

KOMUT TABLOSU İLE İLGİLİ AÇIKLAMALAR:

1) Etkilenen Bayraklar (E.B.) : Bazı komutlar kořturulurken PSW saklayıcısındaki bayrakların değeri değışebilir. Herbir komut için etkilenen bayraklar belirtilmiřtir.

2) Byte: Mikrodenetleyicide kullanılan komutların herbiri farklı uzunluęa sahiptir. Herbir komut için uzunluk (Byte) değeri tabloda belirtilmiřtir.

Program Counter (PC) 'ın değeri her komuttan sonra kořturulan komutun "Byte" değeri kadar donanım tarafından otomatik olarak arttırılır.

Örneęin; "MOV A, R0" komutu 1-Byte'lık bir komuttur. Bu komut kořturulduktan sonra PC değeri 1 artırılır.

Yani; $PC \leftarrow PC + 1$ olur.

Benzer řekilde "MOV 40h, #35d" komutu 3-Byte'lık bir komuttur. Dolayısı ile bu komut icra edildikten sonra PC değeri;

$PC \leftarrow PC + 3$ řeklinde donanım tarafından otomatik olarak güncellenir.

3) Makine Çevrimi (M.Ç.) : Bir komutun kořturulması için kaç makine çevrimi geçtięini dolayısı ile komutun icrası için harcanan süreyi gösterir. Standart 8051'de (12 MHz kristal) bir makine çevrimi 12 osilatör periyodundan oluşur. Dolayısıyla standart 8051'de 1 Makina Çevrimi = $1\mu s$ 'dir.

Aduc841'de ise 1 Makina Çevrimi = 1 osilatör periyodu řeklindedir. Dolayısı ile 12 MHz Kristal ile çalıştırılan Aduc841 için 1 Makina Çevrimi = $1/12 = 0.0833\mu s$ 'dir.

4) Komutlarda; Rn ifadesi R0,R1, R2,R3, R4,R5, R6 ve R7 saklayıcılarını,

Ri ifadesi ise R0 ve R1 saklayıcılarını temsil etmektedir.

direct : Dahili RAM hafızanın doğrudan adreslenebilir alanlarının tümünü ifade etmektedir. Standart 8051'de alt 128-Byte hem doğrudan hem de dolaylı adreslenebilir. SFR bölgesinin ise sadece doğrudan adreslenebilir olduęu unutulmamalıdır.

