

Sayısal Devreler (Lojik Devreleri)

Ders Notlarının Creative Commons lisansı Feza BUZLUCA'ya aittir

Lisans: <http://creativecommons.org/licenses/by-nc-nd/3.0/>

## Lojik Bağlaçlar (Logic Gates)

ANSI/IEEE-1973

ANSI/IEEE-1984

SÜRÜCÜ (BUFFER)  $Y=X$

| X | Y |
|---|---|
| 0 | 0 |
| 1 | 1 |

TÜMLEME (NOT)  $X'$

| X | Y |
|---|---|
| 0 | 1 |
| 1 | 0 |

VE (AND)

$X \cdot Y$

| X | Y | Z |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |

VEYA (OR)

$X + Y$

| X | Y | Z |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 1 |

<http://www.akademi.itu.edu.tr/buzluca>  
<http://www.buzluca.info>

©2000-2015 Feza BUZLUCA

3.1

Sayısal Devreler (Lojik Devreleri)

TVE (NAND)  
(Tümlenmiş VE)

$(xy)'$

$X \rightarrow \text{AND} \rightarrow \neg Z$

$X \rightarrow \boxed{\&} \rightarrow \neg Z$

| X | Y | Z |
|---|---|---|
| 0 | 0 | 1 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |

TVEYA (NOR)  
(Tümlenmiş VEYA)

$(x+y)'$

$X \rightarrow \text{OR} \rightarrow \neg Z$

$X \rightarrow \boxed{\geq 1} \rightarrow \neg Z$

| X | Y | Z |
|---|---|---|
| 0 | 0 | 1 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 0 |

YA DA (XOR)  
 $xy' + x'y$

$X \oplus Y$

$X \rightarrow \text{XOR} \rightarrow Z$

$X \rightarrow \boxed{=1} \rightarrow Z$

| X | Y | Z |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |

EŞDEĞER (XNOR)  
 $xy + x'y'$

$X \odot Y$

$X \rightarrow \text{XOR} \rightarrow \neg Z$

$X \rightarrow \boxed{=1} \rightarrow \neg Z$

| X | Y | Z |
|---|---|---|
| 0 | 0 | 1 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |

<http://www.akademi.itu.edu.tr/buzluca>  
<http://www.buzluca.info>

©2000-2015 Feza BUZLUCA

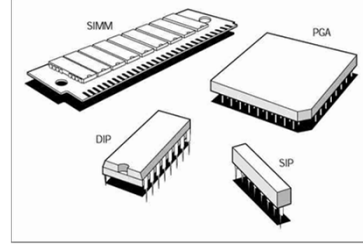
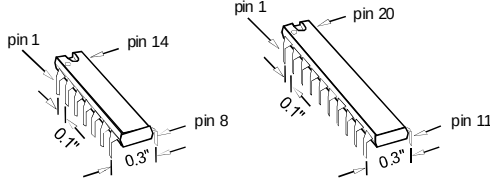
3.2

### Tümdevreler ( *Integrated Circuits - IC* )

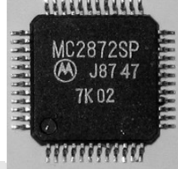
Lojik bağlaçlar, tümdevrelerin içinde yer alacak şekilde üretilir ve pazarlanırlar. Bir tümdevrede, büyüklüğüne ve bağlacın giriş sayısına bağlı olarak birden fazla lojik bağlaç yer alır.

Tümdevreler, farklı şekillerde üretilirler.

