3D NAND memories

A low cost space radiation monitor?

Mathijs Van de Poel

Master thesis report
Faculty of Aerospace Engineering
Department of Space Engineering

3D NAND memories

A low cost space radiation monitor?

by

Mathijs Van de Poel

To obtain the degree of Master of Science
at the Delft University of Technology,
to be defended publicly on the 30th of October, 2018 at 11:00.

Student number:	4185145	
Thesis Committee:	Dr. Alessandra Menicucci	TU Delft, supervisor
	Dr. Angelo Cervone	TU Delft, chair
	Dr. Ir. Dominic Dirkx	TU Delft, external member
Version:	1.0	

Acknowledgments

The completion of this MSc. thesis marks the end of my engineering education at the Delft University of Technology. The continuous lessons in critical thinking and the cultivation of an engineering driven mindset have resulted in the work presented in this document.

Over the final 12 months, I was given the freedom to work on a project with a major hands on component. It required the necessary synthesis to interface different hardware and software components. I am grateful to be given this opportunity to add a couple of last desired technical skills to my engineering toolbox, and to work on a project with future potential. The right person to thank for this opportunity is Alessandra Menicucci, who proposed the topic with excitement and curiosity. Being my daily supervisor, I must thank you for the continued support throughout the project. When things (again) did not develop according to plan and I passed by your office mentioning my "niet hopen" motto, you always greeted me with a positive attitude and sent me off with good courage to continue. I would like to add that your work and ambitions I got to know over the year are a great source of inspiration that I will carry with me.

The necessary parts had to come together in order to reach the final results presented here. For this I need to extend my gratitude to a number of people. First to Alexey Lyashko, who provided the open source basis for the VHDL ONFI memory controller. The staff at the NFI, in particular Martien de Jongh, to provide the time, knowledge and tools to unsolder and reball the memory chips. Chris Verhoeven and the Delft Space Institute need to be thanked for their support by acquiring a logic analyzer, no working end product would have been possible without this tool. Gianluca Furano and Lucana Santos from the European Space Agency for providing the ZedBoard and availability for questions. Mehmet Uludağ, electronic engineer at the department of Space Systems Engineering, has played a key role to help with the design and manufacturing of the PCBs.

Finally, I would not have been able to complete my education and be where I am today, without the continuous support of my parents. No matter what path I chose these past years, with ventures taking me to opposites from the USA to Russia, I could count on your thrust in me. I must thank Timo Vebrugghe, besides being my good friend, for also taking up a role as head of transport. Lastly I would like to thank my colleagues in the master room, for the shared lunches, sharing of ideas and the social events outside of the work environment. In particular, Krishti Das and Arjen Jongschaap must be mentioned for their company during the late night and weekend work sessions.

Mathijs Van de Poel
Delft, October 2018

Summary

The thesis investigates the possible space application of a commercial 3D NAND memory as a radiation monitor. Literature has indicated that both traditional 2D and 3D flash memories are sensitive to ionizing radiation. Where the track of particle could be seen passing through the 3D architecture. A flash memory cell stores data with a collection of charge of which some is lost after a particle event. This loss can be measured as a shift in the threshold voltage. A relationship exists between the energy of a particle and the threshold voltage shift.

A methodology was developed to measure the threshold voltage of memory cells in a Micron 64 layer 3D NAND memory. By modifying the threshold voltage of boundary separating two logic states, a bit flip can be introduced. A page of memory cells will be reread multiple times, each time with a slightly increased or decreased voltage offset. It is determined that a measurement could be made with a theoretical error as low as 7.5 mV.

Based on literature on 3D NAND memory architectures and die and SEM images for the Micron device, the corresponding physical size of a memory cell, page and block were estimated. A memory block measures 7360 μm (depth) x 3.37 μm (height) x 9.5 μm (width). A set of SPENVIS simulations estimate that a single block may see less than one GCR particle event every ten days. If it is possible to detect protons, 10^4 events may be seen in an ISS orbit, up to 10^9 events for a MEO orbit.

In order to validate the proposed measurement methodology, a test setup was developed with which instructions can be executed on a memory. Requirements for this development were based on the ONFI specification, which standardizes the software and hardware interfaces with memories. The setup features a ZedBoard FPGA, where a VHDL memory controller generates the requested control signals. The interface between the FPGA and memory is provided through a custom designed PCB, which also powers the memory. A memory is placed in a socket such that different devices can be tested. Python scripts generate the correct sequence of commands to execute voltage threshold measurements.

A Micron SSD was acquired, from which the 64 layer 3D NAND memory packages were unsoldered. These memories were then used for testing. A first set of tests prove that it is possible to perform voltage threshold shift measurements. The general measurements show the expected voltage distributions. Interesting observations were made when investigating the measurements for a single cell: they do not always transition immediately from one logic state to another. A second interesting observation is that the same cell returned different voltage readings for consequent read cycles. These behaviors should be investigating in further research, as they will impact the errors in the measurement.

Concluding, a methodology is proposed and validated to measure the threshold voltage. A test setup was developed and proves the in house capability of making these measurements. It is possible to advance with the study of the radiation effects on the memories. Based on these results, further developments could lead to a future space radiation monitoring payload based on 3D NAND memories.

Contents

Chapter 1

Monitoring radiation with memories

The aerospace industry was traditionally a source of innovations, where novel technologies caused *spin-off* products and became commercially available. Nowadays, a different trend is seen. Years of computer component development driven be the high commercial demand have enabled us to carry phones in our pockets with data storage and processing power magnitudes larger than that the astronauts had available to land on the moon and return safely back to Earth. These advanced semiconductor devices are now being used for space applications, which is an example of *spin-in* and a characteristic of the current new space era.

This thesis project will follow the *spin in* trend, where a specific space application is proposed for a high tech computer component. The technology under investigation is a 3D NAND memory, which is suggested to be used to monitor space radiation. If space radiation particles can be detected and possible measured to some extend, a satellite payload could be built around a US$30 commercially available memory chip.

This chapter will provide an introduction to the subject of 3D NAND memories and its possible use as a space radiation monitor. 3D NAND memories are shortly described in Section 1.1, after which the space radiation environment and the effects on flash memories are discussed in Section 1.2 and 1.3. The research objective, questions and methods are stated in Section 1.4.The chapter concludes with scope and structure of the thesis in Sections 1.5 and 1.6.

1.1 3D NAND memory storage

Flash memory is a form of electrical data storage, where electrons or holes are stored inside of a semiconductor memory cell. The memory cell resembles a MOSFET transistor: when a voltage is applied to the gate, a current can pass between the source and drain terminals. However, the memory cell has an additional layer between the gate and the channel. The charge in this layer modulates the voltage needed on the gate in order for a current to pass. Single or multiple data bits can be encoded onto such a cell, depending on the present charge. This technology has no moving parts, and can be produced into high density memory solutions such as computer SSDs (solid-state drives).

Traditional flash memory devices have a planar array of memory cells, manufactured using the photolithography microfabrication process. Improvements in data storage density and decreasing cost have been based on down-scaling the memory cell. However, this had reached its lower limits when only a handful of electrons separated two different binary logic states. By switching to an etching manufacturing process, it is possible to create a 3D architecture of memory cells. This allows to further bring down the cost per bit, while having physically larger memory cells for more reliable data storage. The cells are connected along vertical pillars using the NAND layout, making them generally referred to as 3D NAND memories. A pillar may stack up to 96 cells vertically, while forming a grid on the wafer where a single string may be tens of thousands of pillars long. The memory cell technology is based on either floating gate or charge trap flash. A more in depth overview of the 3D NAND technology and architecture will be provided in Section 2.1.

3D NAND memories have been available since 2015 and are the preferred data storage solution for commercial computer platforms such as smart phones, tablets and laptops but have also been popular on the server market. Due to their mass production, they are available for prices as low as US$30 per chip. Four manufacturers are capable of producing the 3D NAND technology: Micron/Intel, Samsung, SK Hynix and Toshiba/WD. The device that will be studied in this thesis is the latest Micron 64-layer 3D NAND memory.

1.2 The space radiation environment

Space is a hostile environment compared to the living conditions present on Earth. Within our atmosphere, we are protected from temperatures hundreds of degrees above or below zero, the vacuum of space and the highly energetic radiation particles.

Once leaving the atmosphere, one may encounter protons, electrons and heavy ions attributed to one of three different sources: trapped radiation, cosmic rays and solar particle events. Protons and electrons are trapped in Earth's magnetic field in the Van Allen belts, extending from about 500 to 50,000 km. Protons with energies between several keV and hundreds of MeV may be encountered. Energies from a couple of eV to about 10 MeV are common for electrons. Cosmic rays include galactic cosmic rays (GCR) and high energy particles emitted from the sun. The GCR consist mostly out of protons (about 85%), 15% alpha particles and 1% high energy heavy ions (HZE) with particle energies in the MeV and GeV range. Solar particle events happen irregularly at the sun's surface, ejecting protons, electrons and some HZE particles. The frequency is dependent on the solar cycle. Proton energies may reach up to hundreds of MeV. The abundance of HZE is generally below the nominal GCR flux, but may for short periods of time be orders of magnitude higher than normal. The flux of different particles and energies is modulated by the location relative to the (Earth's) magnetic field and the solar cycle. [21]

1.3 Radiation effects on flash memories

A high;y energetic charged particle is able to alter the state of data. A very troubling example is the breaking of DNA strands damaging the instructions they have encoded for cell reproduction, which can lead to cancers or other diseases. More relevant within the scope of this thesis are bit flips introduced in semiconductor devices such as computer processors and memories, altering the outcome of a computation or stored data.

The radiation effects on computer memories have been the subject of many studies, where often a relation is sought between the energy and fluence of a particle and the number of bit flips seen in the memory (the so called cross section). More in depth research characterizes the underlying effects causing the bit flips: the basic principle found is that some of the stored charge in the cell encoding the binary data, is lost. The investigation of this phenomena is the subject of many research papers. The RREACT research group at the university of Padua is an active publisher on this topic. The actual underlying phenomena for the lost charge is still subject to debate, however many test campaigns have established relationships between the charge lost and the energy of the particles [10].

Measurements with the 2D memory flash technology established several interesting conclusions. One of the most interesting ones with the application of a radiation monitor in mind is the linear relationship between the linear energy transer (LET) of a particle and the threshold voltage shift of the memory cell. The 3D NAND memory technology has already undergone a radiation test campaign, for which the results are published in [1]. It has been shown that it is possible to see particle tracks passing through the memory by measuring the threshold voltage shift in each memory cell. However, no note was taken on the relationship between the LET of the particles and the threshold shift of a 3D memory cell.

One important comment to make on the literature describing the radiation effects through voltage threshold measurements, is that the methods to extract this data are not described in detail. Often "hidden test routines" are quoted that were obtained through a collaboration with the manufacturer.

1.4 Research objective, questions and method

The space environment is subjected to a flux of both low and highly energetic charged particles. These particles can affect the data stored in flash memory cells, as a striking particle causes charge loss from a memory cell. The effects of ionizing radiation have been characterized as a threshold voltage shift in both 2D and 3D NAND memories. Particle tracks could be visualized when passing through the 3D architecture, upsetting a large number of cells in comparison to the planar technology.

Therefore, a research objective can be formulated: ***can the 3D NAND memory technology be used as a space radiation monitor?*** Based on this objective, several research questions can be formulated to work towards achieving the objective:

1. **What methodology could be used to make threshold voltage measurements in a commercially available 3D NAND memory?**
2. **Can a sensitive volume within the memory be defined and quantified? Which and at what frequency would particle events take place in space?**
3. **What hardware and software is necessary to make such measurements?**

Answering these questions will provide the means to perform threshold voltage shift measurements, with future radiation test campaigns in mind. The computer architecture of a test setup development could serve as a breadboarding example to be advanced into a satellite payload.

The methods of investigation will partially be based on studying 3D NAND memory literature and device data sheets. Radiation simulations can be performed with tools such as SPENVIS or Geant4. To prove the proposed measurement methodology, 3D NAND memory chips will be acquired and read by a custom built test setup.

1.5 Thesis scope

The scope of this thesis will be limited to providing a first order confirmation that it is possible to make threshold voltage measurements. A focus will be made on developing a measurement methodology and validating the method in a test setup with real hardware. No actual radiation measurements will be performed. The study of the radiation effects will be limited to those of the memory cells, not the peripheral electronics.

1.6 Report structure

This chapter gave an introduction to the topic and presented the research objective, questions and methods. Chapter 2 will provide an overview of how the 3D NAND memory could be used as a radiation monitor. It also outlines the methodology to make the threshold voltage, based on the READ OFFSET function present in the Micron B17A 3D NAND memory. Chapter 3 investigates the internal layout of the memory based on die and SEM images, and literature on 3D NAND architectures. A series of SPENVIS simulations are ran to estimate then number of particle events in a single memory block. Chapter 4 is dedicated to the chip procurement and the full development of the test setup. Chapter 5 highlights how the separate components of the test setup work together and what troubleshooting methods are available. In chapter 6, the proposed measurement methodology to measure the threshold voltage is proven to work. Test data from the memory is taken and the data is analyzed. The final part of this thesis, chapter 7 covers the conclusions and recommendations. A number of recommendations are given concerning the future development and use of 3D NAND memories for a radiation monitor.

Chapter 2

Voltage threshold measurements with Micron 3D NAND memories

This chapter will attempt to answer the first research question of taking voltage threshold measurements by presenting a measurement methodology based on a specific Micron 3D NAND memory. The first sections will give a more in depth overview on the research objective of using the 3D NAND memory as a space radiation monitor. The 3D NAND technology is reviewed and the radiation effects on flash memory cells are put in perspective to their possible relation when used for radiation detection.

2.1 The 3D NAND flash architecture

Digital data storage devices retain a series of binary characters. A variety of technologies have been used to capture physical phenomena and convert them into digital forms. An early low tech physical example is the punch card. Where the presence of a hole, or lack thereof, in a stiff piece of paper indicates a one or zero. Optical and magnetic data storage has developed over the last centuries into a compact and popular solution. Optical data is encoded using reflective and non reflective areas on a spinning disk, where a sensor registers the reflected light from a laser diode. Well known example products are the CD and DVD. The direction of a magnetic field is indicative for the binary data stored in a magnetic data storage device. Transitions in the magnetic field of small areas on a rotating platter are measured as small currents by a sensor, positioned just above the surface of the disk. Computer hard drives are a traditional example using this technology. While rather compact data storage is possible, one common feature is the need for mechanically moving parts: the disk rotates at high speed, while a sensor moves radially to address the full disk surface.

The subject of this section is the currently popular data storage solution which is based on using electrons to store data. The products contains no moving parts and can be produced high density memory solutions using the latest microfabrication techniques: the 3D NAND flash memory.

2.1.1 Floating Gate and Charge Trap flash

Flash memory cells are based on one of two designs: floating gate or charge trap flash technology. Planar flash memories traditionally used floating gate technology. In the current 3D NAND market, most manufacturers opted for the charge trap flash technology. Micron, whose 3D NAND memory is the subject of this thesis, uses floating gate technology.

A Floating Gate MOSFET consists of a traditional MOSFET gate, to which an electrically conductive layer is added between the gate and channel. The number of electrons or holes stored in the cell determine the logic state of that memory cell. The FG is separated from the channel with a tunnel oxide layer (TOX), and through the interpoly dielectric layer (IPD) from the control gate. The tunnel oxide is made of a Silicon Dioxide (SiO2) layer, while the IPD is also called the ONO layer, has a Silicon Dioxide (SiO2) - Silicon Nitride (Si4N4) - Silicon Dioxide (SiO2) structure. the floating gate is usually a Phosphorus-doped Silicon.

The charge trap flash technology is as old as the FG technology when it was first presented in 1967. The general structure of the charge trap flash cell is similar to that of the FG cell, however the conductive polysilicon layer has been replaced with a Silicon Nitride film, trapping the stored electrons in the intrinsic defects.

A comparison between two cell structures for the 3D architecture is presented in Fig. 2.1. The charge trap flash layer, shown on the left, is deposited as one thin continuous band (purple) along the entire length of the channel. The floating gate technology, shown on the right, features individually insulated cells surrounding the main channel.

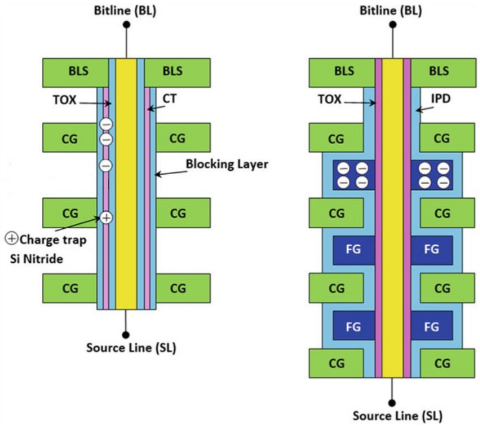


Figure 2.1: Diagram of memory cells in a 3D pillar architecture. (left) Charge Trap technology. (right) Floating gate technology [14]

3D NAND structure

Improvements in the 2D flash technology was based on scaling down the cell size, eventually limiting the number of the electrons stored in the floating gate. Downscaling had reached its limits, where only a couple of electrons separate two different logic levels in 20 nm cells[15]. Due to the constants wafer productions costs, producing and then stacking multiple 2D planar NAND dies is not a scalable and cost effective approach. In order to really fabricate in 3D, using the Z direction of the cell, the manufacturing process was changed from photo lithography to an etching process.

The 3D NAND technology memory technology is based on the basics of the 2D technology, the planar 2D string is rotated vertically, and the gate is fully wrapped around the channel. The process is depicted in Fig. 2.2 and is also referred to as vertical channel with horizontal gate technology. This design is referred to as the Bit-Cost Scalable technology (BiCS). Different memory cell technologies/architectures are presented in the literature (VRAT: Vertical Recess Array Transistor, VSAT: Vertical Stacked Array Transistor etc), but these have not (yet) made it into production.

The comparison could be made between a neighborhood, where the freestanding houses (2D) are replaced with high density apartment buildings (3D). Effectively increasing the population density while keeping the area constant.

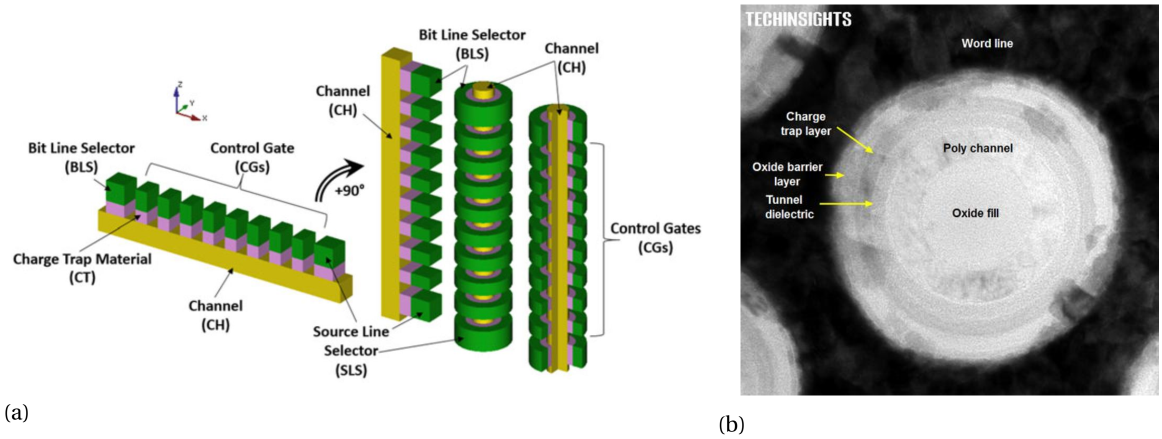


Figure 2.2: (a) Transition from 2D to 3D memory architecture. The channel becomes a vertical pillar with the cells wrapped around its core[14]. (b) Cross section of a Charge Trap flash cell in a Samsung 3D NAND device. Source: TechInsights

The pillars of memory cells are combined into large arrays making up the memory pages and blocks. To increase the memory density from a hardware standpoint, manufacturers have been focusing increasing the height of these pillars, adding more layers. The number of layers has steadily increased from 24 layers in 2015 (Samsung[19]) to 96 layers in 2018 (Intel/Micron[1]). It will be estimated in Section 3.1.4 that the Micron 3D NAND device under investigation in this thesis has a block size made up of an array of 48 by 37184 pillars, 64 layers high.

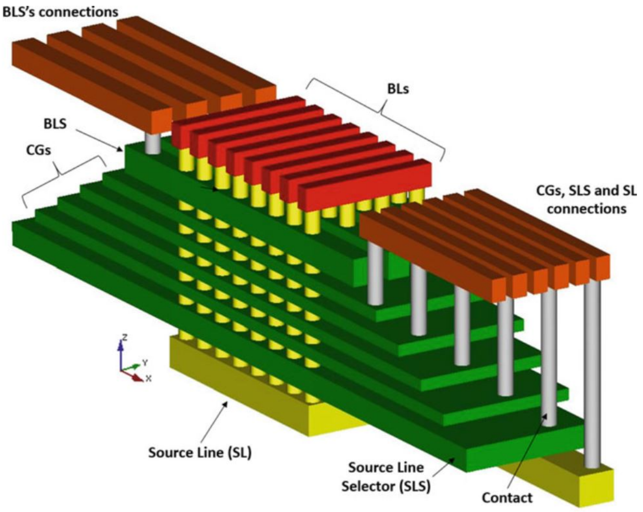


Figure 2.3: Bird's eye view of a 3D NAND memory architecture with necessary fan-out connection [14].

As shown in Fig. 2.3, more hardware is necessary to program, read and erase the memory cells. The extensions around the memory, to make the necessary connections, are also referred to as the fan-out region. The bitline on top of the array (BLs, red) selects which page will be addressed, the bitline select (BLS, BLS's connections, green orange) selects the pillar within that page, finally the layer is selected by the control gate (CGs, CGs connections, green and orange). An image from a 32 layer 3D NAND Samsung memory, Fig. 2.4 shows that about an additional 20 μm is needed, quite substantial when in comparison to the height of the memory stack.

[1] https://newsroom.intel.com/news-releases/micron-intel-extend-their-leadership-3d-nand-flash-memory/

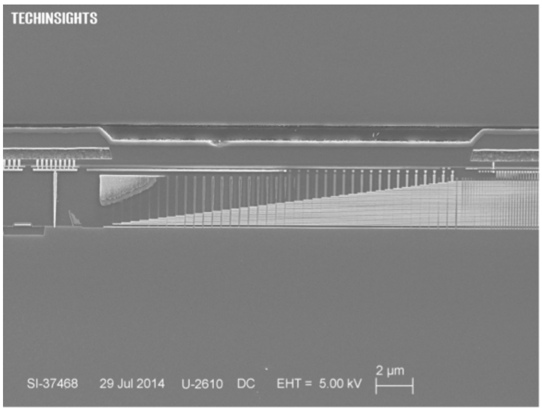


Figure 2.4: The wordline staircase in a second generation 32 layer 3D NAND Samsung memory. Source: TechInsights.

Data storage through electrons

The charge content of a cell is measured as the threshold voltage: the voltage needed to pass a certain current through that cell. With accurate programming control circuits, the threshold voltage can be programmed with increasing accuracy, allowing the threshold voltage to be programmed to a multitude of voltage ranges. This allows to encode multiple bits within a single memory cell. The following names are generally used: SLC, MLC, TLC, QLC, storing 1, 2, 3 and 4 bits.

The stored voltages within a logic level display a more or less Gaussian distribution, when measured for a large amount of cells. The programmed threshold voltage must lay within the upper and lower boundary to represent their correct value. Increasing the number of bits programmed onto a single cell allows for more dense memory storage, however the different levels are closer and closer apart. More sensitive readout electronics are required, and the cells are more sensitive to bit errors due to radiation.

The different logic levels (L0-L7) and their stored logic (000-111) for a TLC cell is visualized in Fig. 2.5. A total of 8 levels encode 3 bits (2^3). The relationship between the logic level and their stored logic is initially unknown, the presented data is representative for the values found for the Micron 3D NAND memory under investigation in this thesis project.

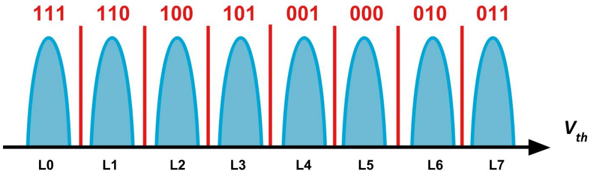


Figure 2.5: Logic levels for TLC memory and an example order for the stored logic in the lower, middle and upper page

2.2 DUT: Micron 3D NAND

The Device Under Test (or DUT) in this thesis is the Micron 64 layer B17A 3D NAND memory. The selection of this memory was based on the presence of the *READ OFFSET* function, allowing for the threshold voltage measurement.

2.2.1 Market Overview

Four main manufacturers are currently able to produce 3D NAND flash memories. Some manufacturers have made partnerships in order to share the large development cost. Table 2.1 indicates which manufacturers are producing what 3D NAND memories at the time of writing (Fall 2018).

Table 2.1: Overview of the four 3D NAND manufacturers.

Manufacturer	Technology	Layers	Announced
Samsung	Charge Trap	96	July 2018
Toshiba/WD	Charge Trap	96	July 2018
SK Hynix	Charge Trap	72	February 2018
Intel/Micron	Floating Gate	96	May 2018

2.2.2 Micron 3D NAND technology

Intel/Micron collaborate on the production of Floating Gate based 3D NAND memories. Their first production 32L memories were presented in 2015, featuring both MLC and TLC storage. Micron has provided a basic insight by publishing about the main technical features[18]. The second generation was announced in 2016, this time featuring TLC technology and 64 layers [22].

According to the Micron part catalog, the 3D NAND memories are available in 48 pin TSOP and 132 pad BGA packages.[2] One is able to register and download the data sheets for the different memories.

2.2.3 Micron B17A FortisFlash

The latest 3D NAND memory technology described in the Micron part catalog is sold as B17A FortisFlash. It features a 64 layer, TLC technology and the data sheet indicates that it is available in 512 Gb, 1 Tb, 2 Tb, 4 Tb and 8 Tb sizes. It is sold in a 132-BGA package. It uses the Open NAND Flash Interface (ONFI) as communication protocol, which is an important feature explained in detail in Section 4.2.

The key feature for which this memory was selected as a potential interesting device to use as a radiation monitor is the READ OFFSET function, which will enable the measurement of the individual memory cell threshold voltage. A detailed description and how this function will be used to measure radiation effects is given in Section 2.4.

The Micron NAND part numbers for this device are MT29F512G08EBHAF, MT29F1T08EEHAF, MT29F2T08EMHAF, MT29F4T08EUHAF and MT29F8T08EWHAF.

[2] `micron.com/products/nand-flash/3d-nand/3d-nand-part-catalog#/`

2.3 Radiation effects on flash memories

The radiation effects on computer memories have been the subject of many research papers. It is attempted to both characterize the effects and to understand the underlying physical phenomenon. Research on this first is quite extensive and conclusive, which is not the case for the latter.

This section will give an introduction to conclusions from the literature on the effects of ionized particles on flash memory cells. Whilst the main measurement is the threshold voltage shift, more detailed conclusions van be made. These are summarized with an indication how this impacts the use of the flash memory technology for radiation monitoring purposes.

2.3.1 Threshold voltage shift

Charged particles interact with the matter through which they pass. The primary interaction is through the coulomb forces between their positive charge and the negative charge of the absorber material. As the particle is slowed down or stopped due to the coulomb interactions, they deposit energy in the absorber material. The deposited energy results in the creation of electron-hole pairs. In SiO_2, one pair is created for every 17 eV [12]. Although the exact underlying physical phenomena is subject to debate, the pair production in the insulating layers surrounding the floating gate or charge trap layer cause the loss of stored charge from the memory cell [10].

When a floating gate or charge trap memory cells is hit by a charged particle, stored charge is lost, effectively decreasing the threshold voltage of a programmed cell. This change is quantified as the voltage threshold shift ΔV_{th}. An example measurement of the radiation effects on memory programmed to a logic level encoded to 7 V, is shown in Fig. 2.6. After irradiation with 276.7 MeV Iodine ions, effective LET of 62 MeV/cm^{2}mg, a secondary distribution appears, with a peak shifter towards a lower voltage and lower than the original distribution. This distribution is representative for the cells that have suffered a threshold voltage shift due to a particle strike.

