

## **EK.1:ASSEMBLER KOMUTLARININ LİSTESİ**

### **AAA - Ascii Adjust for Addition ( ASCII Toplama İçin Ayarla )**

Kullanım : AAA

Değiştirdiği Bayraklar: AF CF (OF,PF,SF,ZF tanımsız)

AL nin içeriğini geçerli bir ondalık sayıya çevirir. Yüksek değerli dörtlüğü sıfırlar.

<u>operandlar</u>	<u>Saat Darbesi</u>	<u>Büüklük (Byte)</u>
yok	8	1

### **AAD -Ascii Adjust for Division ( ASCII Bölme İçin Ayarla )**

Kullanım: AAD

Değiştirdiği Bayraklar: SF ZF PF (AF,CF,OF tanımsız)

Ondalık sayıların bölünmesinden önce kullanılır.AH 'ı 10 ile çarpar ve sonucu AL 'ye ekler. AH'ı sıfırlar.

<u>operandlar</u>	<u>Saat Darbesi</u>	<u>Büüklük (Byte)</u>
yok	60	2

### **AAM- Ascii Adjust for Multiplication ( ASCII Çarpma İçin Ayarla )**

Kullanım : AAM

Değiştirdiği Bayraklar:PF SF ZF (AF,CF,OF tanımsız)

İki ondalık sayının çarpılmasından sonra kullanılır.Bu komut kullanılmadan önce her byte'ın yüksek değerlikli dörtlüğü sıfırlanmış olması gerekir.

<u>operandlar</u>	<u>Saat Darbesi</u>	<u>Büüklük (Byte)</u>
yok	83	2

### **AAS- Ascii Adjust for Substraction ( ASCII çıkartma için ayarlar )**

Kullanım: AAS

Değiştirdiği Bayraklar: AF CF(OF,PF,SF,ZF tanımsız)

AL içindeki ,önceki ondalık sayı çıkartmasının sonucunu düzeltir. Yüksek değerlikli dörtlük'ü sıfırlar.

<u>operandlar</u>	<u>Saat Darbesi</u>	<u>Büüklük (Byte)</u>
yok	8	1

### **ADC -Add with Carry ( Eldeli Toplama )**

Kullanım : ADC hedef,kaynak

Değiştirdiği Bayraklar:AF CF OF SF PF ZF

İki binary operandı toplayarak sonucu hedefe yerleştirir. Eğer CF =1 ise , hedefe 1 eklenir.

<u>operandlar</u>	<u>Büüklük (Byte)</u>
reg,reg	2
hafıza,reg	2-4
reg,hafıza	2-4
reg,öncel	3-4
hafıza,öncel	3-6
accum-öncel	2-3

### **ADD -Arithmetic Addition ( Aritmetik Toplama )**

Kullanım: ADD hedef,kaynak

Değiştirdiği bayraklar: AF CF OF PF SF ZF

Hedefi, kaynağa ekler ve hedefin orijinal içeriğiyle değiştirir. İki operand da binary dir.

<u>operandlar</u>	<u>Büüklük (Byte)</u>
reg,reg	2

hafıza,reg	2-4
reg,hafıza	2-4
reg,öncel	3-4
hafıza,öncel	3-6
accum,öncel	2-3

### **AND -Logical And ( Mantıksal Ve )**

Kullanım : AND hedef, kaynak

Değiştirdiği Bayraklar:CF OF PF SF ZF (AF tanımsız)

İki operanda, mantıksal ve işlemi gerçekleştirerek hedefi sonuçla değiştirir.

operandlar                      Büyükklük (Byte)

reg,reg	2
hafıza,reg	2-4
reg,hafıza	2-4
reg,öncel	3-4
hafıza,öncel	3-6
accum,öncel	2-3

### **CALL-Procedure Call ( Alt Program Çağır )**

Kullanım: CALL hedef

Değiştirdiği Bayraklar:hiçbiri

IP'ı (komut göstergesi) yığın'a iterek , hedef adresi IP ye yükler. Kod CS:IP de görevine devam eder.

operandlar

rel16 (near,IPbağlı)
reg16(near,register dolaylı)
ptr16:16 (far,fullptr)
m16:16 (far,dolaylı)

### **CBW -Convert Byte to Word ( Byte'ı Word'e Değiştir )**

Kullanım:CBW

Değiştirdiği Bayraklar:hiçbiri

AL nin işaretini AH registerina doğru genişleterek ; AL içindeki byte'ı , AX içinde word'e çevirir.

<u>operandlar</u>	<u>Saat Darbesi</u>	<u>Büyükklük (Byte)</u>
yok	2	1

### **CLC -Clear Carry ( Elde'i Sil )**

Kullanım : CLC

Değiştirdiği Bayraklar: CF

Elde (CF=0) bayrağını temizler.

<u>operandlar</u>	<u>Saat Darbesi</u>	<u>Büyükklük (Byte)</u>
yok	2	1

### **CLD-ClearDirectionFlag ( Yön Bayrağını Sil )**

Kullanım: CLD

Değiştirdiği Bayraklar: DF

Yön bayrağını temizleyerek (DF=0) karakter katarı komutlarının; SI ve DI index registerlarının içeriğinin arttırmasına neden olur.

<u>operandlar</u>	<u>Saat Darbesi</u>	<u>Büyükklük (Byte)</u>
yok	2	1

### **CLI-Clear Interrupt Flag (disable) (Kesilim Bayrağını Sil)(engelle)**

Kullanım: CLI

Değiştirdiği Bayraklar: IF

Kesilim bayrağını temizleyerek (IF=0) maskelenebilir donanım interruptlarını engeller. NMI ve yazılım interruptları engellenemez.

<u>operandlar</u>	<u>Saat Darbesi</u>	<u>Büüklük (Byte)</u>
yok	2	1

### **CMC -Complement Carry Flag (Elde Bayrağının Tersini Al )**

Kullanım: CMC

Değiştirdiği Bayraklar: CF

Elde bayrağını tersyüz eder.

<u>operandlar</u>	<u>Saat Darbesi</u>	<u>Büüklük (Byte)</u>
yok	2	1

### **CMP -Compare ( Karşılaştır )**

Kullanım: CMP hedef,kaynak

Değiştirdiği Bayraklar:AF CF OF PF SF ZF

Hedefften kaynağı çıkarır ve bayrakları günceller fakat sonucu saklamaz.Bayraklar sonradan durumlarına göre kontrol edilir.

<u>operandlar</u>	<u>Büüklük (Byte)</u>
reg,reg	2
hafıza,reg	2-4
reg,hafıza	2-4
reg,öncel	3-4
hafıza,öncel	3-6
accum,öncel	4

### **CMPS -Compare String ( Karakter Katarlarını Karşılaştır )**

Kullanım: CMPS hedef,kaynak

CMPSB

CMPSW

Değiştirdiği Bayraklar:AF CF OF PF SF ZF

Hedefin değerini kaynaktan sonuçları saklamadan, çıkartır.Çıkartmaya dayalı bayrakları güncellenir ve yön ( D ) bayrağını durumuna göre SI ve DI index registerları azaltılır yada arttırılır. CMPSB index registerlarını 1 arttırır/azaltır , CMPSW 2 arttırır/azaltır, CMPSD 4 arttırır/azaltır. Bütün data parçaları işlerken REP öneki kullanılabilir.

<u>operandlar</u>	<u>Saat Darbesi</u>	<u>Büüklük (Byte)</u>
hedef,kaynak	22	1

### **CWD-Convert Word to Double word ( Word'ü iki Word' e Çevir )**

Kullanım: CWD

Değiştirdiği Bayraklar: hiçbir

AX registeri içindeki wordün işaretini DX registerına doğru genişleterek, DX:AX içinde bir doubleword değer oluşturur.

<u>operandlar</u>	<u>Saat Darbesi</u>	<u>Büüklük (Byte)</u>
yok	5	1

### **DAA -Decimal Adjust for Addition ( Ondalık Toplama İçin Ayarla )**

Kullanım: DAA

Değiştirdiği Bayraklar:AF CF ĞF SF ZF (OF tanımsız)

(AL nin içindeki) önceki BCD toplama işleminin sonuç değerini düzeltir. AL nin içeriği bir ondalık sayıya çevirilir.

<u>operandlar</u>	<u>Saat Darbesi</u>	<u>Büüklük (Byte)</u>
yok	4	1

### **DAS- Decimal Adjust for Substraction ( Ondalık Çıkarma İçin Ayarla )**

Kullanım: DAS

Değiştirdiği Bayraklar:AF CF PF SF ZF (OF tanımsız)

(AL nin içindeki) önceki BCD çıkartma işleminin sonuç değerini düzeltir. AL nin içeriği bir ondalık sayıya çevirilir.

<u>operandlar</u>	<u>Saat Darbesi</u>	<u>Büüklük (Byte)</u>
yok	4	1

### **DEC- Decrement ( Arttırma)**

Kullanım: DEC hedef

Değiştirdiği Bayraklar: AF OF PF SF ZF

Hedefin içeriği bir çıkartılır.

<u>operandlar</u>	<u>Büüklük (Byte)</u>
reg8	2
hafıza	2-4
reg16	1

### **DIV -Divide ( Böl )**

Kullanımı : DIV kaynak

Değiştirdiği Bayraklar : (AF,CF,OF,PF,SF,ZF tanımsız)

Accumulator'ün kaynağa işaretli binary bölme işlemi. Eğer kaynak (operand)bölen byte değerdeyse , AX kaynağa bölünür ve bölüm AL'ye ,kalan AH'a yerleştirilir. Eğer kaynak (operand) bölen bir word değerse , DX:AX kaynağa bölünür ve bölüm AX de, kalan DX de saklanır.

<u>operandlar</u>	<u>Büüklük (Byte)</u>
reg8	2
reg16	2
hafıza8	2
hafıza16	2-4

### **ESC -Escape**

Kullanım: ESC öncel,kaynak

Değiştirdiği Bayraklar: Hiçbiri

Matematik işlemciler için data bus'a erişim sağlar. CPU buna NOP gibi davranır fakat hafıza operandını bus'a yerleştirir.

<u>operandlar</u>	<u>Saat Darbesi</u>	<u>Büüklük (Byte)</u>
öncel,reg	2	2

### **HLT-Halt CPU ( Dur )**

Kullanım: HLT

Değiştirdiği Bayraklar: Hiçbiri

RESET satırı aktif olana kadar bilgisayarı durdurur,NMI ve maskelenebilir kesilim alınır.CPU etkin olmamaya başlar fakat ,daha sonra restart için, o anki CS:IP değerini tutar.

<u>operandlar</u>	<u>Saat Darbesi</u>	<u>Büüklük (Byte)</u>
-------------------	---------------------	-----------------------

yok

2

1

### **IDIV- Signed Integer Division (İşaretli Tamsayı bölme )**

Kullanımı: IDIV kaynak

Değiştirdiği Bayraklar:(AF,CF,OF,PF,SF,ZF tanımsız)

Akümülatörün kaynağa ,işaretli binary bölümü. Eğer kaynak bir byte değerse , AX kaynağa bölünür ve bölüm AL de kalan AH da saklanır. Eğer kaynak bir word değerse ,DX:AX kaynağa bölünür ve bölüm AL de kalan DX de saklanır.

operandlar                      Büüklük (Byte)

reg8	2
reg16	2
hafıza8	2-4
hafıza16	2-4

### **IMUL- Signed Multiply ( İşaretli Çarpma)**

Kullanım: IMUL kaynak

Değiştirdiği bayraklar: CF OF ( AF,PF,SF,ZF tanımsız)

Akümülatörün kaynakla ,işaretli çarpma işlemi,sonuç akümülatörde saklanır. Eğer kaynak operand bir byte değer ise ,AL ile çarpılır ve sonuç AX de saklanır.Eğer kaynak operand bir word değer ise; AX ile çarpılır ve sonuç DX:AX de saklanır.

operandlar                      Saat Darbesi                      Büüklük (Byte)

reg8	80-98	2
reg16	128-154	2
hafıza8	86-104	2-4
hafıza16	134-160	2-4

### **IN -Input Byte or Word From Port ( Port'dan Byte veya Word Oku)**

Kullanım: IN accum,port

Değiştirdiği Bayraklar:hiçbiri

Bir byte, veya word port'tan okunarak sırayla AL,veya AX 'e yerleştirilir.Eğer port numarası 0-255 arasında ise öncel olarak belirtilebilir, aksi takdirde port numarası DX içinde belirtilmelidir. PC de geçerli port numaraları 0-1024 dür, 65535'e kadar olan değerler üçüncü parti üreticiler ve PS/2 ler tarafından belirtilebilir ve tanınabilir.

operandlar                      Saat Darbesi                      Büüklük (Byte)

accum,öncel8	10/14	2
accum,DX	8/12	1

### **INC- Increment ( Arttır )**

Kullanım: INC hedef

Değiştirdiği Bayraklar: AF OF PF SF ZF

Hedef işaretsiz binary operanda 1 ekler.

operandlar                      Büüklük (Byte)

reg8	2
reg16	1
hafıza	2-4

### **INT-Interrupt ( Kesilim )**

Kullanım: INT sayı

Değiştirdiği Bayraklar: TF IF

Bayrakları iterek ,trap(kapan) ve kesilim bayraklarını temizleyerek (TF=0, IF=0) , yazılım kesilimini başlatır. CS nin itilmesini IP takip eder ve interrupt vektör tablosundan bulunan değer CS:IP 'ye yüklenir.Daha sonra yeni CS:IP tarafından adreslenen yeni konumdan uygulama başlar.

<u>operandlar</u>	<u>Saat Darbesi</u>	<u>Büyükklük (Byte)</u>
3(sabit)	52/72	2
öncel8	51/71	1

### **INTO -Interrupt on Overflow ( Taşma Olursa Kesilime Git )**

Kullanım: INTO

Değiştirdiği Bayraklar: IF TF

Eğer overflow (taşma) bayrağı O=1 ise bu komut INT 4 üretir. Bu 0000:0010 tan adreslenen kodun uygulanmasını sağlar.

