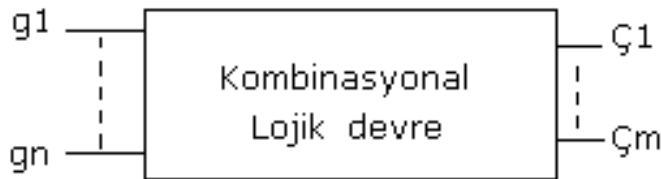


# Kombinasyonel lojik devre tasarımı

# Kombinasyonel Lojik devreler

- Lojik (Mantık) devreleri de denilen Sayısal devreler iki ana kategoride incelenir. Bunlar kombinasyonel (Birleşimsel) ve Ardışıl devreler olarak isimlendirilir. Kombinasyonel devrelerde çıkış, sadece girişin fonksiyonu olduğu halde,
- Ardışıl devrelerde çıkış hem girişin, hem de çıkışın daha önceki zamanlardaki değerlerinin fonksiyonunu



$$\text{Ç} = f(g)$$

a) Kombinasyonel Lojik Devre



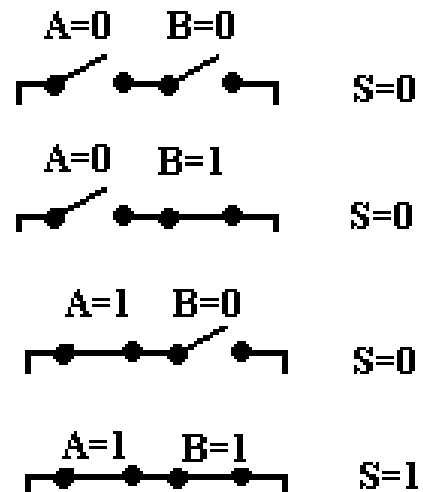
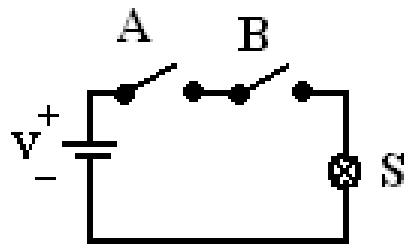
$$\text{Ç} = f(g, \text{Ç}_{\text{önce}})$$

b) Ardışıl Lojik Devre

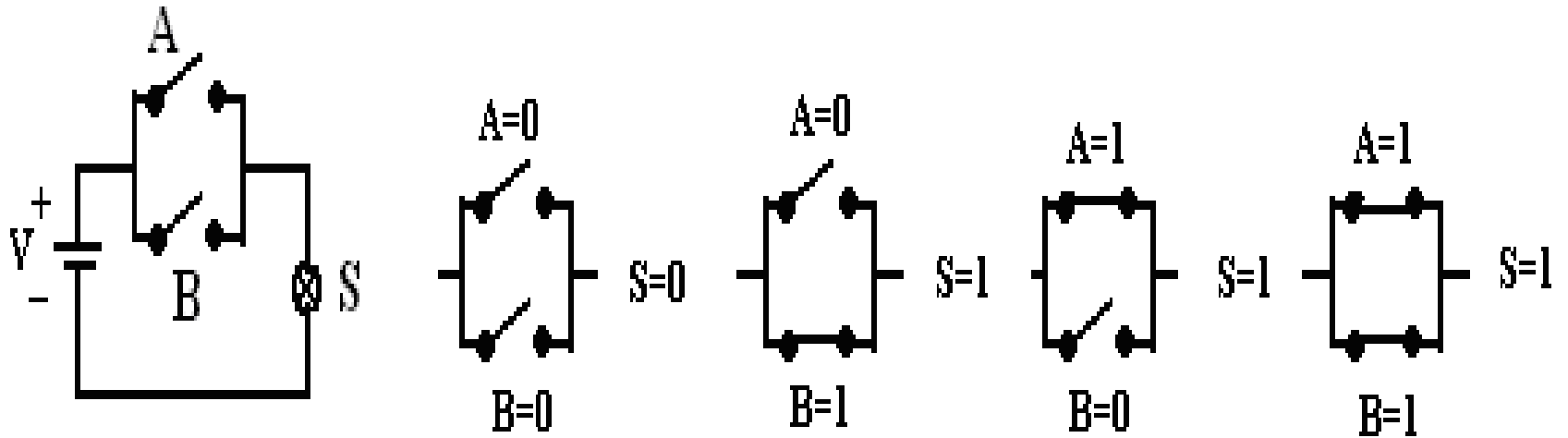
- Kombinasyonel lojik devre tasarlamak demek, problemin sözlü anlatımından hareket edilerek tanım tablosunu oluşturmak, tanım tablosundan fonksiyon denklemini (Lojik ifadeyi, deyimi) elde etmek, fonksiyon denklemini indirgeyerek en az elemanla devreyi oluşturacak hale getirmek, bu denklemi sağlayan lojik şemayı çizmek demektir.
- Lojik Şeması elde edilmiş devre artık, mantıksal bağlaçlarla gerçekleştirilebilir haldedir.
- Bu işlemlerin doğru ve etkili bir biçimde yapılması için mantık matematiğini, tanımlarını, teoremlerini, işlemlerini yerine getiren bağlaç yapılarını iyi anlamak gerekir.

