

## Kısaltılmış Komut Adı

## Komut Kodları

## VERİ AKTARMA KOMUTLARI

**MOV varış,kaynak** = Aktarmak

reg ← reg (2) , reg ← mem (8+EA) , mem ← reg (9+EA)

reg ← imm (10+EA) , mem ← imm (10+EA)

reg ← imm (4)

Aküm ← mem (10)

mem ← Aküm (10)

seg reg ← reg (2) , seg reg ← mem (8+EA)

reg ← seg reg (2) , mem ← seg reg (9+EA)

|             |   |     |     |             |       |              |
|-------------|---|-----|-----|-------------|-------|--------------|
| 1 0 0 0 1 0 | d | w   | mod | reg         | r / m |              |
| 1 1 0 0 0 1 | 1 | w   | mod | 0 0 0       | r / m | veri         |
| 1 0 1 1     | w | reg |     |             |       | w=1 ise veri |
| 1 0 1 0 0 0 | 0 | w   |     | adres-düşük |       | adres-yüksek |
| 1 0 1 0 0 1 | 1 | w   |     | adres-düşük |       | adres-yüksek |
| 1 0 0 0 1 1 | 1 | 0   | mod | 0           | reg   | r / m        |
| 1 0 0 0 1 1 | 0 | 0   | mod | 0           | reg   | r / m        |

**PUSH kaynak** = Yığına word itmek

reg(10) , mem (16+EA)

reg (10)

seg reg (10)

|             |     |       |     |       |       |  |
|-------------|-----|-------|-----|-------|-------|--|
| 1 1 1 1 1 1 | 1   | 1     | mod | 1 1 0 | r / m |  |
| 0 1 0 1     | 0   | reg   |     |       |       |  |
| 0 0 0       | reg | 1 1 0 |     |       |       |  |

**POP varış** = Yığından word çekmek

reg (8) , mem (17+EA)

reg (8)

seg reg (8)

|             |     |       |     |       |       |  |
|-------------|-----|-------|-----|-------|-------|--|
| 1 0 0 0 1 1 | 1   | 1     | mod | 0 0 0 | r / m |  |
| 0 1 0 1     | 1   | reg   |     |       |       |  |
| 0 0 0       | reg | 1 1 1 |     |       |       |  |

**XCHG varış,kaynak** = Yazmaç/Bellek içeriklerini birbiri ile değiştirmek

reg ↔ reg (4) / reg ↔ mem (17+EA)

reg ↔ Aküm (3)

|             |   |     |     |     |       |  |
|-------------|---|-----|-----|-----|-------|--|
| 1 0 0 0 1 1 | 1 | w   | mod | reg | r / m |  |
| 1 0 0 1     | 0 | reg |     |     |       |  |

**IN Aküm,port** = AL/AX' e Byte/word girişi (Giriş/Çıkış)

Aküm ← sabit port (10)

Aküm ← değişken port (DX) (8)

|             |   |   |  |      |  |  |
|-------------|---|---|--|------|--|--|
| 1 1 1 0 0 1 | 0 | w |  | port |  |  |
| 1 1 1 0 1 1 | 0 | w |  |      |  |  |

**OUT port,Aküm** = AL/AX 'den Byte/word çıkışı (Giriş/Çıkış)

sabit port ← Aküm (10)

değişken port (DX) ← Aküm (8)

|             |   |   |  |      |  |  |
|-------------|---|---|--|------|--|--|
| 1 1 1 0 0 1 | 1 | w |  | port |  |  |
| 1 1 1 0 1 1 | 1 | w |  |      |  |  |