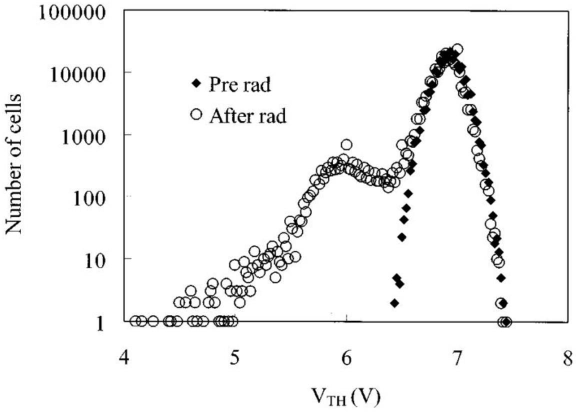


Figure 2.6: Voltage threshold distribution before and after irradiation with 2 x10^7 Iodine ion/cm. The typical change in curve can be observed: a secondary peak appears at a lower voltage with less cells [7].

The effect can be studied for a large number of cells, such as presented above, or for individual memory cells. An example is given in Fig 2.7 for a TLC memory cell . The green arrow indicates the effect of the particle strike. Originally programmed to L7, the memory cell would now be read as L6. The exact shift in the threshold voltage would be measured and quantified as the voltage threshold shift ΔV_{th} specific to that cell.

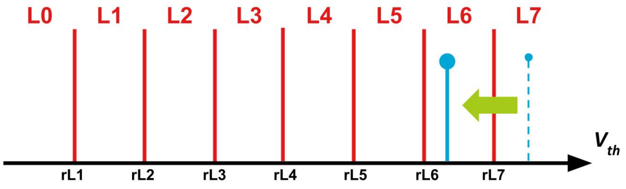


Figure 2.7: Due to a particle strike, a memory cell loses some of its stored charge, decreasing the threshold voltage of that memory cell. Shown here is a TLC memory cell initially programmed to L7, after the particle strike, it would be read as L6: a bit flip has occurred.

2.3.2 Effects on 2D memories

Extensive research has been done on planar flash technology. One leading research group publishing on the topic is the RREACT group at the University of Padua. A list of conclusions by papers published on the radiation effects on 2D flash memories is summarized below. A comment is added in italic what the interpretation of this effect could be when deploying a flash memory as a radiation monitor.

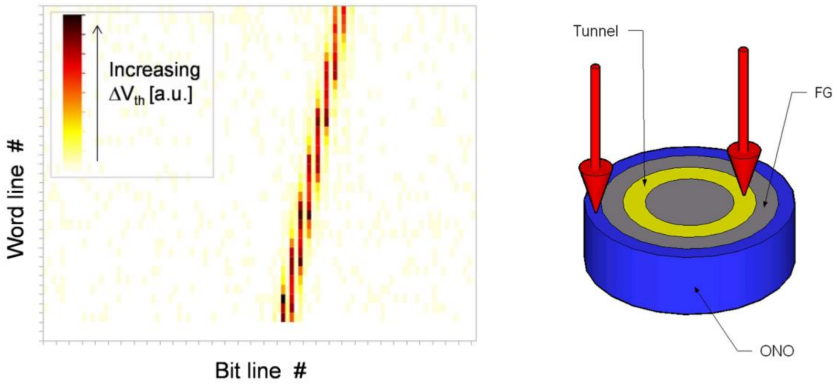
- The ΔV_{th} increases with the programmed V_{th}, and that the cells shift towards a neutral state (no holes/-electrons injected into the floating gate) [7]. *Indicating that ideally, to see the highest shift (most sensitive), the memory shall be programmed to the highest logic level.*
- The ΔV_{th} increases with the LET of the particle. This relationship is of a linear nature[4]. *A characterization campaign can determine the relationship/coeffient for a particular device. One can use the* ΔV_{th} *to estimate the LET of the passing particle.*
- The cell size also has an influence on the ΔV_{th} and a different LET/ΔV_{th} relationship is found between FG and CT technology.[4]. *The* ΔV_{th} *is device and technology specific.*
- The influence of the angle of incidence of the ionizing radiation was investigated in [11]. The Single Event Upset (SEU) cross section varies with the incidence angle, however no general trend can be described for difference angles or devices. *The sensitivity of the device may be angle dependent.*
- Cells may be permanently damaged after a heavy ion strike, causing to continuously loose charge, even after reprogramming [13][5].*Keep track of those cell that have been hit before. If charge is constantly being lost, discard cell as sensitive node.*
- Proton irradiation tests show that a general ΔV_{th} are due to TID effects on the memory cells, the higher energy protons cause a threshold shift of the programmed cells. Recoil atoms may scatter and cause MBU. High LET recoil atoms can be created at higher proton energies [6][2]. *High energy protons may cause a* ΔV_{th}, *in 3D architectures, a more dispersed pattern may be seen due to recoil atoms.*

2.3.3 Effects on 3D memories

Unlike the 2D flash memories, the 3D NAND memories only appeared on the market in 2015. Although the fundamental principle for data storage has remained the same, the physical design of the memory cells and the full architecture is completely different. Two research papers were published on radiation effects on 3D NAND memories. The paper published by D. Chen in 2017 evaluates the susceptibility to bit flips, in comparison to planar architectures [8]. A paper from M. Bagatin in 2017 (RREACT group in Padua) studied the radiation effects in greater detail with threshold voltage shift measurements and simulations [1].

The DUT by Chen was a Hynix 3D NAND memory, using charge trap flash technology. No major interesting conclusions can be taken from this paper: different data patterns (hex values FF, 00 55 and AA) are written to the memory in SLC and MLC mode. However, no assessment is made to which actual voltage level the cells are programmed. Highest sensitivity would be related to not just the data pattern by also the order in which pages are programmed (the highest voltage level might be reached by programming the lower page to 00h and the upper page to FFh).

More interesting results are published by Bagatin. A Micron 32 layer 3D NAND memory (the older version of the DUT studied in this project) was subjected to irradiation by a range of different heavy ion particles. The threshold voltage of the memory cells is measured before and after irradiation. It must be noted that the methodology for the measurements is not published and that the measured threshold voltages are not quantified with the actual values. The paper shows that it is possible to see the track of a particle passing through the 3D structure, Fig. 2.8a. The paper comments on the dispersion of the track, likely attributed to delta electrons, without futher investigation. One major conclusion is that the Micron 3D NAND memory cell architecture has two sensitive volumes. Besides the ONO layer (as in the planar technology), the tunnel oxide layer also contributes to a path through which charge can be lost, Fig. 2.8b. With the use of the 3D NAND memory as a radiation monitor in mind: this has a consequence that it is possible for a particle to strike either one or both of the sensitive volumes, which might result in a less clear relationship between the LET of a passing particle and the voltage threshold shift. And indeed, where a linear relationship between ΔV_{th} and LET was quoted for different devices in planar technology, the paper takes no notes on this relationship for the 3D technology.



(a)

(b)

Figure 2.8: (a) Example of a particle passing the cells in a 3D architecture of a Micron 3D NAND memory. (b) Careful data analysis revealed two identifiable sensitive volumes for a 3D NAND memory cell: the ONO layer and the tunnel oxide layer [1] .

2.4 Methodology to measure threshold voltage shift

Section 2.3 indicated the relationship between the LET of a ionized particle and the threshold voltage of the 3D NAND flash memory cells. The literature indicates the results of the measurements, which are key in taking the presented conclusion, but the methodology behind obtaining these measurements from the flash memory are lacking.

This section will present an original methodology for making threshold voltage measurements of memory cells in the Micron B17A memory. It is the necessary first step towards using the 3D NAND memory technology for radiation monitoring purposes. The key enabler is the READ OFFSET function of the device.

2.4.1 The READ OFFSET function

As a short introduction, a TLC memory cell holds 3 bits of data. To modulate this data onto a single memory cell, it is required to program the memory cell to one of 8 (2^3) *logic levels*. A certain threshold voltage is read as a certain logic level if is it located between two *reference levels*. For the example in Fig. 2.9: the memory cell is programmed to logic level 6 (*L6*), which is located between the lower and upper reference levels *rL6* and *rL7*.

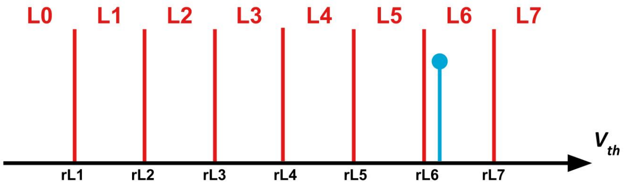


Figure 2.9: A memory cell programmed to *L6*. The threshold voltage is located between reference level 6 and reference level 7 (*rL6* and *rL7*).

The presented methodology to measure the threshold voltages is based on the READ OFFSET function available specifically to the Micron 3D NAND memory. The READ OFFSET function enables to set an offset voltage to each individual reference level, increasing or decreasing the nominal level with an offset between -960 mV and +952.5 mV with a minimum increment of 7.5 mV. Its full functionality is described in the Micron B17A datasheet. The READ OFFSET feature is set in the memory device by issuing a SET FEATURES command. These are manufacturer-specific commands within the ONFI framework described in Section 4.2. Fig. 2.10 indicates the effect of issuing a READ OFFSET with a positive threshold shift: the reference levels have shifted upwards. The logic levels are thus programmed and read differently. Where the data was stored in *L6* before, that same cell would now be read as *L5*.

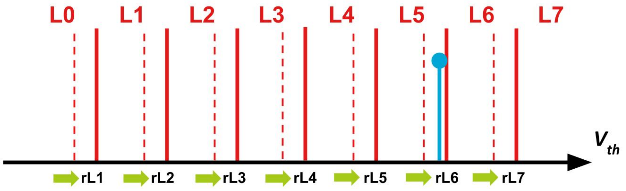


Figure 2.10: Applying a positive READ OFFSET command to all the reference levels will shift the original logic levels towards higher threshold voltages.

2.4.2 Determination of the threshold voltage shift using the READ OFFSET function

As concluded before, the threshold voltage of a memory cell decreases after a particle strike. The largest threshold voltage shift will occur at the highest potentials, therefore the memory cells would be initially programmed to the highest voltage level: *L7*. An example for such a particle event is shown in Fig 2.11, where the memory is initially programmed to *L7*, however it would be read as *L6* after partially losing stored charge. This scenario would result in a bit flip and could be used as a rough measurement, indicating a particle strike has occured. Yet, if one can determine the initial and final threshold voltage, the actual threshold voltage shift can be determined, resulting in a higher accurate measurement.

A method is proposed where the threshold voltage of the memory cell is determined relative to a reference level. The measurement of the initial voltage level for the example would result in a positive voltage offset compared to rL7, the second measurement after irradiation would return a negative offset. The sum of both values is the voltage threshold shift ΔV_{th}.

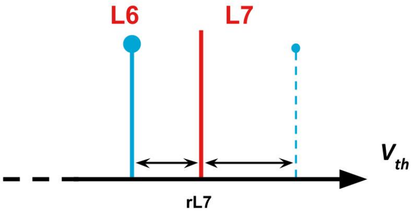


Figure 2.11: The measurement of the threshold voltage shift will be in relation to reference level rL7. The initial voltage and the final offset voltage will be summed.

By steadily increasing the reference level of *rL7* using the READ OFFSET function, a bit flip will be introduced once the actual programmed threshold voltage is passed. The originally programmed cell will now be read as *L6* instead of *L7*. The offset voltage at which this change occurs will be the initial threshold voltage measurement. This method is repeated before and after the radiation event to determine the voltage threshold shift.

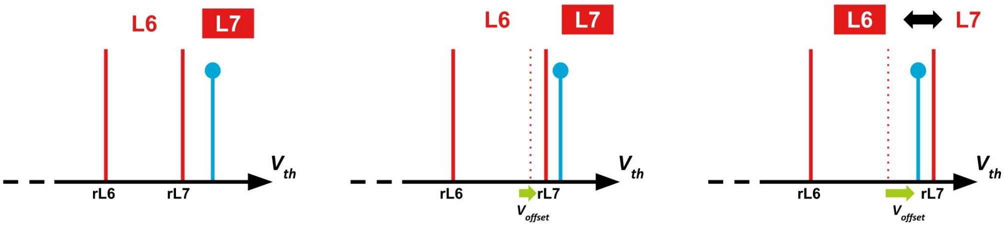


Figure 2.12: Measuring the programmed cell voltage relative to *rL7* by increasing the offset voltage using the READ OFFSET function. When a bit flip is seen in the logic level, the reference level has passed the programmed voltage.

In practice, the offset voltage can be set in increments of 7.5 mV, theoretically determining the relative offset voltage with an error of 3.75 mV and the threshold voltage shift with an error of 5.3 mV, as shown in Section 2.4.4. The results from actual voltage threshold measurements presented in Section 6.3 will show that this will not always be possible and that a larger error will be present.

Because a total of three "virtual" pages will be programmed onto a single "physical" memory page, only the virtual memory page in which the bit flip is expected to occur, needs to be read (for example: *L7* is 011 and *L6* is 010, hence the bit flip will occur on the upper page). The practical implementation will be based on repeated measurements of that same memory page, for which each time, the offset voltage is increased or decreased. This will result in a scan of the memory page, with up to 256 read cycles. When a full memory block (2304 pages for the DUT) is set as the sensitive volume, around 768 pages will be read. This data shall be post processed, looking for each memory cell at what offset voltage the bit flip took place.

2.4.3 Proposed test routine

An overview of the two-step process to measure the threshold voltage shift for a full memory block is presented below. It includes the all the ONFI commands as described in the Micron B17A datasheet that will be sent from a controller to the memory. The command descriptions in bold include the hex commands that will be latched by the memory. This will be discussed in more detail in Section 4.2.

The left flowchart in Fig. 2.13 programs the block to the highest voltage level and determines the initial threshold voltage offset for all the memory cells in a single block. The right flowchart indicates the second set of instructions that will be sent by a memory controller after irradiation, to determine the final threshold voltage offset from which the voltage threshold shift ΔV_{th} is determined.

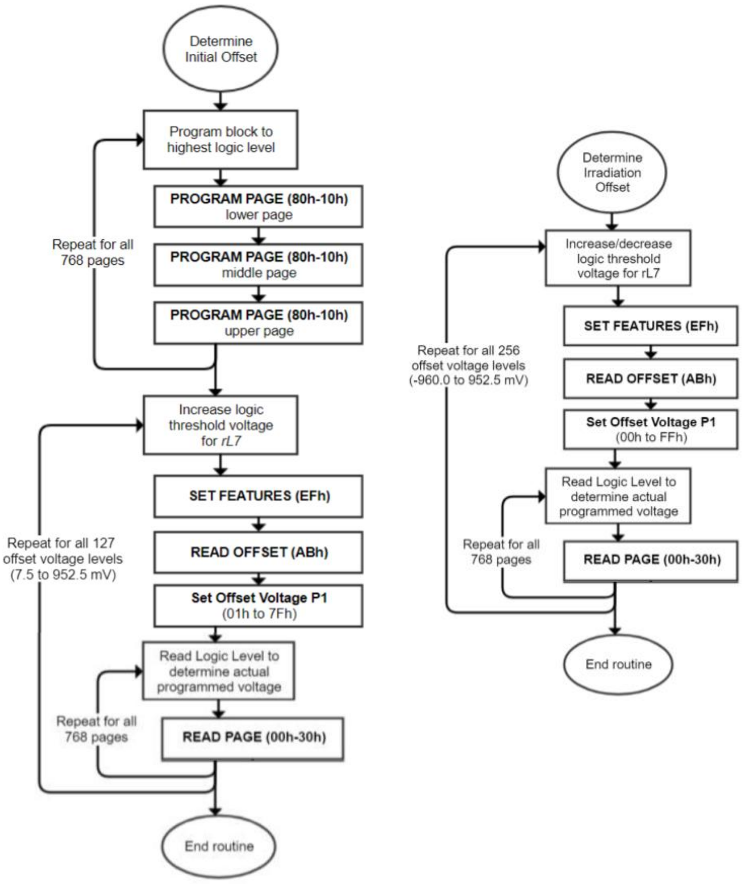

Determine Initial Offset
Program block to highest logic level
PROGRAM PAGE (80h-10h) lower page
PROGRAM PAGE (80h-10h) middle page
PROGRAM PAGE (80h-10h) upper page
Repeat for all 768 pages
Increase logic threshold voltage for rL7
SET FEATURES (EFh)
READ OFFSET (ABh)
Set Offset Voltage P1 (01h to 7Fh)
Repeat for all 127 offset voltage levels (7.5 to 952.5 mV)
Read Logic Level to determine actual programmed voltage
READ PAGE (00h-30h)
Repeat for all 768 pages
End routine
Determine Irradiation Offset
Increase/decrease logic threshold voltage for rL7
SET FEATURES (EFh)
READ OFFSET (ABh)
Set Offset Voltage P1 (00h to FFh)
Repeat for all 256 offset voltage levels (-960.0 to 952.5 mV)
Read Logic Level to determine actual programmed voltage
READ PAGE (00h-30h)
Repeat for all 768 pages
End routine

Figure 2.13: (a) Flowchart of the test routine to determine the initial programmed voltage of the memory cell. The memory instructions as described in the data sheet are shown with their hex command set. (b) Flowchart of the test routine to determine the threshold voltage after irradiation.

Some side notes are made to the presented flowcharts, several changes can be made and optimized for the specific purpose and the results from a characterization campaign with the memory.

- **Program block to highest logic level**
  One has to identify what binary combination of the data on the lower, middle and upper pages programs the memory cell to *L7*. Additionally, the memory is typically not programmed per memory cell or page, but in an order from the first to the last page (page 0 to 2303). A mapping scheme needs to be made, where for each page address, it is determined whether it is a lower, middle or upper page and hence whether the data programmed onto the page should be all 0s or 1s.

- **Increase logic threshold voltage for *rL7* (initial offset)**
  Using the READ OFFSET function, the offset voltage for each memory cell will be determined. The READ OFFSET function shall be issued as described in the Micron B17A datasheet to set an offset voltage to *rL7*. The flowchart proposes 127 offset cycles between 7.5 and 952.5 mV. Only a positive scan shall normally be issued, as the memory is programmed to *L7*. An upper limit can be determined to optimize the procedure, all the cells are likely to be programmed to a voltage below the maximum offset of 952.5 mV. Furthermore, the increment of 7.5 mV can be increased to decrease the number of cycles needed to scan a single page.

- **Read Logic Level to determine actual programmed voltage**
  The only page that should be repeatedly read is the page (lower, middle or upper) in which the bit flip shall occur. This must be investigated and is likely technology/manufacturer-specific. For the Micron B17A memory, the bit flip will occur on the upper page.

- **Increase logic threshold voltage for *rL7* (final offset)**
  It is proposed to read the pages with an offset between -960 and +952.5 mV. This is theoretical maximum scan that can be set to *rL7*, for which again both the lower and upper bounds can be optimized, as well as the increments between two read cycles. The proposed scenario might not be valid where really high LET particles cause a shift surpassing the *L6* logic level. If a bit flip would occur from *L7* to for example *L5* or *L4*, a different reference level should be probed. Additionally, the distance between two logic levels shall be determined.

2.4.4 Error analysis

When using the 3D NAND memory as a radiation monitor, the particle is indirectly detected by the charge lost from the memory cells. If one can determine with great accuracy how much charge is lost from the cell, a better estimation can be made as to which particle had passed. Therefore, the detector performance could be quantified as: *how well can we read the voltage threshold shift for each memory cell*? As part of this assessment, the read error and the measurement error will be discussed.

An error will be introduced into the initial and final voltage threshold measurement, after which the overall shift is determined through subtracting the second value from the first value. The error in the overall threshold shift measurement can be computed according to the following equations.

$$Z = X - Y \tag{2.1}$$

$$\Delta Z = \sqrt{\Delta X^2 + \Delta Y^2} \tag{2.2}$$

Read error

The first error source in a measurements is due to the lower bound increment to measure the threshold voltage. The READ OFFSET function is limited to generating voltage offsets of 7.5 mV. Therefore, the lowest possible reading error ΔX is 0.5*7.5 = 3.75 mV, visualized in Fig. 2.14.

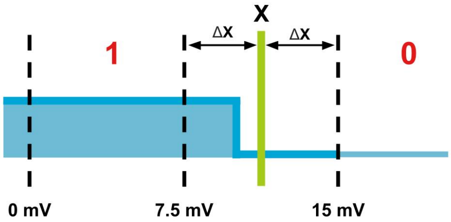


Figure 2.14: The state of the memory cell will be read at 0 mV, 7.5 mV and 15 mV, where the bit flip will be registered at 15 mV. The transition in the logic state, from 1 to 0, will have occurred somewhere between 7.5 and 15 mV. The measurement should then read 11.25 mV ± 3.75 mV. The reading error, Δ x, is half of the increment.

As the error will be present in the first and second measurement, the final error ΔZ can be computed to be 5.3 mV. This would be the error in the best case scenario possible with an offset increment of 7.5 mV.

$$\Delta Z = \sqrt{2 \cdot \Delta X^2} \tag{2.3}$$

$$\Delta Z = \sqrt{2} \cdot \Delta X \tag{2.4}$$

$$\Delta Z = \sqrt{2} \cdot 3.75 = 5.3\text{mV} \tag{2.5}$$

Measurement error

An initial set of voltage threshold measurements with the memory in Section 6.3 indicate that two random errors may be introduced. A first error source is based on the fact that the transition from one state to another may take several offset measurements before a stable new state is reached (later referred to as the flickering period). A second source is caused by reading different cell threshold voltage between multiple read cycles: even though a threshold voltage is measured, a repeated measurement indicates a different value.

An example for the second error source is given in Fig. 2.15. An imaginary single memory cell is read three times, each time returning a different threshold voltage value: the first bit flip would be seen at 15 mV, the second at 7.5 mV and the third at 30 mV. The average measured value and the error for this example would be 15 mV ± 15 mV. This error in the threshold voltage measurement is larger than the reading error due to the measurement resolution of 7.5 mV.

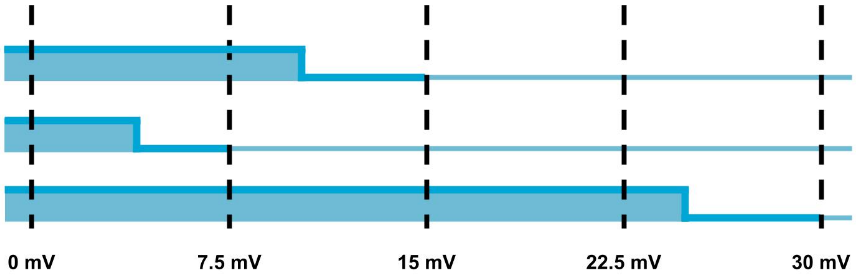


Figure 2.15: The measurement error for a cell may exceed the reading error.

If an estimate of the measurement error is made, the readout cycles of the pages can be optimized by increasing the voltage offset increment between two read cycles. For the example shown above, the page could be read every 30 mV instead of every 7.5 mV and still yield the same results, seen in Fig. 2.16. In this case, the measurement error is matched with the reading error.

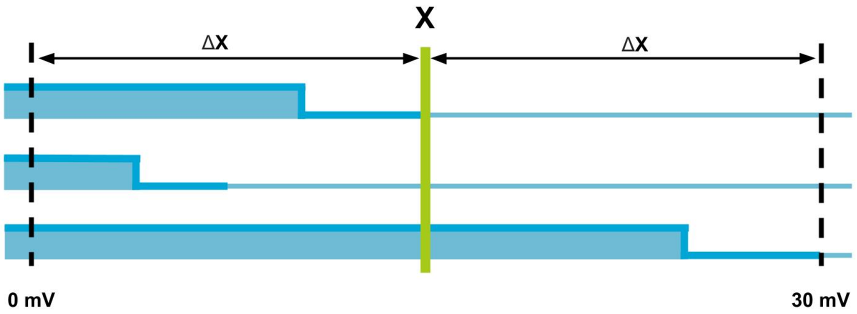


Figure 2.16: Matching the reading error with the measurement error to optimize the detector readouts.

2.4.5 Single particle event: needle in a haystack

A particle event will cause a signature in a memory cell: charge will be lost, which will be measured with the proposed method. The particle is likely to cause a track of events throughout the stack of 64 layers with memory cells. A threshold voltage shift will likely be measured in at least 64 memory cells. 64 upset memory cells is a small number, compared to the total amount of cells in a single memory block. One memory page contains 148,736 memory cells. When measuring 744 pages within a single block , a total number of 110,659,584 memory cells will be assessed. One particle event would upset 0.00067 % of the cells.

It will be interesting to see whether the noise of erroneous cells will stay below the signature of 64 hit cells. With the physical locations relative of each memory to each other known (hypothetically), only the cells showing a track/pattern can be assumed to be hit by a particle. Upsets in individual memory cells outside a funnel around a track could be discarded.

Chapter 3

Sensitive volumes and Radiation simulations

The third chapter will attempt to provide an answer to the second research question on sensitive volumes and the frequency as which particles will encountered in space. The memory has a certain internal organization, corresponding to a physical volume on the memory die. An attempt is made to estimate how the memory is physically structured inside of the die. A specific volume of the memory can be used as sensitive volume, in which a group of cells will be fully charged and be read to check if a particle event took place. Based on the estimated area of a single memory block as sensitive volume, a radiation simulation using the SPENVIS tool will answer the question to how many particle events will be encountered for different Earth orbits.

3.1 Investigation into the internal architecture

The Micron B17A datasheet indicates the internal organization based on page and block: a page contains 18,592 bytes, a block holds 2304 pages. Depending on the storage capacity of the memory, between 2016 and 32,256 blocks are present. The page and block units are addressed by a memory controller in order write and read stored data to a specific location. When using the memory as a radiation monitor, it is desired to also know the physical location of these pages and blocks. Ideally, one has knowledge of the exact or relative location of every single memory cell that can be addressed in the memory. Unfortunately, this mapping data is not released to the public. It is likely to be available upon close contact with Micron and signing a non-disclosure agreement.

In this section, an attempt will be made to estimate the physical layout of the memory, based on the memory organization provided by Micron, images of the memory available on the internet and literature on 3D NAND memory architectures.

3.1.1 Size of a memory cell

The smallest building block of the memory is a single memory cell. The 3D architecture contains a large planar array of pillars, each pillar stacking up to 96 memory cells high. By using the chip surface area and the height of the total number of layers, the size for a single memory cell is estimated.

Memory cell side length

A single 3D NAND memory cell has a cylindrical shape, assumed to be fitting within a rectangular box. It is attempted to estimate the square ground plane size of this box.

Information about the surface area of the chips is provided by TechInsights, a blog post compares the old Micron 32L technology with the new 64L technology [23]. Fig. 3.1 makes a side by side comparison, and quotes the surface areas of both versions. It is quoted that the 256 Gb 64L chip has a die size of 58.18 mm^2 (7.43 by 7.83 mm^2) and memory array efficiency of 89.8 % is computed, and a memory density per die of 4.40Gb/mm^2.

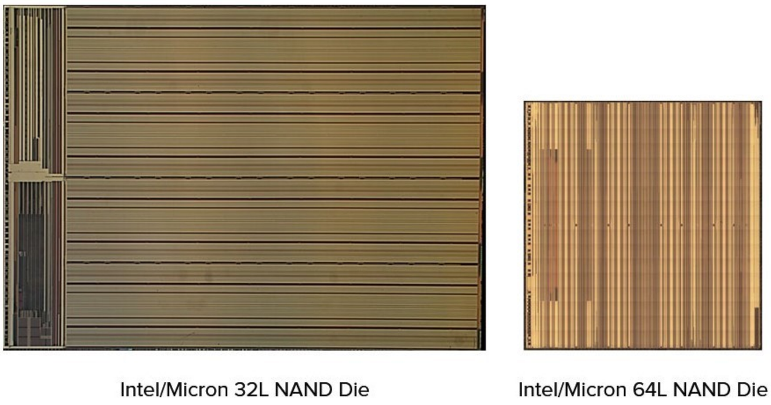


Figure 3.1: The Intel/Micron 3D NAND dies. The 32L has a die area of 168.2 mm^2 (15.36 mm x 10.95 mm) and the 64L version has a die area of 58.18 mm^2 (7.43 mm x 7.83 mm). Source: TechInsights

The memory cell density for the entire stack (64 layers) and each layer can be computer, taking into account the data from TechInsights and the knowledge of TLC operations.

$$\frac{4.4\text{Gb/mm}^2}{3\text{ bit/cell (TLC)}\cdot 89.8\%} = 1.633 * 10^6\text{ cell/mm}^2 \qquad \text{(64 layers)} \tag{3.1}$$

$$\frac{1.633 * 10^6\text{ cell/mm}^2}{64\text{ layers}} = 25.5 * 10^6\text{ cell/mm}^2 \qquad \text{(Single layer)} \tag{3.2}$$

Subsequently, the area for a single memory cell can be estimated. The inversion of the single layer memory cell density provides the area in mm^2 per cell.

$$\frac{1}{25.5 * 10^6\text{ cell/mm}^2} = 39.3 * 10^{-9}\text{ mm}^2\text{/cell} \qquad \text{(Area single cell)} \tag{3.3}$$

From here, the approximate side length for a single memory cell can be determined, assuming a square area.

$$\sqrt{39.3 * 10^{-9}}\cdot 1000\ \mu\text{m/mm} = 0.198\ \mu\text{m/side} \qquad \text{(Side length single cell)} \tag{3.4}$$

Memory cell height

The height of the cell is estimated using scanning electron microscope (SEM) images published by Micron and TechInsights. A first important image is Fig. 3.2, posted in a blog post by TechInsights where the 32L Samsung technology is compared to that of a Micron 32L NAND memory (note, this blog post was later removed from their website). The key takeaway from this figure is the 1 μm scale shown on the bottom side of the Micron SEM image. Based on the scale, the height of the full Micron stack can be measured at approximately 2 μm. Counting a total of 38 layers (extra layers are added for control purposes), the height of a single layer can be estimated at 52.6 nm.