Komut	Komut Kullanımı	Açıklama	Sembolik Gösterim	E.B.	Byte	M.Ç.
MOV	MOV A, Rn	Rn saklayıcısındaki değeri akümülatöre yükle	$(A) \leftarrow (Rn)$	-	1	1
	MOV A, direct	Adresteki değeri akümülatöre yükle	$(A) \leftarrow (\text{direct})$	-	2	1
	MOV A, #data	Sabit değeri akümülatöre yükle	$(A) \leftarrow \#data$	-	2	1
	MOV Rn, A	Akümlatörü Rn saklayıcısına yükle	$(Rn) \leftarrow (A)$	-	1	1
	MOV Rn, direct	Adresteki değeri Rn saklayıcısına yükle	$(Rn) \leftarrow (\text{direct})$	-	2	2
	MOV Rn, #data	Sabit değeri Rn saklayıcısına yükle	$(Rn) \leftarrow \#data$	-	2	1
	MOV direct, A	Akümlatördeki değeri adrese yükle	$(\text{direct}) \leftarrow (A)$	-	2	1
	MOV direct, Rn	Rn saklayıcısındaki değeri adrese yükle	$(\text{direct}) \leftarrow (Rn)$	-	2	2
	MOV direct1, direct2	Adres 2'deki değeri adres 1'e yükle	$(\text{direct1}) \leftarrow (\text{direct2})$	-	3	2
	MOV direct, #data	Sabit değeri direct adrese yükle	$(\text{direct}) \leftarrow \#data$	-	3	2
	MOV A, @Ri	Ri'nin gösterdiği adresteki değeri akümülatöre yükle	$(A) \leftarrow ((Ri))$	-	1	1
	MOV direct, @Ri	Ri'nin gösterdiği adresteki değeri direct adrese yükle	$(\text{direct}) \leftarrow ((Ri))$	-	2	2
	MOV @Ri, A	Akümlatörü Ri'nin gösterdiği adrese yükle	$((Ri)) \leftarrow (A)$	-	1	1
	MOV @Ri, direct	Adresteki değeri Ri'nin gösterdiği adrese yükle	$((Ri)) \leftarrow (\text{direct})$	-	2	2
	MOV @Ri, #data	Sabit değeri Ri'nin gösterdiği adrese yükle	$((Ri)) \leftarrow \#data$	-	2	1
MOVC	MOV DPTR, #data 16	16 bitlik sabit değeri DPTR'ye yükle	$DPH-DPL \leftarrow \#data15-1 - \#data7-0$	-	3	2
	MOV C, bit	Bit değerini elde bayrağına yükle	$(C) \leftarrow (\text{bit})$	C	2	1
	MOV bit, C	Elde bayrağındaki değeri bite yükle	$(\text{bit}) \leftarrow (C)$	C	2	1
MOVC	MOVC A, @A+DPTR	Program hafızanın A+DPTR ile gösterilen adresindeki değeri akümülatöre yükle	$(A) \leftarrow ((A) + (DPTR))$	-	1	2
	MOVC A, @A+PC	Program hafızanın A+PC ile gösterilen adresindeki değeri akümülatöre yükle	$(A) \leftarrow ((A) + (PC))$ $(PC) \leftarrow (PC) + 1$	-	1	2
MOVX	MOVX A, @Ri	Ri saklayıcısının gösterdiği harici RAM adresindeki değeri akümülatöre yükle	$(A) \leftarrow ((Ri))$	-	1	2
	MOVX @Ri, A	Akümlatördeki değeri Ri'nin gösterdiği harici RAM adresine yükle	$((Ri)) \rightarrow (A)$	-	1	2
	MOVX A, @DPTR	DPTR'nin gösterdiği harici RAM (16 bitlik) adresindeki değeri ACC'ye yükle	$(A) \leftarrow ((DPTR))$	-	1	2
	MOVX @DPTR, A	Akümlatördeki değeri DPTR'nin gösterdiği harici RAM adresine yükle	$((DPTR)) \leftarrow (A)$	-	1	2
PUSH POP	PUSH Direct	Direct adresteki değeri yığının (SP) gösterdiği adrese yükle	1. $(SP) \leftarrow (SP) + 1$ 2. $((SP)) \leftarrow (\text{direct})$	-	2	2
	POP Direct	Yığının gösterdiği adresteki bilgiyi direct adrese yükle	1. $(\text{direct}) \leftarrow ((SP))$ 2. $(SP) \leftarrow (SP) - 1$	-	2	2
SWAP	SWAP A	Akümlatörün ilk nibblesi ile ikinci nibblesini yer değiştir	$(A_{3-0}) \leftrightarrow (A_{7-4})$	-	1	1
NOP	NOP	(No operation) Herhangi bir işlem yapılmaz, program akışı 1 makine çevrimi kadar geciktirilmiş olur.	$(PC) \leftarrow (PC) + 1$	-	1	1

