

HACETTEPE UNIVERSITY  
DEPARTMENT OF COMPUTER ENGINEERING



Name : Mehmet Taha

Surname : Usta

E~mail : [b21527472@cs.hacettepe.edu.tr](mailto:b21527472@cs.hacettepe.edu.tr)

Subject : What is an SSD(Solid-State Drive)

## İÇİNDEKİLER

## Sayfa

1. - Tanıtım.....	3
2. - SSD Mimarisi.....	4
2.1 - Giriş - Temel NAND Flaş Hücresi.....	6
2.2 - Giriş - SLC, MLC ve TLC NAND Flaş.....	9
2.3 - NAND Mimari - Dizeler ve Diziler.....	12
2.4 - NAND Mimari - Sayfalar ve Bloklar.....	15
2.5 - NAND Mimari - Planya ve Zar.....	19
2.6 - NAND Mimari - Bileşen Paketleme.....	22
3. - Flaş Denetleyicisi.....	24
3.1 - SSD Denetleyici Mimarisi - Kanallar ve Bankalar.....	28
3.2 - SSD Denetleyici Mimarisi - Blok Diyagramı.....	31
3.3 - SSD Denetleyici Fonksiyonları - Aşınma Seviyelendirme.....	34
3.4 - SSD Denetleyici Fonksiyonları - Çöp Toplama.....	38
3.5 - SSD Denetleyici Fonksiyonları - TRIM Komutu.....	42
3.6 - SSD Denetleyici Fonksiyonları - Aşırı Sağlama.....	44
4. - Kötü Blok Yönetimi.....	47
5. - Hata Düzeltme Kodu (ECC).....	47
6. - SSD Arayüzleri.....	48
7. - SAS and SATA.....	48
8. - PCI-Express.....	51
9. - Yüksek Hızlı Arayüz İhtiyacı.....	54
10. - Kaynakça.....	58

## Solid State Drives (SSDs)

**Teorik olarak** Katı hal sürücüler (SSD'ler), kurum depolama performanslarını bir sonraki seviyeye taşımak için uygun faktör olarak kabul edilir. Gerçekten de, Sabit Disk Sürücülerinin (HDD) rotating-storage (dönen depolama) teknolojisi, yanıt süresinin kritik faktör olduğu uygulamaların gerektirdiği erişim zamanını başaramaz. Aksine, SSD'ler NAND Flash bellekleri gibi katı hal belleklerine dayanmaktadır; bu durumda, herhangi bir mekanik parça ve depolanmış verilere rastgele erişim çok daha hızlı olabilir ve bu şekilde yukarıda belirtilen gereksinimleri karşılar. Ancak birçok uygulamada, ana işlemci ve sürücüler arasındaki arabirim, performans darboğazı olarak kalmaktadır. Bu nedenle SSD'nin arayüzü, SSD'nin ana işlemci ile doğrudan bağlantısını sağlayan SAS ve SATA gibi eski depolama arabirimlerinden PCIe'ya dönüşmüştür.

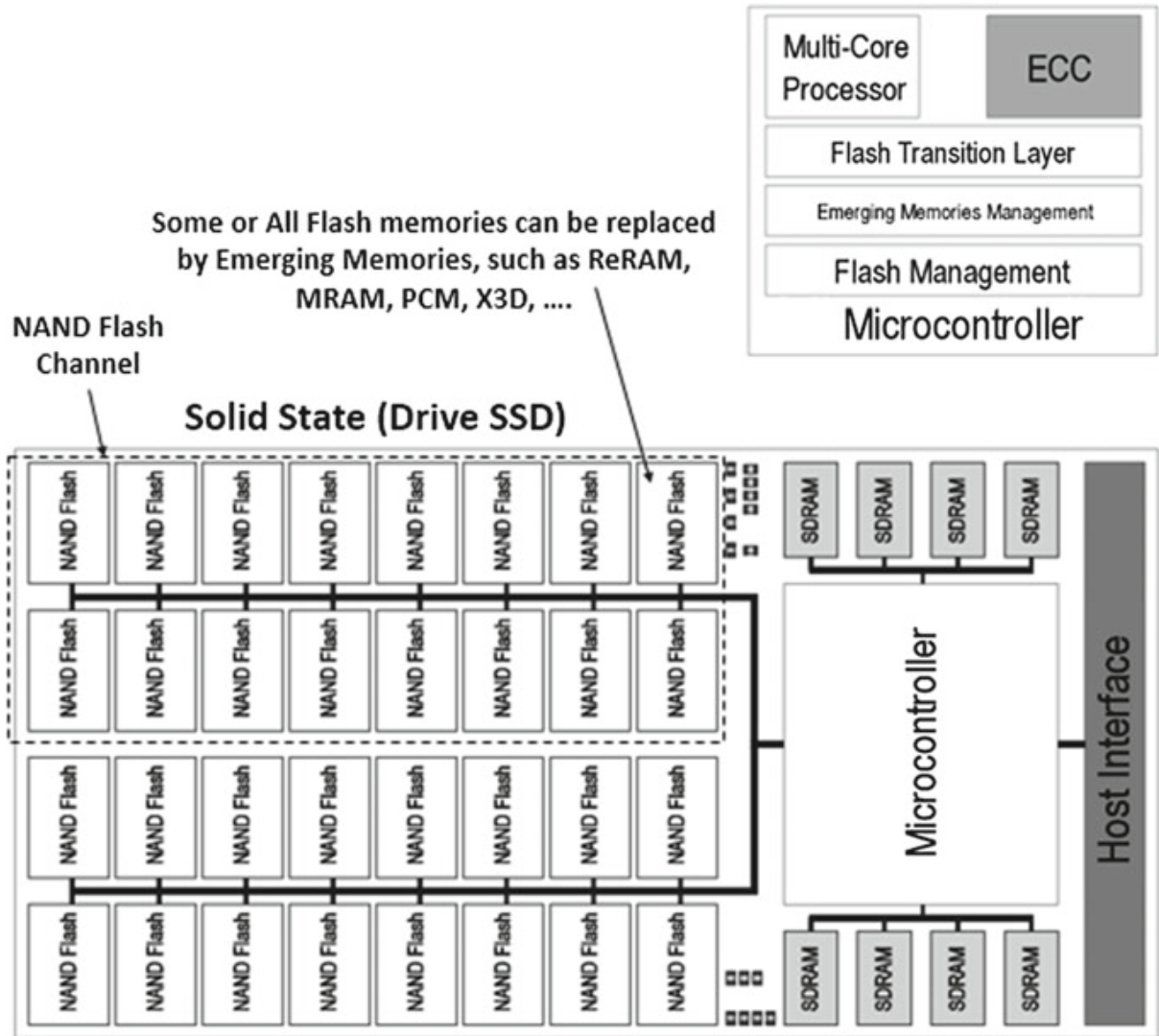
### 1. Tanıtım

Katı Hal Sürücüler (SSD'ler), kurumsal depolama performansını büyük ölçüde geliştirmeyi vaat eder. Elektromekanik Sabit Disk Sürücüler (HDD'ler) kapasiteyi sürekli olarak arttırırken, döner depolama teknolojisi, çevrimiçi işlem, veri madenciliği ve bulut bilişim dahil olmak üzere zorlu kurumsal uygulamalarda gerekli olan erişim zamanını veya aktarım performansını sağlamaz. İstemci uygulamalarının, daha hızlı yanıt süreleri sunabilen, daha az güç harcayan ve daha küçük mobil form faktörlerine sığabilen elektromekanik disk sürücülerine bir alternatif ihtiyacı vardır.

Flash bellek tabanlı SSD'ler, verilere daha hızlı rastgele erişim ve daha hızlı aktarım oranları sunabilir. Dahası, SSD'nin veri depolama kapasitesi artık HDD'lerin yerini alabilecek noktada. Ancak birçok uygulamada, ana bilgisayar ve sürücüler arasındaki arabirim, performans darboğazı olarak kalmaktadır. SAS ve SATA gibi eski depolama arabirimlerine sahip SSD'ler yararlı olduğunu kanıtıyor. PCI-Express (PCIe) SSD'ler, ana işlemci ile doğrudan bağlantılı olarak performansı daha da artıracak ve yanıt vermeyi artıracaktır.

## 2. SSD Mimarisi

Bir SSD'nin temel blok diyagramı Şekil 1.1'de gösterilmiştir. Belleklere ve bir Flash denetleyicisine ek olarak, genellikle başka bileşenler de vardır. Örneğin, dahili güç kaynağını sürmek için harici bir DC-DC dönüştürücü eklenebilir veya daha iyi bir saat hassasiyeti için bir kuvars kullanılabilir. Tabi ki, güç kaynağının stabilize edilmesi için makul filtre kapasitörler yerleştirilir. Güç yönetimi sebepleri için bir dizi sıcaklık sensörüne sahip olmak da çok yaygındır.

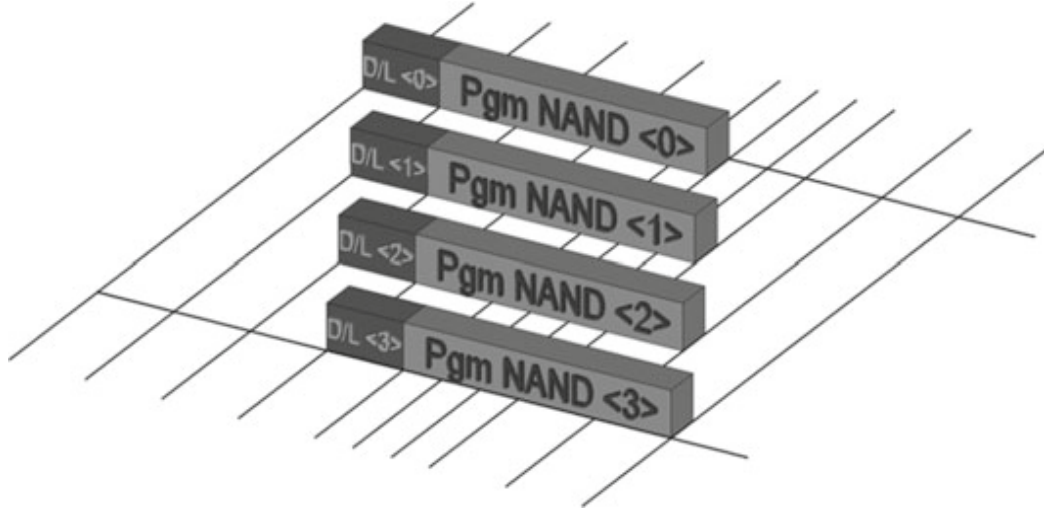


**Fig. 1.1** SSD'nin blok diyagramı

Veri önbellekleme için hızlı bir DDR bellek sıklıkla kullanılır: yazma erişimi sırasında, önbellek, Flaşa aktarılmadan önce verileri depolamak için kullanılır.

Faydası veri güncelleme olup, örn. yönlendirme tabloları daha hızlıdır ve Flaşı yıpratmaz. Tipik bir hafıza sistemi, birkaç NAND belleğinden oluşur. Tipik olarak, genellikle “kanal” olarak adlandırılan 8 bitlik bir veri yolu, farklı bellekleri denetleyiciye bağlamak için kullanılır (Şekil 1.1). Bir sistemdeki birden fazla Flash belleğinin hem depolama yoğunluğunu artırmaya hem de okuma / yazma performanslarına yönelik bir araç olduğunun altını çizmek önemlidir.

Bir kanaldaki işlemler aralıklı olabilir, yani birincisi hala meşgulken ikinci bir çipin ele alınabileceği anlamına gelir. Örneğin, çok sayıda yazma işleminin bir dizisi, Şekil 1.2'de gösterildiği gibi farklı NAND'lara hitap eden bir kanala yönlendirilebilir: Bu şekilde, veri yükleme aşaması boru hattı ile kanal kullanımı maksimuma çıkarılır.



**Fig. 1.2** Interleaved operations on one Flash channel

Aslında, program işlemi bir bellek yongası içinde gerçekleşirken, ilgili Flash kanalı serbesttir. Toplam Flash kanalı sayısı, hedef uygulamanın bir fonksiyonudur, ancak onlarca kanal oldukça yaygın hale gelmektedir. Aynı Flaş programlama süresi göz önüne alındığında, dönüşümlü-çalıştırma(interleaving) sayesinde SSD'nin verimi büyük ölçüde artıyor.

Bellek denetleyicileri, bellek kanallarına erişimi zamanlamaktan sorumludur. Denetleyici, Flash ile düşük seviyeli iletişim protokolü için özel motorlar kullanır.

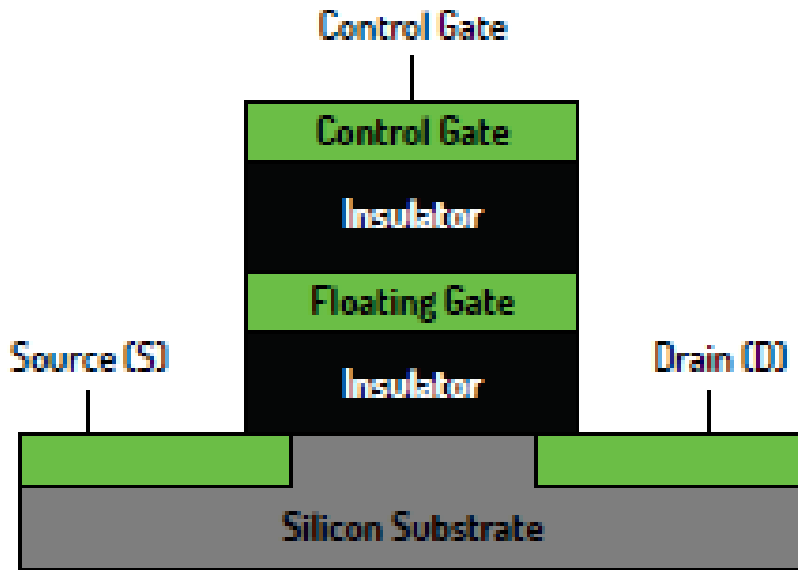
Ayrıca veri yükleme aşamasının, program işlemine kıyasla ihmal edilemediği açıktır (aynı yorum veri çıkışı için geçerlidir): bu nedenle, I / O

arabirimi hızını artırmak, performansları iyileştirmenin başka bir akıllı yolu: DDR benzeri arayüzler. Hız arttıkça, kanalı doyumadan önce daha fazla NAND paralel olarak çalıştırılabilir. Örneğin, 30 MB / sn'lik bir hedef varsayıldığında, yaklaşık 50 MHz'lik bir minimum DDR frekansı ile 2 NAND'a ihtiyaç vardır. 200 µs'lik bir sayfa program süresi verildiğinde, 50 MHz'de dört NAND aralıklı modda çalışabilir ve yazma işinin iki katına çıkar. Tabii ki, güç tüketimi özenle dikkate alınması gereken bir başka ölçüttür.

## 2.1 - Giriş - Temel NAND Flaş Hücresi

Tek bir veriyi bir katı hal sürücüde saklamak için, en küçük yapı bloğuna ihtiyacınız vardır - tek bir NAND flaş hücresi. En basit NAND hücresi, 0 veya 1 durumuna ayarlanabilir. Güç kaldırıldıktan sonra bile bu durumu saklamaya devam edecektir.

**NAND Hücresi neye benziyor?**

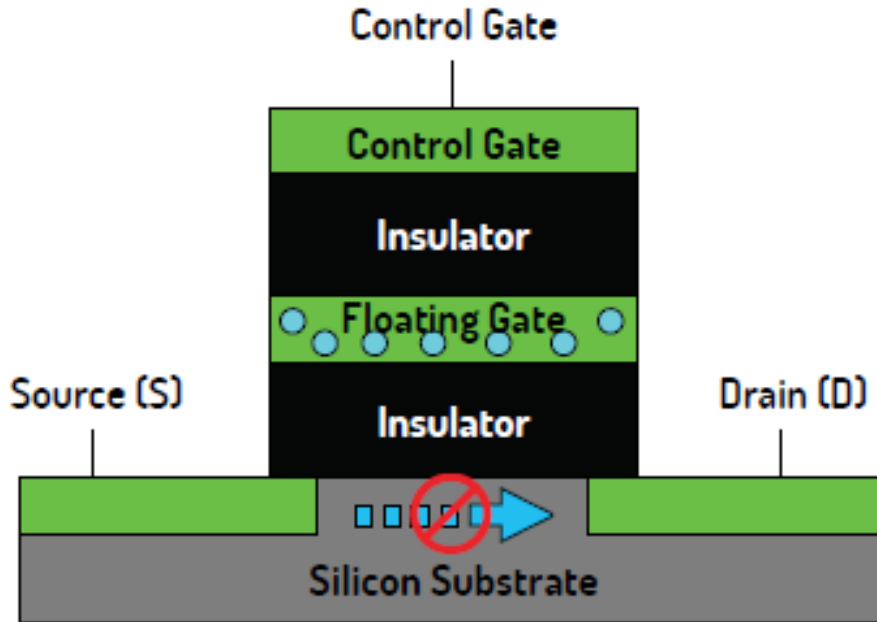


Basit bir NAND Flaş Cell diyagramı yukarıda gösterilmiştir. NAND flaş hücresi **floating gate** transistöründen yapılmıştır. Elektrik yükü, yukarı ve aşağısında oksit yalıtım tabakaları ile izole edilen **floating gate** üzerinde depolanır. En basit haliyle **floating gate** yüklendirildiğinde, programlandırılmıştır ve ikilik tabanda 0 olarak tanımlanır. **floating gate** yüklendirilmediğinde zaman, silinir ve ikilik tabandaki 1 değeri kabul edilir.

FLOATING GATE STATE	REFERRED TO AS	BINARY ASSIGNED VALUE
Charged	Programmed	Zero - 0
No Charge	Erased	One - 1

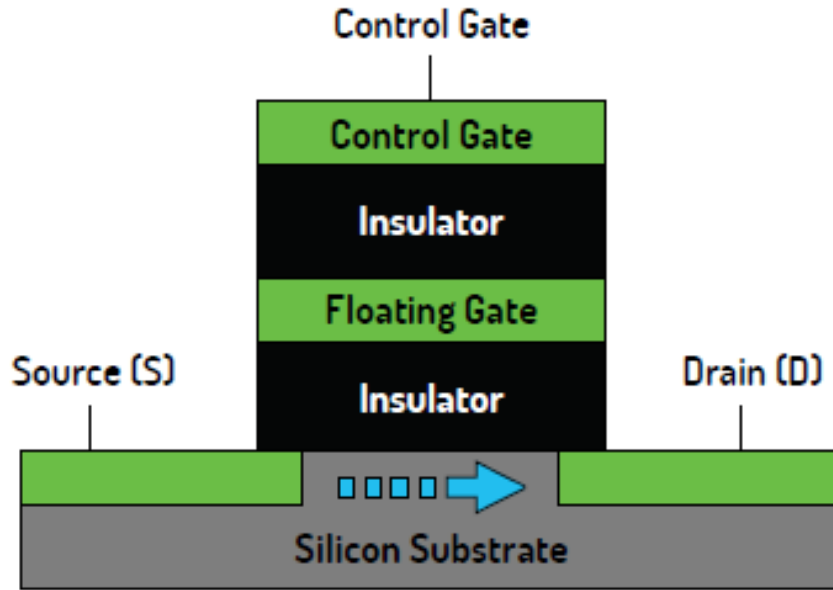
**floating gate**, onu çevreleyen devre tarafından değiştirilene kadar yüklü veya yüksüz durumda kalır. NAND aygıtından güç çıkarmak, **floating gate** durumunu etkilemez, bu nedenle veri depolama için değerli bir aygıttır.

### NAND Hücre Nasıl Okunur



Bir hücreyi okumak için kontrol kapısına voltaj uygulanır ve kaynaktan drenaja akım akışı denenir.

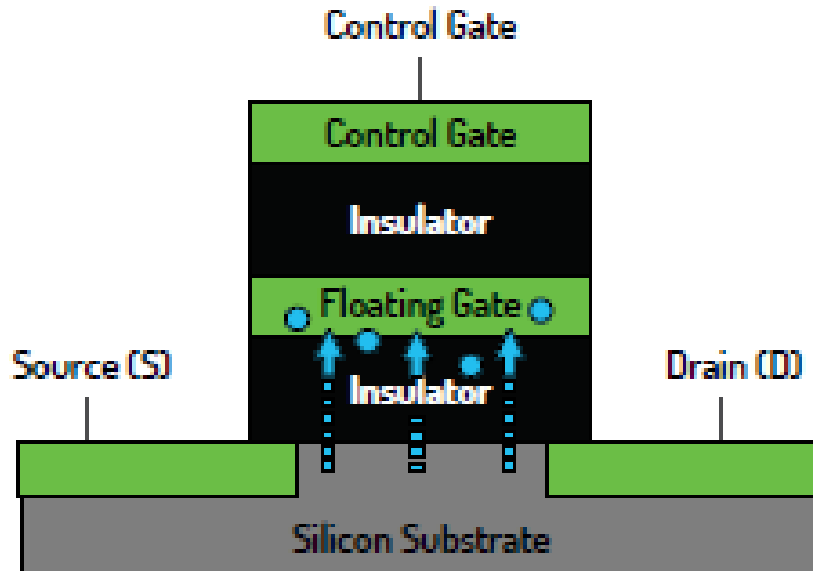
Akım akışı yoksa, yukarıdaki diyagramda olduğu gibi **floating gate**'in yüklü olduğunu gösterir (ikilik tabanda 0). Akım akışı mevcut ise, **floating gate** yüklendirilmez (ikilik tabanda 1) - aşağıdaki şemada olduğu gibi.



### NAND Hücre Nasıl Yazılır

Bir hücre yazmak için, kontrol kapısına yüksek bir voltaj uygulanır ve elektronlar silikon alt tabakadan **floating gate**'e doğru hareket eder.

Bu süreç, tünel açma olarak adlandırılır, çünkü elektronlar oksit izolatöründen geçen "tünel" ile **floating gate**'e ulaşırlar. Aşağıdaki şemaya bakınız.

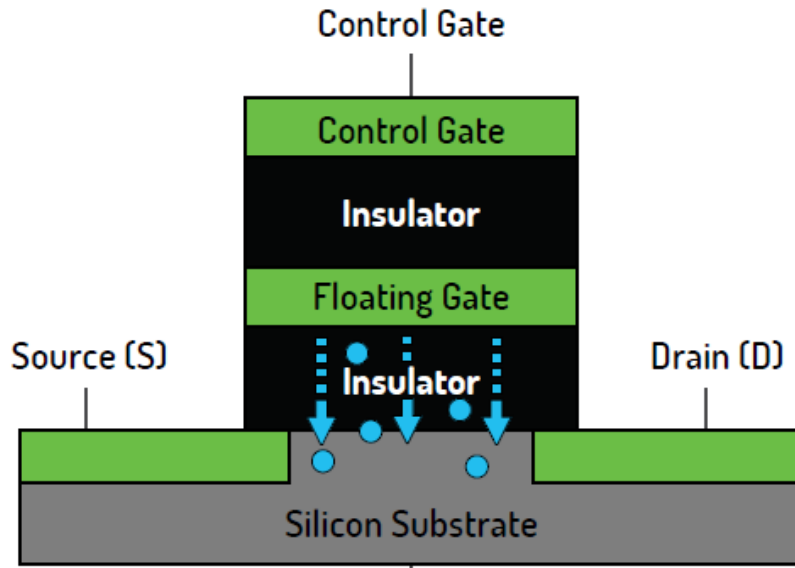


### NAND Hücre Nasıl Silinir

Bir NAND hücrelerini silmek için, silikon alt-tabakaya yüksek bir voltaj uygulanır ve elektronlar **Floating Gate**'den silikon alt-tabakaya hareket



eder. Bu, yazma işlemi ile aynı tünelleme işlemi kullanır. Aşağıdaki şemaya bakınız.