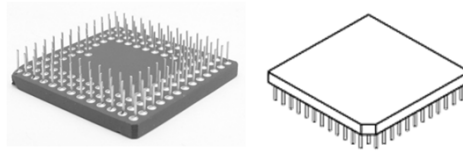
#### Dual in-line Package (DIP) ICs



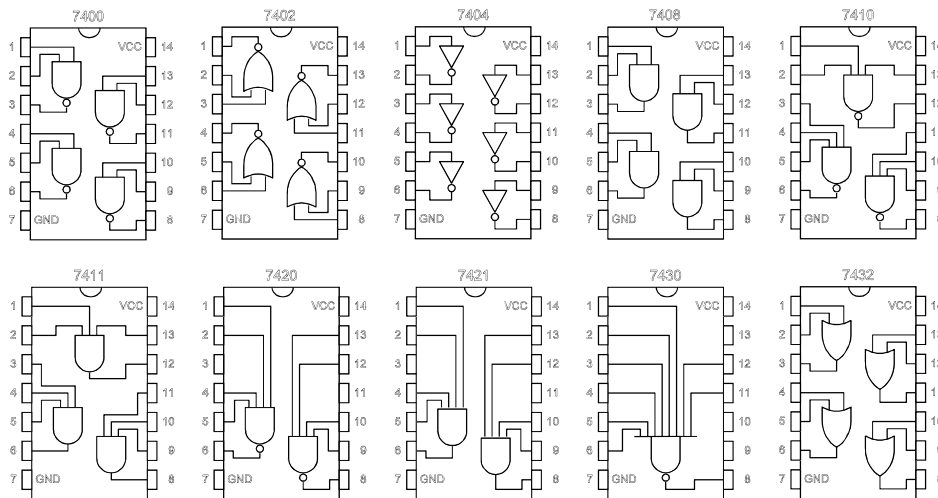
#### Quad Flat Package (QFP)



#### Pin Grid Array (PGA)



### 74xx Serisi Tümdevrelere Örnekler



Tümdevreler ile ilgili bilgiler tümdevre kataloglarında yer alırlar.

**Pozitif ve Negatif Lojik**

Sıfır ve 1 değerini alan girişler ve çıkışlar, genel olarak, fiziksel bir büyüklüğün 2 farklı seviyesine karşı düşer: Gerilim, akım, basınç v.b.

Yüksek seviyeye 1, alçak seviyeye 0 karşı düşürülüyorsa buna pozitif lojik, aksi halde negatif lojik denir.

L (Low) düşük seviye, H (High) yüksek seviye olmak üzere, 2 girişli 1 çıkışlı bir kapının giriş-çıkış ilişkisi aşağıda gösterilmiştir.

Pozitif lojik kullanıldığı takdirde fiziksel devre bir VE kapısı, negatif lojik kullanıldığı takdirde de bir VEYA kapısı gerçeklemektedir.

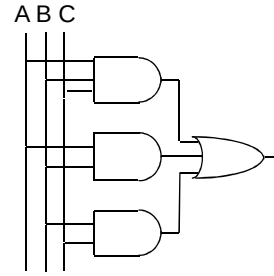
Bir lojik devrenin tümünde ya pozitif ya da negatif lojik kullanılır.

| Fiziksel Devre |    |        | Pozitif Lojik |    |        | Negatif Lojik |    |        |
|----------------|----|--------|---------------|----|--------|---------------|----|--------|
| Girişler:      |    | Çıkış: | Girişler:     |    | Çıkış: | Girişler:     |    | Çıkış: |
| x1             | x2 | z      | x1            | x2 | z      | x1            | x2 | z      |
| L              | L  | L      | 0             | 0  | 0      | 1             | 1  | 1      |
| L              | H  | L      | 0             | 1  | 0      | 1             | 0  | 1      |
| H              | L  | L      | 1             | 0  | 0      | 0             | 1  | 1      |
| H              | H  | H      | 1             | 1  | 1      | 0             | 0  | 0      |

**Lojik Fonksiyonların Bağlaçlar İle Gerçeklenmesi**

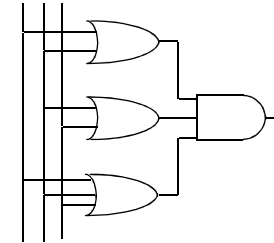
- Çarpımların Toplamı

- VE (AND) kapıları çarpımları gerçekler
- VEYA (OR) kapısı toplamayı gerçekleştirir