**XLAT** = AL'deki baytı çevir AL ← [BX+AL] (11)**LEA varış,kaynak** = reg ← EA (2+EA)**LDS varış,kaynak** = DS:reg ← mem(DS:EA) (16+EA)**LES varış,kaynak** = ES:reg ← mem(ES:EA) (16+EA)**LAHF** = AH ← Bayraklar (4)**SAHF** = Bayraklar ← AH (4)**PUSHF** = yığın ← bayraklar (10)**POPF** = bayraklar ← yığın (8)

|             |   |   |     |     |       |  |
|-------------|---|---|-----|-----|-------|--|
| 1 0 0 0 1 1 | 0 | 1 | mod | reg | r / m |  |
| 1 1 0 0 0 1 | 0 | 1 | mod | reg | r / m |  |
| 1 1 0 0 0 1 | 0 | 0 | mod | reg | r / m |  |
| 1 0 0 1 1 1 | 1 | 1 |     |     |       |  |
| 1 0 0 1 1 1 | 1 | 0 |     |     |       |  |
| 1 0 0 1 1 1 | 0 | 0 |     |     |       |  |
| 1 0 0 1 1 1 | 0 | 1 |     |     |       |  |

## ARİTMETİK İŞLEM KOMUTLARI

**ADD varış,kaynak** = Toplama işlemi

reg ← reg+reg/mem (3/9+EA) , mem ← reg+mem (16+EA)

reg ← reg+imm (4) , mem ← mem+imm (17+EA)

Aküm ← Aküm+imm (4)

|             |   |   |   |     |       |                |
|-------------|---|---|---|-----|-------|----------------|
| 0 0 0 0 0 0 | 0 | d | w | mod | reg   | r / m          |
| 1 0 0 0 0 0 | 0 | s | w | mod | 0 0 0 | r / m          |
| 0 0 0 0 0 1 | 0 | w |   |     | veri  | sw=01 ise veri |
| 0 0 0 0 0 1 | 0 | w |   |     | veri  | w=1 ise veri   |

**ADC varış,kaynak** = Elde ile birlikte topla

reg ← reg+reg/mem+CF (3/9+EA) , mem ← mem+reg+CF (16+EA)

reg ← reg+imm+CF (4) , mem ← mem+imm+CF (17+EA)

Aküm ← Aküm+imm+CF (4)

|             |   |   |   |     |       |                |
|-------------|---|---|---|-----|-------|----------------|
| 0 0 0 1 0 0 | 0 | d | w | mod | reg   | r / m          |
| 1 0 0 0 0 0 | 0 | s | w | mod | 0 1 0 | r / m          |
| 0 0 0 1 0 1 | 0 | w |   |     | veri  | sw=01 ise veri |
| 0 0 0 1 0 1 | 0 | w |   |     | veri  | w=1 ise veri   |

**INC varış** = Artırmak

reg ← reg+1 (2) , mem ← mem+1 (15+EA)

reg ← reg+1 (2)

|             |   |     |     |       |       |  |
|-------------|---|-----|-----|-------|-------|--|
| 1 1 1 1 1 1 | 1 | w   | mod | 0 0 0 | r / m |  |
| 0 1 0 0     | 0 | reg |     |       |       |  |

**AAA** = Toplamadan sonra ASCII ayar (4)**DAA** = Toplamadan sonra Onluk ayar (4)

|             |   |   |  |  |  |  |
|-------------|---|---|--|--|--|--|
| 0 0 1 1 0 1 | 1 | 1 |  |  |  |  |
| 0 0 1 0 0 1 | 1 | 1 |  |  |  |  |

**SUB varış,kaynak** = Çıkarma işlemi

reg ← reg-reg/mem (3/9+EA) , mem ← mem-reg (16+EA)

reg ← reg-imm (4) , mem ← mem-imm (17+EA)

Aküm ← Aküm-imm (4)

|             |   |   |   |     |       |                |
|-------------|---|---|---|-----|-------|----------------|
| 0 0 1 0 1 0 | 0 | d | w | mod | reg   | r / m          |
| 1 0 0 0 0 0 | 0 | s | w | mod | 1 0 1 | r / m          |
| 0 0 1 0 1 1 | 0 | w |   |     | veri  | sw=01 ise veri |
| 0 0 1 0 1 1 | 0 | w |   |     | veri  | w=1 ise veri   |

