BİLGİSAYAR MİMARİSİ VE ORGANİZASYONU

Ders Kodu : BM303
L+T+P : 0-3-1
AKTS : 4

Dersi Veren :
Prof. Dr. Ahmet Akbaş

Mühendislik Fakültesi
Bilgisayar Müh. Bölümü

BİLECİK ŞEYH EDEBALİ ÜNİVERSİTESİ

2025-26 GYY DERSİ

BM303

BİLGİSAYAR MİMARİSİ ve ORGANİZASYONU

Prof. Dr. A. Akbaş

Ders İçeriği :

- Toplama ve çıkarma devreleri
- Yüksek hızlı toplayıcılar
- İşaretli sayıların Booth algoritması ile çarpımı
- Taşı-kaydet toplamasına dayalı yüksek hızlı çarpanlar
- Bölme mantık devreleri
- IEEE standardı ve kayan noktalı sayıları üzerinde aritmetik işlemler

12.nci HAFTA DERSİ

1- Bilgisayar Sistemlerine Giriş (Fonksiyonel Birimler, Sistem Mimarisi, Performans, Teknolojik Arka Plan);

2- Programcı Bakış Açısıyla bir Sistem Turu;

3- Bilgisayar Sistemlerinde Bilginin (information) Temsili ve Manüplasyonu;

4- Komut Seti Mimarisi, Programların Makine Düzeyinde Temsili-I (RISC ve CISC mimariler);

5- Komut Seti Mimarisi, Programların Makine Düzeyinde Temsili-II (IA-32 ve x86-64 mimarileri);

6- İşlemci ve Giriş/Çıkış Birimleri Arasında Veri Transferi (Program Kontrollü G/Ç, Kesmeli G/Ç);

7- Yazılım (Program Hazırlama ve Çalıştırma Süreçleri, C ve Assembly Dilleri Arasındaki Etkileşim, İşletim Sistemleri);

8- İşlemci (CPU) Birimi (Fonksiyonel Birimler, Komut Getirme vr Yürütme Adımları, Kontrol Sinyalleri, HDL);

9- Program Performansının Optimizasyonu (Komut Düzeyinde Paralelleme, Eş-zamanlılık, Süperskaler İşlemciler);

10- Giriş/Çıkış Organizasyonu (Yol Yapısı, Yol Hakimliği, Arayüz Devreleri, Ara-bağlantı Standartları);

11- Bellek Sistemi (Temel Bellek Devreleri, Ana Bellek Organizasyonu, Bellek Teknolojileri);

12- Aritmetik (Tam Sayı ve Kayan Noktalı Sayılar için Aritmetik İşlem Devreleri);

13- Paralel Çalışma ve Performans (Çoklu İş Parçaları, Vektörel Süreçler, Paylaşımlı Bellekli Çoklu İşlemciler);

14- ARM İşlemcili SoC Örnekleri.

BİLECİK ŞEYH EDEBALİ ÜNİVERSİTESİ

2025-26 GYY DERSİ
BM303
BİLGİSAYAR MİMARİSİ ve ORGANİZASYONU
Prof. Dr. A. Akbaş

ENG. TERMINOLOGY & ACRONYMS

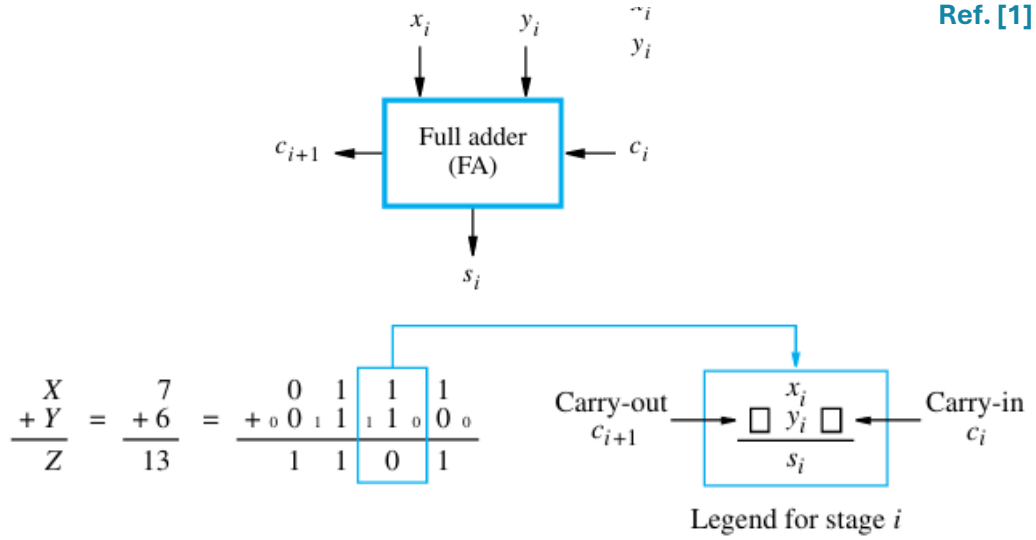
İŞARETLİ TAM SAYILARI TOPLAMA ve ÇIKARMA

İki sayıyı toplama ve çıkarma, tüm bilgisayarlarda makine komutu düzeyindeki temel işlemlerdir. Bu işlemler ve diğer aritmetik ve mantık işlemleri, işlemcinin aritmetik ve mantık biriminde (**ALU**) gerşekleştirilir.

Toplama veya çıkarma işlemlerini gerçekleştirmek için gereken süre, işlemcinin performansını etkiler. Toplama veya çıkarma işlemlerinden daha karmaşık devreler gerektiren çarpma ve bölme işlemleri de performansı etkiler. Bu derste, aritmetik işlemleri gerçekleştirmek için kullanılan mantık devrelerini inceleyeceğiz.

Modern bilgisayarlarda aritmetik işlemleri yüksek hızda gerçekleştirmek için bazı yöntemler geliştirilmiştir. Bunun gibi, kayan noktalı sayıları üzerindeki işlemler için de bazı yöntemler geliştirilmiştir. Buna göre, işaretsiz ikili sayıları toplamak için tasarlanmış bir mantık devresi, 2'ye tümlenmiş işaretli sayıları toplamak için de kullanılabilir.

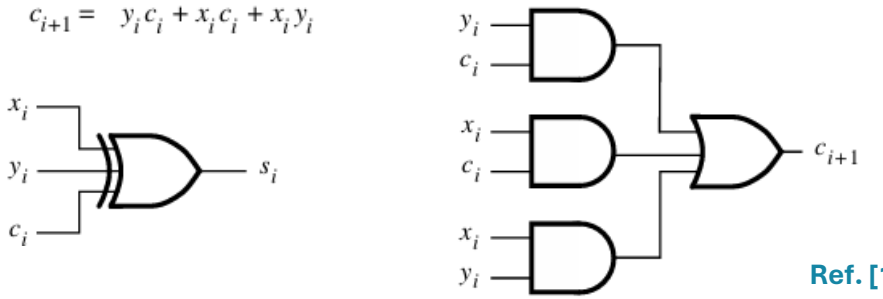
Ref. [1]



x_i	y_i	Carry-in c_i	Sum s_i	Carry-out c_{i+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

$$s_i = \bar{x}_i\bar{y}_i c_i + \bar{x}_i y_i \bar{c}_i + x_i\bar{y}_i\bar{c}_i + x_i y_i c_i = x_i \oplus y_i \oplus c_i$$

$$c_{i+1} = y_i c_i + x_i c_i + x_i y_i$$



Ref. [1]

