Linnaeus University

Sweden

Degree Project

The differences between SSD and HDD technology regarding forensic investigations

Author: Florian Geier
Supervisor: Ola Flygt
Examiner: Johan Hagelbäck
Semester: VT 2015
Subject: Computer Science

Abstract

In the past years solid state disks have developed drastically and are now gaining increased popularity compared to conventional hard drives. While hard disk drives work predictable, transparent SSD routines work in the background without the user's knowledge.

This work describes the changes to the everyday life for forensic specialists; a forensic investigation includes data recovery and the gathering of a digital image of each acquired memory that provides proof of integrity through a checksum. Due to the internal routines, which cannot be stopped, checksums are falsified. Therefore the images cannot prove integrity of evidence anymore. The report proves the inconsistence of checksums of SSD and shows the differences in data recovery through high recovery rates on hard disk drives while SSD drives scored no recovery or very poor rates.

Preface

As a computer science student I specialized in network security and digital forensics and am always interested in the newest technology. I came across the video of Scott Moulton and his speech at DEFCON in Las Vegas, “Solid State Drives Destroy Forensic & Data Recovery Jobs” which sparked my interest in the topic SSD drives and data recovery. It surprised me that there was not much documentation and even less test cases to be found when I first researched the problem which led me to the idea of conducting tests myself. This works aim is to fill this gap and to encourage further testing and research.

Table of Contents

1. Introduction

This section will describe the problems tackled in this report, as well as the necessary background and definitions to understand the structure and extent of this report. A formal problem description will be formulated and presented, along with the limitations of this report.

1.1. Background

Digital memory has been revolutionized over the past ten years, in addition to the known hard disk drive a new technology, flash memory, has emerged and is rapidly gaining market-shares towards the hard disk drive. Flash memory introduced dramatic changes to the principles of computer forensics. Forensic acquisition of computers equipped with flash memory storage is very different from how we used to acquire PCs using traditional hard drives. Instead of predictable and the high likelihood of possible recovery of information, we can no longer assume if and how much data can be recovered.

1.2. Problem discussion

Flash memory has recently become more and more popular. Faster data rates, decreasing prices and higher resistance to shocks are the factors encouraging most buyers. However, when it comes to transparency, data recovery and forensics, flash memory shows significant disadvantages. This can have a major effect on the acquisition of forensic data and can affect how the legal system uses, and gets evidence to hold in court.

1.3. Purpose

The purpose of this report is to show in detail the differences between the two technologies and how they behave after a file has been deleted or the disk is reformatted on purpose or by accident. This report will show through theory and test cases the differences between the two technologies and how these affect the work of a forensic examiner and how and if evidence can still hold in court. This report aims to create awareness of the problematic and inspire further research and the agreement to standards and guidelines for manufacturers and forensic examiners.

1.4. Previous research

Until recent research has been conducted the topic was more or less unknown to both end-users as experts. Not much material could be found online or in books and articles. Scott Moulton's speech "Solid State Drives Destroy Forensics & Data Recovery Jobs," in Las Vegas 2011 drew my attention to the topic as he was one of the first mentioning the problems caused by the new SSD technology [1]. Graeme B Bell and Richard Boddington's work *"Solid State Drives: The Beginning Of The End For Current Practice In Digital Forensic Recovery?"* conducted a research including tests on the topic and are a great starting point for further research [2]. Eoghan Casey's book "Digital Evidence and Computer Crime" provides a solid background about forensic investigations, their procedures and guidelines especially on hard disk drives.

1.5. Research questions

The main research question has been formulated as

RQ1: *The differences between the two technologies, how these affect the work of a forensic examiner and how and if evidence can still hold in court*.

The research question has been divided into sub questions in order to aid in answering them by focusing on specific aspects at a time.

RQ1.1: Is data persistent after deletion on flash memory in the same way as on traditional hard disk drives?

RQ1.2: What is an acceptable method for forensic data acquisition on flash memory?

RQ1.3: What difference makes the TRIM functionality on SSD drives to an acquisition process?

RQ1.4: Does an idle time between deletion and acquisition affect the recovery process?

RQ1.5: Does formatting a medium in comparison to deleting all data affect the acquisition process?

1.6. Hypotheses

The following hypotheses have been derived from the defined research questions.

H1: Data is not or only partially persistent after deletion on flash memory in comparison to traditional hard disk drives.

H2: An acceptable method for forensic data acquisition on flash memory does not exist yet.

H3: The TRIM functionality on SSD drives is expected to be responsible for data loss.

H4: Idle time between deletion and acquisition is expected to influence the result of a recovery process.

H5: Formatting a medium is expected to influence the result of a recovery process.

1.7. Methodology

This chapter describes the methods used to conquer the questions this report is trying to investigate and answer.

The report will consist of theoretical reviews to cover the empirical investigation to address all differences between the two technologies. An in depth research will be conducted using academic publications, books and on-line resources. Secondly testing will be conducted to prove the results gained by the documentary analysis. Hereby differences in architecture between the investigated technologies will be proven. Furthermore the report will show what problems these differences cause by performing test cases simulating real world forensic investigations and data recovery techniques. These tests will use software known and used by forensic investigators (see chapter 2.3.7) and will help investigating workarounds to the problems found. In addition to known software a series of Java programs have been written to perform tests on the different hardware.

It is important to understand that the conducted tests are only there to describe architectural and software based differences rather than resulting in numerical data.

1.7.1. TRIM

The TRIM functionality erases blocks that have been marked as *to be deleted* by the Operating system. The function has a negative effect on forensic analysis and data persistence after deletion cannot be guaranteed anymore because the memory controller of the SSD decides when and how much of the marked blocks to delete. The test cases designed for the TRIM functionality (3.9) log if and at what time certain blocks are physically deleted after the operating system marked all files *to be deleted* with enabled and disabled TRIM functionality.

1.7.2. Garbage collection

The Garbage collection routine works closely together with the TRIM functionality. It keeps track of the *to be deleted* cells and can combine leftover data of different cells to empty ones in order to delete others. This fully works in the background and can only be suspected to work along with TRIM so the same test cases (3.9) are related here.

1.7.3. Erasing patterns

Different SSDs are expected to show different behaviour when deleting data. They are expected to not delete all blocks at a time but a subset of them. Test case 3.5 will show these different patterns.

1.7.4. Wear leveling

Because each cell within a Flash chip has a limited number of write cycles, and usually not all information stored within one device changes with the same frequency, to outrun the wearing out of cells wear leveling tries to even out the wear across the medium (see Chapter 2.2.9). Wear leveling is a function that works fully in the background and therefore cannot be detected by these test cases. This is because the hardware addresses of the cells are not visible and accessible directly by the operating system.

1.8. Outline of the report

Following this introductory section, the report encompasses three major parts.

Chapter 2, the Literature review, will cover the architecture and functionalities of different memory technologies as well as an insight to digital evidence and forensics in order to familiarize the reader with the topic and the problem.

Chapter 3 Testing will show the theory part in practice and investigate how we can prove the existence of different implementations of TRIM, Wear leveling and Garbage collection in the different flash memory applications. Further on this chapter will show how these implementations affect the data-recovery rate on different technologies.

Chapter 4 Discussion will show and discuss the results of the testing in chapter 3 as well as workarounds for the found problems.

Finally, as a final wrap up, chapter 5 Conclusion will be the concluding part, repeating the most important facts from other sections, along with ideas to improve this work in the future.

1.9. Scope and limitations

The focus of this report will be on the architecture and functions of hard drives and different flash memory applications. Hereby SD flash memory cards, USB flash memory drives and solid state disks will be investigated and other technologies and applications are not relevant for this investigation. Further, different test-cases will prove the theory part and display the different implementations of wear leveling and garbage collection in the different flash memory applications and how these implementations affect a recovery process. In depth testing with hardware form many different vendors to show vendor specific variations of implementations cannot be conducted as well as any tests analysing parts of hardware like spindles or chips.

1.10. Ethics and social impacts

Every academic work, published and circulated in a society, has an impact on it. It is therefore necessary for the author to consider the impact in advance and encourage the ethically correct use of the work within the society. Within the field of Computer Science the cornerstone is security consciousness, which contains data integrity and authorisation. Therefore forensic examiners are all confronted with ethical dilemmas because of their privileged access to sensible data. Examiners could be exposed to trade secrets, threats to national security or private information for which or which deletion/alteration third parties may pay handsomely for. The ethical judgment of an examiner can determine the outcome of legal cases and whatever consequences.

2. **Literature review**

The literature review section will provide the necessary background needed to understand the problem this report is trying to answer. It will give insight to the different architectures of the investigated technologies and the basics of digital evidence and digital forensics.

2.1. Hard disk drives throughout history

With the invention of the first computers, IBM released the first computer hard disk drive in 1956. Magnetic hard disk drives became the most used storage device built in computers. The first ever hard disk drive was built in cylindrical form and weighed more than one ton. The IBM Model 350 (Figure 2.1) was as big as a refrigerator and saved up to 5 million digits, approximately 5 Megabytes.

The Hard drive consisted of 50 vertically stacked disks covered in magnetic paint, spinning at speeds of 1,200 rpm. A mechanical arm would move in-between the disks and read or write data on a specific spot. This was achieved by changing the magnetic polarisation on the specific spot [3].

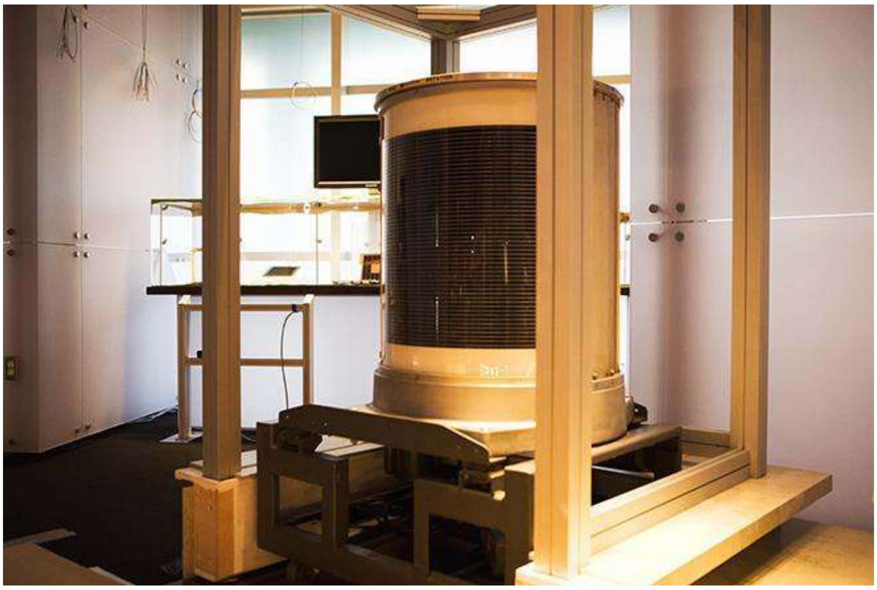


Figure 2.1 The IBM Model 350 [3]

The technology used in the IBM Model 350 is still used in hard disk drives manufactured today. However the form factor was standardized in the early 1980s to a

3.5 inch desktop and 2.5 inch notebook-class drives. Usually today’s desktop class drives spin with a speed of 7,200 rpm and notebook class drives with 5,400 rpm. Today’s 3.5-inch HDDs store up to 6 Terabytes, while 2.5-inch drives up to 2 Terabytes. The internal cable interface has changed from Serial to IDE (Integrated Drive Electronics) to SCSI (Small Computer System Interface) and finally to SATA (Serial ATA) over the years. For the user this meant only performance improvements since each new cable interface operates with higher bitrates per second.

Today transfer rates up to 1,030 Megabits per second are possible while the IBM Model 350 was only able to fetch 100,000 bits per second (0,01 Mb/s) [3].

Today’s hard drives consist of non-moving parts. Flash memory chips store the data instead of magnetic disks, which brings advantages in data rate, energy consumption and shock resistance. While hard disk drives were sensitive to shocks due to the mechanical parts solid state disks are more shock resistant and therefore more suitable for portable devices. Due to the lack of moving parts also less energy is needed to operate SSDs. This is one of the main reasons today’s portable computer have built in SSDs instead of HDDs. The battery life is much longer, and the devices faster and shock resistant.

High production costs slowed down the adoption of SSD’s but world shipments for SSDs are predicted to rise at least 600% between 2012 and 2017, as stated by market researchers [4]. Figure 2.2 shows that by 2015 the shipment of SSDs is predicted to make up over a third of the global shipments of computer storage devices.

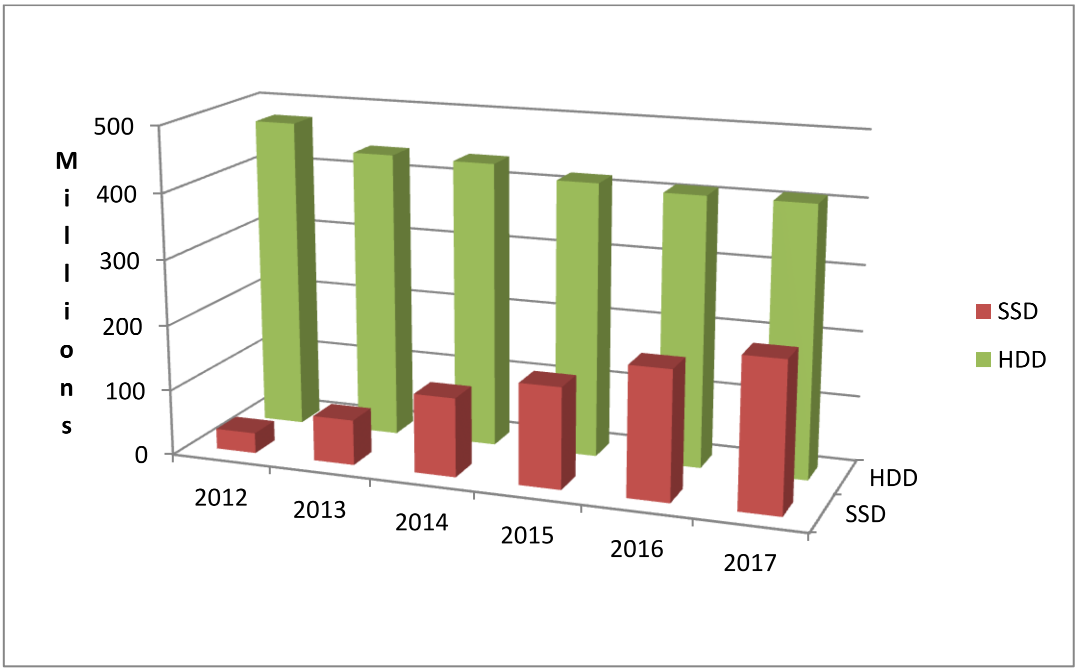


Figure 2.2 Worldwide shipments for HDDs and SSDs, [4]

2.2. The architecture of flash and hard disk drives:

There is a big difference in the architecture between the two technologies flash memory and hard disks. While hard disks save data on spinning disks in form of magnetized areas a flash memory does not consist of any moving parts, which brings multiple advantages in energy consumptions, read and write speeds and robustness.