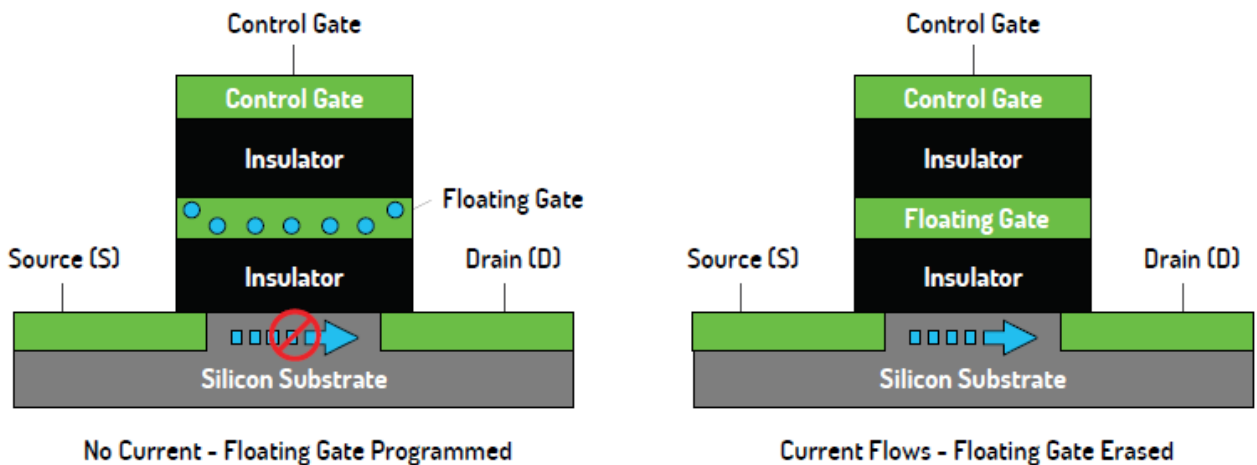


### NAND Hücre Ömrü

Yazma ve Silme işlevlerinde açıklanan tünelleme işlemi, oksit yalıtkan katmanında strese neden olur. Zamanla bu stres oksit tabakasını kırar ve Floating Gate bir yükü tutamaz hale gelir. Bir noktada hücre artık kullanılamaz durumda ve emekli olmalı. Bu durum, her NAND flaş hücresinin sonlu sayıda yazma / silme işleminden sorumlu olan şeydir.

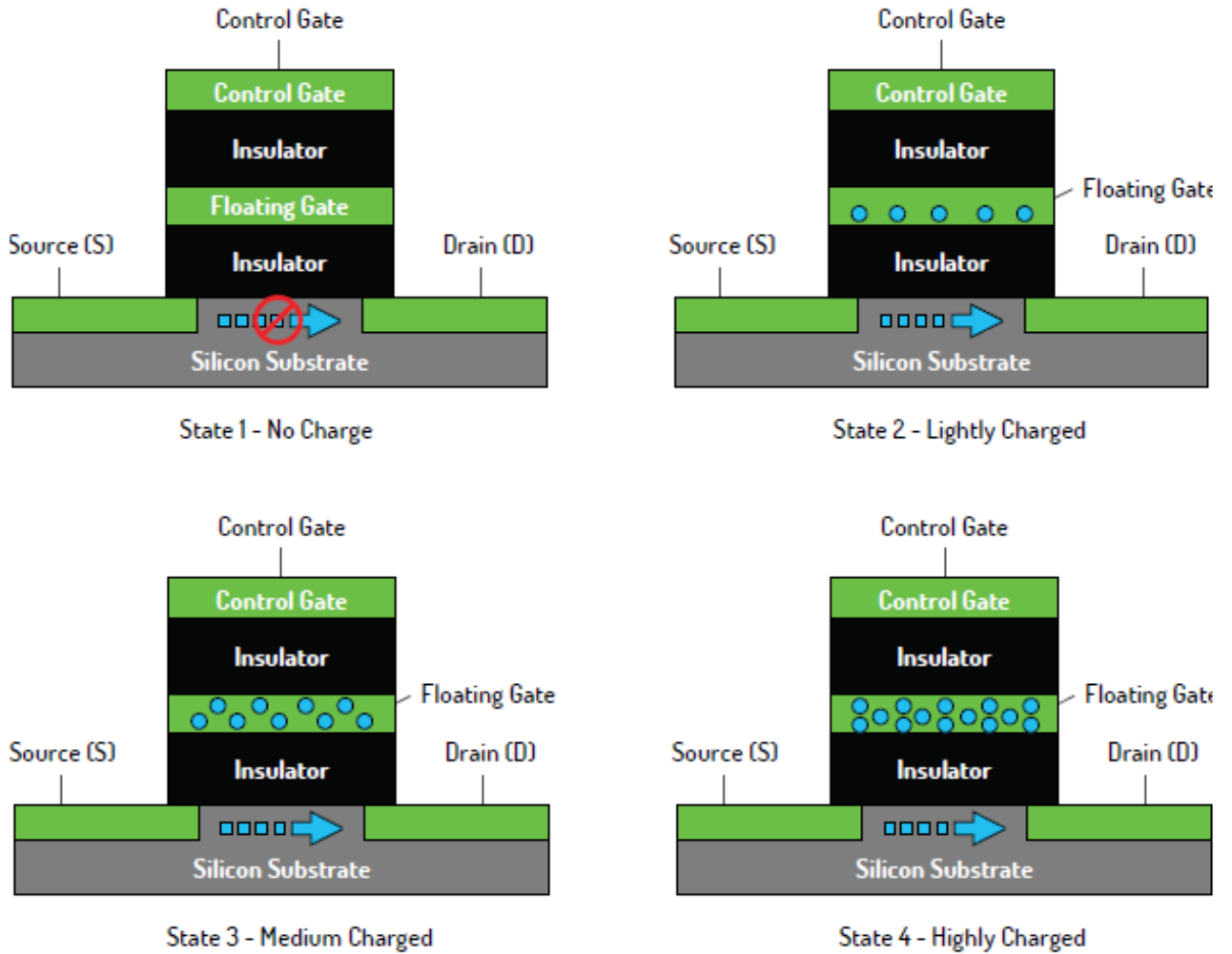
## 2.2 - Giriş - SLC, MLC ve TLC NAND Flaş

Temel NAND Hücresinin basit bir incelemesi için, yükler, bir yalıtkan gibi hareket eden iki oksit tabakası arasına sıkıştırılmış **floating gatede** saklanır veya depolanmaz.



Orijinal ve en basit tipte NAND flaşında, Kaynak ve Drenaj arasında herhangi bir akım yoksa, **floating gate**'in yüklü olduğunu (mavi noktaların elektronları temsil ettiğini) gösterir ve bu nedenle bir ikilik tabanda 0'ı temsil eder ve programlanır. Bkz. yukarıdaki şema.

Eğer akım akışı algılanırsa, **floating gate**'in bir yükü olmadığını ve silindiğini gösterir, bir ikilik tabanda 1'i temsil eder. Aşağıdaki şemaya bakınız.



## Hücre başına birden fazla bit MLC ve TLC nasıl depolanır

Önceki örnek bir SLC (Tek Düzey Hücre) NAND Hücresini göstermektedir. Kaynak ve drenaj arasında herhangi bir akım tespit edildiğinde, hücre programlanmış olduğu sonucuna varılabilir. Bir bit'i temsil etmek için programlanmış veya silinmiş yalnızca iki duruma ihtiyaç duyulduğundan, bütün bunlar gerekli.

MLC (Çoklu Seviye Hücre) NAND ile, 4 farklı durum gerektiren iki bit veriyi saklamak gerekir. Bunu başarmak için, MLC NAND hücresi, dört farklı

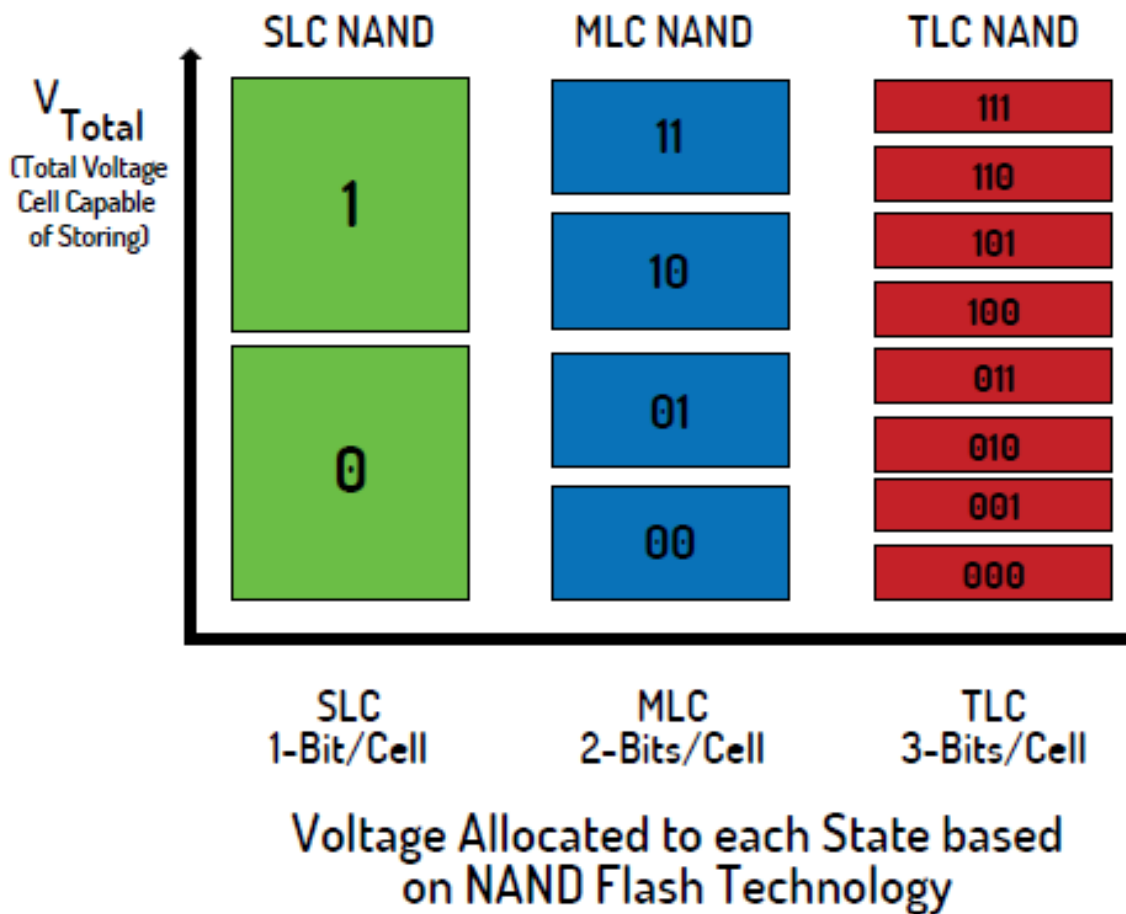
seviyede **floating gate**'e yük uygulayabilmeli ve daha sonra dört seviyeden hangisinin ayarlandığını tespit edebilmelidir.

Yukarıdaki şema, **floating gate**'de bulunan mavi renkteki ek elektronları temsil eder, bunlar daha sonra doğru şekilde okunabilmeleri için kesin seviyelere ayarlanması gerekir. Bu, MLC'yi SLC NAND eşlerinden daha zor ve daha yavaş yazmaktadır.

TLC (Tri Seviye Hücre ve Üç Seviye Hücre olarak da bilinir) NAND daha da karmaşık bir misyona sahiptir. Hücre başına 3 bitlik, 8 farklı durum gerektiren, depolayabilmeli ve tanımlayabilmelidir.

### SLC, MLC ve TLC NAND Hücrelerinde Gerilim Seviyesi

Her bir hücrede maksimum voltaj yaklaşık aynıdır. Böylece SLC hücrelerinin kendi durumları arasında çok sayıda koruma bandı vardır. Bu nedenle, SLC NAND, aşırı sıcaklıklara ve diğer olumsuz etkilere MLC veya TLC NAND'dan çok daha iyi dayanabilir.



Yukarıdaki görüntü, bellek teknolojilerinin her birinde çoklu durumları depolamak için gereken voltaj eşik düzeylerini göstermektedir.

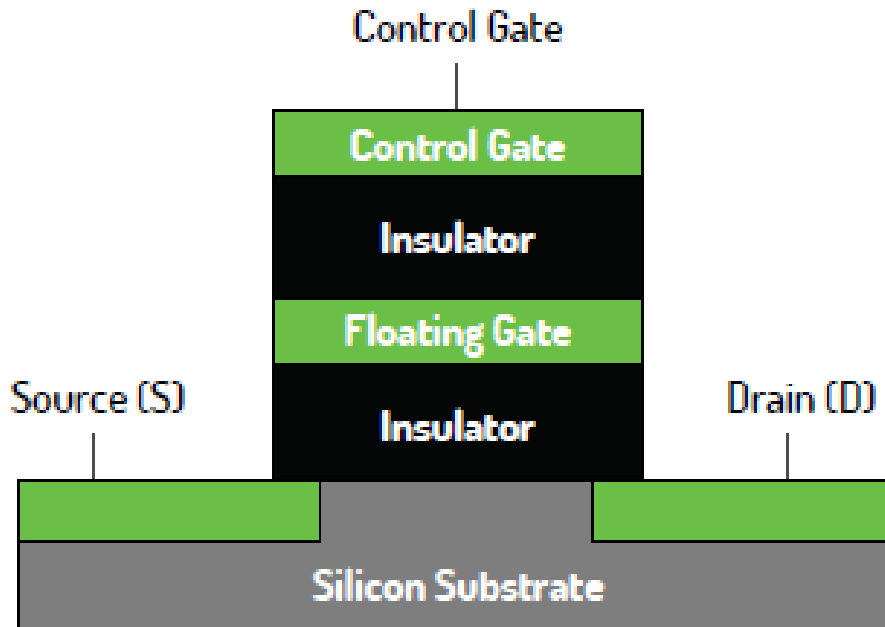
Koruma bandı alanı sayılmaz, her bir SLC durumu, gerilim aralığının % 50'sine ayrılır; MLC% 25 ve TLC% 12.5.

Gördüğünüz gibi, MLC ve TLC daha sıkı toleranslara sahiptir ve SLC NAND'dan dış çevre ve devre etkilerine daha duyarlı olacaktır. Temel avantajları maliyettir.

## 2.3 - NAND Mimari - Dizeler ve Diziler

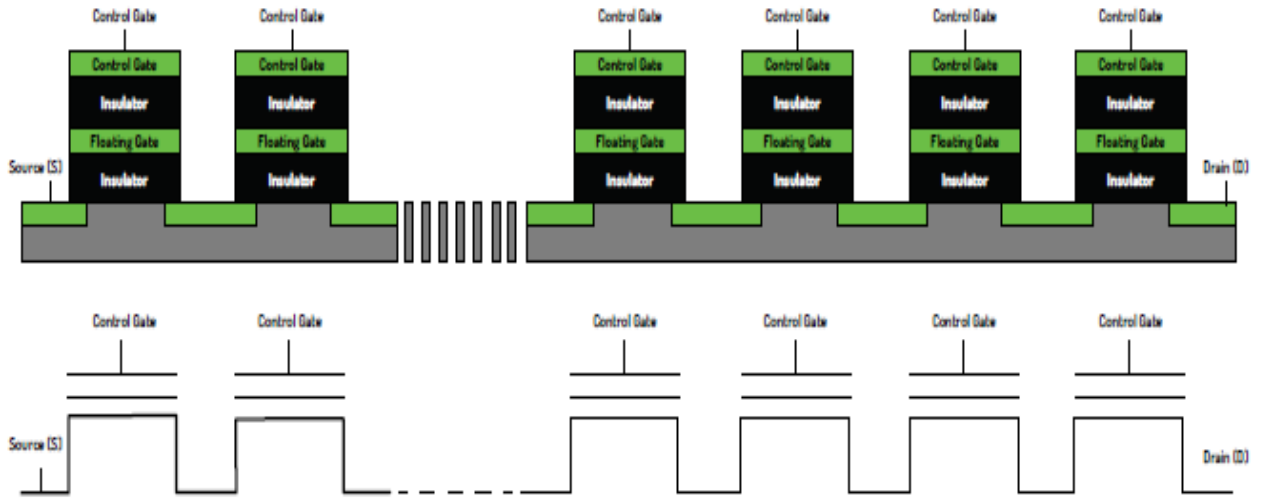
Hızlı bir inceleme için, tek bir NAND flaş hücresi, yukarıda ve aşağıda oksit yalıtım katmanları tarafından izole edilen bir **floating gate**'de bir elektrik yükünü depolar. **floating gate** üzerinde bir yük olduğunda en basit haliyle, programlanmış ve bir ikilik tabanda 0 olarak tanımlanmıştır. **Floating Gate'in** yükü olmadığı zaman, silinir ve 1'in ikilik tabanda değeri olarak kabul edilir.

### Tek bir NAND Flaş Hücresinin Şeması



## Bireysel Nand Flaş Hücreleri Bir Dize İçinde Birleştiriliyor

Tek başına, tek bir flaş hücresi çok değerli olmazdı. Ancak bunların çoğunu birleştirmek, önemli miktarda verinin depolanmasını sağlar. Bireysel NAND hücrelerini birleştirmenin ilk adımı, NAND Dizisidir.

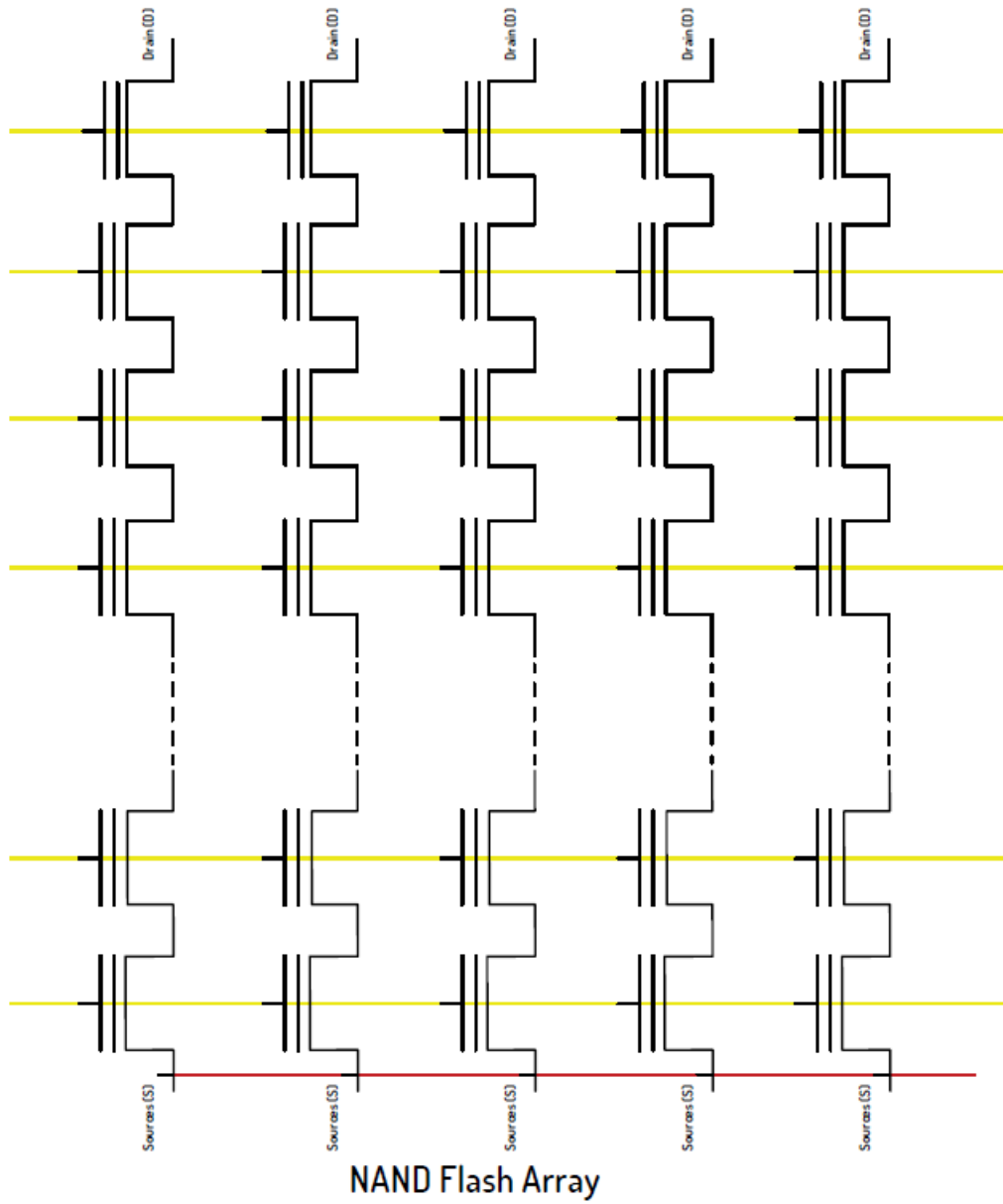


NAND String (Shown in Diagram and Schematic Versions)

Yukarıdaki görüntü, hem bir diyagram formunda hem de şematik formda tasvir edilen NAND Dizisini göstermektedir. Şematik form genellikle daha büyük dizileri göstermek için kullanılır.

NAND hücreleri bir dizi hücre oluşturmak için uçtan uca bağlanır. Tipik olarak 32 veya 64 hücre birbirine seri halde bağlanır, her bir hücre bir bit veriyi temsil eder (0 veya 1).

## Combining NAND Strings into Arrays



Bir NAND Dizisi 32 bit veri saklayabilirken, bu yalnızca 4 baytlık veriye veya 4 karakter için yeterlidir. Dolayısıyla, diziler daha kullanışlı miktarlarda depolama sağlamak için daha büyük diziler halinde birleştirilir.

Yukardaki görüntü, bir dizide birkaç kez tekrarlanan NAND Dizisi şemasını gösterir. Diziyi birbirine bağlamaya hizmet eden NAND dizelerine yapılan ek bağlantılara dikkat edin. Kırmızı çizgi, bireysel dizelerin Kaynaklarını (Sources) bağlar.

Sarı çizgiler NAND dizelerinin Kontrol Kapılarını bağlar. Dizide, kontrol kapıları yatay olarak değil dikey olarak bağlanır. Ek olarak, Drenaj (Drain) satırları, dizide ayrı olarak kullanıldıklarından beri bağlantı göstermiyor.

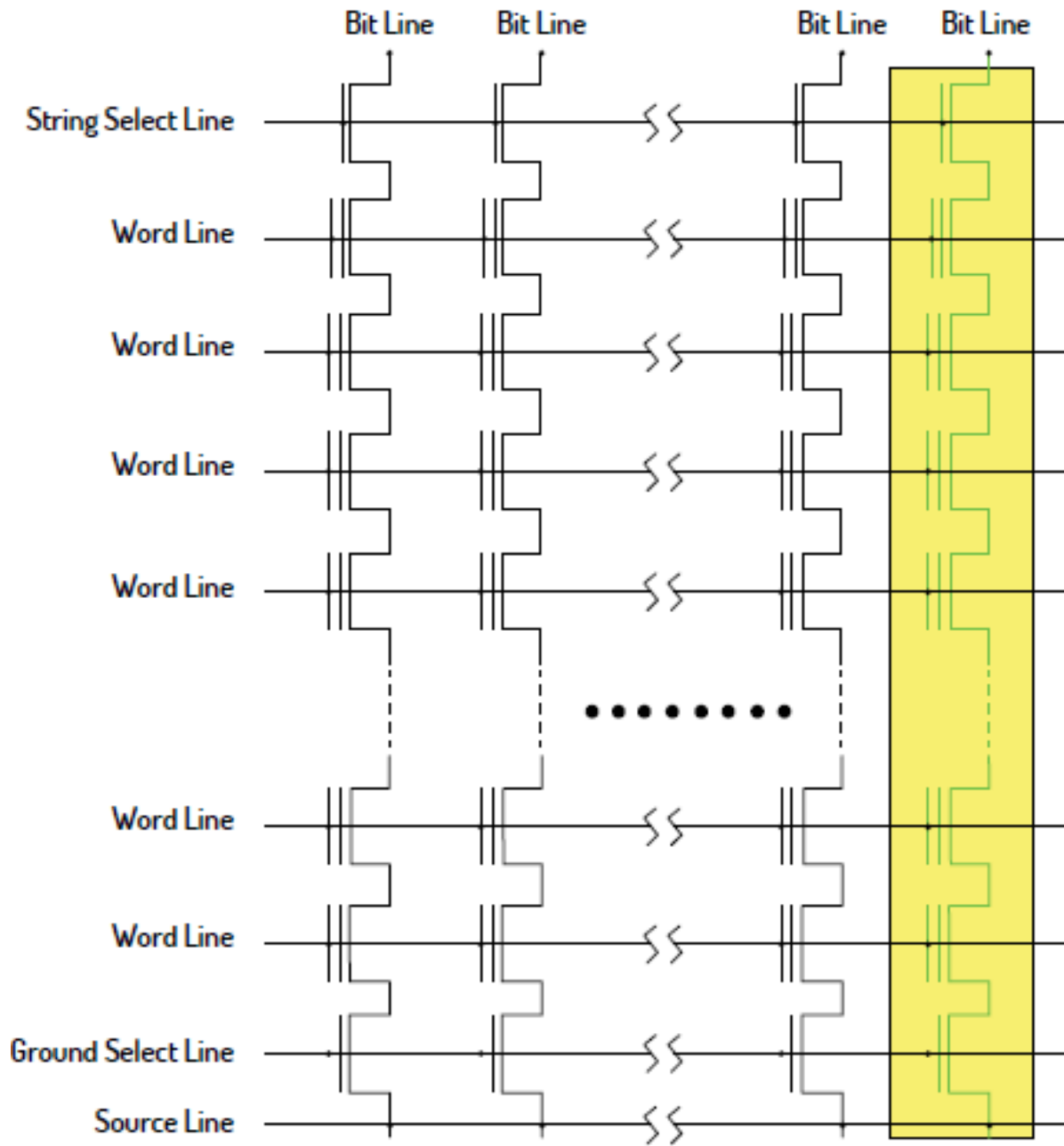
## **2.4 - NAND Mimari - Sayfalar ve Bloklar**

Hızlı bir inceleme için, tek bir NAND flaş hücresi tek bir 0 veya 1 biti tek tek saklamak için çok daha büyük veri depolama yapıları oluşturmak üzere dizeler ve dizilerde birleştirilir. Bu dizeler, seçili hücrelerden verilerin depolanmasını ve alınmasını sağlamak için bağlanır. Diziye daha fazla NAND hücresi ekleyerek çok büyük veri depolama cihazları yapılabilir.