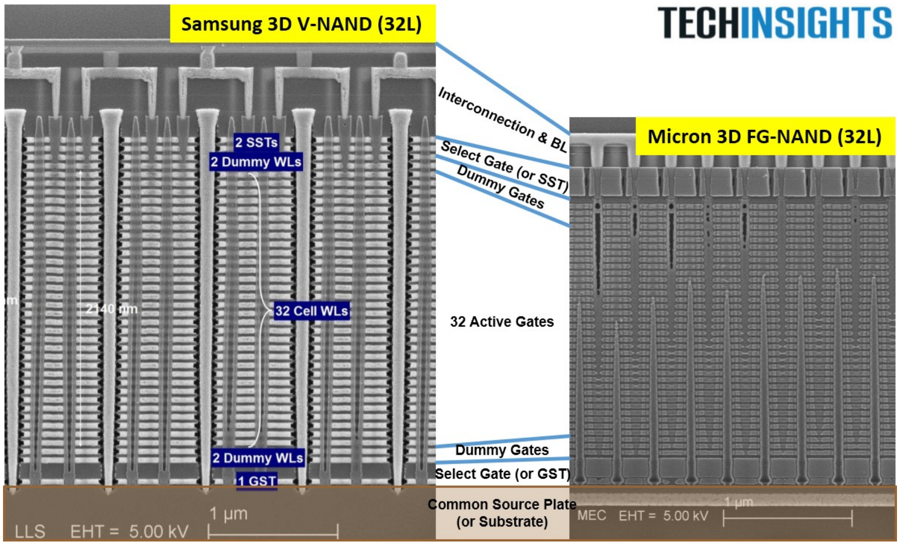


Figure 3.2: SEM image comparing the internal structure of a Samsung and Micron 32L 3D NAND memory [23].

Fig. 3.3 is a SEM image of the Micron 64L technology, published by Micron on their website. It looks as if two 32L dies might have been stacked on one another. However, the quality is not high enough to deduct any information. Making the assumption that no major changes to the architecture have been introduced between the 32L and 64L technology, the height of a single layer will remain at an estimated 52.6 nm.

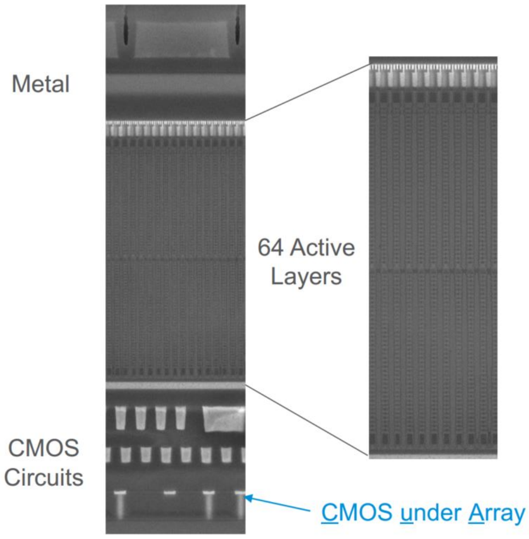


Figure 3.3: SEM image of the latest Micron 64L NAND technology. Source: Micron Website.

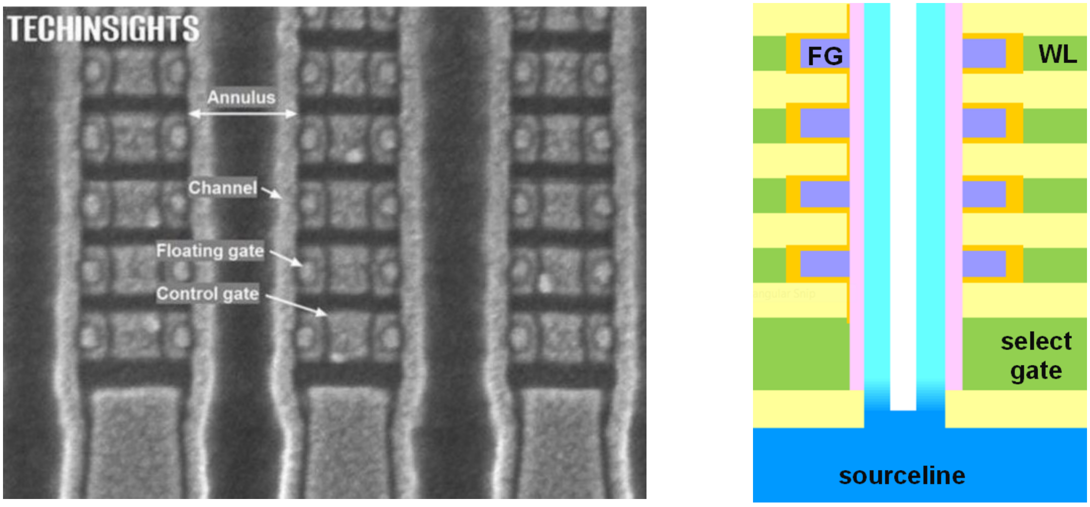
Concluding: the size of single memory cell is estimated at 198 x 198 x 52.6 nm^3.

3.1.2 Memory cell geometry

An attempt is made to describe the individual cell architecture.

Multiple different architectures are proposed in which Floating Gate based designs can be produced [14]. However, none of the presented technologies are exactly similar to the Micron FG cell geometry shown by a SEM image by TechInsights for the 32L technology in Fig. 3.4a.

The schematic in Fig. 3.4b mostly agrees with the cell design in the SEM image. However, it seems that the annotated *Channel* by TechInsights is more likely to be the *Tunnel Oxide*. If the light gray area is indeed the channel as indicated, the tunnel oxide would be less than 1 nm thick which is highly unlikely. A more likely architecture would be a structure where the light gray area is the tunnel oxide, and the dark gray area is the channel.



(a)

(b)

Figure 3.4: (a) SEM image of individual FG memory cells in a Micron 3D NAND memory [23].(b) Proposed FG cell architecture matching that seen in the Micron memory. Source: ChipWorks.

It was estimated that an individual memory cell has a width of 198 nm and a height of 52.6 nm. Based on these sizes, an estimation can be made for the different elements in the cell structure. In the TechInsights SEM image in Fig. 3.4a, one pixel would equal 0.9611 nm. The accuracy of these measurements based on an image can be questioned, however it does provide a sense of scale and serves as a comparison for relative order of magnitude.

The FG cell has a width of 16.3 nm and a height of 20.2 nm. The tunnel oxide has a thickness of 19.5 nm. The IPD has a width of 6.9 nm (between FG and CG) and a height of 11.0 nm (between FG and spacer). The central channel has a diameter of 61.7 nm.

3.1.3 Page size and architecture

The single memory cells are physically combined in the 3D architecture into larger addressable structures of memory pages and blocks. A single memory page is the smallest addressable unit, while the block is the smallest erasable unit.

The Micron B17A FortisFlash 3D NAND memory has the following internal organization. These are referred to as the *virtual* pages and blocks, as this is what is presented to the user, not what is actually represented by the physical layout.

***Virtual* page size:** 18,592 bytes
***Virtual* block size:** 2304 pages

These addressable virtual memory units can be translated into a number of physical memory cells and pages. One byte contains 8 bits, and a single memory cell in TLC mode stores 3 bits. The pages and blocks can hence be represented in units of a "physical" page and block size:

***Physical* page size:** 148,736 memory cells
***Physical* block size:** 768 physical pages

It is investigated how the memory cells are combined in a group to make a physical page. The proposed page structure should fit within the surface area of the die and the architecture shall be based on those proposed in the literature.

Page layout in the basic BiCS architecture

In the basic BiCS 3D NAND architecture, the memory cells of a single page are located on a single layer, in a row perpendicular to the bit lines [14][3]. An example of the physical volume of a single page in a biCS architecture is shown in Fig. 3.5.

The physical length of a page can be computed based on the size of a single physical memory cell and the number of memory cells per page.

$$0.198\ \mu\text{m} \cdot 148736 \text{ memory cells} = 29.5 \text{ mm} \qquad \text{(Length of a single page)} \tag{3.5}$$

The estimated page length of 29.5 mm exceeds the die size of 7.43 by 7.83 mm^2. Therefore this basic page layout is discarted and other page layouts are investigated.

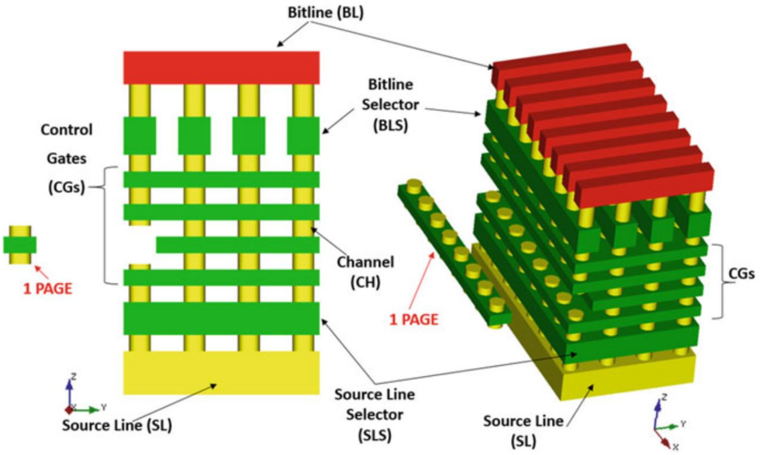


Figure 3.5: A single page in a 3D NAND architecture is proposed as a row of memory cells on the same layer [14].

BiCS architecture with staggered pillars and bitlines

By placing the memory pillars in a staggered arrangement, the density is increased and the page size is doubled. A further improvement by fitting two bitlines over a single column of pillars, again doubles the page size. Increasing the original page size a total of four times compared to the initial single-row architecture. A single NAND memory page is then comprised of two pairs of even-odd rows or pillars.[14]. Fig. 3.6 and 3.7 show the respective layouts.

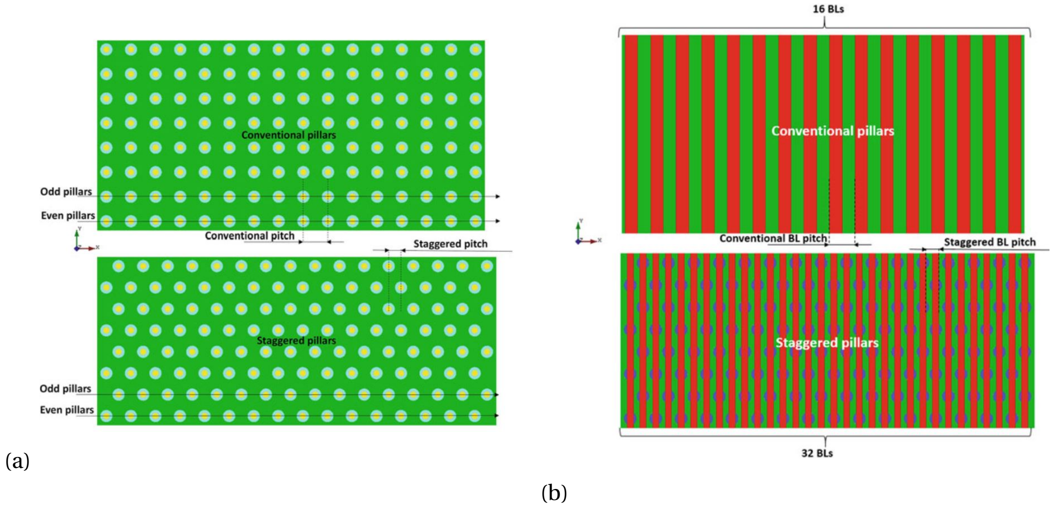


Figure 3.6: (a) Top view comparison of a conventional and staggered pillar layout. (b) By staggering the pillars, the bitline density can be doubled [14].

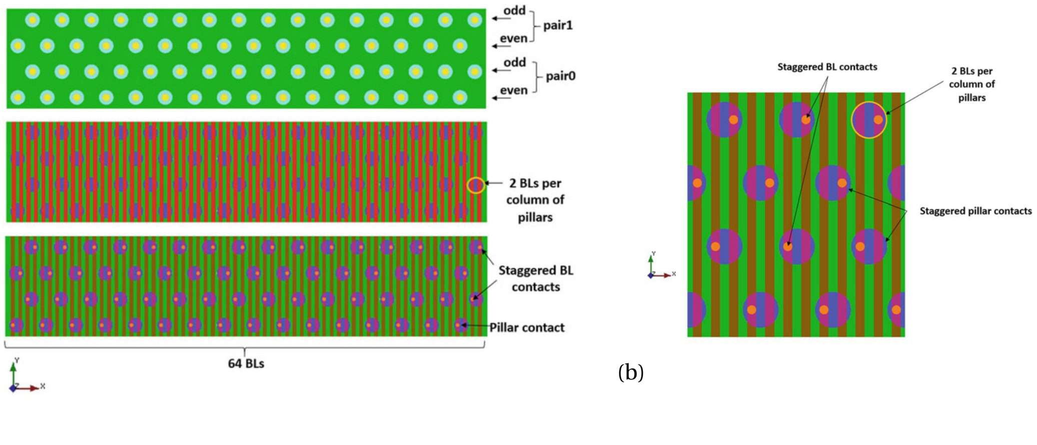


Figure 3.7: The architecture can be further improved by also staggering the bitlines. Two bitlines fit within a single column of pillars [14].

Samsung's V-NAND featured a staggered pillars in their first generation chips (2014) and staggered bitline contacts in their second generation chips (2015) [14]. This makes it reasonable to assume that the Micron chips use a similar technique in the current version of their 3D NAND memories. Additionally, Fig. 3.2 reveals that seemingly, a staggered layout is present due to a slight angle in the cut section, revealing a second row of pillars located in between the initial front row of pillars.

Decreasing the number of memory cells lengthwise by a factor of four, the new page length and width can be computed:

$$0.198\ \mu\text{m} \cdot 148736 \text{ memory cells}/4 = 7.36 \text{ mm} \qquad \text{(Length of a single page)} \tag{3.6}$$

$$0.198\ \mu\text{m} \cdot 4 \text{ memory cells} = 0.792\ \mu\text{mm} \qquad \text{(Width of a single page)} \tag{3.7}$$

The length of a single page in this proposed layout is estimated at 7.36 mm, which fits within the die size of 7.43 mm x 7.83 mm. It is assumed that the internal page layout is based on the staggered pillar and bitline layout as presented above. Concluding that a page measures 7.36 mm in length, 0.792 μm in width and has an approximate height of 52.6 nm.

3.1.4 Memory block architecture and size

Finally, the layout and size for a single memory block can be estimated based on the physical page layout and the number of pages in the block.

A 64 layer memory block with 768 pages should have a width of 12 pages (this number can also be concluded studying Table 69 in the B17A datasheet). Each page is four memory cells wide. The architecture concluded for a single memory block is presented in Fig. 3.8.

The size for a single memory block can be estimated at 7360 μm (depth) x 3.37 μm (height) x 9.5 μm (width).

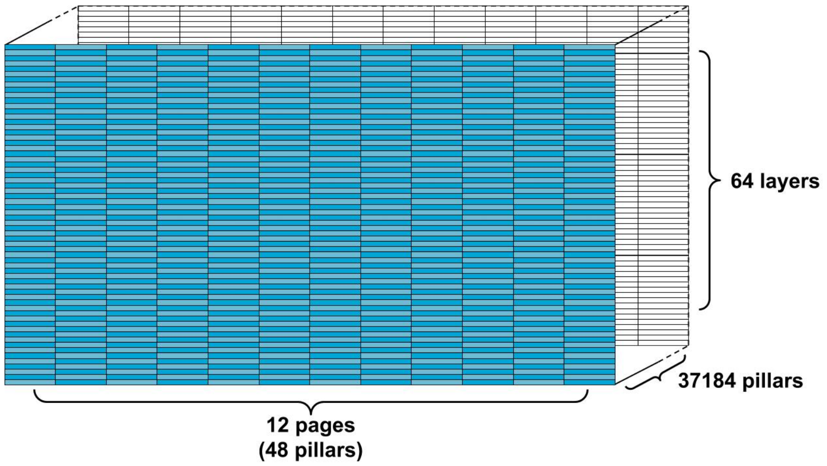


Figure 3.8: The proposed architecture of a single memory block, fitting the observations of the SEM images and surface photographs.

As a final check, one can determine whether all the blocks of a single memory would fit on the die. It is quoted in the TechInsights blog post that the die holds 256 Gb of data [23]. Based on the number of blocks available in the 512 Gb B17A device (2016), it could be estimated that 1008 blocks would fit next to each other onto the die. Calculating the total width results to 9.58 mm, exceeding the maximum die side length of 7.83 mm. This difference could be attributed due to the assumption that the cell size was computed with a squared surface area, while the staggered layout would increase the density of cells per area, hence slightly decreasing the width (and possibly length) of each block.

3.2 SPENVIS radiation simulation

The size of a single block was estimated and could be used as an input for a SPENVIS radiation simulation, to determine how many particle events it would see.

A set of simulations is performed using SPENVIS (SPace ENVironment Information System), an ESA supported web portal with models of the space radiation environment. The tool is free for use upon registering. The model is set up to determine the integral and differential fluence of particles. Based on the lower energy that can be detected by the memory cells, an estimate can be made for the number of particle events seen in different orbits and for different radiation sources. The number of events can be determined by multiplying the integral fluence for a certain threshold LET(Si) with the surface area of one block (7360 x 9.5 μm^2) and integrating over the surface of a sphere (4*π steradians). With the simulated mission time set to one day, the result will be the number of particles seen by a single block in the memory, per day, given a certain threshold surface LET(Si) value.

The calculations will be repeated for four different orbits: ISS, LEO, MEO, GEO.

3.2.1 SPENVIS Model Setup

Workflow

Different setting are configured before a final data product is created using the SEE module. An overview of the different steps is given below:

1. Set the spacecraft trajectory under **Coordinate generators**
   (a) Define *Spacecraft trajectories*
      - Mission duration - *One day*
      - Set *Orbit Start* to coincide with Solar minimum/maximum
      - Set main orbital parameters: Perigee, Apogee and Inclination
2. Set the radiation sources under **Radiation sources and effects**
   (a) Define *Trapped proton and electron fluxes*
      - AP-8 and AE-8 as Proton/Electron model
      - Define solar minimum or mission epoch for solar maximum
   (b) Define *Solar particle mission fluences*
      - SAPPHIRE (total fluence) as Solar particle model
   (c) Define *Galactic cosmic ray fluxes*
      - CREME96 as GCR model at 1 AU
      - Define solar minimum or mission epoch for solar maximum
3. Calculate the effects under **Radiation sources and effects**
   (a) *Long-term SEUs and LET spectra*
      - 0.0 cm Shielding thickness
      - Si (SRIM2008) as the Device Material

SPENVIS Data Products

After running through the presented workflow, the *Average LET (Si, SRIM), proton and ion fluxes* file is downloaded (*spenvis_nlofl_srimsi.txt*) for each user case, containing the integral and differential fluences in function of the particle LET(Si).

3.2.2 Results for the four orbits

The SPENVIS results for the four orbits and the different radiation sources are shown in Fig. 3.9, 3.10, 3.11 and 3.12. The fluence for the trapped proton particles are dependent on the spacecraft's orbit. The fluence due to the Galactic Cosmic Rays has an anti-correlation with the solar cycle. The results are plotted for a solar cycle minimum, hence the highest number of GCR particles, where only the lowest energy cosmic rays are affected by Earth's magnetic field.

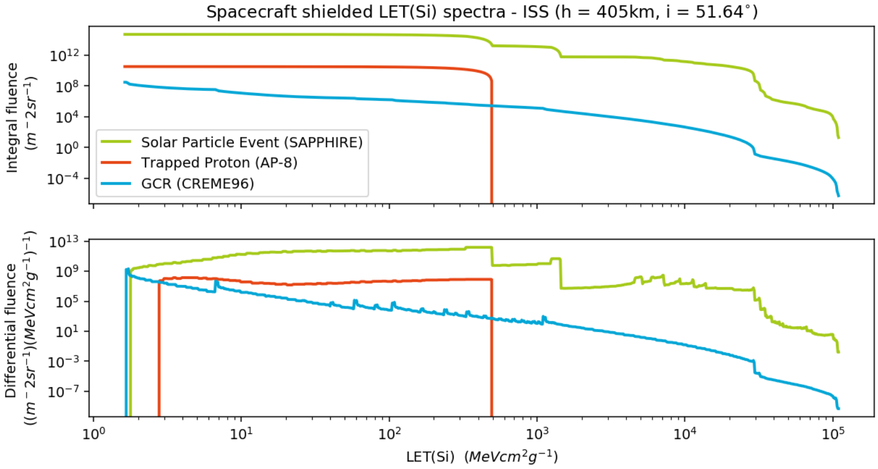


Figure 3.9: Integral and Differential Particle Fluence in function of the particle LET(Si) for an ISS orbit.

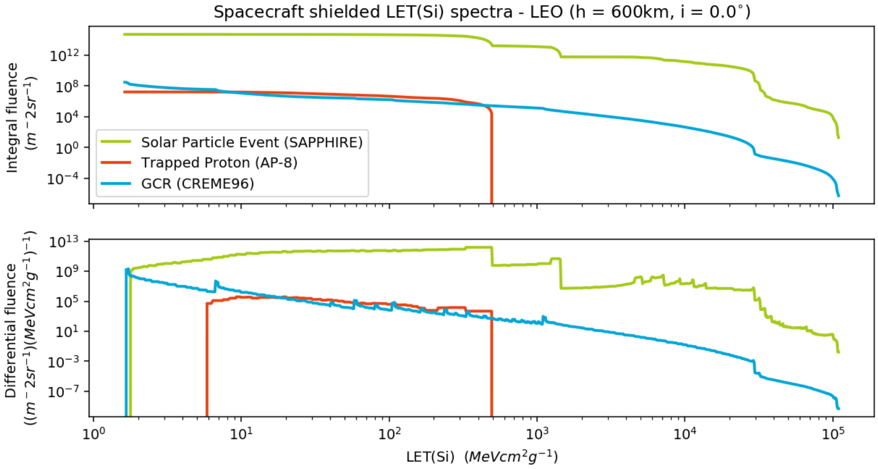


Figure 3.10: Integral and Differential Particle Fluence in function of the particle LET(Si) for a LEO orbit.

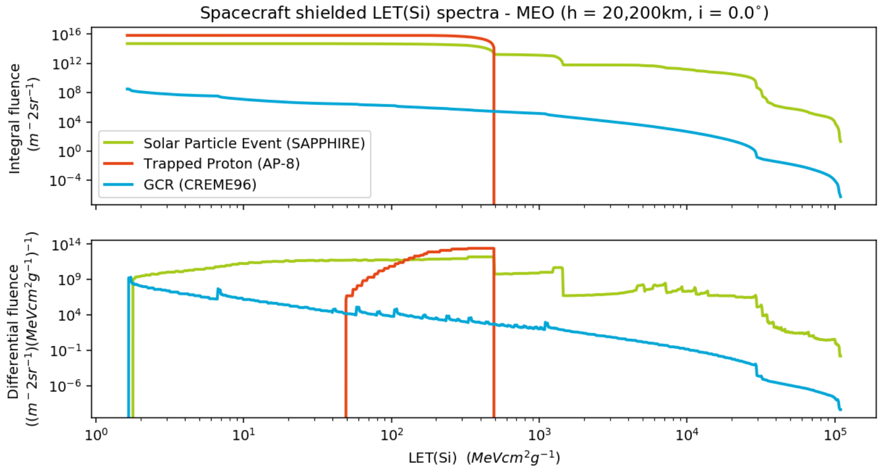


Figure 3.11: Integral and Differential Particle Fluence in function of the particle LET(Si) for an MEO orbit.

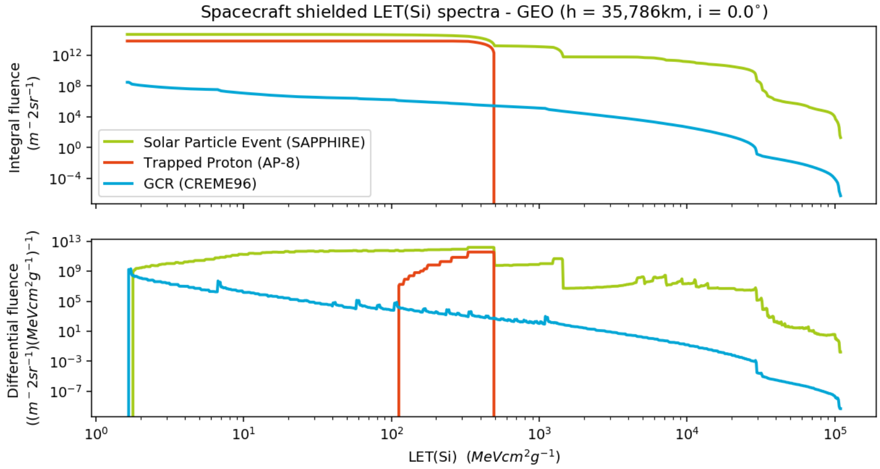
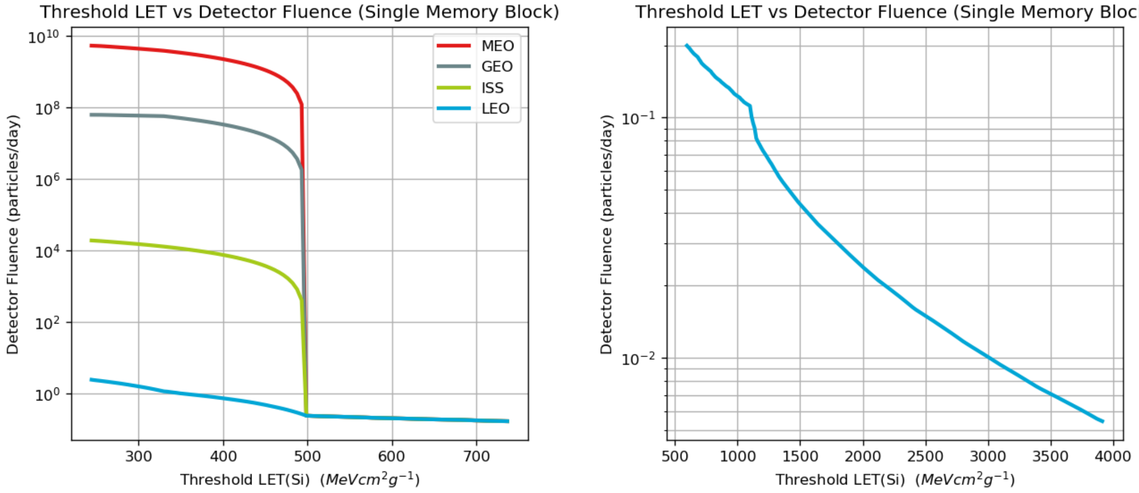


Figure 3.12: Integral and Differential Particle Fluence in function of the particle LET(Si) for an GEO orbit.

From the integral fluence data, the detector fluence at solar minimum taking into account the trapped proton and GCR particles, can be plotted in function of the threshold LET(Si) of the cells. This is plotted in Fig. 3.13 for both lower and higher values for the threshold LET(Si). A test campaign will have to determine the detectable particle with the lowest LET.

As a reference, the lowest LET particle used in [1] is Oxygen at 109 MeV with an LET(Si) of 2850 MeV/cm^2/g. Based on this value, a memory block would see 0.011 particle events/day, a surprisingly low value. The number of events measured could be increased by using more than just a single memory block as a sensitive volume. Depending on the internal layout of the memory packages, up to 32,256 blocks could be available.



(a)

(b)

Figure 3.13: The number of detectable GCR or trapped proton radiation events as a function of threshold LET(Si).