**SBB varış,kaynak** = Ödünç ile birlikte çıkar

reg ← reg-reg/mem-CF (3/9+EA) , mem ← mem-reg-CF (16+EA)

reg ← reg-imm-CF (4) , mem ← mem-imm-CF (17+EA)

Aküm ← Aküm-imm-CF(4)

|             |   |   |   |     |       |                |
|-------------|---|---|---|-----|-------|----------------|
| 0 0 0 1 1 0 | 0 | d | w | mod | reg   | r / m          |
| 1 0 0 0 0 0 | 0 | s | w | mod | 0 1 1 | r / m          |
| 0 0 0 1 1 1 | 0 | w |   |     | veri  | sw=01 ise veri |
| 0 0 0 1 1 1 | 0 | w |   |     | veri  | w=1 ise veri   |

## Kısaltılmış Komut Adı

## Komut Kodları

**DEC varış** = Azaltmak

reg ← reg-1 (2) , mem ← mem-1 (15+EA)

reg ← reg-1 (2)

**NEG varış** = İşaretini değiştir (reg (3),mem (16+EA))**CMP varış,kaynak** = Karşılaştırmak

reg-reg/mem (3/9+EA) ,mem-reg (9+EA)

reg-imm (4) ,mem-imm (17+EA)

Aküm-imm (4)

**AAS** = Çıkarmadan sonra ASCII ayar (4)**DAS** = Çıkarmadan sonra Onluk ayar (4)**MUL kaynak** = Çarpmak (İşaretsiz) (8-bit (71+EA),16-bit (124+EA))**IMUL kaynak** = Çarpmak (İşaretsiz tamsayı) (8-bit (90+EA),16-bit(144+EA))**AAM** = Çarpmadan sonra ASCII ayar (83)**DIV kaynak** = Bölmek (İşaretsiz) (8-bit (90+EA),16-bit (155+EA))**IDIV kaynak** = Bölmek (İşaretsiz tamsayı) (8-bit(112+EA),16-bit(177+EA))**AAD** = Bölmeden önce ASCII ayar (60)**CBW** = byte(AL) → word(AX), dönüştür (2)**CWD** = word(AX) → Dword(DX:AX), dönüştür (5)**LOJİK İŞLEM KOMUTLARI****NOT varış** = Değil (reg (3),mem (16+EA))Aşağıda verilen kaydırma ve döndürme komutları için; yazmacı/belleği 1 bit kaydır/döndür (2)/(15+EA)  
yazmacı/belleği değişken bit sayısında (CL) kaydır/döndür (8+4/bit)/(20+EA+4/bit)**SHL/SAL varış,sayı** = Lojik/Aritmetik sola kaydır**SHR varış,sayı** = Lojik sağa kaydır**SAR varış,sayı** = Aritmetik sağa kaydır**ROL varış,sayı** = sola döndür**ROR varış,sayı** = sağa döndür**RCL varış,sayı** = elde ile beraber sola döndür**RCR varış,sayı** = elde ile beraber sağa döndür**AND varış,kaynak** = Lojik VE işlemi

reg ← reg • reg/mem (3/9+EA) ,mem ← mem • reg (16+EA)

reg ← reg • imm (4) ,mem ← mem • imm (17+EA)

Aküm ← Aküm • imm (4)

**TEST varış,kaynak** = Bayraklar için lojik VE işlemi

reg • reg/mem (3/9+EA) ,mem • reg (16+EA)

reg • imm (4) ,mem • imm (10+EA)

Aküm • imm (4)

**OR varış,kaynak** = Lojik VEYA işlemi

reg ← reg + reg/mem (3/9+EA) ,mem ← mem + reg (16+EA)

reg ← reg + imm (4) ,mem ← mem + imm (17+EA)

Aküm ← Aküm + imm (4)

**XOR varış,kaynak** = Lojik ÖZEL VEYA işlemi

reg ← reg ⊕ reg/mem (3/9+EA) ,mem ← mem ⊕ reg (16+EA)

reg ← reg ⊕ imm (4) ,mem ← mem ⊕ imm (17+EA)

