

# FAHUK BİLEN

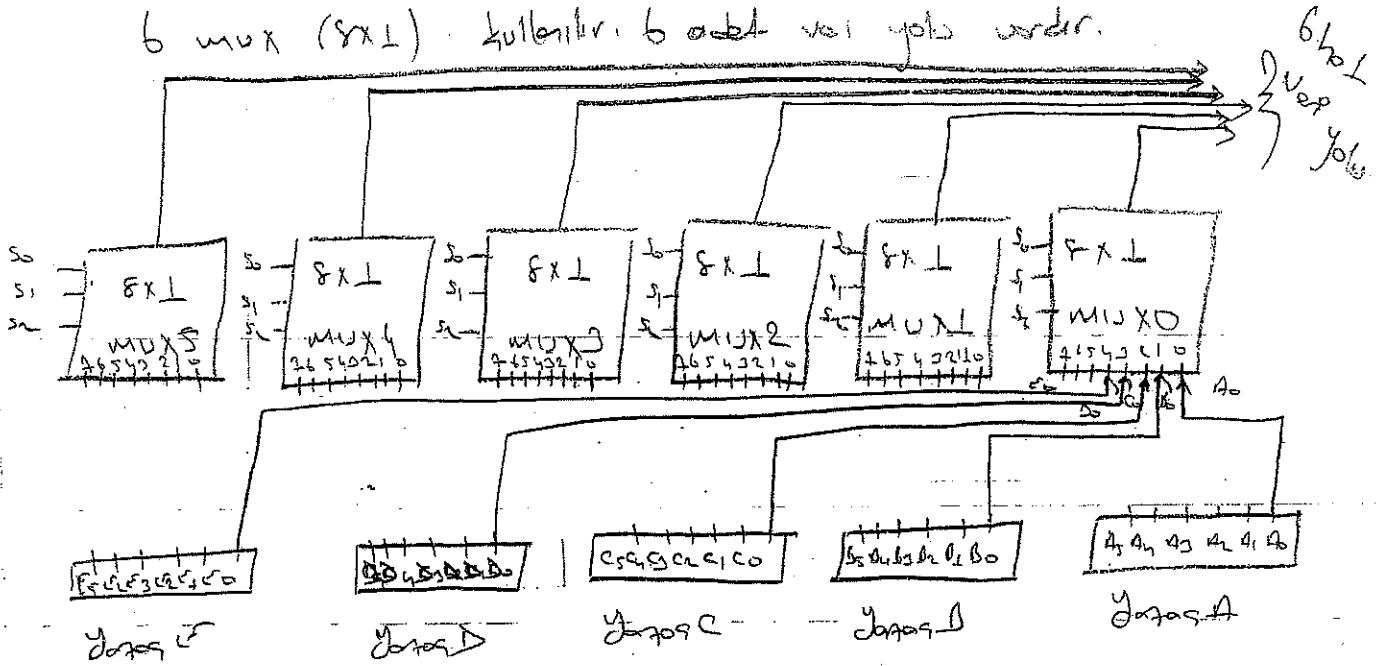
## BİLGİSAYAR MİMARISI

(1)

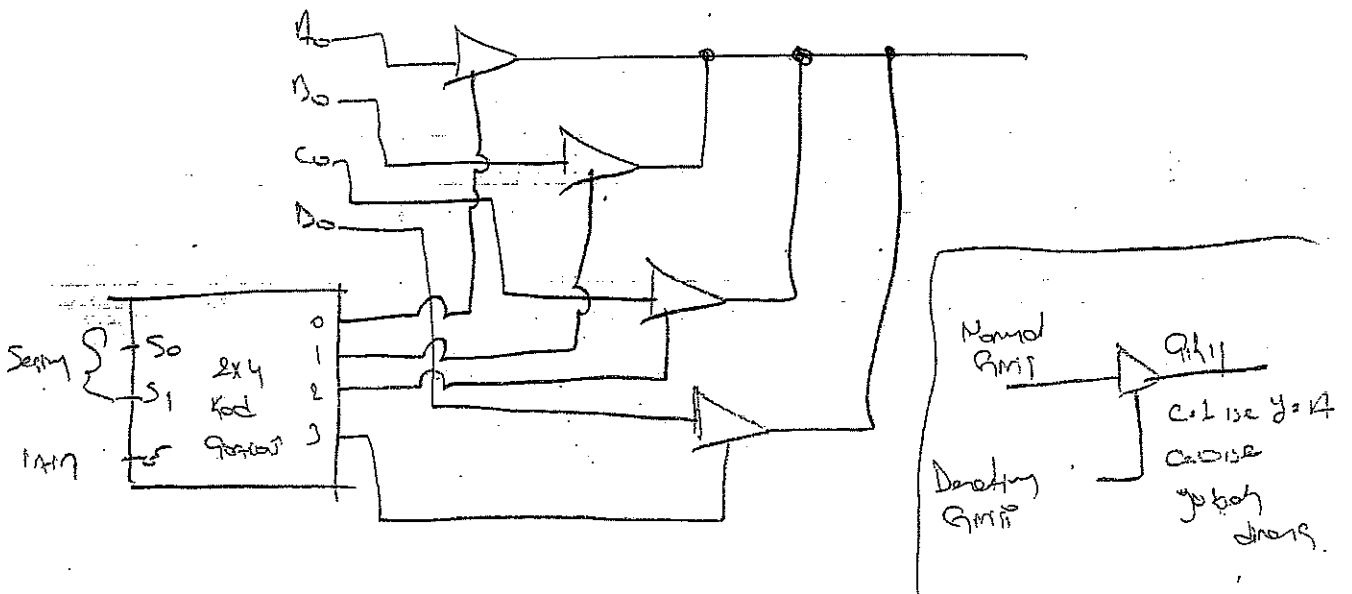
(Register)  
YAAAC AKTARIMI VE MİKRO İŞLEMLERİ (DÖNÜM-4)

SAY = Bir sayısal bilgisayar sisteminde 6 bitlik 5 adet yavaş bulunuyor.  
Bu yavaşları (mux) 2'lik birer aralık var yolu sisteme bağlamak gerekiyor  
bağlantı şemasını çizelim. İşaretleri sisteme her birer mux kullanılır ve  
muxların önceliği ne olmalıdır?