<u>operandlar</u>	<u>Saat Darbesi</u>	<u>Büyükklük (Byte)</u>
Etiket:Dallan	53/73	1
Dallanma yok	4	

### **IRET/IRETD -Interrupt Return ( Kesilimden Deri Dön )**

Kullanım: IRET

Değiştirdiği Bayraklar: AF CF DF IF PF SF TF ZF

IP,CS yi ve sonra yığındaki bayrakları çekerek kesilimin kontrol noktasına geridöner ve bu noktadan uygulamaya devam eder.

<u>operandlar</u>	<u>Saat Darbesi</u>	<u>Büyükklük (Byte)</u>
iret	32/44	1

### **JA/JNBE -Jump Above / Jump not below or equal ( Üstünde / Altında veya Eşit Değilse Dallan )**

Kullanım: JA etiket

JNBE etiket

Değiştirdiği Bayraklar: Hiçbiri

Eğer Elde bayrağı ve sıfır bayrağının ikisi de sıfırsa ( CF=0, ZF=0) uygulamanın etikete dallanmasına neden olur. *İşaretsiz karşılaştırma.*

<u>operandlar</u>	<u>Saat Darbesi</u>	<u>Büyükklük (Byte)</u>
etiket: Dallan	16	2-4
Dallanma yok	4	

### **JAE/JNB-JumpAbove or Equal / Jump on not below (Üstünde veya Eşit /Altında Değilse Dallan )**

Kullanım: JAE etiket

JNB etiket

Değiştirdiği Bayraklar:hiçbiri

Eğer Elde ( CF=0) bayrağı sıfırsa uygulamanın etikete dallanmasına neden olur.Fonksiyonel olarak JNC ye benzer.*İşaretsiz karşılaştırma.*

<u>operandlar</u>	<u>Saat Darbesi</u>	<u>Büyükklük (Byte)</u>
etiket: Dallan	16	2-4
Dallanma yok	4	

### **JB/JNAE-Jump Below / Jump not Above or Equal (Altında / Üstünde veya Eşit Değilse Dallan )**

Kullanım: JB etiket

JNAE etiket

Değiştirdiği Bayraklar: hiçbiri

Elde bayrağı ( CF=1) ise ,uygulamanın etikete dallanmasına neden olur. Fonksiyonel olarak JC ye benzer. **İşaretsiz karşılaştırma.**

<u>operandlar</u>	<u>Saat Darbesi</u>	<u>Büüklük (Byte)</u>
etiket: Dallan	16	2-4
Dallanma yok	4	

### **JBE/JNA -Jump Below or Equal / Jump Not Above (Altında veya Eşit / Üstünde Değilse Dallan )**

Kullanım: JBE etiket

JNA etiket

Değiştirdiği Bayraklar: hiçbiri

Elde bayrağı veya Sıfır bayrağı (CF=1 veya ZF=1) ise uygulamanın etikete dallanmasına neden olur. **İşaretsiz karşılaştırma**

<u>operandlar</u>	<u>Saat Darbesi</u>	<u>Büüklük (Byte)</u>
etiket: Dallan	16	2-4
Dallanma yok	4	

### **JC –Jump on Carry (Elde ise Dallan )**

Kullanım: Jc etiket

Elde bayrağı bir (CF=1) ise uygulamanın etikete dallanmasına neden olur. **İşaretsiz karşılaştırma**

<u>operandlar</u>	<u>Saat Darbesi</u>	<u>Büüklük (Byte)</u>
etiket: Dallan	16	2-4
Dallanma yok	4	

### **JCXZ -Jump if Register CX is Zero ( Cx Registeri Sıfır İse Dallan )**

Kullanım: JCXZ etiket

Değiştirdiği Bayraklar:hiçbiri

Eğer CX registeri sıfır ise uygulamanın etikete dallanmasına neden olur. **İşaretsiz karşılaştırma.**

<u>operandlar</u>	<u>Saat Darbesi</u>	<u>Büüklük (Byte)</u>
etiket: Dallan	18	2
Dallanma yok	6	

### **JE/JZ -Jump Equal / Jump Zero ( Eşitse / Sıfır ise Dallan )**

Kullanım: JE etiket

JZ etiket

Değiştirdiği Bayraklar:hiçbiri

Eğer sıfır bayrağı (ZF=1) ise uygulamanın etikete dallanmasına neden olur.

**İşaretsiz karşılaştırma .**

<u>operandlar</u>	<u>Saat Darbesi</u>	<u>Büüklük (Byte)</u>
etiket: Dallan	16	2-4
Dallanma yok	4	

### **JG/JNLE -Jump Greater /Jump Not Less or Equal ( Büyükse / Küçük veya Eşit Değilse Dallan )**

Kullanım: JG etiket

JNLE etiket

Değiştirdiği Bayraklar: hiçbir

Eğer Sıfır bayrağı sıfır ise veya işaret bayrağı taşma bayrağına eşitse (ZF=0 veya SF=OF) ,uygulamanın etikete dallanmasına neden olur. **İşaretli karşılaştırma.**

<u>operandlar</u>	<u>Saat Darbesi</u>	<u>Büüklük (Byte)</u>
etiket: Dallan	16	2-4
Dallanma yok	4	

### **JGE / JNL -Jump Greater or Equal / Jump Not Less (Büyük veya Eşit / Küçük Değilse Dallan)**

Kullanım : JGE etiket

JNL etiket

Değiştirdiği Bayraklar:hiçbiri

Eğer işaret ve Taşma bayrakları eşitse (SF=OF) ,uygulamanın etikete dallanmasına neden olur. **İşaretsiz karşılaştırma.**

<u>operandlar</u>	<u>Saat Darbesi</u>	<u>Büüklük (Byte)</u>
etiket: Dallan	16	2-4
Dallanma yok	4	

### **JL/JNGE-Jump less / jump not greater or equal ( Küçükse / Büyük veya Eşit Değilse Dallan )**

Kullanım: JL etiket

JNGE etiket

Değiştirdiği Bayraklar:hiçbiri

Eğer İşaret bayrağı Taşma bayrağına eşit değilse (SF ≠ OF) ,uygulamanın etikete dallanmasına neden olur. **İşaretsiz karşılaştırma.**

<u>operandlar</u>	<u>Saat Darbesi</u>	<u>Büüklük (Byte)</u>
etiket : Dallan	16	2-4
Dallanma yok	4	

### **JLE/JNG- Jump Less or Equal / Jump not greater ( Küçük veya Eşit / Büyük Değilse Dallan)**

Kullanım: JLE etiket

JNG etiket

Değiştirdiği Bayraklar:hiçbiri

Eğer Sıfır bayrağı 1 ise veya işaret bayrağı overflow bayrağına eşit değilse (ZF=1 veya SF ≠ OF) uygulamanın etikete dallanmasına neden olur. **İşaretli karşılaştırma .**

<u>operandlar</u>	<u>Saat Darbesi</u>	<u>Büüklük (Byte)</u>
etiket: Dallan	16	2-4
Dallanma yok	4	

### **JMP -Unconditional Jump (Koşulsuz Dallanma )**

Kullanım: JMP hedef

Değiştirdiği Bayraklar:Hiçbiri

Şartsız olarak kontrolü etikete transfer eder. Dallanma standart olarak -32768 ile 32767 byte arasındaki adres değerlerinde olur.NEAR veya SHORT dallanma IP nin değiştirilmesine neden olurken , FAR dallanma CS ve IP nin değiştirilmesine neden olur.

<u>operandlar</u>	<u>Büüklük (Byte)</u>
-------------------	-----------------------



re18 (bağlı) 3  
re16 (bağlı) 3  
reg16 (near,register dolaylı) 5  
hafıza16 (near,hafıza dolaylı) 5

### **JNC-Jump Not Carry ( Elde Değilse Dallan )**

Kullanım: JNC etiket

Değiştirdiği Bayraklar:hiçbiri

Eğer Elde bayrağı sıfırsa (CF=0) ,uygulamanın etikete dallanmasına neden olur. Fonksiyonel olarak JAE ye ve JNB 'ye benzer. **İşaretsiz karşılaştırma.**

<u>operandlar</u>	<u>Saat Darbesi</u>	<u>Büyüklik (Byte)</u>
etiket: Dallan	16	2-4
Dallanma yok	4	

### **JNE/JNZ -Jump not equal / Jump not zero ( Eşit Değilse / Sıfır Değilse Dallan )**

Kullanım: JNE etiket

JNZ etiket

Değiştirdiği Bayraklar:hiçbiri

Eğer zero bayrağı sıfır ise (ZF=0) uygulamanın etikete dallanmasına neden olur.**İşaretsiz karşılaştırma.**

<u>operandlar</u>	<u>Saat Darbesi</u>	<u>Büyüklik (Byte)</u>
etiket: Dallan	16	2-4
Dallanma yok	4	

### **JNO -Jump not overflow (Taşma yoksa Dallan)**

Kullanım: JNO etiket

Değiştirdiği Bayraklar:hiçbiri

Eğer Taşma sıfır ise (OV=0), uygulamanın etikete dallanmasına neden olur.**İşaretli karşılaştırma.**

<u>operandlar</u>	<u>Saat Darbesi</u>	<u>Büyüklik (Byte)</u>
etiket: Dallan	16	2-4
Dallanma yok	4	

### **JNS -Jump Not Signed (İşaretli Değilse Dallan )**

Kullanım: JNS etiket

Değiştirdiği Bayraklar:hiçbiri

Eğer İşaret bayrağı sıfır ise ( SF=0), uygulamanın etikete dallanmasına neden olur.

**İşaretli karşılaştırma.**

<u>operandlar</u>	<u>Saat Darbesi</u>	<u>Büyüklik (Byte)</u>
etiket: Dallan	16	2-4
Dallanma yok	4	

### **JNP/JPO -Jump Not Parity / Jump Parity odd (Parit Yoksa dallan )**

Kullanım: JNP etiket

JPO etiket

Değiştirdiği Bayraklar: hiçbiri

Eğer parity bayrağı sıfır ise (PF=0) uygulamanın etikete dallanmasına neden olur. **İşaretsiz karşılaştırma.**

<u>operandlar</u>	<u>Saat Darbesi</u>	<u>Büyüklik (Byte)</u>
etiket: Dallan	16	2-4
Dallanma yok	4	

### **JO -Jump on overflow ( Taşma varsa Dallan )**

Kullanım: JO etiket

Değiştirdiği Bayraklar: hiçbiri

Taşma bayrağı (OF=1) işaretliyse , uygulamanın etikete dallanmasına neden olur. **İşaretli karşılaştırma.**

<u>operandlar</u>	<u>Saat Darbesi</u>	<u>Büyükük (Byte)</u>
etiket: Dallan	16	2-4
Dallanma yok	4	

### **JP/ JPE -Jump on Parity / Jump on Parity Even ( Parity varsa Dallan )**

Kullanım: JP etiket

JPE etiket

Değiştirdiği Bayraklar: hiçbiri

Eğer parity bayrağı işaretliyse , uygulamanın etikete dallanmasına neden olur. **İşaretsiz karşılaştırma.**

<u>operandlar</u>	<u>Saat Darbesi</u>	<u>Büyükük (Byte)</u>
etiket: Dallan	16	2-4
Dallanma yok	4	

### **JS -Jump Signed (İşaretli İse Dallan )**

Kullanım: JS etiket

Değiştirdiği Bayraklar: hiçbiri

Eğer işaret bayrağı işaretliyse (SF=1), uygulamanın etikete dallanmasına neden olur. **İşaretli karşılaştırma.**

<u>operandlar</u>	<u>Saat Darbesi</u>	<u>Büyükük (Byte)</u>
etiket: Dallan	16	2-4
Dallanma yok	4	

### **LAHF -Load Register AH From Flags ( Bayrak Registerlerinin İçerğini AH registerine kopyala)**

Kullanım: LAHF

Değiştirdiği Bayraklar: hiçbiri

Bayrak registerlarının 0-7 bitlerini AH'a kopyalar. Buna AF,CF,PF,SF ve ZF bayrakları dahildir, diğer bitler tanımsız.

<u>operandlar</u>	<u>Saat Darbesi</u>	<u>Büyükük (Byte)</u>
yok	4	1

### **LDS- Load Pointer Using DS ( DS Registeri Kullanarak Pointeri Yükle )**

Kullanım: LDS hedef,kaynak

Kaynak hafızadan 32-bit pointeri, hedef registera ve DS 'ye yükler. Offset hedef register'a ve segment DS 'ye yerleştirilir. Bu komutun kullanılması için; word ün düşük değerlikli hafızadaki adresinin offseti , word'ün yüksek değerlikli hafıza adresinin de segmenti içermesi gerekir. Bu yığından ve kesilim vector tablosundan uzak pointerların yüklenmesini kolaylaştırır.

<u>operandlar</u>	<u>Büyükük (Byte)</u>
reg16,hafıza32	2-4

### **LEA -Load Effective Address (Etkin Adresi Yükle )**

Kullanım: LEA hedef,kaynak

Değiştirdiği Bayraklar: hiçbiri

Kaynağın offset adresini , hedefe transfer eder.

<u>operandlar</u>	<u>Büyükük (Byte)</u>
-------------------	-----------------------

reg,hafıza

2-4

### **LES -Load Pointer Using ES (ES Registeri Kullanarak Pointeri Yükle )**

Kullanım: LES hedef,kaynak

Değiştirdiği Bayraklar:Hiçbiri

32-bit pointeri hafıza kaynağından hedef registre ve ES'e yükler. Offset hedef registre , ES segmentine yerleştirilir. Bu komutun kullanılması için düşük değerlikli hafıza adresindeki word'ün offseti, yüksek değerlikli adresteki word'ün segmenti içermesi gerekir. Bu yığından ve kesilim vector tablosundan uzak pointerların yüklenmesini kolaylaştırır.

