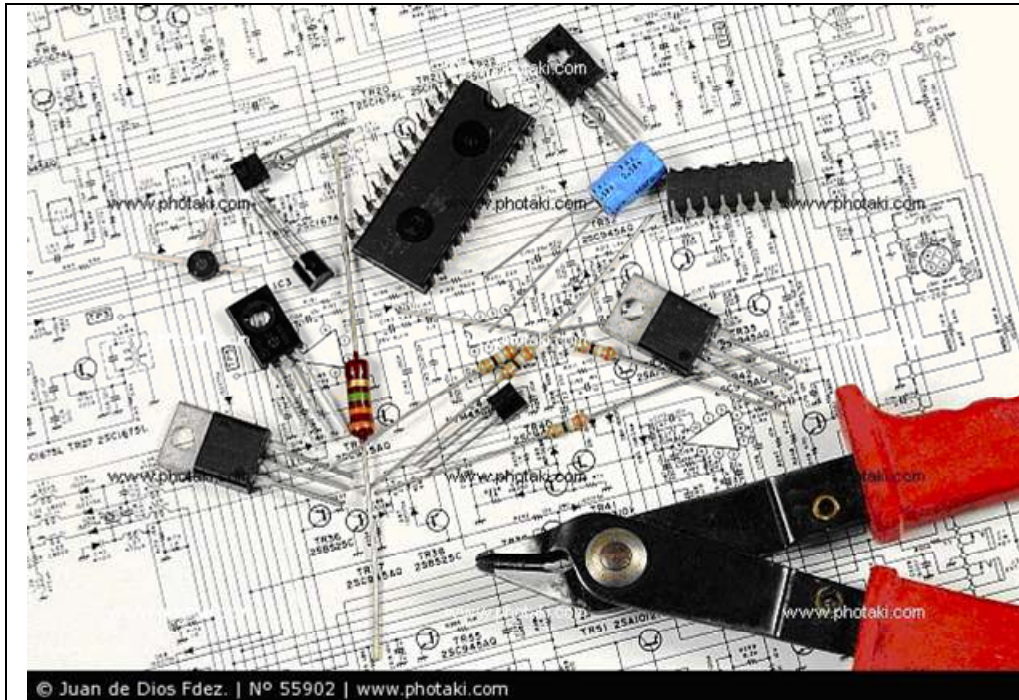


MSI devreler
(Orta Çaplı Entegre devreler)

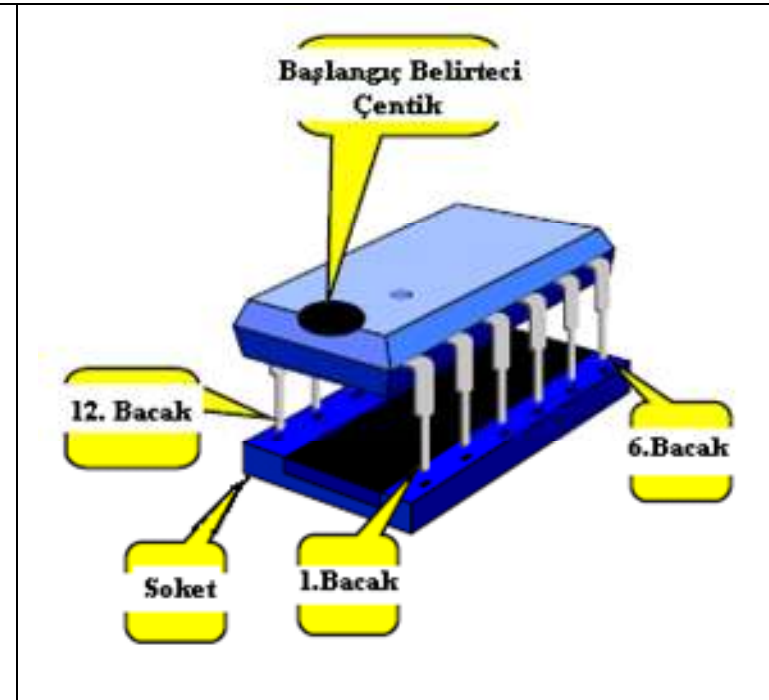
MSI devreler ve fonksiyonel yapıları

- 1940'lı yılların sonlarında yarıiletken devre elemanlarının ve transistörlerin bulunmasıyla bağlaçlar statik devreler olarak gerçekleştirilmeye başlandı.
- 1960'lı yıllarda küçük bir silisyum veya germanyum parça üzerinde birden fazla diyot transistör v.b elemanlar kimyasal yollarla oluşturulabilir hale geldi.
- Oluşan bu devrenin uygun yerlerinden dışarıya bacaklar (pinler) çıkartılır. Daha sonra yonga metal veya plastik bir kılıfla kaplanarak dış etkenlerden korunur. Böylece bir entegre devre elde edilmiş olur.
- Böylelikle sayısal devre elemanları entegre devreler şeklinde üretilmeye başlandı. Bu yapı; entegre devre, chip veya yonga diye isimlendirilir.

Entegre devreler

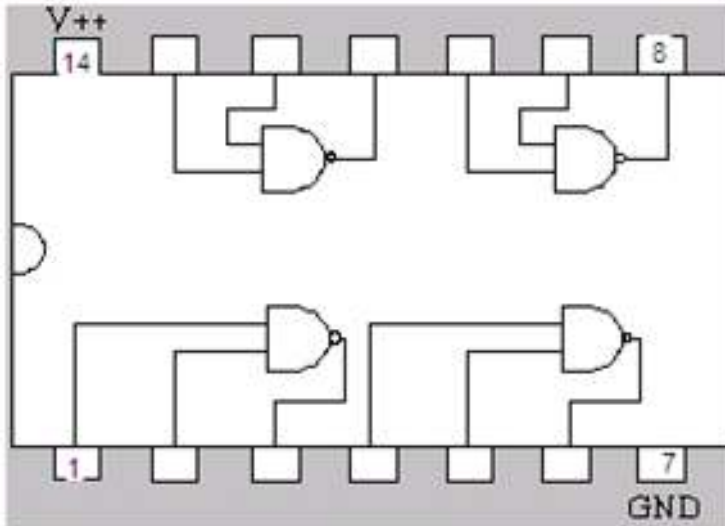


© Juan de Dios Fdez. | N° 55902 | www.photaki.com

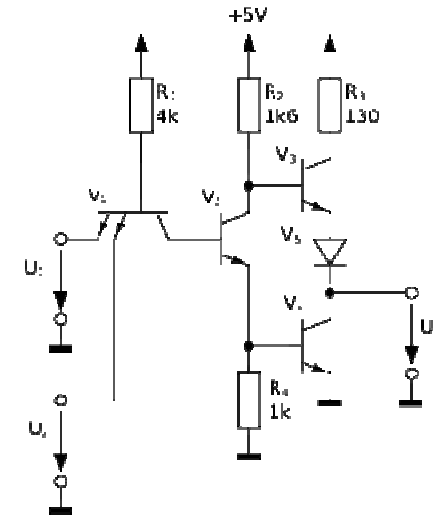


SSI (Small Scale IC - Küçük boyutlu IC

İçinde en fazla 9 adet kapı elemanı bulunan IC'lerdir. Her bağlacın giriş ve çıkışları birbirinden bağımsız şekilde IC'nin dışına çıkartılmıştır. Bu elemanlar tek başına bir işlem yapamazlar. Kombinasyonel lojik devrelerin gerçekleştirilmesin de çok kullanılırlar.