Aküm ← Aküm ⊕ imm (4)

**DİZİ İŞLEM KOMUTLARI****REP/REPZ/REPZ** = Tekrar ön komutu (6)**MOVSB/MOVSW** = Byte/word aktarmak (17)**CMPB/CMPDW** = Byte/word karşılaştırmak (22)**SCASB/SCASW** = Byte/word taramak (15)**LDSB/LODSW** = Aküm ← mem (12)**STOSB/STOSW** = mem ← Aküm (10)**DENETİM AKTARMA KOMUTLARI****CALL prog\_adı** = Altprogram çağırmak

Parça içinden doğrudan çağırmak (11)

Parça içinden dolaylı çağırmak (13+EA)

|   |   |   |   |   |   |   |     |     |   |   |   |       |
|---|---|---|---|---|---|---|-----|-----|---|---|---|-------|
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | w   | mod | 0 | 0 | 1 | r / m |
| 0 | 1 | 0 | 0 | 1 |   |   | reg |     |   |   |   |       |
| 1 | 1 | 1 | 1 | 0 | 1 | 1 | w   | mod | 0 | 1 | 1 | r / m |

|   |   |   |   |   |   |   |   |     |      |       |              |       |
|---|---|---|---|---|---|---|---|-----|------|-------|--------------|-------|
| 0 | 0 | 1 | 1 | 1 | 0 | d | w | mod | reg  | r / m |              |       |
| 1 | 0 | 0 | 0 | 0 | 0 | s | w | mod | 1    | 1     | 1            | r / m |
| 0 | 0 | 1 | 1 | 1 | 1 | 0 | w |     | veri |       | w=1 ise veri |       |
| 0 | 0 | 1 | 1 | 1 | 1 | 1 |   |     |      |       |              |       |
| 0 | 0 | 1 | 0 | 1 | 1 | 1 |   |     |      |       |              |       |

|   |   |   |   |   |   |   |   |     |   |   |   |       |
|---|---|---|---|---|---|---|---|-----|---|---|---|-------|
| 1 | 1 | 1 | 1 | 0 | 1 | 1 | w | mod | 1 | 0 | 0 | r / m |
| 1 | 1 | 1 | 1 | 0 | 1 | 1 | w | mod | 1 | 0 | 1 | r / m |
| 1 | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 0   | 0 | 0 | 1 | 0     |
| 1 | 1 | 1 | 1 | 0 | 1 | 1 | w | mod | 1 | 1 | 0 | r / m |
| 1 | 1 | 1 | 1 | 0 | 1 | 1 | w | mod | 1 | 1 | 1 | r / m |
| 1 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0   | 0 | 0 | 1 | 0     |

|   |   |   |   |   |   |   |   |
|---|---|---|---|---|---|---|---|
| 1 | 0 | 0 | 1 | 1 | 0 | 0 | 0 |
| 1 | 0 | 0 | 1 | 1 | 0 | 0 | 1 |

|   |   |   |   |   |   |   |   |     |   |   |   |       |
|---|---|---|---|---|---|---|---|-----|---|---|---|-------|
| 1 | 1 | 1 | 1 | 0 | 1 | 1 | w | mod | 0 | 1 | 0 | r / m |
|---|---|---|---|---|---|---|---|-----|---|---|---|-------|

|   |   |   |   |   |   |   |   |     |   |   |   |       |
|---|---|---|---|---|---|---|---|-----|---|---|---|-------|
| 1 | 1 | 0 | 1 | 0 | 0 | v | w | mod | 1 | 0 | 0 | r / m |
| 1 | 1 | 0 | 1 | 0 | 0 | v | w | mod | 1 | 0 | 1 | r / m |
| 1 | 1 | 0 | 1 | 0 | 0 | v | w | mod | 1 | 1 | 1 | r / m |
| 1 | 1 | 0 | 1 | 0 | 0 | v | w | mod | 0 | 0 | 0 | r / m |
| 1 | 1 | 0 | 1 | 0 | 0 | v | w | mod | 0 | 0 | 1 | r / m |
| 1 | 1 | 0 | 1 | 0 | 0 | v | w | mod | 0 | 1 | 0 | r / m |
| 1 | 1 | 0 | 1 | 0 | 0 | v | w | mod | 0 | 1 | 1 | r / m |