The data in the figure indicate the effect of the orbit on the detector fluence. The right plot (Fig. 3.13b) with higher threshold LET(Si) values is not affected by the orbit, this is because the primary cause are GCR particles, not mostly unaffected by Earth's magnetic field. The left plot (Fig. 3.13a) sees a variation in the detector fluence due to the lower threshold LET(Si) proton particles. The highest fluences are seen for the MEO and GEO orbits, passing through the Van Allen radiation belts. The ISS orbit has a higher number of events than a general 0° inclination LEO orbit, believed to be attributed due to the passing through the South Atlantic Anomaly. The overall observation can be made that proton events are, by order of magnitudes, more likely to be seen than GCR particle events.

From the SPENVIS simulations, it can be concluded that it will be important to measure the lowest threshold LET(Si) for the 3D NAND memory, to estimate the number of events seen by the memory. The number of high energy particle events for a single memory block will be particularly low (possibly as little as 0.011 particle events/day), therefore it might be necessary to deploy a large number of memory blocks as sensitive volumes.

Chapter 4

Test setup development

This chapter is dedicated to the procurement of a NAND flash memory chip and the development of a technical solution to control this 3D NAND memory. It will answer and materialize the third and final research question: what hardware and software is necessary to make threshold voltage shift measurements. The requirements for the hardware and software development are based on the ONFI standard, for which the key features are discussed in Section 4.2. A breadboarded setup is built, interfacing the memory with an FPGA on which an ONFI memory controller runs. A proven test setup with which memory control is possible marks the start of the in-house capability for future memory related projects and research.

4.1 DUT Procurement

The Micron B17A was chosen for its capabilities to measure the threshold voltage using the READ OFFSET function. The device is listed on the Micron website, however it is not directly available for purchase. In order to have a Micron 3D NAND memory on which to run the desired commands, one or several chips should be procured.

The different purchasing options that were considered will be first presented, after which it is shown how eventually a commercial SSD was bought from which the BGA memory packages are unsoldered.

4.1.1 Embedded and external memory control

Flash memories are typically sold in two versions: with or without an embedded memory controller. These memory controllers manage where the data is stored onto the physical memory cells in the most efficient way and ensure the correct data is returned to the user. Memories with an embedded controller are found in compact handhold devices such as smartphones, tablets and cameras, in which a single memory packages is placed. The embedded controller communicates with one of two JEDEC (Global Standards for the Microelectronic Industry) coordinated standards: UFS or eMMC. An example of this architecture is shown in Fig. 4.1. The memory controller can also be a separate package, managing multiple so called *rawNAND* flash memories, shown in Fig. 4.2. These are common in SSDs found in laptops and desktop computers. Memory controller manufacturers include Sandforce and Silicon Motion. The rawNAND flash memories itself are controlled by a standard also defined by JEDEC: ONFI (Open NAND Flash Interface).

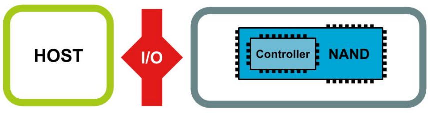


Figure 4.1: Memory package with embedded controller, commonly found in compact hand held devices. The interface with the host uses a UFS or eMMC standard.

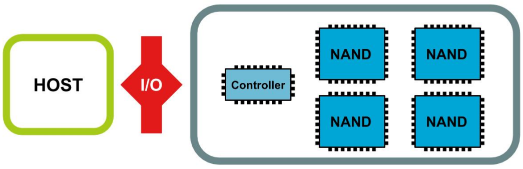


Figure 4.2: Example architecture where a storage device contains multiple rawNAND flash packages, managed by a single controller. Common interfaces when used as an SSD are SATA, mSATA and M.2.

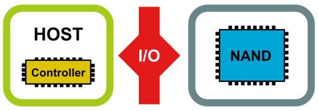


Figure 4.3: In a third and final configuration: a host itself can manage a flash memory, where the controller software is run directly from the main processor.

The memory controllers serve their purpose for efficient data management, but turn the flash memory into a black box. The data is written to a virtual address in the memory controller, which then decides upon a second virtual address in one of the memories to store the data (again virtual, because the memory will then translate a virtual memory address into an actual physical cell or page location). When it is desired to know in which exact physical location data is stored, one must circumvent such controller and directly program data to the virtual addresses of the memory. The exact physical location is still not known, but a first estimation could be made based on the virtual address. The first abstraction layer of the memory controller is removed. In the framework of this project, it is therefore required to obtain a rawNAND memory package.

4.1.2 Micron B17A rawNAND memory procurement

Several options were investigated to procure the specific Micron B17A rawNAND packages. Eventually, it was chosen to desolder the memories found in an SSD.

Chip order from distributor

Micron has a list with authorized distributers in Europe[1], where their devices can be bought. Through a contact with Airbus Defence and Space in Friedrichshafen, a request was made with EBVElectronic to buy Micron B17A chips. However, it was not possible to buy a small quantity of chips, the smallest order was 1000 units. A price of US$20.313,30 including 19% VAT was quoted. This was deemed out of budget for this project.

Buying smaller batches online

Several online resellers[2] provide the option of purchasing smaller batches of electrical components, including flash memory packages. Unfortunately, the Micron 3D NAND technology was not (yet) available due to it is novelty. It might be a good option in the future, when one is looking to test older but still relevant technology.

[1] micron.com/support/sales-network/authorized-distributors

[2] For example kynix.com

Unsoldering from SSD drive

As indicated earlier in Section 4.1.1, SSDs consist out of multiple rawNAND memory package managed by a main memory controller. One could buy such an SSD and unsolder the (BGA) memories.

Micron announced their Micron 5100 series memory solution based on the 32-layer 3D TLC technology in December 2016. Available both in 2.5in SATA and M.2 form factor[3]. An breakdown blog post revealed that the 2.5in SSD contained 12 NAND BGA packages in both the 960 GB and 1.92 TB models[4].

At the time of investigating the different procurement options (January 2018), Micron announced their Micron 5200 series, a line of SSD memory solutions based on the Micron 64-layer 3D NAND architecture. After the announcement, they were available for purchase online in the US. It was chosen to buy a 480 GB Micron 5200 ECO SATA drive from CDW.

Figure 4.4: Micron 5200 SSD drive. Source: Micron

Upon removing the metal enclosure of the drive, a PCB half the size of the 2.5in package was revealed. Only 3 memories were present on the drive, indicated with a blue box in Fig. 4.5. It was somewhat of a disappointment that only 3 memory packages were found, the previous version had 12 memory packages. It is believed that the empty BGA-132 placeholder on the PCB will be used in higher memory capacity configurations. The two memory packages on the front side had the following markings: 7XA22, NW921, 5GJL. The front memory package 7XA22, NW928, 4YBQ. Indicating two types of chips combined to a single 480GB drive. It will later be shown in Section 6.1.1 that two packages with markings 7XA22, NW921, 5GJL are 2 Tb dies (MT29F2T08EMHAFJ) and the third chip with markings 7XA22, NW928, 4YBQ is a 512 Gb chip (MT29F512G08EBHAFJ4).

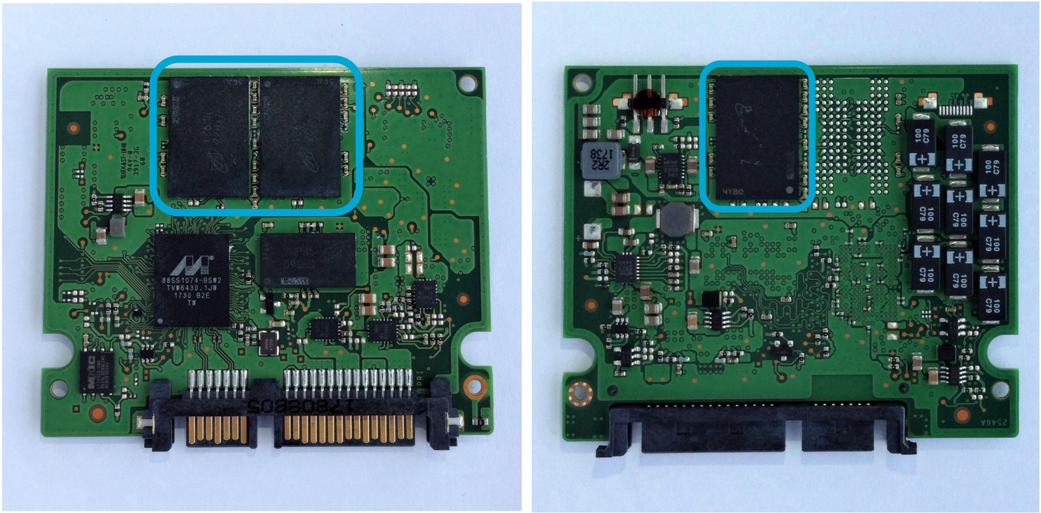


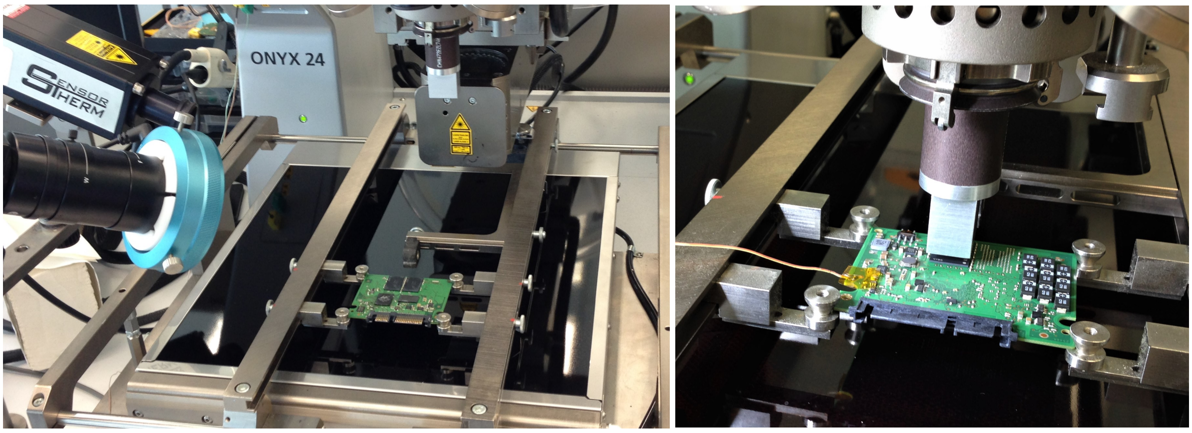
Figure 4.5: The front and back side of the PCB found inside the SSD after removing the 2.5" metal casing. Three NAND packages are present in the 480 GB Micron 5200 ECO SATA drive.

[3] https://www.anandtech.com/show/10886/micron-announces-5100-series-enterprise-sata-ssds-with-3d-tlc-nand
[4] https://www.tweaktown.com/reviews/8037/micron-5100-series-eco-max-sata-enterprise-ssd-review/index2.html

4.1.3 BGA package extraction from PCB

The BGA memory packages need to be unsoldered from the PCB. If one wants to resolder them later, they also need to be reballed, which is the process of applying solder to each of the BGA pads. The unsoldering process of a BGA typically consists of heating the package with hot air until the solder on the pads reach their melting temperature.

The unsoldering and reballing process took place at the Dutch Forensic Institute (Nederlands Forensisch Instituut, NFI in The Hague), where the tools and expertise is available to extract such packages off a PCB with great care, preventing chip damage. The three BGA-132 packages were unsoldered using a ZEVAC Onyx24 selective soldering and desoldering machine. It is able to accurately control both the temperature of chip as the temperature gradient while heating up. Heat is applied over the whole PCB from the bottom heater plate. A hot air gun targets the specific area of the package. It was aimed to keep the temperature gradient below 2°C/s to prevent chip damage, according to their own experiences. Once a preset temperature was reached (145 °C), a vacuum suction cup lifts the chip off the PCB.



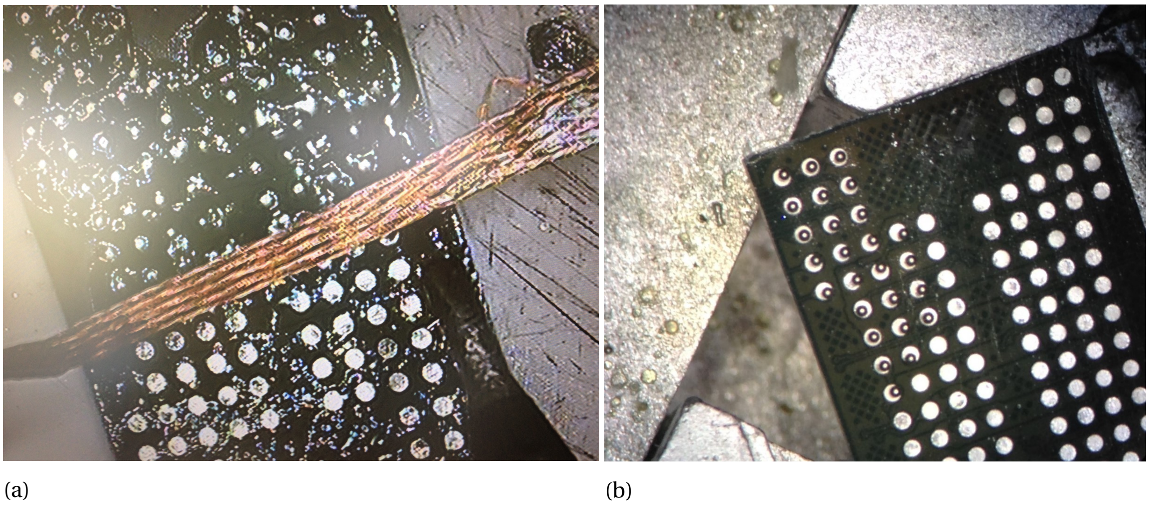
(a)

(b)

Figure 4.6: Unsoldering of the BGA packages using the ZEVAC Onyx24 machine. (a) An optical and thermal camera track the progress. (b) The hot airgun is fitted with a mouth piece to localize the heating.

After unsoldering the three packages off the PCB, the remaining solder on the pads was removed using a solder iron and solder braid. Upon cleaning the area with an alcohol solution, the chips looked brand new.

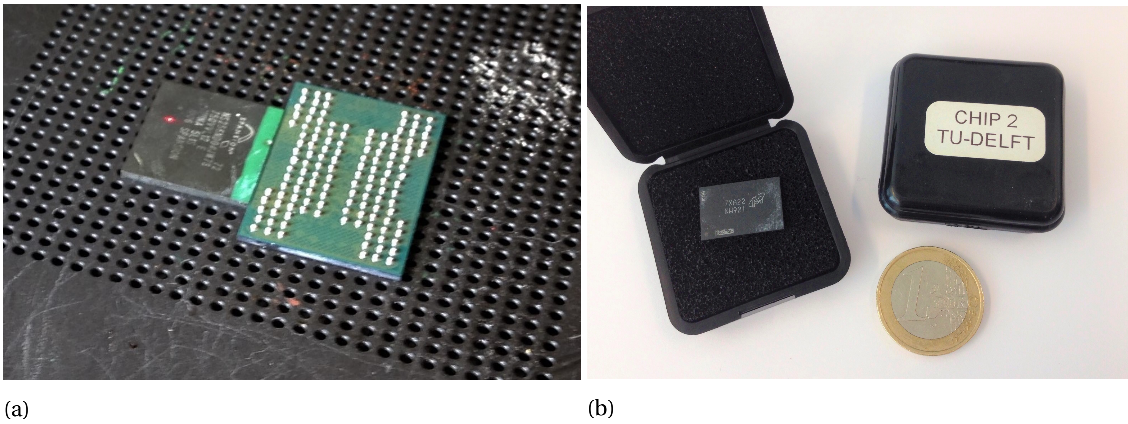
The pads have a diameter of 0.55 mm, upon the NFI's advice, 0.01 inch diameter (0.25 mm) solder balls for reballing were used. First a solder paste was thinly applied on the chip, after which the solder balls were manually placed one by one on the pads.



(a) (b)

Figure 4.7: A (digital) microscope is a necessary tool when working with fingernail-sized packages. (a) Removing leftover solder from the chip using desoldering braid an soldering iron. (b) 0.25 mm diameter solder balls are placed on each of the 132 pads.

The chips were again placed in the ZEVAC Onyx24 machine as shown in Fig. 4.8 (left) and heated to 132°C with a bottom infrared heater, just above the solder ball's melting point. Upon reaching that temperature, the solder balls melted onto the pads enabling future resoldering on a PCB. The memories were placed in three separate containers marked CHIP 1 to CHIP 3. An overview is given in Fig. 4.8 (right). No markings were added to the actual memory packages.



(a) (b)

Figure 4.8: (a) The solder balls are melted using IR radiation from the bottom. A reference chip is placed next to it to measure temperature. (b) The packages are kept safe in three separate marked containers. The euro coin again indicates the small size of the memories.

4.2 The ONFI standard: guide to rawNAND memory control

The hardware and software interfaces between a host and most of the NAND flash memory devices are defined by the Open NAND Flash Interface Specification or ONFI.[17] It standard will define the software and hardware requirements to develop the test setup presented in this chapter, in order to communicate with the rawNAND memory extracted from the SSD. A summary of these requirements will be given in this section. A focus will be made on the features specific to the Micron 3D NAND memory and relevant to this thesis. Through the use of memories compliant with the ONFI standard, other devices could be instructed with the identical requirements.

The ONFI standard it the result of an industry workgroup made up of more than 100 companies that design, build or enable NAND flash memory. The latest major revision was in 2014: ONFI 4.0, by a collaboration between major players in the field of solid state memories: Intel, Micron, Phison Electronics, SanDisk and SK Hynix. The latest revision, 4.1, was released in 2018. The specifications are free for download at `www.onfi.org`.

4.2.1 Software Interface

A memory controller will generate the necessary signals to execute commands on the memory. This section will give an overview of the different software aspects to guide the development of the memory controller. These include the different timing modes which define the transfer speed, the definitions of the control and data signals and the timing diagrams which are the building blocks to generating memory instructions.

Timing Modes

The memory can operate in two main different timing modes based on asynchronous (SDR) and synchronous (NV-DDR) data transfer. Each mode requires specific hardware connections and controller software. The synchronous timing mode provides a faster data interface than the asynchronous mode.

The hardware and software development in this thesis will be based on the Single Data Rate (SDR) asynchronous data transfer. The timing requirements are less strict and it an open source controller is readily available, as will be discussed in [SECTION]. Using the SDR timing mode will allow to show the working principle of controlling the memory and taking the necessary measurements.

The Single Data Rate (SDR) asynchronous data transfer will be enabled when providing the I/O supply voltage (V_{ccQ}) of 1.8V at startup. It has different timing modes (0 to 5), each increasing their data throughput. The NV-DDR2 and NV-DDR3 (Non Volatile-Double Data Rate) is a synchronous mode running off a clock, where data is transfered both on the rising and falling edge of the clock signal. The DDR3 mode sees an increase in speed as it is able to operate at a faster clock speed than the DDR2 mode. NV-DDR3 mode will be enabled at startup with a V_{ccQ} of 1.2V. NV-DDR also supports multiple timing modes (0 to 10).

For the Micron FortisFlash 3D NAND memory, the following read/write throughput per pin are quoted (in MegaTransfers/Second): SDR: 50 MT/s, NV-DDR2: 533 MT/s, NV-DDR3: 667 MT/s. It can be seen that a ten fold speed increase is possible by using the DDR mode.

Control and Data Signals

The memory is controlled by sending digital signals to the pads or connectors of its package. Data will be transferred over 8 signal lines (8-bit data bus). A description will be given about each of the signal lines relevant to controller the memory in SDR mode.

Binary logic has two logic levels: low and high, corresponding to the binary numbers 1 and 0 respectively. A high signal does not necessarily indicate an *On* or *True* state. A high signal indicating an *Off* or *False* state are called *active low*, meaning that a they are active (True) when pulled low (0). Several ONFI signals are active low. Various notations representing an active low signal are used throughout the literature. The following notations are commonly used: $\overline{\text{RE}}$, RE# or RE_n.

Table 4.1 describes the signals used for memory control. Four control signals (RE#, WE#, CLE and ALE) are the main drivers for memory control. The data is be placed on the 8-bit I/O interface (DQ0-DQ7). The status of the memory is given by the Read/Busy signal (R/B#). The Write Protect and Chip Enable (WP#, CE#) can be kept *Off* and *On* at all times.

Table 4.1: Main Signal Descriptions for ONFI memory control in SDR mode.

Signal Name	Input/ Output	Description
RE#	I	**Read Enable** The Read Enable signal enables serial data output.
WE#	I	**Write Enable** The Write Enable signal controls the latching of commands, addresses, and input data in the SDR data interface. Data, commands, and addresses are latched on the rising edge of WE#.
CLE	I	**Command Latch Enable** The Command Latch Enable signal is one of the signals used by the host to indicate the type of bus cycle (command, address, data).
ALE	I	**Address Latch Enable** The Address Latch Enable signal is one of the signals used by the host to indicate the type of bus cycle (command, address, data).
DQ0 - DQ7	I/O	**I/O Port** The I/O port is an 8-bit wide bidirectional port for transferring address, command, and data to and from the device.
R/B#	O	**Read/Busy** The Read/Busy signal indicates the target status. When low, the signal indicates that operations are in progress.
WP#	I	**Write Protect** The Write Protect signal disables Flash array program and erase operations.
CE#	I	**Chip Enable** When Chip Enable is low, the target is selected.

Timing Diagrams

Digital timing diagrams are a representation of a set of signals in the time domain. They provide an overall description and can show timing relationships between the different signals. Three states are generally shown: High or logic 1, Low or logic 0 and grayed out, equivalent to "don't care".

The diagrams indicate the timings with respect to a single reference point. The values of these timings are tabulated in the main ONFI document and Micron documentation for the different timing modes. It is important to note that these values generally represent the *minimum* time that the signal is toggled. This relaxes the timing constraints for a controller, including timing delays that might occur due to hardware related issues.

The ONFI standard defines four main signal combinations that serve as building blocks for all the instructions sent to control the memory: Command, Address, Data In, Data Out. An instruction will be the sequential combination of one or more of these four building blocks. In addition to a defined instruction sequence, minimum timing requirements are set between building blocks. The timing diagrams for the four fundamental signal combinations are shown Fig. 4.9, 4.10, 4.11 and 4.12. An example set of memory instructions be described in Section 4.4.1. Fig. 5.6 shows the actual generated signals by the memory controller built for a command.

The minimum timing requirements are tabulated in the ONFI specification, Table 83 and Table 84 indicate the timings for SDR in mode 0 to 5.

- The *Command* will tell the memory which instruction to execute. A *Command* is written to the memory by placing the 8-bit binary encoded hex instruction (h00 to hFF) on the I/O interface while toggling the *CLE* and *WE_n* signals. It is possible that the memory responds to a command by pulling the *R/B_n* signal low.

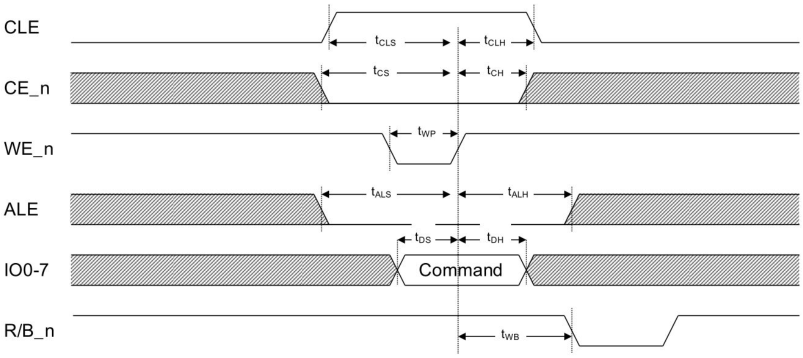


Figure 4.9: Command Latch Timings

- The *Address* instruction will latch a (or multiple) memory address onto the memory. The 8-bit address is placed on the I/O interface while toggling the *ALE* and *WE_n* signals.

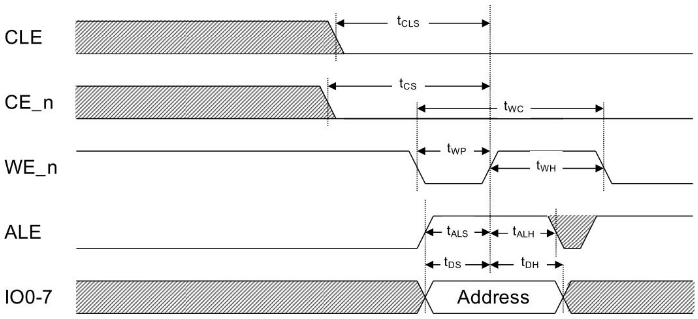


Figure 4.10: Address Latch Timings

- The memory shall either receive data to store, or return stored data to the user. To write data to the memory, the controller will place one byte of data on the I/O interface and repeatedly toggle the *WE_n* signal. This will only take place after the required *Command* and *Address* instructions are sent.

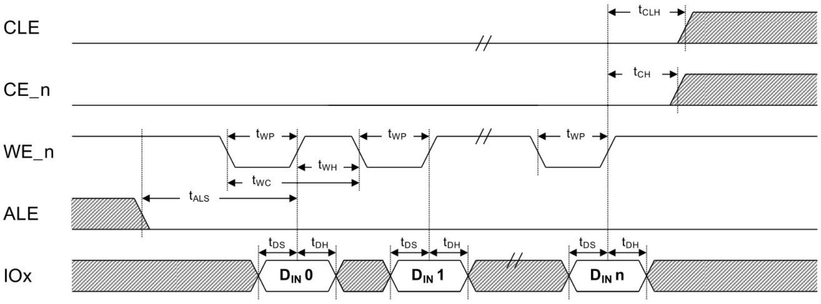


Figure 4.11: Data Input Cycle Timings

- To retrieve data from the memory is similar to writing data, but here the *RE_n* signal is toggled for each byte to be written by the memory to the I/O interface.

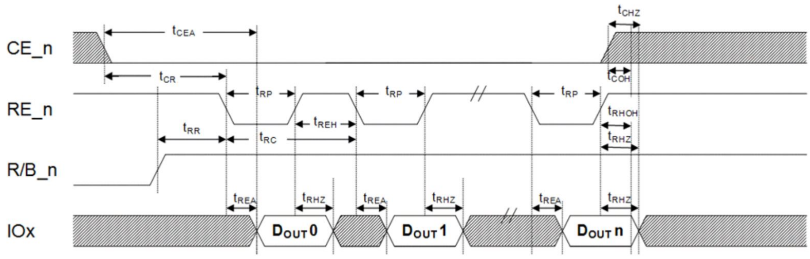


Figure 4.12: Data Output Cycle Timings

A memory controller must be able to replicate the different timing diagrams presented above. For an instruction to be executed, a sequence of control signals must be issued with the specific commands, addresses or data set on the 8-bit data bus.

Memory Addressing

The *Address* instruction is submitted with a memory address to target a specific memory location. A memory address needs to be submitted in a structured matter. An overview will be given on how an address shall be latched into the Micron F17A 3D NAND memory.

Although the smallest unit of data is one single bit, a memory has an internal structure in which the smallest (adressable) unit is a page, containing a large amount of bytes (18.5 MB for the Micron B17A). The internal memory organization includes pages, blocks and LUNs. Multiple pages making up a block and a group of blocks making up a LUN.

Two types of addresses are used to access a memory location within the memory: the *column address* (CA) and the *row address* (RA). The column address locates data within a page, it indicates the offset location within the page. The row address is used to locate pages, blocks and LUNs.

The column and row addresses are issued in multiple 8 bit cycles. When both the column and row address are required (for example to read a single page), the column address is loaded before the row address. Sometimes instructions only require the row address, for example when erasing a block. The column address with information on the pages is of no use.

Fig. 4.13 shows an example timing diagram for the READ PAGE operation, where both the column address and the row address are required. It can be seen that the instruction is made by sequentially issuing three of the four main instruction set building blocks: *Command*, *Address* and *Data Out*.

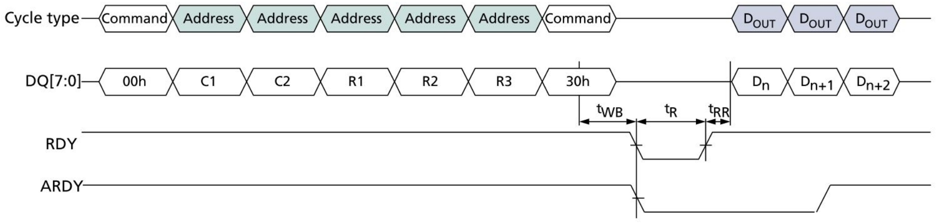
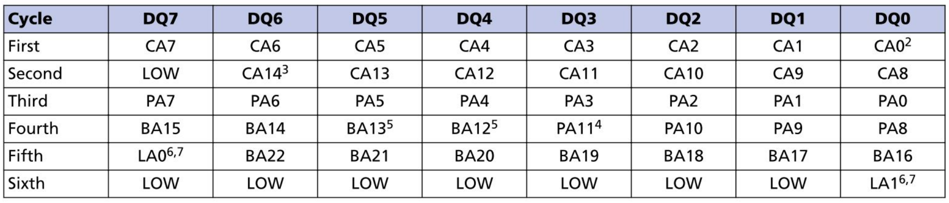


Figure 4.13: Timing Diagram for the READ PAGE operation, both the Column Address (CA) and Row Address (RA) are issued [16].