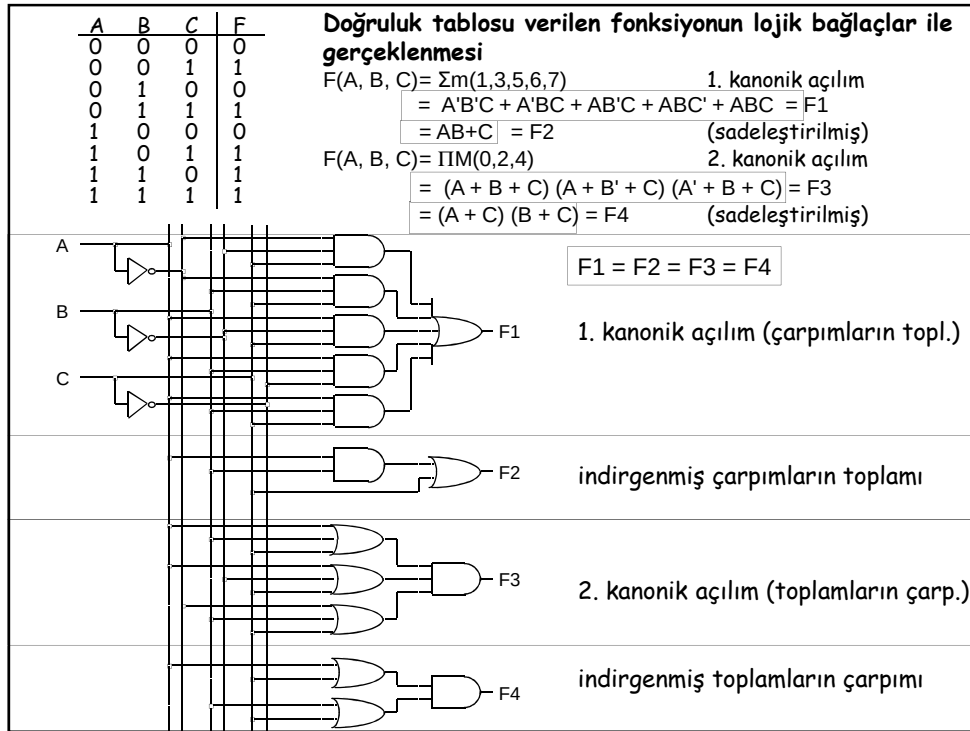


- Toplamların Çarpımı

- VEYA (OR) kapıları toplamaları gerçekler
- VE (AND) kapısı çarpımı gerçekleştirir



Gerekli olan yerlerde tümlleme kapıları da kullanılır.



Sayısal Devreler (Lojik Devreleri) Lisans: <http://creativecommons.org/licenses/by-nc-nd/3.0/>

Bir lojik ifade farklı şekillerde lojik bağlaçlar kullanılarak gerçekleştirilebilir.

**Örnek:**  $Z = A' \cdot B' \cdot (C + D)$   $= (A' \cdot (B' \cdot (C + D)))$

3 girişli kapı

Sadece 2 girişli kapılar

Elinizde var olan fiziksel kapılara göre lojik ifadeyi düzenlemek gerekir.

<http://www.akademi.itu.edu.tr/buzluca>  
<http://www.buzluca.info>

CC BY NC ND

©2000-2015 Feza BUZLUCA

3.8

### Yetkin İşlemler Kümesi (Functionally complete set )

Boole cebrinin tanımına göre VE, VEYA, TÜMLEME işlemleri ile tüm olası lojik fonksiyonları gerçeklemek mümkündür.

Bu nedenle bu işlemler **yetkin bir işlem kümesi** (Functionally complete set ) oluştururlar.

- Bu işlemlerin dışında TVE (NAND) işlemi de tek başına yetkin bir işlemdir.
- Sadece TVE (NAND) bağlacı kullanarak tüm lojik fonksiyonları gerçeklemek mümkündür.
- Benzer şekilde TVEYA (NOR) da yetkin bir işlemdir.
- Sadece TVEYA (NOR) bağlacı kullanarak tüm lojik fonksiyonları gerçeklemek mümkündür.
- Bu özelliklerinden dolayı TVE ve TVEYA bağlaçlarına **evrensel bağlaçlar** (universal gate) adı verilir.

### Yetkin İşlem Özelliğinin Kanıtlanması

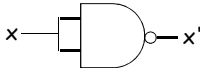
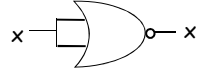
- TVE'nin tek başına bir yetkin işlem olduğunu kanıtlamak için VE, VEYA, TÜMLEME işlemlerinin her birinin sadece TVE bağlaçları (işlemleri) kullanılarak gerçekleştirilebileceğini göstermek gerekir.

