

PIC16F84 KOMUT TABLOSU					
Komutlar	İngilizce Açılımı	Açıklaması	İşlem		
<i>Data Transfer Komutları</i>					
1	MOVLW	k	Move Literal to W	k sabitini W registerine yükler	$k \rightarrow W$
2	MOVWF	f	Move W to f	W registerinin içeriğini f registerine yükler	$W \rightarrow f$
3	MOVF	f,d	Move f	f registerin içeriğini W veya f'e yükler	$f \rightarrow d$
4	CLRW		Clear W	W registerinin içeriğini sıfırlar	$0 \rightarrow W$
5	CLRF	f	Clear f	f registerinin içeriğini sıfırlar	$0 \rightarrow f$
6	SWAPF	f,d	Swap nibbles in f	f registerdeki ilk dört bit ile son dört bit yer değiştirerek sonucu W veya f'e yükler.	$f(00001111) \rightarrow f(11110000)$
<i>Aritmetik ve Lojik Komutları</i>					
7	ADDLW	k	Add Literal and W	W registerin içeriğini k sabiti ile toplayarak sonucu W registerine yükler.	$W+k \rightarrow W$
8	ADDWF	f,d	Add W and f	W ile f registeri içeriğini toplayarak sonucu W veya f registerine yükler.	$W+f \rightarrow d$
9	SUBLW	k	Subtract W from Literal	k sabit değerinden W register içeriğini çıkarır ve sonucu W registerine yükler.	$W-k \rightarrow W$
10	SUBWF	f,d	Subtract W from f	f registerden W register içeriğini çıkararak sonucu W registeri veya f registeri içerisine yükler.	$W-f \rightarrow d$
11	ANDLW	k	AND Literal with W	W registerinin içeriği ile k sabitine AND işlemi uygulayarak sonucu W registerine yükler.	$W \text{ AND } k \rightarrow W$
12	ANDWF	f,d	AND W with f	W ile f register içeriğine AND işlemi uygulayarak sonucu W veya f registerine yükler.	$W \text{ AND } f \rightarrow d$
13	IORLW	k	Inclusive OR Literal with W	W registerinin içeriği ile k sabitine OR işlemi uygulayarak sonucu W registerine yükler.	$W \text{ OR } k \rightarrow W$
14	IORWF	f,d	Inclusive OR W with f	W registeri ile f register içeriğine OR işlemi uygulayarak sonucu W veya f registerine yükler.	$W \text{ OR } f \rightarrow d$
15	XORLW	k	Exclusive OR Literal with W	W registerinin içeriği ile k sabitine XOR işlemi uygulayarak sonucu W registerine yükler.	$W \text{ XOR } k \rightarrow W$
16	XORWF	f,d	Exclusive OR W with f	W ile f register içeriğine XOR işlemi uygulayarak sonucu W veya f registerine yükler.	$W \text{ XOR } f \rightarrow d$
17	INCF	f,d	Increment f	f registerin içeriğini 1 artırarak sonucu W registerine yükler.	$f+1 \rightarrow f$
18	DECF	f,d	Decrement f	f registerin içeriğini 1 azaltarak sonucu W registerine yükler.	$f-1 \rightarrow f$
19	RLF	f,d	Rotate Left f through carry	f registerin içeriğini bir sola kaydırarak sonucu W veya f'e yükler.	$\left[\begin{array}{c} \leftarrow \\ \leftarrow \\ \leftarrow \\ \leftarrow \\ \leftarrow \\ \leftarrow \\ \leftarrow \\ \leftarrow \\ \leftarrow \\ \leftarrow \end{array} \right]$
