

Dosya Sistem Analizi

Hafta - 2

Bilgisayar Temelleri

Amaç

- Bu bölüm bilgisayarların nasıl işlem yaptığını ve düşük seviyeli temellerini kapsamaktadır.
- İkili, onaltılık ve küçük ve büyük endian sıralamaları, bir bilgisayarı başlatmak için gereken önyükleme işlemini, sabit disk geometrilerini, ATA komutlarını, ana bilgisayar korumalı alanları ve SCSI'yi gösterilecektir.

Decimal Sayılar

The values of each symbol in a decimal number.

Decimal Number: 35,812

10,000	1,000	100	10	1
3	5	8	1	2

$$(3 \times 10,000) + (5 \times 1,000) + (8 \times 100) + (1 \times 10) + (2 \times 1) = 35,812$$

En sağdaki sütuna en önemsiz, en soldaki sütuna en önemli sembol adı verilir. 35,812 sayısı için 3 en değerli semboldür, 2 en az değerli semboldür.

İkilik Sayılar

- İkili bir sayı yalnızca iki simgeye (0 ve 1) sahiptir ve her sütun önceki sütunun iki katı kadar bir ondalık değere sahiptir.

Binary Number: 1001 0011

128	64	32	16	8	4	2	1
1	0	0	1	0	0	1	1

$$(1 \times 128) + (0 \times 64) + (0 \times 32) + (1 \times 16) + (0 \times 8) + (0 \times 4) + (1 \times 2) + (1 \times 1) = 147$$

Converting a binary number to its decimal value.

Hexadecimal

- Onaltılık sayıları inceleme aşamasında önemlidir çünkü ham verilere bakarken sıklıkla kullanılırlar. Onluk sayıdan ayırt etmek için onaltılık bir sayının önünde '0x' yazılmaktadır.

Table 2.1. Binary, decimal, and hexadecimal conversion table.

Binary	Decimal	Hexadecimal
0000	00	0
0001	01	1
0010	02	2
0011	03	3
0100	04	4
0101	05	5
0110	06	6
0111	07	7
1000	08	8
1001	09	9
1010	10	A
1011	11	B
1100	12	C
1101	13	D
1110	14	E
1111	15	F

Hexadecimal Number: 0x8BE4

Reference
0xB = 11
0xE = 14

4,096	256	16	1
8	11	14	4

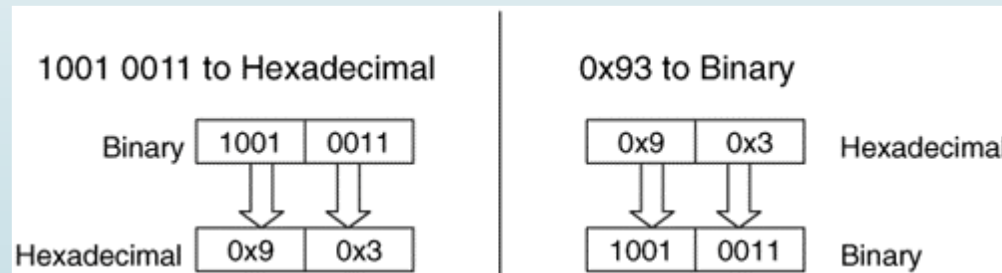
$$(8 \times 4,096) + (11 \times 256) + (14 \times 16) + (4 \times 1) = 35,812$$

Converting a hexadecimal value to its decimal value.

Binary Hexadecimal Dönüşümü

- Belirli sayıda bitle temsil edilebilecek maksimum değeri öğrenmek istendiğinde her bir sütundaki sembol sayısını sütun sayısı ile kuvvetini alıp 1 çıkartırız.
- Örneğin, ikili bir sayı ile 2'deki değeri bit sayısına kuvvetini aldıktan sonra 1 çıkartırız. Dolayısıyla, 32 bitlik bir değer için maksimum onluk değeri elde edilir.

Converting between binary and hexadecimal requires only lookups from Table.



Veri Boyutları

- Dijital verileri depolamak için bir depolama aygıtında bir konum tahsis edilmesi gerekmektedir.
- Bunu adınızdaki her karakteri, küçük kutulara girmeniz gereken kağıt formları gibi düşünebilirsiniz.
- Adınız ve adres alanlarınız, sayfanızda adınızdaki karakterler için alan ayrılmıştır. Sayısal veriler içinde, bir diskte veya bellekte belirli bir değerdeki baytlar için yer ayrılır.
- Bir bayt, verilere genellikle tahsis edilen en küçük alan miktarıdır. Bir bayt yalnızca 256 değeri barındırabilir, bu nedenle baytlar daha büyük sayıları depolamak için birlikte gruplandırılır. Tipik boyutlar arasında 2, 4 veya 8 bayt bulunur.

Endian Kavramı

- Alfabelerde olduğu gibi sağdan sola ya da soldan sağa doğru yazım şeklini işlemciler de byte'ları saklarken önemli byte'ın solda veya sağda olmasına göre sınıflandırılır. Buna **endianness** da denir.
- İşlemcilerin byte'ları saklarken kullandığı sıralamadır. Büyük Sonlu (Big-Endian) ve Küçük Sonlu (Little-Endian) olmak üzere ikiye ayrılır.

Little Endian (Küçük Sonlu)

- i386 ve devamı olan işlemciler Little-Endian kullanır.
- Elimizde 0x34E2561A 4 Byte'lık Integer verimiz olsun. Sağdan itibaren başlanarak 0x1A 0x56 0xE2 0x34 olarak saklanır.

Actual Value: 0x12345678						
	79	80	81	82	83	84
Big-endian	00	12	34	56	78	00
	79	80	81	82	83	84
Little-endian	00	78	56	34	12	00

Big Endian (Büyük Sonlu)

- Sun Sparc, Motorola 68K, PowerPC ve Java Sanal İşlemcisi (Java VM) Big-Endian kullanır.
- 0x7654139A verisi soldan itibaren başlanarak 0x76 0x54 0x13 0x9A olarak saklanır.
- Arap rakamlarında olduğu gibi (İngilizce veya Türkçede kullandığımız rakamlar) önemli byte'ın solda olduğu sıralamaya **Big-endian** denir.

Actual Value: 0x12345678						
	79	80	81	82	83	84
Big-endian	00	12	34	56	78	00
	79	80	81	82	83	84
Little-endian	00	78	56	34	12	00

Ortak Dosya Formatları ve Endian Sırası

- **Adobe Photoshop** — Big Endian
- **BMP (Windows and OS/2 Bitmaps)** — Little Endian
- **DXF (AutoCad)** — Variable
- **GIF** — Little Endian
- **IMG (GEM Raster)** — Big Endian
- **JPEG** — Big Endian
- **FLI (Autodesk Animator)** — Little Endian
- **MacPaint** — Big Endian
- **PCX (PC Paintbrush)** — Little Endian
- **PostScript** — Not Applicable (text!)
- **POV (Persistence of Vision ray-tracer)** — Not Applicable (text!)
- **QTM (Quicktime Movies)** — Little Endian (on a Mac!)
- **Microsoft RIFF (.WAV & .AVI)** — Both
- **Microsoft RTF (Rich Text Format)** — Little Endian
- **SGI (Silicon Graphics)** — Big Endian
- **Sun Raster** — Big Endian
- **TGA (Targa)** — Little Endian
- **TIFF** — Both, Endian identifier encoded into file
- **WPG (WordPerfect Graphics Metafile)** — Big Endian (on a PC!)
- **XWD (X Window Dump)** — Both, Endian identifier encoded into file

String ve Karakter Kodlaması

- En yaygın teknik, karakterleri ASCII veya Unicode kullanarak kodlamaktır.
- ASCII, Amerikan İngilizcesindeki karakterlere sayısal bir değer atar.
- Örneğin, 'A' harfi 0x41'e, '&' ise 0x26'ya eşittir.
- En büyük tanımlanmış değer 0x7E'dir, yani her karakteri saklamada 1 bayt kullanılabilir.
- Kontrol karakterleri tanımlanmıştır ancak yazdırılmaz.

Table 2.2. Hexadecimal to ASCII conversion table.

00 – NULL	10 – DLE	20 – SPC	30 – 0	40 – @	50 – P	60 – `	70 – p
01 – SOH	11 – DC1	21 – !	31 – 1	41 – A	51 – Q	61 – a	71 – q
02 – STX	12 – DC2	22 – "	32 – 2	42 – B	52 – R	62 – b	72 – r
03 – ETX	13 – DC3	23 – #	33 – 3	43 – C	53 – S	63 – c	73 – s
04 – EOT	14 – DC4	24 – \$	34 – 4	44 – D	54 – T	64 – d	74 – t
05 – ENQ	15 – NAK	25 – %	35 – 5	45 – E	55 – U	65 – e	75 – u
06 – ACK	16 – SYN	26 – &	36 – 6	46 – F	56 – V	66 – f	76 – v
07 – BEL	17 – ETB	27 – '	37 – 7	47 – G	57 – W	67 – g	77 – w
08 – BS	18 – CAN	28 – (38 – 8	48 – H	58 – X	68 – h	78 – x
09 – TAB	19 – EM	29 –)	39 – 9	49 – I	59 – Y	69 – i	79 – y
0A – LF	1A – SUB	2A – *	3A – ;	4A – J	5A – Z	6A – j	7A – z
0B – BT	1B – ESC	2B – +	3B – ;	4B – K	5B – [6B – k	7B – {
0C – FF	1C – FS	2C – ,	3C – <	4C – L	5C – \	6C – l	7C –
0D – CR	1D – GS	2D – -	3D – =	4D – M	5D –]	6D – m	7D – }
0E – SO	1E – RS	2E – .	3E – >	4E – N	5E – ^	6E – n	7E – ~
0F – SI	1F – US	2F – /	3F – ?	4F – O	5F – _	6F – o	7F –

String ve Karakter Kodlaması

- Endian sıralaması, karakterlerin nasıl depolandığı konusunda bir rol oynamaz çünkü bunlar ayrı 1 bayt değerlerdir.
- Bu nedenle, kelime veya cümledeki ilk karakter her zaman ilk atanmış baytta bulunur.
- Bir kelimedeki veya cümledeki bayt dizisine bir dizge denir.
- Çoğu zaman dizge, 0x00 olan NULL simgesiyle biter

- Şekilde ASCII'de saklanan örnek bir stringi göstermektedir.
- Dize 10 sembolü içeriyor ve bayt 64'den başlayarak 11 bayt ayrılıyor ve NULL ile sonlandırılıyor.

