretleniyor

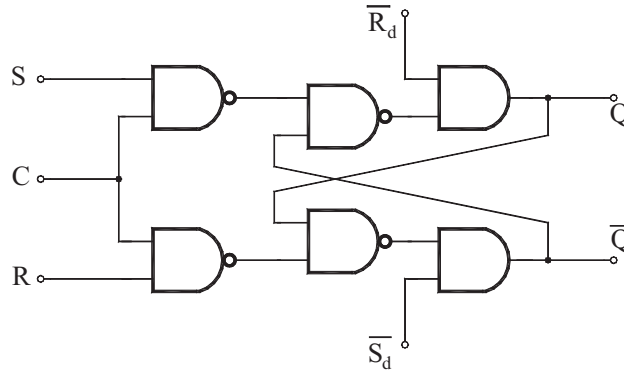


Şek.4-14. Doğrudan (asenكرون) girişli SR flip-flopun mantıksal diyagramı



Şek.4-15 . Doğrudan (asenكرون) girişli palslı SR flip-flopun sembolleri

Bu iki giriş, pals-sinyalinden bağımsız olarak flip-flopun çıkışına doğrudan etkiliyor, bu yüzden onlara *doğrudan* ya da *asenكرون* girişler denir. Onların görevi, flip-flopun başlangıç durumunu belirlemek ve gerekirse palstan bağımsız olarak flip-flop'u yönetmektir.  $\overline{S_d} = 0$  olunca, flip-flop başlangıçta yüksek seviyeye ayarlanıyor ( $Q=1$ ),  $\overline{R_d} = 0$  olunca ise, flip-flop başlangıçta alçak seviyeye ayarlanıyor ( $Q=0$ ). Bu girişler için de aynı zamanda aktif olamaz koşulu geçerlidir. Onların yasak kombinasyonu  $\overline{S_d} = \overline{R_d} = 0$  olacak. Böylece pals-dürtünün yokluğunda, flip-flopun durumu tamamen doğrudan girişlerle belirlenecek. Bu şekilde yapılmış doğrudan girişlerde aktif seviyeler sadece pals seviyesi alçak olunca getirilebilir. Ancak, pals seviyesi yüksekse, onlar da S ve R veri girişleriyle beraber aktif olacak, ancak aramaları ters olacak, örneğin  $S=1$  ve  $\overline{R_d} = 0$ , veya  $R=1$  ve  $\overline{S_d} = 0$ . Böyle durumda flip-flopun çıkışı Q belirlenmemiş olacak. Bu sorun, Şek. 4-14'te gösterilmiş olduğu gibi, flip-flopun çıkışında iki daha VE devrenin eklenmesiyle çözülüyor, öyle ki Şek. 4-16'daki yapılm elde ediliyor. Böyle durumda doğrudan girişler pals-sinyaline karşı üstünlük (hakimiyet) sağlıyor. Şöyle ki, flip-flop çıkışı, pals seviyesi C ve veri girişleri S ve R'den bağımsız olarak sadece  $\overline{S_d}$  ve  $\overline{R_d}$  girişlerin durumuna bağlı olacak.



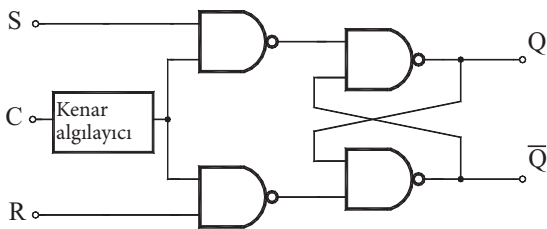
Şek. 4-16.Üstün doğrudan (asenkron) girişli palslı SR flip-flopun mantıksal diyagramı

Doğrudan girişli senkron flip-floplar ne zaman kullansak, genelde doğrudan girişleri palsın üzerine üstün olduklarını tahmin edeceğiz. Palslı flip-floplar, sadece doğrudan (asenkron) girişlerin pasif oldukları durumda, pals-sinyalin etkisi altında çalışabilirler ve onların davranması sadece S ve R senkron girişler bağlı olacak. Bunun için onlar yüksek seviyede korunmalıdır.

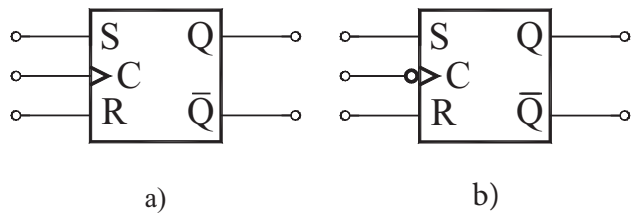
Bazı flip-floplar yapılarında doğrudan girişleri yüksek seviyede de aktif olabileceklerini söylemek gerekiyor. Bu durumlarda onlar  $S_d$  ve  $R_d$ , ya da PRS ve CLS olarak belirtiliyor. Böyle flip-floplar için yasak kombinasyon iki doğrudan girişe iki 1'in getirilmesidir, yani  $S_d=R_d=1$ . Flip-flopun çalışması, sadece doğrudan girişleri aynı anda alçak seviyede ( $S_d=R_d=0$ ) buldukları zaman, S ve R senkron girişlerinden kontrol edilecek.

#### 4.2.4. PALS-SİNYALİNİN KENARIYLA PALSLANAN SR FLİP-FLOP

Pals seviyesiyle yönetilen flip-floplar dışında, pratikte anahtarlamalı flip-floplar da geniş kullanım buluyor. Anahtarlamalı flip-floplar durumlarını sadece pals sinyalin kenarı meydana geldiği zaman değiştiriyor (İng.edge-triggered), hem de alçak seviyeden yüksek geçiş sırasında veya yüksek seviyeden alçak seviyeye geçiş kenarlarında. Birinci durumda, flip flop durumunun palsın pozitif (yükselen) kenarın meydana geldiği sırada değişmesidir, ikinci durumda ise negatif (düşen) kenarda flip flopun durumu değişiyor. Bu flip-floplarda pals sinyali yüksek olabilir ve girişlerin mantıksal durumunda her çeşit değişiklik meydana gelip çıkışların durumlarının üzerine hiçbir etkisi olmayabilir. Bu özellik, şimdiye kadar incelediğimiz flip-floplarda yoktu. Bu tür flip-floplar prensipte Şek.4-10'daki pals sinyal, önce Şek.4-17'ye göre pals sinyalinin kenar algılaması yapan (İng. Edge-triggered detector) devreye getirilerek elde edilebilir. Böyle flip-flopların işareti Şek.4-18 a) ve b)'de gösterilmiştir.

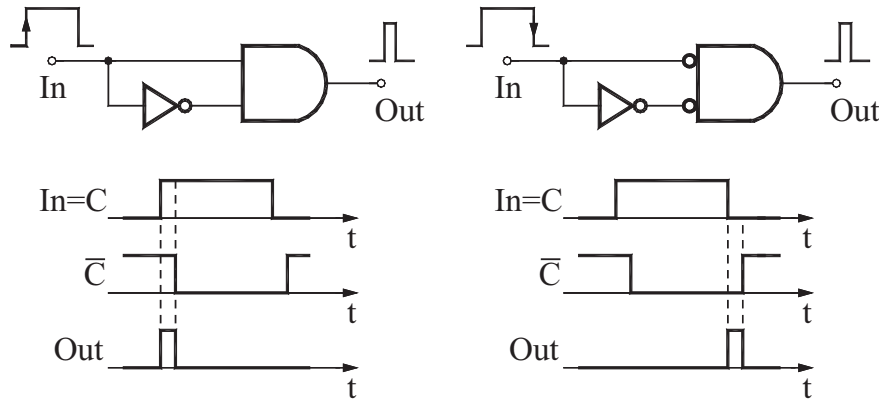


Şek.4-17. Anahtarlamalı flip-flopun mantıksal diyagramı



Şek.4-18. Anahtarlamalı Flip-flopun sembolleri

Yükselen ya da düşen kenarın algılayıcısı, pals-dürtüsünü sadece birkaç nanosaniyelik genişlikte çok dar dürtüyle değiştiriyor (dönüştürüyor). Şek.4-19 a) ve b)'de palsın yükselen ve düşen kenarların algılanması için tipik ve en basit devreler gösterilmiştir. Pals sinyali aynı anda VE devrenin iki girişine getiriliyor. Bu arada, bir giriş palsın girişi evirici devreden geçtikten sonra uyarılıyor. Eviricinin rolü, pals sinyalin ondan geçmesi nedeniyle birkaç nanosaniyelik çok az bir gecikme elde etmektir. Böylece VE devrenin girişinde zamansal olarak çok az oynamış evirilen sinyaller getiriliyor. Bu bağlantı şekli algılayıcının çıkışında tam palsın kenarı sırasına, eviricinin gecikme zamanına eşit süreli pozitif dürtünün oluşmasını sağlıyor (Şek.4-19 a)'ya göre ön kenarda, arka kenarda ise Şek.4-19 b'ye göre). Şek.4-19 b)'deki mantıksal VE devrenin yerine OYA devresi kullanılabilir.



a) ön kenar algılaması

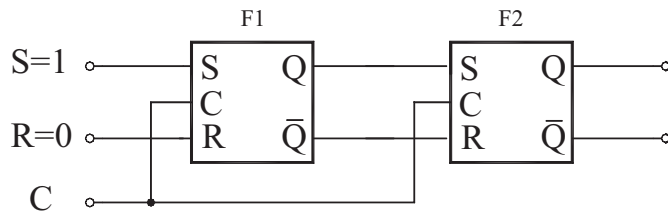
b) arka kenar algılaması

Şek.4-19. Kenar algılamanın uygulaması ve onun zamansal diyagramları

Pals sinyalin seviyesine değil, kenarın meydana gelmesine tepki gösteren böyle flip-flopların sembolik işaretlenmesinde pals Şek. 4-18 a) ve b)'de küçük üçgenle işaretleniyor ( $\Delta$ ). Şek. 4-18 a)'daki flip-flop sembolü palsın pozitif kenarıyla yönetilen flip-flop belirtiyor, Şek. 4-18 b) ise palsın negatif kenarıyla yönetilen flip-flop belirtiyor.

#### 4.2.5. MASTER-SLAVE YAPILI SR FLİP-FLOP

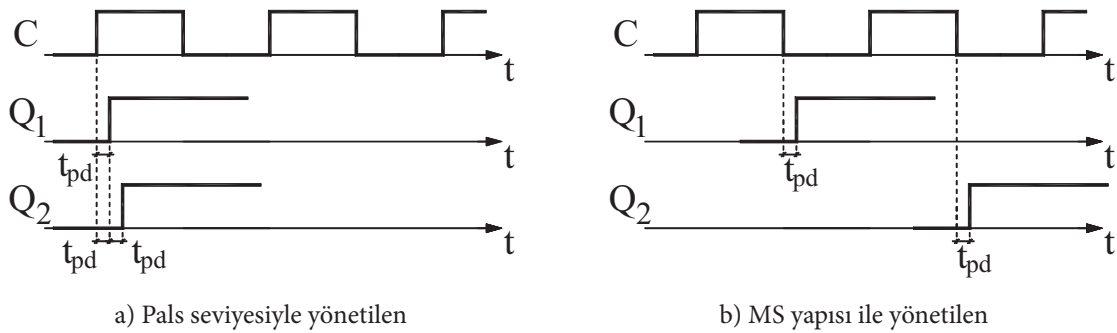
Palslı SR flip-flop çıkışları ( $Q$  ve  $\bar{Q}$ ) pals-sinyal girişinin yükselen kenarın meydana gelmesiyle değişiyor. Ancak, böyle davranış, dijital kurguların çoğunda hatalı çalışmaya yol açabilir. Sorun şundan kaynaklanıyor: dijital cihazlar, aralarında bağlı büyük sayıda flip flop içeriyorlar, öyle ki her flip-flop aktif geçişin meydana gelmesinden önce girişinde bulunan veriye karşılık veriyor, ancak aynı tepkiyi, çıkışlarını değiştiren diğer flip flopların çıkışlarındaki değişiklik sonucu olarak, elde edilen verilere karşı da gösteriyor. Bu sorunu açıklamak için, Şek. 4-20'de basamaklı bağlanmış iki SR flip-flop örneğini inceleyeceğiz.



Şek. 4.20. Basamaklı bağlanmış iki palslı SR flip-flop

Her iki flip-flopun, F1 ve F2, palsın ön kenarın meydana gelmesiyle tepki gösterdiklerini ve flip-flopun başlangıç durumları sıfırlanmış olduğunu ( $Q_1=0$  ve  $Q_2=0$ ) tahmin edeceğiz. Basamaklamamanın girişinde  $S_1=1$  ve  $R_1=0$  giriş kombinasyonu getiriliyor. Giriş kombinasyonuyla birinci flip-flopta birinci pals-döngüsü süresi içinde, ikinci flip flopta ise ikinci pals-döngüsü süresi içinde 1'in girilmesi gerekiyor. Daha doğrusu, birinci pals-dürtüsü meydana gelince  $Q_1$  çıkışın 1'e gitmesi gerekiyor ( $Q_1=1$ ),  $Q_2$  çıkışının ise 0'da kalması gerekiyor ( $Q_2=0$ ). İkinci pals dürtünün meydana gelemsiyle, birinci flip flopun 1'de kalması gerekiyor ( $Q_1=1$ ), ikinci flip-flopun çıkışın ise 1'e geçmesi gerekiyor ( $Q_2=1$ ). Ancak bu durumda kurgu doğru çalışmayacak. Palsın ön kenarı etkinleşince birinci flip flop 1'e gidiyor ( $Q_1=1$ ), çünkü  $S_1=1$  ve  $R_1=0$ , ancak sinyalin flip floptan geçmesi için gereken süreye  $t_{pd}$  gecikme de vardır. Bu kısa süre için ikinci flip-flopun çıkışı da değişmiyor çünkü  $S_2=Q_1=0$  ve  $R_2=\overline{Q_1}=1$ . Ancak, tDP'nin tamamlanmasından hemen sonra,  $S_2$  1'e gidiyor ( $S_2=1$ ),  $R_2$  ise sıfıra ( $R_2=0$ ). Ancak, C palsının hala aktif olduğundan dolayı, ikinci flip-flopun çıkışı, gerektiği gibi ikinci pals-dürtüsü sırasında değil, daha birinci pals-dürtünün sürdüğü sırasında 1'e gidiyor ( $Q_2=1$ ). Bu durum Şek.4-21 a)'daki zamansal diyagramla gösterilmiştir.

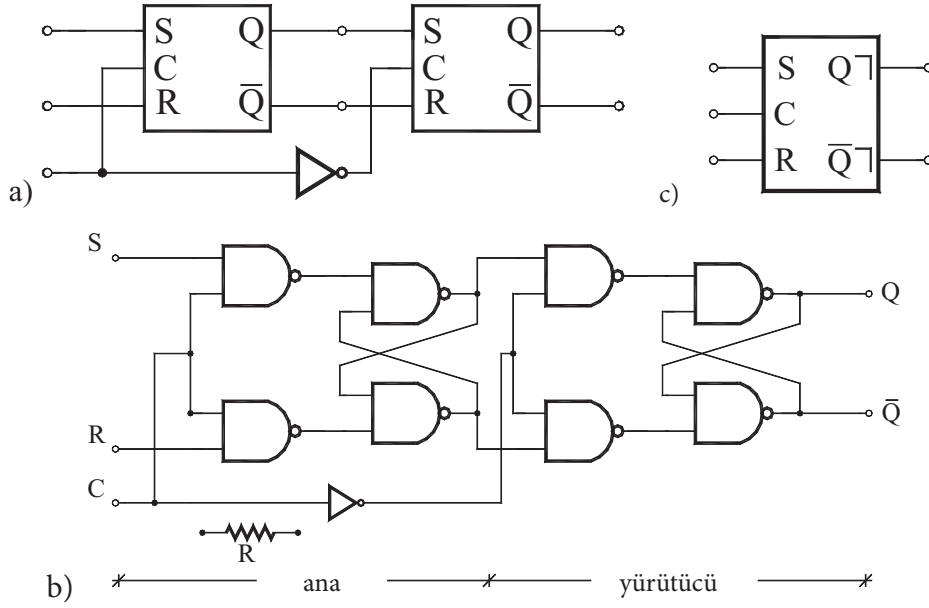
Bunun engellenmesi için bir yöntem duraklamaların sürdüğünden çok daha az süren dürtüleri olan pals sinyalin üretilmesidir.  $t_{pd}$  gecikme zamanından daha kısa süre içinde aktif seviyede kalan böyle bir pals-sinyal flip-flopun taşınması sırasında güvenilir olmayan çalışmaya yol açabilir, çünkü bu süre flip-flopun giriş verilerine karşılık vermesi için yeterli olmayabilir. Kısaca, flip-flopun mantıksal devrelerinin etkin oldukları süre gerekenden daha kısadır. Bu yüzden, bu sorunların giderilmesi için farklı bir çözüm uygulanıyor. Bu çözüm çıkışın değişmesinden önce girişler ve çıkışlar arasındaki bağlantının kesilmesini sağlıyor. **Ana-yürütücü** ya da **MS** (İng. master-slave, ana-yardımcı, iki bellekli) **flip-flopun** kullanımından söz ediliyor. MS flip-flop palsın mantıksal devrelerin çalışma olanağında çalışma olanaksızlığına geçtiği kenarı yönetiyor. Bizim durumda senkronizyon pals dürtünün düşüş kenarı söz konusudur.



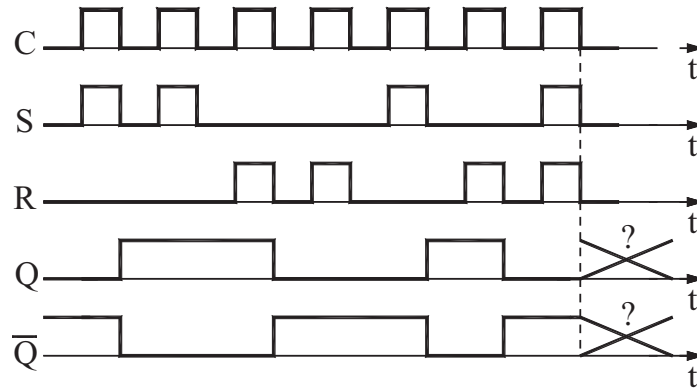
Şek.4-21. Basamaklamalı bağlanmış iki palslı SR flip-flopun zamansal diyagramları

MS flip-flopun aynı uyarıya yansımaları Şek.4-12 b)'de gösterilmiştir. Birinci flip-flop çıkışları  $Q_1$  ve  $\overline{Q_1}$  palsın düşüş kenarında değişiyor ve ikinci flip-flopa etkileyemiyor, çünkü pals seviyesi alçaktır.  $Q_2$  ikinci çıkışın değişikliği pals-sinyalin düşüş kenarının meydana gelmesiyle gerçekleşiyor. Bu durumda kurgu doğru çalışıyor. Bunun dışında, ikinci pals-döngüsü sırasında veriler gereken seviyelere sağlamlaşacak:  $S_2 = Q_1 = 1$  ve  $R_2 = \overline{Q_1} = 0$ .

Flip-flopun MS yapısı Şek. 4-10'dan iki palslı flip-flopun basamaklı bağlanmasıyla elde ediliyor. Böylece SR MS flip-flopun mantıksal yapısını gösteren Şek. 4-22 a) ve b) elde ediliyor. Onun mantıksal sembolü Şek. 4-22 c)'de gösterilmiştir.



Şek. 4-22. SR MS flip-flopun mantıksal yapısı (a, b) ve mantıksal sembolü (c)



Şek. 4-23. Palslı SR MS flip-flopun davranmasının zamansal diyagramları

Bu flip flopun çalışması Şek.4-23'te verilen zamansal diyagramlarla gösterilmiştir. Çalışma prensibi sıradan SR flip-flopun çalışmasıyla aynıdır. Tek fark çıkışın mantıksal durumlarının, pals sinyalin arka kenarında meydana gelmeleridir. Bu özellik birbirine tümleştirici palsla palslanmış iki SR flip-flopun kullanımıyla sağlanmıştır.

Şöyle ki, palsın ön kenarı ana (İng.master, hakim, bellek) flip-flopun durumuna etkiliyor, yürütücü (slave, robot, çıkış) flip-flopun durumu ise değişmeden kalıyor çünkü onun girişinde palsın düşüş kenarı meydana geliyor. Palsın arka kenarı meydana gelince, ana flip-flop üzerine etki sona eriyor, ancak bu gerilim şeklinin evirici değeri, yani ön kenar şimdi yürütücü flip flopta meydana geliyor ve yürütücü flip-flop  $Q_m$  ve  $\bar{Q}_m$  çıkışlarının etkisi altında durumunu değiştiriyor, çünkü  $Q_m$  ve  $\bar{Q}_m$  şimdi yürütücü flip-flopun girişleridir:  $S_s = Q_m$  ve  $R_s = \bar{Q}_m$ . Buna göre pals sabit olduğu sürece (seviye 1'de veya 0'da), flip-floplar birbirinden ayrıdır, bununla beraber MS flip flopun girişi de çıkışından ayrıdır. Palsın ön kenarının meydana gelmesiyle, ana flip-flop çalışıyor, yürütücü flip-flop is kapalıdır. Böylece S ve R girişlerinde bulunan giriş verileri ana flip-flopta yazılıyor. Yürütücü flip-flop verileri, palsın arka kenarının meydana gelmesiyle alıyor, çünkü o zaman devreye giriyor. O anda veriler çıkışta da meydana gelerek, ana flip-flop devre dışında bulunuyor.

MS flip-flopta da, doğru çalışması için, palsın aktif olduğu sürece verilerin değişmesi yasaktır, yani S ve R girişlerinde bulunan sinyaller C=1 olduğu sürece kararlı olmalıdır. Bu dezavantaj, ana flip-flopun mantıksal yapısı, pals sinyalini sadece ön kenarına tepki gösterecek şekilde değiştirilerek giderilebilir.

Flip-flopların MS yapılarında sıkça flip flopu doğrudan ayarladığı veya sildiği asenkron girişler ekleniyor. Onlar yüksek ya da alçak seviyede aktif olabilir ve yapıma bağlı olarak sadece palsın durakladığı anlarda getirilebilir, ya da palsın etkisine tamamen üstünlük sağlayabilirler (onunla hakim olabilir).

Şimdiye kadar açıkladıklarımızı gözden geçirirsek, herhangi SR flip flop çeşitinin uygulanmasını sınırlandıran temel dezavantajı, S ve R girişlerin ikisinin de aynı anda aktif olmamaları gerektiği, yani yasak giriş kombinasyonunun var olduğunu söyleyebiliriz. Böyle bir uyarımın meydana gelmemesi sistemden bile sağlanması, gürültü ve engellerin etkisinden dolayı, izin verilen bazı giriş kombinasyonu geçersiz olabilir. Bu yüzden geçersiz giriş kombinasyonu meydana geldiği zaman girişlerden biri hakim olduğu SR flip-floplar da yapılıyor ve böyle durumda çıkışı bu hakim girişle belirlenecek. Yine de yasak giriş kombinasyon sorunu, her giriş için tanımlanan durumları olan farklı flip-flop türlerinin projelenmesiyle çözülmüştür. Bu tür flip-flopların incelenmesi devamdaki bölümlerde sunulacaktır.

### 4.3. JK FLİP-FLOP

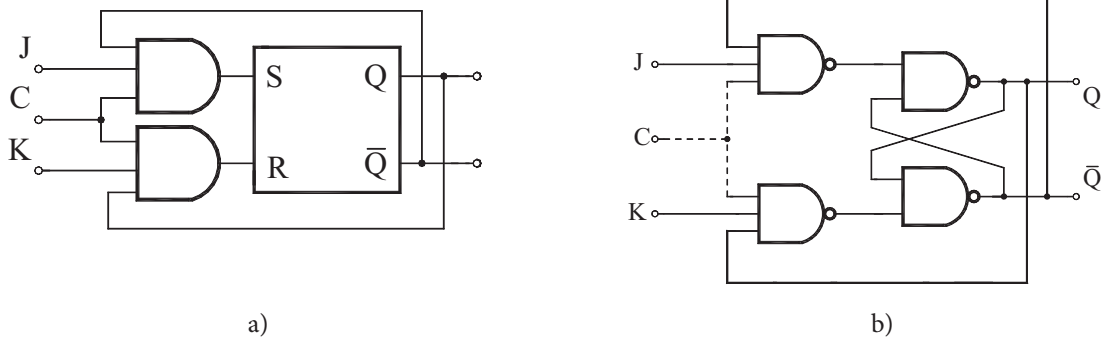
**JK flip-flop** mantıksal yapısına göre yasak giriş kombinasyonu olmayan değişmiş SR flip-floptur. Onun mantıksal yapısını Şek. 4-24 a) ve b) gösterilmiş ve şekilden JK flip-flopun iki çapraz geri bağlantı ile yapılmış SR flip-flopun uygulanmasıyla elde edildiği görünüyor. Bu arada, JK flip-flopun palsli olması için giriş OVE devresinde, Şek. 4-24'te kesilmiş çizgilerle gösterilmiş olduğu gibi palsı taşıyacak bir girişin daha eklenmesi gerekiyor. JK flip-flopun sembolik işareti Şek. 4-25'te verilmiştir, geçiş ve çıkış tablosu tab. 4-8 olarak işaretlenmiş, eksitasyon tablosu ise tab. 4-9 olarak verilmiştir.

Geçiş ve çıkış tablosuna dayanarak JK flip-flopu için karakteristik denklem yazılabilir. Asenkron JK flip-flop için, şu mantıksal denklemle ifade ediliyor:

$$Q^+ = J\bar{Q} + \bar{K}Q \quad (4-3)$$

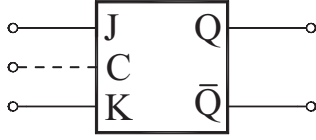
C pals girişli senkron JK flip-flopun mantıksal denklem şu şekildedir:

$$Q^+ = (J\bar{Q} + \bar{K}Q) C + \bar{Q}C \quad (4-4)$$



Şek.4-24. JK flip-flopun mantıksal yapısı

JK flip-flop için yasak giriş kombinasyonu yoktur. Şöyle ki, kombinasyon tablosundan ve geçiş fonksiyonundan çalışma prensibi ortaya çıkıyor: Her iki girişin, J ve K'nin 1 seviyesinde olduğu kombinasyon dışında, tüm giriş kombinasyonu için JK flip-flop SR flip flop gibi çalışıyor. Bu arada S için analog giriş J'dir, R girişine ise K girişi karşılıklıdır. JK flip-flopun J=1 ve K=1 giriş kombinasyonu ile uyarıldığı zaman, çıkış durumu belirlenmiş olacak ve önceki durumun tümleşik durumu, yani  $Q^+ = \bar{Q}$  olacak.



Şek.4-25. JK flip-flopun mantıksal sembolü

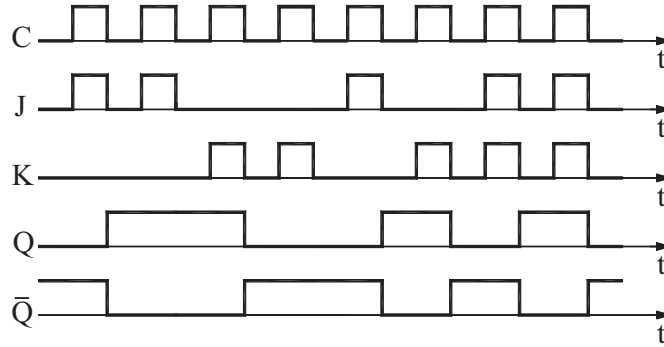
J	K	$Q^+$
0	0	Q
0	1	0
1	0	1
1	1	$\bar{Q}$

Tab.4-8. JK flip flopta geçiş ve çıkış tablosu

Q	$Q^+$	J	K
0	0	0	x
0	1	1	x
1	0	x	1
1	1	x	0

Tab.4-9. JK flip-flopun uyarma tablosu

JK MS flip-flopun çalışmasıyla ilgili bir örnek Şek.4-26'da verilen sıfırlanmış başlangıç durumunda zamansal diyagramlarıyla gösterilmiştir.



Şek.4-26. Palslı JK flip flopun çalışmasının zamansal diyagramları

Palslı JK flip-flopların alçak seviyede de aktif olmaları için genelde asenkron girişler ekleniyor ve  $S_d$  (PRS) ve  $R_d$  (CLR), ya da yüksek seviyede aktif olursa Sd (PRS) ve Rd (CLR) ile işaretleniyor.

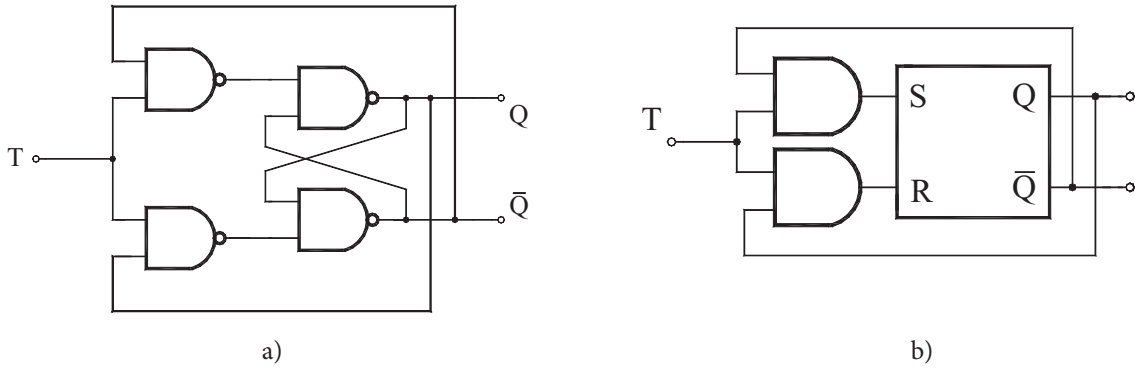
JK flip-floplarda çalışma esnasında belirli sorunlar yaşanabilir. Örneğin, J=1 ve K=1 olduğu ve flip-flopun başlangıç durumu  $Q=0$  ( $\bar{Q}=1$ ) olduğunu tahmin edelim. Palsın yükselen kenarı meydana gelince, pals-sinyalinin dürtüsünün sürdüğü süreden daha kısa süren belirli gecikme tpd ardından, çıkışta  $Q=1$  ( $\bar{Q}=0$ ) durumu elde edilecek. Bu değişiklik çıkıştan girişe dönüyor, pals dürtüsünün hala sürdüğünden dolayı, tpd süresinden sonra çıkışın yeniden değişmesine yol açacak, yani  $Q=0$  ( $\bar{Q}=1$ ) olacak ve bu değişiklik yeniden girişe dönüyor. Bu işlem pals-dürtünün sürdüğüne dek tekrarlanacak. Buna göre, palsın aktif olduğu sürece (C=1) çıkış 0 ile 1 arasında değişiyor. Pals pasif (alçak) seviyeye düşünce, çıkış belirlenmiş olmayacak. Olası bir çözüm dönüş sinyallerin gecikmesidir. Böylece tpd gecikmesi pals dürtüsünün süresinden daha uzun zaman sürüyor, ancak aynı zamanda duraklamadan daha az sürüyor, ikinci dürtü ise tdb gecikme zamanından daha kısa sürecektir.



Her iki çözümün başka sorunlar yarattığından dolayı, bu dezavantaj çıkışın palsin arka kenarın meydana gelmesiyle değişerek, ya da JK flip-flop Şek. 4-22'te benzer MS yapıyla gerçekleştirilerek gideriliyor. Bu durumda çıkış girişi dönüyor, ancak şimdi ikinci (yürütücü) flip-flop için pals-sinyalin girişi, dönen sinyalin ek değişikliğe yol açmadığından dolayı alçak seviyede bulunuyor.

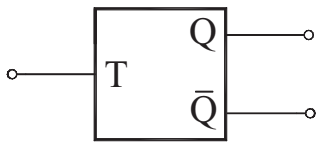
#### 4.4. T FLİP-FLOP

Bu flip-flop türünün T ile işaretlenmiş sadece bir girişi var. T flip-flopun mantıksal yapısı Şek. 4-27 a) ve b)'de gösterilmiştir, sembolü ise Şek. 4-28'de verilmiştir. Şek. 4-27'den görüldüğü gibi bu flip-flop türü de SR (ya da JK) flip flopundan iki çapraz geri bağlantıyla ve iki girişin bir girişe birleşmesiyle elde ediliyor.



Şek. 4-27. T flip-flopun mantıksal yapısı

Geçiş ve çıkış tablosu tab.4-10 olarak gösterilmiştir, uyarma tablosu ise tab. 4-11 olarak sunulmuştur. Onlarda kolayca T flip-flopun çalışma şekli kolayca anlaşılabilir.



Şek. 4-28. T flip-flopun mantıksal sembolü

T	Q <sup>+</sup>
0	Q
1	$\bar{Q}$

Tab. 4-10. T flip-flopunda geçiş ve çıkış tablosu

Q	Q <sup>+</sup>	T
0	0	0
0	1	1
1	0	1
1	1	0

Tab. 4-11. T flip-flopun uyarma tablosu

T girişine 0 getirilirse (T=0), o zaman çıkışın sıradaki durumu önceki duruma kıyasen değişmiyor:  $Q^+ = Q$ . T girişinde ise 1 getirilirse (T=1), o zaman çıkışın sıradaki durumu önceki durumun tümleyeni olacak  $Q^+ = \bar{Q}$ . Böyle çalışma şeklinden dolayı, T flip-flopa geçiş flip flopu (İng. toggle veya trigger) da denir.

Geçiş tablosuna dayanarak T flip-flopun karakteristik denklemi de yazılabilir. Bu denklem şu mantıksal denklemle verilmiştir:

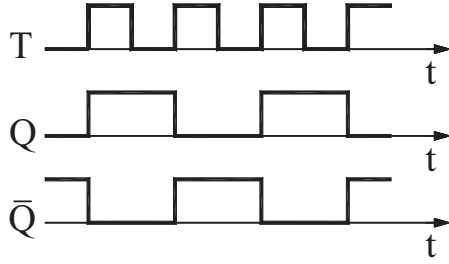
$$Q^+ = (Q \oplus T) \text{ veya } Q^+ = Q\bar{T} + \bar{Q}T \quad (4-5)$$

C pals ile palslı flip-flop için karakteristik denklem şöyle olacak:

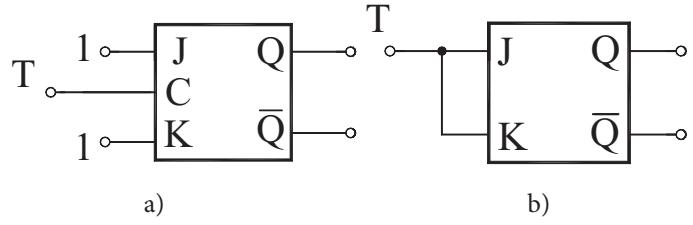
$$Q^+ = (Q \oplus T)C + Q\bar{C} \text{ veya } Q^+ = (Q\bar{T} + \bar{Q}T)C + Q\bar{C} \quad (4-6)$$



Şek. 4-28'de, sıfırlanmış başlangıç durumu  $Q=0$  ( $\bar{Q}=1$ ) ile T flip-flopun çalışmasıyla ilgili bir örnek verilmiştir.



Şek. 4-28. T flip-flopun çalışmasının zamansal diyagramları



Şek. 4-29. T flip flopun JK flip flopla yapılması

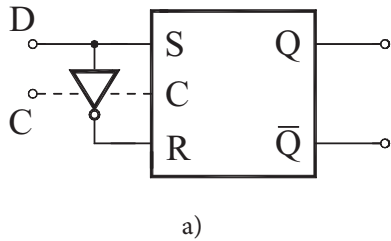
JK flip flopun tab. 4-8'de verilen karakteristik tablosuna bakılırsa, iki girişin bir girişe bağlanmasıyla T flip-flopuna dönüşüm yapıldığı görülüyor. Benzer şekilde, pals aktif olduğu zaman J ve K girişlerine 1'in getirilmesiyle ( $J=K=1$ ), sıradaki durumun önceki durumunun tümleyeni olmasında yol açıyor. Buradan, Şek. 4-29'da gösterildiği gibi, girişleri sabit olarak mantıksal 1 durumunda kalan palslı ve palslı olmayan JK flip flopların uygulanmasıyla T flip flop yapımları elde ediliyor.

Giriş dürtüsünün meydana gelmesiyle T flip-flopun durumu değişmesinden dolayı, flip flopun her dürtünün algılamasını yaptığı ortaya çıkıyor. Bu özellik giriş dürtülerinin sayılması için uygulanabilir. Bu yüzden onlara sayaç flip-flopu denir. Bununla ilgili olarak, devamdaki konulardan birinde incelenecek olan sayaç ağlarının gerçekleşmesi sırasında T flip-flopu temel yapısal eleman olarak kullanılıyor.

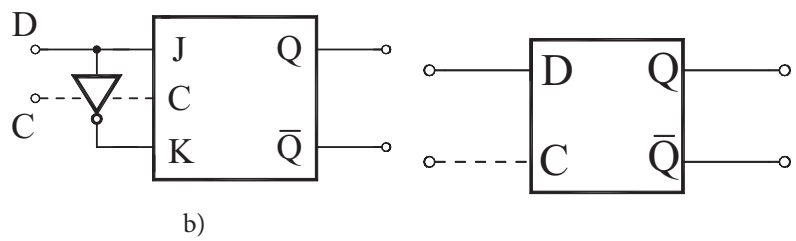
#### 4.5. D FLİP-FLOP

SR ve JK flip-flopların 4-6 ve 4-8 karakteristik tablolarına bakılınca ve incelenince, birbirine tümleşik girişlerin giriş kombinasyonu olarak getirilmesi flip-flopun sıradaki durumu 1 veya 0 olmasına neden olması kolayca görünebilir. Buradan, yapımında SR (ya da JK) flip flop ve bir evirici kullanan D flip-flop Şek. 4-30'da gösterilmiştir. Bu flip-flopun, verinin getirildiği D ile işaretlenmiş sadece bir girişi vardır. Amaç bu girişe getirilen bir biti depolayan dijital bileşenin elde edilmesidir. Böylece flip-flopun sıradaki durumu her zaman D girişinin mantıksal durumuyla aynı olacak.

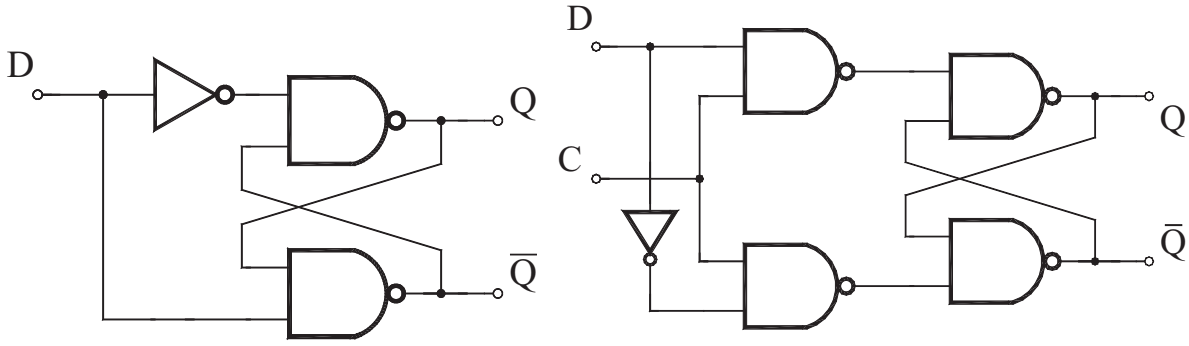
D flip-flopun sembolik işareti Şek. 4-31'de verilmiştir, mantıksal yapısı ise Şek. 4-32 a) ve b)'de gösterilmiştir. Şekilden görüldüğü gibi, D flip flopun çalışması tek bir sistemde diğer dijital devrelerle senkronize olması gerektiğinde, şekillerde kesilmiş çizgilerle gösterilen pals (C) için bir giriş daha ekleniyor.



Şek. 4-30. D flip-flopun SR ve JK flip-flopla uygulaması



Şek. 4-31. D flip-flopun mantıksal sembolü



Şek. 4-32. D flip-flopun mantıksal yapısı

Bu flip-flopun komple çalışması en iyi olarak onun geçiş ve çıkış tablosu 4-12 ve uyarma (ek-sitasyon) tablosu yardımıyla gösterilebilir.

T	$Q^+$
0	0
1	1

Tab. 4-12. D flip flopun geçiş ve çıkış tablosu

Q	$Q^+$	T
0	0	0
0	1	1
1	0	0
1	1	1

Tab. 4-13. D flip-flopun uyarma tablosu

Bu tablolara dayanarak D flip-flopun çalışma prensibi anlaşılabilir. Q çıkışın önceki durumu ne olursa olsun, flip-flopun sıradaki durumu D girişine getirilen durumla aynı olacak:  $Q^+ = D$ , öyle ki D=0 ise o zaman  $Q^+ = 0$  olacak, D=1 ise o zaman  $Q^+ = 1$  olacak. Buradan aslında D flip-flopunun kısaltılmış ismi de kaynaklanıyor. Şöyle ki, girişteki sinyal çıkış iletiliyor, yani girişte bulunan veri (İng. *Data*) çıkışa kadar iletiliyor. Bu arada her zaman sinyalin flip-floptan geçmesi yüzünden, çok az da olsa belirli gecikme (İng. *Delay*) vardır.

Palslı D flip-floplarda D girişi C pals sinyalinin meydana gelmesine bağlıdır ve bu yüzden C ile her zaman eşzamanlı olmalıdır. Bu flip-flopta palsın ön kenarının meydana gelmesiyle, pals yüksek olana kadar, ya da aktif seviyesi olduğu sürece, çıkış girişi takip ediyor. Palslı flip-floplarda genelde palstan bağımsız ve pals üzerine hakimiyet sağlayan bir veya iki doğrudan asenkron giriş ekleniyor.

D flip-flopun çalışma şekli, analitik biçimde de, karakteristik denklemi yardımıyla açıklanabilir:

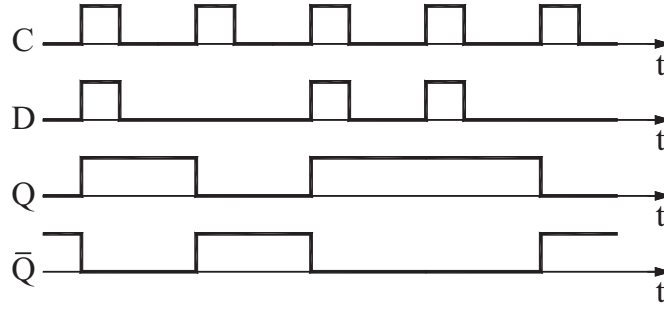
$$Q^+ = D \quad (4-7)$$

Asenkron flip-flop için

$$Q^+ = DC + Q\bar{C} \quad (4-8)$$

ya da senkron flip-flop için

D flip-flopun çalışma prensibi, Şek. 4-33'te gösterilen D flip-flop girişinin ve çıkışının zaman-sal diyagramları örneğiyle gösterilmiştir.



Şek.4-33. D flip-flopun davranış zamansal diyagramları

Bu flip-flop en çok az kapasiteli bellek bileşenleri olarak sabit ve ötelemeli yazmaçların yapılı-mında kullanılıyor. Bu yazmaçlar sıradaki konuda inceleme konusu olacak ve daha detaylı olarak işlenecektir.

D flip-flopun pratik uygulaması, RS ve JK flip-floplara kıyasen daha yüksek yoğunlu paketle-me sağlıyor. Bu yüzden, D flip-flop genelde daha karmaşık ardışık bileşenlerin yapılı-mında temel bileşen olarak kullanılıyor.

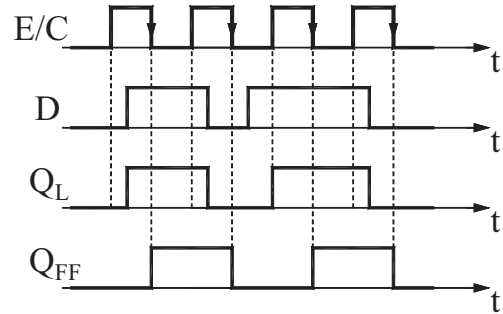
#### 4.5.1 KİLİTLEME DEVRESİ

Yukarıdaki açıkladıklarımızdan, kilitleme (tutma) devresinin ya da mandalın (leçin), aslında tüm flip-flopların temel yapısal birimi olduğunu söyleyebiliriz. Onun en önemli özelliği çalışma sırasında sunduğu saydamlıktır: giriş ve çıkış iletişimdedir ve “birbirini görebiliyorlar”.

D flip-flopun mandal olarak en önemli pratik uygulamalarından biri, verileri kabul eden, kilitleyen (tutan) ve belirli anda onları aktaran devrelerin gerçekleşmesidir. Kilitleme devresinin iki girişi var: biri verinin getirildiği D girişidir, diğeri ise veri sinyalinin D girişinden Q çıkışına geçirilmesini kontrol ettiği için izin girişi olarak adlandırılan E girişidir.

İşlem	Girişler		Çıkış
	E	D	$Q^+$
Geçiriyor (Q, D'yi izliyor)	1	0	0
		1	1
Kilitliyor (Q kilitlidir)	0	x	Q

Tab. 4-14. Kilitleme devresinin kombinasyon tablosu



Şek. 4-14. Kilitleme devresinin davranışının zamansal diyagramları

Bununla ilgili olarak:

- ⊕ Q çıkışı D girişini E aktif olduğu sürece izliyor, yani ön kenarın meydana gelmesinden yüksek seviyede bulunana kadar (E=1);
- ⊕ Q çıkışı, E izin girişinin arka kenarı sırasında ya da izin girişinin yüksek seviyeden alçak seviyeye geçtiği anda (E=1'den E=0'a düşüyor) D girişinin durumunu kilitliyor (tutuyor). Q çıkışının mevcut durumu, E izin girişinin yüksek seviyeden alçak seviyeye geçişi sırasında, E pasif olduğu sürece, ya da E alçak seviyede (E=0) bulunduğu sürece değişmiyor.

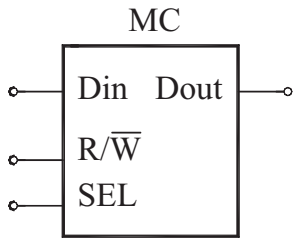
$$Q^+ = DE + D\bar{E} \quad (4-9)$$

Çalışma prensibi 4-14 işlevsel tablosuyla ve (4-9) karakteristik denklem yardımıyla açıklanabilir. Ayrıca, Şek. 4-24'te verilen giriş uyarısı için mandalın davranışıyla ilgili basit örneği gösteren zamansal diyagramları verilmiştir. D mandalın ile palsın düşen kenarıyla yönetilen D flip-flopun çalışması arasındaki farkın belirlenmesi için, Şek. 4-34'te aynı giriş sinyaliyle uyarılan her iki bilimde çıkışın zamansal diyagramları gösterilmiştir.

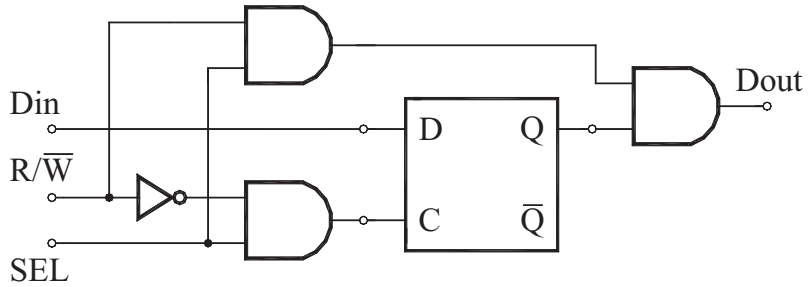
#### 4.5.2. TEMEL BELLEK HÜCRESİ

Şimdiye kadar bahsedilenlerden, flip-flopun bir bitlik verinin depolanabileceği temel bellek hücresi (MC) olarak kullanılabilmesi tahmin edilebilir. Farklı bellek hücreler yapımları vardır, ancak hepsi için ortak özellik birkaç girişin ve en az bir çıkışın olmasıdır.

Burada inceleyeceğimiz bellek hücresi için belirli bir içeriğin (bir bitli verinin) yazdırılması için bir girişi (Din) olduğunu, ikinci girişle (SEL) hücre seçimi yapıldığını ve üçüncü girişle (R/ $\bar{W}$ ) gerçekleşen işlemin, depolanmış verinin okunmasını ya da yeni verinin yazılmasına, seçimi yapıldığını tahmin edeceğiz. Hücre çıkışından (Dout) içeriğin (depolanmış verinin) okunması gerçekleşiyor.



Şek.4-35. Mantıksal sembol



Şek.4-36. Bellek hücresinin mantıksal yapısı

Böyle bir bellek hücresinin sembolik işareti Şek. 4-35'te gösterilmiştir, onu uygulayan mantıksal diyagram ise Şek. 4-36'da verilmiştir. Verilen şekillerden C kontrol girişiyle D mandalın merkezi yeri olduğu görülebilir. Bir bellek hücresinin çalışması gereken prensibi devamda açıklayacağız.

Bellek hücreyle çalışabilmek için, daha doğrusu içinde yazılmış bitin (verinin) okunabilmesi ya da yeni verinin yazılabilmesi için, onun aktif olması gerekiyor. Bellek hücresinin aktif olması için, adreslenmiş olması gerekiyor. Adresleme, adres (seçim) hattına yüksek mantıksal seviyeli sinyalin getirilmesiyle yapılıyor (SEL=1). Bu durumda iki olanak vardır:

- ⊕ **Yeni verinin YAZDIRILMASI:**  $R/\bar{W}$  sinyali alçak seviyede bulunuyorsa ( $R/\bar{W} = 0$ ), o zaman eviriciden 1 elde edilerek VE devresinden geçiyor ve izin sinyalini aktifleştiriyor ( $C=1$ ). Böylece Din girişinde bulunan veri flip-flopta yazılıyor, çıkışta ise sıfır elde ediliyor ( $Dout=0$ );
- ⊕ **depolanmış verinin OKUNMASI:**  $R/\bar{W}$  sinyali yüksek seviyede bulunuyorsa ( $R/\bar{W} = 1$ ), o zaman bir taraftan izin girişi alçak seviyede bulunarak ( $C=0$ ), Din veri hattının seviyesinin flip-flopun durumuna etkilemesine izin vermeyip eski durumunu koruyor; diğer taraftan ise çıkış VE devresi açılarak veri (flip-flopun mevcut durumu) çıkış hattında meydana geliyor ( $Dout=Q$ ).

Bütün zaman sadece aynı bir hücreyle değil, diğer hücrelerle de çalışıldığından dolayı, bellek hücresinin seçilmiş olmaması olanağı da olmalıdır. Bu durumda, hücre adreslenmiş olmadığı zaman, seçim hattının seviyesi alçaktır ( $SEL=0$ ), pasiftir ve onda girilmiş olan veriyi koruyor. Hücre seçilmemiş olduğu zaman, izin girişi de sıfıra gidiyor ( $C=0$ ). Böylece D flip-flopu mevcut durumunu koruyor ve değiştirmiyor, aynı zamanda ise çıkışta alçak seviye elde ediliyor ( $Dout=0$ ).

Şek. 4-35'te sunulan bellek hücresinin çalışma prensibi, 4-15 işlevsel tabloyla daha kompakt şekilde verilmiştir.

İşlem	Girişler			Çıkışlar	
	SEL	$R/\overline{W}$	Din	Q+	Dout
Edilgenlik (Pasiflik)	0	x	x	Q	0
Yazdırılma	1	0	0 1	0 1	0
Okuma	1	1	x	Q	Q

Tab.4-15. Bellek hücrenin işlevsel tablosu

Böyle asenkron bellek hücreleri dışında, çalışması hücrede tüm girişleri ve çıkışları kontrol eden ek pils sinyeli ile belirlenen senkron bellek hücreleri de vardır. Bu durumda bellek hücresi asenkron bellek hücresi gibi aynı şekilde davranıyor, sadece burada okuma ve yazdırma tam olarak belirlenmiş anda, pilsin ön ya da arka kenarın meydana gelmesiyle yapıyor.

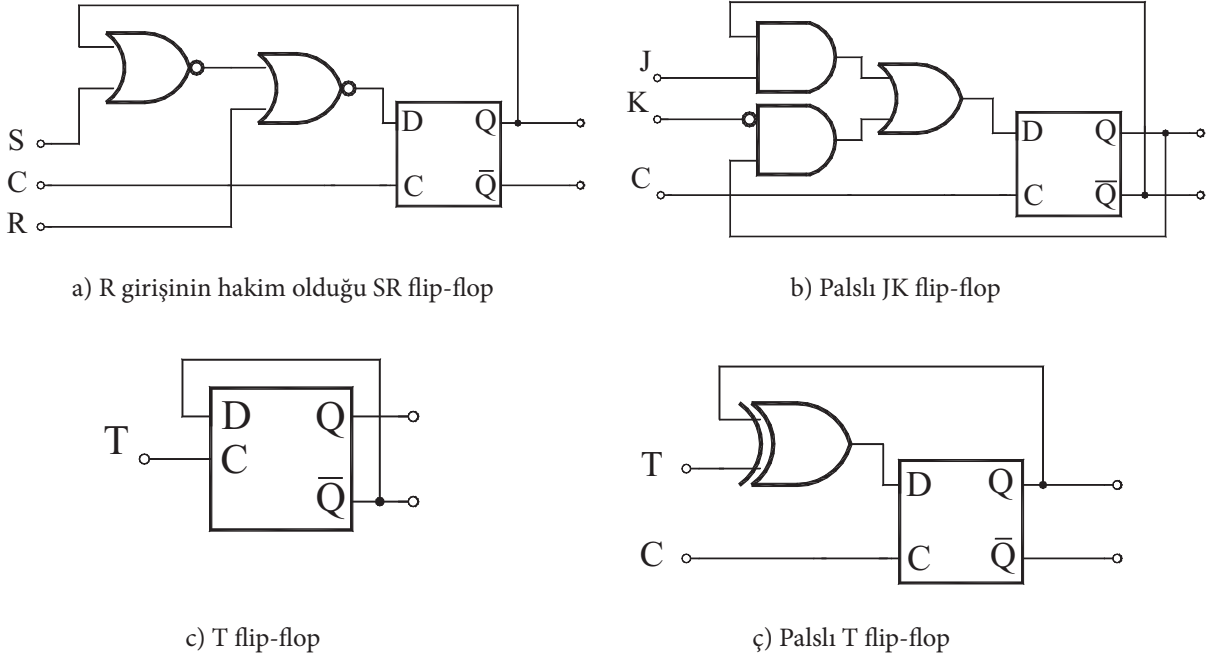
Gerçek bellek bileşenlerde veri için sadece bir girişi olan bellek hücreleri olabilir. Okuma sürecinde bu hat çıkış hattı olur, yazdırma sırasında ise giriş hattıdır. Ayrıca, bellek hücresi adreslenmemiş olduğu zaman, veri hatları genelde üçüncü durumda bulunuyor. Böyle çalışma şeklinin elde edilmesi için hücrenin temel mantıksal yapısında değişiklikler yaparak arabellek devresi eklenmelidir. Bu konu hakkında ders kitabının ilerleyen bölümlerinde daha detaylı bahsedeceğiz.

#### 4.5.3. D FLİP-FLOP MANTIĞININ DEĞİŞMESİ

D flip-flopun başka mantıklı flip-flopların uygulamasında giderek fazla kullanıldığını önceden söylemiştik. Buna göre, flip-flop bir yandan ardışık ağların yapılmasında standart bileşen haline geliyor. Bu yüzden, yukarıda incelediğimiz flip-flop çeşitlerinin D flip-flopun uygulanmasıyla bazı çözümlerin gösterilmesinden yarar vardır. SR flip-flopların en önemli dezavantajlarından biri, tabii ki onların  $R=1$  ve  $S=1$  giriş kombinasyonu için belirlenmemiş durumda olmalarıdır. Ancak, bir hakim girişli SR flip-flopların yapılması mümkün olduğunu söylemiştir. Böylece, örneğin R girişinin hakim olduğu SR flip flop, D flip flopun uygulanmasıyla yapılabilmesi için, Şek. 4-26 a)'da gösterilmiş olan bağlanma şeklinin kullanılması gerekiyor. Böyle yapımda,  $R=1$  ve  $S=1$  giriş kombinasyonunun meydana gelmesi çıkışın sıfırlanmasına yol açıyor. Böylece  $R=1$  ve  $S=1$  uyarısıyla belirlenmemiş durum ortadan kaldırılıyor, çünkü bu durumda flip flop sıfırlandırılıyor ( $Q^+=0$ ).

Şek. 4-37 b)'de D flip floptan, JK flip flopun gerçekleştirildiği bağlanma şekli gösterilmiştir.

T flip-flopu hem palsı olarak da uygulanabilir, ancak sıkça palslı olmayan şekilde de uygulanabilir. Şek. 4-37 c)'de T flip-flopun palslı olmayan yapılı, Şek. 4-37 ç)'de ise palslı T flip-flopun yapılı gösterilmiştir. Her iki uygulamada D flip-flop kullanılmıştır.



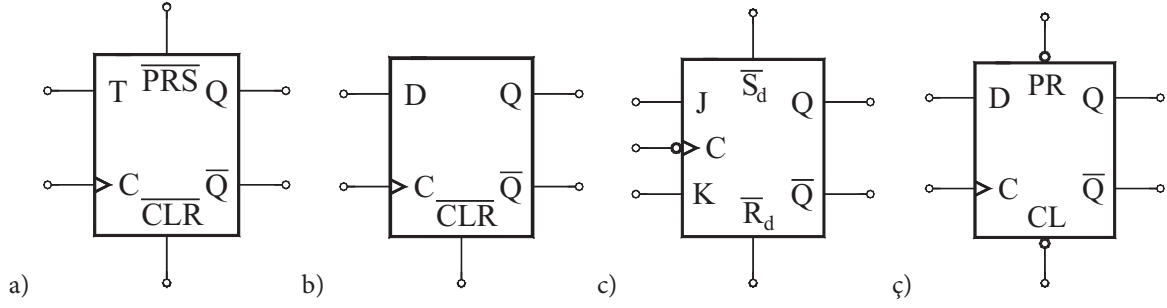
Şek. 4-37. Farklı flip-flop çeşitlerinin D flip-flopun uygulanmasıyla yapılı

## 4.6. TÜMLEŞİK FLİP-FLOPLAR

Yukarıda verilen örneklerden görülebildiği gibi, bazı flip-flopun tümleşik mantıksal devreler yardımıyla yapılmalarına rağmen oldukça karmaşık yapılandırmaya sahiptir. Teknolojinin gelişimi flip flopların çok karmaşık yapılarının da tek monolitik tümleşik devre şeklinde gerçekleştirilmelerini sağlamıştır. Bu flip-flopun daha iyi özellikleri vardır, bileşik sistemlerde uygulamaları ise flip flopu ayrı işlevsel elemanları arasındaki tüm bağlantıların tümleşik devrenin içinde kurulmuş olduğu göz önüne alarak çok daha basittir. Tasarımcı, giriş seviyelerin kombinasyonlarının fonksiyonu olarak sadece çıkış durumların değişikliğini dikkate alarak onları tek işlevsel bütün olarak nitelendiriyor. Bu flip-flopun diyagramlarda şimdiye kadar gösterildiği gibi aynı şekilde, sağ tarafta flip-flopun çıkışları, sol, üst ve alt tarafta flip-flopun girişleri işaretlenmiş kare şeklinde gösteriliyor. Sol tarafta genelde senkron girişler ve pals işaretleniyor, yukarıda ve aşağıda ise doğrudan (asenkron) girişler işaretleniyor.

Şek. 4-38 a), b), c) ve ç)'de sıkça kullanılan bazı tümleşik flip-flopun sembolleri gösterilmiştir. Pratikte asenkron girişlerin küçük çemberle ("O") işaretlemelere de rastlanabilir. Buna göre, flip-flopun durumuna, bununla beraber hem çıkışlarına efektif etki, alçak seviyenin (0) getirilmesiyle gerçekleştirilebilir.

C pals-sinyalin girişindeki çember, flip-flop durumunun pals sinyalinde arka kenarın meydana gelmesiyle değişeceğini belirtiyor. Çember yoksa, değişiklik ön kenarın meydana gelmesiyle yaşanacak. Pals girişinde üçgen ( $\Delta$ ) işareti bulunuyorsa, çember olmadığı durumda palsın ön kenarıyla palstanmış (yönetilen) flip-flopun söz konusu olduğu belirtiliyor, çember işareti de varsa flip flopu palsın arka kenarıyla yönetildiğini gösteriyor.

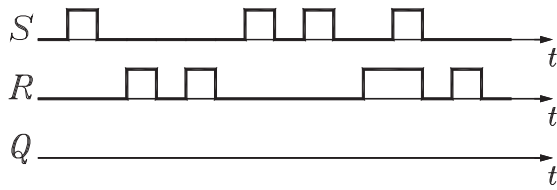


Şek.4-38. Tümeleşik flip-flopların sembolleri

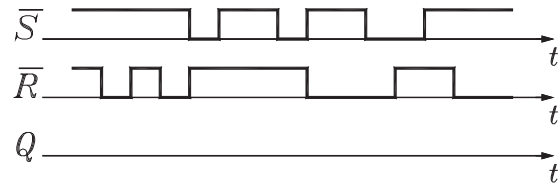
Tümeleşik teknikte yapılan flip-floplarda, iki asenkron girişi veya iki birbirine tümeleşik çıkışın olması her zaman şart değildir, sadece birer tanesinin olması yeterlidir. Tümeleşik devreler teknolojisi, içinde birkaç bağımsız flip-flopun içerildiği tümeleşik devrelerin üretilmesini sağlıyor. Böylece, örneğin 74xx ve 40xx ile işaretlenen TTL ve CMOS teknolojilerde bir bileşende tümeleşik ikili ve dördü flip-flopların farklı yapılarına rastlanabilir.

## TEKRARLAMA SORULAR VE ÖDEVLER

- 4-1. Flip-flop nedir?
- 4-2. Flip-flopun kaç çıkışı vardır, isimleri nedir ve nasıl işaretleniyor?
- 4-3. Flip-flop ne zaman ayarlanmış, ne zaman sıfırlanmıştır?
- 4-4. Flip-flopun genel olarak..... girişi var. Onlardan bazıları.....'dır, bazıları ise .....'dır.
- 4-5. Veri girişlerine ne getiriliyor?
- 4-6. Kontrol girişleri ne için kullanılıyor?
- 4-7. Asenkron flip-floplarda çıkışların durumu neye bağlıdır?
- 4-8. Senkron flip-floplarda çıkışların durumu neye bağlıdır?
- 4-9. Pals sinyali nedir?
- 4-10. Aktif seviye, yani palsın aktif kenarı meydana gelince ne oluyor?
- 4-11. Gecikme zamanı nedir?
- 4-12. Flip-flopun çalışma şeklini açıkla!
- 4-13. Flip-flop çıkışlarının durumu neye bağlıdır? Neden temel bellek bileşeni olarak adlandırılır?
- 4-14. SR flip-flopun (OYA türünden flip-flopun) mantıksal yapısını ve mantıksal sembolünü çiz
- 4-15. SR flip-flopun geçiş ve çıkış tablosunu ve eksitasyon tablosunu çiz ve karakteristik denklemini yaz.
- 4-16. SR flip-flop durumu ne zaman değiştirmiyor? 1'e ayarlanması için ne yapılmalıdır, sıfırlanması için ise ne yapılmalıdır?
- 4-17. SR flip-flopun temel dezavantajı nedir ve bu dezavantaj analitik olarak nasıl ifade ediliyor?
- 4-18. Aşağıdaki şekilde, SR flip-flopta S ve R giriş sinyallerinin zamansal diyagramları gösterilmiştir. Başlangıç durumu (a)  $Q=0$ ; (b)  $Q=1$  ise Q çıkışının zamansal diyagramını çiz.



Soru 4-18 şekli



Soru 4-21 şekli

4-19.  $\overline{S}\overline{R}$  flip-flopun (OVE türünden flip-flopun) mantıksal yapısını, mantıksal sembolünü ve geçiş tablosunu çiz.

4-20.  $\overline{S}\overline{R}$  flip-flop için geçersiz (yasak) giriş kombinasyonu var mı, varsa hangisidir?

4-21. Verilen şekilde  $\overline{S}\overline{R}$  flip-flopta  $\overline{S}$  ve  $\overline{R}$  giriş sinyallerinin zamansal diyagramları gösterilmiştir. Çıkışın başlangıç durumu (a)  $Q=0$ ; (b)  $Q=1$  ise,  $Q$  çıkışının zamansal diyagramını çiz.

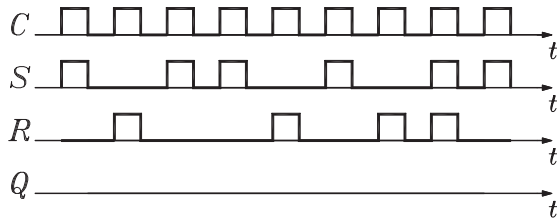
4-22. Palslı SR flip-flopun mantıksal yapısını ve sembolik işaretini çiz!

4-23. Palslı SR flip-flopun geçiş ve çıkış tablosunu ve eksitasyon tablosunu çiz ve karakteristik denklemini yaz.

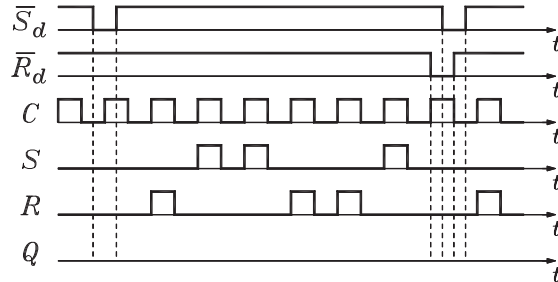
4-24. Palslı SR flip-flopta S ve R girişlerin getirilen sinyallerin ne zaman durumu üzerine efektif etkileri var?

4-25. Palslı SR flip-flopun doğru çalışması için S ve R girişlerin getirilen sinyallerin seviyeleri nasıl olmalıdır? Aksi halde ne olacak?

4-26. Verilen şekilde palslı SR flip-flopta S ve R giriş sinyallerin ve C palsının zamansal diyagramları gösterilmiştir.  $Q$  çıkışı başlangıçta alçak seviyede bulunuyorsa,  $Q$  çıkışının zamansal diyagramını çiz.