<u>operandlar</u>	<u>Büyüklik (Byte)</u>
reg,hafıza	2-4

### **LOCK -Lock Bus (Bus' ı Kilitler )**

Kullanım: LOCK

Değiştirdiği Bayraklar:hiçbiri

Bu komut CPU'nun , sonraki komut işletilirken bus lock sinyali ileri sürmesine neden olan bir önektir. İki işlemcinin aynı data konumunu güncelleştirmesini engellemek için kullanılır.Bu sadece bus öncesi XCHG, MOV, IN ve OUT komutlarını kilitlemek için kullanılmalıdır.

<u>operandlar</u>	<u>Saat Darbesi</u>	<u>Büyüklik (Byte)</u>
yok	2	1

### **LODS -Load String (Byte , word or double) (Karakter Katarını Yükle )**

Kullanım: LODS kaynak

LODSB

LODSW

Değiştirdiği Bayraklar:hiçbiri

DS:SI tarafından adreslenen string elementi akümülatöre transfer eder. SI operandın büyüklüğüne ve kullanılan komuta dayandırılarak arttırılır. Eğer yönlendirme bayrağı işaretliyse (DF=1) ise SI azaltılır, eğer yönlendirme bayrağı sıfır ise (DF=0) SI arttırılır. REP öneki ile kullanılır.

<u>operandlar</u>	<u>Saat Darbesi</u>	<u>Büyüklik (Byte)</u>
etiket: Dallon	18	2
Dallonma yok	5	

### **LOOP -Decrement CX and Loop if CX Not Zero ( CX Registerini Azalt ve CX=0 Değilse Dön )**

Kullanım: LOOP etiket

Değiştirdiği Bayraklar: hiçbiri

CX'i 1 azaltır ve eğer CX sıfır değilse etikete dallanır. Etiket operandı , Loop komutunu takip eden komutun -128 veya 127 byte'ı içinde olmalıdır.

<u>operandlar</u>	<u>Saat Darbesi</u>	<u>Büyüklik (Byte)</u>
etiket: Dallon	18	2
Dallonma yok	5	

### **LOOPE / LOOPZ -Loop While Equal / Loop while zero ( Eşit ise / Sıfır ise Dön )**

Kullanım: LOOPE etiket

LOOPZ etiket

Değiştirdiği Bayraklar: hiçbiri

CX registeri 1 azaltır (bayrakları deęiřtirmeden) ve eęer  $CX \neq 0$  ise ve sıfır bayraęı iřaretliyse (ZF=1) etiketi dallandır. Etiket operandı , loop komutunu takip eden komutun -128 ya da 127 byte 'ı içinde olmalıdır.

<u>operandlar</u>	<u>Saat Darbesi</u>	<u>Büyükük (Byte)</u>
etiket: Dallanma	18	2
Dallanma yok	5	

### **LOOPNZ/LOOPNE -Loop While Not Zero / Loop While Not Equal (Sıfır Deęilse / Eřit Deęilse Dön)**

Kullanım: LOOPNZ etiket  
LOOPNE etiket

Deęiřtirdięi Bayraklar:hiçbiri

CX'i 1 azaltır (bayrakları deęiřtirmeden) ve eęer  $CX \neq 0$  ise ve sıfır bayraęı sıfır ise (ZF=0) etikete dallandır. Etiket operandı , loop komutunu takip eden komutun -128 ya da 127 byte 'ı içinde olmalıdır.

<u>operandlar</u>	<u>Saat Darbesi</u>	<u>Büyükük (Byte)</u>
etiket: Dallan	19	2
Dallanma yok	5	

### **MOV -Move Byte or Word (Byte veya Word Kopyala )**

Kullanım: MOV hedef,kaynak

Deęiřtirdięi Bayraklar: hiçbiri

Kaynak operanddan ,hedef operanda byte veya word kopyalar .

<u>operandlar</u>	<u>Büyükük (Byte)</u>
reg,reg	2
hafıza,reg	2-4
reg,hafıza	2-4
hafıza,öncel	3-6
reg,öncel	2-3
hafıza,accum	3
accum,hafıza	3
segreg,reg16	2
segreg,hafıza16	2-4
reg16,segreg	2
hafıza16,segreg	2-4

### **MOVS -Move String (Byte or Word) ( Byte veya Word Karakter Katarını Kopyala )**

Kullanım: MOVS hedef,kaynak

MOVSB

MOVSW

Deęiřtirdięi Bayraklar:hiçbiri

DS:SI tarafından adreslenen datayı (operandlar verilmiř olsa bile) ES:DI hedef konumuna kopyalar.

SI ve DI 'y1, kullanılan operandın veya komutun boyutuna göre ,günceller. SI ve DI yön bayrağı sıfırlandığında (DF=0) arttırılır ve yön registeri işaretli olduğunda (DF=1) da azaltılır. REP önekleri ile kullanılır.

<u>operandlar</u>	<u>Saat Darbesi</u>	<u>Büyükük (Byte)</u>
hedef,kaynak	18	1

### **MUL -Unsigned Multiply (İşaretsiz Çarpma )**

Kullanım: MUL kaynak

Değiştirdiği Bayraklar: CF OF (AF,PF,SF,ZF tanımsız)

Kaynağın akümülatör ile işaretsiz çarpımı. Eğer kaynak bir byte değer ise, AL diğer çarpılan olarak kullanılır ve sonuç AX'e yerleştirilir. Eğer kaynak bir word değer ise, AX kaynakla çarpılır ve sonuç DX:AX 'e gönderilir.

<u>operandlar</u>	<u>Büyükük (Byte)</u>
reg8	2
reg16	2
hafıza8	2-4
hafıza16	2-4

### **NEG -Two's complement negation ( 2' ye Göre Tümleyenini Al )**

Kullanım: NEG hedef

Değiştirdiği Bayraklar:AF CF OF PF SF ZF

Hedefi sıfırdan çıkartır ve hedefin 2ye tümleyeninin tekrar hedefe yükler.

<u>operandlar</u>	<u>Büyükük (Byte)</u>
reg	2
hafıza	2-4

### **NOP -No operation ( İşlem Yok )**

Kullanım: NOP

Bu hiçbirşey yapma komutudur.Zaman ve yer kazandırır ve en çok code segmentlerini birleştirmede kullanışlıdır.

<u>operandlar</u>	<u>Saat Darbesi</u>	<u>Büyükük (Byte)</u>
yok	3	1

### **NOT -One's Compliment Negation (Logical NOT) (1'e Göre Tümleyenini Al )**

Kullanım: NOT hedef

Değiştirdiği Bayraklar:hiçbiri

Yeni bir 1' göre tümleyenini oluşturarak hedef operandın bitlerinin tersini alır.

<u>operandlar</u>	<u>Büyükük (Byte)</u>
reg	2
hafıza	2-4

### **OR -Inclusive Logical OR ( Veya )**

Kullanım: OR hedef,kaynak

Değiştirdiği Bayraklar: CF OF PF SF ZF (AF tanımsız)

İki operandın mantıksal veya işlemini yapar.Sonucu hedef oprenda yükler.

<u>Operandlar</u>	<u>Büyükük (Byte)</u>
reg,reg	2
hafıza,reg	2-4
reg,hafıza	2-4
reg,öncel	3-4
hafıza8,öncel8	3-6

hafıza16,öncel16      3-6  
accum,öncel              2-3

### **OUT -Output Data to Port ( Porta Yaz )**

Kullanım: OUT port,accum

Değiştirdiği Bayraklar:hiçbiri

Belirli bir port adresine , AL deki byte'ı ,veya AX deki wordü eder.Eğer port numarası 0-255 arası ise öncel olarak belirtilebilir.255 den büyük port numaraları DX de belirtilmesi gerekir.

<u>operandlar</u>	<u>Saat Darbesi</u>	<u>Büyükük (Byte)</u>
öncel8,accum	10/14	2
DX,accum	8/12	1

### **POP -Pop Word off stack ( Yığında Word Çek )**

Kullanım: POP hedef

Değiştirdiği Bayraklar: hiçbiri

O anki yığının en üstündeki word' ü (SS:SP) hedefe transfer eder ve sonra SP' yi yeni yığın üst noktasına doğru iki artırır. CS registeri hedef olarak kullanılmaz.

<u>operandlar</u>	<u>Büyükük (Byte)</u>
reg16	2
segreg	1
hafıza16	2-4

### **POPF -Pop Flags off Stack (Yığında Bayrak Registerine Çek )**

Kullanım: POPF

Değiştirdiği Bayraklar: bütün bayraklar

Yığındaki word değeri bayrak registerlarına çeker ve sonra SP yi 2 arttırır.

<u>operandlar</u>	<u>Saat Darbesi</u>	<u>Büyükük (Byte)</u>
none	8/12	1

### **PUSH -Push Word onto Stack ( Yığına İt )**

Kullanım: PUSH kaynak

Değiştirdiği bayraklar:hiçbiri

SP yi operandın büyüküğü kadar azaltır ve kaynaktan yığın üst noktasına (SS:SP) bir word transfer eder.

<u>operandlar</u>	<u>Büyükük (Byte)</u>
reg16	1
hafıza16	2-4
segreg	1

### **PUSHF -Push Flags onto Stack ( Bayrak Registerinin İçeriğini Yığına İt )**

Kullanım: PUSHF

Değiştirdiği Bayraklar: hiçbiri

Bayrak registerlerini yığına'a transfer eder.

<u>operandlar</u>	<u>Saat Darbesi</u>	<u>Büyükük (Byte)</u>
yok	10/14	1

### **RCL -Rotate Through Carry Left ( Elde Bayrağı İle Sola Doğru Döndür )**

Kullanımı : RCL hedef ,sayım değeri

Değiştirdiği Bayraklar: CF OF

Hedefin bitlerini sola doğru (sayım değeri) kere döndürür. Ve bütün data sağdan tekrar girilerek sol tarafa itilir. Elde bayrağı döndürülen son biti tutar.

operandlar                      Büüklük (Byte)

reg,1	2
hafıza,1	2-4
reg,CL	2
hafıza,CL	2-4

### **RCR -Rotate Through Carry Right ( Elde Bayrağı İle Sağa Doğru Döndür )**

Kullanım: RCR hedef,sayım değeri

Değiştirdiği Bayraklar: CF OF

Hedefin bitlerini sağa doğru (sayım değeri) kere döndürür. Ve bütün data soldan tekrar girilerek sağ tarafa itilir. Elde bayrağı döndürülen son biti tutar.

operandlar                      Büüklük (Byte)

reg,1	2
hafıza,1	2-4
reg,CL	2
hafıza,CL	2-4

### **REP -Repeat String Operation ( Karakter Katarı Tekrarla Öneki )**

Kullanım: REP

Değiştirdiği Bayraklar: hiçbiri

Karakter Katarı komutlarının uygulamasını  $CX \neq 0$  olduğu sürece tekrarlar. Her karakter katari işleminden sonra , CX azaltılır ve Sıfır bayrağı test edilir.

operandlar                      Saat Darbesi                      Büüklük (Byte)

yok	2	1
-----	---	---

### **REPE/REPZ -Repeat Equal /Repeat Zero ( Eşit İse / Sıfır İse Tekrarla )**

Kullanım: REPE

REPZ

Değiştirdiği Bayraklar:hiçbiri

Karakter Katarı komutlarının uygulanmasını  $CX \neq 0$  ve Sıfır bayrağı bir ( ZF=1) olduğu sürece tekrar eder CX azaltılır ve Sıfır bayrağı her karakter katari işleminden sonra test edilir.

operandlar                      Saat Darbesi                      Büüklük (Byte)

yok	2	1
-----	---	---

### **REPNE/REPZ -Repeat not Equal/Repeat Not Zero ( Esit Değil ise / Sıfır değil İse Tekrarla )**

Kullanım: REPNE

REPZ

Değiştirdiği Bayraklar: hiçbiri

Karakter Katarı komutlarının uygulanmasını  $CX \neq 0$  ve Sıfır bayrağı sıfır olduğu ( ZF=0) sürece tekrar eder CX azaltılır ve Sıfır bayrağı her karakter katari işleminden sonra test edilir.

operandlar                      Saat Darbesi                      Büüklük (Byte)

yok	2	1
-----	---	---

### **RET/RETF -Return From Procedure ( Alt Proğramdan Geri Dön )**

Kullanım: RET nBytes

RETF nBytes

RETN nBytes

Değiştirdiği Bayraklar:hiçbiri

Alt programın kontrolünü yığın tarafından tutulan uygulama adresine geri yollar. "n byte" isteğe bağlı bir numaradır.

<u>operandlar</u>	<u>Saat Darbesi</u>	<u>Büyükük (Byte)</u>
retn	16/20	1
retn öncel	20/24	3
retf	26/34	1
retf öncel	25/33	3

### **ROL -Rotate Left ( Sola Döndür )**

Kullanım: ROL hedef,sayım değeri

Değiştirdiği Bayraklar: CF OF

Hedefin bitlerini sola doğru (sayım değeri) kere döndürür. Ve bütün data sağdan tekrar girilerek sol tarafa itilir. Elde bayrağı son döndürülen bitin değerini içerir.

<u>operandlar</u>	<u>Büyükük (Byte)</u>
reg,1	2
hafıza,1	2-4
reg,CL	2
hafıza,CL	2-4

### **ROR -Rotate Right ( Sağa Döndür )**

Kullanım: ROR hedef , sayım değeri

Değiştirdiği Bayraklar: CF OF

Hedefin bitlerini sağa doğru (sayım değeri) kere döndürür. Ve bütün data soldan tekrar girilerek sağ tarafa itilir. Elde bayrağı son döndürülen bitin değerini içerir.