BİLECİK ŞEYH EDEBALİ ÜNİVERSİTESİ

2025-26 GYY DERSİ

BM303

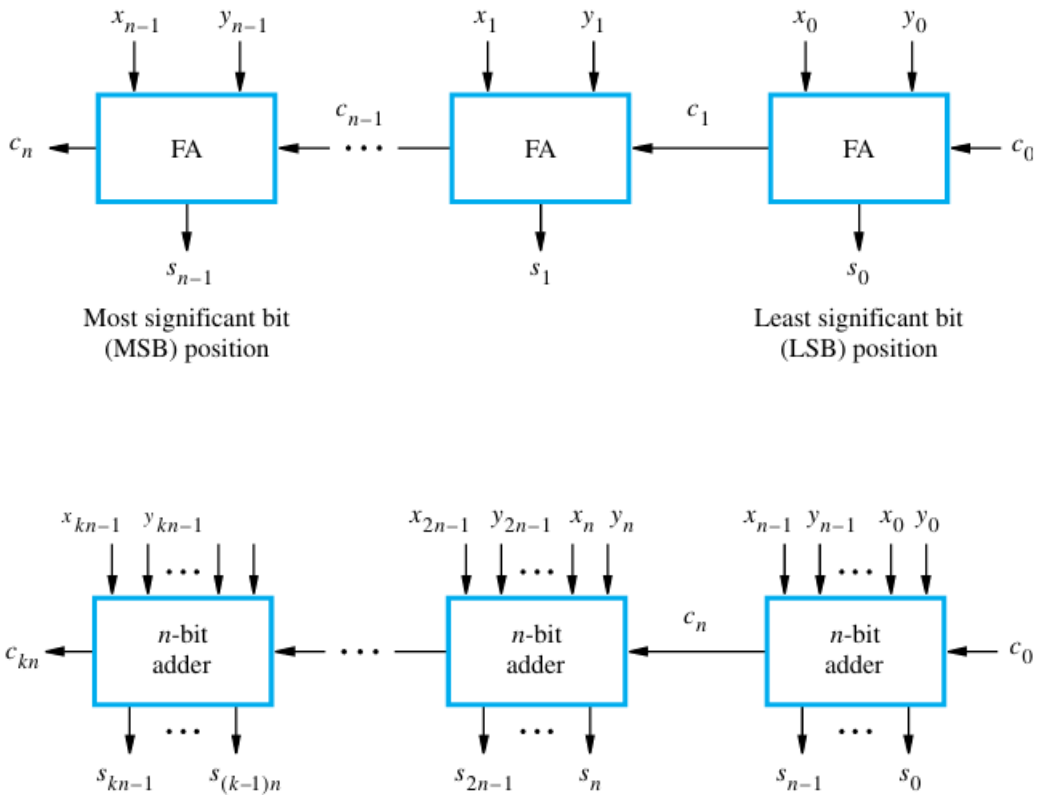
BİLGİSAYAR MİMARİSİ ve ORGANİZASYONU

Prof. Dr. A. Akbaş

ENG. TERMINOLOGY & ACRONYMS

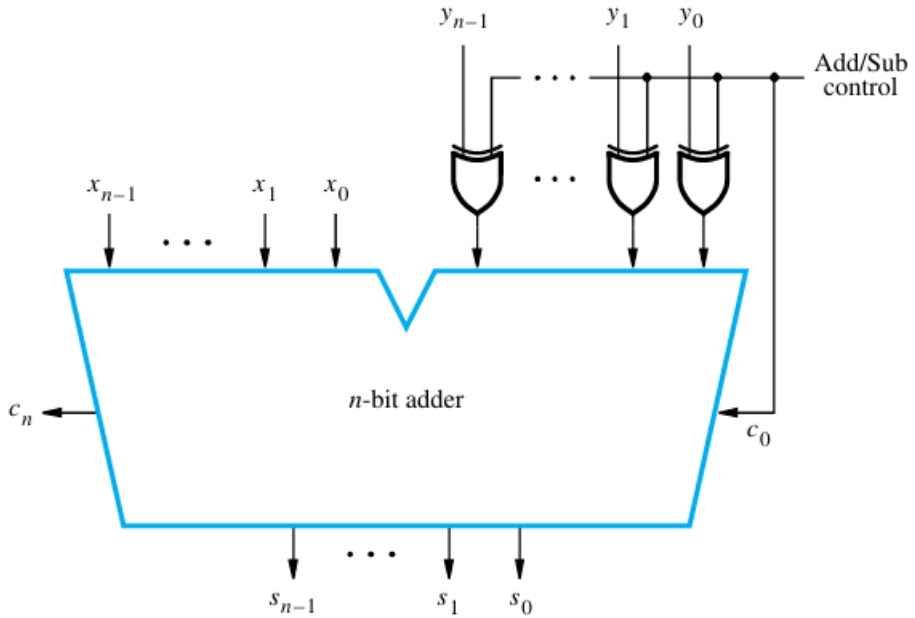
İŞARETLİ TAM SAYILARI TOPLAMA ve ÇIKARMA

***n*-bitlik taşma (elde) taşımalı (ripple-carry) toplayıcı ve bunun *k*-kat kaskatlanmış hali: kxn bit toplayıcı**



Ref. [1]

İkili sayı toplama/çıkarma lojik devresi



Ref. [1]

BİLECİK ŞEYH EDEBALİ ÜNİVERSİTESİ

2025-26 GYY DERSİ
BM303
BİLGİSAYAR MİMARİSİ ve ORGANİZASYONU
Prof. Dr. A. Akbaş

ENG. TERMINOLOGY & ACRONYMS

HIZLI TOPLAYICI TASARIMI

Eldeyi öne taşı (carry-lookahead) yöntemiyle toplama
Hızlı toplayıcı devresinin elde (taşıma) sinyallerinin üretimini hızlandırması gerekir.
Böylece, tüm taşımalar, giriş operantları ***X***, ***Y*** ve c_0 uygulandıktan sonra üç kapı gecikmesi elde edilebilir çünkü tüm ***Pi*** ve ***Gi*** sinyallerini geliştirmek için yalnızca bir kapı gecikmesi gerekir, ardından c_{i+1} için **AND-OR** devresinde iki kapı gecikmesi gelir.

4-bitlik eldeyi öne taşı (**carry-lookahead**) toplayıcı
İkinci **XOR** kapı gecikmesinden sonra, tüm toplam bitleri kullanılabilir. Toplamda, ***n*** bitlik toplama işlemi, ***n***'den bağımsız olarak yalnızca dört kapı gecikmesi gerektirir.

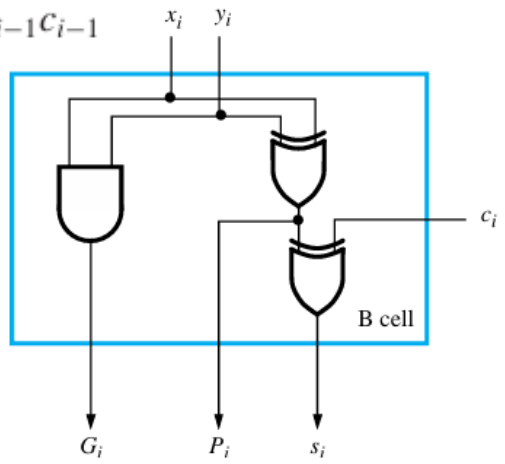
$$s_i = x_i \oplus y_i \oplus c_i \qquad c_{i+1} = x_i y_i + x_i c_i + y_i c_i$$

$$c_{i+1} = x_i y_i + (x_i + y_i) c_i$$

$$G_i = x_i y_i \qquad P_i = x_i + y_i$$

$$c_{i+1} = G_i + P_i c_i$$

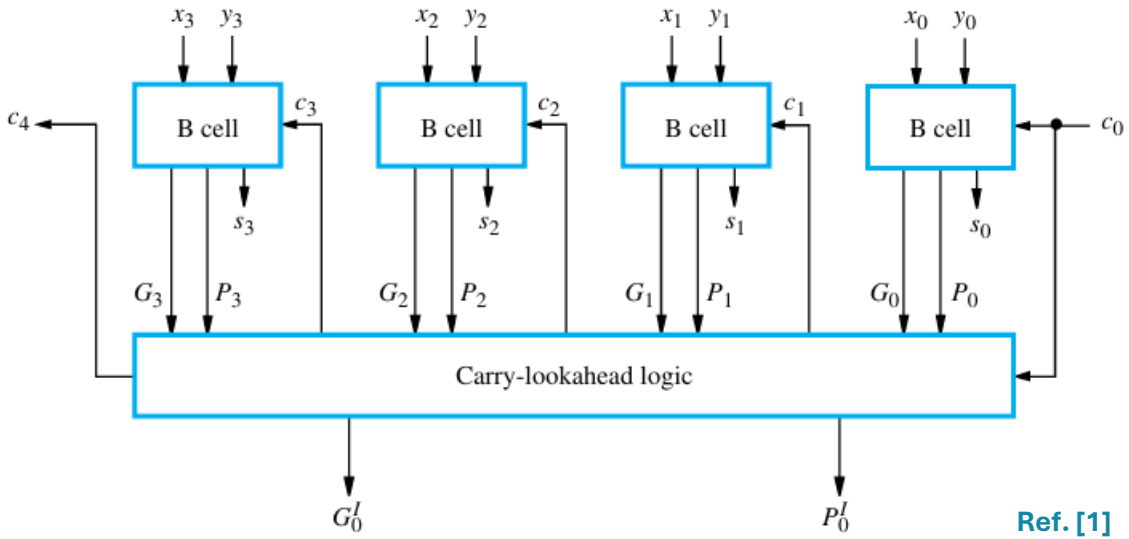
$$c_{i+1} = G_i + P_i G_{i-1} + P_i P_{i-1} c_{i-1}$$



Ref. [1]

$$c_1 = G_0 + P_0 c_0$$
$$c_2 = G_1 + P_1 G_0 + P_1 P_0 c_0$$
$$c_3 = G_2 + P_2 G_1 + P_2 P_1 G_0 + P_2 P_1 P_0 c_0$$
$$c_4 = G_3 + P_3 G_2 + P_3 P_2 G_1 + P_3 P_2 P_1 G_0 + P_3 P_2 P_1 P_0 c_0$$

$$c_{i+1} = G_i + P_i G_{i-1} + P_i P_{i-1} G_{i-2} + \cdots + P_i P_{i-1} \cdots P_1 G_0 + P_i P_{i-1} \cdots P_0 c_0$$



Ref. [1]

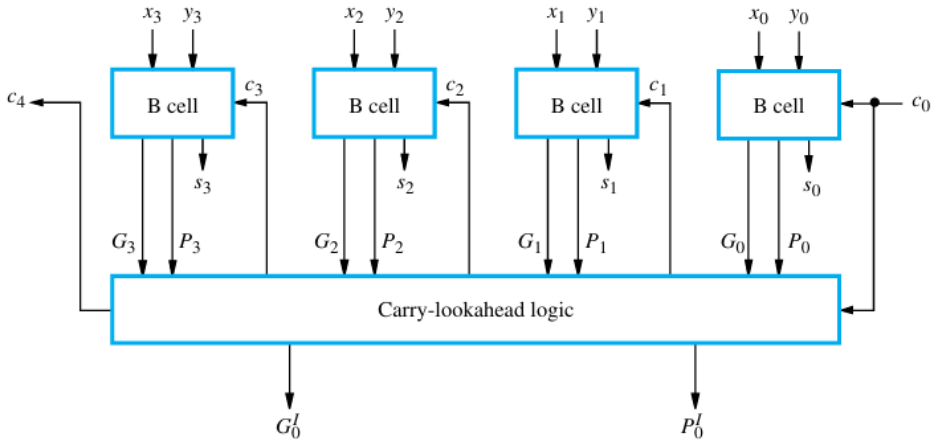
BİLECİK ŞEYH EDEBALİ ÜNİVERSİTESİ

2025-26 GYY DERSİ
BM303
BİLGİSAYAR MİMARİSİ ve ORGANİZASYONU
Prof. Dr. A. Akbaş

ENG. TERMINOLOGY & ACRONYMS

HIZLI TOPLAYICI TASARIMI

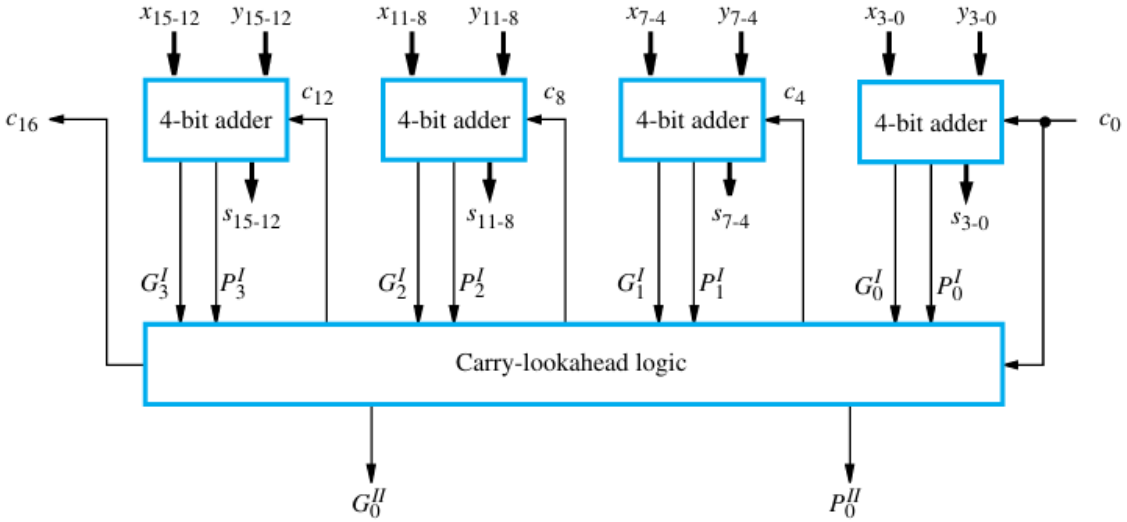
Eldeyi öne taşı (carry-lookahead) yöntemiyle toplama



$$G_0^I = G_3 + P_3G_2 + P_3P_2G_1 + P_3P_2P_1G_0 \qquad P_0^I = P_3P_2P_1P_0$$

Ref. [1]