Komut	Komut Kullanımı	Açıklama	Sembolik Gösterim	E.B.	Byte	M.Ç.
XCH	XCH A, Rn	Rn ve akümülatörün içeriklerini değiştir	$(A) \leftrightarrow (Rn)$	-	1	1
	XCH A, direct	Adresteki değer ve akümülatörün içeriğini değiştir	$(A) \leftrightarrow (\text{direct})$	-	2	1
	XCH A, @Ri	Ri'nin gösterdiği adres ve akümülatörün içeriklerini değiştir	$(A) \leftrightarrow ((Ri))$	-	1	1
	XCHD A, @Ri	Ri'nin gösterdiği adres ve akümülatörün içeriklerinin ilk dört bitini değiştir	$(A_{3-0}) \leftrightarrow ((Ri_{3-0}))$	-	1	1
ANL	ANL A, Rn	Rn ile akümülatörü "VE" işlemine tabi tut	$(A) \leftarrow (A) \wedge (Rn)$	-	1	1
	ANL A, direct	Adresteki değer ile akümülatörü "VE" işlemine tabi tut	$(A) \leftarrow (A) \wedge (\text{direct})$	-	2	1
	ANL A, @Ri	Ri'nin gösterdiği adresteki değer ile akümülatörü "VE" işlemine tabi tut	$(A) \leftarrow (A) \wedge ((Ri))$	-	1	1
	ANL A, #data	Sabit değer ile akümülatörü "VE" işlemine tabi tut	$(A) \leftarrow (A) \wedge \#data$	-	2	1
	ANL direct, A	Adresteki değer ile akümülatörü "VE" işlemine tabi tut	$(\text{direct}) \leftarrow (\text{direct}) \wedge (A)$	-	2	1
	ANL direct, #data	Sabit değer ile adresteki değeri "VE" işlemine tabi tut	$(\text{direct}) \leftarrow (\text{direct}) \wedge \#data$	-	3	2
	ANL C, bit	Elde ile biti "VE" işlemine tabi tut	$(C) \leftarrow (C) \wedge (\text{bit})$	-	2	2
ORL	ANL C, /bit	Elde ile bitin tersini "VE" işlemine tabi tut	$(C) \leftarrow (C) \wedge /(\text{bit})$	-	2	2
	ORL A, Rn	Rn ile akümülatörü "VEYA" işlemine tabi tut	$(A) \leftarrow (A) \vee (Rn)$	-	1	1
	ORL A, direct	Adresteki değer ile akümülatörü "VEYA" işlemine tabi tut	$(A) \leftarrow (A) \vee (\text{direct})$	-	2	1
	ORL A, @Ri	Ri'nin gösterdiği adresteki değer ile akümülatörü "VEYA" işlemine tabi tut	$(A) \leftarrow (A) \vee ((Ri))$	-	1	1
	ORL A, #data	Sabit değer ile akümülatörü "VEYA" işlemine tabi tut	$(A) \leftarrow (A) \vee \#data$	-	2	1
	ORL direct, A	Adresteki değer ile akümülatörü "VEYA" işlemine tabi tut	$(\text{direct}) \leftarrow (\text{direct}) \vee (A)$	-	2	1
	ORL direct, #data	Sabit değer ile adresteki değeri "VEYA" işlemine tabi tut	$(\text{direct}) \leftarrow (\text{direct}) \vee \#data$	-	3	2
XRL	ORL C, bit	Elde ile biti "VEYA" işlemine tabi tut	$(C) \leftarrow (C) \vee (\text{bit})$	-	2	2
	ORL C, /bit	Elde ile bitin tersini "VEYA" işlemine tabi tut	$(C) \leftarrow (C) \vee /(\text{bit})$	-	2	2
	XRL A, Rn	Rn ile akümülatörü "ÖZEL VEYA" işlemine tabi tut	$(A) \leftarrow (A) \veebar (Rn)$	-	1	1
	XRL A, direct	Adresteki değer ile akümülatörü "ÖZEL VEYA" işlemine tabi tut	$(A) \leftarrow (A) \veebar (\text{direct})$	-	2	1
	XRL A, @Ri	Ri'nin gösterdiği adresteki değer ile akümülatörü "ÖZEL VEYA" işlemine tabi tut	$(A) \leftarrow (A) \veebar ((Ri))$	-	1	1
	XRL A, #data	Sabit değer ile akümülatörü "ÖZEL VEYA" işlemine tabi tut	$(A) \leftarrow (A) \veebar \#data$	-	2	1
CLR	XRL direct, A	Adresteki değer ile akümülatörü "ÖZEL VEYA" işlemine tabi tut	$(\text{direct}) \leftarrow (\text{direct}) \veebar (A)$	-	2	1
	XRL direct, #data	Sabit değer ile adresteki değeri "ÖZEL VEYA" işlemine tabi tut	$(\text{direct}) \leftarrow (\text{direct}) \veebar \#data$	-	3	2
	CLR A	Akümlatörü temizle	$(A) \leftarrow 0$	-	1	1
	CLR C	Elde bitini temizle (sıfırla)	$(C) \leftarrow 0$	C	1	1
	CLR bit	Biti temizle	$(\text{bit}) \leftarrow 0$	-	2	1
CPL	CPL A	Akümlatörü tersle	$(A) \leftarrow / (A)$	-	1	1
	CPL C	Elde bitini tersle	$(C) \leftarrow / (C)$	C	1	1
	CPL bit	Biti tersle	$(\text{bit}) \leftarrow / (\text{bit})$	-	2	1