2.2.1. The architecture of hard disc drives

Conventional hard disk drives store data on spinning disks made of aluminium or glass, covered with a thin magnetic material. These disks spin due to a motor that is mounted on a shaft through a hole in the centre of the disk and depending on the application the speed varies between 6,000 and 10,000 revolutions per minute. In desktop computers speeds of 7,200 rpm are standard while in high performance applications 10,000 rpm is more common. Different vendors use different amounts of these disks on top of each other to multiply the storage space [5].

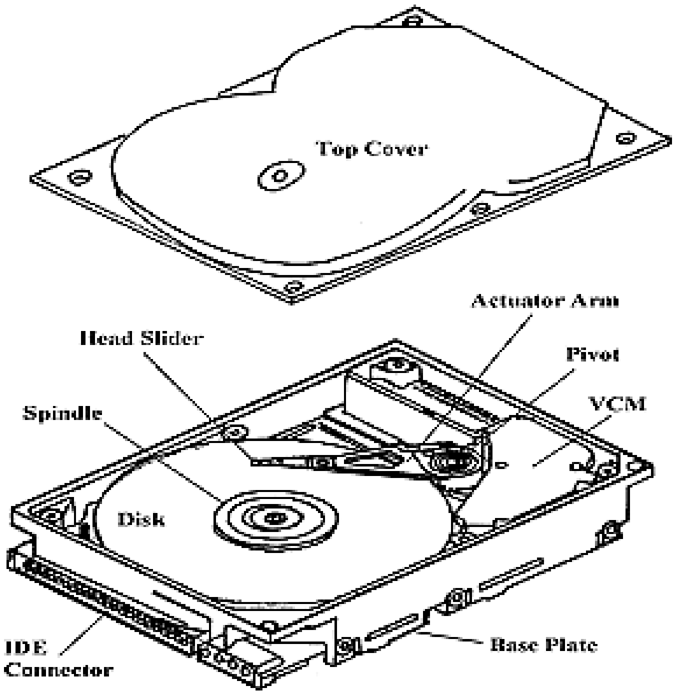


Figure 2.3 Typical components found in HDD [5]

In between these disks the actuator arm or slider moves and on the slider a read and a write head is mounted. The actuator arm brings the heads into close proximity with the magnetized bits so they are flying over the spinning surface. The surface is very smooth in order to provide a uniform read back to the heads. The air in between the head and the surface will make the head float a few nanometers above the surface. This effect only exists while the disks are in motion, otherwise the head will be in contact with the disk.

Hence, to avoid the heads touching the surface of the disks while they are still, two different approaches have been used. Earlier drives used a so called landing zone, a small ring on the disk near the center with an appropriate texture. The arm would drag the head onto this ring before the drive powers off and the disks stop moving. More recent drives use ramps to unload the heads; the arm is moved over a ramp that lifts the heads and brings them to a parking position. Only after the disks begin to spin with a certain speed will the heads move onto the disks. At this speed, as was stated, the head floats above the disk due to its aerodynamic properties.

The write-head, commonly known as the thin film inductive head (TFI head) consists of a thin film coil that gives out a magnetic field when current passes through the coil. The element the coil sits on, known as the core, has a little gap on the bottom. This gap flies over the disk's surface and can change the polarisation of the area on the disk that it passes by changing the polarisation of the current passing through the coil. Figure 2.3 shows the described components in a hard disk drive while Figure 2.4 illustrates the floating write and read head over the magnetized surface.

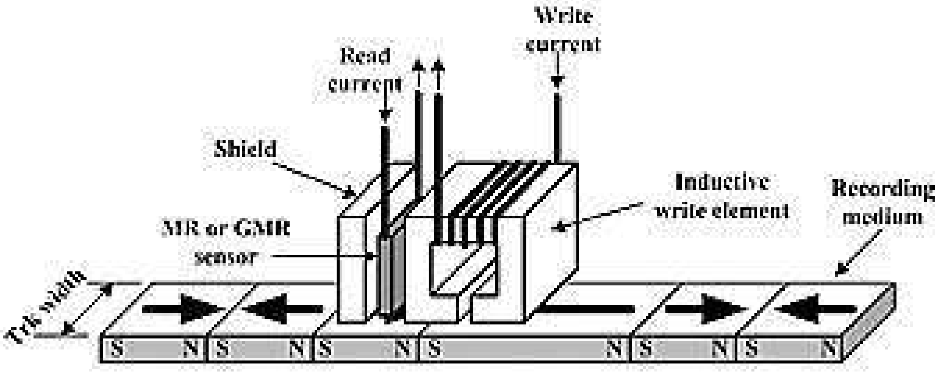


Figure 2.4. Thin film inductive head [5]

The read head works by following the same but reversed principle and also consists of a thin film coil wound around a core that is narrower than the write-head's. The coil uses the magneto-resistive effect that picks up the polarisation of the bit passed over and sends a current, which can then be translated into a zero or one bit. [5].

2.2.2. Arrangement of data on the hard disks

The smallest unit of recorded information on magnetic media is one bit. These bits are arranged in circular forms on tracks around the disk. A typical hard drive disk contains 70,000 to 100,000 tracks on each surface.

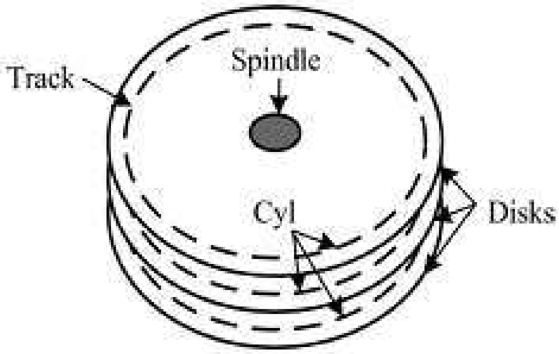


Figure 2.5 Tracks and Cylinders [5]

In order to write on a new track the write head is moved by the arm to the next position on the radius. All data is written in data blocks of 512 bytes which are recorded sequentially along the track. Since a hard drive consists of multiple disks (Figure 2.5) recordable on both surfaces and only one actuator arm, consisting of multiple sliders and heads, a separate head is used for each surface. All heads have the same position on its according surface; the outermost track on any surface is track 0 and so all tracks 0 together are called cylinder 0 (cyl 0). Using cylinder addresses manufacturers could increase access speeds since multiple heads can read simultaneously. Each track is divided into sectors, also called servo sectors. Each sector is typically 512 bytes large and addressed starting from 1 for each track. To identify sectors and tracks, special magnetic patterns are written on the disk during the production [5].

To address a specific sector we can use the CHS, Cylinder-Head-Sector, addressing method. This method allows a sector to be found by the cylinder (starting from 0), the head of the according surface (starting from 0) and the sector number (starting from 1). Recently this addressing method has been replaced by LBA, Logical Block Addressing [5]. Figure 2.6 illustrates the surface of a disk and its arrangement of blocks and sectors.

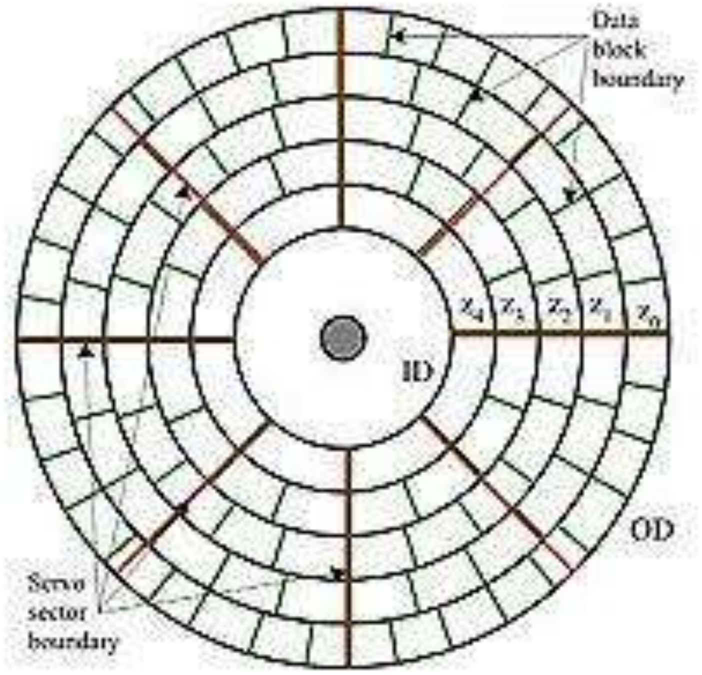


Figure 2.6 Illustration of a disk surface [5]

Before data can be stored on a disk by an operating system the disk must be formatted and a partition must be created. A partition is a logical unit dividing the disk in different logical parts. In the Master Boot Record (MBR), a partition table is stored on the first sector on the disk, telling the operating system how the disk is divided. Operating systems like Linux, Windows or Macintosh lay different file systems over the partitions. While Windows use FAT and NTFS Linux uses EXT2 or EXT3. A file system keeps track of the location on the physical disk that the data is stored. Windows uses the Master File Table (MFT) as an index to the files it stores on hard drives. Contrary to popular belief deleting a partition or reformatting it does not affect the actual data. It simply deletes the file allocation table (FAT), and data can still be recovered [6]. Figure 2.7 shows an example of the disk structure containing two partitions including the partition table and boot sectors.

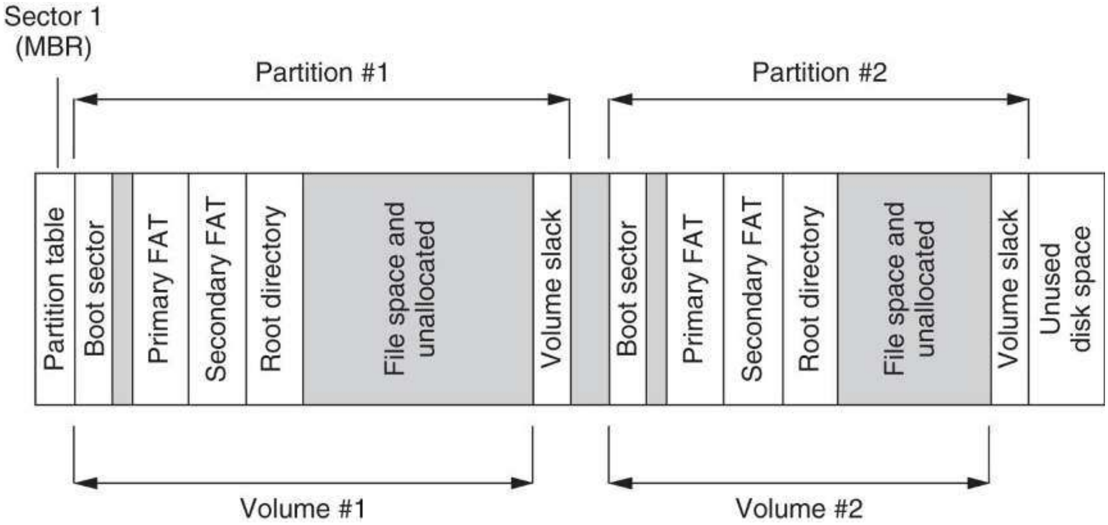


Figure 2.7. Simplified depiction of disk structure [6]

2.2.3. The architecture of flash memory

What makes flash memory faster, more energy efficient and more shock resistant is the lack of moving parts. There are no spinning disks or moving heads reading and writing to a disk. Flash memory devices are complete small systems where every component is soldered to a printed circuit board (PCB). Semiconductor memories (flash memories) can be divided into two major categories: RAM (Random access memory) and ROM (Read only memory). Data on ROM memory can only be written and the information will be stored virtually forever, while RAM memory is rewritable and loses its information as soon as the device loses power. In the 1970's the first non-Volatile memories (NVM) were invented. Stored information on NVMs can be altered but is also preserved after power off. In the early 1990's the first NVMs found application in flash memories used for USB sticks and flash memory cards. Two different types of flash memories exist: NAND and NOR.

2.2.4. NAND flash memory

Flash memories like SD cards, USB drives and SSDs are based on NAND memory; their cells are based on Floating Gate (FG) technology like NOR memory, though NAND chips are smaller and faster they cost about 60% of the price of an equivalent NOR chip to produce. The negative aspect is that not each cell can be written and deleted independently but have to be managed in byte arrays, sectors and blocks, whereas NOR chips handle each cell independently [7].

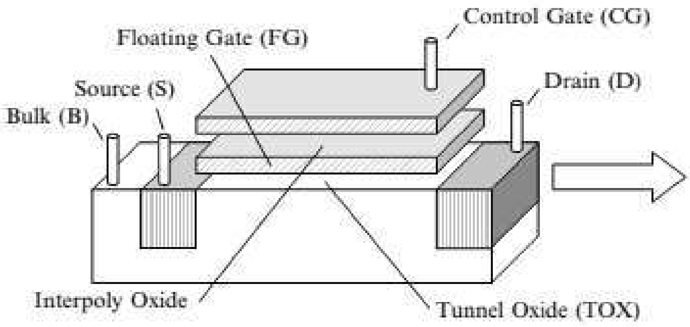
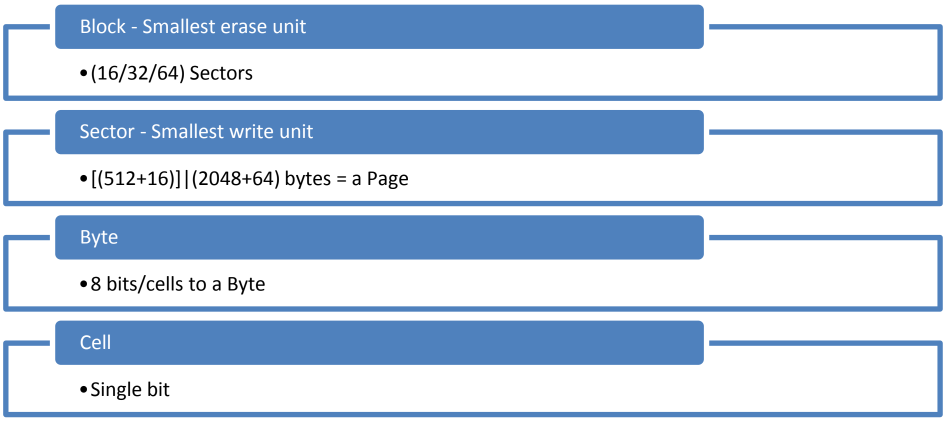


Figure 2.8 Floating gate cell [14]

A NAND cell, as illustrated in Figure 2.8, is built with two overlapping gates, one completely surrounded by oxide and the other forming the gate terminal. If voltage is applied to the control gate, electrons can pass from the source through the dielectrics and settle on the floating gate. Here they are trapped and can stay preserved for decades. This changes the charge of the cell from neutral into negative and is called programming. Only if voltage is applied to the drain the electrons will pass from the floating gate and return the cell to neutral. Each cell contained one bit of information (single-level cell, SLC) until multi-layer cells (MLC) were invented, which contain two or more bits. The cells are connected to arrays as shown in Figure 2.10. An array typically holds 8192 blocks, where a block consists of 64 pages (4000+128 Bytes) (Figure 2.9). On NAND memory, a write operation can be done on page-level, but due to hardware limitations, erase commands always affect entire blocks.