16-bitlik eldeyi öne taşı (carry-lookahead) toplayıcı
4 bitlik carry-lookahead toplayıcı bloklarının basamaklandırılmasıyla 16 bitlik bir toplayıcı oluşturulursa, c16 ve s15'in geliştirilmesindeki gecikmeler, sırasıyla 9 ve 10 kapı gecikmesi olur.



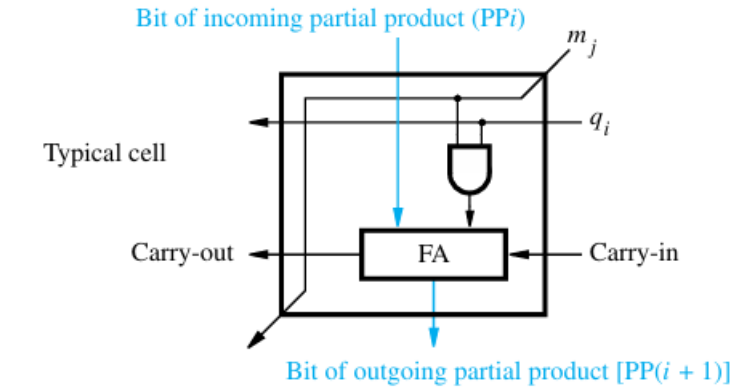
$$c_{16} = G_3^I + P_3^IG_2^I + P_3^IP_2^IG_1^I + P_3^IP_2^IP_1^IG_0^I + P_3^IP_2^IP_1^IP_0^Ic_0$$

Ref. [1]

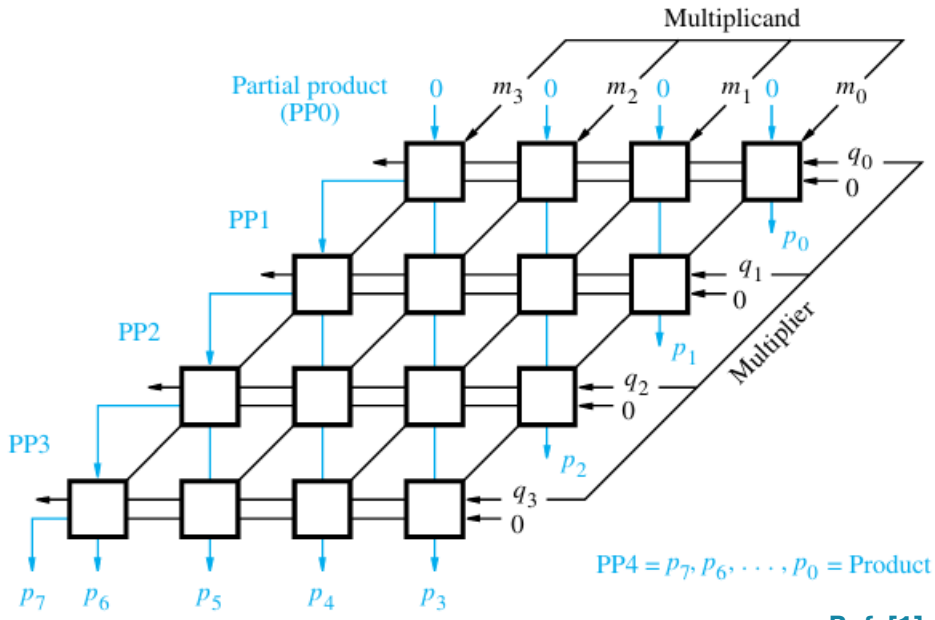
İŞARETSİZ TAM SAYILARIN ÇARPIMI

Manuel çarpım algoritması, **dizi çarpanı**

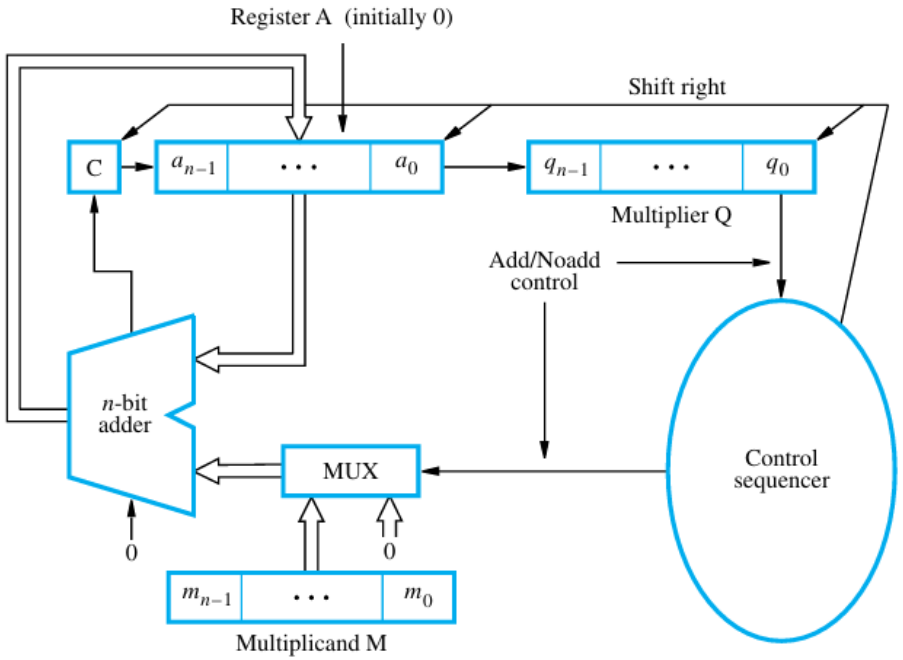
```
          1 1 0 1      (13) Multiplicand M
        × 1 0 1 1      (11) Multiplier Q
        ---------
          1 1 0 1
        1 1 0 1
      0 0 0 0
    1 1 0 1
    ---------------
1 0 0 0 1 1 1 1      (143) Product P
```



Sıralı İkili Çarpan Devresi
Kombinasyonel dizi çarpanı, **32** veya **64** bitlik sayılar gibi pratik boyuttaki sayıları çarpmak için çok sayıda mantık kapısı kullanır. İki ***n*** bitlik sayının çarpımı, tek bir ***n*** bitlik toplayıcı kullanan sıralı bir devrede de gerçekleştirilebilir.



Ref. [1]



Ref. [1]

