



## 5. HAFTA

BLM320

### BİLGİSAYAR MİMARİSİ

Yrd. Doç. Dr. Salih GÖRGÜNOĞLU

[sgorgunoglu@karabuk.edu.tr](mailto:sgorgunoglu@karabuk.edu.tr)

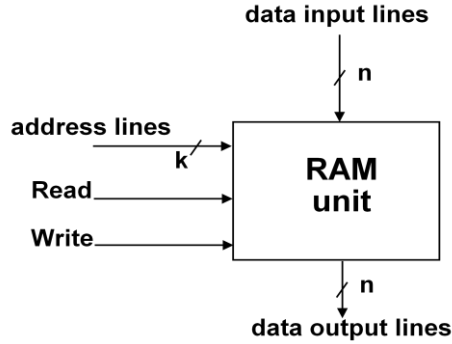
**KBUZEM**

Karabük Üniversitesi

Uzaktan Eğitim Uygulama ve Araştırma Merkezi

## 5. Bellek, Aritmetik, Lojik ve Kaydırma Mikroişlemleri

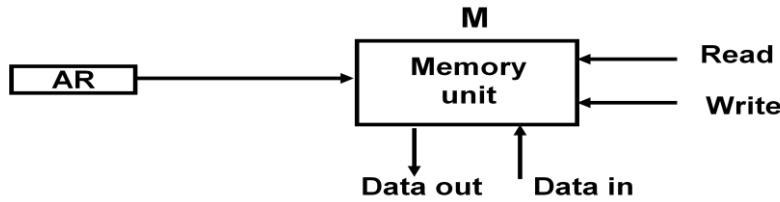
Bellek (RAM) verilerin geçici olarak saklandığı birimlerdir.



Bellek (M) harfi ile gösterilir. Bellek pekçok hücreden oluştuğu için, bellek adresinin belirlenmesi gerekir.

Memory Address Register (MAR, or AR) gibi özel bir saklayıcıda hedef adresi tutarak bir bilgisayar sisteminde belleğe erişim sağlanır.

Belleğe erişildiğinde, MAR içeriği bellek ünitesinin adres hattına gönderilir.



### Bellek okuma

Bellekte bir gözden bir değer okuma ve onu bir saklayıcıya yükleme işlemi RTL olarak gösterimi;

**Read:  $DR \leftarrow M[AR]$**

- Bu işlem aşağıdaki alt işlemlere neden olur:
- AR içeriği bellek adres hattına gönderilir.
- Read (= 1) işareti bellek ünitesine gönderilir.
- Belirlenen adresteki data bellek çıkış hattına gönderilir.
- Bu data DR saklayıcısına yüklenmek üzere ortak yol üzerinden gönderilir.

### Bellek yazma

Bellekte bir alana değer yazma RTL dilinde

**Write:  $M[AR] \leftarrow DR$**

Bu işlem aşağıdaki alt işlemlere neden olur:

- AR içeriği bellek adres hattına gönderilir.
- Write (= 1) işareti bellek ünitesine gönderilir.
- DR saklayıcısındaki data ortak yol üzerinden bellek data giriş hattına gönderilir.
- Bu data bellekte belirlenen adrese yüklenir.

Bilgisayar Sisteminde Mikroişlemleri 4 farklı sınıfa ayırabiliriz.

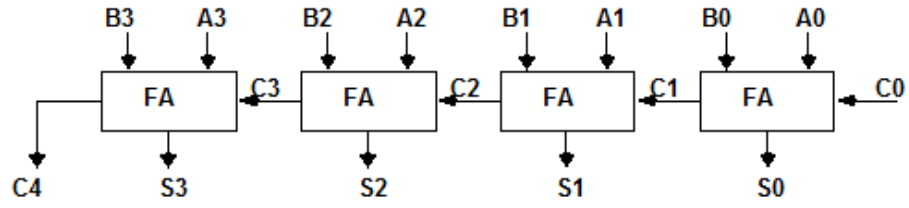
- Register Transfer Mikroişlemleri
- Aritmetik Mikroişlemleri
- Lojik Mikroişlemleri
- Kaydırma Mikroişlemleri