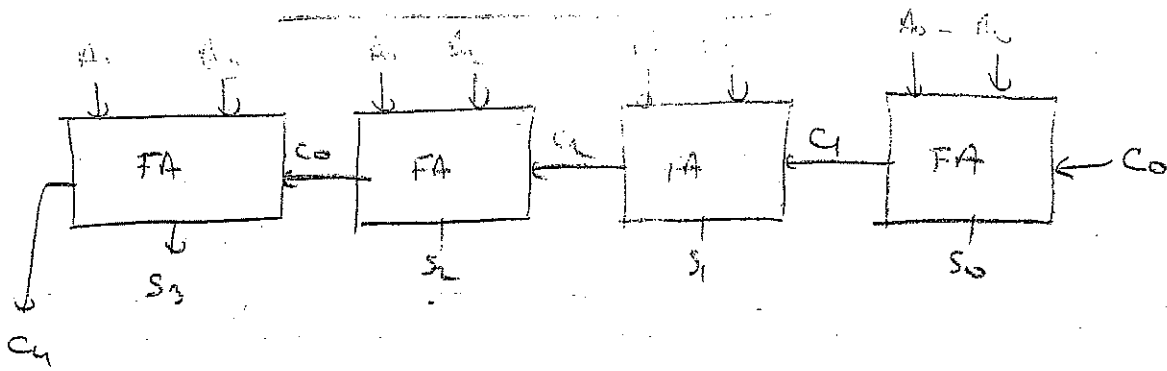
6 mux (8x1) kullanılır. 6 adet var yolu vardır.