Soru 4-26 şekli



Soru 4-29 şekli

4-27. C palsı üzerine hakimiyeti olan ve alçak seviyede aktif olan  $\overline{S}_d$  ve  $\overline{R}_d$  asenkron (doğrudan) girişli palslı SR flip-flopun mantıksal yapısını ve sembolik işaretini çiz.

4-28. Palslı flip-flopta doğrudan girişlerin rolü nedir? Flip-flopun doğru çalışması için doğrudan girişler arasında yerine getirilmesi gereken koşul nedir?

4-29. Verilen şekilde palslı SR flip-flopun  $\overline{S}_d$  ve  $\overline{R}_d$  asenkron girişlerinde, C palsında ve S ve R veri girişlerinde getirilen sinyallerin zamansal diyagramları gösterilmiştir.  $Q$  çıkışı başta alçak seviyede bulunuyorsa, onun zamansal diyagramını çiz.

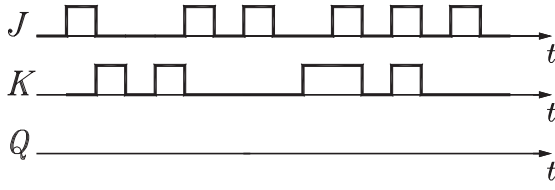
4-30. Pals seviyesinde tepki gösteren ve pals kenarının meydana gelmesine tepki gösteren flip-floplar arasındaki farkı açıkla.

4-31. MS SR flip-flopun mantıksal yapısını ve mantıksal sembolünü çiz.

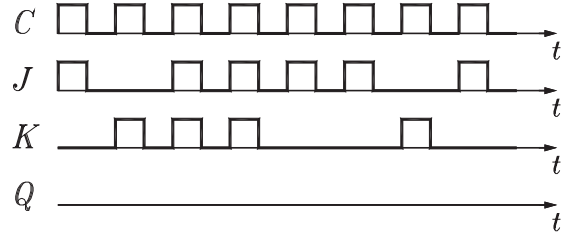
4-32. Palslı SR flip-flop (mandal) ve MS SR flip-flop arasındaki fark nedir?



- 4-33. MS SR flip flop ve pals sinyalinin kenarıyla yönetilen palsli SR flip-flop arasındaki fark nedir?
- 4-34. MS SR flip-flopun doğru çalışması için S ve R veri girişlerinde bulunan verilerin nasıl olmaları gerekiyor?
- 4-35. (a) açlak seviyede; (b) yüksek seviyede aktif olan asenkron girişli MS SR flip-flopun mantıksal sembolünü çiz
- 4-36. (a) asenkron; (b) senkron JK flip-flopun mantıksal diyagramını ve mantıksal sembolünü çiz.
- 4-37. (a) asenkron; (b) senkron JK flip-flopun geçiş tablosunu ve uyarma tablosunu çiz ve karakteristik denklemini yaz.
- 4-38. SR ve JK flip-flopların davranışları arasındaki fark nedir?
- 4-39. Verilen şekilde JK flip-flopta J ve K giriş sinyallerinin zamansal diyagramları gösterilmiştir. Q çıkışı başta alçak seviyede bulunuyorsa, Q çıkışının zamansal diyagramını çiz.

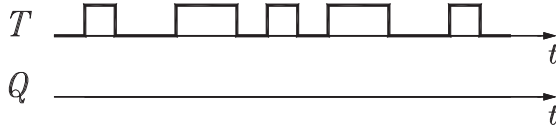


Soru 4-39 şekli

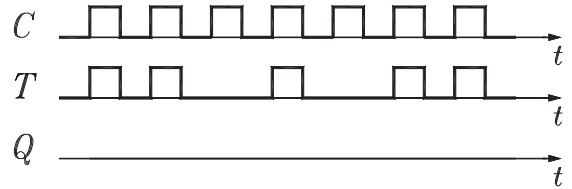


Soru 4-40 şekli

- 4-40. Verilen şekilde palsli JK flip-flopta J ve K giriş sinyallerin ve C palsının zamansal diyagramları gösterilmiştir. Q çıkışı başta alçak seviyede bulunuyorsa, Q çıkışının zamansal diyagramını çiz
- 4-41. (a) asenkron; (b) senkron T flip-flopun mantıksal diyagramını ve mantıksal sembolünü çiz.
- 4-42. (a) asenkron; (b) senkron T flip-flopun geçiş ve çıkış tablosunu ve uyarma tablosunu çiz ve karakteristik denklemini yaz!
- 4-43. Verilen şekilde T flip-flopta T girişine getirilen sinyalin zamansal diyagramı gösterilmiştir. Q çıkışının başlangıç durumu  $Q=0$  ise, onun zamansal diyagramını çiz.

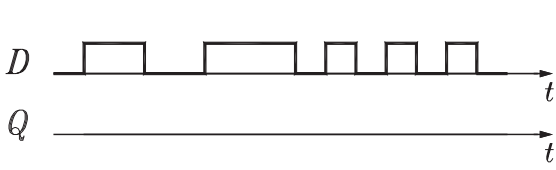


Soru 4-43 şekli

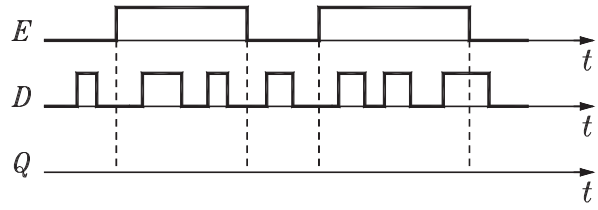


Soru 4-44 şekli

- 4-44. Verilen şekilde T flip-flopunda T veri girişinde ve C palsında getirilen sinyallerin zamansal diyagramları gösterilmiştir. Q çıkışının başlangıç durumu  $Q=0$  ise, çıkışın zamansal diyagramını çiz.
- 4-45. (a) asenkron; (b) senkron D flip-flopun mantıksal diyagramını ve mantıksal sembolünü çiz.
- 4-46. (a) asenkron; (b) senkron D flip-flopun geçiş ve çıkış tablosunu ve uyarma tablosunu çiz ve karakteristik denklemini yaz.
- 4-47. Verilen şekilde D flip-flopuna D veri girişine getirilen sinyalin zamansal diyagramı gösterilmiştir. Q çıkışının başlangıç durumu  $Q=0$  ise, Q çıkışının zamansal diyagramını çiz.



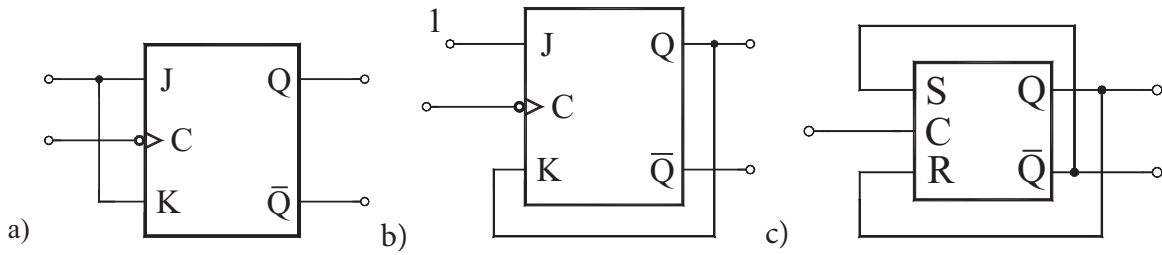
Soru 4-47 şekli



Soru 4-49 şekli

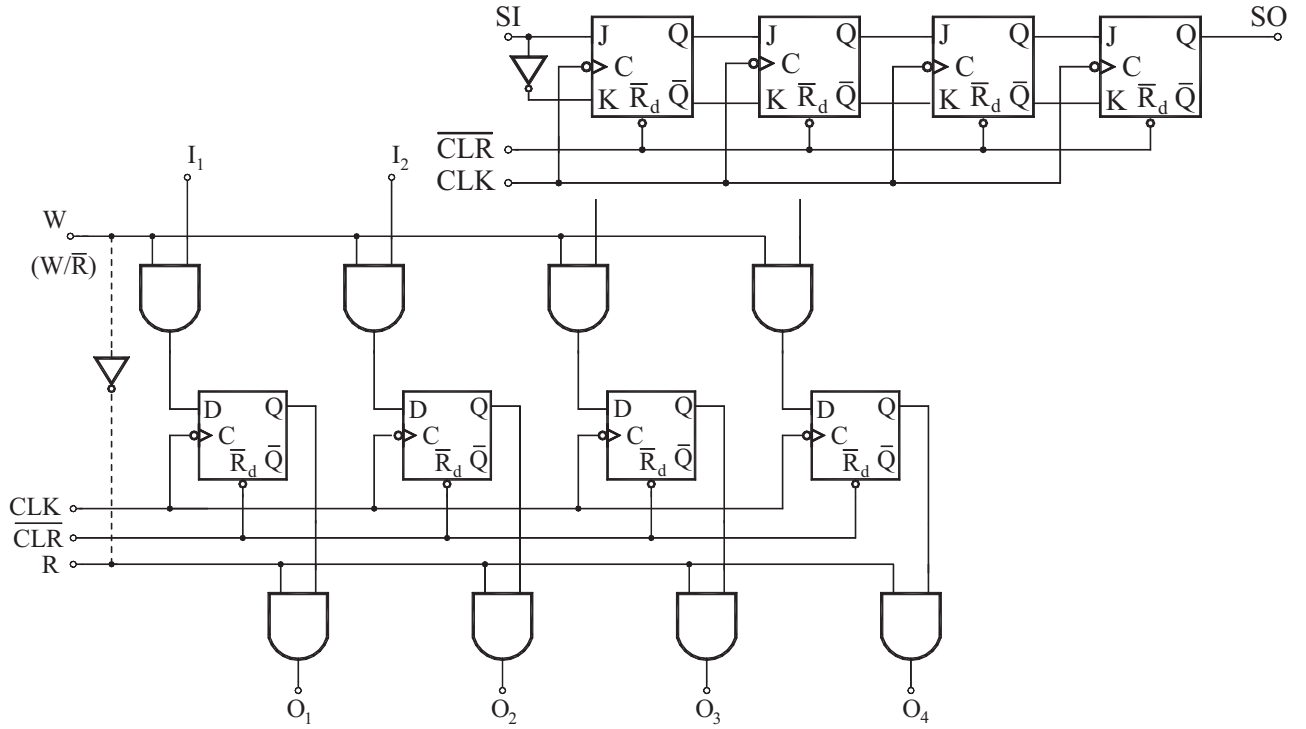
4-48. D flip-flop ve D mandalı (kilitleme devresi) arasındaki farkı açıkla.

4-49. D mandalında (kilitleme devresinde) D veri girişinde ve E kontrol girişinde getirilen sinyallerin zamansal şekilleri gösterilmiştir. Q çıkışı başta alçak seviyede bulunuyorsa, Q çıkışının dalgasal şeklini çiz. Aynı uyarma sinyallerinin, palsın a) ön kenarı; b) arka kenarı ile yönetilen D flip-flopa getirildiğini tahmin ederek Q çıkışının dalgasal şeklini çiz.



Soru 4-50 şekilleri

4-50. Verilen şekillerde gösterilen mantıksal diyagramlarla hangi flip-flopların uygulandıklarını belirle.



# 5. YAZMAÇLAR

Bu konusal birimini öğrendikten sonra:

- ⊕ Yazmaçların mantıksal yapısını anlayacaksınız;
- ⊕ Şu standart yazmaçların çalışma prensibini ve uygulamasını anlayacaksınız ve açıklayabileceksiniz;
  - ⊕ sabit yazmaç,
  - ⊕ ötelemeli yazmaç,
  - ⊕ kombine (karışık) girişli yazmaç,
  - ⊕ kombine (karışık) çıkışlı yazmaç,
  - ⊕ evrensel (genel) yazmaç;
- ⊕ Yazmaç çeşitlerini fonksiyonları ve kullanımına göre ayırt edebileceksiniz;



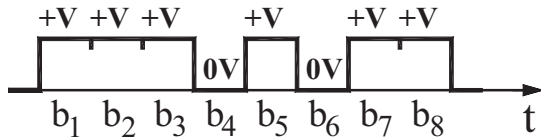
## 5.1. GİRİŞ VE TEMEL TERİMLER VE KAVRAMLAR

Yazmaçlar karmaşık ardışık ağlardır ve dijital cihazların yapımında sıkça kullanılırlar. **Yazmaç** sözcük (kelime) olarak adlandırılan sınırlı uzunlukta ikili verinin depolanması için kullanılan kurgudur. Sözcükten her bitin ezberlenmesi için iki kararlı elemanın, yani flip-flopun gerektiğinden dolayı, n-bit uzunluğunda sözcük için, n flip-flopun gerektiği sonucuna varabiliriz.

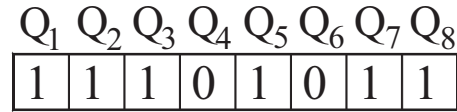
Yazmaçlar geçici olarak ezberlemek için birikeç (akü) olarak kullanılırlar. Onlar giriş verilerini, veri işletim sürecinde elde edilen ara sonuçları ya da sonuçları kabul ediyorlar, ardından farklı hızla çalışan dijital cihaz parçaları arasında bağlantıların kurulması gereken yerlerde bellek olarak kullanılabilirler. Ayrıca, aritmetik ve mantıksal işlemlerin gerçekleşmesi sırasında kullanılırlar: tümleme, özel çarpma ve bölme durumları, bilgi iletiminde verilerin paralel şekilde dizisel şekilde ve ters dönüşüm için eviriciler olarak kullanılabilirler vb.

Şek. 5-1'de dijital bir kurguda gerilimin dalgalı şekli gösterilmiştir. Bu şekilde, belirli 8-bitli verinin, örneğimizde 11101011 verisini tanımlayan sinyal söz konusudur. Mantıksal 1'in gerilim seviyesi yüksektir,  $V(1) = +V_{cc}$ , mantıksal sıfırın ise alçaktır  $V(0) = 0V$ . Şek.5-2'de verilen veriyi yerleştirebilen 8 flip-floplu yazmacın çok basit blok-diyagramı verilmiştir. Verinin yazmaçta depolandığından dolayı, bu veri yazmacın **içeriğini** tanımlıyor. Bizim örneğimizde yazmacın içeriği 11101011 olacak.

Verinin yazmaca girmesi için **yükleme**, **girilme** veya **yerleşme** terimleri (İng. *load*) kullanılıyor, daha seyrek olarak yazmaçta **yazmak** (İng. *write*) terimi de kullanılıyor. Verinin yazmaçtan çıkışı için içeriği **okumak** (İng. *read*) terimi kullanılıyor.



Şek. 5-1. Sekiz bitli veri



Şek. 5-2. 8-bitli yazmacın blok diyagramı

Yazmaçta yazdırma sırasında, önceki (eski) içerik kayboluyor, yazmacın olduğu flip-flopun girişi olduğu veri giriş hatlarında getirilen veri ise ezberleniyor. Yazmadan farklı olarak, yazmacın içeriği genelde birçok kez okunabilir, çünkü okuma sırasında yazmaç içinde yazılmış veri değişmiyor. Böyle okuma yıkıcı (bozucu) değildir. Ancak, yazılan verinin yok olduğu yıkıcı okuma da vardır. Bu yüzden okunan verinin yenilenmesi gerekiyor, yani yazmaçta yeniden yüklenmesi gerekiyor. Okuma sırasında, yazmacın içeriği, aslında flip-flopun çıkışları olan çıkış veri hatlarında meydana geliyor. Yazmacın durumu, belirlenmiş zaman aralığında her ayrı flip-flopun çıkışındaki değerle tanımlanıyor. Bu yüzden çalışmaya başlamadan önce, yazmaçta başlangıç durumu belirleniyor, öyle ki tüm bellek elemanları sıfır durumuna getiriliyor, yani yazmaçta hep sıfırlar yazılıyor. Bu işleme yazmacın **silinmesi** ya da **temizlenmesi** (İng. *clear*) denir. Bazen yazmaç, başlangıç durumu sıfırdan farklı olarak şekilde tasarlanıyor, yani başlangıç durumunda bazı flip-floplarda birler yazılıyor. Bu işleme yazmacın başlangıç **ayarlanması** (İng. *preset*) denir.

Silme ile ayarlama ve yükleme veya okuma gibi yazmaçta girilen içeriğin üzerine yapılan başka işlemler özel kontrol hatları aracılığıyla yönetiliyor. Yazmaç senkronize çalıştığı zaman, özel giriş olarak palslama girişi de bulunuyor. Bu arada yazmacın içeriğinde giren tüm flip-floplar aynı pals sinyali ile palslıdır. Tabii ki, kontrol hatları dışında, yazmacın giriş ve çıkış veri (bilgi) hatları da var. Bu hatların yardımıyla yazmaç yükleniyor ya da yazmacın içeriği (yazmaçta girilen veri) okunuyor.

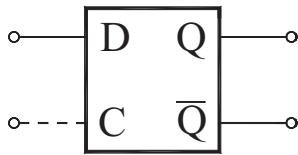
Yazmacın içeriğini okuma şeklini, ya da yazmaçta yeni verinin yazılma şeklini göz önüne alarak, içeriğin iki okunma veya yazdırma şekli vardır:

1. **Paralel (alansal)**. Tüm bitler aynı anda, bir zamansal aralığında yazılıyor veya okunuyor ve
2. **Dizisel (zamansal)**. Bitler birer birer (bit bit olarak), her bit için birer zaman aralığında birinciden son bite kadar yazılıyor veya okunuyor.

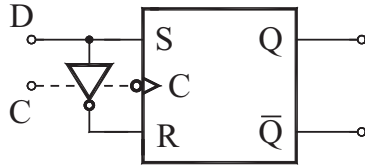
Verilerin açıkladığımız giriş ve çıkış şekillerin birleşmesiyle birkaç farklı yazmaç çeşidi elde ediliyor:

1. Paralel girişli ve paralel çıkışlı yazmaçlar;
2. Dizisel girişli ve dizisel çıkışlı yazmaçlar;
3. Dizisel girişli ve paralel çıkışlı yazmaçlar;
4. Paralel girişli ve dizisel çıkışlı yazmaçlar ve
5. Karışık yazma ve/veya okuma olanaklı yazmaçlar.

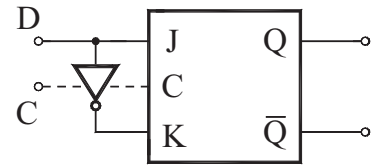
Birinci yazmaç türü **sabittir** (durağandır), çünkü yazmaçta yazılan veri, yazmacın değişmeyen içeriği olarak başlangıç şeklinde kalıyor. Diğer üç yazmaç çeşidinde yazmaç içeriği, yani yazılan veri sürekli birer bit sola ya da sağa kayıyor ve bu yüzden bu yazmaçlara **ötelemeli** veya **dinamik** (İng.*shift*) yazmaçlar denir. Pratikte, sıkça yazmaçların girişleri ve çıkışları karışık yapıyor, öyle ki dizisel veya paralel giriş ve/veya dizisel veya paralel çıkışı olabilir.



Şek. 5-3. D flip-flop



a) SR flip-flop ile yapılmış



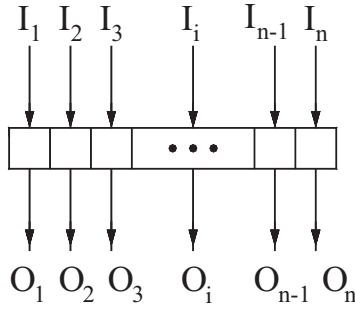
b) JK flip-flop ile yapılmış

Şek. 5-4. D flip-flop

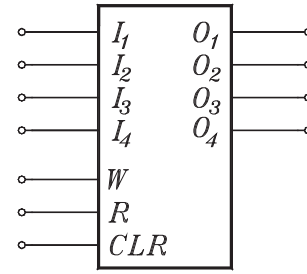
D flip-flopun çalışma şekline göre bir bit uzunluğunda veri saklayabilen temel ardışık bileşen olduğuna göre, yazmaçların yapılması için Şek. 5-3'e göre, söz edilen verinin bitlerle ifade edilmiş uzunluğuna uygun olarak, büyük sayıda palslı D flip-flop kullanılıyor. Kullanılan flip flopların genelde asenkron girişleri de vardır. Bu girişlerle yazmaçların başlangıç durumları ayarlanıyor veya yazmacın olanaklarını genişleterek karışık yazmaç çeşitlerin elde edilmesi için kullanılıyor. Yazmaçların pratik uygulamalarında, temel eleman olarak SR veya JK flip-floplar kullanan mantıksal yapılar da görebiliriz. Ancak bu durum o kadar şaşırtıcı değildir, çünkü böyle flip-floplar her zaman Şek. 5-4 a) veya b)'ye göre D-flip-flopları uygulayan yapımlarda bağlı olacaktır.

### 5.1. SABİT YAZMAÇ

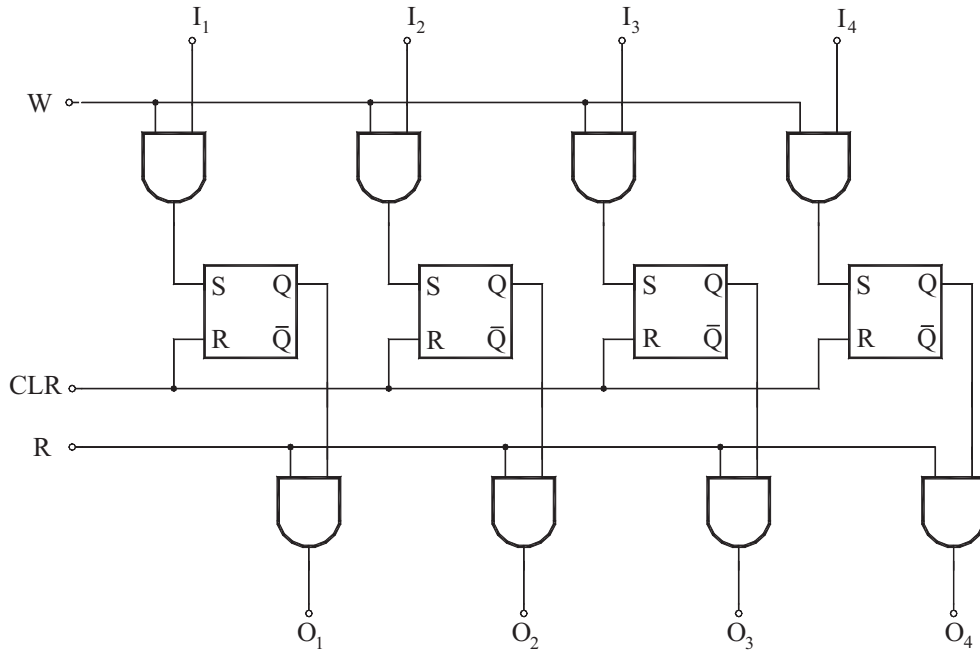
**Sabit (durağan) yazmaçın** Şek. 5-5 a)'daki basit blok-diyagramda gösterilmiş olduğu gibi paralel çıkışı ve paralel girişi vardır. Şek. 5-5 b)'de  $FF_i$  ( $i=1,2,3,4$ ) olarak işaretlenmiş dört SR flip-flopun kullanılmasıyla dört bitli sabit yazmaçın bir uygulaması gösterilmiştir. Gösterilen yazmaçın sekiz veri (bilişim) hattı var. Onlardan dördü giriş hattıdır:  $I_1, I_2, I_3$  ve  $I_4$ , kalan dördü ise çıkış hatlarıdır:  $O_1, O_2, O_3$  ve  $O_4$ . Onların dışında üç kontrol (yönetim) hattı da var: yazmaçın silinmesi için CLR, yeni içeriğin yazılması için W ve okuma hattı R. Bu tür yazmaçın mantıksal sembol Şek.5-5 c)'de gösterilmiştir.



a) basit blok-diyagram



b) Sembolik işaret



c) Mantıksal diyagram

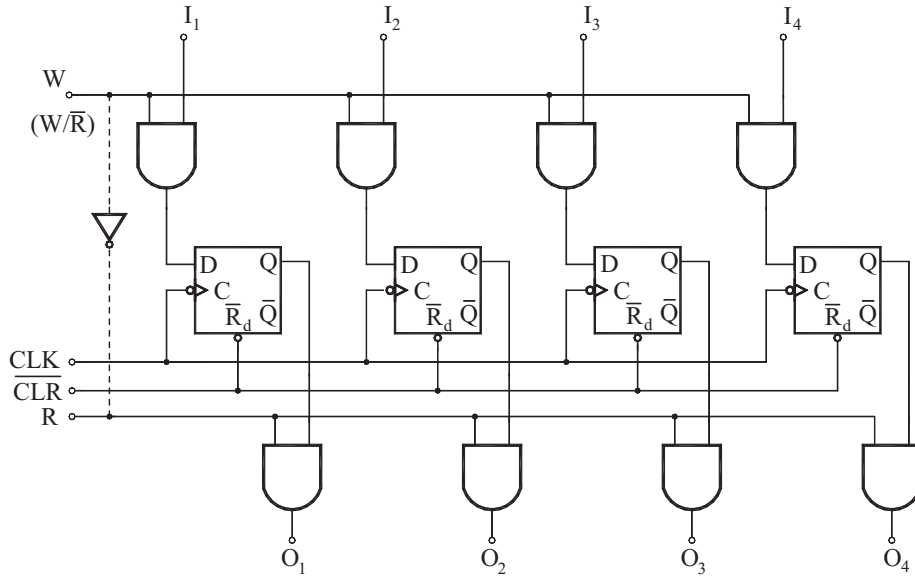
Şek. 5-5. SR flip floplu dört bitli sabit yazmaç

Yazmaç başlangıç durumuna, CLR silme girişine 1 getirilerek, yani yüksek gerilim seviyesi getirilerek ( $CLR=1$ ) ayarlanıyor. Böylece flip-floplar silinecek çünkü onların tüm sıfırlama girişleri aktiftir  $R_i=1$  ( $R_1=R_2=R_3=R_4=1$ ), öyle ki başlangıç durumu şöyle olacak:  $Q_1=0, Q_2=0, Q_3=0$  ve  $Q_4=0$ .

Yazmaçta yazdırma, verinin tüm bitleri  $I_1, I_2, I_3$  ve  $I_4$  girişlerine getirilince ve ardından  $W$  hattına yüksek seviye getirilerek ( $W=1$ ) gerçekleştirilebilir. Böyle durumda tüm VE devreleri aynı anda "açılıyor" ve her bit uygun flip-flopa yazılıyor:  $S_i=I_i$ . Aslında, durumunu sadece girişlerine 1 olan flip-floplar değiştiriyor. Girişleri 0 olan flip-floplar önceki durumunu koruyor. Bu yüzden, yazmacın her doldurulması öncesinde  $CRL$  silme girişine yüksek seviye getirilerek yazmacın silinmesi gerekiyor. Yazılan veri yazmaçta yazmaç içeriği olarak belleklenmiş kalıyor. Bu veri  $I_i$  girişi bilişim hatlarında girilmesi gereken yeni verini meydana gelmesine kadar yazmaçta kalıyor.

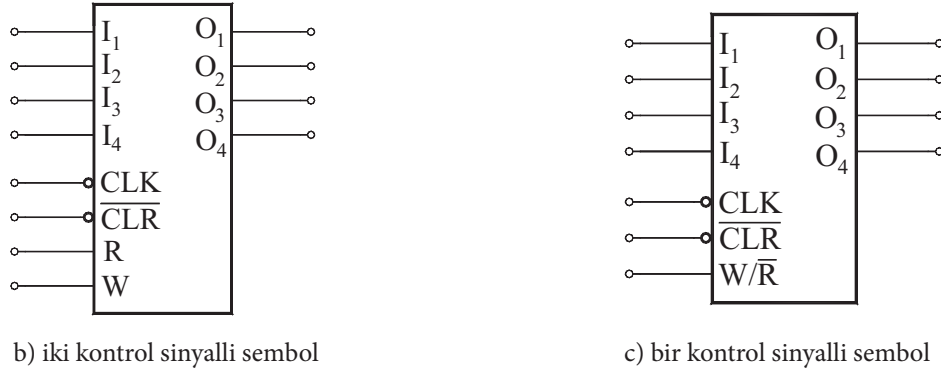
Yazmaç içeriğinin okunması için,  $R$  girişine 1'in getirilmesi gerekiyor, ( $R=1$ ). Böyle durumda aynı anda tüm çıkış VE devreli açılıyor:  $O_i=Q_i$ . Bununla, her flip-flopun çıkıştaki durumu,  $Q_1, Q_2, Q_3, Q_4$  yazmaç çıkışında meydana gelerek yazmaçta bulunan veri değişmeden elde ediliyor. Bu şekilde yazmaçın içeriği hiç değişmeden fazla kez okunabilir. Böyle yazmaçlarda girilen içeriğin kaymadığından, ya da hareket etmediğinden dolayı, onlara sabit (durağan) yazmaç denir. Şek.5-5 a)'da verilen yazmacın doğru çalışması için aynı zamanda  $W=1, R=1$ , veya  $W=1, CRL=1$  veya  $R=1, CRL=0$  olmaması gerektiğini unutmamalıyım, yazıldığı  $W=1, R=0, CRL=0$  ve okunduğu zaman tersi  $R=1, W=0, CLR=0$ , yazmaç içeriği silindiği zaman ise  $CRL=1, W=0, R=0$  olmalıdır.

Bilginin girilmesi, yazmacın mevcut içeriğinin silinme gereksinimi olmadan, olanağı verilecek şekilde yapılırsa, yazmaçta yeni bilginin girmesi için gereken zaman kısalanabilir. Böyle bir yazmacın mantıksal diyagramı Şek. 5-6 a)'da gösterilmiştir. Bu yazmaç D flip-floplarla uygulanmıştır, öyle ki D girişlerin her birinde getirilen verinin herhangi mantıksal seviyesi doğrudan giriliyor. Bunun dışında, palsli D flip-flopların kullanılmasından dolayı, yazmacın çalışması CLK pals dürtülerin arka kenarıyla senkronizedir. Bu yazmacın önemli özelliği, içeriğin silinmesi için kullanılan,  $R_d$  sıfırlandırma için doğrudan girişlerin olmasıdır. Bu özellik, onların alçak seviyede aktif olan,  $\overline{CRL}$  olarak işaretlenmiş ortak bir hatta bağlanmasıyla sağlanıyor ya da bu girişe 0 getirilirse ( $\overline{CRL} = 0$ ), yazmaç siliniyor.



Şek.5-6 a) Palslı D flip-foplu dört bitli sabit yazmacın mantıksal diyagramı





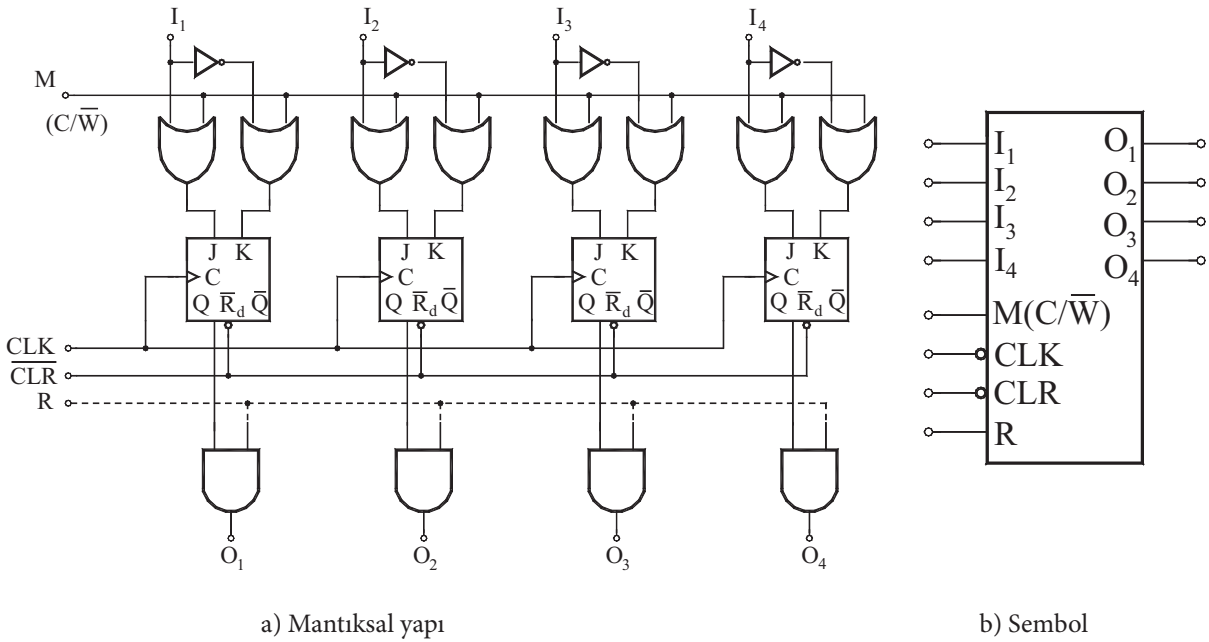
Şek. 5-6. Palsli D flip-floplu dört bitli sabit yazmaç

Yazmacın doğru çalışması için, W giriş (yükleme, yazma) ve R okuma kontrol hatlarının durumlarının karşılıklı tümleşik olma koşulu geçerli olmalıdır:  $WR=0$ , yani iki hatta aynı zamanda yüksek seviye olmamalıdır.

Yazma ve okuma süreci, Şek.5-5 a)'da kesilmiş çizgilerle gösterilmiş olduğu gibi, hem okuma hem yazma için sadece bir kontrol hattının kullanılmasıyla basitleştirilebilir. Böyle durumda R okuma hattı W yazma hattından evirici ile alınarak, W ve R hatlarının durumları arasında tümleme koşulu her zaman geçerli olacak. Bu basit giriş hattı ( $W/\overline{R}$ ) ile işaretlenebilir, çünkü bu hatta yüksek seviye getirilirse yazma olanağı etkindir, alçak seviye getirildiğinde ise okuma olanağı etkindir. Şöyle ki, eğer  $W/\overline{R}=1$  ise, o zaman yazmaçta yeni içeriğin girmesi olanağı sunuluyor, çünkü giriş VE devreli  $D_i=I_i$ , veri bitlerini geçiriyor. Böylece  $Q_i^+=D_i$  ve bu arada yazmaç çıkışlarında sadece sıfırlar elde ediliyor,  $O_i=0$ . Böyle bir durum, her çıkış VE devresinde, okuma/yazma ( $W/\overline{R}$ ) kontrol hattında bulunan 1'in tümleştirilmesinden kaynaklanan birer sıfırın olmasında elde ediliyor. Diğer taraftan ( $W/\overline{R}$ )=0 ise, aslında yazmaçta bulunan verinin okunması gerçekleşiyor. Ancak, bu durumda okuma yıkıcıdır, yani okumayla yazmacın içeriği otomatik olarak siliniyor çünkü flip-flopların tüm girişlerine sıfırlar meydana geliyor,  $D_i=0$ , buna göre  $Q_i^+=0$  olacak.

Okuma ve yazma için iki kontrol hattıyla yapılan yazmacın sembolik işareti Şek.5-6 b)'de gösterilmiştir, okuma ve yazma için sadece bir kontrol hattıyla uygulamanın sembolü ise Şek.5-6 c)'de tanımlanmıştır.

Şek.5-7 a)'da, senronize olarak çalışan, sembolik işareti ise Şek.5-7 b)'de verilmiş olan, JK flip-floplarla uygulanan başka bir sabit yazmacın mantıksal yapısı gösterilmiştir. Bu uygulamada da yeni verinin girilmesinden önce, yazmaç içeriğinin silinmesine gerek yok, çünkü D flip-flopları gibi bağlanmış JK flip-floplar kullanılıyor. Ayrıca, çalışma şeklinin (düzenin) M kontrol hattı (İng.mode) sayesinde,  $M=0$  olunca yeni veri girilebilir ya da  $M=1$  olunca yazmaçta bulunan içerik tümleşebilir. Birinci durumda,  $M=0$  olunca  $J_i=I_i$ ,  $K_i=T_i$  olacak, öyle ki sıradaki zaman aralığı için  $Q_i^+=I_i$  geçerli olacak. M hattında yüksek seviye getirilerek, yani  $M=1$  ise, o zaman  $J_i=K_i=1$  olacak, öyle ki tüm flip-flopların çıkışlarında sıradaki durum, mevcut durumunun tümleyeni olacak:  $Q_i^+=\overline{Q_i}$ . Bu şekilde gerçekleşen yazmaçta, yeni verinin girilmesinden hemen sonra ya da tümlemeden hemen, içeriği çıkış hatlarında bulunuyor. Ancak, Şek.5-7 a)'da kesilmiş çizgilerle gösterilmiş olduğu gibi bir okuma hattın (R) daha eklenmesiyle, R okuma girişinde yüksek seviye ( $R=1$ ) getirilerek okuma anı üzerine de kontrol yapılabilir, çünkü o zaman  $O_i=Q_i$  olacak. Diğer taraftan,  $R=0$  ise, o zaman yazmacın tüm çıkışları 0'dır:  $O_i=0$ .



Şek. 5-7. JK flip-floplarla palslanmış dört bitli sabit yazmaç

Bu yazmaçta da sıfırlamak için  $R_d$  asenkron girişleri kullanılıyor ve bunlarla yazmacın içeriği siliniyor. Bunu işlem, bu hatların alçak seviyede aktif olacak CLR ile işaretlenen tek hatta bağlanmasıyla gerçekleşiyor (CLR=0 koşulu geçerli ise yazmaç siliniyor).

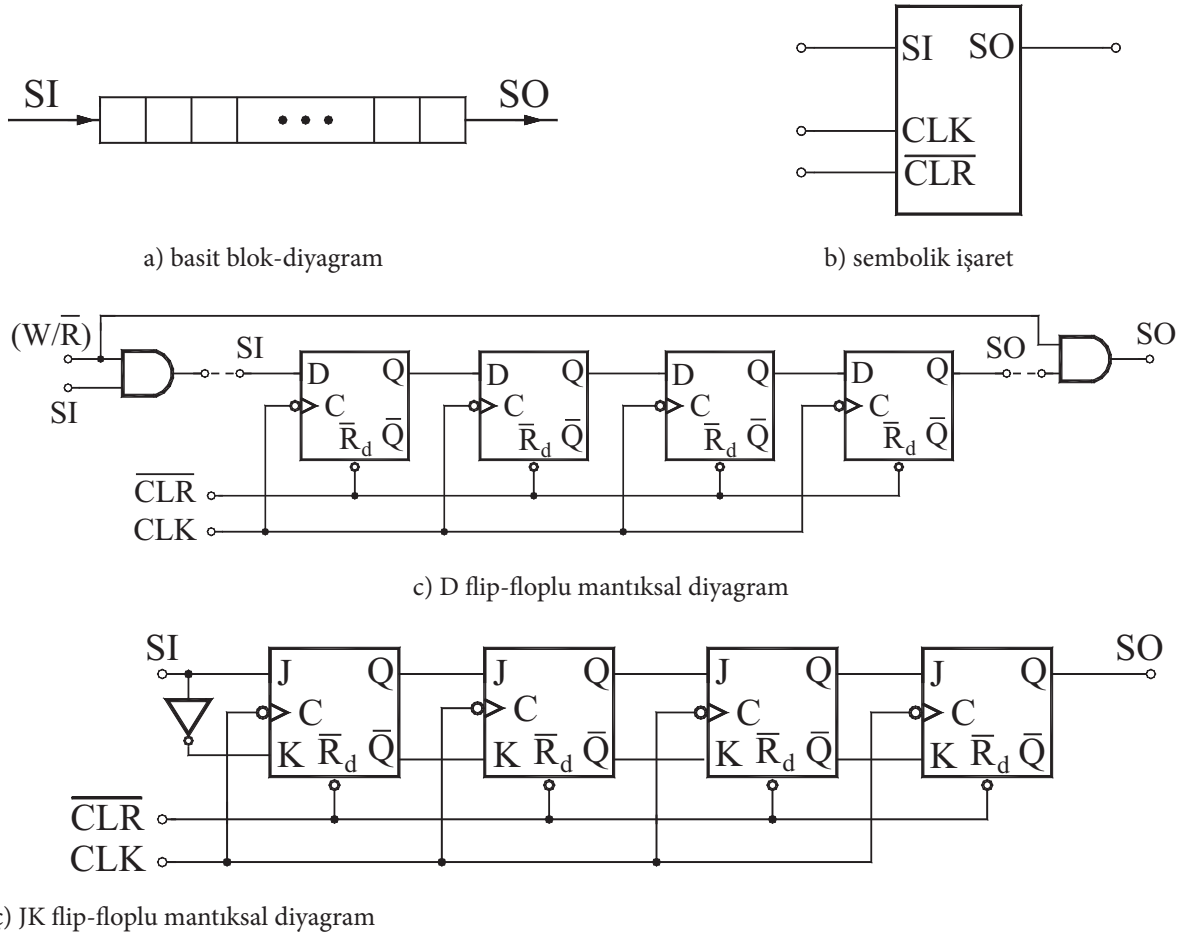
Palslı flip-floplarla uygulanan sabit yazmaçlar için, CLK palsın ön kenarına karşılık veren flip-floplar veya pals-sinyalin arka kenarına karşılık veren MS yapılı flip-floplar kullanılabilir.

## 5.2. ÖTELEMELİ YAZMAÇ

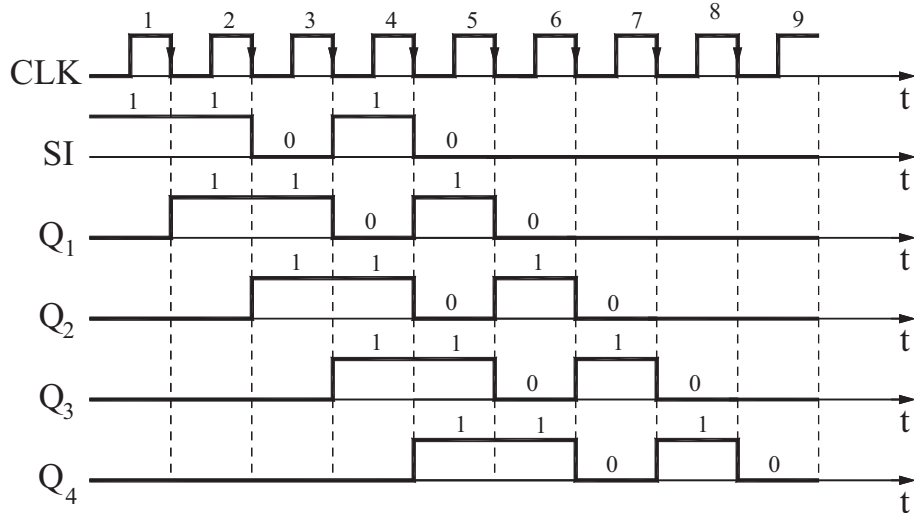
D flip-flopu için, girişindeki pals-sinyalin aktif (anahtarlamalı) kenarın (geçitin) meydana gelmesinden hemen sonra, flip-flopun Q çıkışının, aktif kenarın meydana gelmesinden önce D girişinde bulunan durumu alacağını,  $Q=D$  olacağını biliyoruz. Bu yüzden, Şek. 5-8 a)'da verilen basit blok-diyagramda gösterilmiş olduğu gibi, D flip-flopların yardımıyla, içeriğinin dizisel girişiyle ve dizisel çıkışıyla ötelemeli (İng. *shift*) yazmaç yapılabilir. Şöyle ki **ötelemeli yazmaç** prensipte birkaç D flip-floptan oluşan ardışık bağlantı (basamaklama) olduğunu söylebiliriz.

Giriş verisi birinci flip-flopun girişinde meydana geliyor, çıkışı ise pals-sinyalinin  $T_p$  periyoduna eşit olan bir zaman aralığı için gecikmeli olacak. İkinci flip-flopun çıkışında veri bir pals-aralığı daha kadar gecikmeyle meydana gelecek vs. Böylece bilginin yazmaçtan sağa doğru kademeli kayması meydana geliyor:  $D_i=Q_{i-1}$  ve  $Q_i^+=Q_i$ , ya da  $D_{i+1}=Q_i$  ve  $Q_{i+1}^+=Q_i$ . Yazmacın girişinde nasıl bit dizisinin meydana geldiği önemli olmadan, her pals dürtüyle bu grup bir flip-flop için sağa kayacak (ötelenecek). Buradan da bu yazmaçlar ötelemeli olarak adlandırılır.

Yazmaçların modern yapımlarında, uygulamaları sırasında palsın ön veya arka kenarında aktif olan palslı flip-floplar kullanılıyor ya da MS flip-flopların uygulanmasıyla gerçekleşiyorlar. Pals dürtülerin arka kenarı ile çalışan böyle bir dört bitli ötelemeli yazmaçın iki uygulaması Şek. 5-8 c) ve ç)'de gösterilmiş. D flip-floplar yardımıyla standart uygulama (c) dışında, D olarak bilinen şekilde, yani J ve K girişlerin karşılıklı tümleşmesiyle bağlanmış JK flip-floplar da kullanılmış (ç). Ötelemeli yazmaçın sembolü Şek. 5-8 (b) ile gösterilmiştir.



Şek.5-8.Dört bitli ötelemeli yazmaç



Şek. 5-9. Ötelemeli yazmaçın zamansal diyagramları

Çalışmaya başlamadan önce, CLR asenkron girişine alçak seviye (mantıksal 0) getirilerek, yazmacın önceki içeriği siliniyor ve sıfırlanıyor, başlangıç sıfır durumuna getiriliyor. Bu yazmaçın çalışması en iyi şekilde zamansal diyagramlar ya da geçiş tablosu yardımıyla gösterilebilir. Bu tabloda her ayrı pals-aralığı için bileşendeki her flip flopun durumu yazılıyor.

Bu yazmaçın çalışma şeklini daha iyi anlamak ve daha detaylı açıklamak için basit bir örnek inceleyeceğiz, o da  $t=0$  anında meydana gelen 11010 verisinin yazmaçta yazılma örneğidir. Karakteristik noktalarda zamansal diyagramlar Şek. 5-9'da gösterilmiştir. Yazmaçın başlangıç durumu için, birinci pals-sinyalin gelmesinden önce, daha doğrusu pals sinyalin iniş (arka) kenarında, tüm flip-flopların sıfırlanmış olduklarını tahmin edeceğiz, yani onların içeriği 0'dır ( $Q_1=0, \dots, Q_4=0$ ).

İncelemeye başladığımız  $t=0$  başlangıç anından, birinci flip-flop girişinde yüksek seviyeli (mantıksal 1) giriş verisinin getirildiğinden dolayı, birinci flip-flopun birinci iniş kenarında ayarlanacağı açıkça görünüyor. Tüm diğer flip-flopların girişleri alçak seviyede, yani 0-da olduklarından dolayı, onlar sıfırlanmış kalacak. Birinci flip-floptan ikinci flip-flopa bu aktif geçişin ardından dürtünün ön kenarı gelerek, 0'dan 1'e geçiş gerçekleşiyor, çünkü bu değer birinci flip-flopun çıkışıdır. Ancak bununla ikinci flip-flop ayarlanmıyor çünkü o ön kenara karşılık vermiyor, arka kenarı "bekliyor". Tüm diğer flip flopların girişleri alçak seviyede kalıyor

CLK	SI ( $D_i$ )	$Q_1Q_2Q_3Q_4$
1	1	0 0 0 0
2	1	1 0 0 0
3	0	1 1 0 0
4	1	0 1 1 0
1	0	1 0 1 1
6	0	0 1 0 1
7	0	0 0 1 0
8	0	0 0 0 1
9	0	0 0 0 0
10	0	0 0 0 0

Tab.5-1. Ötelemeli yazmaçta durumların geçiş tablosu

İkinci anahtarlamalı kenarın meydana gelmesi sırasında birinci ve ikinci flip-flopun ayarlanması gerçekleşiyor. Birinci flip-flopun ayarlanması ikinci verinin de 1 olmasından dolayı gerçekleşiyor, ikinci flip-flop ise, girişi aslında önceki aralıkta yüksek seviyeye giden birinci flip-flopun çıkışın olmasından dolayı ayarlanıyor. Üçüncü ve dördüncü flip-flopun çıkışları  $Q_3$  ve  $Q_4$  sıfır kalıyor. Palsın her sonraki aktif geçişiyle, veri yazmaçta "girişiyor". Dördüncü pals-dürtüsünden sonra yazmaçtaki birinci flip-flopun çıkışındaverinin dördüncü bitin meydana geleceği açıkça anlaşılıyor (bu flip-floptan verinin ilk üç biti "geçip" dördüncü bit meydana geliyor), dördüncü flip-flopun çıkışında, yani yazmacın çıkışında ise verinin okunabilen birinci biti meydana geliyor.

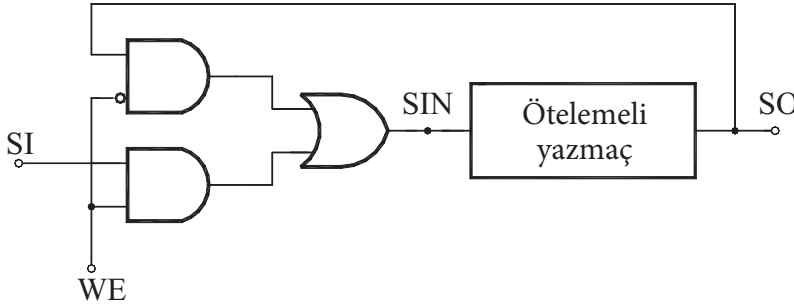
Yazmaç içeriğinden bitlerin bu şekilde okunması *FIFO* (İng. *First-In-First-Out*), ya da ilk giren ilk çıkar prensibine göre çalışma şekildir. Yazmaç içeriğinin okunması dizisel şekilde, birer birer bit yapılıyor ve giriş verisinin bit sayısına eşit pals aralığı kadar sürüyor. Tab.5-1 yazmaç durumlarının geçiş tablosu 1101 giriş kombinasyonuna ilişkindir.

Şimdiye kadar söz edilenlerden, bu yazmacın içeriği, sadece okunmasının tam vaktinde yapıldığı halde kaybolmayacak. Şöyle ki, n-inci dürtüden hemen sonra, yani n-inci iniş kenarından sonra okumaya başlanmalıdır.

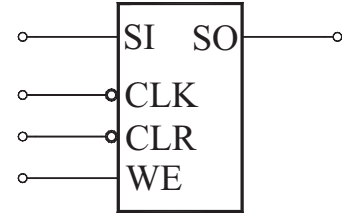
$n$  kullanılan flip-flopların sayısını belirtiyor (örneğimizde  $n=4$ ). Bu arada yeni verinin girmemesi gerekiyor, çünkü son flip-flopun dışına kayan bit kalıcı olarak kayboluyor. Bu amaçla ulaşmak için, Şek. 5-8'deki yazmaça iki VE devresi daha eklenebilir: giriş ve çıkış (şekilde kesilmiş çizgilerle gösterilmiştir). Ek kontrol hattı ( $W/\bar{R}$ ), yüksek seviyede olunca ( $W/\bar{R}=1$ ) yazma olanağı sağlanıyor ya da alçak seviyeye getirildiğinde ( $W/\bar{R}=0$ ) okuma olanağı sağlanıyor.

### 5.3. DAİRESEL YAZMAÇ

Yazmaçta saklanan bitler, son flip-flopun çıkışı birinci flip-flopun girişiyle bağlanırsa bu bitler yazmaçta korunabilir. Böyle yazmaçta veri bitleri, her pals dürtüsünde birer flip-flop için kayarak yazmaçta dönüyor. Bu şekilde bağlanan yazmaça **daireysel** ya da **dönel yazmaç** veya **iletimin sondan döndüğü ötelemeli (kaydıran) yazmaç** (İng. *end-around-carry shift register*) denir. Yazmaçta bitlerin dönmesi sadece yazmaçın "yüklendiği" anda dizisel girişin kopmasıyla ve çıkış bitlerinin birer birer girişe döndüğü geri bağlantının oluşmasıyla mümkün olduğunu söylemek gerekiyor. Bu amaçla, uygun kontrol sinyalli ek mantıksal ağların oluşması gerekecek, öyle ki birinci flip-flopun girişi yazmaçın dizisel girişine ya da yazmaç çıkışına bağlanıyor. Böyle bir daireysel yazmaç örneği Şek. 5-10'da verilmiştir, onun sembolik işareti ise Şek. 5-11'de verilmiştir.



Şek. 5-10. Dairesel ötelemeli yazmaçın mantıksal diyagramı



Şek. 5-11. Dairesel ötelemeli yazmaçın mantıksal sembolü

Yazma olanak sinyali WE (İng. *write-enable*), SIN mantıksal denklemi gerçekleştiren iki VE ve bir VEYA devresinden elde edilen ikiden-bire çoğullayıcı aracılığıyla SIN yazmaçına verilerin girişi için kontrol hattını kontrol ediyor.

$$SIN = SI \cdot WE + SO \cdot \overline{WE} \quad (11-1)$$

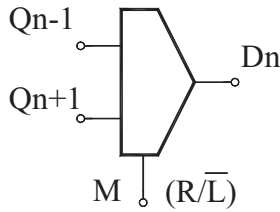
WE=1 olunca, Sİ ( $Y_{SI}=Sİ$ ) veri girişinin bağlı olduğu VE geçiti açılıyor, yazmaç çıkışının SO bağlı olduğu VE geçiti ise kapanıyor ( $Y_{SO}=0$ ). Böylece, VEYA devresi yardımıyla yazmaçta yeni verinin bitleri yazılıyor  $Y_{VEYA}=SIN=Sİ$ .

WE=0 olunca, durum terstir: girilen veriyi geçiren VE geçiti açılıyor ( $Y_{SO}=SO$ ), kapalı olan diğer VE devresiyle ise yeni verinin girmesine izin verilmiyor ( $Y_{SI}=0$ ). Böylece, VEYA devresi aracılığıyla yazmaçta yeniden belleklenmiş bilgi yüklenerek,  $Y_{VEYA}=SIN=SO$ , bu bilginin yazmaçtaki flip-floplardan dönmesi sağlanıyor.

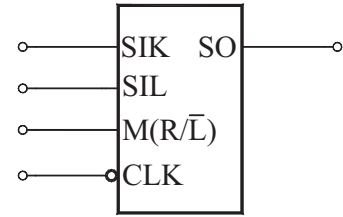
## 5.4. İKİ YÖNLÜ ÖTELEMELİ YAZMAÇ

Yukarıda incelediğimiz ötelemeli-yazmaç veriyi sadece sağa doğru birer birer bit kaydırıyor. Pratikte sıkça verileri sola kaydıran yazmaça ihtiyaç vardır. Sağa doğru ötelemeli yazmaçın çalışma prensibini tanıyarak, sola kaydırma yapması gereken yazmaçta flip-flopların bağlanma şeklini tahmin edebiliriz. Böyle yazmaçta önceki flip-flop girişi sıradaki flip-flopun çıkışı olacak:  $D_i = Q_{i+1}$ .

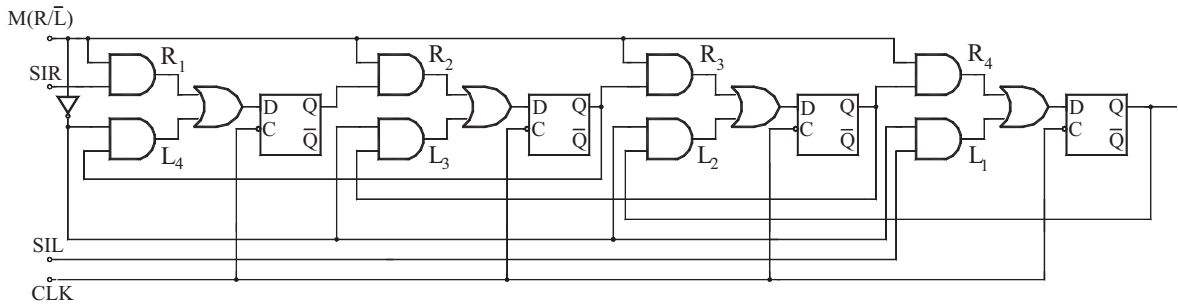
Sıradaki örnekte yazmaç içeriğini hem sola hem sağa kaydırma olanağı olan yazmaç, ya da iki yönlü ötelemeli yazmaç inceleyeceğiz. Her yazmacın girişi, sağa doğru kaydırma yapmak istersek önceki aşamanın çıkışı, ya da sola doğru kaydırma yapmak istersek sıradaki aşamanın girişi olacak:  $D_i = Q_{i-1} + Q_{i+1}$ . Bu durumda da, her flip-flop önünde birer ikiden-bire çoğullayıcının kullanılması gerekecek (Şek. 5-12), her çoğullayıcı tek hareket etme hattıyla kontrol edilecek. Yaptığımız analizden Şek. 5-13 b)'de gösterilen iki yönlü ötelemeli yazmaçın mantıksal diyagramını elde ediyor, sembolü ise Şek. 5-13 a)'da tanımlanmıştır.



Şek. 5-12. İkiyönlü ötelemeli yazmaçın bir aşamasının mantıksal yapısı



Şek. 5-13. a) İki yönlü ötelemeli yazmaçın sembolü



b) Mantıksal yapı

Şek. 5-13. İkiyönlü ötelemeli yazmaç

Kaydırma yönü, çalışma şeklini (düzenini) belirleyen M (İng. *mode*) kontrol girişine bağlıdır, çünkü her flip-flopun girişi  $D_i$ ,  $i=1,2,3,4$ , için bildiğimiz mantıksal denklem:  $D_i = (M \cdot Q_{i-1} + \bar{M} \cdot Q_{i+1})$  geçerlidir.  $M=1$  olunca, yazmaçın içeriği sağa kayıyor,  $M=0$  olunca yazmacın içeriği sola kayıyor. Bu arada, verilerin girilmesi iki girişten gerçekleşiyor: biri STR ile işaretlenmiştir ve onda sağa doğru kaydırılması gereken verinin bitleri getiriliyor, bitleri sola doğru kaydırılması gereken veri için ise STL girişi kullanılıyor.

Şöyle ki,  $M=1$  olunca sinyalin R geçitlerden iletimi mümkün olurken, L geçitinden birer girişinde 0'ın bulunmasından dolayı çıkışları 0 seviyesinde olan L geçitlerden sinyalin iletimi mümkün değildir. Böyle durumda, veri bitleri  $D_0$  birinci flip-flopun girişine getiriliyor. Bu flip-flopun çıkışı  $Q_0$  ikinci flip-flop olan  $D_1$ 'in girişine bağlıdır vs. Böylece yazmaç yukarıda açıkladığımız şekilde içeriği sağa doğru kaydırıyor.

$M=0$  olunca, ters durum vardır, çünkü tüm R geçitleri kapalıdır, çıkışları 0'dadır, tüm L geçitleri ise açıktır, flip floplar arasında ilişkiler ise ters yöndedir: Sinyal önce  $D_3$  üçüncü flip-flopun girişine getiriliyor, onun  $Q_3$  çıkışından ikinci flip-flopun girişine gidiyor, ondan sonra  $Q_2$ 'den  $D_1$ 'e vs. Buna göre, bitler bu durumda birer birer sola kayacak.

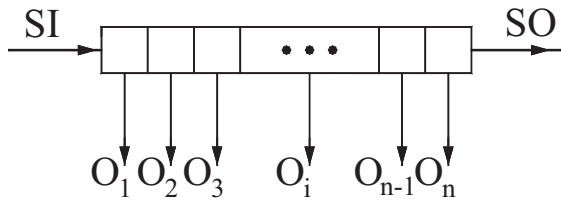
M sinyalinin, sadece pals sinyalinin seviyesi 0 ( $CLK=0$ ) olduğu durumda değiştiğini vurgulamak çok önemlidir. Aksi halde yazmaç içeriğinde istenmeyen değişiklikler meydana gelebilir.

Bu yazmaç türü veri bitlerinin *LİFO* (İng. Last-In-First-Out): son giren ilk çıkar prensibine göre okuma imkânı veriyor. Bu işlemin gerçekleşmesi veri bitlerinin sağa kaymasıyla yazmaçta girilmesi, yani  $M=1$  ayarlanması ve STR girişinin kullanımıyla, ardından ise  $M=0$  getirilerek yazmacın STL girişiyle doldurulması ile gerçekleşiyor.

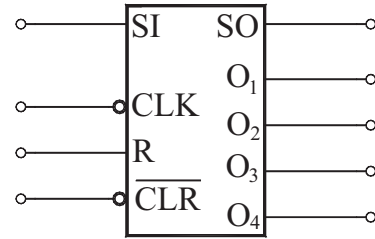
### 5.5. DİZİSEL GİRİŞLİ VE KARIŞIK ÇIKIŞLI ÖTELEMELİ YAZMAÇ

Bir dizisel girişi ve dizisel ya da paralel çıkış olanaklı yazmacın basit blok-diyagramı Şek. 5-14'te gösterilmiştir, bir uygulaması ise Şek. 5-15'te verilmiştir. Bu uygulama aslında dört bitli bir yazmacın diyagramını tanımlıyor. Sembolik işaret Şek. 5-15 a)'da verilmiştir.

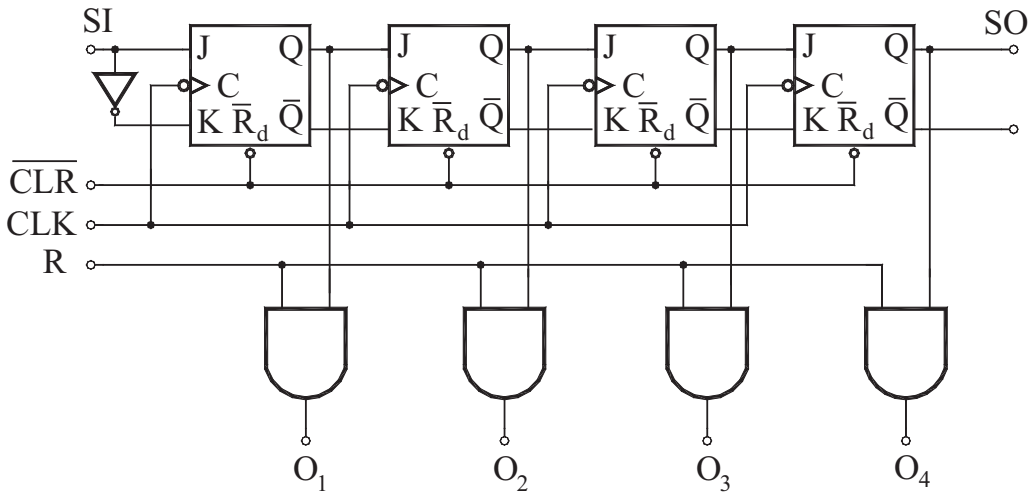
Dizisel giriş ve çıkış söz konusu olunca, yazmaçın çalışmasını önceki açıklamalardan biliyoruz: Veri dizisel şekilde birinci flip-flopun girişine geliyor ve dizisel olarak son flip-flopun çıkışından okunuyor.



Şek. 5-14. Basit blok-diyagram



Şek. 5-15. a) Sembolik işaret



b) Mantıksal diyagram

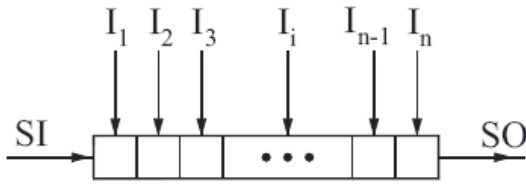
Şek. 5-15. Dizisel girişli ve karışık çıkışlı dört bitli yazmaç



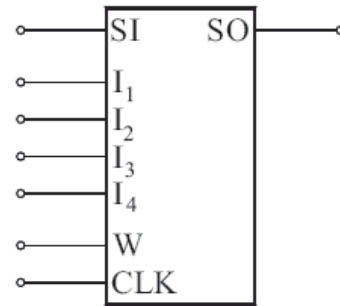
Yazmaç içeriğinin paralel okunması, R okuma hattına pozitif dürtü getirilerek ( $R=1$ ) gerçekleştiriyor. Böylece mantıksal VE devreler açılıyor ve bu şekilde her flip-flopun çıkışındaki durumun uygun çıkış hatlarında meydana geliyor:  $O_1 = Y_{11} = Q_1, \dots, O_4 = Y_{14} = Q_4$ . Böylece yazmacın içeriği (veri) paralel biçimde elde ediliyor. Açıkladıklarımızdan, bu yazmaçın dizisel-paralel kod çeviricinin (dönüştürücünün) olduğu açıkça görülüyor, çünkü veriler dizisel yazılıyor, paralel olarak okunuyor. Dizisel kodlanmış ikili bilgileri paralel ikili koda dönüştürüyor, yani zamansal şekilden alansal şekilde dönüşüm gerçekleştiriyor.

## 5.6. KARIŞIK GİRİŞLİ VE DİZİSEL ÇIKIŞLI ÖTELEMELİ YAZMAÇ

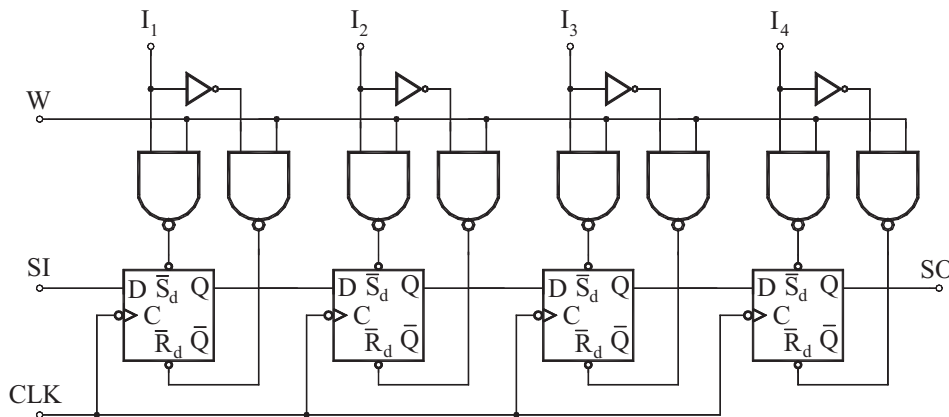
Dizisel veya paralel giriş seçenekli ve dizisel çıkışı olan bu yazmaçın basit blok-diyagramı Şek.5-16'da gösterilmiştir, bir uygulaması ise Şek. 5-17 b)'de verilmiştir. Bu uygulama yeniden dört bitli yazmaçtır, sembolik işaretini ise Şek. 5-17 a)'da görebiliriz. Verilerin paralel giriş her flip-flopta  $\overline{S_d}, \overline{R_d}$  asenkron girişler yardımıyla gerçekleşiyor. Giriş verisinin doğrudan değerleri mantıksal OVE geçitleri aracılığıyla  $\overline{S_d}$  girişini etkinleştiriyor, ikinci giriş sinyali ise W yazma hattıdır. Bu kontrol sinyali (W) ayrıca ikinci girişleri her giriş bitinin tümleyeni olan OVE geçitleri de kontrol ediyor, çıkışlar ise doğrudan Rd sıfırlama girişini uyarıyor. Bu şekilde, her flip-flopun  $\overline{S_d}, \overline{R_d}$  asenkron girişleri birbirine karşı tümleşiktir ve böylece bu dijital bileşenin doğru çalışmasını sağlayan her girişte bulunan mantıksal durumunun flip-flopta girmesi sağlanıyor. Verinin paralel yazdırılması W yazma hattının OVE geçitlerini açan yüksek seviye getirilince gerçekleşiyor.