Tipik Aritmetik Mikroişlemler Özet Olarak :

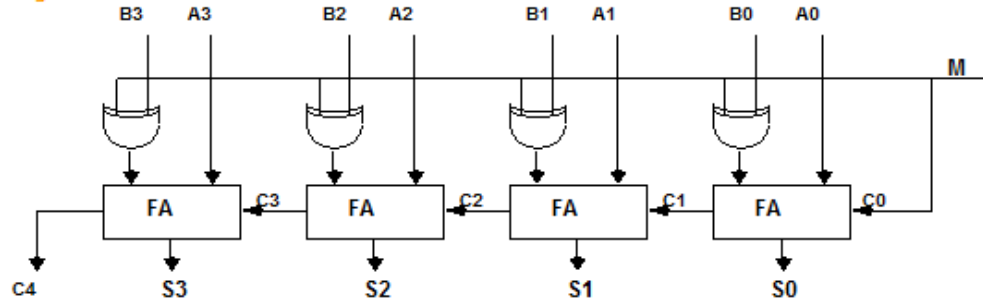
<b><math>R3 \leftarrow R1 + R2</math></b>	<b>Contents of R1 plus R2 transferred to R3</b>
<b><math>R3 \leftarrow R1 - R2</math></b>	<b>Contents of R1 minus R2 transferred to R3</b>
<b><math>R2 \leftarrow R2'</math></b>	<b>Complement the contents of R2</b>
<b><math>R2 \leftarrow R2' + 1</math></b>	<b>2's complement the contents of R2 (negate)</b>
<b><math>R3 \leftarrow R1 + R2' + 1</math></b>	<b>subtraction</b>
<b><math>R1 \leftarrow R1 + 1</math></b>	<b>Increment</b>
<b><math>R1 \leftarrow R1 - 1</math></b>	<b>Decrement</b>