- Aynı durum TVEYA bağlaçları (işlemleri) için de geçerlidir.

Kanıtlama işlemleri aşağıda gösterilmiştir.

| simgesi TVE işlemini,

↓ simgesi ise TVEYA'yı göstermek için kullanılmıştır.

| TVE                                                                                                                                                 | TVEYA                                                                                                                         |
|-----------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------|
| <p>• TÜMLEME:</p> $x' = x \downarrow x$ $= (x \cdot x)'$ $= x'$  | <p><math>x' = x \downarrow x</math></p>  |
| <p>• VE:</p> $x \cdot y = (x \downarrow y)'$                                                                                                        | <p><math>x \cdot y = (x' \downarrow y)'</math> de Morgan</p>                                                                  |
| <p>• VEYA:</p> $x + y = (x' \downarrow y') \text{ de Morgan}$                                                                                       | <p><math>x + y = (x \downarrow y)'</math></p>                                                                                 |

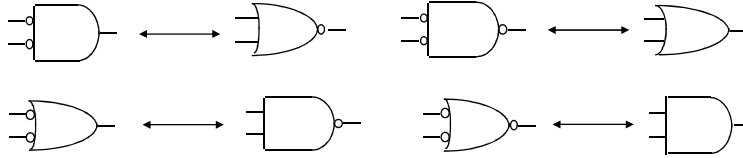
## TVE - TVEYA Arasındaki İlişki

## TVE - TVEYA Dönüşümleri

- de Morgan:  $(A + B)' = A' \cdot B'$   
 $(A \cdot B)' = A' + B'$
- diğer bir yazım şekli:  $(A' \cdot B')' = A + B$   
 $(A' + B')' = (A \cdot B)$

## Buna göre:

- Girişleri tümlenmiş VE kapısı, TVEYA kapısının eşdeğeridir.
- Girişleri tümlenmiş VEYA kapısı, TVE kapısının eşdeğeridir.
- Girişleri tümlenmiş TVE kapısı, VEYA kapısının eşdeğeridir.
- Girişleri tümlenmiş TVEYA kapısı, VE kapısının eşdeğeridir.



## Lojik fonksiyonların sadece TVE (veya TVEYA) bağlaçları ile gerçekleştirilmesi

TVE yetkin bir işlem olduğundan tüm lojik fonksiyonlar sadece TVE bağlaçları kullanılarak gerçekleştirilebilir. Aynı durum TVEYA bağlaçları için de geçerlidir.

## Çarpımların toplamı (VElerin VEYAsı) şeklindeki fonksiyonların

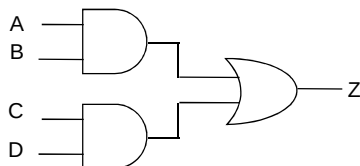
## TVE (NAND) ile gerçekleştirilmesi:

Bu tür devrelerde tüm VE kapıları ve VEYA kapılarının yerine TVE kapıları yerleştirilebilir. Bu değişiklik devrenin çıkış fonksiyonunu etkilemez.

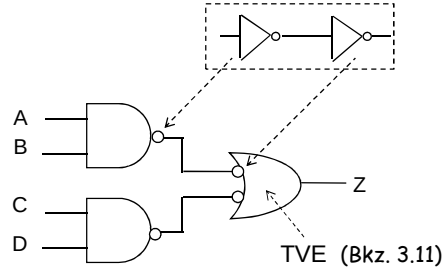
Aşağıda gösterildiği gibi VE kapılarının çıkışlarına, VEYA kapılarının da girişlerine tümleme elemanı yerleştirilirse TVE kapıları elde edilir.