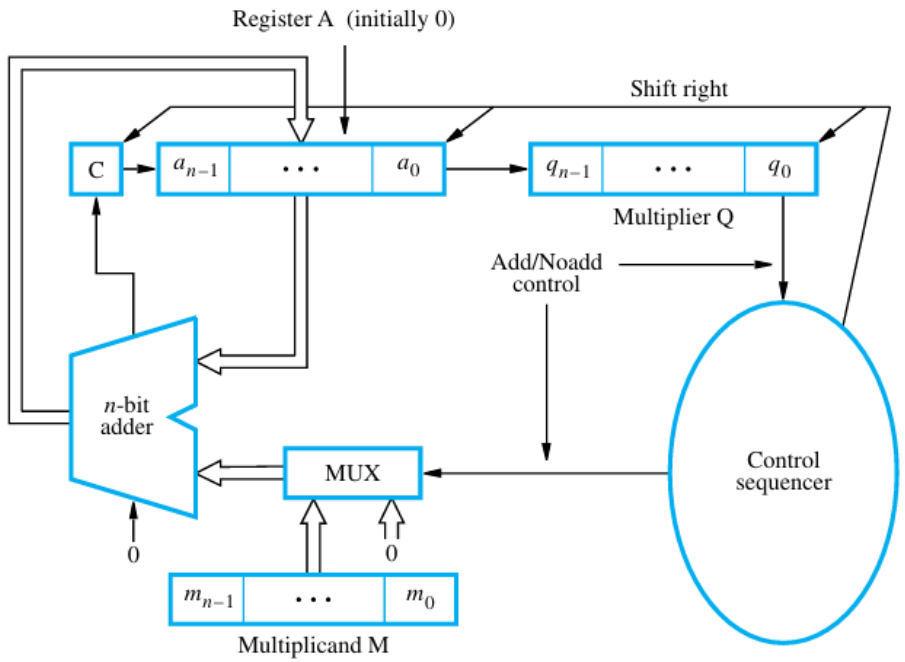
BİLECİK ŞEYH EDEBALİ ÜNİVERSİTESİ

2025-26 GYY DERSİ
BM303
BİLGİSAYAR MİMARİSİ ve ORGANİZASYONU
Prof. Dr. A. Akbaş

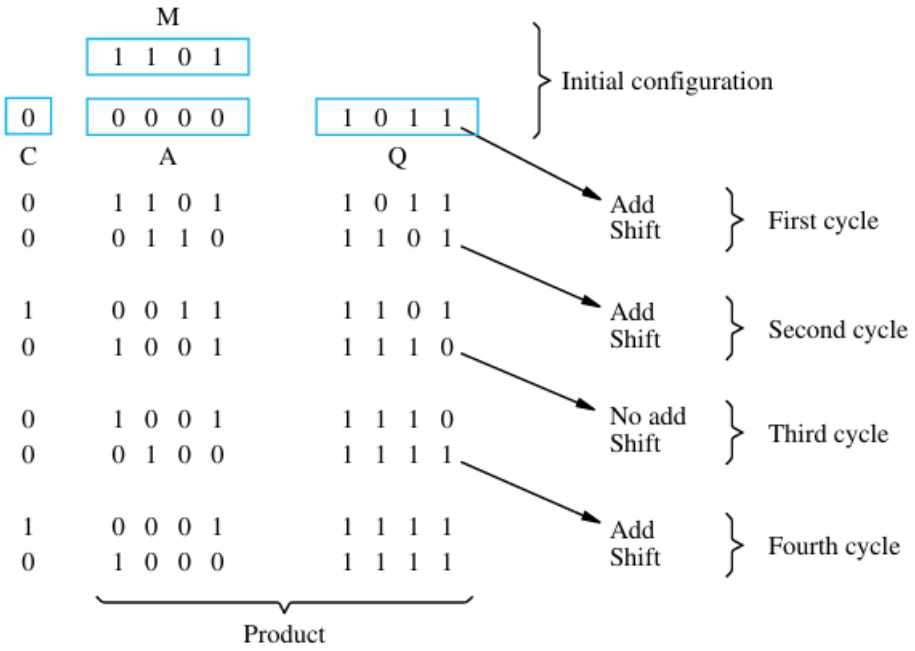
ENG. TERMINOLOGY & ACRONYMS

İŞARETSİZ TAM SAYILARIN ÇARPIMI

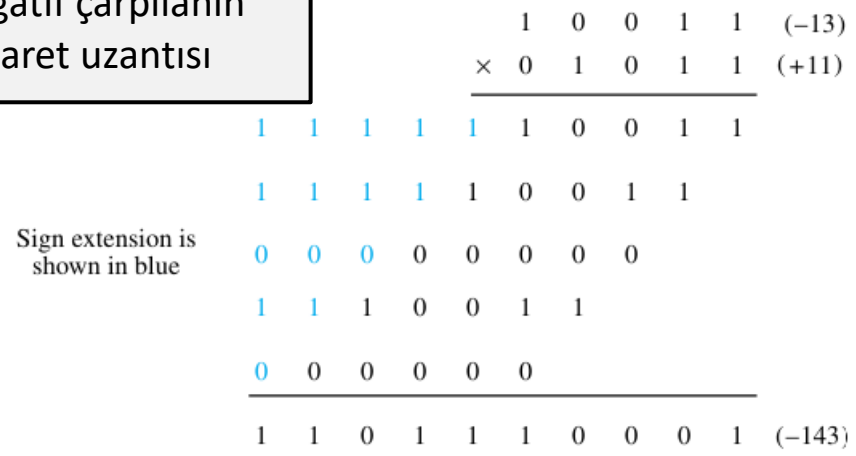
Sıralı İkili Çarpan Devresi
Kombinasyonel dizi çarpanı, 32 veya 64 bitlik sayılar gibi pratik boyuttaki sayıları çarpmak için çok sayıda mantık kapısı kullanır. İki n bitlik sayının çarpımı, tek bir n bitlik toplayıcı kullanan sıralı bir devrede de gerçekleştirilebilir.



Ref. [1]



Negatif çarpılanın işaret uzantısı



Ref. [1]

BİLECİK ŞEYH EDEBALİ ÜNİVERSİTESİ

2025-26 GYY DERSİ
BM303
BİLGİSAYAR MİMARİSİ ve ORGANİZASYONU
Prof. Dr. A. Akbaş

ENG. TERMINOLOGY & ACRONYMS

İŞARETLİ TAM SAYILARIN ÇARPIMI

Negatif bir çarpan için, basit bir çözüm hem çarpanın hem de çarpılanın 2'ye tümleyenini oluşturmak ve pozitif bir çarpan durumunda olduğu gibi devam etmektir. Bu, her iki operantın tümleyeninin sonucun değerini veya işaretini değiştirmediğinden yapılabilir.

Hem negatif hem de pozitif çarpanlar için eşit derecede iyi çalışan bir teknik **Booth algoritması**dır. Booth algoritması 2n bitlik bir sonuç üretir ve hem pozitif hem de negatif 2'nin tümleyeni olan n bitlik operantları ayni şekilde ele alır.

Normal çarpım şeması

```
  0100000   (32)
–
  0000010   (2)
  0011110   (30)
```

```
                                  0  1  0  1  1  0  1
                                  0 +1 +1 +1 +1 +1  0
                                  0  0  0  0  0  0  0
                               0  1  0  1  1  0  1
                            0  1  0  1  1  0  1
                         0  1  0  1  1  0  1
                      0  1  0  1  1  0  1
                   0  0  0  0  0  0  0
                0  0  0  0  0  0  0
             0  0  0  1  0  1  0  1  0  0  0  1  1  0
```

Ref. [1]

Ref. [1]

```
                                  0  1  0  1  1  0  1
                                  0 +1  0  0  0 –1  0
                                  0  0  0  0  0  0  0
0  0  0  0  0  0  0
1  1  1  1  1  1  1              0  1  0  0  1  1   ← 2's complement of the multiplicand
0  0  0  0  0  0  0  0  0  0  0  0
0  0  0  0  0  0  0  0  0  0  0
0  0  0  0  0  0  0  0  0  0
0  0  0  1  0  1  1  0  1
0  0  0  0  0  0  0  0
0  0  0  1  0  1  0  1  0  0  0  1  1  0
```

Booth çarpım şeması

Bir çarpanın **Booth** (kabin) kodlaması

```
0  0  1  0  1  1  0  0  1  1  1  0  1  0  1  1  0  0
0 +1 –1 +1  0 –1  0 +1  0  0 –1 +1 –1 +1  0 –1  0  0
```

```
   0  1  1  0  1   (+13)
× 1  1  0  1  0   (–6)
```

```
   0  1  1  0  1
   0 –1 +1 –1  0
0  0  0  0  0  0  0  0  0  0
1  1  1  1  1  0  0  1  1
0  0  0  0  1  1  0  1
1  1  1  0  0  1  1
0  0  0  0  0  0
1  1  1  0  1  1  0  0  1  0   (–78)
```

Negatif bir çarpan ile **Booth** çarpımıı

İŞARETLİ TAM SAYILARIN ÇARPIMI

X negatif sayısının en soldaki **0**'ının ***k*** bit pozisyonunda olduğu varsayalırsa,

$$X = 11\ldots10x_{k-1}\ldots x_0$$

$$V(X) = -2^{k+1} + x_{k-1} \times 2^{k-1} + \cdots + x_0 \times 2^0$$

$$\begin{array}{rl} & 11\ldots100000\ldots0 \\ + & 00\ldots00x_{k-1}\ldots x_0 \\ \hline X = & 11\ldots10x_{k-1}\ldots x_0 \end{array}$$

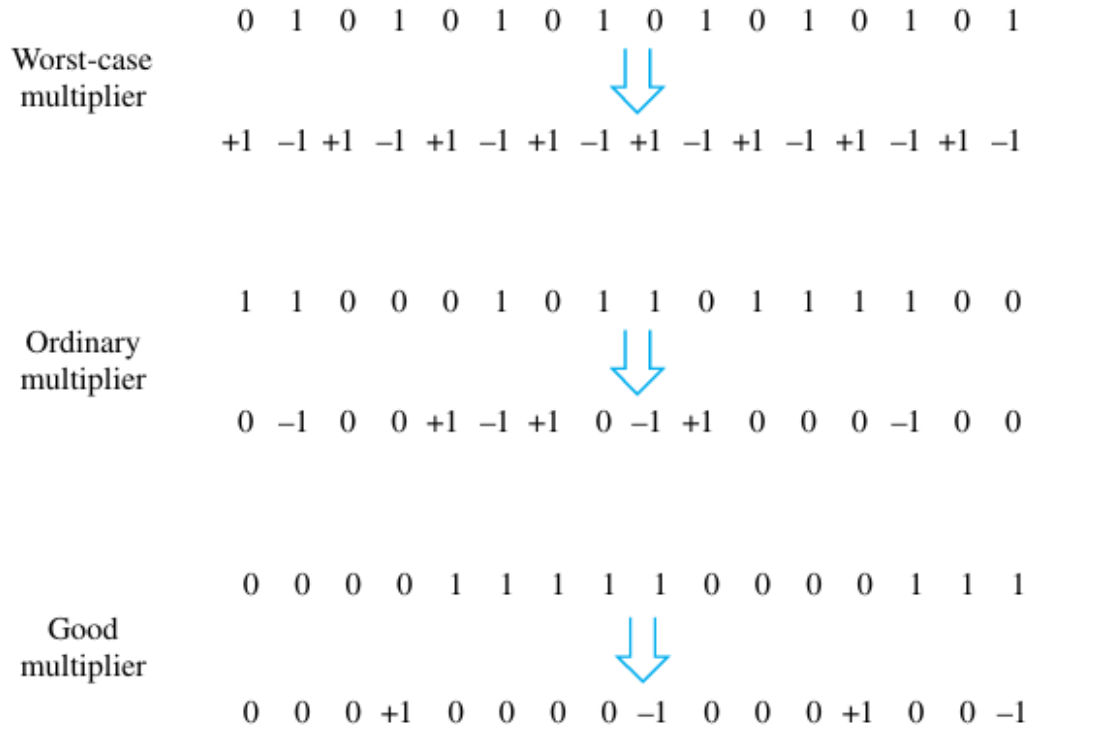
Booth çarpanı kayıt tablosu

Multiplier		Version of multiplicand selected by bit i
Bit i	Bit $i-1$	
0	0	$0 \times M$
0	1	$+1 \times M$
1	0	$-1 \times M$
1	1	$0 \times M$

Ref. [1]

Çarpma işlemini hızlandırmak için 2 yöntem kullanılır. İlki Booth algoritmasından türetilen çarpanın bit çifti yeniden kodlaması, toplam sayısını (çarpan sürümleri) ***n***-bitlik operantlar için ***n***/2'ye indirir. İkincisi, toplayıcıları paralel olarak toplar.