20	RRF	f,d	Rotate Right f through carry	f registerin içeriğini bir sağa kaydırarak sonucu W veya f'e yükler.	$\left[\begin{array}{c} \leftarrow \\ \leftarrow \\ \leftarrow \\ \leftarrow \\ \leftarrow \\ \leftarrow \\ \leftarrow \\ \leftarrow \\ \leftarrow \\ \leftarrow \end{array} \right]$
21	COMF	f,d	Complement f	f register içerisindeki sayıyı tersleyerek sonucu W veya f'e yükler.	$\bar{f} \rightarrow d$
<i>Bit İşlem Komutları</i>					
22	BCF	f,b	Bit Clear f	f register içerisindeki sayının b.inci bitini sıfırlar.	$0 \rightarrow f(b)$
23	BSF	f,b	Bit Set f	f register içerisindeki sayının b.inci bitini 1 yapar.	$1 \rightarrow f(b)$
<i>Program Akışı Yönlendirme Komutları</i>					
24	BTFSZ	f,b	Bit Test f, Skip if Clear	f registerin b.inci bitini test eder. Eğer bu bit sıfırsa program akışı bir sonraki komutu atlar.	$f(b)=0 \Rightarrow$ sıradaki komutu atla
25	BTFSZ	f,b	Bit Test f, Skip if Set	f registerin b.inci bitini test eder. Eğer bu bit 1 ise program akışı bir sonraki komutu atlar.	$f(b)=1 \Rightarrow$ sıradaki komutu atla
26	DECFSZ	f,d	Decrement f, Skip if zero	f registerin içeriğini 1 azaltır. Eğer register içeriği sıfırsa bir sonraki komutu atlayarak sonucu W veya f registerine yazar.	$f-1 \rightarrow d, Z=0 \Rightarrow$ sıradaki komutu atla
27	INCFSZ	f,d	Increment f, Skip if zero	f registerin içeriğini 1 artırır. Eğer register içeriği sıfırsa bir sonraki komutu atlayarak sonucu W veya f registerine yazar.	$f+1 \rightarrow d, Z=0 \Rightarrow$ sıradaki komutu atla
28	GOTO	k	Go to address	Program akışık adresine daller.	GOTO k
29	CALL	k	Call subroutine	Program akışık etiketinin bulunduğu al programa daller.	CALL k
30	RETURN		Return from subroutine	Program akışını ana programa döndürür.	RETURN
31	RETLW	k	Return with Literal in W	Program akışını ana programa döndürür ve W registerine k sabitini yükler.	Ana program+k \rightarrow W
32	RETFIE		Return From Interrupt	Program akışını Interrupt (kesme) alt programından ana programa döndürür.	Kesme \rightarrow Ana program
<i>Diğer Yönlendirme Komutları</i>					
33	NOP		No Operation	Zaman geciktirme işlemi için bir komut sayıklı süresince hiçbir işlem yapmaz.	.
34	CLRWD T		Clear Watchdog Timer	Watchdog Timer'i sıfırlar.	$0 \rightarrow \text{WD T}, 1 \rightarrow \text{T O}, 1 \rightarrow \text{P D}$
35	SLEEP		Go into standby mode	Mikro denetleyici uyuma moduna geçer.	$1 \rightarrow \text{WD T}, 1 \rightarrow \text{T O}, 0 \rightarrow \text{P D}$