# Mantıksal İşlemlerin Gerçekleştirilmesi

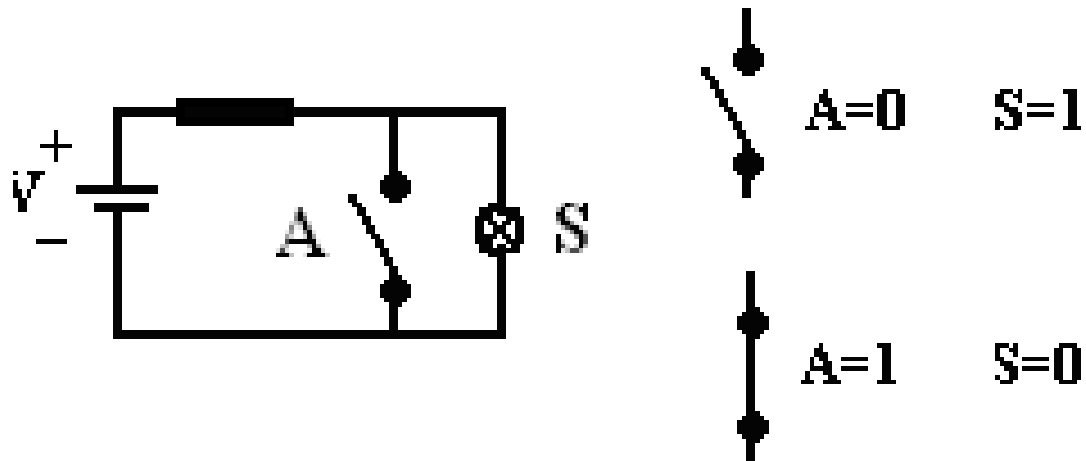
## AND bağlaçları



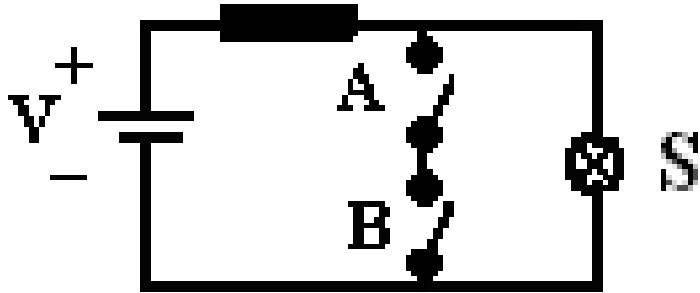
# OR Bağlacı



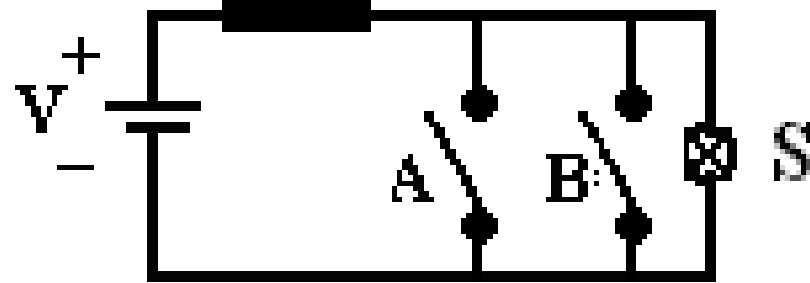
# DEĞİL Bağlacı



# NAND, NOR bağlacı



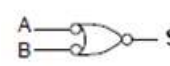


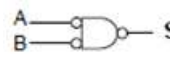
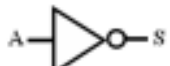