Block - Smallest erase unit

- (16/32/64) Sectors

Sector - Smallest write unit

- [(512+16)]|(2048+64) bytes = a Page

Byte

- 8 bits/cells to a Byte

Cell

- Single bit

Figure 2.9 NAND serial device layout [1]

2.2.5. Memory Controller of a flash memory drive

The memory controller of flash memory has two fundamental tasks; it provides the interface between the disk and the host and handles the data on the disk. The controller hereby translates and keeps track of LBA and physical addresses of the data on the memory. This task is similar to the task of the controller in a HDD. While the controller in HDDs only has small extra functionality, like S.M.A.R.T. (Self-Monitoring, Analysis and Reporting Technology) and bad sector handling, flash controllers have some significant extra features. These functionalities are embedded in the Flash File System (FFS), the file system that enables the use of SSDs like conventional drives. Both these functionalities are completely dependent on the manufacturer. Each manufacturer follows a different approach and no standard has been created yet. The two most important functions are wear leveling and the garbage collection.

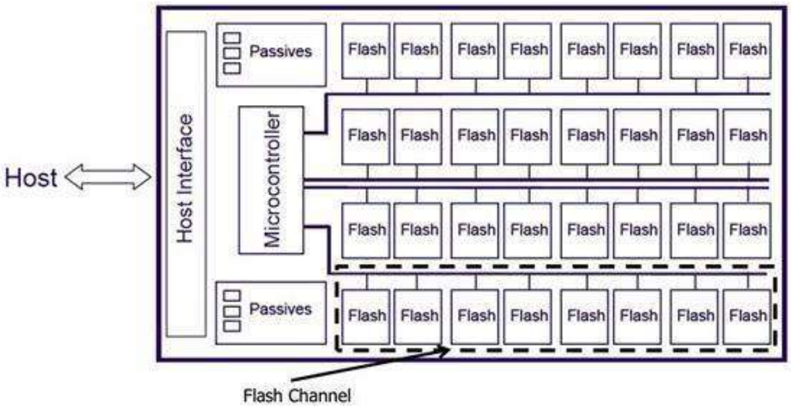


Figure 2.10 Block diagram of a SSD [14]

2.2.6. SSD memory controller

While there exist more than 100 vendors offering SSD drives there are only very few producing SSD controllers [8]. Typically SSD vendors are buying controllers from other companies and combine their own or others NAND memory chips with them. Therefore some vendors have gained a huge market share. The biggest producer of SSD memory controller is SandForce which was an American producer of SSD memory controllers, later bought by LSI, Avago and Seagate in 2014 [9]. Since there are only few companies producing controllers the competition between the manufacturers is really strong. A memory controller’s internal routines, the implementation of wear leveling and garbage collection, compression and encryption is what differentiates one SSDs from another and directly influence directly the drives read and write speeds. This is the reason for a total discretion of the manufacturers about their own approaches to the functionalities of the controllers and the reason no standards have been created yet. Figure 2.11 shows the market shares of the biggest SSD controller manufacturers.

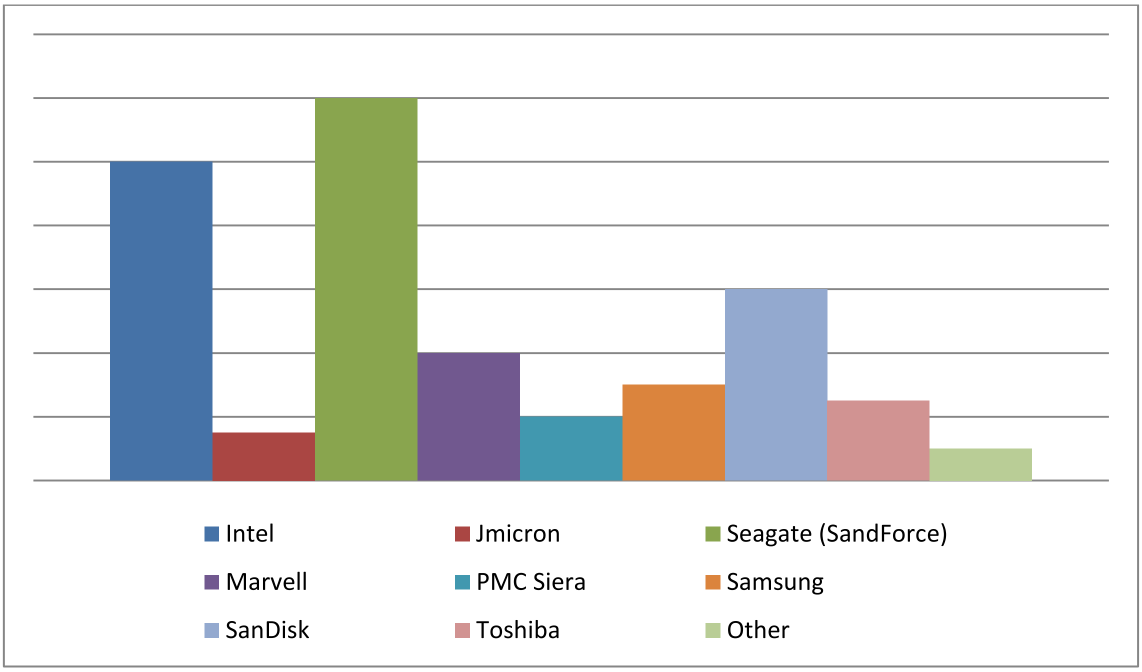


Figure 2.11 SSD controller market share 2014 [10]

2.2.7. SandForce

The biggest player in the SSD memory controller is Seagate based on SandForce technology. SandForce technology has a few advantages compared to other vendor’s solutions like data compression and encryption. These advantages are RAISE Data Protection, Automatic Encryption and DuraWrite, three technologies for improved error correction, security and longer lasting hardware due to less write cycles. Nearly all other vendors store data without encryption on the flash memory and need to use software encryption solutions to secure data stored on the device which creates overhead and slows down the write and read process. Seagate SandForce flash controllers solve this problem by using dual automatic hardware encryption to protect the information it stores on flash and to prevent unauthorized access. This encryption works transparently and independently of the host system [11].

2.2.8. TRIM

An important function of SSDs that does not exist in HDDs is the TRIM command. Trim is an attribute of the ATA Set management command and allows the operating system to inform the SSD of *to be deleted* blocks. It will tell the device what blocks are safe to remove. Using Microsoft Windows 7 or Windows Server 2008 the TRIM command is enabled by default but can be disabled/enabled by the following commands (Code snippet 2.1) in the Windows command prompt [12]. Since the function is enabled by default on operating systems that support TRIM no action is required except for purposely disabling TRIM for test purposes.

```
Enable:
fsutil behavior set disabledeletenotify 0

Disable:
fsutil behavior set disabledeletenotify 1

Check the status of TRIM:
fsutil behavior query disabledeletenotify
Results explained below:
DisableDeleteNotify = 1 (Windows TRIM commands are disabled)
DisableDeleteNotify = 0 (Windows TRIM commands are enabled)
```

Code snippet 2.1 Windows TRIM commands

2.2.9. Wear Leveling

Each NAND cell within a Flash chip has a limited lifespan. It has a limited number of write cycles, typically guaranteed to withstand more than 100 000 cycles. Usually not all information stored within one device changes with the same frequency. Some data gets updated often while other data may not change for a longer time. To outrun the wearing out of some cells and basically leaving others untouched it is important to keep the aging of all cells uniform and to a minimum. Two different approaches are known, dynamic and static wear leveling. Dynamic leveling remaps LBA addresses from the host system to the next free page when the host writes to the drive or updates data on a page. Data will always be written to the next free cell with the least aging level. Using dynamic leveling, unchanged cells will still stay untouched; therefore equal wearing is not guaranteed. Static leveling does dynamic leveling, but in addition it moves static pages periodically to other pages. Data in one of the least aged pages could be moved to an average aged page to free the cell and make it usable for new data. Wear leveling performs in the background and, to minimize the impact on performance, mostly while the memory is in standby. Figure 2.12 illustrates a comparison of the wear across the memory with and without using wear leveling technology.

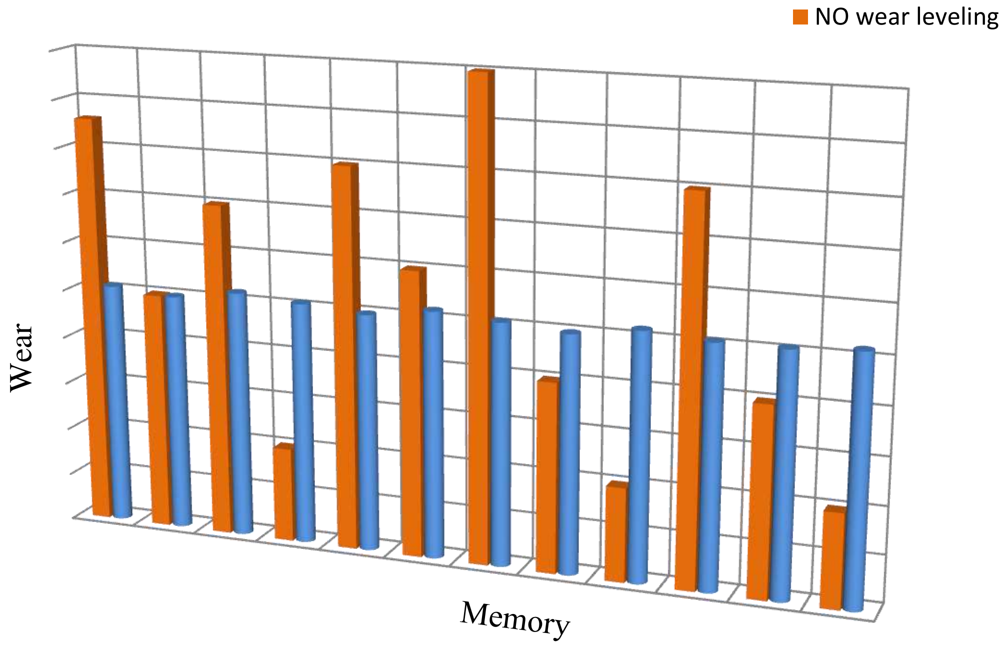


Figure 2.12 Wear leveling [13]

2.2.10. Garbage Collection

When an LBA address has been mapped to a new page or the file-system has instructed the memory to delete an address the page will not be erased immediately, but marked as *to be deleted* using the TRIM command. This is because of the previously mentioned hardware limitation of NAND chips; a block contains multiple pages and could therefore contain more data than the data *to be deleted*, but only whole blocks can be deleted. The garbage collection routine keeps track of *to be deleted* pages and erases whole blocks when a whole block is ready to be deleted. If a block contains too many *to be deleted* files or more empty blocks need to be created, the garbage collection will move remaining pages into different pages before deleting the block. During this routine leftover data from *to be deleted* blocks will be combined in empty blocks to be able to delete others. This operation is performed in the background and is like wear leveling not visible to the host system [14].

2.2.11. Applications of flash memories

There are three major types of flash memories; each has its own target application and has therefore a slightly different implementation, characteristics and architecture. The functions and routines mentioned above are in general the same for all flash memories, although all flash memories need to contain a basic wear leveling and garbage collection their implementations vary from vendor to vendor. Each vendor keeps the exact algorithms a secret and is therefore not or sparsely documented [15].

Secure Digital cards, known as SD cards (Figure 2.13) are memory cards introduced in late 2001 containing flash memory optimized for a small form factor and fast writing processes of relatively small files. The SD Card Association sets the specifications for Secure Digital cards and ensures compatibility for different standards like Secure Digital High Capacity (SDHC), starting at 4GB, and Secure Digital Extended Capacity (SDXC), starting at 64GB or speed class ratings (Class 2, 4, 6, 10) that deliver a minimum data transfer rate (2, 4, 6, 10 Mbit/s). Host devices are mostly cameras and camcorders or mobile phones for the smaller micro SD (Figure 2.13) which are only a fraction of the size of the standard SD card. All SD cards are designed for use with FAT/FAT32/exFAT/NTFS file formats and come with an integrated memory controller, as illustrated in Figure 2.14, which performs very basic wear leveling operations as well as basic garbage collection [16].

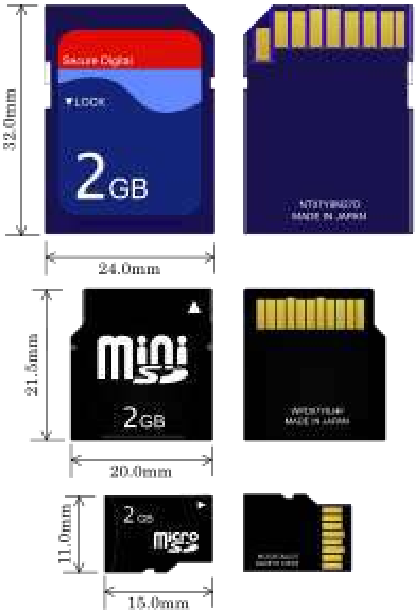


Figure 2.13 SD, mini-SD and micro-SD card [17]

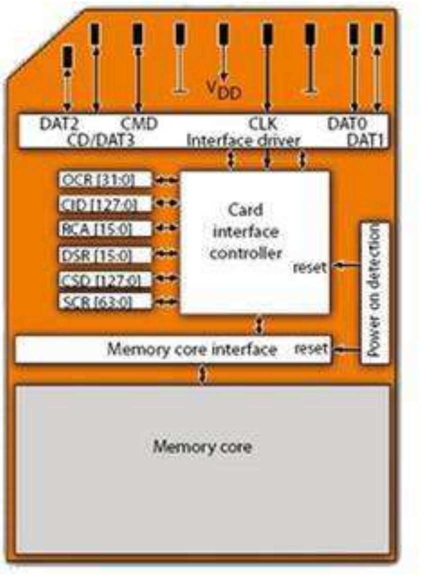


Figure 2.14 Inside an SD Card [18]

USB Flash drives were introduced in 2002 and offer a combination of fast transfer rates and high capacity on a small form factor and were intended as an alternative to CDs and floppy drive and transfer data quickly from one computer to another. A USB flash drive consists of the USB connector, a memory controller and the NAND flash memory chip as illustrated in Figure 2.15 [16].