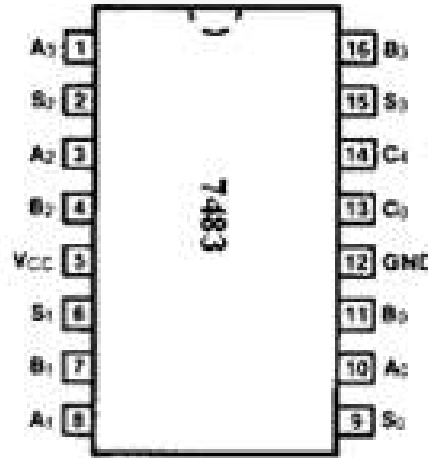
a) NAND bağlacının bağlantı diyagramı



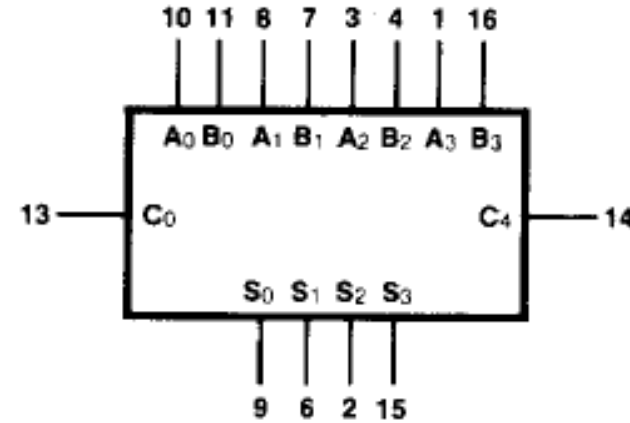
b) 2 girişli NAND bağlacının yapısı

MSI(Medium Scalae IC-Orta boyutlu IC):

- Bu yapıdaki IC'lerin içinde yaklaşık 10-100 adet bağlaca denk gelen eleman vardır. Bu şekilde oluşturulan IC'ler küçük çaplı sonsuz tane problemi tek başına çözebilecek durumdadırlar. MUX, DEMUX, Toplayıcı, Sayıcı, Kodlayıcı, karşılaştırıcı v.b devreleri bunlara örnek gösterebiliriz. Aşağıdaki şekillerde 4 bitlik paralel toplayıcı Entegre devrenin bağlantı uçları, lojik sembol ve lojik diyagramı görülmektedir. Sayısal tasarım dersinin önemli konularından biriside MSI devrelerin önemlilerinden bazılarının fonksiyonel davranışlarının öğrenilmesi ve bunların sayısal tasarımda kullanılmasıdır.

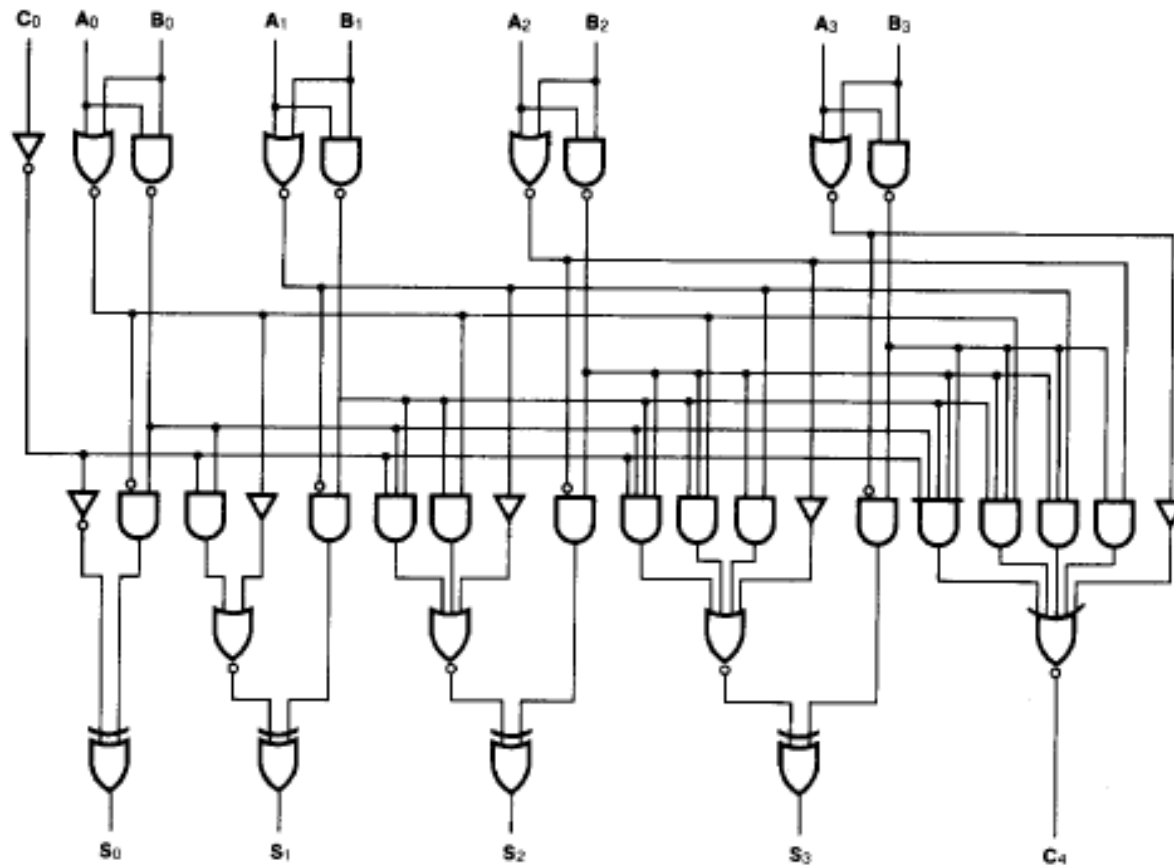


a



b

4 bitlik toplayıcının iç yapısı

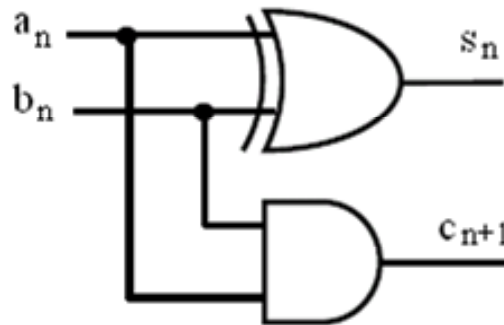


- **LSI(Large Scalae IC): Büyük kapasiteli IC:** Bu IC'ler büyük çaplı karmaşık sayısal problemleri çözmek için üretilirler. Örneğin, İlk Mikroİşlemciler, 1Kb RAM, hesap makinalarının temel IC'leri, saat IC'leri v.b. İçlerinde binlerce transistör gerçekleştirilmiştir.
- **VLSI(Very Large Scalae IC): Çok büyük çaplı IC:** İçlerinde milyonlarca transistör gerçekleştirilmiş olan bu entegreler 1980'li yıllarda geliştirilmiştir. Çoğu zaman yazılım destekli olarak sonsuz tane sayısal problemi çözerler. Mikroİşlemciler, Büyük kapasiteli RAM hafıza devreleri bunlara örnek verilebilir.
- **ULSI(Ultra Large Scalae IC): Ultra büyük çaplı IC:** Milyonlarca kapı eşdeğeri bulunan IC devrelerdir. Büyük bilgisayar işlemcileri v.b sayısal devrelerin oluşturulduğu entegre devrelerdir

TOPLAYICI MSI devreler (FULL ADDERS)

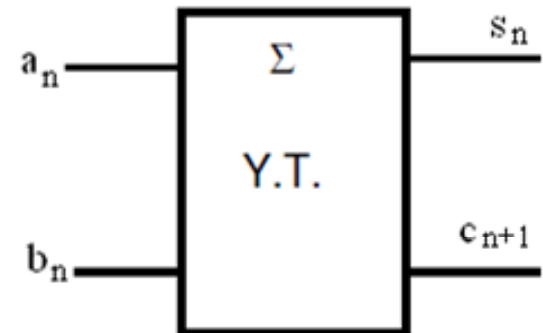
- n bitlik binary sayıları toplamak için kullanılan devrelerdir. Bu devreler MSI entegre devre şeklinde gerçekleştirilir. TTL familyasında en çok kullanılan 4 bitlik paralel toplayıcı entegresi devresi 7483 veya 74283 entegreleri bu iş içindir.
- Karmaşık Sayısal devreler tasarlanırken, devre mümkün olan bir birine benzeyen küçük birimlerden oluşturulur. Toplayıcı devrelerin en küçük birimi ise yarım toplayıcı (half adders)'lardır. Yarım toplayıcılar en sağdaki basamakta işlem yapan toplayıcılardır. Dolayısıyla elde biti girişleri yoktur.
- Yarım toplayıcı ile toplanacak bitler a_n ve b_n olsun. s_n sonuç biti ve c_{n+1} ise oluşan elde biti çıkışı olsun. Buna göre yarım toplayıcının tanım tablosu ve fonksiyon denklemi aşağıdaki gibidir.