Bir hatta peş peşe iki tümleme elemanı yerleştirilmesi herhangi bir değişikliğe neden olmaz.  $(a')' = a$  (Involution)

$$Z = (A \cdot B) + (C \cdot D)$$



≡

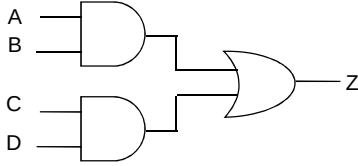


TVE (Bkz. 3.11)

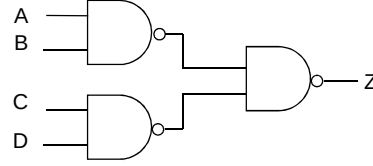
**Cebirsel Dönüşüm:**

İfadenin iki kere tümleyeni alınır.

$$\begin{aligned}
 Z &= (A \cdot B) + (C \cdot D) \\
 &= [(A \cdot B) + (C \cdot D)]' ]' \\
 &= [(A \cdot B)' \cdot (C \cdot D)']' \quad (\text{De Morgan}) \\
 &= (A \mid B) \mid (C \mid D)
 \end{aligned}$$

**Cebirsel olarak sına:**

?



$$\begin{aligned}
 Z &= [(A \cdot B)' \cdot (C \cdot D)']' \quad \text{Sağdaki devrenin ifadesi} \\
 &= [(A' + B') \cdot (C' + D')] \\
 &= [(A' + B') + (C' + D')] \\
 &= (A \cdot B) + (C \cdot D) \quad \checkmark \quad \text{Soldaki devrenin ifadesi}
 \end{aligned}$$

**Sınırlı sayıda girişe sahip kapılar ile tasarım**

Çok sayıda değişkene sahip olan çarpımları (toplamları) bazen az sayıda (örneğin 2) girişe sahip kapılar kullanarak gerçeklemek gerekir.

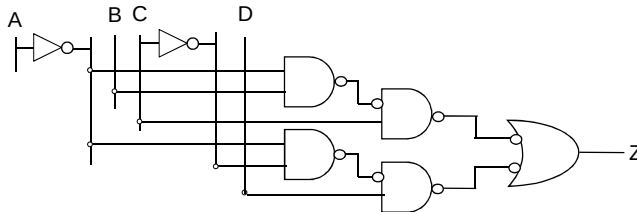
**Örnek:**

$$Z = \overline{A}BC + A\overline{C}D$$

Bu ifadeyi **sadece 2 girişli** TVE (NAND) kapıları kullanarak gerçekleyiniz.

**Çözüm 1:**

1. Bool cebirinin klasik kapıları ile gerçekleştirme

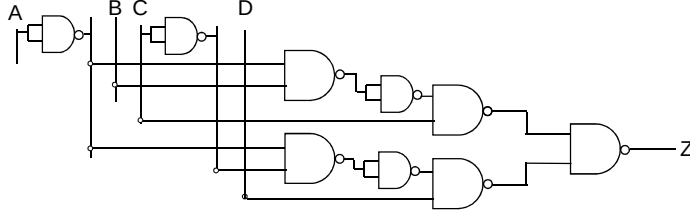


2. TVE kapılarına dönüşümü sağlamak için gerekli yerlere tümleme (NOT) kapılarının yerleştirilmesi

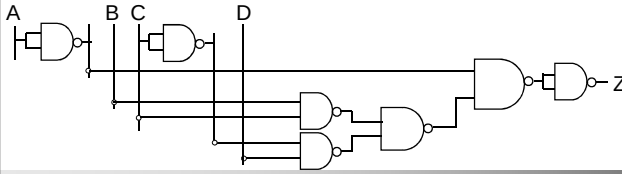
Örnek (devamı):