f : file register

W : Work register

b : f register içerisindeki bit pozisyonu

d : Yönlendirilecek yer (d=0 ise f register, d=1 ise W register)

KOMUTLAR ve ÖRNEKLER

1- MOVLW k: k sabit degerini W registerine yükler.

Örnek : **MOVLW H'FF'** : H'FF' sabit degerini W registerine yükler. Komut öncesi W register içeriği H'00' gibi herhangi bir deger ise komut sonrası H'FF' olur.

2- MOVWF f: W registerinin içeriğini f registerine yükler.

Örnek : **MOVLW TRISB** : W register içeriği H'FF' ise bu degeri TRISB registerine yükler.

3- MOVF f,d : f registerin içeriğini W registerine veya yine f registerine yükler.

a) Eger d = 0 ise W registere yükler.

Örnek : **MOVF FSR,0** : FSR register içeriği H'A0' ise W register içeriği de H'A0' olur.

b) Eger d = 1 ise f registere yükler.

Örnek : **MOVF FSR,1** : W register içeriği sabit kalırken FSR register içeriği H'FF' ise bu degeri yine FSR registere yükler. d=1 durumu status registerdeki Z bayragini (flag) etkiler. Z=1 olur.

4- CLRW : W register içeriğini sıfırlar.

Örnek : **CLRW** : Komut öncesi W register içeriği H'FF' ise komut sonrası W register içeriği H'00' olur.

5- CLRF f: f register içeriğini sıfırlar.

Örnek : **CLRF PORTB** : PORTB'yi sıfırlar. PORTB H'00' veya B'00000000' olur. Bu durum aynı zamanda PIC'in B portunun çıkis olarak tanımlanmasında kullanilir.

6- SWAPF f,d : f registerdeki ilk dört bit ile son dört biti yer degistirerek sonucu W veya f registere yükler.

a) Eger d = 0 ise W registere yükler.

Örnek : **SWAP REG,0** : REG register içeriği H'F0' ise W register içeriği de H'0F' olur. Diger bir ifadeyle binary formda REG register içeriği B'11110000' ise W register içeriği B'00001111' olur. REG, yazilimci tarafından tanımlanan özel bir register adresidir ve isim degistirilebilir.

b) Eger d = 1 ise f registere yükler.

Örnek : **SWAP REG,1** : REG register içeriği H'F0' ise komut sonrası REG register içeriği H'0F' olur. Binary formda REG register içeriği B'11110000' ise komut sonrası B'00001111' olur.

7- ADDLW k : W registerin içeriğini k sabiti ile toplayarak sonucu W registere yükler.

Örnek : **ADDLW H'10'** : W register içeriği H'15' ise komut sonrası W register içeriği H'25' olur.

$$\begin{array}{r} \text{k sabiti H'10'} \\ \text{W register içeriği H'15'} \\ \hline \text{Komut sonrası W register içeriği :} \end{array} \quad \begin{array}{r} 0001 \ 0000 \\ \text{ADDLW} \ 0001 \ 0101 \\ \hline 0010 \ 0101 \end{array} \quad (\text{H'25'})$$

8- ADDWF f,d : W registeri ile f registeri içeriğini toplayarak sonucu W registere veya f registere yükler.

a) Eger d = 0 ise W registere yükler.

Örnek : **ADDWF FSR,0** : FSR register içeriği H'0B' ve W register içeriği de H'11' ise W register içeriği H'1C' olur. FSR register içeriği ise H'0B' olarak kalir.

FSR register içeriđi H'0B'		0000 1011	
W register içeriđi H'11'	ADDLW	0001 0001	
Komut sonrası W register içeriđi :		0001 1100	(H'1C')

b) Eger d = 1 ise f registre yükler.

Örnek : ADDWF FSR,1 : FSR register içeriđi H'4F' ve W register içeriđi de H'A0' ise FSR register içeriđi H'EF' olur. W register içeriđi ise H'A0' olarak kalir.

FSR register içeriđi H'4F'		0100 1111	
W register içeriđi H'A0'	ADDLW	1010 0000	
Komut sonrası FSR register içeriđi :		1110 1111	(H'EF')

9-SUBLW k : k sabit deđerinden W register içeriđini çıkarir ve sonucu W registerine yükler.

Örnek : SUBLW H'15' : W register içeriđi H'10' ise komut sonrası W register içeriđi H'05' olur.

FSR register içeriđi H'4F'		0100 1111	
W register içeriđi H'A0'	ADDLW	1010 0000	
Komut sonrası FSR register içeriđi :		1110 1111	(H'EF')

10-SUBWF f,d : f register içeriđinden W register içeriđini çıkarir ve sonucu W veya f registerine yükler.

a) Eger d = 0 ise W registre yükler.

Örnek : SUBWF KPR,0 : KPR register içeriđi H'D0' ve W register içeriđi de H'10' ise W register içeriđi H'C0' olur. KPR register içeriđi ise H'D0' olarak kalir.

KPR register içeriđi H'D0'		1101 0000	
W register içeriđi H'10'	SUBWF	0001 0000	
Komut sonrası W register içeriđi :		1100 0000	(H'C0')

b) Eger d = 1 ise f registre yükler.