Booth algoritması ile yeniden kodlanan çarpanlar



Ref. [1]

BİLECİK ŞEYH EDEBALİ ÜNİVERSİTESİ

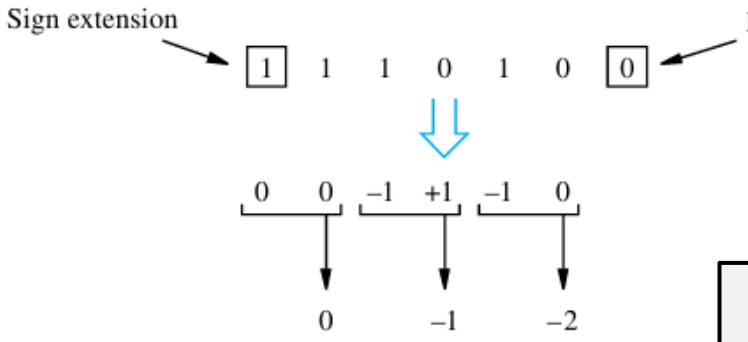
2025-26 GYY DERSİ
BM303
BİLGİSAYAR MİMARİSİ ve ORGANİZASYONU
Prof. Dr. A. Akbaş

ENG. TERMINOLOGY & ACRONYMS

HIZLI ÇARPMA İŞLEMİ

Çarpma işlemini hızlandıran iki yöntemden ilki, toplanması gereken maksimum toplayıcı sayısını (çarpan sürümleri) ***n***-bitlik operantlar için *n*/2 yapar. İkinci yöntemde, toplayıcıları paralel olarak toplanır.

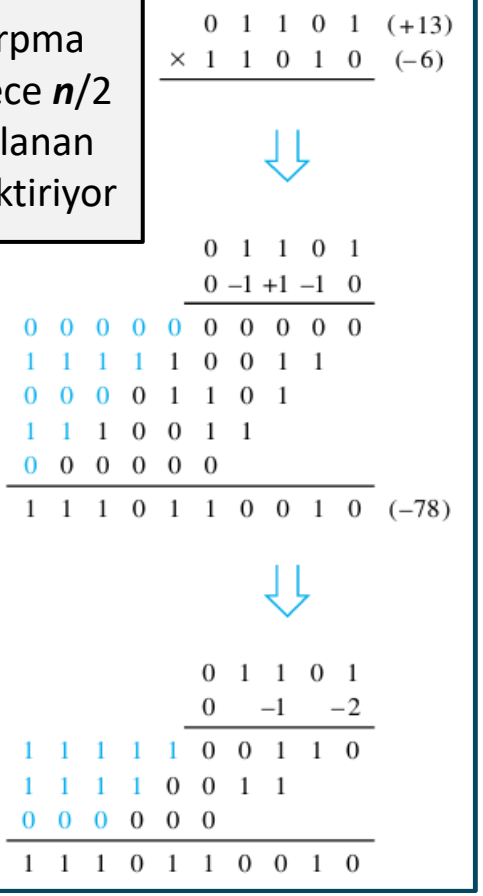
Çarpanın bit-çifti olarak yeniden kodlaması yöntemi, çarpandaki her bit çifti için en fazla bir toplayıcı kullanılır. Bunun için Booth tarafından yeniden kodlanan çarpan bitlerini çiftler halinde gruplanır. (+1 −1) çifti, (0 +1) çiftine eşdeğerdir. Yani, i kaydırma konumundaki çarpan M'nin -1 katını i + 1 konumundaki +1 × M'ye eklemek yerine, i konumundaki +1 × M eklenerek aynı sonuç elde edilir. (+1 0) (0 +2)'ye, (−1 +1) (0 −1)'e eşdeğerdir, vb. Bu nedenle, Booth tarafından yeniden kodlanan çarpan sağdan başlayarak her seferinde iki bit incelenirse, çarpan bitlerinin her çifti için kısmi ürüne en fazla bir çarpım versiyonunun eklenmesini gerektiren bir biçimde yeniden yazılabilir.



Ref. [1]

Booth kaydından türetilen bir bit-çifti kaydı örneği

Çarpma sadece ***n***/2 toplanan gerektiriyor



Çarpılan seçimi için karar tablosu.

Multiplier bit-pair		Multiplier bit on the right	Multiplicand selected at position *i*
i + 1	*i*	*i* − 1	
0	0	0	0 × M
0	0	1	+ 1 × M
0	1	0	+ 1 × M
0	1	1	+ 2 × M
1	0	0	− 2 × M
1	0	1	− 1 × M
1	1	0	− 1 × M
1	1	1	0 × M

BİLECİK ŞEYH EDEBALİ ÜNİVERSİTESİ

2025-26 GYY DERSİ

BM303

BİLGİSAYAR MİMARİSİ ve ORGANİZASYONU

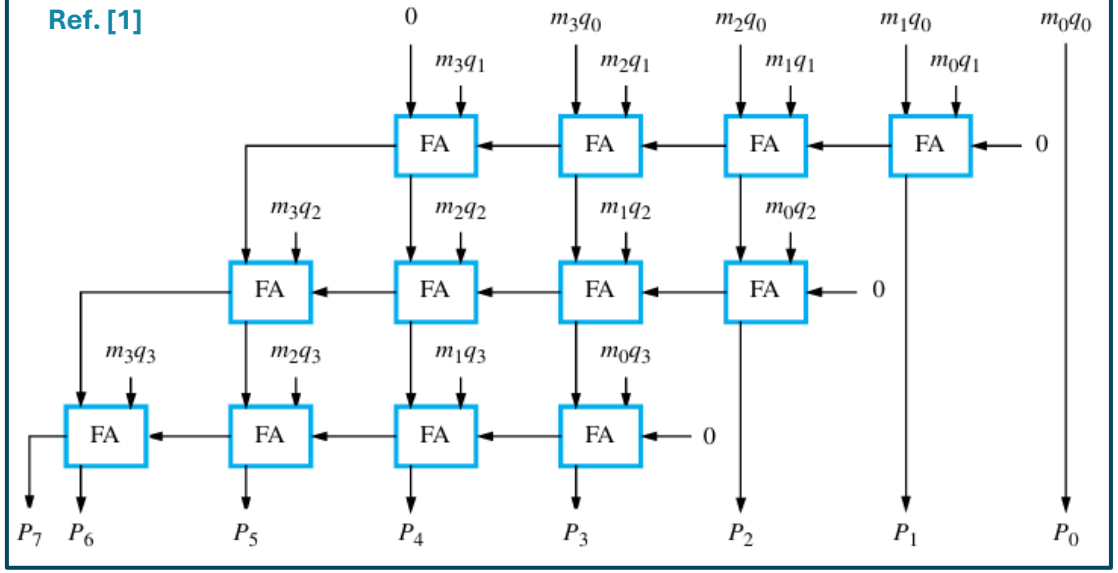
Prof. Dr. A. Akbaş

ENG. TERMINOLOGY & ACRONYMS

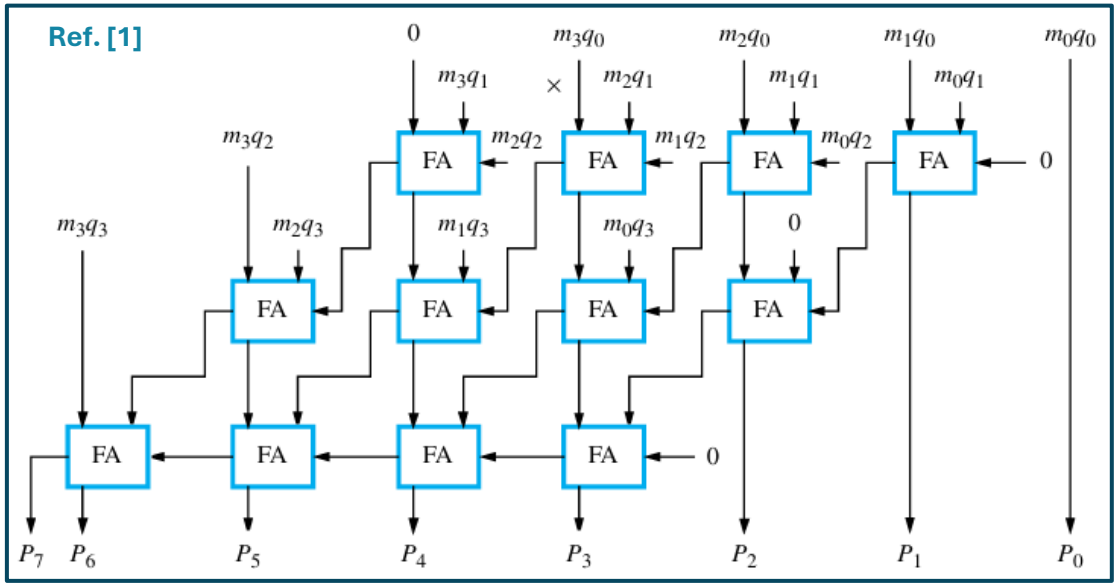
HIZLI ÇARPMA İŞLEMİ : Ara Toplamları Taşı-Kaydet yöntemi

Çarpma, birkaç ara-toplamnın toplanmasını gerektirir. Taşı-kaydet toplama (Carry-Save Addition, CSA) yöntemi çarpma işlemini hızlandırır. Şekildeki 4×4 çarpma ripple-carry dizisinde ilk satır, yalnız dört girdiyi (m3q0, m2q0, m1q0 ve m0q0) üreten VE kapılarından oluşur. Taşımaların satırlar boyunca dalgalanmasına izin vermek yerine, alt Şekildeki caary-save dizisindeki gibi, "kaydedilebilir" ve doğru ağırlıklı konumlarda bir sonraki satıra girebilir. Bu, ilk satırdaki üç tam toplayıcının her biri için bir girişi serbest bırakır. Bu girişler, üçüncü toplama bitleri m2q2, m1q2 ve m0q2'yi sokmak için kullanılabilir. Şimdi, ikinci satırdaki üç tam toplayıcının her birinin iki girişi, ilk satırdaki toplama taşıma çıkışları tarafından beslenir. Üçüncü giriş, dördüncü toplamanın m2q3, m1q3 ve m0q3 bitlerini girmek için kullanılır. Üçüncü ve dördüncü toplamaların yüksek dereceli bitleri m3q2 ve m3q3, ikinci ve üçüncü satırların sol ucunda kalan serbest tam toplayıcı girişlerine girer. Kaydedilen taşıma bitleri ve ikinci satırdaki toplam bitleri artık sonuç bitlerini üretmek için birripple-cary toplayıcısı olan üçüncü satıra eklenir. Carry-save dizisindeki gecikme, ripple-carry dizisindeki gecikmeden daha azdır.

