



**T.C.
KOCAELİ ÜNİVERSİTESİ**

**TEKNOLOJİ FAKÜLTESİ
BİLİŞİM SİSTEMLERİ MÜHENDİSLİĞİ**

Yrd. Doç. Dr. Mustafa Hikmet Bilgehan UÇAR

6. HAFTA

- **BİLEŞİK MANTIK DEVRELERİ (COMBINATIONAL LOGIC)**
- **Aritmetik İşlem Devreleri**
 - **Toplayıcılar**
 - **Çıkarıcılar**
 - **Karşılaştırıcılar**
 - **Aritmetik Mantık Devresi**

ARİTMETİK İŞLEM DEVRELERİ

- ✓ Toplama, çıkarma, çarpma, bölme işlemlerini yapan devrelere, '**Aritmetik İşlem Devreleri**' denir.
- ✓ Bilgisayarlarda ve hesap makinalarında, temel işlemler toplama ve çıkartma işlemleridir.
- ✓ Çarpma işlemi; toplama işleminin tekrarlanması, bölme işlemi ise; çıkartma işleminin tekrarlanması ile yapılır.

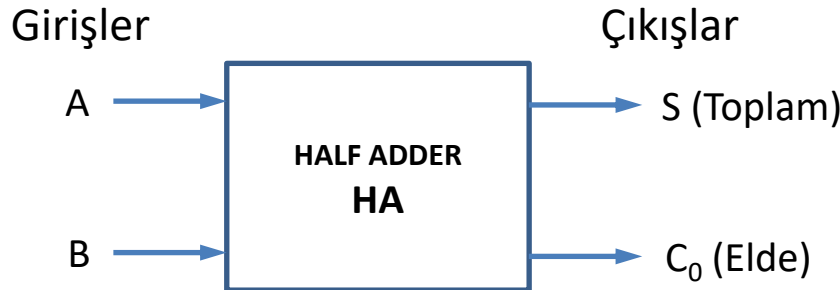
TOPLAYICI (ADDER) DEVRELERİ

- ✓ Bilgisayarlar ve hesap makinaları, her biri çok sayıda bite sahip iki adet ikili sayıyı toplama işlemini gerçekleştirirler.
- ✓ En basit toplama işlemi dört olası temel işlemi içerir.
 - ✓ $0 + 0 = 0$,
 - ✓ $0 + 1 = 1$,
 - ✓ $1 + 0 = 1$,
 - ✓ $1 + 1 = 10$, (Elde 1, Toplam = 0)
- ✓ İlk üç işlemde tek basamaklı bir sayı elde edilirken, son işlemde ikinci basamak ortaya çıkar ve ikinci basamak **'elde biti' (Carry Bit)** olarak isimlendirilir.
- ✓ İki biti toplayan devreler **'yarım toplayıcı'** olarak, elde değerini temsil eden biti üçüncü bit olarak değerlendirilen ve üç bitin toplamını yapan devreler ise **'tam toplayıcı'** olarak isimlendirilir.
- ✓ Tam toplayıcıyı oluşturmak için iki tane yarım toplayıcı kullanılır.

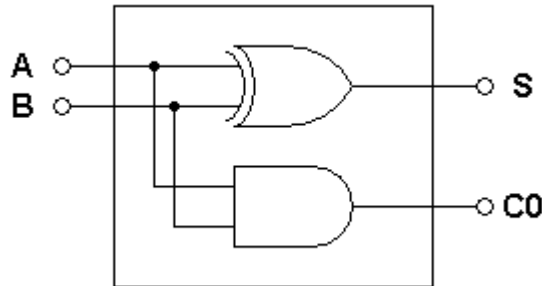
TOPLAYICI (ADDER) DEVRELERİ

✓ YARIM TOPLAYICI (HALF ADDER)

- ✓ Girişine uygulanan iki biti toplayıp, sonucu toplam (Sum) ve elde (Carry) şeklinde veren toplayıcı devresi, 'yarım toplayıcı' olarak isimlendirilir.
- ✓ Yarım toplayıcı devresi, doğruluk tablosundan elde edilen fonksiyonların lojik devresinin çizilmesi ile oluşturulur.
- ✓ Oluşan devrede, 'Toplam' ve 'Elde' değerlerini temsil eden iki çıkış bulunur.



GİRİŞLER		ÇIKIŞLAR	
A	B	S	C ₀
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1



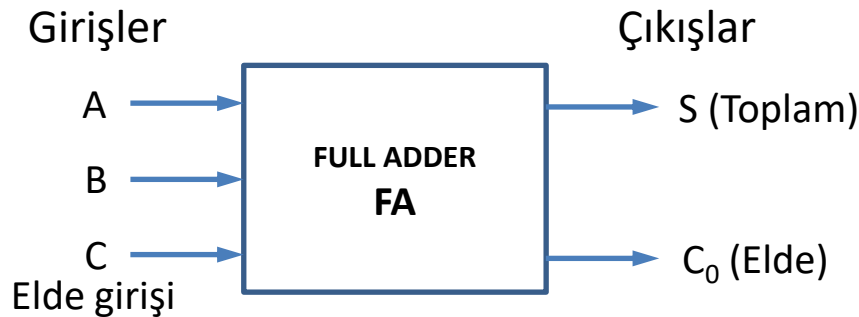
- ✓ Yarım toplayıcı çıkışlarındaki sadeleştirilmiş fonksiyonlar, $S = \bar{A}B + A\bar{B}$ ve $C = AB$ şeklinde elde edilir.

- ✓ Girişlerin A ve B, çıkışların S ve C değişkenleri ile ifade edildiği yarım toplayıcı devresi, bir 'ÖZEL-VEYA' (EXOR) ve bir 'VE' kapısı ile oluşturulabilir.

TOPLAYICI (ADDER) DEVRELERİ

✓ TAM TOPLAYICI (FULL ADDER)

- ✓ Bir bitlik üç adet sayının toplamını gerçekleştiren ve sonucu S ve C₀ olmak üzere iki çıkış hattında gösteren devre, '**Tam Toplayıcı**' olarak isimlendirilir.
- ✓ Girişlerden ikisi toplanacak bitleri gösterirken, üçüncü giriş bir önceki düşük değerlikli basamaktan gelen eldeyi (Carry) ifade etmek için kullanılır.
- ✓ Tam toplayıcı devresi tasarlamak doğruluk tablosundan faydalanılabilir.



