9. HAFTA

BLM320

BİLGİSAYAR MİMARİSİ

Yrd. Doç. Dr. Salih GÖRGÜNOĞLU

sgorgunoglu@karabuk.edu.tr

KBUZEM
Karabük Üniversitesi
Uzaktan Eğitim Uygulama ve AraştırmaMerkezi

9. Temel Bilgisayarı programlama

Bir bilgisayarı programlamak için doğrudan ikili kodlar kullanılabilir. Ancak bu durumda hata yapmak ve hataları düzeltmek çok zorlaşır. Program yazmak çok zahmetli bir süreç hakine dönüşür.

İki sayıyı toplayan binary(ikliliksayı) program

Binary Program to Add Two Numbers

Location	Instruction code
0	0010 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

Komut kodları hexedesimal sayı formatında yazılabilir. Bu işlemi biraz daha kolaylaştırır.

İki sayıyı toplayan hexedesimal program

Hexadecimal Program to Add Two Numbers

Location	Instruction
000	2004
001	1005
002	3006
003	7001
004	0053
005	FFE9
006	0000

Programın daha iyi anlaşılması, daha kolay yazılabilmesi için sayısal komut kodları yerine sembolik kodlar kullanılabilir. Bu şekilde yazılan dile assembly dili denir. Bu program yazmayı oldukça kolaylaştırır.

İki sayıyı toplayan assembly program kodu

Program with Symbolic Operation Codes

Location	Instruction	Comments
000	LDA 004	Load first operand into *AC*
001	ADD 005	Add second operand to *AC*
002	STA 006	Store sum in location 006
003	HLT	Halt computer
004	0053	First operand
005	FFE9	Second operand (negative)
006	0000	Store sum here

Sayısal değerler A,B,C olarak aşağıdaki tanımlanırsa daha anlaşılır bir program olur

Assembly Language Program to Add Two Numbers

```
        ORG 0       /Origin of program is location 0
        LDA A       /Load operand from location A
        ADD B       /Add operand from location B
        STA C       /Store sum in location C
        HLT         /Halt computer
A,      DEC 83      /Decimal operand
B,      DEC -23     /Decimal operand
C,      DEC 0       /Sum stored in location C
        END         /End of symbolic program
```

Temel bilgisayarı programlamak için kullanılan assembly dilinin bazı kuralları vardır. Örneğin programın hangi adresten başlyarak yazılacağı ORG talimatı ile bildirilir. END talimatı programın yazımının sonlandığını ifade eder. DEC talmatı desimal sayı için bir adres gösterir. FEX talimatı hexedesimal sayı için bir adres gösterir. Program yazarken dallanmaların yapılacağı yerler etiketlerlem gösterilir ve bu etiketler virgül "," ile ayrılır.

Definition of Pseudoinstructions

Symbol	Information for the Assembler
ORG N	Hexadecimal number N is the memory location for the instruction or operand listed in the following line
END	Denotes the end of symbolic program
DEC N	Signed decimal number N to be converted to binary
HEX N	Hexadecimal number N to be converted to binary

Ve programın son hali aşağıdaki şekilde verilmiştir.

Assembly Language Program to Subtract Two Numbers

```
        ORG 100     /Origin of program is location 100
        LDA SUB     /Load subtrahend to AC
        CMA         /Complement AC
        INC         /Increment AC
        ADD MIN     /Add minuend to AC
        STA DIF     /Store difference
        HLT         /Halt computer
MIN,    DEC 83      /Minuend
SUB,    DEC -23     /Subtrahend
DIF,    HEX 0       /Difference stored here
        END         /End of symbolic program
```

Programın komut kodları ile birlikte aşağıdaki gibi yazılabilir.