String: 1 Main St.										
64	65	66	67	68	69	70	71	72	73	74
31	20	4D	61	69	6e	20	53	74	2e	00
1		M	a	i	n		S	t	.	

Unicode

- Amerikan İngilizcesi'ni kullanırsanız, ASCII güzel ve basittir, ancak yerli sembolleri gösterilemediği için dünyanın geri kalan kısmı için oldukça sınırlıdır.
- Unicode, bir sembolün sayısal halini depolamak için 1 bayttan fazla alan kullanarak bu sorunu çözmeye yardımcı olur.
- Sürüm 4.0 Unicode standardı, ASCII'nin gerektirdiği 1 bayt yerine 4 baytlık karakteri gerektiren 96.000 karakteri desteklemektedir.

Unicode

- Unicode karakteri saklamak için üç yol vardır.
- İlk yöntem olan UTF-32, her karakter için 4 baytlık bir değer kullanır ve bu da çok fazla alanı boşa harcamasına neden olabilir.
- İkinci yöntem olan UTF-16, en çok kullanılan karakterleri 2 baytlık bir değere, daha az kullanılan karakterleri de 4 baytlık bir değere kaydeder. Bu nedenle, ortalama olarak bu, UTF-32'den daha az alan kullanır.
- Üçüncü yöntem UTF-8 olarak adlandırılır ve bir karakteri saklamak için 1, 2 veya 4 bayt kullanır. Her karakter farklı sayıda bayt gerektirir ve en sık kullanılan baytlar yalnızca 1 bayt kullanır.
- UTF-8 ve UTF-16 her karakteri depolamak için değişken sayıda bayt kullanır ve bu nedenle verileri daha zor işler hale getirir. UTF-8 en az boşa harcanan alana sahip olduğu için sıklıkla kullanılır

Veri Yapıları

- Verilerin belirli dosya sistemlerinde nasıl depolandığına bakmadan önce, veri organizasyonunun genel yapısına bakmamız gerekir.
- Bilgisayarlar, veri yapıları nedeniyle verilerin düzenini bilir. Bir veri yapısı, verilerin nasıl düzenlendiğini açıklar. Bir şablon veya harita gibi çalışır. Veri yapısı alanlara ayrılır ve bu bilgiler veri ile birlikte kaydedilmemesine rağmen her alan bir boyut ve ada sahiptir.
- Örneğin, veri yapımız 'sayı' olarak adlandırılan ilk alanı tanımlayabilir ve 2 bayt uzunluğa sahip olabilir. Ev numarasını adresimize kaydetmek için kullanılır. 'Numara' alanının hemen ardından 'sokak' alanı ve 30 bayt uzunluğundadır.

Table 2.3. A basic data structure for the house number and street name.

Byte Range	Description
0–1	2-byte house number
2–31	30-byte ASCII street name

Veri Yapıları

- Bir depolama aygıtına veri yazmak istiyorsanız, her bir değerin nerede yazılması gerektiğini belirlemek için uygun veri yapısına bakarız.
- Örneğin, «**1 Ana Cadde**» adresini saklamak istersek, adresi ilk olarak numaraya ve isme bölmeliyiz. Depolama alanımızın **0 - 1** baytlarına sayı 1 yazıp sonrasında "**Main St.**" yazarız. Her karakter için ASCII değerlerinin ne olduğunu belirleyerek 2'den 9'a kadar baytlar kullanılır.
- Geriye kalan baytlar ihtiyacımız olmadığından 0 olarak ayarlanabilir.
- Bu durumda, 32 baytlık bir saklama alanı tahsis edilerek depolama aygıtında herhangi bir yerde saklanabilir.
- Bayt uzaklıkları tahsis edildiğimiz alanın başlangıcına göre değişir.
- Ev numarasındaki bayt sırasının bilgisayarın endian yapısına bağlı olduğunu unutmayın.

Veri Yapıları

- Depolama aygıtından veri okumak istediğimizde, verilerin nereden başladığını belirleriz ve daha sonra, gerekli değerlerin nerede olduğunu bulmak için veri yapısına bakarız.
- Örneğin, yazdığımız verileri okumak istiyoruz. Depolama aygıtında nerede başladığını öğrenip veri yapısı şablonumuzu uyguluyoruz. İşte ham verileri okuyan bir aracın çıktısı.

```
0000000: 0100 4d61 696e 2053 742e 0000 0000 0000 ..Main St.....
0000016: 0000 0000 0000 0000 0000 0000 0000 0000 .....
0000032: 1900 536f 7574 6820 5374 2e00 0000 0000 ..South St.....
0000048: 0000 0000 0000 0000 0000 0000 0000 0000 .....
```

- Soldaki sütun onluk satırın bayt ofsetidir, orta 8 sütun verilerin 16 bayt karşılığıdır ve son sütun verinin ASCII karşılığıdır.
- '.' ifadesi yazdırılabılır bir ASCII karakteri bulunmadığını gösterir.
- Her onaltılık simgenin 4 biti temsil ettiğini bu yüzden bir baytın 2 onaltılık sembol gerektiğini unutmayın.

Veri Yapıları

- Veri yapımızın düzenine bakılır ve her adresin 32 bayt olduğunu görülür, bu nedenle ilk adres bayt 0 ila 31 arasındadır.
- Bayt 0 ila 1, 2 baytlık sayı alanıdır ve sokak ismi 2 ila 31 baytlar arasında olmalıdır.
- Bayt 0 - 1 bize 0x0100 değerini gösterir. Veriler, little-endian olan Intel sisteminden gelmektedir. Bu nedenle 0x01 ve 0x00 sırasını 0x0001 olarak değiştiririz.
- Bunu, onluk sayıya çevirdiğimizde, 1 numaralı sayıyı elde ederiz.
- Veri yapısındaki ikinci alan, 2 ile 31 arasındaki baytlardır ve bir ASCII dizesidir; bu, sistem karakter gösterimi olduğunda endian sıralamasından etkilenmez, bu yüzden, yeniden sıralamak zorunda kalınmaz.
- Her bir bayt'ı ASCII eşdeğerine dönüştürebiliriz ya da bu durumda hile yapabilir ve "Ana St'i" görmek için sağdaki sütuna bakabiliriz. Bu, daha önce yazmış olduğumuz değerdir. Başka bir adres veri yapısının bayt 32'de başladığını ve bayt 63'e kadar uzandığını görüyoruz.
- Örneğin, dosya sisteminin ilk sektörü genellikle düzinelerce alan içeren büyük bir veri yapısını içerir ve onu okumamız için dosya sisteminin boyutunun bayt 32 - 35'te verildiğini bilmemiz gerekir. Birçok dosya sistemi çoklu konumlar kullanan farklı büyük veri yapılarına sahiptir.

Ödev

Dosya Sistemlerinin Veri yapılarını araştırınız.

Ntfs, Fat, UFS, HFS, ext

Flag (Bayrak) Değerleri

- Gerçek veri yapılarına bakmadan önce bilinmesi gereken son veri türü flag(bayrak) lardır. Bazı verileri tanımlamak için kullanılır.
- Bazı veriler, 1 veya 0 ile temsil edilebilecek bir durumun oluşup oluşmayacağını belirlemek için kullanılır.
- Örneğin, bir bölümün önyüklenebilir olup olmayacağı olabilir. Bu bilgiyi depolamanın bir yolu tam bir bayt ayırmak ve 0 veya 1 değerini kaydetmektir. Bununla birlikte, çok fazla alan boşa gider, çünkü yalnızca 1 bit gereklidir, ancak 8 bit tahsis edilmiştir. Daha verimli bir yöntem, bu ikili koşullardan birkaçını bir değere paketlemektir.
- Değerin her biti bir özellik veya seçeneğe karşılık gelir. Bunlara sıkça bayraklar denir, çünkü her bit bir durumun doğru olup olmadığını belirtir. Bir bayrak değerini okumak için, sayıyı ikili haline getirip her biti incelememiz gerekir. Bit 1 ise bayrak ayarlanır.

Flag Değerleri

Table 2.4. A data structure with a flag value.

Byte Range Description

0–1	2-byte house number
2–30	29-byte ASCII street name
31–31	Flags
32–47	16-byte ASCII city name (if flag is set)

- Önceki sokak adresi veri yapımızı biraz daha karmaşık hale getirerek bir örneğe göz atalım.
- Orijinal veri yapısı, ev numarası için bir alan ve sokak adı için bir alan içeriyordu.
- Şimdi sokak alanının arkasına isteğe bağlı bir 16 baytlık şehir adı ekleyeceğiz. Şehir adı isteğe bağlı olduğundan, var olup olmadığını belirlemek için bir bayrağa ihtiyacımız var.
- Bayrak, bayt 31'de ve şehir var olduğunda bit 0'dır (yani, 0000 0001).
- Şehir olduğu zaman, veri yapısı 32 yerine 48 bayt'tır..

