

## Hacettepe University Computer Engineering Internship Report



**NAME – SURNAME :** *Mehmet Taha USTA*

**CLASS:** *3rd Grade*

**NUMBER:** *21527472*

**SUBJECT OF INTERNSHIP:** *Software-Hardware*

**START - END DATE:** *02/07/2018-10/08/2018*

**DURATION OF INTERNSHIP:** *6 weeks/30 work day*

**COMPANY NAME/ ADRESS:** University District Hacettepe University  
Beytepe Campus, Sapphire Blocks F Block KAT 11 NO 13, 06800 ÇANKAYA  
/ Ankara

<b>Table Of Contents</b>	<b>Page</b>
<b>1. INTRODUCTION.....</b>	<b>3</b>
<b>2. CORPORATION.....</b>	<b>3</b>
<b>2.1. Company Name.....</b>	<b>3</b>
<b>2.2. Company Location.....</b>	<b>3</b>
<b>2.3. Short History of Company.....</b>	<b>3</b>
<b>2.4. Main Area of Company.....</b>	<b>3</b>
<b>3. INTERNSHIP PROCESS.....</b>	<b>4</b>
<b>3.1. Introduction of Solid State Drive .....</b>	<b>4</b>
<b>3.2. Solid State Drive Architecture .....</b>	<b>4</b>
<b>3.3. The Basic NAND Flash Cell.....</b>	<b>5</b>
<b>3.4. Flash Controller .....</b>	<b>6</b>
<b>3.5. SSD Controller Architecture .....</b>	<b>6</b>
<b>3.6. SSD's Interfaces .....</b>	<b>8</b>
<b>3.7. SAS and SATA .....</b>	<b>8</b>
<b>3.8. PCI-Express .....</b>	<b>9</b>
<b>4.CONCLUSION.....</b>	<b>10</b>
<b>5.REFERENCES.....</b>	<b>11</b>

## **1.INTRODUCTION**

This internship is a software and hardware internship. Main focuses of internship are search What is the SSD(Solid State Drive) , NAND Architecture, SSD Controller Architecture and Functions,SMT Machine after report them . Generate reports after searching the main focus topics. At the end of internship, SSD's Architecture, The Basic NAND Flash Cell, NAND Architecture, Flash Controller, SSD Controller Architecture, SSD Controller Functions, SSD's Interfaces were understood.

## **2. CORPORATION**

### **2.1. Company Name**

SinerjiSoft Bilisim Sistemleri Yazilim Bilgisayar Donanimlari Elektronik Sanayi ve Ticaret A.S

### **2.2. Company Location**

University District Hacettepe University Beytepe Campus, Sapphire Blocks F Block KAT 11 NO 13, 06800 ÇANKAYA / Ankara

E-Mail: [info@sinerjisoft.com.tr](mailto:info@sinerjisoft.com.tr)

Phone: +90 312 484 38 99

Fax: +90 312 227 04 07

### **2.3. Short History of Company**

Sinerjisoft Bilisim Sistemleri Yazilim Bilgisayar Donanimlari Elektronik Sanayi ve Ticaret A.S. continues to operate since 2013. This company within the scope of institutional quality policy; Smart Web and Developing Mobile Applications, Informatics Consultant, Technical Care, Web Software, provides Electronic Security and Infrastructure Services at international standards With our well experienced team and solution partners, reliable, of high quality, professional production – This company provide support and service.

### **2.4. Main Area of Company**

#### **2.4.a. Web Applications**

Web applications that The company developed presents high level user interface interactions and performance.

#### **2.4.b. Mobile Applications**

In developing mobile software industry, The company develops software for Android, IOS, Windows Phone and Mozilla OS devices with The company professional solutions.

#### **2.4.c. Process Modelling**

With implementation of business process analysis and modelling of your Organization, The company provide restoration and increase in performance in your process.

#### **2.4.d. e-Commerce Software**

The company provides you to put up for sale your products at your own e-commerce site, and The company helps you to profit.