|   |   |   |   |   |   |   |   |     |      |       |              |       |
|---|---|---|---|---|---|---|---|-----|------|-------|--------------|-------|
| 0 | 0 | 1 | 0 | 0 | 0 | d | w | mod | reg  | r / m |              |       |
| 1 | 0 | 0 | 0 | 0 | 0 | 0 | w | mod | 1    | 0     | 0            | r / m |
| 0 | 0 | 1 | 0 | 0 | 1 | 0 | w |     | veri |       | w=1 ise veri |       |

|   |   |   |   |   |   |   |   |     |      |       |              |       |
|---|---|---|---|---|---|---|---|-----|------|-------|--------------|-------|
| 1 | 0 | 0 | 0 | 1 | 0 |   | w | mod | reg  | r / m |              |       |
| 1 | 1 | 1 | 1 | 0 | 1 | 1 | w | mod | 0    | 0     | 0            | r / m |
| 1 | 0 | 1 | 0 | 1 | 0 | 0 | w |     | veri |       | w=1 ise veri |       |

|   |   |   |   |   |   |   |   |     |      |       |              |       |
|---|---|---|---|---|---|---|---|-----|------|-------|--------------|-------|
| 0 | 0 | 0 | 0 | 1 | 0 | d | w | mod | reg  | r / m |              |       |
| 1 | 0 | 0 | 0 | 0 | 0 | 0 | w | mod | 0    | 0     | 1            | r / m |
| 0 | 0 | 0 | 0 | 1 | 1 | 0 | w |     | veri |       | w=1 ise veri |       |

|   |   |   |   |   |   |   |   |     |      |       |              |       |
|---|---|---|---|---|---|---|---|-----|------|-------|--------------|-------|
| 0 | 0 | 1 | 1 | 0 | 0 | d | w | mod | reg  | r / m |              |       |
| 1 | 0 | 0 | 0 | 0 | 0 | 0 | w | mod | 1    | 1     | 0            | r / m |
| 0 | 0 | 1 | 1 | 0 | 1 | 0 | w |     | veri |       | w=1 ise veri |       |

|   |   |   |   |   |   |   |   |
|---|---|---|---|---|---|---|---|
| 1 | 1 | 1 | 1 | 0 | 0 | 1 | z |
| 1 | 0 | 1 | 0 | 0 | 1 | 0 | w |
| 1 | 0 | 1 | 0 | 0 | 1 | 1 | w |
| 1 | 0 | 1 | 0 | 1 | 1 | 1 | w |
| 1 | 0 | 1 | 0 | 1 | 1 | 0 | w |
| 1 | 0 | 1 | 0 | 1 | 0 | 1 | w |

|   |   |   |   |   |   |   |   |                 |   |   |                  |       |
|---|---|---|---|---|---|---|---|-----------------|---|---|------------------|-------|
| 1 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | yer değış-düşük |   |   | yer değış-yüksek |       |
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | mod             | 0 | 1 | 0                | r / m |

## Kısaltılmış Komut Adı

Parça dışından doğrudan çağırmak (20)

Parça dışından dolaylı çağırmak (29+EA)

**JMP hedef** = Şartsız dallanmak

Parça içinden doğrudan dallanmak (7)

Parça içinden doğrudan dallanmak - kısa (7)

Parça içinden dolaylı dallanmak (7+EA)

Parça dışından doğrudan dallanmak (7)

Parça dışından dolaylı dallanmak (16+EA)

**RET** = CALL ile çağrılardan dönmek

Parça içinden (8)

SP'ye imm. veri ekleyerek parça içinden (12)

Parça dışından (18)

SP'ye imm. veri ekleyerek parça dışından (17)