Ref. [1]



Ref. [1]



BİLECİK ŞEYH EDEBALİ ÜNİVERSİTESİ

2025-26 GYY DERSİ

BM303

BİLGİSAYAR MİMARİSİ ve ORGANİZASYONU

Prof. Dr. A. Akbaş

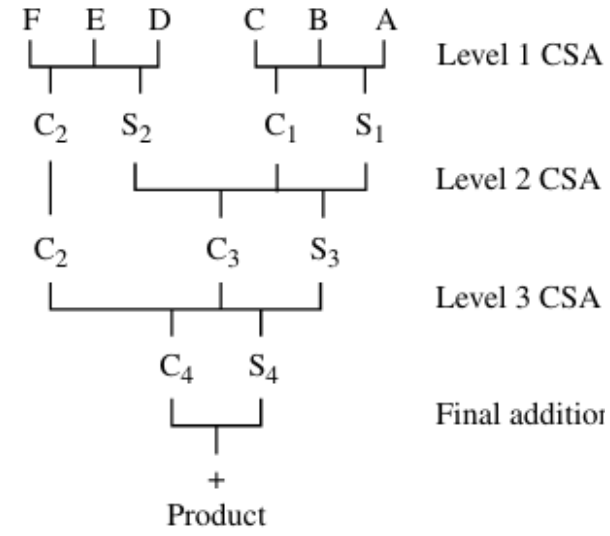
ENG. TERMINOLOGY & ACRONYMS

3-2 İndirgeyici Kullanarak Ara-toplam (Summand) Ekleme Ağacı

Ref. [1]

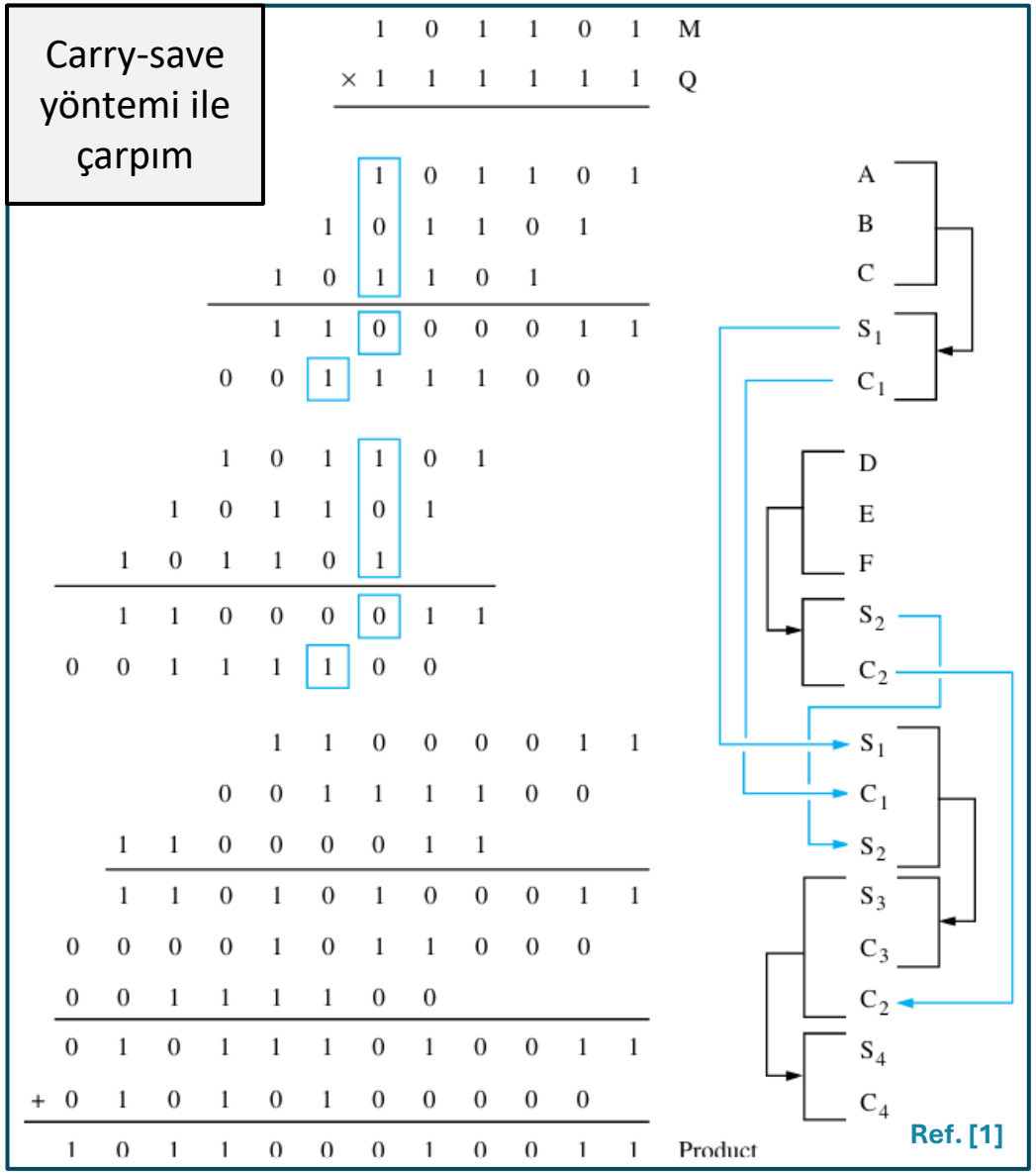
				1	0	1	1	0	1	(45)	M
			×	1	1	1	1	1	1	(63)	Q
			1	0	1	1	0	1		A	
		1	0	1	1	0	1			B	
	1	0	1	1	0	1				C	
1	0	1	1	0	1					D	
1	0	1	1	0	1					E	
1	0	1	1	0	1					F	
1	0	1	1	0	0	0	1	0	0	1	1	(2,835)	Product

Carry-save toplama işlemlerinin sembolik gösterilimi



Ref. [1]

Carry-save yöntemi ile çarpım



Ref. [1]

4-2 İndirgeyici Kullanarak Ara-toplam (Summand) Ekleme Ağacı

• Üç çıktı, s, c ve cout, beş girdinin aritmetik toplamını temsil eder, $w + x + y + z + c_{in} = s + 2(c + c_{out})$

buradaki tüm operatörler aritmetik işlemlerdir.

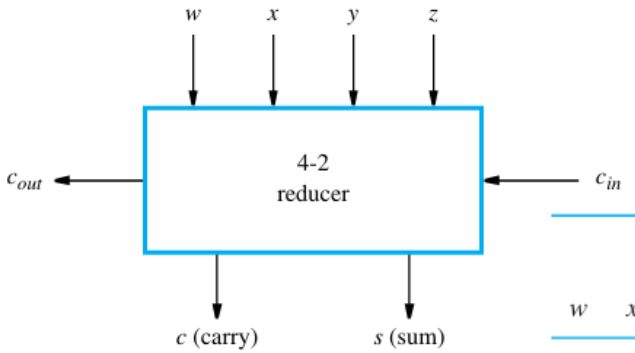
• Çıktı s, olağan toplam değişkenidir; yani s, beş girdi değişkeninin XOR fonksiyonudur.

• Yanal taşıma, cout, cin'den bağımsız olmalıdır. Sadece dört girdi değişkeni w, x, y ve z'nin bir fonksiyonudur.

İki taşıma çıktısını verilen koşulları karşılayacak şekilde belirtmenin farklı yolları vardır. İlk olarak, iki veya daha fazla girdi değişkeni w, x, y ve z 1'e eşit olduğunda yanal taşıma çıktısı cout'u 1 olarak atayalım.

Daha sonra, diğer taşıma çıktısı c, aritmetik koşulu karşılayacak şekilde belirlenir. Bu koşulları sağlayan bir doğruluk tablosu Şekilde verilmiştir. Tablo, farklı bir formda gösterilmiştir. Dört giriş w, x, y ve z ikili sayısal düzende listelenmemiştir. 1 değerine sahip giriş sayısına karşılık gelen gruplar halinde listelenmiştir. Bu, çıktıların verilen koşulları karşılamak için nasıl belirtildiğini görmeyi kolaylaştırır. Mantık kapısı ağı tablodan türetilebilir.

4-2 indirgeyici bloğu



4-2 indirgeme doğruluk tablosu

				c_{in} = 0		c_{in} = 1		
w	x	y	z	c	s	c	s	c_{out}
0	0	0	0	0	0	0	1	0
0	0	0	1	0	1	1	0	0
0	0	1	0	0	1	1	0	0
0	1	0	0	0	1	1	0	0
1	0	0	0	0	1	1	0	0
0	0	1	1	0	0	0	1	1
0	1	0	1	0	0	0	1	1
0	1	1	0	0	0	0	1	1
1	0	0	1	0	0	0	1	1
1	0	1	0	0	0	0	1	1
1	1	0	0	0	0	0	1	1
0	1	1	1	0	1	1	0	1
1	0	1	1	0	1	1	0	1
1	1	0	1	0	1	1	0	1
1	1	1	0	0	1	1	0	1
1	1	1	1	1	0	1	1	1

Ref. [1]

TAMSAYI BÖLME

Daha önce elle yapılan çarpma işleminin bir mantık devresinde nasıl yapıldığını işaretsiz tamsayıların çarpılması örneği ile ele aldık. Şimdi ayni yaklaşımla tamsayı bölmeye bakalım. Şekilde aynı değerlerin ondalık bölme ve ikili bölme örnekleri gösterilmiştir.