Flag Değerleri

```
0000000: 0100 4d61 696e 2053 742e 0000 0000 0000 ..Main St.....
0000016: 0000 0000 0000 0000 0000 0000 0000 0061 .....a
0000032: 426f 7374 6f6e 0000 0000 0000 0000 0000 Boston.....
0000048: 1800 536f 7574 6820 5374 2e00 0000 0000 ..South St.....
0000064: 0000 0000 0000 0000 0000 0000 0000 0060 .....`
```

- İlk satırda, önceki örnekteki ile aynı verileri görüyoruz. Adres 1 Ana St'dir ve bayt 31'deki bayrak değeri 0x61'dir.
- Bayrak sadece 1 bayt boyutunda, dolayısıyla endian yapısında endişelenmeye gerek yoktur. Bu değere ikili olarak bakmamız gerekiyor, bu yüzden daha önce Tablo 2.1'de verilen arama tablosunu kullanıyor ve 0x6 ve 0x1 değerlerini ikili değer 0110 0001'e dönüştürüyoruz.
- En küçük anlamlı bitin şehir olarak ayarlandığını görüyoruz. Diğer bitler, bu adresin işletme adresi olarak tanımlanması gibi diğer bayrak değerleri içindir. Bayrağa dayanarak, 32-47. Baytların "Boston" olan şehir adını içerdiğini biliyoruz.
- Bir sonraki veri yapısı 48 bayttan başlar ve bayrak alanı 79 baytındadır. 0x60 değeridir ve şehir bayrağı ayarlanmamıştır. Bu nedenle, üçüncü veri yapısı, bayt 80'de başlayacaktır.

Boot Süreci

- Bu dersin sonraki bölümlerinde, verilerin bir diskte nerede bulunduğunu ve hangi verilerin bilgisayarın çalışması için gerekli olduğu gösterilecektir.
- Çoğu zaman, bilgisayar başlatıldığında kullanılan makine kodları olan önyükleme kodları bulunmaktadır.
- Bu bölüm önyükleme işlemini ve önyükleme kodunun bulunabileceği yerleri göstermektedir.
- Çoğu disk önyükleme kodu için yer ayırır, ancak bu kullanılmaz. Bu bölüm, hangi önyükleme kodunun kullanıldığını tanımlamanıza yardımcı olacaktır.

Merkezi İşlem Birim ve Makine Kodları

- Modern bilgisayarın beyni bir veya daha fazla Merkezi İşlemci Birimi (CPU) 'dir.
- CPU'lar talimatlarını bellekten alırlar. CPU talimatları makine kodunda yazılmıştır ve okunması zor ve kullanıcı dostu değildir.
- Her bir makine kodu talimatı birkaç bayt uzunluğundadır ve ilk çift bayt, talimat türünü belirtir ve buna **opkod** denir.
- Örneğin, 3 değeri bir toplama talimatı için olabilir. İşlevin ardından talimat argümanlarıdır

Merkezi İşlem Birim ve Makine Kodları

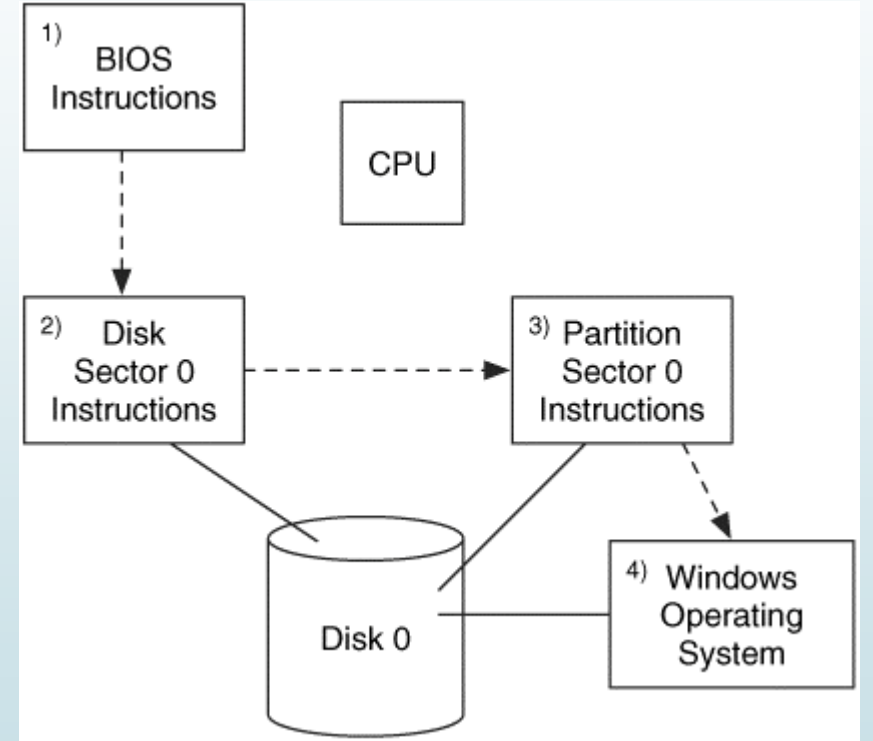
- Bu derste daha fazla ayrıntıya ihtiyacımız yok, ancak basit bir örnekle açıklayalım.
- Makine dili talimatlarından biri, değerleri CPU'nun kayıtlarına taşımaktır.
- Kayıtlar, CPU'ların veriyi depoladığı yerlerdir. Bunu yapmak için bir assembly kodu olan `MOV AH, 00` kullanılır, burada 0 değeri AH kaydına taşınır.
- Makine kodu eşdeğeri, 0xB400 onaltılık değeri olup, burada B4, `MOV AH` için opcode'dır ve 00, onaltılı olarak, taşınması için olan değerdir.
- Makine kodunu sizin için assembly koduna çevirecek araçlar vardır.

Önyükleme (Boot) Kodu Konumları

- Bir bilgisayarı başlatmak için, önyükleme kodu olarak da bilinen CPU işlemlerini barındıran bir aygıt ihtiyacımız vardır.
- Çoğu sistemde, ilk adım tüm donanımı çalışır duruma getirme ve ikinci adım ise OS'u veya diğer yazılımları çalıştırmayı içeren süreçtir.
- Tüm volüm ve dosya sistemlerinin önyükleme kodunu depoladığı belirli bir konumu vardır fakat her zaman gerekli değildir.
- İşlemci çalıştırıldığında ROM bellekte bulunan belirli bir konumdan okuma yapar. ROM'daki kodlar, sistemi donanım için inceleme ve yapılandırmaya zorlar. Donanım yapılandırıldıktan sonra, CPU ek önyükleme kodu içerebilecek bir aygıt arar. Böyle bir aygıt bulursa, önyükleme kodu yürütülür ve kod belirli bir işletim sistemini bulmaya ve yüklemeye çalışır.

Örnek Önyükleme Süreci

- Microsoft Windows sisteminin önyükleme işlemine kısaca bir göz atalım.
- Sistem açık olduğunda, CPU, Temel Giriş / Çıkış Sisteminden (BIOS) talimatları okur ve desteklemek üzere yapılandırılmış olan sabit diskleri, CD sürücülerini ve diğer donanım aygıtlarını arar.
- Donanım bulunduktan sonra, BIOS disketleri, sabit diskleri ve CD'leri yapılandırılmış bazı sırayla inceler ve önyükleme kodu için ilk sektöre bakar.
- Önyüklenebilir bir diskin ilk bölümündeki kod, CPU'nun bölüm tablosunu işleyip Windows işletim sisteminin bulunduğu önyüklenebilir bölümü bulmasına neden olur.

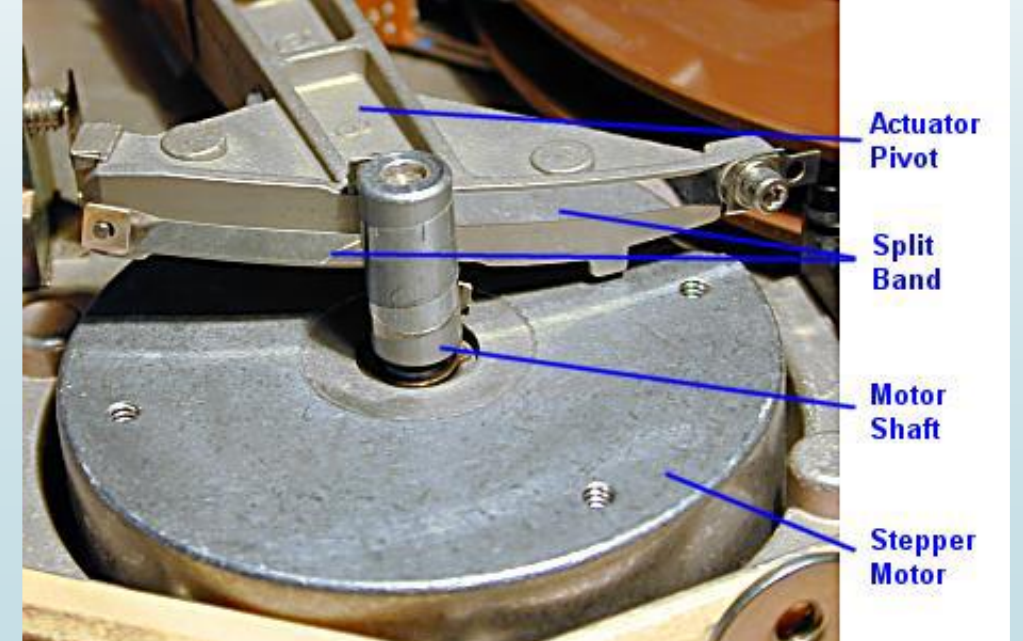
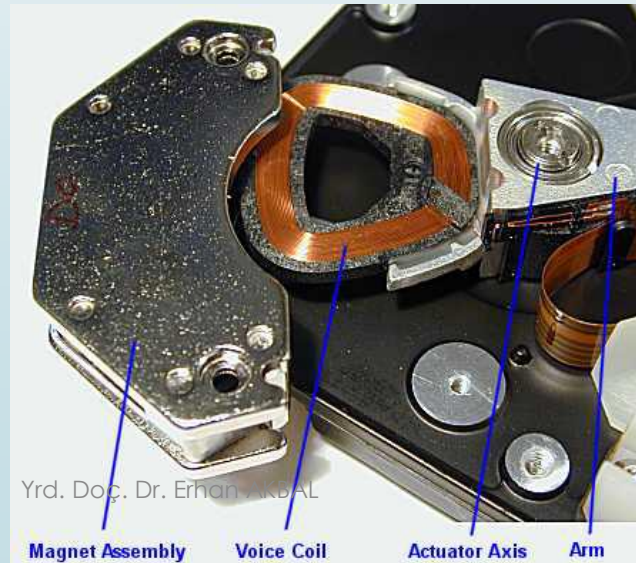
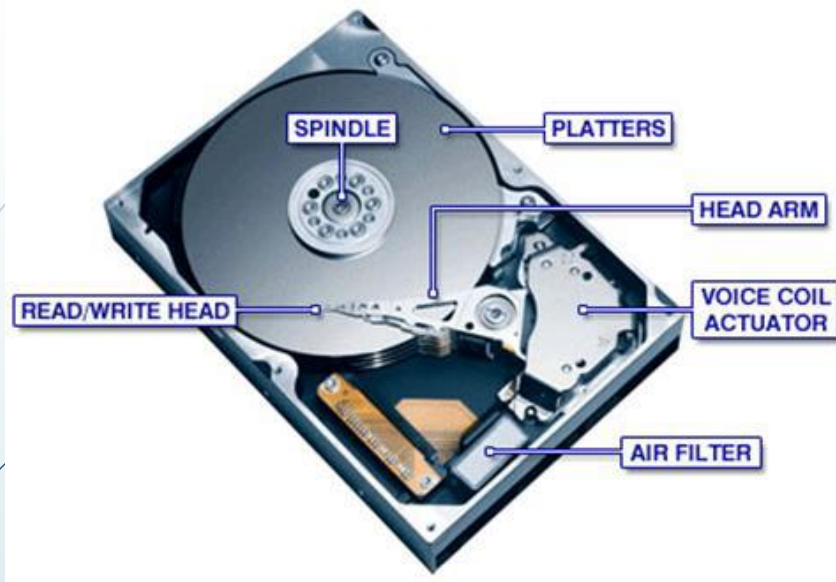