**Çözüm 1:**

3. Sadece 2 girişli TVE (NAND) kapıları ile gerçekleştirme

**Çözüm 2:**Daha basit (ucuz) bir devre elde etmek için orijinal ifadenin dönüştürülmesi

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

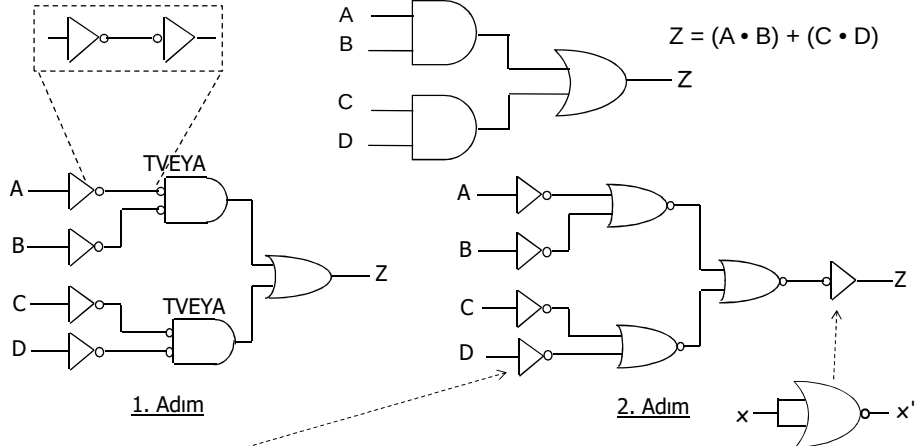


Çözüm 2'deki devre çözüm 1'dekine göre daha ucuzdur. Bu nedenle çözüm 2 tercih edilir.

**Çarpımların toplamı şeklindeki fonksiyonların TVEYA (NOR) ile gerçekleştirilmesi:**

VE lerin VEYA'lanması şeklinde devreler sadece TVEYA kullanılarak da gerçekleştirilebilir.

Bu durumda girişlere ve çıkışa tümleme elemanları yerleştirmek gerekir.



Hatırlatma: Tümleme bağlaçları TVEYA (NOR) bağlaçları ile gerçekleştirilebilir.

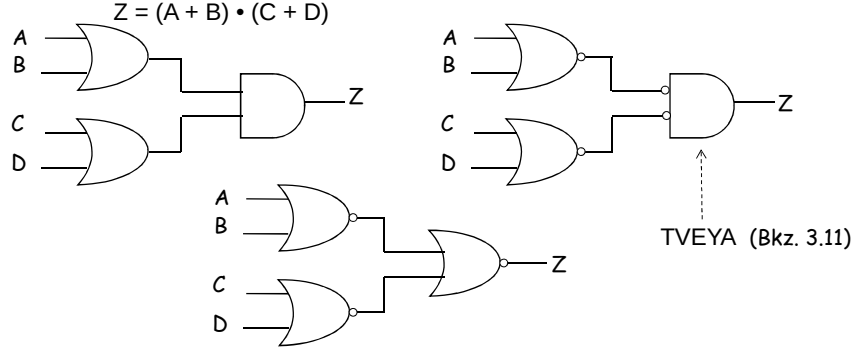


### Toplamların çarpımı (VEYA'ların VE'si) şeklindeki fonksiyonların TVEYA (NOR) ile gerçekleştirilmesi:

Bu tür devrelerde tüm VEYA kapıları ve VE kapılarının yerine TVEYA kapıları yerleştirilebilir. Bu değişiklik devrenin çıkış fonksiyonunu etkilemez.

Aşağıda gösterildiği gibi VEYA kapılarının çıkışlarına, VE kapılarının da girişlerine tümlene elemanı yerleştirilirse TVEYA kapıları elde edilir.

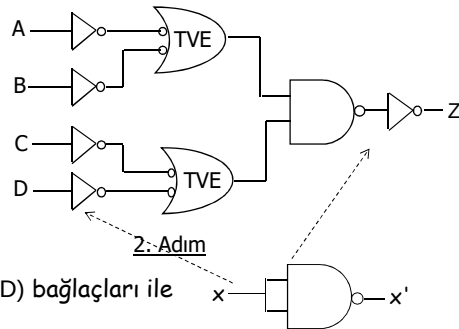
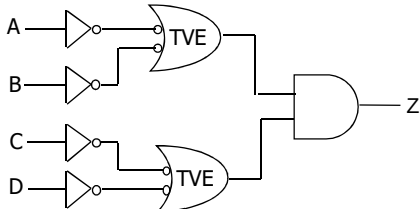
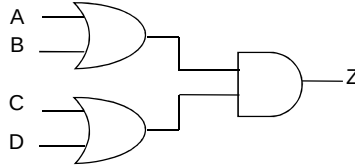
Bir hatta peş peşe iki tümlene elemanı yerleştirilmesi herhangi bir değişikliğe neden olmaz.