Komut	Komut Kullanımı	Açıklama	Sembolik Gösterim	E.B.	Byte	M.Ç.
SETB	SETB C	Elde bitini birle (C=1)	$(C) \leftarrow 1$	C	1	1
	SETB bit	Bit değerini birle	$(\text{bit}) \leftarrow 1$	-	2	1
RL RLC RR RRC	RL A	Akümülatörü 1 bit sola döndür	$(A_{n+1}) \leftarrow (A_n), n = 0 - 6$ $(A_0) \leftarrow (A_7)$	-	1	1
	RLC A	Akümülatörü elde üzerinden 1 bit sola döndür	$(A_{n+1}) \leftarrow (A_n), n = 0 - 6$ $(A_0) \leftarrow (C)$ $(C) \leftarrow (A_7)$	C	1	1
	RR A	Akümülatörü 1 bit sağa döndür	$(A_n) \leftarrow (A_{n+1}), n = 0 - 6$ $(A_7) \leftarrow (A_0)$	-	1	1
	RRC A	Akümülatörü elde üzerinden 1 bit sağa döndür	$(A_n) \leftarrow (A_{n+1}), n = 0 - 6$ $(A_7) \leftarrow (C)$ $(C) \leftarrow (A_0)$	C	1	1
ACALL LCALL	ACALL addr11	Adres11 etiketli alt programı çağır	$(PC) \leftarrow (PC) + 2$ $(SP) \leftarrow (SP) + 1$ $(SP) \leftarrow (PC_{7-0})$ $(SP) \leftarrow (SP) + 1$ $(SP) \leftarrow (PC_{15-8})$ $(PC_{10-0}) \leftarrow \text{adres sayfası}$	-	2	2
	LCALL addr16	Adres 16 etiketli alt programı çağır	$(PC) \leftarrow (PC) + 3$ $(SP) \leftarrow (SP) + 1$ $((SP)) \leftarrow (PC_{7-0})$ $(SP) \leftarrow (SP) + 1$ $((SP)) \leftarrow (PC_{15-8})$ $(PC_{10-0}) \leftarrow \text{adres sayfası}$	-	3	2
	RET	Alt programdan çık, ana programda kaldığın yere dön	$(PC_{15-8}) \leftarrow ((SP))$ $(SP) \leftarrow (SP) - 1$ $(PC_{7-0}) \leftarrow ((SP))$ $(SP) \leftarrow (SP) - 1$	-	1	2
	RETI	Kesme alt programından çık, ana programda kaldığın yere dön	$(PC_{15-8}) \leftarrow ((SP))$ $(SP) \leftarrow (SP) - 1$ $(PC_{7-0}) \leftarrow ((SP))$ $(SP) \leftarrow (SP) - 1$	-	1	2