GİRİŞLER			ÇIKIŞLAR	
C	A	B	S	C ₀
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 toplayıcı çıkışlarındaki sadeleştirilmiş fonksiyonlar,

$$S = \bar{A}\bar{B}C + A\bar{B}\bar{C} + \bar{A}B\bar{C} + ABC$$

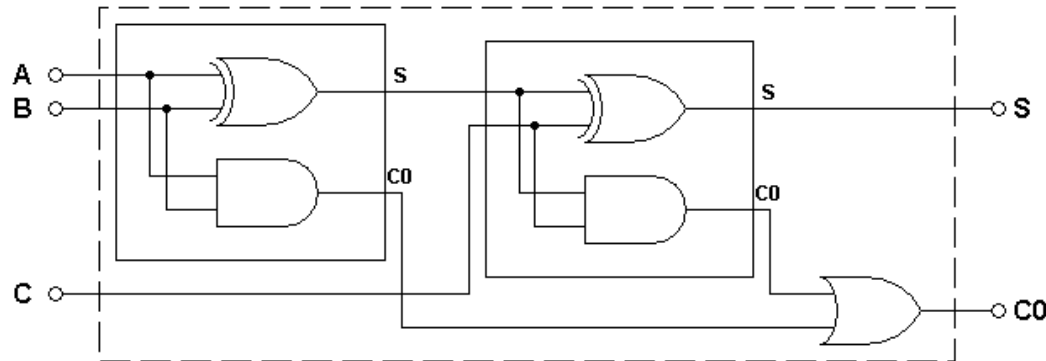
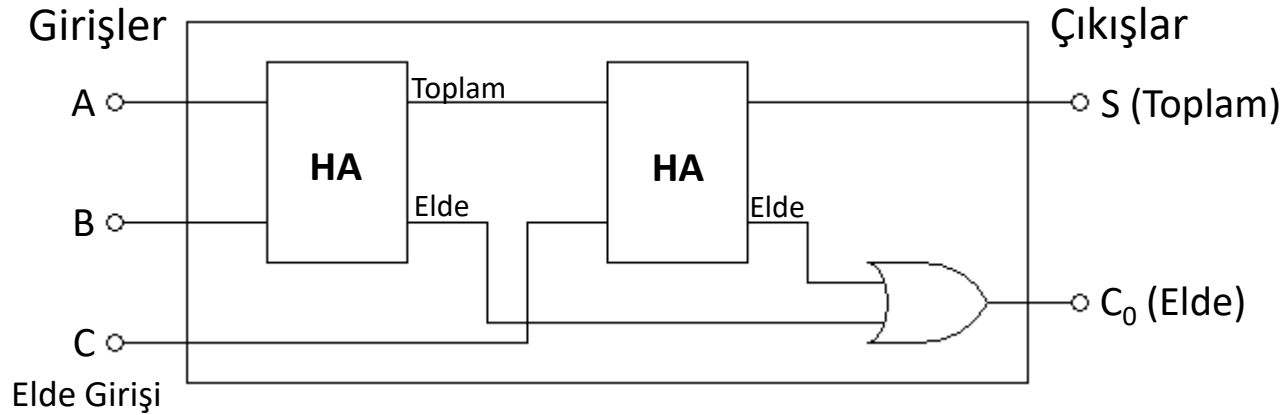
$$C_0 = AB + BC + AC$$

şeklinde elde edilir. Bu ifade matematiksel olarak (A+B)+C olarak gösterilebilir

TOPLAYICI (ADDER) DEVRELERİ

✓ TAM TOPLAYICI (FULL ADDER)

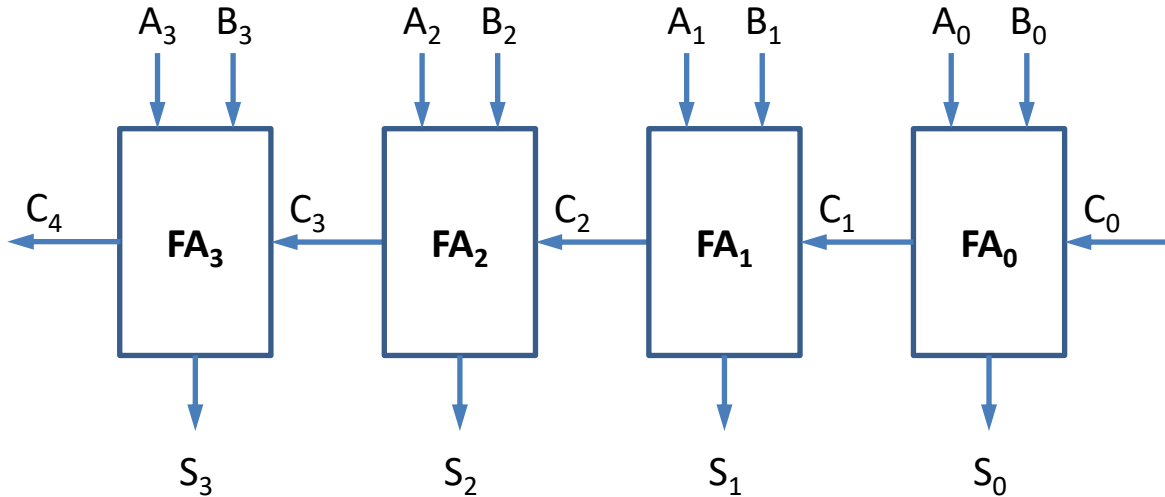
- ✓ Tasarım sonucunda doğrudan lojik devre çizilebileceği gibi tam toplama işlemi, iki adet yarım toplayıcı ve bir 'VEYA' kapısı kullanılarak gerçekleştirilebilir.
- ✓ Bu şekilde gerçekleştirilen devrenin S çıkışı; ilk yarım toplayıcının S çıkışı ile elde girişi C'nin ÖZEL-VEYA'ya uygulanmasının sonucu elde edilir.