### **NAND Flaş Dizesi**

Bir sonraki Görüntü, kontrol ve veri hatları ile daha büyük bir dizi gösterir. Sarı bir arka plan ile yeşil renkte vurgulanan, önceki makalede ele alınan NAND Dizesidir. Nokta olarak tasvir edilen arasında birçok dizeleri vardır.

Dizeler (sütunlar olarak gösterilir) okunacak minimum birimdir ve tipik olarak 32 veya 64 NAND hücresinden oluşur. Dizideki tüm dizeler bir uçtan ortak bir Kaynak Çizgisine (SL) ve diğer ucunda Bit Çizgisine (BL) bağlanır.

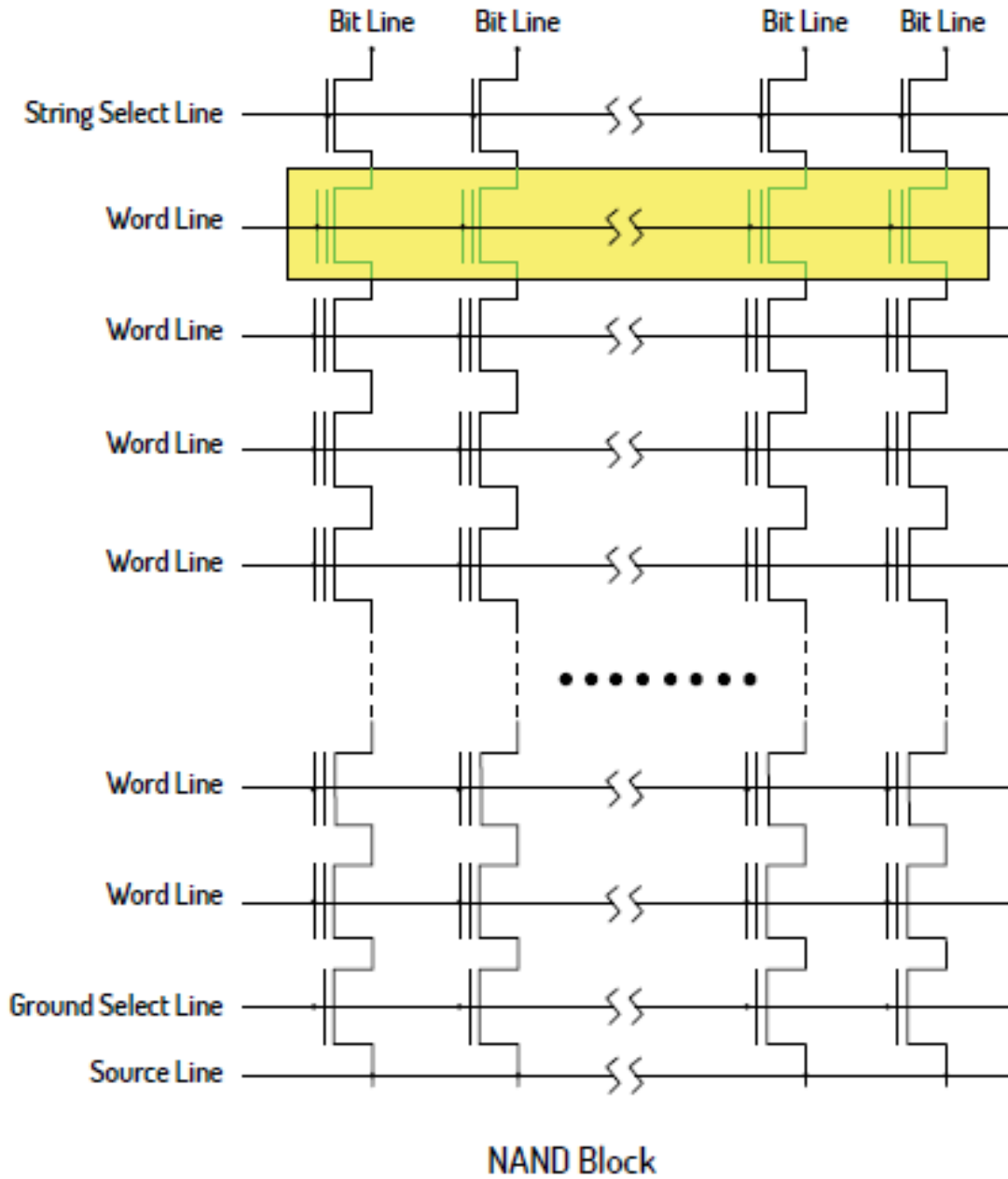


### NAND Block

Her dizgede NAND hücreleri ile seri olarak iki kontrol mekanizması bulunur. String ve ground select transistörler String Select Line (SSL) ve Ground Select Line (GSL) ile bağlantılıdır.



## The NAND Flash Page

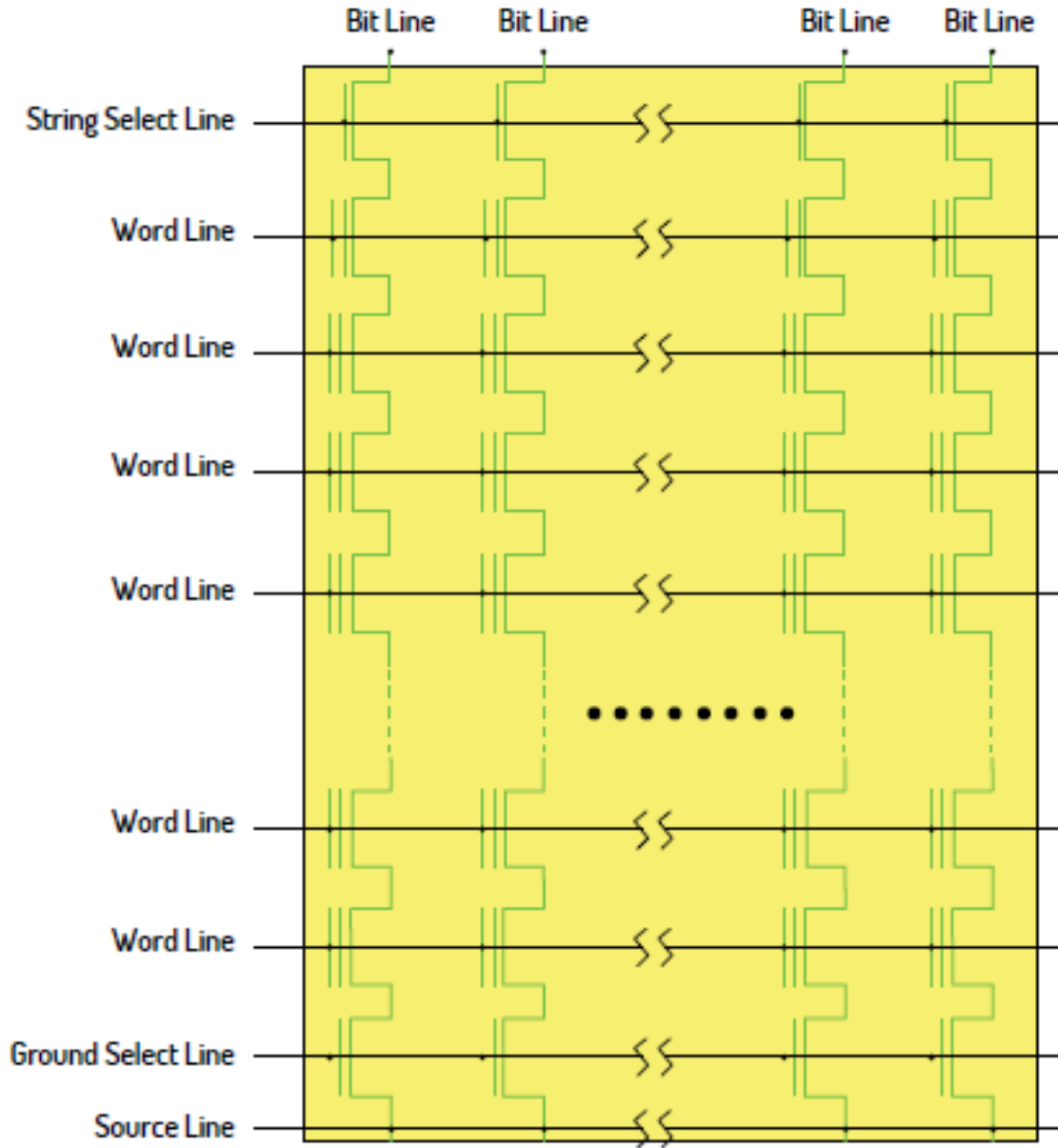


Yukarıdaki görüntü, yeşil çizgiler ve sarı vurgu ile NAND sayfasını göstermektedir.

Sayfalar (satırlar halinde gösterilir) aynı sözcük satırını paylaşır ve programlanacak en düşük birimdir. Genellikle en az 32.768 NAND hücreden oluşurlar, yeni NAND cihazlarının birçoğu sayfa boyutları 64K veya 128K hücrelere sahiptir.

Çoğu sayfa boyutu 2K, 4K, 8K vb. Olarak adlandırılır. Bu, sayfa boyutunu bayt cinsinden belirtir. Dolayısıyla, sayfa boyutu 32,768 NAND Hücresi (bit) içeriyorsa, bu 4096 bayt veya 4K Sayfa boyutuna eşittir.

## The NAND Flash Block



NAND Block

Yukarıdaki bu görüntü yeşil çizgiler ve sarı vurgu ile NAND Bloğunu göstermektedir.

Bir blok, sayfalardan (satırlardan) ve dizelerden (sütunlar) oluşan 2 boyutlu bir matristir. Bir bloktaki toplam bit sayısı, dizi sayısını sayfa sayısına göre çarparak hesaplanabilir.

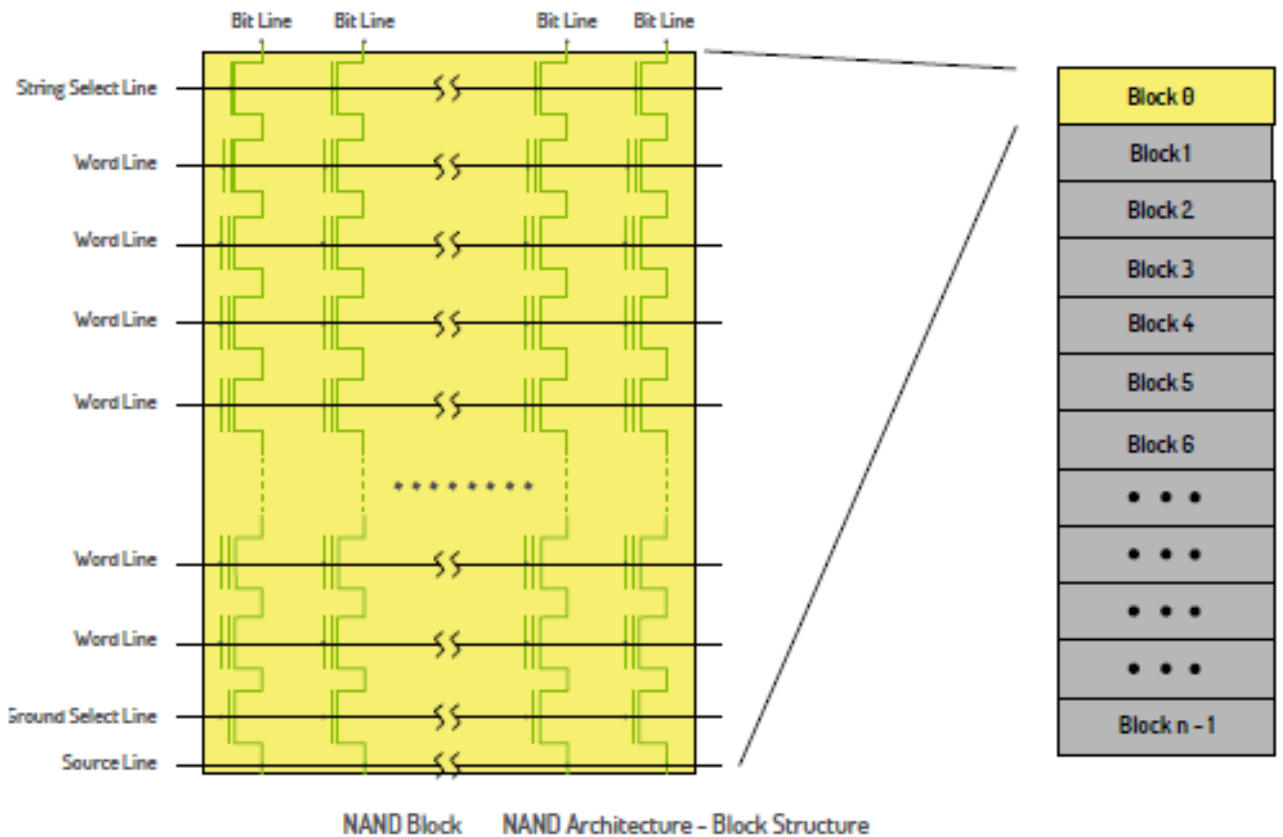
2014 Flaş Bellek Zirvesindeki bir Micron sunumundan, Blok başına Maksimum Sayfa 512'ye yaklaşıyor ve Blok boyutları 8 Mbyte'ye kadar ulaşıyor.

## 2.5 - NAND Mimari - Planı ve Zar

Daha önce tartışıldığı gibi, bireysel NAND hücreleri dizinin sütun ve satırları olarak yapılandırılmış Dizeler ve Sayfalar üzerinde birleştirilir. Genel diziye bir Blok denir. En son NAND bileşenlerinin bazıları, 8Mbyte kadar yüksek blok boyutlarına sahiptir.

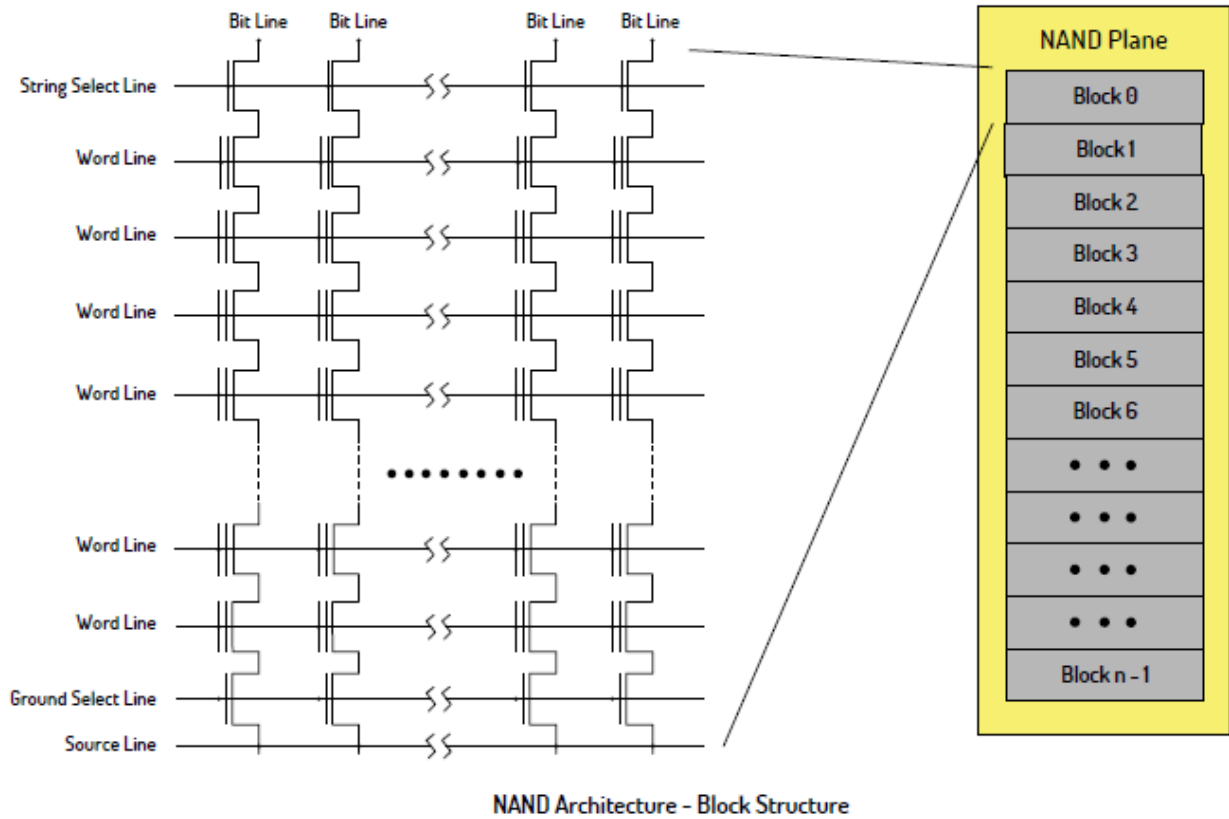
Bu bölüm, bir NAND oluşturan parçaları göstermek için Blok seviyesinden inşa edecektir.

### The NAND Block Structure



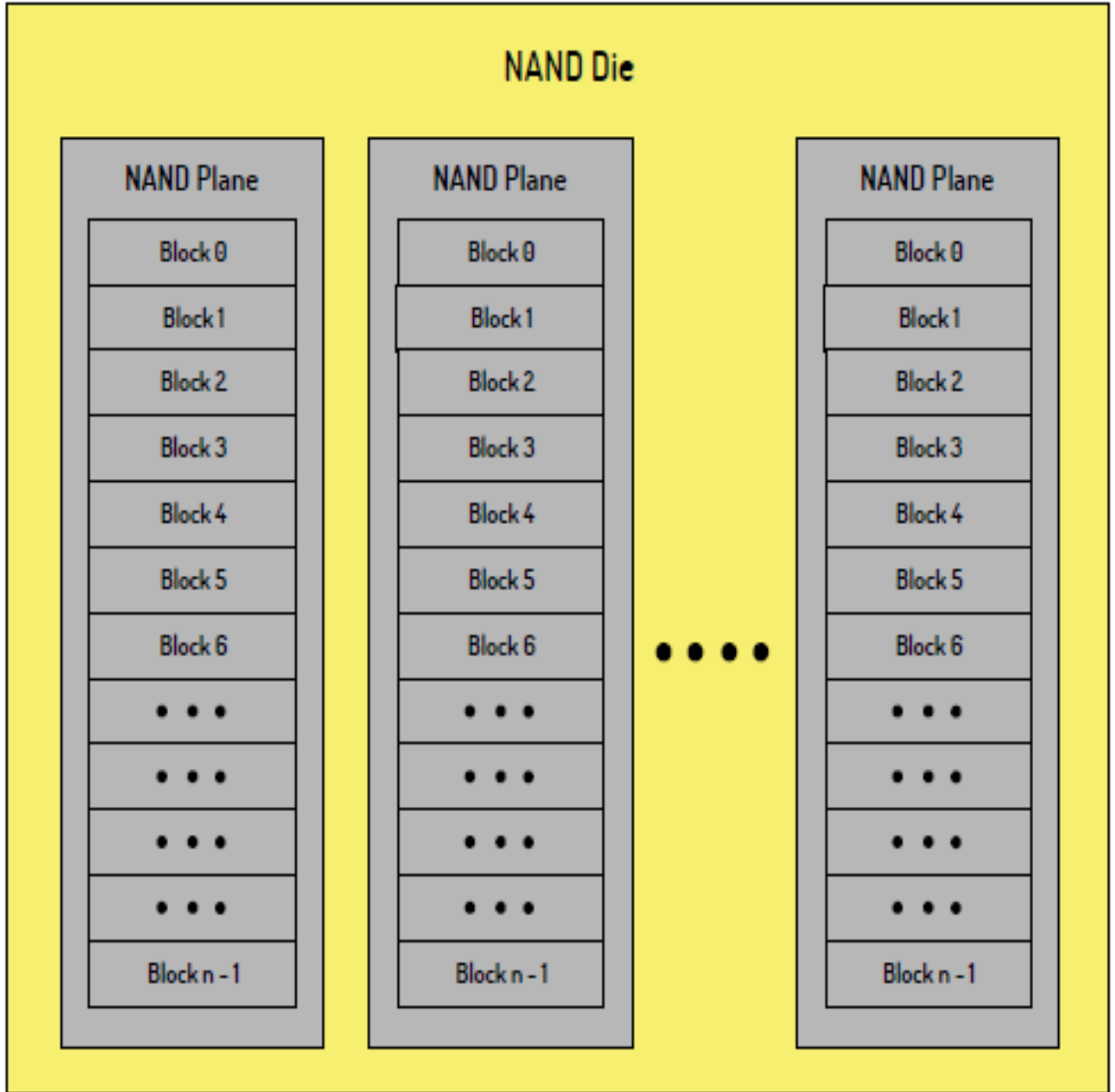
Dizeler ve Sayfalar matrisinden oluşan NAND Bloğu, daha büyük veri yapıları için bir yapı taşıdır. Tek bir blok, yukarıdaki resimde gösterildiği gibi, diğer birçok bloğun bir bankasında gruplanır.

## The NAND Plane and Die



Yukarıdaki sarıyla vurgulanan bu Bloklar bankası bir planya olarak adlandırılmaktadır.

Bir veya daha fazla planya, bir sonraki sayfada yer alan resimde vurgulanan bir NAND Zarını oluşturmak için birlikte gruplandırılmıştır. OEM'lerin birçok farklı tasarım ihtiyacını karşılamak için çok sayıda konfigürasyon vardır.



**NAND Die with Multiple Planes**

Birbirinin üstüne yığılmış tek bir Zar veya çoklu Zarlar, popüler JEDEC TSOP, BGA ve diğer ambalajlarda kullanılabilir bir formda paketlenir.

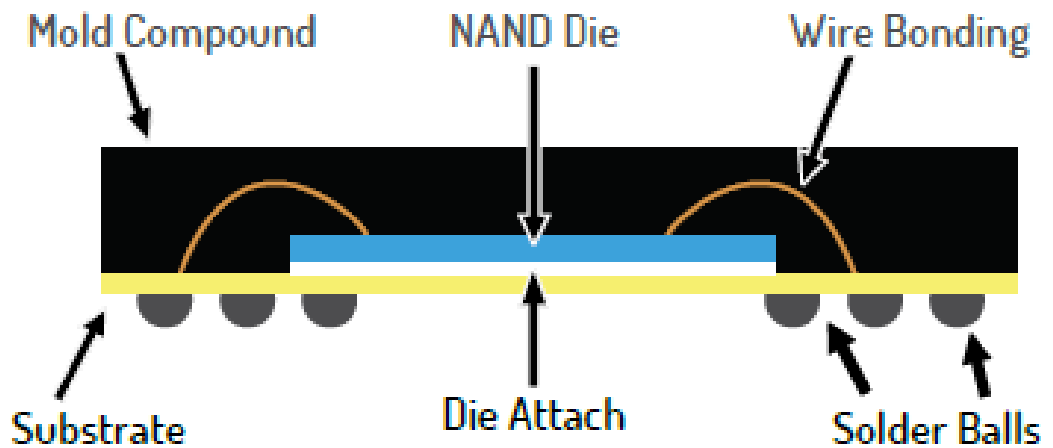
Ek depolama gereksinimleri devam ettikçe, yoğunluğun artması için yeni yollar ortaya çıkmaktadır. Yüksek kapasiteli SSD için ön plana çıkan yeni teknolojilerden biri de 3D Memory.

## 2.6 - NAND Mimari - Bileşen Paketleme

NAND zarı nispeten kırılgandır ve yerleştirme ve bağlama için özel ekipman gerektirir. Tipik olarak NAND Zarı, doğrudan bir devre kartının aksine bir koruyucu bileşen paketinin içine yerleştirilir. Çoğu, JEDEC tarafından tanımlanan açık endüstri standardını takip eder.