Komut	Komut Kullanımı	Açıklama	Sembolik Gösterim	E.B.	Byte	M.Ç.
KOŞULLUSUZ DALLANMA KOMUTLARI	SJMP <adres>	Kısa dallanma (adrese dallan) “+128 byte,-128 byte”lık alanda dallanma yapılabilir	$(PC) \leftarrow (PC) + 2$ $(PC) \leftarrow (PC) + \langle \text{adres} \rangle$	-	2	2
	AJMP <adres>	11 bitlik adres alanı içerisinde ($\pm 2KB$) dallanma sağlar	$(PC) \leftarrow (PC) + 2$ $(PC_{10-0}) \leftarrow \langle \text{adres} \rangle$	-	2	2
	LJMP <adres>	16 bitlik adres alanı (program hafızanın tamamı) içerisinde dallanma sağlar	$(PC) \leftarrow \langle \text{adres} \rangle_{15-0}$	-	3	2
	JMP @A+DPTR	A+DPTR’nin gösterdiği adrese dallan	$(PC) \leftarrow (A) + (DPTR)$	-	1	2
KOŞULLU DALLANMA KOMUTLARI	JC <adres>	Eğer C=1 ise adrese dallan	$(PC) \leftarrow (PC) + 2$ IF (C) = 1 THEN $(PC) \leftarrow (PC) + \langle \text{adres} \rangle$	-	2	2
	JC <adres>	Eğer C=1 ise adrese dallan	$(PC) \leftarrow (PC) + 2$ IF (C) = 1 THEN $(PC) \leftarrow (PC) + \langle \text{adres} \rangle$	-	2	2
	JNC <adres>	Eğer C=0 ise adrese dallan	$(PC) \leftarrow (PC) + 2$ IF (C) = 0 THEN $(PC) \leftarrow (PC) + \langle \text{adres} \rangle$	-	2	2
	JB bit, <adres>	Eğer bit=1 ise adrese dallan	$(PC) \leftarrow (PC) + 3$ IF (bit) = 1 THEN $(PC) \leftarrow (PC) + \langle \text{adres} \rangle$	-	3	2
	JNB bit, <adres>	Eğer bit=0 ise adrese dallan	$(PC) \leftarrow (PC) + 3$ IF (bit) = 0 THEN $(PC) \leftarrow (PC) + \langle \text{adres} \rangle$	-	3	2
	JBC bit, <adres>	Eğer bit=1 ise adrese dallan sonra biti sıfırla (bit=0)	$(PC) \leftarrow (PC) + 3$ IF (bit) = 1 THEN (bit) \leftarrow 0 $(PC) \leftarrow (PC) + \langle \text{adres} \rangle$	-	3	2
	JZ <adres>	Eğer akümülatör sıfır (A=0) ise adrese dallan	$(PC) \leftarrow (PC) + 2$ IF A = 0 THEN $(PC) \leftarrow (PC) + \langle \text{adres} \rangle$	-	2	2
	JNZ <adres>	Eğer A=0 değil ise adrese dallan	$(PC) \leftarrow (PC) + 2$ IF A \neq 0 THEN $(PC) \leftarrow (PC) + \langle \text{adres} \rangle$	-	2	2

Komut	Komut Kullanımı	Açıklama	Sembolik Gösterim	E.B.	Byte	M.Ç.
KOŞULLU DALLANMA KOMUTLARI	DJNZ Rn, <adres>	Rn'i bir azalt ve Rn sıfır değil ise adrese dallan	$(PC) \leftarrow (PC) + 2$ $(Rn) \leftarrow (Rn) - 1$ IF $((Rn) > 0 \vee (Rn) < 0)$ THEN $(PC) \leftarrow (PC) + <adres>$	-	2	2
	DJNZ direct, <adres>	Direct adresin değerini bir azalt, sıfır değil ise belirtilen adrese dallan	$(PC) \leftarrow (PC) + 2$ $(direct) \leftarrow (direct) - 1$ IF $((direct) > 0 \vee (direct) < 0)$ THEN $(PC) \leftarrow (PC) + <adres>$	-	3	2
	CJNE A, direct, <adres>	Akümülatör ile direct adres'deki değeri karşılaştır, eşit değilse belirtilen adrese dallan	$(PC) \leftarrow (PC) + 3$ IF $(A) < > (direct)$ THEN $(PC) \leftarrow (PC) + <adres>$ IF $(A) < (direct)$ THEN $(C) \leftarrow 1$ ELSE $(C) \leftarrow 0$	C	3	2
	CJNE A, #data, <adres>	Akümülatör ve sabit değeri karşılaştır, eşit değil ise direct adrese dallan	$(PC) \leftarrow (PC) + 3$ IF $(A) < > data$ THEN $(PC) \leftarrow (PC) + <adres>$ IF $(A) < data$ THEN $(C) \leftarrow 1$ ELSE $(C) \leftarrow 0$	C	3	2
	CJNE Rn, #data, <adres>	Rn ile değeri karşılaştır, eşit değil ise direct adrese dallan	$(PC) \leftarrow (PC) + 3$ IF $(Rn) < > data$ THEN $(PC) \leftarrow (PC) + <adres>$ IF $(Rn) < data$ THEN $(C) \leftarrow 1$ ELSE $(C) \leftarrow 0$	C	3	2
	CJNE @Ri, #data, <adres>	Ri'nin gösterdiği adresteki değer ile sabit değeri karşılaştır, eşit değil ise direct adrese dallan	$(PC) \leftarrow (PC) + 3$ IF $((Ri)) < > data$ THEN $(PC) \leftarrow (PC) + <adres>$ IF $((Ri)) < data$ THEN $(C) \leftarrow 1$ ELSE $(C) \leftarrow 0$	C	3	2