Örnek : SUBWF KPR,1 : KPR register içeriđi H'7A' ve W register içeriđi de H'51' ise KPR register içeriđi H'29' olur. W register içeriđi ise H'51' olarak kalir.

KPR register içeriđi H'7A'		0111 1010	
W register içeriđi H'51'	SUBWF	0101 0001	
Komut sonrası W register içeriđi :		0010 1001	(H'29')

11-ANDLW k : W registeri içeriđi ile k sabitine AND islemi uygulayarak sonucu W registerine yükler.

Örnek : ANDLW H'CF' : W register içeriđi H'1A' ise komut sonrası W register içeriđi H'0A' olur.

k sabiti H'CF'		1100 1111	
W register içeriđi H'1A'	AND	0001 1010	
Komut sonrası W register içeriđi :		0000 1010	(H'0A')

12-ANDWF f,d : W register ile f register içeriđine AND islemi uygulayarak sonucu W registerine veya f registerine yükler.

a) Eger d = 0 ise W registre yükler.

Örnek : ANDWF KSM,0 : KSM register içeriđi H'2C' ve W register içeriđi de H'12' ise W register içeriđi H'0C' olur. KSM register içeriđi ise H'2C' olarak kalir.

KSM register içeriđi H'2C'		0010 1100	
W register içeriđi H'12'	AND	0001 1100	
Komut sonrası W register içeriđi :		0000 1100	(H'0C')

b) Eger $d = 1$ ise f registere yükler.

Örnek : ANDWF KSM,1 : KSM register içeriği H'A3' ve W register içeriği de H'F0' ise KSM register içeriği H'A0' olur. W register içeriği ise H'F0' olarak kalir.

$$\begin{array}{r} \text{KSM register içeriği H'A3'} \\ \text{W register içeriği H'F0'} \\ \hline \text{Komut sonrası W register içeriği :} \end{array} \quad \begin{array}{r} 1010 \ 0011 \\ \text{AND} \ 1111 \ 0000 \\ \hline 1010 \ 0000 \end{array} \quad (\text{H'A0'})$$

13- IORLW k : W registerinin içeriği ile k sabitine OR islemi uygulayarak sonucu W registerine yükler.

Örnek : IORLW H'B4' : W register içeriği H'26' ise komut sonrası W register içeriği H'B6' olur.

$$\begin{array}{r} \text{k sabiti HB4'} \\ \text{W register içeriği H'26'} \\ \hline \text{Komut sonrası W register içeriği :} \end{array} \quad \begin{array}{r} 1011 \ 0100 \\ \text{OR} \ 0010 \ 0110 \\ \hline 1011 \ 0110 \end{array} \quad (\text{H'B6'})$$

14- IORWF f,d : W registeri ile f registeri içeriğine OR islemi uygulayarak sonucu W registerine veya f registerine yükler.

a) Eger $d = 0$ ise W registere yükler.

Örnek : IORWF MTN,0 : MTN register içeriği H'19' ve W register içeriği de H'4D' ise W register içeriği H'5D' olur. MTN register içeriği ise H'19' olarak kalir.

$$\begin{array}{r} \text{MTN register içeriği H'19'} \\ \text{W register içeriği H'4D'} \\ \hline \text{Komut sonrası W register içeriği :} \end{array} \quad \begin{array}{r} 0001 \ 1001 \\ \text{OR} \ 0100 \ 1101 \\ \hline 0101 \ 1101 \end{array} \quad (\text{H'5D'})$$

b) Eger $d = 1$ ise f registere yükler.

Örnek : IORWF MTN,1 : MTN register içeriği H'DE' ve W register içeriği de H'AA' ise MTN register içeriği H'FE' olur. W register içeriği ise H'AA' olarak kalir.

$$\begin{array}{r} \text{MTN register içeriği H'DE'} \\ \text{W register içeriği H'AA'} \\ \hline \text{Komut sonrası W register içeriği :} \end{array} \quad \begin{array}{r} 1101 \ 1110 \\ \text{OR} \ 1010 \ 1010 \\ \hline 1111 \ 1110 \end{array} \quad (\text{H'FE'})$$

15- XORLW k : W registerinin içeriği ile k sabitine XOR islemi uygulayarak sonucu W registerine yükler.