TOPLAYICI (ADDER) DEVRELERİ

✓ PARALEL TOPLAYICI

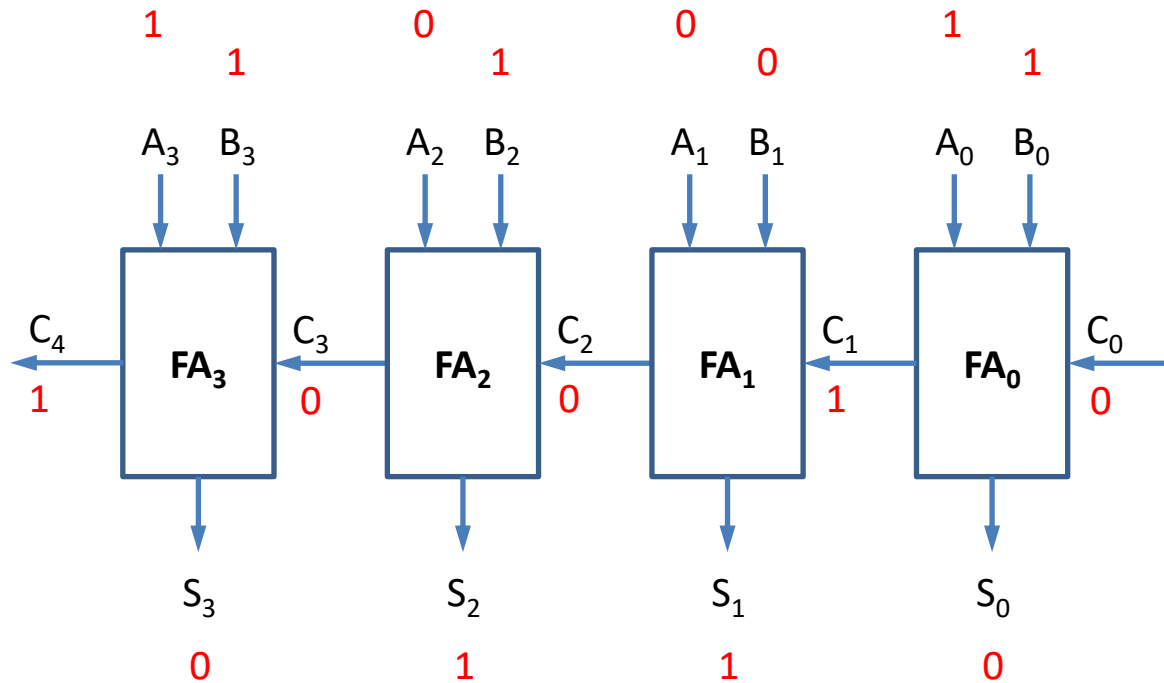
- ✓ Her biri çok sayıda ikili basamak içeren iki sayının toplanması işlemini aynı anda yapan devrelere 'paralel toplayıcı' denir.
- ✓ Bilgisayarlarda ve hesap makinalarında çok sayıda bite sahip iki sayıyı aynı anda toplayan devreler bulunur.
- ✓ En düşük değerli basamakta C_0 biti '0' olduğundan; A_0 ve B_0 değerleri toplanarak S_0 ve C_1 çıkışlarına gönderilir.
- ✓ Bunun dışındaki basamakları toplamak için, A_i , B_i , C_i bitleri toplanarak ilgili S_i ve C_i çıkışlarında gösterilir.



TOPLAYICI (ADDER) DEVRELERİ

✓ **ÖRNEK:**

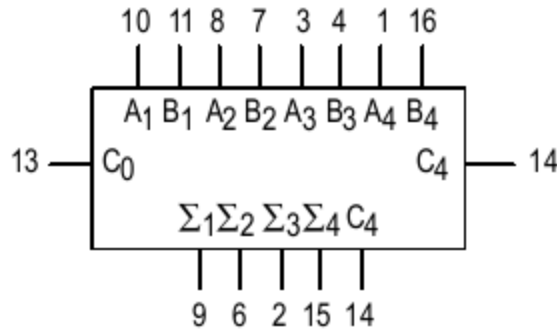
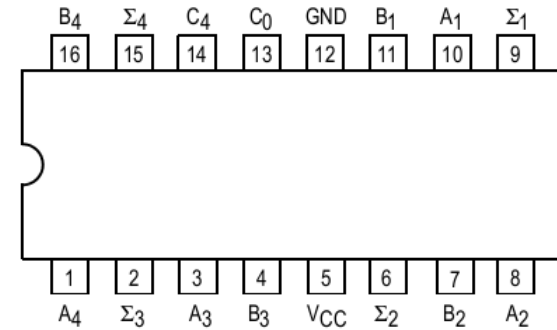
✓ $(1001)_2 + (1101)_2$ işlemini paralel toplayıcı üzerinde gösteriniz.



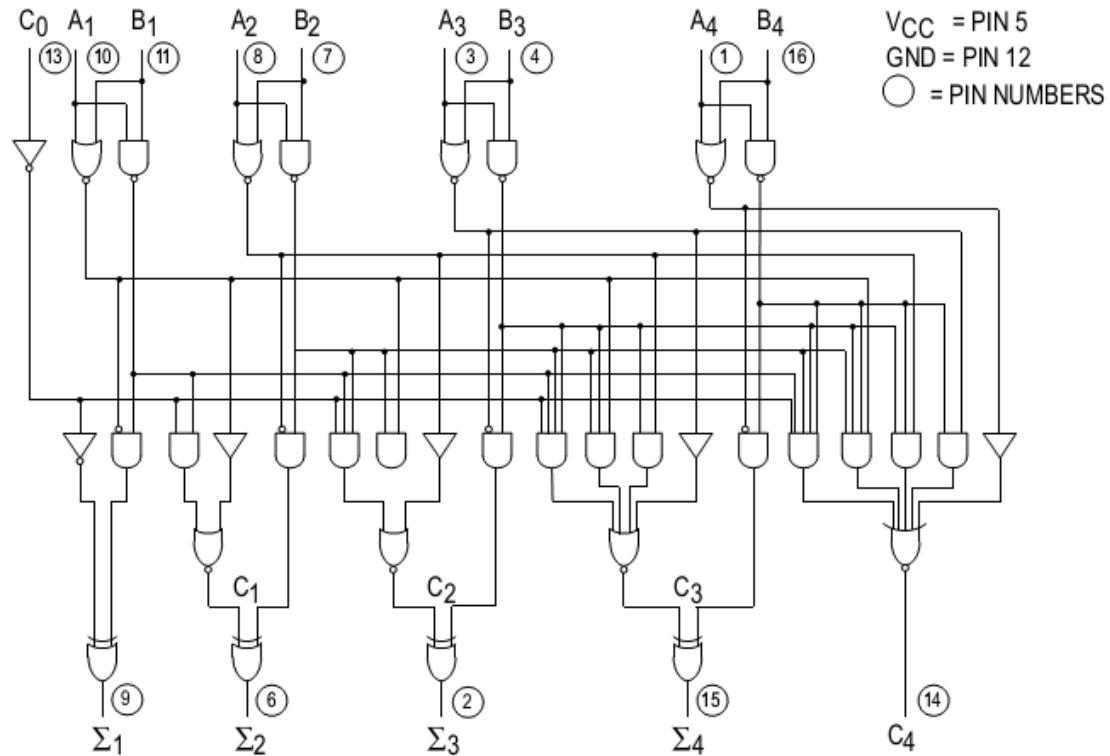
TOPLAYICI (ADDER) DEVRELERİ

✓ PARALEL TOPLAYICI

✓ 74LS83 4 bit paralel toplayıcı entegresidir.