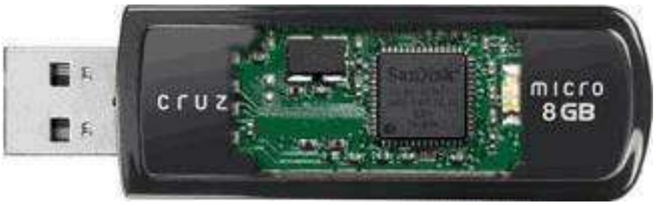


Figure 2.15 USB Flash memory drive [19]

A solid-state drive (SSD) is a storage device introduced in 2007 with much larger capacity than SD cards or USB flash memory. SSDs are designed to replace traditional HDDs, use the same form factor and interface and are therefore easily replaceable in most computer systems. Nowadays much smaller form factors are built in order to fit SSDs in even smaller and thinner hardware. The SSD controller manages functions such as manufacturer dependent intensive wear leveling and garbage collection. SSDs are used in desktop pcs, notebooks, server and storage systems [16]. Figure 2.16 shows the inside of an SSD disk.

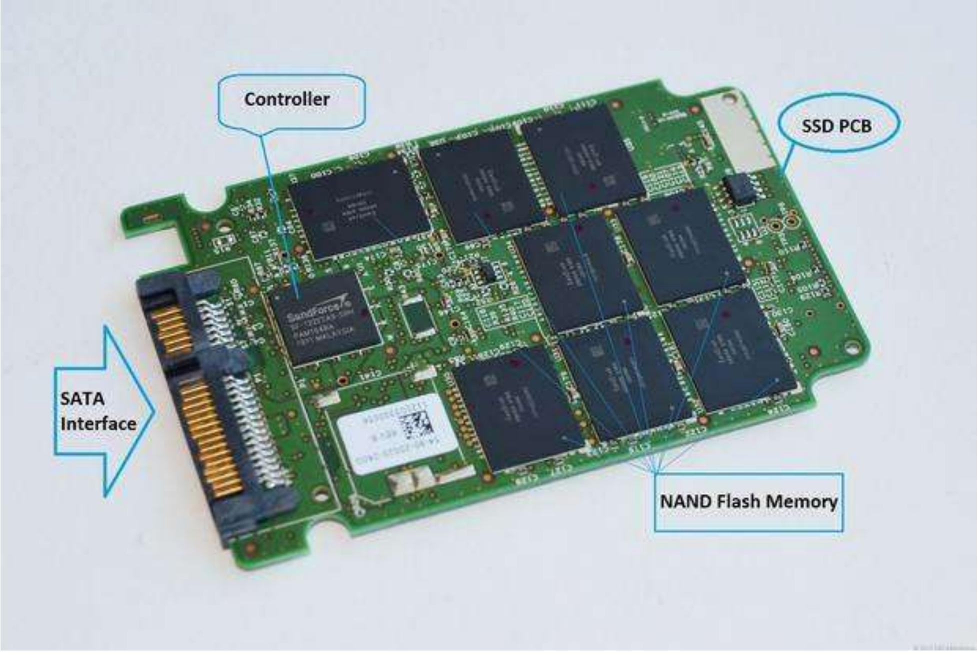


Figure 2.16 Inside an SSD disk [20]

2.2.12. Hybrid applications

High prices for flash memory in comparison to traditional HDD drives led to a hybrid storage technology combining the benefits of both storage technologies into one solution. Different types of applications have been developed since.

One of the first end-user ready solutions was Windows's ReadyBoost technology using an external flash memory device as a fast cache for the operating system. The faster seek time of the flash drive is used to route I/O read requests to populated disk sectors on the Flash memory instead of the actual hard disk's memory sectors.

Seagate released Momentum XT in 2010, an application of *adaptive memory*, a hybrid solution with HDD and SSD memory combined on one drive. This application's memory controller manages the two memories and decides on what memory to store certain data based on user trends and algorithms monitoring data access transactions. Therefore Momentum XT applications are not operating system dependent because the memory controller hides the combination of SSD and HDD technologies and acts as a traditional drive to the operating system. Up to 50% performance improvement, faster data access rates, faster boot processes and decreased power consumption are among the benefits of this technology [14].

Figure 2.17 gives an illustration of a hybrid storage system.

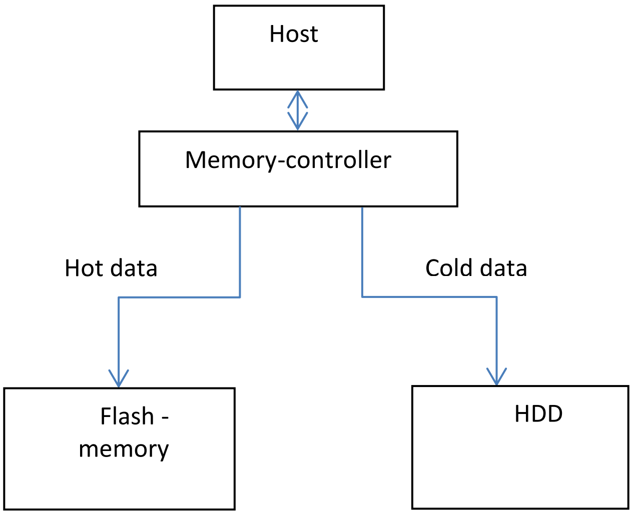


Figure 2.17 Hybrid storage system

2.3. Forensics

“Forensic science is the scientific method of gathering and examining information about the past which is then used in a court of law.” [21]. Evidence is collected to create a link between a crime and a suspect in order to prove its guilt or innocence. In order to provide reliable evidence three concepts are important; Chain of Custody, Admissibility of Tests, Evidence and Testimony and the Expert Witness.

The chain of custody describes carefully the documentation and evaluation of whatever kind of evidence. Certain types of evidence cannot be preserved indefinitely because of its nature, like a human corps and blood spatters, or are destroyed while analysing, like blood tests for drugs and need to be properly documented, evaluated and imaged. Using these documents and images it should be possible to re-evaluate the evidence again at any time. This documentation needs to contain proves about the secured location the evidence has been stored in for the time of discovering until current date. Each change of location must be documented. If this documentation is contains gaps in time the evidence may be rejected and be inadmissible to court.

Admissibility of Tests, Evidence and Testimony involves the existence of legal standards for the admissibility of forensic tests and expert testimony. One legal standard for the admissibility of forensic evidence is the Frye standard, which states that the forensic technique in question must have *general acceptance* by the scientific community.

Expert Witness Relating to all forensic science disciplines is the third issue, the concept of the expert witness. In an investigation of any kind there can be a fact witness, who can usually only relate facts that he/she observed, and an expert witness. The expert witness has specific expertise within a particular discipline and is able to offer opinions that relate to the specific discipline. An expert witness needs to be officially recognized and qualified which usually involves a legal process [22].

The mentioned concepts are the same for all forensic science disciplines like Pathology, Anthropology, Odontology or Digital forensics.

2.3.1. Digital evidence:

Digital evidence is defined by Eoghan Casey as “any data stored or transmitted using a computer that support or refute a theory of how an offense occurred of that address critical elements of the offense such as internet of alibi” [6]. Digital is data that can establish a link between a crime and a victim or a suspect or can prove the occurrence of a crime. Such data can consist of texts, images, audio and video. Examples of digital evidence are email archives, IRC chat histories, images, surveillance videos or log files showing access to certain resources. Case example 1 is an example of a real world legal case in Kansas where digital evidence helped finding and convicting a suspect.

After eluding police for more than 30 years, a serial killer in Kansas re-emerged, took another victim, and then sent police a floppy disk with a letter on it. On the disk forensic investigators found a deleted Microsoft Word file. Inside that file's metadata was metadata containing the name "Dennis" as the last person to modify the deleted file and a link to the Lutheran Church, where Rader was a Deacon. (Ironically, Rader had sent a floppy disk to the police because he had been previously told, by the police themselves, that letters on floppy disks could not be traced.)

Case example 1 (KANSAS, 2005) [23]

2.3.2. Digital forensics

When a crime has been committed in the physical world many times evidence can be found in digital on a suspect's digital devices or on the internet. The internet expands with more sensors surveilling the real world daily like traffic cameras, ATM cameras, and webcams. People also tend to post more messages on social media websites or chat in IRC rooms where IP addresses reveal one's location and conversations are being logged. Whenever an investigation is ongoing and there is chance of digital evidence a digital forensic investigation needs to be conducted. This typically includes seizing a suspect's digital devices like personal computer, mobile phone, navigation device, memory devices and to search them for possible evidence or leads.

TJX, the parent company of T.J. Maxx, Marshalls, and other retail stores in the United States, Canada, and Europe, was the target of cyber criminals who stole over 90 million credit and debit card numbers. After gaining unauthorized access to the inner sanctum of the TJX network in 2005, the thieves spent over 2 years gathering customer information, including credit card numbers, debit card details, and driver's license information. The resulting investigation and law suits cost TJX over $170 million. In 2009, a Ukrainian man named Maksym Yastremskiy was apprehended in Turkey and was convicted to 30 years in prison for trafficking in credit card numbers stolen from TJX. Digital evidence was obtained with some difficulties from computers used by Yastremskiy, ultimately leading investigators to other members of a criminal group that had stolen from TJX and other major retailers by gaining unauthorized access to their networks. In 2010, Albert Gonzalez was convicted to 20 years in prison for his involvement in breaking into and stealing from TJX.

Case example 2 (MASSACHUSETTS, 2005–2010) [6]

When a digital medium is examined by forensic specialists, evidence must sometimes be recovered from broken or purposely destroyed memory, deleted or lost data. Regardless of the state of the device and the data one very important step has to be taken first: create an image that is a digital copy of the state when the device was collected. This image is important to prove the chain of custody, the integrity of the evidence possibly found during the investigation thus it can be proven that the data on the medium has not been altered by the investigator or a third party from the time the device was collected until a possible presentation in court. The step of verifying the integrity generally includes a comparison of the digital fingerprint between the initial image and the evidence presented. This digital evidence mostly consists of a hash value of the image, meaning a computed checksum of the data. This checksum is most commonly calculated by a MD5 or SHA-1 algorithm. All hash algorithms produce a nearly unique fingerprint, which will always be the same given the same input. For example the MD5 hash algorithm produces a 128-bit checksum of any input with arbitrary length. Therefore, an exact copy or image of a device will have the same digital fingerprint as the original; a minor change would cause a different fingerprint from the original as shown in Table 2.1 [6].

Digital Message	MD5 Output
This is a message and possible evidence.	9e2422e9d18d29053e9395baf64d1067
This is a message and possible evidence!	e48c4c419500240e3a8415c67820ab3a

Table 2.1 Different MD5 checksums for two messages

After acquiring a device containing digital memory the investigator will try to take a digital image of the collected device before searching for evidence. Sometimes this is initially not possible due to hardware failure of intentional or unintentional nature. Therefore a hardware or software based recovery has to be performed.

2.3.3. Digital forensics and the law

While investigating a past or ongoing crime the forensic investigators are restricted by the law. It contains regulations to protect the privacy of the public. Hereby differences are made between stored information (saved email, pictures) and transmitted information (VoIP traffic), where the latter is considered more private and is therefore stronger protected by making it harder to obtain a warrant. The *International Organization on Computer Evidence* (IOCE) is an agency that is establishing compatible international standards for the seizure of evidence. The US Electronic Communications

Privacy Act regulates the authority of investigators as well as companies on communications of their employees. The authority of UK investigators is legislated by the Regulation of Investigatory Powers Act [24].

2.3.4. Hardware recovery on HDDs

Hardware failures do not have to be of intentional nature. Electronics in disks are very fragile, so are the read and write heads. Scott Moulton, a forensic specialist, showed in his presentation at ToorCon, an Information Security Conference, what the most typical hardware failures are (Figure 2.18) [25].

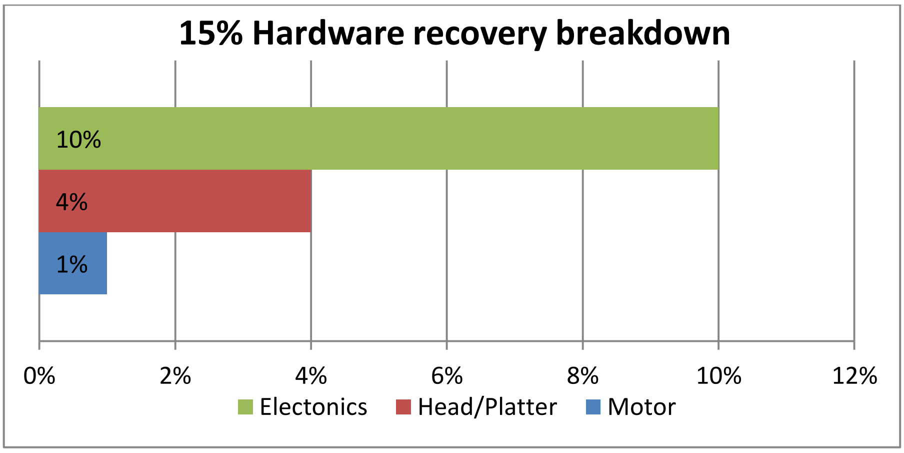


Figure 2.18. Hardware recovery breakdown [25]

In any of the above cases the most promising way of restoring data from a broken device is replacing parts that are broken. Most of the time the platters are still intact, only the mechanisms, to read the information from them, are not working properly. In these cases it is very important to get the exact same hardware as the faulty one, because each vendor and model uses slightly different technologies. Basically three components can be replaced. If an arm, slider or head is broken the whole arm needs to be replaced, otherwise the electronic board containing chips and firmware can be replaced as well as the spindle motor. The whole spindle can be placed in a different casing containing all other hardware. Here it is crucial that the disks in the spindle do not change its position to the other disks. The chances of restoring data from a faulty drive in the above cases are very high if the replacement is done very carefully and in a clean environment [25].

2.3.5. Hardware recovery on flash memory

Recovering data from flash memory is more difficult than from hard disk drives; all control and memory chips are soldered to one board. Therefore we cannot simply replace a part of the device without finding the exact same model and replace parts by re-solder them. Depending on the type of flash memory 2 to 20 chips sit on one board. Re-soldering them by hand is a difficult and fragile job and nearly impossible for multiple chips [1].

Another Possibility is to unsolder each memory chip and read each one separately using special hardware and tools. This is possible for memory sticks with one chip, on SSDs with multiple chips this method becomes very complex because each vendor uses different strategies on how to address chips, how to perform wear leveling and garbage collection and how to distribute data. Figure 2.19 shows hardware used to read a single flash memory chip [26].