<u>operandlar</u>	<u>Büyükük (Byte)</u>
reg,1	2
hafıza,1	2-4
reg,CL	2
hafıza,CL	2-4

### **SAHF -Store AH Register into Flags ( AH Registerinin İçeriğini Bayrak Registerine Yükle )**

Kullanım: SAHF

Değiştirdiği Bayraklar: AF CF PF SF ZF

AH nın 0-7 bitlerini bayrak registerina transfer eder. Bu AF,CF,PF,SF ve ZF yi içerir.

<u>operandlar</u>	<u>Saat Darbesi</u>	<u>Büyükük (Byte)</u>
yok	4	1

### **SAL/SHL -Shift Arithmetic Left/Shift Logical Left ( Aritmetik / Mantıksal Sola Kaydır )**

Kullanım: SAL hedef,sayım değeri

SHL hedef, sayım

Değiştirdiği Bayraklar:CF OF PF SF ZF (AF tanımsız)

Hedefi (sayım değeri) kadar bit sola kaydırır, sağdan sıfır girer. Elde bayrağı en son değiştirilen biti içerir.

<u>operandlar</u>	<u>Büyükük (Byte)</u>
reg,1	2
hafıza,1	2-4



reg,CL 2  
hafıza,CL 2-4

### **SAR -Shift Arithmetic Right ( Aritmetik Sağa Kaydır )**

Kullanım: SAR hedef,sayım değeri

Değiştirdiği Bayraklar: CF OF PF SF ZF (AF tanımsız)

Hedefi sağa (sayım değeri) bit kadar kaydırır, o anki işaret biti en sol bite kopyalanır. Elde bayrağı en son değiştirilen biti içerir.

operandlar Büyükük (Byte)

reg,1 2  
hafıza,1 2-4  
reg,CL 2  
hafıza,CL 2-4

### **SBB -Substract with Borrow ( Ödünç Alarak Çıkart )**

Kullanım: SBB hedef,kaynak

Değiştirdiği Bayraklar: AF CF OF PF SF ZF

Hedeften kaynağı çıkartır, eğer Elde bayrağı işaretliyse ( CF=1) ilave olarak 1 çıkartır. Sonuçlar hedefde saklanır.

operandlar Büyükük (Byte)

reg,reg 2  
hafıza,reg 2-4  
reg,hafıza 2-4  
reg,öncel 3-4  
hafıza,öncel 3-6  
accum,öncel 2-3

### **SCAS -Scan String (byte,word or doubleword) ( Karakter Katarı Ara )**

Kullanım: SCAS karakter katarı

SCASB

SCASW

Değiştirdiği Bayraklar: AF CF OF PF SF ZF

Akümülatörden ES:DI daki değeri karşılaştırır.ve çıkartmadakine benzer şekilde bayrakları

işaretler. komut formatına ve yönlendirme bayrağının konumuna göre DI artırılır/azaltılır .REP önekleriyle kullanılır.

operandlar Saat Darbesi Büyükük (Byte)

karakter katarı 15 1

### **SHR -Shift Logical right ( Mantıksal Sağa Kaydır )**

Kullanım: SHR hedef,sayım değeri

Değiştirdiği Bayraklar: CF OF PF SF ZF (AF tanımsız)

Hedefi (sayım değeri) bit kadar sağa kaydırır ve sıfırlar soldan girer. Elde bayrağı son değiştirilen biti içerir.

operandlar Büyükük (Byte)

reg,1 2  
hafıza,1 2-4  
reg,CL 2  
hafıza,CL 2-4

### **STC -Set Carry ( Elde Bayrağını Bir'le )**

Kullanım: STC

Değiştirdiği Bayraklar: CF

Elde bayrağını 1 yapar.

<u>operandlar</u>	<u>Saat Darbesi</u>	<u>Büüklük (Byte)</u>
yok	2	1

### **STD -Set Direction Flag ( Yön Bayrağını Bir'le )**

Kullanım: STD

Değiştirdiği Bayraklar:DF

Yönlendirme bayrağını 1 yaparak ,Karakter katarı komutlarının SI ve DI nin otomatik olarak arttırılması yerine otomatik azaltılmasına neden olur.

<u>operandlar</u>	<u>Saat Darbesi</u>	<u>Büüklük (Byte)</u>
yok	2	1

### **STI -Set Interrupt Flag (enable interrupts) ( Kesilim Bayrağını Bir'le – kesilimi etkin yap )**

Kullanım: STI

Değiştirdiği Bayraklar: IF

Kesilim bayrağını 1 yapar ve CPU donanım interruptlarının tümünün tanınmasını sağlar.

<u>operandlar</u>	<u>Saat Darbesi</u>	<u>Büüklük (Byte)</u>
yok	2	1

### **STOS -Store String(Byte,word or doubleword) ( Karakter Katarını Sakla )**

Kullanım: STOS hedef

STOSB

STOSW

STOSD

Değiştirdiği Bayraklar: hiçbiri

Akümülatördeki değeri ES:DI konumunda saklar (operand verilse bile). DI operandın

büüklüğüne (veya komut formatına) ve yönlendirme bayrağının durumuna göre arttırılır/azaltılır.

REP önekiyle kullanılır.

<u>operandlar</u>	<u>Saat Darbesi</u>	<u>Büüklük (Byte)</u>
hedef	11	1

### **SUB -Substract ( Çıkart )**

Kullanım: SUB hedef, kaynak

Değiştirdiği Bayraklar: AF CF OF PF SF ZF

Kaynak hedeften çıkartılır ve sonuç hedefte tutulur.

<u>operandlar</u>	<u>Büüklük (Byte)</u>
reg,reg	2
hafıza,reg	2-4
reg,hafıza	2-4
reg,öncel	3-4
hafıza,öncel	3-6
accum,öncel	2-3

### **TEST -Test for Bit Pattern ( Bit Pattreni Test Et )**

Kullanım: TEST hedef,kaynak

Değiştirdiği Bayraklar: CF OF PF SF ZF (AF tanımsız)

İki operanda mantıksal ve uygular ve bayrak registerini günceller, sonuçları saklamaz.

<u>operandlar</u>	<u>Büüklük (Byte)</u>
-------------------	-----------------------

reg,reg	2
reg,hafıza	2-4
hafıza,reg	2-4
reg,öncel	3-4
hafıza,öncel	3-6
accum,öncel	2-3

### **WAIT -Event wait ( Bekle )**

Kullanım: WAIT

Değiştirdiği Bayraklar: hiçbiri

Matematik işlemcinin işlemin bittiğini işaret edene kadar CPU bekleme konumuna geçer .Bu komut geçici olarak Matematik işlemci tarafından kullanılan hafızaya CPU nun erişimini önler.

<u>operandlar</u>	<u>Saat Darbesi</u>	<u>Büyükük (Byte)</u>
yok	4	1

### **XCHG -Exchange ( Değiş tokuş Et )**

Kullanım: XCHG hedef,kaynak

Değiştirdiği Bayraklar: hiçbiri

Kaynak ve hedefin içeriklerini birbiriyle değiştirir.

<u>operandlar</u>	<u>Büyükük (Byte)</u>
reg,reg	2
hafıza,reg	2-4
reg,hafıza	2-4
accum,reg	1
reg,accum	1

### **XLAT/XLATB –Translate ( Çevir )**

Kullanım: XLAT çeviri tablosu

Değiştirdiği Bayraklar: hiçbiri

AL içerisindeki byte'ı BX tarafından adreslenen bir kullanıcı tablosundaki byte ile değiştirir.AL nin orjinal değeri çeviri tablosuna indextir.

<u>operandlar</u>	<u>Saat Darbesi</u>	<u>Büyükük (Byte)</u>
tablo	11	1

### **XOR -Exclusive OR ( EXOR Özel Veya )**

Kullanım: XOR hedef,kaynak

Değiştirdiği Bayraklar: CF OF PF SF ZF (AF tanımsız)

Operandlar arasında Exor işlemi yapar ve sonucu hedefte saklar.

<u>operandlar</u>	<u>Büyükük (Byte)</u>
reg,reg	2
hafıza,reg	2-4
reg,hafıza	2-4
reg,öncel	3-4
hafıza,öncel	3-6
accum,öncel	2-3

## **EK.2:INTERRUPTLAR**

### **BIOS Interrupt INT 13H**

#### **Fonksiyon 00h – (Disket/Hard Disk) Başlangıç Durumuna Getirir:**

Bu fonksiyon floppy denetçisini ve sürücüsünü başlangıç durumuna getirir ve sonunda o anki fonksiyonu durdurur. Bu fonksiyonun tamamlanmasından sonra denetçi ve sürücü iyi tanımlanmış durumda olur.

---

Register:	Çağrı Değeri:	Dönen Değer:
AH	00h	hata kodu <sup>1)</sup>
DL	sürücü <sup>2)</sup>	
Elde		hata varsa

1) bkz F-2  
2) Floppy disk sürücüsü : 00h

---

#### **Fonksiyon 01h: En son Disket ya da Hard Disk İşleminin durumunu okur. (Hata Kodu)**

Bu fonksiyon en son sabit disk ya da disket sürücüsü işleminin sonlandırma durumunu belirtir. Bu durum kodu genellikle işlemin sonlandırılmasından sonra ah ile aynı formatta Register'ın içine geri döner. Bu fonksiyon bir işlemin tamamlanmasından sonraki durumu belirtmek istemiyorsanız yararlıdır. Ve durum byte'ıyla birlikte ah'ın içeriği zaten başka bir talimatla yok edilmiştir.

---

Register:	Çağrı Değeri:	Dönen Değer:
AH	01h	hata kodu <sup>1)</sup>
DL	sürücü <sup>2)</sup>	
Elde		hata varsa

1) bkz F-2  
2) Disket sürücüsü : 00h

---

#### **Fonksiyon 02h – Sektörleri Okur (Disket / Hard Disk)**

Disket / Hard Disk'in okuma tamponunun içinden bir ya da daha fazla sektör okunur. Tampon okuma sektörleri barındıracak kadar uzun olmalıdır. Bu durum olmazsa, 02h fonksiyonu veriyi ana belleğin üzerine yazar; ve sonuçta sistem çöker.

---

Register:	Çağrı Değeri:	Dönen Değer:
AH	02h	hata kodu <sup>1)</sup>
AL	okunacak sektörün numarası	
CH	iz/silindir	
CL	sektör	
DH	kafa	
DL	sürücü	
ES	okuma tamponunun segmenti	
BX	okuma tamponunun ofseti	
Elde		hata varsa

2) bkz F-2

---

#### **Fonksiyon 03h – Sektörleri Yazar (Disket / Hard Disk)**

Bu fonksiyon ana belleğin yazma tamponundan disket ya da Hard Disk'in üzerinde bir ya da daha fazla sektör yazar. Bu tampon yazılabilecek bütün verileri içerir. Bu veri transferi yalnızca 512 byte ile olur. Eğer tamponunuz kısmen yazma verisiyle dolmuşsa 03h fonksiyonun değerini aktarır. Bütün sektörler tüm register'ların programlanana kadar girintisiz veri de disk üzerindedir.

Register:	Çağrı Değeri:	Dönen Değer:
AH	03h	hata kodu <sup>1)</sup>
AL	okunacak sektörün numarası	
CH	iz/silindir	
CL	sektör	
DH	kafa	
DL	sürücü	
ES	okuma tamponunun segmenti	
BX	okuma tamponunun ofseti	
Elde		hata varsa
2) bkz F-2		

### **Fonksiyon 04h – Sektörleri Doğrular (Disket/ HD)**

Bu fonksiyon ana belleğin içindeki doğrulama tamponunun içeriği ile disket ya da hard disk üzerindeki bir ya da dah fazla sektörün içeriğini karşılaştırır. Ya da bir ya da daha çok sektör bulunup okunabildiğinde ve ve geçerli CRC kodu geri gönderildiğinde son durumda hiçbir data karşılaştırılmamıştır.

Register:	Çağrı Değeri:	Dönen Değer:
AH	04h	hata kodu <sup>1)</sup>
AL	okunacak sektörün numarası	
CH	iz/silindir	
CL	sektör	
DH	kafa	
DL	sürücü	
ES	okuma tamponunun segmenti	
BX	okuma tamponunun ofseti	
Elde		hata varsa
2) bkz F-2		

### **Fonksiyon 05h – İz ve Silindiri Formatlar (Disket / HD)**

Bu fonksiyon sektörlerin bir iz ya da silindirini formatlar. Bir AT üzerinde ilk olarak 17h ya da 18h orta tipi onarabilirsiniz. Formatlama işlemi için her sektör formatlamak için bir format tamponunun format bilgisini içermesi gereklidir.

Eğer birkaç sektör formatlamak isterseniz format tamponun bütün sektörler için format bilgisini saklamak için yeteri kadar büyük olması gerekir. Denetçi bilgiyi format tamponunun içine yazar. (Kararlı sektörün ID alanı içine) ve data'yı okurken ve yazarken doğru sektörü tanımlamak için kullanılır.

Register:	Çağrı Değeri:	Dönen Değer:
AH	05h	hata kodu <sup>1)</sup>
AL	okunacak sektörün numarası	
CH	iz/silindir	
CL	sektör	
DH	kafa	

DL	sürücü	
ES	okuma tamponunun segmenti	
BX	okuma tamponunun ofseti	
Elde		hata varsa
1) bkz F-2		
2) bkz F-4		

### **Fonksiyon 06h – Formatlar ve Track’i İşaretler (HD)**

Bu fonksiyon daha fazla data kaydında kullanılmayacak ve hatalı sektörsüz bir iz işaretler. Bu fonksiyon yalnızca XT hard disk denetçisi için geçerlidir.

<u>Register:</u>	<u>Çağrı Değeri:</u>	<u>Dönen Değer:</u>
AH	06h	hata kodu <sup>1)</sup>
AL	boş sayfa ekle	
CH	silindir	
CL	sektör	
DH	kafa	
DL	sürücü	
Elde		hata varsa
2) bkz F-2		

### **Fonksiyon 07h – Sürücüyü Formatlar (HD)**

Bu fonksiyon sürücüyü tanımlanmış silindirden itibaren formatlar. Bu fonksiyon yalnızca XT hard disk denetçisi için geçerlidir.