Aşağıda verilen koşullu dallanmalarda, dallanma olursa (8) / oluşmazsa (4)

**JE/JZ** = Eşit/Sıfır ise dallanmak (ZF=1)**JL/JNGE** = Küçük/Büyük-eşit değil ise ((SF xor OF) =1)**JLE/JNG** = Küçük-eşit/Büyük değil ise (((SP xor OF) or ZF)=1)**JB/JNAE** = Aşağıda/Yukarı-eşit değil ise (CF=1)**JBE/JNA** = Aşağı-eşit/Yukarıda değil ise ((CF or ZF)=1)**JP/JPE** = Parity var/Çift parity ise (PF=1)**JO** = Taşma var ise (OF=1)**JS** = İşaret var ise (SF=1)**JNE/JNZ** = Eşit değil/Sıfır değil ise dallanmak (ZF=0)**JNL/JGE** = Küçük değil/Büyük-eşit ise ((SF xor OF)=0)**JNLE/JG** = Küçük-eşit değil/Büyük ise (((SF xor OF) or ZF)=0)**JNB/JAE** = Aşağıda değil/Yukarı-eşit ise (CF=0)**JNBE/JA** = Aşağı-eşit değil/Yukarıda ise ((CF or ZF)=0)**JNP/JPO** = Parity yok/Tek parity ise (PF=0)**JNO** = Taşma yok ise (OF=0)**JNS** = İşaret yok ise (SF=0)**LOOP** = CX defa döngü (Dallanma Var/Yok=9/5)**LOOPZ/LOOPE** = Sıfır/Eşitken döngü (Dallanma Var/Yok=11/5)**LOOPNZ/LOOPNE** = Sıfır/Eşit değilken döngü (Dall.Var/Yok=11/5)**JCXZ** = CX sıfır ise dallanmak (Dallanma Var/Yok=9/5)**INT Kesme\_Tipi** =Yazılım ile kesme komutu

Özel tip (50)

Tip 3 (51)

**INTO** =Taşma var ise kesme (Taşma Var/Yok=52/4)**IRET** =Kesme servis programından dönmek (24)**İŞLEMCİ DENETİM KOMUTLARI****CLC** = Elde bayrağını temizlemek (2) (CF←0)**CMC** = Elde bayrağının tümleyenini almak (2) (CF← CF)**STC** = Elde bayrağını set etmek (2) (CF← 1)**CLD** = Yön bayrağını temizlemek (2) (DF← 0)**STD** = Yön bayrağını set etmek (2) (DF← 1)**CLI** = Kesme izin bayrağını temizlemek (2) (IF← 0)**STI** = Kesme izin bayrağını set etmek (2) (IF← 1)**HLT** = Durmak (2)**WAIT** = Beklemek (3)**LOCK** = Yol kilitleme ön eki (2)**NOP** = İşlem yok (3)**ESC** = Kaçmak (Dıştaki alete) (7+EA)

## Komut Kodları

|                 |               |                |
|-----------------|---------------|----------------|
| 1 0 0 1 1 0 1 0 | uzaklık-düşük | uzaklık-yüksek |
|                 | parça-düşük   | parça-yüksek   |
| 1 1 1 1 1 1 1 1 | mod 0 1 1     | r / m          |

|                 |                 |                  |
|-----------------|-----------------|------------------|
| 1 1 1 0 1 0 0 1 | yer değiş-düşük | yer değiş-yüksek |
| 1 1 1 0 1 0 1 1 | yer değiş       |                  |
| 1 1 1 1 1 1 1 1 | mod 1 0 0       | r / m            |
| 1 1 1 0 1 0 1 0 | uzaklık-düşük   | uzaklık-yüksek   |
|                 | parça-düşük     | parça-yüksek     |
| 1 1 1 1 1 1 1 1 | mod 1 0 1       | r / m            |

|                 |            |             |
|-----------------|------------|-------------|
| 1 1 0 0 0 0 1 1 |            |             |
| 1 1 0 0 0 0 1 0 | veri-düşük | veri-yüksek |
| 1 1 0 0 1 0 1 1 |            |             |
| 1 1 0 0 1 0 1 0 | veri-düşük | veri-yüksek |