Hard Disk Teknolojisi

- Adli bilişim uzmanının bilmesi gereken en temel donanım yapısı hard disklerdir. Çünkü hard diskler dijital delillerin en önemli parçasıdır.
- Bu bölüm, sabit disk temellerini kapsamaktadır ve erişim metotları, yazma engelleme ve verilerin gizlenebileceği yerler gibi bir inceleme uzmanının öğrenmesi gereken alanlardır.
- İlk bölüm, bir diskin nasıl çalıştığına genel bir bakış sunarken, sonraki iki bölüm AT Attachment (ATA / IDE) diskleri ve Small Computer Systems Interface (SCSI) diskleri kapsar.

Sabit Disk Geometrisi ve İç Donanımları

30



Sabit Disk Geometrisi ve İç Donanımları

- Verilerin nasıl saklandığına ilişkin temel bilgiler ,eski dosya sistemleri, partition düzenleri, disk geometrisi, modern disklerde kullanılan gizlenmiş olan diğer iç birimlerin anlaşılmasını gerekmektedir.
- Bu nedenle, disk geometrisi hakkında bilgi sahibi olmak, bir dosya sistemindeki bazı değerlerin anlaşılmasına yardımcı olacaktır.
- Bu bölümün amacı, sabit diskleri tamir etmeyi öğrenmek değildir. Bunun yerine, amaç içeride olup bitenlerin anlaşılmasıdır.

Sabit Disk Geometrisi ve İç Donanımları

- Sabit diskler, birbirinin üstüne istiflenmiş ve aynı anda dönen bir veya daha fazla dairesel tabağa sahiptir.
- Her plakanın alt kısmı ve üstü bir manyetik ortam ile kaplanmıştır ve disk üretildiğinde plakalar üniform ve boştur.
- Diskin içinde ileri geri hareket eden bir kol bulunur ve her bir tabağın üstünde ve altında bir kafa vardır, bu da veri okuyup yazabilir, ancak yalnızca bir kafa bir kerede okuyabilir veya yazabilir.
- Tracklar ve sektörlerde veri yapıları oluşturmak için boş plakalar üzerinde düşük düzeyli bir format uygulanır.

Sabit Disk Geometrisi ve İç Donanımları

- Bir track, plakayı çevreleyen dairesel bir halkadır.
- Çalışan bir track bir şerite benzer; böylece tüm dairenin etrafında dolaşırsanız, başlanılan aynı yere dönülür.
- Sabit diskteki her track, 0'dan başlayarak dışarıdan içe bir adres verilir.
- Örneğin, her bir plaka 10,000 parçaya ayrıldıysa, her bir plakanın dış iz numarası 0, en iç (dairesinin merkezine en yakın) 9999 olur.
- Her plakanın düzeni aynı olduğundan ve her bir plakaya ait tracklara aynı adres verildiğinden silindir terimi, tüm plakalarda belirli bir adresteki tüm parçaları tanımlamak için kullanılır.
- Her bir track, sabit diskteki en küçük adreslenebilir depolama birimi olan ve genellikle 512-4096 bayt olan sektörlerle ayrılmıştır.

Sabit Disk Geometrisi ve İç Donanımları

- Her sektörde her bir track için 1'den başlayarak bir adres verilir. Bu nedenle trackları göstermek için silindir adresini (C), plaka ve kenarı alacak baş numarasını (H) ve track üzerindeki sektöri göstermek için (S) kullanarak adresleme yapabiliriz. Bu adresleme yöntemi ve disk geometrisine CHS denir.

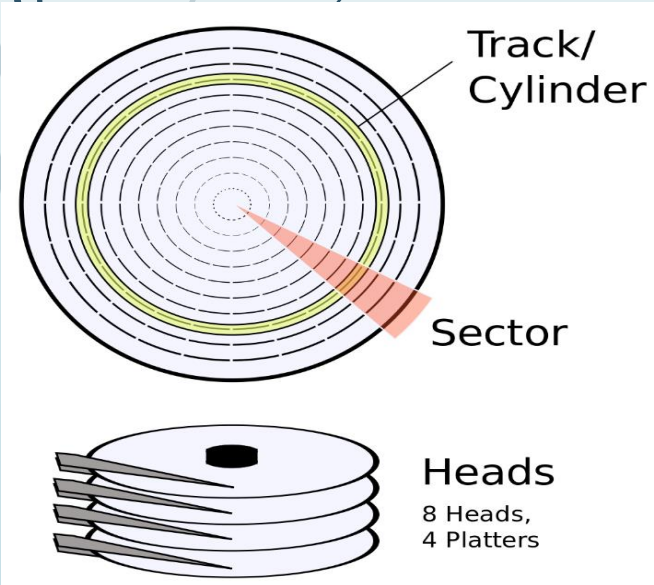
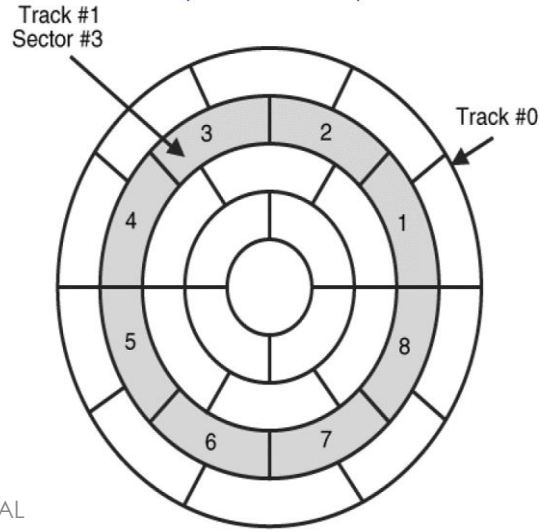
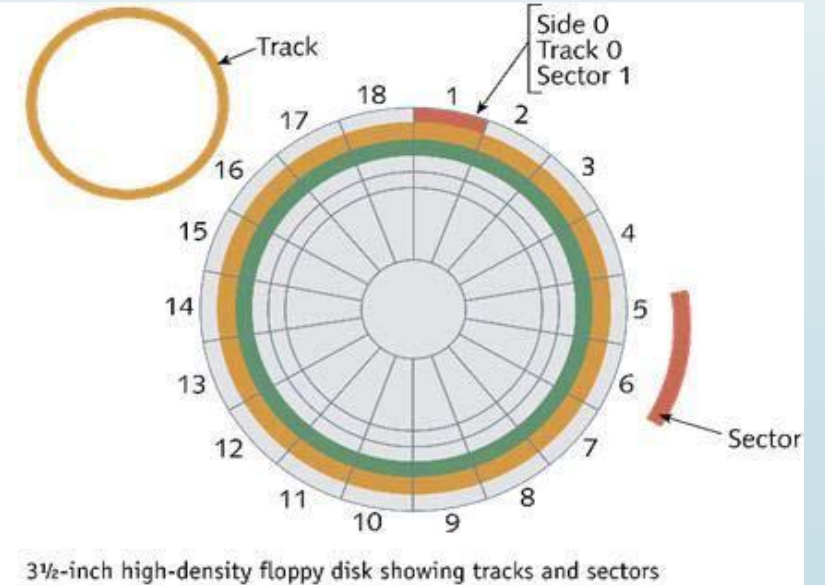


Figure 2.9. Disk geometry of one platter showing the track (or cylinder) and sector addresses (not even close to scale).



rhah AKBAL



Sabit Disk Geometrisi ve İç Donanımları

- CHS adreslemesinin yerine, Mantıksal Blok Adresi (Logical Block Address - LBA) kullanılır ve her bir sektör için sıralı adresler atar. LBA adresi, fiziksel konum ile ilgilenmez.
- Bir sektör hatalı olduğunda kullanıcı verilerini depolamak için artık bu alanı kullanmayabilir. Eski nesil disklerde, bad sektörlerin nerede olduğunu bilmek ve onları dosyalar için ayırmamak işletim sisteminin sorumluluğundaydı. Kullanıcılar hangi sektörleri bad sektör olarak görmesi gerektiğini elle diske söyleyebilirdi.
- Aslında birçok dosya sistemi, sektörleri bad olarak işaretleme seçeneği sunar. Modern diskler bad sektörü belirleyebilir ve adresi disk üzerinde başka bir yerde iyi bir konuma yeniden eşleştirebilir. Kullanıcı arka planda olan süreçlerin neler olduğunu asla bilmez.