## İkili toplayıcı / çıkarıcı / artırıcı

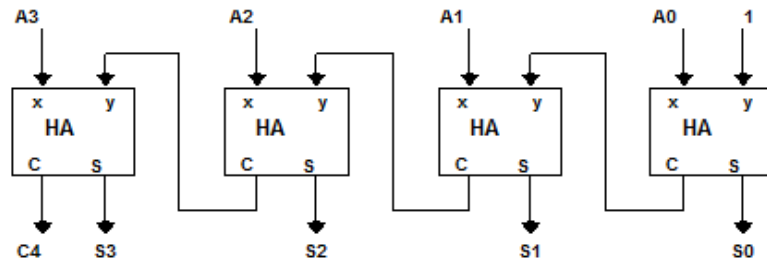
### İkili Toplayıcı



### İkili Toplayıcı- Çıkarıcı



### İkili Artırıcı



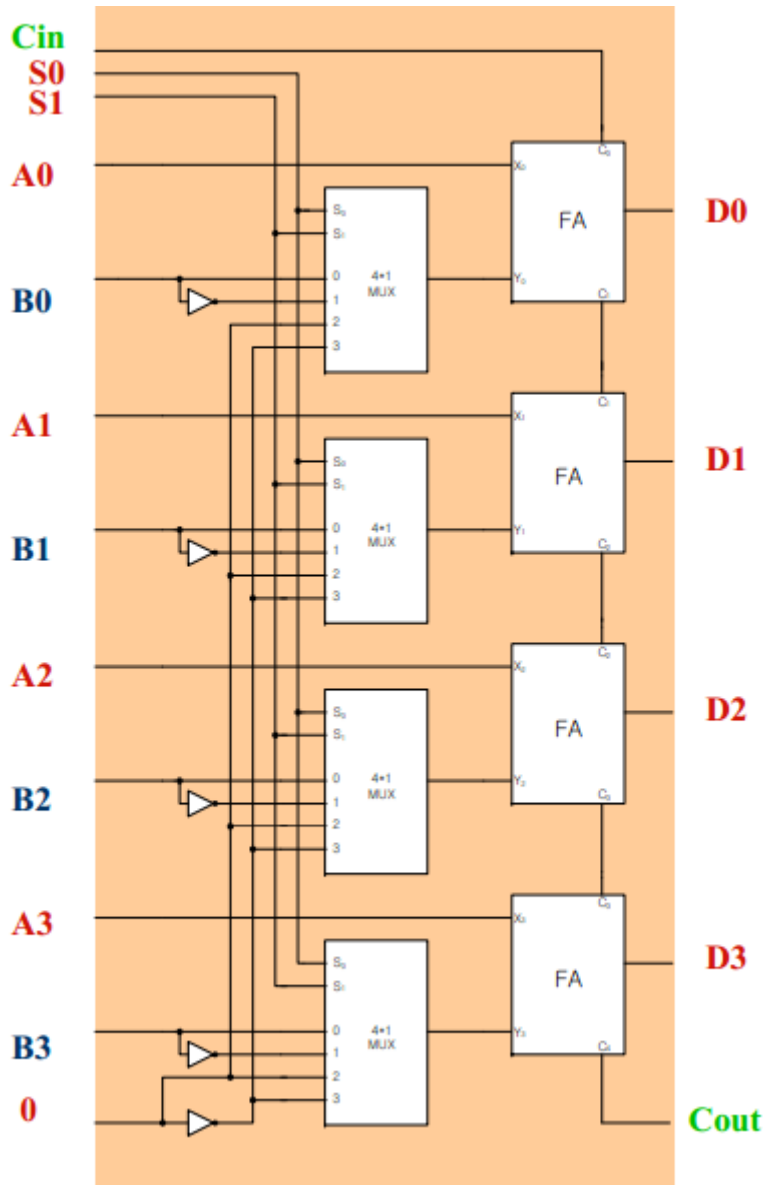
## Aritmetik ünite tasarımı

Select			Input	Output	Microoperat
S1	S0	C <sub>in</sub>	Y	D=A+Y+C <sub>in</sub>	
0	0	0	B	D=A+B	Add
0	0	1	B	D=A+B+1	Add wit
0	1	0	B'	D=A+B'	Subtract with borrow
0	1	1	B'	D=A+B'+1	Subtract
1	0	0	0	D=A	Transfer A
1	0	1	0	D=A+1	İn
1	1	0	1	D=A-1	Decremen
1	1	1	1	D=A	Transfer A

$$A+1111=A-1$$

$$A-1+1=A$$

$$A+B'=A+B'+1-1 \\ = A-B-1$$



## Lojik mikroişlemler

Saklayıcılardaki bit dizinleri üzerinde ikili işlemleri belirler.

- Lojik mikroişlemler bit düzeyinde gerçekleştirilen işlemlerdir.
- İkili data üzerinde bit manipasyonu için yararlıdır.
- Bit değerine bağlı lojik karar verme durumu için yararlıdır.
- İki değişkenle oluşturulabilecek 16 lojik fonksiyon tanımlanabilir.

Bununla birlikte, çoğu sistemler sadece bunlardan 4 tanesini gerçekleştirir. Bunlar:

- AND ( $\wedge$ ), OR ( $\vee$ ), XOR ( $\oplus$ ), Complement/NOT

Diğer fonksiyonlar bunlar cinsinden tanımlanabilirler

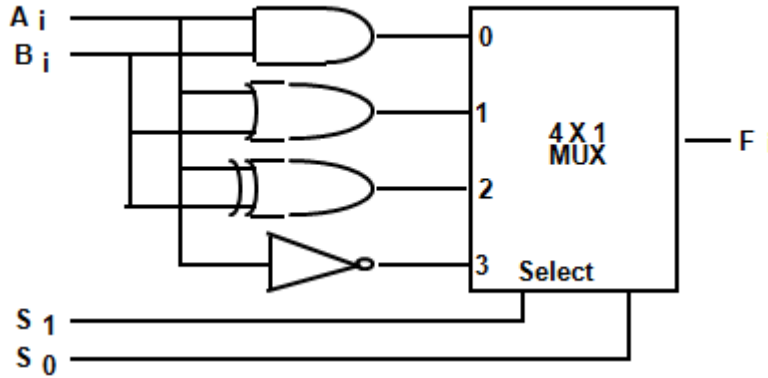
$P+Q: R1 \leftarrow R2 + R3, R4 \leftarrow R5 \vee R6$