<u>Register:</u>	<u>Çağrı Değeri:</u>	<u>Dönen Değer:</u>
AH	07h	hata kodu <sup>1)</sup>
AL	boş sayfa ekle	
CH	silindir	
CL	sektör	
DH	kafa	
DL	sürücü	
Elde		hata varsa
2) bkz F-2		

### **Fonksiyon 08h – Sürücü Parametrelerini Tanımlar (Disket)**

Bu fonksiyon floppuy disk’in geometrik parametrelerini tanımlar. Data BIOS tablosundan açılır ve kurulu ürüdüden geometrisi yansıtılır. Fakat bu data ortamına girmiş değildir.

<u>Register:</u>	<u>Çağrı Değeri:</u>	<u>Dönen Değer:</u>
AH	08h	hata kodu <sup>1)</sup>
BH		0
BL		sürücü tipi <sup>2)</sup>
CH		silindirini numarası -1
CL		her izin sektörü
DH		kafaların numarası
DL	sürücü	sürücülerin numarası
ES		tablo segmenti parametresi
DI		tablo ofseti parametresi

Elde hata varsa  
1) bkz F-2  
2) 0=hard disk, 1=360 kbyte, 2=1,2 Mbyte, 3=720 kbyte

### **Fonksiyon 08h – Sürücü Parametrelerini Tanımlar (HD)**

Register:	Çağrı Değeri:	Dönen Değer:
AH	08h	hata kodu <sup>1)</sup>
BH		0
BL		sürücü tipi <sup>2)</sup>
CH		silindirini numarası -1
CL		her izin sektörü
DH		kafaların numarası
DL	sürücü	sürücülerin numarası
ES		tablo segmenti parametresi
DI		tablo ofseti parametresi
Elde		hata varsa

3) bkz F-2  
4) 0=hard disk, 1=360 kbyte, 2=1,2 Mbyte, 3=720 kbyte

### **Fonksiyon 09h – Sürücü Parametrelerini Beirtir (HD)**

Bu fonksiyon hard disk geometrik parametrelerini belirtir ve uydurur. Kararlı parametreler bir tablo içinde saklanır. (bkz. F-3) Bunlar yalancı kesme vektörleri olan 41h ve 46h tarafından uzak adres olarak tutulur.

Bu fonksiyon çağrıldıktan sonra, BIOS kararlı tablodaki değerleri kullanır.

### **Fonksiyon 01h – Mouse İmlecini Gösterir / İmleç Bayrağının Artımı:**

Register:	Çağrı Değeri:	Dönen Değer:
AH	09h	hata kodu <sup>1)</sup>
DL	sürücü	sürücünün numarası
Elde		hata varsa

2) bkz F-2

### **Fonksiyon 0Dh – Hard Disk Reseti (HD)**

Bu fonksiyon adreslenmiş sürücüyü resetler.

Register:	Çağrı Değeri:	Dönen Değer:
AH	09h	hata kodu <sup>1)</sup>
DL	sürücü	
Elde		hata varsa

2) bkz F-2

### **F-2 Hata Kodları:**

<u>Hata Kodu</u> (AH değeri)	<u>Anlamı</u>	<u>Floppy için</u> <u>geçerliliği</u>	<u>HD için</u> <u>geçerliliği</u>
00h	hata yok	evet	evet

01h	geçersiz fonksiyon numarası	evet	evet
02h	adres işareti bulunamadı	evet	evet
03h	disk yazmaya karşı korumalı	evet	hayır
04h	sektör bulunamadı	evet	evet
05h	başarısız reset	hayır	evet
07h	hatalı başlatma	evet	evet
08h	DMA taşması	evet	hayır
09h	DMA segment taşması	evet	evet
10h	okuma hatası	evet	evet
11h	data okuma hatası	hayır	evet
	ECC doğrulaması başarılı		
20h	denetçi hatası	evet	evet
40h	iz bulunamadı	evet	evet
80h	sürücü yasıtı	evet	evet
BBh	BIOS hatası	hayır	evet
FFh	Tanımlanamayan hata	hayır	evet

### **F-3 Hard disk Sürücüsü Parametre Tablosu:**

#### **XT-Denetçisi**

00h	Silindirelerin Sayısı	(Word)
02h	Kafaların Sayısı	(Byte)
03h	Azaltılmış Yazma Güncelinin Başlangıç Silindiri	(Word)
05h	Ön Gidermenin Başlangıç Adresi	(Word)
07h	Max. ECC Patlama Uzunluğu	(Byte)
08h	Denetim Byte'ı (1)	(Byte)
09h	Standart Zaman-Bitim Değeri (2)	(Byte)
0ah	Formatlama için Zaman-Bitim Değeri (2)	(Byte)
0bh	Sürücü Denetimi için Zaman-Bitim Değeri (2)	(Byte)
0ch	Ayrılmış	(4 Byte)
0fh		

- (1) Bit 0..2: Sürücü Seçeneği  
 Bit 3..5: Boş (Null)  
 Bit 6 : 1=ECC Yenilemleri Etkisiz  
 Bit 7 : 1= Arama Yenilemleri Etkisiz  
 (2) Zamanlayıcı İşaretinde

#### **AT-Denetçisi**

00h	Silindirelerin Sayısı	(Word)
02h	Kafaların Sayısı	(Byte)
03h	Ayrılmış	(Word)
05h	Ön Gidermenin Başlangıç Adresi	(Word)
07h	Ayrılmış	(Byte)
08h	Denetim Byte'ı (1)	(Byte)
09h	Ayrılmış	(3 Byte)
0ah	Kafa Parkı için İniş Bölgesi	(Word)
0bh	Sektör / İz Sayısı	(Byte)
0ch	Ayrılmış	(Byte)
0fh		

- (1) Bit 0..2: Ayrılmış  
 Bit 3 : 1= 8 kafadan daha fazla  
 Bit 4 : Ayrılmış  
 Bit 5 : 1= MaxSilindir+1'de Listeyi Boz  
 Bit 6 : 1=ECC Yenilemleri Etkisiz  
 Bit 7 : 1= Arama Yenilemleri Etkisiz

### **F-4 Format Tamponu:**

Ofset	Boyut	İçerik
00h	byte	sektörün izi formatsız



01h	byte	sektörün kafası formatsız
02h	byte	sektörün numarası
03h	byte	her sektör byte'ının numarası *)

\*) 0=128, 1=256, 2=512, 3=1024

## **Arabirimlere Erişim:**

### **I-1 Paralel Arayüz:**

Paralel arabirime erişmek için DOS'un 21h kesmesinin dört fonksiyonu BIOS'un 17h kesmesinin üç fonksiyonu kullanılır.

#### **I-1-1 DOS Fonksiyonları:**

Bu fonksiyonlar DOS'un 21h kesmesini çağırarak erişirler.

#### **Fonksiyon 05h – Yazma:**

Fonksiyon paralel arayüz PRN'ye bir karakter aktarır.

Register	Çağrı Değeri	Dönen Değer
AH	05h	
DL	yazılan karakterin ASCII kodu	
Elde		

#### **Fonksiyon 40h – Dosya / Unit Yazar:**

Bu fonksiyon bir tampondan paralel arayüze bir ya da daha çok karakter aktarır. Genellikle PRN tutucu 4'e atanmıştır. Aksi halde tutucuyu 3. fonksiyonla geri döndürmelisiniz. “dosya / aygıt açma”

Register	Çağrı Değeri	Dönen Değer
AH	40h	
AX		hata kodu / byte numarası *)
BX	tutucu	
CX	yazılan byte'ların numarası	
DX	yazma tamponu ofseti	
DS	yazma tamponu segmenti	
Elde		hata varsa

\*) sistem hata kodu, eğer elde ayarlanmışsa ya da güncel yazılmış byte'ların numarası ayarlanmışsa

#### **I-1-2 BIOS Fonksiyonları:**

Bu fonksiyonlar BIOS interrupt 17h'ı çağırarak erişirler.

#### **Fonksiyon 00h – Paralel Arayüz ve Yazıcıya Karakter Çıktı Verir:**

Bu fonksiyon paralel arabirime bir karakter çıktı verir.

Register	Çağrı Değeri	Dönen Değer
AH	00h	durum <sup>1)</sup>
AL	karakterin ASCII kodu	
DX	arayüz / yazıcı numarası <sup>2)</sup>	

Elde

- 1) bkz. I-1-3
- 2) 0=LPT1=PRN, 1=LPT2, 2=LPT3, 3=LPT4

### **Fonksiyon 01h – Paralel Arabirimi ve Yazıcıyı Başlangıç Durumuna Getirir:**

Bu fonksiyon paralel arabirimi ve bağlı yazıcıyı başlangıç durumuna getirir.

Register	Çağrı Değeri	Dönen Değer
AH	00h	durum <sup>1)</sup>
DX	arabirim / yazıcı numarası <sup>2)</sup>	

Elde

- 1) bkz. I-1-3
- 2) 0=LPT1=PRN, 1=LPT2, 2=LPT3, 3=LPT4

### **Fonksiyon 02h – Yazıcı Durumunu Belirtir:**

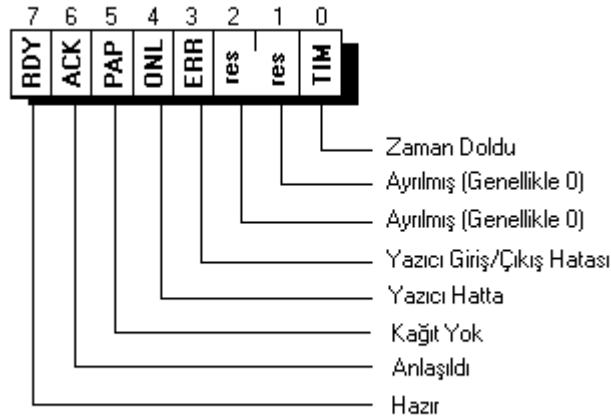
Bu fonksiyon bağlı yazıcının ve arabirimin durumunu belirtir.

Register	Çağrı Değeri	Dönen Değer
AH	00h	durum <sup>1)</sup>
DX	arabirim / yazıcı numarası <sup>2)</sup>	

Elde

- 1) bkz. I-1-3
- 2) 0=LPT1=PRN, 1=LPT2, 2=LPT3, 3=LPT4

### **I-1-3 Yazıcı Durum Byte'ı:**



### **I-2 Seri Arabirim:**

Seri arabirim için DOS 21h kesmesinin 4 fonksiyonu BIOS 14h kesmesinin 7 fonksiyonu uygundur.

#### **I-2-1 DOS Fonksiyonları:**

Bu fonksiyonlar DOS'un 21h kesmesini çağırarak erişirler.

**Fonksiyon 03h – Seri Arayüzden Karakter Okur:**

Bu fonksiyon AUX'a uydurarak COM1 seri arabirimden bir karakter okur.

Register	Çağrı Değeri	Dönen Değer
AH	03h	
AL		kabul edilmiş karakter

**Fonksiyon 04h – Seri Arabirim Yoluyla Karakter Çıktı Verir:**

Bu fonksiyon COM1 seri arabirim yoluyla AUX'a uydurarak bir karakter çıktı verir.

Register	Çağrı Değeri	Dönen Değer
AH	04h	
DL	karakter çıktısının ASCII kodu	
Elde		

**Fonksiyon 03fh – Dosya / Aygıt Okur:**

Bu fonksiyon bir tampon içerisinden bir seri arabirimden bir ya da daha çok karakter okur.

Genellikle AUX COM1'i atanmış tutucu 3'e uydurur. Aksi halde tutucuyu 3dh fonksiyonuyla geri döndürmelisiniz. Dosya / Sürücü açmak arayüzü ilgilendirir.

Register	Çağrı Değeri	Dönen Değer
AH	03fh	
AX		hata kodu / byte numarası *)
BX	tutucu	
CX	okunan byte'ların numarası	
DX	okuma tamponu ofseti	
DS	okuma tamponu segmenti	
Elde		hata varsa

\*) Elde ayarlanmışsa sistem hata kodu; ya da byte'ların numarası hale okunuyor.

**Fonksiyon 40h – Dosya / Sürücü Yazar:**

Bu fonksiyon seri arabirim yoluyla bir tampondan bir ya da daha çok karakter çıktı verir.

Genellikle AUX COM1'i atanmış tutucu 3'e uydurur. Aksi halde tutucuyu 3dh fonksiyonuyla geri döndürmelisiniz. Dosya / Sürücü açmak arabirimi ilgilendirir.

Register	Çağrı Değeri	Dönen Değer
AH	03fh	
AX		hata kodu / byte numarası *)
BX	tutucu	
CX	okunan byte'ların numarası	
DX	okuma tamponu ofseti	
DS	okuma tamponu segmenti	
Elde		hata varsa

\*) Elde ayarlanmışsa sistem hata kodu; ya da byte'ların numarası hale okunuyor.

**I-2-2 BIOS Fonksiyonları:**

Bu fonksiyonlar BIOS'un 14h kesmesi çağrısı yoluyla erişirler.

**Fonksiyon 00h – Seri Arabirimi Başlangıç Durumuna Getirir:**

Bu fonksiyon seri arabirimi başlangıç durumuna getirir.

Register	Çağrı Değeri	Dönen Değer
AH	00h	gönderme durumu <sup>1)</sup>
AL	parametre byte'ı <sup>2)</sup>	modem durumu <sup>3)</sup>
DX	arabirim numarası <sup>4)</sup>	

1) bkz. I-2-3  
2) bkz. I-2-5  
3) bkz. I-2-4  
4) 0=COM1=AUX, 1=COM2, 2=COM3, 3=COM4

**Fonksiyon 01h – Seri Arayüz Yoluyla Karakter Çıktısı Verir:**

Bu fonksiyon seri arabirim yoluyla karakter çıktısı verir.