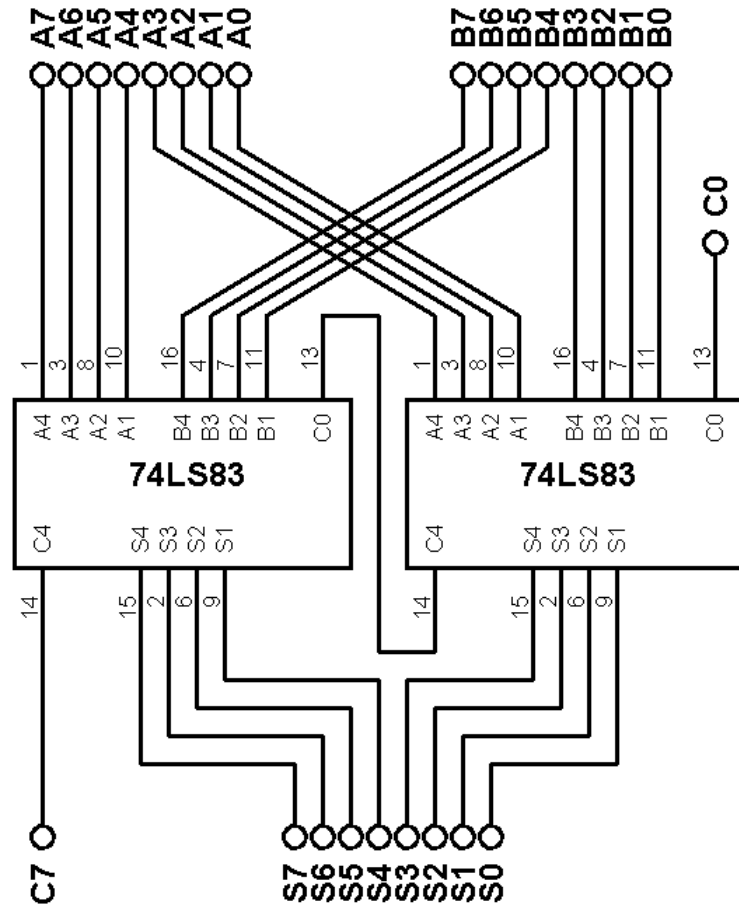
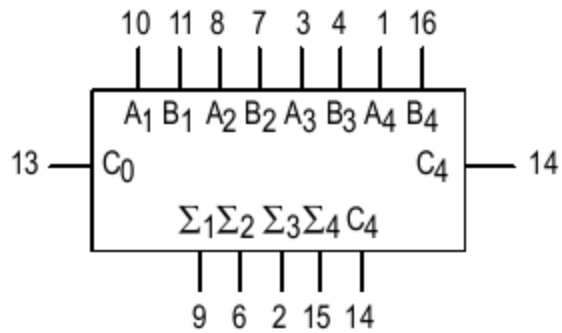
C (n-1)	A _n	B _n	Σ _n	C _n
L	L	L	L	L
L	L	H	H	L
L	H	L	H	L
L	H	H	L	H
H	L	L	H	L
H	L	H	L	H
H	H	L	L	H
H	H	H	H	H



TOPLAYICI (ADDER) DEVRELERİ

✓ PARALEL TOPLAYICI

- ✓ İki adet 74LS83 dört bit paralel toplayıcı entegresi kullanarak 8 bit paralel toplayıcı devresinin kurulması



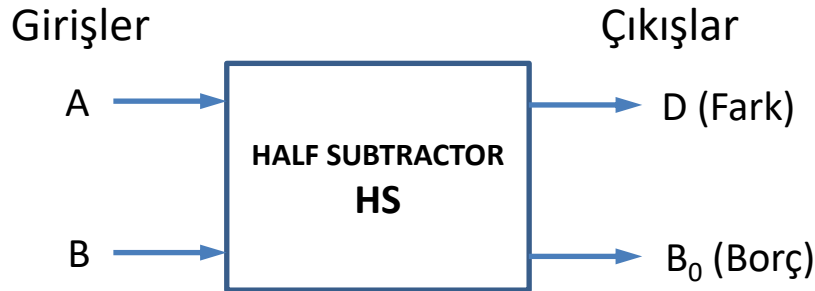
ÇIKARMA (SUBTRACTOR) DEVRELERİ

- ✓ En basit çıkarma işlemi dört olası temel işlemi içerir.
 - ✓ $0 - 0 = 0$,
 - ✓ $0 - 1 = 1$, (Borç = 1)
 - ✓ $1 - 0 = 1$,
 - ✓ $1 - 1 = 0$,
- ✓ '0'dan '1' çıkarılırken bir soldaki basamaktan '1' alındığı kabul edilerek işlem yapılan sütuna 2 olarak aktarılır. Bu nedenle fark '1' ödünç alınan değer '1' olur.
- ✓ Küçük sayıdan büyük sayı çıkarıldığında **'borç biti' (Borrow Bit)** oluşur.
- ✓ Dolayısıyla çıkarıcının fark ve borç olmak üzere iki çıkışı mevcuttur.
- ✓ İki bit arasında çıkarma işlemi yapan devrelere **'yarım çıkarıcı'**, borç değerini temsil eden biti üçüncü bit olarak değerlendirip üç bitle çıkarma işlemi yapan devrelere ise **'tam çıkarıcı'** denir.
- ✓ Yarım çıkarıcı terimi, tam çıkarıcıyı oluşturmak için iki tane yarım çıkarıcı kullanılmasından ileri gelir.

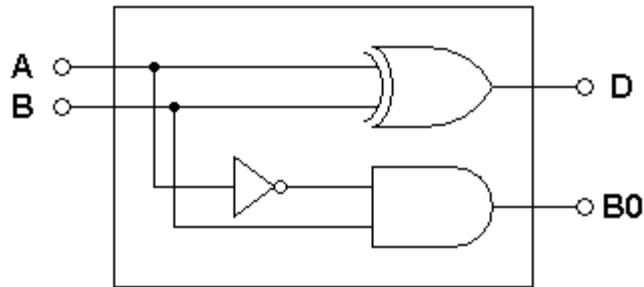
ÇIKARMA (SUBTRACTOR) DEVRELERİ

✓ YARIM ÇIKARICI (HALF SUBTRACTOR)

- ✓ Girişine uygulanan iki bitin farkını alıp, sonucu Fark (Difference) ve borç (Borrow) şeklinde veren çıkarıcı devresi, 'yarım çıkarıcı' olarak isimlendirilir.
- ✓ Yarım çıkarıcı devresi, doğruluk tablosundan elde edilen fonksiyonların lojik devresinin çizilmesi ile oluşturulur.



GİRİŞLER		ÇIKIŞLAR	
A	B	D	B ₀
0	0	0	0
0	1	1	1
1	0	1	0
1	1	0	0

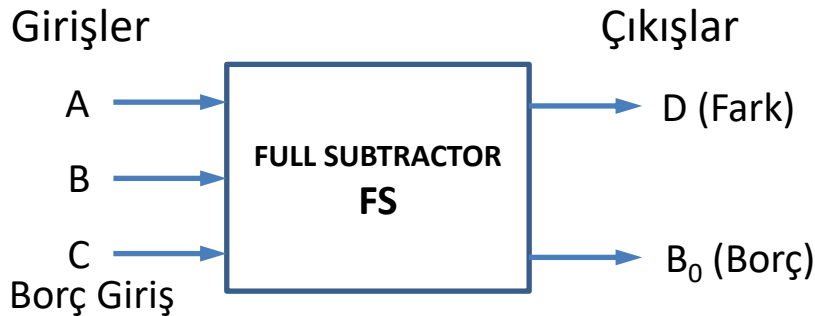


- ✓ Yarım toplayıcı çıkışlarındaki sadeleştirilmiş fonksiyonlar, $D = \bar{A}B + A\bar{B}$ ve $B_0 = \bar{A}B$ şeklinde elde edilir.