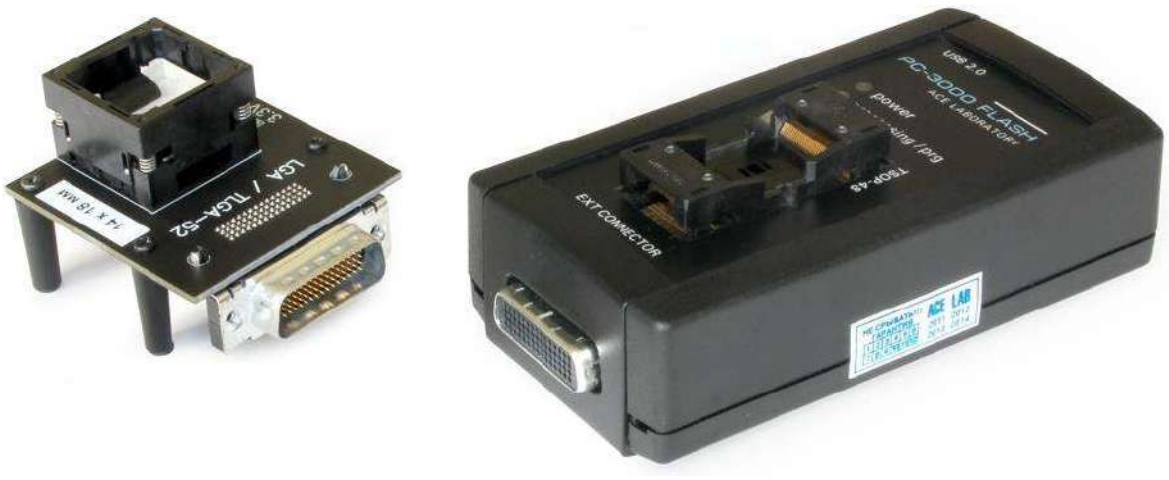


Figure 2.19 PC-3000 Flash SSD Edition [26]

2.3.6. Software recovery from HDDs

Data recovery is not always hardware related. In far more cases analysis of the disk using software is enough to recover information from a disk. As mentioned before, the actual file is not deleted from hard disk drives and they will eventually be over-written by a new file. This fact is commonly used to recover data. Most importantly, while restoring data from disks and gathering evidence, the original data must stay untouched and altered as little as possible. Therefore specialists use specific hardware to bitwise copy the information on the disk to an image file or another disk. This device is called the write blocker which it is used as the connection between the hard drive and computer and monitors the commands that are being issued and prevents the computer from writing data to the disk, as illustrated in Figure 2.20. Read commands are passed to the device while write commands will be blocked. Such an image of the original file system will then be examined using software tools as mentioned in the next chapter [27].

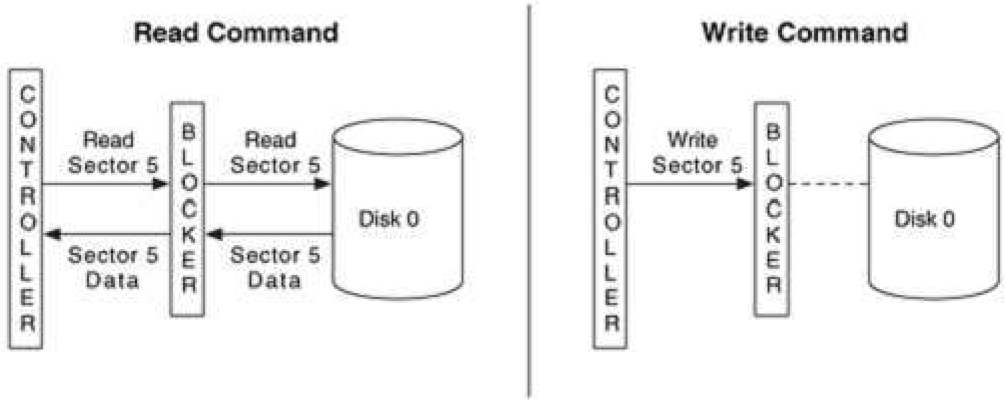


Figure 2.20. Logical view of the write blocker [27]

2.3.7. Forensics software tools

Many tools have been developed that forensic personnel can use to recover data from hard disk drives and other digital memory, expensive software suites as well as open source tools. However, one of the most known and common forensic evidence gathering tools is EnCase. It can copy disks using bit stream technology to create a virtual reconstruction of the file system. FTK (Forensic Toolkit by Access Data) and X-Ways are two different windows based tools and the special feature of these three tools is the additional data stored with the disk image like MD5 hash values to prove the integrity of the image. Sleuth Kit is an open source software suite that runs on different operating systems and supports all common file systems. Autopsy is “a digital forensics platform and graphical interface to The Sleuth Kit and other digital forensics tools.” [28]. Other well-known freeware tools are Recuva [29], rated as *very good* [30], and PCI File Inspector also rated “3.5 of 5” [31]. After imaging a hard drive using bit stream technology every bit on the original drive is stored in the image file and can then be examined. Above mentioned tools can help the examiner to gather possible evidence in existing files and are also able to restore data from deleted files or formatted partitions. All mentioned tools are only able to process unencrypted disks, if Encrypted File System (EFS) is used an image can be made, but analysing the data requires much more effort [6].

2.3.8. Software recovery from flash memory

In order to examine a SSD and to gather evidence of existing files the same technology is used as with conventional hard disk drives. EnCase or any other described tool is used to capture an image of the medium, in order not to alter the original data and to gather potential evidence files. [6] When partitions have been formatted or files deleted prior the examination examiners have few chances for recovery of data. This is because in contrast to hard disk drives flash memory and in particular SSDs have internal routines that cannot be influenced from outside for example with a write blocker. [1].

2.3.9. Forensic tools for flash memory

The tools that can be used to capture images and gather potential evidence on SSDs are the same as for HDDs. In order to read out single memory chips from an SSDs or other flash memory in case of a hardware problem or to avoid internal routines to alter the data saved on the memory chips, these four tools can be used:

- PC-3000 Flash SSD Edition (ACE Data Recovery - Russia) [26]
- Dumppicker (Russia) [32]
- Flash Extractor (Russia) [33]
- Flash Doctor (China) [34]

All the above tools work in a similar way. The hardware as shown in Figure 2.18 reads the content of a memory chip. The software then compares the chip manufacturer and model with a database and assists recovering existing files [26].

In 2015 ACE Data Recovery announced an extended cooperation between the Data recovery firm and SandForce and the development of a new custom software to improve SandForce based SSD data recovery. As mentioned previously SSD controller's manufacturers face a very strong competition and are not willing to share the insight of the internal routines, encryption, wear leveling and garbage collection. Therefore the cooperation between a big data recovery firm and the biggest manufacturer of SSD controllers is a huge step and improvement for forensic examiners and data recovery specialists and has increased the recovery rate for SandForce based SSDs drastically [35].

3. **Testing**

The Testing section will analyse and prove the theory and background provided in Chapter 2 and will investigate what effects these have on digital forensic and data recovery processes.

In this section a series of tests will be carried out on different hardware. The tests will be repeated on two to three different drives for each type of hardware, hard disk drives, SD memory cards, USB memory drives and SSD drives. As a preparation of the tests all hardware had been formatted with the NTFS filesystem and the memory has been completely filled with a jpg file (Figure 3.1).

Figure 3.1 Test image file

This file was chosen because of its fairly big size (18.2 MB) and format (jpg) is one that all forensic and recovery software will recognize. In addition it would be rather simple to check if a recovered file is intact or not by simply opening the picture file.

3.1. Tested hardware

As a test system a standard Windows PC has been used. Hardware has been attached either via USB in case of the USB memory devices, the internal card reader for the SD cards and the secondary SATA port for HDD and SSD drives.

Model:	HP Z230 Tower-Workstation
Operating system:	7 Professional 64- bit
Processor type:	Intel Core i7-4770
Installed memory:	4.00 GB

Two to three different models of each type of memory has been tested. Table 3.1 below lists the tested devices.

	Description	Size
Hard disk drives	Hitachi HTS541616J9SA00 SATA 2.5’’	150 GB
	Hitachi HTS542525K9SA00 SATA 2.5’’	250 GB
USB flash drives	General UDisk USB Device	8 GB
	SanDisk Cruzer	8 GB
SD memory cards	SanDisk Extreme Class 10 SDHC I	32 GB
	SanDisk Ultra II Class 4	2 GB
	SanDisk Micro SD Class 4 SDHC	8 GB
Solid state disks	Kingston SSD Now 300V	120 GB
	Patriot Pyro	120 GB

Table 3.1 Tested memory devices

3.2. Software used for testing

As recovery software four different programs have been used;

- Autopsy 3.1.1 [28]
- PCI File Inspector 4.0 [31]
- Recuva 1.52 [29]
- FTK imager 3.2.0 [36]

These four programs have been chosen because all of them are free for non-commercial use and come with good ratings. Autopsy is the front end for the open source forensic tool Sleuth kit.

Especially Recuva is among the best rated recovery software on Cnet.com [30]. Recuva scans the Master File Table (MFT) for files marked as deleted. Since MFT index entries are still intact even for deleted files, including entries for size and where it physically resides on the hard drive, Recuva can make a very quick estimate of which deleted files that can be recovered. Otherwise a bitwise scan of the memory in order to find file headers can be conducted.

Forensic Toolkit FTK from AccessData is very well known amongst forensic investigators. FTK imager is part of the same software suite and is used to create images and checksums of memory drives.

In addition to free software two different Java programs were written for the test cases. Program 1 (Code snippet 3.1) was written to fill up the device’s complete memory automatically using a 10 MB picture.

Program 2 (Code snippet 3.2) was designed to capture specific sections of the tested device at specific time intervals to analyse the change over time.

```
public class Fill {
        static int counter=0;
        static int folder=0;

        public static void main(String[] args) {

        try {
                BufferedImage img = null;
                img = ImageIO.read(new File("c:/image.jpg"));
                while (1!=0){
                        File dir = new File("f:/folder("+folder+")");
                        if (!dir.exists())
                        System.out.println(dir.mkdir());
                        File outputfile = new
                        File("f:/folder("+folder+")/image("+counter+").jpg");
                        if(!outputfile.exists())
                        ImageIO.write(img, "jpg", outputfile);
                        System.out.println("Folder:    "    +folder+    "    file:
                        "+counter);
                        counter++;
                        if (counter==1000){
                                counter=0;
                                folder++;
                        }
                }
        } catch (IOException e) {e.printStackTrace();
System.out.println("Volume is full after " + folder+ " folders and " +counter
+ " files.");
        }}
```

Code snippet 3.1 Java code used to fill memory with sample data

```
public class Sampler {
        public static void main(String[] args) {
                RandomAccessFile raf = null;
                try {
                byte [] block = new byte [1024];
                ArrayList<Byte> array = new ArrayList<Byte>();
                ArrayList<Integer> ChangeArray = new ArrayList<Integer>();
                long disksize=120;              //fill in Disk-size in GB
                long interval=1000*1000*10;  //each 10 MB
                Timer timer= new Timer();
                disksize=(disksize-disksize/100*5)*1000*1000*1000;
                File outputfile = new File("d:/changelog.txt");
                double decimal=100; int cycle=0;
                for (int i = 0; i < disksize/interval; i++) {
                   ChangeArray.add(0); //filling the array with zeros}
                while (true){
                   raf = new RandomAccessFile("\\\\.\\PhysicalDrive1","r");
                   FileWriter filewr = new FileWriter(outputfile);
                   int offset=0;
                   while (offset*interval<disksize){
                        raf.seek(offset*interval);
                        raf.readFully(block);
                        if (array.size()<=offset)
                           array.add(block[0]);
                        else {   if (array.get(offset)!=block[0]){
                                ChangeArray.set(offset,cycle);
                                array.set(offset,block[0]);}}
                   System.out.println("READ  BYTES  at  :  "+offset/decimal+"
                   GB: " + array.get(offset).toString());
                   offset++;     }
                raf.close();
                //write the arraylist in file.
                System.out.print("Round "+ cycle + " - ");
                for (int i = 0; i < ChangeArray.size(); i++) {
                        System.out.print(ChangeArray.get(i)+";");
                        filewr.append(ChangeArray.get(i)+";");}
                filewr.close();
                cycle++;
                synchronized (timer)
                {timer.wait(10000);}   }} catch …
```

Code snippet 3.2 Java code used to document changes on memory over time

3.3. Test cases

The following seven test cases will investigate in detail the different behaviour between the different memory devices. While *Test case 1* – Timeline of the write process investigates the timeline of the different devices being filled with the sample file *Test case 2* - Timeline of the delete process investigates the delete process to compare the timeline of the physical deletion of data on different media. Both tests use the same software Code snippet 3.1 and Code snippet 3.2.

Test case 3 – Recovery after deletion, *Test case 4* – Recovery after deletion and idle and *Test case 5* – Recovery after formatting investigate the data recovery rate of the different devices using three different recovery software tools. All devices will be filled with the sample data using Code snippet 3.1, which then will be deleted. The tests are investigating differences in recovery rate between deletion and formatting devices and if an idle time between deletion and recovery process has any influence on the recovery rate.

Test case 6 - TRIM investigates the influence of the operating system's TRIM functionality on the recovery rate while *Test case 7 – MD5 checksum comparison* compares computed checksums of the entire memory of before and after an idle time of three hours.

3.4. Test case 1 – Timeline of the write process

Test case 1 analyses the differences between SSD and HDD during the filling process and shows them in a timeline.

3.4.1. Purpose of experiment:

The purpose of this experiment is to analyse the differences of the two technologies in the writing process.

3.4.2. Method of experiment:

All devices are formatted and partitioned with one NTFS partition spanning the entire disk using the default allocation size. The tested device is analysed using the sampler Java program (Code snippet 3.2) and by using the Java code (Code snippet 3.1) filled with sample data, a jpg file (18.2 MB).

The code (Code snippet 3.2) reads bitwise in intervals of 100 Megabyte across the whole memory, as shown in Figure 3.2, saves the result, waits 10 seconds and starts the same routine again. The program then compares the results of the previous rounds with the new ones and saves the cycle number where a change happened. By doing this we can track at what time and at what position of the disk a change happened. After 12 hours the process will be stopped and the output file analysed.

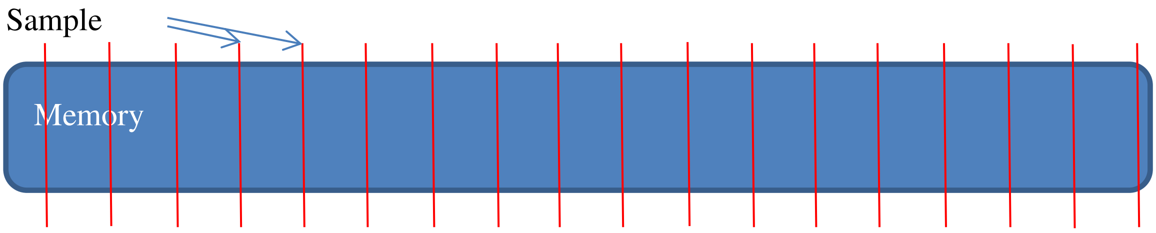


Figure 3.2 Samples in Test case 1 & 2

3.4.3. Expected result:

The expected result of this experiment is that the tested SSD drives will be faster than HDDs and will not gradually write on the disk but in chunks spread over the entire disk, whereas HDDs are expected to write bit after bit.

3.4.4. Actual result:

The tests seem to show that, against the expectations, the HDD was faster than the SSDs. Each read cycle on the HDD took an average of 2:58 minutes while on both SSDs the average time for a run cycle of the program (see Code snippet 3.2) took 8 seconds. Therefore the HDD was in fact slower than SSDs due to the fact that the read cycle took about 22 times more time which could be used for additional write cycles. The test also shows how both mediums wrote gradually on the disk while we expected the SSD to write in a different pattern. We expect this to be due to the fact that the Java program uses logical block addressing and the drive internally manages the actual location of the data.