Register	Çağrı Değeri	Dönen Değer
AH	01h	gönderme durumu <sup>1)</sup>
AL	karakter	
DX	arabirim numarası <sup>2)</sup>	

1) bkz. I-2-3  
2) 0=COM1=AUX, 1=COM2, 2=COM3, 3=COM4

**Fonksiyon 02h – Seri Arabirim Yoluyla Karakter Çıktısı Okur:**

Bu fonksiyon seri arabirim yoluyla karakter çıktısı okur.

Register	Çağrı Değeri	Dönen Değer
AH	02h	gönderme durumu <sup>1)</sup>
AL		karakter
DX	arabirim numarası <sup>2)</sup>	

3) bkz. I-2-3  
4) 0=COM1=AUX, 1=COM2, 2=COM3, 3=COM4

**Fonksiyon 03h – Seri Arabirim'in Durumunu Belirtir:**

Bu fonksiyon seri arabirim'in o anki durumunu belirtir.

Register	Çağrı Değeri	Dönen Değer
AH	03h	gönderme durumu <sup>1)</sup>
AL		modem durumu
DX	arabirim numarası <sup>2)</sup>	

1) bkz. I-2-3  
2) bkz. I-2-4  
3) 0=COM1=AUX, 1=COM2, 2=COM3, 3=COM4

**Fonksiyon 04h – Seri Arabirim Uzatımını Başlatır.**

Bu fonksiyon seri arabirim'in bir uzatılmış başlatmasını dışa taşır. Bu fonksiyon yalnız PS/2'lerde geçerlidir.

Register	Çağrı Değeri	Dönen Değer
----------	--------------	-------------

AH	04h	gönderme durumu <sup>1)</sup>
AL	kesme ayarı	modem durumu <sup>2)</sup>
BH	parite	
BL	stop bitleri	
CH	data bitleri	
CL	boud rate	
DX	arabirim numarası <sup>3)</sup>	

1) bkz. I-2-3

2) bkz. I-2-4

3) 0=COM1=AUX, 1=COM2, 2=COM3, 3=COM4

### <<<K-5>>>

#### **Fonksiyon 05h – Alt Fonksiyon 00h – Modem Kontrol Register’ını Okur :**

Bu fonksiyon seri arabirim modem kontrol register’ını okur. Bu fonksiyon yalnız PS/2’de geçerlidir.

Register	Çağrı Değeri	Dönen Değer
AH	05h	
AL	00h	
BL		modem kontrol register’ı
DX	arayüz numarası <sup>3)</sup>	

1) bkz. I-2-6

2) 0=COM1=AUX, 1=COM2, 2=COM3, 3=COM4

#### **Fonksiyon 05h – Alt Fonksiyon 00h – Modem Kontrol Register’ını Yazar :**

Bu fonksiyon seri arayüzün modem kontrol register’ını yazar. Bu fonksiyon yalnız PS/2’de geçerli.

Register	Çağrı Değeri	Dönen Değer
AH	05h	
AL	00h	
BL		modem kontrol register’ı
DX	arabirim numarası <sup>3)</sup>	

3) bkz. I-2-6

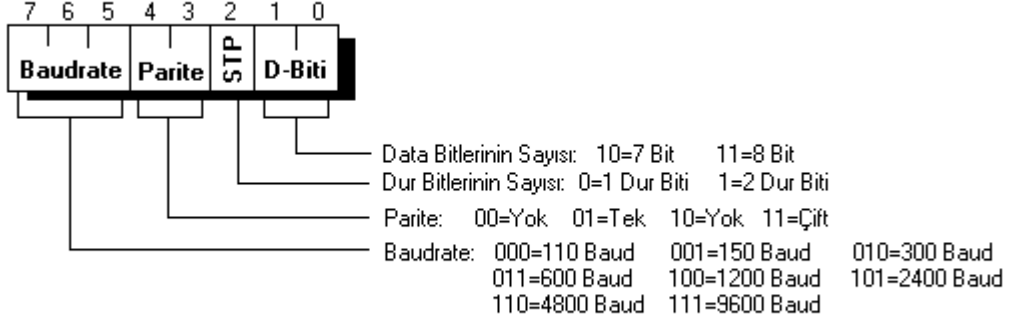
4) 0=COM1=AUX, 1=COM2, 2=COM3, 3=COM4

#### **I-2-3 Gönderme Durumu:**

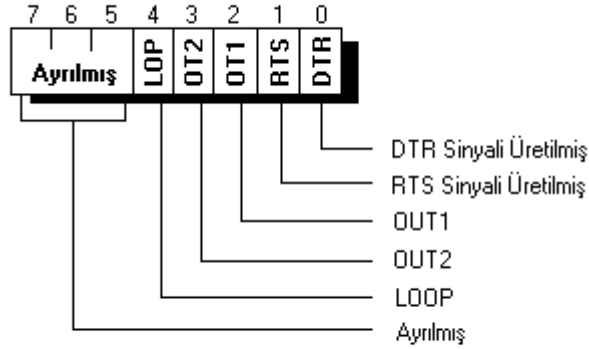
### <<<K-6>>>

#### **I-2-4 Modem Durumu:**

### I-2-3 Parametre Byte'ı



### I-2-3 Modem Kontrol Register'ı



### J Klavye ve Fare Erişimi:

#### J-1: Klavye:

- 7 DOS fonksiyonu DOS'un 21h kesmesi çağrısıyla erişirler.
- Klavye BIOS'un 15h kesmesinin anlamıyla doğrudan erişebilir.
- Bazı BIOS versiyonlarıyla, donanım kesme tutucusu 09h INT 15h'in 4fh fonksiyonunu çağırır ve tutucunun tarama koduna geçer.
- SysReg tuşuna basıldığında ya da bırakıldığında tutucunun 09h kümesi INT 15h'in 85h fonksiyonunu çağırır.

#### J-1-1 DOS Fonksiyonları:

##### Fonksiyon 01h – Echo'yla Karakter Girişi:

Bu fonksiyon klavye tamponundan bir karakter byte okur. Ve aynı zamanda karakter yoluyla standart çıktı aygıtına çıktı verir. Fonksiyon tuşları gibi non-ASCII (ASCII olmayan)'lar için

fonksiyonu ikinci kez çağırılmalıdır. Birincide 00h kodu döner, ikincide tuşun tarama kodu çağrılır.

Register	Çağrı Değeri	Dönen Değer
AH	01h	
AL		ASCII kod

**Fonksiyon 06h – Standart Girdi Aygıtından Denetimsiz Olarak Karakter Girer:**

Bu fonksiyon klavye tamponundan bir karakter okumaya teşebbüs eder. Ve bir uygun karakteri beklemez. F1 gibi non-ASCII tuşlar ile ilk çağrıda 0'a eşit kod geri döner ve ikincide basılan tuşun tarama kodu çağrılır. Bu fonksiyon klavyeden standart girdi aygıtı kadar uzun bir şekilde okur. Ctrl-C gibi kontrol karakterleri açıklanmamıştır.

Register	Çağrı Değeri	Dönen Değer
AH	06h	
AL		ASCII kod
DL	ffh	
Zero		1= cr karakter okuma 0= hiçbir karakter uygun değil

**Fonksiyon 07h- Klavyeden Doğrudan Karakter Girdisi Alır:**

Bu fonksiyon klavye tamponundan bir karakter okumaya teşebbüs eder. Ve bir uygun karakteri bekler. F1 gibi non-ASCII tuşlar ile ilk çağrıda 0'a eşit kod geri döner ve ikincide basılan tuşun tarama kodu çağrılır. 06h fonksiyonundan farklı olarak, bu fonksiyon klavye tamponu boşsa tuş basılana kadar bekler. Ctrl-C gibi kontrol karakterleri açıklanmamıştır.

**Fonksiyon 06h – Standart Girdi Aygıtından Denetimle Karakter Girdisi Alır:**

Bu fonksiyon klavye tamponundan bir karakter okumaya teşebbüs eder. Ve bir uygun karakteri bekler. F1 gibi non-ASCII tuşlar ile ilk çağrıda 0'a eşit kod geri döner ve ikincide basılan tuşun tarama kodu çağrılır. Bu fonksiyon klavyeden standart girdi aygıtı kadar uzun bir şekilde okur. Eğer yol <dosya ya da <aygıt aktif şeklindeyse, karakterler dosya ya da aygıttan okunur. Ctrl-C gibi kontrol karakterleri açıklanmamıştır.

Register	Çağrı Değeri	Dönen Değer
AH	08h	
AL		ASCII kod

**Fonksiyon 0ah – Standart Girdi Aygıtından Echo'yla tamponlanmış karakter girer:**

Bu fonksiyon tampondan karakter stringi okur. RETURN'e basıldığında girdi sonlandırılır. Ctrl-C (program iptali) ve Ctrl-P (Printer da eko) hariç karakter kodlarından önce “^” simgesi getirilir. (Örneğin Ctrl-R, ^R). Tamponun ilk byte'ı girdinin maksimumunu gösterir ve ikinci byte asıl uzunluktur. Bu yüzden tampon maksimum uzunluk+2 byte içerir. Bu fonksiyon klavyeden standart girdi aygıtındaki kadar uzun okur. Eğer yönelimi <dosya ya da <aygıt aktif şeklindeyse, bütün karakterler dosya ya da aygıttan okur.

Register	Çağrı Değeri	Dönen Değer
AH	0ah	
DX	tampon ofseti	
DS	tampon segmenti	

### **Fonksiyonu 0bh – Standart Girdi Aygıtının Durumunu Denetler:**

Bu fonksiyon karakter standart girdi aygıtı yoluyla alındığında tanımlanır. Bu fonksiyon klavyeye standart girdi aygıtı kadar uzun erişir. Yönelim <dosya ya da <aygıt aktif şeklindeyse bütün karakterler dosya ya da aygıttan okunur.

Register	Çağrı Değeri	Dönen Değer
AH	0bh	
AL		00h=hiçbir karakter uygun değil ffh= bir karakter uygun

### **Fonksiyon 3fh – Dosya / Aygıt Okur:**

Bu fonksiyon klavyeden tampon içine bir ya da daha çok karakter okur. Genellikle klavye (CON) tutucu 0 atanmıştır.

Register	Çağrı Değeri	Dönen Değer
AH	3fh	
AX		hata kodu byte numarası
BX	tutucu	
CX	okunan byte'ların numarası	
DX	tampon ofsetini okur	
DS	tampon segmentini okur	error if <>0

\* Elde ayarlanmışsa sistem hata kodu, yada o anda okunan byte'ların numarası.

### **J-1-2 BIOS İnterrupt INT 16h:**

#### **Fonksiyon 00h – Gelecek Karakteri Okur:**

Bu fonksiyon klavye tamponunun dışından gelecek karakteri okur ve uygun getirme işaretçisi günceller. Fonksiyon gelecek tuşu bekler.

Register	Çağrı Değeri	Dönen Değer
AH	00h	tarama kodu
AL		ASCII kodu

#### **Fonksiyon 01h – Tampon Durumunu Tanımlar:**

Bu fonksiyon klavye tamponunun durumunu tanımlar. Ve karakter uygun olduğunda gösterir. 00h fonksiyonunun aksine, klavye işaretçisi güncellenmemiştir.

Register	Çağrı Değeri	Dönen Değer
AH	01h	tarama kodu
AL		ASCII kodu
Sıfır		0= karakter uygun 1= hiçbir karakter uygun değil

#### **Fonksiyon 02h – Shift Durumunu Tanımlar:**

Bu fonksiyon klavye bayrağı 0040:0017'yi denetleyerek shift tuşlarının durumunu tanımlar.



Register	Çağrı Değeri	Dönen Değer
AH	02h	
AL		shift durumu *)

\*) bkz. J-1-4

### **Fonksiyon 03h – Yazma Aralığını ve Duraklamayı Ayarlar:**

Bu fonksiyon AT ve MFII klavyelerin yazma aralığını ve duraklamayı ayarlar.

Register	Çağrı Değeri	Dönen Değer
AH	03h	
BL	tipmatik aralığı <sup>1)</sup>	
BH	duraklama <sup>2)</sup>	

2) L-21

### **Fonksiyon 05h – Klavye Tamponunda Karakter ve tarama Kodunu Geri Yazar:**

Bu fonksiyon bir ASCII karakter yazar ve tarama kodunun tarama tamponuna geri gönderir. Ve bir tuş basılmasını benzetir.

Register	Çağrı Değeri	Dönen Değer
AH	05h	
AL		durum *)
CH	tarama kodu	
CL	ASCII kodu	

\*) 00h=O.K. 01h = klavye tamponu dolu

### **Fonksiyon 10h – Uzatılmış Tampondan bir Karakter Okur:**

Bu fonksiyon uzatılmış (MFII) klavyeden bir karakter okur. Bu fonksiyon 00h'a banzer ama uzatılmış klavyenin kodlarını destekler. Bu yeni fonksiyonun kodları için önkod byte'ı 0eh geçilir. Yerine 00h değeri gelir. Böylece yeni tuşlar geleneksel olandan ayrılabilir.

Register	Çağrı Değeri	Dönen Değer
AH	10h	tarama kodu
AL		ASCII kodu

### **Fonksiyon 11h – Uzatılmış Klavye İçin Tampon Durumunu Tanımlar:**

Bu fonksiyon uzatılmış klavye için tampon durumunu tanımlar. Bu fonksiyon 01h fonksiyonuna benzer ancak uzatılmış klavye kodlarını destekler. Bu yeni fonksiyonun kodları için önkod byte'ı 0eh geçilir. Yerine 00h değeri gelir. Böylece yeni tuşlar geleneksel olandan ayrılabilir.