Örnek : XORLW H'F5' : W register içeriği H'67' ise komut sonrası W register içeriği H'92' olur.

$$\begin{array}{r} \text{k sabiti H'F5'} \\ \text{W register içeriği H'67'} \\ \hline \text{Komut sonrası W register içeriği :} \end{array} \quad \begin{array}{r} 1111 \ 0101 \\ \text{XOR} \ 0110 \ 0111 \\ \hline 1001 \ 0010 \end{array} \quad (\text{H'92'})$$

16- XORWF f, d : W registeri ile f registeri içeriğine XOR islemi uygulayarak sonucu W registerine veya f registerine yükler.

a) Eger $d = 0$ ise W registere yükler.

Örnek : XORWF ENG,0 : ENG register içeriği H'C2' ve W register içeriği de H'CF' ise W register içeriği H'0D' olur. ENG register içeriği ise H'C2' olarak kalir.

$$\begin{array}{r} \text{ENG register içeriği H'C2'} \\ \text{W register içeriği H'CF'} \\ \hline \text{Komut sonrası W register içeriği :} \end{array} \quad \begin{array}{r} 1100 \ 0010 \\ \text{XOR} \ 1100 \ 1111 \\ \hline 0000 \ 1101 \end{array} \quad (\text{H'0D'})$$

b) Eger $d = 1$ ise f registere yükler.

Örnek : XORWF ENG,1 : ENG register içeriği H'AA' ve W register içeriği de H'0F' ise ENG register içeriği H'A5' olur. W register içeriği ise H'0F' olarak kalır.

ENG register içeriği H'AA'	1010 1010	
W register içeriği H'0F'	XOR 0000 1111	
Komut sonrası W register içeriği :	1010 0101	(H'A5')

17- INCF f,d : f registerin içeriğini 1 artırarak sonucu W registere yükler.

a) Eger d = 0 ise W registere yükler.

Örnek : INCF REG,0 : REG register içeriği H'D4' ise komut sonrası W register içeriği H'D5' olur.

b) Eger d = 1 ise f registere yükler.

Örnek : INCF REG,1 : REG register içeriği H'A1' ise komut sonrası REG register içeriği H'A2' olur

18- DECF f, d : f registerin içeriğini 1 azaltarak sonucu W registere yükler.

a) Eger d = 0 ise W registere yükler.

Örnek : DECF KPR,0 : KPR register içeriği H'10' ise komut sonrası W register içeriği H'0F' olur.

b) Eger d = 1 ise f registere yükler.

Örnek : DECF KPR,1 : KPR register içeriği H'11' ise komut sonrası KPR register içeriği H'10' olur

19- RLF f,d : f registerin içeriğini bir sola kaydırarak sonucu W registere veya f registere yükler.



a) Eger d = 0 ise W registere yükler.

Örnek : RLF MTN,0 : MTN register içeriği B'11001011' ve C biti (Carry) = 0 ise komut sonrası W register içeriği B'10010110' ve C=1 olur. MTN register içeriği ise B'11001011' olarak kalır.

b) Eger d = 1 ise f registere yükler.

Örnek : RLF MTN,1 : MTN register içeriği B'11001011' ve C biti (Carry) = 0 ise komut sonrası MTN register içeriği B'10010110' ve C=1 olur.

20- RRF f,d : f registerin içeriğini bir sağa kaydırarak sonucu W registere veya f registere yükler.



a) Eger d = 0 ise W registere yükler.

Örnek : RRF MTN,0 : MTN register içeriği B'11001011' ve C biti (Carry) = 0 ise komut sonrası W register içeriği B'01100101' ve C=1 olur. MTN register içeriği ise B'11001011' olarak kalır.

b) Eger d = 1 ise f registere yükler.