ÇIKARMA (SUBTRACTOR) DEVRELERİ

✓ TAM ÇIKARICI (FULL SUBTRACTOR)

- ✓ Bir bitlik üç adet sayının fark alınması işlemini gerçekleştiren ve sonucu D ve B₀ olmak üzere iki çıkış hattında gösteren devre, '**Tam Çıkarıcı**' olarak isimlendirilir.
- ✓ Girişlerden ikisi farkı alınacak bitleri gösterirken, üçüncü giriş bir önceki düşük değerlikli basamaktan gelen borç yükünü (Borrow) ifade etmek için kullanılır.
- ✓ Tam çıkarıcı devresi tasarlamak doğruluk tablosundan faydalanılabilir.



GİRİŞLER			ÇIKIŞLAR	
C	A	B	D	B ₀
0	0	0	0	0
0	0	1	1	1
0	1	0	1	0
0	1	1	0	0
1	0	0	1	1
1	0	1	0	1
1	1	0	0	0
1	1	1	1	1

- ✓ Tam çıkarıcı çıkışlarındaki sadeleştirilmiş fonksiyonlar,

$$D = \bar{A}B\bar{C} + A\bar{B}\bar{C} + \bar{A}\bar{B}C + ABC$$

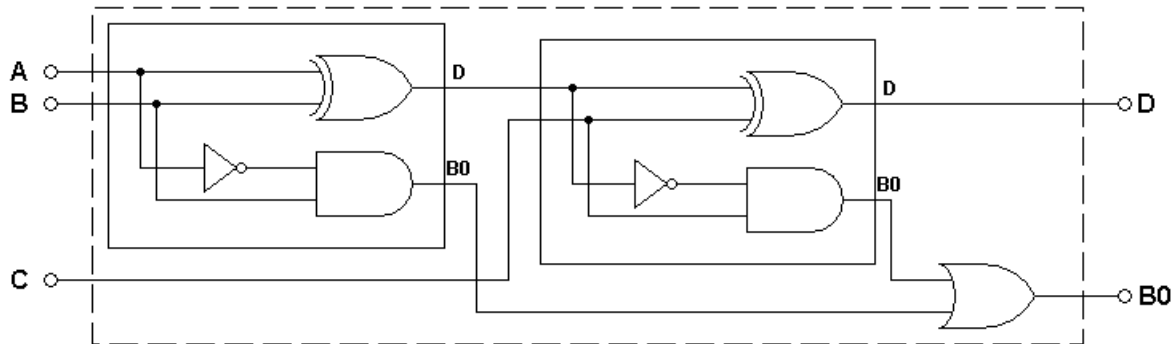
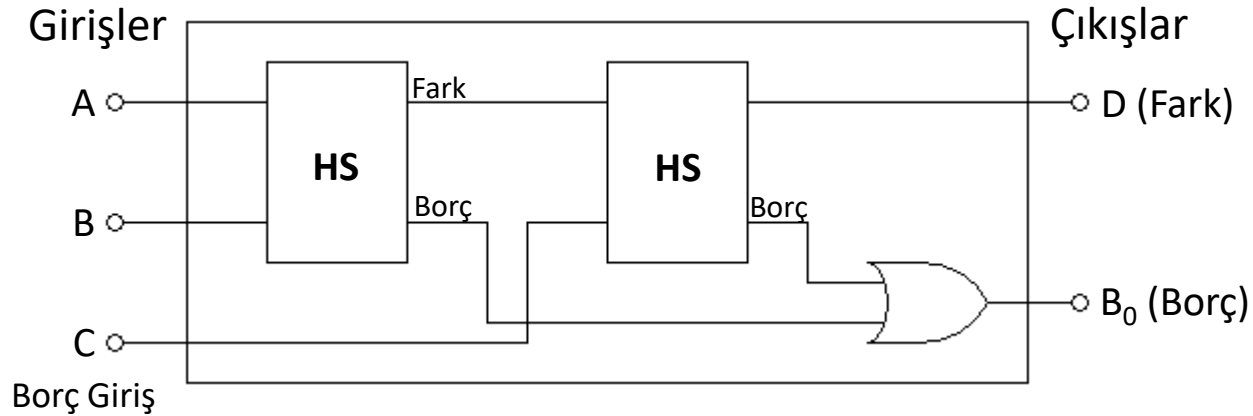
$$B_0 = \bar{A}B + \bar{A}C + BC$$

şeklinde elde edilir. Bu ifade matematiksel olarak (A-B)-C olarak gösterilebilir.

ÇIKARMA (SUBTRACTOR) DEVRELERİ

✓ TAM ÇIKARICI (FULL SUBTRACTOR)

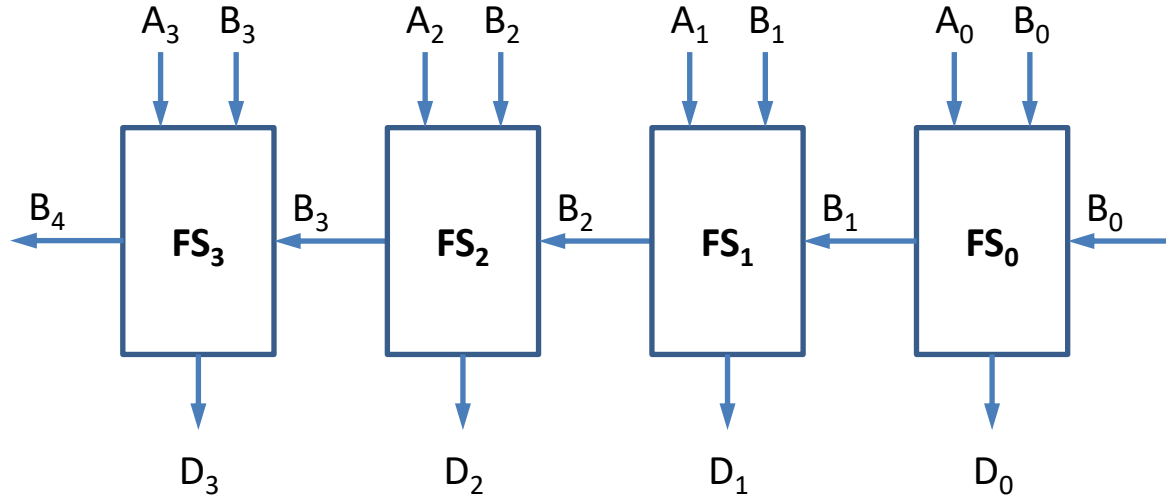
- ✓ Tasarım sonucunda doğrudan lojik devre çizilebileceği gibi tam çıkarıcı işlemi, iki adet yarım çıkarıcı ve bir 'VEYA' kapısı kullanılarak gerçekleştirilebilir.
- ✓ Bu şekilde gerçekleştirilen devrede D çıkışı; ilk çıkarıcının D çıkışı ile borç girişine ÖZEL-VEYA uygulanması sonucu elde edilir.