The CA and RA have a predefined order in which the locations (page, block and LUN) are specified. The addresses are encoded in a set of 8-bit instructions which are latched onto the data bus. Fig. 4.14 indicates the orders for the B17A Micron memory. Each cycle is latched with the *Address* instruction with the right binary data present on the I/O interface. It must be noted that the sixth cycle may be omitted if LA1 is 0.



Cycle	DQ7	DQ6	DQ5	DQ4	DQ3	DQ2	DQ1	DQ0
First	CA7	CA6	CA5	CA4	CA3	CA2	CA1	CA0[2]
Second	LOW	CA14[3]	CA13	CA12	CA11	CA10	CA9	CA8
Third	PA7	PA6	PA5	PA4	PA3	PA2	PA1	PA0
Fourth	BA15	BA14	BA13[5]	BA12[5]	PA11[4]	PA10	PA9	PA8
Fifth	LA0[6,7]	BA22	BA21	BA20	BA19	BA18	BA17	BA16
Sixth	LOW	LOW	LOW	LOW	LOW	LOW	LOW	LA1[6,7]

Figure 4.14: Array Addressing for Logical Unit (LUN) for B17A in TLC mode [16].

The length of each unit is such that it fits the maximum binary address specific to the memory internal configuration.

- **CA0 to CA14**: 15 bits are needed to select the word (8 bit) within a page. A single memory page holds 18,592 words, can be represented in 15 bits (18,591 in binary is 100100010011111).
- **PA0 to PA11**: 12 bits are needed to select the page. A block consists out of 2304 pages, hence 12 bits are needed (2303 in binary is 100011111111).
- **BA12 to BA13**: plane select bits. There are four planes present and the plane number can be determined by taking the rest division with 4 of the block. A visual representation is given by Figure 12 in [16].
- **BA14 to BA22**: 9 bits to select on out of 504 blocks, (503 in binary is 111110111).
- **LA0 and LA1**: LUN select bits. Present when more than one LUN shared on a target. Otherwise Low.

Feature Addressing

The address command cycle is also used for to specify a specific sub function to set a parameter. For example: the SET FEATURES operation requires an address which defines the feature that will be modified. The READ OFFSET function will be called by issuing a command between A0h and ACh (hex values shall be converted into 8 bit binary), which specifies the reference level to which the offset shall be applied. An example diagram is shown in Fig. 4.15. The P1-P4 are the data settings applied to that feature and are described in the data sheet of the memory for each feature.

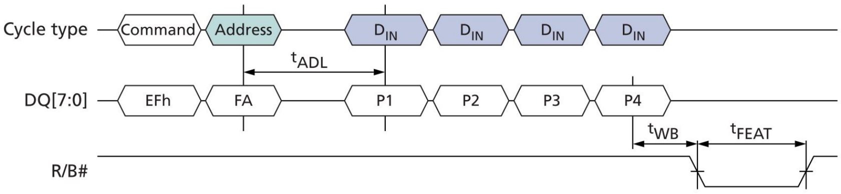


Figure 4.15: The SET FEATURES instruction requires an address to specify which feature to modify. The modification of that feature will be defined in the following data cycles P1 to P4 [16].

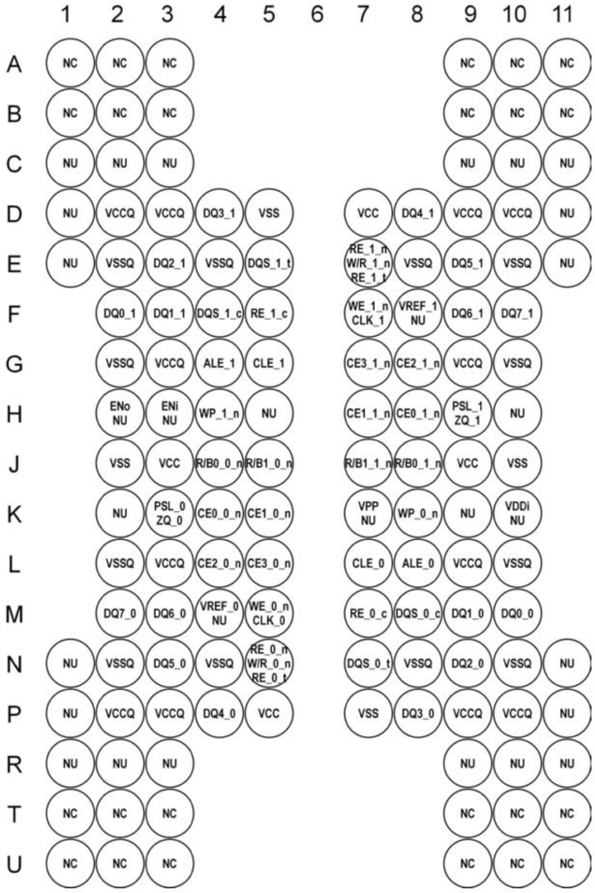
4.2.2 Physical Interface

The interface between a memory and their respective memory controller are defined in the ONFI standard. The signal assignment to the pins or pads of different packages are specified, as well as the required voltages to drive the memory.

BGA-132 ball assignment

ONFI defines the signal-pad assignment for many different standardized memory packages. The 132-BGA assignment, the package of the DUT, is shown in Fig. 4.16 (Ball-down, top view). Depending on the data interface (SDR or NV-DDR), the pads may have a different function. Devices with dual 8-bit data access will use the pads marked both with "_0" and "_1". The single 8-bit Micron 3D NAND memory will only use the "_0" assigned pads.

Furthermore, it can be noticed that only the center rectangular pads are assigned their respective signals, and that the outer rows and columns are assigned NU (Not Usable) or NC (No (internal) connection). The BGA-152 packaging features an identical core pad layout, with an additional column left and right of the BGA-132 outline with more NU and NC assigned pads. This makes both packages reverse compatible.



	1	2	3	4	5	6	7	8	9	10	11
A	NC	NC	NC						NC	NC	NC
B	NC	NC	NC						NC	NC	NC
C	NU	NU	NU						NU	NU	NU
D	NU	VCCQ	VCCQ	DQ3_1	VSS		VCC	DQ4_1	VCCQ	VCCQ	NU
E	NU	VSSQ	DQ2_1	VSSQ	DQS_1_t		RE_1_n W/R_1_n RE_1_t	VSSQ	DQ5_1	VSSQ	NU
F		DQ0_1	DQ1_1	DQS_1_c	RE_1_c		WE_1_n CLK_1	VREF_1 NU	DQ6_1	DQ7_1	
G		VSSQ	VCCQ	ALE_1	CLE_1		CE3_1_n	CE2_1_n	VCCQ	VSSQ	
H		ENo NU	ENi NU	WP_1_n	NU		CE1_1_n	CE0_1_n	PSL_1 ZQ_1	NU	
J		VSS	VCC	R/B0_0_n	R/B1_0_n		R/B1_1_n	R/B0_1_n	VCC	VSS	
K		NU	PSL_0 ZQ_0	CE0_0_n	CE1_0_n		VPP NU	WP_0_n	NU	VDDi NU	
L		VSSQ	VCCQ	CE2_0_n	CE3_0_n		CLE_0	ALE_0	VCCQ	VSSQ	
M		DQ7_0	DQ6_0	VREF_0 NU	WE_0_n CLK_0		RE_0_c	DQS_0_c	DQ1_0	DQ0_0	
N	NU	VSSQ	DQ5_0	VSSQ	RE_0_n W/R_0_n RE_0_t		DQS_0_t	VSSQ	DQ2_0	VSSQ	NU
P	NU	VCCQ	VCCQ	DQ4_0	VCC		VSS	DQ3_0	VCCQ	VCCQ	NU
R	NU	NU	NU						NU	NU	NU
T	NC	NC	NC						NC	NC	NC
U	NC	NC	NC						NC	NC	NC

Figure 4.16: Ball-down, top view of the BGA-132 package with the signal-pad assignments [17]. The layout will guide the PCB design, connecting the correct signals to the pads.

Supply Voltages

The ONFI standard defines four different DC voltages: V_{cc}, V_{ccQ}, V_{ss} and V_{ssQ}. Power is supplied through the two main voltage rails: main supply voltage V_{cc} and output stage logic supply voltage V_{ccQ}. These supply voltages are both relative to the source voltages V_{ss} and V_{ssQ}.

The main supply voltage V_{cc} has a nominal voltage of 3.3V. The output stage logic power voltage V_{ccQ} is intended to power the output transistors to supply the energy to the load applied to the input/output pins. The Micron B17A memory uses both 1.8V and 1.2V for I/O signaling. 1.8V V_{ccQ} shall be used for SDR and NV-DDR2 data interfaces and 1.2V for the NV-DDR3 data interface.

The recommended DC operating conditions for both supply voltages are shown in Table 4.2.

Table 4.2: DC supply voltage operating ranges. Based on Table 7 in [17].

Parameter	Symbol	Min	Typ	Max	Unit
Supply Voltage	V_{cc}	2.7	3.3	3.6	V
Supply Voltage for 1.8V I/O signaling	V_{ccQ}	1.7	1.8	1.95	V
Supply Voltage for 1.2V I/O signaling	V_{ccQ}	1.14	1.2	1.26	V

Signal Voltages/Logic Thresholds

Digital devices represent binary data by encoding different analog voltage levels. The memory operates with a V_{ccQ} of 1.8V for SDR, a voltage near 1.8V will be seen as "high" and represented as binary 1, while a voltage near 0V will be "low" and encoded with a 0.

The threshold values for each logic state are defined for input low/high: VIH, VIL in Table 4.3.

Table 4.3: DC characteristics for single ended signals. Based on Table 12 in [17].

Parameter	Symbol	Min	Typ	Max	Unit
DC input high voltage	VIH(DC)	V_{ccQ}*0.7	-	V_{ccQ}+0.3	V
DC input low voltage	VIL(DC)	-0.3	-	V_{ccQ}*0.3	V

R/B#: pull-up Resistor

The Read/Busy signal is the only memory output signal aside from the 8-bit data bus. It indicates the readiness of the memory to receive the next instruction.

ONFI document describes that R/B# is open drain output and will therefore require an external pull-up resistor to V_{ccQ}. A guideline to select the resistance value for the pull-up resistor is not given. The Micron datasheet indicates the relationship between the resistance of the pull-up resistor and the rise time of the R/B# signal, quoting values between 2k to 10k Ω for a total capacitive load of 100 pF.

4.3 Hardware Development

The necessary hardware is developed or acquired in order to perform operations on the 3D NAND memory. The ONFI hardware requirements were addressed in Section 4.2.2. This section will give an overview of the different hardware components to provide an integrated solution.

The main items which of the test setup are listed below:

3D NAND memory The Device Under Test (DUT) is a 3D NAND memory comprised of a standard 132-BGA package.

132-BGA socket A BGA socket provides the (electrical) interface between the DUT and the PCB. The socket allows for the necessary flexibility to test different DUTs.

Custom PCB A custom designed PCB interfaces the external memory controller with the DUT placed in the socket. It performs the necessary voltage conversions to provide the correct voltages to the chip.

ZedBoard™ development board The memory controller on the FPGA will generate the command signals for the memory. The ZedBoard is connects to the PCB with two cables and to a computer for communication.

Pmod USBUART module A Pmod extension module allows for FPGA-Computer communication using the UART protocol over a USB cable.

4.3.1 FPGA: ZedBoard™

The ZedBoard™ is a development board based on the Xilinx Zynq®-7000 all programmable SoC. It features a dual core ARM Cortex-A9 and an FGPA architecture. The board was kindly provided by ESA TEC, and will run a VHDL ONFI memory controller, capable of generating the correct waveforms to interface with the Micron 3D NAND memory. The ZedBoard will be connected to a custom interface PCB board described in Section 4.3.3, and be connected to a computer to receive commands and send data.

The ARM processor can offload specific tasks to the FPGA, but the FPGA can also by used as a stand-alone unit. VHDL or Verilog code can be synthesized in Xilinx Vivado and uploaded directly on to the ZedBoard's FGPA over the JTAG port. The ONFI memory controller programmed in VHDL will be ran off the FPGA, and further discussed in Section 4.4.1.

The ZedBoard has four Pmod™ connectors available directly to the FPGA. Pmod is a standard for peripheral modules, defined by Digilent[5]. The Pmod connectors on the ZedBoard have a 12 pin female header form factor, supplying 3.3V, GND and 8 user definable signals, conforming to the LVCMOS 3.3V or LVTTL 3.3V logic conventions. These are conveniently chosen to interface with the memory and the computer, a more in depth discussion is provided in Section 4.4.3.

Two Pmod connectors will handle the data I/O and the command signals (JA1 and JB2). A third will be configured to communicate with a computer over USB (JC1). The ZedBoard with the attached connectors is shown in Fig. 4.17.

[5] digilentinc.com/Pmods/Digilent-Pmod_%20Interface_Specification.pdf

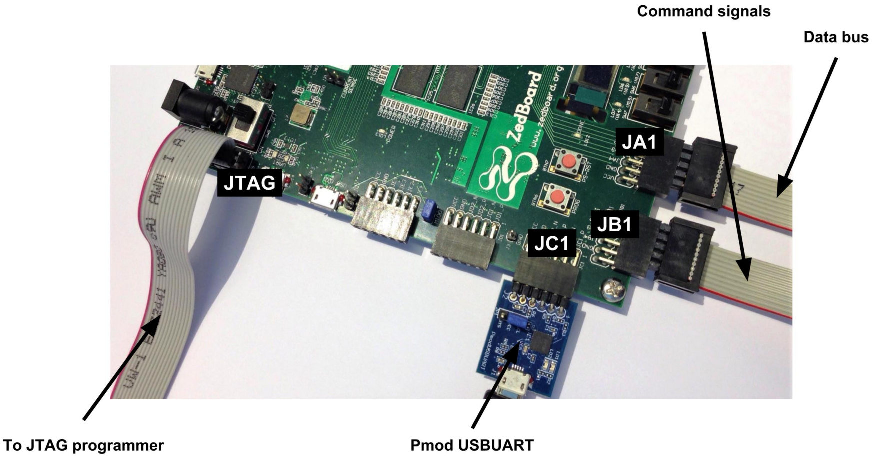


Figure 4.17: The ZedBoard with the connectors attached to the Pmod ports and JTAG programmer cable.

ZedBoard - PCB interface

Two Pmod connectors on the FPGA (JA1 and JB1) are configured in the controller software to serve as the 8-bit data bus and to channel the control signals.

Two ten core ribbon cables connect the FPGA with the 10 pin female header pins on the custom designed PCB interface board. A male-male header insert is placed between the female header of the ZedBoard Pmod connector and the female ribbon cable connector. A common ground is created between the PCB and the FPGA. The 3.3V port of the Pmod port is not used. An overview of the signal-pin assignment is given in Section 4.4.3.

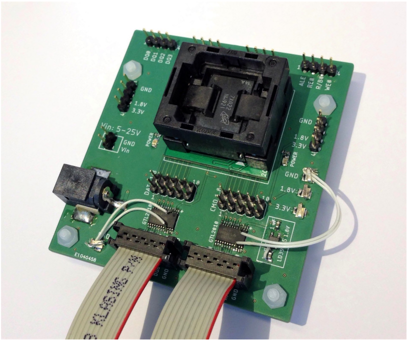


Figure 4.18: PCB with data cable (left) and command cable (right).

ZedBoard - Computer interface

A control interface to a main computer is desired to send instructions to the memory controller and to receive the page data downloaded from the memory. The Digilent Pmod USBUART board connects an receiving and transmitting signal to the data signals of the Pmod connector, while taking power through the 3.3V port. [6] It provides a "plug-and play" solution, where only the Pmod pin assignment needs to configured in Vivado. The FTDI FT232R chip on the board is theoretically capable of data transfer speeds up to 3 MBaud. A software description of the interface will be given in Section 4.4.3.

4.3.2 BGA-132 Interface: Socket

The signal layout for NAND flash BGA-132 packages is specified according to the ONFI standard, described in Section 4.2.1. The 8-bit data bus, the control signals and the necessary power lines need to be connected to the correct pads of the package. In order to connect the memory controller to the chip, a physical connection needs to be made with the pads. The following three main solutions could be proposed:

[6] store.digilentinc.com/pmod-usbuart-usb-to-uart-interface/

- **Soldering wires to the chip**
  After desoldering and reballing the chip, thin wires could be soldered directly onto the solder pads of the BGA package. This concept is proven to work for BGA packages with 0.3mm PTFE wires [20].

- **Soldering the memory to a custom PCB**
  A reballed package could again be soldered onto a PCB with a BGA-132 pad layout. It is desired to have the proper soldering tools available to prevent damaging the chip. The disadvantage is that there a little room for errors, and no replacement is possible. It is deemed a feasible option future work, where the package is soldered on an interchangeable board and connected to the controller/hardware interface.

- **Placing the memory into a BGA compatible socket**
  The unsoldered (and possible reballed) package is placed into a socket, where the pads are pushed onto (pogo) pins, forming an electrical connection. Pressure is applied on the top of the chip by a spring mechanism or a hinged/screwed interface. The advantage of this solution is that it is a non-permanent solution. Multiple DUT can be investigated with the same test hardware.

It was chosen to proceed the current development with a BGA compatible socket. In this first iteration where a proof of concept is the goal, it is desired to have enough flexibility, without dealing with soldering wires to the individual pads of the package.

In the following sections, an overview is made of the sockets available for purchase. A socket is chosen, ordered and characterized.

Socket market overview

An investigation looked at which manufacturers produce and sell off the shelf and custom BGA-132 sockets. A number of options are available from Chinese manufacturers through *Alibaba*. Several European data recovery companies, such as ACELAB and RUSOLUT, have used the Chinese components as their main socket around which a product is built. Custom high quality sockets designs are possible with Ironwood Electronics, a USA based company.

- **Alibaba Option 1: Clamshell**
  Clamshell design where the pressure is applied onto the memory with a surface pressing down on the top of the memory. The device is closed with a manual latch. A set of breakout pads allow for easy integration to another PCB. Prices between US$85.00[7] and US$105.80[8].

- **Alibaba Option 2: Open Socket with breakout board**
  Spring loaded open socket with two clamps pressing down on the memory. A set of breakout pads allow for easy integration to another PCB. For sale at US$128.00[9].

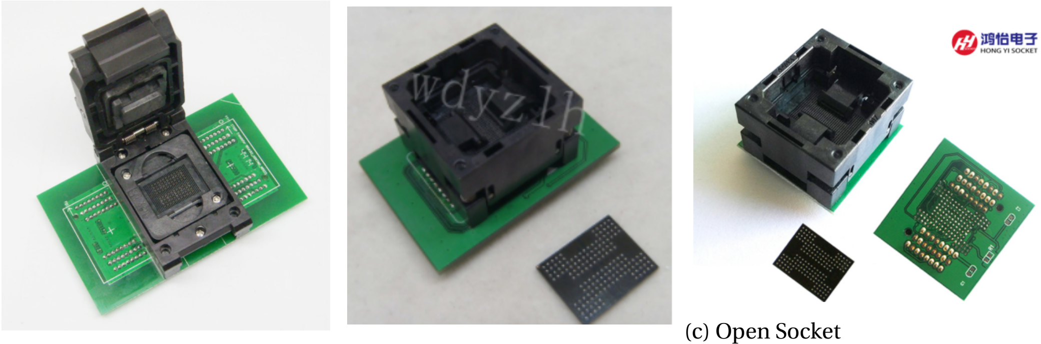
- **Alibaba Option 3: Open Socket with breakout board**
  The same socket as presented under *Alibaba Option 2*, but with a smaller breakout board footprint. For sale at US$59.00[10].

[7] vipprogrammer.com/bga132-bga152-to-dip96-96pin-8ce-ssd-test-socket-adapter-programmer-1759

[8] nl.aliexpress.com/item/BGA132-BGA152-Test-Socket-to-DIP96-adapter-for-SSD-8CE-test-ClamShell-\
socket-BGA152-Flash-Memory/32793860625.html?spm=a2g0z.10010108.1000023.12.192d4894PyJzTd

[9] iphsocket.com/product_info.php/cPath/42/products_id/71

[10] nl.aliexpress.com/item/BGA132-BGA152-to-DIP48-Adapter-IC-Test-Socket-BGA132-BGA152-Burn-\
in-Socket-Programmer-Socket-Open/32770127939.html?spm=a2g0z.10010108.1000023.8.7276175aPr52M6



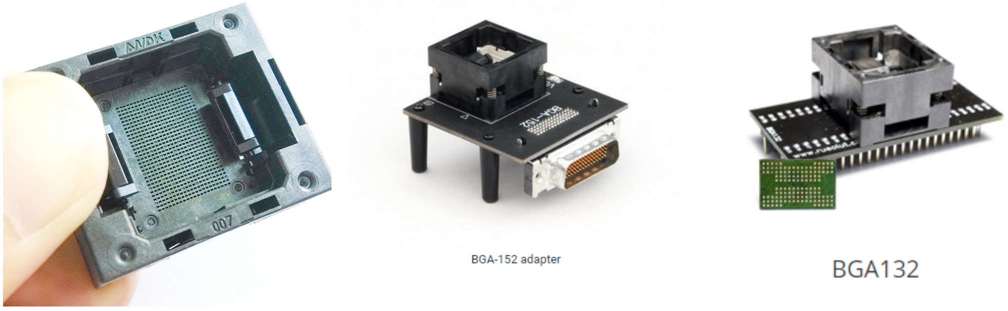
(a) Clamshell socket (b) Open Socket (c) Open Socket

Figure 4.19: Various sockets with breakout boards available through *Alibaba*.

- **Alibaba Option 4: Open Socket without breakout board**
  The socket used in *Alibaba Option 2: Open Socket* and *Alibaba Option 3: Open Socket* is also for sale, US$72.00[11] and US$73.00 [12]

- **Option 5: ACELAB**
  ACELAB is a "Professional Data Recovery Tools" business based in Czechia that has developed hardware to read out NAND memories. Their for sale hardware includes a "Open Socket" with a connector[13].

- **Option 6: RUSOLUT**
  RUSOLUT is a "Data Investigation & Recovery Tools" based in Poland that has developed hardware to read out NAND memories. They have developed a breakout solution based on a 0.1in header connection[14].



(a) Single socket (b) RUSOLUT socket (c) ACELAB socket

Figure 4.20: (a) Single socket to be soldered directly onto a custom PCB. (b)(c) Socket solutions with integrated interface.

- **Option 7: IronWood Electronics**
  Ironwood Electronics, a USA based company provides custom made "High Performance Sockets & Adapters".[15] A suitable socket was quoted at US$500. It contains various aluminium/steel parts that might disturb the beam line. A custom socket could be designed, an estimate of US$5000 was quoted for a socket design with breakout board.

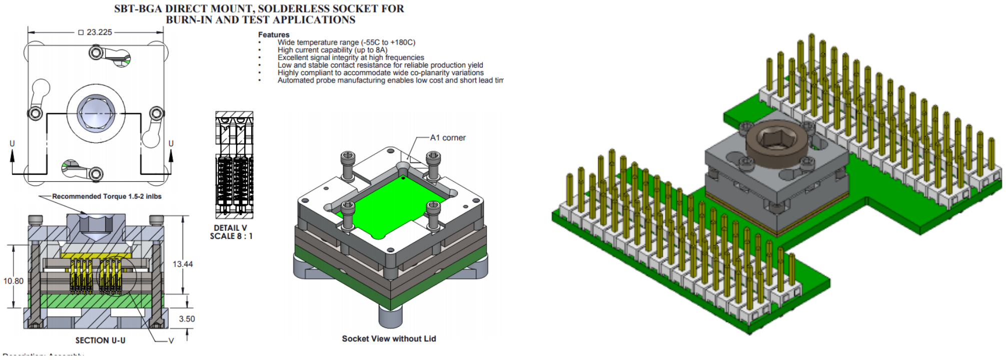
[11] iphsocket.com/product_info.php/products_id/125

[12] nl.aliexpress.com/item/BGA132-BGA152-Burn-in-Socket-BGA-Adapter-IC-Test-Socket-For-BGA132-BGA152-\
Flash-Testing-Programmer/32767790353.html?spm=a2g0z.10010108.1000023.10.7d6c44b5E6psQi

[13] acelab.eu.com/pc3000flash.php

[14] rusolut.com/visual-nand-reconstructor/nand-adapters/

[15] ironwoodelectronics.com


SBT-BGA DIRECT MOUNT, SOLDERLESS SOCKET FOR
BURN-IN AND TEST APPLICATIONS
□ 23.225
Features
• Wide temperature range (-55C to +180C)
• High current capability (up to 8A)
• Excellent signal integrity at high frequencies
• Low and stable contact resistance for reliable production yield
• Highly compliant to accommodate wide co-planarity variations
• Automated probe manufacturing enables low cost and short lead tim
U
U
Recommended Torque 1.5-2 inlbs
DETAIL V
SCALE 8 : 1
A1 corner
13.44
10.80
3.50
SECTION U-U
V
Socket View without Lid

(a)

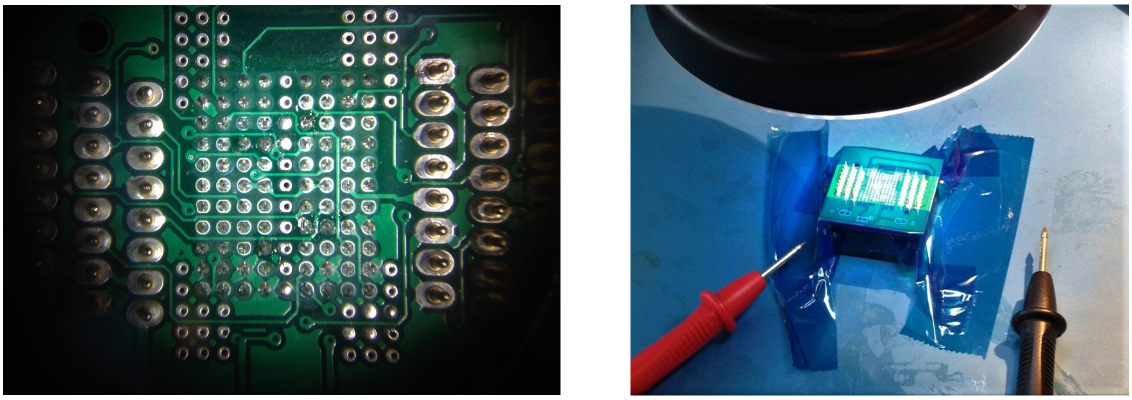
(b)

Figure 4.21: IronWood Electronics (custom) bracket solutions. Source: IronWood website.

Alibaba Socket characterization

The *Alibaba Option 3: Open Socket with breakout board* was chosen as it was the cheapest option, while the open socket solution would prevent any with the radiation beam if being tested while the DUT is mounted (up to a certain angle of incidence).

The socket did not have a datasheet attached with the pin specifications, which are connected to the ONFI NAND flash signals. To investigate the mapping, the socket is turned upside down and the connection of each pin is checked with a multimeter. Fortunately, the leads of the socket are protruding through the bottom of the breakout PCB, allowing for convenient checking. Caution must be taken: when the socket is placed upside down, the pads are horizontally mirrored.



(a)

(b)

Figure 4.22: (a) Central pins connect straight up to the pads on the chip. 48 protruding pins located left and right of the central area connect to the chip. (b) Mapping the connections under a microscope with a multimeter.