Örnek : RLF MTN,1 : MTN register içeriği B'11001011' ve C biti (Carry) = 0 ise komut sonrası içeriği B'01100101' ve C=1 olur.

MTN register

21- COMF f,d : f register içerişindeki sayiyi tersleyerek sonucu W registre veya f registre yükler.

a) Eger d = 0 ise W registre yükler.

Örnek : COMF ENG,0 : ENG register içeriği H'15' ise komut sonrası W register içeriği H'EA' olur.

ENG register içeriği H'15'	0001 0101	
ENG register içeriğinin tersi	Complement	
Komut sonrası W register içeriği	1110 1010	(H'EA')

b) Eger d = 1 ise f registre yükler.

Örnek : COMF ENG,1 : ENG register içeriği H'F5' ise komut sonrası ENG register içeriği H'0A' olur.

ENG register içeriği H'F5'	1111 0101	
ENG register içeriğinin tersi	Complement	
Komut sonrası ENG register içeriği	0000 1010	(H'0A')

22- BCF f,b : f register içindeki sayinin b. bitini sıfırlar.

Örnek : BCF PORTB,7 : Komut öncesi PORTB içeriği H'8A' ise komut sonrası H'6A' olur.

PORTB içeriği H'8A'	1000 1010	
Komut sonrası PORTB içeriği	0110 1010	(H'6A')

7.bit

23- BSF f,b : f register içindeki sayinin b. bitini 1 yapar.

Örnek : BSF PORTB,5 : Komut öncesi PORTB içeriği H'4B' ise komut sonrası H'6B' olur.

PORTB içeriği H'4B'	0100 1011	
Komut sonrası PORTB içeriği	0110 1011	(H'6B')

5.bit

24- BTFSC f,b : f registerin b. bitini test eder. Eger bu bit sıfır ise program akisi bir sonraki komutu atlar.

Örnek : BTFSC PORTA,1 : PORTA'nin 1. biti sıfır ise sıradaki komutu atlayarak bir sonraki komutu uygular. Eger PORTA'nin 1. biti 1 ise sıradaki komutu uygular.

PORTA içeriği H'1D'	0001 1101	
	BTFSC	PORTA,1
	GOTO	BUTON
	BSF	PORTB,0
		0001 1111
		PORTA içeriği H'1F'

1.bit

25- BTFSS f,b : f registerin b. bitini test eder. Eger bu bit 1 ise program akisi bir sonraki komutu atlar.

Örnek : BTFSS PORTA,1 : PORTA'nin 1. biti 1 ise sıradaki komutu atlayarak bir sonraki komutu uygular. Eger PORTA'nin 1. biti sıfır ise sıradaki komutu uygular.

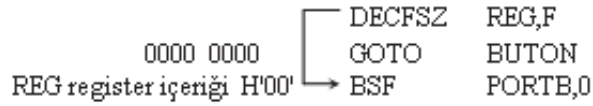
PORTA içeriği H'AA'	1010 1010	
	BTFSS	PORTA,1
	GOTO	BUTON
	BSF	PORTB,0
		0101 0101
		PORTA içeriği H'55'

1.bit

26- DECFSZ f,d : f registerin içeriğini 1 azaltır. Eger register içeriği sıfır ise bir sonraki komutu atlayarak sonucu W registre veya f registre yükler.

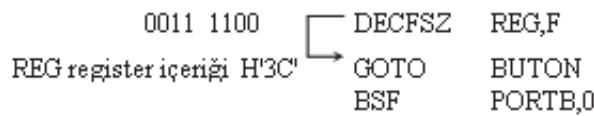
a) Eger d = 0 ise W registre yükler.

Örnek : DECFSZ REG,0 : REG = H'01' ise, REG-1 = H'00' olur ve register içeriği sıfır olduğundan sıradaki komut atlanarak bir sonraki komut isletilir ve sonuç W registerine yazilir.



b) Eger d = 1 ise f registre yükler.