Şek. 5-16. Basit blok-diyagram



Şek. 5-17. a) Sembolik işaret



b) Mantıksal diyagram

Şek. 5-17. Karışık girişli ve dizisel çıkışlı dört bitli yazmaç



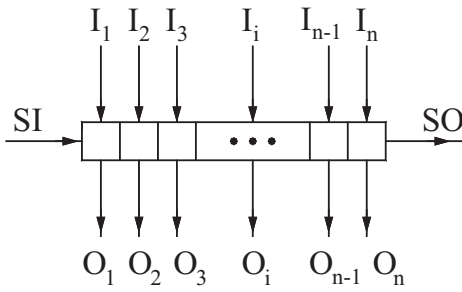
Bu şekilde, her flip-flopun ve bununla beraber yazmacın da son durumu, mevcut içeriğine değil, sadece giriş sinyallerin mantıksal seviyelerine bağlıdır. Böylece yeni içeriklerin yüklenmesi sırasında daha hızlı çalışma imkanı oluşuyor, çünkü yazmaçta bulunan verinin silinmesine gerek kalmıyor.

Bilgi birinci flip-flopun girişi üzerinden dizisel şekilde de yazılabilir, yazmaçın içeriğinin okunması ise son flip-flopun çıkışında dizisel şekilde gerçekleşiyor.

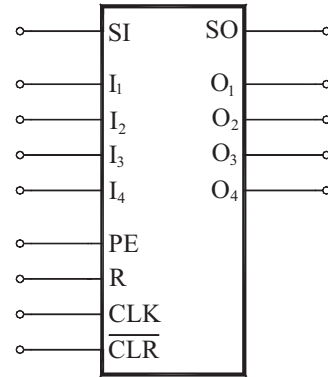
Paralel yazma ve dizisel okuma yapıldığı zaman, paralel kodlanmış ikili bilgi, aslında dizisel kodlanmış bilgiye dönüşüyor, yani alansal kodundan zamansal koda dönüşüm gerçekleşiyor. Bu çok kullanışlı bir özelliktir çünkü bu şekilde bilgiler periyodik olmayan şekilde, her bit farklı zamanda yazılabilir, ardından ise periyodik olarak, her pals-dürtüsüyle dizisel şekilde birer birer bit okunabilir.

## 5.7. EVRENSEL YAZMAÇ

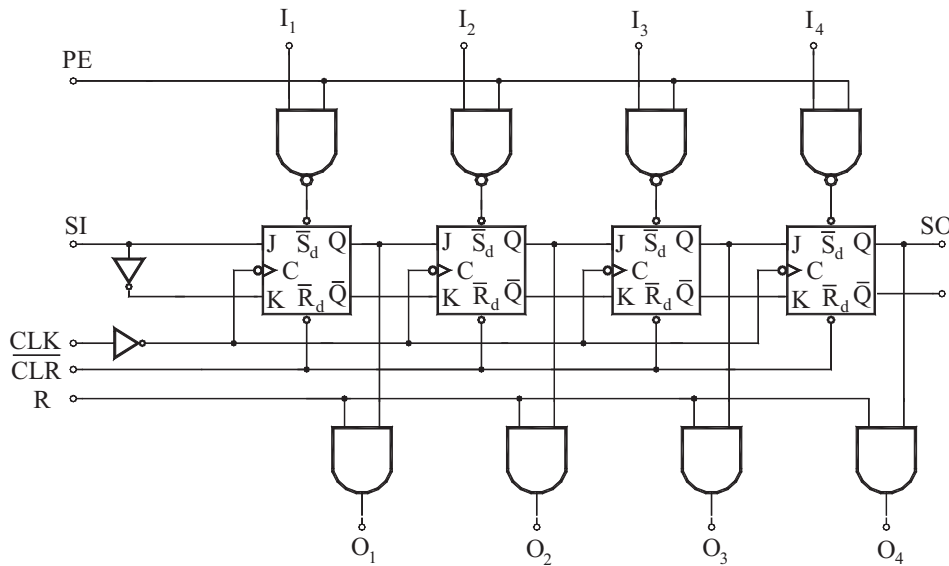
Böyle yazmaçın basit blok-diyagramı Şek. 5-18'de gösterilmiştir. Buradan bu yazmaçta verilerin her iki şekilde yazma ya da okuma olanağı olduğu görünüyor. Şek. 5-19 b)'de böyle bir dört bitli yazmaçın uygulaması verilmiştir, sembolik işareti ise Şek. 5-19'da tanıtılmıştır.



Şek. 5-18. Basit blok-diyagram



Şek. 5-19. a) Sembolik işaret



b) Mantıksal diyagram

Şek. 5-19. Dört bitli evrensel yazmaç

Silme,  $\overline{CLR}$  girişine alçak seviye getirilerek yapılıyor. Sİ (serial input) girişi üzerinden veriler birer birer bir olarak dizisel şekilde giriyor, T1,T2T3,T4 paralel girişleri aracılığıyla yazmaç herhangi başlangıç durumuna getirilebilir. Başlangıç durumuna ayarlama girişi PE (preset enable-ayarlama olanağı), yüksek seviye getirilince tam bu işlemi sağlıyor. Öncesinde, yazmacın silinmiş olması gerekiyor. Bu kural genel olarak geçerlidir: silme işlemi paralel şekilde verinin girme işlemi öncesinde gelmelidir. Çıkış bilgisi, son flip-flopun çıkışında dizisel şekilde elde edilebilir, ya da R (*read*) okuma hattına yüksek seviye getirilerek her flip-flopun durumu çıkış hatlarına meydana geliyor ve böylece paralel şekilde elde edilebilir.

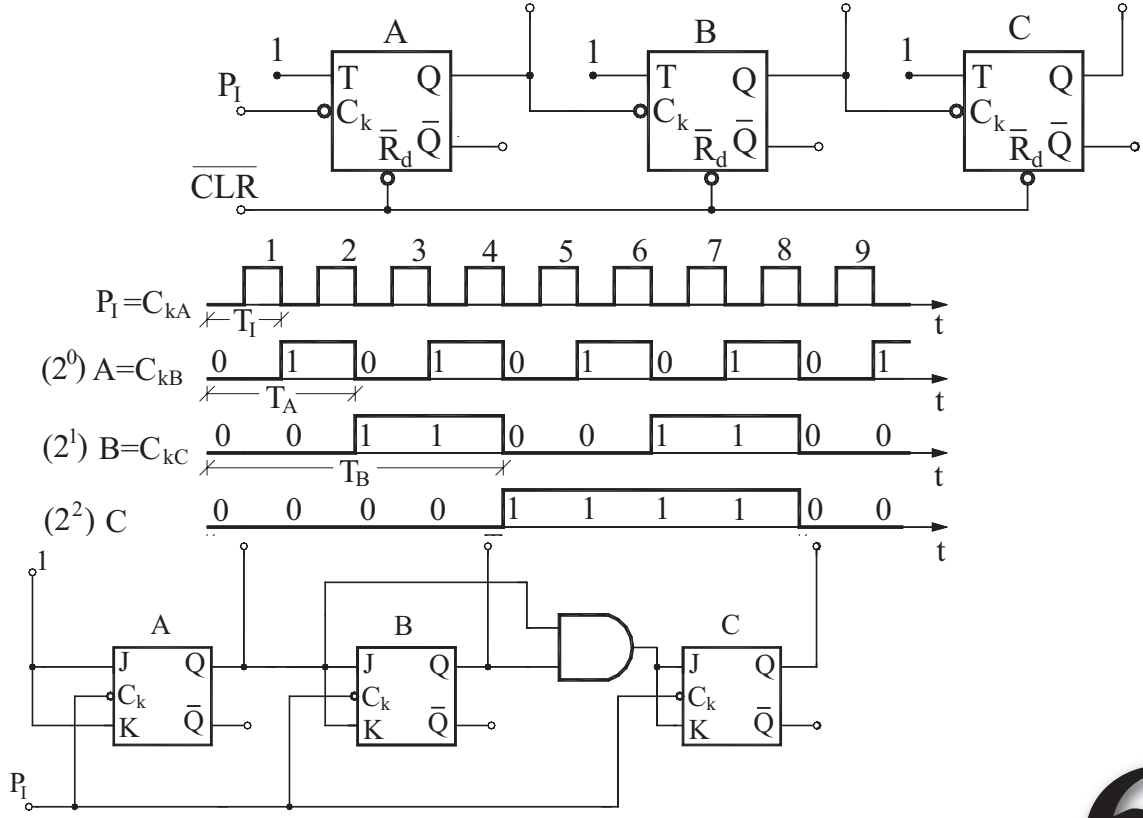
Bu yazmaç örneğinde, pals-dürtüleri önce eviriliyor, ardından tüm flip-floplarda C pals-sinyal girişlerine getiriliyor. Bununla aslında, flip-flopların MS yapılı olmalarına rağmen, palsın ön kenarında (bir peryod gecikmeyle) yazmacın çalışması sağlanıyor.

## TEKRARLAMA SORULARI VE ÖDEVLERİ

- 11-1. Yazmaç nedir? Hangi parçalardan oluşuyor?
- 11-2. Dijital hesaplama makinelerinde yazmaçların kullanımını nedir?
- 11-3. Yazmaç içeriği nedir?
- 11-4. Yükleme (girilme, yazdırma) işlemi nedir?
- 11-5. Okuma işlemiyle ne yapılıyor?
- 11-6. Yazmacın yeni içerikle yüklenmesi sırasında eski içerikle ne oluyor?
- 11-7. Yazmacın okunması sırasında içeriğin kaybolup kabolmayacağına bağlı olarak kaç okuma şekli vardır?
- 11-8. Yazmacın içeriğinin okunması sırasında her bit nerede meydana geliyor?
- 11-9. Yazmacın durumu ne ile tanımlanıyor?
- 11-10. Silme işlemiyle ne yapılıyor?
- 11-11. Ayarlama işlemiyle ne yapılıyor?
- 11-12. Her yazmaçta hangi ek hatlar vardır ve ne için kullanılıyor?
- 11-13. Yazmaç içeriğinin farklı yazma ve okuma şekillerini say!
- 11-14. Paralel şekil sırasında içeriğin bitleriyle ne oluyor?
- 11-15. Dizisel şekil sırasında içeriğin bitleriyle ne oluyor?
- 11-16. Sabit (durağan) yazmaçların özellikleri nedir?
- 11-17. Ötelemeli yazmaçların özellikleri nedir?
- 11-18. Şek. 5-5 c)'de nasıl yazmaç gösterilmiştir? Bu yazmacın blok-diyagramını ve mantıksal sembolünü çiz.
- 11-19. Şek. 5-5 c)'deki yazmacın başlangıç durumuna nasıl getirildiğini açıkla!
- 11-20. Şek. 5-5 c)'deki yazmacın nasıl yüklendiğini açıkla!

- 11-21. Şek. 5-5 c)'deki yazmacın içeriğinin okunması nasıl gerçekleştiğini yaz.
- 11-22. Şek. 5-6 c)'de nasıl yazmaç gösterilmiştir? Bu yazmaçın blok-diyagramını ve mantıksal sembolünü çiz.
- 11-23. Şek. 5-6 c)'deki yazmaç başlangıç durumuna nasıl getiriliyor?
- 11-24. Şek. 5-6 c)'de verilen okuma ve yazmak için a) iki hatlı, (b) bir hatlı yazmacın yüklenmesi nasıl gerçekleşiyor?
- 11-25. Şek. 5-6 c)'de verilen okuma ve yazmak için a) iki hatlı, (b) bir hatlı yazmaç içeriğinin okunması nasıl gerçekleşiyor?
- 11-26. Şek. 5-6 c)'de verilen yazmaçta 1)1101; 2)1011 verinin a) yazılması, b) okunması için tüm hatların mantıksal durumu nasıl olmalıdır?
- 11-27. Şek. 5-7 a)'da nasıl yazmaç gösterilmiştir? Bu yazmacın blok-diyagramını ve mantıksal sembolünü çiz.
- 11-28. Şek. 5-7 a)'daki yazmaç başlangıç durumuna nasıl getiriliyor?
- 11-29. a)  $M=1$  olunca, b)  $M=0$  olunca Şek.5-7 a)'daki yazmaçta M hattı ne yapıyor?
- 11-30. Şek. 5-7 a)'daki yazmaçın içeriği nasıl okunuyor?
- 11-31. Şek. 5-7 a)'da verilen yazmaçta 1)1001; 2)1110 verisinin a) yazılması için; b) okunması için tüm hatların mantıksal durumu nasıl olmalıdır?
- 11-32. Şek. 5-8 c) ve ç)'de nasıl yazmaç gösterilmiştir? Bu yazmaçın blok-diyagramını ve mantıksal sembolünü çiz.
- 11-33. Şek. 5-8 c) ve ç)'deki yazmaç başlangıç durumuna nasıl getiriliyor?
- 11-34. Şek. 5-8 c) ve ç)'deki yazmaçta içerik hangi hatlar üzerinden giriliyor ve okunuyor?
- 11-35. Şek. 5-8 c) ve ç)'deki yazmacın girişinde 1011 verisi gelince Q1, Q2,...,Q4 flip-flop çıkışlarının, yani yazmaç durumunun zamansal diyagramlarını çiz. Pals-sinyalinin kenarıyla yönetilen flip-floplar yerine a) MS D flip-floplar; b) D mandallar kullanılırsa zamansal diyagramların görünüşü nasıl olacak? Böyle yazmaç doğru çalışıyor mu? Açıkla!
- 11-36. Şek. 5-8 c) ve ç)'deki yazmaçın girişinde 1) 10110; 2)01010 verisi getirilirse yazmaçın zamansal diyagramlarını ve geçiş tablosunu çiz.
- 11-37. Şek. 5-8 c) ve ç)'deki yazmaç içeriğinin okunması hangi prensibe göre gerçekleşiyor? Neden?
- 11-38. Şek. 5-8 a) ve b)'de gösterilen yazmaç içeriğinin okunması sırasında içeriğin kaybolma imkânı var mıdır ve hangi koşullar altında kaybolabilir?
- 11-39. Dizisel giriş hatlı, dizisel çıkış hatlı, silme hatlı ve yeni içeriğin yazılması için kontrol hatlı dört bitli dairesel yazmacın, yani Şek. 5-10'da gösterilen eski yazmaçın dönmesi için mantıksal diyagramı açıkla.
- 11-40. \*) Pals sinyalin kenarıyla yönetilen ve silmek için ayrıdan hatı olan D flip-floplu dört bitli sola ötelemeli yazmaçın mantıksal diyagramını ve mantıksal sembolünü çiz.

- 11-41. Şek. 5-12'deki iki yönlü yazmacın (a)  $M=0$ , (b)  $M=1$  için çalışma şeklini açıkla.
- 11-42. Şek. 5-12'deki yazmaç içeriğinin ilk giren bitin son okunacağı ya da LIFO prensibine göre okunması hangi şekilde yapılabilir?
- 11-43. Şek. 5-15 b)'de nasıl yazmaç gösterilmiştir? Bu yazmacın blok-diyagramını ve sembolik işaretini çiz.
- 11-44. Şek. 5-15 b)'deki yazmaç hangi şekilde yükleniyor? Bu yazmaçın içeriği nasıl okunabilir?
- 11-45. Şek. 5-15 b)'deki yazmaç nasıl dönüşüm imkânı veriyor?
- 11-46. Şek. 5-17 b)'de nasıl yazmaç gösterilmiştir? Bu yazmacın blok-diyagramını ve sembolik işaretini çiz.
- 11-47. Şek. 5-17 b)'deki yazmaç hangi şekilde yükleniyor? Bu yazmacın içeriği nasıl okunabilir?
- 11-48. Şek. 5-19 b)'de nasıl yazmaç gösterilmiştir? Bu yazmacın blok-diyagramını ve mantıksal sembolünü çiz.
- 11-49. Şek. 5-19 b)'deki yazmacın çalışma prensibini açıkla.



# 6. SAYAÇLAR

Bu konusal birimini öğrendikten sonra:

- ⊕ Sayaçların mantıksal yapısını anlayacaksınız;
- ⊕ Asenkron sayaçların çalışma prensiplerini açıklamayı bileceksiniz:
  - ⊕ İkili sayaç,
  - ⊕ Rastgele modüllü sayaç,
  - ⊕ Geriye sayaç
  - ⊕ Karışık çıkışlı yazmaç,
  - ⊕ İkiyönlü sayaç.
- ⊕ Senron sayaçların çalışma prensiplerini ve kullanımlarını açıklamayı bileceksiniz:
  - ⊕ İkili sayaç,
  - ⊕ Rastgele modüllü sayaç,
  - ⊕ Onlu sayaç,
  - ⊕ Dairesel sayaç.
- ⊕ Farklı sayma modüllü sayaçlar projelemeyi becereceksiniz.



## 6.1. GİRİŞ VE TEMEL TERİMLER VE KAVRAMLAR

*Sayaçlar* (İng. *counters*), girişlerinde getirilen dürtüleri sayan ardışık mantıksal ağlardır, çıkışta ise her giriş dürtünün belirli ikili koda sıra sayısını veriyor. Genelde doğal ikili sistemi, yani doğal ikili kodu ya da NBCD veya 8421 kodu kullanılıyor. Bu dijital kurgular, temel yapısal bileşenleri olarak farklı mantıksal devrelerle kombinasyonda sonlu sayıda bellek elemanları, daha doğrusu flip floplar kullanıyorlar.

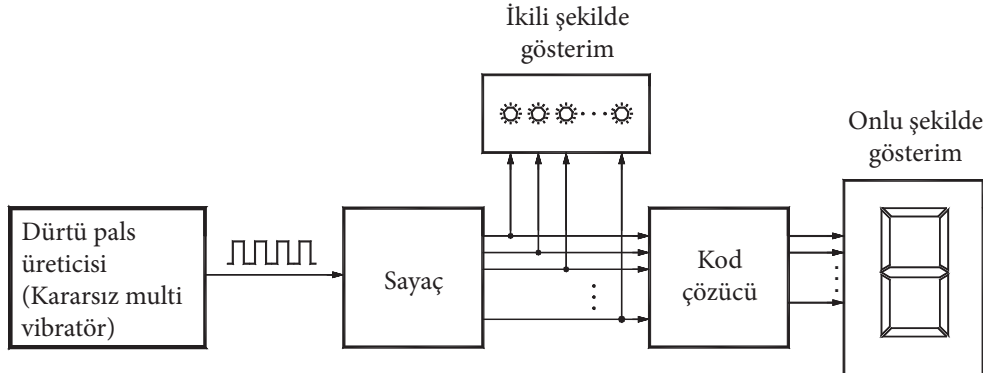
Her flip-flopun iki durumu olduğuna göre,  $n$  flip-floplar kümesi en çok  $2^n$  durumda olabilir. Kümenin bir durumu bireysel flip-flopların durumlarının belirli kombinasyonudur. Sayaçlarda flip-floplar öyle bir mantıksal yapıda bağlıdır ki giriş sinyalinin her dürtü sayaçın durumunun değişmesine ve bir durumdan başka duruma geçmesine yol açılıyor. Sayaç belirli bir başlangıç durumundan saymaya başlarsa ve  $M$  giriş dürtüden sonra aynı duruma geri dönerse, o zaman böyle sayaça *M tabanlı* ya da *M modüllü* sayaç denir. Bir sayaç  $n$  flip floptan oluşuyorsa ve  $2^n$  olası durumdan geçiyorsa, öyle sayaça *ikili sayaç* denir.

Sayaçlar tab.4-10'a göre  $T=1$  girişinin pals sinyalinin her dürtüsü için flip-flop durumunun değişmesine yol açan geçiş (anahtarlamalı) çalışma düzeninde çalışan palsli  $T$  flip-floplara dayanıyor. Söz konusu olan flip-floplar pals sinyalinin arka kenarında aktif olan MS yapılı (İng. master-slave, yani ana-yürütücü)  $T$  flip-floplardır. Flip-floplar basamaklamalı olarak bağlanıyor, öyle ki birinci flip-flop durumunu her sayaç dürtüsünde değiştiriyor, ikinci flip-flop her ikinci dürtü, üçüncüsü her dördüncü dürtüde, dördüncü flip-flop her sekizinci dürtüde vs. değişerek, doğal ikili sistemde farklı ağırlıklı bitlerin değişimine uygundur. Bu yüzden, sayacın çıkış bilgisi, sayaç ağının içeriğinde giren her bir flip-floptan çıkış durumu aynı anda (paralel) alındığı zaman elde ediliyor. Bu yüzden sayaç çıkışı  $n$ -bit uzunluğunda ikili vektör tanımlıyor,  $n$  kullanılan flip-floplar, yani aşamalar sayısıdır.

Buna göre  $T$  flip-flopların basamaklamalı bağlanması ve onların çalışma şekli, sayma dışında sayaç ağın içeriğinde her flip-floptan sonra, giriş sayaç dürtülerin frekansının ikiye bölünmesini sağlıyor ve bu yüzden sayaçlar frekans ayırıcıları olarak da kullanılabilir.

Sayaç ağların pratik uygulamalarında sıkça,  $T$  yerine basamaklamalı bağlanmış JK MS flip-floplar kullanılıyor. Tab. 4-8'e göre JK flip-floplar, geçiş düzeninde  $T$  flip-floplar gibi çalışıyor, çünkü  $J$  ve  $K$  girişlerinde yüksek seviye veriliyor. Şöyle ki, iki girişlerinden dolayı, JK flip-floplar, hem mantıksal yapıları açısından hem sayma tabanına göre farklı sayaç türlerinin yapımlarında daha geniş olanaklar sunuyorlar.

$T$  veya JK basamaklamasının önceki bölümlerde saydığımız çalışma prensibini ve sayaçta çıkış bilgisinin oluşma şeklini göz önüne alarak, sayaç çıkışındaki her sayılan dürtünün, genel bir sayacın blok-diyagramını tanımlayan Şek. 6-1'den görüldüğü gibi LED (ışıklı) diyotlar yardımıyla kolayca gösterilebilen uygun ikili kodlanmış sayı ile ifade edilebileceği açıkça görülüyor.



Şek.6.1. İkili ve onlu gösterimli sayacın blok-diyagramı

Sayma sonuçlarının görsel tanımlanması için, dürtü sayısının insan tarafından anlaşılabilir olması için, Şek.6-1'e göre sayaç çıkışında kod çözücü ve gösterge olarak yedi-bölütlü ekran eklenebilir.

## 6.2. SAYAÇLARIN TABANI VE KAPASİTESİ

Sayaçın durumu, sayaç ağının içeriğinde giren her  $n$  flip-flopun (aşamanın) durumuyla belirlendiğini söylemiştik. Bununla ilgili olarak, her sayacın kendi *başlangıç durumu* vardır, o da birinci giriş dürtüsünün gelmesinden önce flip-floplardan herbirinin durumudur. Her yeni dürtüyle sayaç durumunu değiştirerek, tam bir sayma kapsamın tamamlanmasından sonra, yeniden başlangıç durumuna dönüyor ve yeni döngü başlıyor. Farklı durumların toplam sayısına *sayma döngüsünün uzunluğu* denir ve saymanın yapıldığı sayı sisteminin *tabanını* ya da sayacın *modülünü* tanımlıyor. Sayma tabanına göre sayaçlar iki gruba ayrılıyor:

- ☞ ikili sayaçlar, ve
- ☞ ikili olmayan sayaçlar, ya da rastgele tabanlı sayaçlar.

İkili sayaçların tabanı  $M_0$  ile isaretleniyor ve 2 sayısının herhengi üssü olabilir, örneğin 2, 4, 8, 16 vs. ve şu denklemle belirleniyor:

$$M_0 = 2^n, \quad (6-1)$$

$n = 1, 2, 3, \dots$  sayacın oluşturmasında kullanılan flip-flopların sayısıdır.

Rastgele  $M$  tabanlı sayaçlar, tabanı 10 olan ( $M = 10$ ) onlu sayaçlar ve başka ikili olmayan sayaçlara ayrılıyor.

Verilen sayacın *kapasitesi*,  $N_K$  ile işaretleniyor ve sayacın çıkışında verebileceği en büyük ondalık değeri olarak tanımlanıyor. Sayacın kapasitesi, sayacın tabanı aracılığıyla ifade edilebilir:

$$N_K = M_0 - 1 \quad (6-2)$$

Eğer ikili (doğal ikili kodunda) sayaç ise, ya da Rastgele tabanlı sayaç durumunda

$$N_K = M - 1 \quad (6-3)$$

İkili sayaç söz konusu olduğunda

$$N_K = 2^n - 1. \quad (6-4)$$



Buna göre, sayacın  $N_K$  kapasitesine ulaşılnca, tüm flip-floplar ayarlanmış olacak (çıkışları mantıksal bir durumda bulunacak)

**Örnek:** a)  $n=3$  ve b)  $n=4$  flip-flopla uygulanmış iki ikili sayaç verilmiş olsun. Her biri için tabanı ya da modülü ( $M_0$ ) ve kapasitesi ( $N_K$ ) belirlensin.

a)  $n=3$  için  $M_0=2^3=8$  elde ediliyor ve buradan  $N_K=8-1=7$  ( $=111_{\text{BIN}}$ ) elde ediliyor. Buna göre tabanı 8 olan sayaç, yani sekizli sayaç söz konusu olduğu elde ediliyor

b)  $n=4$  için  $M_0=2^4=16$  ve buradan  $N_K=16-1=15$  ( $=1111_{\text{BIN}}$ ) elde ediliyor. Buna göre 16 modüllü sayaç, yani on altılı sayaç sözkonusudur.

### 6.3. SAYAÇLARIN AYRIMI

Sayaçlar farklı ölçütlere ya da kriterlere göre ayrılıyor. Örneğin sayma modülüne göre önceki bölümde gördüğümüz gibi sayaçlar ikili sayaçlara ve rastgele tabanlı sayaçlara ayrılıyor. Diğer taraftan, dürtünün getirildiği şekile göre *asen kron sayaçlar* ya da başka adıyla *dizisel* ya da *sıralı girişli* sayaçlar var ve *sen kron sayaçlar* veya *paralel girişli sayaçlar* var. Asen kron sayaçlarda dürtüler birinci flip flopun pals sinyal girişinde katılıyor, bir önceki flip-flopun her çıkışı ise bir sonraki flip-flopun palsıyla bağlanıyor. Onlardan farklı olarak, sen kron sayaçlarda dürtü tüm flip-flopların pals girişlerine paralel olarak (aynı anda) getiriliyor.

Sayacın başlangıç ve son durumu sayma yönüne bağlıdır ve bu kriterle göre *ileriye sayaçlar* ve *geriye sayaçlar* olarak ayrılıyor. İleriye sayaçlar, sayacın en küçük değerine uyan durumdan başlıyor ve her yeni dürtüyle en yüksek değere doğru, bir için yükseliyor. En yüksek değere ulaşılnca ardından, tekrar baştan başlanıyor: her önceki durumu bir için arttarak en küçük değerden en yüksek değere doğru gidilerek, sayma süreci kesintisiz sürüyor. İleriye sayaçlara ters olarak, geriye sayaçlar en büyük değerden başlıyor ve her yeni dürtüyle bu değer bir için azalarak en küçük değer doğru sayıyor. En küçük değere ulaşılnca yeniden en yüksek değere dönüp en küçük değere doğru saymaya başlıyor vs. Bu sayaçlarda da sayma döngüsü durmadan tekrarlanıyor.

Açıkladıklarımıza göre, doğal sayı sistemine göre sayan her ileriye sayaç için başlangıç durum sıfırlanmış durum olacak, yani tüm flip-flopların çıkışları 0'dır, ikili geriye sayaç durumunda ise başlangıç durum ayarlanmış durum olacak, yani flip-flopların çıkışları 1 olacak. Demek ki, ileriye sayaç, ondalık yazımla saymak  $0, 1, 2, 3, \dots, N_K-1, N_K$  olarak başlıyor, ardından yeniden  $0, 1, 2, 3, \dots$  olarak devam ediyor vs. Geriye sayaç  $N_K, N_K-1, \dots, 2, 1, 0$  olarak başlıyor, devamda yeniden  $N_K, N_K-1, \dots, N_K$  sayacın kapasitesi, ya da sayacın en yüksek onlu değeridir.

**Örnek:** a) ileriye, b)geriye sayan sekizli sayaç ( $M_0=8$  tabanlı ikili sayaç) verilmiş olsun. Bu sayaç için sayma dizisi belirlensin.

Sekizli sayaç  $n=3$  flip-floptan oluşuyor.  $M_0=2^3=8$  olduğundan dolayı  $N_K=8-1=7$ . a) Sekizli ileriye sayaç  $000$  ( $0_{\text{DEC}}$ ) 'dan başlayarak  $111$  ( $7_{\text{DEC}}$ ) 'e kadar sayıyor ve şu diziyi üretecek:  $\dots, 000, 001, 010, 011, 100, 101, 110, 111, \dots$  ya da onlu yazımda  $\dots, 0, 1, 2, 3, 4, 5, 6, 7, \dots$  b) Sekizli geriye sayaç ters olara sayıyor:  $111$  ( $7_{\text{DEC}}$ ) 'den başlıyor ve  $000$  ( $0_{\text{DEC}}$ ) 'a kadar devam ediyor ve şu dizi elde ediliyor:  $\dots, 111, 110, 101, 100, 011, 010, 001, 000, \dots$  ya da onlu yazımda  $\dots, 7, 6, 5, 4, 3, 2, 1, \dots$

Pratikte sıkça *çevrim (dairesel) sayaçlara* da rastlanabilir. Bu sayaçlar mantıksal yapılarına göre hem asen kron hem sen kron sayaçlardan büyük ölçüde farklıdır. Şöyle ki, çevrim sayaçları da fazla flip-floptan oluşan ardışık ağlar olarak gerçekleşiyor, ancak onlar son flip-flopun çıkışı birinci flip-flopun girişiyle mantıksal devreleri kullanılmadan geri bağlantı ile kapalıdır.

Bunun dışında, dairesel sayaçlar T veya JK flip-floplardan değil, genelde basamaklamalı bağlanmış D flip-floplardan yapılıdır. Bu yüzden yapısal açıdan ötelemeli yazmaçlara andırıyolar. Dairesel sayaçlarda, kapalı mantıksal ağda özel bağlanma şeklinden dolayı, giriş sayaç dürtüler dizisine ileriye/geriye sayaçlarda olduğu gibi artan ya da düşen sıralamada sıradaki sayılar değil, dairesel sayaçın yapısına bağlı farklı ikili kodunda sayılar elde ediliyor.

Sayaçların incelemesi onların analizini ve sentezini kapsıyor. Analiz, bilinen mantıksal yapı verilen sayaçın çalışma şeklini tespit edilmesiyle ve sayacın kombinasyon tablosu ve zamansal diyagramların oluşmasıyla ilgilidir. Diğer taraftan, sentez süreci sırasında önceden belirlenmiş özelliklere: sayaç türü (senkron veya asenkron), sayma tabanı (modülü) (ikili ya da rastgele tabanlı), sayma yönü (ileriye ya da geriye) ve flip-flop türüne (T, JK veya başka) göre sayaç projelendirilmelidir (tasarlanmalıdır). Tam bir görünümün elde edilmesi için kitabın devamında her iki sorun işlenecek.

## 6.4. ASENKRON SAYAÇLAR

*Dizisel veya sıralı girişli* (İng. *Ripple counters*) olarak da adlandırılan *aseenkron sayaçların* en basit yapımları var ve çalışma hızının önemli olmadığı durumlarda kullanılıyorlar.

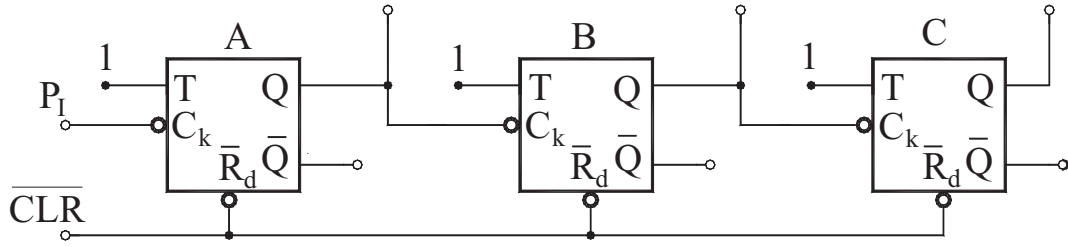
Dikkatimizi ilk önce ileriye sayan sayaçlara yönlendireceğiz, ardından kısaca geriye sayaçların çalışma prensibini inceleyeceğiz. Ayrıca, sayma yönünü seçen özel kontrol girişinin mantıksal durumuna bağlı olarak hem ileriye hem geriye sayan iki yönlü ya da iki taraflı sayaçları da inceleyeceğiz. Sonunda, ikili olmayan tabanlı sayaçların projelendirme sürecini de açıklayacağız.

### 6.4.1. İKİLİ ASENKRON SAYAÇ

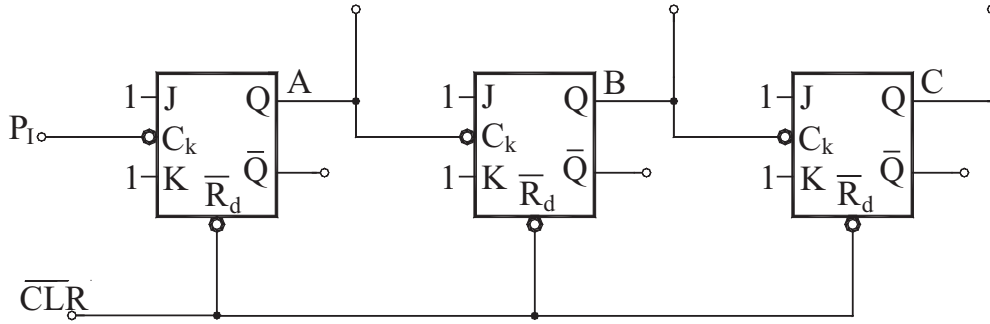
Dijital cihazlarda saymak, en basit uygulamaları olan ikili sayaçlara dayanıyor. Bunun dışında, ikili sayaçların yapısından çekerek, belirli dönüşümlerle oldukça basit şekilde ikili olmayan sayaçlar elde edilebilir.

Herhangi ikili sayacın tabanını ( $M_0$ ) hesaplamak için (6-1) denklemini ( $M_0=2^n$ ) göz önüne alarak ve sayaçların saydığı dürtüleri ikili şekilde gösterdiğini bilerek, 2 sayısının tam üssü olan  $M_0$  tabanlı ikili sayacın gerçekleşmesi için n flip-flopun kullanılmasının gerektiği açıkça görülüyor.

Şek. 6-2 a) ve b)'de üç T ve JK MS flip-flopun uygulanmasıyla (n=3) ikili sayaçın iki temel uygulaması gösterilmiştir. Buna göre üç dereceli sayaç ağ söz konusudur. Flip-flopların doğrudan sıfırlama (silme) girişleri var, aralarında ise çıkışları aracılığıyla basamaklamalı bağlanmıştır. Bu arada tüm flip-flopların T, yani J ve K girişleri mantıksal 1 seviyesinde tutuluyor. Bu yüzden tab. 4-8'de ve tab. 4-10'da son satırlara göre onlar geçiş düzenine göre çalışıyor: her pals dürtüyle flip-flopun durumu önceki durumun tümleşik durumuna değişiyor ( $Q = Q^+$ ). Flip-floplar MS yapılı olduğundan dolayı, durumları pals sinyalin arka kenarın meydana gelmesiyle değişiyor ve her sıradaki aşama, önceki aşamanın çıkışı yüksek seviyeden alçak seviyeye (1'den 0'a) geçtiği anda değişiyor. Buna göre, böyle bir yapıda, flip-floplardan herhangi birinin pals girişinde dürtünün meydana gelmesi (CK), mevcut durumun ona tümleyen duruma değişmesine yol açıyor.



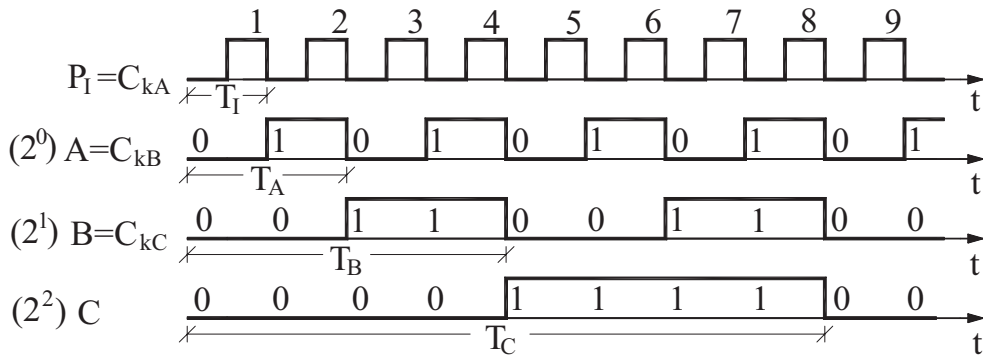
a) T flip-floplu sekizli asenkron sayaç



b) JK flip-floplu sekizli asenkron sayaç

Şek. 6-2. Mantıksal yapı

Sayılması gereken uyarma dürtüler ( $P_I$ ) birinci flip-flopun (A) pals sinyal girişine getiriliyor ( $C_{KA} = P_I$ ), her önceki flip-flopun çıkışı Q ise sıradaki flip flop için palsı tanımlıyor.  $T=1$ , yani  $J=1$  ve  $K=1$  olduğu palslı T, yani JK MA flip-floplar söz konusu olduğundan dolayı, her sayaç dürtüsünün arka kenarının meydana gelmesiyle birinci flip-flopun (A'nın) durumu değişiyor. B ve C flip-flopları da sadece onların öncesindeki flip-flopların çıkışlarında yüksek seviyeden alçak seviyeye değişiklik meydana gelince, yani mantıksal sıfırın elde edildiği anda durumlarını değiştiriyorlar. B flip flopun durumu, ya da çıkışı  $Q_B$ , A'nın çıkışına ( $Q_A$ ) bağlıdır çünkü B'nin palsı aslında A flip-flopun çıkışıdır ( $C_{KB} = Q_A$ ); benzer şekilde C'nin çıkışı ( $Q_C$ ), B'nin çıkışına ( $Q_B$ ) bağlıdır, çünkü C flip-flop için pals B'nin çıkışıdır ( $C_{KC} = Q_B$ ). Sayaçın çalışma şekli en iyi olarak bu bellek elemanlarının çıkışlarındaki gerilimlerin zamansal diyagramlarından görülebilir (Şek.6-3). Ardışık ağ söz konusu olduğundan dolayı, ilk önce başlangıç durumunun belirtilmesi gerekiyor. Bu durumda, ileriye sayaç olduğuna göre, başlangıç durumu  $0_{DEC}$  olması gerekiyor.  $0_{DEC}$  ikili şekilde  $ABC=000$  olarak kodlanıyor, ya da  $Q_A=Q_B=Q_C=0$ . Başlangıç durumu, her flip flopun doğrudan sıfırlama girişine bağlı olan CLR silme girişine kısa süreli alçak seviyenin (mantıksal 0) getirilmesiyle belirtiliyor. Bu şekilde aynı anda tüm flip-floplar sıfırlandırılıyor.



Şek. 6-3. Sekizli asenkron sayacın zamansal diyagramları

Sayacın girişinde birinci dürtünün gelmesiyle A flip-flopu ayarlanıyor ve onun çıkışında yüksek gerilim seviyesi kuruluyor ( $Q_A=1$ ). Bu durum B flip-flopuna etkilemiyor çünkü palsin ön kenarı söz konusudur ve B sıfırlanmış durumda kalıyor ( $Q_B=0$ ). B flip-flopun çıkışı C flip-flopuna için palstır, ancak alçak olduğu için C flip-flopunun durumunu değiştirmiyor ve C flip-flopunu sıfırlanmış kalıyor ( $Q_C=0$ ). İkinci giriş dürtüsü A flip-flopun durumunu tekrar değiştiriyor. Şimdi A sıfırlanıyor ve çıkışında düşen kenar ve alçak gerilim seviyesi meydana geliyor ( $Q_A=0$ ). Bu durum sıradaki aşamanın, B flip-flopun pals girişine aktarılıyor. B flip-flopunu ayarlanıyor ve çıkışında mantıksal bir ( $Q_B=1$ ), ön yükselen kenar meydana geliyor. Yükselen kenar sıradaki aşamaya, ya da C flip-flopuna etkilemiyor ve C sıfırlanmış kalıyor ( $Q_C=0$ ).

Üçüncü dürtü A flip flopunu tekrar ayarlıyor, ancak B ve C flip-floplarına etkilemiyor. ( $Q_A=1$ ,  $Q_B=1$ ,  $Q_C=0$ ). Dördüncü dürtünün ardından A'nın durum değişmesinin iki döngüsü, B'nin bir döngüsü tamamlanmıştır ve C flip flop çıkışının değişmesi başlıyor ( $Q_C=1$ ). C'nin bir döngüsü, ya da durumunun yeniden değişmesi, ancak bu kez yüksek seviyeden alçak seviyeye gelmesi (1'den 0'a) ancak sekizinci dürtüden sonra oluyor, yani tüm flip floplar ayarlanmış durumdan, onlu sistemde 7 sayısına eşit olan  $ABC=111$ , ya da  $Q_A Q_B Q_C = 111$  durumdan, sıfırlanmış duruma  $ABC=000$ , ya da  $Q_A Q_B Q_C=000$  ( $0_{DEC}$ ) geçtiği anda oluyor. Bu anda yeniden başlangıç durumu elde ediliyor ve sayma yeniden 1, 2, 7'ye doğru şekilde devam ediyor. 6-1 denklemden ( $M_0=2^n$ ) de elde edildiği gibi sayılan 8 dürtüden sonra sayma döngüsü de tamamlanıyor. Bu denklemden  $n=3$  için

$$M_0 = 2^3 = 8 \quad (6-5)$$

elde ediliyor. Buna göre bu ikili sayacın sayma modülü (tabanı) 8'dir ( $M_0=8$ ). Bu yüzden bu sayaca *sekizli sayaç* denir.

Sekizli sayacın çalışmasını açıklamak için, sayacın tüm durumları  $S_i$ , her durum için flip flop çıkışlarının ikili şekilde değerleri (kombinasyonları) ve her kombinasyonun uygun indisleri  $K_i$  (onlu değerleri) verilmiş olduğu Tab. 6-1 kombinasyon tablosu da kullanılabilir. Tab. 6-1'den sayacın tüm değerleri  $S_0, S_1, S_2, \dots, S_7$  sekizli sayı sisteminde ikili triyatların kombinasyon değerlerine uygun olduğu görülüyor:  $000, 001, 010, 011, 101, 110$  ve  $111$ , onlu şekilde  $0, 1, 2, 3, 4, 5, 6$  ve  $7$ .

Durumlar		Flip-floplar
$S_i$	$K_i$	C B A
0	0	000
1	1	001
2	2	010
3	3	011
4	4	100
5	5	101
6	6	110
7	7	111
0	0	000

Şöyle ki, sayacın kombinasyon değerleri A, B ve C flip-flopların çıkışları yardımıyla belirlenebilir. Her satırda sayacın bulunabileceği durum yazılıyor ( $S_i$ ) ve her flip-flopun bireysel durumuna bağlı olarak durumu tanımlayan indis ( $K_i$ ) yazılıyor,  $i=0, 1, 2, 3, 4, 5, 6, 7$ .

Durumların ve indis değerlerinin tablodaki sıralaması doğal ikili sayı sisteminde saymayla tamamıyla uyumludur.

Tab. 6-1. Sekizli asenkron sayacın kombinasyon tablosu

Tablodan, birinci aşamanın, yani A flip-flopunun en az ağırlıklı değeri  $2^0$  olduğu, sıradaki B flip-flopun ağırlığı  $2^1$  olduğu, son aşama C'nin ise en büyük pozisyon ağırlığı  $2^2$  olduğu sonucuna varılıyor. Buna göre, onlu işaretlemeye, her sayılan dürtü uygun N sayısı ile ifade edilebilir:

$$N = 2^0 Q_A + 2^1 Q_B + 2^2 Q_C \quad (6-6)$$

Bu sayaç için toplam durumlar sayısı sekizdir, sayma ise  $N=0$  ( $S_0 = Q_A Q_B Q_C = 000$ ) olduğu başlangıç, birinci durumdan başlayarak, sayacın kapasitesine ulaşıldığı  $N=N_K=7$  ( $S_7 = Q_A Q_B Q_C = 111$ ) son-sekizinci duruma kadar devam ediyor.

Genel olarak, n-dereceli ikili sayaç için durumların toplam sayısı  $2^n$ 'dir. Bu arada  $N=0$  olduğu başlangıç durumundan başlayarak,  $N=N_K=2^n-1$  son duruma kadar devam ediyor.

$$N = N_K = 2^n - 1.$$

$$N = 2^0 Q_A + 2^1 Q_B + 2^2 Q_C + 2^3 Q_D + \dots + 2^{n-1} Q \dots \quad (6-7)$$

**Frekans ayırıcısı:** İkili sayaç, Şek. 6-3'te verilen zamansal diyagramlarından kolayca görüldüğü gibi, frekans ayırıcısı olarak da çalışıyor. Her flip-flopun davranışı, önceki flip-flopun aslında çıkışı olduğu pals sinyaline bağlı olmasından dolayı, her flip-flopun çıkışında, önceki flip-flopa kıyasen iki kat daha az dürtü, yani iki kat daha yüksek periyotlu sinyal üretiliyor, öyle ki giriş sayaç dürtülerinin ayırma (bölme) oranı, çıkışın hangi flip-floptan ardından alındığına bağlıdır. Ayırma (bölme) faktörü  $2^i$ 'dir. i flip-flopun sıra numarasıdır ( $i=1,2,3,\dots,n$ ). Böylece, örneğin giriş sayaç dizisi birinci flip-flopun palsidir ve onun A çıkışındaki ayırma oranı  $2^1$ 'dir ( $2^1$ ), B ikinci flip-flopun ardından oran  $4$ 'tür ( $2^2$ ), vs. Bu yüzden, giriş sayaç dürtülerinin frekansı  $f_i = 1/T_i$  ise (T onların periyodudur), aşağıdaki denkleme göre, birinci flip-floptan (A-dan) sonra dürtülerin frekansı  $f_A = f_i/2^1 = f_i/2$  olacak, ikinci flip floptan sonra  $f_B = f_A/2 = f_i/2^2 = f_i/4$  olacak, üçüncü flip-floptan sonra  $f_C = f_B/2 = f_i/2^3 = f_i/8$  olacak vs.

$$f_i = \frac{f_i}{2^i}, \quad i = 1(A), 2(B), 3(C), \dots n. \quad (6-8)$$

Frekansları (6-8) denklemi ile verilen sinyallerden her birinin seçimi, sayacın uygun çoğultayıcı ile bağlanmasıyla elde edilebilir.

#### 6.4.2. GERİYE ASENKRON İKİLİ SAYAÇ

Geriye sayaçların çalışma prensibi ileriye sayaçlara göre terstir. En küçük değerden saymaya bağlayarak en büyük değere doğru sayan ileriye sayaçlardan farklı olarak, *geriye sayaçlar* en büyük değerden başlayarak sayılan her yeni dürtüyle değerleri, en küçük değere-sıfıra ulaşana kadar bir için azalıyor. Sıfıra ulaşıldıktan hemen sonra tekrar başlangıç duruma-sayaçın en büyük değerine girip, yeniden bir için azalıp en küçük değere kadar azalıyor vs. Bu sayma döngüsü durmadan tekrarlanıyor.

İleriye ikili sayaçlarda olduğu gibi, 2 sayısının tam üssü olan  $M_0$  tabanlı geriye ikili sayacın gerçekleşmesi için gereken flip-flop sayısını (n) bildiğimiz (6-1) denkleminden, yani  $M_0 = 2^n$  denkleminden elde ediliyor.

Analiz örneği olarak tekrar 8 tabanlı ( $M_0=8$ ) sekizli ikili sayacı alacağız. Bu sayaç 3 flip-flop ile uygulanıyor ( $n=3$ ), ancak şimdi geriye sayacak. Kombinasyon değerleri, ya da sayaçın geçtiği durumlar tab.6-2 tablosundan verilmiştir.

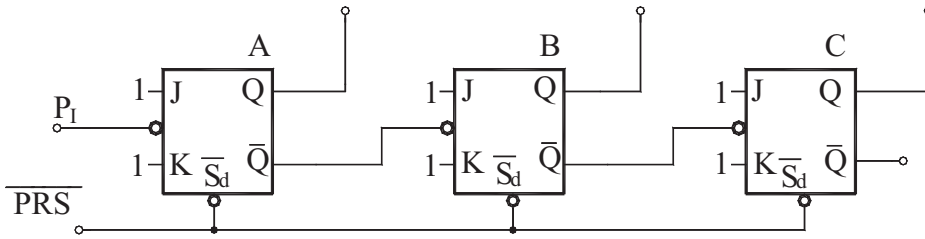
Durumlar		Flip-floplar
$S_i$	$K_i$	CBA
0	7	1 1 1
1	6	0 1 1
2	5	1 0 1
3	4	1 0 0
4	3	0 1 1
5	2	0 1 0
6	1	0 0 1
7	0	0 0 0
0	7	1 1 1

Tab. 6-2. Sekizli geriye sayacın doğruluk tablosu

En yüksek değerden başladığına göre, başlangıç anında tüm flip-floplar ayarlanmış durumdadır:  $Q_A Q_B Q_C = 111$  ( $S_0 = 111$  başlangıç durumu, onlu yazılımda 7'dir). Onların durumu önce 110'a değişiyor ( $S_1 = 110$ , yani 6), ardından 101 ( $S_2 = 101$ , yani 5) vs. Bu süreç, tüm flip-flopların sıfırlanmış olduğu, sayacın son-sekizinci durumuna girene kadar devam ediyor:  $Q_A Q_B Q_C = 000$  ( $S_7 = 000$ , ya da onlu 0). Ardından sayaç yeniden 7'yi gösteren başlangıç durumuna giriyor ve sayma 6,5,4... şeklinde devam ediyor.

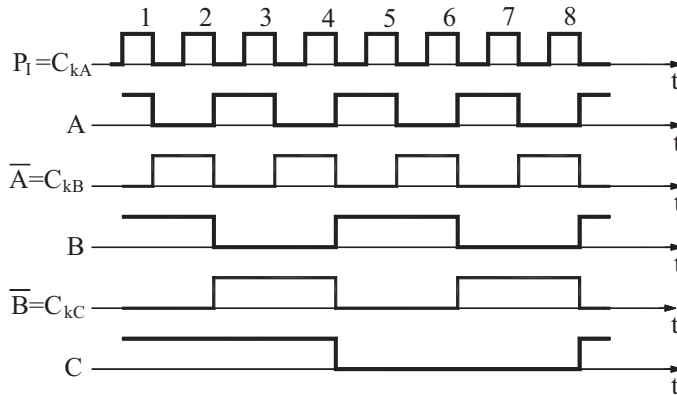
Bu çalışma şekli en basit olarak, ileriye ikili sayacılara benzer şekilde,  $n$  basamaklı bağlanmış T veya JK flip-floplar kullanarak uygulanabilir ancak bu durumda bir flip-floptan diğerine bağlanma doğrudan çıkışlar üzerinden değil, onların tümleyen çıkışları üzerinden yapılıyor.

Sayacın mantıksal yapısı Şek. 6-4'te gösterilmiştir. Bu sayaç geçiş çalışma düzeninde ( $J=1$  ve  $K=1$ ) JK MS flip-floplarla gerçekleşiyor, zamansal diyagramları ise Şek. 6-5'te tanımlanmıştır.



Şek. 6-4. Sekizli geriye sayacın mantıksal yapısı

Geriye sayaç olduğundan dolayı, başlangıç durumu  $S_0$ , ikili şekilde  $ABC=111$  olarak kodlanan onlu yazılımda 7 olmalıdır, yani  $Q_A=1$ ,  $Q_B=1$  ve  $Q_C=1$ . Bu yüzden, kullanılan flip-flopların doğrudan ayarlama girişleri vardır -  $\overline{S_d}$ . Bu doğrudan ayarlama girişlerinde  $\overline{PRS}$  ayarlama girişi bağlanıyor. Bu girişe kısa süreli mantıksal sıfırın getirilmesiyle sayacın başlangıç durumu düzenleniyor, yani tüm flip-floplar ayarlanıyor ( $QAQBQC=111$ ).



Şek. 6-5. Geriye sekizli sayacın zamansal diyagramları



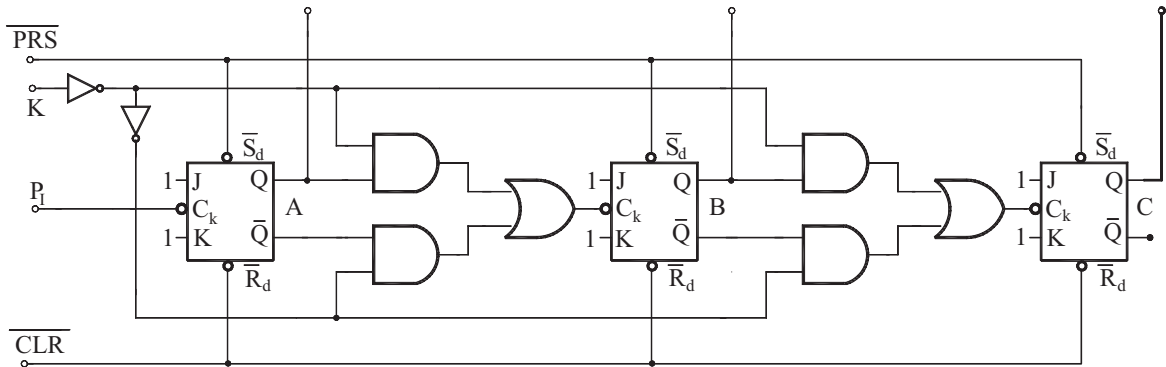
Bu sayacın çalışma prensibi o şekildedir ki birinci flip-flop giren her yeni sayaç dürtüyle durumunu değiştiriyor, tüm diğer flip-floplar ise durumlarını sadece önceki flip-flopun tümleyen çıkışında mantıksal sıfır durumu gelince değiştiriyor. Buna göre, tüm flip-flopların  $Q_A=1$ ,  $Q_B=1$  ve  $Q_C=1$  ayarlanmış oldukları zaman, sayacın  $S_0=111$  başlangıç durumunda bulunduğunu tahmin ederek, birinci sayma dürtünün getirilmesi A flip-flopunu sıfırlayacak ( $Q_A=0$ ) ve onun çıkışında düşen kenar meydana gelecek, tümleyen çıkış ise yüksek seviyeye gidiyor. A flip-flopun tümleyen çıkışı, sıradaki B flip-flopun palsıdır, onda ise gerilimin yükselen kenarı meydana geliyor ( $Q_A=1$ ), B'nin durumuna etkilemiyor ve değişmemiş kalıyor ( $Q_B=1$ ). Sıradaki dürtü A flip-flopunu mantıksal bir duruma geri getiriyor ( $Q_A=1$ ), tümleyen çıkışında, B flip-flopunu sıfırlayan düşen kenar meydana geliyor ( $Q_B=1$ ), ancak palsı B'nin tümleyen çıkışı olan C flip-flopun durumuna etkilemiyor. B flip-flopunda yükselen kenar meydana geliyor ve bu yüzden C ayarlanmış kalıyor ( $Q_C=1$ ). C flip-flopun durumunun ayarlanmış durumdan sıfırlanmış duruma değişmesi, dördüncü dürtünün arka kenarın meydana gelmesiyle oluyor, çünkü o zaman B'nin tümleyen çıkışında düşen kenar meydana gelecek ( $Q_A=0$ ). Sayma döngüsü sekizinci dürtüden sonra, sayacın ayarlanmış flip-floplar başlangıç durumuna ( $S_0=Q_A Q_B Q_C=111$ ) dönerek tamamlanıyor.

### 6.4.3. İKİLİ ASENKRON İKİ YÖNLÜ SAYAÇ

Şimdiye kadar bahsettiklerimizden ileriye ve geriye sayaçların aynı yapıları, ancak farklı ardaşık bağlanmaları olduğunu söyleyebiliriz. Bu durum, her iki sayma şeklini uygulayabilecek mantıksal ağın kullanılmasına yöneltiliyor.

Böyle özelliği olan sayaca *iki yönlü sayaç* (İng.up-down counter) denir çünkü iki yönde sayma olanağı vardır: sayma yönünün kontrol edildiği özel duruma bağlı olarak ileriye ve geriye olanağı vardır.

Asenkron iki yönlü sayacın elde edilmesi için, normal yapısına her iki sayma şekli için flip-flopların doğru bağlanmasını sağlayan mantıksal devrelerin eklenmesi gerekiyor. Geçiş düzeninde ( $J=1$  ve  $K=1$ ) bağlanmış JK MS flip-floplarla uygulanan böyle sayaç Şek.6-6'da gösterilmiştir.



Şek.6-6. Sekizli ikiyönlü sayacın mantıksal yapısı

Sayacın hem ileriye hem geriye sayma olanağı olması gereğinden dolayı, flip-flopların bağlanması, karşılıklı seçkinlik ile sayma yönünün seçimini sağlayan 2-den-1-e çoğullayıcı ile gerçekleşiyor (her biri ikişer VE ve birer VEYA devresiyle uygulanıyor). Sayaç ileriye sayarsa, geriye sayamaz ve tersi. Şekilden görüldüğü gibi çoğullayıcılarda bir giriş hattı önceki flip-floplardan herbirinin doğrudan çıkışın her sonraki flip-flopun girişiyle bağlanmasını sağlıyor, pals girişinin ikinci hattında tümlenmiş çıkışlar bağlanıyor.

Sayma yönünün seçimi, tüm çoğullayıcıların seçim hattı olarak eklenen K kontrol hattıyla gerçekleşiyor. Sayacın ileriye sayması gerektiğinde, sayma yönünü belirleyen bu hatta (K) mantıksal sıfır ayarlıyor (K=0). Bu sinyal tümleme devresi aracılığıyla 1 olarak üst VE devrelere aktarılıp, onları açıyor. Bu VE-geçitlerin girişleri flip-flopların doğrudan çıkışlarıyla bağlıdır, çıkışları ise VEYA devresi üzerinden sıradaki her flip-flopun pals sinyal girişine dağıtılıyor. Bununla aslında her çoğullayıcı sinyalin önceki flip-flopun doğrudan çıkışından, sıradaki flip-flopun pals girişine kadar iletilmesini sağlayarak aynı zamanda ileriye saymayı da sağlıyor.

Geriye sayma gerektiği zaman, K kontrol hattına yüksek seviye getiriliyor (K=1). Bu durumda, tümleyen çıkışların VE-geçitleri açılıyor ve sinyalin önceki flip-flopun tümleyen çıkışlarından sonraki flip flopların pals-girişlerine kadar iletilmesi nedeniyle geriye sayma evirici süreci elde ediliyor.

$\overline{CLR}$  ve  $\overline{PRS}$  girişleri üzerinden sayacın başlangıç durumu düzenleniyor: ileriye saymanın yapılması gerektiği zaman tüm flip-floplar sıfırlanıyor ve bu yüzden  $\overline{CLR}$  girişine alçak seviye,  $\overline{PRS}$  girişine yüksek seviye getirilmelidir ( $\overline{CLR} = 0$  ve  $\overline{PRS} = 1$ ), geriye sayma için ise bu iki girişin uyarılması, önceki duruma göre tümleşik olmalıdır ( $\overline{CLR} = 1$  ve  $\overline{PRS} = 0$ ). Böylece başlangıçta tüm flip-floplar ayarlanacak. Buna göre başlangıç durumu bir giriş hattıyla, örneğin P ile de kurulabilir. Bu durum aynı zamanda tüm flip-flopların  $S_d$  girişlerine doğrudan gönderiliyor ve evirici yardımıyla tümleşik olarak  $R_d$  girişlerine gönderiliyor

#### 6.4.4. RASTGELE TABANLI ASENKRON SAYACIN PROJELENMESİ

Dijital cihazlar ikili sayı sistemine dayandığına göre, temel çalışma prensibi, yani ikili sayaçların saymalarının doğal temeli  $2^n$ 'dir, n kullanılan bellek elemanları (flip-flopları) belirten tam sayıdır. Buna göre, ikili sayaçlar sadece  $M_0$  ikili modüllerle sayabilir. Bu şekilde bir flip-flop ile 2 ( $M_0=2^1$ ) tabanlı (modüllü) sayaç elde edilir, iki flip-flopla 4 tabanlı sayaç ( $M_0=2^2$ ), üç flip-flopla 8 tabanlı sayaç ( $M_0=2^3$ ), dört flip-flopla 16 modüllü ( $M_0=2^4$ ) sayaç elde edilir vs. Ancak diğer taraftan, pratik gereksinimlerden dolayı, sıkça M rastgele tabanlı ikili olmayan sayaçların yapılması aranıyor. Bunu ikili sayacın temel yapılına uygun geri bağlantı gerçekleştirerek yapabiliriz.

n flip-floptan oluşan M rastgele tabanlı sayaç, aynı sayıda flip-floptan (n) oluşan,  $M_0$  doğal tabanına göre sayan ikili sayaçtan daha küçük tabanı olacak, öyle ki aşağıdaki ilişki her zaman geçerlidir:

$$M \leq M_0 = 2^n. \quad (6-9)$$

Böylece, örneğin her dört dereceli sayacın yapısında 4 flip-flopun (n=4) olduğundan dolayı, taban olarak  $M_0=2^4=16$  'dan daha küçük tüm M modülleri olabilir, yani  $M \leq M_0 = 2^4 = 16$  geçerli olmalıdır.

Rastgele tabanlı M ile ikili olmayan sayacın projelenmesi gerektiğini tahmin edersek, demek ki sayma döngüsünün M giriş dürtüden sonra tamamlanması gerekiyor ve ardından sayaç kendi başlangıç durumuna geri dönüyor. M sayma modülünün 2 sayı üssüden farklı olduğunu ve  $M_0=2^n$  olduğu  $M_0$  doğal (ikili) sayma temelinden farklı olduğunu ( $M \neq M_0$ ) bilerek, aralarında  $\Delta M$  ile işaretlenen fark meydana geliyor.  $\Delta M$ , sayacın M tabanıyla sayması için, atlaması gereken durumların sayısını belirtiyor:

$$\Delta M = M_0 - M. \quad (6-10)$$



Sayacın saydığı zaman bulunabildiği durumlar  $M$ , *izin verilir* ya da *geçerli* durumlar olarak adlandırılıyor, atlanması gereken durumlar ise *izin verilmeyen* veya *geçersiz* durumlardır ( $AM$ )

Kullanılmayan, yani yasak  $AM$  durumların atlanması için, prensipte şu çözüm uygulanabilir. İlk önce, sayılan dürtünün ikili kodu çok kısa sürede üretiliyor. Ondan sonra flip-flopların sıfırlanması gerçekleşiyor, yani başlangıç durumuna geri dönme ve sayma süreciyle devam edilmesi gerçekleşiyor. Burada önemli özellik, önceki sayma döngüsünün son geçerli (yasal) durumu ve sıradaki döngüden birinci geçerli durumu arasında geçersiz kuasi-durumun (ara durumun) meydana gelmesidir. Bu ara-durum okunursa, sayma sürecinde hata meydana gelir.

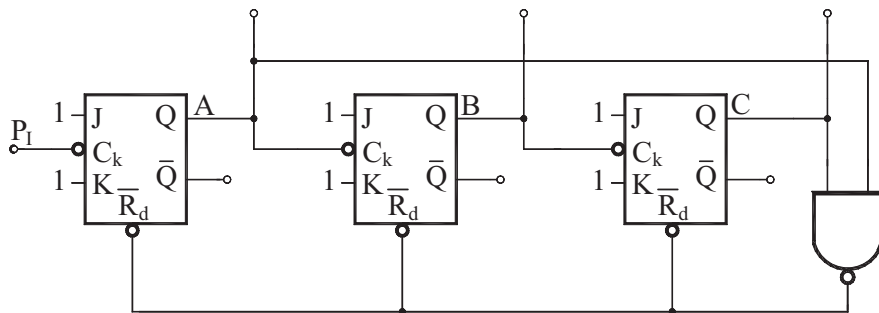
**Örnek:** 5 modüllü sayaç veya beşli sayaç için, birinci yöntemle göre, sayaç ...,000, 001, 010, 011,100,101/000..., ya da onlu yazılımda 0,1, 2, 3, 4,5 /0 dizisini tekrarlayacak. Ara durum 5'tir, geçersiz durumlar ise 110 ve 111 durumlarıdır, yani ondalık yazılımda 6 ve 7 sayılarıdır. Fakat ikinci çözüm uygulanırsa, sayaç 000, 001, 010, 011, 100, 111/000 dizisini , ya da onluk işaretlemeye 0, 1, 2, 3, 4, 7/0 dizisini üretecek. Bu duruma ara-durum onlu 7 olacak, geçersiz (yasak) durumlar ise 101 ve 110 durumları ya da onlu yazılımda 5 ve 6 sayıları olacak. Her iki durumda, geçersiz durumlardan biri, kuasi ya da ara durum olarak, her beşinci dürtüden sonra, bir döngünün bittiği ve yeni döngünün başladığı zaman meydana geliyor.