ATA/IDE Arayüzü

- AT Attachment (ATA) arayüzü en popüler sabit disk arabirimidir. Bu arayüzü kullanan diskler IDE diskleri denir, ancak IDE sadece Integrated Disk Electronics'i (Orijinal Disk Elektroniği) temsil eder ve daha eski disklerde yer almayan mantık paneli bulunan bir sabit disk tanımlar.
- "IDE" disklerinin kullandığı gerçek arabirim ATA'dır.
- ATA spesifikasyonları Uluslararası Bilgi Teknolojisi Standartları Komitesi (INCITS) de bulunan T13 Teknik Komitesi (<http://www.t13.org>) tarafından geliştirilmiştir.

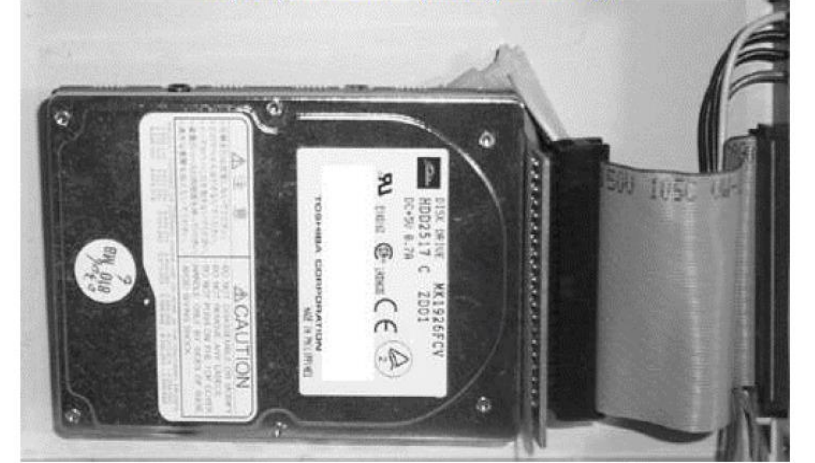
ATA/IDE Arayüzü

- ▶ ATA diskler, modern sistemlerin ana kartına yerleştirilmiş bir denetleyici gerektirir. Denetleyici, şerit kablo kullanarak bir veya iki ATA diskinde komutlar gönderir. Kablonun maksimum uzunluğu 18 inçtir ve 40 iğne vardır.
- ▶ Teller arasındaki paraziti önlemek için fazladan teller bulunur. Dizüstü bilgisayarlar sıklıkla daha küçük bir diske sahipler ve güç için iğneler içeren 44 pinli bir arabirim kullanırlar.
- ▶ Her kanal iki diske sahip olabilir ve diğerleri üzerinde kontrol sahibi olmasa da, "master (ana)" ve "slave (köle)" terimleri kendilerine verilir.

Figure 2.10. An ATA disk with the 40-pin connector, jumpers, and power connector.



Figure 2.11. A 44-pin ATA laptop drive connected to a 40-pin ATA ribbon cable using an adaptor (Photo courtesy of Eoghan Casey).



Sektör Adres Türleri

- İki farklı fiziksel adresleme yöntemi vardır. Eski sabit diskler, disk geometrisini ve CHS yöntemini kullanır. CHS adresleme şeması iyi gibi gözükse de, çok sınırlayıcı olduğu kanıtlanmıştır ve artık çok fazla kullanılmamaktadır.
- Orijinal ATA spesifikasyonu, 16 bitlik silindir değeri, 4 bit baş değeri ve 8 bitlik bir sektör değeri kullanır. Ancak daha eski BIOS'lar 10 bitlik silindir, 8 bit baş ve 6 bitlik bir sektör değeri kullanırlar. Bu nedenle, sabit disk ile BIOS üzerinden iletişim kurmak için, her değer için en küçük boyut kullanılmalıdır; bu da yalnızca 504 MB diske izin verir.
- 504 MB sınırını aşmak için, adres aralıklarını ATA spesifikasyonunun benzer adreslerine çeviren yeni BIOS'lar geliştirilmiştir. Bir uygulama silindir 8, kafa 4 ve sektör 32 den veri istediğinde BIOS bunu silindire 26, kafa 2, sektör 32 olarak dönüştürerek diskten isteyebilir.
- Dönüştürme işlemi için BIOS, diskte var olandan farklı olan bir sabit disk geometrisini rapor eder. Çeviri işlemi, 8.1 GB'dan daha büyük diskler için çalışmaz.

Sektör Adres Türleri

- Adres çevirisi gerçekleştiren BIOS'lar artık yaygın değildir, ancak bir inceleme uzmanı böyle bir sistemle karşılaşursa zorlanabilir. Diski sistemden çıkarıp kendi sistemine taktığında çeviri olmayabilir veya farklı olabilir ve yanlış sektörler döndürüleceğinden doğru bir analiz yapılamayabilir.
- Bu sorunu çözmek için bir inceleme uzmanının orijinal sistemi kullanması veya benzer çeviriyi yapan benzer bir sistem bulması gerekir. Bir incelemeci, bir sistemin, satıcılarının web sitesinde bulunan BIOS sürümüne bakarak veya BIOS'taki başvuruları arayarak çeviri yapıp yapmadığını belirleyebilir.

Sektör Adres Türleri - LBA

- Dönüştürmeden kaynaklı 8.1 GB sınırını aşmak için, CHS adresleri kullanımı bırakılarak Mantıksal Blok Adresleri (LBA) standart haline geldi.
- LBA, her sektörü adreslemek için 0'dan başlayan tek bir sayı kullanır ve ilk resmi ATA spesifikasyonundan beri desteklenmiştir. LBA ile yazılımın geometri hakkında bir şey bilmesine gerek yoktur; Yalnızca tek bir numarayı bilmelidir. CHS adresleri için destek ATA-6'daki ATA spesifikasyonundan kaldırılmıştır.
- Ne yazık ki, bazı dosya sistemi ve veri yapıları hala CHS adreslerini kullanmaktadır, bu nedenle CHS'den LBA'ya dönüşüm yapabilmeyi bilmek gereklidir. LBA 0 adresi, CHS 0,0,1 adresine ve LBA 1 adresi, CHS adresi 0,0,2 ye karşılık gelmektedir. Track içindeki tüm sektörler kullanıldığında, aynı silindirdeki bir sonraki kafadaki ilk sektör CHS adresi 0,1,1 olarak Bunu, alt plakanın dış halkasını doldurarak, daha sonra üst plakaya ulaşana kadar plakaları yukarıya doğru hareket ettirerek görselleştirebiliriz.
- **$LBA = (((CYLINDER * \text{Her Silindirdeki Head Sayısı}) + HEAD) * \text{Her Traktaki Sektör Sayısı}) + SECTOR - 1$**

Sektör Adres Türleri - LBA

- Örneğin, her silindirde 16 baş ve Track başına 63 sektörü olan bir disk düşünelim. Silindir 2 (C), kafa 3 (H) ve sektör 4 (S) olan CHS adresimiz olsaydı, LBA'ya dönüşümü şu şekilde olurdu:
- **$2208 = (((2 * 16) + 3) * 63) + 4 - 1$**

Arayüz Standartları

- Tüketici sabit disk arenasında kafa karıştırıcı olabilecek birçok arayüz terimi vardır.
- Bazı terimler, standart bir komitenin seçtiği ile aynıdır, fakat bir sabit disk şirketinde başka bir standardı seçebilir. "Geliştirilmiş IDE" ve "Ultra ATA" gibi terimler gayri resmi terimlerdir.
- ATA özelliklerinin yalnızca sabit diskler için geçerli olduğunu unutmayın. CD-ROM'lar ve ZIP diskler gibi çıkarılabilir ortamlar için AT Attachment Paketi Arabirimi (ATAPI) adı verilen özel bir belirtim kullanılır. ATAPI aygıtları genellikle aynı kabloları ve denetleyiciyi kullanır, ancak özel sürücüler gerektirir.

Arayüz Standartları

- Burada bir inceleme uzmanının ilgisini çeken özellikler şunlardır:
- ATA-1: 1994 yılında yayınlandı. Bu özellik CHS ve 28 bit LBA adreslerini destekler.
- ATA-3: 1997'de yayınlandı ve güvenilirlik ve güvenlik özellikleri eklendi. Diskin çeşitli bölümlerini izleyerek güvenilirliği artırmaya çalışan SMART (Kendini İzleme Analizi ve Raporlama Teknolojisi) tanıtıldı.
- ATA / ATAPI-4: Çıkarılabilir ortamın belirtimi olan ATAPI, 1998'de yayınlanan ATA-4'teki ATA spesifikasyonudur. 80 telli kablo bozulmanın azaltılmasına yardımcı olmak için tanıtıldı. ATA-4 de HPA eklendi;
- ATA / ATAPI-6: 2002'de yayınlandı, 48-bit LBA adresleri eklendi, CHS adresleri için destek kaldırıldı ve DCO eklendi.
- ATA / ATAPI-7: 2005 yılında yayınlandı. SATA 1.0, Akış özelliği seti, paket dışı cihazlar için uzun mantıksal / fiziksel sektör özelliği seti özellikleri eklendi.
- ATA / ATAPI-8: 2013 yılında yayınlandı. Kritik OS dosyalarını hızlandırmak için kalıcı önbellek içeren karma sürücü desteği geldi.