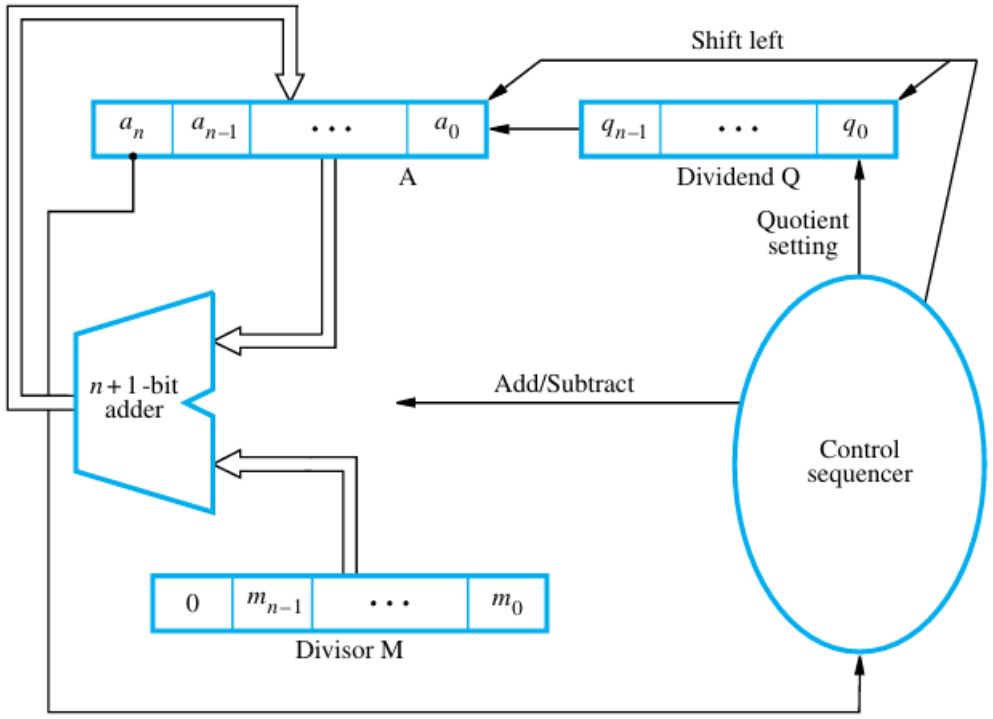
Uzunel (longhand) bölme örnekleri

```
      21
  13 ) 274
       26
       14
       13
        1
```

```
          10101
  1101 ) 100010010
         1101
         10000
          1101
          1110
          1101
             1
```

Ref. [1]

İkili bölme için düzenlenen devre

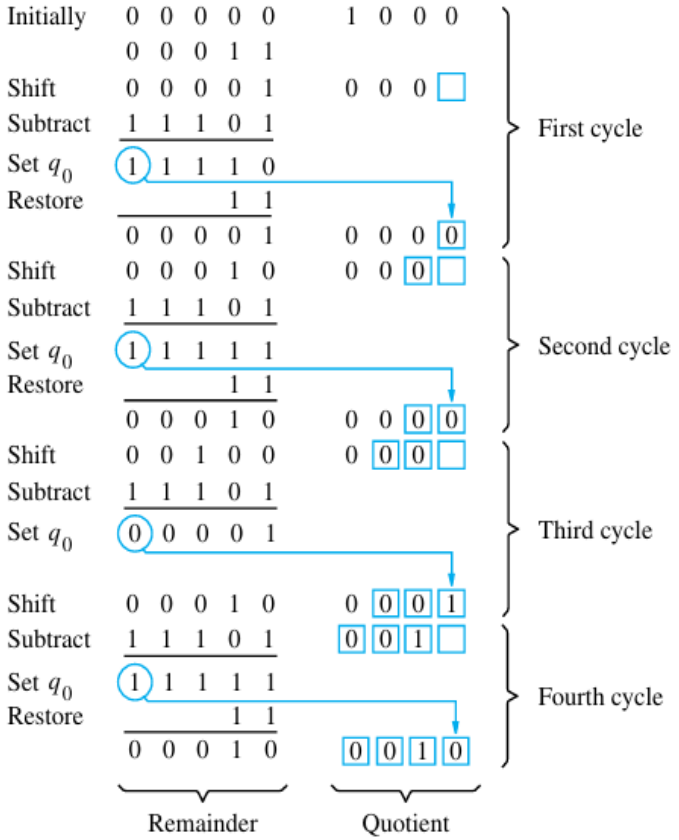


Ref. [1]

TAMSAYI BÖLME

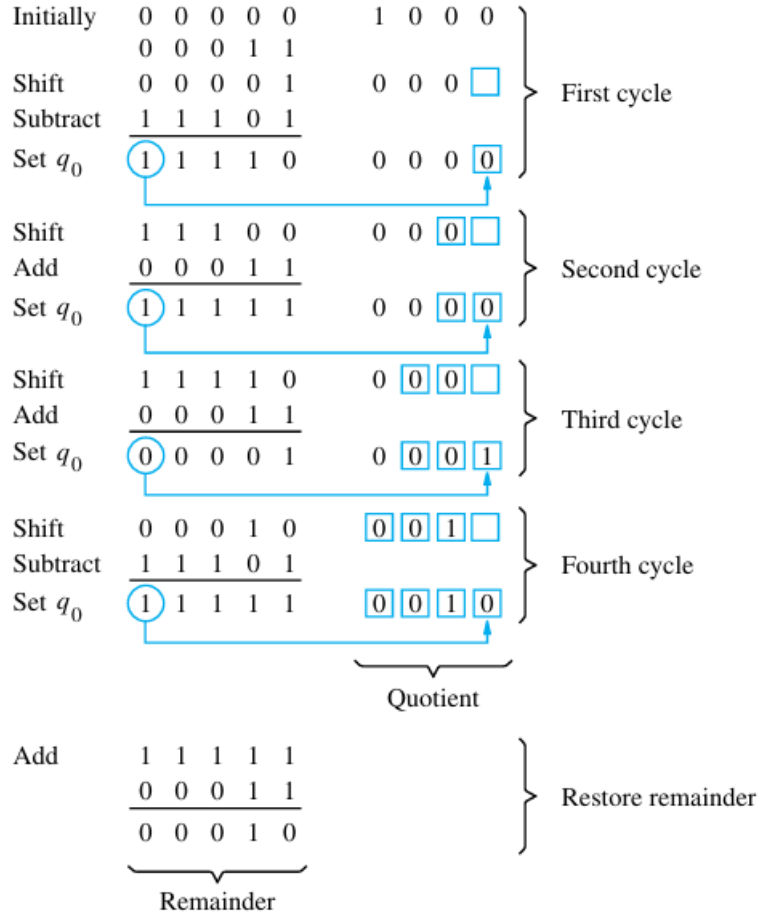
Geri yüklemeli bir bölme örneği

$$\begin{array}{r} 10 \\ 11\overline{)\,1000} \\ \underline{11} \\ 10 \end{array}$$



Ref. [1]

Geri yüklemesiz bir bölme örneği



Ref. [1]

SABİT NOKTALI KESİRLİ SAYILAR

Bilgisayar sistemlerinde veri nesneleri görüntülendiği bağlama göre farklılaşır. Bu kapsamda tam sayı kısmı ***n***-bit, kesir kısmı ***m***-bit olan

$$B = b_{n-1}b_{n-2}\ldots b_0.b_{-1}b_{-2}\ldots b_{-m} \qquad b_i = 0,1 \;;\; 0 \le i \le n-1 \;;\; 1 \le j \le m$$

nesnesi **sabit noktalı işaretsiz bir kesirli sayıyı** temsil eder. Buna göre, böyle bir sayının onluk (desimal) değeri

$$V(B) = b_{n-1} \times 2^{n-1} + b_{n-2} \times 2^{n-2} + \cdots + b_0 \times 2^0 + b_{-1} \times 2^{-1} + b_{-2} \times 2^{-2} + \cdots + b_{-m} \times 2^{-m}$$

şeklinde hesaplanır.

10-tabanlı (desimal) sabit noktalı bir kesirli sayıyı 2-tabanlı (ikili, binary) sayıya dönüştürmek için tam sayı ve kesir kısımları ayrı ayrı ele alınır. Tam sayı kısmının dönüşümü, **2**'ye bölünmeyle başlar. Kalan (**0** veya **1**), ***B***'nin tam sayı kısmının en düşük değerlikli b_0 bitidir (**LSB**). Bölüm tekrar **2**'ye bölünür. Kalan, ***B***'nin bir sonraki b_1 bitidir. Bu işlem, bölümün **0** olduğu, dolayısıyla en yüksek değerlikli bitin (**MSB**) elde edildiği adıma kadar tekrarlanır.

Kesir kısmının dönüşümü, **2** ile çarpılarak başlar. Ondalık noktanın solundaki çarpım (**0** veya **1**), ***B***'nin kesir kısmının b_{-1} bitidir. Çarpımın kesir kısmı tekrar **2** ile çarpılarak ***B***'nin kesir kısmının bir sonraki biti olan b_{-2} üretilir. İşlem, çarpımın kesir kısmı **0** olana veya gerekli hassasiyetin elde edildiği basamağa varılıncaya kadar tekrarlanır.

Şekildeki 10-tabanlı **927.45** sayısının 2-tabanlı sayıya dönüşüm örneğinde görüldüğü gibi, tam sayı kısmının dönüşümü her zaman tam ve bölüm **0**'dır. Ancak ondalık kesir **0.45**'in tam bir ikili eşdeğeri yoktur. Bu gibi durumlarda, ikili kesir istenen bir hassasiyet seviyesine kadar üretilir. Örnekte **2** basamaklık hassasiyet yeterli görülmüştür. Bununla beraber, **0.25** gibi bazı ondalık kesirlerin **0.01** gibi tam bir ikili eşdeğer karşılığı olabilir.