a_n	b_n	s_n	c_{n+1}
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1



$$s_n = a_n' \cdot b_n + a_n \cdot b_n' = a_n \oplus b_n$$

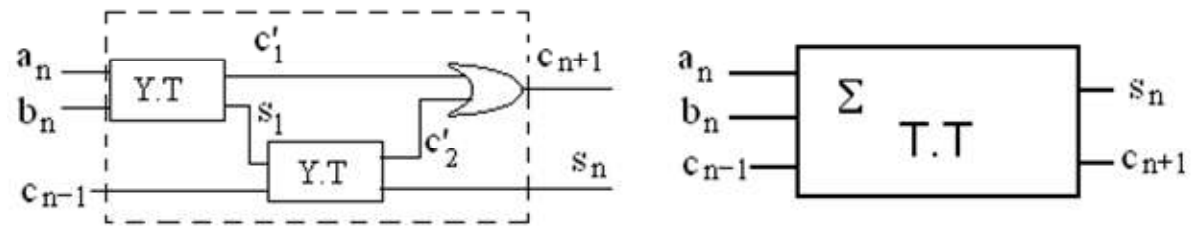
$$c_{n+1} = a_n' \cdot b_n$$



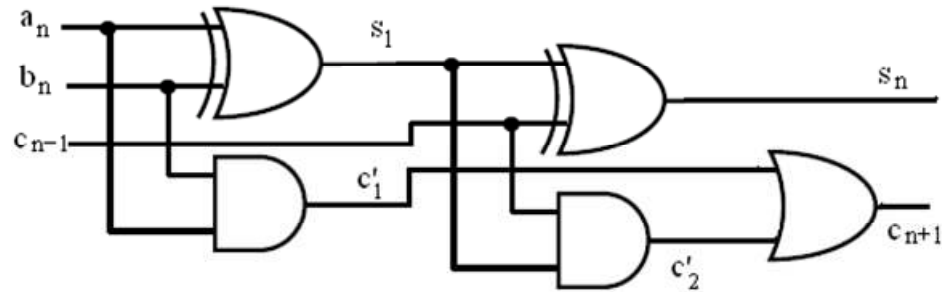
- Her hangi bir basamakta birer bitlik iki sayıyı, alt basamaktan gelen elde bitini de göz önüne alarak toplayan tam toplayıcının tanım tablosu ise aşağıda verilmiştir. Burada c_{n-1} girişi bir alt basamaktan gelen elde biti girişidir. Bu tanım tablosunu bu şekilde gerçekleştirmek yerine; Bir tam toplayıcının 2 tane yarım toplayıcıdan oluşturulması teknik olarak daha uygundur. Bunun matematiksel gösterimi aşağıdadır.
- $$a_n + b_n = s_1 \text{ (elde } c_1')$$
$$s_1 + c_{n-1} = s_n \text{ (elde } c_2') \text{ olur.}$$
- Görüldüğü gibi toplama işlemi iki girişli yarım toplayıcılarla yapılabilmektedir. Aşağıda bunun blok şeması verilmiştir.

Tam toplayıcının 2 tane yarım toplayıcıdan oluşması

a_n	b_n	c_{n-1}	s_n	c_{n+1}
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

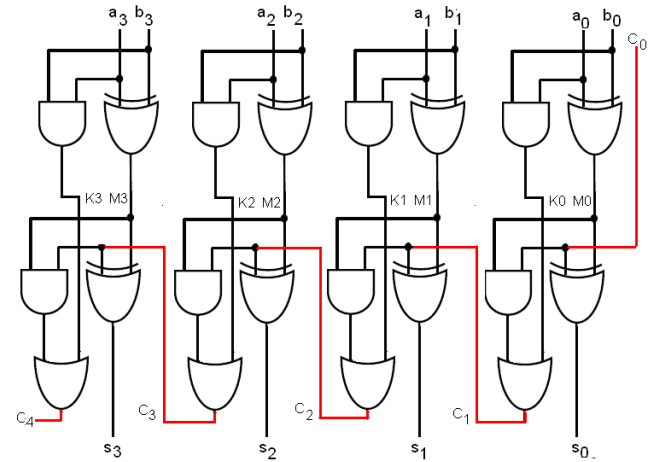
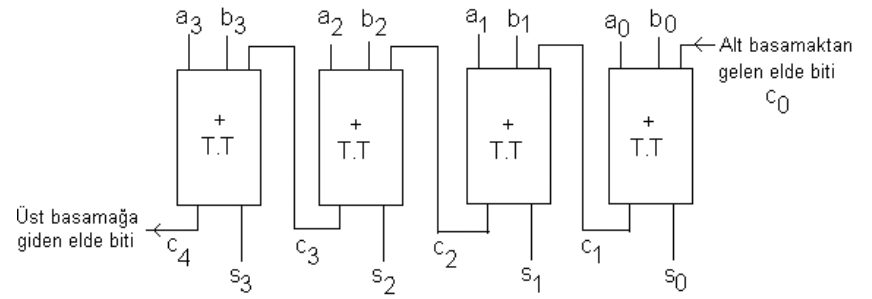


Tam toplayıcının iki yarım toplayıcıdan oluşturulması.



Yarım toplayıcılardan oluşan tam toplayıcının lojik şeması.

- Toplama işlemlerini gerçekleştiren temel devre bir bitlik tam toplayıcı devresidir.
- Toplayıcılar iki değişik şekilde gerçekleştirilir. Paralel toplayıcılar, bir bitlik tam toplayıcıların paralel bağlanmasıyla elde edilen bir birim zamanda n bitlik iki sayıyı toplayıp sonucu veren sayısal donanımlardır.
- Seri toplayıcılar ise n bitlik sayıları n birim zamanda toplayan, tek bir tam toplayıcı kullanılan devrelerdir.
- 4 bitlik paralel toplayıcının blok şeması verilmiştir.
- Eğer 4 bit'den daha uzun sayıların paralel olarak toplanması isteniyorsa, paralel toplayıcıları kaskad bağlamak gerekir.



4 bitlik paralel toplayıcı uygulaması:

- **4 bitlik BCD kodlanmış sayıları toplayıp düzelten toplama devresini 4 bitlik paralel toplayıcılar ve gerekli bağlaçları kullanarak tasarlayınız.**
- **Çözüm:** $a(a_3 a_2 a_1 a_0)$, $b(b_3 b_2 b_1 b_0)$ şeklindeki BCD sayılarının toplanmasında sonuç her zaman doğru çıkmıyordu. Önceki bölümlerde verilen kuralları hatırlarsak;
- **1-) $0 \leq s' \leq 9$** ise sonuç doğru. Düzeltmeye gerek yok (Sonuca 0(0000) ekle.
- **2-) $10 \leq s' \leq 15$** ise sonuç yanlış. Sonuca 6(0110) ekle. Oluşan elde bitini üst basamağa gönder.
- **3-) $s' \leq 16$** ise sonuç yanlış. Elde bitini üst basamağa gönder. Sonucu 6(0110) ile topla.
- Bu kurallara göre ara sonucu değerlendirerek sonuca 0 veya 6 eklenmelidir. Sadece 1. ve 2. kuralları sağlayan tanım tablosu aşağıdaki gibi oluşturulur. Burada bahsedilen ara sonuç $s'(s_3', s_2', s_1', s_0')$, $\zeta_3', \zeta_2', \zeta_1', \zeta_0'$ ise ara sonucun toplanacağı 4 bit'dir. Tanım tablosundan da görüldüğü gibi ara sonucun MSB ve LSB bitleri devamlı 0 ile toplanır. Diğer 2 bit ise ya 0 ile ya da 1 ile toplanır.

$$\begin{array}{r}
 a_3 a_2 a_1 a_0 \\
 + b_3 b_2 b_1 b_0 \\
 \hline
 e' s_3' s_2' s_1' s_0'
 \end{array}$$