#### **2.4.e. Security Systems**

The company is providing electronic systems and security systems that institutions needed to their service by preparing projects.

#### **2.5.f. Event Management Systems**

The company develops software products that following processes. For example, National and International Olympics, events etc.

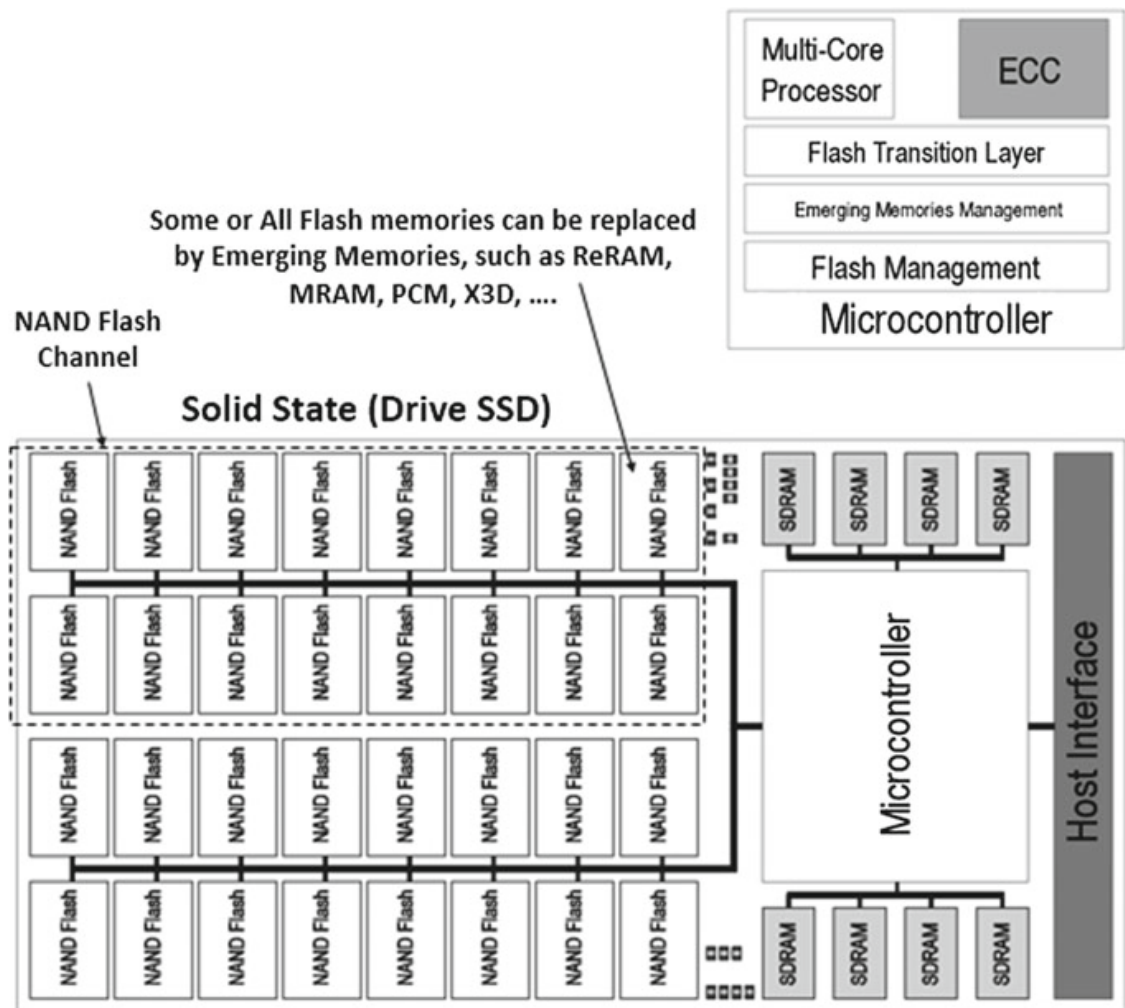
### **3. INTERNSHIP PROCESS**

#### **3.1. Introduction of Solid State Drive**

Solid-state drives (SSDs) are unanimously considered the enabling factor for bringing enterprise storage performances to the next level. Indeed, the rotating storage technology of Hard Disk Drives (HDDs) can't achieve the access-time required by applications where response time is the critical factor. On the contrary, SSDs are based on solid state memories, namely NAND Flash memories: in this case, there aren't any mechanical parts and random access to stored data can be much faster. In many applications though, the interface between host processors and drives remains the performance bottleneck. This is why SSD's interface has evolved from legacy storage interfaces, such as SAS and SATA, to PCIe, which enables a direct connection of the SSD to the host processor.

#### **3.2. Solid State Drive Architecture**

A basic block diagram of a solid state drive is shown in Fig. 1.1. In addition to memories and a Flash controller, there are usually other components. For instance, an external DC-DC converter can be added in order to drive the internal power supply, or a quartz can be used for a better clock precision. Of course, reasonable filter capacitors are inserted for stabilizing the power