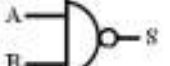


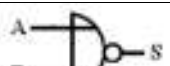





NAND



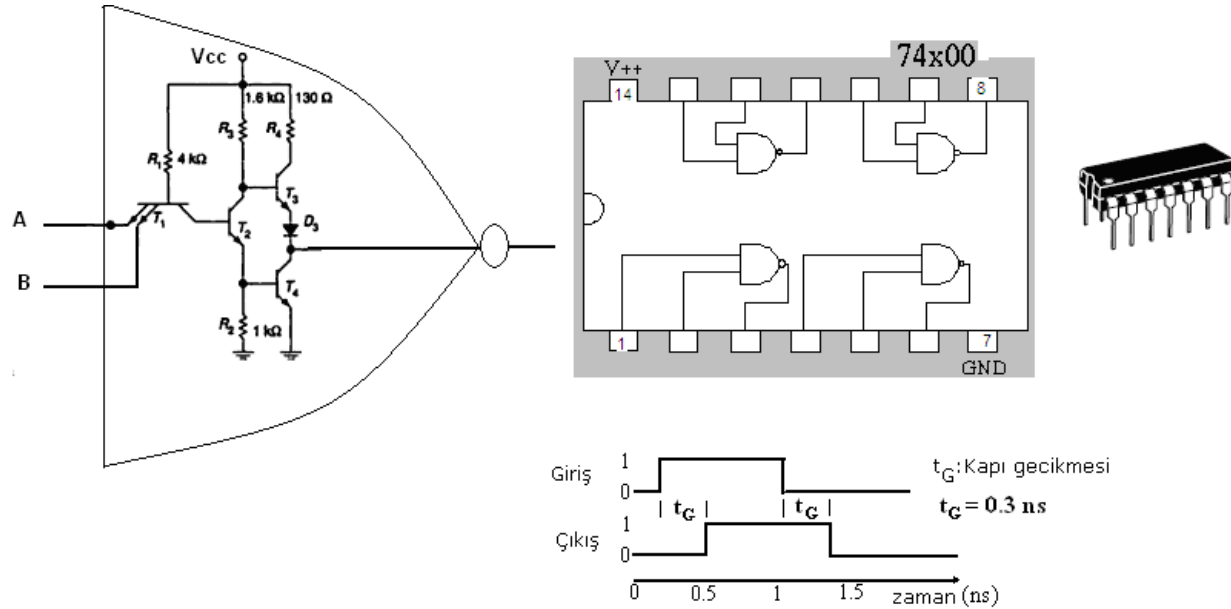
NOR

# Tanım tablosu, Fonksiyon denklemleri ve Şematik gösterim arasındaki ilişki

apı tipi	I.Form	II.Form	III.Form																
VE KAPISI (AND GATE)	 $S = A \cdot B$	 $S = A \cdot B$	 $S = A \cdot B = \overline{\overline{A} + \overline{B}}$	<table><tr><th>A</th><th>B</th><th>S</th></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	S	0	0	0	0	1	0	1	0	0	1	1	1
A	B	S																	
0	0	0																	
0	1	0																	
1	0	0																	
1	1	1																	
VEYA KAPISI (OR GATE)	 $S = A + B$	 $S = A + B$	 $S = A + B = \overline{\overline{A} \cdot \overline{B}}$	<table><tr><th>A</th><th>B</th><th>S</th></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	S	0	0	0	0	1	1	1	0	1	1	1	1
A	B	S																	
0	0	0																	
0	1	1																	
1	0	1																	
1	1	1																	
DEĞİL KAPISI (NOT GATE)	 $S = \overline{A}$	 $S = \overline{A}$	 $S = \overline{A}$	<table><tr><th>A</th><th>S</th></tr><tr><td>0</td><td>1</td></tr><tr><td>1</td><td>0</td></tr></table>	A	S	0	1	1	0									
A	S																		
0	1																		
1	0																		
VE-DEĞİL (NAND GATE)	 $S = \overline{A \cdot B}$	 $S = \overline{A \cdot B}$	 $S = \overline{A \cdot B} = \overline{A} + \overline{B}$	<table><tr><th>A</th><th>B</th><th>S</th></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	S	0	0	1	0	1	1	1	0	1	1	1	0
A	B	S																	
0	0	1																	
0	1	1																	
1	0	1																	
1	1	0																	
VEYA-DEĞİL KAPISI (NOR GATE)	 $S = \overline{A + B}$	 $S = \overline{A + B}$	 $S = \overline{A + B} = \overline{A} \cdot \overline{B}$	<table><tr><th>A</th><th>B</th><th>S</th></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	S	0	0	1	0	1	0	1	0	0	1	1	0
A	B	S																	
0	0	1																	
0	1	0																	
1	0	0																	
1	1	0																	
ÖZEL -OR (EX-OR)	 $S = A \oplus B$ $S = A \overline{B} + \overline{A} B$			<table><tr><th>A</th><th>B</th><th>S</th></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	S	0	0	0	0	1	1	1	0	1	1	1	0
A	B	S																	
0	0	0																	
0	1	1																	
1	0	1																	
1	1	0																	
ÖZEL-NOR (EX-NOR)	 $S = A \odot B$ $S = A B + \overline{A} \overline{B}$			<table><tr><th>A</th><th>B</th><th>S</th></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	S	0	0	1	0	1	0	1	0	0	1	1	1
A	B	S																	
0	0	1																	
0	1	0																	
1	0	0																	
1	1	1																	



- Sayısal sistemlerde mantıksal işlemlere karşılık gelen semboller değişik standartlarda tarif edilmişlerdir. Bundan sonra fonksiyon denklemlerinin şematik olarak gösterilmesi için bu semboller kullanılacaktır.
- Günümüzde sayısal elektronik disiplinde yukarıdaki sembollerle açıklanan mantıksal işlemleri gerçekleştiren birçok sayısal entegre devre ailesi mevcuttur. Küçük Çaplı Entegre devrelerde en çok kullanılan Transistör-Transistör Lojiği (TTL) ailesinden bir NAND bağlacının iç yapısı Şekilde verilmiştir.