ÇIKARMA (SUBTRACTOR) DEVRELERİ

✓ PARALEL ÇIKARICI

- ✓ Her biri çok sayıda ikili basamak içeren iki sayının farkının alınması işlemini aynı anda yapan devrelere 'paralel çıkarıcı' denir.
- ✓ Bilgisayarlarda ve hesap makinalarında çok sayıda bite sahip iki sayıyı aynı anda farkını alan devreler bulunur.
- ✓ En düşük değerli basamakta B_0 biti '0' olduğundan; A_0 ve B_0 değerleri farkı alınarak D_0 ve B_1 çıkışlarına gönderilir.
- ✓ Bunun dışındaki basamakların farkını almak için, A_i , B_i , C_i bitleri farkı alınarak ilgili D_i ve B_i çıkışlarında gösterilir.



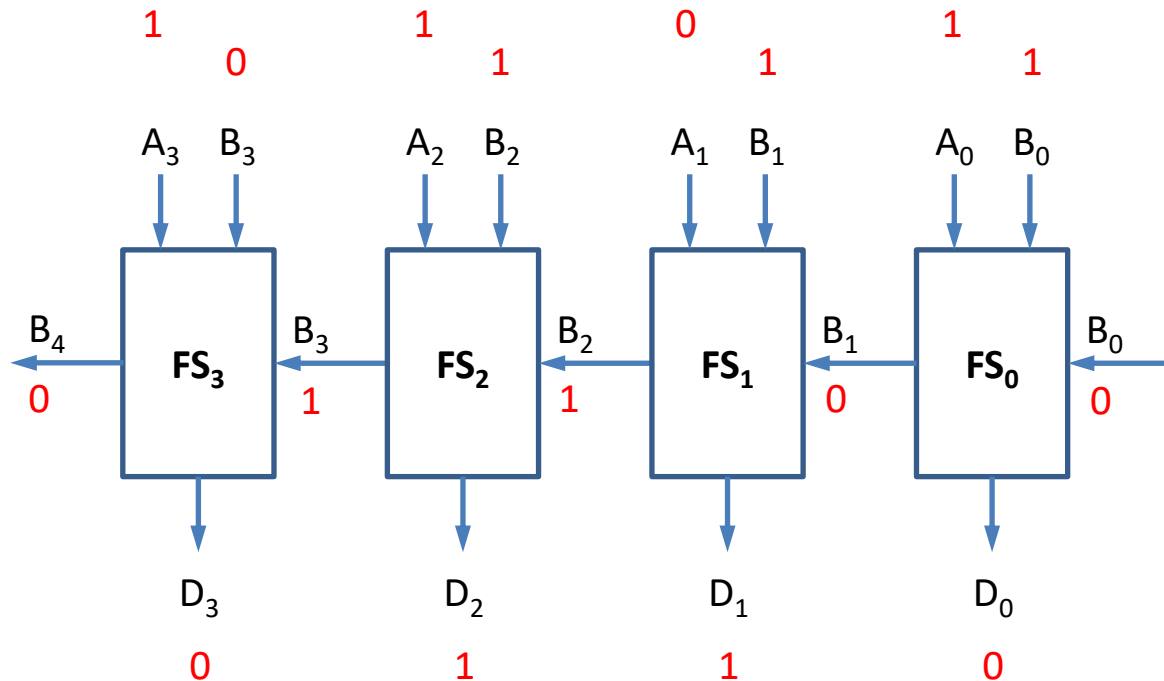
ÇIKARMA (SUBTRACTOR) DEVRELERİ

✓ **ÖRNEK:**

✓ $(1101)_2 - (0111)_2$ işlemini paralel çıkarıcı üzerinde gösteriniz.

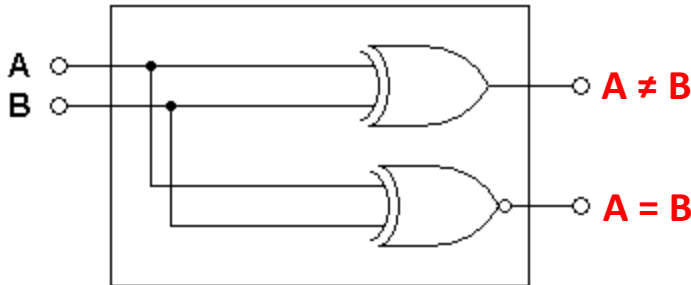
1101
0111

0110



KARŞILAŞTIRMA (COMPARATOR) DEVRELERİ

- ✓ İki sayıyı karşılaştıran ve büyüklüklerini belirleyen bileşik devreler, '**büyüklik karşılaştırıcı**' (**magnitude comparator**) olarak isimlendirilir.
- ✓ Karşılaştırma sonucu; ' $A > B$ ', ' $A = B$ ' veya ' $A < B$ 'yi belirleyen üç konum ile belirlenir.
- ✓ En yaygın kullanım yerleri aritmetik lojik devrelerdir.
- ✓ Karşılaştırıcı devreleri, girişleri aynı veya farklı iken çıkış veren kontrol devrelerinde ve ikili karşılaştırmanın kullanıldığı adres bulma devrelerinde kullanılır.
- ✓ En basit karşılaştırıcı devresi, tek bitlik A ve B sayılarının eşitlik durumunu karşılaştıran karşılaştırıcı devresidir.
- ✓ Bu devrede ' $A = B$ ' durumunda çıkışlardan birisi '1' olurken, ' $A \neq B$ ' durumunda diğeri '1' olur.



GİRİŞLER		ÇIKIŞLAR	
A	B	$A \neq B$	$A = B$
0	0	0	1
0	1	1	0
1	0	1	0
1	1	0	1

KARŞILAŞTIRMA (COMPARATOR) DEVRELERİ

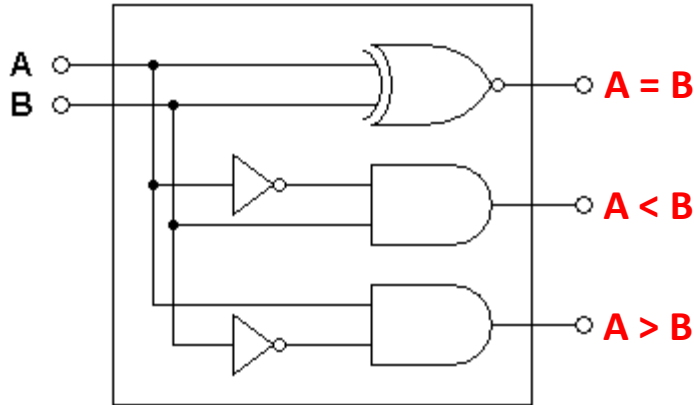
✓ Bir bitlik sayıyı karşılaştıran, 'A > B', 'A = B' ve 'A < B' çıkışları üreten devreyi tasarlayalım.