### Toplamların çarpımı (VEYA'ların VE'si) şeklindeki fonksiyonların TVE (NAND) ile gerçekleştirilmesi:

Bu durumda girişlere ve çıkışa tümlene elemanları yerleştirmek gerekir.

**Örnek:**  $Z = (A + B) \cdot (C + D)$

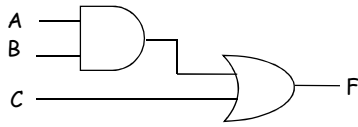


Hatırlatma: Tümlene bağlaçları TVE (NAND) bağlaçları ile gerçekleştirilebilir.

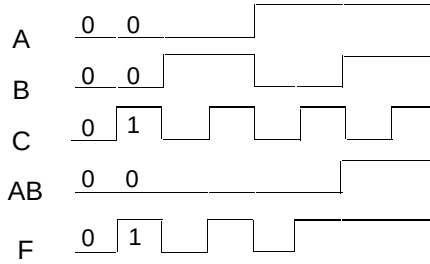
### Zaman Diyagramları (Timing Diagrams)

- Sayısal devrelerin zaman içindeki davranışlarını (giriş/çıkış ilişkisini) gösteren diyagramlardır.
- x ekseninde zaman, y ekseninde ise girişlerin ve çıkışların lojik değerleri (0/1 veya L/H) yer alır.
- Daha ayrıntılı zaman diyagramlarında y ekseninde elektriksel büyüklükler de (gerilim veya akım) yazılır.
- Fiziksel elemanların elektriksel özelliklerinden dolayı ortaya çıkan bazı durumların doğruluk tablosu ile gösterilmesi mümkün değildir. Böyle durumlarda devrelerin zaman diyagramlarını incelemek gerekir.

Örnek:



Yandaki diyagramda devrenin sadece lojik davranışı gösterilmiş, daha sonra anlatılacak olan gecikmeler dikkate alınmamıştır.

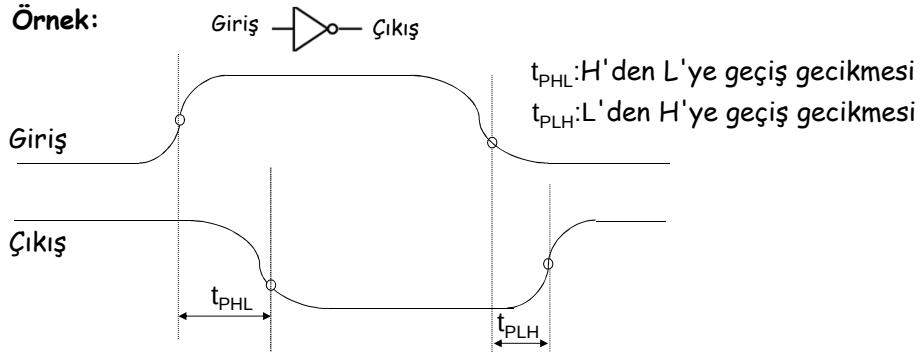


### Propagasyon (Yayılma) Gecikmesi (Propagation Delay)

Sayısal elemanları oluşturan elektronik devrelerin fiziksel yapılarından dolayı bir sayısal elemanın (örneğin bir lojik bağlacın) girişine uygulanan işaret (lojik değer) ancak belli bir süre geçtikten sonra o elemanın çıkışında etkili olur.

Giriş işaretinin elemanın içinde yol alarak çıkışı etkilemesi için geçen zaman o elemanın **propagasyon (yayılma) gecikmesini** belirler.

Örnek:



Propagasyon gecikmesi lojik devrenin çalışma hızını belirler.

Devrenin girişlerindeki değerler, devre bir önceki işini bitirinceye kadar sabit tutulmalıdır.

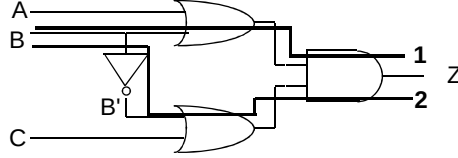
Devrenin girişine yeni değerler ancak eski girişler işlendikten sonra verilebilir.