- elimizde bir fonksiyon denklemi varsa bu denklemi şematik olarak göstermek için, ilgili işlem sembollerini denklemi sağlamak için uygun bir şekilde birbirine bağlarız.
- Burada, tanım tablosu ve fonksiyon denklemi verilen bir problemin lojik sembolleri kullanarak şematik gösterimini bir örnekle açıklayalım. Şekilde görüldüğü gibi tanım tablosunu sağlayan fonksiyon denkleminin kısaltılmış şekli  $F = X + Y'.Z$  dir.
- Bu mantıksal eşitlikte 1 adet *değil*, 1 adet 2 girişli VE, 1 adet 2 girişli VEYA bağlacına gerek vardır. Denklemi gerçekleştiren lojik şema bunların uygun şekilde biri birine bağlanmasıyla oluşturulmuştur.
- Aslında problemin tanımı için tanım tablosu yeterlidir. Fonksiyon denklemi ve lojik şema gerçekleştirme aşaması için önemlidir. Bunların hepsi birden bir kombinasyonel (Birleşimsel) lojik devrenin tasarım ve gerçekleştirme aşamasıdır.

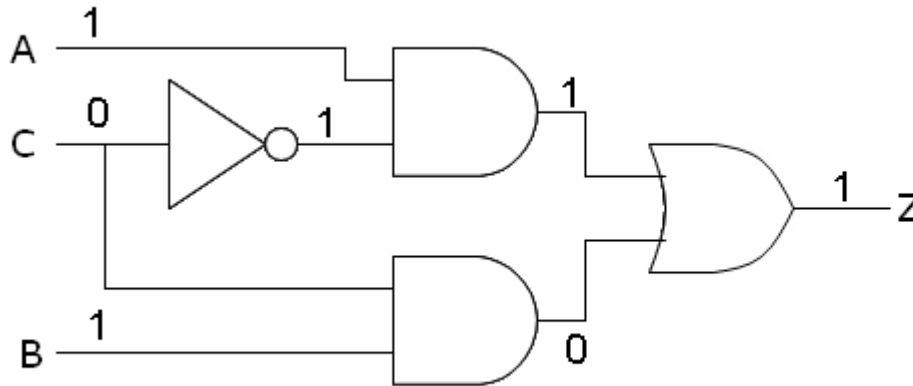
X Y Z	$F = X + Y'.Z$	<p>Lojik ( Mantık) Şema</p>
0 0 0	0	
0 0 1	1	
0 1 0	0	
0 1 1	0	
1 0 0	1	
1 0 1	1	
1 1 0	1	
1 1 1	1	

# Kombinasyonel Lojik devrelerin analizi

- Kombinasyonel lojik devrelerinin analizi, lojik şemadan yola çıkarak, tanım tablosunu ve fonksiyon denklemini elde etmektir. Kısaca Lojik şeması verilen devrenin ne işe yaradığının sistematik olarak açıklanması işlemidir.

## 1-Literal Analiz:

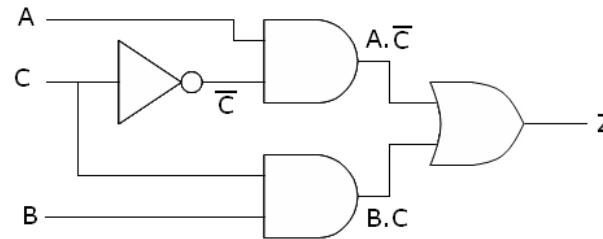
- Literal, devredeki değişkenlerin ve deęillerinin her birine denir. Bu tür analizde, giriş deęişkenlerinin oluşturduęu her bir farklı sözcük için şematik devrede sembollerin giriş ve çıkışları, devre girişinden çıkışına kadar deęerlendirilerek çıkış deęeri elde edilir. n tane giriş deęişkeni için olabilecek  $2^n$  tane giriş sözcüğü için bu izleme yapılır. Şematik devre yardımıyla sadece tanım tablosu elde edilebilir. Şekil4.8'de 3 girişli bir devre örneęi görölmektedir.



A	B	C	Z
0	0	0	0
0	0	1	X
0	1	0	X
0	1	1	X
1	0	0	X
1	0	1	X
1	1	0	1
1	1	1	1

# Sembolik Analiz

- Bu tür analizde, lojik şemada girişten itibaren her tabakadaki bağlaçların çıkış denklemleri yazılır. Girişten uygulanan her bir sözcük için bu denklemlerin ve sonuç için oluşturulan denklemin aldığı değerler hesaplanarak, hem fonksiyon denklemi, hem de tanım tablosu elde edilmiş olur. Şekil 4.9'da şematik devre bundan elde edilen tanım tablosu ve fonksiyon denklemi görülmektedir.