Bu bileşenler, üreticinin, tek bir pakette bir veya birkaç NAND Zarını, tipik olarak bir TSOP veya BGA paketinin standart pin çıkışı ile yerleştirmesine olanak tanır. Standart paketler, lehimleme makineleri ile çalıştırılmadan önce parçaların baskılı devre kartlarına (PCB) yapıştırılmasından dolayı çekme ve yerleştirme sistemleri tarafından kolayca işlenir.

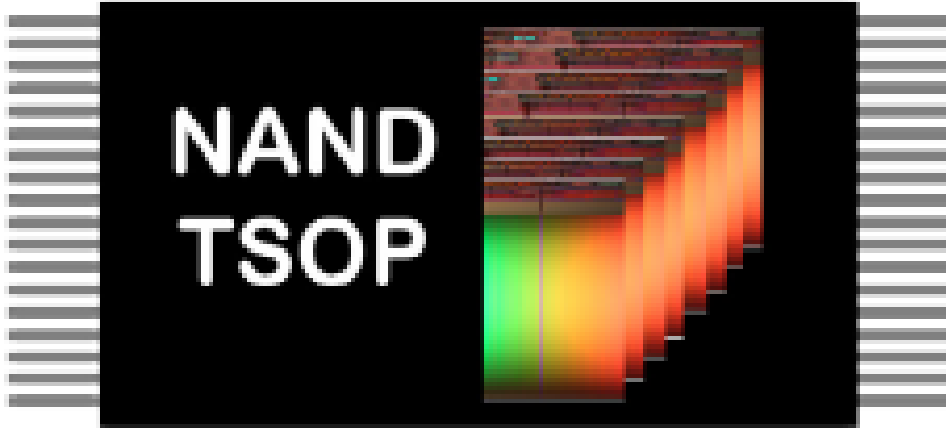
### BGA Packaged NAND



Basit bir tek NAND Zarı BGA bileşeni paketi görüntüsü yukarıda gösterilmiştir.

Ona bağlı bir NAND Zarı olan bir substrat vardır. Tel bağlama makineleri, NAND Zarını alt taraftaki toplara bağlanan substrata bağlar. Kablolama tamamlandıktan sonra, bir kalıplama bileşiği sağlam bir fiziksel paket sağlamak için substratın üstünü kapsüller.

Aşağıdaki şekil, tek bir büyük kapasiteli NAND bellek cihazı oluşturmak için birbiri üzerine yığılmış çok sayıda kat kat ile bir NAND bileşeninin bir kesitini göstermektedir.

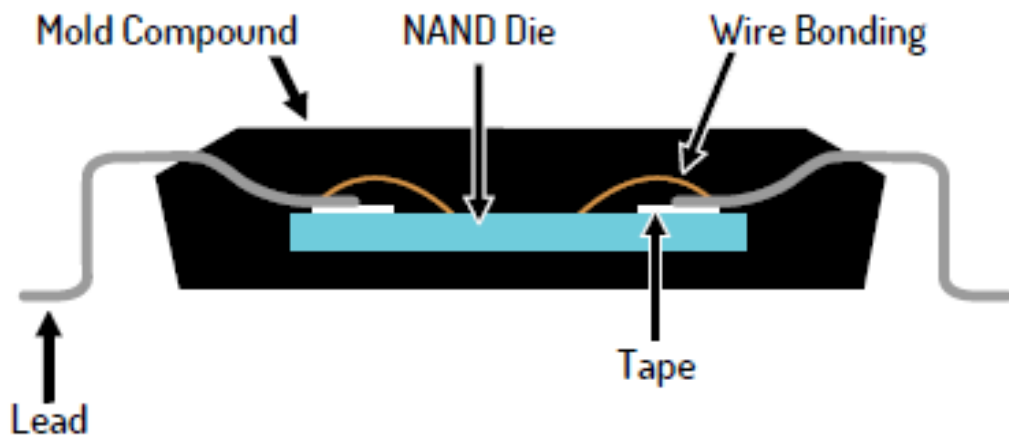


Her bir NAND Zarı arasında bir yalıtım tabakası bulunur ve her bir NAND Zarından bir tel bağlama makinesi kullanılarak substrata bağlantılar yapılır.

Substrat, yığılmış kalıbın tabanı olan çok ince bir PCB (Baskılı Devre Kartı) gibidir. Daha önce gösterilen tek Zar kesitinde olduğu gibi, bileşenin tüm üstü kapsülendir.

Parçalar bir JEDEC standart form faktörünü paylaşıyorsa, parçalar içerideki Zar sayısına bakılmaksızın fiziksel olarak aynı görünür.

### TSOP Packaged NAND



Bu resimde TSOP paketindeki bir NAND Zarı gösterilmektedir. BGA paketinden TSOP paketi ile birkaç fark var. İlk olarak, NAND ölçeğindeki bağlantıları dış dünyaya taşıyan bilyaların aksine kurşunlar vardır. İkincisi, tüm montaj sadece en üstte değil, kapsülendir. Tek çıkan bağlantı, lead çerçevesinin sonu. BGA paketinde olduğu gibi, bir TSOP paketinin içinde birden çok kalıp istiflenebilir.

### 3. Flaş Denetleyicisi

Bir bellek denetleyicisinin iki temel görevi vardır:

1. hem ana bilgisayara hem de Flaş belleğine en uygun arabirimi ve protokolü sağlamak;
2. Verilerin verimli bir şekilde ele alınması, aktarım hızının en üst düzeye çıkarılması, veri bütünlüğü ve saklanan bilgilerin saklanması.

Bu tür görevlerin yerine getirilmesi için, zaman-kritik görevleri ele almak üzere özel bir donanım ile birlikte standart bir işlemci (genellikle 8–16 bit) yerleştirilerek uygulamaya özel bir aygıt tasarlanmıştır.

Genel olarak, bellek denetleyicisi donanımda veya bellek yönetiminde uygulanan dört bölüme ayrılabilir (Şekil 1.3).

Ana bilgisayardan Flaş'a geçerken, ilk kısım, gerekli endüstri standardı protokolünü (PCIe, SAS, SATA, vb.) Uygulayan ana bilgisayar arabirimidir. Böylece SSD'ler ve ana bilgisayarlar arasında hem mantıksal hem de elektrik birliğinde çalışabilirlik sağlar. Bu blok, ana makine tarafından çağrılan komut dizisini çözen ve Flash belleklerine veri akışını işleyen bir donanım (arabellek, sürücü, vb.) Ve ürün yazılımı karışımıdır.

İkinci kısım Flaş Dosya Sistemi (Flash File System) : yani SSD'lerin manyetik diskler gibi kullanılmasını sağlayan dosya sistemi.

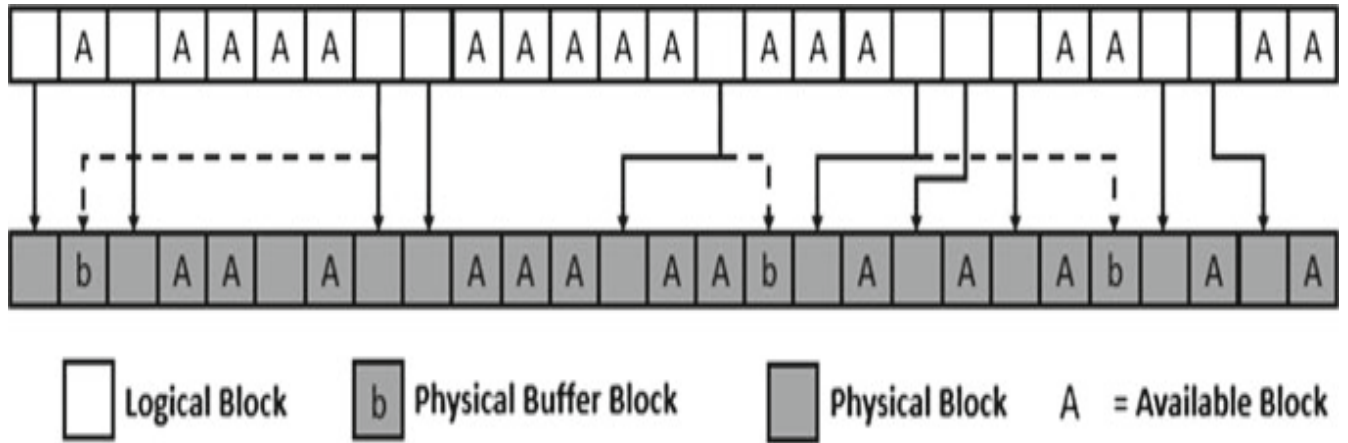
Örneğin, bir dosya oluşturan çok sayıda alt sektördeki sıralı bellek erişimi, dosya tarafından Dosya Ayırma Tablosunu (File Allocation Table-FAT) oluşturmak için kullanılan bağlı listeler (SSD'nin kendisinde saklanır) tarafından düzenlenir.



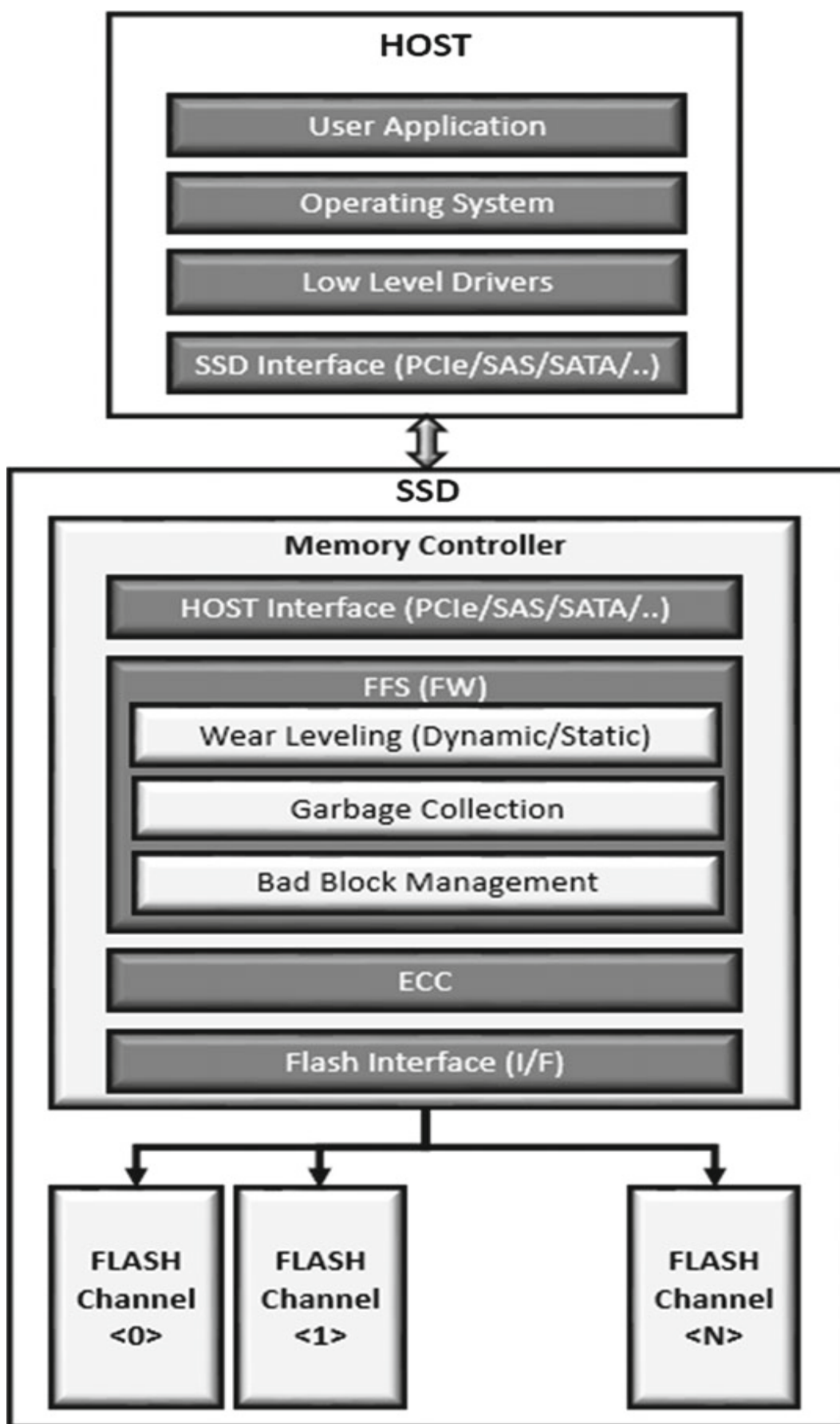
FFS genellikle denetleyicinin içindeki yerleşik yazılım biçiminde gerçekleştirilir; her alt katman belirli bir işlevi yerine getirir. Ana fonksiyonlar şunlardır: Aşınma dengeleme yönetimi, Çöp Toplama ve Kötü Blok Yönetimi.

Tüm bu işlevler için tabloları, Şekil 1.4'te gösterildiği gibi, sektörleri ve sayfaları mantıksal etki alanından fiziksel etki alanına (Flash Çeviri Katmanı veya FTL) eşlemek için yaygın olarak kullanılır. En üstteki satır hafızanın mantıksal görünümüdür, alt satır ise fiziksel olanıdır. Ev sahibi perspektiften, veriler belirli bir mantıksal sektörde şeffaf bir şekilde yazılır ve üzerine yazılır: Flaş sınırlamaları nedeniyle, aynı sayfada yer almak mümkün değildir; bu nedenle, fiziksel bloğa yeni bir sayfa (sektör) atanmalı ve bir önceki geçersiz olarak işaretlenmelidir. Bir noktada, mevcut fiziksel bloğun dolduğunu ve ikincisinin ("tampon" blok havuzundan gelen) bu mantık adresini ele geçirmesi gerektiği açıktır.

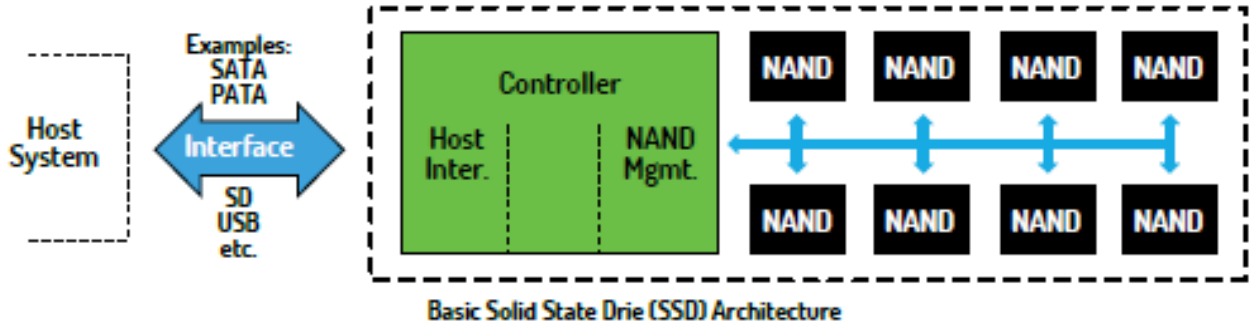
Gerekli çeviri tabloları SSD'nin kendisinde depolanır ve böylece toplam depolama kapasitesini azaltır.



**Fig. 1.4** Logical to physical block management



**Fig. 1.3** High level view of a Flash controller



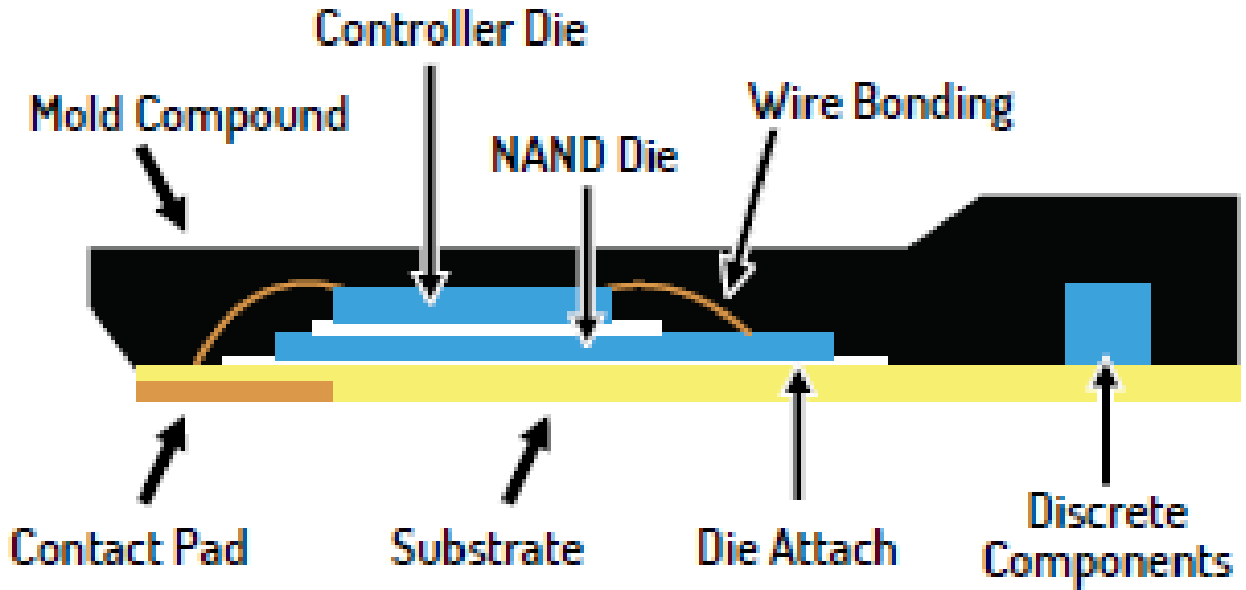
Yukarıda gösterildiği gibi, temel SSD, bir veya daha fazla NAND bileşenini yöneten bir denetleyici yongasından oluşur, bunların her biri birden fazla NAND Zarflarından oluşabilir. Diyagram, son ana bilgisayar arabiriminin kullanılmasının önemli olmadığı anlamda geneldir.



Gerçek bir yaşam örneği olarak, yukarıda SATA SSD'nin dahili devre kartının görüntüsüne bakın. Tüm dikdörtgen bileşenler, bu SSD için tek denetleyici olarak kare bileşenli NAND yongalardır. Denetleyici ünitesinin mutlaka kare bir pakette olması gerekmez, sadece bu durumda olur.

Önceki görüntü hakkında dikkat edeceğiniz bir şey, NAND bileşenlerinin birbirine ne kadar sıkı bir şekilde yerleştirileceğidir. Bir SSD'nin depolama kapasitesindeki sınırlama, bir denetleyicinin her bir Zarı (okuma / yazma)

yeteneđi ile birlikte endüstri standardı paketine kaç tane NAND Zarının entegre edilebileceđidir.



### Cutaway Illustration of microSD Card

MicroSD kartlar gibi çok küçük ambalajlı SSD'ler için, paketlenmiş NAND ve kontrolörün kullanılması için yeterli fiziksel alan yoktur.

Bu durumlarda, denetleyici Zarı ve NAND Zarı üst üste yığılır ve bağlantılar tel ile yapılır.

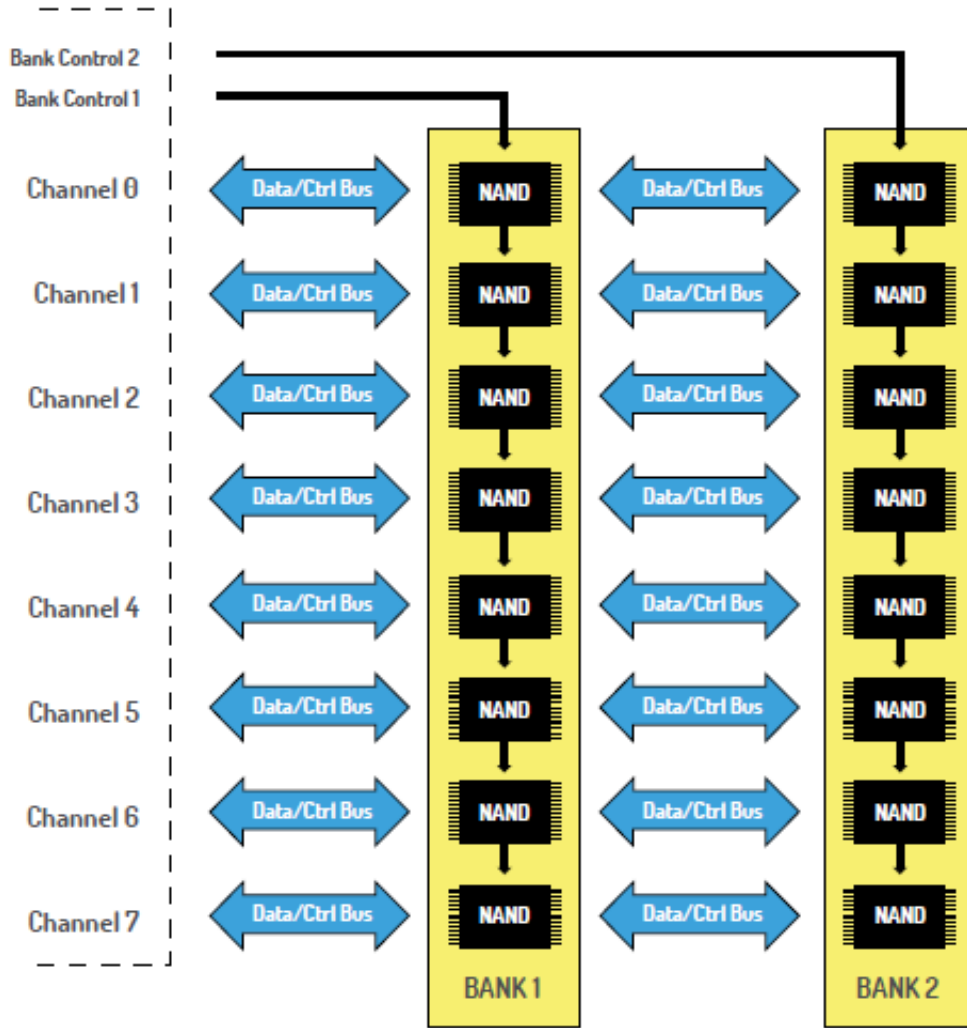
Yukarıdaki şemada, microSD paketinde sadece tek bir NAND Zarı gösterilmektedir, ancak çoklu NAND Zarı, daha yüksek yoğunluk / kapasite parçaları yapmak için denetleyici ile istiflenebilir. Kesit içindeki Zarın boyutu, netlik için gerçek boyut ile orantılı değildir. Birçok 16 NAND Zarı gibi gofret sırtlama teknikleri ile tek bir microSD karta başarılı bir şekilde entegre edilmiştir.

### 3.1 - SSD Denetleyici Mimarisi - Kanallar ve Bankalar

Bu bölüm SSD kontrol cihazı ve NAND flaşı arasındaki bağlantıya odaklanır. SSD tasarımında birçok NAND yapılandırması vardır ve SSD'nin genel gücü, performansı ve maliyeti açısından büyük bir fark yaratır.

Aşağıdaki resim, ortak bir 2.5 "SATA III SSD NAND yapılandırmasını gösterir. Bu örnekte, NAND çiplerine bağlı 8 Kanal bulunmaktadır. Her kanal için 2 NAND bileşeni olan Bankalar vardır.

Belirli bir kanal için Veri / Kontrol Veriyolu üzerinde Banka 1 veya Banka 2'yi aktif hale getiren bir kontrol hattı vardır. Bu kontrol hattı, bileşeni etkinleştirmek veya devre dışı bırakmak için her bir NAND bileşeninin Çip belirleyicisine(chip select) bağlanır.



NAND Banks and Channels Illustration

## NAND Kanalları

Kanallar, denetleyicinin eşzamanlı olarak konuşabileceği flaş yongalarının sayısını ifade eder. Düşük uçlu(low end) SSD'ler genellikle 2 veya 4 kanalıdır; yüksek uç SSD'ler genellikle 8 kanala sahiptir, bazıları 10 kanala sahiptir.

SSD üreticileri, üretim sırasında daha az kanal doldurarak performans ve güç tüketimini dengeleyebilir. Daha fazla kanaldaki sınırlama, tümünün maliyeti artıran kalıp boyutu, iğne sayısı ve güç tüketimi eklenir.