Önceden verilmiş  $M$  rastgele tabanlı sayacın sentezi için, alçak seviyede aktif olan doğrudan sıfırlama girişli flip-floplar uygulanıyor ( $R_d$ ). Mantıksal yapı, yani gereken bellek elamanların sayısı ve onların bağlanma şekli aşağıdaki adımların uygulanmasıyla elde ediliyor:

1. Gereken flip-flopların sayısı  $n$ , (6-9) koşulundan elde ediliyor, daha doğrusu  $2^{n-1} \leq M \leq 2^n$  geçerli olmalıdır. Bu ilişkide  $M$ , sayacın verilmiş tabanıdır (modülüdür), yani sayma döngünün uzunluğudur;
2. Flip-floplar basamaklamalı bağlanıyorlar ve sıradan  $n$ -dereceli asenkron sayaç gerçekleştiren mantıksal diyagram oluşuyor;
3. Verilen sayma modülünün onlu sayıdan ikili sayıya dönüşümü (çevirmesi) yapılıyor:  $M_{(10)} \rightarrow M_{(2)}$  ( $M_{DEC} \rightarrow M_{BIN}$ );
4. Çıktıları  $M_{(2)}$  sırasında yüksek seviyede ( $Q=1$ ) olan tüm flip-floplar OVE devresinde bağlanmalıdır;
5. OVE devresinin çıkışı flip-flopların tüm doğrudan sıfırlama girişlerine bağlanıyor.

#### 6.4.4.1. 5 TABANLI ASENKRON SAYAÇ

$M$  rastgele tabanlı sayacın tasarlama sürecini, sayma tabanı 5 ( $M=5$ ) olduğu yukarıdaki *beşli sayaç* örneğiyle açıklayacağız.

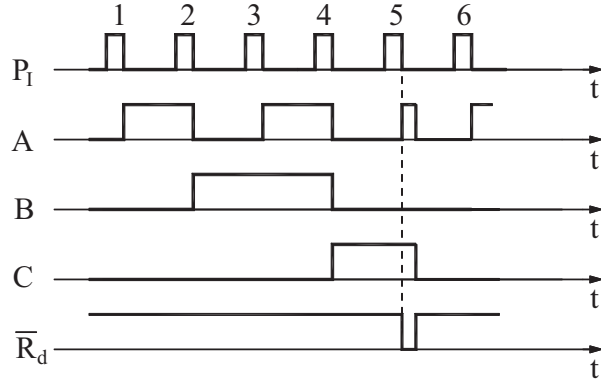


Şek.6-7. Beşli asenkron sayacın mantıksal yapısı

Elimizde  $\overline{R_d}$  doğrudan sıfırlama girişli JK MS flip-flopların olduğu tahmin edeceğiz. Elimizde ger giriş pals dürtüsünde durumlarını değiştirmeleri gereken JK flip-flopların olduğundan dolayı, onların J ve K girişlerine aynı anda yüksek seviye bağlıyoruz, yani sabit 1 durumunda tutuyoruz ( $J=1$  ve  $K=1$ ). Böylece JK flip-floplarını aslında  $T=1$  olduğu T flip-floplara dönüştürüyoruz. Bu şekilde bu flip floplar pals girişinde her dürtüyle durumlarını tümleşik duruma geçtiği geçiş düzeninde çalışıyor.

Durumlar		Flip-floplar		
$S_i$	$K_i$	C B A		
0	7	0	0	0
1	6	0	0	1
2	5	0	1	0
3	4	0	1	1
4	3	1	0	0
0	5/0	1/0	0	1/0

Tab.6-3. Beşli asenkron sayacın doğruluk tablosu



Şek.6-8. Beşli asenkron sayacın zamansal diyagramları

Yukarıda açıklanan sürecin adımlarını sırayla uygulayarak, beşli sayacın uygulaması için 3 flip-flopun gerektiğini elde ediyoruz ( $n=3$ ) ( $2^2=4 < 5 < 2^3=8$ ). Bu arada  $5_{DEC}=101_{BIN}$  olduğundan dolayı, bu sayaç için Şek. 6-7'de verilen mantıksal diyagramı, Şek. 6-8'de verilen zamansal diyagramları ve Tab. 6-4 doğruluk tablosu elde ediliyor.

Şek. 6-8'den sayacın 5-nci dürtüye kadar normal ikili sayaç gibi çalıştığı görülüyor. Bu arada OVE-devresinin çıkışı sürekli olarak yüksektir ve alçak seviyede aktif olan doğrudan sıfırlama girişlerine etkilemiyor. Fakat, 5-nci sayma dürtüsünün düşen kenarı meydana gelmesinin ardından, sayaç ikili yazılımda sayma modülüne uygun ara-(kuazi)-durumuna giriyor ( $M=5_{DEC}=101_{BIN}$ ). Bu yüzden A ve C flip-flopların çıkışları yüksektir, B'nin çıkış ise alçaktır:  $Q_A=1$ ,  $Q_B=0$  ve  $Q_C=1$ . A'nın ve C'nin çıkışları OVE devresine bağlı olduklarından ve ikisi de aynı zamanda mantıksal bir seviyesinde olduklarından dolayı, bu durum OVE-devresinin çıkışı yüksek seviyeden alçak seviyeye gitmesine yol açıyor ve tüm flip-floplar sıfırlandırılıyor. Böylece sayaç başlangıç (doğru) durumuna düzenleniyor:  $Q_A=0$ ,  $Q_B=0$  ve  $Q_C=0$ .

*Onlu sayacın*, ya da 10 tabanlı ( $M=10$ ) sayacın elde edilmesi için, 4 flip-flop gerekiyor ( $n=4$ ): A, B, C ve D. Bunu birinci adımın uygulanmasıyla ya da  $2^{n-1} < M < 2^n$ , ilişkisinin yerine getirilmesiyle elde ediyoruz çünkü  $2^3=8 < 10 < 2^4=16$ . Sayacın modülü 10 ( $M=10$ ) olduğundan çekerek, ikili sayıya dönüşüm yapılıyor ve 1010 ( $10_{DEC}=1010_{BIN}$ ) elde ediliyor. Buna göre, OVE devresinin girişleri B ve D flip-flopların çıkışlarından alınmalıdır ( $1010=Q_D Q_C Q_B Q_A$ ). Buradan sayacın JK ve T türünden flip-flopların uygulanmasıyla mantıksal diyagramı, sayacın zamansal diyagramları ve doğruluk tabloları kolayca tamamlanabilir.

## 6.5. SENKRON SAYAÇLAR

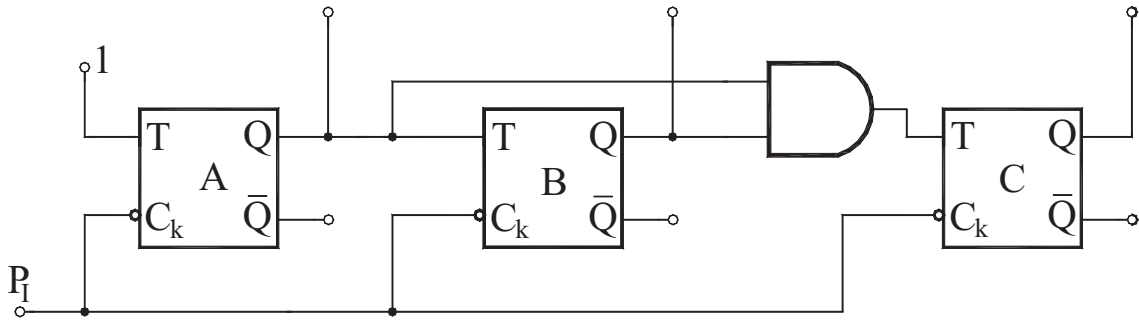
Asenkron sayaçlarda, çalışma hızının sınırlandırılması sayma dürtülerin sadece birinci flip-flopa değil sayaç yapısındaki tüm flip-floplarda pals girişlerine paralel getirilerek kaçınırılabilir. Bu şekilde flip-flopların senkron (eşzamanlı) uyarılması sağlanıyor ve bu yüzden bu sayaçlara *senkron (eşzamanlı) sayaçlar* ya da *paralel girişli sayaçlar* denir. Bu tür sayaçlarda, tüm flip-floplar (dereceler) için durum değişikliği, palsın gelmesiyle aynı anda, yani ona bağlanan sayısal diziyle yapılabilir. Bu değişiklik flip-flopun başlangıç flip flopuna göre sıralamasına bağlı olmayacak, yani bekleme olmayacak. Böylece geçiş düzeni kısalıyor ve bununla tabii ki çalışma hızı da artıyor.

Devamda gelen bölümlerde göreceğimiz gibi, hızın artması sayaç yapısına ek mantıksal devreler ekleyerek sağlanabilir, ancak ek mantıksal devrelerin eklenmesi diğer taraftan sayaç yapısının karışıklığına yol açabilir.

### 6.5.1. İKİLİ SENKRON SAYAÇ

Tüm sayaçların sayılan dürtüleri ikili şekilde gösterdiklerini bilerek, ikili asenkron sayaçlarda olduğu gibi, 2 sayısının tam üssü olan  $M_0$  tabanlı (modüllü) ( $M_0=2^n$ ) ikili senkron sayacın yapılması için  $n$  bellek elemanın (flip-flopun) kullanılması gerektiği açıkça görülüyor.

Senkron sayaçlarda sayma dürtülerin tüm flip-floplarda (derecelerde) pals girişlerine aynı anda götürüldüklerinden dolayı, birinci dürtünün tüm flip-flopların durumlarını aynı anda değişmesini engelleyen ek mantığın kullanılması gerekiyor. Aksi halde, çalışmada hataya yol açılabilir. Sayacın doğru çalışması için birinci derece durumunu her giriş dürtüyle değiştirdiğini, ikinci derecenin sadece önceki derece mantıksal bir durumunda olduğu zaman durumunu değiştirdiğini, üçüncü derecenin uyarılması için ise öncesinde bulunan her iki flip flopun aynı anda mantıksal bir durumunda olması gerektiğini hatırlayalım. Dördüncü derece D durumunu sadece A, B ve C yüksek seviyede oldukları zaman durumunu değiştirmesi gerekecek vs. Bununla ilgili, pals-sinyali göz önüne alarak flip-flopların çalışmasında uyumluluk sağlamak görevi olan VE devrelerin kullanılması gerekiyor,yani bu sayaçlarda sayma dizisi pals-sinyali olarak tüm flip-flopların pals girişlerine aynı anda götürülüyor.



Şek.6-9. T MS flip-foplu sekizli senkron sayacın mantıksal diyagramı

Bununla ilgili olarak, Şek.6-9'da T MS flip-foplarla gerçekleştirilen sekizli senkron sayacın (8 tabanlı ikili sayaç) ( $M_0=8$ ), mantıksal diyagramı için bir örnek verilmiştir.

Birinci flip-flop her giriş dürtüsüne karşılık veriyor, ikincisi her ikinci dürtüye, üçüncüsü ise her dördüncü dürtüye karşık veriyor. Bu yüzden her flip flopun T girişine şöyle uyarı getiriyor:

$$\left. \begin{array}{l} \text{☞ A'ya mantıksal bir } (T_A=1), \\ \text{☞ B'ye A'nın çıkışı } (T_B=A, \text{ yani } T_B=Q_A), \text{ ve} \\ \text{☞ C'ye A'nın ve B'nin çıkışları } (T_C=AB, \text{ yani } T_C=Q_A Q_B). \end{array} \right\} \quad (6-11)$$

Yukarıdaki ifadelere göre, A flip-flopun T girişinde sürekli olarak yüksek seviye bulunmalıdır, A derecesi B derecesiyle doğrudan bağlanıyor, C flip-flopun uyarılması için ise A'nın ve B'nin çıkışları bağlanacağı iki girişli VE devresi gerekiyor.

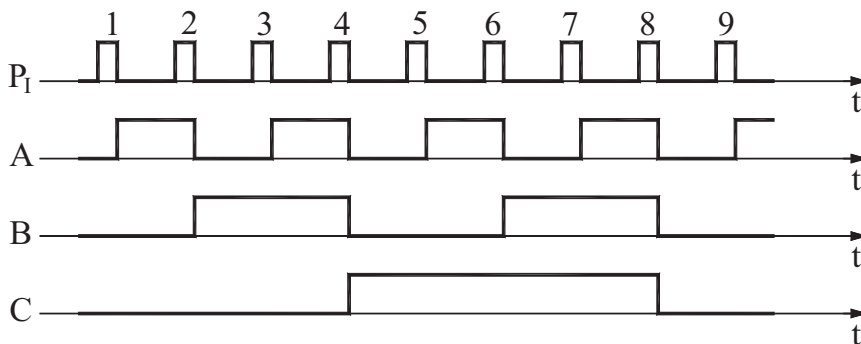
Böyle yapılmı, ardışık olarak üç bitli doğal ikili kodu üreterek asenkron sekizli sayaçta gibi aynı çalışma şekli elde ediyoruz: ... 000, 001, 010, 011, 100, 101, 110, 111 ... ya da onlu şekilde ... 0, 1, 2, 3, 4, 5, 6, 7 ... vs.

Durumlar		Çıkışlar			Girişler		
S <sub>i</sub>	K <sub>i</sub>	C	B	A	T <sub>C</sub>	T <sub>B</sub>	T <sub>A</sub>
0	0	0	0	0	0	0	1
1	1	0	0	1	0	1	1
2	2	0	1	0	0	0	1
3	3	0	1	1	1	1	1
4	4	1	0	0	0	0	1
5	5	1	0	1	0	1	1
6	6	1	1	0	0	0	1
7	7	1	1	1	1	1	1
0	0	0	0	0			

Tab. 6-4. Sekizli senkron ileriye sayacın doğruluk tablosu

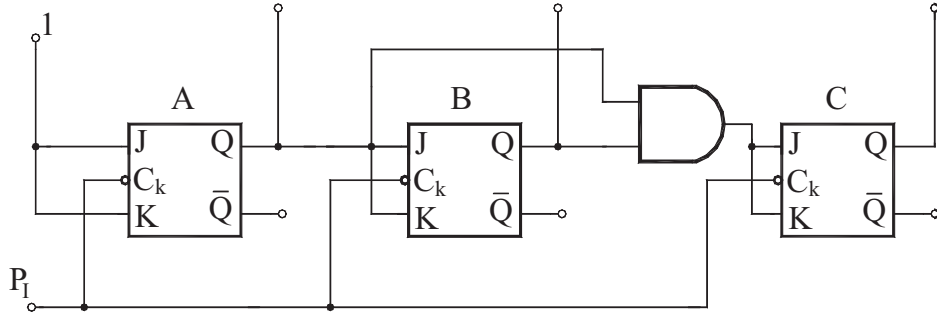
Tab. 6-4 kombinasyon tablosu sayacın tüm durumlarını ( $S_i$ ), onların ikili değerlerini, yani üç flip flopun çıkış kombinasyonları ve onların ondalık yazılımda indisleri ( $K_i$ ) göstermektedir. Herbir durum farklı ikili kombinasyonla ve uygun indisle tanımlanmıştır.

Sayacın durumları doğal ikili sayı sisteminde ikili sayıların sıralamasına göre 000'dan 111'e kadar (onlu yazılımda 0'dan 7'ye kadar) değişiyor. Buna göre, sayaçta durumun sayısı, o durumu gösteren kombinasyonun indisine eşittir ( $S_i=K_i$ ). Sayacın çalışma prensibini daha detaylı açıklayan zamansal diyagramları Şek. 6-10'da verilmiştir.



Şek. 6-10. Sekizli senkron sayacın zamansal diyagramları

Şek. 6-11'de 8 modüllü ( $M_0=8$ ) ikili senkron sayacın başka bir uygulaması gösterilmiştir, ancak burada JK MS flip-flopları uygulanmıştır.

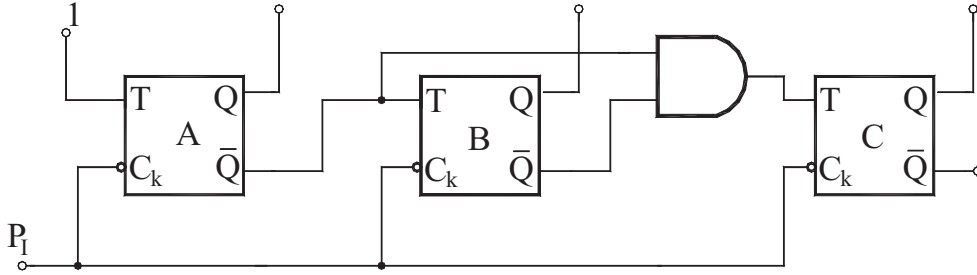


Şek.6-11. JK MS flip-floplu sekizli sayacın mantıksal yapısı

### 6.5.2. İKİLİ SENKRON GERİYE SAYAÇ

Geriye sayaçların çalışma prensibinden çekerek ve ikili asenkron geriye sayaçların ve senkron ileriye sayaçların mantıksal yapılarını bilerek, senkron geriye sayacın mantıksal yapısını tahmin edilebilir. T MS flip-floplarla yapılmış sekizli geriye sayacın öyle bir prensip diyagramını Şek. 6-12'de gösterilmiştir.

Sayaç değerinin bir için azalması önceki flip-floptan tümleyen çıkışın sıradaki flip-flopun girişiyle bağlanarak sağlanıyor. Başlangıç durumu en büyük olduğuna göre ( $S_0=111$ ), sayma en küçüğüne doğru gidiyor ( $S_7=000$ ): ... 111, 110, 101, 100, 011, 010, 001, 000 ya da onlu yazılışla... 7, 6, 5, 4, 3,2,1,0...

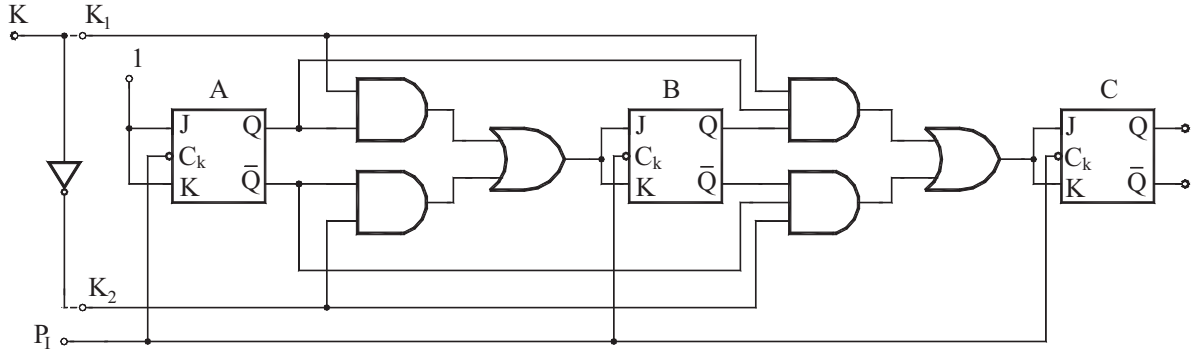


Şek. 6-12. Sekizli senkron geriye sayacın mantıksal yapısı

### 6.5.3. İKİLİ SENKRON İKİYÖNLÜ SAYAÇ

Devamda, dikkatimizi kısaca iki yönde: ileriye ve geriye sayabilen iki yönlü senkron sayacına ayıracağız. Hem ileriye hem geriye sayaçların aynı yapımları, ancak farklı ardışık bağlanmaları olduğundan dolayı, her flip-flopun önünde, 2-den-1-e çoğullayıcı olarak çalışacak ve her iki olanağı gerçekleştirebilecek mantıksal ağın oluşması gerekecek.

Şek. 6-13'teki sayaç karışık sekizli sayaç, yani ileriye veya geriye sayabilen, JK MS flip-floplarla gerçekleşen, paralel girilşi 8 modüllü ( $M_0=8$ ) ikili sayaç örneği verilmiştir. Bunu elde etmek için, her flip-flopun uyarısı, K kontrol hattıyla sayma yönünün seçilmesini sağlayan mantıksal VE-VEYA ağından gidiyor.Şöyle ki, kontrol hattına yüksek seviye getirilince ( $K=1$ ), sıradaki flip-flopları uyararak VEYA devreleri aracılığıyla flip-flopların doğrudan çıkışların bağlı olduğu tüm VE devreleri açarak, sayaç ileriye doğru sayıyor. Aksine, K hattında seviye alçak olunca ( $K=0$ ), sıradaki flip-floplara VEYA devreleri aracılığıyla aktarılan flip-flopların tümleyen çıkışlarıyla bağlı olan tüm VE devreler evirici yardımıyla açılarak, sayaç geriye sayıyor.



Şek.6-13. Sekizli senkron iki yönlü sayacın mantıksal yapısı

#### 6.5.4. RASTGELE TABANLI SENKRON SAYACIN PROJELENMESİ

Pratikte farklı sorunlar sıkça rastgele (ikili olmayan) tabanlı sayaçlar gerektiriyor. Bunlardan en çok onlu sayaç, yani 10 modüllü ( $M=10$ ) sayaçlar kullanılıyor. Bununla ilgili olarak, projelenmesi gereken sayacın  $2^i$ 'den ( $i=1, 2, 3, \dots$ ) farklı tabanı  $M$  ve sayacın yapımı için kullanılan flip-flop'ların türü önceden belirleniyor. Senkron sayaç ağlarının sentezi sırasında, önceden belirlenmiş koşulların yerine getirilmesi için flip-flop'lar üzerine geri bağlantıları gerçekleştiren ek mantıksal devreler belirlenmelidir. Daha doğrusu, sayacın yapısı ve geri bağlantıların gerçekleşmesi için gereken birleşimsel ağın oluşması, sayaç modülü  $M$  ve sayacın yapımı için kullanılan flip-flop'ların türüne bağlıdır.

Atlanması gereken durumlar istenmeyen (kullanışsız, geçersiz veya yasak) durumların ve onların sayısı  $\Delta M$ 'dir ( $\Delta M = M_0 - M$ ). Genel olarak, sayacın bu geçersiz durumlardan birine girme tehlikesi var ve bu durumlarda kalıp hatalı çalışabilir. Bu yüzden, sayacın projelenmesi ek koşullarla yapılabilir, o da sayaç istenmeyen durumdan çıksın ve geçerli, yasak olmayan duruma girsin. Rastgele modüllü sayacın tasarlanması sırasında genelde geçersiz durumdan en sıkça sıfırlanmış durum olan başlangıç durumuna girmesi aranıyor, yani flip-flop'ların tüm çıkışları sıfır olsun.

Senkron sayacın tasarlandığı bu durumda da, önceki ikili ve ikili olmayan sayaçların da uygulamalarında olduğu gibi, en sıkça olarak T ya da JK flip-flop'ları kullanılıyor. Gereken flip-flop'ların sayısı  $n, 2^{n-1} < M < 2^n$  bilinen koşuldaki belirleniyor,  $M$  ise sayacın verilen tabanıdır (sayma döngüsünün uzunluğu). Bu iki durum arasındaki fark, senkron sayacın projelenmesi sırasında tab. 4-9. uyarma (eksitasyon) tablosunu ve tab. 4-11 kullanılan flip-flop'lar tablosunun kullanılması gerekir.

Süreç aranan sayacın doğruluk tablosunun oluşmasıyla başlıyor. Sütunlarda, bağımsız değişkenler olarak A, B, C, ... elimizde olan flip-flop'ların çıkışları alınıyor ( $Q_A, Q_B, Q_C, \dots$ ), fonksiyonlar ise tüm verilen flip-flop'ların T, ya da J ve K girişleriyle tanımlanıyor.

Ardından, sayacın tüm durumlarını ( $S_i$ ), onların ikili kombinasyonlarını ve indislerini ( $K_i$ ),  $i=0, 1, 2, 3, \dots$  yazarak flip-flop'ların çıkışları için ( $M+1$ ) satırın doldurulmasıyla devam ediyor. Sayacın başlangıç, ilk durumu  $S_0$  ve onun indisi olan  $K_0$  ile başlanıyor, ardından ikinci durumla ( $S_1$ ), üçüncü durumla ( $S_2$ ), vs.  $M$ 'inci duruma ( $S_{M-1}$ ) kadar devam ediyor ve ondan sonra bir kez daha başlangıç durumu alınıyor, ancak bu kez onu ( $M+1$ )'inci durum olarak alıyoruz.

Uygun uyarı tablosunu kullanarak, hangi flip-flopları kullanacağımıza bağlı olarak, JK için tab.4-9'da, yani T için tab.4-11'de, kombinasyon tablosundan sağ sütunları, yani fonksiyonların değerleri birer birer dolduruluyor. Bu arada, her flip-flopun önceki durumu sıradaki durumuyla kıyaslanıyor ve durumun değişip değişmediği tespit ediliyor. Durumun değişip değişmediğine göre uyarının nasıl olması gerektiği belirleniyor (T girişinin değeri, yani J ve K girişlerin değerleri) ve bu değer incelenen satırın ve incelenen flip-flop sütunun kesiştiği yerde yazılıyor.

Sonunda her fonksiyonun minimizasyonu ile, daha doğrusu her flip-flopun giriş değişkenlerinden (T yani J ve K), verilen M tabanlı aranan sayacın yapısını tanımlayan tek ardışık ağın elde edilmesi için, n flip-flopun bağlanacağı birleşimsel ağ elde ediliyor.

#### 6.5.4.1. 10 TABANLI SENKRON SAYACIN SENTEZİ

Pratikte, ikili olmayan sayaçlardan en geniş kullanımı, sayma tabanı (modülü) 10 olan ( $M=10$ ) onlu *sayaçların* vardır. Sayma döngüsü 10 durumdan geçiyor, yani 10 farklı ikili durumdan ya da diziden oluşuyor. Bu kadar durum sayısı, dört dereceli sayacın uygulamasıyla elde edilebilir, yani 4 flip-flopun kullanımıyla ( $n=4$ ) elde edilebilir çünkü  $2^3 < 10 < 2^4$ . Dört dereceli sayaç  $2^4=16$  farklı durum sağlayarak, onlardan sadece 10 tanesi kullanılmış olacak,  $\Delta M=16-10=6$  dizi ise kullanılmamış kalacak, daha doğrusu 6 durumun atlatılması gerekiyor çünkü bu durumlar geçersiz olacak. Flip-flopların başlangıç durumuna dönüşü onuncu dürtüden sonra olmalıdır ve onuncu dürtüden sonra bir sayma döngüsü tamamlanıyor. Hangi durumların gereksiz ya da ara durum olarak seçileceği, hangi ikili (BCD) kodun uygulanacağına bağlıdır.

Durumlar		Çıkişlar	Girişler
$S_i$	$K_i$	DCBA	$T_D T_C T_B T_A$
0	0	0000	0001
1	1	0001	0011
2	2	0010	0001
3	3	0011	0111
4	4	0100	0001
5	5	0101	0011
6	6	0110	0001
7	7	0111	1111
8	8	0101	0001
9	9	0101	1001
0	0	0000	

Sıkça olarak doğal BCD kodunda (NBCD ya da 8421 kodu) onlu sayaç kullanılıyor çünkü bu kodun durumlar sıralaması, olası 16 kombinasyondan tüm derecelerde mantıksal sıfır durumundan başlayarak  $Q_A Q_B Q_C Q_D = 0000$  (onlu 0),  $Q_A Q_B Q_C Q_D = 1001$  (onlu 9) olduğu onuncu duruma kadar ilk 10 kombinasyonu kapsayan sıradan ikili sayacın kombinasyonlar sıralamasıyla aynıdır. Son altı durum:  $Q_A Q_B Q_C Q_D = 1010, 1011, 1100, 1101, 1110, 1111$  (onlu 10, 11, 12, 13, 14 ve 15) yastaktır ve kullanılmıyor.

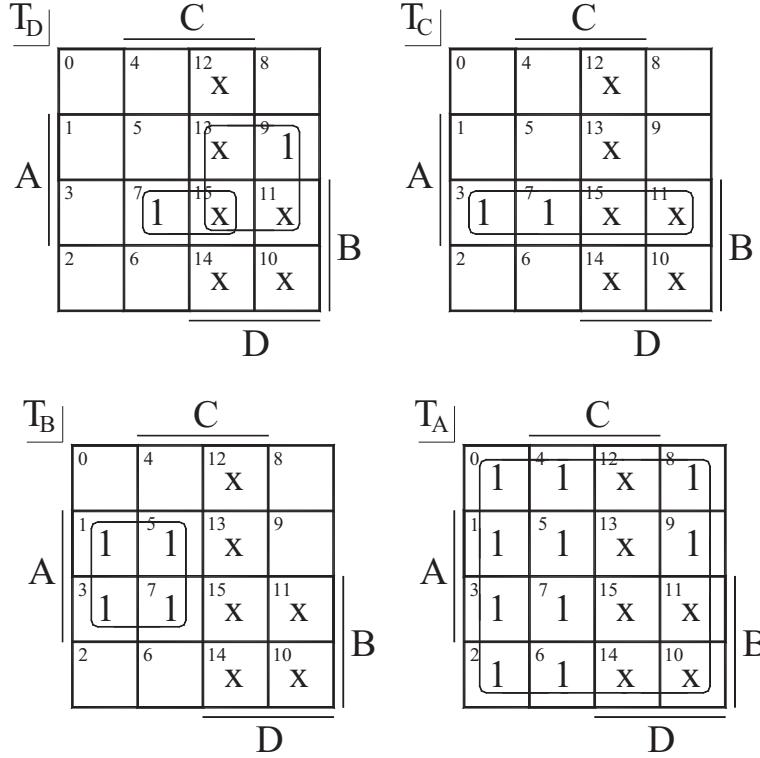
Sayma döngüsü on dürtüden sonra tamamlanıyor ve ardından sayaç yeniden başlangıç durumuna yerleşiyor  $Q_A Q_B Q_C Q_D = 0000$ .

Tab. 6-5. Onlu sayacın doğruluk tablosu



Onlu NBCD sayacın sentezi sırasında elimizde T MS flip-floplar olduğunu tahmin edeceğiz. T flip-flopun eksitasyon tablosunu (t.10-10) kullanarak ve her sayılan dürtüden sonra flip-flopların durumlarını göz önüne alarak tab. 6-5 doğruluk tablosu doldurulmalıdır. Bu tabloya dayanarak kullanılan flip-flopların tüm T girişleri minimizasyonuyla devam ediliyor.

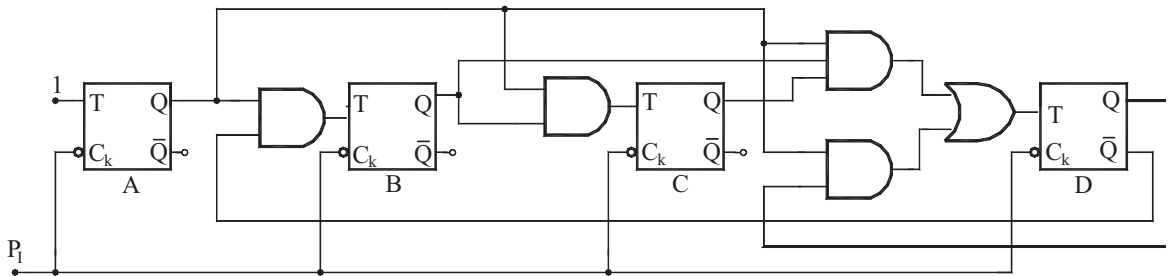
Her fonksiyonun, daha doğrusu flip-flopların giriş değişkenlerin minimizasyonu için, Şek.6-14 a), b), c) ve ç)'de gösterilen Karno yöntemi uygulanıyor. Minimizasyon sürecinde, altı geçersiz durumun: 10,11,12,13,14 ve 15, meydana gelmeyeceği tahmin ediliyor. Bu yüzden bu kombinasyonlar için herhangi fonksiyonun değeri önemsizdir. Bu fonksiyonların değerleri için ya 0 ya da 1 alabiliriz ve bu şekilde „x” ile işaretleniyor.



Şek. 6-14. Onlu senkron sayaçta Karno kartları yöntemiyle minimizasyon

Gerçekleşen minimizasyonun ardından bellek elemanların girişleri için şu mantıksal denklemler elde ediliyor:

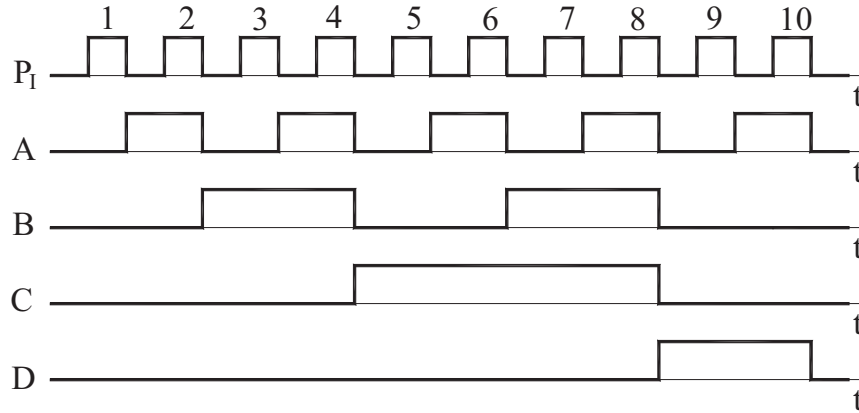
$$\begin{aligned}
 & \text{A flip-flopu için: } T_A = 1, \\
 & \text{B flip-flopu için: } T_B = AD, \\
 & \text{C flip-flopu için: } T_C = AB \text{ ve} \\
 & \text{D flip-flopu için: } T_D = ABC + AD
 \end{aligned}
 \quad (6-12)$$



Şek. 6-15. Onlu NBCD sayacın mantıksal diyagramı



6-12 denklemleriyle aslında mantıksal devreler türü ve onların flip-floplarla bağlanma şekli belirleniyor. Buradan Şek. 6-15'te gösterilen NBCD sayacın mantıksal devresi de elde ediliyor.



Şek. 6-16. Onlu NBCD sayacın zamansal diyagramları

Şek. 6-15'teki mantıksal yapı yanısıra, çalışma sürecini daha iyi anlamak için, Şek. 6-16'da onlu sayacın zamansal diyagramları da gösterilmiştir. Onlardan görüldüğü gibi 10-uncu dürtüye kadar sayaç ikili sayaç gibi çalışıyor,  $A, B, C$  dereceleri ise geçiş çalışma düzeninde T flip-floplar olarak davranıyor:  $A$  aşaması her dürtüye karşılık gösteriyor,  $B$  aşaması her ikinci dürtüye,  $C$  aşaması ise her dördüncü dürtüye karşılık veriyor. Yedinci dürtü ilk üç flip-flopu ayarlıyor, sekizinci dürtü ise son flip-flopu,  $D$ 'yi ayarlıyor ve aynı zamanda diğerlerini,  $A, B$  ve  $C$ 'yi sıfırlandırıyor. Dokuzuncu dürtüden sonra  $B$  ve  $C$  flip-flopları sıfırlanmış olacak, sadece birinci ve son aşama,  $A$  ve  $D$  ayarlanmış olacak. Onuncu dürtü tüm flip floplarda başlangıç durumunu kuruyor ( $A, B, C, D = 0000$ , ya da  $Q_A=0, Q_B=0, Q_C=0$  ve  $Q_D=0$ ). Bu şekilde bir sayma döngüsü tamamlanıyor.

## 6.6. DAİRESEL SAYAÇLAR

Dairesel sayaçların mantıksal yapısı ötelemeli yazmaçlara andırıyor çünkü D türü flip-floplardan oluşuyor. Şimdiye kadar incelediğimiz ikili sayaçlara kıyasen bu en önemli farktır, çünkü ikili sayaçlar genelde T flip-floplara ya da T olarak bağlanmış JK flip-floplara dayanıyorlardı.

Bunun dışında dairesel sayaçlarda, mantıksal devreleri kullanmadan çıkıştan girişe geri bağlantıyla kapanmış basit sayaç ağı söz konusudur. Bu da sayaçlara kıyasen başka bir farktır, çünkü sıradan sayaçlarda flip-floplar basamaklamalı bağlanmıştı (art arda) ve sayma tabanına (modülüne) bağlı olarak, aralarında bağlanmak için ve/veya geri bağlantıların kurulması için belirli mantıksal devrelerin olası uygulanması ile açık veya serbest çıkışları vardı

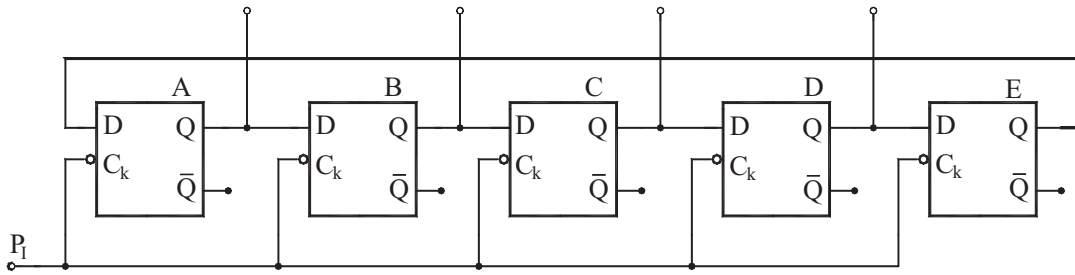
Başka bir fark da sayma tabanı ile kullanılan bellek elemanların sayısı ( $n$ ) arasındaki ilişki dedir. Şöyle ki, ikili sayacın yapılması için  $n$  flip-flop kullanılınca, sayacın tabanı  $M=M_0=2^n$  'e eşittir. Dairesel sayaçlarda modül  $n$  ya da  $2^n$ 'dir, yani  $M=n$  veya  $M=2n$ . Bu özellik bir taraftan dairesel sayaçların ikili sayaçlara karşı bir dezavantajdır, çünkü dairesel sayaçlarda modülün artması, flip-flop sayısının hızlı artmasına yol açıyor. Fakat, diğer taraftan dairesel sayaçların daha basit bağlanma şekli ve çalışma ve sayma prensibi bakımından da avantajları vardır.

### 6.6.1. 5 TABANLI DAİRESEL SAYAÇ

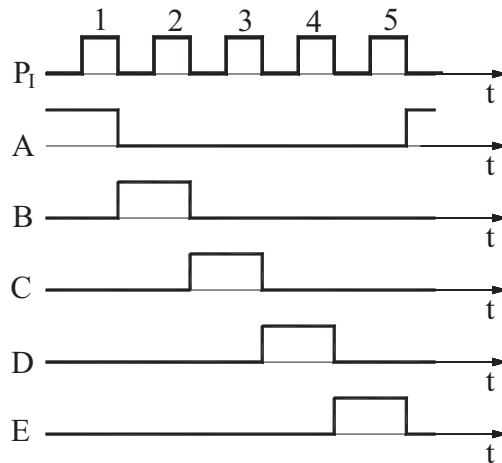
n tabanlı dairesel sayaç için örnek olarak, sayma tabanı 5 olan ( $M=n=5$ ) ve D türünden 5 flip-flop kullanan beşli sayacı inceleyeceğiz. Tab. 6-6 tablosunda, onlu yazılımda uygun indislerle sayacın bütün durumları (flip-flopların çıkış kombinasyonları), ikili şekilde kombinasyon değerleri ve D girişlerin eksitasyon değerleri gösterilmiştir. Tab. 6-6'ya dayanarak, Şek. 6-17'de bu sayaç için mantıksal diyagramı çizilmiştir.

$S_i$	$K_i$	EDCBA	$D_E D_D D_C D_B D_A$
0	1	00001	00010
1	2	00010	00100
2	4	00100	01000
3	8	01000	10000
4	16	10000	00001
0	1	00001	

Tab. 6-6. Beşli dairesel sayacın kombinasyon tablosu



Şek. 6-17. Beşli dairesel sayacın mantıksal diyagramı



Şek. 6-18. Beşli dairesel sayacın zamansal diyagramları

Flip-flopların giriş fonksiyonları için şu denklemler yazılabilir:

$$D_A = EP_P \quad D_B = AP_P \quad D_C = BP_P \quad D_D = CP_P \quad D_E = DP_P \quad (6-13)$$

Denklemlerde kullanılan A, B, C, D ve E, flip-flopların çıkışlarıdır  $Q_A$ ,  $Q_B$ ,  $Q_C$ ,  $Q_D$  ve  $Q_E$ .

Sayacın çalışmasını daha görünür şekilde açıklayabilen zamansal diyagramları Şek.6-18'de verilmiştir. Sayacın başlangıç durumunda sadece ilk aşama ayarlanmış durumdadır ( $A=1$ ), tüm diğer aşamalar ise sıfırlanmış durumdadır ( $B=C=D=E=0$ ). Tüm pals girişlerine paralel olarak birinci sayma dürtüsü getirildiğinde, dürtü birinci ve ikinci flip-flopun: A ve B'nin durumunu değiştiriyor. Şöyle ki, son E flip flopun çıkışına bağlı olan A flip flopun D girişi alçak seviyede bulunuyor ( $D_A=Q_E=0$ ), B'nin D girişine bağlı olan A'nın çıkışı ise yüksek seviyede bulunuyor ( $Q_A=D_B=1$ ). Bu yüzden A flip-flopu sıfırlanıyor, B flip-flopu ise ayarlanıyor. İkinci dürtünün getirilmesiyle B aşaması sıfırlanılarak, sadece C aşaması ayarlanıyor. Tüm diğer çıkışlar alçak seviyede kalıyor.

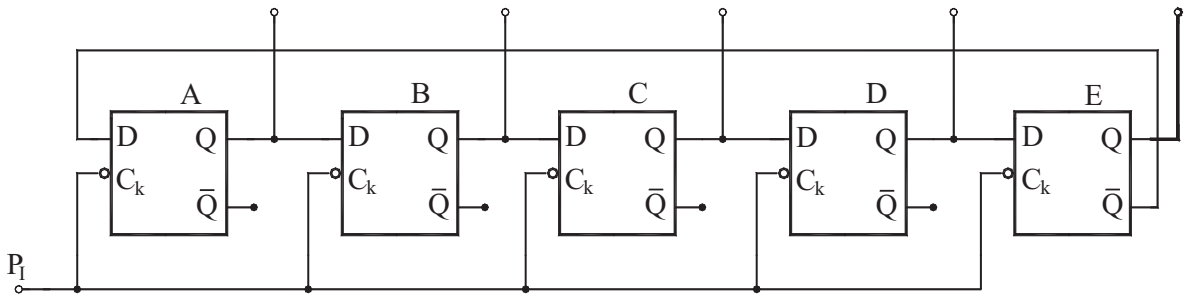
Bu süreç tekrarlanıyor ve her pals sinyaliyle birinci flip-flopun, A'nın çıkışında bulunan yüksek seviye art arda sıradaki flip-floplara: B, C, D ve E'ye aktarılıyor. Son flip-floptan (E) birinci flip-flopa (A) olan geri bağlantı nedeniyle, beşinci dürtü A flip-flopunu yeniden ayarlıyor, E flip flopunu sıfırlanıyor. Bu şekilde sayma döngüsü tamamlanıyor ve başlangıç durumu kuruluyor.

n modüllü dairesel sayaç, JK veya SR flip-flopların kullanımıyla da gerçekleştirilebilir, öyle ki her önceki flip-flopun çıkışı Q her sıradaki flip-flopun S veya J girişine gidiyor,  $\bar{Q}$  çıkışı ise R veya K girişine bağlanmalıdır. Aynı bağlanma şekli son flip-floptan birinci flip-flopa: E'den A'ya doğru uygulanıyor. Aslında A flip-flopun aynı yapılmış söz konusudur, çünkü böyle bağlanma aslında JK veya SR flip-flopların D türünden flip floplara dönüşümüdür.

### 6.6.2. ONLU DAİRESEL SAYAÇ

$2n$  modüllü dairesel sayaç, yapı açısından  $n$  tabanlı sayacın modifikasyonudur. Değişiklik geri bağlantının oluşmasında vardır, öyle ki geri bağlantı son flip-flopun doğrudan çıkışından değil, tümleyen çıkışından birinci derecenin D girişine kadar kuruluyor.  $2n$  tabanlı dairesel sayaç genelde *Johnson sayacı*, ya da *çapraz geri bağlantılı dairesel sayaç* adıyla rastlanıyor.

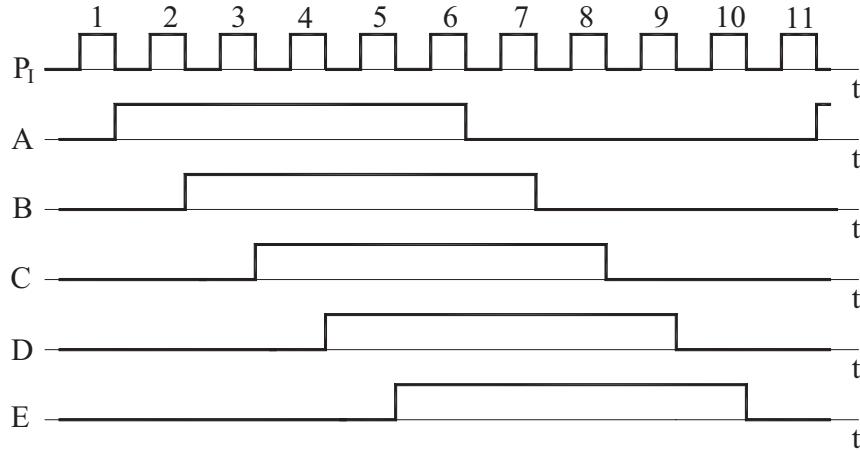
$2n$  modüllü sayaç örneği olarak, modülü 10 olan ( $M=10$ ) Şek. 6-19'daki dairesel sayacı tanıta-  
cağız. Bu onlu sayaç, Şek.6-17'deki 5 modüllü sayaçta gibi benzer şekilde bağlanan D türünden 5 flip-floptan oluşuyor ( $n=5$ ,  $M=2n=2 \times 5=10$ ). Tek fark birinci A flip flopun D girişine, son E flip-flopun doğrudan çıkışın değil, tümleyen çıkışın bağlanmasıdır.



Şek. 6-19. D flip-foplu onlu dairesel sayacın mantıksal diyagramı

Johnson sayacının çalışma prensibini Şek. 6-20'de gösterilen zamansal diyagramlarıyla açıklayacağız. Başlangıçta tüm flip-floplar sıfırlanmıştır ve buna göre sayaç sıfırlanmış durumdadır:  $ABCDE=00000$ , ya da  $Q_A Q_B Q_C Q_D Q_E=00000$ . Birinci sayma dürtünün getirilmesi bi-

rinci flip-flop olan A'yı ayarlıyor, çünkü sadece onun girişinde son aşamanın tümleyen çıkışından yüksek seviyede bulunuyor ( $D_A = \overline{Q_E}$ ), diğer flip-floplar ise sıfırlanmış kalıyor. İkinci dürtü A'nın durumunu değiştirmiyor, onun çıkışı yüksek seviyede kalıyor, aynı zamanda B flip-flopunu da ayarlıyor. Böyle çalışma şekli, ya da C, D, E flip-flopların art arda ayarlanması altıncı dürtüye kadar devam ediyor. Altıncı dürtüde A flip-flopı sıfırlanıyor, tüm diğer flip-floplar ise ayarlanmış durumdadır. Şimdi tüm flip-flopların art arda sıfırlanması başlıyor. A'dan başlayarak, B, C ve D üzerinden E'ye kadar her sayma dürtünün meydana gelmesiyle sıfırlanıyor. Onuncu dürtüde, yeniden başlangıç durumu elde ediliyor ABCDE=00000.



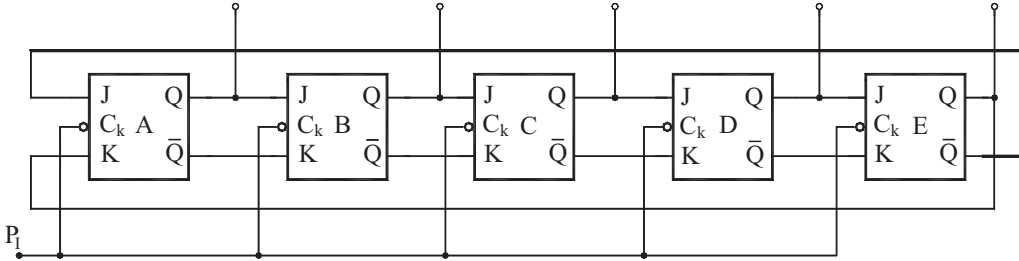
Şek. 6-20. Dairesel onlu sayacın zamansal diyagramları

Tab. 6-7 kombinasyon tablosu yardımıyla sayacın çalışmasını daha detaylı açıklanıyor. Bu tabloda sayacın bulunabildiği tüm 10 durum sırasıyla sunulmuştur.

$S_i$	$K_i$	EDCBA	$D_E D_D D_C D_B D_A$
0	0	00000	00001
1	1	00001	00011
2	3	00011	00111
3	7	00111	01111
4	15	01111	11111
5	31	11111	11110
6	30	11110	11100
7	28	11100	11000
8	24	11000	10000
9	16	10000	00000
0	0	00000	

Tab. 6-7. Dairesel onlu sayacın kombinasyon değerler tablosu

Böyle bir sayacın yapılımları için JK veya SR flip-floplar da D flip-floplara dönüşerek kullanılabilir. Şöyle ki, önceki flip-floptan her doğrudan çıkış sıradaki flip-flopun J girişine bağlanması gerekecek, tümleyen çıkış ise K girişine bağlanacak ( $J=Q$ ,  $K=\bar{Q}$ ), son flip-floptan ilk flip-flopa doğru bağlantı, Şek. 6-21'de gösterilen mantıksal blok diyagrama göre, son flip-flopun doğrudan çıkışı birinci flip-flopun K girişine, tümleyen çıkış ise J girişine bağlanarak kuruluyor.



Şek. 6-21. JK flip-floplu onlu dairesel sayacın mantıksal diyagramı

## TEKRARLAMA SORULARI VE ÖDEVLERİ

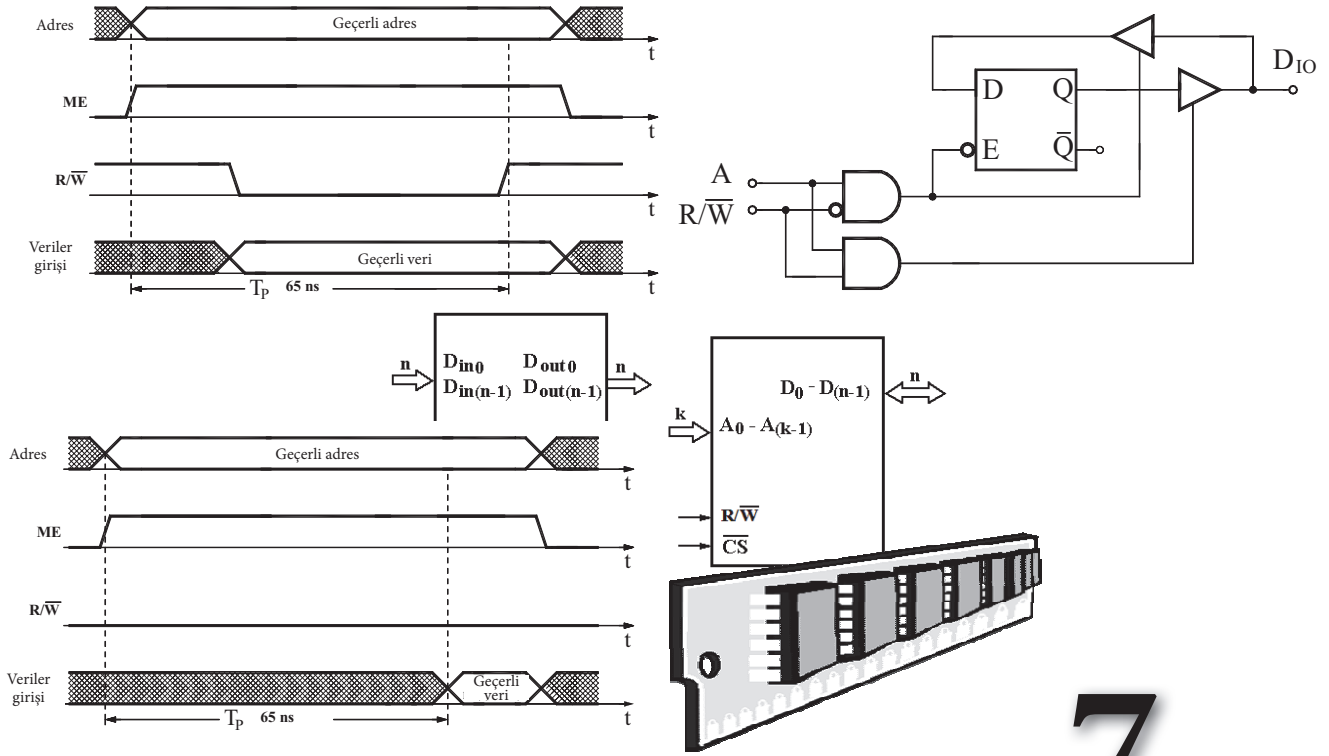
- 6-1. Sayaçların temel amacı nedir? Sayacın yapısını oluşturan temel yapı elemanları hangileridir?
- 6-2. Verilen dijital bileşenlerden hangileri sayaçların temel yapı elemanı olarak kullanılabilir : a) pals-sinyalin kenarıyla yönetilen flip-floplar; b) MS flip-floplar; c) mandal(latch) devreleri? Cevabımı açıkla.
- 6-3. Sayacın bir durumu neyi belirtiyor?
- 6-4. Sayacın yapısında kullanılan flip-flopların sayısı ve sayacın bulunabileceği durumların sayısı arasındaki ilişki nasıldır?
- 6-5. Sayacın sayma tabanı (modülü) nedir? Sayma döngüsü nedir? Sayacın kapasitesi nedir? Bunlar sayacın yapısında kullanılan flip-flopların sayısı ve sayacın tabanı ile nasıl bağlıdır?
- 6-6. 1)  $n=2$ , 2)  $n=3$ , 3)  $n=4$  flip-floptan oluşan sayacın a) sayma tabanı ve b) kapasitesi nedir?
- 6-7. Sayma tabanına (modülüne) göre sayaçlar nasıl ayrılıyor?
- 6-8. Sayacın yapılımları için hangi flip-flop türleri en sıkça kullanılıyor? Nasıl çalışma düzeninde çalışıyorlar? Nasıl bağlanıyorlar?
- 6-9. Sayaçlar frekans ayırıcıları olarak kullanılabilir mi? Cevabı açıkla!
- 6-10. a) ikili; b) onlu gösterimde sayacın blok-diyagramını çiz.
- 6-11. İkili sayaçlar ve ikili olmayan tabanlı sayaçlar arasındaki fark nedir?
- 6-12. Sayaçlar sayma yönüne göre nasıl ayrılıyor?
- 6-13. Dürtülerin bağlanma şekline göre hangi sayaç türleri vardır?
- 6-14. Asenkron (paralel) ve senkron (dizisel) sayaçlar arasındaki fark nedir?
- 6-15. a)  $n=2$ ; b)  $n=3$ ; c)  $n=4$  flip-flop ile gerçekleşen ikili sayaç hangi durumlardan geçiyor? Bu kombinasyonlar ikili ve onlu işaretlemelerde hangileridir ve hangi sıralamaya göre tekrarlanıyorlar?

- 6-16. Verilen sayacın analizi neyi kapsıyor? Ne biliniyor, ne aranıyor?,
- 6-17. Belirli sayacın tasarlanması için neyin bilinmesi gerekiyor?
- 6-18. a)  $n=2$ ; b)  $n=3$ ; c)  $n=4$ ; ç)  $n=5$  flip-flopla gerçekleşen ikili asenkron sayaç verilmiş olsun. Bu sayacın 1) Modülünü ( $M_o$ ), 2) Kapasitesini ( $N_K$ ), 3) Başlangıç durumunu ve 4) Son durumunu belirle.
- 6-19.  $\overline{R_d}$  doğrudan sıfırlama girişi olan 1) T, 2) JK, SR MS(master-slave) flip-flopla gerçekleşen a)  $M_o=4$ ; b)  $M_o=8$ , c)  $M_o=16$  tabanlı ikili asenkron sayaç için mantıksal diyagramı, kombinasyon tablosunu ve zamansal diyagramları çiz.
- 6-20.  $+V_{cc}=5V$  güç kaynağıyla beslenen ve  $T_p = [\mu\text{sec}]$  peryotlu dürtü dizisiyle uyarılan a)  $M_o=4$ ; b)  $M_o=8$ , c)  $M_o=16$  modülüne göre asenkron ikili sayaç verilmiştir. Sayacın zamansal diyagramlarını çiz ve giriş dürtülerin frekansını ( $f_p$ ) ve birinci ( $f_1$ ), ikinci ( $f_2$ ),..., son flip-flopın çıkışında elde edilen dürtülerin frekanslarını hesapla.
- 6-21.  $\overline{R_d}$  doğrudan sıfırlama girişi olan 1) T, 2) JK, SR MS(master-slave) flip-flopla gerçekleşen a)  $M_o=4$ ; b)  $M_o=8$ , c)  $M_o=16$  tabanlı ikili asenkron geriye sayaç için mantıksal diyagramı, kombinasyon tablosunu ve zamansal diyagramları çiz.
- 6-22. İkiyönlü sayacın en önemli özelliği nedir? İkiyönlü sayaçta iki yönde sayma olanağı veren temel prensip hangisidir?
- 6-23. Elimizde a)  $n=3$ ; b)  $n=4$ ; c)  $n=5$  flip-flopun olduğunu tahmin ederek, ikili olmayan  $M \neq M_o$  tabanlı sayaçların tüm olası modüllerini belirle.
- 6-24. Rastgele tabanlı sayaçlarda geçerli (izin verilir) ve geçersiz (izin verilmeyen, yasak) durum arasında temel fark nedir?
- 6-25. Rastgele tabanlı asenkron ikili olmayan sayaçların elde edilmesi için süreç (işlem) adımlarını ayrı ayrı say ve açıkla.
- 6-26. Elimizde  $\overline{R_d}$  doğrudan girişli 1)T, 2) JK MS flip-floplarımız varsa,  $M=$  a)5; b)6; c)7; ç) 9; d)10; e)11; f)12; g)13; h)14; ı)15 rastgele tabanlı asenkron sayaç projeye.
- 6-27. Asenkron ve senkron (dizisel ve paralel) sayaçlar aralarında neye göre farklıdır?
- 6-28.  $\overline{R_d}$  doğrudan sıfırlama girişi olan 1) T, 2) JK MS flip-flopların uygulanmasıyla a)  $M_o=4$ ; b)  $M_o=8$ , c)  $M_o=16$  tabanlı ikili senkron sayaç için mantıksal diyagramı, kombinasyon tablosunu ve zamansal diyagramları çiz. Elimizde gerektiği kadar girişli VE devreleri vardır.
- 6-29.  $M \neq M_o$  rastgele tabanlı senkron sayaçların tasarlama prensibini açıkla.
- 6-30. 1) JK, 2) T MS flip-floplar kullanarak a) 5; b) 6; c) 9; ç) 10; d) 11, e)6; f)14 tabanlı senkron sayaç projeye. Çözme sürecinde 1) sayacın hiçbir zaman geçersiz durumda bulunmayacağını, 2) geçersiz durumda bulunuyorsa başlangıç (sıfırlanmış) duruma geçmesine gerektiğini tahmin et.
- 6-31. Dairesel sayaçların mantıksal yapısının oluşmasında flip-flopların temel bağlanma şekli hangisidir?
- 6-32. Kullanılan flip-flopların sayısı ve onlarla yapılan dairesel sayacın sayma tabanı (modülü) arasında nasıl ilişki vardır?

- 6-33. Asenkron ve senkron sayaçlar arasında dairesel sayaçlara göre hangi farklar vardır?
- 6-34. Elimizde a)  $n=4$ , b)  $n=5$ , c)  $n=6$  flip-flopumuz varsa, sayma tabanı açısından nasıl dairesel sayaçların gerçekleştirilebileceğini cevapla.
- 6-35. Beşli dairesel sayacın çalışma prensibini incele.
- 6-36. 1)D, 2)JK, 3) SR MS flip-floplar uygulayarak a) $M=4$ , b) $M=5$ , c) $M=6$  tabanlı dairesel sayacın mantıksal diyagramını, doğruluk tablosunu ve zamansal diyagramlarını çiz. Çalışma prensibini açıkla.
- 6-37. Onlu dairesel sayacın çalışma prensibini incele.
- 6-38. 1)D, 2)JK, 3) SR MS flip-floplar uygulayarak a) $M=8$ , b) $M=10$ , c) $M=6$  tabanlı dairesel sayacın mantıksal diyagramını, doğruluk tablosunu ve zamansal diyagramlarını çiz. Çalışma prensibini açıkla.







# 7. BELLEK BİLEŞENLERİ

Bu konusal birimi öğrendikten sonra

- ⊕ Bellek bileşenlerle ve cihazlarla ilgili temel terimleri ve kavramları bileceksiniz;
- ⊕ Belleğin organizasyonunu tanıyacaksınız;
- ⊕ Farklı bellek türlerini: ROM, PROM, EPROM, EEPROM, RAM belleklerini kıyaslayabileceksiniz;
- ⊕ Bellek tümleşik devreler arasındaki benzerlikleri ve farkları açıklayabileceksiniz;
- ⊕ Bellek adresleme şekillerini açıklayabileceksiniz;
- ⊕ RAM bellek hücresinin mantıksal yapısını ve çalışmasını tanıyacaksınız;
- ⊕ Bellekten okuma ve yazma süreçlerin nasıl gerçekleştiğini anlayarak çalışma prensibini anlayacaksınız;



## 7.1. GİRİŞ

Bellek bileşenleri dijital sistemlerin temel ve ana parçalarından biridir. Belleğin, özellikle bilgisayarlarda temel rolü vardır, çünkü esas rolü ikili şekilde kodlanmış bilgiler tanımlayan farklı veri türlerini korumaktır (ezberlemektir). Belleğin temel işlevi girilen verilerin okunmasını sağlamaktır ve aynı zamanda yeni veriler yazdırarak onların değişmesini (yenilenmesini) sağlamaktır. Bellekler iki kutuplu ve tek kutuplu elektronik (yarı iletken) elemanlarla gerçekleşen büyük sayıda mantıksal devrelerden oluşan özel bileşenlerdir.

Bu konuda ilk önce bellek bileşenlerin ve cihazların hiyerarşisi, terminolojiyi ve belleklerle ilgili en önemli terimleri tanıyacağız. Özel dikkat yarı iletken belleklere ve onların iç organizasyonuna verilecek. Ardından farklı kriterlere göre belleklerin ayrımını inceleyeceğiz. Bu arada birçok bellek bileşenler türleri arasındaki benzerlikleri ve farklılıkları karşılaştırarak onların avantajları ve dezavantajları hakkında uygun sonuçlar getireceğiz.

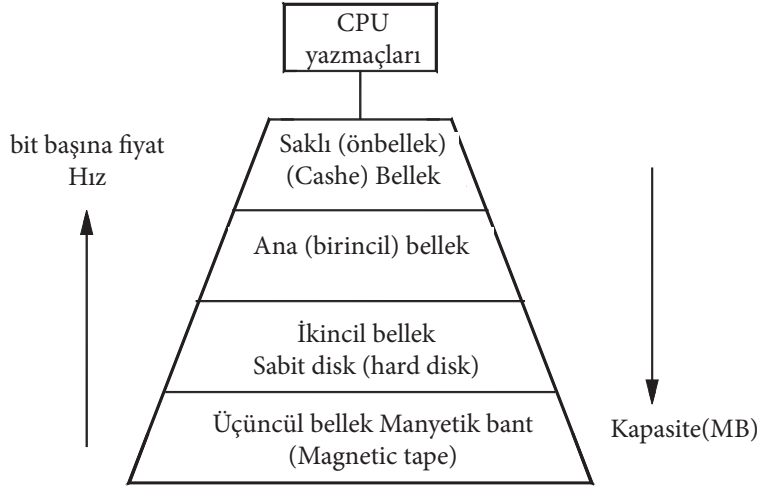
Bellek bileşenlerin çalışmasını daha kolay anlamak için şu kavramlar tanımlanacak ve incelenecektir: bellek modeli, temel elektrik şeması-mantıksal yapı ve belleğin en küçük elementer yapı birimi olan bellek hücrelerin mantıksal diyagramları. Bununla ilgili, giriş ve çıkış hatların rolünü inceleyerek ve iki temel işlemin: okuma ve yazmanın incelenmesi sırasında zamansal diyagramları kullanarak belleğin çalışma prensibi açıklanacaktır.

## 7.2. BELLEK HİYERARŞİSİ

Konu olarak bellek incelendiği zaman, bellek hiyerarşi piramidin incelenmesi kaçınılmazdır. Bellek hiyerarşi piramidi bellek bileşenlerin ve cihazların ayrımını grafiksel (resimli) şekilde göstererek aynı zamanda maliyetleri, çalışma hızı ve kapasitesi arasındaki ilişkiyi tanıtıyor.

Şek. 7-1'de gösterilen bellek piramidi, tepesinde (ucunda), diğer bellek türlerine göre en küçük kapasitesi, ancak diğer taraftan en yüksek çalışma hızı olan saklı bellek ya da önbellek (İng. Cache) bulunuyor. İkinci seviyede ana ya da birincil bellek yer alıyor. Ana bellek önbellekten kapasiteye göre daha büyüktür, ancak çalışma hızı açısından daha yavaştır. Bu iki en yüksek seviyede bulunan belleklerin özelliği yarı iletken elemanlardan yapılı olmalarıdır. Onlar günümüzde genelde tek kutuplu NMOS transistörlerden ya da tümleyici MOS (CMOS) teknolojisinde yapılıyorlar. Sıradaki hiyerarşi seviyenin en önemli temsilcileri sabit diskli birimleridir (İng. hard-disk). Onlar ikincil bellek temsilcileridir. Manyetik bant (İng. Magnetic tape) birimleri daha alçak seviyeye, ya da üçüncül bellek seviyesine aittir.

Son iki bellek cihaz türü aslında manyetik belleklerdir, çünkü verilerin okunması ve yazdırılması manyetik ortamda yapılıyor: hareket etmeleri için mekanik elemanın gerektiği miknatıslanmış ince diskler ya da miknatıslanmış bantlar. Bunun sonucu olarak, bu bellek cihazların en küçük çalışma hızları var, ancak yarı iletken belleklere göre çok daha büyük kapasiteleri vardır.