ng of Translated Program of Table 6-8

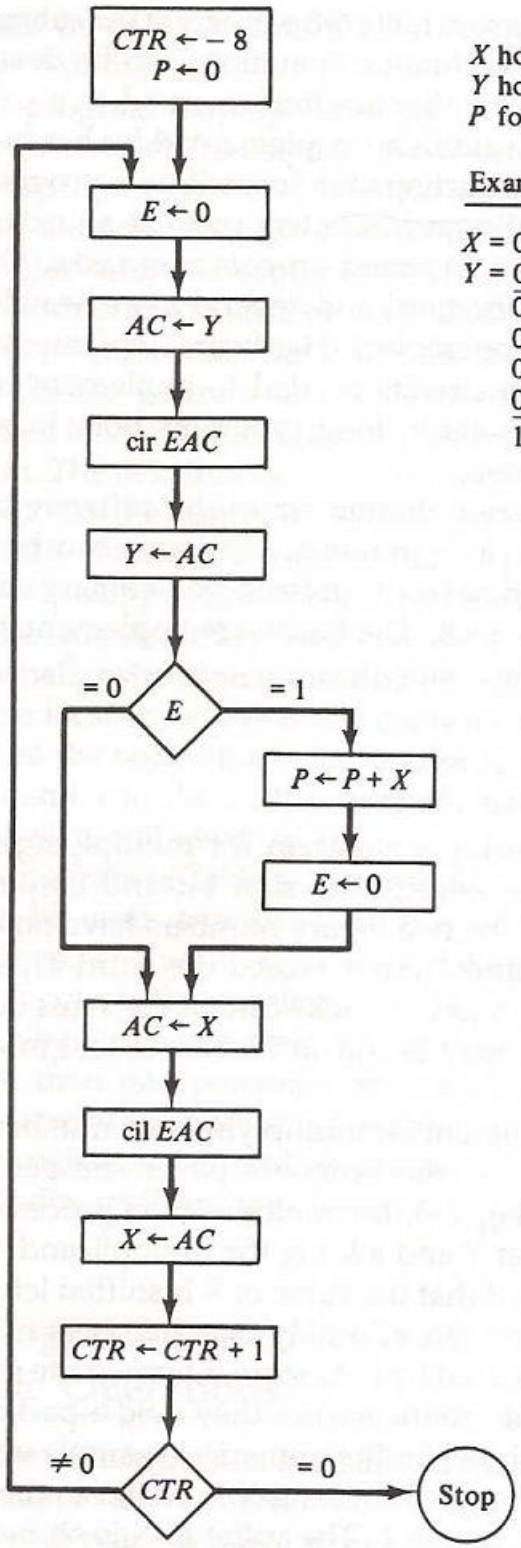
Hexadecimal code			
Location	Content		Symbolic program
			ORG 100
100	2107		LDA SUB
101	7200		CMA
102	7020		INC
103	1106		ADD MIN
104	3108		STA DIF
105	7001		HLT
106	0053	MIN,	DEC 83
107	FFE9	SUB,	DEC −23
108	0000	DIF,	HEX 0
			END

100 adet sayıyı toplayan program

Symbolic Program to Add 100 Numbers

Line			
1		ORG 100	/Origin of program is HEX 100
2		LDA ADS	/Load first address of operands
3		STA PTR	/Store in pointer
4		LDA NBR	/Load minus 100
5		STA CTR	/Store in counter
6		CLA	/Clear accumulator
7	LOP,	ADD PTR I	/Add an operand to *AC*
8		ISZ PTR	/Increment pointer
9		ISZ CTR	/Increment counter
10		BUN LOP	/Repeat loop again
11		STA SUM	/Store sum
12		HLT	/Halt
13	ADS,	HEX 150	/First address of operands
14	PTR,	HEX 0	/This location reserved for a pointer
15	NBR,	DEC −100	/Constant to initialized counter
16	CTR,	HEX 0	/This location reserved for a counter
17	SUM,	HEX 0	/Sum is stored here
18		ORG 150	/Origin of operands is HEX 150
19		DEC 75	/First operand
•			
•			
•			
118		DEC 23	/Last operand
119		END	/End of symbolic program

İki sayıyınçarpma akış şeması



X holds the multiplicand
Y holds the multiplier
P forms the product

Example with four significant digits

		P
X =	0000 1111	0000 0000
Y =	0000 1011	0000 1111
	0000 1111	0010 1101
	0001 1110	0010 1101
	0000 0000	1010 0101
	0111 1000	
	1010 0101	

Flowchart for multiplication program.