Logic OR

Arithmetic ADD

Logic OR

## Lojik mikroişlemlerinin donanımsal gerçekleştirilmesi



Function table

$S_1$	$S_0$	Output	$\mu$ -operation
0	0	$F = A \wedge B$	AND
0	1	$F = A \vee B$	OR
1	0	$F = A \oplus B$	XOR
1	1	$F = A'$	Complement

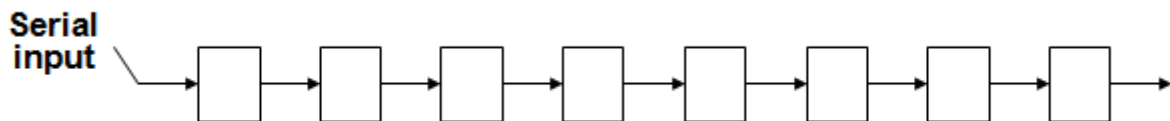
## Kaydırmamikroişlemleri

3 farklı kaydırma tipi bulunmaktadır:

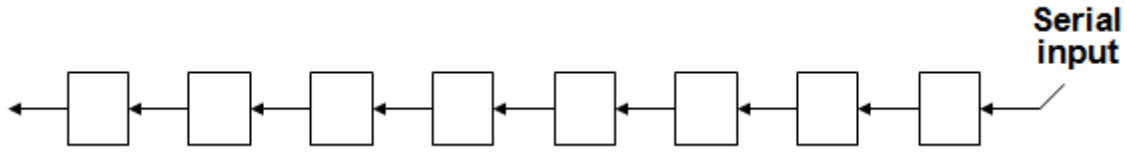
- Lojik kaydırma
- Dairesel kaydırma
- Aritmetik kaydırma

kaydırmatürlerinin aralarındaki temel fark seri giriş bilgisinin farklılığından kaynaklanır

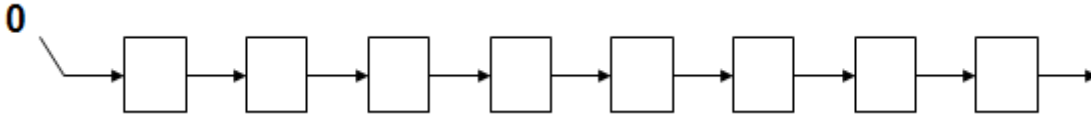
Sağa kaydırma işlemi



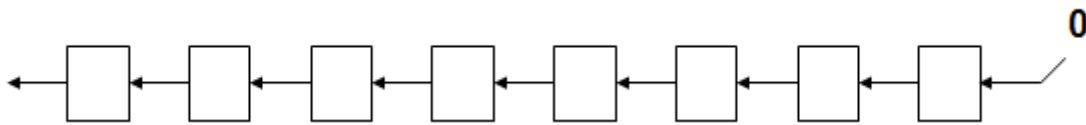
Sola kaydırma işlemi



Lojik kaydırmada seri giriş 0 olur  
Sağa lojik kaydırma işlemi,



Sola lojik kaydırma işlemi



RTL de aşağıdaki notasyon kullanılır:

shl Sola lojik kaydırma

shr Sağa lojik kaydırma

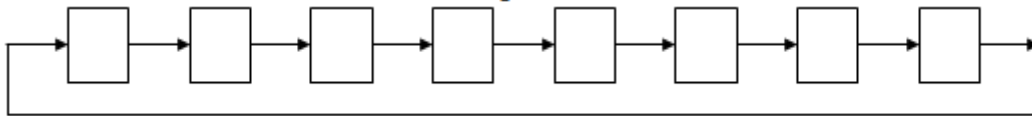
Örnekler:

$R2 \leftarrow shr R2$

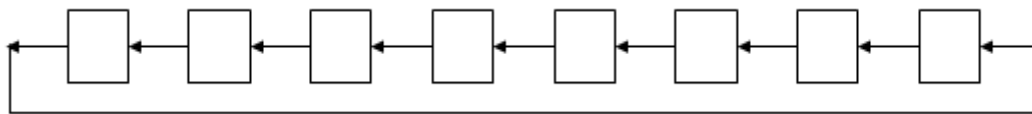
$R3 \leftarrow shl R3$

Dairesel kaydırma

Dairesel sağa kaydırma işlemi:



Dairesel sola kaydırma işlemi:



RTL de, aşağıdaki notasyon kullanılır:

cil Dairesel sola kaydırma

cir Dairesel sağa kaydırma

Örnekler:

$R2 \leftarrow cir R2$

$R3 \leftarrow cil R3$

### Aritmetik Kaydırma

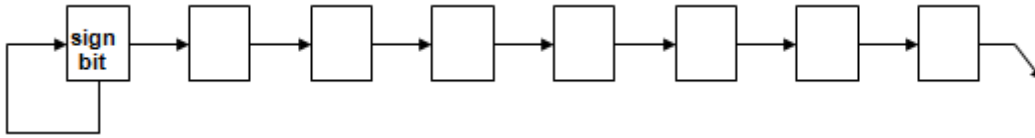
İşaretili sayıların kaydırma için kullanılır (işaretili tamsayı)

Aritmetik sola kaydırma işaretili sayıyı 2 ile çarpmak demektir.

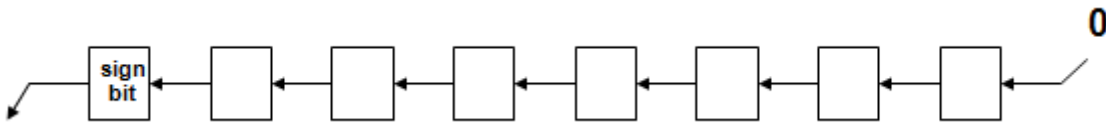
Aritmetik sağa kaydırma işaretili sayıyı 2 ye bölmek demektir.

İşaretili sayılar 2 ile çarpılınca veya bölününce işaretleri değişmeyeceğinden işaret bitleri korunmalıdır.

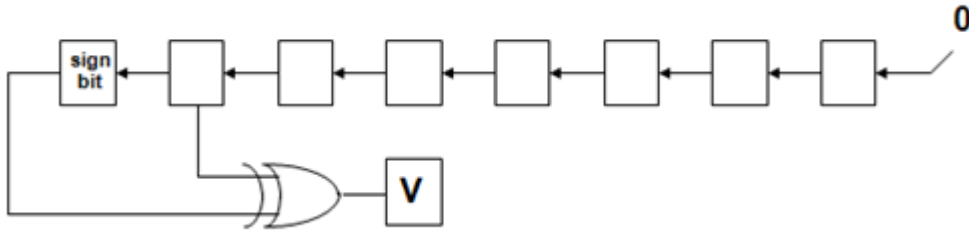
Sağa aritmetik kaydırma işlemi:



Sola aritmetik kaydırma işlemi:



Sola aritmetik kaydırma işleminde sayı aralığının sışına taşma olabileceğinden (işaret değişimi) taşma kontrolu yapılmalıdır.



RTL de, aşağıdaki gösterim kullanılır:

ashl Sola aritmetik kaydırma

ashr Sağa aritmetik kaydırma

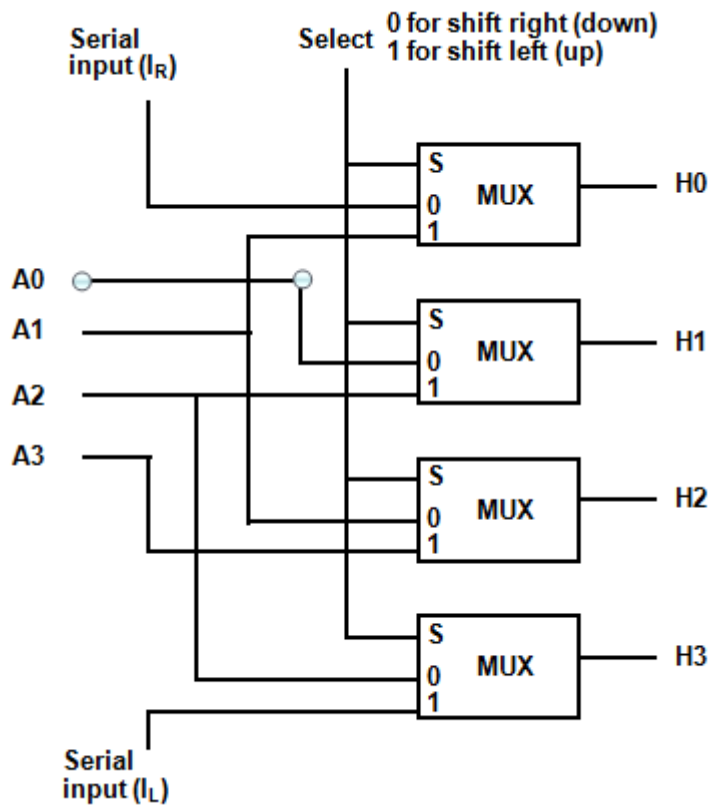
Örnekler:

$R2 \leftarrow \text{ashr } R2$

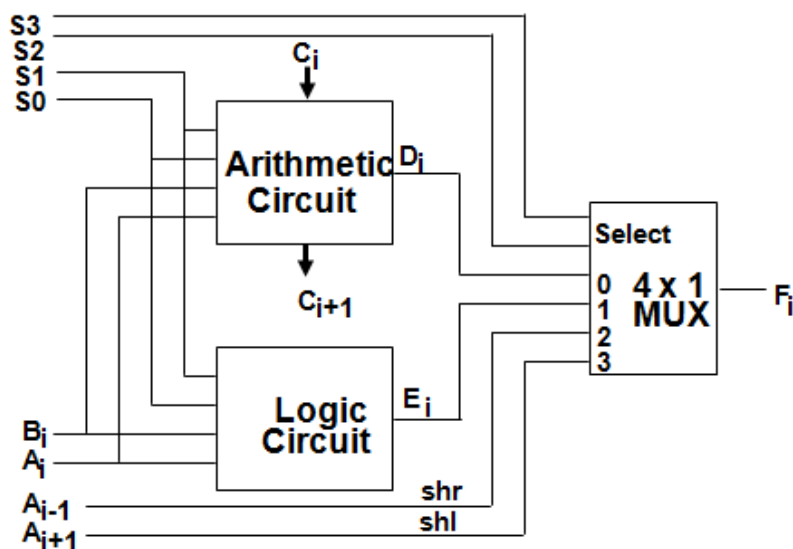
$R3 \leftarrow \text{ashl } R3$



## Kaydırmamikroişlemlerinin donanımsal gerçekleştirilmesi



## ALU Birimi



Operation select					Operation	Function
$S_3$	$S_2$	$S_1$	$S_0$	$C_{in}$		
0	0	0	0	0	$F = A$	Transfer $A$
0	0	0	0	1	$F = A + 1$	Increment $A$
0	0	0	1	0	$F = A + B$	Addition
0	0	0	1	1	$F = A + B + 1$	Add with carry
0	0	1	0	0	$F = A + \overline{B}$	Subtract with borrow
0	0	1	0	1	$F = A + \overline{B} + 1$	Subtraction
0	0	1	1	0	$F = A - 1$	Decrement $A$
0	0	1	1	1	$F = A$	Transfer $A$
0	1	0	0	$\times$	$F = A \wedge B$	AND
0	1	0	1	$\times$	$F = A \vee B$	OR
0	1	1	0	$\times$	$F = A \oplus B$	XOR
0	1	1	1	$\times$	$F = \overline{A}$	Complement $A$
1	0	$\times$	$\times$	$\times$	$F = \text{shr } A$	Shift right $A$ into $F$
1	1	$\times$	$\times$	$\times$	$F = \text{shl } A$	Shift left $A$ into $F$