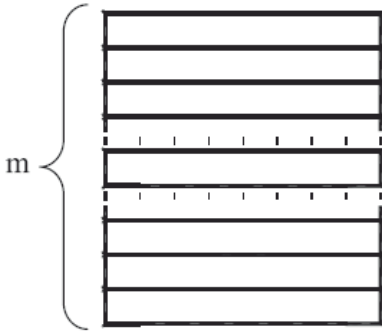
Şek. 7-1. Bellek hiyerarşi piramidi

Şek. 7-1'den yarı iletken bellek bileşenlerin, manyetik belleklere kıyasen yüksek çalışma hızı açısından avantajları olduğu, ancak olumsuz tarafları küçük kapasite ve bit başına yüksek fiyatları olduğu görülmüştür. Diğer taraftan, piramidin dibinde bulunan manyetik bellek cihazların bit başına en alçak fiyatları ve en büyük kapasiteleri vardır, fakat yarı iletken belleklere kıyasen en büyük zayıflıkları küçük çalışma hızıdır.

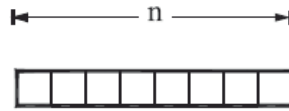
Devamda, en yüksek hiyerarşi seviyesinde, önbellek seviyesinde bulunan bellek yarı iletken tümleşik bileşenlerin çalışma prensibinin analizinin ve özelliklerin sunulmasına dikkat veracağız.

### 7.3. BELLEĞİN İÇ ORGANİZASYONU VE TEMEL TERİMLER VE KAVRAMLAR

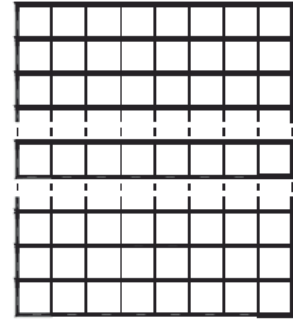
Organizasyon açısından belleğin mantıksal yapısı, büyük sayıda bellek konumları ( $m$ ), Şek. 7-2'de basit şekilde gösterilmiş olduğu gibi tablo veya matris şeklinde sonlu ve düzenlenmiş kümedir.



Şek. 7-2. Bellek tablosu



Şek. 7-3. Bellek sözcüğü



Şek. 7-4. Bellek matrisi

Her konumda, Şek. 7-3'e göre bitlerle ifade edilen sabit uzunluğunda ( $n$ ), 0-lar ve 1-ler dizisi (kombinasyonu) olarak ikili şekilde kodlanan veri tanımlayan belli *bellek sözcüğü* bulunabilir. Bellek sözcüğü belirli bellek konumunda yerleşebilir. Bellek konumu, Şek. 7-4'e göre sözcüğün uzunluğuna (bitler sayısına) eşit son sayıda bellek hücresi içeriyor. Şöyle ki, bellek hücresi bellek bileşenin en küçük yapı birimini tanımlıyor, çünkü onda sadece bir bit depolanabilir

(belleklenir) (tek bitli veri): 0 ya da 1. En sol pozisyonda bulunan verinin en değerli biti (MSB) (en büyük ağırlıklı biti), genelde  $d_{(n-1)}$  ile işaretleniyor, yanındaki  $d_{(n-2)}$  ve en sağda bulunan ve  $d_0$  ile işaretlenen en az değerli (LSB) (en küçük ağırlıklı bit) bite kadar öyle devam ediyor. Pratikte sözcükleri 1, 2, 4 veya 8 bayt uzunluğunda olan bellekler kullanıldığından dolayı,  $n$  genelde 8, 16, 32 ya da 64 olacak. Her bellek sözcüğünün ikili şekilde kodlanmış ve uzunluğu  $n$  bit olan belirli verinin olduğunu tahmin ederek, bellek konumunda yerleşmiş söz konusu olan ikili vektör, yani 0-lar ve 1-ler kombinasyonu bellek konumun içeriğidir. Buna göre, bellek çerçevesinde her *bellek sözcüğü*, içeriğinin okunması ya da yeni içeriğin yazılmasıyla değişmesi amacıyla ulaşılabilen  $n$  bit uzunluğunda ayrı bir bütün (öge) tanımlıyor. Bellek bileşeni kendi işlevini, *okuma ve yazma* iki temel işlemin uygulanmasıyla gerçekleştiriyor.

Bununla ilgili olarak, *yazma terimi* kullanıldığı zaman, bellek dışında bulunan bileşenden ya da cihazdan gelen bellek sözcüğün (yeni verinin) depolanmasına (belleklenmesine, girilmesine, İn.Write) düşünülüyor ve kullanılabilir tüm bellek yerleri kümesinden verilen bir (herhangi) bellek yerin yeni içeriği olarak yazılıyor. Belirtilen bellek konumun eski içeriği geri dönmez şekilde kayboluyor, çünkü yeni içerikle değişmiş olacak.

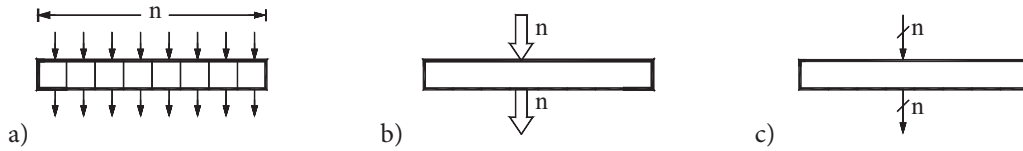
Diğer taraftan, *okuma terimi* kullanıldığı zaman (İng. Read), bellek sözcüğünün (var olan depolanmış verinin) bellekten bir dış bileşene ya da cihaza çıkarılması düşünülüyor. Bununla aslında, verilen bellek konumunda yerleşmiş içerik okunuyor. İki okuma şekli vardır: yıkıcı ve yıkıcı olmayan. *Yıkıcı (bozucu) okuma* sırasında bellek konumun içeriği kayboluyor (siliniyor), *yıkıcı olmayan okuma* sırasında ise bellek konumun içeriği okunmadan önce olduğu gibi kalıyor. Pratikte çok daha sıkça yıkıcı olmayan okuma şekli uygulanıyor ve okuma terimi kullanıldığı zaman yıkıcı olmadığı düşünülüyor.

Buna göre, bellek konumun içeriği sadece  $n$  bitlik sabit uzunluğunda tek bir bütün olarak okunabilir ya da onda aynı uzunlukta yeni içerik yazılabilir.

Gerçek bellek tümleşik devreler, genelde  $R/\overline{WE}$ ,  $\overline{WE}$  ya da  $\overline{WE}$  ile işaretlenen ayrı giriş kontrol hattına sahiptir. Bu hatta 1 getirilirse ( $R/\overline{WE} = 1$ ) bellekten okunabilir demektir, bu hattın durumu 0 ise ( $R/\overline{WE} = 0$ ), bellek bileşeninde yazılabilir anlamına geliyor.

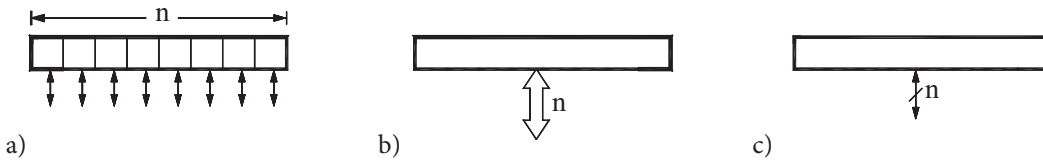
Yeni içeriğin en hızlı olarak paralel şekilde, tüm bellek hücrelerine aynı anda ulaşarak okunabildiğini veya değişebileceğini göz önüne alarak, veriden her bit için birer ayrı veri hattı kullanılması gerektiği sonucuna varılıyor, çünkü verilen anda bir hattan sadece bir bit aktarılabilir. Şöyle ki, bu hatta bulunan gerilim seviyesi alçak olabilir,  $V_L = V_{LOW} = V(0)$  ve bu durum mantıksal 0'a uygundur, ya da yüksek olabilir  $V_H = V_{HIGH} = V(1)$  ve bu durum mantıksal 1'e uygundur.

Buna göre, verilen bellek konumunda yeni verinin (içeriğin) yazdırılması belirli hatlardan, okuma ise başka hatlardan gerçekleşiyorsa, sembolik olarak bunu Şek. 7-5'e göre tanımlayabiliriz. Şekillerde paralel veri hatların (Şek. 7-5 a) aynı amaçları var ve bu yüzden daha basit olarak çift geniş çizgiyle (Şek. 7-5 b) ya da daha kalın çizilmiş çizgiyle (Şek. 7-5 c) işaretlenerek yanında toplam hatlar (teller) sayısı yazılıyor. Böyle gösterimle mantıksal diyagramların çizilmesi basitleştiriliyor ve aynı zamanda onların görünürlüğü artıyor.



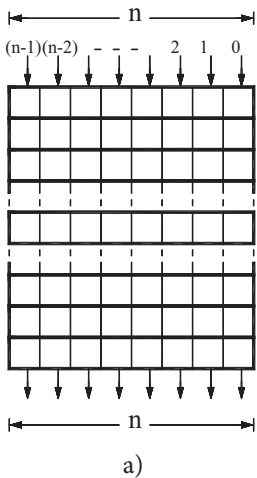
Şek. 7-5. Ayrı okuma ve yazma hatlı bellek konumlarının sembolleri

Aynı anda hem okuma hem yazma mümkün olmadığını göz önüne alarak, verilen anda bir bellek konumuna erişerek içeriğin okunduğundan ya da yeni içeriğin yazıldığından dolayı, biri giriş için diğeri çıkış için olmak üzere iki hatlar kümesinin kullanılması yerine, sadece bir hatlar kümesi kullanılabilir. Bu hatlar okunduğu zaman ( $R/\overline{W}=1$ ) çıkış hatları olacak, yazıldığı zaman dae ( $R/\overline{W}=0$ ) giriş hatları olacak. Bu sembolik olarak Şek. 7-6 a), b) ve c) ile gösterilmiştir. Pratik uygulamalarda gerçek bellek bileşenlerinde veri hatları genelde iki yönlüdür.

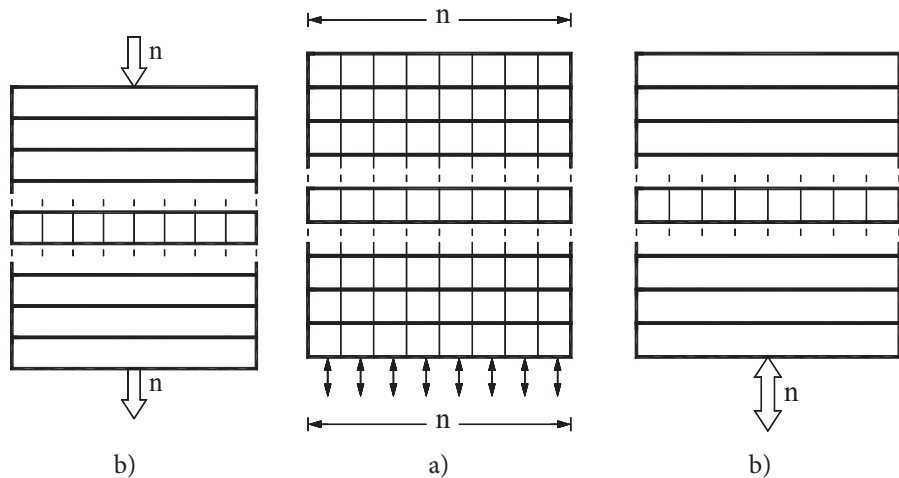


Şek. 7-6. İki yönlü okuma/yazma hatlı bellek konumunun sembolleri

Verilen anda sadece bir bellek konumuna erişilebildiğinden dolayı, bir bellek bileşenin sahip olduğu veri hatların sayısı, Şek. 7-7 a) b) ve Şek. 7-8 a) b)'ye göre bu konumda yerleşmiş olan bellek sözcüğünün uzunluğuna eşittir. Bellek ve başka bir bileşen arasında iletişimin kurulduğu anda, bellek devresinde tek veri hatlarında sadece belirtilmiş konuma ilişkin bitler meydana gelecek ya da düzenlenecek. Böyle yapı ve bağlanma şekli veri hatlarının sayısının sadece bir bellek sözcüğünün uzunluğuyla, yani bir bellek konumunda bellek hücrelerin sayısı ile (n) eşit olmasını sağlıyor.



Şek. 7-7. Ayrı okuma ve yazma hatlı bellek bileşenlerinin sembolleri

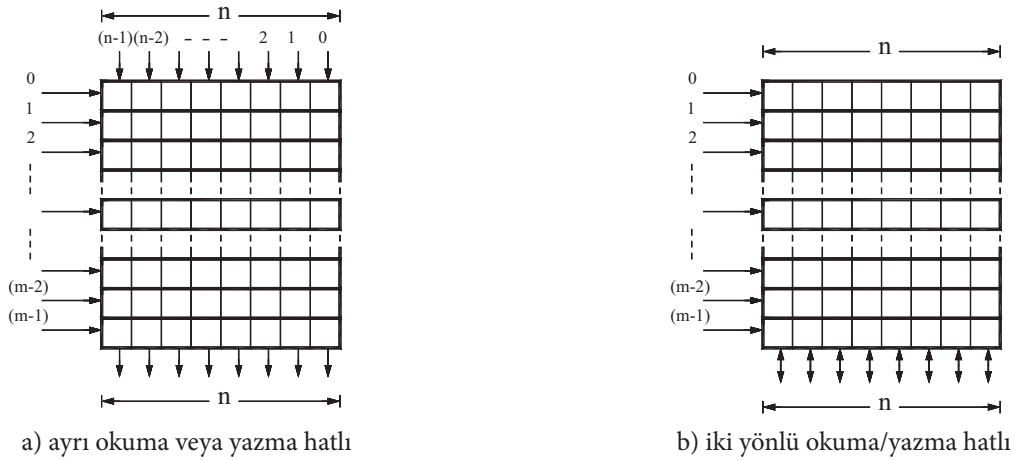


Şek. 7-8. İki yönlü okuma/yazma hatlı bellek bileşenlerinin sembolleri

Gerçek bellek tümleşik devrelerin çok büyük sayıda konumları vardır. Verilen bellek bileşenin sahip olduğu toplam bellek konumlarının sayısı m ile işaretleniyor ve bu arada m, 1'den çok daha büyüktür ( $m \gg 1$ ) ve söz konusu olan dijital sisteme: mikroişlemci, mikrodenetleyici, kişisel bilgisayar veya başka bir daha büyük sistem olmasına bağlı olarak genelde  $2^{10}=1024=1K \approx 10^3$  ile  $2^{20}=1024 \times 1024=1M \approx 10^6$ ,  $2^{30}=1024 \times 1024 \times 1024=1G \approx 10^9$  aralığındadır.

Okuma ve yazma işlemlerinin herhangi bellek yeriyle ilişkin olabildiğinden dolayı, belleği birinci konumdan son konuma kadar erişim ve arama olanağı olmalıdır. Herhangi bir bellek konumuna erişimin etkili gerçekleşmesi için, her bellek sözcüğü özel sayı ile numaralandırılıyor (indisleniyor) ve bu sözcüğün benzersiz verilen *adresidir*. Bu adres verilen kelimenin yazıldığı ya da okunduğu bellek yerinin belirlenmesi için kullanılacaktır. Her bellek sözcüğünün ona benzersiz şekilde bağlanmış olduğu özel adresi ve bellek yeri olmasından dolayı, okuma veya yazma işlemi sırasında erişebilen en küçük veri, belirtilmiş bellek yerinde bulunan bellek sözcüğüdür. Bununla ilgili olarak, verilen okuma veya yazma işleminin gerçekleşmesi için belirli bellek konumuna erişildiği zaman, *adresleme* veya *belirtme* terimi kullanılıyor. Bu şekilde, adreslerin kullanımıyla, bellek sözcüklerin dış bileşenlerden belleğe ve ters yönde hareket etmeleri sağlanabilir. Böylece her veri, adresi verilmiş konumdan okunabilir ya da orada yazılabilir.

Söylediklerimizden, bellek devresinin, tüm bellek yerlerinin toplam sayısından ( $m$ ) herhangi bir bellek yerinin adreslenmesi (seçilmesi) gerçekleşebildiği özel adres hatlarının olmasını gerekli olduğu sonucu ortaya çıkıyor. Buna göre adreslerin sayısı bellek sözcüklerin toplam sayısına eşit olmalıdır. Bu şekilde, diğerleri pasif olurken belirli bir adres hattının aktifleştirilmesiyle, adreslenmiş bellek konumuna erişiliyor, veya içeriği (sözcük, orada korunmuş veri) okunabilecek, diğer bellek yerleri ise pasif kalacak. O anda,  $n$  veri hattında okuma gerektiğinde gösterilen konumun içeriği meydana geliyor, ya da bu yerde yeni içeriğin yazılması gerektiğinde veri hatlarına yeni veri yerleşiyor. Genel olarak, birinci bellek sözcüğün sıfırın adresi vardır, yani birinci bellek yerinin adresi 0 ile işaretleniyor ( $m=0$ ), son sözcüğün, yani son bellek yerinin ise en yüksek adresi, Şek. 7-9 a) veya b)'ye göre,  $(m-1)$  değeri olacaktır.



Şek. 7-9. Bellek matrisinde adreslerin belirlenmesi

Bellek yerlerin sayısı ( $m$ ) çok büyük olduğundan dolayı, belleğin dışardan adreslendiği adres hatlarının da sayısı büyük olur. Bu yüzden, bellek konumların adreslenmesi (tüm adres alanına, bellek matrisine erişim), bellek bileşenin içeriğinde giren adres kod çözücü yardımıyla yapılıyor.

Adres kod çözücünün kullanımı dış adres hatlarının sayısını büyük ölçüde, daha doğrusu logaritmik bağlantıyla azaltıyor, çünkü kod çözücünün çıkışında  $m$  hatın elde edilmesi için, girişinde sadece  $k$  hat gerekecek ve bu arada  $k=\log_2 m$ , yani  $m=2^k$  koşulu yerine getirilmelidir.

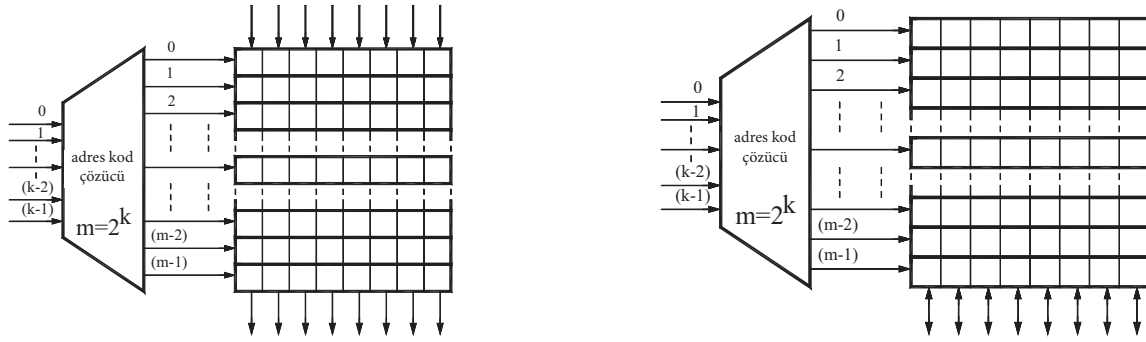
Son sonuçla ilgili verilen bellek bileşenin sahip olduğu bellek konumlarının toplam sayısı  $m$ , 2 sayısının tam üssü olarak elde ediliyor, öyle ki şu denklem her zaman geçerli olacak:

$$m = 2^k \quad (7-1)$$

$k$ , 1'den büyük tam sayıdır ( $k \gg 1$ ), ve genelde 10 ile 30 arasındadır.

Adres kod çözücünün kullanıldığı yüzünden, adres bilgisi, kodun çözümlenmesi ardından tam olarak belirlenmiş bellek konumunu etkinleştiren bitler dizisi olacak (ikili vektör). Daha doğrusu, onlu yazılımda adresi, adres hatlarında getirilen ikili kodlanmış adrese uygun olan bellek sözcüğü etkinleştirecek. Bellek sözcüklerin toplam sayısının  $m = 2^k$  olduğunu ve birinci bellek sözcüğün sıfır adresi, ( $m=0$ ) olduğunu göz önüne alarak, son sözcüğün en yüksek adresi olarak ( $m-1$ )'e uyan ( $2^k-1$ ) değeri olacak.

Adresin tüm bitleri kod çözücünün girişinde aynı anda (paralel) götürüldüğünden dolayı,  $k$  adres hattan oluşan grup (küme) gerekecek. Buna göre her bellek konumun adresi, Şek.7-10 a) ve b)'ye göre, genişliği  $a_{(k-1)}a_{(k-2)}\dots a_2a_1a_0$  adres hatların sayısına eşit  $k$ -bitli vektör (ikili sayı) tanımlayacak.



a) okuma ve yazma için özel hatlı

b) iki yönlü okuma/yazma hatlı

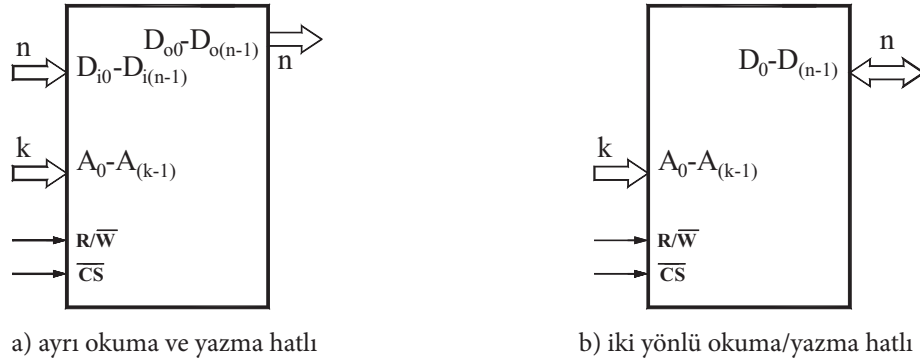
Şek.7-10. Bellek konumların adres kod çözücü ile bir boyutlu adresleme

Adresin kod çözümlenmesi, en sol pozisyonda bulunan bitin en büyük ağırlığı  $a^{(k-1)}$ , en sağda bulunan bitin ise en düşük ağırlığı  $2^0=1$  olduğu doğal ikili sayı sistemine göre yapılıyor. Bellek alanına erişiminde adresleme için tek adres kod çözücünü kullanıldığı adresleme *doğrusal* ya da *bir boyutlu* adresleme tanımlıyor. Pratikte en sıkça olarak iki boyutlu adreslemeye rastlanıyor. Bu adresleme şeklinde bellek matrisine iki adres kod çözücüyle ulaşıyor: bir kod çözücü satırlar için ve bir kod çözücü sütunlar için.

Veri hatları ( $d_{(n-1)}d_{(n-2)}\dots d_2d_1d_0$ ) ve adres hatları ( $a_{(k-1)}a_{(k-2)}\dots a_2a_1a_0$ ) ile okuma/yazma kontrol hatları ( $R/\overline{W}$  ya da  $\overline{WE}$ ) dışında, bellek bileşenlerin, Chip Select veya Memory Enable İngilizce terimlerine göre genelde  $\overline{CS}$  ya da  $\overline{ME}$  ile işaretlenen çok önemli kontrol girişleri de vardır. Bu girişte bellek bileşenin (bellek yongası, bellek tümleşik devrenin) seçilmesi için ve çalışma olanağı veren sinyal geliyor. Bu sinyal  $\overline{CS}$  (ya da  $\overline{ME}$ ) alçak seviyede aktiftir. Buna göre bu sinyalin bellek tümleşik devrelerin üzerine efektif etkisi, sadece  $\overline{CS} = 0$  (ya da  $\overline{ME}=0$ ) koşulu yerine getirilmiş durumda vardır. Sadece bu durumda bellek yongası seçilmiş oluyor ve bellek fonksiyona koyulmuştur: bellek adreslenip okuma ve yazma işlemleri gerçekleşebilir. Ancak, bu girişe yüksek mantıksal seviye getirilirse  $\overline{CS}$  (ya da  $\overline{ME}$ ), bellek bileşeni pasif olacak (hiçbir şey çalışmayacak) çünkü onun tüm hatları yüksek dirençlik durumuna giderek, bellek bulunduğu dijital sisteminden aslında devre dışı kalacak.



Sonuç olarak, Şek. 7-11 a) ve b)'de tüm giriş ve çıkış veri, adres ve kontrol hatlarıyla bellek bileşenlerin sembolik işaretleri verilmiştir.



Şek. 7-11. Bellek bileşenlerin sembolik işaretleri

### **Bellek kapasitesi**

*Bellek kapasitesi* verilen bellek bileşenin sahip olduğu toplam bit(b) ya da bayt (B) sayısını belirtiyor ve onu  $w$  ile işaretleyeceğiz. Kapasite bellek konumların (sözcüklerin) toplam sayısını uzunluklarıyla çarparak, aşağıdaki denklemlerle kolayca elde edilebilir:

$$w = m \times n \quad (7-2)$$

Kapasite genelde bitlerden (b) veya baytlardan (B) çok daha büyük birimlerle ifade ediliyor, çünkü pratikte uygulanan gerçek bellek bileşenlerin büyük kapasiteleri vardır. Bununla ilgili olarak, en sıkça kilo (K), mega (M) ve giga (G) gibi birimlerle karşılaşacağız:

$$\Rightarrow 1K = 2^{10} = 1024 \approx 10^3,$$

$$\Rightarrow 1M = 2^{20} = 2^{10} \times 2^{10} = 1024 \times 1024 \approx 10^6,$$

$$\Rightarrow 1G = 2^{30} = 2^{10} \times 2^{10} \times 2^{10} = 1024 \times 1024 \times 1024 \approx 10^9,$$

(7-1) denklemini, bellek kapasitesi denkleminde (7-2) uygulayarak, bellek kapasitesini belirleyen çok daha sıkça kullanılan denklem elde ediyor:

$$w = 2^k \times n \quad (7-3)$$

### **Çözülmüş örnekler:**

Bu bölüme kadar sunduğumuz ve incelediğimiz konuları daha kolay anlamak için, devamda verilmiş değerlerle birkaç örnek çözeceğiz.

**Örnek 7-1:**  $m=16$  sözcüğe sahip olan küçük kapasiteli bir bellek bileşenin içeriği tablo ya da matris şeklinde gösterilmiştir (Tab.7-1). Matriste sözcükler satırlar olarak verilmiştir ve bu arada her sözcük sütunları tanımlayan sonlu sayıda bitlerden oluşuyor ( $n=8$ ). Bellek devrenin sözcükleri 8 bit, yani 1 bayt (B) uzunluğundadır ( $n=8$ ). Böylece bileşenin kapasitesi (7-1) denklemine göre bitlerle ifade edilirse  $w=16$  sözcük  $\times$  8b/sözcük = 128b değeri elde ediliyor, baytlarla ifade edilirse  $w=16$  sözcük  $\times$  1B = 16B değeri elde ediliyor.

Bellek konumun adresi			Bellek konumun içeriği			
İkili	Onlu	On altılı.	İkili	Onlu	On altılı	ASCII sembol
0000	0	1	01100001	97	61	a
0001	1	2	01100010	98	62	b
0010	2	3	01100011	99	63	c
0011	3	4	01100100	100	64	d
0100	4	5	01100101	101	65	e
0101	5	6	01100110	102	66	f
0110	6	6	01100111	103	67	g
0111	7	7	01101000	104	68	h
1000	8	8	01101001	105	69	i
1001	9	9	01101010	106	6A	j
1010	10	A	01101011	107	6B	k
1011	11	B	01101100	108	6C	l
1100	12	C	01101101	109	6D	m
1101	13	D	01101110	110	6E	n
1110	14	E	01101111	111	6F	o
1111	15	F	01110000	112	70	p

Tab. 7-1. 1 baytlık 16 sözcüklü bellek devresinin içeriği

Bellekte tüm bilgiler, tablodaki adres ve verilerin ilk iki sütunda işaretlenmiş olduğu gibi ikili şekilde korunuyor. Ancak, böyle işaretler sadece makine (bilgisayar) tarafından anlaşılır, insan için ise uzunlukları ve ikili taban 2 yüzünden sorun yaratıyorlar. Bu yüzden bellekteki sözcüklerin adresleri ve içerikleri onlu şekilde ve aynı zamanda on altılı gösterimde verilmiştir. Onlu işaretleme insanın anlayış şekline çok daha yakındır, ancak on altılı işaretlemeye uzun ikili diziler (vektörler) en basit ve en kompakt şekilde gösteriliyor: her dördü bit (her nibıl) tek ve uygun on altılı rakamla değiştiriliyor. Tab.7-1'den bellekte İngilizce alfabesinden ilk 16 küçük harfin ASCII kodların da girilmiş olduğu görülebilir.

**Örnek 7-2:** Tab.7-2 tablosu bellek bileşenin ilk dört bellek konumun içeriğini gösteriyor. Bu bellek bileşeni önceki örnekteki bellek bileşenine kıyasen daha çok sayıda sözcüklere sahiptir ve tablodan görüldüğü gibi kapasitesi  $m=1024_{DEC}$  bellek sözcüğüdür, ya da 1K sözcüktür. Bu sözcükler bellek alanını tanımlayan birincisinden ( $a=0_{DEC}$ ) sonuncusuna kadar ( $1023_{DEC}$ ) adreslerle işaretlenmiştir. Ayrıca şekilden her satırın 8'er alanı (bellek hücresinin) olduğu görülüyor ve her alanda 1'er bitin yerleşebileceğine göre, sonuç olarak her sözcüğün  $n=8$  bit veya 1 bayt (1B) uzunlukta olduğunu söyleyebiliriz.

Ayrıca, verilen tablodan 0-nci konumun içeriği  $d_7d_6d_5d_4d_3d_2d_1d_0=01000001$  olduğu, 1-nci konumun içeriği  $01000010_{(2)}$ , 2-nci konumun içeriği  $01000011$  olduğunu, 3-ncü konumda ise  $01000100$  içeriğin yerleşmiş olduğu görülüyor. Standart ASCII kodun tablosunu önümüze alırsak, ilk dört konumda depolanmış verileri kolayca tanıyabiliriz. ASCII tablosuna göre, sıfırıncı konumda A harfinin kodu yerleşiktir, birinci konumda B'nin kodu, ikinci konumda C'nin kodu ve üçüncü konumda D'nin kodu bulunuyor.

Bellek konumun adresi		Bellek konumun içeriği					
İndis		k =10 bit	ASCII sembol			n=8 bit	
Hex	Ai(dec)	A <sub>9</sub> A <sub>8</sub> ... A <sub>2</sub> A <sub>1</sub> A <sub>0</sub>	Ch	HEX	DEC	D <sub>7</sub> D <sub>6</sub> D <sub>5</sub> D <sub>4</sub> D <sub>3</sub> D <sub>2</sub> D <sub>1</sub> D <sub>0</sub>	
000	0000	0 0 0 0 0 0 0 0 0 0	A	41	65	0 1 0 0 0 0 0 1	
001	0001	0 0 0 0 0 0 0 0 0 1	B	42	66	0 1 0 0 0 0 1 0	
002	0002	0 0 0 0 0 0 0 0 1 0	C	43	67	0 1 0 0 0 0 1 1	
003	0003	0 0 0 0 0 0 0 0 1 1	D	44	68	0 1 0 0 0 1 0 0	
004	0004	0 0 0 0 0 0 0 1 0 0	E	45	69	0 1 0 0 0 1 0 1	
005	0005	0 0 0 0 0 0 0 1 0 1	F	46	70	0 1 0 0 0 1 1 0	
006	0006	0 0 0 0 0 0 0 1 1 0	G	47	71	0 1 0 0 0 1 1 1	
007	0007	0 0 0 0 0 0 0 1 1 1	H	48	72	0 1 0 0 1 0 0 0	
008	0008	0 0 0 0 0 0 1 0 0 0	I	49	73	0 1 0 0 1 0 0 1	
009	0009	0 0 0 0 0 0 1 0 0 1	J	4A	74	0 1 0 0 1 0 1 0	
00A	0010	0 0 0 0 0 0 1 0 1 0	K	4B	75	0 1 0 0 1 0 1 1	
...	...	...	...	...	...	...	
...	...	...	...	...	...	...	
3FE	1022	1 1 1 1 1 1 1 1 1 0	b	62	98	0 1 1 0 0 0 1 0	
3FF	1023	1 1 1 1 1 1 1 1 1 1	a	61	97	0 1 1 0 0 0 0 1	

Tab. 7-2. 1 baytlık 1K sözcüklü bellek devresinin içeriği

Bellek sözcüklerin toplam sayısını ve her bellek sözcüğünün bitlerle ifade edilmiş uzunluğunu bilerek, verilen bellek bileşenin (7-1) denklemine göre  $w$  kapasitesini belirleyebiliriz. Baytlarla (B) ve bitlerle (b) ifade edilen  $w$  bellek kapasitesinin hesaplanması şöyle yapılır:

$$w = m \times n = 1024 \text{ sözcük} \times 8 \text{ b/sözcük} = 2^{10} \times 1 \text{ B} = 1\text{K} \times 1\text{B}, \text{ ya da } 1\text{KB veya } 1024 \text{ B},$$

$$w = 1\text{K} \times 8\text{b}, \text{ ya da } 8 \text{ Kb veya } 8096 \text{ b}.$$

Tab. 7-2 gösterilen belleği, okuma ve yazma işlemlerin ifade edilmesi için de kullanabiliriz. Şöyle ki, bilgisayarı iki yönergenin gerçekleştirilmesi gerektiğini alalım: önce adresi 3 olan bellek konumunu okuyor ve ardından okunulan içeriği 0 adresli başlangıç konuma yazdırıyor.

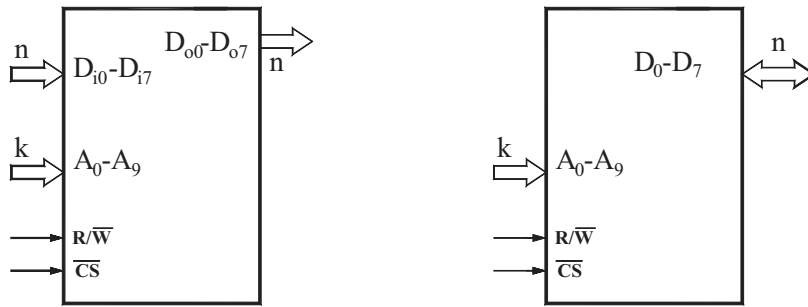
Okumanın, varsayılan yıkıcı olmadığını tahmin edersek, o zaman programın tamamlanmasından sonra 0 adresli bellek konumunun içeriği 3 adresli konumun içeriğiyle aynı olacak ve her iki konumda  $01000100_{(2)}$  sözcüğü yerleşmiş olacak, yani her iki konumda D sembolünün ASCII kodu bulunacak. Bellekten  $01000001_{(2)}$ , ya da Şek.7-3'e göre sıfırıncı adresli birinci bellek sözcüğünde yerleşmiş olan A harfinin ASCII kodu geri dönmez şekilde kaybolacak.

Hex	Ai(dec)	A <sub>9</sub> A <sub>8</sub> ... A <sub>2</sub> A <sub>1</sub> A <sub>0</sub>	Ch	HEX	DEC	D <sub>7</sub>	D <sub>6</sub>	D <sub>5</sub>	D <sub>4</sub>	D <sub>3</sub>	D <sub>2</sub>	D <sub>1</sub>	D <sub>0</sub>
000	0000	0000000000	D	44	68	0	1	0	0	0	1	0	0
001	0001	0000000001	B	42	66	0	1	0	0	0	0	1	0
002	0002	0000000010	C	43	67	0	1	0	0	0	0	1	1
003	0003	0000000011	D	44	68	0	1	0	0	0	1	0	0

Tab. 7-3. 1'er baytlık 1K sözcüklü bellek devrenin ilk dört konumun içeriği

**Örnek 7-3:** Şek.7-12 a) ve b)'de bir bellek bileşenin blok-diyagram-modeli gösterilmiştir. Bu bellek bileşenin de önceki bileşen gibi on adres hattı ( $k=10$ ) ve sekiz veri hattı ( $n=8$ ) vardır. Önceki analizi ve (7-1) denklemini göz önüne alarak, bellek tümleşik devrenin 8 bit uzunluğunda ( $n=8b=1B$ ) 1K bellek konumu vardır ( $k=10 \Rightarrow m=2^{10}=1024=1K$ ). Buna göre bellek bileşenin iç organizasyonu  $n=8$ 'er bitli  $m=1024$  sözcüktür. (7-2) ve (7-3) denklemine göre belleğin kapasitesi  $w=1024$  sözcük  $\times 8$  bit =  $2^{10} \times 1$  bayt = 1KB olduğu elde ediliyor.

Şek. 7-8 a)'da verilmiş olan birinci tümleşik devrenin iki veri hatlar grubu var: giriş ve çıkış. Giriş hatlarda bellekte yazılması gereken veri yerleşiyor, çıkış hatlarında ise bellekten okunan veri meydana geliyor. Şek.7-8 b)'de verilen ikinci devrede iki yönlü olan sadece bir veri hatlar kümesi vardır: bellekte veri yazılınca bu hatlar giriş hatlarıdır, okunduğu zaman ise çıkış hatlarıdır.



a) ayrı okuma ve yazma hatlı

b) iki yönlü okuma/yazma hatlı

Şek. 7-12. 1K sözcük x 1 bayt olarak organize edilmiş bellek bileşenin sembolleri

**Örnek 7-4:** Bellekte yazma. „J” sembolünün (ASCII koduna göre büyük J latin harfi)  $5_{DEC}$  adresli bellek konumunda yazılması için verilen bellek devrenin tüm hatların mantıksal durumları belirlensin.

1. Yonga seçimi:  $\overline{CS} = 0$ ;
2. Bellek konumun adresinin seçilmesi:  $5_{DEC} = a_{BIN} = 0000000101$ . Bu adres bilgisi adres hatlarına yerleşiyor ( $A_9$ 'dan  $A_0$ 'a kadar);
3. Yazma işleminin seçilmesi:  $R / \overline{W} = 0$ ;
4. „J” =  $d_{BIN} = 01001010$  verinin girilmesi. Bu veri, veri hatlarına yerleşiyor ( $D_7$ 'den  $D_0$ 'a kadar).

**Örnek 7-5:** Okuma.  $6_{DEC}$  adresli bellek konumun içeriğinin okunması için verilen bellek devresinin tüm hatlarının mantıksal durumları belirlensin.

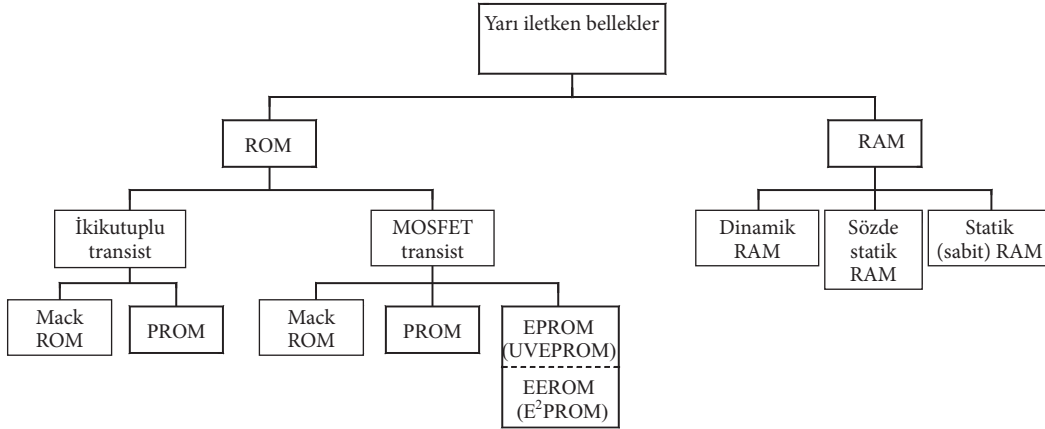
1. Yonga seçimi:  $\overline{CS} = 1$ ;
2. Bellek konumun adresinin seçilmesi:  $6_{DEC} = a_{BIN} = 0000001100$ . Bu adres bilgisi adres hatlarına yerleşiyor ( $A_9$ 'dan  $A_0$ 'a kadar);
3. Yazma işleminin seçilmesi:  $R / \overline{W} = 1$ ;
4. „G” =  $d_{BIN} = 01000111$  verinin çıkarılması. Bu veri, veri hatlarına yerleşiyor ( $D_7$ 'den  $D_0$ 'a kadar).

Bellek yongaların büyük kısmı iki kontrol hatla üretiliyor: biri,  $\overline{WE}$ , yazma olanağı veriyor, ikincisi  $\overline{OE}$  ise, okumayı sağlıyor. Bu arada onların mantıksal durumu birbirine göre tümleyen olmalıdır:  $\overline{WE} = 1$  ve  $\overline{OE} = 0$  olduğu zaman okunuyor,  $\overline{WE} = 0$  ve  $\overline{OE} = 1$  koşulu geçerliyse yazılıyor.

## 7.4. BELLEK BİLEŞENLERİN AYRIMI

İlerdeki bölümde dikkatimizi yarı iletken belleklerin ayırımına yönlendireceğiz, çünkü bu konu çerçevesinde bu bellekler inceleme konusu olacak.

Belleğin beslemenin kesilmesinden sonra içeriğini kaybedip kaybetmediğine göre, bellek bileşenleri Şek. 7-13'te gösterildiği gibi iki büyük gruba ayrılıyor: uçucu (İng. Volatile) ve uçucu olmayan bellekler (İng. Non/Volatile memories)



Şek. 7-13 Yarı iletken bellek bileşenlerinin ayrımı

*Uçucu (geçici) olmayan belleklerde* girilen veriler beslemenin kesilmesinden sonra kaybolmuyor. Bu grubun en tipik ve en çok bilinen temsilcisi ROM belleğidir, ya da sadece okunabilen bellektir (İng. Read Only Memory). ROM verilerin bellek yapısının iç bağlantılarının programlanmasıyla yazılan tümleşik devre olarak yapılıyor. ROM bellek bileşenlerinde içerik sadece bir kez, tümleşik devrenin üretim sürecinde girilir. ROM tümleşik mantıksal devrelerin sadece bir kontrol sinyali var, o da okuma olanağı ( $\overline{OE}$ ), sinyalidir, çünkü onlarda yeni içeriğin yazılması mümkün değildir.

Programlanabilir ROM belleği, ya da kısaca PROM, kullanıcıların PROM programlayıcı olarak adlandırılan özel bir cihazın yardımıyla bağlantıları kendi başına programlayabildiği ve istediği içerikleri yerleştirdiği ROM bileşenidir. Programlandıktan sonra, PROM ROM gibi davranıyor çünkü içeriği ondan sonra artık değişmez.

Özel PROM bellekler kategorisi silinebilir PROM tümleşik devreleridir (İng.erasable programmable ROM) ya da kısaca EPROM-lar. Onlar PROM-bellekler gibi programlanabilir, ancak onun dışında morötesi ışığın (UV) etkisi altında içerikleri silinebilir. Böylece bağlantıların yeniden programlanması yoluyla her bellek konumuna ayrıdan yeni içeriklerin girme olanağı vardır.

EEPROM, PROM belleğin başka bir türüdür, çünkü onun içeriği silinebilir ve yeniden programlanabilir, ancak silme, EPROM bileşenlerinde gibi mor ötesi (UV) ışığıyla değil, elektrik yoluyla, daha kuvvetli elektrik akımının salınmasıyla yapılıyor. Çakar (fleş) bellek tümleşik devreler (İng. Flash) özel EEPROM bellek türüdür. Onlarda yazma klasik EEPROM'da olduğu gibi birer birer bayt şeklinde değil, bloklarla gerçekleşiyor.

Tab.7-4'te farklı ROM bellek türlerinin temel parametreleri verilmiştir.

Tür	Teknoloji	Okuma döngüsü	Yazma döngüsü	Yorum
Mack ROM	NMOS, CMOS	20-200 ns	4 hafta	Bir kez yazılabilir, düşük güç;
Mack ROM	İki kutuplu	<100 ns	4 hafta	Bir kez yazılabilir: yüksek güç; yüksek yoğunluk;
PROM	İki kutuplu	<100 ns	5 dakika	Bir kez yazılabilir: yüksek güç; maskesiz;
EPROM	NMOS, CMOS	25-200 ns	5 dakika	Fazla kez yazılabilir; düşük güç; maskesiz;
EEPROM	NMOS	50-200 ns	10 µs/bayt	10,000 yazma/konum;
FLASH	CMOS	25-200 ns	10 µs/blok	100,000 silme döngüsü;

Tab. 7-4. Farklı ROM bellek türlerin kıyaslama özellikleri

*Uçucu (geçerli) belleklerde*, elektrikle beslemenin kesilmesinden sonra içerikleri kayboluyor, ancak besleme olduğu sürece, içerikleri her zaman değişebilir, girilen veriler okunabilir, varolan veriler silinebilir ve yenileri yazılabilir. Bu yüzden onlar için Read/Write Memory İngilizce teriminden RWM kısaltması kullanılıyor ve okunabilir ve yazılabilir bellek anlamına geliyor. Ancak, bu bellek bileşenleri için pratikte geniş çapta yayılmış ve hemen her yerde çok daha iyi bilinen RAM kısaltmasının kullanılması kabul edilmiştir. Bu kısaltma İngilizce Random-Access Memory ifadesinden geliyor ve rastgele erişimli bellek demektir. Bu ifade, herhangi bir bitin okunması için gereken zamanın bitin bellekteki pozisyonuna bağlı olmamasıyla ilgilidir, daha doğrusu bellekte herhangi bir veriye erişim zamanı eşittir (düzgündür).

Statik RAM bellekler (SRAM) özel RAM bileşenler grubudur. Onlarda veriler, mandal flip-floplarla (kilitleme devreleriyle) gerçekleşen bellek hücrelerinde giriliyor ve korunuyor. Bu veriler onlarda bellek tümleşik devrenin beslemeye bağlı olduğu sürece korunabilir, okunabilir ya da değişebilir. Beslemenin kestirilmesiyle veriler geçici olarak kayboluyor. SRAM bellek hücreleri en büyük çalışma hızı olan önbellek (saklı) belleğin yapımında kullanılıyor.

Dinamik RAM (DRAM) bellekler, başka bir RAM bileşen grubudur, çünkü onlarda veriler çok küçük kondansatörde (sığaçta) elektrik yükü olarak korunuyorlar. Bu yüzden zaman geçtikçe boşalıyorlar ve sıkça periyodik olarak yükün yenilemesine (yinelemesine, İng.refresh) gereksinimi vardır. Yükün yenilenmesiyle depolanan bilgilerin korunması sağlanıyor. Asenkron DRAM dışında, son zamanlarda kişisel bilgisayarlarda senkron DRAM (SDRAM) giderek fazla kullanılıyor. SDRAM'da iletme özel pals sinyaliyle, pasın bir kenarın meydana gelmesiyle eşzamanlı (senkron) olarak yapılıyor. Bu türden özellikle hızlı bellek devreleri iyi bilinen DDR (Double Data Rate) SDRAM tümleşik devreleridir. Onlar veri iletiminde iki kat daha büyük hızla çalışıyorlar, çünkü pals sinyalinin hem ön hem arka kenarında veriler iletiliyorlar. DRAM bellek tümleşik devreler yardımıyla ana (işletim, birincil) bellek gerçekleşiyor. Ana belleğin önbellekten daha büyük kapasitesi var, ancak çalışma hızı açısından daha yavaştır.

Sözde (pseudo) statik RAM belleği (PSRAM), aslında SRAM gibi davranan ve yenileme için tümleşik devreli DRAM belleğidir.



## 7.5. ROM BELLEK BİLEŞENLERİ

**ROM:** ROM, verilerin yazılması belleğin yapısındaki iç bağlantıların programlanmasıyla gerçekleştirildiği tümleşik devre olarak yapılıyor. Daha doğrusu, programlama sürecinde, içerde di-yotlar olarak bağlanmış ve iki kutuplu ya da MOS teknolojisiyle yapılmış transistörler aracılığıyla oluşan belirli iç bağlantılar kuruluyor ya da kesiliyor. Böyle ROM belleklerde içerik sadece bir kez, fabrika üretimi sürecinde, önceden belirlenmiş maskeye göre giriliyor ve bu yüzden mask-programlanabilir ROM bellekler olarak da biliniyor. Maskeyi, ROM'un üreticisi, kullanıcının içeriği hakkında aradığı gereksinimlere dayanarak üretiyor.

**PROM:** PROM belleğinin ROM belleğine kıyasen büyük avantajı var çünkü kullanıcı bağlantıları kendi başına PROM programlayıcı kullanarak programlıyor. Programlama sürecinde kurulan her bağlantı, içinden elektrik darbeler akarak kesilebilir. Bu işlem programlayıcı yardımıyla yapılarak mantıksal 0 giriyor ve bu şekilde kullanıcı aslında kendi isteği üzerine 1'leri ve 0'ları nerede yerleştireceğini seçiyor. Programlama tamamlandıktan sonra PROM, ROM belleği gibi davranıyor çünkü programlandıktan sonra içeriği değişmez.

**EPROM:** EPROM-lar da PROM-bellekleri gibi programlanabilir, ancak onların içeriği silinebilir ve yeni içeriğin girme olanağı vardır. EPROM'ların böyle çalışma şekli, yüzücü geçitli MOS transistörlerin kullanımıyla sağlanıyor, çünkü onların sayesinde potansiyeli değişebilir. EPROM-lar genelde 10 ile 100 kez arasında programlanabilir. EPROM içeriğinin silinmesi, mor ötesi ışığın etkisi altında gerçekleşiyor ve bu yüzden bu bellek devrelerine UV EPROM-lar da denir. EPROM devrelerin kasalarında şeffaf bir kısım var ve bu kısımdan devre içerisinde yonga görünüyor. Programlamadan sonra şeffaf kısım koruyucu folyo ile örtülüyor. Silme işleminden önce, yonganın mor ötesi ışıkların etkisi altında kalması için bu folyonun çıkarılması gerekiyor.

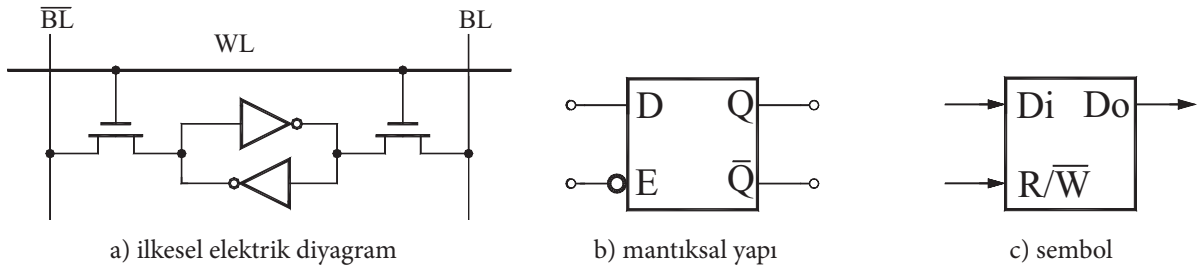
**EEPROM:** EEPROM (ya da E<sup>2</sup>PROM) PROM belleğin başka bir türevidir, ancak bu bellek bileşenlerinde içeriğin silinmesi UV ışınımının etkisiyle değil, kuvvetli elektrik akımının salınmasıyla yapılıyor. Silme ve yazma döngüleri yaklaşık on bin kez tekrarlanabilir. Flash belleği (FEPROM) EEPROM'un bir türevidir çünkü elektronik yoluyla silinebilir, ancak onlarda silme yaklaşık 100.000 kez tekrarlanabilir.

## 7.6. RAM

SRAM bellek kavramı anlamak için oldukça basittir, çünkü kilitleme flip floplardan oluşmuş belle hücreler matrisi, yani satırlara ve sütunlara organize edilmiş düzgün bellek hücreler kümesi söz konusudur. Onların önceliğinde okuma/yazma süreçlerin kontrol mantığı ve adres bilginin kod çözümü vardır. Pratikte senkron ve asenkron SRAM yongalar kullanılıyor. Asenkron SRAM, bilgisayarın çalışmasını senkronize eden palstan bağımsızdır. Asenkron SRAM belleğinde verilerin girişi ve çıkışı yonga ve okuma/yazma seçimi için kontrol sinyallerin meydana gelmesiyle yönetiliyor. Senkron SRAM-da adresin, giriş/çıkış verilerin ve diğer kontrol sinyallerin zamansal uyumlaşması pals sinyaline bağlıdır ve onun ön/arka kenarın meydana gelmesiyle başlatılıyorlar. Bu işlem, asenkron SRAM'ın arabirimine göre senkron SRAM'ın daha karmaşık arabirimi sayesinde sağlanıyor, çünkü senkron SRAM, çalışmasının senkronizasyonunu sağlayan, iç yazmaçlar içeriyor.

### 7.6.1. SRAM BELLEK HÜCRELERİ

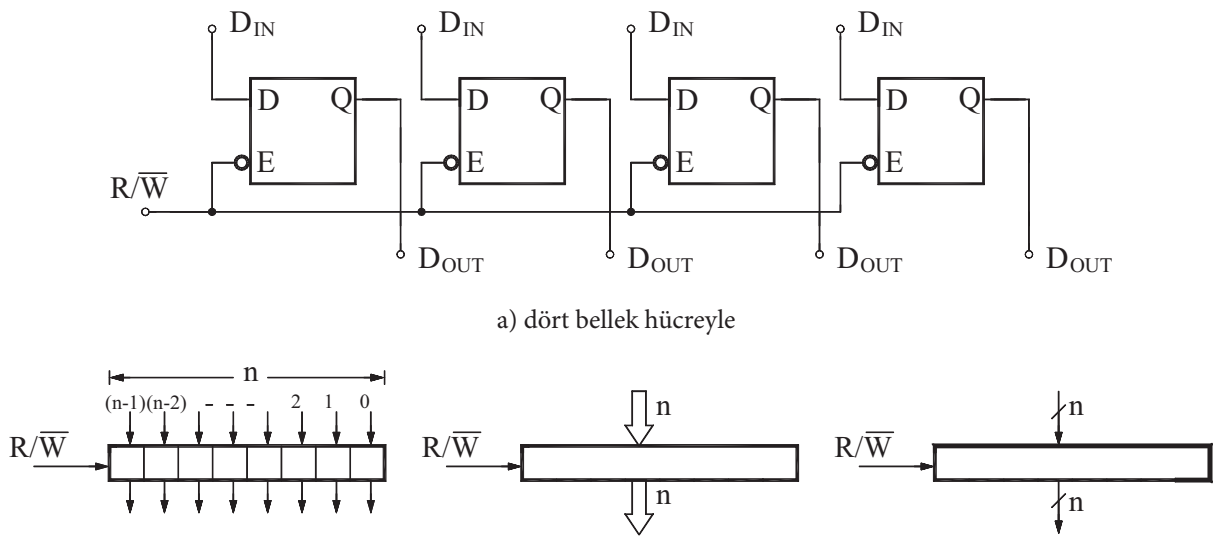
SRAM bellek bileşenlerin çalışmasını daha detaylı anlayabilmek için, önce onun Şek.7-14'te gösterilen tam yapı biriminin mantıksal yapısını inceleyeceğiz. Burada birbitli veri depolayabilen bellek hücresi söz konusudur. Aslında iki kutuplu multi vibratör kullanılıyor, yani D-türünden flip-flop için, SR türünden flip-flop ile de gerçekleştirilebilen mandal (kilitleme, tutma devresi) uygulanıyor. Pratikte bellek hücresi sıkça olarak altı MOSFET transistörle uygulanıyor: veriyi (biti) iki karşılıklı bağlı evirici (biri CMOS transistör çiftinden oluşuyorlar) ve ek olarak iki transistörün okunması-yazılması gerçekleştirildiği hatların bağlanması için kullanılıyor.



Şek. 7-14. SRAM bellek hücresi

Yeni içerik, yani yeni veri (yeni bit) D giriş hattı üzerinden yazılıyor, depolanmış içerik ise, yani hücrede korunan bit Q flip-flopun çıkışından okunuyor. Yazma sırasında E (izin) kontrol hattında alçak seviye getiriliyor ( $E=0$ ), yeni bit ise D girişine yerleşiyor. Okuma sırasında E kontrol hattında yüksek seviye getirilerek ( $E=1$ ), Q çıkışında flip-flopun mevcut durumu, yani onda bulunan bit meydana geliyor. Kilitleme devrenin tümleyen çıkışı  $\overline{Q}$  kullanılmıyor.

Bir bit korunabilen en küçük bilgi miktarı olduğundan dolayı, sıradaki Şek. 7-15 a)'da, 4 bit uzunluğunda bir bellek sözcüğünü elde etmemiz için dört bellek hücresinin hangi şekilde bağlayabileceğimizi göstereceğiz, Şek. 7-15 b) bu işlem sembolik olarak tanıtılmıştır. Tüm dört bitin bir bütün olarak alınması gerektiğini göz önüne alarak, okuma/yazma kontrol hattı tüm flip-flopların aynı girişinde aynı anda getiriliyor.

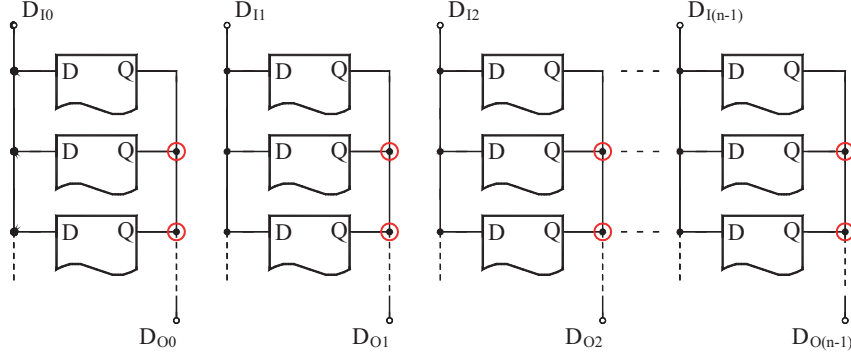


b) n-bellek hücre için sembolik gösterim

Şek. 7-15. Bir bellek sözcüğünün oluşması

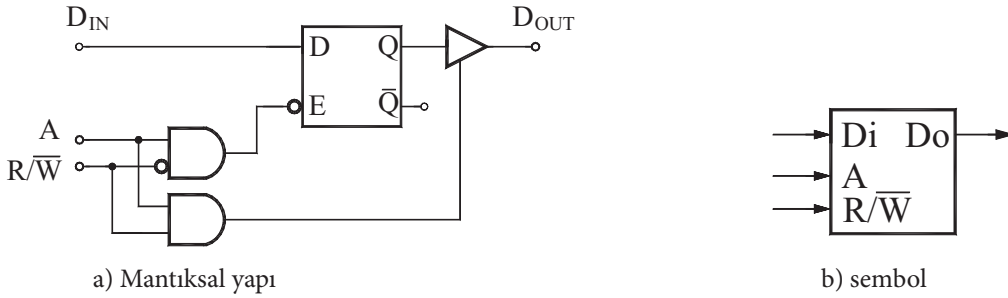


Ancak, belleğin daha büyük sayıda bellek hücresinden oluşması sırasında, sorun yaşanacak çünkü bu bellek hücrelerin adresleme hatları yoktur. Böylece bellek konumların seçimi yapılamaz, yani yazma süreci sırasında giriş hatlarında bulunan verinin hangi hücrede yazılacağına karar getirilemiyor. Ayrıca, farklı konumlardaki hücrelerden çıkış veri hatları, bellekten veri çıkış hatların oluşması için ortak bir noktaya bağlanamıyor, çünkü o zaman onların durumu çakışmalı olacak: Şek. 7-16'ya göre, ortak noktada potansiyel ve bununla beraber veri hatların herbiri tanımlanmamış olacak.



Şek. 7-16. Veri hatların oluşması için bellek hücrelerin çıkışlarının yanlış (geçersiz) bağlanması

Bu nedenlerden dolayı, sıradaki Şek. 7-17'de, A ile işaretlenmiş ek adrekleme hatı ve üç durumlu arabellek devresi içeren bellek hücresinin daha karmaşık ve yaygın türü gösterilmiştir.



Şek. 7-17. Üç durumlu adres girişi, veri girişi ve veri çıkışlı bellek hücresi.

Verilen mantıksal yapıdan, Tab. 7-5 ile sunulan A adres hattının durumunun, bellek hücresinin çalışması üzerine hakim rolü olduğunu görebiliriz.

A	$R/\overline{W}$	Düzen (İşlem)	Çıkış	Harcama
0	X	Seçilmiş değildir	HiZ	Aktif değil (İng. standby)
1	0	Yazma	HiZ	Aktif
1	1	Okuma	$D_{OUT}$	Aktif

Tab. 7-5. Şek.7-17'deki bellek hücresinin işlevsel tablosu

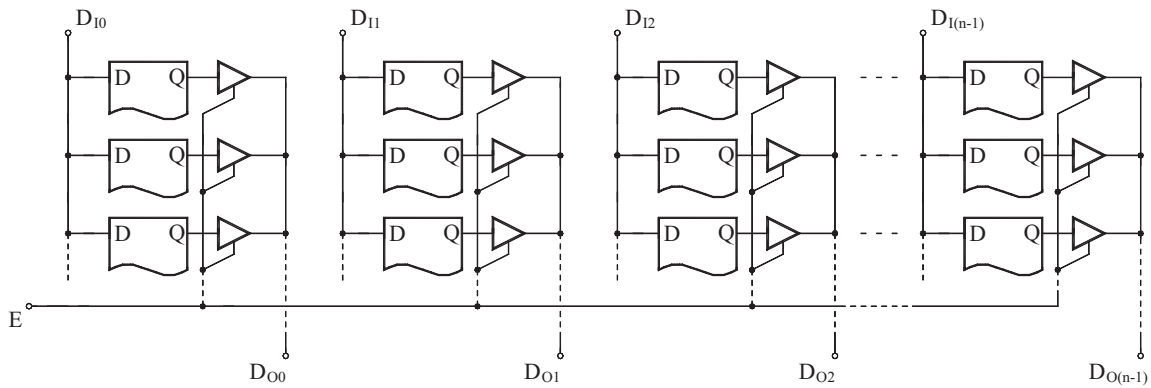
A adres hattı alçak seviyede bulunuyorsa ( $A=0$ ), arabelleğin kontrol hattı alçak seviyeye gidecek. Böylece verinin flip-flop çıkışından çıkış veri hattına iletim engellenecektir, çünkü çıkış arabelleği üçüncü duruma gidecek (HiZ). Aynı zamanda E kontrol hattı da alçak seviyeye geçecek ( $E=0$ ) ve bu yüzden flip-flopun mevcut durumunun değiştirme imkânı yoktur ve flip-flop önceden girilmiş içeriği koruyor. Buna göre bellek hücresi adreslenmiş değilse, onda ne yeni içerik yazılabilir ne de mevcut içerik okunabilir.

Hücresinin etkili çalışması (hücre aktif olması) sadece adreslenmiş olduğu zaman mümkündür, daha doğrusu sadece A hattı aktifleştirilirse ve ona yüksek seviye getirilirse ( $A=1$ ) olabilir. Bu şekilde  $R/\overline{W}$  (veya  $\overline{WE}$ ) sinyaline bağlı olarak, yazma veya okuma olanağı verilecek. Bu arada:

→  $R/\overline{W} = 0$  ise, o zaman DİN girişinde bulunan verinin yazılması için E kontrol hattını aktifleştiren VE-devresi açılıyor, diğer VE-devrenin aracılığıyla çıkış arabelleği devre dışı bırakılıyor, çünkü onun kontrol hattına alçak seviye gönderilerek üçüncü duruma gidiyor (HiZ).

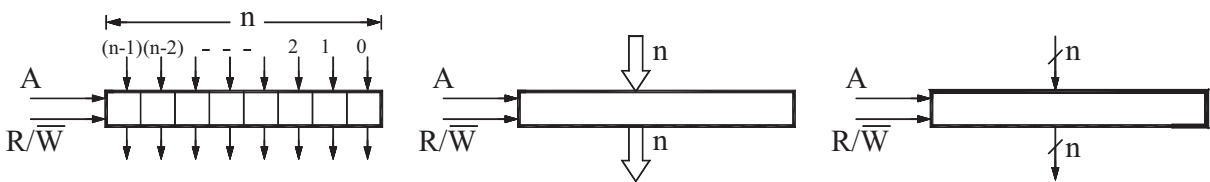
→  $R/\overline{W} = 1$  ise, o zaman çıkış arabelleğinin kontrol hattını etkinleştiren VE-devresi açılarak yüksek seviyeye gidiyor, öyle ki flip-flopun durumu arabellek üzerinden bellek hücrenin çıkışına iletiliyor. Bu sırada verinin yazılması için E kontrol hattı alçak seviyede bulunuyor ( $E=0$ ) ve flip-flopun kendi durumunu koruyor.

Sıradaki şekilden ve yukarıda yaptığımız açıklamadan çıkış arabelleğinin rolü de açıkça belirleniyor. Çıkış arabelleği hücre çıkışın, flip-flopun durumunun (1 veya 0) çıkış hattına yansımalarını sağlıyor, ancak ek olarak arabellek yüksek dirençlik üçüncü durumda da bulunabilir ( $Hiz, RD \rightarrow \infty$ ). Üçüncü durum, Şek. 7-18'e göre her konumdan "paralel" bellek hücrelerin çıkışlarının tek bir ortak noktada bağlanmasını sağlayarak, uygun bit için bellekten tek çıkış veri hattının oluşmasını sağlıyor



Şek. 7-18. Bellek hücrelerin çıkışlarının bağlanması ve veri çıkış hatlarının oluşması

Şek. 7-19 a), b) ve c)'de sekiz hücreden oluşan bellek konumunun sembolik görüntüleri verilmiştir. Bu arada ayrı adres hatları tek bir adres hatına bağlanıyor ve aynı şekilde okuma/yazma kontrolü için ayrı hatlar da tek bir kontrol hattına bağlanıyor.