A	B	C	$\overline{C}$	$A.\overline{C}$	$B.C$	$Z = A.\overline{C} + B.C$
0	0	0	1	0	0	0
0	0	1	0	0	0	0
0	1	0	1	0	0	0
0	1	1	0	0	1	1
1	0	0	1	1	0	1
1	0	1	0	0	0	0
1	1	0	1	1	0	1
1	1	1	0	0	1	1

# Kombinasyonel Lojik devrelerin tasarımı

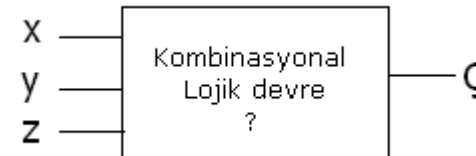
- **1-Problemin sözle anlatılması:** Burada hiçbir belirsiz durumun kalmamasına dikkat edilmelidir. Her bir durum için devrenin nasıl davranacağı net bir biçimde açıklanmalıdır.
- **2-Tanım tablosunun oluşturulması:** Sözle anlatımdan yola çıkarak giriş değişkenlerinin belirlenmesi ve her bir giriş kombinasyonu (sözcüğü) için tarif edilen çıkış veya çıkışlar değerlerinin sonuç sütununda belirlenmesi.
- **3- Fonksiyon denkleminin elde edilmesi ve indirgenmesi:** Tanım tablosunu kullanarak, fonksiyon denklemi tam açılım şeklinde kanonik formlardan birinde elde edilir. Belirlenmiş kriterlere göre indirgenir ve/veya kullanılacak bağlaçlara göre optimize edilir.
- **4- Lojik Şemanın Çizimi:** Elde edilmiş, kısaltılmış ve optimize edilmiş fonksiyon denklemi, lojik bağlaçlar kullanılarak ifade edilir. Bu aşamada tasarımın büyük bölümü tamamlanmıştır.
- **5-Lojik şemanın test edilmesi:** Optimize edilmiş denkleme göre çizilmiş lojik şema, analiz yöntemleriyle test edilerek tanım tablosunu doğruladığının sağlanması yapılır.
- **6- Teknolojik Gerçekleştirme:** Tasarlanan devre, seçilen lojik familyasından entegre devre bağlaçlar kullanılarak gerçekleştirilir. Bu aşamada, kullanılacak lojik familyanın seçim

- Yukarıdaki adımlara göre elde edilen kombinasyonel lojik devreler 3 tiptir.
- **1- Çok girişli, tek çıkışlı bütünüyle tanımlanmış devreler:** Giriş değişkenleri 1'den fazladır. Tek bir çıkışı vardır. Devrenin alabileceği durumlar için çıkış tanımlanmıştır.
- **2- Çok girişli, çok çıkışlı devreler bütünüyle tanımlanmış devreler:** Bu devrelerin giriş değişkenleri ve çıkışları 1'den fazla olup, devrenin alabileceği giriş durumları için çıkışlar tanımlanmıştır.
- **3-Bütünüyle tanımlanmamış lojik devreler :** Bu devreler tek çıkışlı veya çok çıkışlı olabilir. Devre girişindeki olabilecek kombinasyonlardan bazıları için çıkış tanımlanmamıştır. Giriş kombinasyonlarından hiçbir zaman oluşmayacaklar için veya devre çıkışının önemsiz olduğu giriş kombinasyonları için tanımlama yapılmayabilir. Bu şekildeki tanımlanmamış kombinasyonlar

# Örnek1

## Problemin sözle ifadesi:

Girişine uygulanan 3 bitlik sayıların 1,2,3,4 olduğu durumlarda çıkışı 1 olan çok girişli, tek çıkışlı kombinasyonel lojik devreyi tasarlayınız.



x	y	z	Ç	Minterim	Maksterim
0	0	0	0	$x'y'z'$	$x+y+z$
0	0	1	1	$x'y'z$	$x+y+z'$
0	1	0	1	$x'y z'$	$x+y'+z$
0	1	1	1	$x'y z$	$x+y'+z'$
1	0	0	1	$x y'z'$	$x'+y+z$
1	0	1	0	$x y'z$	$x'+y+z'$
1	1	0	0	$x y z'$	$x'+y'+z$
1	1	1	0	$x y z$	$x'+y'+z'$

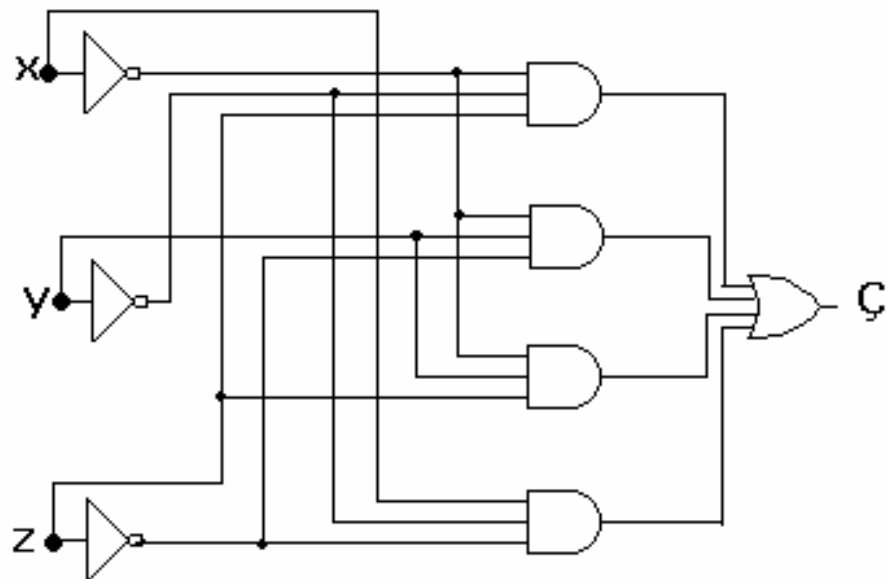
## **Fonksiyon denkleminin elde edilmesi ve indirgenmesi:**

Fonksiyon denklemini 2 farklı standart form'da elde edilebilir.  
Denklemini 1'li açılıma göre yazmak için, sonucu 1 olan minterim'ler alınıp kendi aralarında VEYA bağlacı ile bağlanır.