✓ $Q_{(A=B)} = \bar{A}\bar{B} + AB = \overline{A \oplus B}$

✓ $Q_{(A < B)} = \bar{A}B$

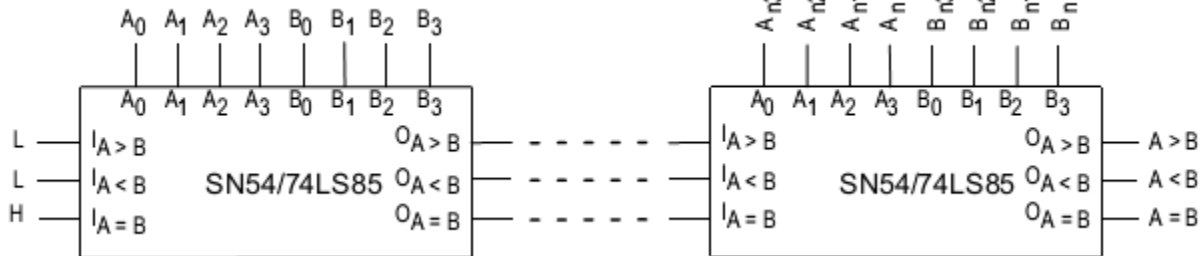
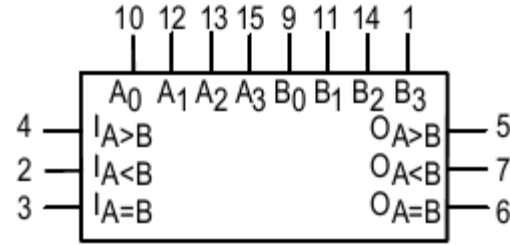
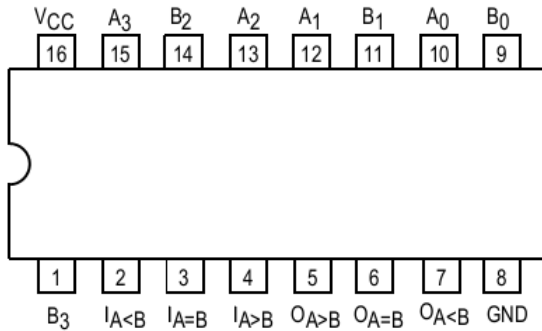
✓ $Q_{(A > B)} = A\bar{B}$

GİRİŞLER		ÇIKIŞLAR		
A	B	A > B	A = B	A < B
0	0	0	1	0
0	1	0	0	1
1	0	1	0	0
1	1	0	1	0



KARŞILAŞTIRMA (COMPARATOR) DEVRELERİ

- ✓ 74LS85 dört bit karşılaştırıcı entegresidir.
- ✓ Dört bitlik iki sayıyı karşılaştırarak 'A > B', 'A = B' ve 'A < B' olmak üzere çıkışlar üretir.
- ✓ Birden fazla 74LS85 entegresi kullanılarak karşılatırmadaki bit sayısı arttırılabilir.
- ✓ Bu sebeple 3 farklı çıkış ile birlikte, devir amaçlı kullanılmak üzere karşılaştırma girişleri(kaskat girişler) de bulunur.



KARŞILAŞTIRMA (COMPARATOR) DEVRELERİ

- ✓ Büyüklük karşılaştırıcılar, bilgisayarlarda (mikroişlemci içeren sistemlerde) adres kod çözücü devrelerin bir parçası olarak kullanılır.
- ✓ Bu kullanımda, bir karşılaştırıcı bilgisayarın merkezi işlem biriminin ürettiği adres kodu ile bellekte bulunan adresin kodunu karşılaştırır.
- ✓ Karşılaştırılan kodlar eşitse, istenilen işlemi yapmak üzere ilgili devreyi aktif hale getirir.
- ✓ İki tane 74LS85 4 bit karşılaştırıcı ile 8 bitlik iki sayı karşılaştırılsın. Eğer A sayısının en yüksek değerlikli biti (A_3) B sayısının en yüksek değerlikli bitinden (B_3) büyükse her zaman $Q_{A>B}=1$ 'dir. Eğer her iki sayının en yüksek değerlikli 4 biti eşitse bu sefer $A=B$ ise Birinci 74LS85 4 bitlik karşılaştırıcıdan gelen sonuca bakılır. Eğer

✓ $I_{A>B}$ ise $Q_{A>B}$

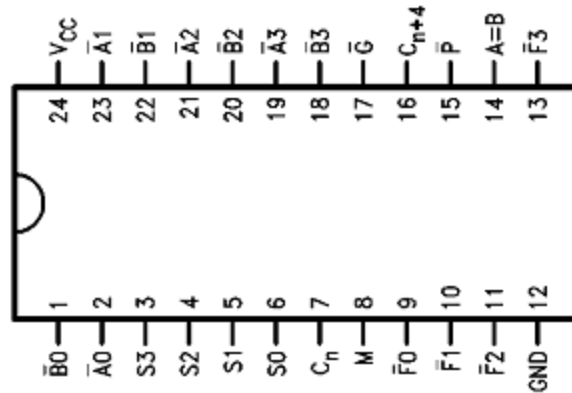
✓ $I_{A<B}$ ise $Q_{A>B}$

✓ $I_{A=B}$ ise $Q_{A=B}$

A_3, B_3	A_2, B_2	A_1, B_1	A_0, B_0	$I_{A>B}$	$I_{A<B}$	$I_{A=B}$	$Q_{A>B}$	$Q_{A<B}$	$Q_{A=B}$
$A_3 > B_3$	X	X	X	X	X	X	1	0	0
$A_3 < B_3$	X	X	X	X	X	X	0	1	0
$A_3 = B_3$	$A_2 > B_2$	X	X	X	X	X	1	0	0
$A_3 = B_3$	$A_2 < B_2$	X	X	X	X	X	0	1	0
$A_3 = B_3$	$A_2 = B_2$	$A_1 > B_1$	X	X	X	X	1	0	0
$A_3 = B_3$	$A_2 = B_2$	$A_1 < B_1$	X	X	X	X	0	1	0
$A_3 = B_3$	$A_2 = B_2$	$A_1 = B_1$	$A_0 > B_0$	X	X	X	1	0	0
$A_3 = B_3$	$A_2 = B_2$	$A_1 = B_1$	$A_0 < B_0$	X	X	X	0	1	0
$A_3 = B_3$	$A_2 = B_2$	$A_1 = B_1$	$A_0 = B_0$	1	0	0	1	0	0
$A_3 = B_3$	$A_2 = B_2$	$A_1 = B_1$	$A_0 = B_0$	0	1	0	0	1	0
$A_3 = B_3$	$A_2 = B_2$	$A_1 = B_1$	$A_0 = B_0$	0	0	1	0	0	1
$A_3 = B_3$	$A_2 = B_2$	$A_1 = B_1$	$A_0 = B_0$	0	0	0	0	0	1