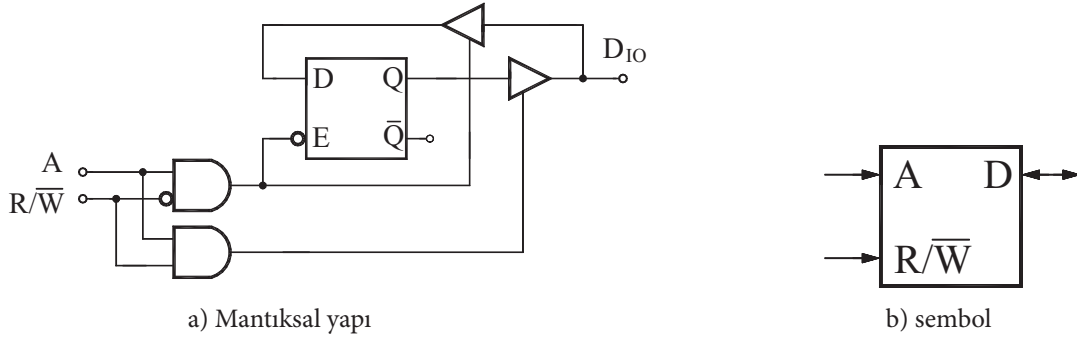


Şek. 7-19. Ayrı adresleme ve okuma/yazma kontrol hatlı bellek konumunun sembolik işaretleri

Şek.7-17'de mantıksal diyagramı gösterilmiş bellek hücrede, veri biti için iki ayrı hattın olması karakteristiktir: birinde yazılıyor, diğerinden ise okunuyor. Ancak, Şek.7-20 a)'ya varolan arabelleğe paralel ancak ters yönde bir arabellek daha eklenirse veri için iki yönlü olacak tek bit hat kullanılabilir, yani o hattan verinin hem yazılması hem de okunması sağlanacak.

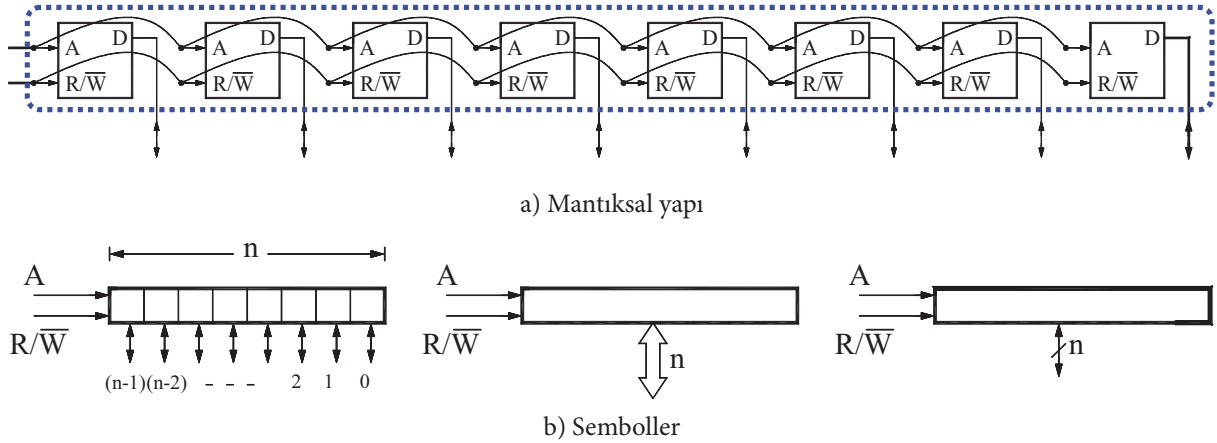
Şek.7-20 a)'dan, bu şekilde gerçekleşen bellek hücrenin çalışma prensibi hakkında sonuç getirilebilir.

1. A adres hattına alçak seviye getirilirse ( $A=0$ ), hücre adreslenmeyecek. O zaman flip-flop ve üçüncü durumda (HiZ) bulunan hücre çıkışı arasında iletişim olmadığından dolayı her iki arabellek devre dışı kalacak. Flip-flop pasiftir ve mevcut durumu koruyor.
2. Hücre, adres hattında yüksek seviyenin getirilmesiyle adresleniyor ( $A=1$ ). Bu arada  $R/\overline{W}$  sinyalin seviyesine bağlı iki durum mümkündür:
  - ⊕ Okuma sırasında  $R/\overline{W}$  yüksek seviyeye gitmelidir ( $R/\overline{W}=1$ ). Böylece üst (çıkış) arabelleğin flip-flopların durumunu bu durumda çıkış hattı olan veri hattına kadar iletmesi sağlanıyor, alt (giriş) hattı ise kapalıdır:
  - ⊕ Yazma sırasında  $R/\overline{W}$  alçak seviyeye gitmelidir ( $R/\overline{W}=0$ ). Böylece alt (giriş) arabelleğin açılması/aktifleştirilmesi sağlanıyor. Bununla, bu durumda giriş hattı olan veri hattında bulunan veri bitin de flip-flopa yeni içerik gibi girmesi sağlanıyor. Bu arada üst (çıkış) arabellek kapalıdır. Bu bellek hücrenin sembolik işareti Şek.7-20 b)'de gösterilmiştir.



Şek. 7-20 Üç durumlu adres girişli ve iki yönlü veri hatlı bellek hücresi

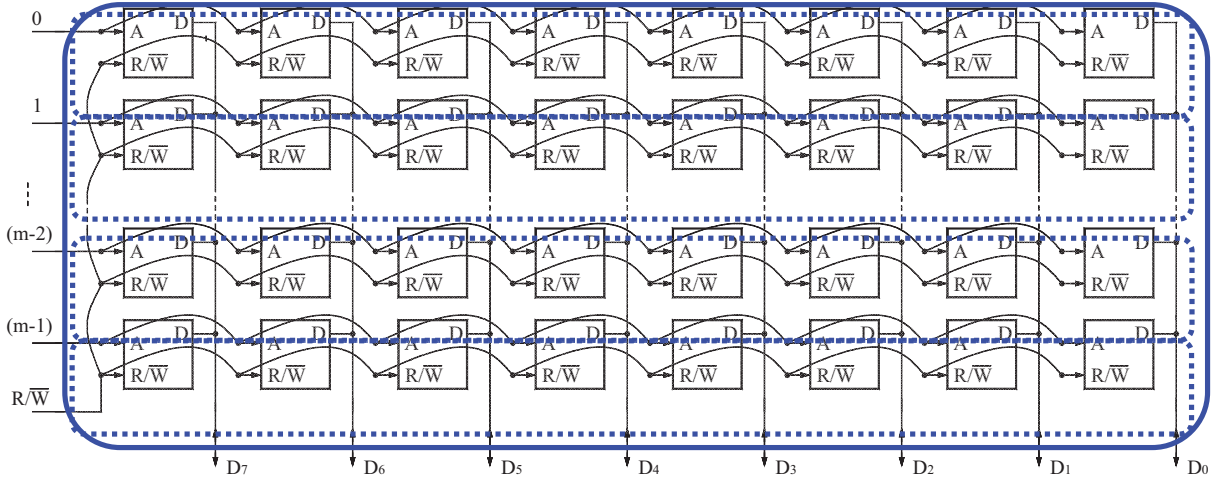
8 bitli bellek sözcüğün (konumun) oluşması için 8 bellek hücrenin Şek. 7-21 a)'ya göre bağlanması gerekiyor. Şek. 7-21 b)'de onun sembolik işareti gösterilmiştir.



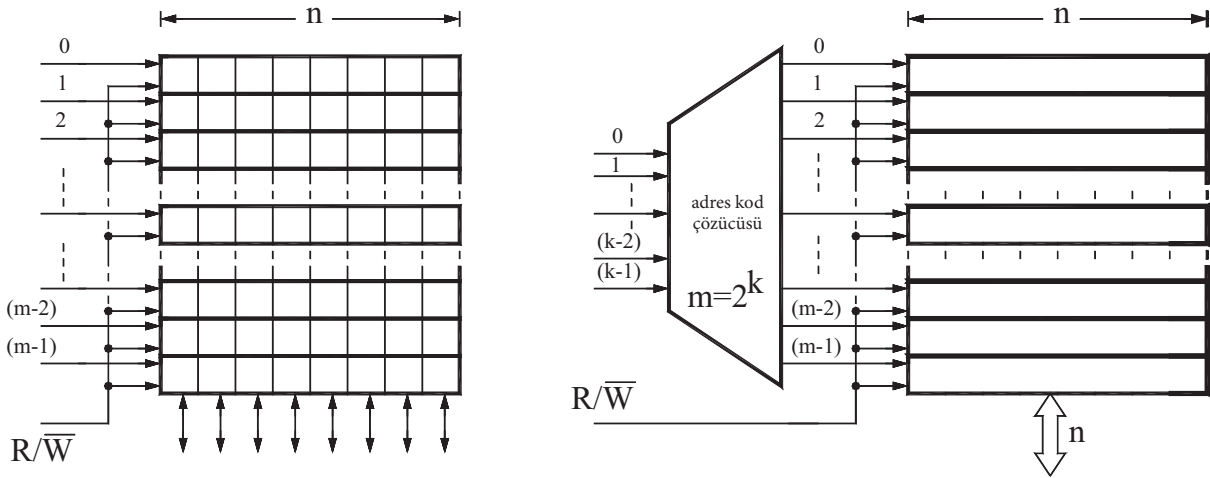
Şek. 7-21. Adres girişli, okuma/yazma kontrol girişli ve iki yönlü veri hatlı sekiz bitli bellek konumu

Şekilden görüldüğü gibi bellek sözcüğü için adres hattı tek bir hücreye seçimi sağlar. Adres hattı tüm hücre seçim girişlerine paralel olarak bağlanıyor. Paralel olarak tek hatta okuma/yazma girişleri de bağlanıyor. Bellek hücrelerin sadece böyle bağlanma şekli, belleğe erişebilen en küçük adreslenebilir yapı olarak tek bellek konumunun oluşmasını sağlıyor.

Bellek matrisin oluşması için büyük sayıda konumların aşağıdaki şekile (Şek. 7-22) göre bağlanması gerekiyor. Şekildeki görünümün çizgilerle oldukça yüklü olduğunu ve görünürlüğü zayıf olduğunu göz önüne alarak, çok daha sıkça Şek.7-23 a)'daki sembolik işaretleme ya da bellek matrisini kod çözücüyle beraber gösteren Şek.7-23 b)'deki işaretleme kullanılıyor. Her bellek sözcüğün ayrı adres hattı vardır, tüm bireysel okuma/yazma girişleri ise paralel olarak tek okuma/yazma kontrol hattına bağlanıyor.



Şek. 7-22. m-adres girişli, tek okuma/yazma kontrol girişli ve iki yönlü veri hatlı 8 bitli bellek matrisin oluşması için SRAM bellek hücrelerinin bağlanması



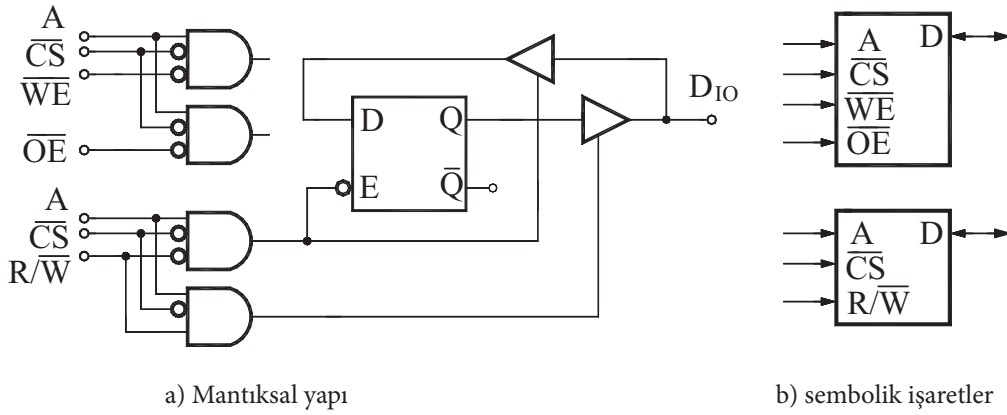
a) Adres kod çözücüsüz

b)  $m=2^k$  geçerli olduğu k/m adres kod çözücülü

Şek. 7-23. m-adres girişli, okuma/yazma kontrol girişli ve iki yönlü veri hatlı 8 bitli bellek matrisin mantıksal yapısı

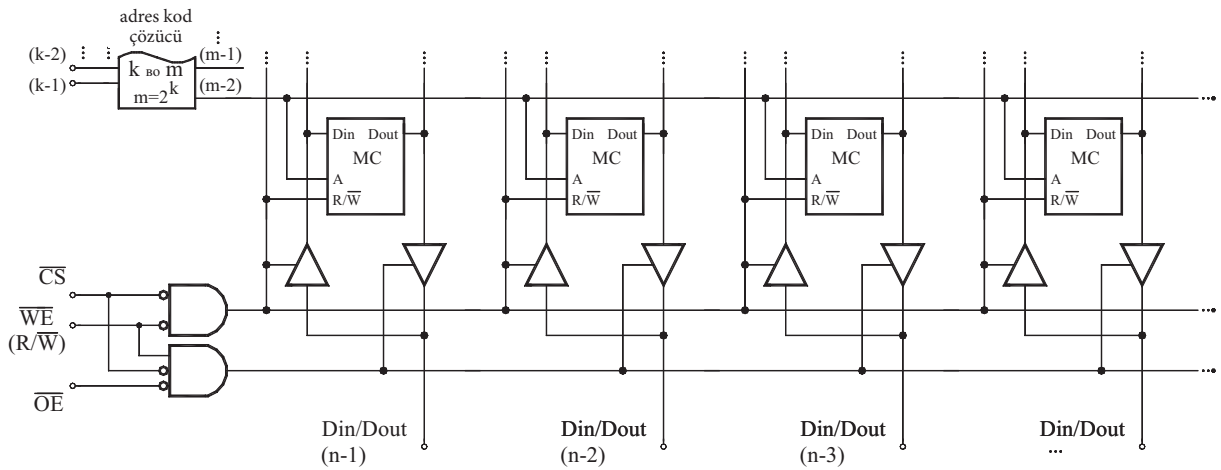
k dış, yani  $m=2^k$  iç adres hatları ve veri hatları ile  $R/\overline{W}$  (veya  $\overline{WE}$ ) okuma/yazma girişi dışında, RAM bellek bileşenlerin Şek.7-24'e göre iki ek kontrol girişi daha vardır. Şek. 7-24 a) ve b)'ye uygun olarak bir girişle bellek tümleşik devre seçiliyor ve bu yüzden bu giriş CS ile işaretleniyor (İng. chip select ya da yonga seçimi), diğer kontrol girişi ise OE ile belirtilip verilerin çıkmasını (okumayı) kontrol ediyor ve sağlıyor veya engelliyor (İng. output enable). Gerçek bellek bileşenlerinde bu sinyaller genelde alçak seviyede aktiftir.

Yonga seçilmiş değilse, o zaman  $\overline{CS} = 1$  olacak ve tüm veri hatları üçüncü durumda bulunacak (HiZ). Bellekten okumak ya da belleğe yazmak için yonga seçim hattına aktif (alçak) seviye getirilmelidir ( $\overline{CS} = 0$ ). Bu arada adres hatlarında ve dolayısıyla adres kod çözücünün girişinde, kod çözücünün uygun çıkış adres hattının aktifleştirilmesi için, belirtilen bellek konumun adres bilgisi yerleşmelidir.



Şek. 7-24. Seçim, adresleme ve okuma/yazma kontrol mantıklı bellek hücresi

Bu koşullar altında, okumak için, okuma ve yazma bir kontrol sinyalle,  $R/\overline{W}$  ile yönetilirse  $R/\overline{W} = 1$  koşulu geçerli olması gerekiyor, ya da bu süreçleri yönetmek için bellek yongasının iki kontrol sinyali varsa  $\overline{WE} = 1$  ve  $\overline{OE} = 0$  geçerli olmalıdır. Ancak yazıldığı zaman, sinyallerin mantıksal durumları önceki durumlardan ters olmalıdır: okuma/yazma için tek kontrol sinyal durumunda  $R/\overline{W} = 0$  olmalıdır ya da bellekte okuma/yazma için iki kontrol sinyali varsa  $\overline{WE} = 0$  ve  $\overline{OE} = 1$  olmalıdır.



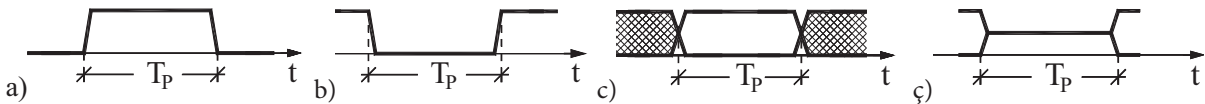
Şek. 7-25. SRAM bellek matrisi çerçevesinde oluşan bellek sözcüğü ve kontrol sinyalleri

Şek. 7-25 bir SRAM bellek matrisinden dört bellek hücrenin tek bir bellek sözcüğüne bağlanmasını ve kontrol mantığı ile adres kod çözücüyü gösteriyor. Şekile göre, en büyük  $a_{(m-1)} = 2^k - 1$  adresi olan belleğin son konumu söz konusudur. Tüm giriş adres hatları yüksek seviyede, ya da  $a_{(k-1)}, a_{(k-2)}, a_{(k-3)} \dots a_1, a_0 = 111 \dots 11$  geçerli olunca bu konum adreslenecektir.

### 7.6.2. ASENKRON OKUMA VE YAZMA

Belleğin çalışmasını, onunla beraber belleği okuma ve bellekte yazma süreçlerini de merkezi işlemci kontrol ediyor ve yönetiyor. İşlemcinin çalışması onun iç pals sinyalinin senkron iken, bellek asenkron da çalışabilir. Belleğin çalışması, yonga seçimi için giriş kontrol sinyallerinin mantıksal seviyelerinin değişmesiyle ve okuma/yazma için iki kontrol hatında yerleşmesi gereken sinyallerle yönetiliyor. Adresleme hatlarında bulunan sinyaller, içeriği veri hatlarında meydana gelen bellek konumunu belirliyorlar.

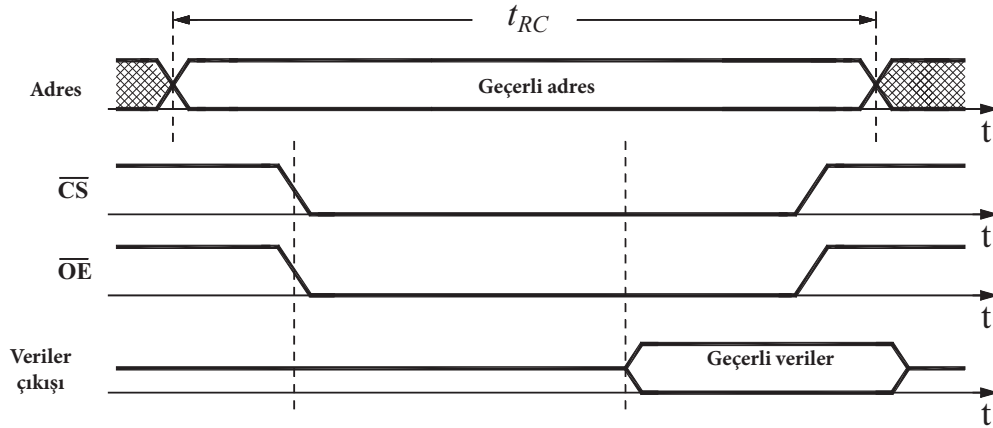
Adreslenen bellek konumundan okuma sürecini, ya da bellekte yazmayı açıklamak için, Şek. 7-26'da verilen zamansal diyagramları kullanacağız. Bu şekilde dijital sistemin bileşenleri arasında iletişim kuran giriş ve çıkış hatlarının gerilim seviyeleri ve bununla beraber mantıksal durumları gösterilmiştir. Yüksek seviyenin (mantıksal 1 seviyesi) sunmak için belirli süre  $T_p$  süren gerilim dikdörtgen dürtüsü kullanacağız (Şek. 7-26 a). Alçak seviye için (mantıksal 0 seviyesi) gerilim sıfırdır ve bu yüzden  $T_p$  zaman aralığının zamansal çizgisinde gerilimde hiçbir türlü değişim olmayacak ve seviye sıfır olacak (Şek. 7-26 b).  $T_p$  zaman aralığında hatta herhangi bir durumda, yüksek veya alçak (1 veya 0) veri meydana gelebilir,  $T_p$  zaman aralığının başlangıç ve son anında "X" harfi şeklinde iki çapraz çizgi çekiliyor (Şek. 7-26 c). Onların arasında t-eksenine paralel, alçak ve yüksek seviye için iki çizgi çekiliyor. Bu çizgiler, süresi  $T_p$ 'ye eşit olan ikili verinin meydana geldiğini belirtiyor.  $T_p$  zaman aralığı sırasında, hat sonsuz yüksek direnç durumuna bulunuyorsa (yüksek empedans, HiZ), zaman ekseninde gerilim seviyesine paralel, yüksek ve alçak seviyenin ortasında çizgi çekiliyor. (Şek.7-26 ç)



Şek. 7-26. Farklı mantıksal durumlara uyan gerilimlerin dalgasal şekilleri

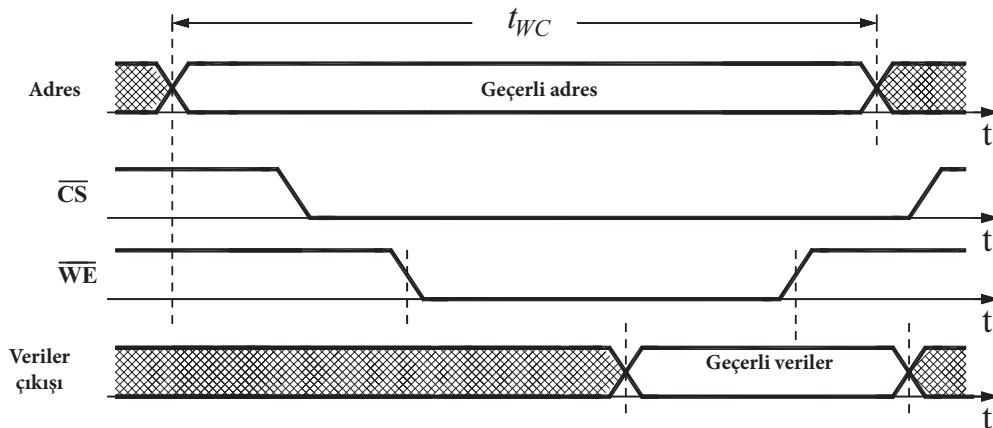
Sıradaki zamansal diyagramlarda, görsel olarak alt alta verilen bellek hatlarının mantıksal durumları sunulmuştur. Böylece bu hatlarda bulunan sinyallerin meydana gelme sıralamalı ve süreleri kolayca dikkat edilebiliyor. İşlemlerin gerçekleşmesi için her iki durumda, yonga seçim sinyali alçak olmalıdır, belleğe ise adres bitleri adres hatlarına yerleşen bilinen adres ile erişiliyor. Okuma/yazma kontrol hatlarının durumları, gerçekleşmesi gereken işleme bağlı olarak kendi değerlerini değiştirecek.

Okuma süreci,  $\overline{WE} = 1$  ve  $\overline{OE} = 0$  koşulu geçerli olduğu zaman gerçekleşiyor. Okunan ve belirlenen bellek konumunda depolanan veri, adres hatlarında adres bilginin meydana gelmesiyle aynı anda meydana gelemez. Onun için belirli zaman aralığı gerekiyor. Bunu göz önüne alarak *erişim zamanı* tanımlanıyor. Şek.7-27'deki zamansal diyagramlarında erişim zamanı, adres bilgilerin adres hatlarında yerleştiği andan, çıkış veri hatlarında adreslenen konumun içeriği meydana geldiği ana kadar, en uzun süre olarak gösterilmiştir. Adres bilgisinin adres hatlarına getirildiği ve okunması için gereken verilen süre kadar orada kaldığı zaman aralığı *okuma döngüsünün süresi* olarak tanımlanıyor ve  $t_{RC}$  ile işaretleniyor.



Şek.7-27. Okuma döngüsü ve erişim zamanı

Yazma işlemi gerçekleştiği sırada kontrol hatlarının mantıksal durumu  $\overline{WE}=0$  ve  $\overline{OE}=1$  olmalıdır. Okumaya benzer olarak, yazmada *yazma döngüsünün süresi* tanımlanıyor. Şek. 7-28'deki zamansal diyagramlarda gösterilen bu süre  $t_{WC}$  ile işaretleniyor. Bu zaman aralığı adres bilgisinin yerleşmesinden, belirlenen adres konumunda yeni verinin (bellek sözcüğün) yazıldığı tüm iç (içsel) bellek işlemlerin tamamlanmasına kadar en uzun süreyi tanımlıyor.



Şek. 7-27. Yazma döngüsü ve yazma döngüsünün süresi

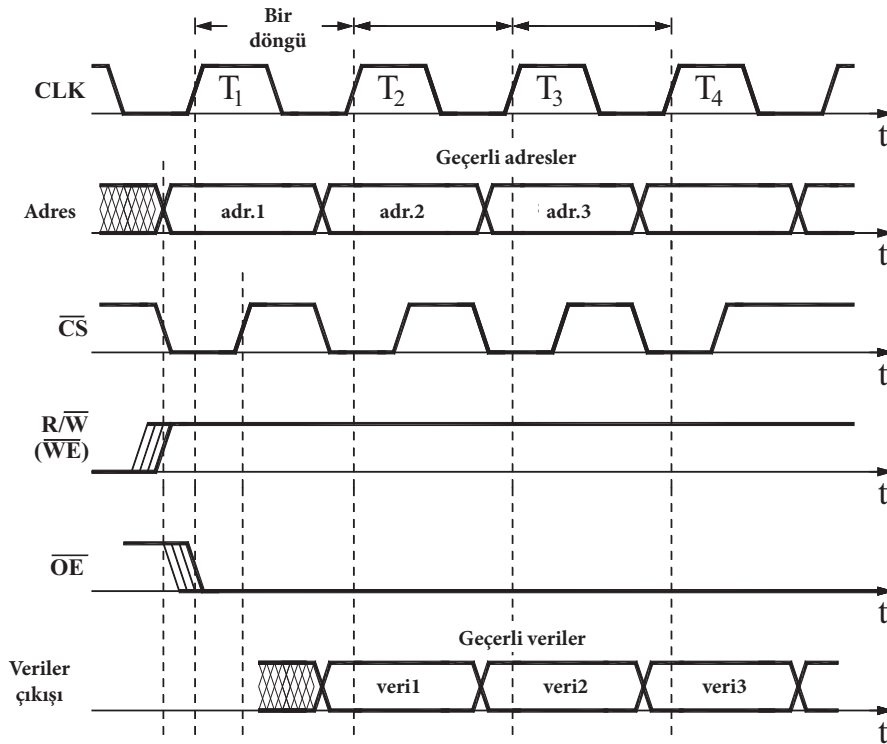
Yukarıdaki açıklamayı ve Şek. 7-27 ve Şek. 7-28'deki zamansal diyagramları dikkate alarak, bellekte okuma ve yazma işlemleri sırasında, işlemcinin kontrol hatlarının durumlarını yöneten dijital bileşen olduğunu bilmek gerekiyor. Bu arada, dijital sistemin doğru çalışması için, sistemin okuma döngüsünün, ya da yazma döngüsünün süresi çerçevesinde, kontrol sinyallerin art arda meydana gelmesi arasında, gecikme zamanını göz önüne alması gerekiyor.

### 7.6.3. SENKRON OKUMANIN BELLEK DÖNGÜSÜ

Senkron SRAM'ın çalışması, zamansal olarak, işlemcinin de çalıştığı sistem palsla, daha doğrusu okuma veya yazma sürecinin başladığı aktif kenarın meydana gelmesiyle uyumludur. Bellek matrisi ve kontrol sinyalleri açısından senkron ve asenkron SRAM arasında ilkesel ayrım yoktur. Fakat, temel fark, senkron SRAM'ın içinde tüm sinyallerin: veriler, adresler ve kontrol sinyalleri sistem palsın ön kenarının meydana gelmesiyle eşzamanlı kilitlendiği farklı yazmaçların içermesidir.



Verilen bellek konumundan veri okuma işlemi bir kaç temel adımda gerçekleşiyor. Bu adımlar Şek. 7-29'da sunulan zamansal diyagramlarda daha görünebilir şekilde açıklanmıştır.



Şek. 7-29. Bellekten okuma döngüsünün gerçekleştiği zaman karakteristik sinyallerin zamansal diyagramları

(1) Pals sinyalin yükselen (aktif) kenarının meydana gelmesinden önce, mikro-işlemci adres hatlarında bellek sözcüğün (ikili verinin) okunması gereken bellek konumunu belirleyen adresi yerleştiriyor,

(2) Bununla aynı zamanda işlemci bellek yonganın seçim  $\overline{CS}$  hattına alçak seviye getirerek,  $\overline{CS}$  hattını aktifleştiriyor ( $\overline{CS} = 0$ ), okuma süreci ise WE okuma/yazma kontrol hattını yüksek seviyeye ayarlanarak ve  $\overline{OE}$  çıkış olanağı hattına alçak seviyeye ayarlanarak sağlanıyor ( $\overline{WE} = 1$  ve  $\overline{OE} = 0$ ),

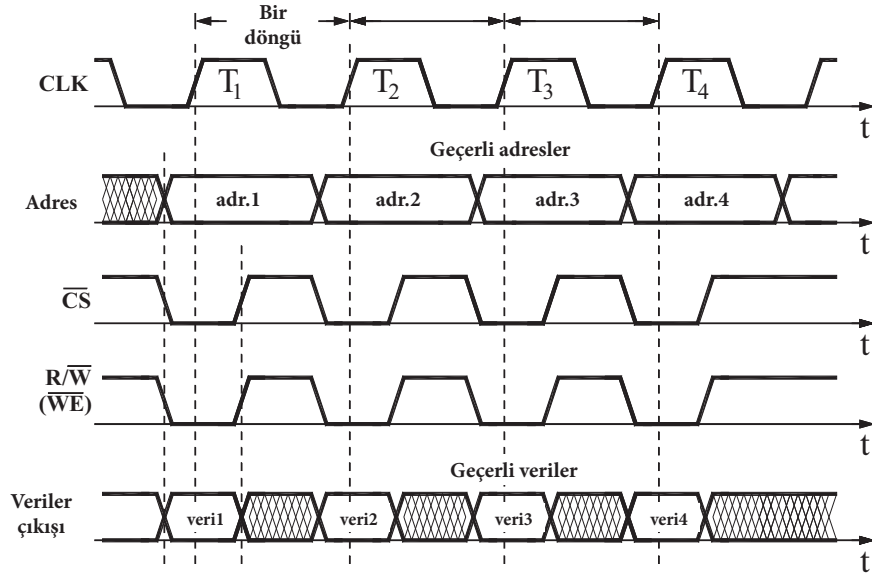
(3) Adres hatlarına adres bilginin yerleştiği andan ve erişim zamanının eşit olan gecikmeden sonra başlayarak bellek veri hatlarında adreslenmiş bellek konumdaki verinin (içeriğinin) sekiz biti yerleşiyor. Söz konusu okuma şekline bağlı olarak bu bitler aynı pals aralığında ya da sonraki pals aralığında meydana geliyor.

Belirtilen (adreslenen) bellek konumundan okunan veriyi (baytı) işlemci kendi iç yazmacında onun yeni içeriği olarak yazarak okuma işlemi tamamlanıyor ve işlemci başka okuma veya yazma döngüsü gerçekleştirebilir.



## 7.6.4. SENKRON YAZMANIN BELLEK DÖNGÜSÜ

Adreslenmiş bellek konumuna verinin yazma işlemi de, Şek. 7-30'da verilen zamansal diyagramlara göre birkaç temel adımdan oluşuyor.

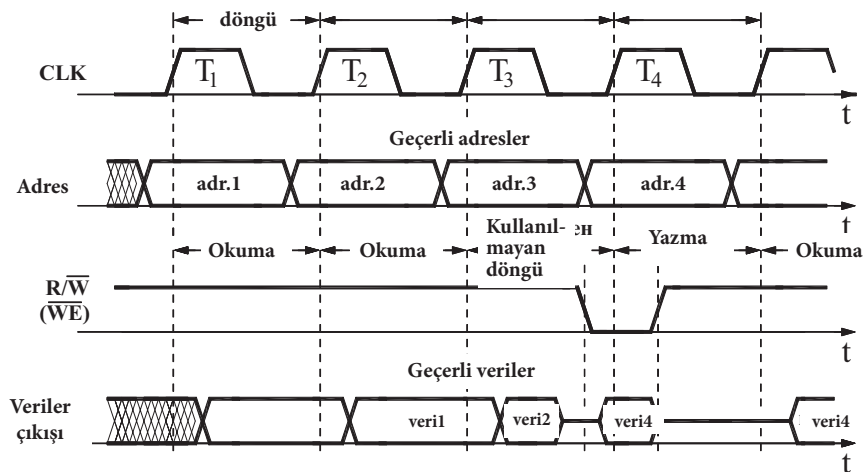


Şek. 7-30. Yazma döngüsü sırasında karakteristik sinyallerin zamansal diyagramları

(1) Pals sinyalin aktif (ön) kenarın meydana gelmesinden önce, bellek sözcüğün (ikili verinin) yazılması (girilmesi) gereken bellek konumunu belirleyen adres bilgisi, mikro-işlemci tarafından adres hatlarına yerleşiyor. Aynı zamanda işlemci bellek yongasını da seçiyor, yani uygun  $\overline{CS}$  yonga seçim hattını aktifleştirerek ve onu alçak seviyeye indirerek ( $\overline{CS} = 0$ ) yonganın çalışmasını sağlıyor,

(2) Aynı anda  $\overline{WE}$  okuma/yazma için kontrol hattına işlemci yazmak için aktif sinyal ayarlıyor, o da alçak seviyededir:  $\overline{WE} = 0$ . Bu sinyalin sürdüğü süresi veri hattında bulunan verilerin bellek bileşeninde yazılması için yeterince uzun olmalıdır. Verinin yazılmasından sonra bu sinyal pasif oluyor, yani yüksek seviyeye geri dönüyor.

Bu şekilde veri veriyolunda yerleşmiş olan yeni sözcüğün (verinin) yazma sürecinde, yazma şekline bağlı olarak, işlemcinin aynı palsında ya da sonraki palsta, veri belirtilen (adreslenen) bellek konumunda, yeni içerik olarak yazılıyor.



Şek. 7-31. Okuma ve yazma sırasında karakteristik sinyallerin zamansal diyagramları

**Örnek:** Devamda tab.7-6 ile sunacağımız örnek için, okuma ve yazma süreçlerin merkezi işlemci ve basit bir mikro-bilgisayarın SRAM bellek yongası arasındaki iletişime ilişkin olduğunu tahmin edeceğiz. Onun mikro-işlemcisi  $k=16$  adres ve  $n=8$  veri hatlarına sahip olduğunu alırsak, adreslenebildiği bellek alanın  $2^{16}$  bellek sözcüğü olduğunu elde ediyoruz, ya da sekiz bit (bir bayt) uzunluğunda  $2^6 \times 2^{10} = 64 \times 1024 = 64K$  sözcük olduğunu elde ediyoruz. Bu yüzden  $k=16$  adres ve  $n=8$  veri hatlı ve  $2^{16} = 64KB$  kapasiteli uygun SRAM bellek tümleşik devre kullanılıyor.

Kontrol hatları			Adres hatları		Veri hatları		
İşlem	$\overline{CS}$	$R/\overline{W}$	İkili	On altılı	İkili	On altılı	Yön
Okuma	0	1	0000000011001010	00CA	11110000	F0	Çıkış
Yazma	0	0	0000000011001111	00CF	01111001	79	Giriş
Aktif olmayan	1	X	xxxxxxxxxxxxxxxxxxx	xxxx	xxxxxxxxxxx	xx	HiZ (Yüksek dirençlik)

Tab.7-6. Temel bellek işlemler örneği

Tab.7-6'da a) adresi  $202_{DEC} = 0000000011001010_{BIN}$  olan bellek konumunda bulunan  $240_{DEC} = 11110000_{BIN}$  verinin okunduğu zaman; b)  $207_{DEC} = 0000000011001111_{BIN}$  adresli bellek konumunda  $121_{DEC} = 01111001_{BIN}$  verisi yazıldığı zaman; c) işlemci ve bellek arasında iletişim olmadığı zaman, RAM bellek bileşeninde tüm hatların durumları gösterilmiştir

## TEKRARLAMA SORULARI VE ÖDEVLERİ

- 7-1. Bellek bileşenlerin ve cihazların temel rolü (görevleri) nedir?
- 7-2. Yarı iletken bellekleri ve sabit diskleri karşılaştır ve çalışma hızı, kapasite ve maliyetleri açısından avantajlarını ve dezavantajlarını say.
- 7-3. Organizasyon açısından bellek nasıl mantıksal yapıdır?
- 7-4. Bellek sözcüğü.....
- 7-5. Bellek konumu.....
- 7-6. Bellek hücresi ..... tanımlıyor.
- 7-7. Sözcük uzunluğu ..... tanımlıyor ve .....ile ifade ediliyor.
- 7-8. Sözcük içeriği.....dir.
- 7-9. Veri hatların rolü nedir? Onlarda nasıl bilgi meydana geliyor? Onların sayısı neye bağlıdır?
- 7-10. Bellekte en küçük adreslenebilir bütün .....dir.
- 7-11. Belleğin kendi işlevini gerçekleştirdiği iki temel işlem hangisidir ?
- 7-12. Okuma.....tanımlıyor.
- 7-13. Yazma.....tanımlıyor.
- 7-14. Yıkıcı (bozucu) ve yıkıcı olmayan okuma arasında fark nedir?
- 7-15. Adres.....tanımlıyor.
- 7-16. Adres hatların rolü nedir? Onlarda ne tür bilgi meydana geliyor? Onların sayısı neye bağlıdır?
- 7-17. Adres kod çözücünün rolü nedir?
- 7-18. Bellek kapasitesi.....tanımlıyor.

7-19. Bayt-organize edilmiş bellek devrenin a) 512 MB, b) 8 KB c) 64 KB kapasitesi varsa kaç adres ve veri hatı olması gerekiyor? Bellek sözcüklerin iki bayt uzunlukları varsa (1sözcük = 2 B) aynı soruya cevap ver.

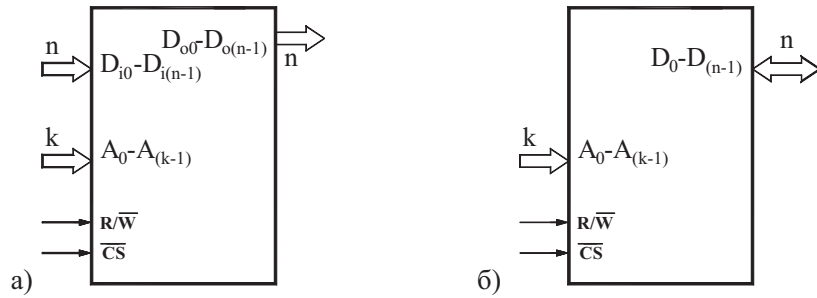
7-20.  $R/\overline{W}$  kontrol hattının amacı nedir?

7-21.  $\overline{WE}$  kontrol hattının amacı nedir?

7-22.  $\overline{OE}$  kontrol hattının amacı nedir?

7-23.  $\overline{CS}$  ( $\overline{ME}$ ) kontrol hattı ne için kullanılıyor?

7-24. Sembolik işaretleri aşağıdaki şekillerde gösterilen bellek bileşenleri için giriş/çıkış hatlarından her birinin rolü nedir?



Soru 7-24 şekilleri

7-24.a)  $k=10$ , b)  $k=16$ , c)  $k=20$ , ç)  $k=30$  adres hattı olan ve a)  $n=8$ , b)  $n=16$ , c)  $32$ , ç)  $n=64$  veri hattı olan bellek bileşeni nasıl organize edilmiştir ve baytlarla ifade edilmiş kapasitesi nekadardır? Onun sembolünü çiz!

7-25. a)  $n=8$ , b)  $n=16$ , c)  $n=32$ , ç)  $n=64$  bit uzunluğunda a)  $m=1024$ , b)  $m=4096$ , c)  $m=65536$ , ç)  $m=536870912$  bellek sözcüğüne sahip olan bellek bileşenin kaç adres ve veri hattı olmalıdır. Onun sembolünü çiz.

7-26. Metni 8-bitli ASCII koduna göre girmemiz gereken a) 100, b) 250, c) 320 sayfalık kitabı yerleştirmemiz için bellek tümleşik devrenin satın almamız gerektiğini tahmin edelim. Verilen kitapların metnini girmemiz için satın alacağımız bellek bileşenin baytlarla ifade edilmiş kapasitesi ne kadar olmalıdır? Metnin her sembolüne ayrıdan erişebilmek için belleğin organizasyonu nasıl olmalıdır? Pentiuma dayanan kişisel bilgisayarda (1) 1 GB'lık, (2) 2 GB'lık işletim belleğinde kaç böyle kitap yerleşebilir?

7-27. Verilere erişim ve verilerin ezberlenmesi için süre açısından ROM ve RAM bellek bileşenleri arasında temel fark ve benzerlik nedir?

7-28. ROM bellek bileşenlerin ayrımını belirt. Onların arasındaki fark nedir?

7-29. RAM bellek devrelerin ayrımını belirt. Onların arasındaki fark nedir?

7-30. SRAM ve DRAM bellek bileşenleri arasındaki farklar nedir?

7-31. Asenkron ve senkron SRAM arasındaki farkları say.

7-32. SRAM bellek hücresinin ilkel elektrik diyagramını çiz.

7-33. İki veri hatlı SRAM bellek hücresinin mantıksal yapısını çiz. İşlevsel tablo kullanarak bu bellek hücresinin çalışma prensibini açıkla.

7-34. Bir veri hatlı SRAM bellek hücresinin mantıksal yapısını çiz. İşlevsel tablo kullanarak bu bellek hücresinin çalışma prensibini açıkla.

7-35. Verilen tabloda ilk ve son üç bellek konumunda a,b,A ve B harflerin ASCII kodları , ile 8-bitli ASCII koduna göre 1 ve 2 rakamları yerleşmişse, bu konumların içeriklerini yaz. Bellek bileşenin (1) 1 bayt, (2) 2 bayt uzunluğunda a)16, b)32 bellek sözcüğüne sahip olduğunu tahmin et.

7-36. Bellek sözcükleri 1) 1 bayt, 2) 2 bayt uzunluğunda olan bellek bileşenin a) 1K, b) 4K, c) 16K, ç)64K kapasitesi var. a) 255(10), b) 256(10), c) 1023(10), ç) 1024(10), d) 4095(10), e) 4096(10) adresli bellek konumlarında sırasıyla a,b,A ve B harfleri ile 8-bitli ASCII koduna göre 1 ve 2 rakamları bulunuyor. Verilen tabloyu doldur.

Adres		İçerik		
İkili	On altılı	İkili	On altılı	ASCII sembol

Soru 7-35 ve 7-36 için tablo

7-37. a) 4095, b) 4096 adresli bellek konumlarında bulunan verinin okunması için ve ardından bu verinin a)1023, b) 1024 adresli bellek konumlarına yazdırılması için kontrol, adres ve veri hatların durumları nasıldır. Verilen tabloyu doldur.

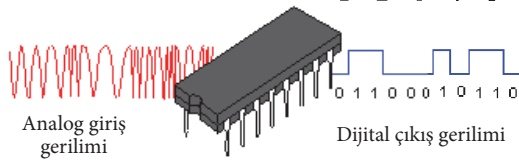
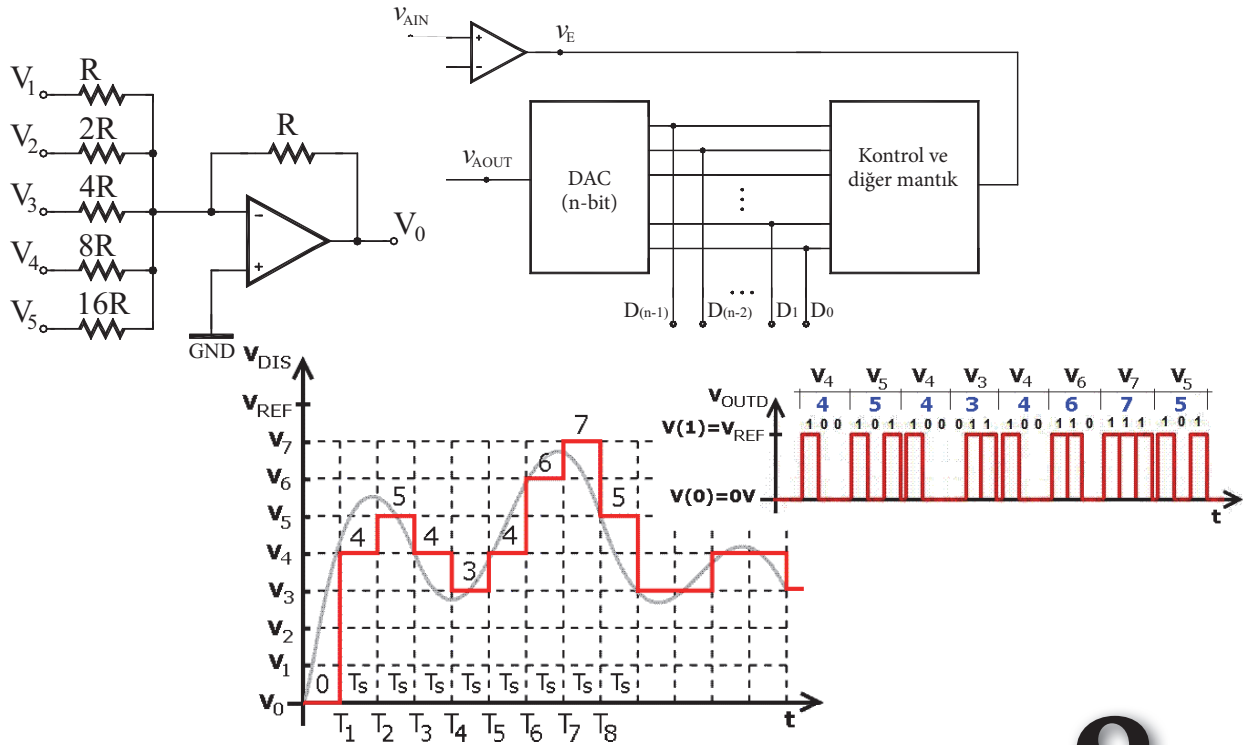
Kontrol hatları			Adres hatları		Veri hatları		
İşlem	$\overline{CS}$	$R/\overline{W}$	İkili	On altılı	İkili	On altılı	Yön

Soru 7-37 için tablo

7-38. Asenkron RAM yongada, Şek.7-26 ve Şek.7-27'de gösterilen zamansal diyagramlara göre, a) okuma, b) yazma süreciyle ilgili sinyallerin rollerini ve sinyallerin meydana gelmelerinin zamansal sıralamasını açıkla.

7-39. Senkron RAM yongada, Şek.7-28 ve Şek.7-29'da gösterilen zamansal diyagramlara göre, a) okuma, b) yazma süreciyle ilgili sinyallerin rollerini ve sinyallerin meydana gelmelerinin zamansal sıralamasını açıkla.Sistem paslın frekansı 5 MHz olduğunu tahmin et.

7-40. (\*) İnternette 74LS189 bellek tümleşik devre hakkında detaylı teknik özellikler dosyasını (İng.datasheet) bulmaya çalış, bak ve incele. Ardından daha detaylı şekilde a) nasıl bellek türünün sözkonusu olduğunu, b) devredeki pinlerin (iğneciklerin) rolü ne olduğunu, c) çalışma prensibi ne olduğunu açıkla ve ç) devrenin çalışmasını göstereceğin bir örnek ver.



# 8. DİJİTAL - ANALOG VE ANALOG-DİJİTAL DÖNÜŞÜM

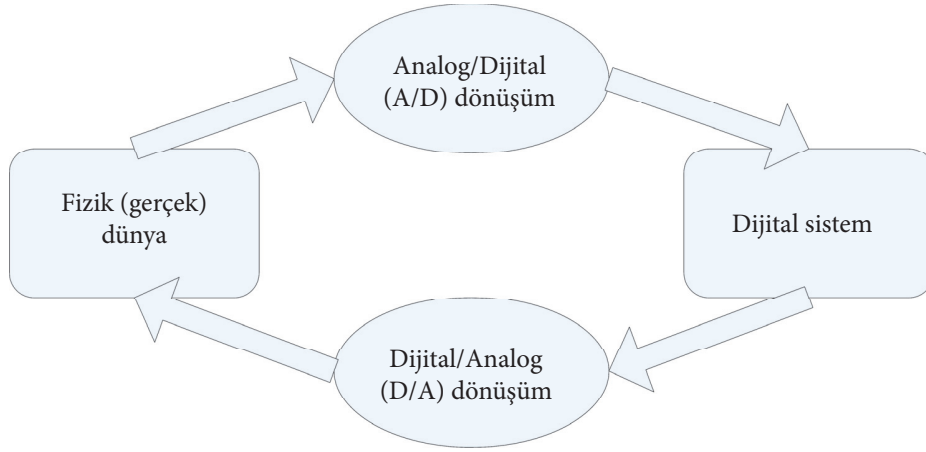
Bu konusal birimi öğrendikten sonra

- ⊕ D/A ve A/D dönüşüm ile ilgili temel terimleri tanıyacaksınız;
- ⊕ D/A ve A/D dönüşüm süreçlerin temel kavramlarını anlayacaksınız;
- ⊕ D/A ve A/D dönüşüm süreci;
- ⊕ Farklı D/A ve A/D dönüşüm yöntemlerini tanıyacaksınız ve ilkeli olarak açıklamayı bileceksiniz;
- ⊕ Farklı D/A ve A/D dönüştürücü türlerini bileceksiniz ve onların çalışmasını anlayacaksınız.



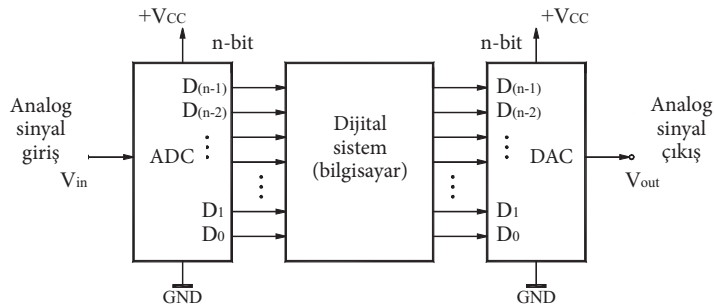
## 8.1. GİRİŞ

Analog-dijital (A/D) ve dijital-analog (D/A) dönüşüm konusunun evrensel ve aynı zamanda en genel görünümü Şek.8-1'de gösterilmiştir. Bir tarafta gerçek (fiziksel dünya varken, diğer tarafta dijital sistemler bulunuyor. Bu iki taraf arasında iletişim sadece A/D ve D/A dönüştürücüleri yardımıyla kurabilir. A/D ve D/A dönüştürücülerin özel bileşenler olarak uyum sağlamak görevleri vardır. Gerçek dünya ve dijital sistemler arasında temel bir fark var: Gerçek dünyada olaylar genelde süreklidir, sonsuz sayıda değer alıyorlar ve bu yüzden bu olayları tanımlayan sinyaller analogdur. Diğer taraftan, dijital sistemler sadece dijital sinyallerle çalışıyorlar. Dijital sinyaller ikili sözcükler dizileridir ve her biri dizi sadece iki gerilim seviyenin kombinasyonudur: yüksek  $V(1) = +V_{cc}$  ve alçak  $V(0)=0$ . Bu iki seviye sınırlı sayıda seviyeli ayrık değerleri olan sinyallerin ikili kodlanmış temsilcilerini tanımlayan mantıksal 1'in ve mantıksal 0'ın karşılığıdır.

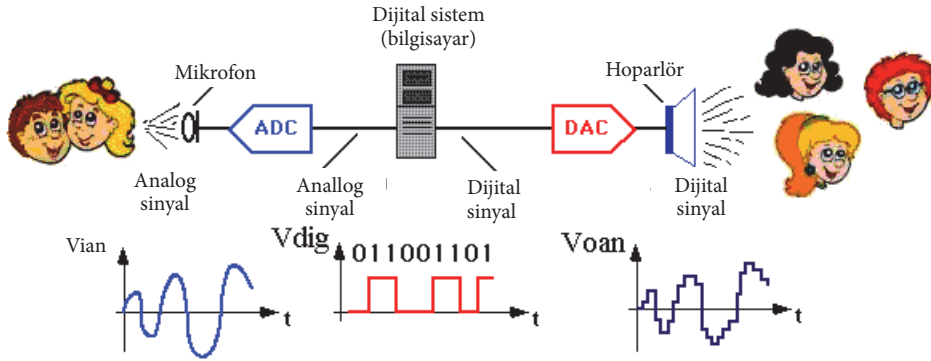


Şek. 8-1. Gerçek dünya ve dijital sistemler arasındaki etkileşimde A/D ve D/A dönüştürücülerin yerini ve rolünü gösteren blok-diyagram

A/D ve D/A dönüştürücülerin dijital cihazlarda yer almaları şarttır, ancak daha geniş çapta da yer almaları gerekiyor, çünkü yakın tüm elektronik cihazlarda sıkça kullanılıyorlar. A/D ve D/A dönüştürücülerin ortak kullanımı, elektronik sistemlerde, süreçlerde ve gerçek dünyadan olaylar üzerine kontrol ve yönetim rolünü gerçekleştirmek amacıyla, girişte farklı sensörlerle (İng. sensors) ve çıkışta çıkış dönüştürücü elemanlarla (İng.transducers) tam bağlanma (İng.interface) sağlıyorlar. Şek.8-2'de, A/D ve DA dönüştürücüler en basit şekilde, genel amaçlı bir dijital sistemde giriş ve çıkış bloğu olarak gösterilmiştir. Şek. 8-3'te ise ses sinyalin kişisel bilgisayarla A/D ve D/A dönüşümü için örnek verilmiştir.



Şek. 8-2. Bilgisayarın A/D ve D/A dönüştürücü ile bağlanmasının en basit blok diyagramı



Şek. 8-3. Ses sinyalin kişisel bilgisayarla A/D ve D/A dönüşümü

Dijital cihazın girişine analog cihazın bağlanması gerekince, çalışma oldukça kompleksli olur, çünkü analog çıkışların analog girişleriyle doğrudan bağlanması mümkün değildir çünkü analog sinyallerin sonsuz sayıda değerleri olabilir, dijital sinyallerin ise sınırlı kapsamda sonlu sayıda ayrık değerleri olabilir. Yapılması gereken şu ki analog sinyalin, sınırlı uzunlukta mantıksal 1-ler ve 0-lar dizisi olarak (ikili sözcükler, vektörler) dijital şekile elektronik dönüşümün gerçekleşmesi için bir yol bulunmalıdır. Bununla ilgili, A/D dönüştürücü (İng.analog-to digital converter, ADC), tek girişinde analog sinyal, sıkça olarak gerilim veya akım alan ve bu sinyalin dijital sinyal, yani belirli ikili koda göre bitler dizisi tanımlayan sözcüklere veya doğal ikili sisteme göre ikili sayılara dönüştüren (çeviren) elektronik bileşen ya da cihazdır. A/D dönüştürücünün çıkışında her bit için ayrı hat vardır. Bu hattan bit, işletilmesi için dijital sistemin işlemcisine aktarılıyor.

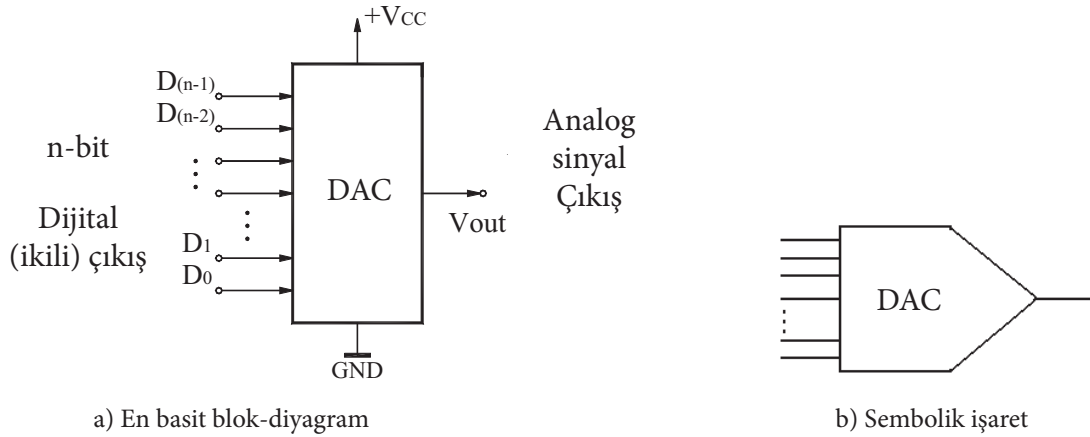
Diğer taraftan, D/A dönüştürücü (İng.digital-to analog converter, DAC) ters işlemi gerçekleştiriyor. Şöyle ki verilerin işletimi tamamlandıktan sonra, bitler kombinasyonu olarak dijital şekilde elde edilen sonuçlar, özel hatlardan D/A dönüştürücülerin girişlerine gönderiliyor. D/A dönüştürücü sonuçları analog sinyale dönüştürüyor ve onları tek çıkışına iletiyor. Mantıksal devrelerin çıkışlarının dönüştürücü elemanlarla (İng.transducers) bağlanması oldukça basittir, çünkü onlar doğaları açısından dijital elemanlardır. Mantıksal devrelerin çıkışları, transistörler, röleler ve benzer cihazların, açık-kapalı olmak üzere iki durumda çalışma prensibinden dolayı, bu cihazlarla kolayca bağlanıyorlar. Benzer özellikleri dijital bileşenlerin çıkışlarında elde edilen sinyallerinde de vardır.

Bu konusal biriminde dikkatimiz A/D dönüşümüne ve onun ters sürecine, D/A dönüşümüne ayıracağız. Farklı dönüşüm süreçlerini ve farklı dönüştürücü türlerin çalışma prensibini inceleyeceğiz ve açıklayacağız.

## 8.2. DİJİTAL - ANALOG DÖNÜŞÜM

D/A dönüşümün, ters A/D dönüşümün gerçekleşmesinden daha basit süreç olduğunu önceden vurgulamıştık. Daha da önemlisi, pratikte içeriğinde D/A dönüştürücü içeren A/D dönüştürücüler kullanılıyor. Bu yüzden önce D/A dönüşüm süreci, ondan sonra ise A/D dönüşüm süreci incelenecektir.





Şek. 8-4. D/A dönüştürücü

D/A dönüştürücünün, girişinde gelen, sözcükler kümesi olarak tanımlanan ve her birinin sonlu sayıda n bit uzunluğu olan dijital sinyalini analog şekile dönüştürme görevi var. Şek.8-4 a) ve b)'de D/A dönüştürücünün en basit blok-diyagramı ve sembolik işareti gösterilmiştir.

### 8.3. TEMEL DENKLEMLER, TERİMLER VE İLETİM ÖZELLİĞİ

D/A dönüştürücü, kelimenin,  $i=0, 1, 2, \dots, (n-1)$  olduğu her n biti  $D_i$  için, ikili sözcüğünde o bitin  $G_i$  ağırlığının evrik değerine uygun analog değer üretmelidir ve ardından tüm bu analog değerleri toplaması gerekiyor ve çıkışında tek analog değerini  $V_{ODAC}$  ( $V_{OUT}$ ) şu denkleme göre vermelidir:

$$V_{ODAC} = K \cdot V_{REF} (D_{n-1} G_{n-1} + D_{n-2} G_{n-2} + \dots + D_1 G_1 + D_0 G_0) \quad (8-1)$$

Denklemden dönüşen sözcüğün her  $D_i$  bitin sadece mantıksal 0 veya 1 değeri olabilir,  $G_i$  ağırlıkları ise 2 sayısının üssü olarak ifade edilmiştir, en büyük  $2^{(n-1)}$ den başlayarak en küçüğüne  $2^0=1$ 'e doğru:  $2^{(n-1)}, \dots, 8, 4, 2, 1, +V_{REF}$  ise D/A dönüştürücünün bağlı olduğu referans gerilimidir.

Yukarıdaki denklemden K orantılılık sabitinin değeri, çıkış geriliminin 0 ve  $V_{REF}$  sınırları arasında değer almasını sağlayacak şekilde olmalıdır. Bu koşulun geçerli olması için K sabitinin,  $V_{REF}$  referans gerilimi daha küçük parçalara ayıran kesir olması gerekiyor, öyle ki  $K=1/2^n$  doğru olacaktır. Bunu (8-1) denkleminde değiştirerek, denklem değişiyor ve şu şekli alıyor:

$$V_{ODAC} = V_{REF}/2^n \cdot (D_{n-1} 2^{n-1} + D_{n-2} 2^{n-2} + \dots + D_1 2^1 + D_0 2^0) \quad (8-2)$$

$$V_{ODAC} = V_{REF} \cdot (D_{n-1} 1/2^0 + D_{n-2} 1/2^1 + \dots + D_1 1/2^{n-1} + D_0 1/2^n) \quad (8-3)$$

K sabiti için  $K=1/2^n$  seçimi ile (bir kutuplu kod şeması veya bir kutuplu kodlama için geçerli olan), tüm bitlerin 0 olduğu en küçük dijital değer için en küçük, sıfırıncı çıkış seviyesi  $V_0=0V$  elde ediliyor, tüm bitlerin 1 olduğu en büyük dijital değer için ise en büyük çıkış seviyesi  $V_{n-1}=[(2^n-1)/2^n] \cdot V_{REF}$  elde ediliyor.

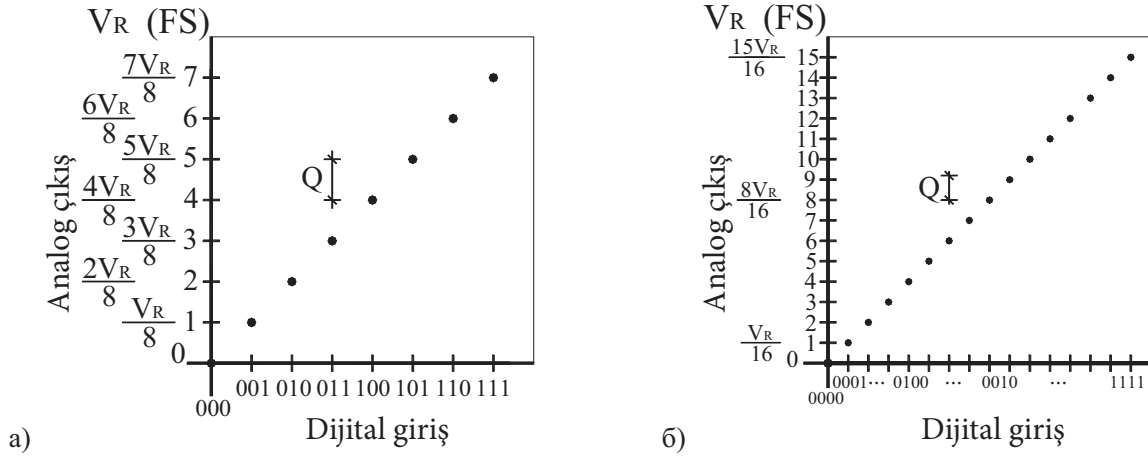
(8-3) denklemini göz önüne alarak, bitlerin sayısı n ne kadar daha büyük ise, elde edilen çıkış analog seviyesi o kadar daha doğru olacağı açıkça görülüyor, çünkü her yeni eklenen bit, çıkış gerilim seviyesini belirleyen toplamda ek olarak daha küçük kesir ekliyor ve  $V_{REF}$ 'i daha küçük parçalara ayırıyor.

(8-2) denklemi şu biçimde de yazılabilir:

$$V_{ODAC} = V_{REF}/2^n \cdot d \quad (8-4)$$

Son denklemde d ile ( $D_{n-1}G_{n-1} + D_{n-2}G_{n-2} + \dots + D_1G_1 + D_0G_0$ ) toplamıyla ifade edilen ikili vektörün onlu değeri tanımlanıyor.

Aşağıda verilen Şek. 8-5 a)'da üçbitli D/A dönüştürücünün idaelleşmiş iletim özelliği (karakteristiği) gösterilmiştir. Böyle dönüştürücü girişinde, 3 bitle  $2^3=8$  farklı kombinasyonu kodlanabileceğinden dolayı, bu sekiz olası kombinasyondan, 000,001,010,...,110,111, herhangi kombinasyonla tanımlanan dijital sinyal alabilir. Herbir kombinasyon için D/A dönüştürücünün çıkışında önceden verilmiş sınırlarda uygun gerilim seviye üretmesi gerekiyor. Çıkış analog sinyalinin yüksek olası seviyeyi tanımlayan kapsama ölçeğin tam (bütün) değeri denir ve FS veya FSR (İng.full scale range) ile işaretleniyor. Bu değer, A/D dönüştürücüye bağlı olan  $V_{REF}$  referans gerilime bağlıdır.



Şek. 8-5. a) 3-bitli, b) 4-bitli D/A dönüştürücünün idaelleşmiş iletim özellikleri

4-bitli D/A dönüştürücü söz konusu olunca, o zaman iletim özellikleri gösterilen Şek.8-5 b)'ye göre girişinde  $2^4=16$  farklı 4-bitli vektör meydana gelebilir. FS ölçeğin tam değeri iki durumda eşit olduğunu tahmin edersek, ikinci örnek için çıkış sinyali daha isabetli ve daha kaliteli olduğu görülüyor, çünkü tüm FS kapsamı iki kat fazla seviyeye ayrılıyor. Bununla ilgili olarak, D/A dönüştürücünün isabetlik ve kalite ölçüsü olarak çözünürlük (İng. Resolution) terimi tanımlanıyor. Çözünürlük dönüştürücünün sözcükten dönüştürebileceği bit sayısı n ile ilgilidir ve bununla beraber olarak D/A dönüştürücünün çıkışında elde edilebilen farklı gerilim seviyelerin toplam sayısı N ile ilgilidir. D/A dönüştürücünün girişinde n-bit uzunluğunda ikili sözcüğün girdiğini göz önüne alırsak, N farklı seviyelerin toplam sayısı bildiğimiz  $N=2^n$  denkleminde elde ediliyor.