DMA (Direct Memory Access)

- Doğrudan belleğe erişim (Direct memory access; DMA), modern bilgisayarlarda bulunan, merkezi işlem biriminden bağımsız olarak okuma ve/veya yazmak için, belirli donanım alt sistemleri içinde sistem belleğine erişim sağlayan bir özelliktir.
- DMA, disk sürücü kontrol birimleri, grafik kartları, ağ kartları ve ses kartları dahil birçok donanım sistemi tarafından kullanılmaktadır. DMA kanalı olan bilgisayarlar, DMA kanalı olmayan bilgisayarlara nazaran, cihazdan cihaza çok daha hızlı bir şekilde veri transferi yapabilmektedir.
- Bir DMA transferi esas olarak, bir bellek bloğunu bir cihazdan diğerine kopyalar. Her ne kadar transferi CPU başlatsa da, o gerçekleştirmez.
- DMA'nın tipik bir kullanımı, sistem RAM'ından bir bellek blokunun cihazdaki arabelleğe kopyalanmasıdır. Böyle bir işlem işlemciyi geciktirmez, o da, bunun sonucunda, başka görevleri yapmak üzere planlama yapar. DMA transferleri, yüksek performanslı tümleşik sistemler için çok önemlidir.

DMA

DMA Mode	Cycle Time (nanoseconds)	Maximum Transfer Rate (MB/s)	Defining Standard
Single Word Mode 0	960	2.1	ATA
Single Word Mode 1	480	4.2	ATA
Single Word Mode 2	240	8.3	ATA
DMA Mode	Cycle Time (nanoseconds)	Maximum Transfer Rate (MB/s)	Defining Standard
Multiword Mode 0	480	4.2	ATA
Multiword Mode 1	150	13.3	ATA-2
Multiword Mode 2	120	16.7	ATA-2

UDMA (Ultra DMA)

- Sabit disklerin performanslarının artışı ile PIO (Programmed I/O) modları kullanmak performansa engel oldu. PIO arabirimi farklı transfer oranlarını karşılayan değişik modlara gruplandırılmıştır. Bunlara PIO Mode denir.

Ultra DMA Mode	Cycle Time (nanoseconds)	Maximum Transfer Rate (MB/s)	Defining Standard
Mode 0	240	16.7	ATA/ATAPI-4
Mode 1	160	25.0	ATA/ATAPI-4
Mode 2	120	33.3	ATA/ATAPI-4
Mode 3	90	44.4	ATA/ATAPI-5
Mode 4	60	66.7	ATA/ATAPI-5
Mode 5	40	100.0	ATA/ATAPI-6

ATAPI-7 – SATA, eSATA

- SATA yani Seri ATA, bilgisayar donanımı içerisinde bir veri taşıma teknolojisidir. eSATA yani (External SATA) SATA veri standardının harici disklerde kullanılan versiyonudur.
- SATA için standart arayüz Advanced Host Controller Interface (AHCI)'dir. AHCI bize disklerin çalışır durumdayken çıkarılıp takılmasını, Native Command Queuing(Dahili Komut Sıralama) özelliklerini kazandırır. AHCI anakart çipsetinde veya BIOS açık durumda değilse SATA IDE modda çalışır.

Name	Raw data rate	Data rate
eSATA	6 Gbit/s	600 MB/s
eSATAp	3 Gbit/s	300 MB/s
SATA revision 3.2	16 Gbit/s	1.97 GB/s ^[e]
SATA revision 3.0	6 Gbit/s	600 MB/s ^[73]
SATA revision 2.0	3 Gbit/s	300 MB/s
SATA revision 1.0	1.5 Gbit/s	150 MB/s ^[74]

Disk Komutları

Bu bölümde, donanım yazma koruyucularını ve Host Protected Area Alanların ile denetleyicinin ve sabit diskin nasıl iletişim kuracağı konusunda genel bir bakış sunulmaktadır.

Denetleyiciler

- Denetleyici, şerit kablo üzerinden sabit diske komutlar gönderir. Komutlar kablo üzerindeki her iki diske de gönderilir, ancak komutun bir bölümü master veya slave için olup olmadığını belirler. Denetleyici, sabit disk ile küçük bellek parçaları olan kayıt defterine yazarak iletişim kurar.
- Örneğin, denetleyicinin diskin bir kesimini okumak istediği bir durumu düşünün. Okunacak sektör adresini ve sektör sayısını uygun kayıt defterine yazması gerekecektir.
- Komut detayları kayıtlara yazıldıktan sonra, denetleyici sabit diske, komut defterine bakarak okuma eylemini gerçekleştirecektir.

Hard Disk Şifreleri

- ATA-3 spesifikasyonu, BIOS veya çeşitli yazılım uygulamaları aracılığıyla ayarlanabilen şifreler de dahil olmak üzere yeni isteğe bağlı güvenlik özelliklerini tanıttı.
- Sabit disklerdeki iki parola, kullanıcı ve ana parolalardır. Ana parola, kullanıcı parolasının kaybolması durumunda bir şirket yöneticisinin bilgisayara erişebilmesi için tasarlanmıştır.
- Sabit disk, diğer ATA komutlarının birçoğu yürütülemeden önce doğru şifre ile yürütülecek SECURITY_UNLOCK komutunu gerektirir. Doğru şifre girildikten sonra, disk kapatılıncaya kadar disk normal olarak çalışır.

Kullanıcı Korumalı Alanlar (Host Protected Area)

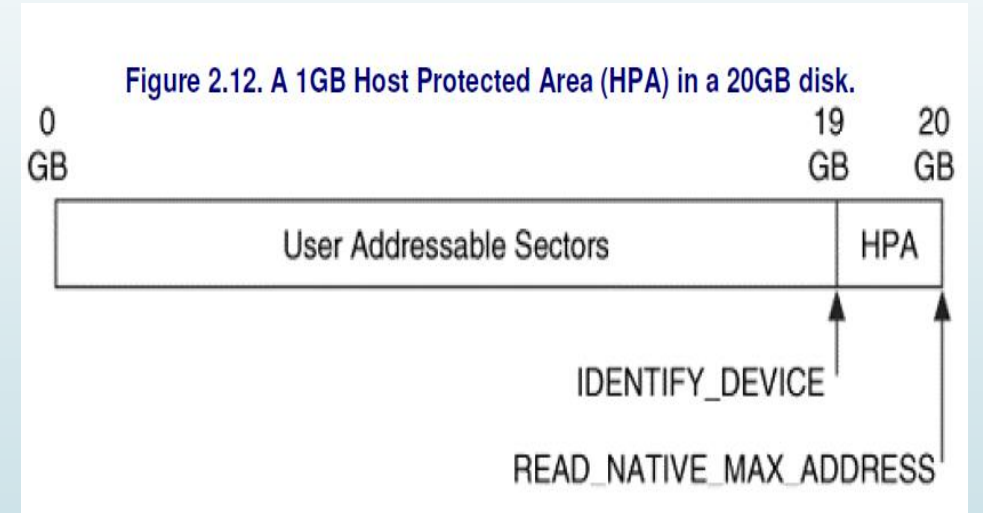
- HPA, verilerin kaydedilmesi için kullanılabilen, disk alanının özel bir alanıdır ve sıradan bir incelemeci bunu göremeyebilir.
- Bu alanın boyutu ATA komutlarını kullanarak yapılandırılabilir ve birçok disk varsayılan olarak 0 boyutuna sahiptir. HPA, ATA-4'te eklenmiştir.
- Amaç, kullanıcı sabit disk içeriğini biçimlendirip sildikten sonra, bilgisayar üreticilerinin silinmeyecek verileri depolayabileceği bir alan ayırmaktır.
- HPA, diskin sonundadır ve kullanıldığında, yalnızca sabit diski yeniden yapılandırarak erişilebilir.

HPA Komutları

- **READ_NATIVE_MAX_ADDRESS** komutu en fazla fiziksel adresi döndürür, ancak **IDENTIFY_DEVICE** komutu yalnızca bir kullanıcının erişebileceği kesim sayısını döndürür. Bu nedenle, bir HPA varsa, **READ_NATIVE_MAX_ADDRESS** diskin gerçek ucunu döndürür ve **IDENTIFY_DEVICE** komutu kullanıcı alanının sonunu döndürür (ve HPA'nın başlangıcı). Sonraki bölümde **READ_NATIVE_MAX_ADDRESS**'in her zaman diskin son fiziksel adresi olmadığını göstereceğini unutmayın.
- HPA oluşturma için **SET_MAX_ADDRESS** komutu ile kullanıcının erişimde kullanacağı maksimum adres değeri belirlenir. Bir HPA silmek için **SET_MAX_ADDRESS** komutu ile **READ_NATIVE_MAX_ADDRESS** komutu ile öğrenilebilecek diskin gerçekte olan maksimum boyutu tekrar ayarlanır.

Örnek HPA

- Örneğin disk 20 GB ise `READ_NATIVE_MAX_ADDRESS` 20GB(41,943,040) diskin sektör sayısını döndürecektir.
- 1 GB lık bir HPA oluşturmak için `SET_MAX_ADDRESS` değeri olarak 39,845,888 ayarlanır.
- Herhangi bir okuma yada yazma durumunda kalan 2,097,152 sektör (1 GB) hata kodu üretir ve `IDENTIFY_DEVICE` komutu maksimum adres değeri olarak 39,845,888 değerini gösterecektir.
- HPA yı silmek için `SET_MAX_ADDRESS` değerini tekrar diskin gerçek sektör sayısına ayarlamamız gerekir. Şekilde gösterilmiştir.



Cihaz Yapılandırma Yerleşimi (Device Configuration Overlay – DCO)

- Verilerin bir HPA'da gizlenmesine ilaveten, Aygıt Yapılandırma Kaplaması (DCO) kullanılarak da gizlenebilir. DCO ATA-6 ya eklenmiştir ve bir sabit diskin belirgin yetenekleri sınırlı olmasını sağlar.
- Bilgisayar, bir sabit diskin hangi özelliklerin desteklendiğini belirlemek için IDENTIFY_DEVICE komutunu kullanır.