9 veya 9'dan küçük mü?
 9'dan büyük 16'dan küçük mü ?
 16'dan büyük mü?

s_3'	s_2'	s_1'	s_0'	ζ_3'	ζ_2'	ζ_1'	ζ_0'
0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	0
0	0	1	0	0	0	0	0
0	0	1	1	0	0	0	0
0	1	0	0	0	0	0	0
0	1	0	1	0	0	0	0
0	1	1	0	0	0	0	0
0	1	1	1	0	0	0	0
1	0	0	0	0	0	0	0
1	0	0	1	0	0	0	0
1	0	1	0	0	1	1	0
1	0	1	1	0	1	1	0
1	1	0	0	0	1	1	0
1	1	0	1	0	1	1	0
1	1	1	0	0	1	1	0
1	1	1	1	0	1	1	0

$s_3' s_2'$				
$s_1' s_0'$	00	01	11	10
00	0	0	1	0
01	0	0	1	0
11	0	0	1	1
10	0	0	1	1

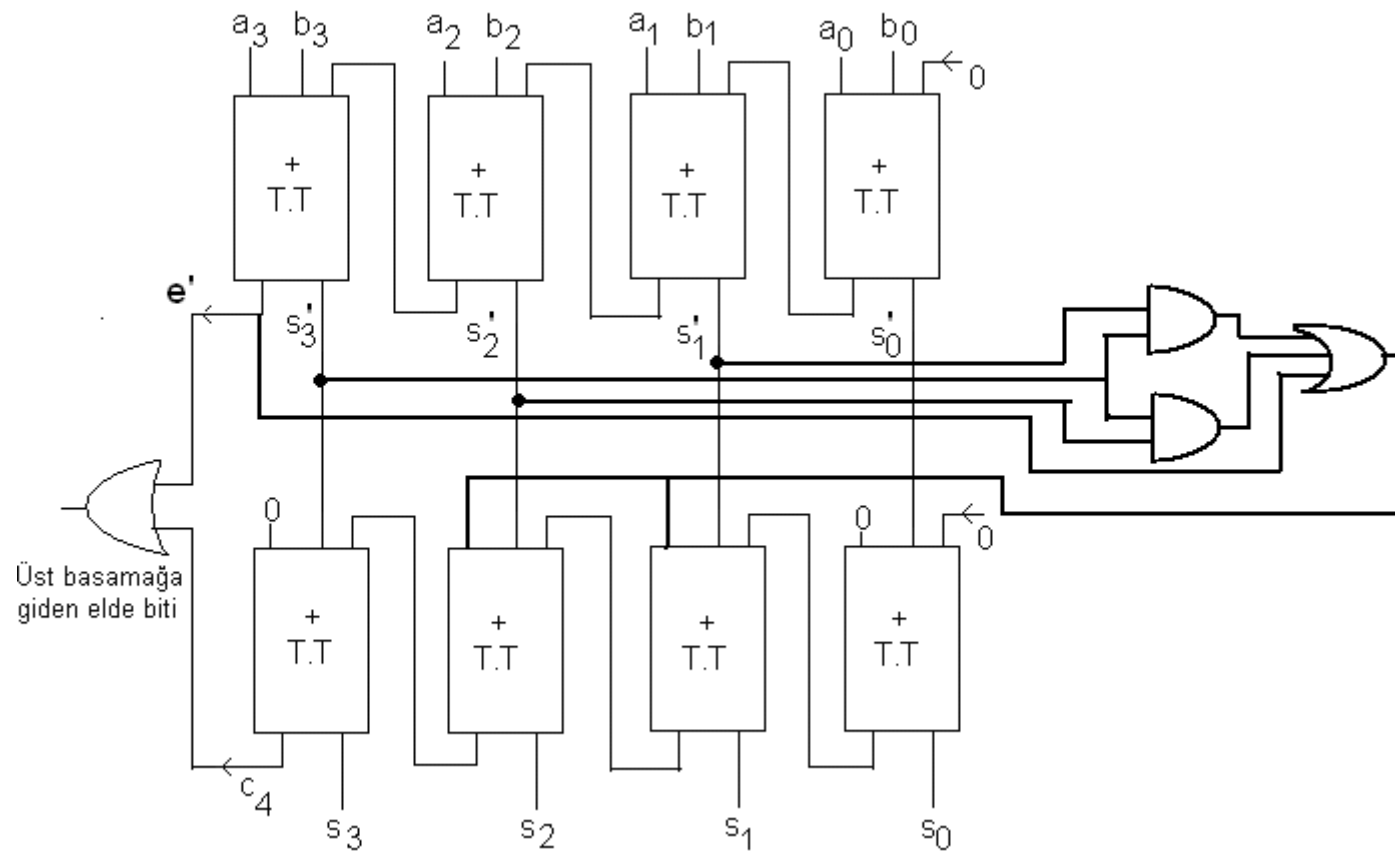
Kısaltılmış fonksiyon denklemi,

$$\zeta_2' = \zeta_1' = s_3' . s_2' + s_3' , s_1'$$

şeklindedir. 3.kuralı gerçekleştirmek için e' elde bitini üstteki denkleme VEYA bağlacı ile eklemek yeterlidir. Buna göre düzeltme devresi;

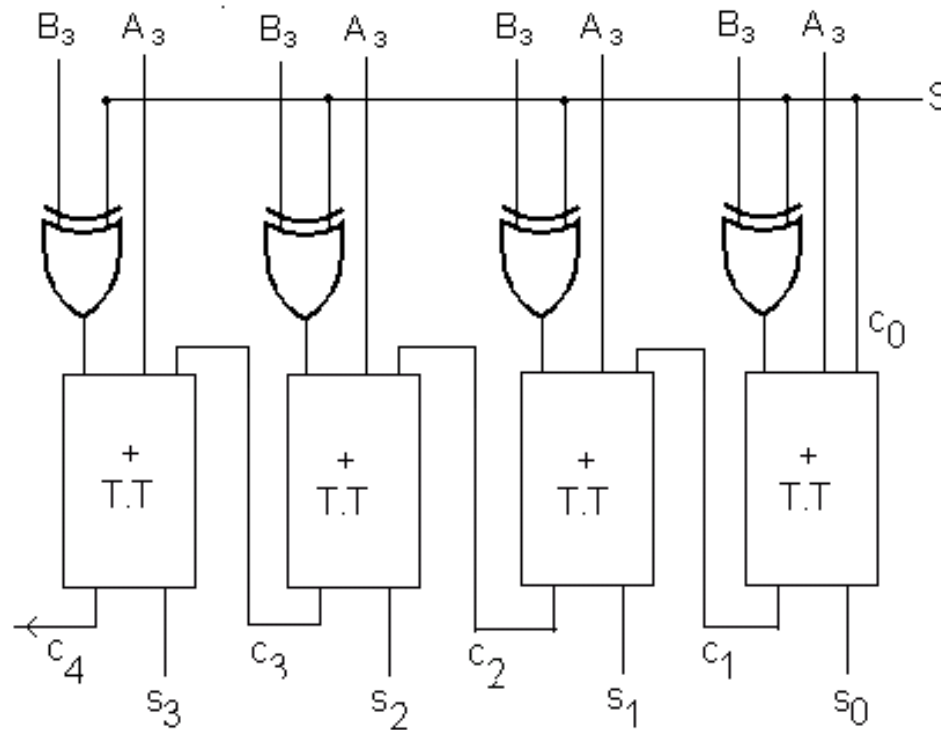
$$\zeta_2' = \zeta_1' = s_3' . s_2' + s_3' , s_1' + e'$$

şeklinde olur. Düzeltme devresi ile birlikte BCD toplayıcının lojik şeması Şekil 7.8'de verilmiştir.