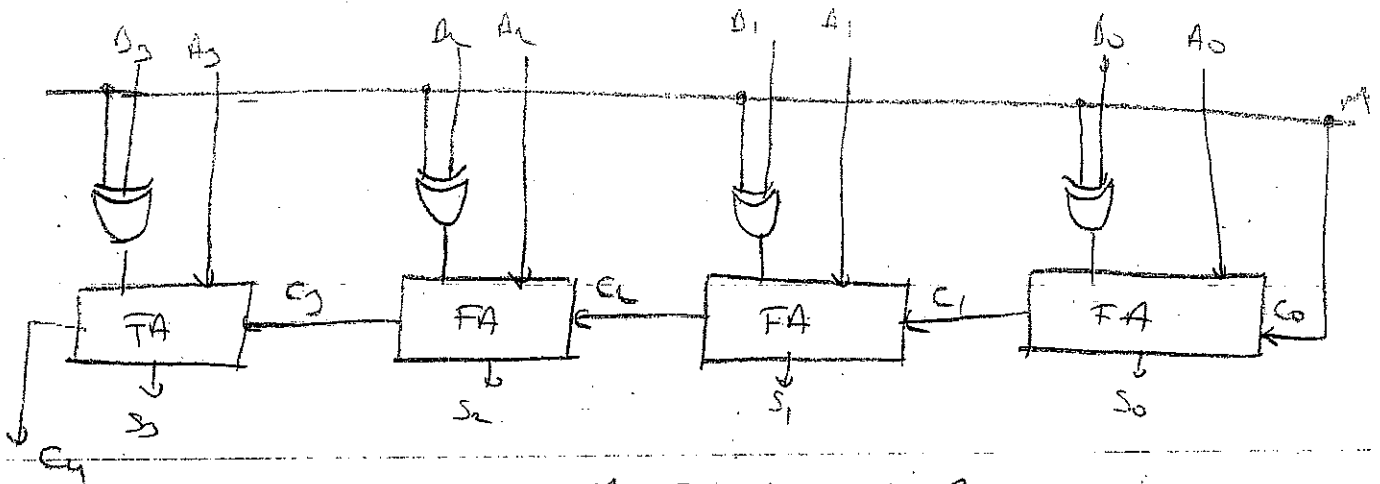
LS DÖNÜMÜSÜ BUFFER



### 4 BIT RYAN'S CARRY LOOK-AROUND



### 4 BIT RYAN'S CARRY LOOK-AROUND DEVICES



A	B	S
0	0	0
0	1	1
1	0	1
1	1	0

$M=0$  ism

$$\underline{\underline{A+B}}$$

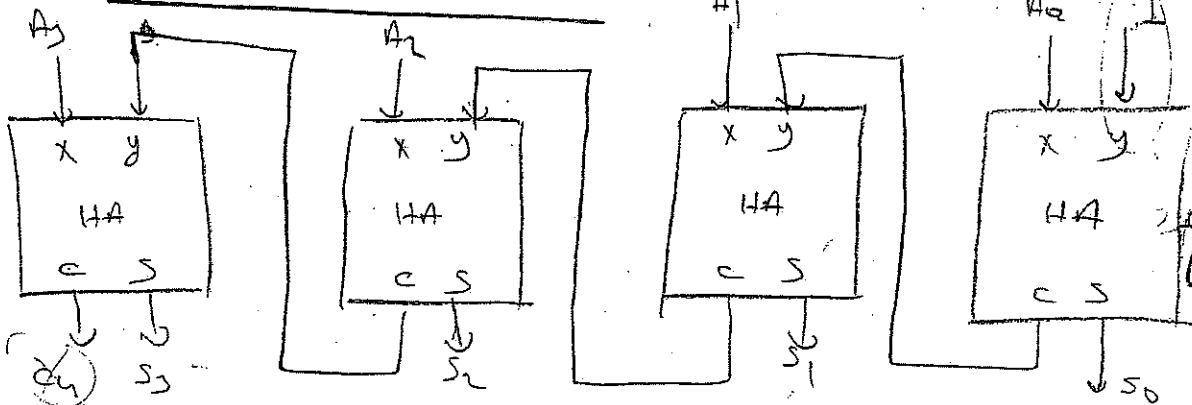
$M=1$  ism

$$\underline{\underline{A+\bar{B}+1}}$$

$$A \oplus B$$

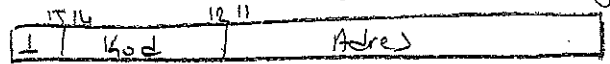
$$\bar{B}+1$$

### 4 BIT RYAN'S CARRY LOOK-AROUND

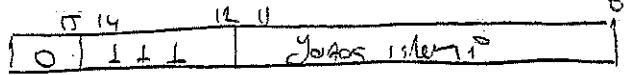


PC (12 bit)  $\Rightarrow$  Program Sayısı  
 AR (12 bit)  $\Rightarrow$  Adres yatağı  
 IR (16 bit)  $\Rightarrow$  komut  
 TR (16 bit)  $\Rightarrow$  Çeşitli yatağı  
 DR (16 bit)  $\Rightarrow$  Veri yatağı  
 AC (16 bit)  $\Rightarrow$  İşlemci yatağı  
 OADR (8 bit)  $\Rightarrow$  Giriş yatağı  
 ENPR (8 bit)  $\Rightarrow$  Çıkış yatağı

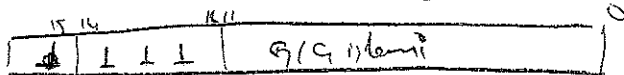
$\Rightarrow$  Bellek Adresleme Bunuşu



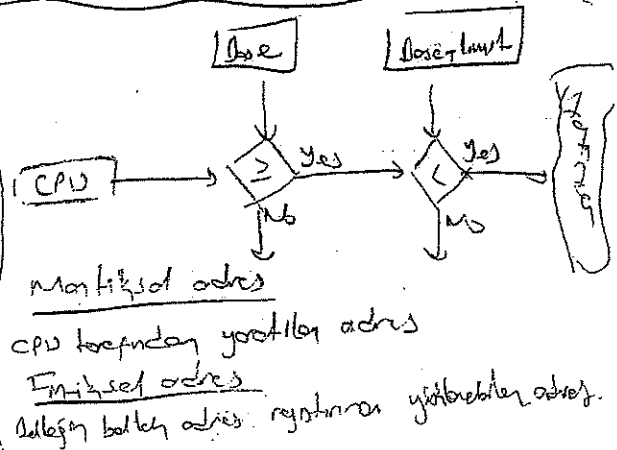
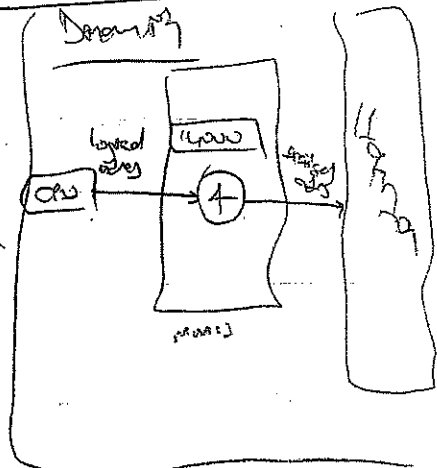
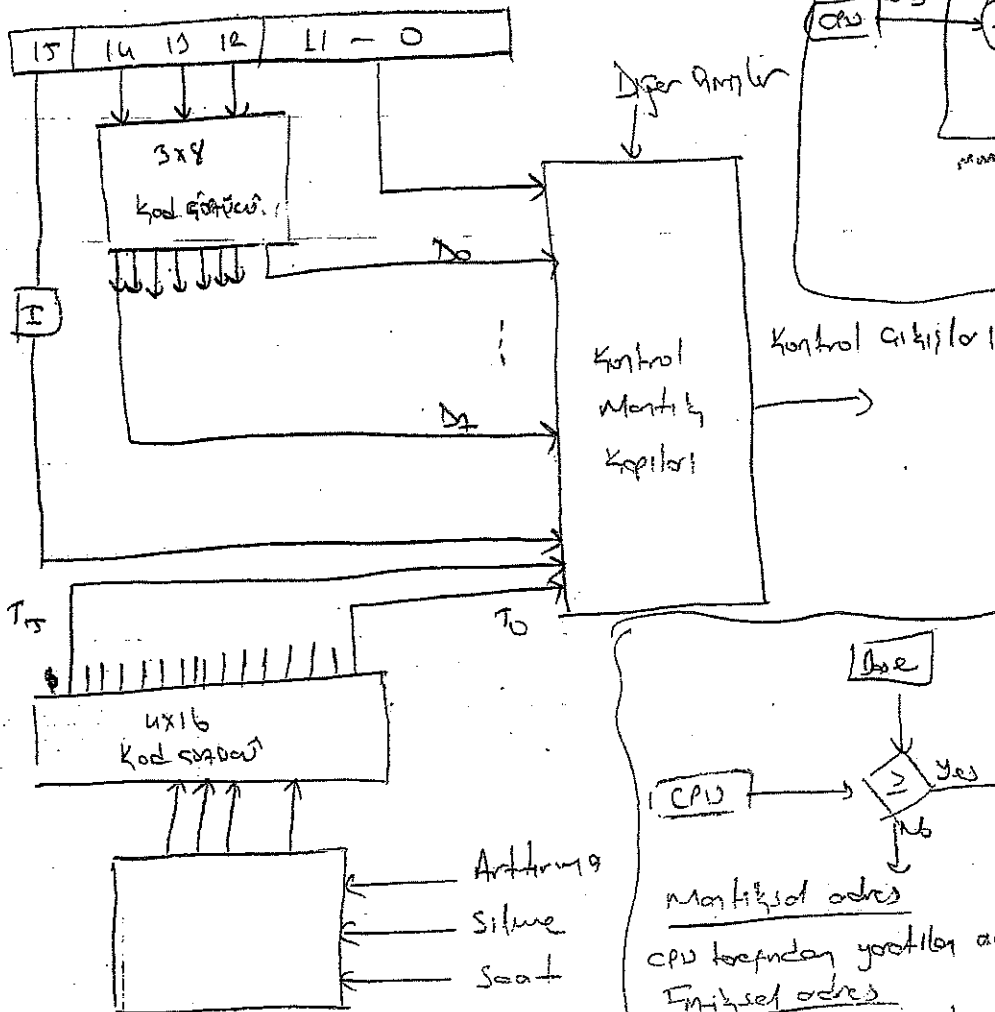
$\Rightarrow$  Veri Adresleme Bunuşu