Strangely all three devices show a spike at the beginning of the memory, which means this sector has been written at a later time. We have not found a reason for this behaviour.

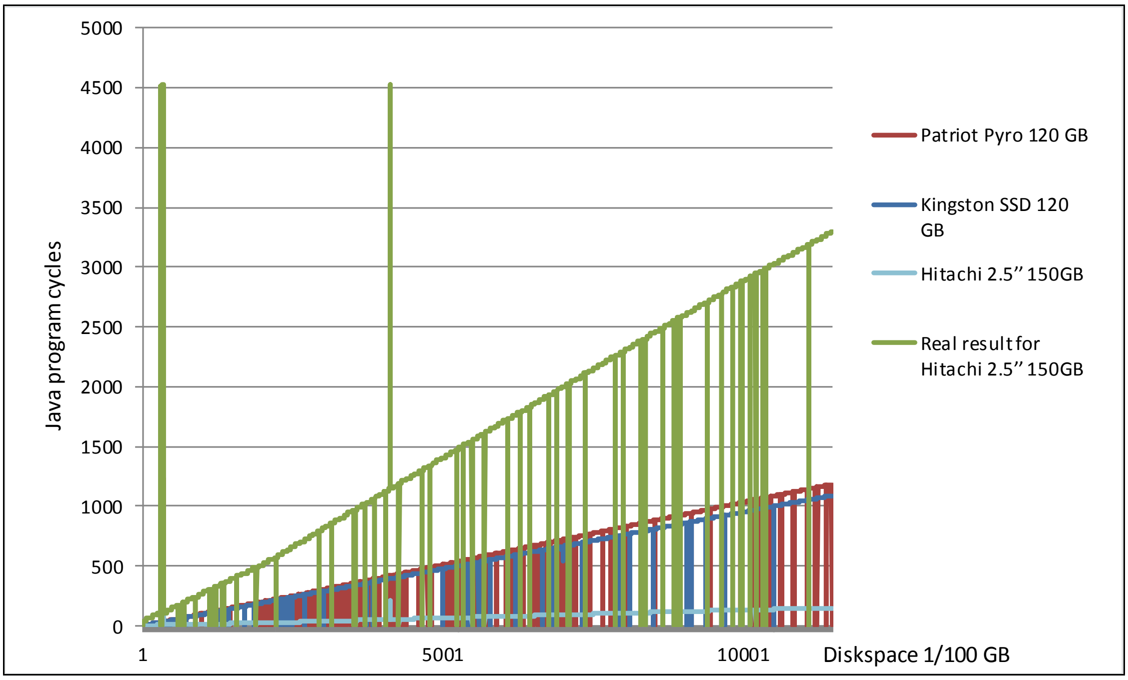


Figure 3.3 Results Test case 1

Figure 3.3 shows the program run-time cycles of the program (see Code snippet 3.2) on the y axis compared to the memory's space on the x axis. Each program cycle probes the memory space in 100 MB steps until the end of the memory is reached, waits ten seconds and starts the process from the beginning, probing the exact same addresses again. The other program (see Code snippet 3.1) does simultaneous fill the memory with the sample image file. The graph labelled as real result for Hitachi 2.5'' 150 GB shows the results of the test on the Hitachi 150 GB disk multiplied by 22 since the duration of each read cycle was 22 times longer as on the SSD drives.

3.5. Test case 2 - Timeline of the delete process

Test case 2 consists of a Java program that has been written for this test case (Code snippet 3.2). The Java program samples different parts of the tested disk repeatedly and saves changes in a file.

3.5.1. Purpose of experiment:

The purpose of this experiment is to analyse and document what and when changes to data on the disk happened. This can prove when the garbage collection and other routines in SSDs start to work.

3.5.2. Method of experiment:

All devices are formatted and partitioned with one NTFS partition spanning the entire disk using the default allocation size. The tested device is filled using the provided Java code (Code snippet 3.1) with sample data, a jpg file (18.2 MB). The files will be deleted using standard windows commands and the Java code (Code snippet 3.2) will be run. This code will, as in test case 1, monitor at what time a change happened and at what part of the memory.

3.5.3. Expected result:

The expected result of this experiment is that the hard drives, flash memory cards and USB flash drives do not show any changes in the output file. In contrary the solid state drives are expected to show changes shortly after the start of the experiment.

3.5.4. Actual result:

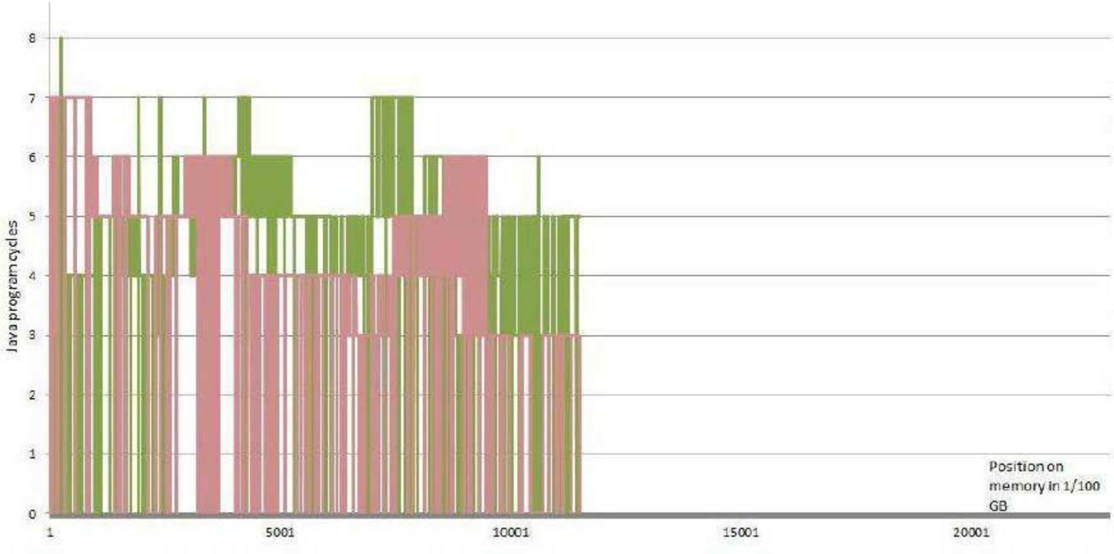


Figure 3.4 Results Test case 2

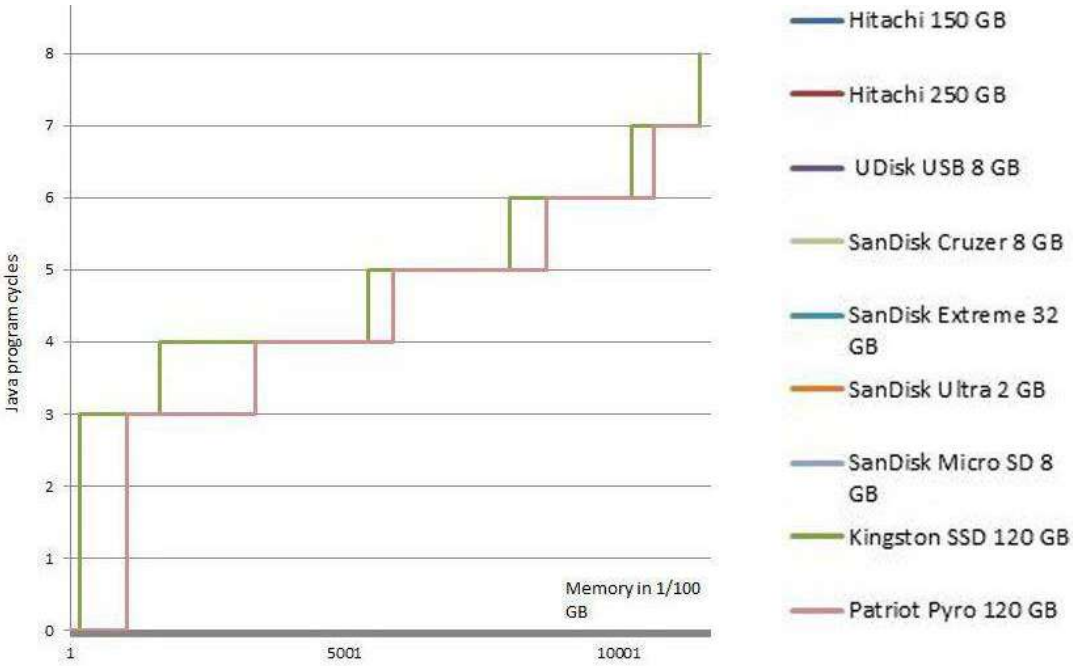


Figure 3.5 Sorted Results Test case 2

The tests show how no data changed on any device except the SSDs where the controller actively deletes cells. Figure 3.4 shows how the actions took place in big blocks at a time. Another difference is shown between the two SSD drives; while the Kingston SSD deletes nearly all data immediately the Patriot drive leaves at least about 10% untouched. Figure 3.5 shows the test results sorted by cycles to better illustrate the big blocks deleted at a time.

3.6. Test case 3 – Recovery after deletion

Test case 3 consists of a scenario where the tested devices are filled with a jpg file. The files will be deleted and a recovery process will be started.

3.6.1. Purpose of experiment:

The purpose of this experiment is to see the differences in behaviour between the different devices when files have been deleted, how many files that can be recovered and how many are lost. This experiment examines whether the flash memory continues to store the data after deletion, no overwriting, in the same manner as hard disk drives.

3.6.2. Method of experiment:

All devices are formatted and partitioned with one NTFS partition spanning the entire disk using the default allocation size. The tested device is filled using the provided Java code (Code snippet 3.1) with sample data, a jpg file (18.2 MB) and deleted using standard windows commands. A recovery process is started immediately after deleting the files to recover the data, giving the devices the least possible time for internal routines to work and possibly delete files completely from the memory. The trim command is enabled for the test on the SSDs.

3.6.3. Expected result:

The expected result of this experiment is that the hard drives, flash memory cards and USB flash drives do not lose any data or very little. In contrary the solid state drives are expected to permanently lose a big amount of data in short time.

3.6.4. Actual result:

Hardware	Test files	Recovered intact files					
		Autopsy		PCI file recovery		Recuva	
Hitachi HTS541616J9SA00 SATA 2.5’’ 150GB	18770	18769	99.995%	18769	99.995%	18769	99.995%
Hitachi HTS542525K9SA00 SATA 2.5’’ 250 GB	29334	29333	99.997%	29333	99.997%	29333	99.997%
General UDisk USB Device 8 GB	952	946	99.370%	951	99.895%	951	99.895%
SanDisk Cruzer	936	932	99.573%	932	99.573%	934	99.786%
SanDisk Extreme Class 10 SDHC I 32 GB	3734	3733	99.973%	3733	99.973%	3731	99.920%
SanDisk Ultra II Class 4 2 GB	234	233	99.573%	233	99.573%	233	99.573%
SanDisk Micro SD Class 4 SDHC 8GB	925	924	99.892%	924	99.892%	924	99.892%
Kingston SSD Now 300V 120GB	14080	0	0.000%	0	0.000%	0	0.000%
Patriot Pyro 120 GB	14083	0	0.000%	441	3.131%	766	5.439%
Average			77.597%		78.003%		78.277%

Table 3.2 Results Test case 3

The **SanDisk Ultra II Class 4 2 GB** holds 233 intact files, and one partial file and six empty files when using Autopsy. PCI file recovery and Recuva recovered 233 intact files and one partial file. The same resulted from a recovery process using Recuva and PCI File Inspector. On the **SanDisk Micro SD Class 4 SDHC 8GB** 924 intact files and one partial file have been recovered using Autopsy. Recovery on the **SanDisk Extreme Class 10 SDHC I 32 GB** resulted in 3731 complete files, 1 partial and two overwritten files when using Recuva. This time Recuva identified the reason for the data loss. The files have been overwritten by files in \$Extend\$RmMetadata which is a hidden system directory containing NTFS metadata [37]. Using Autopsy 3733 intact pictures and one partial picture were recovered the same result as when using PCI File recovery and Autopsy. Recovery processes on the **Hitachi HTS541616J9SA00 SATA 2.5’’ 150GB** using Recuva showed only one partial file, all others were intact. On the **Hitachi HTS541616J9SA00 SATA 2.5’’ 150GB** using Recuva and Autopsy resulted in all but one intact file. Recovering from the USB memory drive **UDisk USB Device 8 GB** Autopsy was able to restore 946 correct files, one partial image and around 100 empty files without any content. Recuva and PCI File Recovery recovered 951 correct and one partial file. The **SanDisk Cruzer** held 932, 932 and 934 intact files using the three recovery programs. The **Kingston SSD Now 300V 120GB** showed no single intact file after a recovery process right after deletion while the Patriot drive held about 5% of intact data using PCI File recovery and Recuva.

3.7. Test case 4 – Recovery after deletion and idle

Test case 4 consists of the same scenario as Test case 3 where the tested devices are filled with jpg files. These files will be deleted and a recovery process will be started. The difference from test case 3 is that the device will be left in idle mode for two hours.

3.7.1. Purpose of experiment:

The purpose of this experiment is to see the differences in behaviour between the different devices when files have been deleted, how many files that can be recovered and how many are lost. This experiment examines whether flash memory continues to store the data after deletion; no overwriting, in the same manner as hard disk drives. In addition this test scenario examines the impact of leaving the device in idle mode (supplied with power but without accessing or altering the data) for two hours before the recovery process.

3.7.2. Method of experiment:

All devices are formatted and partitioned with one NTFS partition spanning the entire disk using the default allocation size. The tested device is filled using the provided Java code (Code snippet 3.1) with sample data, a jpg file (18.2 MB), and deleted using standard windows commands. After leaving the device in idle mode for two hours a recovery process is started to recover the deleted files, giving each device time for internal routines to work and possibly delete files completely from the memory. The trim command is enabled for the test on the SSDs.

3.7.3. Expected result:

The expected result of this experiment is that the hard drives, flash memory cards and USB flash drives do not lose any or very little data. In contrary the solid state drives are expected to lose even larger amounts of data forever than in test case 1.