|                 |           |
|-----------------|-----------|
| 0 1 1 1 0 1 0 0 | yer değiş |
| 0 1 1 1 1 1 0 0 | yer değiş |
| 0 1 1 1 1 1 1 0 | yer değiş |
| 0 1 1 1 0 0 1 0 | yer değiş |
| 0 1 1 1 0 1 1 0 | yer değiş |
| 0 1 1 1 1 0 1 0 | yer değiş |
| 0 1 1 1 0 0 0 0 | yer değiş |
| 0 1 1 1 1 0 0 0 | yer değiş |
| 0 1 1 1 0 1 0 1 | yer değiş |
| 0 1 1 1 1 1 0 1 | yer değiş |
| 0 1 1 1 1 1 1 1 | yer değiş |
| 0 1 1 1 0 0 1 1 | yer değiş |
| 0 1 1 1 0 1 1 1 | yer değiş |
| 0 1 1 1 1 0 1 1 | yer değiş |
| 0 1 1 1 0 0 0 1 | yer değiş |
| 0 1 1 1 1 0 0 1 | yer değiş |
| 1 1 1 0 0 0 1 0 | yer değiş |
| 1 1 1 0 0 0 0 1 | yer değiş |
| 1 1 1 0 0 0 0 0 | yer değiş |
| 1 1 1 0 0 0 1 1 | yer değiş |

|                 |     |
|-----------------|-----|
| 1 1 0 0 1 1 0 1 | tip |
| 1 1 0 0 1 1 0 0 |     |
| 1 1 0 0 1 1 1 0 |     |
| 1 1 0 0 1 1 1 1 |     |

|                 |                 |
|-----------------|-----------------|
| 1 1 1 1 1 0 0 0 |                 |
| 1 1 1 1 0 1 0 1 |                 |
| 1 1 1 1 1 0 0 1 |                 |
| 1 1 1 1 1 1 0 0 |                 |
| 1 1 1 1 1 1 0 1 |                 |
| 1 1 1 1 1 0 1 0 |                 |
| 1 1 1 1 1 0 1 1 |                 |
| 1 1 1 1 0 1 0 0 |                 |
| 1 0 0 1 1 0 1 1 |                 |
| 1 1 1 1 0 0 0 0 |                 |
| 1 0 0 1 0 0 0 0 |                 |
| 1 1 0 1 1 x x x | mod x x x r / m |

8086 komut tablosunda bulunan kısaltmaların açıklamaları

- reg / seg reg = yazmaç / parça yazmacı
- mem / imm = bellek / hemen adr. veri
- ( ) = saat olarak zamanlama
- AL = 8-bit akümülatör (Aküm)
- AX = 16-bit akümülatör (Aküm)
- BX = Taban yazmacı
- CX = Sayıcı yazmacı
- DX = Değişken port yazmacı
- SP = Yığın işaretçi yazmacı
- Yukarıda/Aşağıda işaretli değerler için kullanılır.
- Çok büyük = daha pozitif, Çok küçük = az pozitif (daha negatif) işaretli değerler.
- EA = Etkin Adres (işlenenin hesap sonucu bulunan mantıksal adresi)
- d = 1 ise '-e doğru, d = 0 ise '-den
- w = 1 ise word, w = 0 ise byte komut.

- BP = Taban işaretçi yazmacı
- SI = Kaynak dizin yazmacı
- DI = Varış dizin yazmacı
- IP = Komut işaretçi
- F = Durum bayrakları
- CS = Kod parça yazmacı
- DS = Veri parça yazmacı
- SS = Yığın parça yazmacı
- ES = Diğer parça yazmacı
- s:w = 01 ise 16-bit imm. işlenen
- s:w = 11 ise işareti 16-bit'e genişletilmiş 8-bit imm. işlenen.
- v = 0 ise "sayma" = 1 ; v = 1 ise "sayma" CL yazmacının belirlediği değerdir.
- x = dikkate alınmayacak.
- z = bazı dizi işlemlerinde sıfır bayrağıyla karşılaştırmada kullanılır.