$\Rightarrow$  Giriş/Çıkış Adresleme Bunuşu



## Temel Bilgisayar Devlet Bunuşu



## Sırasıyla İşleme

=> Tercih bilgisayarlarda her bir işlemin 4 adıma ayrılması

- 1) Buyruğun bellekten alınıp getirilmesi (Fetch)
- 2) Buyruğun kodunun ayrıştırılması (Decode)
- 3) Buyruğun dolaylı adreslenmek ise etkin adresin bulunması
- 4) Buyruğun icrası (Execute)

ALGORTMA KODU GÖR

116  
15122320578

T0:  $AR \leftarrow PC$   
 T1:  $IR \leftarrow M[AR], PC \leftarrow PC + 1$  } Fetch  
 T2:  $D0, \dots, D4 \leftarrow \text{kodu gördü } IR(0-14), AR \leftarrow IR(0-11), IR(15) \Rightarrow \text{Decode}$

### BİRLİKTE ADRESLENEN BUYRUKLAR

1) AND (AC'yi vitle) =

D0T4:  $DR \leftarrow M[AR]$   
 D0T5:  $AC \leftarrow AC \wedge DR, SC \leftarrow 0$

2) ADD (AC'ye ekle) =

D0T4:  $DR \leftarrow M[AR]$   
 D0T5:  $AC \leftarrow AC + DR, PC \leftarrow PC + 1, SC \leftarrow 0$

3) LDA (AC'ye yükle) =

D0T4:  $DR \leftarrow M[AR]$   
 D0T5:  $AC \leftarrow DR, SC \leftarrow 0$

4) STA (AC'yi belleğe atar) =

D0T4:  $M[AR] \leftarrow AC, SC \leftarrow 0$

5) BUN (Sartlı olarak atlar) =

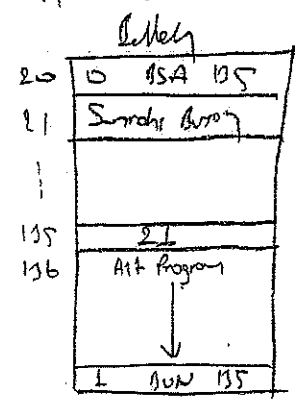
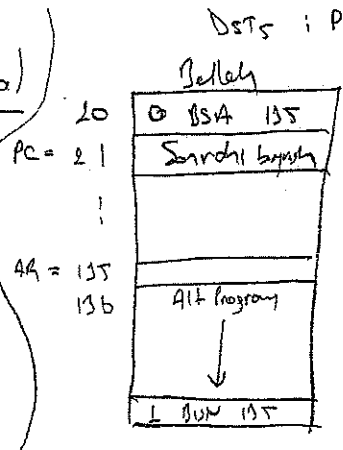
D0T4:  $PC \leftarrow AR, SC \leftarrow 0$

6) JSA (Dolan ve Geri Dönüş Adresini Sola) =

D0T4:  $M[AR] \leftarrow PC, AR \leftarrow AR + 1$   
 D0T5:  $PC \leftarrow AR, SC \leftarrow 0$

7) JSE (Arttır ve Eğer Sıfır ise atla) =

D0T4:  $DR \leftarrow M[AR]$   
 D0T5:  $DR \leftarrow DR + 1$   
 D0T6:  $M[AR] \leftarrow DR$   
 eğer  $(DR = 0)$  ise  $(PC \leftarrow PC + 1), SC \leftarrow 0$





# Assembly Language

1) Belki Adresleri Bayrak=

15 12 11 10 9 8 7 6 5 4 3 2 1 0

2) Yorum Adresleri Bayrak=

15 12 11 10 9 8 7 6 5 4 3 2 1 0

3) Cihazlar Adresleri Bayrak=

15 12 11 10 9 8 7 6 5 4 3 2 1 0

## Binary

Location

Instruction Code

0	0000 0000 0000 0100
1	0001 0000 0000 0101
10	0011 0000 0000 0110
11	0111 0000 0000 0001
100	0000 0000 0101 0011
101	1111 1111 1110 1001
110	0000 0000 0000 0000

## Hex 4

Location

Instruction

000	2004
001	1005
002	3006
003	7001
004	0053
005	FF09
006	0000

## Symbolic

Location

Instruction

000	LDA 004
001	ADD 005
002	STA 006
003	HCT
004	0053
005	FF09
006	0000

## Assembly

ORG 0

LDA 4

ADD 3

STA C

HCT

A, DEC 83

B, DEC -23

C, DEC 0

END

Deretim Birimi = Bir bilgisayarda deretim biriminin görevi mikro işlemlerin sırasını koşturmadır.

=> Bir deretim biriminin binary değişkenleri bellekte saklanır, Mikro programlaması deretim olarak adlandırılır.

=> Deretim belleğindeki her ifade bir mikro buyruk iserir.

=> Bir mikro buyruk bir veya daha fazla mikro işlem kerr.

=> Mikro buyrukların sırası da bir mikro program oluşturur.

=> Deretim birimi oluşturulmuş, mikro buyrukların işleme göre değişim.

=> Deretim belleği kolay okunabilir.

=> Daha ileri bir uygulamada ise deretim mikro programı bir HD üzerinde birleştirilir. Buna deretmiş mikro program denir.

=> Mikro programlaması, deretim birimi iseren bilgisayarlarda iki ayrı bellek vardır.

1) Ana Bellek = Kullanıcı programlarının ve verilerinin bulunduğu bellektir.

2) Deretim Belleği = Değişkenler, sabit bir mikrogramın tutulduğu bellektir.

HARİTALAMA = Deretim belleğine yerleştirilmesi olan ustis (yardım) buyruk kod bitlerinin bir adres dönüşümü ile bulunması, haritalama olarak isimlendirilir. Haritalama işlemi, bir buyruk kodunu deretim belleği adresine dönüştüren bir durumdur.