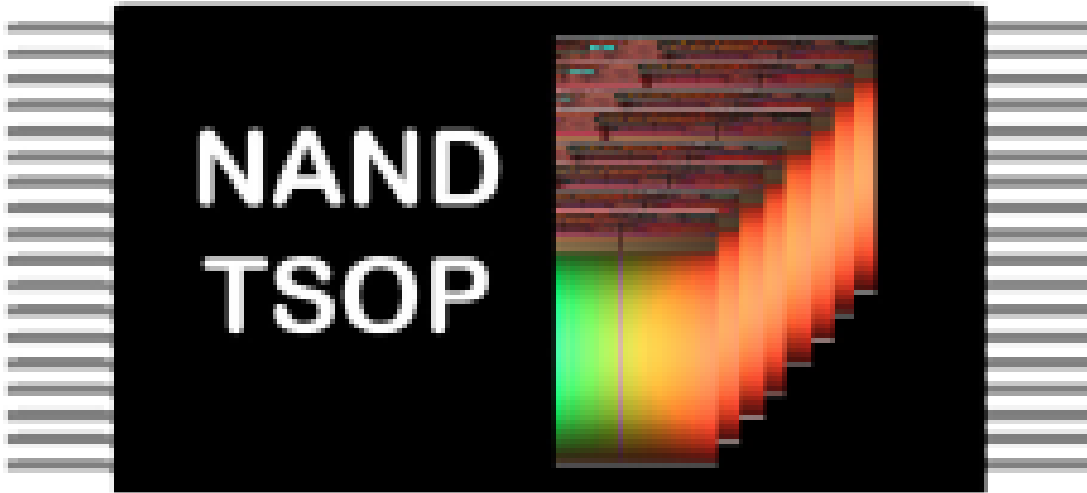
### **NAND Bankaları**

Bir kanalda aynı yerde bulunan her bir flaş çipi bir banka oluşturur. Bir önceki sayfadaki şemaya bakınız. Her kanalın birden fazla çipi olabilir. Maksimum çip sayısı sınırlaması, iğne sayısı, kalıp boyutu ve maliyet faktörlerinin bir sonucudur.

### **Ek SSD Performans Teknikleri**

Performansı daha da artırmak için denetleyiciler serpiştirmeden yararlanabilirler. Her bir NAND flaş bileşeninde çok sayıda kalıp bulunabilir, bu özellikle yüksek yoğunluklu parçalar için geçerlidir. 2, 4 ve 8 kalıplı paketler yaygındır.

Aşağıdaki şekil, tek bir büyük kapasiteli NAND flaş yongası oluşturmak için birbiri üzerine yığılmış çok sayıda katmanı olan bir TSOP NAND bileşeninin bir kesitini göstermektedir.



Çok zarın bir paket için, her bir zarın bir komut gerçekleştirilmesi mümkündür; Bu, serpiştirme olarak adlandırılır ve cihaz performansını önemli ölçüde artırabilir. Serpiştirme yeteneği flaşa, denetleyiciye ve ürün yazılımı desteğine bağlıdır.

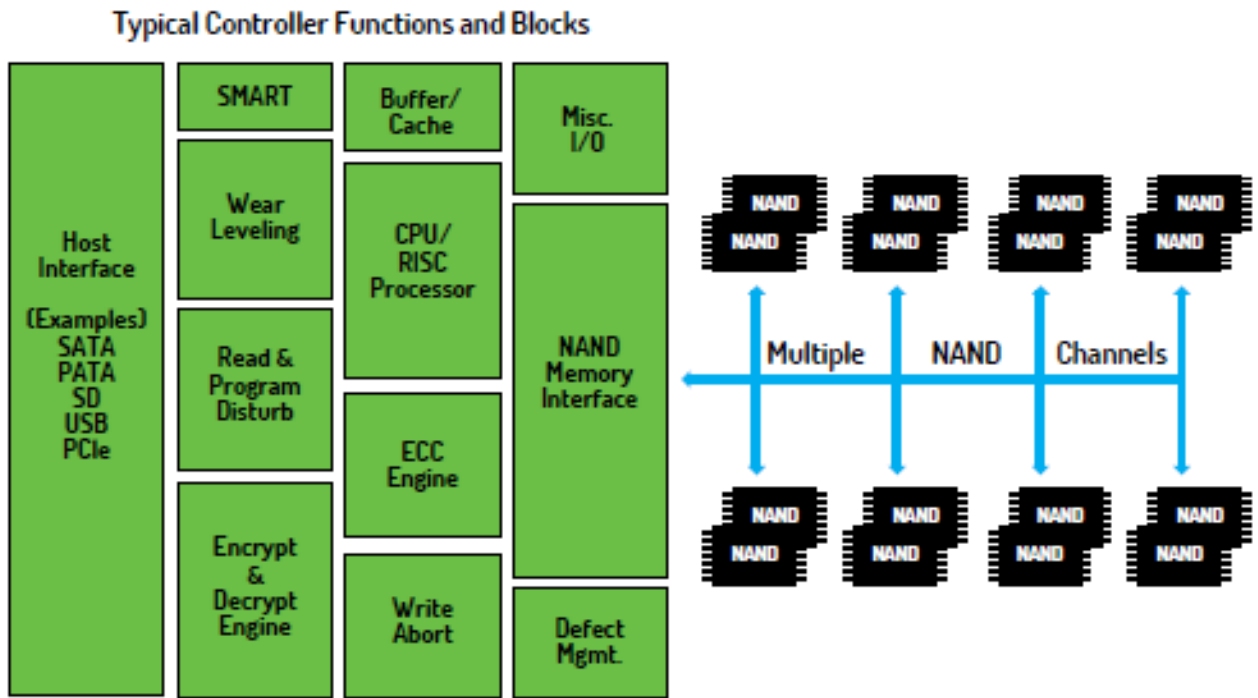
Performansı iyileştirmek için başka bir mekanizma çok planyalı(multi-plane) işlemdir. Bir flaş yongası düzlemlerde, düşük yoğunluklu cihazlar tek planyalı ve 2, 4 veya daha fazla planyalı yüksek yoğunluklu cihazlarla dahili olarak düzenlenmiştir.

Çok planyalı bir düzenekte, tüm planyaların paralel olarak bir komut yürütmesi mümkün olabilir (bu, bir araya getirme gibidir, ancak tek bir zar için). Çok düzlemlili çalışma, mümkün olduğunda, cihaz performansını önemli ölçüde artırabilir.

### 3.2 - SSD Denetleyici Mimarisi - Blok Diyagramı

Basit bir tüketici SD kart denetleyicisi, maliyet ve bazı durumlarda performans için tasarlanmıştır. Bu uygulama için, silikona bir şifreleme ve şifre çözme motoru eklemek için aşırı ve gereksiz bir harcama olurdu.

Güvenli askeri sınıf SSD gibi diğer durumlarda, şifreleme ve şifre çözme mutlak bir zorunluluk olabilir. Diğer uygulamalar gelecekte oluşacak bir hatayı tahmin etmek için SMART verilerine güvenmektedir, bu nedenle SSD beklenmedik bir arızadan önce değiştirilebilir.



**Generic Solid State Drive (SSD) Controller Architecture**

Yukarıdaki şekilde bir SSD'nin temel blokları gösterilmektedir.

## **Ana Bilgisayar Arabirimi**

Bir denetleyicinin ana arabirimi tipik olarak bir endüstri standardı arabirim belirtimine göre tasarlanmıştır. Farklı sistem ve tasarım gereksinimlerini ele almak için çeşitli arayüzler vardır. En popüler SATA, SD, USB, PATA / IDE ve PCIe.

## **SMART (Kendi Kendini İzleme, Analiz ve Raporlama Teknolojisi)**

Bazı denetleyicilerde bulunan SMART işlevi, SSD'nin ve belleğin birçok niteliğine ilişkin verileri izler ve kaydeder. Bunun bir örneği, SSD'de kalan dayanıklılık döngüleri yüzdesini izleme yeteneğidir çünkü bu kalan yaşamın önemli bir belirleyici faktörüdür.

## **Aşınma Seviyelendirme**

Aşınma Seviyelendirme, mevcut NAND boyunca yazım döngülerinin sayısını bile çıkarma yeteneğidir. Her bir NAND bloğu sınırlı sayıda silme / yazma döngüsüne sahip olduğundan, sadece bir fiziksel blok sürekli yazılırsa, dayanıklılık döngülerinden hızlı bir şekilde tükenecektir. Bir denetleyicinin Aşınma Seviyelendirme algoritması, yazılanları farklı fiziksel NAND bloklarına izler ve dağıtır.

## **Okuma ve Program Rahatsızlığı**

NAND flaşının daha ince ve ince iz genişlikleriyle, NAND hücrelerinin veri içeriğini korumak daha fazla sorun ortaya çıkar. Okuma ve Program Rahatsızlığı, hücreler okunduğunda veya yazıldığında, bitişik hücrelere çapraz bağlanmaya ve bazen de değerlerini değiştirerek meydana gelir. Denetleyiciler, algoritmalara ihtiyaç duyarlar ve bazı durumlarda bu olguyu telafi etmek için devrelere ihtiyaç duyarlar.

## **Şifrele ve Şifresini Çöz**

Daha yüksek güvenlik uygulamaları için genellikle bir donanım şifreleme ve şifre çözme motoru, kontrolörün silikonuna yerleştirilir. Şifreleme motoru tipik olarak, anında şifreleme / şifresini çözme hızını sağlamak için donanımda uygulanır. Günümüzde SSD'ler için en popüler şifreleme yöntemi AES256'dır.



## **Tampon / Önbellek**

Denetleyiciler genellikle SSD'nin okuma ve / veya yazma verilerini tamponlamak için kullanılan yüksek hızlı bir SRAM / DRAM önbellek tamponuna sahiptir. Bu önbellek, geçici bellek kullandığı için, güç beklenmedik bir şekilde kaldırılırsa, veri kaybına neden olur. Denetleyici yongasının kendi iç önbelleklerinin yanı sıra harici RAM önbellek çiplerini de görmek normaldir.

## **CPU / RISC İşlemci**

Her SSD'nin kalbi ana işlem çekirdeğidir. CPU / RISC işlemcisinin boyutu ve performansı, denetleyicinin ne kadar yetenekli olabileceğini belirler.

## **ECC Motoru**

Hata Denetleme ve Düzeltme, bugünün SSD'sinin önemli bir parçasıdır. ECC, veri bloğu başına belirli sayıda bite kadar düzeltecektir. ECC olmadan, çok ucuz bellek kullanan düşük maliyetli tüketici flaş kartlarının çoğu mümkün olmazdı.

## **Durdur – Yaz(Write Abort)**

Durdur Yaz, SSD'nin NAND flaşına yazma sırasında ne zaman kaybolduğudur. Pil veya SuperCap destekli önbellek olmadan, aktarımdaki bu veri kaybolur. Bunun daha önemli yönü, SSD'nin dahili meta verilerinin ve ürün yazılımının bozulmamasını sağlamaktır. Bu, Endüstriyel Sınıf ürünlerinde bulunan Yazma Durdurma devresinin işlevidir.

## **Çeşitli G / Ç (I / O)**

NAND bileşenleri için çip seçme pimleri gibi basit işlevler birkaç giriş / çıkış pini ile işlenir. İlk programlama ve üretim için gerekli bir dizi I/O işlevi vardır.

## **NAND Bellek Arayüzü**

NAND bellek arayüzü, NAND bankaları ve kanalları hakkındaki önceki makalede ele alınmıştır. Denetleyicilere bağlı olarak, 10 veya daha fazla tek bir NAND kanalı olabilir. Her kanal bir veya daha fazla NAND çipine sahip olabilir.

## Hata Yönetimi

Her denetleyicinin, kötü bellek blokları ve yeni kusurlarla uğraşmak için bir yöneme ihtiyacı vardır. Bu noktada, bir NAND bloğu kullanılamaz hale geldiğinde, SSD kontrol cihazının parçası üzerinde bazı eylemler gerçekleşmelidir. Bazı durumlarda, yedek bir sektör başarısız bloğun yerini alır. Kötü bir denetleyici tasarımında SSD başarısız olur. Her denetleyici, kusurlarla başa çıkma yöntemine sahiptir.

### 3.3 SSD Denetleyici Fonksiyonları - Aşınma Seviyelendirme

Genellikle, aynı hafıza konumunda saklanan tüm veriler aynı sıklıkta değişmez: bazı veriler genellikle diğerleri çok uzun bir süre değişmezken güncellenir - (en uç durumda, cihazın tüm ömrü boyunca). Sıklıkla güncellenen bilgileri içeren blokların daha fazla sayıda yazma / silme döngüsü ile vurgulandığı, çok nadiren güncellenen bilgileri içeren blokların çok daha az stresli(kullanılmış) olduğu açıktır.

Sorunları gidermek için, her bir sayfanın / bloğun yaşlanmasını mümkün olduğunca minimum ve tekdüze olarak tutmak önemlidir: yani, her sayfaya uygulanan hem okuma hem de program döngülerinin sayısı izlenmelidir. Ayrıca, bir blok için izin verilen maksimum program / silme döngüsü sayısı (yani dayanıklılığı) göz önünde bulundurulmalıdır: SLC NAND hafızalarının kullanılması durumunda, bu sayı 20–30 k döngülerindedir, bu da 10'a düşürülür. Sırasıyla MLC ve TLC NAND için 15 k ve 1-5 k.

Aşınma Seviyelendirme teknikleri, mantıksal olarak fiziksel çevirme kavramını kullanır: ana bilgisayar uygulamasının her zaman aynı (mantıksal) sektörü güncellemesi gerektiğinde, bellek denetleyicisi bu sektörü dinamik olarak farklı (fiziksel) bir sektörle eşleştirir, tabi ki haritalamayı izler. Sektörün güncel olmayan kopyası geçersiz olarak etiketlendi ve silme için uygun. Bu şekilde, tüm fiziksel bloklar eşit olarak kullanılır, böylece yaşlanmayı makul bir değerde tutar.

İki tür yaklaşım mümkündür: Dinamik Aşınma Seviyelendirme, kullanıcının güncelleme talebini takip etmek için kullanılır; en düşük silme sayısına sahip ilk silinen bloğa yazma; Statik Aşınma Seviyelendirme ile Her blok,

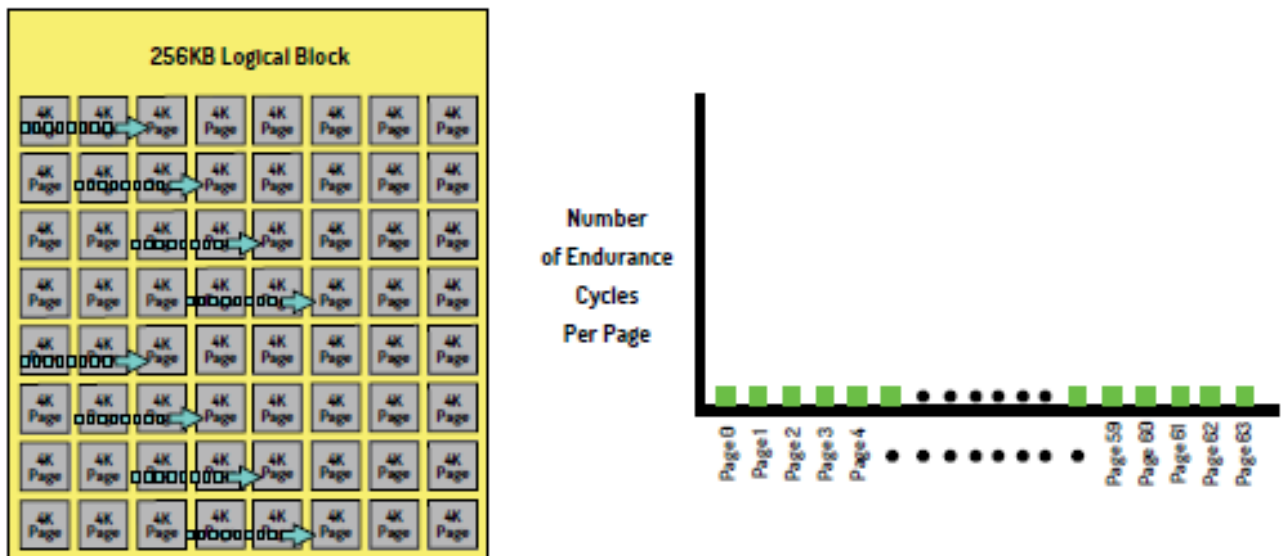
en az modifiye edilmiş olsa bile, yaşlanma ortalama değerinden saptığında yeniden haritalama için uygundur.

Bir SSD kontrol cihazı, ana sistemden, bir veri parçasının nerede okunacağını veya yazılacağını söyleyen komutlar alır. Aşınma Seviyelendirme ile ilgili bu makalenin basitliği için iki varsayım yapacağız:

1) Her bir veri parçası 4KB'dir ve; 2) NAND sayfaları da 4KB'dir. Gerçek dünya koşullarında, veri ve sayfa boyutlarının boyutları, kullanılan ana sisteme ve NAND belleğine bağlı olarak değişebilir.

Ana sistem, okumak veya yazmak istediği verilerin Mantıksal Blok Adresini (logical block address - LBA) sağlar. SSD kontrol ünitesinin LBA'yı aynı Fiziksel Blok Adresine yazmasını veya yazmasını göreceli olarak doğru olacaktır.

## Wear Leveling and Sequential Writing to NAND Memory



Sequentially Written Flash Block

Yukarıdaki resim, her biri 4 KB depolama kapasitesine sahip 64 sayfadan oluşan 256 KB'lık NAND bloğunu gösterir. Ana sistem, sıralı LBA'larda 256 KB veri yazıyorsa ve SSD denetleyicisi bu verileri sıralı Fiziksel Blok Adreslerinde depolar.- İlk 4K Sayfasından başlayıp blokta 64. ile biten - Daha sonra tüm blok için etkili bir şekilde 1 dayanıklılık döngüsü kullanılır.

256 KB bloğunun sağındaki grafik, sıralı verilerin NAND flaş cihazında veri depolamanın ideal yöntemi olduğunu gösterir. Yazma döngülerinin tümünü NAND Sayfalarına ve Bloklarına eşit olarak dağıtır, böylece tek bir NAND Sayfası / Blok diğer Sayfalar / Bloklar öncesinde yıpranmaz.

Birkaç açıklamaya devam etmeden önce:

- Bir Dayanıklılık döngüsü sadece bir silme gerçekleştiğinde ortaya çıkar, bu nedenle ilk yazıda aslında bir dayanıklılık döngüsü olmaz.
- NAND hücresinin okunması, NAND hücresinin dayanıklılık döngüsünü etkilemez.

### Nand Belleğe Aşınma Seviyelendirme ve Sıralı Olmayan Yazma



Non-Sequentially Written Flash Block

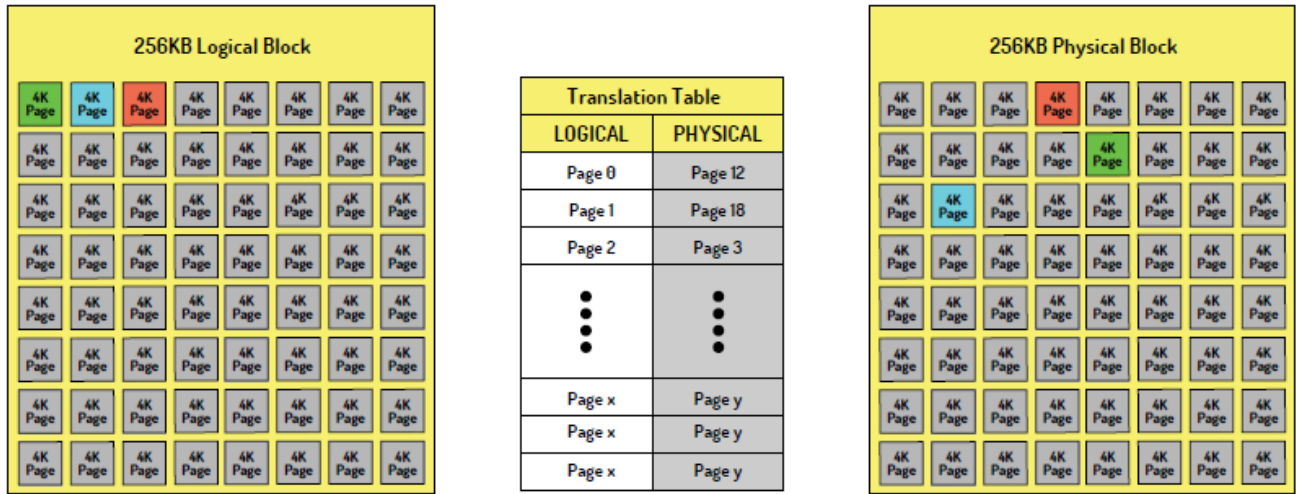
Ne yazık ki, SSD'lerin gerçek hayattaki kullanımı nadiren ideal sıralı kullanım modellerine sahiptir. Herhangi bir dosya değiştirildikten sonra güncellenmesi gereken Dosya sistemi dizinleri vardır. Ana bilgisayardaki veri boyutlarında ve SSD'deki NAND'ın Sayfa / Blok boyutlarında uyumsuzluklar var. Önceden yazılmış blokları ve diğer birçok faktörleri geri alma ihtiyacı var.

Yukarıdaki Sıra Olmayan görüntü, verilerin sürekli olarak ilk iki 4K Sayfasına yazıldığı en kötü durum durumunu gösterir. SSD kontrol cihazı bu verileri NAND'ın bu fiziksel sayfalarına yazmaya devam ederse, bu hücrelerin toplam dayanıklılık döngü sayısını hızla tüketir.

## Aşınma Seviyelendirme Ne Yapar?

SSD kontrol cihazlarındaki Aşınma Seviyelendirme algoritmaları, tüm SSD'de ana sistem yazımlarını eşit bir şekilde dağıtmaya çalışır. NAND flaş, blok başına sınırlı sayıda yazara sahip olduğundan, aşınma seviyelendirme SSD'nin her kullanım ömrünün bitiminden önce her dayanıklılık döngüsünü kullanmaya çalışır.

Farklı SSD kontrol tasarımcıları tarafından kullanılan birçok farklı şema vardır, ancak hepsi bir çift özelliği paylaşırlar.



Logical to Physical Translation

Yukarıda gösterildiği gibi, Mantıksal Blok Adres (LBA) için yazılmış ana veriyi, kullanılan en az dayanıklılık döngüsüne sahip fiziksel bir yere depolarlar. Aynı LBA'ya yazılan ana bilgisayar verileri tipik olarak NAND'ın aynı fiziksel konumunda saklanmaz. Denetleyici, bir tablo veya başka bir yöntemde Mantıksal Fiziksel bloktan çeviriyi takip etmelidir.

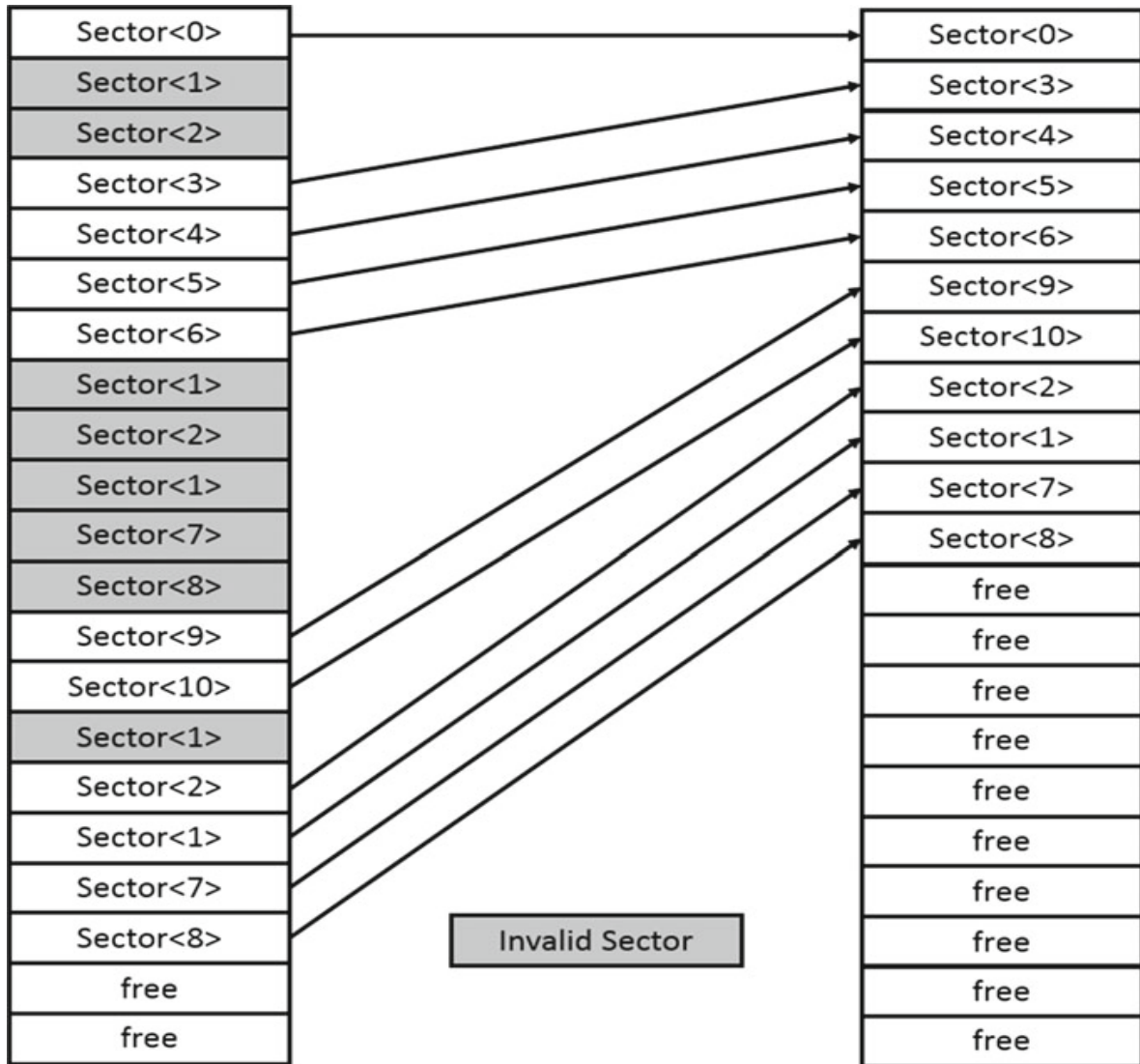
Diğer bir yaygın durum SSD'deki statik verilerin hiçbir zaman hareket etmemesidir - İşletim Sistemi ve Uygulama verileri gibi. SSD'de bir kez depolanır ve daha sonra yalnız bırakılır. Bu durumlar için çoğu yeni kontrol cihazı bu statik verileri diğer NAND konumlarına otomatik olarak taşıyacak ve böylece bu NAND hücrelerinin dayanıklılık döngülerinden yararlanabileceklerdir.