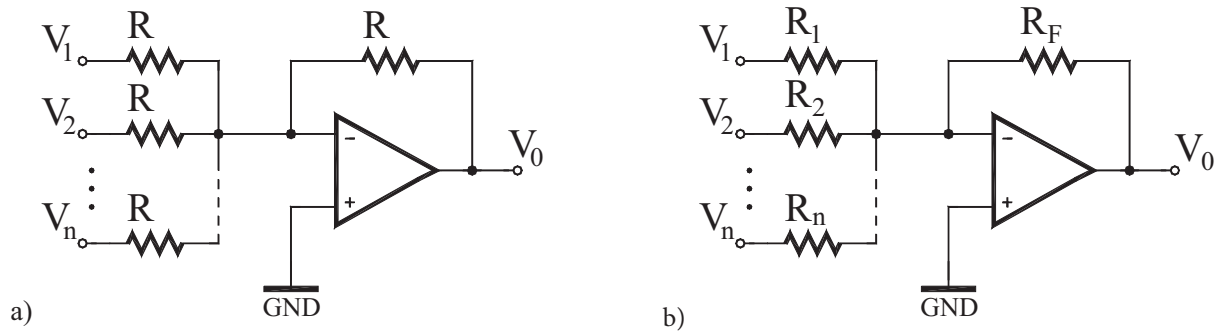
## 8.4. D/A DÖNÜŞTÜRÜCÜLER

D/A dönüştürücüler, aslında çıkış gerilimi fazla giriş gerilimin toplamını tanımlayan evirici kuvvetlendirici derecelere dayanan devrelerdir. Bu arada maksimum değeri önceden verilemiş referans seviyeye  $+V_{REF}$  sınırlı olan çıkış gerilimi, değerleri referans seviyeden tam kat daha küçük olan giriş ayrık gerilim seviyelerin toplamı olarak elde ediliyor. Böyle parçaların ikili tabanları vardır. Daha doğrusu onlar,  $+V_{REF}$  referans seviyenin, 2 sayısının tam üsleri ile katsayılarıdır, öyle ki D/A dönüştürücünün çıkış gerilimi  $V_{REF}/2, V_{REF}/4, V_{REF}/8, V_{REF}/16, \dots$  vs toplamı olarak elde ediliyor. Bu gerilim seviyesi D/A dönüştürücülerin büyük bölümünde, toplayıcının girişinde bağlı olan farklı değerli hassas rezistörlerden ağlar yardımıyla elde ediliyor. Uygun girişteki bit 0 ise girişlerde sıfırcı gerilim seviyesi meydana geliyor ya da girişteki bit 1 ise  $+V_{REF}$  meydana geliyor.

### 8.4.1. R/2<sup>n</sup>R AĞIRLIKLI DİRENÇ AĞLI D/A DÖNÜŞTRÜRÜCÜ

Ağırlıklı R/2<sup>n</sup>R ağı, ya da diğer adıyla ikili ağırlıklı direnç ağı D/A dönüştürücü, işlem kuvvetlendirici ile gerçekleşen toplayıcının iyi bilinen diyagramının bir çeşididir. Bu dönüştürücü, Şek. 8-6 a)'da gösterilen ilkesel elektrik diyagramına göre, birkaç giriş gerilimden uyarılan işlem kuvvetlendiricinin evirici yapılıdır. Toplayıcıda giriş sinyalleri rezistörler üzerinden getiriliyor ve bu arada her ayrı V<sub>i</sub> giriş geriliminin, V<sub>0</sub> çıkış gerilime eşit etkisi olması amacıyla rezistörlerin değerleri eşittir: R<sub>1</sub>=R<sub>2</sub>=...=R<sub>i</sub> = ...R<sub>n</sub>=R. Negatif geri bağlantının, giriş rezistörlerle aynı değeri olan R<sub>F</sub> rezistörüyle (R<sub>F</sub>=R) gerçekleştiğini göz önüne alarak, V<sub>0</sub> çıkışındaki gerilim, denklem (8-5)'e göre onların evirici şeklindeki toplamını tanımlayacak.

$$V_0 = -(V_1 + V_2 + \dots + V_n) = -\sum V_i \quad (8-5)$$



Şek. 8-6. İşlem kuvvetlendiricinin a) eşit rezistörlü b) rezistörlerin farklı değerleri ile toplayıcının elektrik diyagramları

Giriş dirençleri Şek. 8-6 b)'ye göre birbirinden farklıysa, bireysel gerilmelerin çıkış gerilimin üzerine etkiler de farklı olacak. Şöyle ki, çıkış gerilimin onların toplamı ile yeniden orantılı olacak, ancak toplamın doğru değerini belirlemeyecek. Ayrıca, geri bağlantıda R<sub>F</sub> rezistörün değişmesiyle kuvvetlendirici aşağıdaki denkleme göre kontrol ediliyor ve belirleniyor:

$$V_0 = -R_F \left( \frac{1}{R_1} V_1 + \frac{1}{R_2} V_2 + \dots + \frac{1}{R_n} V_n \right) \quad (8-6)$$

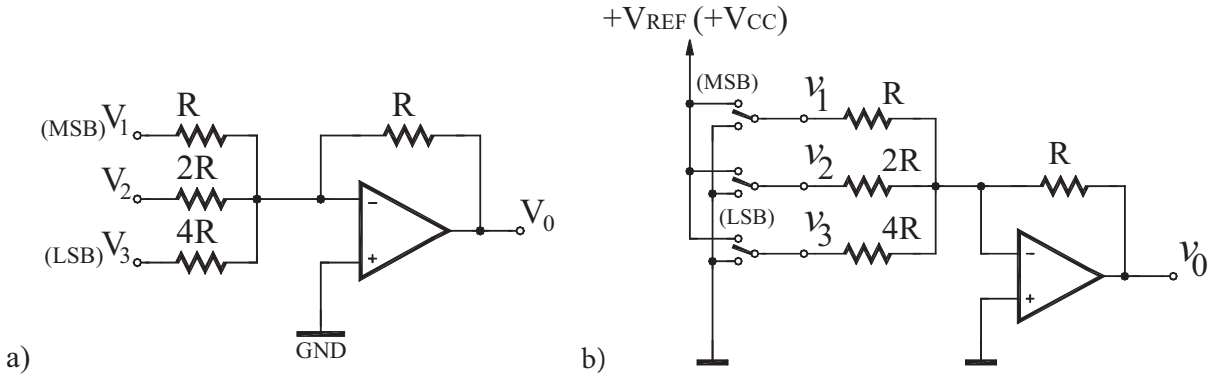
Toplayıcının bu özelliği D/A dönüştürücünün doğru çalışmasının temelidir. V<sub>1</sub>, V<sub>2</sub>, ..., V<sub>n</sub> giriş gerilim seviyelerden herhangi seviyesi, verilen pozisyonda bitin 1 veya 0 olmasına bağlı olarak yüksek (+V<sub>REF</sub>) veya alçak (0V) olabilir. Buradan aşağıdaki denklem elde ediliyor:

$$V_0 = -R_F \cdot V_{REF} \cdot \left( \frac{1}{R_1} D_1 + \frac{1}{R_2} D_2 + \dots + \frac{1}{R_n} D_n \right) \quad (8-7)$$

Son açıklama ile ilgili olarak, giriş rezistörlerin değerleri birbirinden en küçüğünden başlayarak en büyüğüne doğru iki kat farklı olmaları gereken evirici toplayıcı devrenin yaptığımızı tahmin edelim. Daha doğrusu, birinci gerilim V<sub>1</sub>'in getirildiği R<sub>1</sub> rezistörün R<sub>1</sub>=R değeri varsa, sıradaki rezistörün 2R değeri olmalıdır, her sıradaki rezistör önceki dirence kıyasen iki kat artıyor. Buna göre, üç girişli (n=3) devre, yani diyagramı Şek.8-7 a) veya b)'de gösterilmiş olan üç bitli D/A dönüştürücü söz konusu olunca, dirençler şöyle olacak: R<sub>1</sub>=2<sup>0</sup>R, R<sub>2</sub>=2<sup>1</sup>R<sub>1</sub>=2R ve R<sub>3</sub>=2<sup>2</sup>R<sub>1</sub>=R<sub>4</sub>

Bunun dışında, geri bağlantıda değeri  $R_i=R$  direncinden iki kat daha az olan  $R_F$  direnç seçelim, öyle ki  $R_F=R_i/2 =R/2$  geçerli olacak. Bu yüzden, (8-6) v (8-7) denklemlerine göre, çıkış gerilimi şöyle olacak:

$$V_0 = -\left(\frac{1}{2}V_1 + \frac{1}{4}V_2 + \frac{1}{8}V_3\right) \quad (8-8)$$



Şek. 8-7. R/2nR ağırlıklı direnç ağı üç bitli D/A dönüştürücü

(8-6) ve (8-7) denklemlerden, böyle dirençlerin seçimi, her giriş gerilimin ( $V_1$ ,  $V_2$  ve  $V_3$ ), (8-8) denkleminde göre çıkış gerilim üzerine önceki gerilime kıyasen tam yarım etkisi olmasını sağladığı ortaya çıkıyor. Başka bir deyişle,  $V_1$  giriş gerilimin en büyük etkisi olacak ve çıkış gerilime en yüksek referans gerilimin yarısıyla etkiliyor, sıradaki  $V_2$  gerilimi çıkışa birinci gerilimden yarım için daha az etkileyecek, yani  $V_{REF}$ 'in çeyreği kadar etkileyecek,  $V_3$  ise  $V_2$ 'den bir yarım için daha az etkileyecek, yani sadece sekizde bir orantı ile etkileyecek. Dirençlerin böyle değerleri ve çıkış gerilim üzerindeki etkileri tesadüfen seçilmemiş çünkü bu gerilimler (8-3) denklemini gerçekleştirmek amacıyla doğal ikili sisteminde bitlerin uygun yerlerine uygun ağırlık ilişkileri tanımlıyorlar. Bu analize göre, Şek.8-7 a) ve b)'deki diyagramda en küçük direncin bağlı olduğu girişte en büyük ağırlıklı bitin,  $2^{n-1}$  (MSB) getirilmesi gerekiyor, en büyük direncin bulunduğu girişte ise en az ağırlıklı bit  $2^0$  (LSB) bağlanmalıdır. Son olarak, devre girişinde üç mantıksal devrenin dijital çıkışı getirirsek, yani üç bittten herhangi kombinasyonu getirirsek, Volt ile ifade edilmiş çıkış gerilim, aşağıdaki denkleme göre bu üç bitin ikili değerinin onlu şekilde analog temsilcisi olacak:

$$V_0 = -V_{REF} \cdot \left(\frac{1}{2}D_2 + \frac{1}{4}D_1 + \frac{1}{8}D_0\right) \quad (8-9)$$

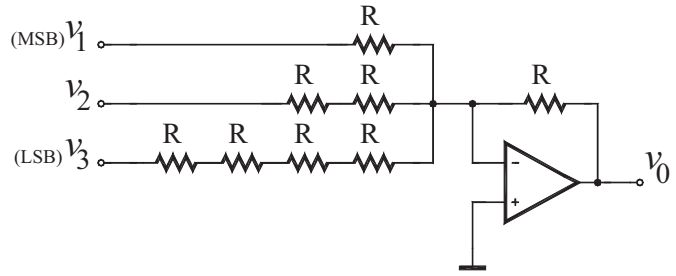
Son (8-9) denkleminde,  $V_{REF}$  mantıksal 1 seviyesini tanımlıyor,  $V(1)=V_{REF}$ ,  $D_2$ ,  $D_1$  ve  $D_0$  ise değerleri 0 veya 1 olabilen giriş bitleridir. Bu arada MSB,  $V_1$  girişine getirilen  $D_2$ 'dir, üçüncü  $V_3$  girişini uyarın LSB ise  $D_0$  dir.

Genel olarak, n giriş (bit) için, bu D/A dönüştürücünün denklemini aşağıdaki şekilde de yazılabilir.

$$V_{ODAC} = -R_F/R_0 \times V_{REF} \times d = -R_F/R_0 \times V_{REF} \times (2^{n-1}D_{n-1} + \dots + 2^0D_0);$$

Devrenin girişinde meydana gelen, 000'dan 111'e kadar tüm sekiz üç-bitli kombinasyondan her biri için çıkış gerilim değerinin doğru hesaplanması için (8-9) denklemini uygulayacağız ve tab.8-1'i elde edeceğiz. Örnekte, D/A devircinin referans seviyesi olarak 5V'luk besleme tahmin edilmiştir ( $V_{REF}=+5V$ ). Bu yüzden  $V(1) = +5V$  ve  $V(0)=0V$  geçerli olacak.

Onlu değer	İkili eşdeğer	Çıkıl gerilimi [V]
0	000	0.000
1	001	- 0.625
2	010	- 1.250
3	011	- 1.875
4	100	- 2.500
5	101	- 3.125
6	110	- 3.750
7	111	- 4.375



Şek.8-8. Eşit dirençli  $R/2^nR$  ağırlıklı direnç ağı üç bitli DAC

Tab.8-1. Üç bitli DAC'de eşdeğerli değerler

Verilen analiz genişletilebilir ve D/A dönüştürücünün dirençlerinden birinin bilinen değerine dayanarak diğer dirençler için doğru seçim yapılabilir. Bu arada, eğer:

- ⊕ LCB en düşük ağırlıklı bitteki en büyük direnç  $R_0$  biliniyorsa, o zaman geri bağlantıdaki  $R_F$  direncin  $R_F = R_0/2^n$  değeri olması gerekiyor;
- ⊕ MSB en yüksek ağırlıklı bitte en küçük  $R_{n-1}$  direnci biliniyorsa, o zaman geri bağlantıdaki  $R_F$  direncin  $R_F = R_{n-1}/2$  değeri olması gerekiyor;
- ⊕ geri bağlantı direnci  $R_F$  biliniyorsa, o zaman en düşük ağırlıklı LSB bitte en büyük  $R_0$  direncin  $R_0 = 2^n \cdot R_F$  değeri olması gerekiyor.

Giriş dirençlerin değerleri her sıradaki direnç en küçüğünden  $R_{n-1}$  başlayarak en büyüğüne  $R_0$  doğru her durumda iki kat artması gerektiğini unutmamalıyım.

D/A dönüştürücünün daha isabetli olması aranırsa, bitlerin sayısı  $n$  ve bununla beraber girişlerin sayısı da artmalıdır. Bu sırada, her yeni eklenen dijital girişine birer ek direnç eklenecek. Bu ek direncin, LSB-bitin getirildiği dirençten iki kat daha büyük değeri olacak, ya da ağda yer alan en büyük dirençten iki kat daha büyük direnç ekleniyor. Bu koşul bu tür dönüştürücünün olumsuz tarafını ortaya koyuyor çünkü farklı değerlerle isabetli dirençlerin sayısı giderek artıyor.

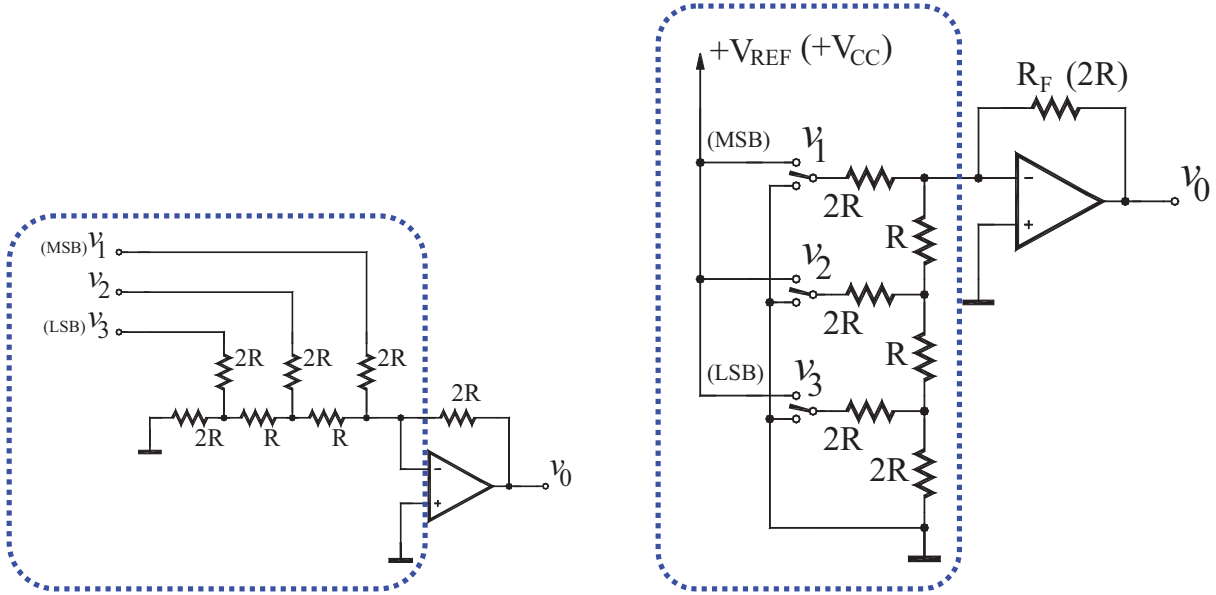
Ağırlıklı  $R/2^nR$  direnç ağı D/A dönüştürücü, her giriş için ayrıdan çok sayıda aynı değeri olan dizisel şekilde bağlanmış dirençlerle gerçekleştirilebilir. Bu arada, Şek 8-8'e göre her yeni girişe, önceki girişten iki kat fazla direnç bağlanmalıdır. Böyle bağlanma şekli yukarıda yaşanan sorunu çözüyor ancak yeni bir sorun ortaya çıkıyor, o da direnç ağının çok daha karmaşık yapılandırma olmasıdır.

#### 8.4.2. R/2R BASAMAKLI DİRENÇ AĞLI D/A DÖNÜŞTÜRÜCÜ

Yukarıda incelediğimiz ikili ağırlıklı ağı D/A dönüştürücünün en büyük dezavantajı, büyük sayıda farklı ve giderek daha yüksek değerleri olan isabetli ve benzersiz dirençler gereksinimidir. Bu sorunu çözmek için pratikte sıkça topolojisi basamağa benzeyen oldukça etkili bir çözüm kullanılıyor.

Bu yüzden, bu şekilde uygulanan dönüştürücü basamaklı ağılı dönüştürücü ismini kazanmış, kullanılan dirençlerin değerleri yüzünden popüler olarak R/2R D/A dönüştürücü olarak da adlandırılıyor (İng. R/2R Ladder DAC).

Bu yapılandırmada da yine toplayıcı kullanılıyor, ancak burada girişinde Şek. 8-9 a) veya b)'ye göre sadece iki değeri olan dirençler kullanan direnç ağı oluşuyor. Ağırlıklı ağılı D/A dönüştürücü ile kıyaslama yapılırsa dirençlerin sayısı bu durumda az ölçüde arttığı görülüyor.



Şek. 8-9. a) b) basamaklı R/2R direnç ağılı üç bitli D/A dönüştürücü

Uygulanan basamaklı ağılı matematisel analizi, önceki durumdan daha karmaşıktır ve Tevenen teoremi kullanılmalıdır. Burada elde edilme şeklini incelemeyi sonucunu kullanacağız. Bu sonuç ağırlıklı R/2nR ağılı diyagramın sol tarafı  $R_T$  iç dirençli karşılıklı Tevenen kaynağı  $V_T$  ile değiştirebileceğini gösteriyor. Bu değerler önceki ağırlıklı ağılı D/A dönüştürücüdeki değerlere eşittir.

$$V_T = \frac{1}{2}V_1 + \frac{1}{4}V_2 + \frac{1}{8}V_3 \quad \text{ve} \quad R_T = R \quad (8-10)$$

Böylece, tab. 8-1'e uygun çıkış gerilimin seviyelerine ilişkin sonuç denklemi de elde ediliyor.

$$V_0 = -\frac{R_F}{R_T}V_T = -\frac{R_F}{R_T}\left(\frac{1}{2}V_1 + \frac{1}{4}V_2 + \frac{1}{8}V_3\right) = -\frac{R_F}{R}\left(\frac{1}{2}V_1 + \frac{1}{4}V_2 + \frac{1}{8}V_3\right) \quad (8-11)$$

$R_F$  için  $R_F=R$  seçersek, o zaman çıkış gerilimi için şu ifade elde ediliyor:

$$V_0 = -\left(\frac{1}{2}V_1 + \frac{1}{4}V_2 + \frac{1}{8}V_3\right) = -\left(\frac{1}{2} \cdot D_2 + \frac{1}{4}D_1 + \frac{1}{8}D_0\right) \cdot V_{REF} \quad (8-12)$$

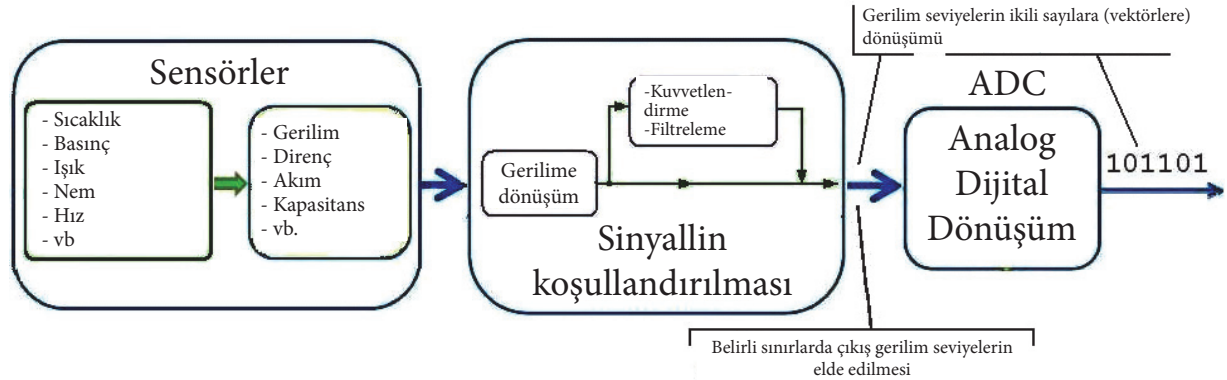
$V_1$ ,  $V_2$  ve  $V_3$  giriş gerilimlerinin değerleri, uygun bitin 0 ya da 1 olduğuna bağlı olarak sadece alçak seviyede (0V) ya da yüksek seviyede ( $+V_{REF}$ ) olabildiğinden dolayı, önceki bölümden bildiğimiz (8-9) denkleminin eşit ifade elde ediyoruz.



## 8.5. ANALOG-DİJİTAL DÖNÜŞÜM

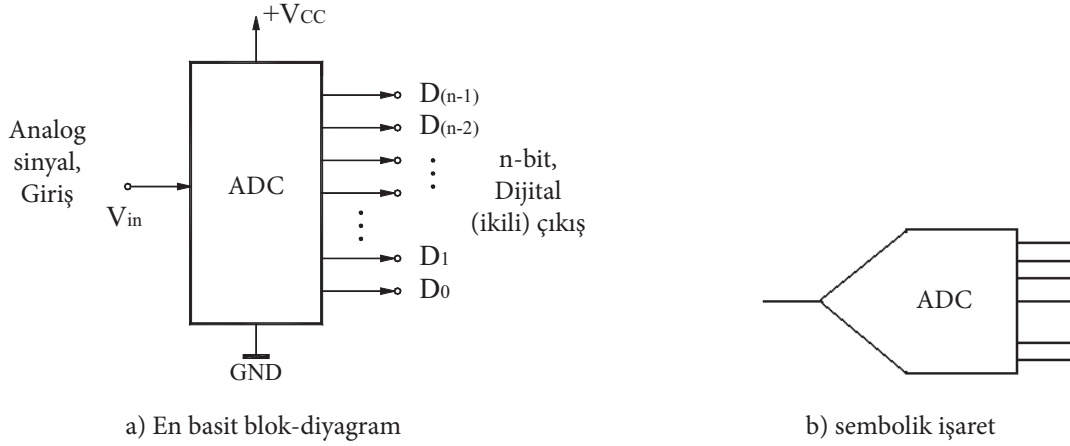
Dijital sistemler dijital sinyellerle, dijital şekilde ikili sözcükler olarak ifade edilen verilerle çalışıyorlar. Her sözcük sistemin giriş olarak kabul ettiği sonlu sayıda bitler kombinasyonu tanımlıyor ve dış dünyadan belirli bir bilgiyle ilgilidir. İnsanlar arasında iletişim konusunda, farklı şekillerde kurulu, ancak her zaman cümlelerin ve sayıların dijital şekilde dönüşüm, daha doğrusu kodlama araması var. Burada farklı grafiksel semboller (rakamlar ve sayılar) ve işaretler (İng. characters) kombinasyonu olarak elde edilen bilgiler söz konusudur. Hesaplamaların gerçekleştiği sırada aritmetik ve mantıksal işlemler işaretleriyle kombinasyonda onlu rakamlar kullanılıyor, kelimeler ve tümceler için noktalama işaretlerle kombinasyonda harfler kullanılıyor. Böyle verilerin dijital şekile dönüşmesi oldukça basittir çünkü hem onlu sayı sistemi hem alfabeler sonlu sayıda rakamlar, harfler ve işaretler kullanıyor. Buna göre, bu veriler doğal ikili sistemin, ya da başka bir ikili sistem veya kod uygulanarak dönüşen ayrık büyüklüklerdir.

Fakat, örneğin ses, ışık, basınç, sıcaklık, hız ve çok sayıda başka doğal olaylar ve süreçler sürekli büyüklüklerdir. Şöyle ki bu büyüklükler zamanla sürekli olarak değişiyor ve bu yüzden herhangi değer alabilirler. Bunun dışında, her doğal olayın kendi özellikleri var ve buna uygun olarak bu özelliklerin ölçüldüğü ayrı birimleri vardır, öyle ki her sürekli sinyalin dijital şekilde işlenebilmesi için analog elektrik büyüklük, gerilim ya da akıma dönüşmesi ve ardından dijitalleşmesi gerekiyor. Bilgisayarda veya başka dijital sistemde işletmek için uygun olan sürekli sinyallerin dijital sinyallere dönüşüm sorunu oldukça karmaşık bir işlemdir ve onu veri toplama (edinme) sistemi yapmalıdır (İng. data acquisition system). Bu sistemin basitleştirilmiş blok diyagramı Şek.8-10'da gösterilmiştir.



Şek. 8-10. Veri toplama sistemin blok-diyagramı

Uyumlaşma birkaç adımda yapılıyor. Sürekli olay ona akım, direnç ve benzeri gibi uygun analog büyüklüğüne dönüşmeyle başlıyor, ardından onun gerilime dönüşüm yapılıyor ve gerekirse onun koşullandırılması yapılıyor ve en sonunda o gerilimin dijital (ikili) şekile dönüşmesi yapılıyor. Sıcaklık, ışık, basınç vb. fiziksel büyüklüklerin elektirik büyüklüklere dönüşmesi sensör (İng. sensor) elemanlar yardımıyla yapılıyor. Koşullandırma farklı seviyelerde gerilim sinyalin kesin belirlenmiş sınırlar içinde iyileştirilmesi ve uyumlaşmasıdır. Son adım analog gerilim sinyalim dijital sinyale dönüşümdür. Bu dönüşümü en önemli blok yapıyor, o da Şek.8-11 a)'da blok diyagramı gösterilmiş, Şek.8-11 b)'de ise sembolik işareti verilmiş A/D dönüştürücüsüdür.



Şek. 8-11. A/D dönüştürücü

A/D dönüştürücü girişinde, zamanla sürekli değişen amplitütlü  $V_{inA}$  analog gerilim alıyor, çıkışlarda ise ikili kodlanmış sözcükler (vektörler), daha doğrusu giriş gerilimden (örnekler, örneklemeler) mevcut (seçilen) değerlere uyan, sonlu uzunlukta belirli bitler kombinasyonları (dizileri) üretiyor. Analog sinyalin amplitüdü sınırlı olmalıdır ve izin verilen en küçük ve en yüksek giriş seviyesi ( $V_{AMAX}$  ve  $V_{AMİN}$ ) ile belirlenen sınır içinde olmalıdır. Pratikte sıkça rastlanan A/D dönüştürücüleri için giriş kapsamı pozitif olabilir ve o zaman  $V_{AMAX}=V_{REF}$  ve  $V_{AMİN}=0V$  geçerlidir ya da simetrik belirlenmiş kapsam sınırlarında giriş sinyali hem pozitif hem negatif olabilir ve bu durumda  $V_{AMAX}=+V_{REF}$  ve  $V_{AMİN}=-V_{REF}$  geçerlidir. Bu seviyeler aslında sıkça olarak dönüştürücünün beslemeleridir.

## 8.6. TEMEL TERİMLER VE KAVRAMLAR

A/D dönüşüm süreci iki aşamada gerçekleşiyor. İlk önce, birinci aşamada analog sinyal zamana göre ayrıklaştırılıyor, ardından ikinci aşamada seviyeye (amplitüde) göre ayrıklaştırılıyor, yani nicemleniyor ve kodlanıyor.

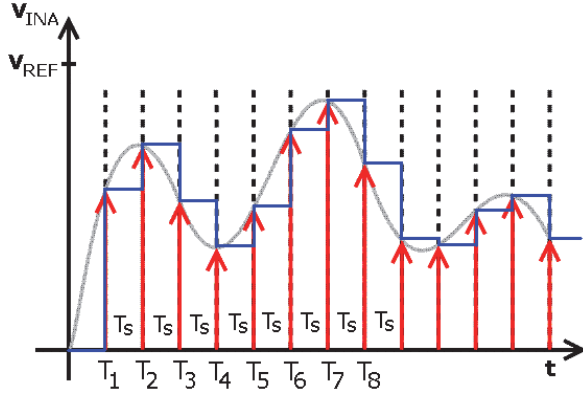
**Zamana göre ayrıklaştırma** örneklemeler (örnekler) alma sürecidir, daha doğrusu  $T_s$  periyodlu belli zaman aralıklarda, ya da  $f_s$  belirli seçme frekansla, sinyalden belirli gerilim seviyeleri alma sürecidir. Bu yüzden seçme ya da sempleme (İng. sample) terimi de kullanılıyor.

**Seviyeye göre ayrıklaştırma ya da** (pratikte en çok kullanılan terim) **nicemleme**, sürekli (sonsuz isabetli, doğru) değerden her örnekleme amplitüdünü (seviyesini), belirli sonlu isabetli ayrık gerilim seviyenin değerini tanımlayan belli sayıya dönüştürüyor.

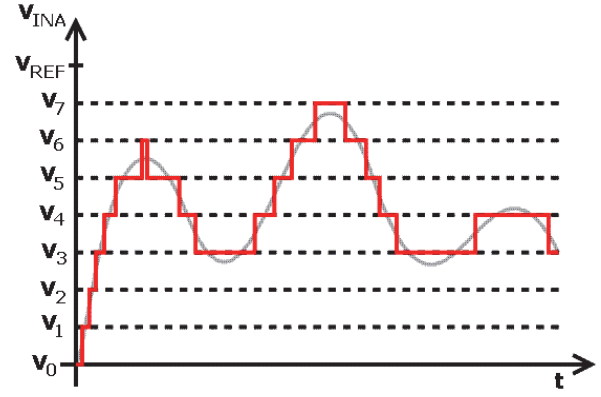
Nicemlemeden sonra belirli onlu değeri ( $d$ ) olan her ayrık seviye, belirli koda göre ayrı ikili sözcüğe (vektöre) ya da sonlu uzunlukta ( $n$ ) bitler kombinasyonuna dönüşüyor. Tüm kelimeler kümesi, farklı kodlama teknikleriyle elde edilebilen dijital sinyalin son şeklini tanımlıyor. Tek kutuplu kodlama doğal ikili sayı sistemiyle ilgilidir ve giriş analog gerilim sinyalin pozitif kapsamı durumunda uygulanıyor. Giriş gerilimin simetrik kapsamı söz konusu olunca iki kutuplu kodlama uygulanıyor. İki kutuplu kodlama bazı işaretli ikili sisteme göre yapılıyor, örneğin ikinci tümleyen (RC sistemi).



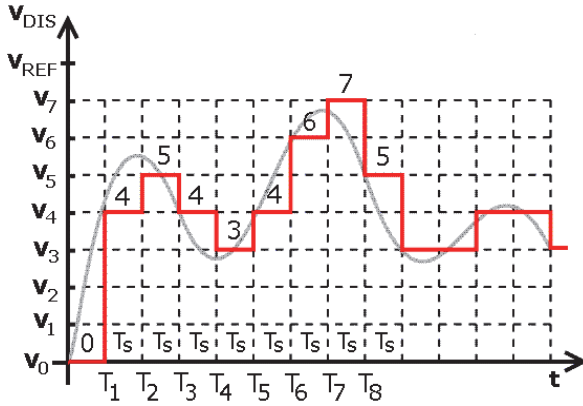
Sıradaki şekillerde, Şek. 8-12 a), b), c) ve ç), giriş analog sinyali için bir örnek verilmiştir ve sırasıyla onun zamana göre ayrıklaştığı zaman, sekiz seviyede seviyeye göre ayrıklaştığı (nicemlendiği) zaman, hem zamana hem seviyeye göre ayrıklaştığı zaman görünüşü ve dijital sistem tarafından işlenebilen bitler dizisi olarak dijital sinyalin son görünüşü gösterilmiştir. Her örnekleme üç bitle kodlanan sekiz olası seviyeden biri seviyesi vardır. Bununla kod sözcüğün uzunluğu da belirlenmiş oluyor.



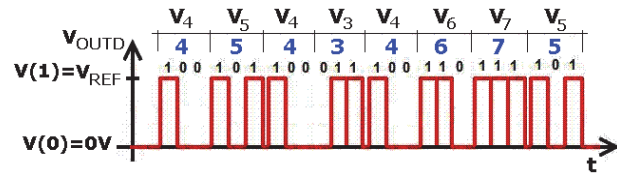
Şek. 8-12. a) zamansal ayrıklaştırma - giriş analog gerilimden örnekleme almak (zaman aralıkları ayrıktır, gerilim seviyeleri ise sürekli)



Şek. 8-12. b) Seviyeye göre ayrıklaştırma ya da Nicemleme - Gerilim kapsamını sonlu sayıda gerilim seviyelerine ayırmak (zaman aralığı sürekli, gerilim seviyeleri ise ayrıktır)



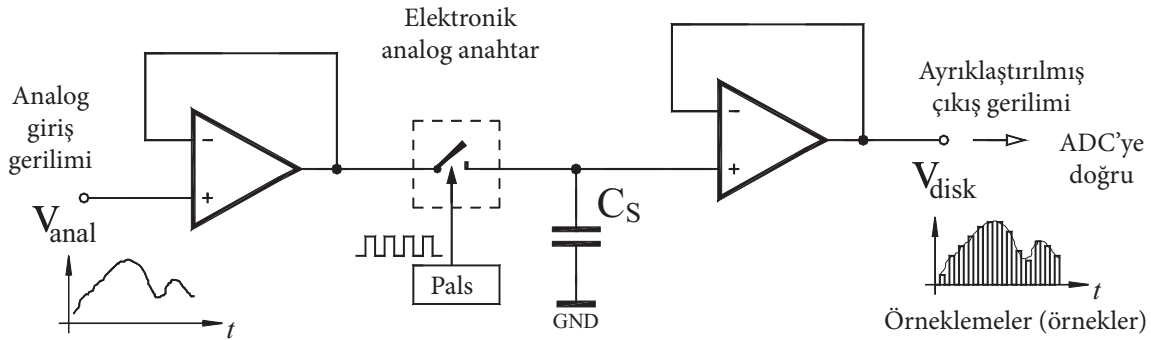
Şek. 8-12. c) Ayrıklaşmış ve nicemlenmiş sinyal - ayrıktır zaman aralıkları ve gerilim seviyelerinin ayrıktır değerleri



Şek. 8-12. ç) Dijital gerilim seviyesi - A/D dönüşüm süreciyle elde edilen giriş analog gerilime karşılıklı sinyal

Şek. 8-13'te ayrıklaşma sürecinin ya da bir gerilim sinyalinden örnekleme alma sürecinin blok-diyagramı gösterilmiştir. Giriş analog gerilimi, tek gerilim kuvvetlendirici ile gerilim iletici üzerinden, A/D dönüştürücü girişindeki örnekleme (seçme) ve tutma (İng. Sample-and-Hold, SH) devresine getiriliyor. Bu devre, en basit olarak, analog anahtar ve sığaç (kondansatör) kombinasyonudur. Anahtar, açılmasını ve kapanmasını kontrol eden  $T_s$  pals dürtülerin periyodunun süresine bağlı olan  $f_s$  örnekleme frekansına uyumlu olarak periyodik şekilde açılıyor ve kapanıyor. Palsın bir yarı periyodu sırasında anahtar kapalıdır ve o zaman giriş geriliminin bir seviyesi seçilerek zamana göre ayrıklaştırma gerçekleşiyor. Sıradaki yarı periyotta anahtar açılıyor ve kondansatör anahtarının açılmasından hemen önce kondansatörde bulunan ve o seviyeye kadar doldurulan giriş geriliminin mevcut değeri korunuyor

Anlattıklarımıza göre, kondansatörde ve dolayısıyla birincisinde olduğu gibi tek gerilim kuvvetlendirmesi olan ikinci gerilim ileticinin çıkışında zamansal ayrıklaştırılmış basamak şeklinde, ardışık örneklemeler dizisinden oluşan sinyal elde ediliyor. Örneklemelerin amplitütleri devamda seviyeye göre ayrıklaşmalı, yani nicemlenmelidir.



Şek. 8-13. Analog sinyalden örneklemeler (örnekler) alma süreci

Anahtarın yeniden kapanması için geçen süre (kondansatörde ve gerilimi ileticisinin çıkışında bulunan) mevcut örneklemenin gerilim seviyesinin ilerdeki işletilemsi amacıyla iletilmesi için, daha doğrusu nicemlenmesi için kullanılıyor. Bununla ilgili olarak, dönüşüm döngüsü, A/D dönüştürücünün giriş analog sinyalden alınan gerilim örneklemesi seviyesinin ayrık değerini belirlemek için, yani dönüştürücünün giriş sinyalinin, çıkışta uzunluğu  $n$ , gereğe göre, ancak genelde 4 ile 16 bit arasında olan sözcük (ikili vektör, kombinasyon) tanımlayan uygun dijital şekline dönüştürmesi için gereken zaman aralığıdır.

Örneklemesi frekansı  $f_s$  ve analog giriş sinyalin seviyesinin değişme hızı, yani A/D dönüşüm sürecinde en büyük frekans kapsamı  $f_{MAX}$  arasında, ters olan D/A dönüşüm sürecinde analog sinyalinin yeniden oluşması sırasında biçim bozukluğunu engellemek amacıyla basit ilişki vardır. Daha doğrusu örneklemesi teoremi (İng. *Sampling Theorem*) geçerli ise analog sinyali alınan örneklemelerden tamamiyle yeniden oluşabilir. Örneklemesi teoremine göre, biçim bozukluğu ve gürültünün meydana gelmemesi için, tayfında en yüksek frekansı  $f_{MAF}$  olan sürekli sinyalden, Nyquist kriteri olarak bilinen (8-13) denklemine göre  $f_{MAX}$ 'tan en az iki kat daha yüksek olan  $f_s$  frekansla örneklemelerin alınması gerekiyor.

$$f_s \geq 2f_{MAX} \quad (8-13)$$

Örneklemesi frekansı  $f_s$  yukarıdaki koşulu yerine getirmiyorsa, ya da ters  $f_s < 2f_{MAX}$  koşulu geçerli ise sahte frekansların (İng. *aliasing*) meydana gelmesine yol açılacak. Bu durum yeniden oluşmuş sinyalin, orijinel sinyalde olmayan, ancak giriş sinyalinden gerçek (var olan) bir frekansın yansıma sonucu olarak meydana gelen frekansın ortaya çıktığı olaydır.

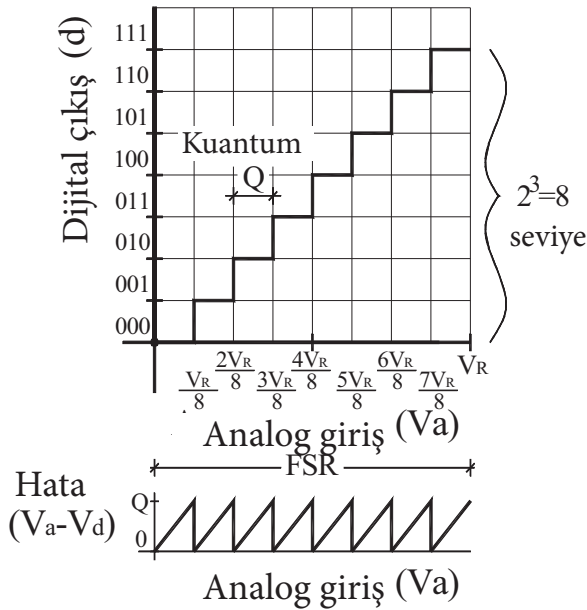
Şimdiye kadar açıkladıklarımızdan şu sonuca varabiliriz: D/A dönüştürücünün doğru çalışması için dönüşüm döngünün süresi, önceki örneklemesi dönüştürülmeden yeni örneklemesi meydana gelmesinin önlenmesi için örneklemesi periyodu olan  $T_s$ 'ten daha kısa olmalıdır .

Nicemlemeye gelince, bu süreç devamda daha büyük detaylarla açıklanacaktır.

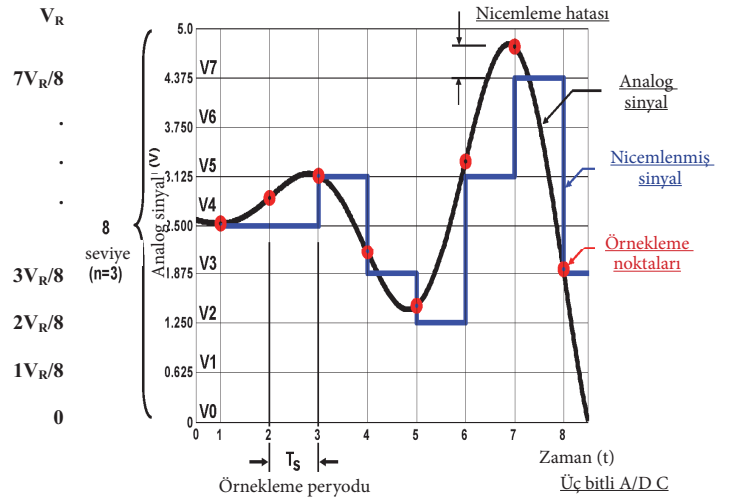
## 8.7. KARAKTERİSTİK PARAMETRELER VE İLETİM ÖZELLİĞİ

Şimdiye kadar söylediklerimizden, A/D dönüştürücünün nicemleme süreci sırasında bir TS pals aralığı süresinde, verilen referans gerilim seviyesi  $V_i$  etrafında iki yan yana referans gerilim seviyenin ( $V_i; V_{i+1}$ ) ve  $i=0,1,2,\dots,n$  arasında bulunan giriş analog sinyalinin (gerilim değerler kümesi)  $Q$  gerilim aralığını, tek ayrık değer gibi getirdiğini söyleyebiliriz. Her sıradaki referans seviyesi  $V_{i+1}$  önceki referans seviyesi  $V_i$ 'den aynı değer için daha büyüktür. Bu değere nicemleme adımı ya da kuantum (nicem) denir ve  $Q$  ile işaretlenerek,  $V_{i+1}=V_i+Q$  geçerlidir.

A/D dönüşüm süreci en sıkça, Şek.14'te olduğu gibi, basamaklı şekilde iletim fonksiyonunu gerçekleştiren devre aracılığıyla gerçekleşiyor. Şek.8-14'te üç bitle A/D dönüşüm sürecinin gerçekleştirilmesini sağlayan belirli iletim özelliği (karakteristiği) gösterilmiştir. (A/D dönüşüm sürecinin gerçekleşmesi için şekli Şek.8-14'te gibi olan iletim fonksiyonu gerekir). Nicemleme sürecini daha detaylı açıklamak için, Şek.8-15'te analog giriş sinyali ve verilen iletim özelliği için uygun ayrılmış çıkış sinyali için bir örnek verilmiştir.



Şek. 8-14. Üç bitli A/D dönüştürücünün idealleşmiş iletim özelliği



Şek. 8-15. Analog giriş sinyali ve uygun çıkış ayrılmış çıkış sinyali

Verilen şekillerden analog sinyalin amplitüdü iki seviye arasında sınırlı olduğu görülüyor: en alçak sıfır gerilim seviyesi  $V_0=0V$  ve en yüksek referans gerilim seviyesi  $+V_{REF}$ . Bu yüzden bu kapsama tam ya da bütün basamak (ölçek) kapsamı denir ve FS ya da FSR (İng.Full Scale Range) ile işaretleniyor. Genel olarak, FSR tam ölçeği giriş analog gerilimin alabildiği en büyük ( $V_{INmax}$ ) ve en küçük ( $V_{INmin}$ ) değer arasındaki farktır, ya da  $FSR = V_{INmax} - V_{INmin}$ . Giriş gerilimi, bizim durumda olduğu gibi pozitif ise ve  $0V$  ile belirli referans seviyesi  $+V_{REF}$  arasındaki sınırdaki değişiyorsa, o zaman son denklem basitleşiyor ve şu şekli alıyor:  $FSR = V_{REF} - 0 = V_{REF}$ . Şek.8-14 ve Şek.8-15'teki örnekte referans seviyesi  $V_{REF} = +5V$ , en alçak gerilim seviyesi ise  $0V$  olduğundan dolayı tam ölçek  $FSR = 5V$  olacak.

Bunun dışında, şekillerden üç bitli A/D dönüşüm söz konusu olduğundan dolayı tam kapsamın sekiz gerilim kapsamına ayrıldığını görebiliriz. Şöyle ki, gerilim kapsamların sayısı  $N_U$  dönüşümün yapıldığı bitler sayısı ne bağlıdır ve bildiğimiz denklemle belirleniyor:

$$N_U = 2^n \quad (8-14)$$

İncelediğimiz örnekte  $n=3$ , öyle ki  $N_U=2^3=8$ 'dir. Böylece, tam ölçek FS değeri gerilim kapsamların toplam sayısı  $N_U$  ile bölünerek, her gerilim kapsamın  $Q$  gerilim aralığının genişliği elde edilebilir.

$$Q = FSR/N_U = FSR/2^n \quad (8-15)$$

Bizim örneğimizde  $Q = 5V/2^3 = 5V/8 = 0,625V$ .

Bu gerilim aralığında bulunan tüm gerilimler, belirli aralığın nerde bulunduğuna bağlı tek bir değerle değiştiriliyor. Şöyle ki, bizim örneğimizde, 0 ve 0,625V' tan daha küçük giriş gerilim seviyesi arasında bulunan tüm giriş gerilimler 000 olarak kodlanan  $V_0=0V$  sıfıncı referans seviyesine dönüşecek, 0,625 ve 1,250V arasındaki giriş gerilimleri ise 001 olarak kodlanan birinci referent seviyesine  $V_1=0,625V$  dönüşüyorlar vs. 4,325V ile 5V arasındaki kapsam son, yedinci referans seviyesine  $V_7=4,325V$ 'a dönüşüyor ve 111 olarak kodlanıyor.  $V_0$  sıfıncı referans seviyeden başlayarak son seviyeye kadar  $V_{n-1}$  (örneğimizde  $V_{n-1}=V_7$ ), aradaki gerilim farkı  $Q$  kapsamına eşit olduğu  $N_Q$  referans gerilim seviyelerin sayısı  $N_U$  kapsamların sayısına  $N_U$  eşit olduğu görülüyor ( $N_Q=N_U=2^n$ ). Her  $V_i$  gerilim seviyenin onlu değeri ve volt ile ifade edilen değer arasındaki ilişki aşağıdaki denkleme göre hesaplanıyor:

$$V_i = i \cdot \frac{V_{REF}}{N_Q} = i \cdot \frac{V_{REF}}{2^n}, \text{ ku } i = 0, 1, 2, (n-2), (n-1) \quad (8-16)$$

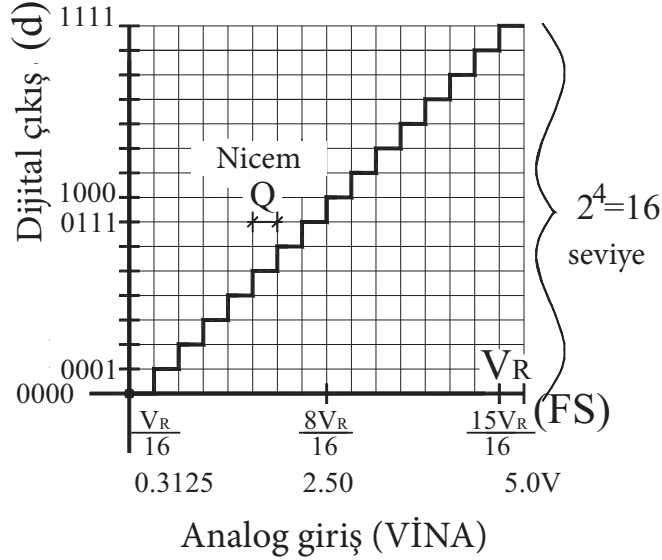
Ayrı referans gerilim seviyelerini birbirinden ayıran gerilim aralığının büyüklüğüne nicem (kuantum) denir ve  $Q$  ile işaretleniyor. İncelediğimiz örnekte,  $V_{REF}=+5V$  referans gerilimi ve iç bitli A/D dönüşüm için, nicemin büyüklüğü tab. 8-2'den görüldüğü gibi  $1/8 \cdot V_{REF} = 0,625V$ 'tur.

Analog gerilim aralığı	Gerilim farkı	Referans gerilim seviyesi [V]	Onlu değer	İkili karşılık
$V_0 - V_1$ 0.000 – 0.625	Q	$V_0=0V_R/8$ 0.000	0	000
$V_1 - V_2$ 0.625 – 1.250	Q	$V_1=1V_R/8$ 0.625	1	001
$V_2 - V_3$ 1.250 – 1.875	Q	$V_2=2V_R/8$ 1.250	2	010
$V_3 - V_4$ 1.875 – 2.500	Q	$V_3=3V_R/8$ 1.875	3	011
$V_4 - V_5$ 2.500 – 3.125	Q	$V_4=4V_R/8$ 2.500	4	100
$V_5 - V_6$ 3.125 – 3.750	Q	$V_5=5V_R/8$ 3.125	5	101
$V_6 - V_7$ 3.750 – 4.375	Q	$V_6=6V_R/8$ 3.750	6	110
$V_7 - V_R$ 4.375 – 5.000	Q	$V_7=7V_R/8$ 4.375	7	111

Tab. 8-2. Şek. 8-14'teki iletim özelliğine göre A/D dönüşümünde gerilim seviyeleri

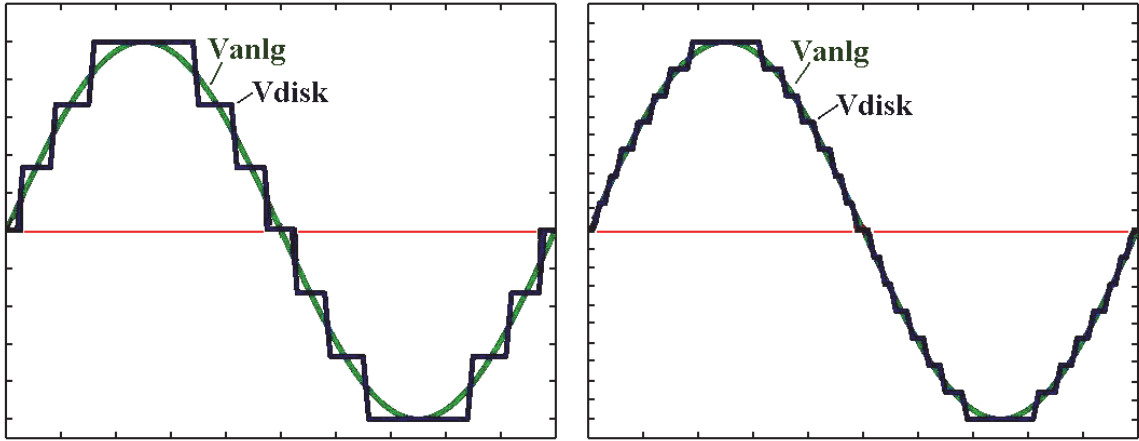
Şimdiye kadar incelediklerimizden, bit sayısı  $n$  artarsa nicemin büyüklüğü azalacağı, dolayısıyla referans seviyelerin sayısı artacağı ve nicemleyicinin isabetliği artacağı açıkça görülüyor çünkü daha küçük gerilim kapsamı tek bir gerilim seviyeye dönüşüyor.

Böylece, örneğin  $n=4$  bitle A/D dönüştürme uyguluyorsak, referans seviyelerin sayısı üç bitli dönüştürme kıyasına iki kat daha büyük olacak  $N_U=2^4=16$ ,  $V_0=0V$ 'dan  $V_{15}=4,6875V$ 'a kadar, Şek.8-16'da verilen iletim özelliğine göre nicem ise iki kat daha küçük olacak,  $Q=5V/2^4 = 5V/16=0.3125V$  olacak ve dolayısıyla nicemleyicinin isabetliği de artacaktır.



Şek. 8-16 Dört bitli A/D dönüştürücünün idealleşmiş iletim özelliği

Böylece, basit periyotlu sinyalin bir periyodunu alırsak ve bir kez üç bitli A/D dönüştürücüye ayrıklaştırıyoruz, bir kez ise dört bitli A/D dönüştürücüyle ayrıklaştırarak Şek.8-17 a) ve Şek.8-17 b) ile işaretlenmiş iki şekli elde ediyoruz.



Şek. 8-17. a) üç bit ile b) dört bit ile basit periyotlu sinyalin nicemlenmesi

Şek. 8-17 a) ve b) şekillerinden ikinci durumda kalitenin çok daha iyi olduğunu görebiliriz. Buna göre nicemleme için daha büyük sayıda bitler kullanılıyorsa isabetlik daha büyüktür. Bununla ilgili dönüştürücünün çözünürlüğü (İng. resolution) terimi tanımlanıyor ve herhangi giriş ayrık gerilim seviyesinin kodlandığı ikili vektörün bitler sayısı  $n$  (sözcük uzunluğu) olarak ifade ediliyor.

Buradan çözünürlüğün, FSR giriş değerlerin verilmiş tam gerilim kapsamı için A/D dönüştürücünün üretebildiği ayrık değerlerin toplam sayısını gösterdiğini elde ediyoruz.

Çözünürlüğün değeri, FSR giriş analog kapsamın tam ölçeği ayrık gerilim seviyelerin toplam sayısı ile ( $N_Q$ ) aşağıdaki denkleme göre ayrılınca elde ediliyor.

$$Q = \frac{FSR}{N_Q} = \frac{FSR}{2^n} \quad (8-17)$$

Çözünürlük aslında A/D dönüştürücünün meydana gelmesi çıkış dijital sinyalinde bitlerle kodlanmış çıkış seviyesinde değişikliğe yol açan en küçük gerilimi belirtiyor, o da en az ağırlıklı LSB bitin değişikliği demektir. Buradan çözünürlüğe nicem, LSB gerilim, çözünürlük gerilimi ya da nicemleme adımı da denir ve Q dışında  $\Delta$  ile de işaretleniyor.

Açıkladıklarımıza matematiksel destek olarak A/D dönüştürücünün denklemi olarak adlandırılan denklemi sunacağız. Bu denklemden dönüşüm birlerin sayısının A/D dönüştürücünün isabetliğine etkisi matematiksel açıdan daha iyi görülüyor.

n bitli çözünürlüğü olan A/D dönüştürücüyü gözetlediğimizi tahmin edelim.  $V_{REF}$  referans gerilimidir,  $V_{INA}$  ise dönüştürmemiz gereken analog giriş gerilimi olduğunu tahmin edelim. A/D dönüştürücünün çıkış gerilimi, n bitten oluşan ( $D_{n-1}D_{n-2}...D_1D_0$ ) ikili sözcük (vektör) şeklinde, onun karşılıklı onlu değeri d (onlu gösterimde) ile elde edilmelidir. Dijital şekline dönüştürülen giriş analog sinyaline, ya da ikili vektörüne karşılıklı d onlu değer, analog giriş gerilimin mevcut değeri ve karşılaştırıcı nicemin  $Q(\Delta)$  arasındaki bölümünün tam bölümü olarak, şu denkleme göre hesaplanıyor:

$$d = \left[ \frac{V_{in}}{Q} \right] \quad (8-18)$$

Yukarıdaki denklemde  $Q(\Delta)$  nicemi (çözünürlüğü), A/D dönüştürücü tarafından giriş gerilimin algılanabilen (tespit edilebilen) en küçük değişikliğini belirliyor. Böylece aslında dönüştürücünün isabetliği (hassasiyeti) de belirleniyor:

$$Q = \frac{V_{an \max} - V_{an \min}}{N_Q} = \frac{V_{REF} - 0}{N_Q} = \frac{V_{REF}}{2^n} \quad (8-19)$$

n çıkış dijital sinyalin bitlerinin sayısıdır.

Nicemleyicinin isabetliği için ölçü olarak nicemleme hatası tanımlanıyor (İng. Quantization Error). Nicemleme hatası, analog giriş sinyalin örnekleme seviyesi ve nicemlemenin çıkışındaki ayrık değeri arasındaki farktır, yani belirli  $V_i$  referans seviyesine göre sapma olarak tanımlanıyor. Bu değer analog girişi ve onun ayrılmış değeri arasındaki farkın orta değeridir. İletim özelliği Şek. 8-14'te verilmiş nicemleyiciyle ilgili ideal A/D dönüşüm durumunda, nicemleme hatası 0 ile Q arasındaki sınırlarda, ya da  $V_i$ 'den ( $V_i+Q$ )'ya kadar değişiyor.

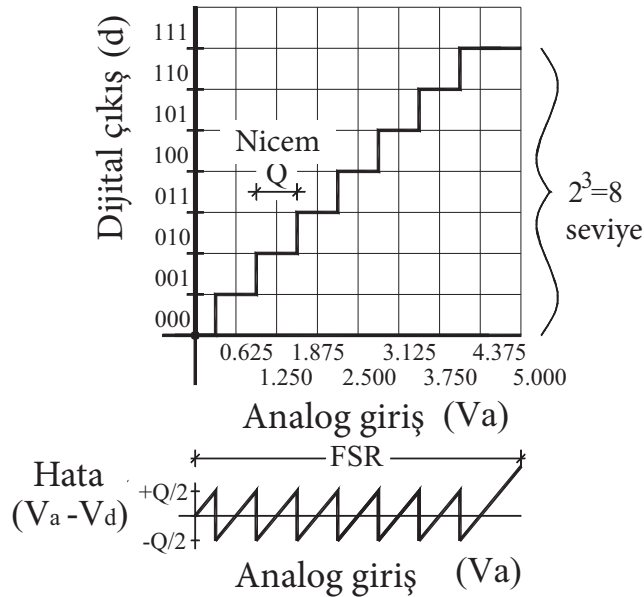
Pratikte genelde A/D dönüştürücünün Şek. 8-14 veya 8-16'daki iletim özelliğinin Q/2 için kayması yapılıyor, öyle ki A/D dönüştürücünün iletim özelliğinin genelde Şek. 8-18'de gösterilen şekli vardır. Bu iletim özelliğine, üç bitli A/D dönüştürücüyle ilgili gerilim seviyeli tab. 8-3 uygundur. Bu tablo ile, Şek. 8-14'ten önceki örneğe kıyasen farklar daha kolay görülebilir.



Analog gerilim aralığı	Gerilim farkı	Referans gerilim seviyesi [V]	Onlu değer	İkili karşılık	
$V_0 - V_1$	0.0000–0.3125	$0,5 \cdot Q$	$V_0=0V_R/8$ 0.000	0	000
$V_1 - V_2$	0.3125–0.9375	Q	$V_1=1V_R/8$ 0.625	1	001
$V_2 - V_3$	0.9375–1,5625	Q	$V_2=2V_R/8$ 1.250	2	010
$V_3 - V_4$	1,5625–2,1875	Q	$V_3=3V_R/8$ 1.875	3	011
$V_4 - V_5$	2,1875–2,8125	Q	$V_4=4V_R/8$ 2.500	4	100
$V_5 - V_6$	2,8125–3,4375	Q	$V_5=5V_R/8$ 3.125	5	101
$V_6 - V_7$	3,4375–4,0625	Q	$V_6=6V_R/8$ 3.750	6	110
$V_7 - V_R$	4,0625–5.000	$1,5 \cdot Q$	$V_7=7V_R/8$ 4.375	7	111

Tab. 8-3. Şek. 8-18'deki iletim özelliğine göre A/D dönüşümünde gerilim seviyeleri

Böyle değişim verilen referans gerilim seviyesi etrafında değerlerin yuvarlanması sağlanır ve bu işlemin matematiksel özürüllüğü vardır. Şek.8-18'de verilen durumda da, en büyük nicemleme hatası, mutlak değer olarak, Şek.8-14'te olduğu gibi bir nicem (kuantum) Q değerine eşit olduğu görülüyor, ancak şimdi bu hata referent seviyesi artı/eksi yarım nicem etrafındadır:  $(V_i - Q/2)$ 'den  $(V_i + Q/2)$ 'ye kadar.



Şek. 8-18. Üç bitli A/D dönüştürücüde idealleşmiş iletim özelliği ve nicemleme hatası

## 8.8. A/D DÖNÜŞTÜRÜCÜLERİN AYRIMI VE ÇEŞİTLERİ

A/D dönüştürücülerin yapılmasının dayandığı birçok temel süreci vardır. Onların en büyük kısmı şu büyük gruplara ayrılabilir:

1. Paralel (flaş) A/D dönüştürücüler (İng. Flash ya da Parallel ADC),
2. D/A dönüşümüne dayanan A/D dönüştürücüler (İng. DAC based ADC),
3. Tümlüşmeli A/D dönüştürücüler (İng. Integrating ADC) ve
4. Delta-sigma (sigma-delta) A/D dönüştürücüler (İng. delta-sigma ya da sigma-delta ADC).

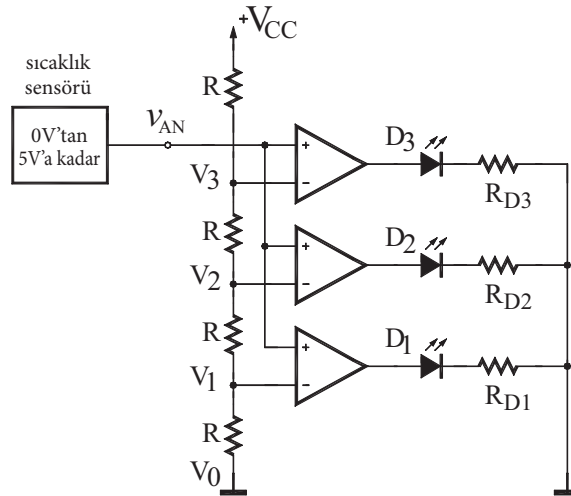
Farklı gruplardan A/D dönüştürücüler aralarında, çalışma hızı, isabetlik ve maliyeti gibi performansları açısından farklıdır ve onların pratik uygulaması bu performanslara bağlıdır. Devamda her gruptan tipik temsilcilerin yapılandırılmasında ve çalışma prensibinin incelenmesine değineceğiz. Her A/D dönüştürücü grubunun farklı pratik uygulaması olduğundan, devamda her birini ayrıdan inceleyeceğiz.

### 8.8.1. PARALEL ADC

Paralel A/D dönüştürücü ya da doğrudan dönüşümlü A/D dönüştürücüsü, nerdeyse yıldırım çalışma hızında dolayı, popüler olarak flaş (İng.flash) A/D dönüştürücü olarak da adlandırılıyor. Bu A/D dönüştürücü oldukça basit şekilde çalışıyor, öyle ki çalışmasını anlamak için kolaydır. Şöyle ki dönüştürücü karşılaştırıcı aracılığıyla, giriş analog gerilimi, verilmiş  $V_{REF}$  referans gerilimin ayırmasıyla elde edilen önceden ayarlanmış ve belirlenmiş sabit gerilim seviyesiyle karşılaştırılıyor. Bu seviyenin değeri, giriş analog gerilimin ulaşabildiği en yüksek gerilim seviyesine eşittir. Böylece, örneğin referans gerilimi 5V ise, o zaman giriş analog gerilimin en yüksek seviyesi de 5V olmalıdır. Referans gerilimi, eşit dirençlerden oluşmuş olan gerilim, ayırıcı rolü olan uygun direnç ağı yardımıyla ayrılıyor. Bu arada her direncin gerilimi, bağlı olduğu karşılaştırıcı için ayrı referans gerilimi olarak alınıyor, öyle ki giriş referans gerilimin verilen andaki değeri sonlu sayıda ayrı referans gerilimlerle karşılaştırılıyor.

Aşağıdaki Şek. 8-19'da  $V_{REF} = +5V$  referans gerilimli ve dört ayrı karşılaştırıcı seviyeden oluşan, sıcaklık sensörünün ürettiği gerilimden uyarılan çok basit flaş A/D dönüştürücü örneği gösterilmiştir. Tüm rezistörlerin eşit dirençleri ( $R$ ) olduğundan dolayı, ayrı gerilim seviyelerin değerleri şu denklemlerle elde edilecektir:

$$V_3 = \frac{V_{CC}}{4R} \cdot R = 3.75V, V_2 = \frac{V_{CC}}{4R} \cdot 2R = 2.5V, V_1 = \frac{V_{CC}}{4R} \cdot 3R = 1.25V, V_0 = 0V \quad (8-20)$$



Şek. 8-19. Dört seviyeli flaş A/D dönüştürücü

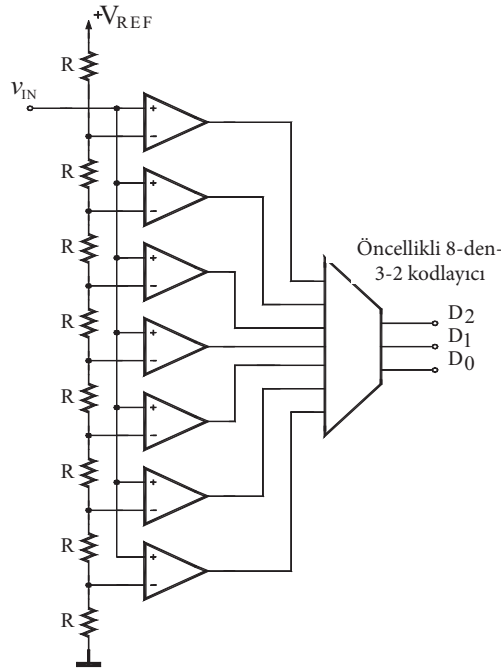
Her karşılaştırıcının çıkışında ışıklanan LED diyotları bulunuyor. Bu arada belirli diyot, sıcaklık sensörünün çıkış gerilim seviyesi, karşılaştırıcı çıkışına bağlı olduğu uygun karşılaştırma seviyesine ulaştığı anda yanıyor. Diyot ışığının yanması, o çıkışta yüksek seviyenin bulunduğunu gösterecek.