ARİTMETİK MANTIK DEVRESİ (ARITHMETIC LOGIC UNIT - ALU)

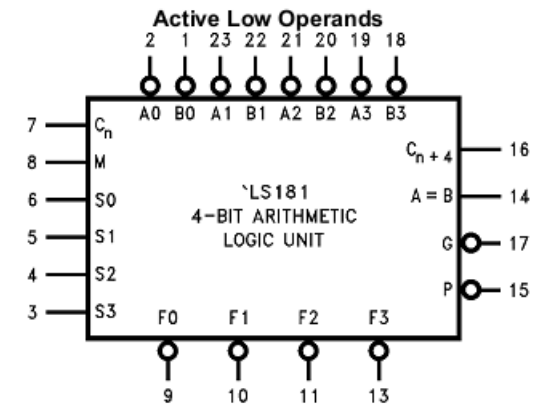
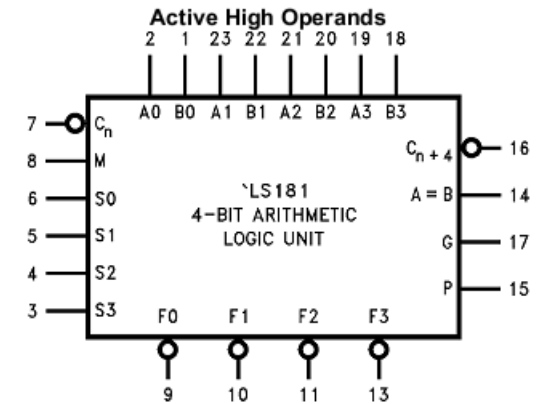
- ✓ Temel aritmetik işlemlerin (toplama, çıkarma, çarpma) ve lojik işlemlerin (VE, VEYA, VEDEĞİL, v.b) yapılmasında kullanılan çok fonksiyonlu Aritmetik Mantık Devresi entegreleri bulunmaktadır.
- ✓ Bu entegrelere örnek olarak 74181 entegresi verilebilir.
- ✓ 74181 entegresi, aritmetik veya mantık işlemlerinden birisini yapacak şekilde programlanabilir.
- ✓ Bu işlem entegrenin 8 numaralı M giriş ucu ile yapılır.
- ✓ Entegre, 'M = 0' iken aritmetik işlemleri, M=1 iken ise lojik işlemleri yapmak üzere programlanır.



ARİTMETİK MANTIK DEVRESİ (ARITHMETIC LOGIC UNIT - ALU)

- ✓ 4 tanesi A sayısı girişi (A_3, A_2, A_1, A_0), 4 tanesi B sayısı girişi (B_3, B_2, B_1, B_0), 4 tanesi de A ve B sayıları arasında yapılan aritmetik ve lojik işlemler sonucu ortaya çıkan değerlerin (F ile gösterilen) çıkışı (S_0, S_1, S_2, S_3) için kullanılır.
- ✓ S_0, S_1, S_2, S_3 dizisi ile yapılacak işlemi (toplama, çıkarma, AND, OR, v.b.) seçmek için kullanılır.
- ✓ Örneğin; $M=1$ ve $S_3S_2S_1S_0 = 1110$ ise ' $F = A + B$ ' işlemi gerçekleştirilir.
- ✓ Diğer bir deyişle, A girişleri ile B girişleri 'VEYA' işlemine tabi tutulur.

Pin Names	Description
$\overline{A0}-\overline{A3}$	Operand Inputs (Active LOW)
$\overline{B0}-\overline{B3}$	Operand Inputs (Active LOW)
$S0-S3$	Function Select Inputs
M	Mode Control Input
C_n	Carry Input
$\overline{F0}-\overline{F3}$	Function Outputs (Active LOW)
A = B	Comparator Output
\overline{G}	Carry Generate Output (Active LOW)
\overline{P}	Carry Propagate Output (Active LOW)
C_{n+4}	Carry Output



V_{CC} = Pin 24
GND = Pin 12

ARİTMETİK MANTIK DEVRESİ (ARITHMETIC LOGIC UNIT - ALU)

- ✓ Bu girişlerin farklı kombinasyona sahip olması ile yapılabilecek işlemler, fonksiyon tablosu olarak entegre veri kataloglarında verilmektedir.
- ✓ 74181 fonksiyon tablosuna göre, entegreyi beş çeşit aritmetik işleme, 16 tip lojik işleme ve 8 tip hem lojik hem de aritmetik işleme programlamak mümkündür.
- ✓ Bunların anlamı; 74181 entegresi ile 32 çeşit işlem yapmanın mümkün olduğudur.

Mode Select Inputs				Active LOW Operands & F _n Outputs		Active HIGH Operands & F _n Outputs	
S3	S2	S1	S0	Logic	Arithmetic	Logic	Arithmetic
				(M = H)	(M = L) (C _n = L)	(M = H)	(M = L) (C _n = H)
L	L	L	L	\bar{A}	A minus 1	\bar{A}	A
L	L	L	H	\overline{AB}	AB minus 1	$\bar{A} + \bar{B}$	A + B
L	L	H	L	$\bar{A} + \bar{B}$	\overline{AB} minus 1	$\bar{A} B$	A + \bar{B}
L	L	H	H	Logic 1	minus 1	Logic 0	minus 1
L	H	L	L	$\bar{A} + \bar{B}$	A plus (A + \bar{B})	\overline{AB}	A plus \overline{AB}
L	H	L	H	\bar{B}	AB plus (A + \bar{B})	\bar{B}	(A + B) plus \overline{AB}
L	H	H	L	$\bar{A} \oplus \bar{B}$	A minus B minus 1	A \oplus B	A minus B minus 1
L	H	H	H	A + \bar{B}	A + \bar{B}	\overline{AB}	AB minus 1
H	L	L	L	$\bar{A} B$	A plus (A + B)	$\bar{A} + B$	A plus AB
H	L	L	H	A \oplus B	A plus B	$\bar{A} \oplus \bar{B}$	A plus B
H	L	H	L	B	\overline{AB} plus (A + B)	B	(A + \bar{B}) plus AB
H	L	H	H	A + B	A + B	AB	AB minus 1
H	H	L	L	Logic 0	A plus A (Note 1)	Logic 1	A plus A (Note 1)
H	H	L	H	\overline{AB}	AB plus A	A + \bar{B}	(A + B) plus A
H	H	H	L	AB	\overline{AB} minus A	A + B	(A + \bar{B}) plus A
H	H	H	H	A	A	A	A minus 1