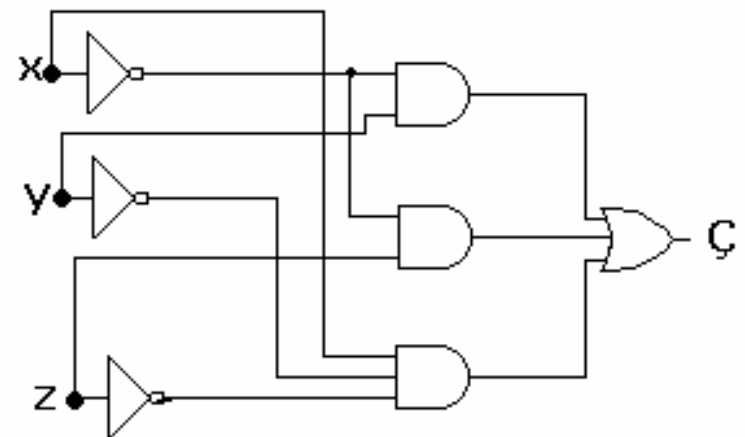
- **$\Sigma_1(x,y,z) = \Sigma_{m\ 1,2,3,4} = x'y'z + x'y z' + x'y z + x y'z'$**
- Oluşturulan bu denklemde her terimde bütün değişkenlerin kendileri veya değilleri mevcuttur. Buna tam açılım şekli denir. Denklem bu durumda indirgenmemiş (Kısaltılmamış) haldedir.
- Yukarıdaki denklem, tanım tablosunun en açık ifadesi olmasına karşılık gerçekleştirilme açısından en pahalı olanıdır. Çünkü Literal sayısı, bağlaç ve bağlaç giriş sayısı bakımından en fazla olan denklem kısaltılmamış denklemdir.
- Fonksiyon denklemlerinin optimizasyonu bir sonraki konuda açıklanacaktır. Burada denklemin kısaltılması teoremler yardımıyla yapılacaktır.



$$\begin{aligned}
C_1(x,y,z) &= x'y'z + x'y z' + x'y z + x y'z' \\
&= x'y'z + x'y (z' + z) + x y'z' \\
&= x'y'z + x'y + x y'z' \\
&= x'(y+y'z) + x y'z' \\
&= x'(y+z) + x y'z' \\
&= x'y + x'z + x y'z'
\end{aligned}$$



$$C_1(x,y,z) = x'y'z + x'y z' + x'y z + x y'z'$$



$$C_1(x,y,z) = x'y + x'z + x y'z'$$

- Aynı problem için 0'lı açılıma göre denklemini oluşturmak için;
- Tanım tablosunda sonucu 0 olan sözcüklerden Maksterimler oluşturularak kendi aralarında VE bağlacı ile bağlanmalıdırlar.

$$\begin{aligned}\zeta 0(x,y,z) &= \prod_M 0,5,6,7 \\ &= (x+y+z).(x'+y+z').(x'+y'+z).(x'+y'+z')\end{aligned}$$

- Denklemin bu hali tam açılım şeklidir. Bu denklem teoremler kullanılarak kısaltılırsa;

$$C_0(x,y,z)=(x+y+z).(x'+y+z').(x'+y'+z).(x'+y'+z')$$

$$C_0(x,y,z)=(x.x'+x.y+x.z'+y.x'+y.y+y.z'+z.x'+z.y+z.z')$$

$$\cdot (x'.x'+x'.y'+x'.z'+y'.x'+y'.y'+y'.z'+z.x'+z.y'+z.z')$$

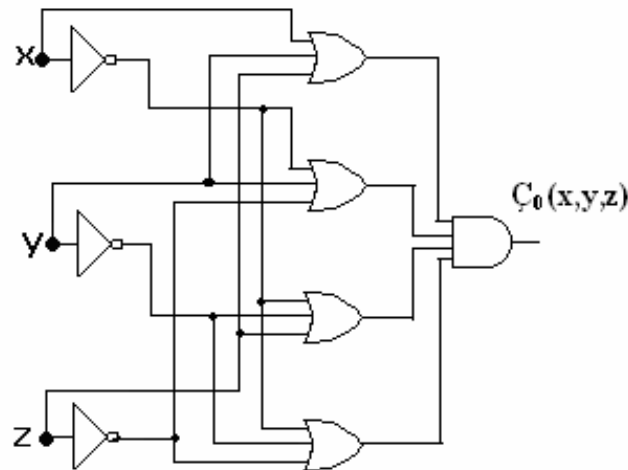
$$= (0+y(x+x'+1+z'+z)+xz'+z.x').(x'(1+y'+z'+y'+z)+y'(1+z'+z))$$