3.7.4. Actual result:

Hardware	Test files	Recovered intact files					
		Autopsy		PCI file recovery		Recuva	
Hitachi HTS541616J9SA00 SATA 2.5’’ 150GB	18770	18769	99.995%	18769	99.995%	18769	99.995%
Hitachi HTS542525K9SA00 SATA 2.5’’ 250 GB	29334	29333	99.997%	29333	99.997%	29333	99.997%
General UDisk USB Device 8 GB	952	946	99.370%	951	99.895%	951	99.895%
SanDisk Cruzer	936	932	99.573%	932	99.573%	934	99.786%
SanDisk Extreme Class 10 SDHC I 32 GB	3734	3733	99.973%	3733	99.973%	3731	99.920%
SanDisk Ultra II Class 4 2 GB	234	233	99.573%	233	99.573%	233	99.573%
SanDisk Micro SD Class 4 SDHC 8GB	925	924	99.892%	924	99.892%	924	99.892%
Kingston SSD Now 300V 120GB	14080	0	0.000%	0	0.000%	0	0.000%
Patriot Pyro 120 GB	14083	0	0.000%	441	3.131%	766	5.439%
Average			77.597%		78.003%		78.277%

Table 3.3 Results Test case 4

The SD cards: **The SanDisk Ultra II Class 4 2 GB** holds the same 233 intact files, one partial file and six empty files as in test case 3. Also the **SanDisk Micro SD Class 4 SDHC 8GB** and **SanDisk Extreme Class 10 SDHC I 32 GB** gave the same results when repeating the test waiting for two hours between deleting and sampling the data with all three software tools. Recovering from the **UDisk USB Device 8 GB**, the **SanDisk Cruzer**, the **Hitachi HTS542525K9SA00 SATA 2.5'' 250 GB** and the **Hitachi HTS541616J9SA00 SATA 2.5'' 150GB** resulted in the same recovered files as in test case 3. This test resulted in the same results for both SSD drives as in test number 3.

3.8. Test case 5 – Recovery after formatting

Test case 5 consists of the same scenario as Test case 3 where the tested devices are filled with jpg files, but the device will be quick-formatted with the NTFS file system to one partition spanning the whole memory and a recovery process will then be started.

3.8.1. Purpose of experiment:

The purpose of this experiment is analysing if formatting a device will have a different result than the previous test cases.

3.8.2. Method of experiment:

All devices are formatted and partitioned with one NTFS partition spanning the entire disk using the default allocation size. The tested device is filled using the provided Java code (Code snippet 3.1) with sample data, a jpg file (18.2 MB). The devices will be quick-formatted with the NTFS file system to one partition spanning the whole memory and a recovery process will be started. The trim command is enabled for the test on the SSDs.

3.8.3. Expected result:

The expected result of this experiment is that the hard drives, flash memory cards and USB flash drives do not lose any or very little data. In contrary the solid state drives are expected to delete big amounts of data until the recovery process started.

3.8.4. Actual result:

Hardware	Test files	Recovered intact files					
		Autopsy		PCI file recovery		Recuva	
Hitachi HTS541616J9SA00 SATA 2.5’’ 150GB	18770	0	0.000%	0	0.000%	18768	99.989%
Hitachi HTS542525K9SA00 SATA 2.5’’ 250 GB	29334	0	0.000%	0	0.000%	29333	99.997%
General UDisk USB Device 8 GB	952	0	0.000%	0	0.000%	950	99.790%
SanDisk Cruzer	936	0	0.000%	0	0.000%	933	99.679%
SanDisk Extreme Class 10 SDHC I 32 GB	3734	0	0.000%	0	0.000%	3730	99.893%
SanDisk Ultra II Class 4 2 GB	234	0	0.000%	0	0.000%	225	96.154%
SanDisk Micro SD Class 4 SDHC 8GB	925	0	0.000%	0	0.000%	925	100.000%
Kingston SSD Now 300V 120GB	14080	0	0.000%	0	0.000%	0	0.000%
Patriot Pyro 120 GB	14083	0	0.000%	0	0.000%	0	0.000%
Average			0.000%		0.000%		77.278%

Table 3.4 Results Test case 5

For the SD cards, when recovering from the **SanDisk Ultra II Class 4 2 GB** with Autopsy and PCI file recovery no intact picture could be saved, all pictures were only partial. Using Recuva 225 intact files could be recovered, 9 other files were partly complete but had missing pieces or were miscoloured. With the **SanDisk Micro SD Class 4 SDHC 8GB** the results were similar, both Autopsy and PCI File Inspector could not recover any or only partial files, Recuva recovered all 925 files. The test on the **SanDisk Extreme Class 10 SDHC I 32 GB** resulted in 3730 pictures, with PCI File Recovery only partial pictures could again be recovered and Autopsy could not find any data at all.

For the USB memory drives, recovering from the **UDisk USB Device 8 GB** using Recuva was a surprise since it recovered 950 complete pictures and an additional 1.5 Gb of data which was on the drive before formatting, filling the drive and formatting again. Therefore the test was repeated where 949 and some additional 1GB of data had been restored. On the SanDisk Cruzer 933 intact files could only be restored using Recuva.

For the Hard disk drives the recovery with Recuva on the **Hitachi HTS542525K9SA00 SATA 2.5'' 250 GB** resulted in all but one intact picture using Recuva, using PCI File Recovery and Autopsy again resulted in only partial pictures. When recovering from the **Hitachi HTS541616J9SA00 SATA 2.5'' 150GB** PCI File Recovery and Autopsy could not find any data, Recuva missed only one file and recovered one partial file.

Both the SSD drives could recover about four to five thousand files all of them where correctly named but filled with zeros. Therefore no intact file could be recovered as shown in Figure 3.5.

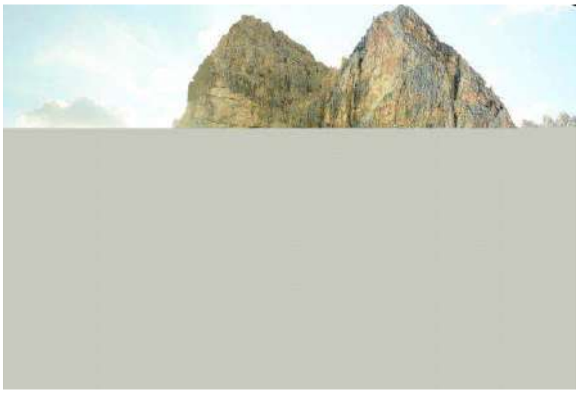


Figure 3.6 Partially recovered image

3.9. Test case 6 - TRIM

Test case 6 tests the effect of the TRIM function on the recovery process on SSD drives.

3.9.1. Purpose of experiment:

The purpose of this experiment is to analyse and document what the effects of the TRIM commands issued by the operating system are on the recovery process.

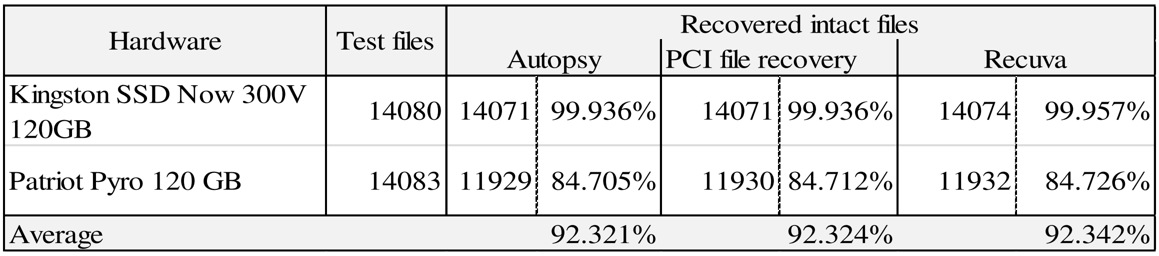
3.9.2. Method of experiment:

All devices are formatted and partitioned with one NTFS partition spanning the entire disk using the default allocation size. The tested device is filled using the provided Java code (Code snippet 3.1) with sample data, a jpg file (18.2 MB). The trim command is disabled for this test (Code snippet 2.1). Then the same Java code will be run as in test case 5 and the output file analysed.

3.9.3. Expected result:

The expected result of this experiment is that the tested SSD drives will delete less data. The data should be deleted slowly and steadily by the garbage collection instead of quickly and in big chunks.

3.9.4. Actual result:



Hardware	Test files	Recovered intact files					
		Autopsy		PCI file recovery		Recuva	
Kingston SSD Now 300V 120GB	14080	14071	99.936%	14071	99.936%	14074	99.957%
Patriot Pyro 120 GB	14083	11929	84.705%	11930	84.712%	11932	84.726%
Average			92.321%		92.324%		92.342%

Table 3.5 Result Test case 6

The results on the Kingston SSD 120 GB show that with a disabled TRIM function almost no data will actually be deleted. Only a short spike can be noticed on the **Kingston SSD,** all other files could be recovered even after leaving the drive in idle mode for 12 hours.

3.10. Test case 7 – MD5 checksum comparison

Test case 7 compares two checksums of the complete memory of different drives after a deletion before and after an idle time of three hours.

3.10.1. Purpose of experiment:

The purpose of this experiment is to analyse and document if checksums of different drives are consistent over time after deletion of data. This can show if the integrity of an image of a device issued by a forensic investigator can be proven or not.

3.10.2. Method of experiment:

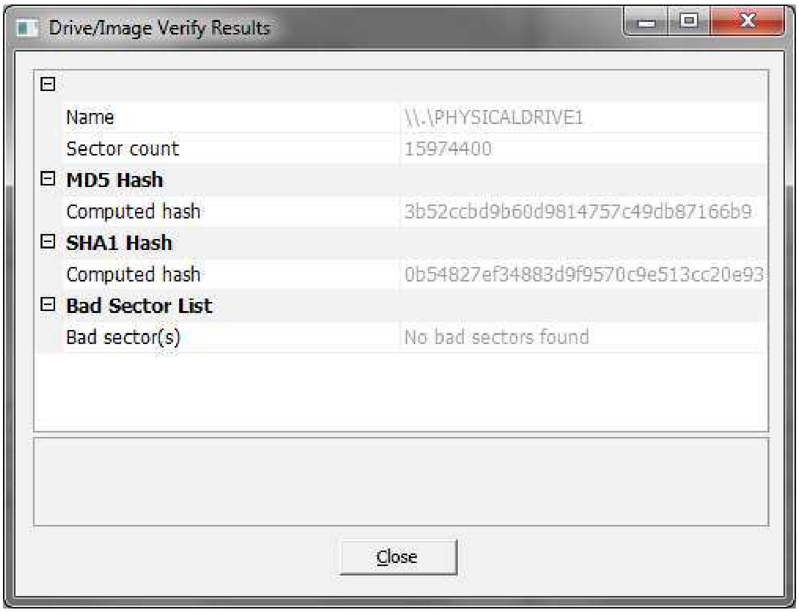
The tested devices are formatted and partitioned with one NTFS partition spanning the entire disk using the default allocation size. The tested device is filled using the provided Java code (Code snippet 3.1) with sample data, a jpg file (18.2 MB). All files are then deleted and AccessData FTK imager is used immediately to create a checksum of the complete memory. After three hours idle time the checksum will be calculated again and compared.

3.10.3. Expected result:

The expected result of this experiment is that the tested SSD drives will have a different checksum at both calculations while HDDs and other flash memory will have the same.

3.10.4. Actual result:

The results show how the checksums match in all cases of HHD drives, USB memory drives and SD cards while SSD checksums do not match after the idle time of three hours. Figure 3.6 shows the result of the checksum calculation using AccessData FTK imager.



⊟

Name	\\.\PHYSICALDRIVE1
Sector count	15974400
⊟ **MD5 Hash**	
Computed hash	3b52ccbd9b60d9814757c49db87166b9
⊟ **SHA1 Hash**	
Computed hash	0b54827ef34883d9f9570c9e513cc20e93
⊟ **Bad Sector List**	
Bad sector(s)	No bad sectors found

Figure 3.6 MD5 hash calculation result

Hardware	MD5 checksum 1	MD5 checksum 2	Match:
Hitachi HTS541616J9SA00 SATA 2.5'' 150GB	d116ed8d064ea3939ba650d6beca6efd	d116ed8d064ea3939ba650d6beca6efd	TRUE
Hitachi HTS542525K9SA00 SATA 2.5'' 250 GB	5a6d311c0d8f6d1dd03c1c129061d3b1	5a6d311c0d8f6d1dd03c1c129061d3b1	TRUE
General UDisk USB Device 8 GB	3b52ccbd9b60d9814757c49db87166b9	3b52ccbd9b60d9814757c49db87166b9	TRUE
SanDisk Cruzer	95f4d9cdae0d7f36652e43d	95f4d9cdae0d7f36652e4	TRUE
SanDisk Extreme Class 10 SDHC I 32 GB	6a8c29918d00b17c7b2aa1fc9d8b16a6	6a8c29918d00b17c7b2aa1fc9d8b16a6	TRUE
SanDisk Ultra II Class 4 2 GB	3a5f69a4124feb62f52be4619acb492a	3a5f69a4124feb62f52be4619acb492a	TRUE
SanDisk Micro SD Class 4 SDHC 8GB	55887e2daf4303193f82d6fb7594a51a	55887e2daf4303193f82d6fb7594a51a	TRUE
Kingston SSD Now 300V 120GB	9693fb0c9bfa45b5ff5f99dcdcbdc56a	8fd42ff19a450d123488d63abf88be0f	FALSE
Patriot Pyro 120 GB	ee7cf0403d7d66e14bc8c4ce83612563	d719535a44deba8b251a5136392216d5	FALSE

Table 3.6 Results test case 7

4. **Discussion**

The discussion chapter is used to summarize and discuss the results gathered in the theory and testing chapters and these results are used to answer the research question.

4.1. The research questions

The research question is defined in chapter 1.2 and 1.3. The report shows through theory and test cases the differences between hard disk drives and different flash applications in architecture and behaviour, how these affect the work of a forensic examiner and if collected evidence from such devices can hold in court.

The Literature review section provided in-depth knowledge of the architecture of hard disk drives and different applications of flash memory as well as internal routines and the position and arrangement of data on both technologies. Also an insight into forensics, digital evidence and recovery processes has been given. Using the background provided in section 2 tests have been carried out to prove the hypotheses made in chapter 1.6.

The research question has been divided into sub-questions in order to aid in answering them by focusing on specific aspects at a time.

RQ1.1: Is data persistent after deletion on flash memory in the same way as on traditional hard disk drives?

The test cases 3, 4 and 5 (chapter 3.6, 3.7 and 3.8) show how data persistence varies between traditional hard drives and flash memory. While hard disk drives, flash memory cards and USB memory devices continue to store data after deletion SSD memory devices delete almost all data immediately or make it unreadable.

RQ1.2: What is an acceptable method for forensic data acquisition on flash memory?

All test cases show how data acquisition on SSD memory is unpredictable and varies for different models and manufacturers. Therefore no acceptable method for data acquisition on flash memory has been found yet.

RQ1.3: What difference makes the TRIM functionality on SSD drives to an acquisition process?

Test case 6 (chapter 3.9) showed that TRIM has a huge impact on the acquisition process. A disabled TRIM functionality makes an SSD function similar to a hard disk drive by continuing storing deleted data an enabled TRIM function triggers the internal routines to permanently delete almost all data.

RQ1.4: Does an idle time between deletion and acquisition affect the recovery process?

An idle time between deletion and acquisition process did not affect the results as seen in test case 4 (chapter 3.7). This behaviour is expected to be dependent on the firmware implementation and therefore to vary between models and manufacturers.

RQ1.5: Does formatting a medium in comparison to deleting all data affect the acquisition process

Formatting triggered an even more aggressive deletion process on the SSD drives but a constant behaviour on other memory types.

4.2. Hypotheses testing

The hypotheses describes in chapter 1.6 have been tested and the results are summarized here.

H1: Data is not or only partially persistent after deletion on flash memory in comparison to traditional hard disk drives.