=> Özetle, Deretim belleğinde adres sıralama işlemi:

- 1) Deretim adres yapısında artırma
- 2) Deretim bitlerinin konumuna bağlı olarak sırtı, veya sırtıya dalama
- 3) Buyruk bitlerinin bir adrese deretim belleği için haritalama işlemi
- 4) Alt programa giriş ve çıkış için bağlantılar.

3) MİKRO BUYRUK KOD BİTİMLERİ (20 Bit)

3	3	3	2	2	4
F1	F2	F3	CD	B4	AD

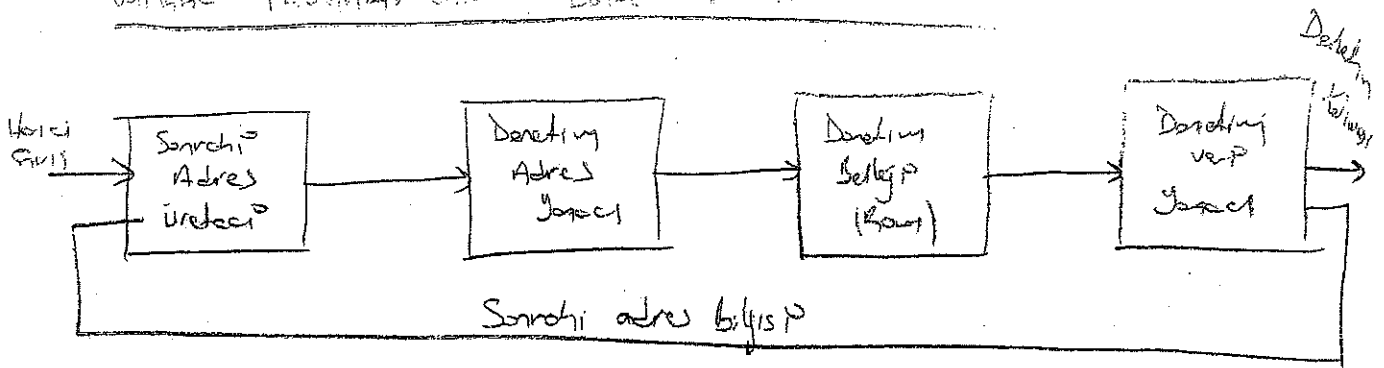
F1, F2, F3 = Mikro işlem denetimi

CD = Dalama işi sırt

B4 = Dalama alanı

AD = Adres alanı

# VIRAC PROGRAMIRANJE DOKU I Osnovne instrukcije



<u>F1</u>	<u>mitroslav</u>	<u>simbol</u>
000	None	NOP
001	$AC \leftarrow DR$	ADD
010	$AC \leftarrow 0$	CLAC
011	$AC \leftarrow AC + 1$	INCAC
100	$AC \leftarrow DR$	DATA
101	$AR \leftarrow DR(0-10)$	DATA
110	$AR \leftarrow PC$	PCDATA
111	$MCAR \leftarrow DR$	WRITE

<u>F2</u>		
000	none	NOP
001	$AC \leftarrow AC + DR$	SUB
010	$AC \leftarrow AC + DR$	OR
011	$AC \leftarrow AC + DR$	AND
100	$DR \leftarrow MCAR$	READ
101	$DR \leftarrow AC$	ACTDR
110	$DR \leftarrow DR + 1$	INCDR
111	$DR(0-10) \leftarrow PC$	PCDR

<u>F3</u>		
000	none	NOP
001	$AC \leftarrow AC + DR$	XOR
010	$AC \leftarrow AC$	COM
011	$AC \leftarrow SHAC$	SIL
100	$AC \leftarrow SHAC$	SILB
101	$PC \leftarrow PC + 1$	INCP
110	$PC \leftarrow AR$	ARPC
111	Arithmetic	

SBR = Adres program Yajaci

CAR = Kontrol Adres Yajaci

<u>CD</u>	<u>sort</u>	<u>Simbol</u>	<u>Yajaci</u>
00	hromat	L	Sadržaj dolazna
01	$DR(15)$	I	abljki adres bit
10	$AC(15)$	S	Ac'ing isret bit
11	$AC=0$	E	Ac'ing defop sign

<u>DR</u>	<u>Simbol</u>	<u>Funkcija</u>
00	JMP	$CAR \leftarrow AD$ efer sort = 1 ise $CAR \leftarrow CAR + 1$ efer sort = 0 ise
01	CALL	$CAR \leftarrow AR$ , $SBR \leftarrow CAR + 1$ efer sort = 1 ise $CAR \leftarrow CAR + 1$ efer sort = 0 ise
10	RFT	$CAR \leftarrow SBR$
11	MAP	$CAR(2-5) \leftarrow DR(11-14)$ , $CAR(0,1,6) \leftarrow 0$

<u>Smoy adres</u>	<u>F1</u>	<u>F2</u>	<u>F3</u>	<u>CD</u>	<u>DR</u>	<u>AD</u>
1000000	110	000	000	00	00	1000001
1000001	000	100	101	00	00	1000010
1000010	101	000	000	00	11	0000000

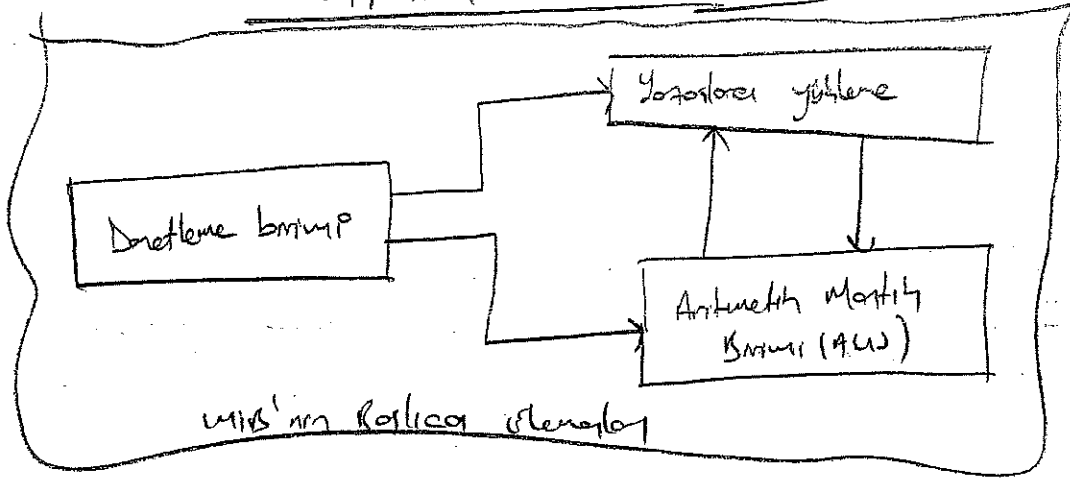
Smoy

<u>DR</u>	<u>by</u>	<u>Simbol</u>
00	PCDATA	JMP NEXT
01	READ INCP	JMP NEXT
10	DATA	MAP

NEXT = Sirodahi bir sirodahi adres gssstiri.