$$= (y+x.z'+z.x').(x'+y') = y.x'+y.y'+x.z'.x'+x.z'.y'+z.x'.x'+z.x'.y'$$

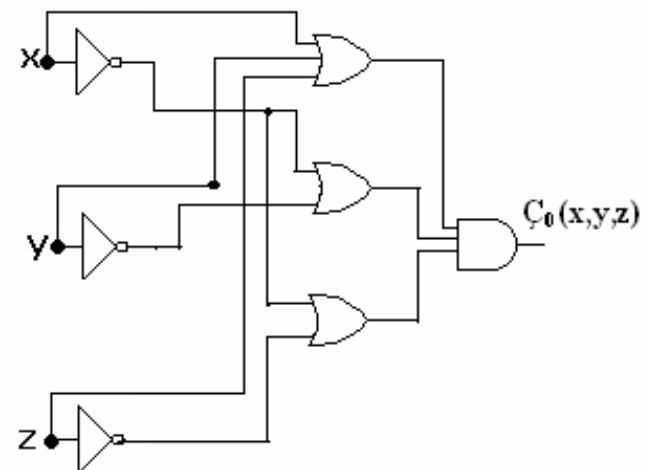
$$= x'.y + 0 + 0 + x.z'.y' + z.x' + z.x'.y' = x'.y + x'.z.(1+y) + x.z'.y'$$

$$= x'.y+x'.z+x.y'.z'=x'.(y+z)+x.(y'.z')=(x+(y+z)).(x'+y'.z') \text{ (T.11)}$$

$$= (x + y + z) . (x' + y').(x' + z') \quad \text{(A3.b)}$$



$$C_0 = (x+y+z).(x'+y+z').(x'+y'+z).(x'+y'+z')$$



$$C_0 = (x + y + z) . (x' + y').(x' + z')$$

## Örnek 2.

- **Problemin sözle anlatımı:** Birer bitlik iki binary sayıyı alt basamaktan gelebilecek elde bit'inide göz önüne alarak toplayıp sonucu ve oluşabilecek elde biti çıkışı veren tam toplayıcı devreyi tasarlayınız.
- Bu örnek, çok girişli çok çıkışlı bir devre problemidir.

$a_n$  : A sayısının n.biti

$b_n$  : B sayısının n.biti

$e_{n-1}$ : n-1. basamaktan gelen elde

$e_{n+1}$ : n.basamaktaki toplamadan üst basamağı  
gönderilecek elde biti.

$S_n$  : n. bitlerin toplanmasından oluşan sonuç.



## Tanım tablosunun oluşturulması ve çıkış denklemleri.:

- Problem 3 girişli 2 çıkışlı bir devreyle çözülür. Binary sayılardaki toplama kuralı göz önünde tutularak, tanım tablosunda iki çıkış sütunundan  $s_n$ , n.basamaktaki toplama sonucunu,  $e_{n+1}$  ise üst basamağa gönderilecek elde bitini oluşturmak içindir.

$a_n$	$b_n$	$e_{n-1}$	$s_n$	$e_{n+1}$
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

*Her bir çıkış denklemini bulmak için, giriş kombinasyonaları ile, ilgili çıkış sütunu göz önüne alınır.  $S_n$  çıkışı, 1'li açılıma göre;*

$$S_n = a_n' \cdot b_n' \cdot e_{n-1} + a_n' \cdot b_n \cdot e_{n-1}' + a_n \cdot b_n' \cdot e_{n-1}' + a_n \cdot b_n \cdot e_{n-1}$$

*$e_{n+1}$  çıkışı ise;*

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

$$e_{n+1} = a_n' \cdot b_n \cdot e_{n-1} + a_n \cdot b_n' \cdot e_{n-1} + a_n \cdot b_n \cdot (e_{n-1}' + e_{n-1})$$

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

$$e_{n+1} = a_n' \cdot b_n \cdot e_{n-1} + a_n \cdot (b_n' \cdot e_{n-1} + b_n)$$