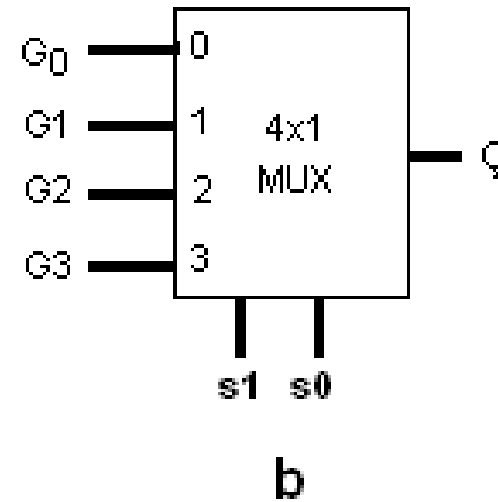
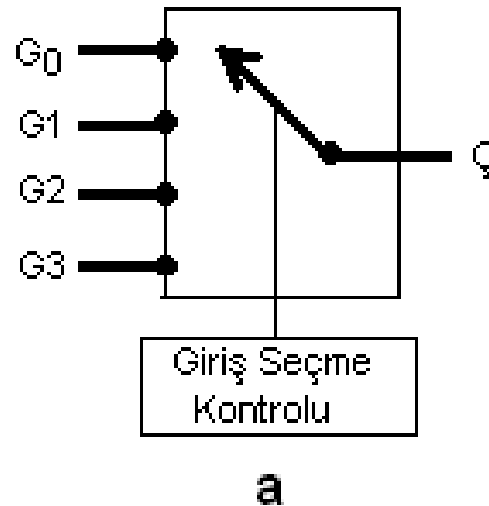
4 bitlik paralel toplayıcı uygulanması – 2

- 4 bitlik toplama ve tamamlayıcı toplama yoluyla çıkarma yapan devre aşağıda verilmiştir.
- Burada S girişi toplama veya çıkarma işlemi seçme ucudur. S=0 olduğu zaman A+B işlemi 4 bitlik paralel toplayıcı tarafından gerçekleştirilir.
- S=1 olduğu zaman ise A-B işlemini tamamlayıcı toplama yoluyla yapmalıdır. Yani B sözcüğünün taban -1' göre tümleyenini alarak A sözcüğü ile toplamalı ve 1 eklenmelidir.
- Aşağıdaki devrede ex-or bağlaçları tümleme için kullanılabilir. Sonucun 1 ile toplanması ise ilk basamağın elde girişine S'yı uygulamakla sağlanır.



Multiplexer (MUX – Çoktan bire çoğullayıcılar)

- Şekill a'daki gibi bir mekanik anahtar düşünelim. Bu anahtar herhangi bir girişe bağlanırsa, çıkışa sadece anahtarla bağlanmış giriş aktarılır. Giriş seçme kontrol devresi ile herhangi bir giriş herhangi bir zamanda çıkışa aktarılabilir. Bu yapı Çoktan bire çoğullayıcı bir devredir. N tane girişten yalnızca 1 tanesini istenile süre içerisinde çıkışa yönlendirmek için kullanılır. Şekil.b'de sayısal Mux'un prensip şeması görülmektedir. Burada sayısal Multiplexer'lar üzerinde durulacaktır.
- Sayısal Mux'lar değişik kanallardan gelen sayısal bilgileri tek çıkış kanalına yönlendirmek için kullanılırlar. Örneğin 4 farklı kanaldaki sayısal veriyi tek bir iletim kanalı üzerinden farklı yerlere ulaştırmak için kullanılabilir.



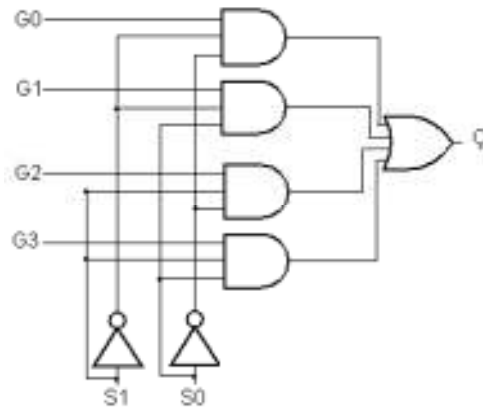
4x1 MUX yapısı

s_1	s_0	G_0	G_1	G_2	G_3	ζ
0	0	0	x	x	x	0
0	0	1	x	x	x	1
0	1	x	0	x	x	0
0	1	x	1	x	x	1
1	0	x	x	0	x	0
1	0	x	x	1	x	1
1	1	x	x	x	0	0
1	1	x	x	x	1	0

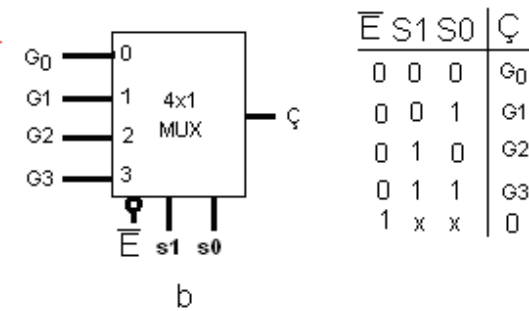
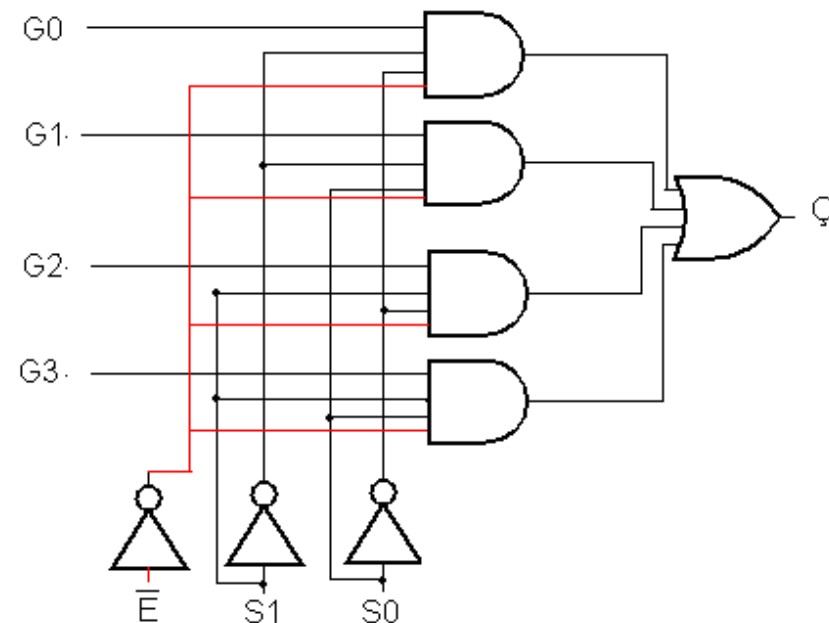
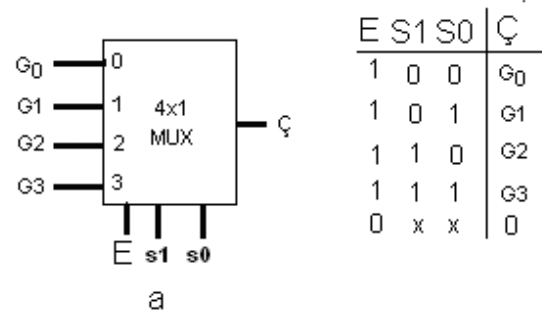
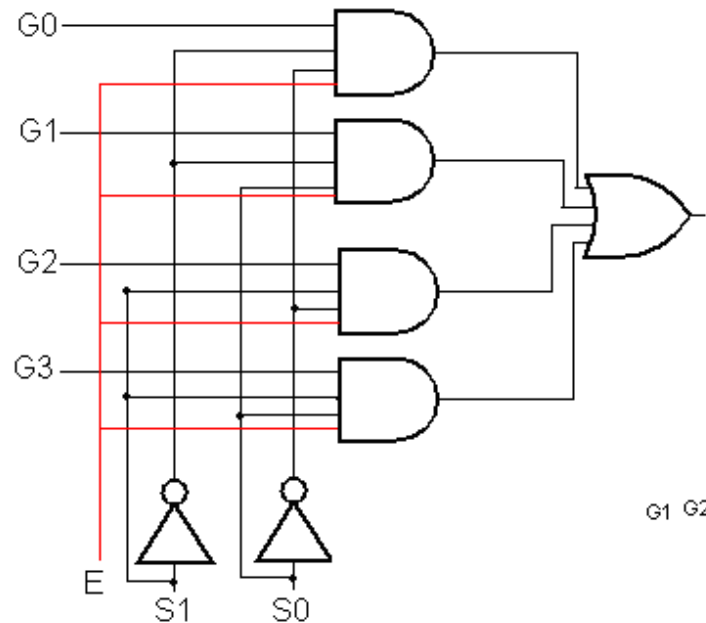
Tanım tablosunu aşağıdaki şekilde yazabiliriz.

s_1	s_0	ζ
0	0	G_0
0	1	G_1
1	0	G_2
1	1	G_3

$\zeta = s_1'.s_0'.G_0 + s_1'.s_0.G_1 + s_1.s_2'.G_3 + s_1.s_2.G_3$
Bu denkleme göre lojik şema ise Şekil 7.11'dedir.