Böylece, örneğin giriş analog gerilimi  $V_i = 1.25V$  altında ise hiçbir karşılaştırıcı karşılık vermiyor ve tabii ki LED diyotlardan hiçbiri yanmayacak. Giriş gerilimi  $V_i = 1.25V$ 'luk birinci referans sınırını aşarsa, birinci LED diyodu  $D_1$  yanacak,  $V_2 = 2.5V$ 'luk ikinci seviyeyi geçerse birinci LED diyodu dışında, ikinci  $D_2$  LED diyodu da yanacak,  $V_3 = 3.75V$ 'luk üçüncü seviyeyi aşarsa ilk iki diyot dışında üçüncü LED diyodu  $D_3$  da yanacak. Sinyalizasyon oldukça yararlı ve uyarıcıdır, çünkü sıcaklığın artmasıyla gerilim de artıyor, dolayısıyla yanan diyotların sayısı da artıyor. Ancak, böyle bir çalışma prensibi, bilgisayar ya da başka bir dijital sistem tarafından işletilmesi için en uygun değildir çünkü çıkışta uygun ikili şekilde kodlanmamış sinyal elde ediliyor.

Yukarıdaki örnekte, ikili kodlanmış çıkış sinyalin elde edilmesi için, karşılaştırıcıların çıkışı 4-ten-2 kodlayıcıya bağlanmaları gerekecek. Kodlayıcının girişlerini işlem kuvvetlendiricinin çıkışına bağlayarak, üç referans seviyeden birine ulaşıncaya kadar, kodlayıcının çıkışında iki bitten benzersiz ikili kombinasyon üretilen. Sıcaklığın sürekli arttığından ya da azaldığından dolayı, böyle davranma (çalışma) şekli sadece öncelikli kodlayıcının uygulanmasıyla sağlanabilir.

Bu tür A/D dönüştürücünün çalışma prensibini daha kolay anlamak için, Şek. 8-20'de verilmiş olan ilkel elektrik diyagramını inceleyeceğiz. Şek. 8-20'de oldukça basit, 8-den-3'e öncelikli kodlayıcı içeren üç bitli flaş A/D dönüştürücü gösterilmiştir. Bu arada karşılaştırıcılar olarak ideal işlem kuvvetlendiricileri kullanılıyor. Çıkışta üç bitli çıkış vektörün üretilmesi gerektiğini tahmin ettiğimizden, çıkışın en küçük değeri  $0_{(10)}$  ya da  $000_{(2)}$  olacak, en yüksek değer olarak ise  $111_{(2)}$  ya da  $7_{(10)}$  meydana gelecek.



Şek. 8-20. 3-bitli flaş ADC'nin ilkel elektrik diyagramı

Üç bitli A/D dönüştürücü söz konusu olduğundan dolayı, direnç ayırıcısı eşit değerleri olan  $2^3 = 8$  dirençten (rezistörden) ve yedi karşılaştırıcıdan oluşuyor. Karşılaştırıcının çıkışları, kodlayıcının girişleri olarak getiriliyor. Kodlayıcı giriş gerilimin seviyesine öncelikli karşılık verdiği için, çıkışında sıfırcıdan farklı yedi referans seviyesinden en büyüğüne en yakın giriş analog seviyesine uyan kod sözcüğü meydana geliyor. 8-den-3'e öncelikli kodlayıcının en basit uygulaması 74x148 işaretli ik'nin uygulanmasıdır.

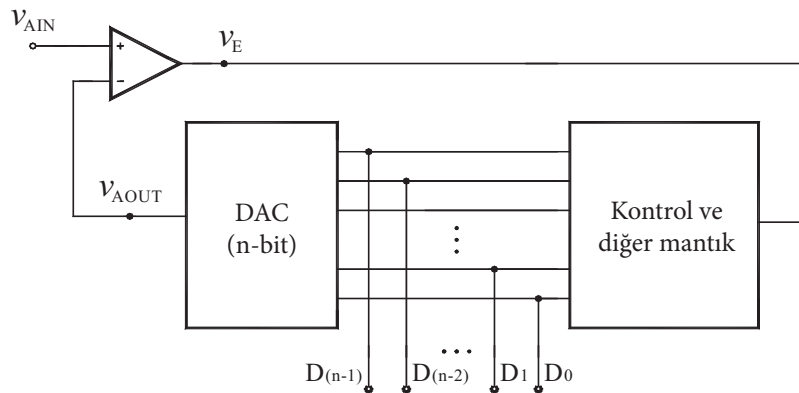
İncelenen A/D dönüştürücünün üç bitli olduğunu göz önüne alarak, çıkış gerilim deyişleri arasındaki en küçük fark, ya da nicem aralığı  $Q$ ,  $625\text{mV}$  için artma ya da azalma olacak. Şöyle ki, dirençlerin sıfır referans seviyesi (topraklama, “tablo”) ve  $5\text{V}$ ’luk besleme ( $V_{\text{REF}} = +5\text{V}$ ) arasında bağlı olduğundan dolayı, bu seviye aslında aynı zamanda kodlayıcının girişinde meydana gelebilen en yüksek seviyedir. Böylece, FSR tam kapsam,  $5\text{V}$ ’luk dirençlerin gerilimine eşit olacak. Üç bitle  $N=2^3=8$  farklı seviyenin kodlanabileceğine göre, (8-17) denkleminin uygulanmasıyla  $625\text{mV}$ ’luk nicem farkı  $Q$  elde ediliyor ( $Q=5\text{V}/8 = 0.625\text{V}$ ). Bu şekilde, birinci referans seviyesi  $V_1=625\text{mV}$  olacak, ikinci seviye  $V_2 = 1.25\text{V}$  olacak vs. Sıradaki her referans seviyesi daha büyük olacak ve gerilimin önceki referans seviyesinden  $625\text{mV}$  için artarak, son sekizinci seviye  $V_7 = 4.375\text{V}$  olacak. Tabii ki en alçak referans seviyesi sıfıncı seviye, tablonun potansiyeli olacak, yani  $V_0 = 0\text{V}$ .

Şek.8-20’den flaş A/D dönüştürücünün elektrik diyagramı oldukça basit olduğunu görebiliriz. Fakat, daha ciddi kullanım için, sekiz veya fazla bitten daha büyük isabetlik arandığı zaman, bu A/D dönüştürücü pahalı olacak çünkü onun içeriliğinde büyük sayıda karşılaştırıcı olacak, daha doğrusu  $n$  çıkış kod sözcüğünde bitlerin sayısı ise o zaman  $(2^n-1)$  karşılaştırıcı gerekecek. Böylece, örneğin 8-bitli flaş A/D dönüştürücü için  $2^8-1=255$  karşılaştırıcı gerekecek, profesyonel 16-bitli A/D dönüştürücü için ise  $2^{16}-1=65.535$  karşılaştırıcı gerekecek! Ayrıca, çözünürlüğün ve dolayısıyla isabetliğin artması gerektiği zamanı, flaş A/D dönüştürücü güç tüketimi her yeni eklenen bitle büyük ölçüde artıyor, çünkü her yeni eklenen bitle karşılaştırıcıların sayısı iki kat artıyor.

Diğer taraftan, bu A/D dönüştürücünün çok büyük çalışma hızı vardır, çünkü A/D dönüştürücü çıkışındaki girişte getirilen analog gerilime uygun ikili karşılık çıkışta doğrudan ve nerdeyse aynı anda elde ediliyor. Şöyle ki, gecikme sadece sinyalin kodlayıcının işlem kuvvetlendiricileri ve mantıksal devrelerden geçmesi için gereken zamana kalıyor ve bu süre gerçekten çok kısadır. Bu A/D dönüştürücünün büyük çalışma hızından dolayı flaş ya da yıldırım A/D dönüştürücü olarak da adlandırılıyor.

### 8.8.2. D/A DÖNÜŞÜMÜNE DAYANAN A/D DÖNÜŞTÜRÜCÜLER

Bu A/D dönüştürücülerin en önemli özelliği, giriş analog sinyalinin ona en uygun olan dijital bitle karşılaştırılması ve bu şekilde belirli bit çıkış dijital sinyalinde yerleşiyor ya da yerleşmiyor (siliniyor). Böyle A/D dönüştürücülerin çalışma prensibi, Şek.8-21’de verilen blok diyagrama göre,  $n$  bit D/A dönüştürücü aracılığıyla geri bağlantının kurulmasına dayanıyor. Temel fikir, kontrol mantığının  $n$  uzunluğunda ikili sözcük ( $D_{n-1}, D_{n-2}, \dots, D_1$ ) üreterek ve bu sözcüğü D/A dönüştürücünün aracılığıyla  $n$  onlu değeri olan  $V_{\text{ADAC}}$  iç analog gerilime dönüştürmektir. Ardından bu iç analog gerilimi, giriş analog gerilimle karşılaştırılıyor.



Şek. 8-21.D/A dönüştürücüye dayanan A/D dönüştürücünün blok-diyagramı

Karşılaştırıcının bir girişinde, bu tür A/D dönüştürücünün parçası olan D/A dönüştürücüden çıkış  $V_{ADAC}$  getiriliyor, diğer girişinde ise giriş analog gerilimi  $V_{AIN}$  getiriliyor. Sinyal hatası olarak meydana gelen farkı, iki sinyal arasındaki farkı (hatayı) azaltma amacıyla dijital sinyali düzelteren kontrol mantığının çalışmasını belirtiyor. D/A dönüştürücünün çıkış gerilimi, giriş analog geriliminin nicemin (çözünürlüğün) değerinden daha küçük olan hata değerini seviyesini aştığı anda, karşılaştırıcı çıkış değerini değiştiriyor ve kontrol devrenin baştan başlaması için onu sıfırlıyor. Aynı anda dijital değer, D/A dönüştürücüden çıkış geriliminde n uzunluğunda ikili sözcük şeklinde bellekleniyor. Bununla örneklemenin dönüşüm süreci tamamlanıyor ve giriş analog sinyalinden sıradaki örneklemenin dönüşümü başlıyor.

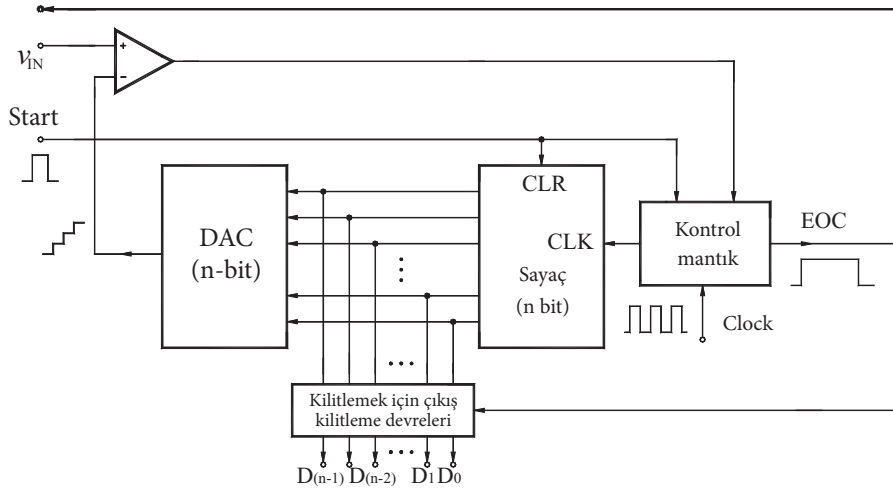
Karşılaştırma bloğunda temel parçası olarak D/A dönüştürücü uygulayan A/D dönüştürücülerin yapılması için birkaç farklı yöntem vardır. Bunlardan devamda bu gruba ait iki dönüştürücü incelenecektir:

- sayaç (sayaç dönüşümlü) ya da dijital rampalı A/D dönüştürücü (İng. ADC with ramp counter ya da digital ramp ADC) ve
- Ardaşıl yaklaşılmalı A/D dönüştürücü (İng. successive-approximation ADC).

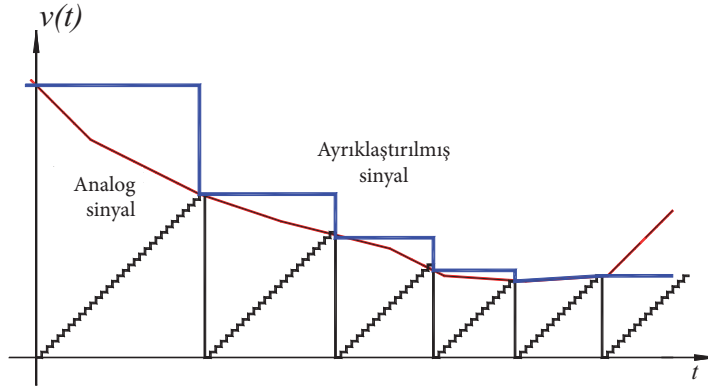
### 8.5.2.1. SAYAÇ RAMPALI ADC

Bu A/D dönüştürücüde, dönüştürmeyi yapan temel eleman iç ikili sayaçtır. Bu sayaç sıfırdan sayma başlayıp en yüksek değere doğru sayıyor. Sayma satacın durumu iç A/D dönüştürücü aracılığıyla  $V_{in}$  giriş analog geriliminin örneklemesine uyan değere ulaşılan kadar ya da o seviyeden büyük olan en yakın dijital değere kadar devam ediyor. Sayaçın en yüksek değeri  $(2^n - 1)$ 'dir ve buradan dönüştürücünün çözünürlüğüne, ya da bitler sayısı  $n$ 'e bağlı olduğu görülüyor.

Böyle çalışma prensibinden dolayı bu A/D dönüştürücüye sayaçlı, basamaklı veya dijital rampa denir (İng. Ramp counter ADC ya da digital ramp ADC). Bu dönüştürücünün ilkeli blok-diyagramı Şek. 8-22 olarak belirtilmiş aşağıdaki şekilde gösterilmiştir, Şek. 8-23'te ise verilen devrenin karakteristik noktalarında gerilimlerin zamansal diyagramları bu dönüştürücünün çalışmasını daha iyi açıklıyor.



Şek. 8-22. Sayaç rampalı A/D dönüştürücünün blok-diyagramı



Şek. 8-23. Giriş analog gerilimin ve sayac rampalı A/D dönüştürücüde iç D/A dönüştürücüden üretilen karşılaştırma çıkış gerilimi

Şekilde  $V_{in}$  ile analog giriş gerilimi gösterilmiştir,  $n$  dijital çıkış (bit) ise  $D_{n-1}$ 'den  $D_0$ 'a kadar işaretlenmiştir. Sayacın çalışması START giriş hattıyla kontrol ediliyor. Bu giriş hattına yüksek seviyenin getirildiği anda, sayacı sıfırlandırıyor ve kontrol mantığı aracılığıyla pals dürtülerin (Clock) sayaç girişine gelmesi sağlanıyor ve sayaç saymaya başlıyor. Pals sinyalin her dürtüsüyle sayacın değeri 1 için artıyor. Sayaç çıkışları D/A dönüştürücüyü uyarıyor ve bu giriş kombinasyonları 1 için arttığandan dolayı D/A dönüştürücünün çıkışında sürekli büyüyen basamak şeklinde gerilim sinyali, yani dijital rampa elde ediliyor.

Bu A/D dönüştürücünün temel fikri, sayacın değerini, sayacın D/A dönüştürücünü aracılığıyla verdiği değer giriş analog sinyalinden örneklemenin değerine ulaşana kadar artmaktadır. Her dönüşüm döngünün başlangıcında, giriş analog sinyalini ve amplitüdü sürekli artan içerde üretilen analog sinyalini karşılaştıran karşılaştırıcının çıkışı alçak seviyede bulunuyor. D/A dönüştürücünün çıkışı giriş gerilimin seviyesine ulaştığı anda, girişinde bulunan ve sayaç çıkışının değeriyle eşit olan ikili kombinasyon, analog sinyalin dijital karşılığı olarak alınıyor. Bu anda karşılaştırıcının çıkışı alçak seviyeden yüksek seviyeye değişiyor. Bu, tamamlama sinyali gönderen (İng.End), yani örneklemenin dönüşüm sürecinin tamamlandığını belirleyen EOC (İng. End of Conversion) ve giriş analog gerilimden yani örneklemenin dönüşümü başlayabileceğini belirleyen sinyal kontrol mantığının tepki vermesine yol açıyor. EOC sinyali aynı zamanda sayacın değerinin okunabilmesi ve işletilmesi için mevcut değeri kilitleyen (tutan) çıkış mandal devreleri etkinleştiriyor.

Mandal devrelerin, içeriğinde bulunduğu çıkış arabelleği dönüştürücünün parçası olması şart değildir, ancak, arabelleğin olması büyük avantaj sağlıyor çünkü örneklemenin dijital değerinin tutulması, A/D dönüştürücünün dönüşmüş değerleri alan devrede, A/D dönüştürücü giriş analog gerilimin sıradaki örnekleme üzerine çalışırken, kilitlenen değerlerin okunması ve işletilmesi sağlanıyor.

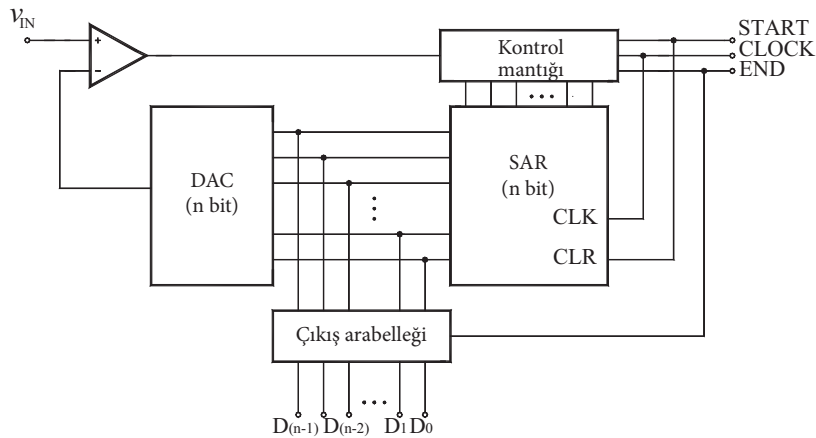
EOC sinyalinin elde edilmesinden sonra, karşılaştırıcı girişindeki D/A dönüştürücünün giriş devreleri, analog sinyalden sıradaki örneklemeyle ilgili ve aynı anda onun dijital değere dönüştürülmesinin başlatılması için START sinyali gönderiyor. Buna göre, giriş analog sinyalinden, dönüştürülmesi gereken her örnekleme için,  $f_s$  seçme (örnekleme) frekansına eşit frekanslı birer START dürtüsü gönderiliyor.

Kısaca, sayaç 0'dan başlayarak en yüksek değerine kadar  $(2n-1)$ , A/D dönüştürücünün girişinde bulunan analog gerilimden örnekleme seviyesinin tam ya da en uygun dijital seviyesine ulaşana kadar sayıyor. Bu değer ulaşıncaya kadar, EOC sinyali üretiliyor ve analog gerilimin (örneklemenin)  $V_{in}$  giriş seviyesinin dijital şekli, sayacın  $D_{n-1}$ 'den  $D_0$ 'a kadar çıkışlarında bulunan ikili vektör yardımıyla elde ediliyor. Bununla ilgili olarak, bu ADC türünün en büyük sorunu çalışma hızıdır. Şöyle ki, gerilim seviyesi için küçük değeri olan örneklemeler için, dönüşüm süresi kısadır, ancak daha büyük giriş seviyeleri için dönüşüm süresi büyük ölçüde artıyor. Bu dönüştürücü oldukça yavaştır, çünkü dönüştürülmesi gereken ve en yüksek değere yakın değeri olan her örnekleme için,  $(2^n-1)$  pils aralığı (döngü) geçebilir. Böylece, örneğin 8 bitli ADC'de, en yüksek seviyesi olan örnekleme için  $2^8-1=256-1=255$  pils gerekecek, ancak 12 bitli ADC ile çalışılıyorsa, o zaman bu örneklemenin dönüştürülmesi için  $2^{12}-1=(4x1024)-1=4096-1=4.095$  pils aralığı gerekecek. Ayrıca, böyle çalışma şekli,  $f_s$  seçme frekansından  $2n$  kat daha yüksek frekanlı pilsinyalin ( $f_{CLK}$ ) uygulamasını gerektiriyor.

### 8.8.2.2. ARDAŞIK YAKLAŞIMLA ADC

Ardışık yaklaşımla A/D dönüşüm (İng. successive approximation ADC), analog sinyallerin dijital sinyallere dönüştürmek için en çok kullanılan prensiplerden biridir. Aşağıdaki Şek.8-24'te bir ilkeli blok-diyagram gösterilmiştir. Buradan, bu tür A/D dönüştürücünün yapısını, saymalı A/D dönüştürücüye benzer olarak, analog giriş sinyalinden örnekleme seviyesiyle kıyaslamak için sinyal üreten iç D/A dönüştürücüye dayandığını görebiliriz.

Analog giriş gerilimi  $V_{in}$  ile işaretlenmiştir, örnekleme mevcut giriş gerilim seviyesine dijital karşılığının elde edildiği dijital çıkışlar ise  $D_{n-1}, D_{n-2}, \dots, D_1, D_0$  olarak belirtilmiş  $n$  bittir. SAR ile işaretlenmiş olan blok, ardışık yaklaşım yazmacıyla ilgilidir (İng. Successive Approximation Register). Kontrol mantığı, sayaç rampalı A/D dönüştürücüde olduğu gibi tüm kontrol sinyalleri gönderiyor: yazmaçın çalışmasını başlatan ve herhangi bir örnekleme dönüştürülmesini başlatan START, verilen örnekleme dönüştürülmesi tamamlandığını gösteren EOC (END) sinyali ve CLOK pils sinyali. Şekilden, bu durumda da çıkış arabelleğinin kullanıldığı görülüyor, öyle ki dönüştürücü giriş analog gerilimden sıradaki örneklemeyi işletirken, dijital veri (ikili sözcük) işletilmesi gereken mantıksal ağda duruyor. Yine de, basamaklı rampalı A/D dönüştürüye kıyasen, çalışma prensibinde büyük fark vardır.

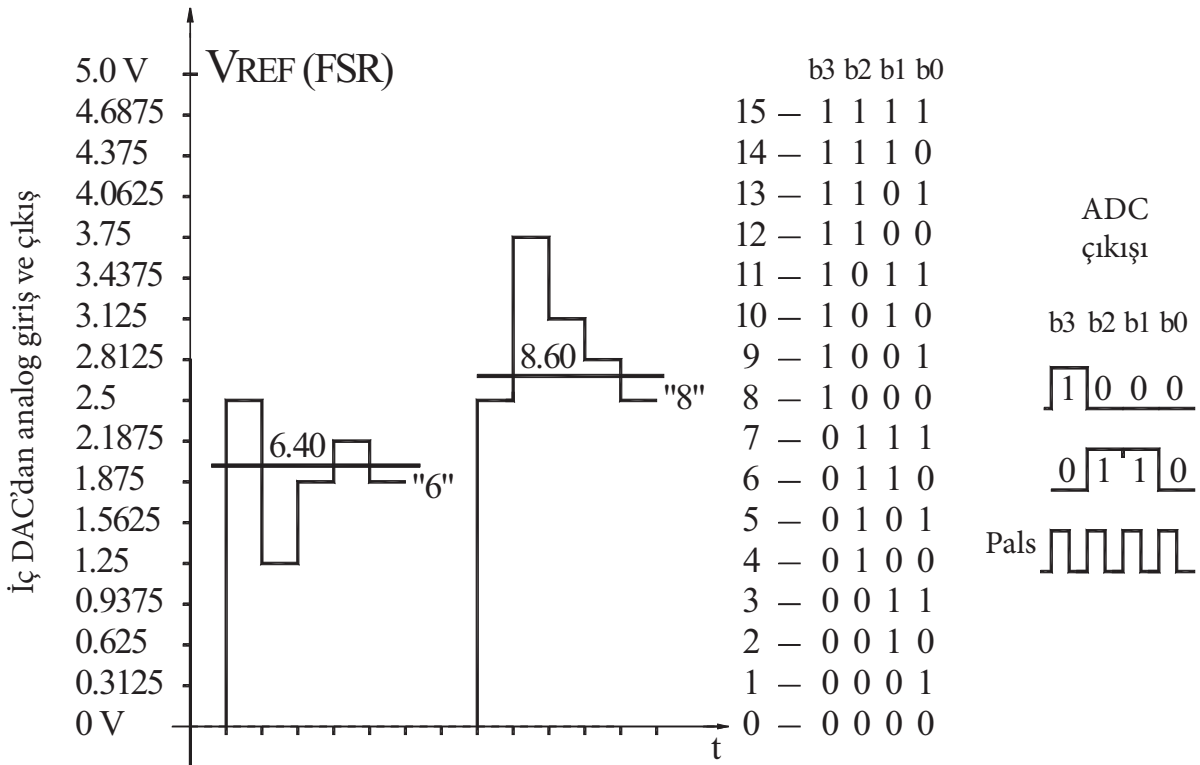


Şek. 8-24. Ardışık yaklaşımla A/D dönüştürücünün blok-diyagramı

Daha kesin olarak, bu dönüştürücü, en büyük ağırlığı ( $2^{n-1}$ ) olan MSB  $D_{n-1}$  bitini mantıksal 1 seviyesine getirmesiyle başlayarak analog giriş örnekleme verileri verilen andaki değerine ardışık olarak yaklaşıyor. Bununla iç D/A dönüştürücünün çıkışında aslında,  $+V_{REF}$  referans gerilime eşit olan en büyük kapsamın yarısına eşit gerilim seviyesi olan sinyal meydana geliyor. Karşılaştırıcı bu gerilimi giriş analog gerilimiyle karşılaştırıyor. Karşılaştırma sonucu olarak karşılaştırıcı kontrol blokuna kadar uygun sinyal üretiyor ve bu sinyal ile neyin yapılması gerektiği söyleniyor:

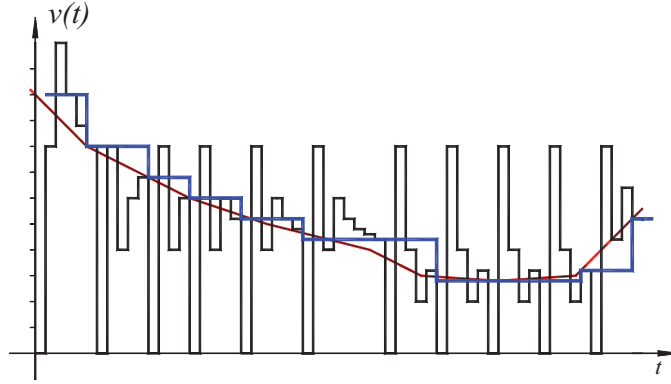
- bitin 1'de ayarlanmış kalması gerekiyor (bu durum iç D/A dönüştürücüden çıkış gerilimin örnekleme seviyesinden daha küçük olduğu sırada meydana geliyor, yani giriş örnekleme seviyesi, iç D/A dönüştürücü tarafından üretilen analog gerilimden daha büyük olduğu sırada meydana geliyor) ya da
- 1'de ayarlanmış bit silinmelidir, ya da 0'a ayarlanmalıdır (bu durum iç D/A dönüştürücünün çıkış gerilimi giriş gerilimden, ya da örneklemeden daha büyük seviyede olduğu zaman meydana geliyor).

Bu çalılma prensibi, önceki bittten iki kat daha düşük ağırlığı ( $2^{n-2}$ ) olan sıradaki  $D_{n-2}$  biti için de tekrarlanıyor. D/A dönüştürücü çıkışının analog değeri şimdi A/D dönüştürücünün en yüksek seviyesinin bir çeyreğine eşit değeri olacak. Karşılaştırıcıda tekrar yeni üretilen seviye ve örnekleme seviyesi arasında kıyaslama yapılıyor. Yapılan kıyaslama ardından karşılaştırıcının çıkış seviyesi kontrol mantığı aracılığıyla tekrar bu bitin 1 kalması ya da sıfırlanması gerekip gerekmediği belirleniyor. Bu işlem en düşük ağırlığı ( $2^0$ ) olan son LSB  $D_0$  bitine kadar devam ediyor. Tüm bu işlem, Şek 8-25'te ve 8-26'da grafiksel olarak gösterilmiştir.



Şek. 8-25. İki örnekleme için ardışık yaklaşma sürecinin zamansal diyagramı





Şek. 8-26. Ardışık yaklaşımlı A/D dönüştürücüsünde giriş analog gerilimin ve iç D/A dönüştürücüsünde üretilen kıyaslama çıkış geriliminin zamansal diyagramları

Bu A/D dönüştürücünün en önemli avantajı çalışma hızıdır. Şöyle ki analog örnekleme seviyesi için uygun dijital değerin belirlenmesi, en kötü ihtimalle, palsın  $n$  döngüsü gerekecek.  $n$  analog girişim dijital sunumu için kullanılan bitlerin sayısıdır. Buna göre sayaç rampalı A/D dönüştürücüsünde olduğu gibi  $2^n$  pals döngüsü gerekmiyor. Böylece, örneğin 8 bitli ardışık yaklaşımlı A/D dönüştürücü kullanılırsa, bir örneklemin dijital değeri en çok 8 pals için belirlenecek. Sayaç rampalı A/D dönüştürücü durumunda harcanabilen  $2^8 - 1 = 255$  palsa kıyaslan, gerçekten çok büyük ilerlemedir. 12 bitli A/D dönüşüm durumunda, örneklemin dijital değeri 12 pals döngü içinde elde edilebilir, önceki basamaklı rampalı dönüştürücüsünde aynı işlem için  $2^{12} - 1 = 4096$  zaman aralıkları harcanabilir. Buna göre, dijital rampalı dönüştürücüsünde olduğu gibi  $f_s$  seçme frekansından  $2^n$  kat daha yüksek,  $f_{CLK}$  frekanslı pals-sinyali yerine, bu durumda  $f_s$  seçme frekansından sadece  $n$  kat daha yüksek  $f_{CLK}$  frekanslı pals sinyali uygulanabilir.

### 8.8.3. TÜMLEŞİK DEVREYE DAYANAN ADC

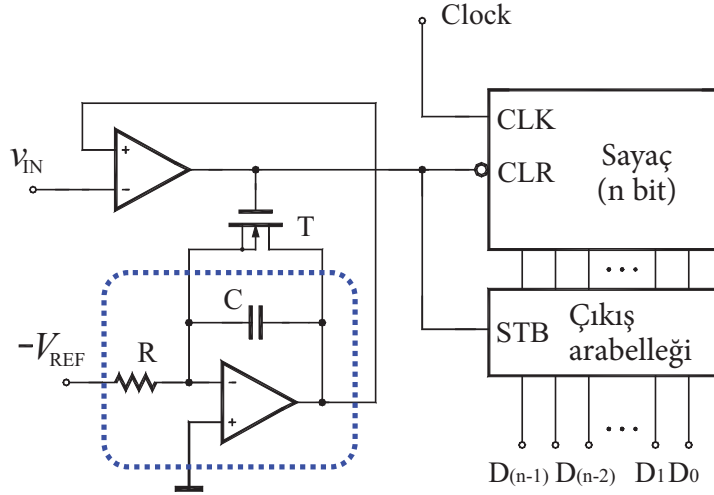
Tümleşme yardımıyla ADC (İng. Integrating ADC) tasarlamak için farklı yöntemler vardır. Devamda tek eğimli A/D dönüştürücüyü (İng. Single-slope ADC) ve çift eğimli A/D dönüştürücüyü (İng. Dual-slope ADC) inceleyeceğiz.

İlerleyen bölümde göreceğimiz gibi, bu iki A/D dönüştürücünün ortak özelliği, sayaç rampalı A/D dönüştürücünün çalışma prensibiyle benzerliktir. Bu dönüşüm şeklinde de sayaç kullanılıyor, sadece tek fark, analog giriş sinyalinin örnekleme değeriyle kıyaslanan iç analog gerilimin üretilmesi için D/A dönüştürücünün değil, tümleşik devrenin kullanılmasıdır. Bu durumda tümleştirici çıkışında basamak şeklinde kıyaslama gerilimi yerine, yükselen doğrusal şekilde kıyaslama gerilimi elde ediliyor.

#### 8.8.3.1. TEK EĞİMLİ A/D DÖNÜŞTÜRÜCÜ

Tümleşme ile A/D dönüştürücünün ilkeli blok-diyagramı Şek.8-27'de sunulmuştur. Devamda daha detaylı açıklayacağımız çalışma prensibinden dolayı, bu A/D dönüştürücüyü tek eğimli dönüştürücüsünde denir (İng. single-slope ADC). Verilen şekile daha dikkatli bakarsak, bu diyagramın sayaç rampalı A/D dönüştürücünün blok-diyagramına çok benzer olduğunu göreceğiz.

Sadece burada analog girişle karşılaştırılan iç analog gerilimin üretilmesi için, çıkışta doğrusal şekilde değil, basamak şeklinde yükselen sinyal üreten D/A dönüştürücü yerine, tümleşik devre, daha doğrusu testere gerilim üretici (İng.integrator, saw-tooth generator) kullanılıyor. Tümleştirici devre genelde kapasitif negtaif geri bağlantılı işlem kuvvetlendiriciden oluşuyor. Bu arada uyarma gerilimi evirici girişe rezistör aracılığıyla götürülüyor. Tümleştiricinin girişinde sabit tek yönlü negatif gerilim  $-V_{REF}$  getirilince, tümleştiricinin çıkışında doğrusal yükselen sinyal (gerilim şekli) elde edilecek, çünkü devre işaretin değişmesini gerçekleştiriyor. MOSFET'in, her dönüşüm döngüsünün başlangıcını kontrol eden anahtar eleman rolü vardır.



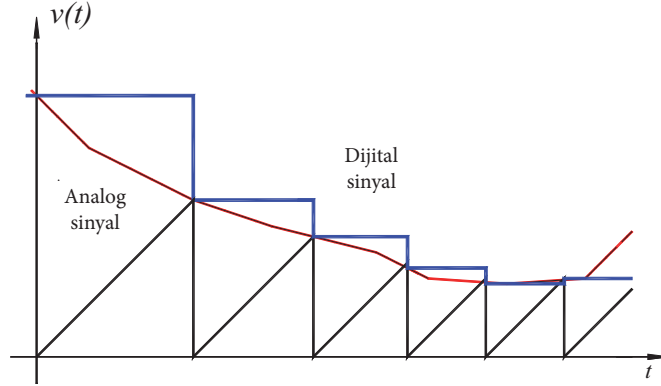
Şek. 8-27. Tümleştirici devreli ve tek eğimli A/D dönüştürücünün blok-diyagramı

Tümleştiricinin çıkışında elde edilen testere gerilim, en alçak sıfırıncı seviyeden çekerek, mutlak değeri giriş sinyalinin olabildiği en yüksek seviyesine eşit olan  $|-V_{REF}| = +V_{REF}$  en yüksek değere doğru doğrusal olarak yükseliyor. Bu zaman aralığında giriş analog gerilimi testere gerilimden daha büyüktür ve karşılaştırıcı çıkışında alçak seviye elde ediliyor. Karşılaştırıcının çıkışı MOSFET geçidine getiriliyor, öyle ki onun alçak değeri MOSFET'i durma alanında kutuplaştırıyor ve bu şekilde onun devre dışı bırakıyor, çünkü bu durumda MOSFET iletmiyor. Kondansatör drain-source (boşalma kaynağı) bağlantısına paralel olarak bağlı olduğundan ve ondan akım akmadığından ( $I_D=0$ ) dolayı, tüm akım C kondansatöründen geçip dolduruluyor. Bu doldurma akımının sabit kuvveti var ( $I_C=V_{REF}/R$ ) ve işlem kuvvetlendirici çıkışında yükselen testere gerilim meydana geliyor.

Testereri dürtünün üretilmesi bağladığı anda, sayaç 0-dan  $(2^n-1)$ 'e doğru saymaya başlıyor, n sayısı A/D dönüştürücünün kullandığı bit sayısıdır. Tümleştiricide üretilen testere gerilim, analog giriş sinyalinin Vin örnekleme seviyesine ulaşınca, karşılaştırıcı kendi çıkış seviyesini değiştirir sayacın o anda olduğu, yani son değeri tespit ediyor. O anda çıkış arabelleği kontrol girişi aktifleştiriyor ve onda giriş geriliminden dönüştürülen örneklemenin analog seviyesine uyan dijital değer yerleşiyor. Aynı anda karşılaştırıcının çıkış seviyesinin değişmesi sayacı sıfırlandırıyor ve MOSFET aracılığıyla tümleştiriciden kondansatörü boşaltıyor. Şöyle ki, yükselme eğilimi olan doğrusal gerilimi giriş analog gerilimiyle eşitlendiği zaman, karşılaştırıcı çıkış seviyesi alçak seviyeden yüksek seviyeye değiştiriyor. Yüksek seviye MOSFET geçidini uyarıyor ve MOSFET devreye giriyor. Drain-source gerilimi kısa bağlantı olarak davrandığından dolayı ( $U_{DS}=0V$ ) tümleştirici çıkışındaki potansiyel sıfıra düşüyor ve neredeyse aynı anda C kondansatörü boşalıyor ve bu şekilde yeni (sıradaki) dönüşüm seviyenin başlaması sağlanıyor. Şek. 8-28'de verilen tümleşik



devreli ve tek eğimli A/D dönüştürücünün karakteristik noktalarındaki gerilimlerin zamansal diyagramları çalışma prensibini görsel şekilde gösteriyor.



Şek. 8-28. Giriş analog gerilimin ve sayaç rampalı A/D dönüştürücüde tümleştirici tarafından üretilen kıyaslama çıkış geriliminin zamansal diyagramları

Sunulan analizi ve bu A/D dönüştürücünün blok-diyagramını göz önüne alarak, burada da çıkış arabellek devresinin kullanıldığını söyleyebiliriz. Buna göre son dönüş değeri okunurken, dönüştürücü artık giriş analog gerilim sinyalinden sıradaki örneklemeyi işletiyor.

Dönüşümün bu uygulaması, sayaç rampasına dayanan dönüşüme kıyasen daha basit olmasına rağmen, burada da 1 için artan ardışık sayma söz konusudur, öyle ki bu A/D dönüştürücüde de aynı sorun vardır, o da alçak çalışma hızıdır, çünkü yüksek gerilim seviyeli örneklemenin dönüştürülmesi için  $(2^n - 1)$  pals aralığı gerekebilir. Ayrıca, bu tek eğimli A/D dönüştürücü türünde bir sorun daha meydana gelebilir, o da kalibrasyon kaymasıdır. Şöyle ki, tümleştirici sayacın pals sinyaliyle bağlı olmadığına göre, zamanla belirli sapmalar meydana gelebilir.

### 8.8.3.2. ÇİFT EĞİMLİ A/D DÖNÜŞTÜRÜCÜ

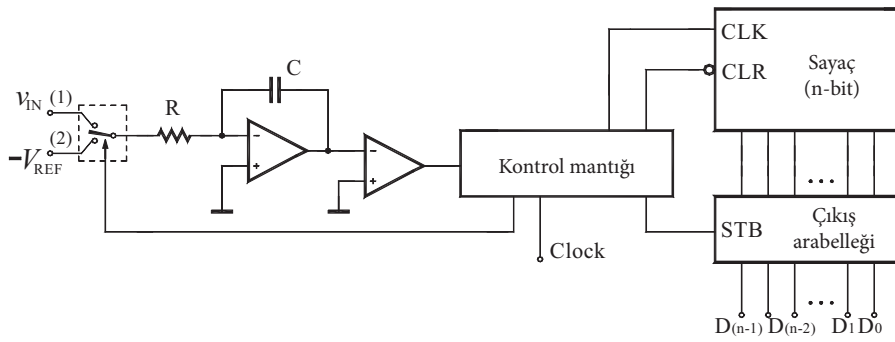
Tümleştirici devresine dayanan ikinci popüler tasarım çift eğimli A/D dönüştürücü olarak biliniliyor (İng. Dual-Slope ADC). Onun standart ancak basitleştirilmiş blok diyagramı Şek.8-29'da gösterilmiştir. Bu A/D dönüştürücü, tek eğimli A/D dönüştürücüde testereli sinyalin üretilmesi sayacın palsıyla senkronize olmadığından dolayı hataya yol açan kalibrasyon kayması sorununu çözüyor.

Çift eğimli A/D dönüştürücüde, dönüşüm döngüsü iki adımda gerçekleşiyor. Birinci aşamada kontrol mantığı S anahtarını 1 durumuna koyuyor ( $S=1$ ) ve bu şekilde  $V_{INA}$  analog giriş gerilimini bağlıyor, ikinci aşamada ise kontrol mantığı anahtarı  $-V_{REF}$  negatif değeri olan sabit referans gerilimi olan 2 durumuna bağlıyor ( $S=2$ ).

Dönüşüm sürecin başlangıcında, S anahtarı (1) durumunda bulunduğu zaman, analog sinyal örnekleme seviyesi tümleştiricinin (testereli gerilim üreticinin) girişine getiriliyor. Bu yüzden onun çıkışında ters, negatif eğimi olan testereli gerilim üretilmeye başlıyor. Bu doğrusal düşüş sinyali belli bir sabit zaman aralığı  $T_1$  için üretiliyor. Bu süre içinde karşılaştırıcı pasiftir, kontrol mantığı ise, tam olarak belirlenmiş pals-dürtü sayıda sıfırdan saymaya başlayan sayacın çalışmasını yönetiyor.

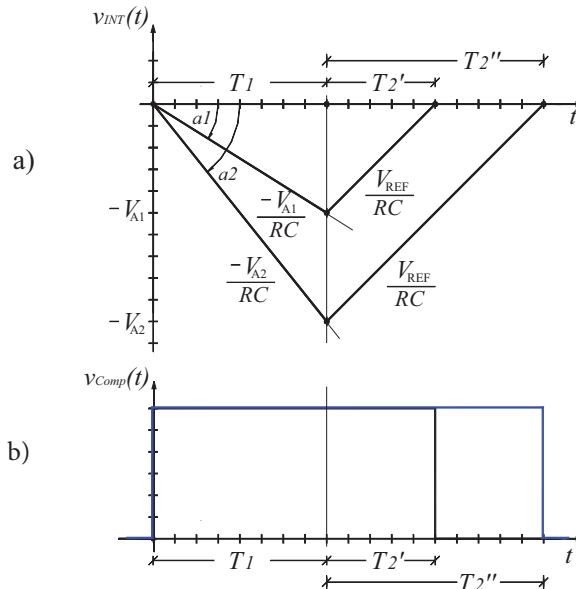
Böylece, aslında  $T_1$  zaman aralığının sabit uzunluğu belirleniyor. Bu sabit zaman aralığı geçtikçe, kontrol mantığı karşılık olarak anahtarın durumunu durum (1)'den durum (2)'ye değiştiriyor. Böylece, tümleştiricinin girişinde  $-V_{REF}$  referans gerilim bağlanıyor. Aynı zamanda kontrol mantığı sayaca iki kontrol dürtüsü gönderiyor. Biriyle sayacı sıfırlandırıyor (saymayı başlangıca geri döndürüyor), ikinci dürtüyle ise pal dürtülerini geçirip sayacın sıfırlanmasından sonra tekrar saymaya başlaması için uyarıyor. Bu arada karşılaştırıcı pasif durumda kalıyor.

Tümleştiricinin girişinde şimdi negatif referans gerilim  $-V_{REF}$  bağlı olduğu için, tümleştiricinin çıkışında, gerilimin önceki adımda girişinde analog sinyali bağlandığı zaman düştüğü negatif seviyeden başlayarak sıfırınca seviyeye doğru doğrusal olarak yükselmeye başlayan testereleli gerilim elde edilecek. Tümleştiricinin sıfırını seviye ulaştığı, belirli  $T_2$  zaman aralığı ardından, karşılaştırıcı tekrar karşılık vererek sayacın durumu, çıkış arabelleğinde hafız edilmiş analog giriş sinyalin dijital değerini tanımlayacak.



Şek. 8-29. Tümleşmeli ve çift eğimli A/D dönüştürücünün blok-diyagramı

Söylenenleri, yeniden dönüşüm döngüsüne bir kez daha dönerek daha iyi anlayabiliriz. Anahtarın birinci pozisyonunda ( $S=1$ ) sabit olan  $T_1$  zaman aralığında, eğimi  $V_{INA}$  analog giriş seviyesine bağlı olan testereleli gerilim üretiliyordu: seviye yüksek değerdeyse eğim de daha büyük olacak ve tersi. Bundan farklı olarak ikinci durumda ( $S=2$ ) tümleştiricinin uyarısı sabittir ( $-V_{REF}$ ) ve tümleştiricinin çıkış gerilimi sıfırınca seviyeye ulaştığı  $T_2$  zaman aralığı hangi gerilim seviyeden başladığına bağlıdır.



Şek. 8-30. a) Tümleştirici ve b) Karşılaştırıcı çıkışlarının dalgalı şekilleri

Bu süreç, tümleştiricinin ve karşılaştırıcının çıkışlarında diyagramdan karakteristik noktalarındaki sinyallerin zamansal diyagramları (dalgalı şekilleri) verilmiş olduğu Şek.8-30'a bakarak daha iyi anlaşılır.

$T_1$  süresi sabittir ve değişmez,  $T_2$  aralığı ise  $V_{INA}$  giriş gerilim örneklemelerin değeriyle orantılıdır. Aslında  $V_{INA}$  analog sinyalin mevcut seviyesi eğimin (yamacın) değerini doğrusal gerilime ayarlıyor: analog sinyalin seviyesi ne kadar yüksekse, eğim de o kadar büyük olacak. Buna göre açı da daha büyük olurken, sıfırinci seviyeye ulaşmak için gereken zaman  $T_2$  de daha uzun olacak, çünkü sinyalin sıfıra ulaşma eğildiği eğim sabittir. (8-21) denklemi, önceden saydığımız gerilim seviyeleri ve zaman aralıkları arasındaki bağlantıyı belirtiyor.

$$\frac{V_{INA}}{T_2} = \frac{V_{REF}}{T_1} \Rightarrow V_{INA} = V_{REF} \cdot \frac{T_2}{T_1} \quad (8-21)$$

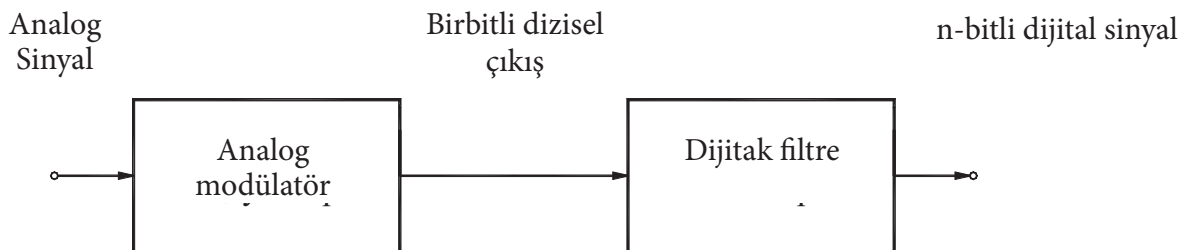
Bu yüzden  $T_2$  aralığı, sayaç tümleşik devre sıfırinci seviyeye ulaşana kadar saydığı, karşılaştırıcının tepki gösterdiği (çıkış seviyesini değiştiriyor) dönüşüm döngüsünün ikinci adımıyla ilgilidir. Bu adıma etkinleşen kontrol sinyali bu durumda şu işlemler için yönetim sinyalleri gönderiyor:

- ⊕ sayacın sıfırlanması için,
- ⊕ Pals sinyallerinin dürtülerini sayaca kadar geçirilmesi ve saymanın başlaması için,
- ⊕ ikili sözcüğün – dijital dönüşmüş analog örneklemin arabellekte korunması için ve
- ⊕ S analog anahtarının (1) durumuna geri dönmesi için.

Böylece yeni örnekleme alınabilir ve sıradaki dönüşüm döngüsü başlayabilir.

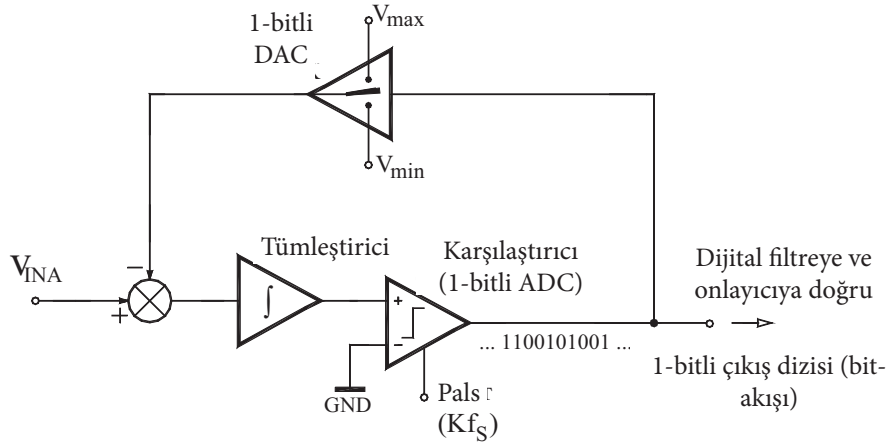
#### 8.5.4. DELTA-SİGMA A/D DÖNÜŞTÜRÜCÜLER

İlkesel bakış açısından, delta-sigma ( $\Delta\Sigma$ ) veya sigma-delta ( $\Sigma\Delta$ ) A/D dönüştürücüler (İng. delta-sigma ya da sigma-delta ADC), çift eğimli A/D dönüştürücülere biraz andırdığına rağmen, diğer A/D dönüştürücülerden oldukça farklıdır. Şek. 8-31'de gösterilen delta-sigma dönüştürücünün en basit blok diyagramı iki blok içeriyor. Birinci blok analog giriş sinyalini kabul eden ve sinyali büyük sayıda bitlerden oluşan uzun diziye dönüştüren analog modülatördür. Bu blok bir-bitli akış (İng. stream) üretiyor ve bu yüzden delta-sigma ( $\Delta\Sigma$ ) dönüştürücüler bir bitli dönüştürücüler olarak da biliniyor. İkinci blok onlayıcı olarak bilinen kısım içeren dijital AF filtresidir. Bu blok modülatörden gelen uzun bitler dizisini dijital çıkışa dönüştürüyor, yani n uzunlukta ikili sözcükler şekline dönüştürüyor.



Şek.8-31. Sigma-delta A/D dönüştürücünün blok diyagramı

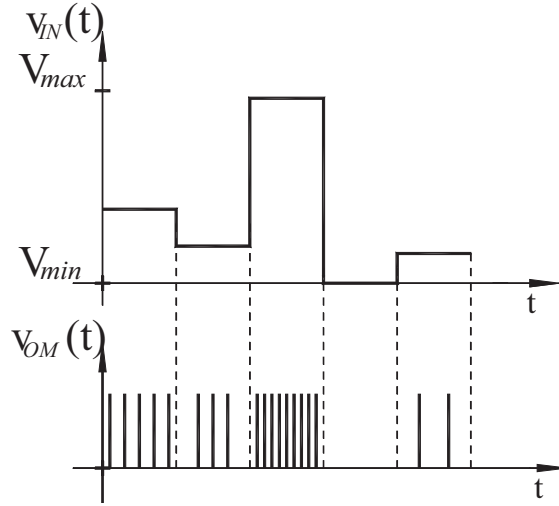
Analog delta-sigma modülörün çalışma prensibini, onun Şek.8-32'de verilmiş blok diyagramı yardımıyla açıklayacağız. Karşılaştırıcı tümleştirici çıkışı ile sıfırıncı seviye arasındaki farkı ( $\Delta$ ) tespit ediyor, tümleştirici ise karşılaştırıcının (birbitli D/A dönüştürücünün) çıkış gerilimleri ile analog giriş sinyali arasındaki farkları tümleştiriyor (topluyor,  $\Sigma$ ). Bu işlem, bir bitli D/A dönüştürücü üzerinden gerçekleşen negatif geri bağlantı sayesinde sağlanıyor. Onun çıkış seviyesinin,  $V_{R_{max}}$  ve  $V_{R_{min}}$  ile işaretlenen sadece iki farklı değeri olabilir. Bu şekilde giriş sinyalin en yüksek seviyesi sınırlanıyor. Böylece, örneğin  $V_{R_{min}} = 0V$  ve  $V_{R_{max}} = +5V$  ise, giriş gerilim seviyelerin kapsamı  $0V$  ile  $5V$  arasında olacak, ancak eğer  $V_{R_{min}} = -10V$  ve  $V_{R_{max}} = +10V$  ise, o zaman gerilimlerin giriş kapsamı  $-10V$  ile  $+10V$  arasında olacak



Şek. 8-32. Sigma-delta A/D dönüştürücüde modülörün blok-diyagramı

Delta-sigma dönüştürücünün özelliği karşılaştırıcının ya da bir bitli D/A dönüştürücünün çalıştığı puls frekansının büyük olmasıdır, öyle ki analog giriş sinyalinden aşırı sayıda örnekleme alınıyor. Bu yüzden bu dönüştürücüler aşırı sayıda örnekleme dönüştürücüler olarak da biliniyor (İng. over-sampling ADC). Bu yüzden, analog modülör çıkışında elde edilen bitler akışı,  $f_{CLK}$  frekansı  $f_s = 2 \cdot f_{max}$  seçme frekansına göre birkaç misli artmış dizisel bir-bitli sinyaldir. Bu frekans  $f_s$  Nyquist frekansından tam kat daha büyüktür:  $f_{CLK} = K \cdot f_s = K \cdot 2 \cdot f_{max}$ . Kısa zaman aralığı için böyle uzun ve yoğun bitler dizisinin en önemli özelliği, onların orta seviyesi bu zaman aralığı için giriş analog sinyalinin orta seviyesini tanımlamasıdır. Zaman biriminde daha büyük sayıda 1'ler, giriş analog sinyalinin daha yüksek seviyede olduğunu gösteriyor ve tersi. A/D dönüştürücünün çıkışı dijital olduğundan dolayı, en yüksek seviye çıkış dijital sözcüğünde tüm bitlerin 1 olmasıyla tanımlanıyor, en alçak seviye ise tüm bitlerin 0 olmasıyla belirtiliyor. 1'lerin sayısı ne kadar büyükse, yani bitler ne kadar daha yoğunsa, giriş analog sinyalin de seviyesi daha büyük olacak. Buna ters olarak, Şek.8-33'teki örnekte gösterilmiş olduğu gibi giriş seviyesi düşerse, 1'lerin sayısı da azalıyor. Buradan puls frekansı ne kadar daha büyükse, delta-sigma modülörün isabetliğin de o kadar daha büyük olacağı açıkça görülüyor.

Giriş sinyalinin seviyesini belirlemek için oldukça basit bit yöntem, belirli sabit zaman aralığında 1-leri sayab-n sayacın kullanılmasıdır. Sayam tamamlanca, sayacın çıkışları bu zaman aralığı için analog sinyalin dijital karşılığını tanımlayacak. Fakat, pratikte uygulanan delta-sigma dönüştürücülerde bir bitli bit akışın dijital çıkışa, yani ikili kodlanmış sözcüklere dönüştürülmesi için seyreltme (desimasyon) (İng. Decimation) olarak adlandırılan teknik uygulanıyor. Bu teknikle zamansal ayrıklaştırılmış giriş sinyalde örnekleme sayısının azalıyor.



Şek. 8-33. Birkaç örneklemenin ve modülatör çıkış sinyalin dalgalı şekilleri

Seyreltme ile aslında örnekleme frekansı azaldığından dolayı, Nyquist kriterin bozulmamasına dikkat edilmelidir. Bu yüzden A/D çıkışında modülatör önce dijital AF filtreyle, ardından onlayıcı ile bağlanıyor. AF filtresi sahte frekansların elenmesi için filtre olarak kullanılıyor (İng.anti-aliasing filter). Böylece sinyalin frekans kapsamı azalıyor, ardından seyreltici ile sinyalin örnekleme frekansının azalması yapıyor (downsampling). Bu şekilde en sonunda verilen zaman aralığında giriş sinyalin orta seviyesi elde ediliyor.

Delta-sigma A/D dönüştürücüler hassas ölçmeler için kullanılıyor ve bu yüzden ölçme aletlerine, sanayide ve başka alanlarda kullanılıyorlar.

## TEKRARLAMA SORULARI VE ÖDEVLERİ

- 8-1. Analog sinyaller .....tanımlıyor.
- 8-2. Dijital sinyaller .....tanımlıyor.
- 8-3. A/D dönüştürücünün en basit blok-diyagramını çiz ve onun rolünü yorumla.
- 8-4. A/D dönüşüm terimi altında..... tanımlanıyor.
- 8-5. D/A dönüşüm terimi altında..... tanımlanıyor.
- 8-6. D/A dönüştürücünün en basit blok-diyagramını çiz ve onun rolünü yorumla.
- 8-7. Bilgisayarın A/D veya D/A dönüştürücüyle bağlanmasının en basit blok-diyagramını çiz.
- 8-8. a)n-bitli çözünürlüklü genel durumda, b) n=3 bit çözünürlüklü, c) n=4 bit çözünürlüklü ideal D/A dönüştürücüde çıkış gerilim denklemini yaz.
- 8-9. a) n=3 bit, b)n=4 çözünürlüklü ideal D/A dönüştürücünün iletim özelliğini çiz.
- 8-10. (\*) Yüksek mantıksal seviyenin (1 seviyesi) (1) 5V, (2) 10V olduğunu tahmin ederek, 1V adımla  
a) 3-bitli, b) 4-bitli D/A dönüştürücü tasarla. Gerçekleştireceğin D/A dönüştürücünün çıkış gerilim kapsamı ne olacak?

- 8-11. (\*) Yüksek mantıksal seviyenin (1 seviyesi) (1) 5V, (2) 10V olduğunu tahmin ederek, 0,2V'luk adımla a) 3-bitli, b) 4-bitli D/A dönüştürücü tasarla. Gerçekleştireceğin D/A dönüştürücünün çıkış gerilim kapsamı ne olacak?
- 8-12. Ağırlıklı  $R/2^nR$  direnç ağı ve a)  $n=3$  bit, b)  $n=4$  bit, c) (\*)  $n=5$  bit çözünürlüklü D/A dönüştürücünün elektrik diyagramını çiz.
- 8-13. Basamaklı  $R/2R$  direnç ağı ve a)  $n=3$  bit, b)  $n=4$  bit, c) (\*)  $n=5$  bit çözünürlüklü D/A dönüştürücünün elektrik diyagramını çiz.
- 8-14. Ağırlıklı  $R/2^nR$  ve basamaklı  $R/2R$  direnç ağı D/A dönüştürücülerin iyi ve kötü taraflarını (avantajlarını ve dezavantajlarını) kıyasla.
- 8-15. Analog sinyalin zamana göre ayrıklaşması nedir?
- 8-16. Analog sinyalin nicemlenmesi (seviyeye göre ayrıklaşması) nedir?
- 8-17. 0 ile 5V kapsamında analog gerilim sinyalinin zamansal diyagramının bir kısmını çiz. a)  $n=3$  bit, b)  $n=4$  bit ile seviyeye göre ayrıklaştırılmış (nicemlenmiş) 5 örnekleme olarak onun zamana göre ayrıklaşmasını yap. Onun dönüştürülmüş dijital sinyalini de çiz.
- 8-18. Dönüşüm döngüsü.....dır.
- 8-19. Seçme frekansı.....dır.
- 8-20. Naykvist kriteri neden önemlidir?
- 8-21. İki analog sinyalin verilmiş olduğunu tahmin edelim: biri 0 ile 20KHz arasında frekans aralığı kapsayan ses sinyalidir, diğeri ise 0 ile 5MHz arası kapsamında video sinyalidir. Bu sinyeller için en alçak seçme frekansı nekadardır?
- 8-22. a)  $n=3$  bit,  $n=4$  bit çözünürlüklü ideal A/D dönüştürücünün iletim özelliğini çiz.
- 8-23. FS(R) tam (bütün) ölçek kapsamı nedir?
- 8-24. Nicemleyicinin çözünürlüğü nedir? Neye bağlıdır? Hangi birimlerde ifade ediliyor? Hangi denkleme göre hesaplanıyor?
- 8-25. Nicem (Nicemleme adımı) (Q) nedir? Neye bağlıdır? Hangi denkleme göre hesaplanıyor?
- 8-26. Nicemleme adımın (nicemin) küçük olması neden önemlidir? Yorumla!
- 8-27. Nicemleme hatası nedir? Hangi sınırlarda değişiyor?
- 8-28. İdeal A/D dönüştürücünün girişinde 0 ile 10V arasında gerilim analog sinyalin geldiğini tahmin edelim. A/D dönüştürücü a)  $n=3$  bitle, b)  $n=4$  bitle, c)  $n=12$  bitle, ç)  $n=16$  bitle dönüşüm gerçekleştiriyorsa nicemleme seviyelerin sayısını ve çözünürlük gerilimi (nicem büyüklüğünü) ile nicemleme hatasını hesapla.
- 8-29. A/D dönüştürücülerin yapıldığı temel süreç gruplarını say.
- 8-30. Paralel (flaş) A/D dönüştürücünün ilkesel diyagramını göz önüne alarak, onun çalışma prensibini açıkla.
- 8-31. a) 8-bitli, b) 12-bitli, c) 16-bitli flaş A/D dönüştürücünün gerçekleştirilmesi için nekadarda direnç gerekiyor? Kaç karşılaştırıcı gerekecek?
- 8-32. Sayaç rampalı, ardışık yaklaşıklama ve izlemeli A/D dönüştürücülerin ortak özellikleri nedir?

8-33. Sayaç rampalı A/D dönüştürücünün ilkesel diyagramını göz önüne alarak onun çalışma prensibini açıkla.

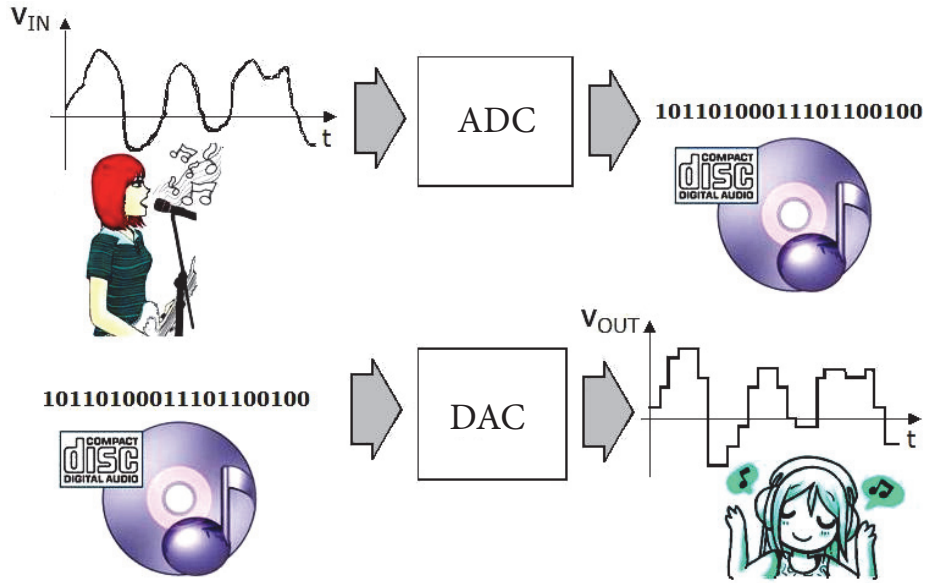
8-34. Ardışık yaklaşıklamalı A/D dönüştürücünün ilkesel diyagramını göz önüne alarak onun çalışma prensibini açıkla.

8-35. Hangi A/D süreçler tümlük devrenin uygulanmasına dayanıyor?

8-36. Tek eğimli A/D dönüştürücünün ilkesel diyagramını göz önüne alarak onun çalışma prensibini açıkla.

8-37. Çift eğimli A/D dönüştürücünün ilkesel diyagramını göz önüne alarak onun çalışma prensibini açıkla.

8-38. A/D dönüşümü için farklı süreçleri ve bu süreçleri gerçekleştiren A/D dönüştürücülerin avantajlarını ve dezavantajlarını kıyasla ve yorumla.



8-39 sorusunun şekli

8-39. (\*) Bir müzik stüdyosunda 44.1 kHz seçme frekansıyla dijital olarak CD için müzik kaydediliyor. a) (1) 3 dakika, (2) 4 dakika süren şarkının kaydedilmesi için nekadara bitlerle (b) ve baytlarla (B) ifade edilmiş bellek alanı gerekiyor? b) Bir CD'nin kapasitesi 700 MB ise, onda kaç dakika müzik kaydedilebilir? c) Analog ses sinyalinin amplitüdü 0 ile 5V arası sınırlarda değişiyorsa, nicemleme seviyelerinin sayısı, nicemleme adımı (çözünürlük gerilimi, nicem) ve nicemleme hatası ne kadardır?





## KAYNAKÇA

1. Balch, M. (2003). Complete Digital Design, *McGraw-Hill Companies, Inc.*
2. Floyd, T. (2006). Digital Fundamentals, Ninth Edition, *Pearson Prentice Hall.*
3. Holdsworth, B., Woods, C. (2003). Digital Logic Design, Fourth Edition, *Elsevier Private Ltd.*
4. Maini, A. (2007). Digital Electronics, *John Wiley & Sons Ltd.*
5. M. Morris, Mano, Charles Kime (2008). Logic and Computer Design Fundamentals, Fourth Edition, *Prentice Hall.*
6. Saha, A., Manna, N. (2007). Digital Principles and Logic Design, *Laxmi Publications Ltd.*
7. Сервини, Ј. (2008). Импулсна и дигитална електроника I (прв дел), трето издание, *Просвешно дело.*
8. Сервини, Ј., Дужевиќ, М. (2008). Импулсна и дигитална електроника II (втор дел), четврто издание, *Просвешно дело.* Скопје.