## Mikro İŞLEM BİRİMİ (CPU)



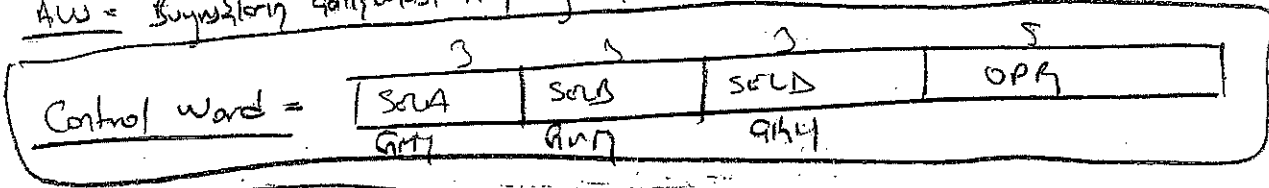
5

1.2.2

Mikro İŞLEM BİRİMİ = Veri işleme işlemleri yapar.

Denetim Birimi = Yatırımlar bilgisi transferi ve ALU birimlerinin denetiminde sorumludur.

ALU = Birimlerin çalışması için gerekli mikro işlemi gerçekleştirir.

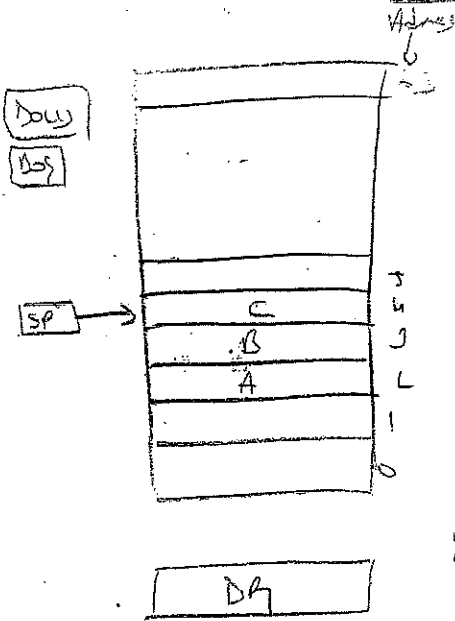


### ALU İşlemleri

00000	=> A'nın değeri	=> STP A
00001	=> A'ya 1 ekle	=> INCA
00010	=> A+B	=> ADD
00101	=> A-B	=> SUB
00110	=> A'ya 1 ekleme	=> DECA
01000	=> A ve B'yi VS'le	=> AND
01010	=> veya'la	=> OR
01100	=> XOR veya'la	=> XOR
01110	=> Tamlayıcı ol	=> COMA
10000	=> A'ya sağ kaydır	=> SHRA
11000	=> A'ya sola kaydır	=> SHLA

İşlem	Yatırımlar	İşlem
LOAD	A1, A	A1 ← m[A1]
LOAD	A2, B	A2 ← m[A2]
LOAD	A3, C	A3 ← m[A3]
LOAD	A4, D	A4 ← m[A4]
ADD	A1, A1, A2	A1 ← A1 + A2
ADD	A3, A3, A4	A3 ← A3 + A4
MUL	A1, A1, A3	A1 ← A1 * A3
STORE	X, A1	m[X] ← A1

## Yığılma Yığılması



$SP =$  Yığılma gösterir ve yığılma son indeksin kelimenin adresinin birer defa arttırılması

=> Üstteki bilgiyi silmek için; 3 adresindeki bilgi okunur ve SP'nin isareti 1 azaltılır.

=> Yığılma yeni bir sayı yazmak için; SP bir arttırılır ve yeni adres yeni bilgi yazılır.

=> Başlangıçta  $SP = 0$ ,  $BOJ = 1$ .  $DOW = 0$  yapılır.

Yığılma yapma (itme)

$SP \leftarrow SP + 1$   
 $M[SP] \leftarrow DA$   
 if ( $SP = 0$ ) then ( $DOW \leftarrow 1$ )  
 $BOJ \leftarrow 0$

Yığılma okuma (çıkarma)

$DA \leftarrow M[SP]$   
 $SP \leftarrow SP - 1$   
 if ( $SP = 0$ ) then ( $BOJ \leftarrow 1$ )  
 $DOW \leftarrow 0$

## İT POLONYALI GÖSTERİMİ

=> Matematikte  $= A + B$   
 => İn ek veya polonik gösterimi  $= + A B$   
 => Art ek veya art polonyalı gösterimi  $= A B +$

ÜÇ ADRESLİ BÜYÜKLÜK =

$V = (A + B) + (C + D)$  y yapan birleştirilirdi

ADD R1, A, B  $R1 \leftarrow M[A] + M[B]$   
 ADD R2, C, D  $R2 \leftarrow M[C] + M[D]$   
 MUL X, R1, R2  $M[X] \leftarrow R1 + R2$

İKİ ADRESLİ BÜYÜKLÜK

MOV R1, A  $R1 \leftarrow M[A]$   
 ADD R1, B  $R1 \leftarrow R1 + M[B]$   
 MOV R2, C  $R2 \leftarrow M[C]$   
 ADD R2, D  $R2 \leftarrow R2 + M[D]$   
 MUL R1, R2  $R1 \leftarrow R1 * R2$   
 MOV X, R1  $M[X] \leftarrow R1$