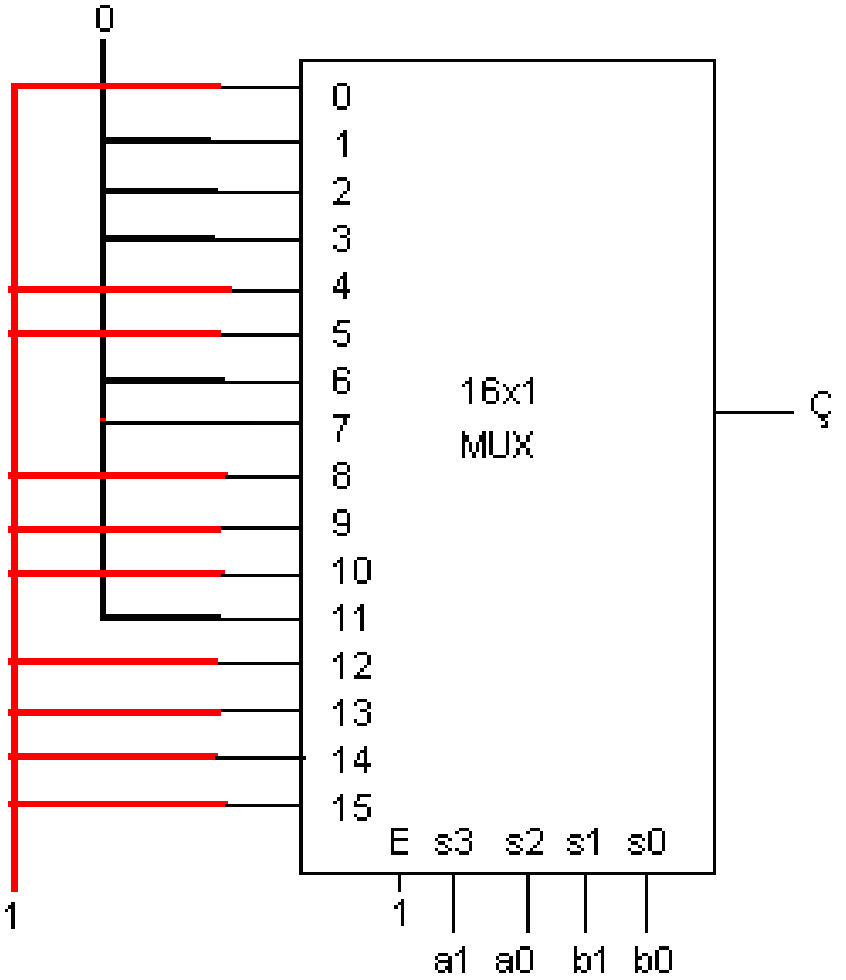
4x1 Mux'un yapısı (Enable giriřli)



Multiplexer Uygulamaları:

- Mux'un temel işlevinin, çok sayıda giriş kanalından yalnızca seçilenini çıkışa yönlendirmek olduğu yukarıdaki bölümlerde ifade edilmişti. Burada MUX'ların kombinasyonel devre tasarımında kullanılmasına ait örnekler verilecektir.
- **Uygulama 1:** $a(a_1 a_0)$ ve $b(b_1 b_0)$ ikişer bitlik sayılar olduğuna göre, $a \geq b$ olduğu durumlarda çıkışı 1 olan lojik devrenin tanım tablosunu yazarak 16:1 Mux ile gerçekleştiriniz

a_1	a_0	b_1	b_0	C
00	00	00	00	1
00	00	00	11	0
00	00	11	00	0
00	00	11	11	0
00	11	00	00	1
00	11	00	11	1
00	11	11	00	0
00	11	11	11	0
11	00	00	00	1
11	00	00	11	1
11	00	11	00	1
11	00	11	11	0
11	11	00	00	1
11	11	00	11	1
11	11	11	00	1
11	11	11	11	1

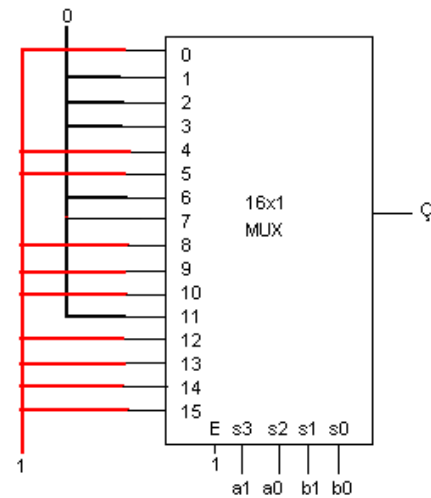


- Örnek uygulamadan da anlaşıldığı gibi, eğer problemin giriş değişken sayısı, kullanılan MUX'un giriş seçme ucu sayısına eşitse, başka ek devre kullanılmadan devre çözülebilir. Giriş kanallarına, problemin tanım tablosundaki ilgili kombinasyonlar için belirlenmiş çıkış değerleri girilir ve seçme ucuna da giriş değişkenlerinin uygulanması ile çözüm tamamlanmıştır.

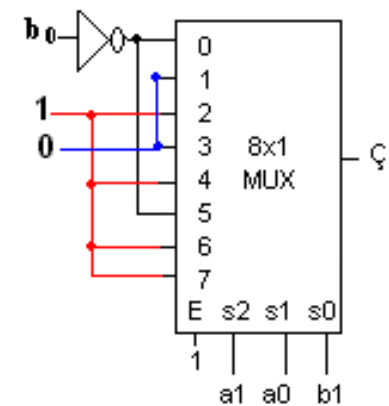
Uygulama 2: Aynı problemi 8x1, 4x1, 2x1 Mux kullanarak ayrı ayrı çözüünüz. Yorumlayınız.

- **Çözüm:** Eğer kullanılacak Mux'un giriş seçme ucu sayısı (ss), problemin değişken sayısından (n) eksik ise yeni tanım tabloları hazırlanır. Tablolarda **n-ss** kadar en ağırlıksız değişkenden başlanmak üzere değişken dışarıda bırakılır. Tablolarda kullanılmayan değişken veya değişkenlere bağlı olarak sonuç kısmı elde edilir. cinsinden ifade edilir. Bunlara göre şema çizilir.