Register	Çağrı Değeri	Dönen Değer
AH	11h	tarama kodu
AL		ASCII kodu
Sıfır		0= karakter uygun 1= hiçbir karakter uygun değil

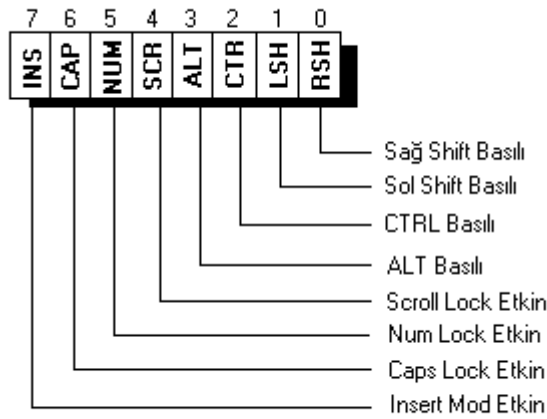
### **Fonksiyon 12h – Uzatılmış Klavye için Shift Durumunu Tanımlar:**

Bu fonksiyon klavye bayrağı 0040:0017 ve 0040:0018'i denetleyerek SHIFT tuşlarının durumunu tanımlar. Bu fonksiyon 11h fonksiyonuyla uyuşur. Ancak uzatılmış klavye için ek SHIFT tuşlarını destekler.

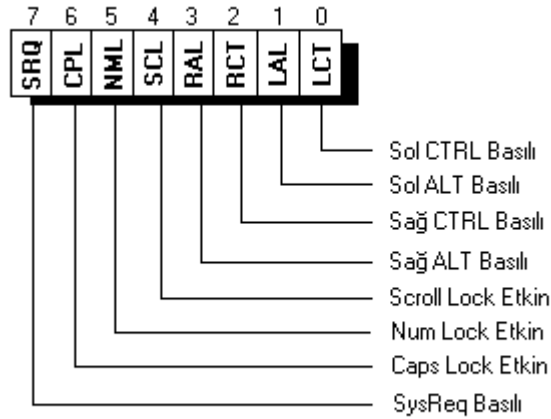
Register	Çağrı Değeri	Dönen Değer
AH	12h	2. Shift durumu byte'ı <sup>1)</sup>
AL		1. Shift durumu byte'ı <sup>2)</sup>

1) bkz. J-1-5  
2) bkz. J-1-4

#### **J-1-4 – Birinci Shift Durum Byte'ı:**



#### **J-1-4 – Birinci Shift Durum Byte'ı:**



### **J-2 – Mouse İnterrupt 33h:**

Mouse ve Mouse sürücüsü 33h kesmesi yoluyla erişilebilirler.

#### **J-2-1 – INT 33h'in Fonkiyonları:**

Bu fonksiyon mouse var olduğunda tanımlanır. Ve mouse sürücüsünün standart ayarlarını getirir. BX register'ı içinde uygun mouse düğmelerinin numarası geri döner.

Register	Çağrı Değeri	Dönen Değer
AX	00h	durum <sup>1)</sup>
BX		Mouse Düğmeleri <sup>2)</sup>

1) 1= mouse var; 0=hiçbir mouse yok

2) 2= iki düğme (Microsoft); 3= üç düğme

### **Fonksiyon 01h – Mouse İmlecini Gözterir / İmleç Bayrağının Artımı:**

Bu fonksiyon imleç bayrağını arttırır. Bayrak 0'a eşitse mouse'un imleci ekran üzerinde görünür. Normalde imleç bayrağı –1 değerindedir.

Register	Çağrı Değeri	Dönen Değer
AX	01h	

### **Fonksiyon 02h – Mouse İmlecini Temizler / İmleç Bayrağının Azalımı:**

Bu fonksiyon imleç bayrağını bir azaltır. Böylece Mouse'un imleci ekrandan temizlenir. Bayrak 0'a eşitse mouse'un imleci ekran üzerinde görünür. Normalde imleç bayrağı –1 değerindedir.

Register	Çağrı Değeri	Dönen Değer
AX	02h	

### **Fonksiyon 03h- Mouse Düğmelerini ve Mouse İmleç Konumunu Tanımlar:**

Bu fonksiyon mouse düğmelerinin güncel durumunu ve mouse imlecinin ekran üzerindeki güncel konumunu tanımlar.

Register	Çağrı Değeri	Dönen Değer
AX	03h	
BX		düğme byte'ı *)
CX		imleç konumunun X-değeri
DX		imleç konumunun Y-değeri

\*) bkz. J-2-2 : (1=düğme basımına uydurma, 0=düğme bırakımına uydurma)

### **Fonksiyon 04h – Mouse'un İmleç Konumunu Avarlar:**

Bu fonksiyon mouse imlecinin ekran üzerindeki konumunu ayarlar.

Register	Çağrı Değeri	Dönen Değer
AX	04h	
CX		imleç konumunun X-değeri
DX		imleç konumunun Y-değeri

### **Fonksiyon 05 – Mouse'un Basılan Düğme Numarasını Ve Mouse'un İmleç Konumunu Tanımlar:**

Bu fonksiyon kesin mouse düğmesinin hangi sıklıkla basıldığını ve son tıklamada mouse imlecinin ekranın neresinde olduğunu tanımlar.

Register	Çağrı Değeri	Dönen Değer
AX	05h	düğme byte'ı <sup>1)</sup>
BX	mouse düğmesi <sup>2)</sup>	sayaç değeri <sup>3)</sup>
CX		imleç konumunun X-değeri
DX		imleç konumunun Y-değeri

- 1) bkz – J-2-2 (1=düğme basımına uydurma, 0=düğme bırakımına uydurma)  
 2) 1= sol düğme denetle 2= sağ düğme denetle 4= orta düğme denetle  
 3) tıklamaların numarası (0 dan 32'ye, 767)

**Fonksiyon 06h – Bırakılan Mouse Düğmesini ve Mouse İmleç Konumunu Belirtir:**

Bu fonksiyon kesin mouse düğmesinin son araştırmadan bu yana hangi sıklıkla bırakıldığını ve son bırakmada mouse imlecinin ekranın neresinde olduğunu belirtir.

Register	Çağrı Değeri	Dönen Değer
AX	06h	düğme byte'ı <sup>1)</sup>
BX	mouse düğmesi <sup>2)</sup>	sayaç değeri <sup>3)</sup>
CX		imleç konumunun X-değeri
DX		imleç konumunun Y-değeri

- 1) bkz – J-2-2 (1=düğme basımına uydurma, 0=düğme bırakımına uydurma)  
 2) 1= sol düğme denetle 2= sağ düğme denetle 4= orta düğme denetle  
 3) tıklamaların numarası (0 dan 32'ye, 767)

**Fonksiyon 07h – Mouse İmleci İçin Yatay Kenarlık Tanımlar:**

Bu fonksiyon mouse imleci hareketinin yatay kenarlıklarını belirler. Mouse imleci bunun dışına çıkamaz. Mouse daha uzağa giderse, böylece fonksiyon 08h ile birlikte mouse imleci için bir pencere tanımlayabilirsiniz.

Register	Çağrı Değeri	Dönen Değer
AX	07h	
BX		
CX	sol kenarlık	
DX	sağ kenarlık	

**Fonksiyon 08h – Mouse imleci İçin Dikey Kenarlık Tanımlar:**

Bu fonksiyon mouse imleci için dikey kenarlık tanımlar. Mouse imleci bunun dışına çıkamaz. Mouse daha uzağa giderse, böylece fonksiyon 07h ile birlikte mouse imleci için bir pencere tanımlayabilirsiniz.

Register	Çağrı Değeri	Dönen Değer
AX	08h	
BX		
CX	üst kenarlık	
DX	alt kenarlık	

**Fonksiyon 09h – Grafik Modda Mouse İmlecinin Tanımları:**

Bu fonksiyon mouse imlecinin grafik modda şeklini ve davranışını tanımlar. Ekran ve imleç maskesi tampon içinde bu düzende desteklenir. Eylem noktası göreceli mouse imlecinin konumunun bir sorgusunda geri dönen değer olan üst sol köşesi olarak tanımlanır. olarak tanımlanır.

Register	Çağrı Değeri	Dönen Değer
AX	09h	
BX	yatay eylem noktası *)	
CX	dikey eylem noktası *)	
DX	maske tampon ofseti	

ES maske tampon segmenti  
\*) değerlerin aralığı : -16...+16

### **Fonksiyon 0ah – Mouse İmleci Text Modda Tanımlar:**

Bu fonksiyon mouse imlecinin şeklini ve davranışını tanımlar.

Register	Çağrı Değeri	Dönen Değer
AX	0ah	
BX	mouse imleç tipi <sup>1)</sup>	
CX	ekran maskesi <sup>2)</sup>	
DX	imleç maskesi <sup>3)</sup>	

- 1) 0= yazılım mouse imleç 1= donanım mouse imleç
- 2) yazılım mouse imleç ekran maske kodu , donanım mouse imleç ; mouse imlecini ilk tarama çizgisi
- 3) yazılım maske imleç ; imleç maske kodu
- 4) donanım mouse imleç ; mouse imlecini son tarama çizgisi

### **Fonksiyon 0bh – Mouse'un Hareket Sayacını Okur:**

Bu fonksiyon mouse'un hareket sayacını okur. Ve son son fonksiyon çağrısından bu yana ne kadar uzağa hareket ettiğini belirtir. Bir Sayaç değeri 1/250 veya 0,13mm'ye eşittir.

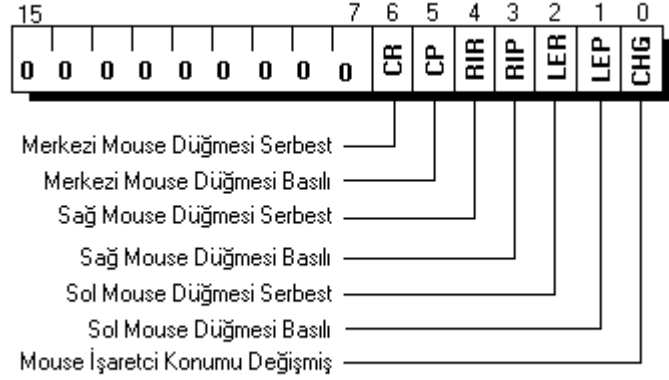
Register	Çağrı Değeri	Dönen Değer
AX	0bh	
CX		yatay sayaç değeri *)
DX		dikey sayaç değeri *)

\*) değerlerin aralığı - -32768 ..... +32767

### **Fonksiyon 0ch – Kullanıcı Prosedürü için Çağrı Maskesi Tanımlar:**

Bu fonksiyon uzun çağrı yoluyla mouse sürücüsünün bir kullanıcı tanımlı prosedür çağrısı için durumları tanımlar.

Register	Çağrı Değeri	Dönen Değer
AX	0ch	
CX	çağrı maskesi *)	
DX	prosedür ofseti	
ES	prosedür segmenti	



### **Fonksiyon 0fh – Piksel Oranını Tanımlar:**

Bu fonksiyon her pikselin piksel piksel oranını tanımlar. Yatay standart değeri 8'e, dikey ise 16'ya eşittir.

Register	Çağrı Değeri	Dönen Değer
AX	0fh	
CX	yatay oran *)	
DX	dikey oran *)	
*) değerlerin aralığı 1.....32,767		

### **K – Grafik Bağdaştırıcılara Erişim:**

- DOS fonksiyonları DOS'un 21h kesmesiyle çağırılırlar.
- Sistem BIOS grafik bağdaştırıcınının text ve grafik modda standart fonksiyonlarını içerir.
- EGA ve VGA dah fazla güçlü bağdaştırıcı mantık kontrolleri ile BIOS uzantılarına sahiptir. Bunlar yani text ve grafik formları içinde karakter ve grafik çıktıları içindir.

### **K-1 . DOS Fonksiyonları:**

#### **Fonksiyon 02h – Karakter Çıktı:**

Bu fonksiyon standart çıktı aygıtından bir karakter çıktı verir. (genellikle CON) Karakter kodu 07h (zil), 08h (backspace), 09h (tab), 0ah (çizgi besleme) ve 0dh (taşım döngüsü) kontrol karakterlerinden önce gelir. Ve BIOS uydurma sürecini yürütür.

Register	Çağrı Değeri	Dönen Değer
AX	02h	
DL	ASCII karakter kodu	

#### **Fonksiyon 06h – Satandard Çıktı Aygıtlarına Denetimsiz Olarak Karakter Çıktı Verir:**

Bu fonksiyon standart çıktı aygıtından bir karakter çıktı verir. (genellikle CON). Eğer yönelim <dosya ya da <aygıt aktif ise bütün karakterler dosya ya da aygıtından çıktı alınır. Ctrl-C gibi DOS kontrol karakterleri açıklanmamıştır. Ama yalnızca aktarılır. Karakter kodlar 07h (zil), 08h (backspace), 09h (tab), 0ah (çizgi besleme) ve 0dh (taşım döngüsü) BIOS tarafından tanınır ve açıklanırlar.

Register	Çağrı Değeri	Dönen Değer
AH	06h	
DL	ASCII karakter kodu	

### **Fonksiyon 09h – String Çıktı:**

Bu fonksiyon standart çıktı aygıtına string çıktı verir (genellikle CON). String “\$ (ASCII kod 36) karakteriyle sonlandırılmalıdır.” . Ctrl-C gibi DOS kontrol karakterleri açıklanmamıştır. Ama yalnızca aktarılır.

Karakter kodlar 07h (zil), 08h (backspace), 09h (tab), 0ah (çizgi besleme) ve 0dh (taşım döngüsü) BIOS tarafından tanınır ve açıklanırlar.

Register	Çağrı Değeri	Dönen Değer
AH	09h	
DX	string ofseti	
DS	string segmenti	
AL		

### **Fonksiyon 40h – Dosya / Aygıt Yazar :**

Bu fonksiyon tutucunun açıklamasıyla bir string çıktı verir. Tutucu standart çıktı aygıtı için 1'e eşittir. Eğer standart çıktı aygıtı CON ise ekran üzerine bir ya da daha çok çıktı almak istiyorsanız. CON'u dosya/aygıt açma fonksiyonuyla yazma erişimi için açmanız gerekiyor.