supply. It is also very common to have an array of temperature sensors for power management reasons. For data caching, a fast DDR memory is frequently used: during a write access, the cache is used for storing data before their transfer to the Flash. The benefit is that data updating, e.g. of routing tables, is faster and does not wear out the Flash.

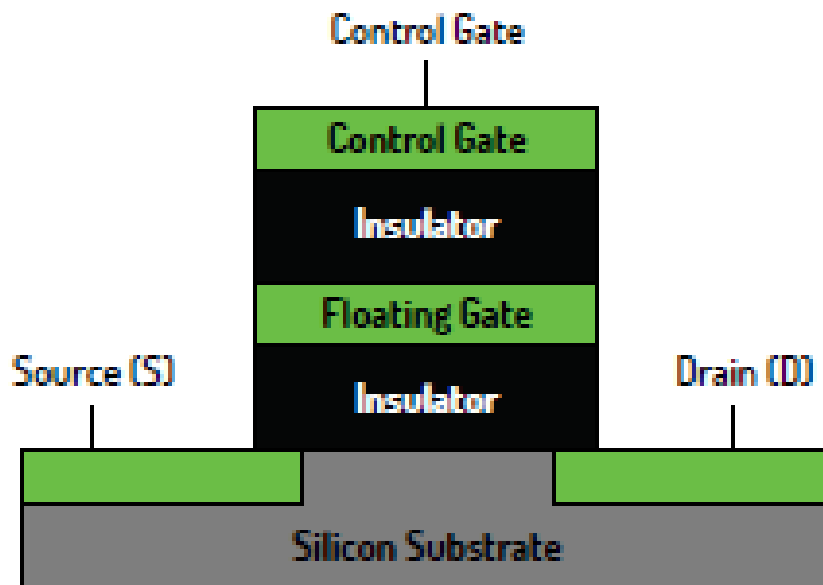


**Fig. 1.1** Block diagram of an SSD

### 3.3. The Basic NAND Flash Cell

In order to store a single bit of data on a solid state drive, we need the smallest building block - a single NAND flash cell. The simplest NAND cell can be set to either a 0 or 1 state. It will continue to store that state even after power has been removed.

### What does a NAND Cell look like?



A simple NAND Flash Cell diagram is shown above. The NAND flash cell is made from a floating gate transistor. Electrical charge is stored on the floating gate which is isolated above and below by oxide insulating layers. In its simplest form when the floating gate is charged, it is programmed and recognized as a binary 0. When the floating gate has no charge it is erased and recognized as a binary value of 1.

### 3.4. Flash Controller

A memory controller has two fundamental tasks:

1. provide the most suitable interface and protocol towards both the host and the Flash memories;
2. efficiently handle data, maximizing transfer speed, data integrity and retention of the stored information.

In order to carry out such tasks, an application specific device is designed, embedding a standard processor—usually 8–16 bits—together with dedicated hardware to handle time-critical tasks.