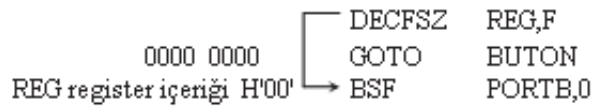
Örnek : DECFSZ REG,1 : REG = H'3D' ise, REG-1 = H'3C' olur ve register içeriği sıfır olmadığından sıradaki komut isletilerek sonuç REG registerine yazilir.



27- INCFSZ f, d : f registerin içeriğini 1 artırır. Eger register içeriği sıfır ise bir sonraki komutu atlayarak sonucu W registre veya f registre yükler.

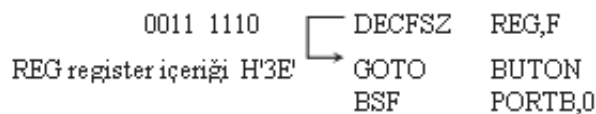
a) Eger d = 0 ise W registre yükler.

Örnek : DECFSZ REG,0 : REG = H'FF' ise, REG+1 = H'00' olur ve register içeriği sıfır olduğundan sıradaki komut atlanarak bir sonraki komut isletilir ve sonuç W registerine yazilir.



b) Eger d = 1 ise f registre yükler.

Örnek : DECFSZ REG,1 : REG = H'3D' ise, REG+1 = H'3E' olur ve register içeriği sıfır olmadığından sıradaki komut isletilerek sonuç REG registerine yazilir.



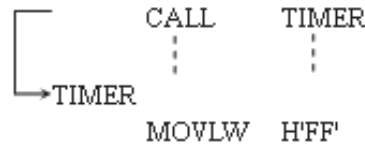
28- GOTO k : Program akisi k adresine dallanir.

Örnek : GOTO DONGU : Program akisi GOTO komutu ile sıradaki komutlari atlayarak DONGU etiketine dallanir.



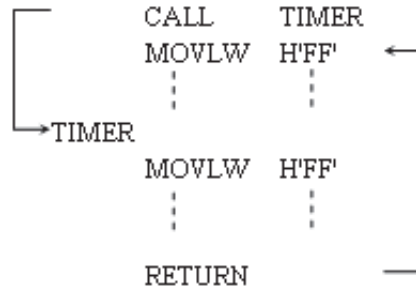
29- CALL k : Program akisi k etiketinin bulunduğu alt programa dallanir. Bu komut alt programlara dallanmak amacıyla kullanilir.

Örnek : CALL TIMER : Program akisi CALL komutuna geldiğinde devamındaki komutlari atlayarak TIMER etiketine dallanir.



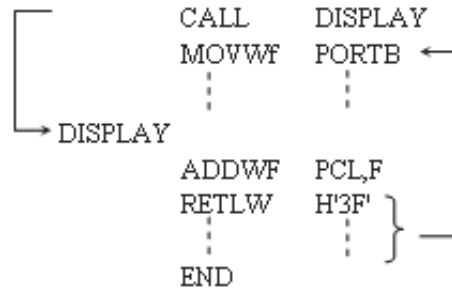
30- RETURN : Program akisini ana programa döndürür. Program akisi CALL komutu ile alt programa yönlendirildiğinde, alt programın sonuna yazılan RETURN komutu ana programa dönmeyi sağlar. Ana program kaldığı yerden devam eder.

Örnek : RETURN :



31- RETLW k : Program akisini RETURN komutunda olduğu gibi ana programa döndürür ancak ana programa dönüş esnasında W registerine k sabitini yükler.

Örnek : RETLW :



32- RETFIE : Program akisini Interrupt (kesme) alt programından ana programa döndürür.

33- NOP : Zaman geciktirme işlemi için kullanılır ve bir komut saykili süresince hiçbir işlem yapmaz.

34- CLRWDT : Watchdog Timer'i sıfırlar.

35- SLEEP : Mikrodenetleyiciyi uyuma moduna geçirir.