Register	Çağrı Değeri	Dönen Değer
AH	40h	
AX		hata kodu / byte numarası *)
BX	tutucu (01h)	
CX	çıktı byte'larının numarası	
DX	yazma tamponu ofseti	
DS	yazma tamponu segment	
Elde		hata kodu

\*) Elde ayarlı ise sistem hata kodu, ya da güncel yazılan byte'ların numarası

### **K-2 – BIOS İnterrupt 10h – Sistem BIOS'un Standart Fonksiyonları:**

Listelenmiş fonksiyonlar anakart üzerinde sistem BIOS içinde yürütülmüşlerdir.

### **Fonksiyon 00h – Video Modu Ayarlar:**

Bu fonksiyon uygun değerlerle modu ve grafik çipinin kontrol register'ını yükleyerek kesin video modunu ayarlar. Bir 00h fonksiyonu çağrısından sonra, imleç 0 çizgisindedir., sütun 0, palet standart renklere getirilir ve ekran temizlenir.

Register	Çağrı Değeri	Dönen Değer
AH	00h	
AL	video modu *)	

\*) her video RAM katmanı

### **Fonksiyon 01h – İmleç Boyutunu Ayarlar:**

Bu fonksiyon text modda imleç boyutunu ayarlar. Eğer tarama çizgisinin başlangıç değeri bitiş değerinden büyükse, bölünmüş imleç görünür. Eğer tarama çizgilerinin değeri aktif video modundaki karakter kutusunda verilen aralığın dışındaysa her bir imleç görüntülenmez. İmleç parametreleri BIOS'un veri alanında (40:60h) saklıdır. CGA için standart ayarlar: tarama çizgisi başlangıç= 6 ; tarama çizgisi sonu = 7; MDA için tarama çizgisi başlangıcı=11; tarama çizgisi sonu=12 'dir.

Register	Çağrı Değeri	Dönen Değer
AH	01h	
CH	imlecin tarama çizgisi başlangıcı <sup>1)</sup>	
CL	imlecin tarama çizgisi sonu <sup>2)</sup>	

1) bit 17 ayrılmış (=0); bit 6-5:00=normal 01= imleç görüntülenmemiş

2) bit 7-5 ayrılmış (=0); bit 4-0=tarama çizgisi sonu

### **Fonksiyon 02h – İmleç Konumunu Ayarlar:**

Bu fonksiyon aktif ekran sayfasında imleç konumunu ayarlar. Ekran sayfasının gösterimde olması gerekir. Gizli sayfa adreslenebilir. Text Modda text imleci bu bölgede görülebilir. Grafik modda görünmez arta kalır. Ancak Nerede olduğu koordinatları tanımlanır. Örneğin nokta ayarlanabilir. Bu fonksiyon imleç koordinatlarını 40:50h'ta saklar.

Register	Çağrı Değeri	Dönen Değer
AH	02h	
BH	ekran sayfası	
DH	sıra	
DL	sütun	

### **Fonksiyon 03h- İmleç Konumunu Okur:**

Bu fonksiyon aktif ekran sayfasında imleç konumunu belirtir. Ekran sayfası görünümde olmamalıdır. Zaten gizli sayfa adreslenebilir. Bu fonksiyon imleç koordinatlarını 40:50h'tan okur.

Register	Çağrı Değeri	Dönen Değer
AH	03h	
BH	ekran sayfası	
CH		imleç – tarama çizgisi başlangıcı
CL		imleç – tarama çizgisi sonu
DH		satır
DL		sütun

### **Fonksiyon 04h – Işık Kalem Konumunu Okur: (Text / Grafik)**

Bu fonksiyon ışık kaleminin konumunu ve durumunu belirtir. Bu günlerde ışık kaleminin yerini mouse ve iz topu almıştır.

Register	Çağrı Değeri	Dönen Değer
AH	04h	durum <sup>1)</sup>



BH	piksel sütunu (grafik mod)
CH	piksel çizgisi 0-199 (grafik mod)
CL	imleç çizgisi (0 – xxx <sup>2</sup> ) (gm)
DH	sıra (text mod)
DL	sütun (text mod)

- 1) 00h = ışık kalemi açma anahtarı  
01h= anahtar kapalı, koordinatlar geçerli
- 2) 200 çizgiden büyük çizgilerle mod için (xxx 200)

### **Fonksiyon 05h – Ekran Sayfası Secer (Text / Grafik):**

Bu fonksiyon aktif ekran sayfası'nı belirtir. (görünene sayfa). Uygun ekran sayfasının numarası video mod olarak korunur, ve grafik bağdaştırıcı kullanılır.

Register	Çağrı Değeri	Dönen Değer
AH	05h	
AL	ekran sayfası	

### **Fonksiyon 06h – Pencereyi Yukarı Kaydırır (Text / Grafik):**

Bu fonksiyon ekran üzerinde bir pencere tanımlar ve pencere içeriğini belirtilmiş sıra numarasını yukarı kaydırır. AL=00h ile pencere temizlenir. Boş sıralar pencerenin dibinde görünür ve boş (blank) karakterlerle doldurulur. Bu boş karakterler BH registerından atanmış özelliğindedir.

Register	Çağrı Değeri	Dönen Değer
AH	06h	
AL	kaydırılan satırların numarası	
BH	boş sıraların özelliği	
CH	pencerenin üst sırası	
CL	pencerenin sol sütunu	
DH	pencerenin alt sırası	
DL	pencerenin sağ sütunu	

### **Fonksiyon 07h – Pencereyi Yukarı Kaydırır (Text / Grafik):**

Bu fonksiyon ekran üzerinde bir pencere tanımlar ve pencere içeriğini belirtilmiş sıra numarasını aşağı kaydırır. AL=00h ile pencere temizlenir. Boş sıralar pencerenin dibinde görünür ve boş (blank) karakterlerle doldurulur. Bu boş karakterler BH registerından atanmış özelliğindedir.

Register	Çağrı Değeri	Dönen Değer
AH	07h	
AL	kaydırılan satırların numarası	
BH	boş sıraların özelliği	
CH	pencerenin üst sırası	
CL	pencerenin sol sütunu	
DH	pencerenin alt sırası	
DL	pencerenin sağ sütunu	

### **Fonksiyon 08h – Ekrandan Karakter / Özellik Okur (Text / Grafik):**

Bu fonksiyon güncel imleç konumundan karakter okur. Muhtemelen imleci arzu ettiğiniz konuma getirmelisiniz (fonksiyon 02h ile). Text modda bu fonksiyon sadece karakter kodu geri döndürmez ancak karakter özelliğidir. Grafik modda karakter matrisi belirtilmiş imleç konumunda karakter koduyla belirtilmiş aktif karakter tablosuyla karşılaştırılmıştır. Eğer matris deseni tablodaki karakterle karşılaşılmazsa fonksiyon bir 00h kodu geri döndürür.

Register	Çağrı Değeri	Dönen Değer(yalnızca text modda)
AH	08h	karakter özelliği
AL		karakter kodu
BL	ekran sayfası	

#### **Fonksiyon 09h – Ekran Üzerinde Karakter Özellik Yazar (Text / Grafik):**

Bu fonksiyon bir karakter yazar, güncel imleç konumunu başlatır, ve süreç CX kadar döner. İmleç konumunu bununla değiştirmiştir. BL özelliği belirtir. Ve grafik modda karakterin önalın rengi yazılmış olabilir. Eğer karakterler grafik modda yazılmışsa ve BL'nin bit 7'si ayarlanmışsa karakterin karakter matrisi XOR – lanmış video RAM'in içeriğiysa bu bölgeye yazılmış olabilir.

Register	Çağrı Değeri	Dönen Değer(yalnızca text modda)
AH	09h	
AL	karakter kodu	
BH	ekran sayfası	
BL	karakter özelliği ya da önalın rengi	
CX	tekrarlanmışsa numarası	

#### **Fonk 0ah – Ekran Üzerine Karakter Yazar (Text / Grafik):**

Bu fonksiyon bir karakter yazar, güncel imleç konumunu başlatır, ve süreç CX kadar döner. İmleç konumunu bununla değiştirmiştir. Eğer karakterler grafik modda yazılmışsa ve BL'nin bit 7'si ayarlanmışsa karakterin karakter matrisi XOR – lanmış video RAM'in içeriğiysa bu bölgeye yazılmış olabilir.

Register	Çağrı Değeri	Dönen Değer(yalnızca text modda)
AH	09h	
AL	karakter kodu	
BH	ekran sayfası	
CX	tekrarlanmışsa numarası	

#### **Fonksiyon 0bh – Renk Paletini Ayarlar (Grafik):**

Bu fonksiyon CGA'nın orta çözünürlüğü için renkleri ayarlar. BH register değeri olarak saklanır. BL register farklı anlamdadır. Palet Değiştirilerek görüntülenmiş ekranın rengi ani olarak değiştirilebilir. Ve kullanıcı yanıp sönen ekran izlenimi alır.

BH = 00h --

Register	Çağrı Değeri	Dönen Değer(yalnızca text modda)
AH	0bh	
BH	00h	
BL	renk 0-31 *)	

\*) mod 4,5, artalan rengi BL'ye eşit

mod 0,1,2,3 : kenarlık rengi BL'ye eşit  
mod 6,11 : önalın rengi BL'ye eşit

BH = 01h --

Register	Çağrı Değeri	Döner Değeri(yalnızca text modda)
AH	0bh	
BH	01h	
BL	palet *)	

\*) 00h : palet = yeşil (1), kırmızı (2), sarı (3)  
01h : palet = cyan (1), magenta (2), beyaz (3)

### **Fonksiyon 0ch – Ekran Üzerine Piksel Yazar (Grafik):**

Bu fonksiyon ekran üzerinde arzu edilen bölgeye bir piksel yazar. AL register'ının bit 7'si bit2i tersine çevirme hizmeti verir. AL içindeki bit, renk adreslenmiş bölgede güncel yerleşmiş piksel ile XOR'lanmıştır. Eğer tekrar aynı çeşit piksel yazarsanız, ekrandan kalkmalı. Böylece nesnelere ekranda görünüş olarak hareket etmiştir.

Register	Çağrı Değeri	Döner Değeri
AH	0ch	
AL	renk	
BH	ekran sayfası *)	
CX	piksel sütunu	
DX	piksel çizgisi	

\*) yalnızca video mod birden fazla sayfa desteklerse gerekli

### **Fonksiyon 0dh – Ekrandan Piksel Okur (Grafik):**

Bu fonksiyon ekran üzerinde kesin bölgede bir piksel okur.

Register	Çağrı Değeri	Döner Değeri
AH	0dh	
AL		okunana pikselin rengi
BH	ekran sayfası *)	
CX	piksel sütunu	
DX	piksel çizgisi	

\*) yalnızca video mod birden fazla sayfa desteklerse gerekli

### **Fonksiyon 0fh – Video Durumunu Belirtir. (Text / Grafik):**

Bu fonksiyon güncel video modunu ve video durumunu belirtir.

Register	Çağrı Değeri	Döner Değeri
AH	0fh	sütunların numarası
AL		video mod (40:49h'dan)
BH		aktif ekran sayfası (40:62h 'dan)

### **Fonksiyon 13h – String Yazar (Text / Grafik):**

Bu fonksiyon ekran üzerine geniş stringler yazar. AL register'ının 0 ve 1 bitleri fonksiyonun davranışını belirtir. String karakter kodlarını ve farklı olarak karakter özelliklerini ya da yalnızca karakter özelliklerini içerir. İkinci durumda BL register'ının içinde kullanıldığında özellik geçilmiştir. Şekil TM3

### **K3 – BIOS İnterrupt INT 10h – EGA/VGA BIOS için EK Fonksiyonlar:**

#### **Fonksiyon 00h – video Modunu Ayarlar:**

Bu fonksiyon uygun değerlerle grafik yongasının mod ve kontrol register'ını yükleyerek EGA ve VGA'nın keim video modunu ayarlar. Yüksek çözünürlüklü EGA ve VGA bağdaştırıcı kartlar sizin bağdaştırıcı kartınızda ayarlama için bulabileceğiniz text ve grafik modu için çözünürlük bolluğuna sahiptir. 00h fonksiyon çağrısından sonra imleç çizgi 0'da sütun 0, palet standart renklere ayarlı, ekran temizlenmiş. S-1

#### **Fonksiyon 10h – Palet Register'ını Ayarlar:**

Bu fonksiyon EGA ve VGA'nın palet register'ını ayarlar. EGA ile 64 farklı renk olasıdır. VGA ile  $2^{18} = 262,144$  renk olasıdır.

### **K4 – BIOS İnterrupt INT 10h – Ek SVGA BIOS Fonksiyonlar:**

#### **Fonksiyon 4fh – SVGA BIOS Fonksiyonları Çağrısı :**

##### **Alt Fonksiyon 00h – SVGA bilgisini Belirtir:**

Bu fonksiyon SVGA BIOS varken belirtilir ve bu güncel durumdur. M-1

##### **Alt Fonksiyon 01h – SVGA Mod Bilgisini Belirtir:**

Bu fonksiyon bir 256 byte tamponun içinde SVGA moduna bakarak bilgi geri döndürür. M-2

##### **Alt Fonksiyon 02h – SVGA Modunu Ayarlar:**

Bu fonksiyon SVGA modlarından birine VESA tarafından tanımlı olarak ayarlar. M-3

##### **Alt Fonksiyon 03h – SVGA Modunu Belirtir:**

Bu fonksiyon güncel SVGA modunu belirtir. M-3a \*

##### **Alt Fonksiyon 04h – SVGA Video durumu Kaydet / Eski Haline Getir:**

Bu fonksiyon SVGA BIOS'un durumunu kaydeder ya da eski haline getirir.