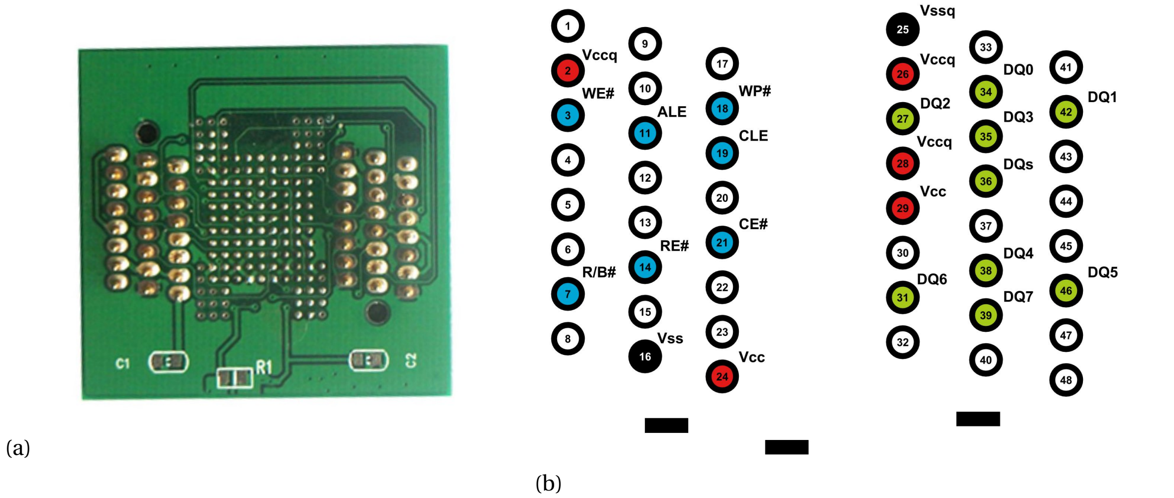


Figure 4.23: (a) Image of the bottom side of the socket with the PCB. (b) Mapped pins to the external leads of the socket board. The black rectangulars on the bottom represent placeholders for SMD as seen on the socket PCB in (a).

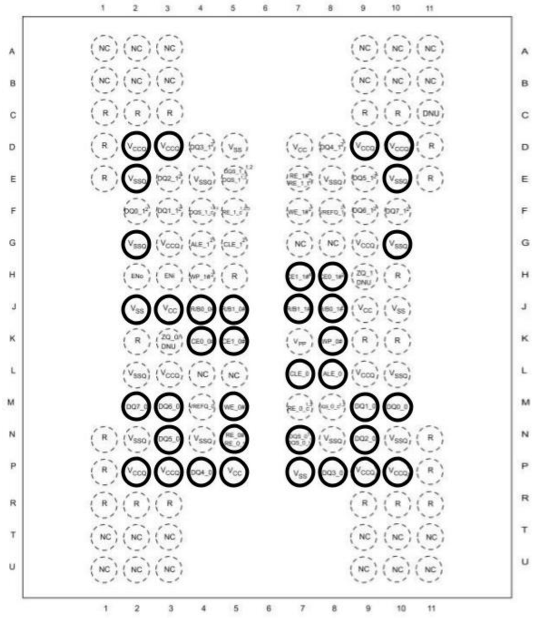


Figure 4.24: Ball-Down, Top View of the BGA-132 signal assignment. The marked pads are connected to the pins on the PCB.

4.3.3 PCB design

A custom PCB was designed and manufactured to interface the socket with the ZedBoard. The main function of the PCB is to provide a structured interface between the memory controller and the memory chip. The PCB shall supply the correct voltages to the chip, and convert input/output signals to the correct voltages to be registered by the memory and the Pmod ports on the Zedboard.

The schematics and PCB layout was done using EAGLE, Autodesk's PCB design software. The final design is a 70 x 75 mm^2, two layer PCB, printed by EuroCircuits and populated in-house with mostly surface mount components, ordered from Farnell.

An overview of the main design features are given next.

Power supply

The power shall be applied from an external supply to the board, where the correct voltages are provided to the memory chip. The power can be supplied to a 2 mm female barrel jack (compatible with a wall mount adaptor) or to a 2x1 male header pin header.The supply voltage should be between 5 and 24 V.

As defined in Table 4.2, the main supply voltage V_{cc} to the memory is 3.3V and the supply voltage for SDR signaling,V_{ccQ}, is 1.8V. Two linear voltage regulators are used to provide the 3.3V and 1.8V, given any supply voltage between 5 and 25 V. The first regulator (LM2937IMP-3.3/NOPB) takes an input voltage between 4.75 V and 26 V, whilst providing a stable 3.3V output. A second regulator (LD39015M18R) is placed on the output of the first regulator, itself providing a stable 1.8V output.

Although stable voltages could have been applied by an external power supply, the use of the two linear regulators provides a built-in overvoltage protection mechanism to prevent from damaging the DUT. In order to further protect the regulators, a Zener diode is placed in parallel to the first regulator to ground. The chosen Zener diode has a Zener voltage of 24V, (1W, KDZVTR24B) and is placed in series with a 620 Ohm resistor. Two red LEDs are connected to the 3.3V power plane. It provides a quick visual reference check to indicate whether or not the board is powered.

The schematic designs for the full power supply subsystem are shown in Fig. 4.25 and 4.26.

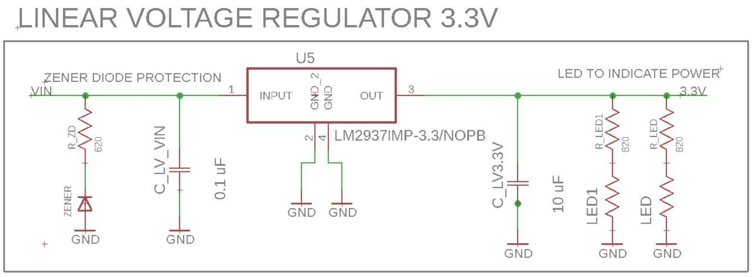


Figure 4.25: Schematic of the main input power regulator, including two LEDs to indicate the power.

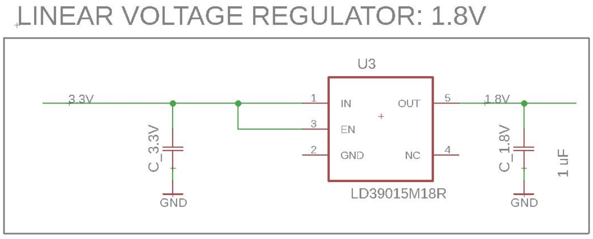


Figure 4.26: Second stage linear voltage regulator to 1.8V.

Voltage Conversion

The control and data signals generated by the memory controller on the ZedBoard are sent through two Pmod ports. These ports are based on the LVCMOS 3.3V or LVTTL 3.3V logic conventions. Because the signaling in SDR mode of the memory is based on 1.8V, a down conversion to send signals to the memory is needed. Likewise, the conversion up to 3.3V is needed to comply with the logic convention used by the Pmod ports.

A suitable solution is the use of a bidirectional voltage translator, which can translate both received and sent signals. The Texas Instruments GTL2010 (SN74GTL2010PW) was chosen, the IC in a TSOP package can handle up to 10 channels. The chip is given a reference input and output voltage. Two GTL2010 chips provide the conversion, one for the eight data signals (DQ0-DQ7) and one for the command signals.

1k Ohm Pull Up resistors to V_{cc} (3.3V) are placed on the output data signal lines (DQ0-DQ7), which serve as an input to the Pmod connector. As seen in the schematics in Fig. 4.27 and 4.28, a placeholder Pull Up and

Pull Down resistor are drawn for all the signal lines to the memory chip after the level shifter. These are only populated on the command lines, where the *active low* signals (WE#, RE#, CE# and WP#) are pulled high to 1.8V using a 1k Ohm resistor. The Read/Busy signal (R/B#) has been given a dedicated 3.3k Ohm resistor. The other signals (ALE, CLE) are pulled low to 0V using a 1k Ohm resistor.

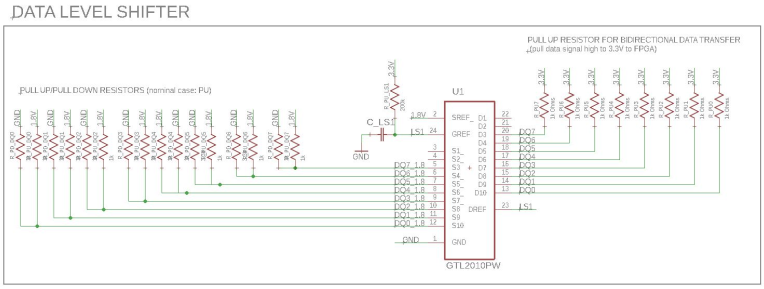


Figure 4.27: Schematic for the IO data lines level shifter. Because data is also sent back to the Pmod ports on the FPGA, Pull-Up resistors are placed.

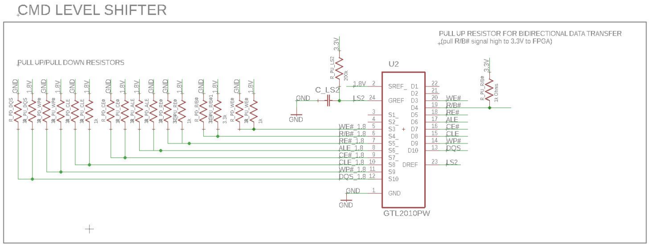


Figure 4.28: Schematic for the second level shift, dealing with the command signals. A dedicated Pull Up resistor is added for the R/B# line.

Connectors and Test points

The interface between the PCB and the ZedBoard is based on two 5x2 male header pin connectors, connected by two ribbon cables to the Pmod port, as was shown in Fig. 4.18. A header pin connector is placed both before and after the level shifter: if a new controller is being used that is able to already provide 1.8V, it can be directly connected. The signal assignment to each of the header pins (DQ0-DQ7 and the command signals) are based on the convenience of the hardware design, such that the signal lines on the PCB can be connected to the socket needing a minimal number of vias crossing other signal lines. The signal pins can be configured in the ZedBoard to the required Pmod pin positions.

In addition to the signals, a common ground is established between the PCB and the Zedboard by connecting the ground of the Pmod port with the ground on the PCB. The schematic for the four male header pins is shown in Fig. 4.29.

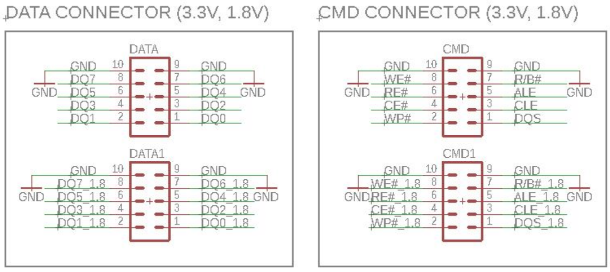


Figure 4.29: The main interface is done through two header pins before the level shifter. A second set of headers is placed after the level shifter, they could provide a direct 1.8V interface to the memory.

A series of test points are added to allow for quick troubleshooting. Six 4x1 male header pins are placed on the edge of the board, making them easily accessible to the probes of a logic analyzer. Four header connectors are attached to each of the signals. Two are placed to probe the ground, 1.8V and 3.3V. Additional surface mount test pins are added on the board to probe the ground, 1.8V and 3.3V.

Socket Interface

The signal and voltage lines are all connected to the correct pins of the socket, which are defined in Section 4.3.2. The BGA socket was imported as a custom component and wired accordingly, as shown in Fig. 4.30. Because no data sheet was available on the physical layout of the pin, the distances between the pins were measured and drawn in the new component. A copy of the pin layout is given in Addendum A.

As can be seen, a capacitor is placed between each supply line and ground to suppress any noise coming from the power supply line.

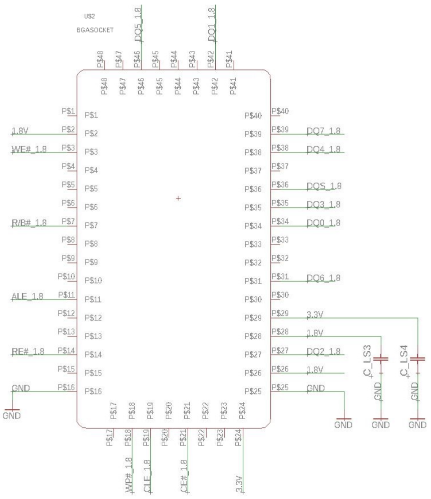


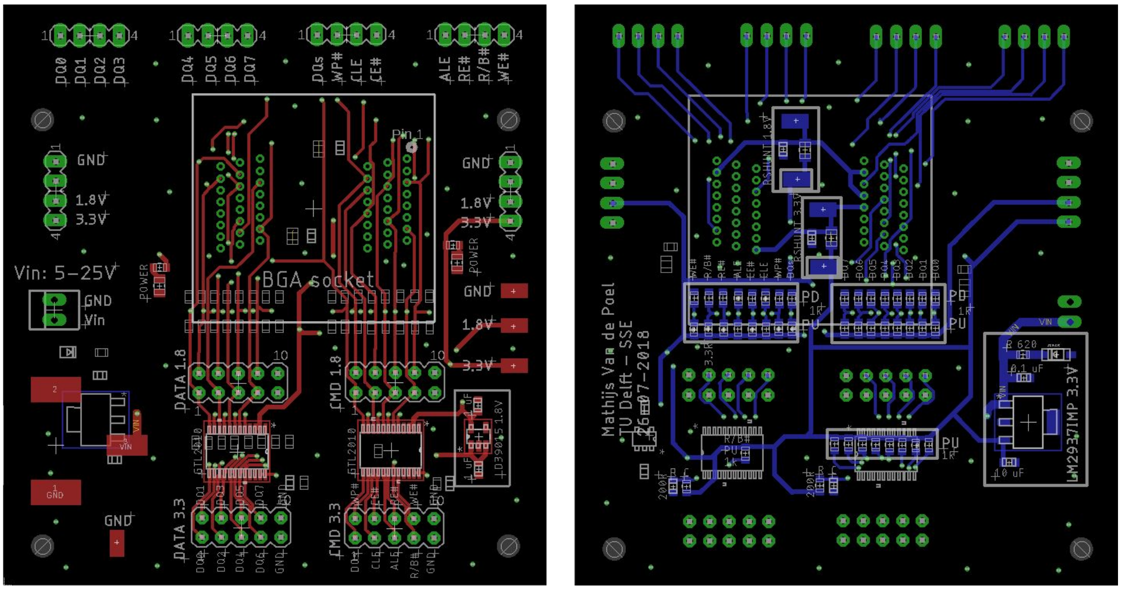
Figure 4.30: The socket interface board was made as a custom component to which all the signals are connector.

4.3.4 PCB Layout

The schematic of the PCB design presented above is converted into a printable PCB design. The board layout is structured around the centrally placed BGA socket. The interface connector points (5x2 header pins) are placed in line with the logic level shifters and their respective connectors on the socket. This efficient layout shortens the lines and makes it easy to connect the two ribbon cables between the ZedBoard and the PCB. The test points to probe the signals with a logic analyzer are placed on the opposite side of the interface connectors. Having all the test points lined up and labeled, makes it easy to connect all the probes from the logic analyzer, used for trouble shooting.

Four drilled holes provide the option to connect spacers to keep the PCB elevated from the surface below, or to attach the board to an interface in a test environment.

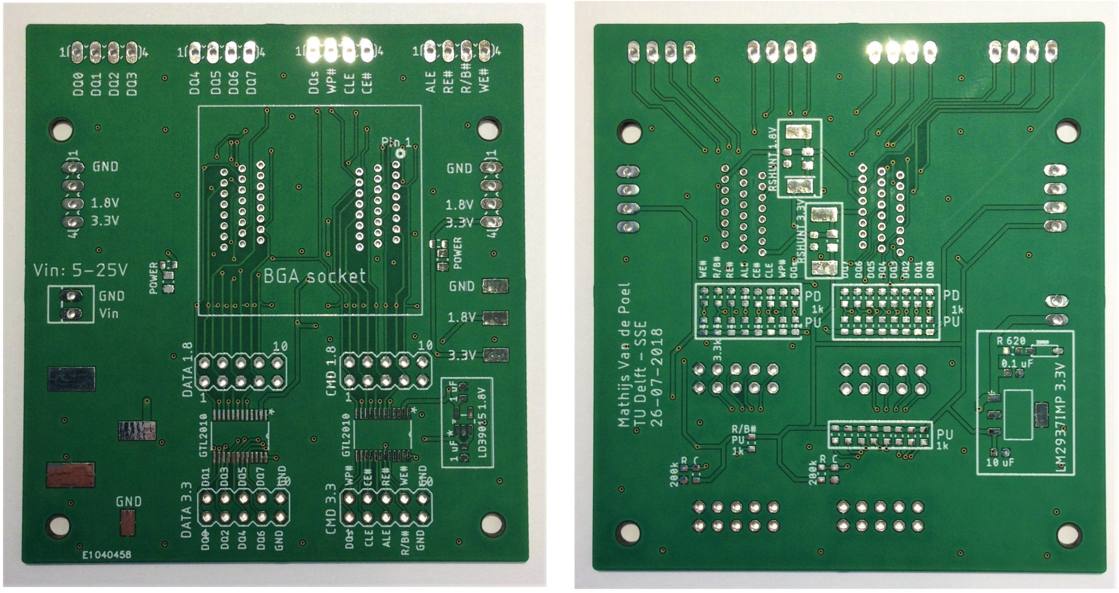
The EAGLE design is shown in Fig. 4.31 and the empty boards received after ordering from EuroCircuits can be seen in Fig. 4.32.



(a)

(b)

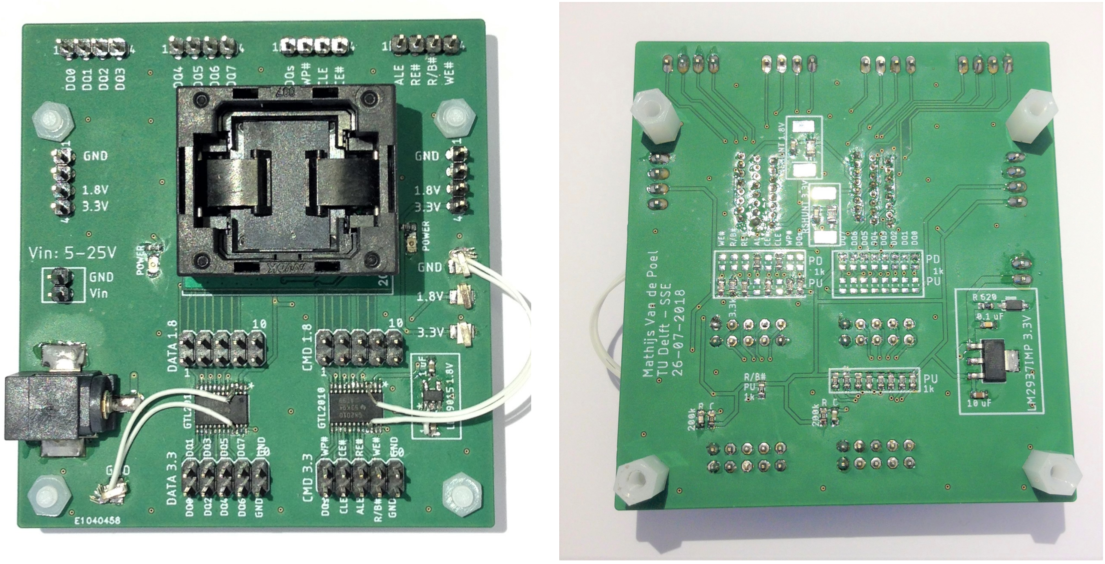
Figure 4.31: Front and back of the PCB design in EAGLE. The socket is placed in the center of the board, the lower side is dedicated to the interface with the ZedBoard, while the top side has the necessary pins available to be probed by a logic analyzer. The barrel jack and header pin to supply power are positioned on the left side.



(a)

(b)

Figure 4.32: The front and back side of the empty PCBs after ordering through EuroCircuits.



(a)

(b)

Figure 4.33: Front and back side of the fully assembled board. Four plastic spacers are placed in each of the corners to keep the board elevated from the workbench surface. It can be seen that a 132-BGA memory is inserted in the centrally located socket.

4.4 Software Development

The test setup is comprised of multiple software solutions running across different platforms. An overview will be given here as to what tools are used, how they are set up, tested and the function of each software unit.

The software system is comprised of the following main items:

ONFI Memory Controller A program that is capable of generating the correct ONFI instructions on the Pmod ports, to execute operations on the memory. Programmed in VHDL, synthesized with Vivado and then uploaded onto the Zedboard.

UART Communication A routine that is able to both receive and send commands through the Pmod USBUART module from the ZedBoard. It will receive the commands to instruct the memory controller which commands to execute. Programmed in VHDL, synthesized with Vivado and then uploaded onto the Zedboard.

VHDL Controller Testbench A digital testbench to simulate the synthesized ONFI memory controller and UART communication. It will replicate the signals that are generated by the controller that will be used to control the memory. Computer based simulations using ModelSim.

Python Control Module Python based scripts from which the desired commands are sent to the FPGA to execute instructions on the memory. It automates running repeated test sequences. Scripts store the test data. Furthermore, it is the main tool for data processing.

4.4.1 ONFI memory control: instructions

The main goal is to have a software solution which is able to communicate with the rawNAND flash memory. As described in Section 4.2.1, the control instructions are defined by the ONFI standard. The software capable of generating these instructions is the ONFI memory controller presented here.

The core of the memory control code is based on an open source VHDL ONFI memory controller available through OpenCores.[16] OpenCores is an online community for the development of free and open source IP cores. This particular project was developed by Alexey Lyashko. Mr. Lyashko is aware of the use of his code for this thesis, and is available to contact at *alexey@bitmazing.com*.

The controller is based on a Finite state machine (FSM). The different steps of sending the commands are each a finite state which are being run through.

The ONFI specification defines four main control signals (ALE, CLE, RE# and WE#) with which the four main instructions (Command, Address, Data In, Data Out), can be constructed. The two latch units: Address Latch Enable (ALE) and Command Latch Enable (CLE) are set by a separate module, called from the main file: *latch_unit.vhd*. The latch unit latches the commands and addresses into the memory (signals nand_ale and nand_cle). The Read Enable (RE#) and Write Enable (WE#) are called from another module: *IO_unit.vhd*. The IO unit latches the writing of data available onto the bus, or requests the memory to present data onto the data bus (signals nand_nre and nand_nwe).

The controller runs through the main routines by selecting a *STATE* of the FSM. Where each state runs a certain operation on the memory. After successfully completing a routine, the controller returns to an *IDLE* state where it awaits its next instruction.

An overview is given below of commonly used instructions that are originally programmed in the controller code, and the commands that were later added. The bolded description indicates the command defined

[16] `opencores.org/project/nand_controller`

in the VHDL controller code. The next capitalized command is how it is addressed according to the ONFI standard, including the hex values for the command to be latched into the memory.

Commonly used instructions originally programmed

- **M_NAND_READ**
  READ PAGE (00h-30h)
  Main command used to read a page at the desired memory address. Will be issued after setting the desired read offset voltage. By toggling the RE# signal, a new byte will be made available on the data bus. The controller then stores the page data locally on the FPGA, after which it can be read out to the computer.
- **M_NAND_RESET**
  RESET(FFh)
  Must be issued after the memory is powered up (as defined by Device Initialization). Can be used to restore the memory after a freeze up.
- **M_NAND_BLOCK_ERASE**
  ERASE BLOCK (60h-D0h)
  Erases a specified block in the memory. This is needed before writing a fresh set of pages into the memory. Only the row address is given.

Custom added instructions to the original instruction set

- **M_NAND_SET_READ_OFFSET**
  SET FEATURES (EFh)
  The read offset is a feature implemented by issuing the SET FEATURES (EFh) command. It is further specified which level read offset is to be set. In this case, the level is set to ABh, referring to the *rL7* reference level. Within the command, the P1-P4 subfeatures are set. P1 defines the offset voltage, between -x and +x V. The readed is advised to study Table 35 in the Micron B17A data sheet [16]. The reference level to which the offset is applied (ABh) can be changed in the VHDL code.
- **M_NAND_PAGE_PROGRAM_HIGH**
  PROGRAM PAGE (80h-10h)
  Writes the addressed page to logic state 1.
- **M_NAND_PAGE_PROGRAM_LOW**
  PROGRAM PAGE (80h-10h)
  Writes the addressed page to logic state 0.
- **MI_GET_NAND_PAGE**
  This command returns the page data from a READ PAGE instruction from the buffer on the FPGA to the computer over the UART communication link.

Adaptations to the original controller software

Aside from extending the instruction set, a small set of changes was made to the original code that should be mentioned.

The first change is based on the lacking reliability of the controller to read the R/B# signal. The M_WAIT state is waiting for a response from the R/B# signal before proceeding to the next state. During testing, this was found to be not reliable, the signal was not always properly latched into the controller, causing the FSM to stay put in this state. The M_WAIT state was therefore replaced with the M_DELAY state, where instead of waiting for a response from the R/B# signal, the next state is initiated after a preset delay time. The delay times can be based on the times specified by the Micron manual for each instruction, therefore it was deemed a robust solution without the need of modifying any hardware.

A second and small change is to set the Write Protect and Chip Enable signals to their *Off* (Write Protect) and *On* (Chip Enable) states. These signals are not used during a testing routine and the chip should be available to the controller.

4.4.2 UART Communication

A UART communication protocol was implemented to control the memory controller program running on the ZedBoard FPGA. A communication line to the controller is desired to select the instruction, set a target address (where to execute the instruction) and set the desired offset voltage. Secondly, the pages read from the memory are stored in a buffer on the FPGA, which should first be downloaded to a permanent storage solution (in this case: on the user's computer), before overwriting it with new data.

The UART communication system was chosen after investigating the different possibilities of providing a data connection between the FPGA and an external entity. Fig. 4.34 indicates the I/O ports available through the different parts of the ZedBoard Zynq-7000 SoC chip. These ports could be used to set up a channel of communication with the ONFI memory controller. Note the difference between the Processing System (PS) and the Programmable Logic (PL). The ARM processor runs in the Processing System block, while the FPGA (used thus far to control the memory) is based in the Programmable Logic (PL) block.

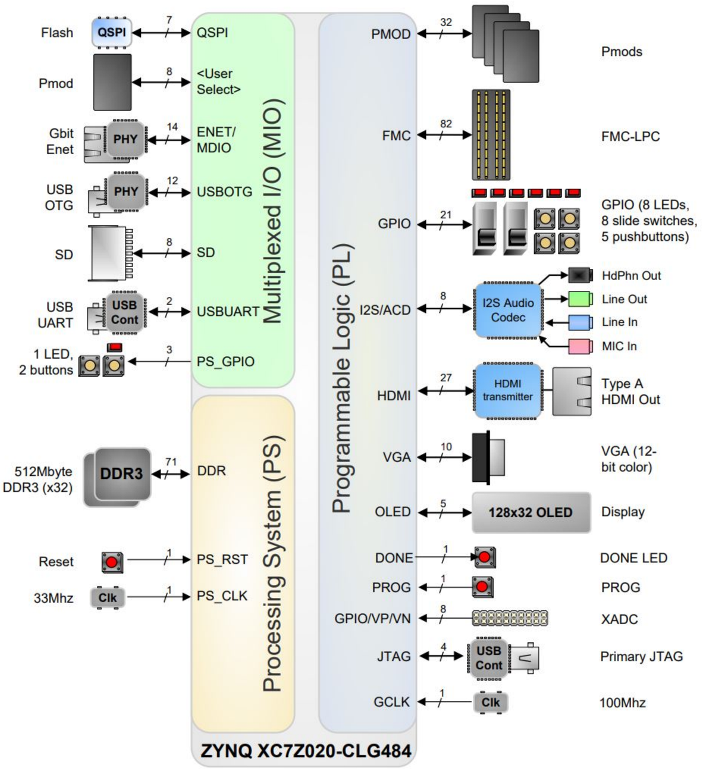


Figure 4.34: The ZedBoard block diagram, indicating the availability of I/O ports to the Processing System, Programmable Logic and Multiplexed I/O. It is desired to have a communication solution based on solely the Programmable Logic part (the FPGA), on which the ONFI controller is programmed [9].

Interesting ports have a high speed data throughput, are for example Ethernet and USB. However, these are not directly available to the Programmable Logic (PL) part of the ZedBoard. Another interesting feature in the Processing System (PS) is the presence of local storage in the form of a DDR3 memory, or expandable data storage through and SD card.

The theoretical implementation for the use of these PS ports and data storage solutions is based on creating an AXI interface between the PL and the PS in Vivado. One can create a custom IP block containing the VHDL controller. Through an AXI DMA interface, data could be passed to the DDR3 memory, from where it could be either locally processed on the ARM or transferred through the USB or Ethernet port. This option was investigated but abandoned because of its complex implementation.

However, 4 Pmod connectors (Peripheral Module Interface) are directly accessible by the PL, and are also used for the communication with the PCB containing the DUT as described in [SECTION]. The choice was made to implement a solution solely based on the Programmable Logic part of the ZedBoard, by implementing a serial UART communication through a Pmod port.

UART Communication through Pmod

A serial communication protocol can be implemented, where one pin of the Pmod port acts as a receiver and another as a transmitter. This is exactly what the Pmod USBUART extension board does, presented before in Section 4.3.1 and shown in Fig. 4.35. It provides a micro USB interface, which is convenient for connecting the board to a computer. The FT232R chip on the extension Pmod board is capable of transferring serial data up to 3 Mbaud. The serial UART signals will be generated or received accordingly by the VHDL controller described below.