Yalnız Aşınma Seviyelendirmesi, bir SSD'nin tüm görevlerini çözemez, ancak NAND belleğinde bulunan sınırlı dayanıklılığı verimli bir şekilde kullanan daha güvenilir bir SSD oluşturma'nın önemli bir parçasıdır.

### 3.4 SSD Denetleyici Fonksiyonları - Çöp Toplama

Her Aşınma Seviyelendirme teknikleri, güncellemelerle doldurulabilecek serbest sektörlerin kullanılabilirliğine güvenmektedir: serbest sektörlerin sayısı belirli bir eşiğin altına düştüğünde, sektörler “sıkıştırılmış” ve birden fazla, eski kopya silinir. Bu işlem, geçersiz sektörleri içeren blokları seçen Çöp Toplama modülü tarafından gerçekleştirilir, en son geçerli içeriği serbest sektörlerle kopyalar ve ardından bu blokları siler (Şekil 1.5).

Performans üzerindeki etkileri en aza indirmek için, çöp toplama arka planda gerçekleştirilebilir. Aşınma Seviyelendirme dağılımları tarafından tahrik edilen yaşlanma tekdüzeliği, tek sıcak noktalardan ziyade tüm dizi üzerinde stres yıpranmaktadır. Bu nedenle, belirli bir iş yükü ve kullanım süresi verildiğinde, bellek yoğunluğu arttıkça, hücre başına daha az aşınır.



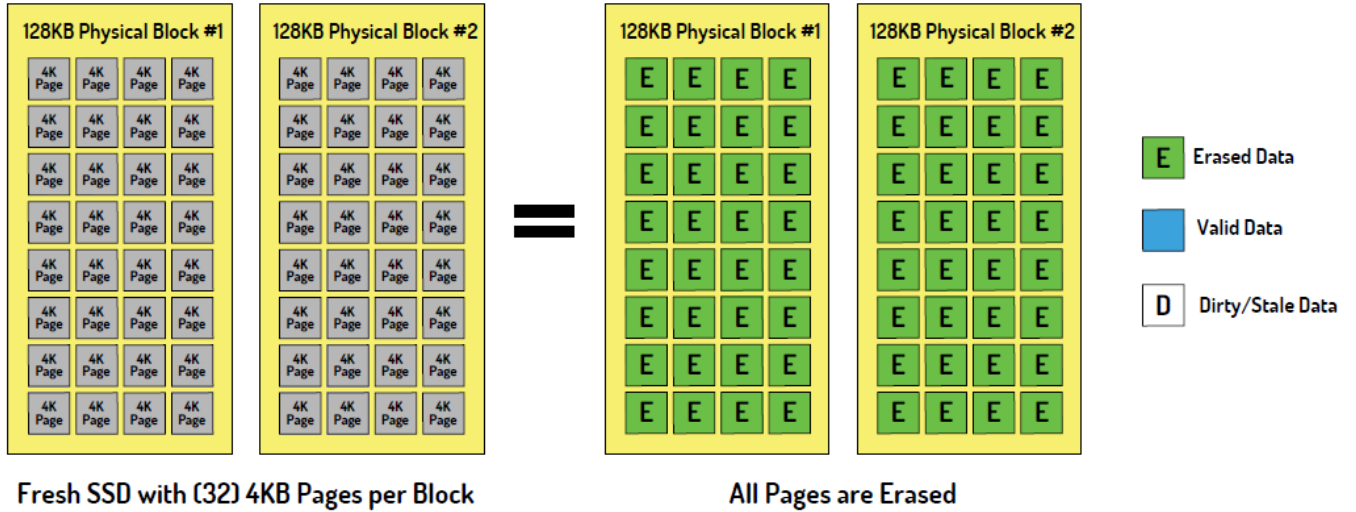
**Fig. 1.5** Garbage collection

Çöp Toplama, daha önce yazılmış veri bloklarını geri almak için yeni verilerle yeniden yazılabilen bir süreçtir.

Çöp Toplama'ya ihtiyaç duyulmasının nedeni, yazılmadan önce NAND'ın silinmesi gereğidir. Bir blok, NAND flaşın en küçük silme birimidir ve genellikle 32 ile 64 arası sayfalardan oluşur.

Bir veri sayfası zaten silinmiş olduğu sürece yazılabilir, ancak sadece tek bir Sayfayı geri almak için bir blokta 32 veya 64 sayfanın silinmesi çok etkisizdir. Çöp Toplama algoritmaları ile ilgili birçok zorluk vardır ve verimlilik, güvenilirlik ve performans tasarımından etkilenebilir.

### Aşınma Seviyelendirme Ne Yapar?

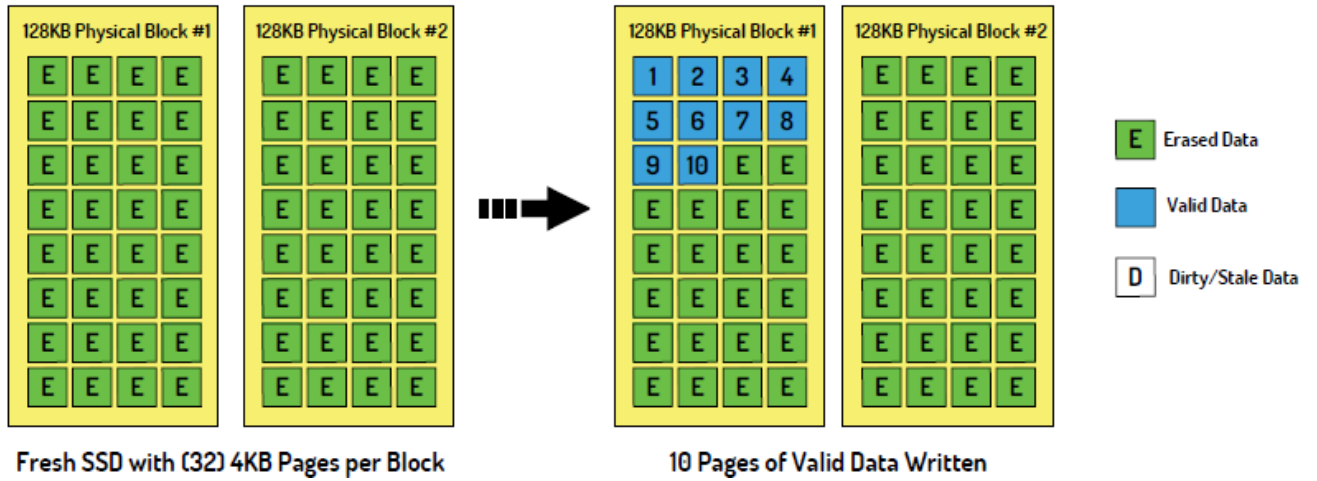


Yukarıdaki şemada, her biri 4 KB veri depolayabilen 32 sayfa içeren yeni bir SSD bloğu gösterilmektedir. Bu, NAND'ın tipik bir yapılandırmasıdır ve bu bölümün geri kalanı için kullanacağız.

Gördüğünüz gibi tüm Sayfalar fabrikadan silinir ve yazılmaya hazırdır.



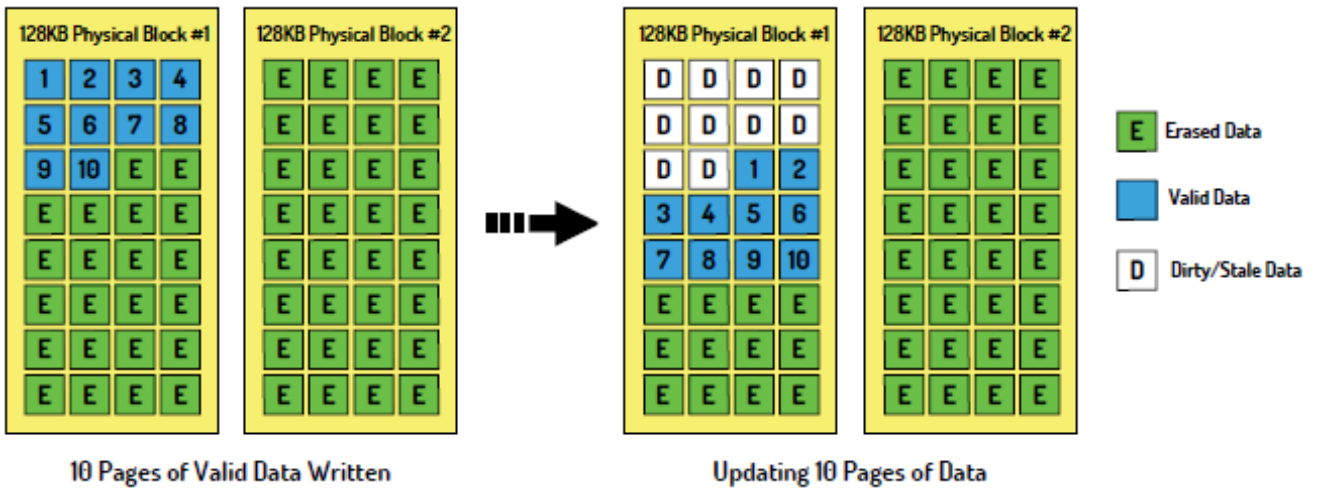
## İlk Sayfa Taze SSD'ye Yazılıyor



Taze bir NAND bloğuna yapılan ilk yazıda, işler kolay. Tüm sayfalar silinir ve bu silinmiş sayfalara sıralı olarak yazılması meselesidir.

Yukarıda gösterilen, tipik bir Çöp Toplama algoritmasının, önceden silinmiş bir bloğa ilişkin 10 yeni sayfa verisinin ilk görevini nasıl yerine getireceğidir.

## Sayfa Verilerinin Güncellenmesi

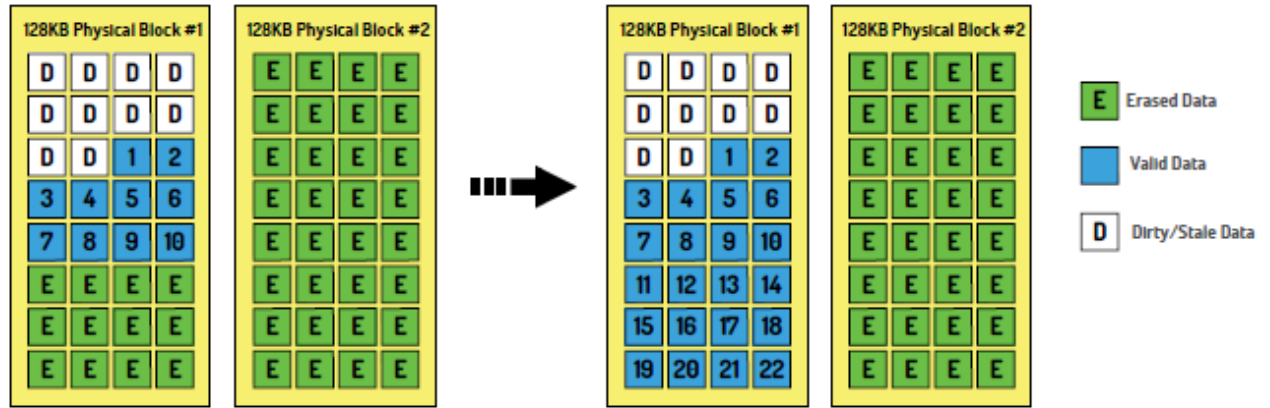


NAND'a orijinal olarak yazdığımız 10 sayfayı güncelleyerek yukarıdaki diyagramda bir adım daha ileri gidiyoruz. Güncellemek istediğimiz sayfaları silmek için bloğun tamamını silmemiz gerektiğinden, güncellenmiş verileri bir sonraki silinebilir sayfalara kopyalayıp yazdık ve önceki sayfaları kirli veya eski olarak işaretledik.

Bu noktada, 10 sayfa geçerli verilerle depolanır ve NAND Bloğu'nun tamamı silinene kadar yazılmayan 10 sayfa kirli olarak işaretlenmiştir.



## NAND Blokunun Doldurulması



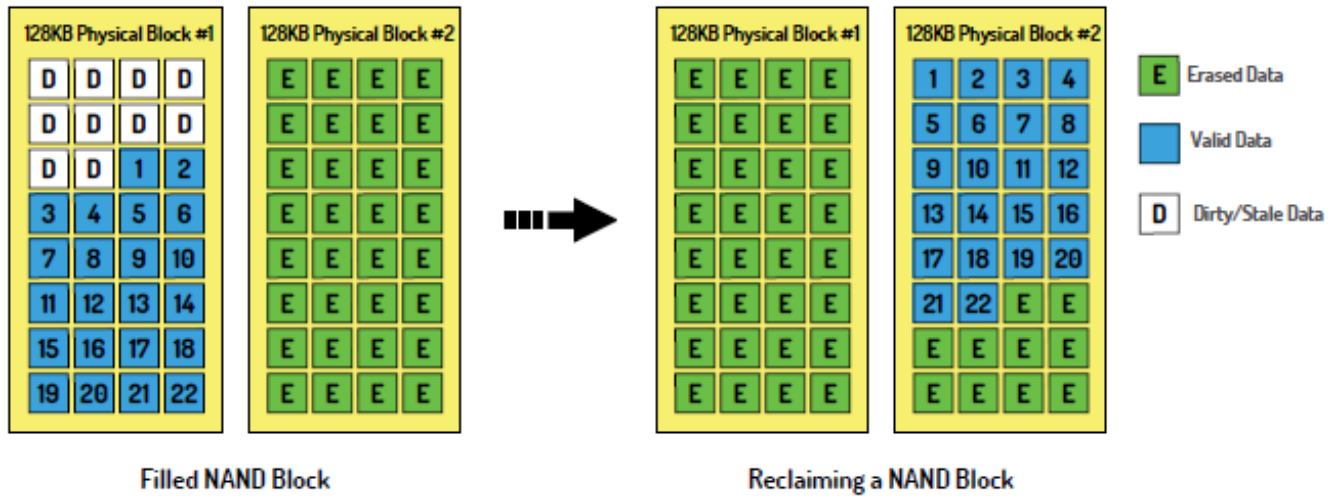
Mixed Block with Valid, Dirty and Erased Pages

Filling the NAND Block

Yukarıdaki bu son diyagram, NAND bloğuna yazılan 12 sayfa daha gösterir. Bu yeni öğeler 11 - 22 NAND Blok # 1'in sonundaki 12 silinen sayfalara yerleştirildi.

Bu noktada, tüm NAND Bloğu kullanılmıştır. Bu sayfaları geri almak için silinmesi gereken yalnızca kayıtlı veriler veya kirli veriler olduğundan, verileri saklamak için artık silinmemiş sayfalar kalmamıştır.

## Sonra ne olur - Çöp Toplama?



Filled NAND Block

Reclaiming a NAND Block

NAND bloğunun tüm sayfaları iyi verilerle veya daha önce geçerli olmayan daha önceden yazılmış verilerle dolu olduğunda ne olur?

Çöp Toplama'nın devreye girdiği yer burası. NAND sayfalarını silinen bir duruma geri almak için, önce bu bloktaki geçerli herhangi bir verinin, yeni bir NAND bloğunda silinecek sayfalara kopyalanması ve yazılması gerekir.

En son diyagramdan da görebileceğiniz gibi, 22 numaralı geçerli sayfa tam Blok 1'den kopyalandı ve tamamen silinmiş Blok # 2'ye yazıldı.

Geçerli veri yeni bloğa başarılı bir şekilde yazıldığında, tüm Blok # 1 silinir. Bu tek adımlı bir işlemdir ve geçerli veya kirli / eski bir veriyi siler. Şimdi Block # 1, yeni bir blokmuş gibi yazılabilir.

Çöp Toplama algoritmasının tasarımı, Yazma Amplifikasyonu gibi diğer faktörlerle çok ilgili olduğundan, genel SSD tasarımının önemli bir parçasıdır.

### **3.5 - SSD Denetleyici Fonksiyonları - TRIM Komutu**

Bu bölüm, bazı SATA, SCSI ve diğer SSD denetleyicileri tarafından desteklenen TRIM komutunu kapsar. TRIM komutu, önceki bölümde açıklanan SSD'nin Çöp Toplama işlemi ile ilgilidir.

Çöp Toplama ile, İşletim Sistemi, bir dosya güncelleme / üzerine yazma gibi, zaten veri içeren LBA'nın (Mantıksal Blok Adresleri) yerini aldığı anda, SSD güncelleştirilmiş verileri yeni sayfalarda saklar ve mevcut sayfaları Kirli (veya Eski) olarak işaretler.

Daha sonra, Çöp Toplama, NAND flaşındaki en küçük silme alanı olan tüm bloğu sildiğinde bu kirli sayfaları geri çağırır. Bu yeni silinen blok artık yeni verilerle yazılabilir.

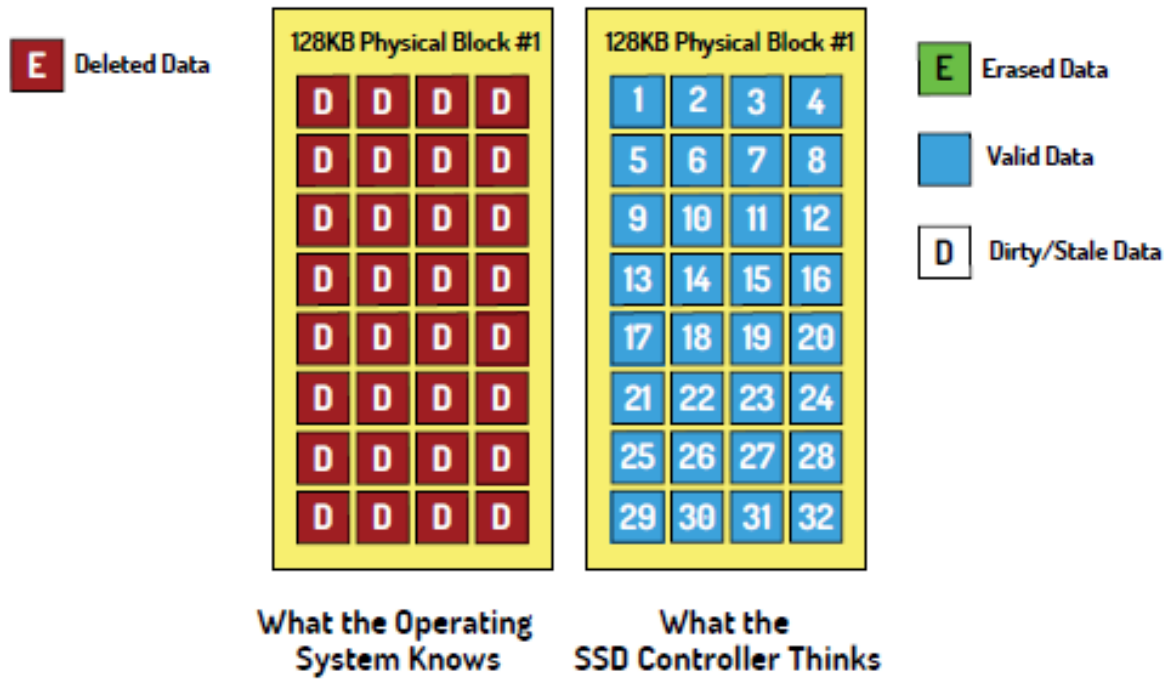
#### **Peki neden TRIM Komutu?**

İşletim sistemi aynı LBA'ların üzerine yazdığı anda ve SSD denetleyicisi bunları gelecekteki silme işlemleri için kirli olarak işaretlediğinde, bu sayfalarda artık geçerli veri olmadığı açıktır.

Ancak, bir dosya işletim sisteminde silindiğinde, çoğu kez yalnızca işletim sisteminin dizini güncellenir. Bu durumlarda, işletim sistemi tarafından artık geçerli sayılmayan LBA'lar için SSD'ye gönderilen bir silme komutu yoktur.

TRIM komutu gelir. Bir dosya kalıcı olarak silindiğinde TRIM komutunu yayınlayarak, ilgili LBA'ların artık geçerli verilere sahip olmadığını ve bir sonraki veri toplama turu için kirli olarak işaretlenebileceğini SSD'ye bildirir.

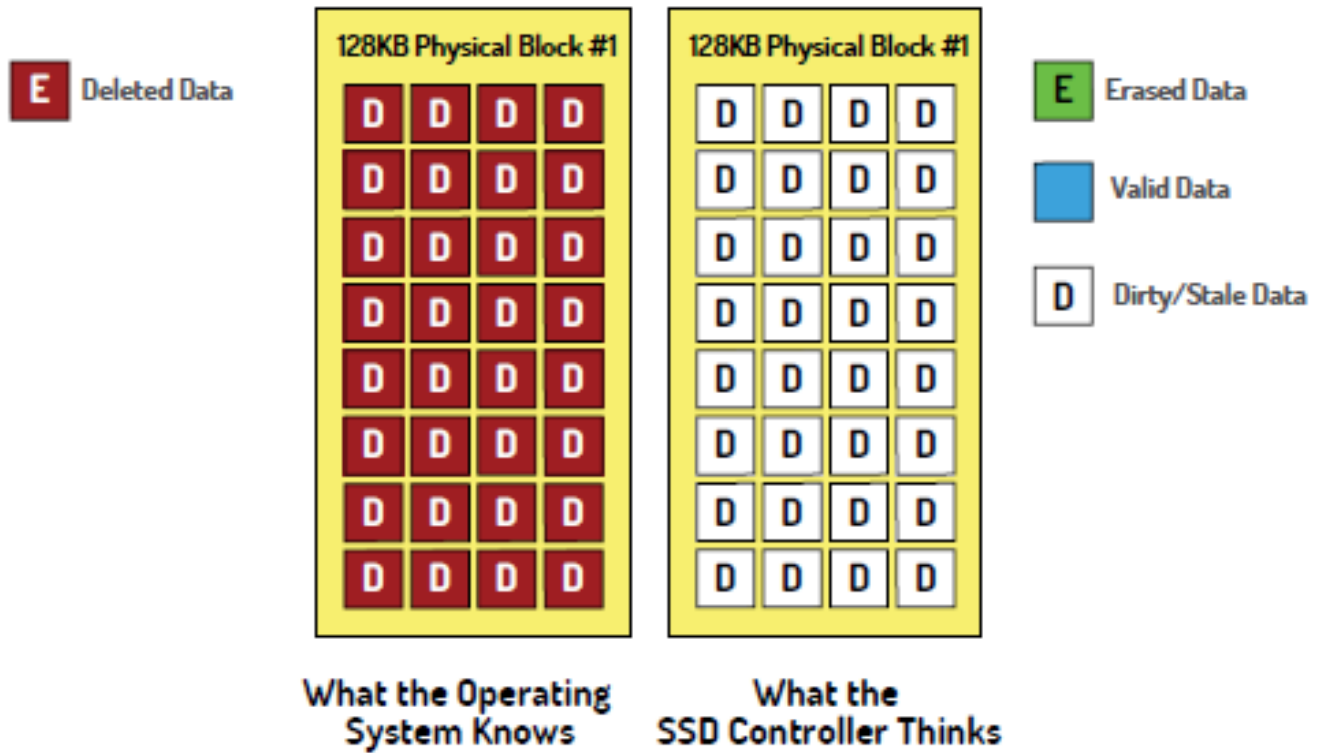
## TRIM Komutu Olmayan İşletim Sistemi ve / veya SSD Örneği



TRIM'in işlevsel olması için, komut hem İşletim Sistemi hem de SSD tarafından desteklenmelidir. Diğerlerinin yanı sıra Windows, Mac ve Linux işletim sisteminin daha yeni sürümleri TRIM komutunu desteklemektedir.