Program to Multiply Two Positive Numbers

```
        ORG 100
LOP,    CLE         /Clear E
        LDA Y       /Load multiplier
        CIR         /Transfer multiplier bit to E
        STA Y       /Store shifted multiplier
        SZE         /Check if bit is zero
        BUN ONE     /Bit is one; go to ONE
        BUN ZRO     /Bit is zero; go to ZRO
ONE,    LDA X       /Load multiplicand
        ADD P       /Add to partial product
        STA P       /Store partial product
        CLE         /Clear E
ZRO,    LDA X       /Load multiplicand
        CIL         /Shift left
        STA X       /Store shifted multiplicand
        ISZ CTR     /Increment counter
        BUN LOP     /Counter not zero; repeat loop
        HLT         /Counter is zero; halt
CTR,    DEC -8      /This location serves as a counter
X,      HEX 000F    /Multiplicand stored here
Y,      HEX 000B    /Multiplier stored here
P,      HEX 0       /Product formed here
        END
```

Altprogram kullanma.

Aşağıda altprogram kullanarak bir sayıyı 4 sefa sola kaydıran ve saklayan bir program görülmektedir.

Program to Demonstrate the Use of Subroutines

Location			
		ORG 100	/Main program
100		LDA X	/Load X
101		BSA SH4	/Branch to subroutine
102		STA X	/Store shifted number
103		LDA Y	/Load Y
104		BSA SH4	/Branch to subroutine again
105		STA Y	/Store shifted number
106		HLT	
107	X,	HEX 1234	
108	Y,	HEX 4321	
			/Subroutine to shift left 4 times
109	SH4,	HEX 0	/Store return address here
10A		CIL	/Circulate left once
10B		CIL	
10C		CIL	
10D		CIL	/Circulate left fourth time
10E		AND MSK	/Set *AC*(13–16) to zero
10F		BUN SH4 I	/Return to main program
110	MSK,	HEX FFF0	/Mask operand
		END	

Aşağıda verilen program verileri taşımaktadır.

Subroutine to Move a Block of Data

```
                        /Main program
        BSA MVE         /Branch to subroutine
        HEX 100         /First address of source data
        HEX 200         /First address of destination data
        DEC -16         /Number of items to move
        HLT
MVE,    HEX 0           /Subroutine MVE
        LDA MVE I       /Bring address of source
        STA PT1         /Store in first pointer
        ISZ MVE         /Increment return address
        LDA MVE I       /Bring address of destination
        STA PT2         /Store in second pointer
        ISZ MVE         /Increment return address
        LDA MVE I       /Bring number of items
        STA CTR         /Store in counter
        ISZ MVE         /Increment return address
LOP,    LDA PT1 I       /Load source item
        STA PT2 I       /Store in destination
        ISZ PT1         /Increment source pointer
        ISZ PT2         /Increment destination pointer
        ISZ CTR         /Increment counter
        BUN LOP         /Repeat 16 times
        BUN MVE I       /Return to main program
PT1,    —
PT2,    —
CTR,    —
```

Aşağıda verilen program giriş çıkış işlemlerini göstermektedir.

Programs to Input and Output One Character

```
(a) Input a character:
    CIF,    SKI       /Check input flag
            BUN CIF   /Flag=0, branch to check again
            INP       /Flag=1, input character
            OUT       /Print character
            STA CHR   /Store character
            HLT
    CHR,    —         /Store character here
(b) Output one character:
            LDA CHR   /Load character into AC
    COF,    SKO       /Check output flag
            BUN COF   /Flag=0, branch to check again
            OUT       /Flag=1, output character
            HLT
    CHR,    HEX 0057  /Character is "W"
```

Aşağıda verilen program iki sayıyı lkarşılaştırmaktadır.