Hypothesis 1 has been partially proven valid in test cases 2, 3, 4 and 5 (chapter 3.5, 3.6, 3.7 and 3.8) where data stored on SSD memory showed the expected result where 95-100% of the data was not recoverable or unusable after deletion. Other flash memory applications showed persistent data and good recovery rates. This difference is explained by different implementations of wear leveling, garbage collection and TRIM functionality on flash memory cards and USB memory drives. Chapter 4.5.1 describes a problem encountered when discovering that the TRIM functionality is not working over an USB interfacing device. Since memory card readers and USB memory devices both use the USB interface as a connection the TRIM functionality is suspected to be none-functioning on these devices.

H2: An acceptable method for forensic data acquisition on flash memory does not exist yet.

The reason is the forensic examiner being unable to prove the integrity of an image taken at time of acquiring a memory device at the time presenting evidence in court due to the self-destructive behaviour of the SSD devices. Test case 7 (chapter 3.10) shows that SSD drive's internal routines are working in the background and change the result of a checksum calculation constantly which makes it impossible to prove integrity for a forensic examiner.

H3: The TRIM functionality on SSD drives is expected to be responsible for data loss.

Test case 3.9 proves hypothesis 3 and the effect of the TRIM functionality on the recovery process on SSD drives. A disabled TRIM functionality resulted in very good recovery rates while enabled TRIM led to minimum 95% permanent data loss.

H4: Idle time between deletion and acquisition is expected to influence the result of a recovery process.

Hypothesis 4 could not be proven in the carried out test cases, all permanent data loss happened almost immediately after the deletion as test case 2 (chapter 3.5) showed. Different implementations from different models or manufacturers are expected to have different results.

H5: Formatting a medium is expected to influence the result of a recovery process.

Test case 5 (chapter 3.8) shows that formatting SSDs triggered an even more aggressive permanent deletion of data while other memory devices showed very similar results as after normal deletion.

4.3. Discussion of findings

The research shows the evolvement of digital memory from HDD being the most used memory device to SSD slowly taking over due to some obvious advantages against HDDs. This research includes architectural differences of HDD and flash memory as well as different applications for flash memory. The conducted tests investigated the influences of the found differences on real world simulations of data recovery processes and how they could affect forensic investigations.

The results complied in almost all cases with the hypotheses proposed before the research. Tests showed how predictable HDD recovery can be and that, with the right software, recovery rates of over 99% can be reached while SSD recovery was very unpredictable with very low recovery rates of 0 to 5%. A surprise was the behaviour of other flash applications, which behaved more like HDDs with recovery rates of over 95%.

Previous research from Bell and Boddington showed similar results and while my tests investigate more in detail the different internal routines our conclusions are the same. Given the really low data recovery rate data SSDs have been accused of being the end of recovery and forensic investigations and given the fact that SSD memory is violating the golden rule of forensics and digital evidence, to maintain a chain of custody and the ability to prove the integrity of evidence to all times, this is not too far off at the current state of technology. The internal routines are not controllable from an operating system and are falsifying checksums and data deletion is completely invisible and is run in the background.

Another interesting finding was the impact of the TRIM function on the recovery rate. If disabled before deletion the tested SSDs would act like HDDs and would have a recovery rate from 85-99%. This finding does not affect the problems of the forensic examiner since TRIM functionality is enabled by default and would be enabled especially if evidence has been deleted purposely.

It appears that the key to the differences and unpredictability in data recovery lies in the firmware of the memory controller. There are only a handful of manufacturers producing SSD memory controllers and they are controlling the whole SSD market and still each protects its algorithms as a secret black box creating a market without using any standards. Overriding this firmware could therefore be the key to more reliable and higher recovery rates as well as constant data on memory while in a read only mode which would lead back to well documented guidelines and standards for data recovery specialists and forensic examiners. Evidence acquired from SSD memory could then be proven of integrity and would not be risked of not being accepted as evidence in court because alterations after acquiring would be exceptions.

The differences found, leading to a new behaviour of memory devices during a forensic investigation, will change the work of future investigations. Three possibilities could be estimated. Firstly forensic investigations would need to adapt new guidelines and standards in order to maintain prove of integrity of data before and after an investigation. This could be by a chip based analysis of the memory and disconnecting the memory chips from the SSD memory controller in order to prevent internal routines to falsify checksums and data. Secondly data recovery specialists could create custom firmware for SSD memory controllers that could be used to prevent internal routines from working. Massive research including reverse engineering of memory controllers would need to be conducted if no collaboration with manufacturers was possible. The third possibility could be collaboration between SSD controller manufacturers in order to create standardized routines and ways to prevent internal routines from working. Such measurements could be installed either by a hardware switch, comparable to a write block switch on cassette tapes or floppy drive, or software based.

A first step in this direction was taken by the collaboration between ACE Data Recovery solution and SandForce where the recovery specialists would need to have insight to the internal routines in SandForce driven drives in order to improve their recovery rates drastically. Measurements binding manufacturers to standards restricting routines either by hardware or software would need to be taken by authorities.

4.4. Method reflection

The report consists of theoretical reviews to cover the empirical investigation to address all differences between the two technologies using academic publications, books and on-line resources. Secondly tests have been carried out to prove these differences and show the problems caused by these differences through simulations of real world forensic investigations and data recovery techniques. By using known forensic software tools these tests can resemble real world scenarios.

4.5. Encountered problems

The below section describes encountered problems during the testing process.

4.5.1. Interfacing device

Initially the tests were planned using a Sharkoon SATA to USB interface device. The decision was based on the plug and play functionality of the interface device as well as less mechanical work. Tests and research showed that TRIM functions on SSD devices do not work properly over USB connection. Therefore test results were incorrect and had to be repeated using the internal secondary SATA connection.

4.5.2. Panic mode on SandForce driven devices

The Patriot Pyro 120 GB SSD of the tested hardware went into a panic mode, a measure designed to prevent the hardware from damage due to shortcuts while the test system was sent into standby mode. This is not the intended behaviour and a known bug for older versions of the SandForce firmware.

5. **Conclusion**

The concluding chapter sums the overall outcome of this report and findings in previous sections and discusses the possibilities of extending it in the future through further research.

5.1. Conclusions

The research and tests show the problems of the topic and show the difference newly introduced memory technologies have created for data recovery specialists and forensic investigators. Four out of five hypotheses were proven to be true which underlined the problems once again. Tests showed significant differences between different models and also showed how unpredictable data recovery suddenly has become and checksum tests of entire drives proved that the internal routines running in the background on SSDs falsifies the results without a possibility to disable these. Test cases 3, 4 and 5 (chapter 3.6, 3.7 and 3.8) show how data persistence varies between traditional hard drives and flash memory. While hard disk drives, flash memory cards and USB memory devices continue to store data after deletion SSD memory devices delete 95 to 100% of all data immediately or make it unreadable. All test cases show how data acquisition on SSD memory is unpredictable and varies between different models and manufacturers. Therefore no acceptable method for data acquisition on flash memory has been found yet. It is therefore a fact that forensic examiners cannot follow traditional guidelines in order to process and gain digital evidence that can be proven to be 100% untampered with.

Previous research shows that general awareness of the topic is rising after a period where no one seemed to notice the problem. This is a first important step and the foundation for further research with a hope to create standards and new guidelines in the future.

5.2. Further research

The section 1.9 Scope and limitations described the limitations of this report. Tests on multiple models of hardware from different manufacturers were not possible for this research paper. Further research could begin here. Data recovery specialized companies could repeat similar tests on all manufacturer's models in order to see the different behaviour of different memory controllers. Different tests should also cover different firmware revisions in order to see how firmware improves and changes and affects data recovery rates. Firmware tests could go as far as installing custom firmware to prevent internal routines from working. This research could be backed by memory chip analysis where flash memory chips could be unsoldered to analyse the exact behaviour of the devices and the location of data.

After in depth testing improved data recovery solutions could be offered that are manufacturer and memory controller independent in contrary to the existing ACE Data Recovery solution for SandForce driven devices resulting of a collaboration of the two companies.

A better collaboration between SSD manufacturers could for example lead to a standardized read only switch on all drives and collaborations with data recovery firms could lead to standards and guidelines how to gain better data recovery rates and make evidence gathered again hold in court.

Reference List

[1] S. Moulton, "Solid State Drives Destroy Forensics & Data Recovery Jobs," Las Vegas, 2011.

[2] G. B. Bell and R. Boddington, "Solid State Drives: The Beginning Of The End For Current Practice In Digital Forensic Recovery?," *The Journal of digital forensics, Security and Law,* pp. Volume 5, Number 3, 2010.

[3] WIRED, "WIRED," 01 03 2014. [Online]. Available: http://www.wired.com/2014/01/tech-time-warp-ibm-ramac/. [Accessed 09 02 2015].

[4] C. Mellor, "The Register," 03 05 2013. [Online]. Available: http://www.theregister.co.uk/2013/05/09/ihs_on_pc_hdd_ssd_units/. [Accessed 10 02 2015].

[5] A. A. Mamun, Hard Drive Mechatronics and Control, Boka Raton, FL: CRC Press, 2007.

[6] E. Casey, Digital Evidence and Computer Crime Third Edition, Watham, San Diego, London: Academic Press, 2011.

[7] W. Miller, "Understanding the Differences Between NAND Flash and NOR Flash Memory and Key Future Trends," [Online]. Available: http://www.em.avnet.com/en-us/design/technical-rticles/Pages/Articles/Understanding-the-Differences-Between-NAND-Flash-and-NOR-Flash-Memory-and-Key-Future-Trends.aspx. [Accessed 13 02 2015].

[8] J.-J. Maleval, "storagenewsletter.com," 14 06 2011. [Online]. Available: http://www.storagenewsletter.com/rubriques/solid-state-ssd-flash-key/91-ssd-manufacturers-in-the-world-document/. [Accessed 11 05 2015].

[9] K. Vättö, "anandtech.com," 29 05 2014. [Online]. Available: http://www.anandtech.com/show/8073/seagate-acquires-sandforce-from-avagolsi. [Accessed 11 05 2015].

[10] Slidesharecdn, "slidesharecdn.com," 2014. [Online]. Available: http://image.slidesharecdn.com/flashmemorysummit-itbrandpulse-enterprisessd-whoisadoptingthemandwhy-150221192132-conversion-gate01/95/enterprise-ssd-who-is-adopting-them-and-why-38-638.jpg?cb=1424568237. [Accessed 11 05 2015].

[11] Seagate, "seagate.com" n.a.. [Online]. Available: http://www.seagate.com /ca/en/tech-insights/duraclass-technology-master-ti/. [Accessed 11 05 2015].

[12] Intel.com, "intel.com," 02 2015. [Online]. Available: http://www.intel.com/support/ssdc/hpssd/sb/CS-031846.htm?wapkw=(TRIM). [Accessed 11 05 2015].

[13] Adata, "adata.com," Adata, [Online]. Available: http://www.adata.com/?action=product_feature&cid=3&piid=33. [Accessed 2015 02 13].

[14] R. Micheoloni, A. Marelli and K. Eshghi, Inside Solid State Drives (SSD), Dortrecht: Springer, 2013.

[15] SanDisk, "ugweb.cs.ualberta.ca," 2003. [Online]. Available: http://ugweb.cs.ualberta.ca/~c274/resources/hardware/SDcards/WPaperWearLevelv1.0.pdf. [Accessed 17 02 2015].

[16] T. C. Kingston, "media.kingston.com," 2012. [Online]. Available: http://media.kingston.com/pdfs/FlashMemGuide.pdf. [Accessed 26 02 2015].

[17] Elinux, "elinux.org," 23 02 2013. [Online]. Available: http://elinux.org/File:SD_Card_dimensions.png. [Accessed 26 02 2015].

[18] colorfoto.de, "colorfoto.de," 07 03 2008. [Online]. Available: http://www.colorfoto.de/ratgeber/halbleiter-372321.html. [Accessed 26 02 2015].

[19] USB2U, "USB2U," 10 07 2009. [Online]. Available: http://www.usb2u.co.uk/articles/2009/07/usb-flash-drives-explained/. [Accessed 26 02 2015].

[20] D. Ngo, "cnet.com," 01 03 2013. [Online]. Available: http://www.cnet.com/how-to/digital-storage-basics-part-4-ssd-explained/. [Accessed 26 02 2015].

[21] Wikipedia, "wikipedia.org," n.a.. [Online]. Available: http://en.wikipedia.org/wiki/Forensic_science. [Accessed 11 05 2015].

[22] J. Prahlow, Forensic Pathology for Police, Death Investigators, Attorneys, and Forensic Scientists, New York: Springer, 2010.

[23] forensicswiki.org, "forensicswiki.org," 26 07 2013. [Online]. Available: http://forensicswiki.org/wiki/Famous_Cases_Involving_Digital_Forensics. [Accessed 11 05 2015].

[24] gov.uk, "legislation.gov.uk," 01 08 2000. [Online]. Available: http://www.legislation.gov.uk/ukpga/2000/23/pdfs/ukpga_20000023_en.pdf. [Accessed 11 05 2015].

[25] S. Moulton, "Hard Drive Recovery Part 3 at Toorcon," San Diego, 2006.

[26] Ace Laboratory, "Ace Laboratory," 2014. [Online]. Available: http://www.acelaboratory.com/. [Accessed 11 02 2015].

[27] B. Carrier, File System Forensic Analysis, Upper Saddle River: Pearson, 2005.

[28] sleuthkit.org, "sleuthkit.org," 2015. [Online]. Available: http://www.sleuthkit.org/autopsy/. [Accessed 23 04 2015].

[29] piriform, "piriform.com/recuva," 2015. [Online]. Available: http://www.piriform.com/recuva. [Accessed 23 04 2015].

[30] Download.com, "download.com," 08 04 2015. [Online]. Available: http://download.cnet.com/Recuva/3000-2242_4-10753287.html. [Accessed 23 04 2015].

[31] pcinspector.de, "pcinspector.de," 2015. [Online]. Available: http://www.pcinspector.de/. [Accessed 23 04 2015].

[32] nrteam, "nrteam.ru," 2014. [Online]. Available: http://www.nrteam.ru/en. [Accessed 20 02 2015].

[33] Flash-extractor.com, "flash-extractor.com," 2014. [Online]. Available: http://flash-extractor.com/. [Accessed 18 02 2015].

[34] Salvationdata, "Salvationdata," 2014. [Online]. Available: http://www.salvationdata.com/. [Accessed 20 04 2014].

[35] datarecovery.net, "datarecovery.net," 06 02 2015. [Online]. Available: http://www.datarecovery.net/pressreleases/pr_20150206.html. [Accessed 11 05 2015].

[36] Accessdata, "accessdata.com," Accessdata, 02 07 2014. [Online]. Available: http://accessdata.com/product-download/digital-forensics/ftk-imager-version-3.2.0. [Accessed 20 05 2015].

[37] ntfs.com, "ntfs.com," n.a.. [Online]. Available: http://ntfs.com/ntfs-system-files.htm. [Accessed 11 05 2015].

Table of figures