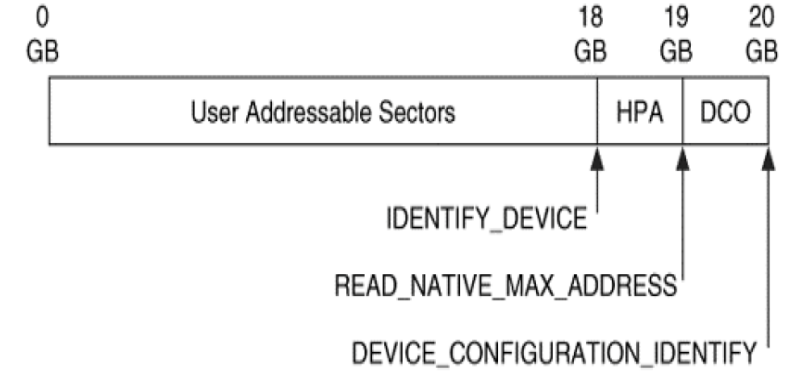
DCO Komutları

- Bir DCO oluşturmak veya değiştirmek için **DEVICE_CONFIGURATION_SET** komutu kullanılır. Bir DCO'yu kaldırmak için **DEVICE_CONFIGURATION_RESET** komutu kullanılır.
- **DEVICE_CONFIGURATION_IDENTIFY** komutu, bir diskin gerçek özelliklerini ve boyutunu döndürür. Bu nedenle, **DEVICE_CONFIGURATION_IDENTIFY** ve **IDENTIFY_DEVICE** değerleri karşılaştırarak DCO olup olmadığını tespit edebiliriz.
- Ayrıca, **READ_NATIVE_MAX_ADDRESS** komutu bir HPA'dan sonra diskin boyutunu döndürdüğünü hatırlayın. **READ_NATIVE_MAX_ADDRESS** çıktısını **DEVICE_CONFIGURATION_IDENTIFY** ile karşılaştırarak sektörleri gizleyen bir DCO olup olmadığını saptayabiliriz.

Örnek DCO

- Örneğin, 20GB lık bir diskte DCO'nun maksimum adresi 19GB olarak belirlediğini düşünelim.
- READ_NATIVE_MAX_ADDRESS ve IDENTIFY_DEVICE, diskin yalnızca 19GB olduğunu gösterir.
- Bir 1GB HPA da oluşturulmuşsa, IDENTIFY_DEVICE komutu diskin boyutunun 18GB olduğunu gösterir.

Figure 2.13. A DCO can hide sectors at the end of the disk, in addition to sectors hidden by an HPA.



Serial ATA

- ATA cihazlarıyla çalışmanın dezavantajları vardır. Kablolar büyüktür, esnek değildir ve bağlayıcıları genellikle istenmeyen yerlere yerleştirirler. Diskleri, master ve slave atlama kablolarıyla yapılandırmak da zor olabilir.
- Arayüz, seri olarak adlandırılır; çünkü, orijinal arabirim veya paralel ATA ile aynı anda 16 bit ile karşılaştırıldığında, denetleyici ile disk arasında sadece bir bit bilgi aktarılır.
- Seri ATA bağlantı uçları, paralel ATA bağlantı uçlarının yaklaşık dörtte biri kadardır ve yalnızca yedi konnektöre sahiptir.
- Her seri ATA aygıtı doğrudan denetleyiciye bağlanır ve birden fazla aygıtı zincirleme yapılmaz.

Disk Erişim Yöntemleri

Kontrol Ünitesi ile Erişim

- Verileri okumak ve yazmak için kullanılan teknik, yazılımın doğrudan sabit disk ile iletişim kuran sabit disk denetleyicisi ile iletişim kurmasıdır.
- Bu şekilde iletişim kurmak için, yazılımın denetleyiciye nasıl adres atayacağını ve komutları nasıl yayınlayacağını bilmesi gerekir.

BIOS Üzerinden Erişim

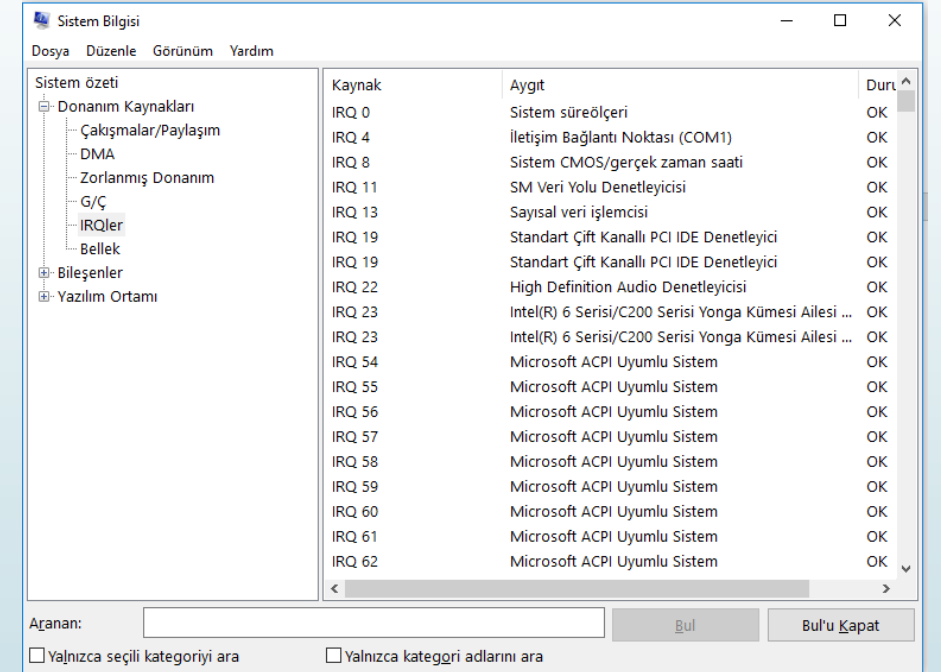
- Doğrudan sabit diske erişmek, diskten veri almak için en hızlı yoldur ancak yazılımların donanım hakkında biraz bilgi sahibi olmasını gerektirir.
- BIOS'un işlerinden biri, yazılımların bu ayrıntıları bilmesine mani olmaktır.
- BIOS, donanımı bilir ve donanımla daha kolay iletişim kurabilmesi için yazılıma hizmet sunar.
- Bilgisayar başlatıldığında BIOS'un kullanıldığı "Önyükleme Kodu Konumları" bölümünden çağrılır.
- BIOS önyükleme işlemi sırasında birçok görev gerçekleştirir, ancak adli bilişim açısından ilgilendiğimiz iki konu vardır.
- İlk şu anda yüklü olan disklerin ayrıntılarını belirlendiği kısım, ikincisi ise, işletim sistemine ve yazılıma hizmet sunmak için kullanılacak olan kesme tablosunun yüklenmesidir.

BIOS Üzerinden Erişim

- BIOS sabit disk hizmetlerini kullanmak için, yazılım sektör adresi ve boyutları gibi verileri CPU kayıtlarına yüklemeli ve 0x13 yazılım kesme komutunu (genellikle INT13h olarak adlandırılır) yürütmelidir.
- Yazılım kesme komutu, işlemcinin kesme tablosuna bakmasına ve hizmet isteğini işleyecek kodu bulmasına neden olur.
- Genelde, kesme 0x13 için tablo girişi, denetleyiciyle iletişim kurmak için sabit disk bilgisini kullanacak olan BIOS kodunun adresini içerir. Özünde BIOS, yazılım ve sabit disk arasında bir aracı olarak çalışır.
- INT13h aslında bir disk işlevleri kategorisidir ve diske yazma, diski okuma, diskteki parçaları biçimlendirme ve bilgi için diski sorgulama işlevleri içerir. Kullanılan CHS adreslerini okumak ve yazmak için orijinal INT13h işlevleri ve yazılımın sadece 8,1 GB veya daha az bir diske erişmesine izin verdi. Bu kısıtlamanın üstesinden gelmek için BIOS'da INT13h'ye "genişletilmiş INT13h" olarak adlandırılan yeni fonksiyonlar eklenmiştir.
- INT 13h kesmesi Disk işlemlerinin yapıldığı BIOS kesmesidir.

INT 13h Kesmesi

- Disk işlemlerinin yapıldığı BIOS kesmesidir.
- Aşağıda ilgili fonksiyonları verilmiştir.
 - 00–diskin resetlenmesi
 - 02 –disk sektörlerinin okunması
 - 03–disk sektörlerine veri yazılması
 - 04–disk sektörlerinin test edilmesi
 - 05–disket trackformat işlemi yapılır



SCSI Sürücüler

- Taşınabilir bir olay müdahale kiti oluştururken, hangi Bilgisayar Sistemleri Arayüzü (SCSI) kablolarının, sürücülerinin ve bağlayıcıların dahil edilmesi gerektiğini belirlemek önemlidir.
- SCSI sabit diskler, tüketici bilgisayarlarında ATA sabit diskler kadar yaygın değildir, ancak çoğu sunucuda standarttır.
- SCSI disklerin birçok özellikleri vardır.
- SCSI-1, SCSI-2, SCSI-3, Ultra SCSI, Wide Ultra2 SCSI, Ultra3 SCSI SCSI-3 gibi.