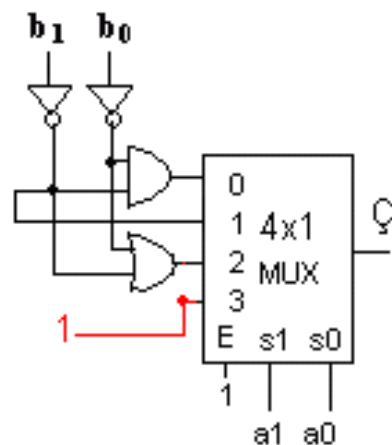
a_1	a_0	b_1	b_0	ζ
0	0	0	0	1
0	0	0	1	0
0	0	1	0	0
0	0	1	1	0
0	1	0	0	1
0	1	0	1	1
0	1	1	0	0
0	1	1	1	0
1	0	0	0	1
1	0	0	1	1
1	0	1	0	1
1	0	1	1	0
1	1	0	0	1
1	1	0	1	1
1	1	1	0	1
1	1	1	1	1



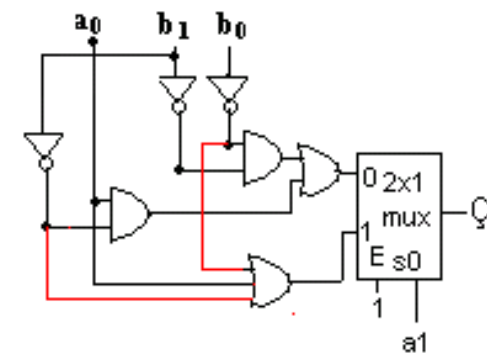
a_1	a_0	b_1	ζ
0	0	0	b_0'
0	0	1	0
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	b_0'
1	1	0	1
1	1	1	1



a_1	a_0	ζ
0	0	$b_1' \cdot b_0'$
0	1	b_1
1	0	$b_1' + b_0'$
1	1	1



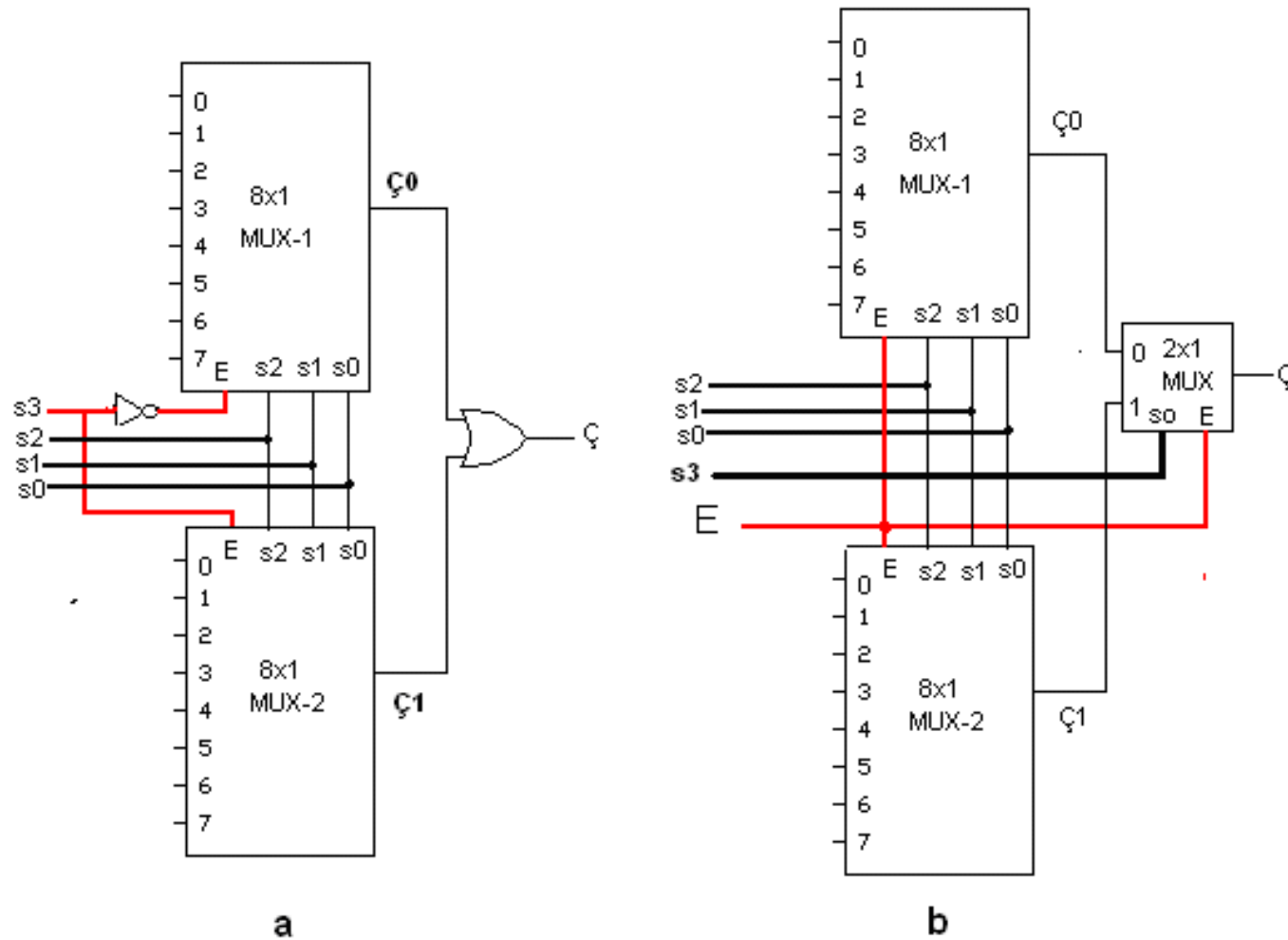
a_1	ζ
0	$b_1' \cdot b_0' + a_0 \cdot b_1$
1	$a_0 + b_1 + b_0'$



Uygulama 3: Düşük giriş kanallı Mux'lar kullanılarak daha fazla giriş kanallı Mux'ların oluşturulması.

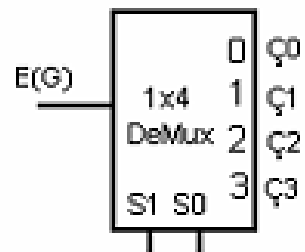
- Bazı uygulamalarda çok sayıda giriş kanalı olan Multiplexerlara ihtiyaç duyulur. Mux'ların giriş kanal sayısını yükseltmek için değişik uygulamalar mevcuttur. Bunlardan 2 tanesi Şekilde verilmiştir. Şekil.a'da 2 adet 8x1 Mux'tan 1 adet 16x1 Mux'un elde edilmesi verilmiştir. 16x1 Mux'un 4 adet giriş seçme ucu olmalıdır. Bunun için 8x1 Muxların Enable girişleri bir değil bağlacı ile birleştirilmiş ve 4.giriş seçme ucu olarak değerlendirilmiştir.
- Şekil.a'daki yapının dezavantajı, Enable uçlarının başka fonksiyon için kullanılmasıdır. Şekilde ise Enable uçları da olan bir 16x1 Mux yapısı görülmektedir.

16x1 Mux'un 8x1 Muxlar ile gerçekleştirilmesi



Demultiplexer/Decoder (birden Çoğa çoğullayıcı/ Kod açıcılar)

- Demultiplexer (DEMUX), MUX'un tersi işlevi yapar. Yani tek bir giriş kanalı fazla sayıda çıkış kanallarından, seçilen bir tanesine yönlendirilir. DeMUX'ların giriş kanalı ucu, çıkış kanalları ve çıkış seçme uçları mevcuttur. Herhangi bir anda seçilen çıkış kanalına girişi yönlendirir. Bu fonksiyonları yerine getiren 1x4 Demux'un tasarım bilgileri şekilde gösterilmektedir.