$$e_{n+1} = a_n' \cdot b_n \cdot e_{n-1} + a_n \cdot (e_{n-1}' + b_n)$$

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

$$e_{n+1} = a_n \cdot e_{n-1}' + b_n \cdot (a_n' + a_n' \cdot e_{n-1})$$

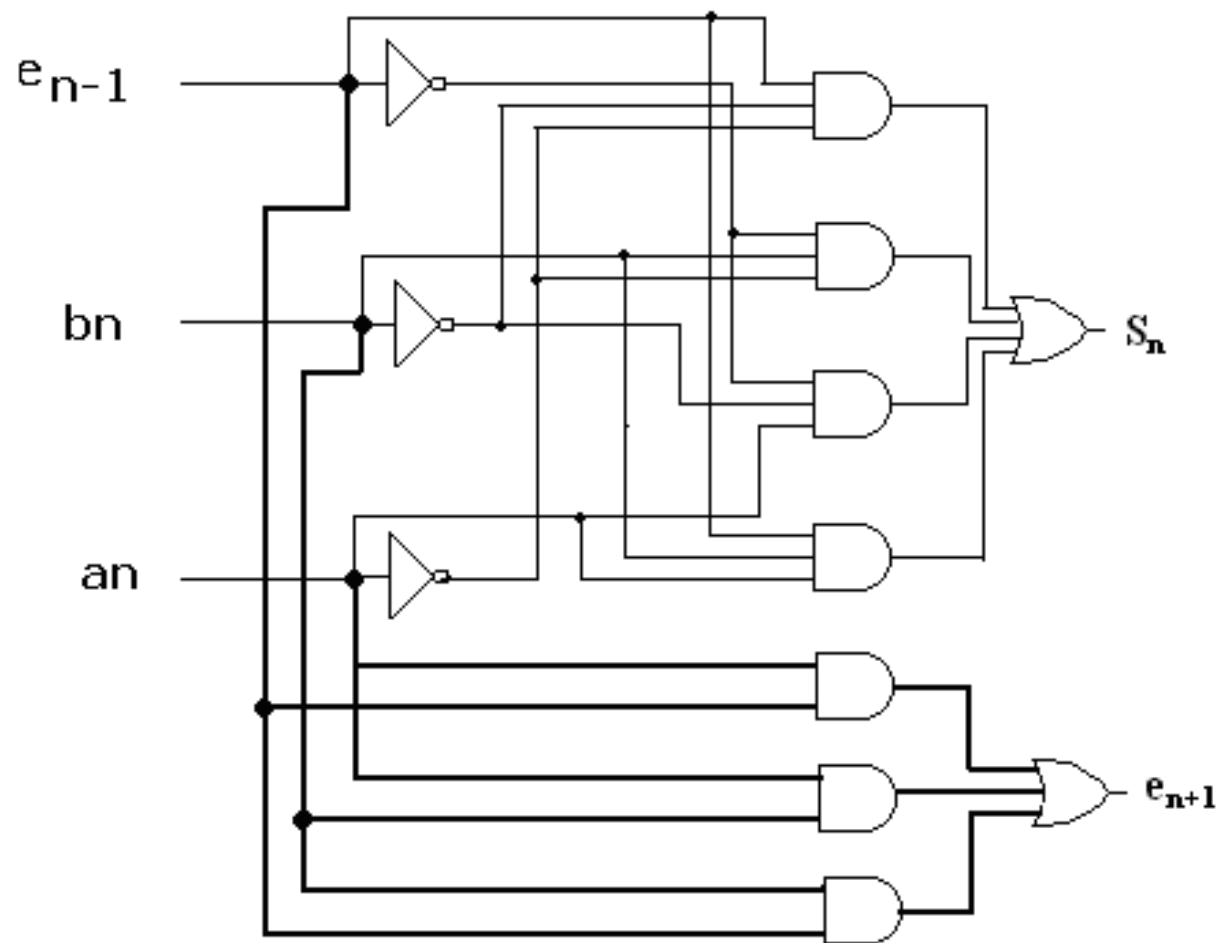
$$e_{n+1} = a_n \cdot e_{n-1}' + b_n \cdot (a_n' + e_{n-1})$$

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

# Problemin Lojik Şeması

$$S_n = a_n' \cdot b_n' \cdot e_{n-1} + a_n' \cdot b_n \cdot e_{n-1}' + a_n \cdot b_n' \cdot e_{n-1}' + a_n \cdot b_n \cdot e_{n-1}$$

$$e_{n+1} = a_n \cdot e_{n-1} + a_n \cdot b_n + b_n \cdot e_{n-1}$$



## **Örnek 3.**

- **Problemin sözle anlatımı:** Girişine uygulanan BCD kodlanmış 4 bitlik sözcüklerden, içerisinde lojik 1 sayısı çift olan kombinasyonlarda çıkışı 1 olan kombinasyonel devreyi tasarlayınız.

## Tanım tablosunun oluşturulması ve çıkış denklemleri

- Girişler, BCD kodlanmış rakam sözcükleri olduğundan, devrenin girişine 10 farklı 4 bitlik sözcük uygulanabilir.
- Girişte diğer 6 farklı durum oluşmayacağından bütünüyle tanımlanmamış fonksiyon şeklinde düşünülecektir.
- Girişte oluşmayacak kombinasyonlar için tanım tablosunun çıkışında Keyfi (K– 0 veya 1 olabilir) değerler alınabilir.

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



- 1'li açılıma göre denklemi yazmak istersek, minterimlerin sayısının az olması için Keyfi (K) değerleri 0 alınır. Buna göre denklem;
- $S = \sum_m 0,3,5,6,9$   
 $= A'.B'.C'.D' + A'.B'.C.D + A'.B.C'.D + A'.B.C.D' + A.B'.C'.D$
- Bu denklem kısaltılamaz. Bunu sağlayan lojik şema aşağıdadır.