Program to Compare Two Words

```
        LDA WD1   /Load first word
        CMA
        INC       /Form 2's complement
        ADD WD2   /Add second word
        SZA       /Skip if AC is zero
        BUN UEQ   /Branch to "unequal" routine
        BUN EQL   /Branch to "equal" routine
WD1,    —
WD2,    —
```

Aşağıda verieln program kesme kullanımı ile ilgilidir.

Program to Service an Interrupt

Location			
0	ZRO,	—	/Return address stored here
1		BUN SRV	/Branch to service routine
100		CLA	/Portion of running program
101		ION	/Turn on interrupt facility
102		LDA X	
103		ADD Y	/Interrupt occurs here
104		STA Z	/Program returns here after interrupt
•		•	
•		•	
•			/Interrupt service routine
200	SRV,	STA SAC	/Store content of *AC*
		CIR	/Move *E* into *AC*(1)
		STA SE	/Store content of *E*
		SKI	/Check input flag
		BUN NXT	/Flag is off, check next flag
		INP	/Flag is on, input character
		OUT	/Print character
		STA PT1 I	/Store it in input buffer
		ISZ PT1	/Increment input pointer
	NXT,	SKO	/Check output flag
		BUN EXT	/Flag is off, exit
		LDA PT2 I	/Load character from output buffer
		OUT	/Output character
		ISZ PT2	/Increment output pointer
	EXT,	LDA SE	/Restore value of *AC*(1)
		CIL	/Shift it to *E*
		LDA SAC	/Restore content of *AC*
		ION	/Turn interrupt on
		BUN ZRO I	/Return to running program
	SAC,	—	/*AC* is stored here
	SE,	—	/*E* is stored here
	PT1,	—	/Pointer of input buffer
	PT2,	—	/Pointer of output buffer

Örnek Programlar

1.

;Bu program bir kesme geldiğinde ACC deki değeri 2 defa artırır.

;Ana programda 200 adresindeki değerle 201 adresindeki değer toplanır ve sonuc 202 adresine atılır

Adres(location)	İçerik		Program
000	101		
001	4300		BUN 300
Ana progrm			
100	2200		LDA 200
101	1201		ADD 201
102	3202		STA 202
103	7001		HLT
Adres içeriği			
200	5		
201	6		
202			
Kesme hizmet programı (ISR – Interrupt Service Routine)			
300	7020		INC
301	7020		INC
302	C000		BUN 000 I

Aynı programı daha anlaşılır ve esnek bir biçimde yazabiliriz.

Adres(location)	İçerik	Program	
000	101		
001	4300		BUN ISR
		Ana program	
100	2200		LDA X
101	1201		ADD Y
102	3202		STA Z
103	7001		HLT
		Değişken (adres) içerikleri	
200	5	X,	5
201	6	Y,	6
202		Z,	-
		ISR alt programı	
300	7020	ISR,	INC
301	7020		INC
302	C000		BUN 000 I

Aşağıda verilen programları yapınız

1.

```
;Ac ye 1 yükle ve 4 defa 1 ekle, 4 olunca çik
;sonucu outr'ye yaz. Bir döngü oluşturarak yapınız.
```

2.

```
; Önce ac'ye ff yükle, 4 defa sağa kaydir
; sonucu e0 adresine yaz, ac'ye f0 al
; e0 adresindeki veriyle AND işlemi sonucu outr'ye yaz
```

3.

```
;Önce ac ye 01 değerini yükler ve
;sürekli sola kaydir
;interrupt gelince mevcut değeri 1 azalt ve
;outr ye at
```

4.

```
;Önce girişten bilgi al, alinan değeri 01 ile
;karşilaştir 1 ise çikişa 1; 0 ise çikişa 2;
; 1 den BÜYÜK İSE ÇIKIŞA 3 GÖNDER
```

5.

```
;Girişten döngü değerini al, alinan değer kadar
;ac'yi sağa kaydir, kaydirma işini prosedür ile
;yap, sonucu outr'ye yaz
```