**Gecikmeler nedeniyle oluşan problemler: Kaza (Hazard)**

Bir giriş değerinin, farklı bir kaç yoldan çıkışı etkilemesi nedeniyle çıkışta beklenmedik değer değişiklikleri (kazalar) oluşur.

Örneğin aşağıdaki devrede B girişinin değeri Z çıkışına iki farklı yoldan etki eder.

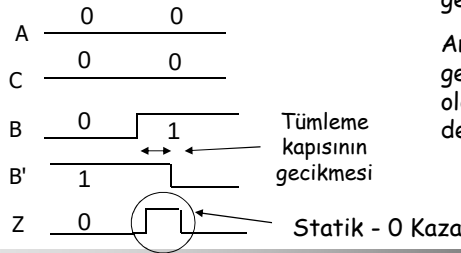
$$Z = (A+B) \cdot (B'+C)$$



Bu devrenin doğruluk tablosu incelendiğinde  $A=0, B=0, C=0$  girişi için  $Z=0$  olduğu görülür.

Bu durumdayken  $B=1$  olursa lojik ifadeye göre devrenin çıkışının  $Z=0$  olarak kalması gerekir.

Ancak 1. yol, 2. yola göre gecikmeler açısından daha "kısa" olduğundan Z çıkışında anlık bir değişim (kaza) oluşur.



Üç tür kaza (hazard) vardır:

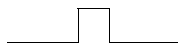
a) Statik 0: Çıkış lojik 0'da kalması gerekirken kısa bir süre "1" olup tekrar 0'a iner.

Statik 0 kaza, toplamların çarpımı şeklinde gerçekleştirilen devrelerde oluşur.

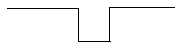
b) Statik 1: Çıkış lojik 1'de kalması gerekirken kısa bir süre "0" olup tekrar 1'e çıkar.

Statik 1 kaza, çarpımların toplamı şeklinde gerçekleştirilen devrelerde oluşur.

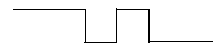
b) Dinamik: Çıkış bir kez değer değiştirmesi gerekirken bir kaç defa değer değiştirir.



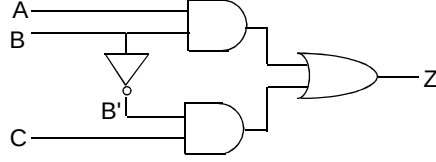
Statik 0



Statik 1



Dinamik

**Kazaların önlenmesi:**

Çarpımların toplamı şeklinde gerçekleştirilen yandaki devrenin doğruluk tablosu incelendiğinde  $A=1, B=1, C=1$  girişi için  $Z=1$  olduğu görülür. Bu durumdayken  $B=1$ 'den  $0$ 'a inerse devrenin çıkışının  $Z=1$  olarak kalması gerekir. Ancak  $Z$  çıkışında anlık bir değişim (statik 1 kaza) oluşur.

Bir devrede kaza tehlikesi olup olmadığı Karnaugh diyagramından da anlaşılabilir.

| Z |   | BC |    |    |    | B |  |
|---|---|----|----|----|----|---|--|
|   |   | 00 | 01 | 11 | 10 |   |  |
| A | 0 |    | 1  |    |    |   |  |
|   | 1 |    | 1  | 1  | 1  |   |  |
|   |   | C  |    |    |    |   |  |

$Z = AB + B'C$

$B'$ 'deki değişim ( $1 \rightarrow 0$ ) bir asal çarpımdan diğerine geçilmesine neden olmaktadır. Böyle geçişler gecikmelerden dolayı kazalara neden olurlar.

Eğer kazalar kesinlikle önlenmek isteniyorsa devrenin maliyeti arttırılarak, aralarında geçiş olan iki çarpımın **konsensüsü** de tasarıma eklenir.

| Z |   | BC |    |    |    | B |  |
|---|---|----|----|----|----|---|--|
|   |   | 00 | 01 | 11 | 10 |   |  |
| A | 0 |    | 1  |    |    |   |  |
|   | 1 |    | 1  | 1  | 1  |   |  |
|   |   | C  |    |    |    |   |  |

$Z = AB + B'C + AC$