BİR ADRESLİ BÜYÜKLÜK =

LOAD A  $AC \leftarrow M[A]$   
 ADDS B  $AC \leftarrow AC + M[B]$   
 STORG T  $M[T] \leftarrow AC$   
 LOAD C  $AC \leftarrow M[C]$   
 ADD D  $AC \leftarrow AC + M[D]$   
 MUL T  $AC \leftarrow AC * M[T]$   
 STORG X  $M[X] \leftarrow AC$

SİFİR ADRESLİ BÜYÜKLÜK

PUSH A  $TOS \leftarrow A$   
 PUSH B  $TOS \leftarrow B$   
 ADD  $TOS \leftarrow (A + B)$   
 PUSH C  $TOS \leftarrow C$   
 PUSH D  $TOS \leftarrow D$   
 ADD  $TOS \leftarrow (C + D)$   
 MUL  $TOS \leftarrow (A + B) * (C + D)$   
 POP X  $M[X] \leftarrow TOS$

⊗  $TOS =$  yığılma göstergesi  
 ⊗ ADD, MUL, POP, PUSH adresleridir.

## KIP ALANI

6

- => İmza edilmiş kipi alanı = Bu kipte veriler buyruğun tanımıyla ilgili olarak verilir.
- => Derhal kipi = Bu kipte veri buyruğun içinde verilmektedir.
- => Yatay kipi = Bu kipte veriler MIB içindeki yataklar içinde verilir.
- => Yatay Dolaylı kipi = Bu kipte adres alanında bir yatağın adresi verilir. Yatağın içinde ise verilerin bellekte bulunduğu adres verilir.
- => Oto atama veya oto atama kipi = Bu kipi yatay dolaylı kipi gibi başka bir kipi içindeki belirli adresi kullanılmadan önce atar veya atar.

Derhal ver =  $\sqrt{16 \text{ LS MIB}}$

Etkin Adres =

Doğrudan Adres Kipi = Bu kipte etkin adres buyruğun adres alanındaki adrestir.

Dolaylı Adres Kipi = Bu kipteki adres kipi verinin bellekteki gerçek adresin bulunduğu yerin adresidir. (2)

Göreceli Adres Kipi = Bu kipte PC işareti buyruğun adres kısmına eklenir ve etkin adres elde edilir. (1)

İndirilmiş Adres Kipi = Bu kipte adres yatağının işareti buyruğun adres kısmına eklenir.

İndirilmiş adres olan bir MIB yatağı olup indirilmiş değeri  $\text{indir}(\text{indir})$

Taban Yatağı Adres Kipi = Bu kipte taban yatağının işareti buyruğun adres kısmına eklenir ve etkin adres elde edilir.

## CISC ÖZELLİKLERİ

- => Çok sayıda buyruk. Tipik olarak 100-500 adet
- => Bazı buyruklar çok özel işler yaparlar ve çok nadir kullanılırlar

- => Çok sayıda adresleme kipi tipi olarak 5-20 adet
- => Değişken uzunlukta buyruk baskımları
- => Buyrukların verileri bellekten kullanılmaları

### Kayan Noktalı (Floating point)

- => Üstlerin kısıtlandırılması (Range is report)
- => Kesirlerin hesaplandırılması
- => Kesirlerin toplanması veya çıkarılması
- => Normalizasyon

## RISC ÖZELLİKLERİ

- => Göreceli olarak az sayıda buyruk
- => Az sayıda adresleme kipi
- => Bellek erişimi sadece LOAD ve STORE buyrukları ile yapılır.
- => Basit işlemci MIB yatağında yapılır.
- => Sabit uzunlukta kolay gerçekleştirilebilir kod
- => Tek evreli buyruk isrası
- => Mikro programlı donatıların yerine donatımlı donatılar.

## Baru Hattı ve Veri İşlemeleri

### Paralel İşleme

=> Bilgisayarların büyükler sayıyla işi eder. Paralel işlemlerde ise aynı anda birçok büyük işi eder. Böylece işleri hızlanır.

=> Paralel işlemin amacı bilgisayarların işlem kapasitesini hızlandırmaktır.

SISD = Tek büyük işlemi, tek veri işlemi (Tek bir işlem birimi, bir işlemci, bir bellek)

SIMD = Tek büyük işlemi, çok veri işlemi (Tek bir işlem birimi, birçok işlemci)

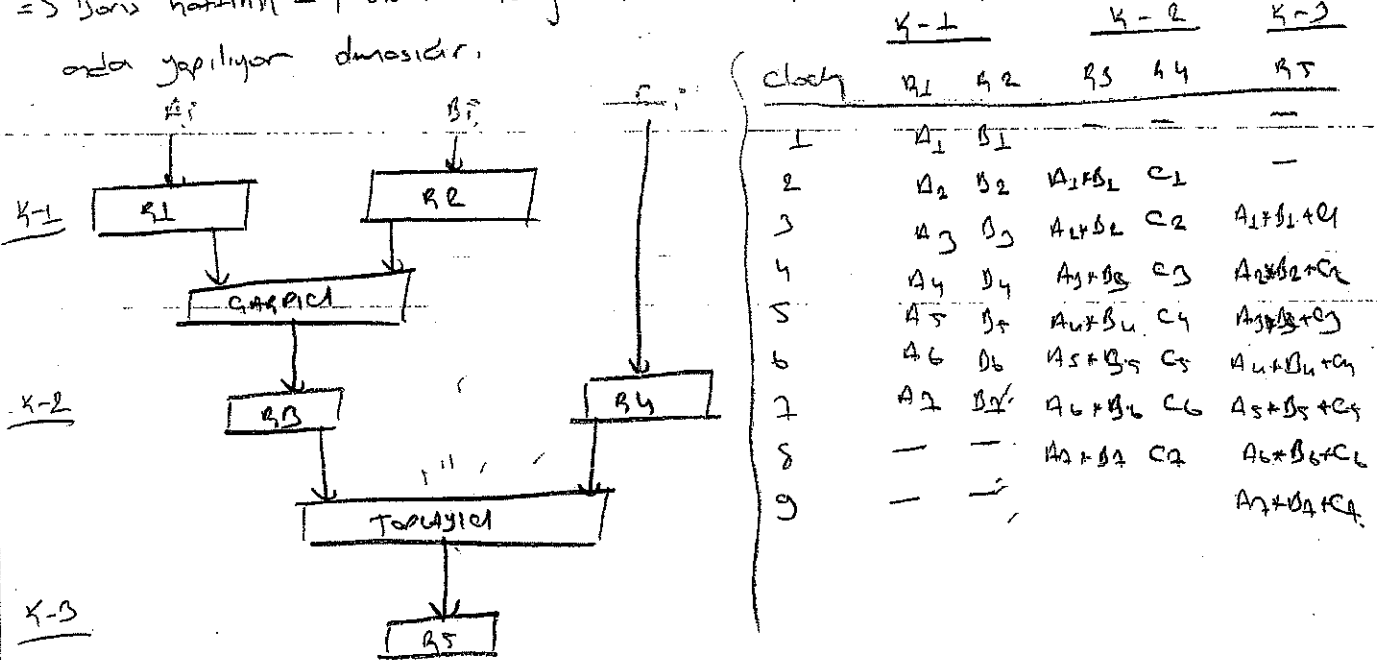
MISD = Çok büyük işlemi, tek veri işlemi