### 3.5. - SSD Controller Architecture

#### 3.5.a. Host Interface

The host interface of a controller is typically designed to one industry standard interface specification. There are several interfaces made to address different system and design requirements. The most popular are SATA, SD, USB, PATA/IDE and PCIe.

### **3.5.b. SMART (Self-Monitoring, Analysis and Reporting Technology)**

The SMART function, available in some controllers, monitors and records data regarding many attributes of the SSD and memory. An example is the ability to monitor the percentage of endurance cycles remaining in the SSD since this is a key determining factor of the life remaining.

### **3.5.c. Wear Leveling**

Wear Leveling is the ability to even out the number of write cycles throughout the available NAND. Since each NAND block has a limited number of erase/write cycles, if only one physical block is written continuously, it will quickly deplete its endurance cycles. A controller's Wear Leveling algorithm monitors and spreads out the writes to different physical NAND blocks.

### **3.5.d. Read & Program Disturb**

With the finer and finer trace widths of the NAND flash, more issues arise to maintain the data contents of the NAND cells. Read & Program Disturb occur when cells are read or written causing cross coupling to adjacent cells and occasionally changing their values. Controllers need algorithms and in some cases circuitry to compensate for this phenomena.

### **3.5.e. Encrypt & Decrypt Engine**

For higher security applications, a hardware encryption and decryption engine is generally built into the silicon of the controller. The encryption engine is typically implemented in hardware to ensure speed for encrypting/decrypting on the fly. The most popular encryption method for SSDs today is AES256.

### **3.5.f. Buffer/Cache**

Controllers generally have a high speed SRAM/DRAM cache buffer used for buffering the read and/or write data of the SSD. Since this cache uses volatile memory, it subjects data to loss if power is unexpectedly removed. It is common to see both internal caches in the controller chip itself as well as external RAM cache chips.

### **3.5.g. CPU/RISC Processor**

The heart of every SSD is the main processing core. This can be a CPU or RISC processor. The size and performance of the CPU/RISC processor determines how capable the controller can be.

### **3.5.h. ECC Engine**

Error Checking & Correction are a key part of today's SSD. ECC will correct up to a certain number of bits per block of data. Without ECC, many of the low cost consumer flash cards using very inexpensive memory would not be possible.

### **3.5.i. Write Abort**

Write Abort is the when power to the SSD is lost during a write to the NAND flash. Without a battery or SuperCap backed cache, it is likely this data in transit will be lost. The more important aspect of this is to ensure the SSD's internal metadata and firmware remain uncorrupted. This is the function of Write Abort circuitry mainly found in Industrial Grade products.

### **3.5.a. Miscellaneous I/O**

Simple functions such as chip select pins for the NAND components are handled with several input/ output pins. There are also a number IO functions required for initial programming and production.