ASSEMBLER KOMUT DİREKTİFLERİ

Assembler kontrol direktifleri, programın işletilmesinde etkisi olmayan program akisinde zaman kazanmak, kaynak kodunu kontrol etmek ve direktif vermek için isteğe bağlı kullanılan komutlardır. Bu komutlardan sık kullanılan belli başlıları aşağıda verilmiştir.

DEFINE : Bir komut ve değişken dizininin program içerisinde birden fazla kullanılması durumunda tek bir isimle ifade edilmesini sağlar.

Örnek : #DEFINE BUTON PORTA,1

Örnekta BUTON, PORTA'nın 1. biti olarak tanımlanmıştır. Bu komut ile programın ilerleyen bölümünde PORTA'nın 1. biti ile ilgili işlemlerde direkt olarak BUTON ismi kullanılabilir. Böylece BSF BUTON gibi bir

komut satiri yazilabilir. Eger DEFINE komutu kullanilmaz ise ayni komut satirini `BSF PORTA,1` olarak yazmak gerekir. Ayni komutun program içerisinde birden fazla kullanilmasi durumunda DEFINE komutu program yazimini sadelestirir.

INCLUDE : Microchip firmasi her PIC mikrodenetleyicisi için kullanilmak üzere “Header file-Baslik dosyaları” denilen dosyalar düzenlemiştir. Bu dosyalar aracılığıyla PIC’in RAM belleğindeki özel register adreslerinin program yazimi asamasında bir çok defa girilmesi zorunluluğu ortadan kalkmıştır. Bunun yerine INCLUDE komutu kullanarak register adresleri otomatik olarak atanabilir. INCLUDE komutu çift tırnak içerisinde yer alan bir dosya ismi olarak tanımlandığında PIC için hazırlanmış sistem dosyalarını, eğer parantez şeklinde tanımlanmış ise kullanıcının sonradan geliştirdiği dosyaları içerir.

Örnek : INCLUDE “PIC16F84.INC” veya INCLUDE <MTN.h>

ORG : Programın PIC belleğinde depolandığı adresi tanımlar. Böylece programın hangi adresten başlayacağı veya kesme alt programlarının başlangıç adresleri belirlenebilir.

Örnek : ORG 0x00 : Programı 0x00 adresinden başlatır.

EQU : PIC bellek adreslerine atama yapmak için kullanılan etiket olarak isimlendirilen komuttur.

Örnek : Aşağıdaki komut satiri status registerinin PIC belleğinde H’03’ adresinde olduğunu tanımlar.

```
STATUS EQU H’03’
```

CONFIG : PIC kontrolünde osilatör tipi, Watchdog Timer, Power on Reset gibi koşulların belirlenmesini sağlayan komuttur. Bu komut satirindeki değişkenler PIC uygulama devresinin donanım özellikleri ve program içerisine bağlı olarak değişirler. CONFIG komutunun önünde iki alt çizgi bulunur, diğer değişkenlerin ise önlerinde birer alt çizgi ve aralarında birer boşluk ile & sembolü bulunur.

Örnek : __CONFIG __CP_OFF & __WDT_OFF & __PWRTE_ON & __XT_OSC

Konfigürasyon satirında PIC programlamada kullanılan koşullar aşağıda verilmiştir. Code Protect (CP) komutları PIC entegresinin programlanmasının ardından içerisindeki yazılımın okunarak kopyalanmasını önlemek için kullanılır. Watchdog Timer programı belirli zaman aralıklarında basa döndürür. Power on Reset, PIC’e enerji verilme durumunu (On/Off) kontrol eder.

- `__CP_ON` : Kod koruma açık.
- `__CP_OFF` : Kod koruma kapalı.
- `__WDT_ON` : Watchdog Timer açık.
- `__WDT_OFF` : Watchdog Timer kapalı.
- `__PWRTE_ON` : Power on reset açık.
- `__PWRTE_OFF` : Power on reset kapalı.
- `__XT_OSC` : Kristal osilatör.
- `__RC_OSC` : RC osilatör.
- `__LP_OSC` : Low Power osilatör.
- `__HS_OSC` : High Speed osilatör.