Komut	Komut Kullanımı	Açıklama	Sembolik Gösterim	E.B.	Byte	M.Ç.
INC	INC A	Akümülatörün değerini 1 arttır	$(A) \leftarrow (A) + 1$	-	1	1
	INC Rn	Saklayıcının değerini 1 arttır	$(Rn) \leftarrow (Rn) + 1$	-	1	1
	INC direct	Adresteki değeri 1 arttır	$(direct) \leftarrow (direct) + 1$	-	2	1
	INC @Ri	Ri saklayıcısının gösterdiği adresteki değeri 1 artır	$((Ri)) \leftarrow ((Ri)) + 1$	-	1	1
	INC DPTR	DPTR saklayıcısının değerini 1 arttır	$(DPTR) \leftarrow (DPTR) + 1$	-	1	2
DEC	DEC A	Akümülatörün değerini 1 azalt	$(A) \leftarrow (A) - 1$	-	1	1
	DEC Rn	Saklayıcının değerini 1 azalt	$(Rn) \leftarrow (Rn) - 1$	-	1	1
	DEC direct	Adresteki değeri 1 azalt	$(direct) \leftarrow (direct) - 1$	-	2	1
	DEC @Ri	Ri saklayıcısının gösterdiği adresteki değeri 1 azalt	$((Ri)) \leftarrow ((Ri)) - 1$	-	1	1
ADD	ADD A,Rn	Rn saklayıcı değerini akümülatöre ekle	$(A) \leftarrow (A) + (Rn)$	C,OV,AC	1	1
	ADD A,direct	Adresteki değeri akümülatöre ekle	$(A) \leftarrow (A) + (direct)$	C,OV,AC	2	1
	ADD A,@Ri	Saklayıcının gösterdiği adresteki değeri akümülatöre ekle	$(A) \leftarrow (A) + ((Ri))$	C,OV,AC	1	1
	ADD A,#data	Sabit değeri akümülatöre ekle	$(A) \leftarrow (A) + \#data$	C,OV,AC	2	1
ADDC	ADDC A,Rn	Rn saklayıcı değerini ve elde bitini (C) akümülatöre ekle	$(A) \leftarrow (A) + (C) + (Rn)$	C,OV,AC	1	1
	ADDC A,direct	Adresteki değeri ve elde bitini (C) akümülatöre ekle	$(A) \leftarrow (A) + (C) + (direct)$	C,OV,AC	2	1
	ADDC A,@Ri	Saklayıcının gösterdiği adresteki değeri ve elde bitini (C) akümülatöre ekle	$(A) \leftarrow (A) + (C) + ((Ri))$	C,OV,AC	1	1
	ADDC A,#data	Sabit değeri ve elde bitini (C) akümülatöre ekle	$(A) \leftarrow (A) + (C) + \#data$	C,OV,AC	2	1
SUBB	SUBB A, Rn	Akümülatörden saklayıcının değerini ve borcu (C) çıkart	$(A) \leftarrow (A) - (C) - (Rn)$	C,OV,AC	1	1
	SUBB A, direct	Akümülatörden adresteki değeri ve borcu (C) çıkart	$(A) \leftarrow (A) - (C) - (direct)$	C,OV,AC	2	1
	SUBB A, @Ri	Akümülatörden saklayıcının gösterdiği adresteki değeri ve borcu (C) çıkart	$(A) \leftarrow (A) - (C) - (Ri)$	C,OV,AC	1	1
	SUBB A, #data	Akümülatörden sabit değeri ve borcu (C) çıkart	$(A) \leftarrow (A) - (C) - (\#data)$	C,OV,AC	2	1
MUL	MUL AB	Akümülatördeki değeri ile B saklayıcısındaki değeri çarp. Çarpım sonucunun yüksek anlamlı byte'ını B saklayıcısına, düşük anlamlı byte'ını ise akümülatöre yaz.	$(A)_{7-0} \leftarrow (A) \times (B)$ $(B)_{15-8}$	C,OV	1	4
DIV	DIV AB	Akümülatördeki değeri B saklayıcısındaki değere böl. İşlem sonucunda bölüm değerini akümülatöre, kalan değerini B saklayıcısına yükle.	$(A)_{15-8} \leftarrow (A)/(B)$ $(B)_{7-0}$	C,OV	1	4
DA	DA A	Akümülatörü onluk tabana ayarla	IF $[(A_{3-0}) > 9] [(AC) = 1]$ THEN $(A_{3-0}) \leftarrow (A_{3-0}) + 6$ AND IF $[(A_{7-4}) > 9] [(C) = 1]$ THEN $(A_{7-4}) \leftarrow (A_{7-4}) + 6$	C	1	1