ATA SCSI Karşılaştırması

- SCSI ve ATA arasında hem üst seviye hem de alt seviye farklar vardır.
- En bariz üst seviye fark, çok sayıda bağlayıcı türü içerir.
- ATA ile yalnızca 40 ve 44 pinli konektörler vardı, ancak SCSI'in birçok şekli ve modeli vardır.
- SCSI kabloları ATA kablolarından daha uzun ve aynı kablo üzerinde ikiden fazla aygıt olabilir.
- SCSI kablosundaki her aygıtın, diskte veya yazılımdaki atlama (Jumper) kablolarıyla yapılandırılabilen benzersiz bir sayısal ID si olmalıdır. Birçok SCSI disk de diski salt okunur yapmak için ATA yazma engelleyiciye benzer bir işlev sağlayan bir jumper teli bulunur.

ATA SCSI Karşılaştırması

- ATA ve SCSI arasındaki ilk alt seviye fark, SCSI'in bir denetleyicisinin bulunmamasıdır.
- ATA arayüzü, bir veya iki sabit diske ne yapılacağını söylemek için tek bir denetleyiciyle tasarlanmıştır.
- SCSI, farklı aygıtların birbirleriyle iletişim kurduğu ve aygıtların sabit disk sayısının sınırlı olmadığı bir veri yolu kullanılarak tasarlanmıştır.
- Bir SCSI konfigürasyonunda bilgisayara takılan kart kontrolcü değildir, çünkü SCSI kablосundaki her cihaz eşittir ve birbirlerinden istekte bulunabilirler.

SCSI Türleri

- SCSI sürümlerindeki farklılıklar, bir defada kaç bit gönderildiğine, kablodaki sinyallerin frekansına (aktarım hızı) ve hangi sinyaller kullanıldığına göre değişir.
- Örneğin, Ultra SCSI aygıtı 8 bitlik bir aktarım gerçekleştirir ve Geniş Ultra SCSI aygıtı 16 bitlik bir aktarım gerçekleştirir.

Table 2.5. Speed differences among the different types of SCSI.

Type	Frequency	8-bit Transfer Rate	16-bit (wide) Transfer Rate
SCSI (normal)	5 MHz	5MB/s	10MB/s
Fast SCSI	10 MHz	10MB/s	20MB/s
Ultra SCSI	20 MHz	20MB/s	40MB/s
Ultra2 SCSI	40 MHz	40MB/s	80MB/s
Ultra3 SCSI	80 MHz	N/A	160MB/s
Ultra160 SCSI	80 MHz	N/A	160MB/s
Ultra320 SCSI	160 MHz	N/A	320MB/s

Current SCSI Standards

Technology name	Maximum cable length (meters)	Maximum speed (MBps)	Maximum number of devices
SCSI-I	6 Single Ended	5	8
Fast SCSI	3 Single Ended	10	8
Fast Wide SCSI	3 Single Ended	20	16
Ultra SCSI	1.5 Single Ended	20	8
Ultra SCSI	3 Single Ended	20	4
Wide Ultra SCSI	-	40	16
Wide Ultra SCSI	1.5 Single Ended	40	8
Wide Ultra SCSI	3 Single Ended	40	4
Ultra2 SCSI	12 LVD/25 HVD	40	8
Wide Ultra2 SCSI	12 LVD/25 HVD	80	16
Ultra3 SCSI (Ultra160 SCSI)	12 LVD	160	16
Ultra320 SCSI	12 LVD	320	16
Ultra640 SCSI	10 LVD	640	16

SCSI Veri İletim Türleri

- Üç metot kullanılır. Bunlar Single Ended (SE), Low Voltage Differential (LDV), High Voltage Differential (HVD) dir.
- Single Ended (SE), kabloda 1 gönderilirse yüksek voltaj kablo üzerine yerleştirilir ve 0 iletilirse kabloda voltaj olmaz.
- Bu yöntem yüksek hızda ve daha uzun kablolarda sorunlara neden oluyor, çünkü elektrik sinyali yüksek saat hızında stabilize olamıyor ve kablolar birbirlerine müdahale ediyor.
- Diferansiyel Voltaj her bit için iki kablo gerektirir.
- Bir 0 iletilirse, her iki kabloya da voltaj uygulanmaz.
- Bir 1 iletilirse, bir tele pozitif bir voltaj uygulanır ve ikinci tele karşıt voltaj uygulanır.
- Bir aygıt kablodan sinyalleri okurken, iki tel arasındaki farkı alır.
- İlk sürümlerde düşük voltaj diferansiyel (LVD) sinyal seçeneği kullanılırken, Ultra2 SCSI'den beri yeni diskler için yüksek voltaj farkı (HVD) sinyal seçeneği kullanılır.

Table 2.6. Signal types that can found in each type of SCSI.

Signal Type SCSI Types

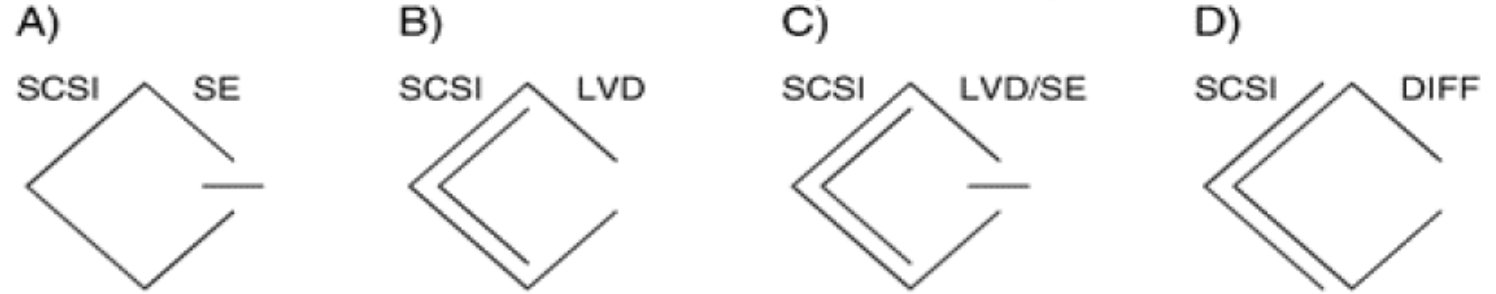
SE	SCSI, Fast SCSI, Ultra SCSI
HVD	SCSI, Fast SCSI, Ultra SCSI, Ultra2 SCSI
LVD	Ultra2 SCSI, Ultra3 SCSI, Ultra160 SCSI, Ultra 320 SCSI

Yrd.

SCSI İletim Türleri

- Farklı sinyal türlerini karıştırmamak çok önemlidir. SE cihazlarını HVD'ye ve bazı LVD cihazlarına bağladığınızda cihazlarınıza zarar verebilirsiniz.
- LVD disklerinden bazıları SE uyumludur, bu nedenle hasar görmeden bir SE ortamında kullanılabilir, ancak yalnızca SE cihazlarının kullandığı hızlarda çalışırlar. SCSI aygıtlarında hangi sinyal türünü kullandığını belirten semboller vardır.

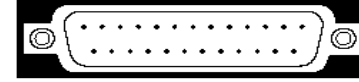
Figure 2.14. Symbols for SCSI signal types. (A) is for SE devices, (B) is for LVD devices, (C) is for devices that are both SE and LVD compatible, and (D) is for HVD devices.



Konnektör Tipleri

- Birçok SCSI bağlacı türü vardır.
- Genel olarak, 8 bitlik bir veri yoluna sahip olan sürücüler 50 iğneli bir kabloya sahiptir. 16 bitlik bir veri yoluna sahip olan sürücüler 68 iğneli bir kabloya sahiptir.
- Yüksek yoğunluklu 68-pinli adaptör günümüzde en yaygın olanlardan biridir

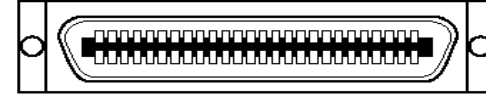
SCSI Connectors, Actual Size



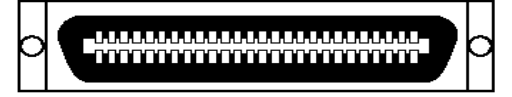
DB-25, Male External



DB-25, Female External



Low-Density, 50-pin, Male External



Low-Density, 50-pin, Female External



High-Density, 50-pin, Male External



High-Density, 50-pin, Female External



Low-Density, 50-pin, Male Internal



Low-Density, 50-pin, Female Internal



High-Density, 68-pin, Male External



High-Density, 68-pin, Female External



High-Density, 68-pin, Male Internal



High-Density, 68-pin, Female Internal

Adaptec Terminology	Alternative Terminology
Low-density 50-pin	Centronics 50-pin
High-density 50-pin	Micro DB50 or Mini DB50
High-density 68-pin	Micro DB68 or Mini DB68
Very high-density condensed 68-pin	Ultra Micro DB68

SCSI Özet

- Farklı SCSI bağlaçları için birçok bağdaştırıcı vardır ve SCSI aygıtlarının geriye dönük olarak uyumlu olması gerekir.
- Bununla birlikte, tüm adaptör ve aygıt yapılandırmaları işe yaramayabilir.
- Kurulum işe yararsa, en yavaş cihazın hızında çalışacaktır. Herhangi bir HVD cihazını, uygun adaptörler olmadan LVD veya SE cihazlarıyla karıştırmayınız.
- LVD cihazları SE modunda çalışıyorsa sadece LVD ve SE cihazlarını çalıştırın.

Boyut Engelleri

- SCSI belirtimleri her zaman 32-bit ve 64-bit LBA adreslerini kullandığından, SCSI diskleri ATA diskleri ile aynı boyut sınırlamalarına sahiptir.
- BIOS'un (INT13h kullanıyorsanız) veya dosya sisteminin boyut sınırlamaları, SCSI'nin desteklediği boyuttan daha küçük olabilir ve sınırlayıcı faktör olacaktır.
- BIOS kullanıldığında, SCSI denetleyicisi bir CHS adresini bir LBA adresine çevirir. Farklı denetleyiciler farklı haritalama teknikleri kullanabilir, ancak birçoğu bir bölüm tablosundaki girdiler tarafından tanımlanan geometriye dayalı olarak bir tane seçerler.