SABİT NOKTALI KESİRLİ ONLU SAYILARIN İKİLİ SAYIYA DÖNÜŞÜMÜ

$$\frac{927}{2} = 463 + \frac{1}{2} \rightarrow 1 \text{ LSB}$$
$$\frac{463}{2} = 231 + \frac{1}{2} \rightarrow 1$$
$$\frac{231}{2} = 115 + \frac{1}{2} \rightarrow 1$$
$$\frac{115}{2} = 57 + \frac{1}{2} \rightarrow 1$$
$$\frac{57}{2} = 28 + \frac{1}{2} \rightarrow 1$$
$$\frac{28}{2} = 14 + \frac{0}{2} \rightarrow 0$$
$$\frac{14}{2} = 7 + \frac{0}{2} \rightarrow 0$$
$$\frac{7}{2} = 3 + \frac{1}{2} \rightarrow 1$$
$$\frac{3}{2} = 1 + \frac{1}{2} \rightarrow 1$$
$$\frac{1}{2} = 0 + \frac{1}{2} \rightarrow 1 \text{ MSB}$$

$$0.45 \times 2 = 0.90 \rightarrow 0 \text{ MSB}$$
$$0.90 \times 2 = 1.80 \rightarrow 1$$
$$0.80 \times 2 = 1.60 \rightarrow 1$$
$$0.60 \times 2 = 1.20 \rightarrow 1$$
$$0.20 \times 2 = 0.40 \rightarrow 0$$
$$0.40 \times 2 = 0.80 \rightarrow 0$$
$$0.80 \times 2 = 1.60 \rightarrow 1 \text{ LSB}$$

$$(927.45)_{10} = (1110011111.0111001\ldots)_2$$

MSB: Most Significant (En Yüksek Değerlikli) Bit
LSB: Least Significant (En Düşük Değerlikli) Bit

Ref. [1]

BİLECİK ŞEYH EDEBALİ ÜNİVERSİTESİ

2025-26 GYY DERSİ

BM303

BİLGİSAYAR MİMARİSİ ve ORGANİZASYONU

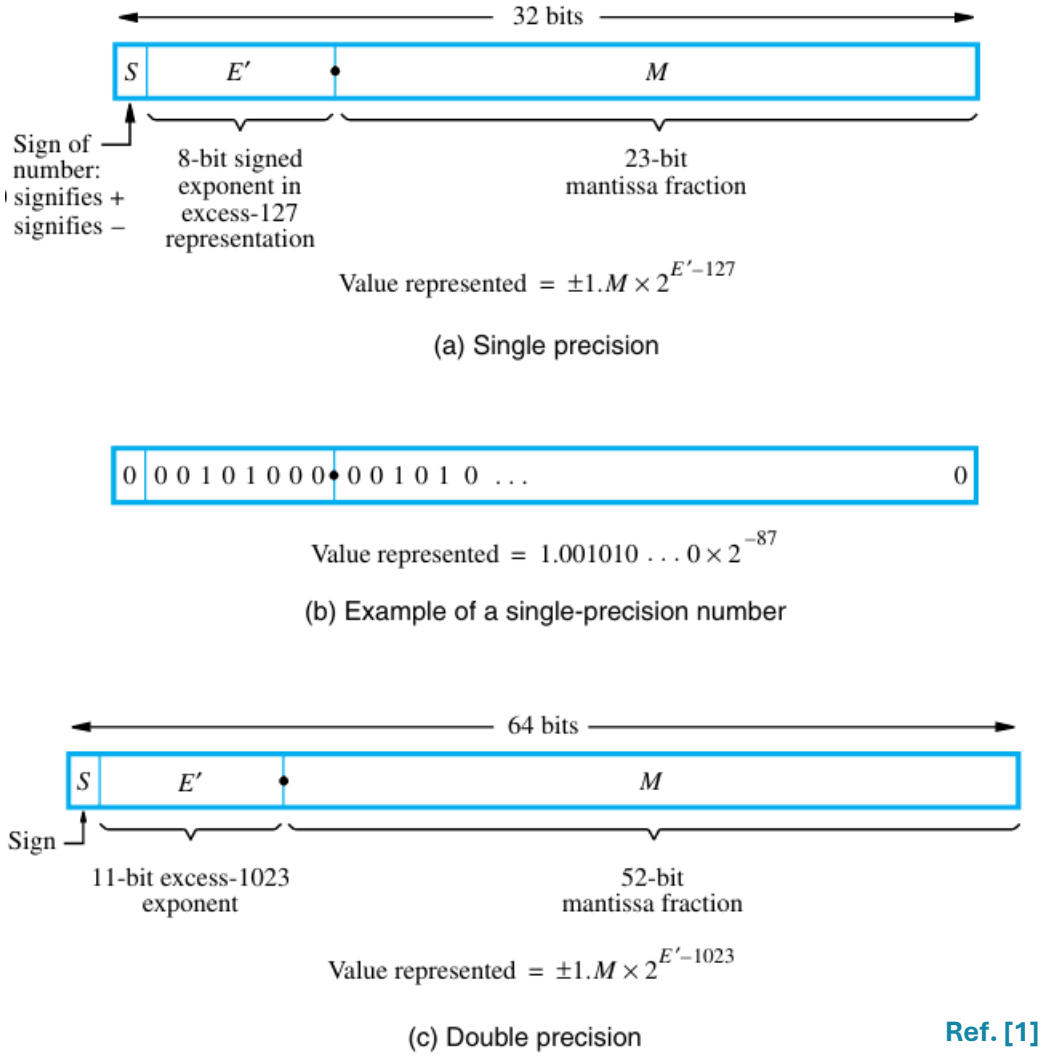
Prof. Dr. A. Akbaş

ENG. TERMINOLOGY & ACRONYMS

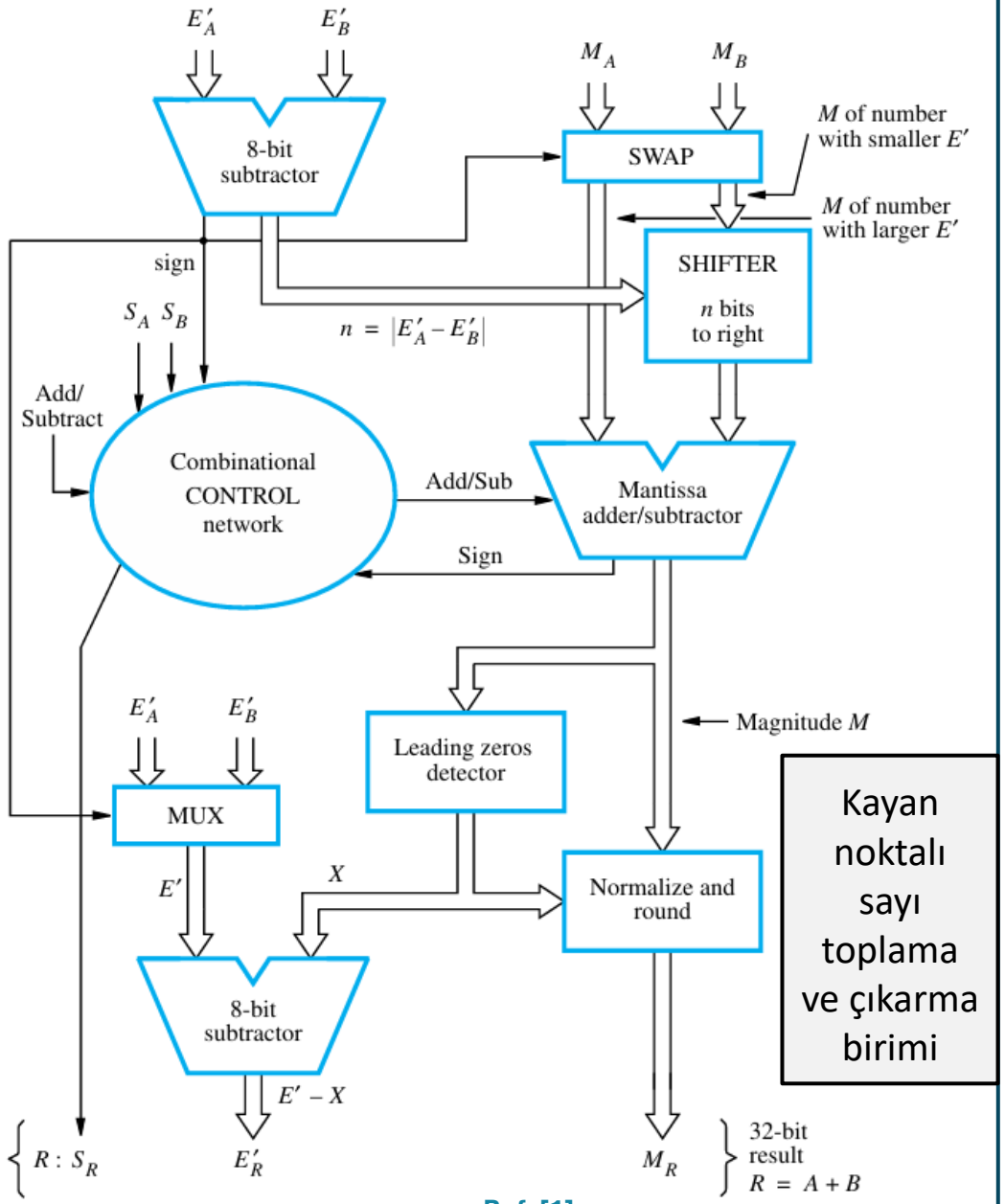
KAYAN NOKTALI SAYI İŞLEMLERİ

IEEE kayan noktalı sayı formatları

$$B = 1.M = 1.b_{-1}b_{-2}\ldots b_{-23}$$



Ref. [1]



Ref. [1]

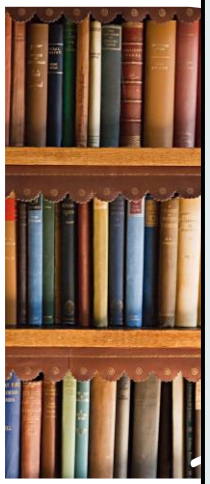
Kayan noktalı sayı toplama ve çıkarma birimi

BİLECİK ŞEYH EDEBALİ ÜNİVERSİTESİ

2025-26 GYY DERSİ
BM303
BİLGİSAYAR MİMARİSİ ve ORGANİZASYONU
Prof. Dr. A. Akbaş

Kaynaklar

REFERANSLAR



1- Hamacher, C., Vranesic, Z., Zaky, S. and Manjikian, N. (2012). *Computer Organization and Embedded Systems.* McGraw Hill, New York. ISBN 978–0–07–338065–0

2- Bryant, R.E. and O'Hallaron, D.R. (2016). *Computer Systems, A Programmer's Perspective.* Pearson Education Limited, Malaysia. ISBN 10: 1-292-10176-8

3- Mono, M.M. (2001). *Bilgisayar Sistem Mimarisi*. Literatür Yayınları, İstanbul. ISBN 975-843-31-5

4- Stallings, W. (2005). *Computer Organisation and Architecture, Designing for Performance.* Pearson, Prentice Hall, NJ. ISBN 0-13-607373-5

5- Abd-El-Barr, M. and El-Rewini, H. (2005). *Fundamentals of Computer Organization and Architecture.* Wiley-Interscience, Hoboken-New Jersy. ISBN 0-471-46741-3

6- Reynolds, C. and Tymann, P. (2008). *Principles of Computer Science.* McGraw-Hill Schaum's Outline Series, New York. 0-07-151037-0

7- https://www.wikipedia.org v.b internet siteleri