Oku-Değiştir-Yaz (Read-Modify-Write) Komutları

8051 komut kümesinde I/O portlarından (P0,P1,P2,P3) veri okuyan komutlardan bazıları (kullanımlarına bağlı olarak) porta ait tutucuyu (Latch) bazıları ise portun fiziksel pinlerindeki değeri okur. Tutucuyu okuyan komutlar ve **tutucunun okunduğu** örnek kullanımları aşağıda verilmiştir.

Komut	Açıklama
ANL	Lojik “VE” işlemi ANL P1, A
ORL	Lojik “VEYA” işlemi ORL P2, #55h
XRL	Lojik “ÖZEL-VEYA” işlemi XRL P3, #0AAh
JBC	Eğer bit=1 ise adrese dallan sonra biti sıfırla JBC P2.0, devam
CPL	Biti tersle CPL P3.2
INC	INC P2
DEC	DEC P3
DJNZ	DJNZ P2, git
MOV Px.y, C *	MOV P2.3, C
CLR Px.y *	CLR P3.6
SETB Px.y *	SETB P0.3

* Bu komutların icrasında ilk önce portun tutucuları (8-Bit’in tamamı) okunur, ilgili bitin değeri değiştirilir ve ardından oluşan yeni Byte değeri tutuculara yazılır.

Tabloda verilen komutların kullanımında eğer **hedef operand** herhangi bir port veya portun bir biti ise komutun koşturulmasında ilk önce tutucu değeri okunur, okunan veri üzerinde işlem yapılır ve oluşan yeni değer tekrar tutucuya yazılır. Bu işlemlerin hepsi donanım tarafından otomatik olarak yerine getirilir. Eğer port veya portun bir biti **kaynak operand** ise bu durumda komutun koşturulmasında portun fiziksel pinlerindeki değeri okunur.

Örneğin ANL komutunun “**ANL P1, A**” şeklindeki kullanımında P1 portu **hedef operand’dır**. Bu durumda komut koşturulurken donanım P1 portuna ait tutucuyu okur, akümülatördeki değeri ile lojik ve işlemine tabi tutar ve ardından lojik ve işleminin sonucunu P1 tutucularına yazar.

Fakat, ANL komutunun “**ANL A, P1**” şeklindeki kullanımında ise P1 portu **kaynak operand’dır**. Bu durumda komut koşturulurken donanım P1 portuna fiziksel pinleri okur akümülatördeki değeri ile lojik ve işlemine tabi tutar ve ardından lojik ve işleminin sonucu akümülatöre yazılır.