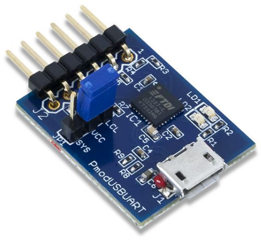


Figure 4.35: Pmod USBUART extension board. Source: Digilent.

The basic UART data block contains a start bit (low), one byte of data, finished with a stop bit (high), shown in Fig. 4.36. The speed can be increased by decreasing the time of each signal.

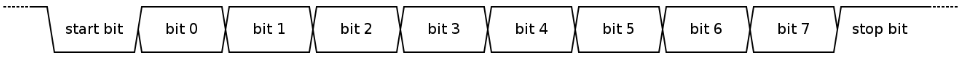


Figure 4.36: Data Framing for the UART. Source: Wikipedia.

UART Controller

A UART controller is able to extract the data bytes from an incoming binary data stream, by detecting the start and stop bits. It is also able to generate the stream of data, and correctly placing the start and stop bits for each sent byte. A basic version of an UART receiver and transmitter was implemented in VHDL on the FPGA. The code was based on an example of NANDLAND[17]. The website also provides a good source of information on the use of UART communication. The UART controller is again based on a Finite State Machine, running through different states as it is receiving and sending data.

Interface with ONFI Memory Controller

A custom VHDL based FSM machine was added to provide a buffer for the incoming data and to interface with the ONFI controller program. Based on the received byte by the UART control routine, it can send an instruction to the ONFI memory controller, set a buffer with six address bytes in preparation for an instruction, or set the four offset bytes needed to set a threshold offset.

The following four FSM states are defined, which are entered by writing the arbitrarily chosen hex commands to the FPGA.

- **Idle** (01h)
  Resets the communication interface back into *Idle*.
- **Command** (AAh)
  Receives 1 byte that contains the decimal number of the instruction to be executed on the ONFI controller.
- **Addr** (CCh)
  Receives 6 bytes that will be stored in the Address buffer, to be used by the ONFI controller when writing to or reading from a memory page.
- **Offset** (33h)
  Receives 4 bytes that will be stored in the Offset buffer, to be used by the ONFI controller. The data will contain the offset value applied by the READ OFFSET function.

After receiving the expected number of bytes, the interface controller FSM returns to Idle, awaiting the next command.

4.4.3 Software Setup on the ZedBoard

Bitstream generation using the Xilinx Vivado toolbox

The ONFI memory controller and UART communication protocol are programmed in VHDL, several more steps are needed before the code can be run on the FGPA: the code needs to be compiled into a bitstream, which is then uploaded to the ZedBoard's FPGA using a JTAG programmer.

[17] `nandland.com/vhdl/modules/module-uart-serial-port-rs232.html`

The process of code synthesis, implementation and bitstream generation is handled by Xilinx' Vivado toolbox. In the logic synthesis process, the hardware description language (HDL), VHDL in this case, is transformed and optimized into a gate-level netlist. The results of Vivado's synthesis process for the used VHDL code is visualised in Fig. 4.37. The following implementation phase places and routes the schematic of the synthesized code onto the FPGA. The defined signals in the constraints file are set to the requested pins on the Zynq chip. The code configured onto the FPGA gates and the Zynq pins placements are shown in Fig. 4.38. Finally, Vivado generates a bitsteam, this is a sequence of bits, describing the configuration to be loaded onto the FPGA. When the ZedBoard is powered up and connected to the computer, it will be recognized by Vivado, from where the bitstream is programmed. A confirmation message will appear after successful after a successfully completed upload. Additionally, a blue led (LD12) will light up.

Note that the bitstream needs to be uploaded each time the ZedBoard is turned off/on.

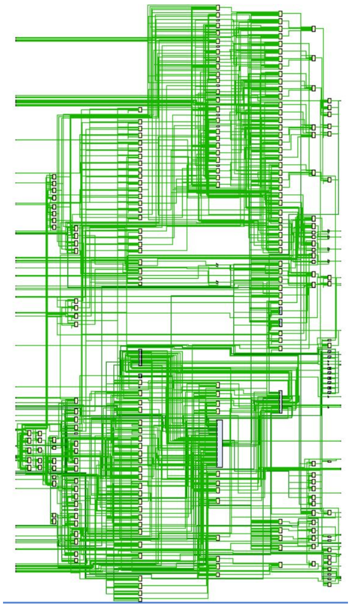
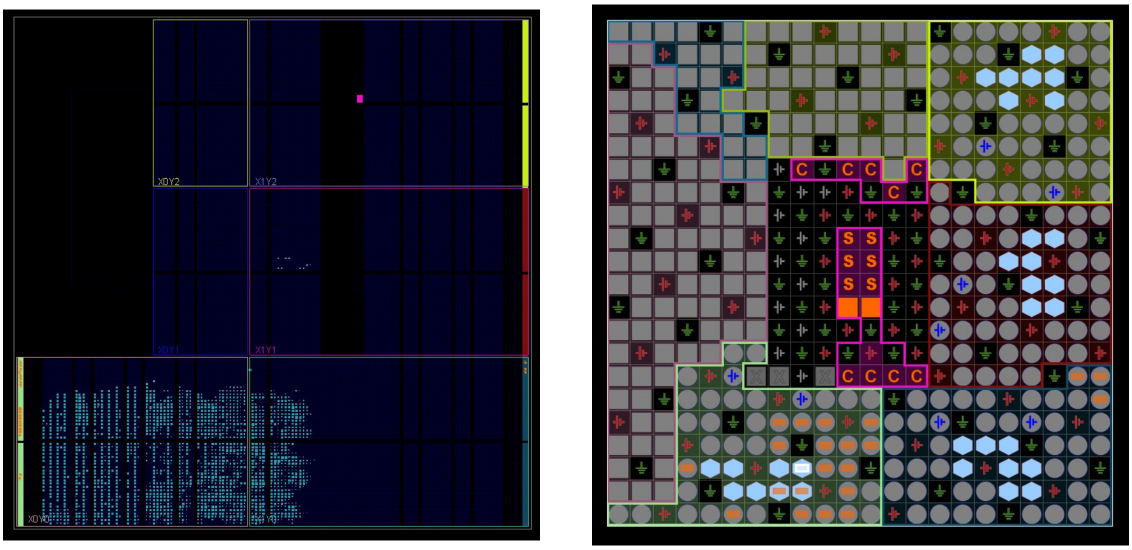


Figure 4.37: The Vivado synthesized VHDL code as a gate-level netlist.



(a)

(b)

Figure 4.38: (a) A visualization of the programmed logic on the ZedBoard FPGA. (b) Pin assignment on the Zynq chip, routed to the peripherals of the development board.

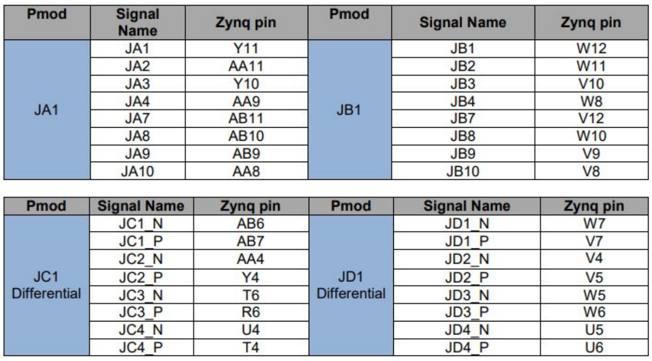
The current, unoptimized, VHDL code has the following FPGA resource utilization:

Resource	**Utilization**
LUT	5675
LUTRAM	2336
FF	2583
IO	22
BUFG	1

Pmod connection setup in Vivado

Three Pmod connectors on the Programmable Logic (PL) of the ZedBoard will be used to communicate with the memory and a computer. It is necessary to define the connections between the signals in the VHDL code and the correct Zynq pins, which are tied to the Pmod connectors. Two Pmod ports (*JA1* and *JB1*) will be configured for I/O data and commands, while a third Pmod port (*JC1 Differential*) is used for UART communication. The connections between the software and hardware are defined in a *Configuration file*, which Vivado takes into account when compiling the VHDL code.

The Pmod signal assignment (*Signal Name*) and the Zynq pin assignment (*Zynq Pin*) are tabulated in the ZedBoard Hardware User's Guide and are shown in Fig. 4.39.



Pmod	Signal Name	Zynq pin	Pmod	Signal Name	Zynq pin
JA1	JA1	Y11	JB1	JB1	W12
	JA2	AA11		JB2	W11
	JA3	Y10		JB3	V10
	JA4	AA9		JB4	W8
	JA7	AB11		JB7	V12
	JA8	AB10		JB8	W10
	JA9	AB9		JB9	V9
	JA10	AA8		JB10	V8

Pmod	Signal Name	Zynq pin	Pmod	Signal Name	Zynq pin
JC1 Differential	JC1_N	AB6	JD1 Differential	JD1_N	W7
	JC1_P	AB7		JD1_P	V7
	JC2_N	AA4		JD2_N	V4
	JC2_P	Y4		JD2_P	V5
	JC3_N	T6		JD3_N	W5
	JC3_P	R6		JD3_P	W6
	JC4_N	U4		JD4_N	U5
	JC4_P	T4		JD4_P	U6

Figure 4.39: In order to connect signals from the FPGA to the Pmod connectors, a connection needs to be defined in a *configuration file*. This table indicates the relationship between the physical Pmod pins and the Zynq pins, to which the signals are connected [9].

An overview of the connections between the signals defined in the ONFI controller code and UART communication, the connection on the interface PCB and their designated Pmod pins are shown in Table 4.4, 4.5 and 4.6.

Table 4.4: Configuration of Pmod JA1: dedicated to the 8-bit data bus.

PMOD	ZYNQ PIN	PCB PIN	SOFTWARE
JA1	Y11	DQ0	nand_data[0]
JA2	AA11	DQ2	nand_data[2]
JA3	Y10	DQ4	nand_data[4]
JA4	AA9	DQ6	nand_data[6]
JA7	AB11	DQ1	nand_data[1]
JA8	AB10	DQ3	nand_data[3]
JA9	AB9	DQ5	nand_data[5]
JA10	AA8	DQ7	nand_data[7]

Table 4.5: Configuration of Pmod JB1: dedicated to the command signals.

PMOD	ZYNQ PIN	PCB PIN	SOFTWARE
JB1	W12	DQs	nand_dqs
JB2	W11	CLE	nand_cle
JB3	V10	ALE	nand_ale
JB4	W8	R/B#	nand_rnb
JB7	V12	WP#	nand_nwp
JB8	W10	CE#	nand_nce
JB9	V9	RE#	nand_nre
JB10	V8	WE#	nand_nwe

Table 4.6: Configuration of Pmod JC1 Differential: dedicated to the Rx and Tx signals to the Digilent USBUART extension board.

PMOD	ZYNQ PIN	PCB PIN	SOFTWARE
JC1_N	AB6	-	o_TX_Serial
JC2_P	Y4	-	i_RX_Serial

Use of the ZedBoard clock

The timings on the FPGA are based on the on-board 100 MHz oscillator. This clock is available to the VHDL software through pin Y9. This connection is also made in the *configuration file*, along with the connections to the Pmod connectors.

4.4.4 Python Control Scripts

In order to execute the desired instructions on the memory through the ONFI controller on the FPGA, a Python script sends the necessary commands over a serial port, connected with USB to the ZedBoard.

The control scripts first open a serial communication link using the pySerial package. For a windows machine, a COM port is selected and a baud rate is set. The necessary typeshifting is needed to provide the UART VHDL receiver with the correct commands. The commands are written using *ser.write(command_byte)*, where *command_byte* is the hex or decimal command converted into an 8-bit binary number.

Two main operations will be ran from a Python script:

- Program the memory cells of one or multiple blocks to a desired logic level.
- Determine the threshold voltage of the programmed memory cells.

Programming the memory cells to *L7*

All the memory cells of a block are normally programmed to the highest logic level, *L7*, to see the highest possible voltage threshold shift possible. A diagram of the different steps executed by the script is shown in Fig. 4.41. A set of comments to each of the steps is added below the figure.

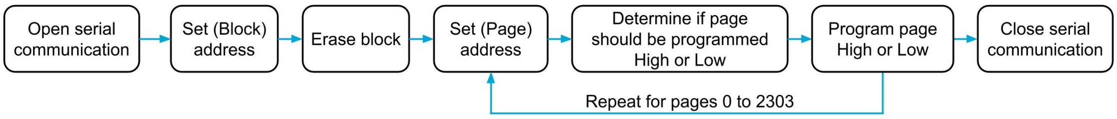

Open serial communication
Set (Block) address
Erase block
Set (Page) address
Determine if page should be programmed High or Low
Program page High or Low
Close serial communication
Repeat for pages 0 to 2303

Figure 4.40: Flowchart of the different steps executed by a Python script to program a memory block to a desired logic level.

1. **Open Serial Communication**, the pySerial communication channel is opened on the desired COM port with a selected baud rate.
2. **Set (Block) Address**, 6 address bytes are passed to the memory controller, on which the next operation (Erase Block) will be executed. First, the column, page, block and LUN are converted to the format specified in Section 4.2, then (CCh) is written, indicating the UART controller that 6 address bytes will follow. Then the 6 address bytes are writen to the FPGA. Because a block will be addressed, the column and page are not taken into account.
3. **Erase Block**, command AAh is passed to the UART controller, preparing it to pass an instruction to the ONFI memory controller. Instruction number 04 is then written, requesting the ONFI memory controller to execute the BLOCK ERASE instruction. The selected block is now erased and is ready to be (re)programmed.
4. **Set (Page) Address**, the pages in the block are programmed sequentially from the first to last page (page 0 to 2303). Similar to the previous address command, the selected location is parsed into 6 bytes and passed on to the memory controller through the UART.
5. **Determine if page should be programmed High or Low**, the memory contains SLC, MLC and TLC cells. The TLC cell contains a Lower, Middle and Upper page. Each page should be programmed in the right binary state to reach the highest logic level (*L7*). The table determines for each page number if it should be programmed high or low.

6. **Program page High or Low**, the ONFI memory controller has two instructions in its instruction set that will automatically program full page with 0 or 1's. First, command AAh is sent to the UART controller, after which 32 (program page high) or 33 (program page low) is issued. This process is repeated for the 2304 memory pages present in a memory block.
7. **Close serial communication**, finally, the pySerial communication channel is closed, preparing it for the next sequence of instructions to be sent to the FPGA.

Determining the threshold voltage

The threshold voltage of the memory cells will be determined by rereading the same page over again, each time increasing or decreasing the reference level using the READ OFFSET function. The flowchart shown in Fig. 4.41 indicates the different steps executed by the Python script.

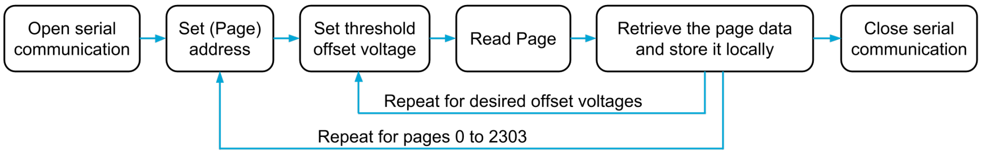

Open serial communication
Set (Page) address
Set threshold offset voltage
Read Page
Retrieve the page data and store it locally
Close serial communication
Repeat for desired offset voltages
Repeat for pages 0 to 2303

Figure 4.41: Flowchart of the steps executed by a Python script to measure the threshold voltages of all the memory cells in a memory block.

1. **Open Serial Communication**, the pySerial communication channel is opened on the desired COM port with a selected Baud rate.
2. **Set (Page) address**, 6 address bytes are passed to a buffer of the memory controller, on which the READ PAGE operation will be executed. The desired column, page, block and LUN are formatted to the requirements of the memory as specified in Section 4.2. CCh is written to the FPGA, indicating the UART controller that 6 address bytes will follow. Then the 6 address bytes are sent. Throughout the loop, the block and LUN address will stay identical, while the page number will increment with one page each full cycle.
3. **Set threshold offset voltage**, the command 33h is passed to the controller after which the four bytes defining the offset voltage are written to the controller. Then the *Set Read Offset* instruction is executed by the controller (command 31).
4. **Read Page**, with the offset set, the page will be read (READ PAGE, command 02) and stored on the FPGA.
5. **Retrieve the page data and store it locally**, the page data stored on a buffer in the FPGA is downloaded over the UART communication and cached in a Python array, after which it is stored on the computer. The instruction MI_GET_NAND_PAGE (instruction 36) is sent, after which the data is transferred. The pySerial connection waits until 18592 bytes are received, which are then stored for later data processing.
6. **Close serial communication**, after all the pages are scanned with the desired offset voltages, the pySerial communication channel is closed, preparing it for the next sequence of instructions to be sent to the FPGA.

Chapter 5

System integration, testing and performance

This chapter provides the framework to correctly interface the different software and hardware components. Every part has to be configured and set up correctly before instructions can be executed on the memory, possible to enable enable voltage threshold measurements. A section is dedicated to the different trouble shooting methods that were used. Efficient and targeted error solving is key to develop a system with many different parts. Finally it is shown how well the system performs in terms of speed and data storage.

5.1 Hardware Integration

The hardware side of the test setup features the following components, these need to be set up and configured correctly.

1. ZedBaord FPGA running the controller code
2. Custom PCB, providing the interface between the 3D NAND memory and the ZedBoard
3. The DUT: NAND memory with 132-BGA package, which needs to be correctly placed into the socket
4. Saleae Logic Analyzer for trouble shooting purposes.

5.1.1 DUT: 3D NAND memory

The BGA-132 package is placed in the socket by applying pressure on the outer spring loaded bracket. Two arms located in the center will be lifted, after which the memory device is placed as indicated in Fig. 5.1. The blue circle indicates the white mark on the memory, which should be located in the upper left corner of the bracket, towards the edge of the PCB where the test pints placed. The socket also fits the wider BGA-152 packages and it therefore required to manually center the memory chip. Proper placement with good connections can be confirmed by looking at the response of the R/B# signal or by requesting a READ PARAMETER PAGE. This instruction should return a set of data on the data bus, indicating proper connection.

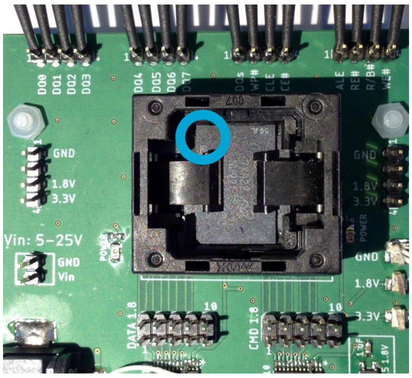


Figure 5.1: Indication of the white marker on the memory for correct placement in the socket.

5.1.2 PCB-FPGA Pmod ribbon cables

Two flat, ten core ribbon cables connect the JA1 and JB1 Pmod ports on the Zedboard to the header pins on the PCB, indicated with *DATA 3.3* and *CMD 3.3*. The Pmod ports pass signals as 0 or 3.3V, hence the connectors are mated with the 3.3V pins, before the logic level shifters, which converts the signals to 1.8V, accepted by the memory.

The Pmod ports are female header pins, so are the used ribbon cables. Therefore a male-male 2x5 header pin is inserted to allow for mating. Note that the Pmod port also has two 3.3V pins (the pins closest to the right side of the board in Fig. 5.2), these are not used, as power is provided to the interface PCB by an external power source.

The ribbon cables should be attached as shown in Fig. 5.2 to make the correct connections.

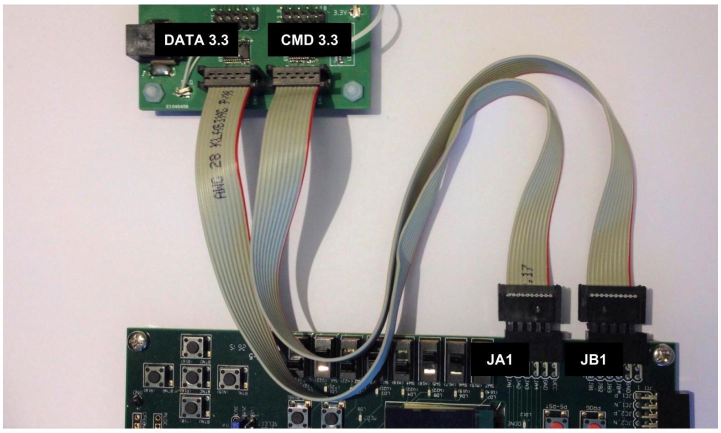


Figure 5.2: Two ribbon cables connect the FPGA with the PCB. The JA1 port connects with DATA 3.3 and JB1 with CMD 3.3.

ZedBoard

The ZedBoard is powered through a barrel plug with a cable to a wall adapter. The JTAG programmer cable should be connected, such that the controller software can be uploaded via a bitstream through the Vivado software package. A blue light will indicate a successful upload.

Pmod USBUART extension board

The Digilent extension board in the upper row of female header slots of the JC1 Pmod (as indicated in Fig. 4.17), where the Rx and Tx signals of the UART communication are connected to Pmod port JC1_N and JC2_P. A USB cable with a micro USB connector connects the USBUART board to the main computer. Through a UART communication channel with the ONFI memory controller on the ZedBoard FPGA, the necessary sequence of commands will be sent from a Python script, as described in Section 4.4.4.

Power Supply to the PCB

Power is supplied to the PCB either through the 2mm female barrel jack or to the male header pins. This can be done by using a power supply or a wall adapter. The voltage source shall be between 5 and 24V. Two red LEDs will light up when power is applied.

Logic Analyzer

The PCB has 16 test pins through which the signals sent and received from the memory can be probed. The 16 channel Saleae logic analyzer can probe each of these pins. An example for this colorful spaghetti of probe wires is shown in Fig. 5.3. The communication signals from the USBUART extension board can also be checked by attaching the probes to the JC2_P and JC1_N Pmod pins. The logic analyzer is powered through USB, and required USB3.0 for high speed data acquisition.

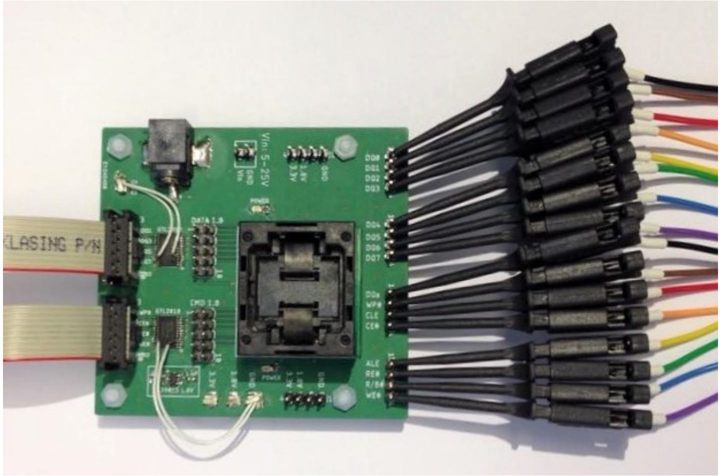


Figure 5.3: All 16 data and command signals to communicate with the memory can be probed on the PCB. The image shows how the probes of the logic analyzer are attached to the test pins on the right side of the board.

5.2 System Debugging

This section of the report is dedicated to the debugging methods used with which the final working test setup was reached that is presented in this thesis. Debugging relied mainly on (VHDL) software simulations and signal measurements with the logic analyzer.

5.2.1 VHDL Simulation with ModelSim

The first line of testing will happen before compiling the code and uploading the software onto the hardware platform. VHDL code can be synthesized and simulated using Mentor Graphics' ModelSim, available for download online.

A so called testbench is created in which the to be tested VHDL program is loaded, one needs to add for example a virtual clock module (normally supplied by the FPGA hardware) or a set of instructions for which the response of the VHDL program is under scrutiny. Small parts of the code can be tested, or an entire program as a whole. As an example for this project, a testbench was created to simulate the UART communication protocol. The testbench emulates the commands sent from the Python scripts to the UART and memory controller. As such, it is possible to simulate the full system, without the memory, in ModelSim.

As an example, the testbench file commanded the BLOCK ERASE instruction. The output of the simulation is shown in Fig. 5.4, where the controller *state*, *next state*, *substate* can be seen and the generated control signals (CLE, ALE, WE# and RE#). It can be noted that such a simulation can be more powerful than probing the generated signals when implemented on hardware, because the internal state of the program is also shown.

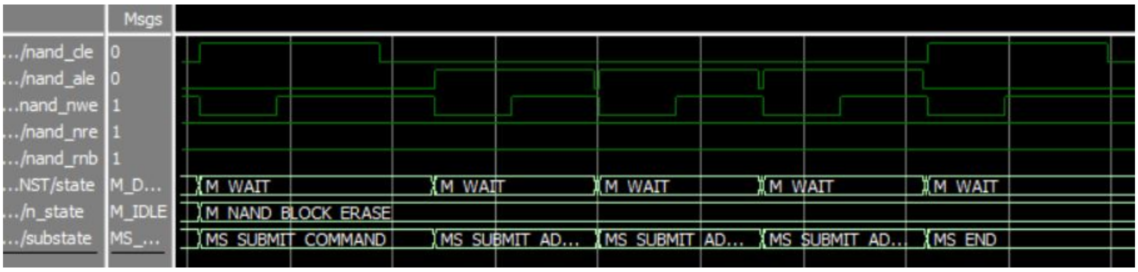


Figure 5.4: Executing of the BLOCK ERASE instruction by the controller, simulated in ModelSim.

One obvious drawback is that it is not always possible to fully recreate a digital representation of a controller program when expecting a response from an external device. The original VHDL ONFI controller code has certain states in the Finite State Machine that are triggered by the response of the Read/Busy (R/B#) signal. This must be taken into account by for example forcing this signal into a certain state, or triggering it such that the controller can continue its simulated operations. It must be noted that Micron provides Verilog files for their (3D NAND) memories, which can possible be simulated while interfacing with controller software. This was however not attempted in this project.

A series of commands can be automated in ModelSim with a *.do* file, when called from the command line. It can reduce the repeated effort of configuring a set of simulations.

5.2.2 Probing the generated signals with a Logic Analyzer

A logic analyzer can serve as a quick reference check whether the controller or memory are acting as expected. It can probe the individual commands and signals in detail, to look for any discrepancies. The used hardware is a Saleae Logic Pro 16, which has 16 digital/analog channels, and is connected to the computer using USB 3.0. The logic analyzer comes with its custom software (Logic 1.2.18), where the gathered data is visualized with the option of saving it into different formats for further processing.

As an example user case is to check whether the memory is accepting the instructions sent from the controller. The R/B# signal should be pulled low while the memory is completing an instruction. A second example is to investigate whether the memory outputs data on the data bus while the controller toggles the RE# signal.

An example of a probed instruction is shown in Fig. 5.5, the sent 0x02 command instructs the memory controller to read a page of the memory (READ PAGE 00h-30h). The signal indicates how first a command (00h on the data bus, not shown) is latched by enabling the Command Latch Enable (CLE, red) and Write Enable (WE#, purple), after that, 5 address cycles are latched with the Address Latch Enable (ALE, yellow) and Write Enable (WE#, purple), after which the second command (30h) is latched. It can be seen that the full instruction is accepted by the memory, as the Read/Busy signal (R/B#, blue) is pulled low by the memory.

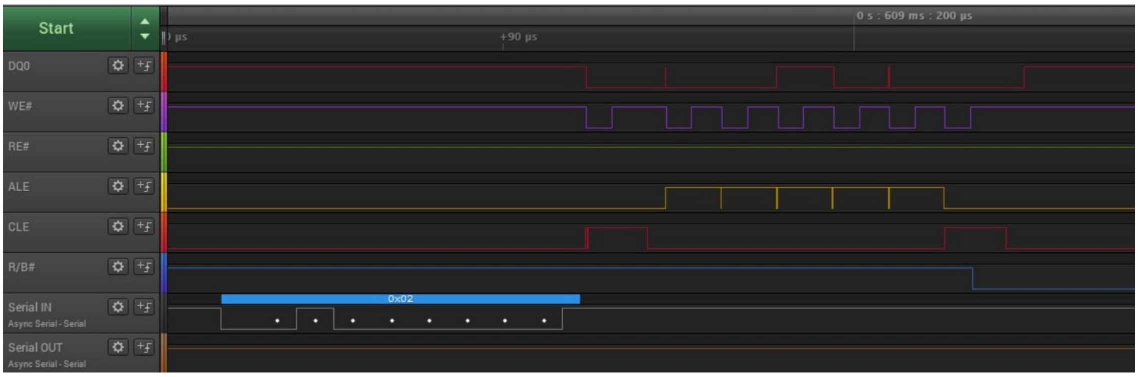


Figure 5.5: Example of a probed READ PAGE instruction. It can be seen how the controller first receives the command over the UART communication channel (0x02), proceeded by generating the signals to latch the starting command, the address, and a second command.