## **3.6. SSD's Interfaces**

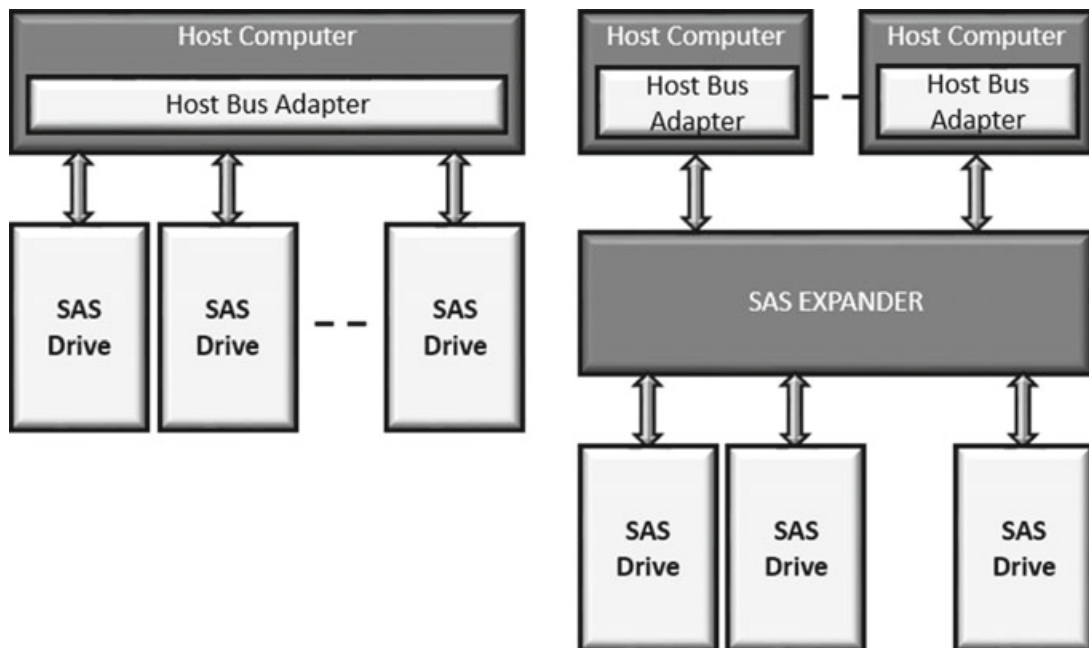
There are 3 main interface protocols used to connect SSDs into server and/or storage infrastructure: *Serial Attached SCSI (SAS)*, *Serial ATA (SATA)* and *PCI-Express*. *PCI-Express* based SSDs deliver the highest performances and are mainly used in server based deployments as a plug-in card inside the server itself. *SAS* SSDs deliver pretty good level of performances and are used in both high-end servers and midrange and high-end storage enclosures. *SATA* based SSDs are used mainly in client applications and in entry-level and mid-range server and storage enclosures.

## **3.7. SAS and SATA**

*Serial Attached SCSI (SAS)* is a communication protocol traditionally used to move data between storage devices and host. *SAS* is based on a serial point-to-point physical connection. It uses a standard *SCSI* command set to drive device communications. Today, *SAS* based devices most commonly run at 6 Gbps, but 12 Gbps *SAS* are available too. On the other side, *SAS* interface can also be run at slower speeds—1.5 Gbps and/or 3 Gbps to support legacy systems.

*Serial ATA(SATA* or *Serial Advanced Technology Attachment)* is another interface protocol used for connecting host bus adapters to mass storage devices, such as hard disk drives and solid state drives. *Serial ATA* was designed to replace the older parallel *ATA/IDE* protocol. *SATA* is also based on a point-to-point connection. It uses *ATA* and *ATAPI* command sets to drive device communications. Today, *SATA* based devices run either at 3 or 6 Gbps.





Similarities between SAS and SATA are:

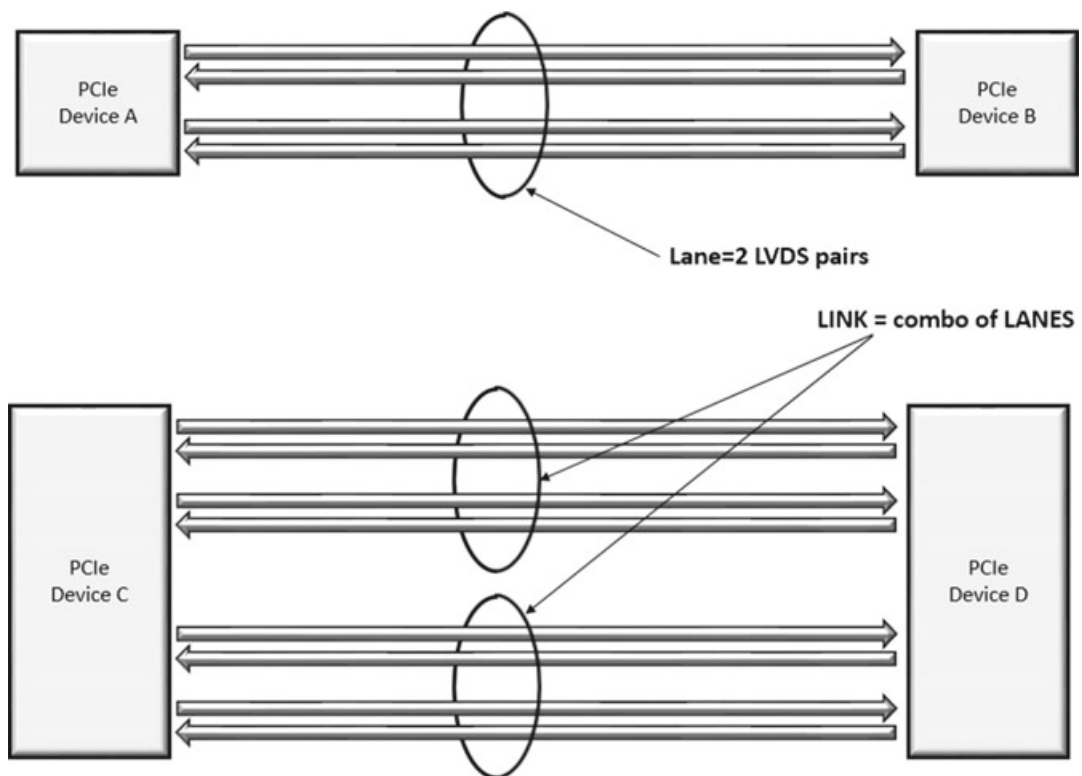
- Both types plug into the SAS backplane;
- The drives are interchangeable within a SAS drive bay module;
- Both are long proven technologies, with worldwide acceptance;
- Both employ point-to-point architecture;
- Both are hot pluggable.

Differences between SAS and SATA are:

- SATA devices are cheaper;
- SATA devices use the ATA command set, SAS the SCSI command set;
- SAS drives have dual port capability and lower latencies;
- While both types plug into the SAS backplane, a SATA backplane cannot accommodate SAS drives;
- SAS drives are tested against much more rigid specifications;
- SAS drives are faster and offer additional features, like variable sector size, LED indicators, dual port, and data integrity;
- SAS supports link aggregation (wide port).

### 3.8. PCI-Express

PCI-Express (*Peripheral Component Interconnect Express*) or PCIe is a bus standard that replaced PCI and PCI-X. PCI-SIG (*PCI Special Interest Group*) creates and maintains the PCIe specification.



**Fig. 1.2** PCI Express lane and link. In Gen2, 1 lane runs at 5Gbps/direction; a 2-lane link runs at 10 Gbps/direction

Unlike the older PCI bus topology, which uses shared parallel bus architecture, PCIe is based on point-to-point topology, with separate serial links connecting every device to the root complex (host). Additionally, a PCIe link supports full-duplex communication between two endpoints. Data can flow upstream (UP) and downstream (DP) simultaneously. Each pair of these dedicated unidirectional serial point-to-point connections is called a *lane*, as depicted in Fig. 1.2.

#### 4. CONCLUSION

This internship basically gave me idea about software development and business industry. After things i learned, i have gained vision about my future plans.

I learned so many things about Solid State Drives (SSDs). Topics that i became familiar:

- SSD's Architecture
- The Basic NAND Flash Cell
- NAND Architecture
- Flash Controller
- SSD Controller Architecture and Functions
- SSD's Interfaces

Also i learned how to program with "C#" and making projects on "Visual Studio". I learned some basics about "AForge.NET" framework and how to make new applications on Windows Form Application.

Besides all of these topics, i have seen creating program process, teamwork importance, debugging and what developers environment really look like.

## 5. REFERENCES

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=aHR0cHM6Ly9lbi53aWtpcGVkaWEub3JnL3dpa2kvU29saWQtc3RhdGVfZHJpd mU>
11. <http://codecapsule.com/2014/02/12/coding-for-ssds-part-2-architecture-of-an-ssd-and-benchmarking/>
12. <http://sinerjisoft.com.tr>