Komut kodlarındaki ikinci baytın 2-bit'lik "mod" bölgesinin açıklama tablosu :

| mod | Açıklama                                                                                                                              |
|-----|---------------------------------------------------------------------------------------------------------------------------------------|
| 0 0 | Yer değiştirme yok. Özel durum : r/m = 110 ise 2-byte yer değiştirme var ve r/m adresleme modu ile belirlenen (EA) bellek adreslenir. |
| 0 1 | İşareti 16-bit'e genişletilmiş 1-byte işaretli yer değiştirme var.                                                                    |
| 1 0 | 2-byte işaretli yer değiştirme var.                                                                                                   |
| 1 1 | r/m, 'reg' alanı gibi davranır.                                                                                                       |

"reg" alan bit atamaları :

| reg | w = 0 | w = 1 | reg | Parça |
|-----|-------|-------|-----|-------|
| 000 | AL    | AX    | 00  | ES    |
| 001 | CL    | CX    | 01  | CS    |
| 010 | DL    | DX    | 10  | SS    |
| 011 | BL    | BX    | 11  | DS    |
| 100 | AH    | SP    |     |       |
| 101 | CH    | BP    |     |       |
| 110 | DH    | SI    |     |       |
| 111 | BH    | DI    |     |       |

Varsayılan parçayı değiştirmek için kullanılan komut öneki

**ES:** , **CS:** , **SS:** , **DS:** (2) 0 0 1 reg 1 1 0

Komut önekinin kullanımı

| İşlenen Yazmaç                       | Varsayılan | komut öneki ile |
|--------------------------------------|------------|-----------------|
| IP (kod adresi)                      | CS         | yok             |
| SP (yığın adresi)                    | SS         | yok             |
| BP (yığın adresi veya yığın markörü) | SS         | ES, CS, BP+DS   |
| SI veya DI (Dizi işlemi içermiyor)   | DS         | ES, SS, CS      |
| SI (Diziler için kaynak adresi)      | DS         | ES, SS, CS      |
| DI (Diziler için varış adresi)       | ES         | yok             |

İşlenen Adresleri (EA) zamanlaması (saat olarak) :

- Tek adresteki kelime işlenenleri için 4 saat ekle
- Imm. Yer deęiş. = 6
- Taban (BX, BP, SI, DI) = 5
- Taban + yer deęiş. = 9
- Taban + dizin (BP+DI, BX+SI) = 7
- Taban + dizin (BP+SI, BX+DI) = 8
- Taban + dizin (BP+DI, BX+SI) + yer deęiş. = 11
- Taban + dizin (BP+SI, BX+DI) + yer deęiş. = 12

Durum Bayrakları Yazmacı (F) :

|    |    |    |    |
|----|----|----|----|
| 15 | 8  | 7  | 0  |
| X  | X  | X  | X  |
| OF | DF | IF | TF |
| SF | ZF | X  | AF |
| X  | PF | X  | CF |

Komut kodlarındaki 3-bit'lik

"r/m" bölgesinin "mod"

bölgesine baęlı açıklaması :

| r/m | işlenen adres   |
|-----|-----------------|
| 000 | BX+SI+yer deęiş |
| 001 | BX+DI+yer deęiş |
| 010 | BP+SI+yer deęiş |
| 011 | BP+DI+yer deęiş |
| 100 | SI+yer deęiş    |
| 101 | DI+yer deęiş    |
| 110 | BP+yer deęiş*   |
| 111 | BX+yer deęiş    |

\* mod=00 ise EA

- CF = Elde bayrağı
- PF = Eşlik bayrağı
- AF = Yarım elde bayrağı
- ZF = Sıfır bayrağı
- SF = İşaret bayrağı
- TF = Tek adım bayrağı
- IF = Kesme bayrağı
- DF = Yön bayrağı
- OF = Taşma bayrağı