Yukarıdaki resimde TRIM komut desteği olmayan bir sistemin basitleştirilmiş bir örneği gösterilmektedir. Bu örnekte, ana bilgisayar sisteminde, silinmiş 32 sayfa (128 KB) olan bir dosya vardır. İşletim sistemi bunun silindiğini biliyor ve bunların, yazmak için kullanabileceği serbest alanlar olduğunu biliyor.

Ancak, işletim sistemi bu alanları dizinde olduğu gibi işaretlediğinden, SSD denetleyicisine bu sayfaların silinmiş olduğunu bildirecek bir mekanizma yoktur. Sağdaki resimde SSD'nin fiziksel bloklarında, SSD kontrolörünün hala bu bloklarda geçerli veriler olduğuna inanıyor görebilirsiniz. Bu nedenle, bunları kirli olarak işaretlemez ve daha sonra işletim sistemi tarafından üzerine yazılana kadar üzerlerinde çöp toplama işlemi gerçekleştirmez.



### Örnek: TRIM Komutu ile İşletim Sistemi ve SSD

Hem İşletim Sistemi hem de SSD tarafından TRIM desteğine sahip sistemler için, işletim sistemleri ve SSD'ler senkronize edilir. Bir dosyayı sildikten sonra işletim sistemi, silinmiş LBA'lar ile SSD'ye TRIM komutu verir. SSD daha sonra bu LBA'ları kirli olarak işaretler ve çöp toplama işlemi sırasında verimli bir şekilde geri kazanabilir.

Yukarıdaki resimde TRIM komutunun basitleştirilmiş bir örneği gösterilmektedir. Önceki örnekte olduğu gibi, ana bilgisayar sisteminde, silinmiş 32 sayfa (128 KB) olan bir dosya vardır. İşletim sistemi bu alanları dizinde mevcut olarak işaretler, ardından LBA'ları ile TRIM komutunu SSD'ye gönderir. SSD kontrolörü bu sayfaları çöp toplama işlemi için kirli olarak işaretler.

### 3.6 - SSD Denetleyici Fonksiyonları - Aşırı Sağlama

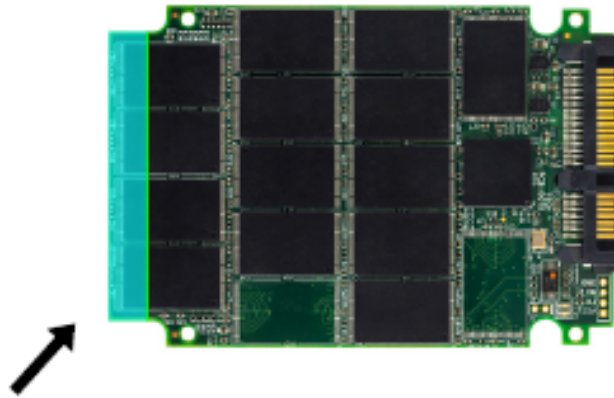
Temelde SSD tarafından kötü blok haritalama, Aşınma Seviyelendirme ve çöp toplama gibi dahili görevleri yerine getirmek için kullanılan işletim sistemi için mevcut olmayan veri alanıdır.

Aşırı Sağlanan alan miktarı genellikle SSD'nin son düşük seviye formatı sırasında fabrikada ayarlanır. Aşırı Sağlama'nın tipik yüzdeleri% 0,% 7 ve% 28'dir.

Piyasada görülen bu yüzdelerin bir örneği, 128GB SSD,% 0 Aşırı Sağlama, % 7 ile 120 GB,% 28 de% 100GB olarak pazarlanabilir. Farklı miktarlarda “kullanıcı alanı” ile aynı SSD'dir.

### Tipik Olarak Sayılmayan Aşırı Sağlama

Çoğu kullanıcı için kolayca görülmeyen bir öge, NAND bileşenlerinin ondalık kapasitelerde satılan SSD'ye karşı ikili kapasite noktalarında satılmasıdır. Aşağıda gösterildiği gibi, SSD'nin NAND kapasitesinin% 7'den biraz fazlası mevcut değildir.



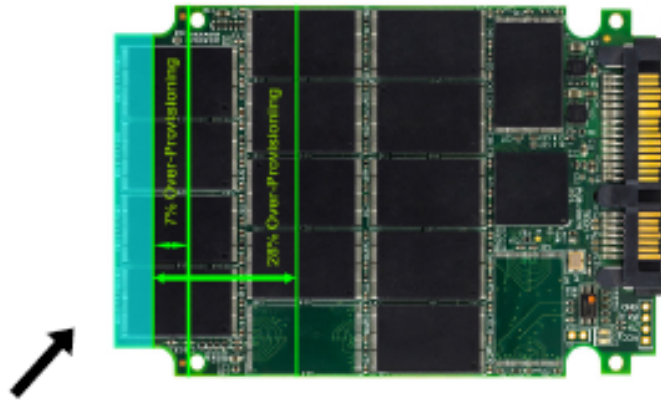
**7% Difference in Binary to Decimal SSD Capacity  
used internally by SSD Controller**

MARKETED CAPACITY	IDEMA DECIMAL CAPACITY (BYTES)	BINARY CAPACITY (BYTES)	DIFFERENCE
128GB	128,035,676,160	137,438,953,472	7.34%
512GB	512,110,190,592	549,755,813,888	7.35%

Uluslararası Disk Sürücü Ekipman ve Malzemeleri Derneği (IDEMA), çoğu HDD ve SSD üreticisinin takip ettiği sürücü kapasiteleri için bir standart (LBA 1-03) yayınladı. Yukarıdaki tabloda 128GB ve 512GB kapasiteleri gösterilmektedir.

İkilik tabandan onluk tabana dönüşümdeki fark Aşırı Sağlama olarak sayılmaz. Boşluk, belleğin depolama, yedek sektörler, kötü blok eşleştirmesi, Aşınma Seviyelendirme ve Aşırı Sağlama'ya benzer diğer görevler için denetleyici tarafından dahili olarak kullanılır.

### % 0,% 7 ve% 28'de Aşırı Sağlama



**7% Difference in Binary to Decimal SSD Capacity used internally by SSD Controller**

0% OVERPROVISIONING	7% OVERPROVISIONING	28% OVERPROVISIONING
128GB	120GB	100GB
512GB	480GB	400GB

Bu tartışma için,% 7'den ikilik tabandan ondalığa olan kapasite zaten kaldırılmıştır. Yukarıdaki tablo, en yaygın üç En Çok Önceden Hazırlık Yüzdesi'nde ondalık 128GB ve 512GB olduğunu göstermektedir.

Aşırı Sağlama, genellikle% 0,% 7 veya% 28'lik parçalar halinde gelir. Düşük kapasiteli ürünlerin çoğu, aşırı yüksek dayanımlı SLC NAND ile üretilen Endüstriyel Sınıf cihazların çoğuna ek olarak% 0 Aşırı Sağlama özelliğine sahiptir.

Bugün piyasada MLC veya TLC NAND ile üretilen birçok istemci SSD için% 7 Aşırı Sağlama kullanılır ve Enterprise SSD genellikle% 28 kullanılır.

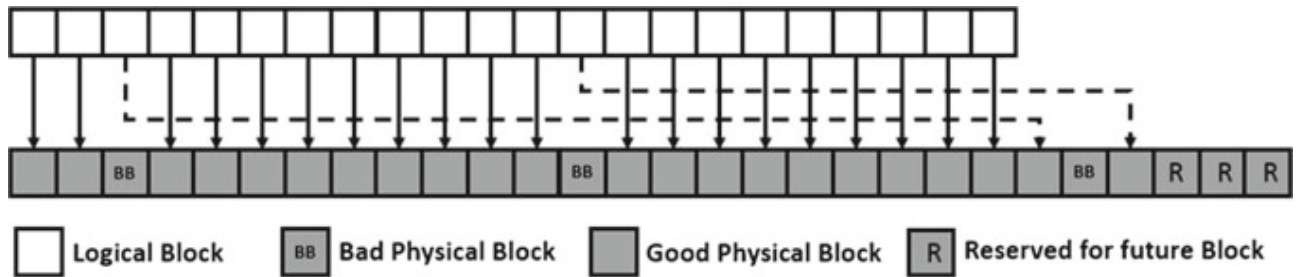
## Aşırı Sağlamanın Avantaj ve Dezavantajları Nelerdir?

Aşırı Sağlama'nın avantajları, dayanıklılığı ve performansı artırırken Yazma Amplifikasyonunu azaltmaktır.

Aşırı Sağlama'nın temel dezavantajı, kullanıcı kapasitesi kaybıdır. Bir kullanıcı daha yüksek güvenilirlik ve daha fazla dayanıklılık ve bazı durumlarda daha az kullanılabilir alan için daha iyi performans ticaretidir.

## 4. Kötü Blok Yönetimi

Aşınma Seviyelendirme algoritması ne kadar akıllı olursa olsun, NAND Flash belleklerinin içsel bir sınırlaması, sözde Kötü Blokların (BB) varlığıyla temsil edilir. Diğer bir deyişle güvenilirliği garanti edilmeyen bir veya daha fazla konum içeren bloklardır. Kötü Blok Yönetimi (BBM) modülü, Şekil 1.6'da gösterildiği gibi, hatalı blokların bir haritasını oluşturur ve korur. Bu harita fabrikada oluşturulur ve SSD'nin ömrü boyunca, bir blok ne zaman kötü olursa olsun güncellenir.



**Fig. 1.6** Bad Block Management (BBM)

## 5. Hata Düzeltme Kodu (ECC)

Bu görev genellikle bellek denetleyicisinin içindeki bir donanım hızlandırıcısı tarafından yürütülür. Gömülü ECC ile hafıza örnekleri de rapor edilmiştir. Birden fazla hatayı düzelten en popüler ECC kodları Reed-Solomon ve BCH'dir.

NAND raw BER, üretimin yaklaşmasından sonra, Shannon sınırının bir sonucu olarak, daha da kötüye gidiyor. Sonuç olarak, yumuşak bilgi işlemeye dayalı düzeltme teknikleri giderek daha popüler hale gelmektedir: LDPC (Düşük Yoğunluklu Parite Kontrolü) kodları bu yumuşak bilgi yaklaşımının bir örneğidir.

## 6. SSD Arayüzleri

SSD'leri sunucuya ve / veya depolama altyapısına bağlamak için kullanılan 3 ana arayüz protokolü vardır: Seri Bağlı SCSI (SAS), Seri ATA (SATA) ve PCI-Express. PCI-Express tabanlı SSD'ler, en yüksek performansı sunar ve çoğunlukla sunucu tabanlı dağıtımlarda, sunucunun içinde bir eklenti kart olarak kullanılır.

SAS SSD'ler oldukça iyi düzeyde performans sunar ve hem üst düzey sunucularda hem de orta ve üst düzey depolama kasalarında kullanılır. SATA tabanlı SSD'ler çoğunlukla istemci uygulamalarında ve giriş seviyesi ve orta sınıf sunucu ve depolama kasalarında kullanılır.

## 7. SAS and SATA

Seri Bağlı SCSI (SAS), depolama aygıtları ve ana bilgisayar arasındaki verileri taşımak için geleneksel olarak kullanılan bir iletişim protokolüdür. SAS, bir seri noktadan noktaya fiziksel bağlantıya dayanmaktadır. Cihaz iletişimini sürmek için standart bir SCSI komut seti kullanır. Bugün, SAS tabanlı cihazlar en çok 6 Gbps hızında çalışmaktadır, ancak 12 Gbps SAS da kullanılabilir. Diğer tarafta, SAS arayüzü ayrıca eski sistemleri desteklemek için daha düşük hızlarda (1.5 Gbps ve / veya 3 Gbps) çalıştırılabilir.

SAS, ikinci nesil SATA sürücülerle geriye dönük uyumluluk sunar. Uluslararası Bilgi Teknolojileri Komitesi (INCITS) T10 teknik komitesi, SAS protokolünü geliştirir ve sürdürür; SCSI Ticaret Birliği (SCSITA) teknolojiyi desteklemektedir.

Seri ATA (SATA veya Serial Advanced Technology Attachment), ana bilgisayar veri yolu adaptörlerini, sabit disk sürücülerini ve katı hal sürücülerini gibi yoğun depolama aygıtlarına bağlamak için kullanılan başka bir arabirim protokolüdür. Seri ATA, eski paralel ATA / IDE protokolünün yerini alacak şekilde tasarlanmıştır. SATA ayrıca bir noktadan noktaya bağlantıya dayanmaktadır. Cihaz iletişimini yürütmek için ATA ve ATAPI komut setlerini kullanır. Bugün, SATA tabanlı cihazlar 3 veya 6 Gbps'de çalışır.

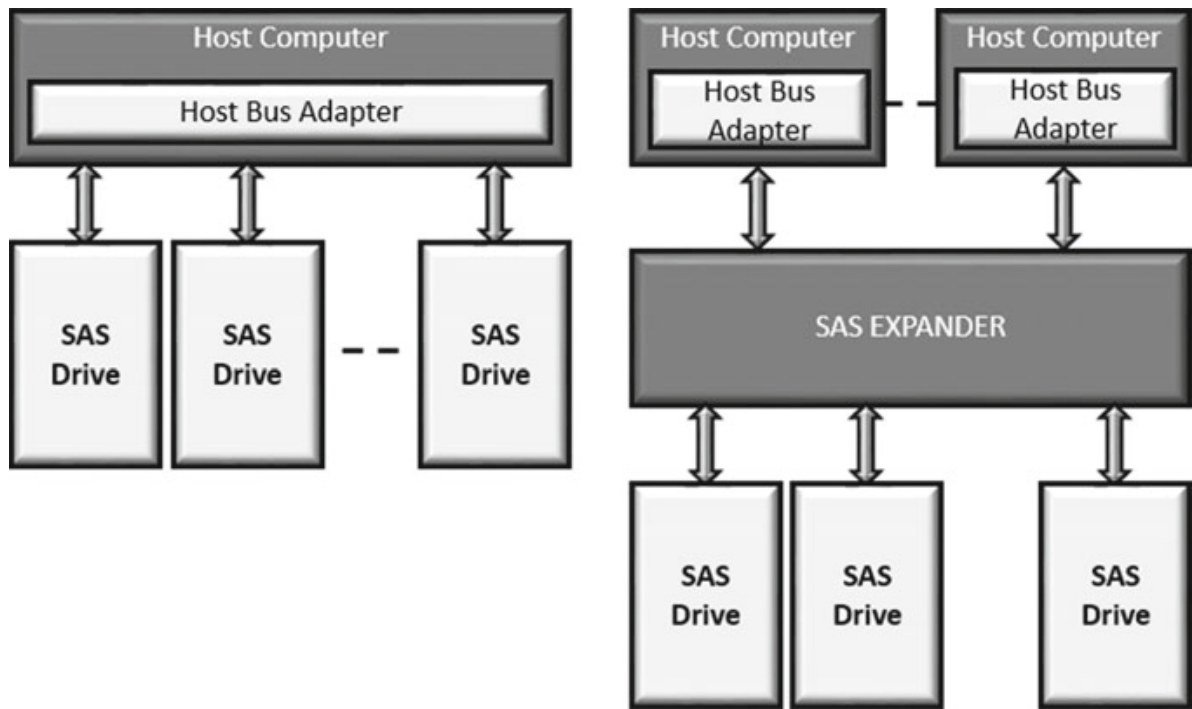
Seri ATA endüstri uyumluluk özellikleri, Seri ATA Uluslararası Organizasyonundan (aka. SATA-IO) kaynaklanmaktadır.



Tipik bir SAS eko-sistemi, bir SAS arka paneline ya da bir noktadan noktaya bağlantı yoluyla bir ana bilgisayar veri yolu adaptörüne takılan SAS SSD'lerden oluşur; bu da, ya bir genişletici ya da doğrudan yoluyla ana mikroişlemciye bağlanır. Şekil 1.7'de gösterildiği gibi.

Her genişletici, toplam 65535 (64k) SAS bağlantısı sağlamak için 255'e kadar bağlantıyı destekleyebilir. Aslında, SAS tabanlı dağıtımlar, paylaşılan bir depolama ortamında çok sayıda SAS SSD'nin kullanılmasını sağlar.

SAS SSD'ler iki bağlantı noktası ile üretilmiştir. Bu çift bağlantı noktası işlevi, ana sistemlerin SAS SSD'lerine yedek bağlantı kurmasına izin verir. SSD'ye olan bağlantılardan birinin kırılması veya düzgün çalışma olmaması durumunda, ana sistemlerin hala SAS SSD'ye sürekli erişimi sağlamak için kullanılabilen ikinci bağlantı noktası vardır. Yüksek kullanılabilirliğin mutlak bir gereklilik olduğu kurumsal uygulamalarda bu özellik önemlidir.



**Fig. 1.7** SAS connectivity

SAS SSD'ler ayrıca hot-plug'ı da destekler. Bu özellik SAS SSD'lerinin sistem çalışırken dinamik olarak kaldırılmasını veya takılmasını sağlar; yeni takılan SAS SSD'lerin otomatik olarak algılanmasını sağlar. Aslında, bir sunucu veya depolama sistemi çalışırken, yeni takılan SAS SSD'leri dinamik olarak yapılandırılabilir ve kullanıma sokulabilir. Ayrıca, SAS SSD'leri çalışan bir sistemden çekilirse, ana sisteme zaten bağlı olan tüm

uçuş verileri SAS sürücüsünde saklanır ve SSD tekrar açıldığında daha sonraki bir noktadan erişilebilir.

SAS'tan farklı olarak tipik bir SATA altyapısı, ana bilgisayar mikroişlemcisi tarafından çalıştırılan bir ana bilgisayar veri yolu adaptörüne bağlanan SATA SSD'lerden nokta-noktadan oluşur. Ayrıca SATA diskleri, SAS SSD'lerden farklı olarak, tek bir portla üretilmiştir. Bu iki ana farklılık, SATA tabanlı SSD'leri giriş veya orta ölçekli dağıtımlar ve tüketici uygulamaları için iyi bir uyum haline getirmektedir.

SATA protokolü hot-plug'ı destekler; Ancak, tüm SATA sürücüleri bunun için tasarlanmamıştır. Aslında hot-plug, gerçekte hala in-flight (havalı\*) olan işlenmiş verilerin bir güç düşüşü sırasında güvenli bir şekilde saklanmasını garantilemek için özel bir donanım (yani ek maliyet) gerektirir.

SATA sürücülerinin SAS arka planlarına bağlanabileceğini vurgulamak gerekir, ancak SAS sürücüleri SATA arka planlarına bağlanamaz. Tabi ki, piyasada geniş SATA etkisi için bu başka bir sebep.

SAS ve SATA arasındaki benzerlikler şunlardır:

- Her iki tip de SAS arka paneline takılır;
- Sürücüler bir SAS sürücü bölmesi modülünde değiştirilebilir;
- Her ikisi de dünya çapında kabul görmüş uzun süredir kanıtlanmış teknolojilerdir;
- Her ikisi de noktadan noktaya(point-to-point) mimarisini kullanır;
- Her ikisi de takılabilir.

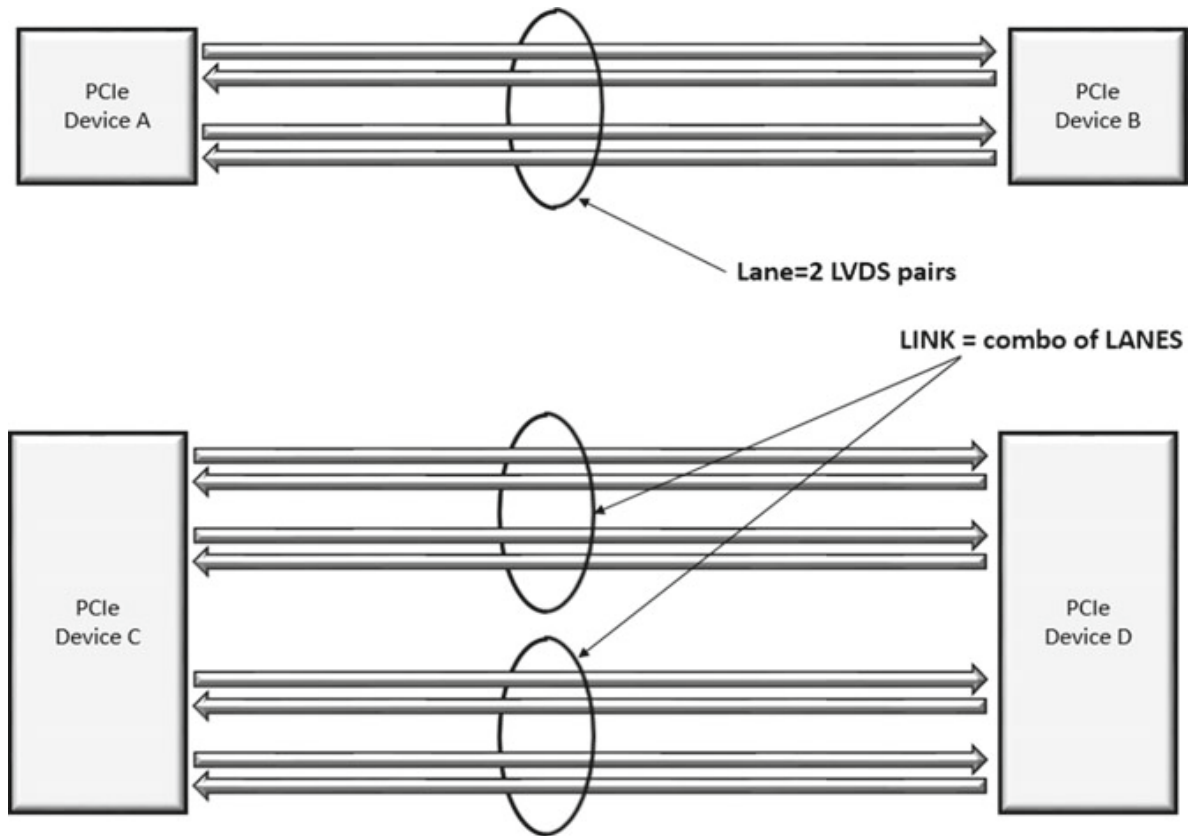
SAS ve SATA arasındaki farklar şunlardır:

- SATA cihazları daha ucuzdur;
- SATA cihazları, ATA komut setini, SAS'ı SCSI komut setini kullanır;
- SAS sürücülerinde çift bağlantı noktası kapasitesi ve daha düşük gecikmeler vardır;
- Her iki tip de SAS arka paneline takılıyken, bir SATA arka paneli SAS sürücülerini barındıramaz;
- SAS sürücüleri çok daha katı özelliklere karşı test edilir;
- SAS sürücüleri daha hızlıdır ve değişken sektör boyutu, LED göstergeleri, çift bağlantı noktası ve veri bütünlüğü gibi ek özellikler sunar;
- SAS bağlantı kümelemeyi (geniş bağlantı noktası) destekler.

## 8. PCI-Express

PCI-Express (Peripheral Component Interconnect Express) veya PCIe, PCI ve PCI-X'in yerini alan bir veriyolu standardıdır. PCI-SIG (PCI Special Interest Group) PCIe tanımını oluşturur ve sürdürür.


PCIe, kurumsal sunucular, tüketici kişisel bilgisayarlar (PC), iletişim sistemleri ve endüstriyel uygulamalar dahil olmak üzere tüm bilgisayar uygulamalarında kullanılır.