MIMD => Çok büyük işlemi, çok veri işlemi (Aynı anda farklı programlar işi ederler)

### Baru Hattı İşlemeleri

=> Barı hattı işlemi bir yöntem olup sıralı bir işlemi alt işlemlere böler. Her alt işlem kendine ait bir kısımda işi eder. Diğer kısımlarda da aynı anda başka alt işlemlerde işi eder. Veri bütün kısımlarda geçince sonuç elde edilir.

=> Barı hattının en önemli özelliği farklı hesaplamaların farklı kısımlarda aynı anda yapılmasıdır.



"n" 4 kısımla 6 görev için bütün işlemlerin bitmesi için gerekli zaman

$$tp = \left\{ \begin{aligned} & k + (n - 1) = 4 + 6 - 1 = 9 \text{ saat sürmüştür} \\ & \text{Toplam süre} = k + tp \end{aligned} \right.$$

ör =  $tp = 2000$  olsun ve 4 kısımlı baru hattı ve 100 görev işi ediyorsa hızlanma oranı 5!

=> Barı hattı olan bilgisayarda  $\Rightarrow (k + n - 1) * tp = (4 + 100 - 1) * 20 = 20600$  dur.  $\cdot 17$

=> " " olan " "  $\Rightarrow t_n = k + tp * n = 4 + 20 * 100 = 20004$  olur

$$S = 20000 / 20600 = 3,88$$

program  
1000

100 101 102 103 104 105 106 107 108 109 110  
 1006 1200 1104 2020 3108 2001 0109 0100 7705 0000

a)

ORG 100  
 LDA A106  
 CMA  
 ADD 0  
 INC  
 STA C  
 HLT  
 A 0105  
 B 0053  
 C 0100  
 FF09  
 0000

b

1006  
 2200  
 1104  
 2020  
 3108  
 2001  
 0109  
 0053  
 0064  
 FF09  
 0000

END

7

program 1

100  
 101  
 102  
 103

- 1) AC = 7705 (1111 1111 1110 1001)
- 2) AC = 0016 (0000 0000 0001 0110)
- 3) AC = AC + 0053 = 0016 + 0053 = 0069
- 4) AC = 0069 + 1 = 006A
- 5) 104 e 246

11 11 11 11 11 11 11 11

(1) (10) (10)

Seet	$k_1$	$k_2$	$k_3$	$k_4$	$k_5$	$k_6$	$k_7$
1	$A_1$	$B_1$	$C_1$	$D_1$	—	—	—
2	$A_2$	$B_2$	$C_2$	$D_2$	$(A_1+B_1)$	$(C_1+D_1)$	—
3	$A_3$	$B_3$	$C_3$	$D_3$	$A_2+B_2$	$C_2+D_2$	$(A_1+B_1), (C_1+D_1)$
4	$A_4$	$B_4$	$C_4$	$D_4$	$A_3+B_3$	$C_3+D_3$	$(A_2+B_2), (C_2+D_2)$
5	$A_5$	$B_5$	$C_5$	$D_5$	$(A_4+B_4)$	$(C_4+D_4)$	$(A_3+B_3), (C_3+D_3)$
6	$A_6$	$B_6$	$C_6$	$D_6$	$(A_5+B_5)$	$(C_5+D_5)$	$(A_4+B_4), (C_4+D_4)$
7	$A_7$	$B_7$	$C_7$	$D_7$	$(A_6+B_6)$	$(C_6+D_6)$	$(A_5+B_5), (C_5+D_5)$
8	$A_8$	$B_8$	$C_8$	$D_8$	—	—	—
9	$A_9$	$B_9$	$C_9$	$D_9$	—	—	—
10	$A_{10}$	$B_{10}$	$C_{10}$	$D_{10}$	$(A_9+B_9)$	$(C_9+D_9)$	$(A_8+B_8), (C_8+D_8)$
11	—	—	—	—	$(A_{10}+B_{10})$	$(C_{10}+D_{10})$	$(A_9+B_9), (C_9+D_9)$
12	—	—	—	—	—	—	$(A_{11}+B_{11}), (C_{11}+D_{11})$

	1	2	3	4	5	6	7	8	9	10	11	12
$k=1$	$T_1$	$T_2$	$T_3$	$T_4$	$T_5$	$T_6$	$T_7$	$T_8$	$T_9$	$T_{10}$	—	—
$k=2$	—	$T_1$	$T_2$	$T_3$	$T_4$	$T_5$	$T_6$	$T_7$	$T_8$	$T_9$	$T_{10}$	—
$k=3$	—	—	$T_1$	$T_2$	$T_3$	$T_4$	$T_5$	$T_6$	$T_7$	$T_8$	$T_9$	$T_{10}$

①  $\Rightarrow$  bushott, olugay biyogayden  $\Rightarrow 10 \times 3 = 30$   
 bushott olay "  $\Rightarrow \frac{30}{12} = 2,5$