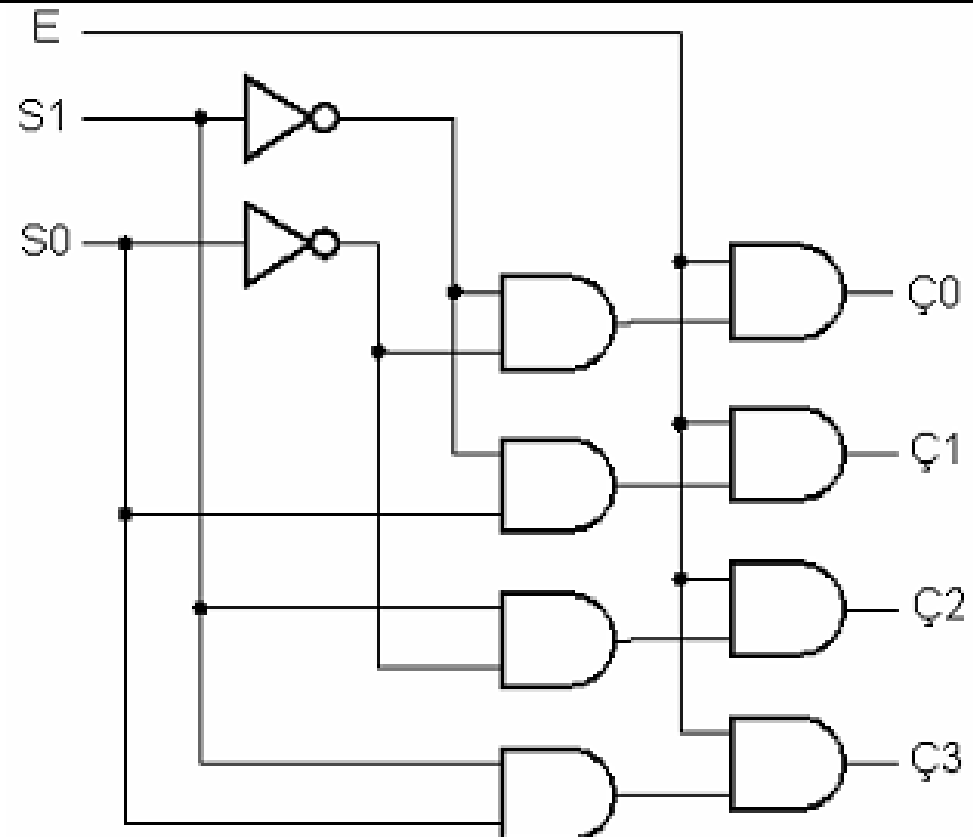
E	S1	S0	Ç0	Ç1	Ç2	Ç3
0	X	X	0	0	0	0
1	0	0	1	0	0	0
1	0	1	0	1	0	0
1	1	0	0	0	1	0
1	1	1	0	0	0	1

$$\text{Ç0} = E \cdot S1' \cdot S0'$$

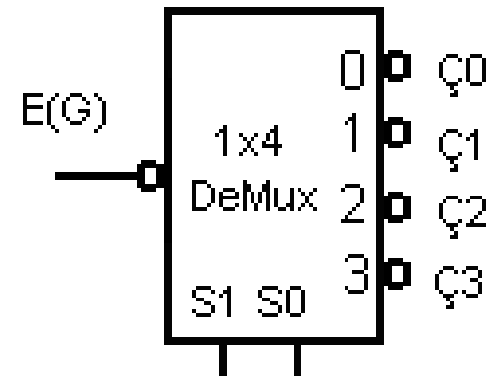
$$\text{Ç1} = E \cdot S1' \cdot S0$$

$$\text{Ç2} = E \cdot S1 \cdot S0'$$

$$\text{Ç3} = E \cdot S1 \cdot S0$$



E	S1	S0	Ç0	Ç1	Ç2	Ç3
1	X	X	1	1	1	1
0	0	0	0	1	1	1
0	0	1	1	0	1	1
0	1	0	1	1	0	1
0	1	1	1	1	1	0



Demuxun seçilmeyen çıkışları Lojik 1 olacak şekilde düzenlenmiştir. Demuxların temel işlevi tek giriş kanalını değişik çıkışlara yönlendirmektir.

Örneğin tek bir iletişim kanalını paylaşan gönderici ve alıcı sisteminin temel prensibi aşağıda verilmiştir. Bu yapı zaman paylaşımli çoğullamalı (TDM) iletişimin prensibidir. Giriş aboneleri tek bir seri iletişim hattını kullanarak zaman paylaşımli olarak çıkış aboneleriyle haberleşirler. Burada kimin ne kadar süreyle kiminle haberleşeceği giriş ve çıkış kontrol devreleriyle sağlanır.

Bu dersin konusu iletişim üzerine olmadığından üzerinde fazla durulmayacaktır.

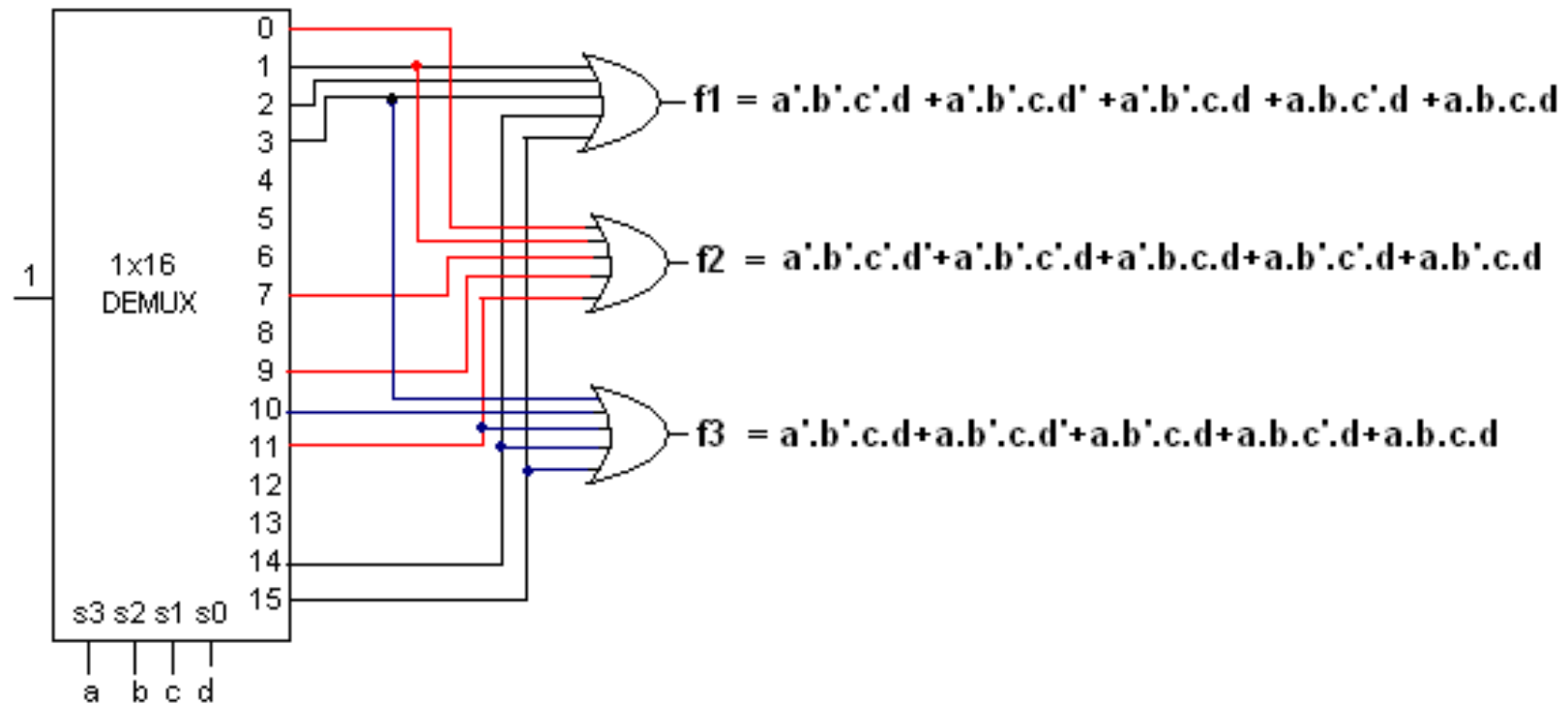
Daha çok bir DEmux'la kombinasyonel devre gerçekleştirme uygulamaları üzerinde durulacaktır.

Örnek



Uygulama -1: $f1(a,b,c) = \sum 1,2,3,14,15$, $f2(a,b,c) = \sum 0,1,7,9,11$, $f3(a,b,c) = \sum 3,10,11,14,15$ çok girişli çok çıkışlı devreyi 1x16 Demux kullanarak gerçekleştiriniz.

•**Çözüm:** Burada 1x16 Demuxun 4 tane çıkış seçme ucu bulunduğundan, problemin değişkenleri çıkış seçme uçlarına uygulanır. İlgili çıkışlar ise VEYA'lanarak problem çözülmüş olur. Aşağıda problemi çözen lojik şema görülmektedir



Uygulama -2: 1x8 Demux ve 8 Adet 8x1 MUX kullanarak 64x1 Mux'un gerçekleştirilmesi