**Fig. 1.8** PCI Express lane and link. In Gen2, 1 lane runs at 5Gbps/direction; a 2-lane link runs at 10Gbps/direction

Paylaşılan paralel veri yolu mimarisini kullanan eski PCI veri yolu topolojisinin aksine, PCIe, her aygıtı kök kompleksine (ana bilgisayar) bağlayan ayrı seri bağlantılarla noktadan noktaya topolojiye dayanır. Ayrıca, bir PCIe bağlantısı iki uç nokta arasında tam çift yönlü iletişimi destekler. Veri aynı anda yukarı yönde (UP) ve aşağı yönde (DP) akabilir. Bu özel tek yönlü seri noktadan noktaya bağlantıların her bir çiftine, Şekil 1.8'de gösterildiği gibi bir şerit denir. PCIe standardı sürekli olarak geliştirilmekte olup, PCIe 4.0 piyasadaki standartların en son versiyonudur (Tablo 1.1). Standardizasyon kuruluşu şu anda Gen5'ü

tanımlamak için çalışıyor. PCIe'nin diğer önemli özellikleri arasında güç yönetimi, çalışırken değiştirilebilir cihazlar ve eşler arası veri aktarımlarını işleme yeteneği (ana bilgisayar üzerinden yönlendirmeden iki uç nokta arasında veri gönderme) bulunur.

PCIe Bandwidth & Frequency 		
Year	Bandwidth	Frequency/Speed
1992	133MB/s (32 bit simplex)	33 Mhz (PCI)
1993	533MB/s (64 bit simplex)	66 Mhz (PCI 2.0)
1999	1.06GB/s (64 bit simplex)	133 Mhz (PCI-X)
2002	2.13GB/s (64 bit simplex)	266 Mhz (PCI-X 2.0)
2002	8GB/s (x16 duplex)	2.5 GHz (PCIe 1.x)
2006	16GB/s (x16 duplex)	5.0 GHz (PCIe 2.x)
2010	32GB/s (x16 duplex)	8.0 GHz (PCIe 3.x)
2017	64GB/s (x16 duplex)	16.0 GHz (PCIe 4.0)
2019	128GB/s (x16 duplex)	32.0 GHz (PCIe 5.0)

**Table 1.1** Throughput of different PCIe generations

Buna ek olarak, PCIe, paralel veri yolu mimarilerine göre kablo sayısını önemli ölçüde azaltan bir seri teknolojiden yararlanarak pano tasarımını basitleştirir.

İki cihaz arasındaki PCIe bağlantısı 1–32 şeritli olabilir. Paket verileri şerit boyunca şeritlidir ve aygıt başlatması sırasında şerit sayısı otomatik olarak görüşülür.

PCIe standardı, çoklu genişlikler için yuvaları ve konektörleri tanımlar:  $\times 1$ ,  $\times 4$ ,  $\times 8$ ,  $\times 16$ ,  $\times 32$ . Bu, PCIe'nin düşük verimlilik, düşük maliyetli uygulamalar ve performans açısından kritik uygulamalar sunmasına izin verir.

PCIe, bir işlem katmanı, bir veri bağlantı katmanı ve bir fiziksel katmandan oluşan paket tabanlı katmanlı bir protokol kullanır. (Şekil 1.9'da gösterildiği gibi.) İşlem katmanı, verilerin ve durum mesajı trafiğinin paketlenmesini ve paketten çıkartmasını (de-packetizing) sağlar. Veri bağlantı katmanı, bu İşlem Katmanı Paketlerini (TLP'ler) diziler ve iki uç

nokta arasında güvenilir bir şekilde teslim edilmelerini sağlar. Bir verici cihaz bir uzak alıcı cihaza bir TLP gönderir ve bir CRC hatası tespit edilirse, verici cihaz bir bildirim alır. Verici cihaz TLP'yi otomatik olarak tekrar eder. Hata kontrolü ve arızalı paketlerin otomatik tekrarlanması ile PCIe çok düşük Bit Hata Oranı (Bit Error Rate) sağlar.

Fiziksel Katman iki bölümden oluşmaktadır: Mantıksal Fiziksel Katman ve Elektriksel Fiziksel Katman.

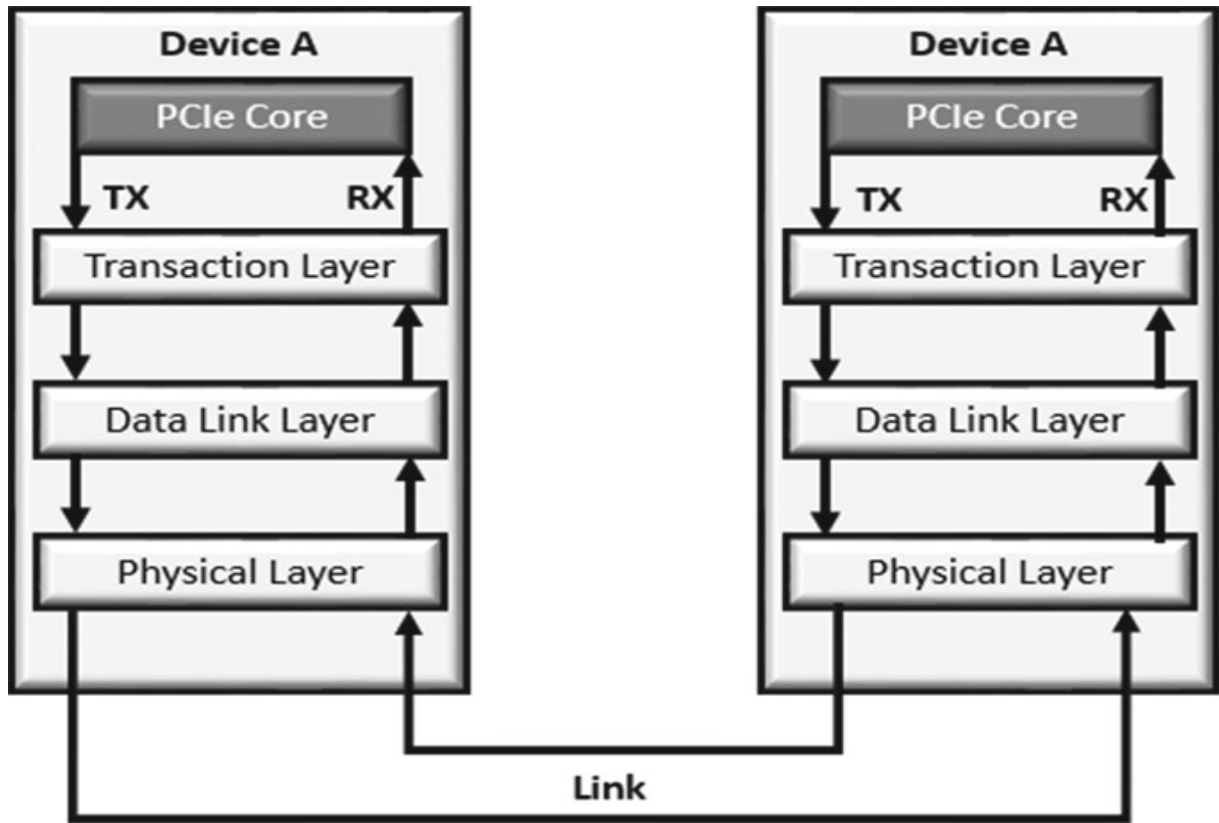
Mantıksal Fiziksel Katman: Link üzerinde iletim yapmadan önce paketlerin işlenmesi için mantık kapılarını ve Bağlantıdan Veri Bağlantısı Katmanına giden paketleri işlemektedir.

Elektriksel Fiziksel Katman: Fiziksel Katman'ın analog arabirimidir: her şerit için farklı sürücü ve alıcılardan oluşur.

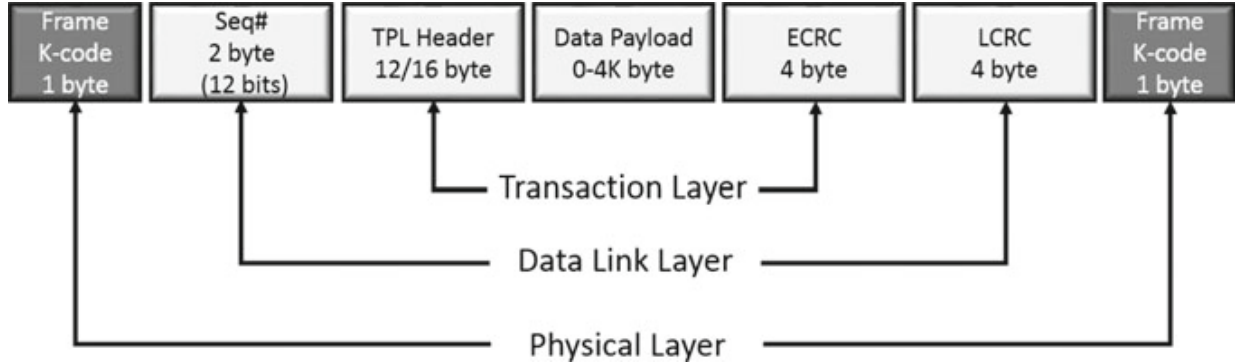
TLP montajı Şekil 1.10'da gösterilmiştir. Başlık ve Veri Yüğü TLP'nin temel bilgileridir: İşlem Katmanı, bu bölümü uygulama yazılımı katmanından alınan verilere dayanarak toplar.

Pakete isteğe bağlı bir Uçtan Uca CRC (End-to-End CRC- ECRC) alanı eklenebilir. ECRC, Header ve Data Payload içindeki CRC hatalarını kontrol etmek için bu paketin son(nihai\*) hedeflenen cihazı tarafından kullanılır. Bu noktada Veri Bağlantısı Katmanı, kimliği korumak için bir sıra kimliği ve yerel CRC (local CRC-LCRC) alanı ekler. Ortaya çıkan TLP, her biri pakete 1 baytlık bir Başlangıç ve Bitiş çerçeveleme karakterlerini birleştiren Fiziksel Katmana iletilir. Son olarak, paket mevcut şeritleri kullanılarak Bağlantıda kodlanır ve farklı şekilde iletilir.

Bugün, PCIe, ana bilgisayar CPU'larını yerleşik IC'lerle ve ek çevresel genişletme kartlarıyla birbirine bağlayan ana anakart teknolojisi olarak, tüketici dizüstü bilgisayarlarından kurumsal sunuculara kadar hemen hemen tüm bilgisayarlarda kullanılan yüksek hacimli bir emtia(satılacak şeyler, mallar) ara bağlantıdır.



**Fig. 1.9** PCIe Layered architecture



**Fig. 1.10** Transaction Layer Packet (TLP) assembly

## 9. Yüksek Hızlı Arayüz İhtiyacı

İşlemci satıcıları, bireysel işlemci çekirdeğinin performansını artırmaya, birden çok çekirdeği tek bir çipte birleştirmeye ve çoklu işlemci sistemlerinde birden fazla çipi birbirine yakınlaştırabilecek teknolojiler geliştirmeye devam etti. Sonuçta, böyle bir senaryodaki tüm çekirdeklerin aynı depolama alt sistemine erişmesi gerekir.

Kurumsal BT (bilgi teknolojisi) yöneticileri, çok işlemcili sistemleri kullanmaya isteklidir çünkü bir sistemin işlem yapabileceği saniye başına

I / O işlem sayısını (IOPS) ve ayrıca Watt başına IOPS sayısını artırma potansiyeline sahiptir.

Bu çok işlemcili bilgi işlem yeteneği, maliyet ve güç tüketimine göre daha iyi IOPS sunar - işlem öğelerinin verilere zamanında erişebildiğini varsayar. Verileri bekleyen aktif işlemciler zaman ve para harcar. Elbette, sistemde her işlemci çekirdeğine kod ve veri besleyen çoklu depolama teknolojisi seviyeleri vardır. Genellikle her çekirdek, çekirdek hızında çalışan yerel önbellek içerir. Bir çipte birden çok çekirdek ikinci düzey ve bazen de üçüncü düzey bir önbellek paylaşır. Ve DRAM önbellekleri besler. DRAM ve önbellek erişim süreleri, veri aktarım hızıyla birlikte işlemcinin performansını eşleştirecek şekilde ölçeklenmiştir.

Sorun, erişim süresi ve veri hızı açısından DRAM ve HDD arasındaki performans boşluğu. Disk / sürücü satıcıları, daha yüksek kapasiteli, daha düşük-Gbyte diskleri / diskleri tasarlamak ve üretmek için harika bir iş çıkardılar; Ancak sürücüler, ne kadar hızlı bir şekilde veriye erişebilecekleri konusunda sınırlamalara sahiptir ve bu verileri DRAM'e ne kadar hızlı aktarabilirler.

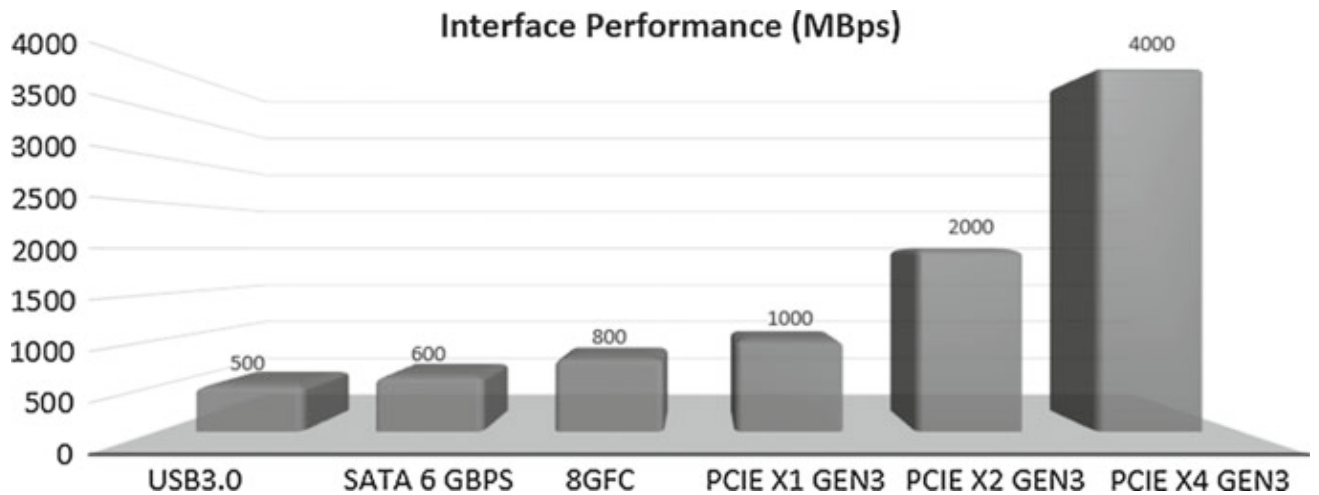
Erişim süresi, bir sabit diskin ne kadar hızlı bir şekilde okuma kafasını bir diskin üzerindeki gerekli veri yolu üzerinde hareket ettirebileceğine ve adreslenen sektörün dönme gecikmesinin kafanın altına hareket etmesine bağlıdır. Maksimum aktarım hızı, diskin dönme hızı ve veri kodlama şeması tarafından belirlenir: birlikte, diskten okunabilecek saniye başına bayt sayısını belirlerler.

Sabit diskler, sıralı verileri okumak ve aktarmak için nispeten iyi performans gösterir. Ancak rastgele arama işlemleri gecikme katıyor ve sıralı okuma işlemleri bile en son işlemcilerin veri iştahı ile eşleşemez.

Bu arada, finansal işlemler ve veri madenciliği (örneğin müşteri ilişkileri yönetimi için başvurular) gibi çevrimiçi işlem gerçekleştiren kurumsal sistemler, verilere oldukça rastgele erişim gerektirir. Ayrıca bulut bilişim, özellikle sanallaştırmaya bakıldığında, tek bir sistemin tek seferde aktif olduğu farklı uygulamaların kapsamını genişleten güçlü rastgele gereksinimlere sahiptir. Her bir mikro saniyelik gecikme süresi doğrudan para, işlemcilerin kullanımı ve sistem gücü ile ilgilidir.

Neyse ki, Flash bellekler DRAM ve HDD arasındaki performans farkını azaltmaya yardımcı olabilir. Flash, DRAM'den daha yavaştır, ancak Gbyte depolama alanı başına daha düşük bir maliyet sunar. Bu maliyet, disk depolamadan daha pahalıdır, ancak işletmeler, Premium'u, Mbyte / s ve rasgele verilere daha hızlı erişim açısından çok daha iyi bir çıktı sunarak, döner depolamaya kıyasla daha iyi bir IOPS maliyeti sağladığından, prim ödemesini memnuniyetle karşılayacaktır.

Ne eski disk sürücüsü form faktörü ne de arabirim Flash tabanlı depolama için idealdir. SSD üreticileri, disk sürücüleri için geliştirilmiş güç profilini kolayca aşacak şekilde 2,5 inç form faktöründe yeterli Flash cihazı paketleyebilir. Ve Flash, en yeni nesil disk arabirimlerinden bile daha yüksek veri aktarım hızlarını destekleyebilir.



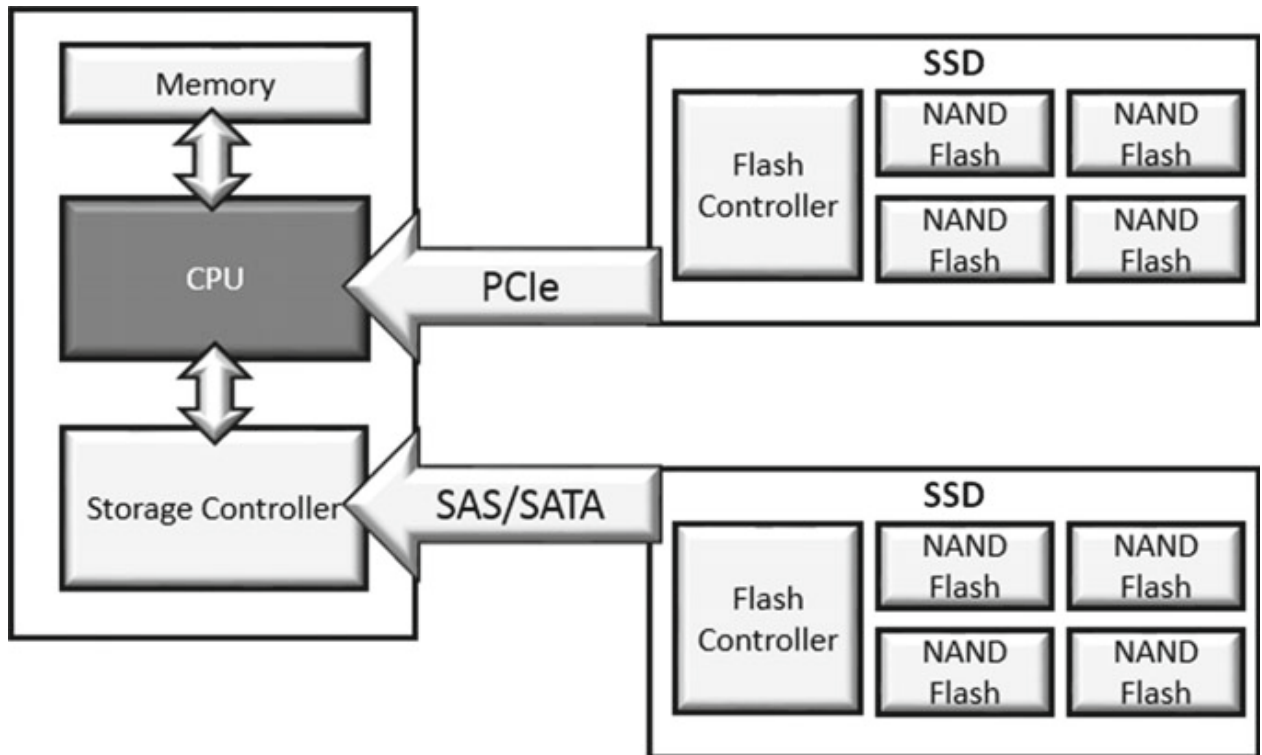
**Fig. 1.11** Arabirim performansı. PCIe, gecikmeyi azaltarak ve işlem hacmini artırarak genel sistem performansını artırır

Disk arayüzlerini performansı (Şekil 1.11). Üçüncü nesil SATA ve SAS 600 Mbyte / s verimini destekler ve bu arabirimlere dayalı sürücüler, kurumsal sistemlerde zaten kullanım bulmuşlardır. Bu veri hızları en hızlı elektromekanik sürücülerini desteklerken, yeni NAND Flash mimarileri ve çoklu kalıplı Flash paketleme, SATA ve SAS arabirimlerinin çıktı kapasitesini aşan toplu Flash bant genişliği sağlar. Kısacası, SSD performans darboğazı depolama ortamından ana bilgisayar arayüzüne kaymıştır. Bu nedenle, birçok uygulamanın Flash depolamadan tam olarak yararlanmak için daha hızlı bir ana bilgisayar bağlantısı gerekir.



PCIe ana bilgisayar arayüzü, bu depolama performansı darboğazının üstesinden gelebilir ve SSD'yi doğrudan PCIe ana bilgisayar veriyoluna bağlayarak benzersiz bir performans sunar. Örneğin, 4 şeritli (x4) PCIe Generation 3 (Gen3) bağlantısı 4 GB / sn veri hızı sağlayabilir. Basitçe söylemek gerekirse, PCIe istenen depolama bant genişliğini karşılar. Ayrıca, doğrudan PCIe bağlantısı sistem gücünü azaltabilir ve eski depolama altyapısına atfedilen gecikmeyi kesebilir.

Açıkçası, PCIe gibi bir arabirim çok kanallı bir Flash depolama alt sisteminin bant genişliğini ele alabilir ve ek performans avantajları sunabilir. Disk arabirimi kullanan SSD'ler, disk I / O işlemlerini gerçekleştiren bir depolama denetleyicisi IC tarafından eklenmiş gecikme de yaşar. PCIe cihazları doğrudan ana veri yoluna bağlanır, böylece eski depolama altyapısı ile ilişkili mimari katman ortadan kaldırılır. PCIe SSD'lerin zorlayıcı performansı, sistem üreticilerinin PCIe diskleri sunuculara ve depolama dizilerine yerleştirerek, IOPS başına maliyeti artırırken uygulamaları hızlandıran katmanlı depolama sistemleri oluşturmasıyla sonuçlandı.



**Fig. 1.12** PCIe SSD vs SAS/SATA SSD

PCIe'yi bir depolama ara bağlantısı olarak kullanmanın yararları açıktır. SATA veya SAS'a kıyasla 6 kattan fazla veri çıkışı elde edebilirsiniz. SATA ve SAS arayüzlerinde ana bilgisayar veri yolu adaptörleri ve SerDes IC'leri gibi bileşenleri ortadan kaldırabilirsiniz; sistem düzeyinde para ve güç tasarrufu sağlar ve Şekil 1.12'de gösterildiği gibi PCIe depolamayı ana CPU'ya yaklaştırır ve gecikmeyi azaltır.

Gecikme, IOPS, bant genişliği, güç, arabirim hızı, kanal sayısı, NAND türü (SLC, MLC, TLC, QLC), SSD tasarımcılarının hedef özelliklerini en düşük maliyetle karşılamak için dikkate almaları gereken parametrelerdir. Gelecekte, ortaya çıkan bellekler de oyunun bir parçası olacak. Bu değişkenler göz önünde bulundurulduğunda, özellikle prototip yapımına dayanan basit bir yaklaşımın, özellikle pazara çıkış zamanına bakıldığında, takip edilmesi zordur. Bu nedenle, SSD simülatörleri olması zorunludur.

## 10. Kaynakça

1. G. Campardo, R. Micheloni, D. Novosel, "VLSI-Design of Non-Volatile Memories", SpringerVerlag, 2005.
2. <https://tr.scribd.com/document/264622256/SSD-ARCHITECTURE>
3. <https://mytxt.xyz/solid-state-drives-ssds-modeling-simulation-tools-strategies/>
4. <https://www.cactus-tech.com/files/cactus-tech.com/documents/ebooks/Solid%20State%20Drives%20101%20EBook.pdf>
5. <https://www.enpedi.com/2012/08/ssd-solid-state-drive-hakknda-hersey-ii.html>
6. <http://www.ssd labs.net/ssd-rehberi/ince-ayrintilari-ile-ssd-arayuzleri/>
7. <https://www.extremetech.com/extreme/210492-extremetech-explains-how-do-ssds-work>
8. <https://tr.scribd.com/document/264622256/SSD-ARCHITECTURE#logout>
9. [https://books.google.com.tr/books?id=vaq11vKwo\\_kC&printsec=frontcover&source=gbs\\_atb&redir\\_esc=y#v=onepage&q&f=false](https://books.google.com.tr/books?id=vaq11vKwo_kC&printsec=frontcover&source=gbs_atb&redir_esc=y#v=onepage&q&f=false)
10. <http://www.wiki-zero.co/index.php?q=aHR0cHM6Ly9lbi53aWtpcGVkaWEub3JnL3dpa2k vU29saWQtc3RhdGVfZHIJpdmU>
11. <http://codecapsule.com/2014/02/12/coding-for-ssds-part-2-architecture-of-an-ssd-and-benchmarking/>