Overall trends can also be observed, in Fig 5.6, a full block is being programmed to the highest logic level, where the right pages need to be programmed to either 0 or 1. It can be seen that the write operations are successful, by the R/B# signal being toggled after each programmed page. The controller is shown to alternate between two pages being programmed low and one page being programmed high (indicated by the DQ0 data signal), which is in accordance to programmed a page to logic level 7, with logic 001.

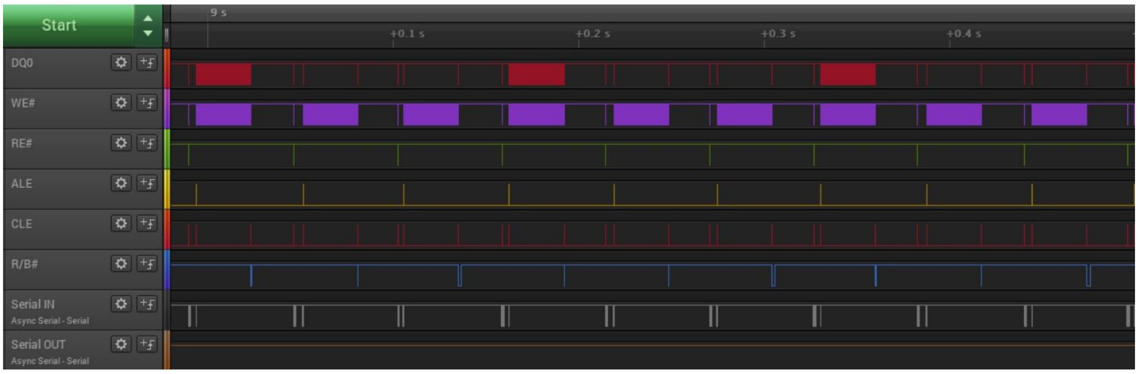


Figure 5.6: Programming blocks to 0 or 1. It is a quick visual reference check whether the programming is working accordingly by looking at the DQ signal (shown here is only DQ0) and the R/B# signal.

The UART communication signals that are being received and sent by the ZedBoard can also be probed, as already shown in the example above. Two probes of the logic analyzer are attached to the x and Tx pins of the JC1 Pmod port (JC2_P and JC1_N). The UART protocol can be decoded through the by the logic analyzer software into binary, ASCII or HEX, to assist the user.

Aside from digital signals, the analyzer can process analog signals and act as a digital oscilloscope. It can be used to check for overshoot/undershoot, which could cause signals to get latched prematurely or twice. It can also serve as a quick check whether the correct voltages (1.8V) are generated by the level shifter.

5.2.3 Xilinx Vivado Integrated Logic Analyzer

Another debugging tool is the Integrated Logic Analyzer in the Xilinx Vivado toolbox. When connecting the Zedboard with the computer through the JTAG programmer cable, it can show the internal states of the VHDL controllers running on the FPGA. It requires the necessary setup within the VHDL code, to make the data available to the logic analyzer. With a proper setup, it can be a useful tool to check the internal state transitions and where error appear. Due to the extended setup required, it was not fully implemented and used as a tool in this project.

5.2.4 ZedBoard LEDs

8 LEDs (LD0-LD7) are available to the programmable logic (FPGA) of the ZedBoard. These can be connected to certain signals within the controller, and for example be turned on/off based on a certain FSM state or status. The LEDs can serve as a quick visual reference check.

5.3 System Performance

A couple of data points can be given as indication of how the system performs.

5.3.1 Read cycle timing

The measurements as a radiation monitor will be based on the threshold voltage of each individual memory cell, which is measured by reading a cell multiple times, while looking for a change in its logic level. This is a repetitive process in which the same operation is repeated for a number of cycles per page, and then for a large number of pages within a memory block (nominally, 744 pages per block).

It is shown here that a single page read cycle consists of three parts: retrieving the page data from the memory and store it locally on the FPGA, transfer the data from the buffer on the FPGA to the computer over the UART communication and finally save it locally for later data processing. An example of such a full cycle is shown in Fig. 5.7, visualized by the logic analyzer.

The figure indicates that the actual reading of the page (in SDR mode, see Section 4.2.1) is only a fraction of the total time it takes to complete one cycle. The data is placed on the 8-bit data bus (only DQ0 in red is shown here) each time the Read Enable (RE#, green) signal is toggled. After completion, the page data is sent from the buffer in the FGPA to the computer, seen by the data on the Serial OUT line (orange block) with a baud rate of 921600 bit/s. The remaining time before a next cycle is initiated is spent by the Python script storing the data. The total time for one cycle is measured at 712 ms, where the first 35 ms are spent reading the page, 207 ms are needed to send the data and 470 ms are needed to store the data.

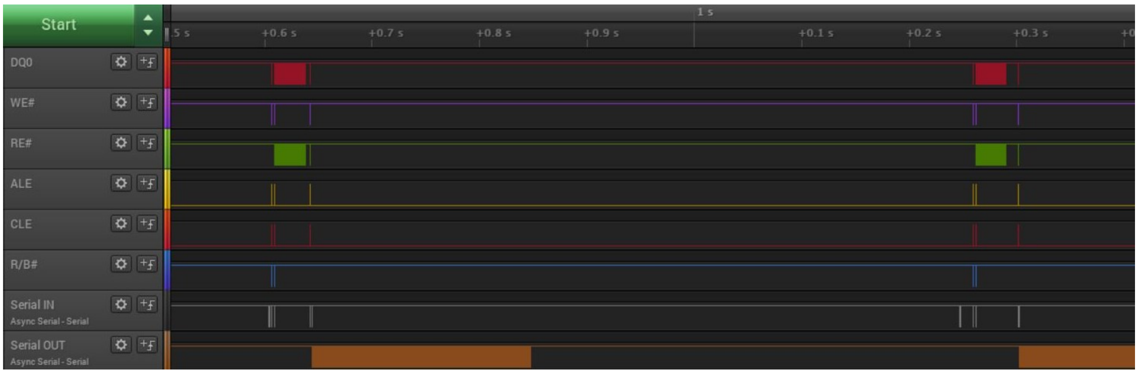


Figure 5.7: Overview of one cycle to read, transfer and store one page with the logic analyzer.

As a tentative calculation, where the 744 pages are each read 64 times (30 mV increments, see Section 2.4.3), a single block read operation would take 33,902 seconds or 9,4 hours. This is rather undesired. However, the two bottlenecks: data transfer speed and storage time, are identified and included in the recommendations for future work.

5.3.2 UART Communication

It has been shown that it takes 207 ms or 22% of the total cycle time to transfer the page data from the FPGA. This is based on a baud rate of 921600 bit/s, this was made possible by providing a buffer to pySerial to accommodate the incoming data. It is indicated by the datasheet of the Digilent USBUART extension module that the FTDI chip should be capable of handling speeds up to 3 MBaud.

5.3.3 Data Storage

A single page holds 18,852 bytes of data. Assuming the identical parameters from earlier (744 pages, 64 read cycles per page (30 mV increments)), 904 MB is required for a single block scan. As the state of a single memory block will be determined before and after irradiation exposure to determine the voltage threshold shift, a total of 1.8 GB is needed.

Chapter 6

Memory Test Data

The full system has been proven to work. Using a Python script, the ONFI memory controller can be given commands to execute on the memory. The possible results of those commands can be stored onto the main computer and be processed. This chapter will give an overview of the initial set of characterization measurements taken and the analysis of those measurements. It is proven that the methodology presented in Section 2.4 can be used to determine the threshold voltage of individual memory cells. The data reveals some unexpected behavior which will need further investigation.

6.1 Memory Characterization

Three B17A Micron memory packages were acquired by unsoldering them from a commercially purchased SSD. These rawNAND devices are placed in the BGA socket on the PCB to be read through commands sent from the Python scripts.

Figure 6.1: The three Micron memory packages are placed in separate protecting containers.

6.1.1 Memory Device Model

Three 3D NAND memories were obtained by unsoldering them from an SSD. Two chips had identical markings, while a third one was marked differently. By issuing the READ PARAMETER PAGE instruction, the memory returns the 256 byte long JEDEC parameter page which holds general and device specific information. Table 20 in the Micron data sheet gives a full overview of how the data should be interpreted [16]. Bytes 44-63 indicate the actual device model.

The instruction was issued for the three chips returning the following data:

CHIP 1	MT29F2T08EMHAFJ4
CHIP 2	MT29F2T08EMHAFJ4
CHIP 3	MT29F512G08EBHAFJ4

The first two chips store each 2 Tb of data, the third chip 512 Gb. The theoretical maximum storage capacity is 564 GB. The SSD from which the memories were extracted has an advertised storage capacity of 480 GB.

6.2 Overall analysis

6.2.1 Logic Level Order

A single memory cell in TLC mode stores 3 bits of data, which are being encoded in the threshold voltage by 8 separate logic levels. The mapping between the logic level and the binary data of the logic levels is important to determine in order to later know the page (lower, middle or upper) where the bit flip due to a particle event occurs.

By trial and error, the cell logic of the highest logic level, *L7* is found. An arbitrary state is programmed and the *rL7* offset voltage was increased until a bit flip was seen. This data gives the cell logic for *L7* and *L6*. Next, the *rL6* can be modified until the cell logic of *L5* is found, repeating the process for the remaining logic levels. The results visualized in Fig. 6.2.

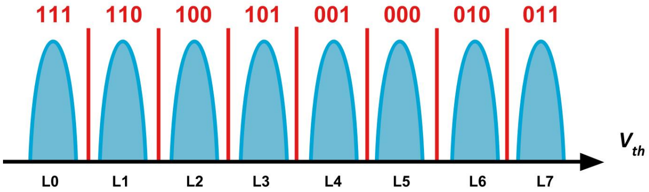


Figure 6.2: The logic levels with their discovered stored binary data.

As it is desired to program the memory to the highest threshold voltage (*L7*) to get the largest sensitivity, the lower page shall be programmed to 0, while the middle and upper page shall be programmed to 1. The upper page shall be checked for a bit flip, where the programmed state will change from *L7* to *L6*.

6.2.2 Threshold Level Shift: Overall Behavior (Full Page level)

The presented method in Section 2.4 is applied to the memory to gain insight in the actual programmed cell threshold voltages. A visual representation combined with real measurements is shown in Fig. 6.3. When increasing the threshold shift offset of *rL7* (shown on the x-axis), the binary data stored in the upper page will be shifted from a binary 1 state to 0, as the data stored in *L7* will be registered as L6. While increasing the threshold voltage offset, the logic state will gradually shift from fully high to low, as is shown in the figure. When the offset is 0 mV, (almost) 100% of the cells within that page (148,736 cells) will be programmed to state 011, gradually increasing the offset voltage, the number of pages programmed to *L7* (011) lower and more cells are read as *L6* (010).

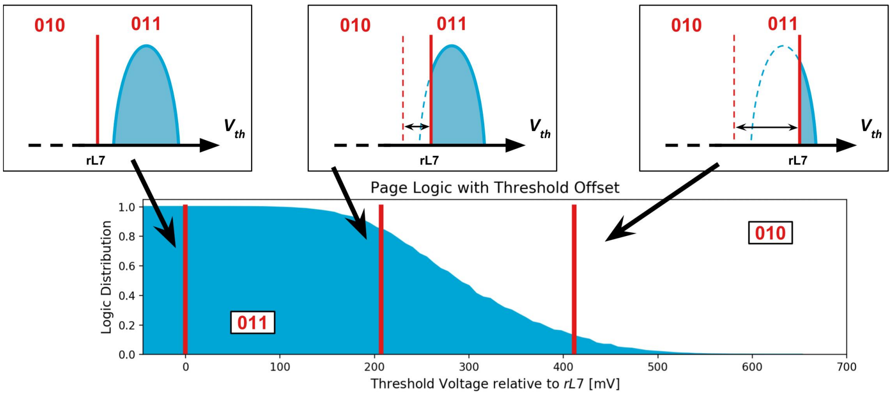


Figure 6.3: The distribution of the cells programmed in *L7* (011) or *L6* (010), when increasing the *rL7* threshold voltage.

According to literature, the programmed states is supposed to have a Gaussian distribution [15]. By taking the derivative of the previous distribution curve (where the offset voltage was increased with 7.5 mV increments, the smallest possible), the voltage distribution of the cells programmed in *L7* can be found, this is presented in Fig. 6.4. The peak of the real data is slightly skewed to the lower end. It seems however to by a rather good approximation.

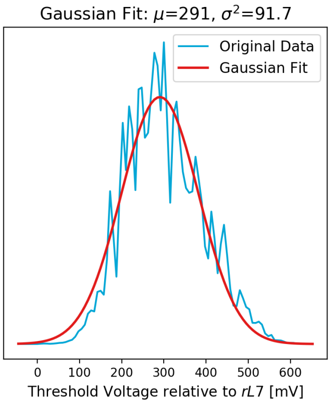


Figure 6.4: An attempt to fit the original voltage distribution data with a Gaussian curve.

The voltage threshold cell distribution is now known in function of the (relative) threshold voltage. As shown in Fig. 6.4, the distribution curve is about 500 mV wide. The one paper which published on the radiation effects in 3D NAND memory, showed voltage threshold measurements without presenting the actual voltages [1]. The DUT in question was the 32L Micron 3D NAND memory, the previous generation of the DUT with which the presented data is taken. Assuming similar characteristics, the plots from the paper could be analyzed and an estimation of the voltage threshold shifts could be made.

6.2.3 Distribution analysis

Every read cycles requires time, storage area and processing power. It might be of interest at some point to make a trade-off between the number of offset steps (engineering requirements) and the amount of cells being taken as active detector nodes (science requirements). Based on the data shown in the distribution curve of Fig. 6.3, one can determine the upper and lower bounds of the offset threshold voltage to capture 1σ, 2σ or 3σ of the data points.

UPPER BOUND

std	percentage	# cells
1 σ	84.1%	125087
2 σ	97.7%	145315
3 σ	99.9%	148587

LOWER BOUND

std	percentage	# cells
1 σ	84.1%	23649
2 σ	97.7%	3421
3 σ	99.9%	149

The corresponding values are visualized in the distribution plot, shown in Fig. 6.5. A slight skew towards the left side can be seen, which is in agreement with the near Gaussian distribution found in Fig. 6.4.

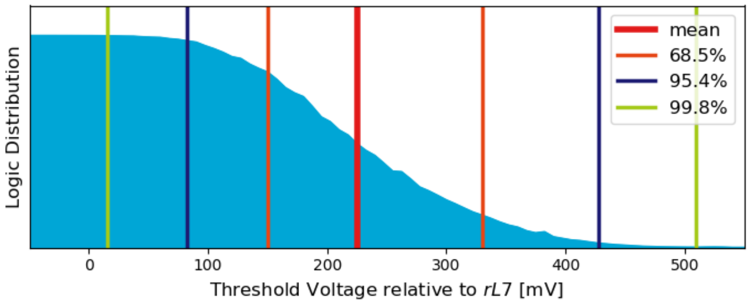


Figure 6.5: Visualization of the lower and upper limits for 1,2 and 3 σ distributions.

To determine the threshold voltage offset values corresponding to the lower and upper boundaries, a total of 20 pages were sampled, for which the average and standard deviation values were determined. The results are tabulated below. In addition to the data in the table, mean offset voltage is 210.4 mV with standard deviation of 7.67 mV.

UPPER BOUND

std	offset voltage [mV]	std [mV]
1 σ	312.2	7.60
2 σ	406.9	8.18
3 σ	490.9	8.03

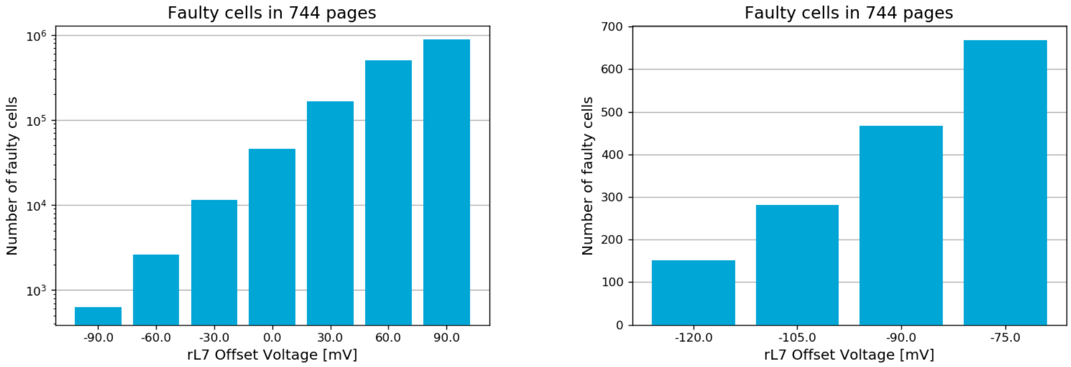
LOWER BOUND

std	offset voltage [mV]	std [mV]
1 σ	131.2	6.05
2 σ	65.6	7.88
3 σ	-4.1	6.48

6.2.4 Cells correctly programmed to *L7*

Cells programmed to *L7*, are also expected to be read at *L7*. Due to the limitations of the programming circuits, not all the cells are read as *L7*, but a small amount of cells are registered as *L6*. A full memory block was programmed was programmed to the highest logic level, after which it was checked how many fault cells are registered, while decreasing the offset level of *rL7*. The results are shown in Fig. 6.6.

744 TLC pages are read (110,659,584 memory cells), where an initial rough comparison is made (left figure, note the log scale) and a more detailed analysis for the lower offset voltages (right figure). At the original *rL7* level with a 0.0 mV threshold shift, about 40,000 faulty cells are programmed, which is equivalent to about 0.037%. With an offset voltage of -90 mV, this number has reduced to 480 cells, about 0.00046 %.



(a)

(b)

Figure 6.6: Investigation into the number of faulty cells programmed, when the memory is programmed to the *L7* logic level. In the original memory state (no offset voltage), about 0.037% of the cells were programmed to *L6* instead of *L7*.

The originally faulty programmed cells are normally handled by an error correcting code (ECC) in the memory controller. The memory as well can be calibrated, where an initial offset is given to the different reference levels, which might offset intrinsic anomalies which are device specific. The results shown here indicate that it might be needed to accept a certain number of memory cells that will have to be discarded as sensitive nodes for a detector. The trade off must be made where a larger number of scanning cycles are done to capture each and every cell, whilst the added benefit diminishes.

6.3 Threshold level shift: Individual cell analysis

The analysis thus far has focused on the overall page or block. This section will investigate the threhsold voltage measurements of the individual memory cells. A *flickering* phenomena is seen, where a memory cell displays a state of uncertainty between two logic levels.

6.3.1 Flickering behavior, a description

The smallest offset voltage with which an reference level can be increased or decreased is 7.5 mV, it is expected that within a single increase, the binary behavior of a logic state change should be observed. However, it is noticed that not all the cells change immediately from one logic state to another. A considerable part of the cells within a page show "flickering" behavior between the two states. Fig. 6.7 shows the transitions between the logic states for a set of individual memory cells. An uncertainty in state, the flickering behavior, is indicated by the red ellipse, while the cells displaying an immediate transition are desired (indicated with the green ellipse) .

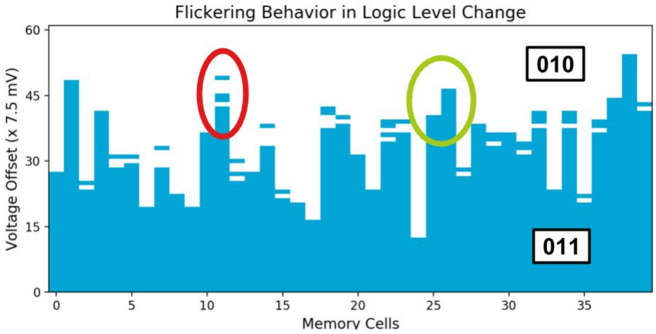


Figure 6.7: The transition between two logic states when increasing the offset voltage is visualized for individual memory cells. Several cells show flickering behavior, a state in which the cell transitions between two logic levels.

When the transition between two logic states is not instant, it will be the source of an added error in the measurement of the programmed threshold voltage of each cell. The flickering behavior will be further investigated.

A first attempt is made to characterize the flickering behavior. One "flicker" could be described by a *Mean* and *Width* value, as visualized in Fig. 6.8. The Mean value is an estimation of where the logic change would occur, and is assumed to be located halfway of the flickering period. The Width is based on the offset voltages where the first and last transition occurs, before the final stable new logic state is reached.

It can be noted that this is a rather basic description of the found behavior. More analysis may show that far outliers could be discarded, or that a weight could be given to certain transitions, based on how frequently a change occurs. An algorithm could be deployed in future work.

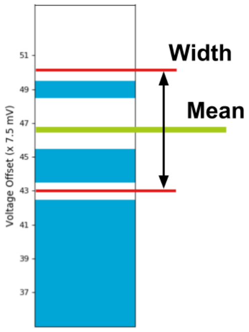


Figure 6.8: Characterization of the flickering behavior with a value for its Mean and Width.

6.3.2 Analysis of the flickering behavior

We attempt to quantify the flickering behavior seen in a memory page. The histogram in Fig. 6.9 indicates the widths of the flickering period for a single page in the memory, i.e. how unstable is the transition from one logic state to the other. A considerable amount of the cells (66.2%) displays an immediate transition, these cells will have a low error in their initial measurement. No cells can have a width of 7.5 mV, as at least two logic state transitions are needed to reach a new final state. The frequency of cells with an increasing width gradually decreases.

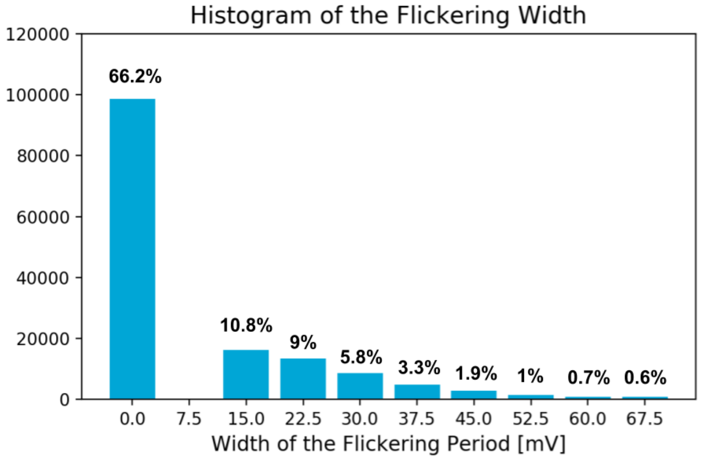


Figure 6.9: Histogram indicating the width of the flickering period before a stable new logic state is reached.

Without a method of measuring the threshold voltage of each cell with a higher accuracy (lower flickering width), the error could be accepted. When using the memory as a detector, each cell measurement will have a different measurement error, based on the width of the flickering behavior.

The scatter plot in Fig. 6.10 investigates a possible correlation between the mean programmed voltage and the width of that cell. No real relation is found.

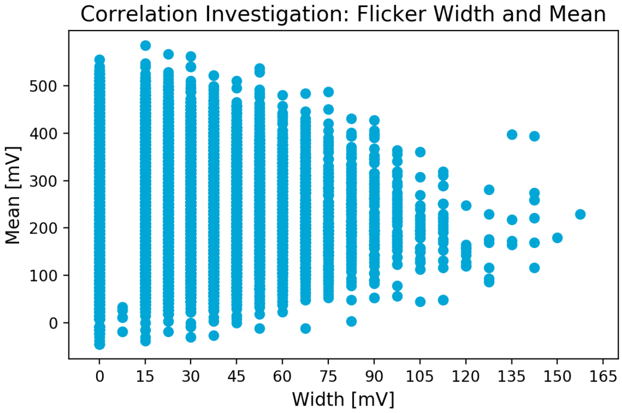


Figure 6.10: Scatter plot of the width and Mean values of the cell flickering.

6.3.3 Threshold Voltage between different read cycles

A baseline has been established where a certain amount of cells can be read with an immediate transition, and about 30% of the cells display some transition period in which it juggles between two logic states. However, the question was asked if that exact same voltage would be read again during a second read cycle. A page was programmed once, and then read out multiple times.

A visual representation of 6 cells with 20 and 60 read cycles are shown in Fig. 6.11 and 6.12. The logic state is shown in function of the read cycle and the threshold voltage offset. The red line indicates the mean threshold voltage determined of that cell for each read cycle, based on the estimation as presented in Section 6.3.1. The 20 reread cycles in the first figure were done over the course of approximately half an hour, the 60 read cycles over the course of 5 hours.

The two figures indicate the presence of rather different behavior. In the top graph, the measured mean offset voltage at which the bit flip occurs for each cycle varies about plus or minus 20 mV around a mean value, the general trend is constant. The bottom figure indicates a drifting behavior, where the measured threshold voltage gradually decreases while the cell is being reread. A spike is seen after the 50th read cycle, where the threshold is restored to a higher voltage.

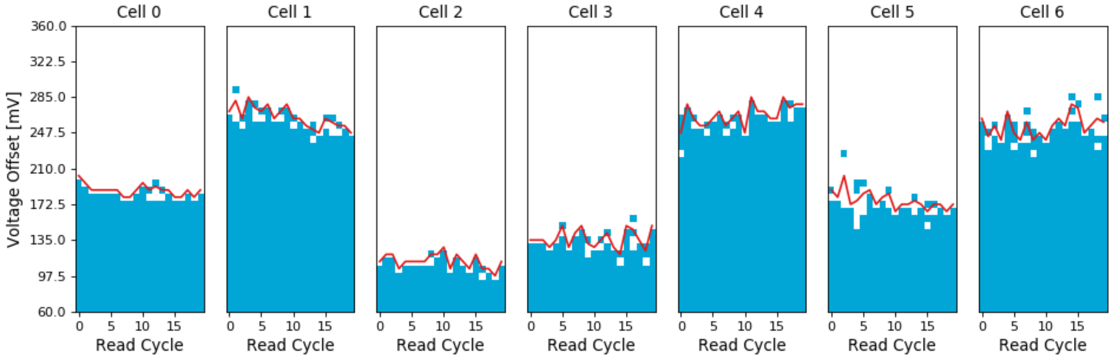


Figure 6.11: Logic state of multiple cells over multiple read cycles.

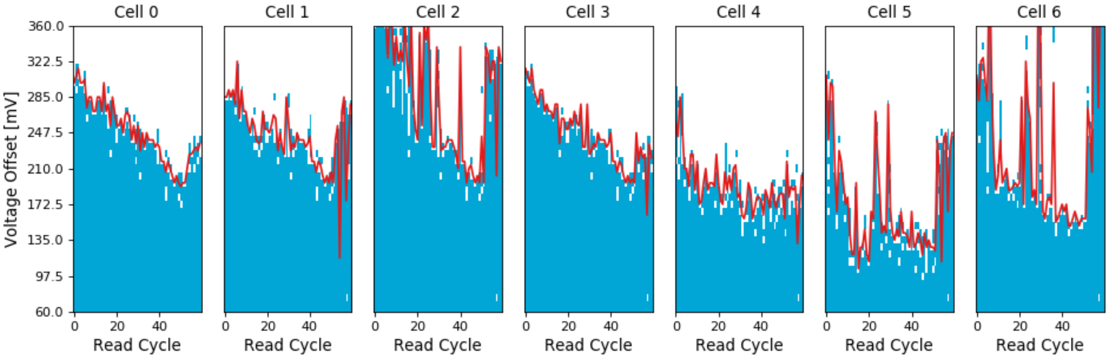


Figure 6.12: Logic state of multiple cells over multiple read cycles.

This behavior may question the validity of the threshold voltage measured during the first read cycle. Is this data point a representative measure of the stored charge? Further questions are raised towards the reason of the downwards drifting behavior, and the spike seen after a large number of read cycles. Could this be attributed to an internal calibration done by the memory or due to environmental effects such as temperature? A further investigation and characterization will be needed to better understand the behavior. If no drift is seen, the initial error bars of the measurement based on the 7.5 mV increment could be increased to levels equivalent measured during a large number of read cycles. An example of how the behavior could affect the measurement frequency and the error is discussed in Section 2.4.4. If the measurements gradually change during a large number of read cycles, could this be caused by the underlying physical measurement method, during which each cycle some charge is lost from the cell?

Chapter 7

Conclusions and recommendations

This final chapter presents an overview of the conclusions and recommendations that are found in this thesis. Section 7.1 summarizes the conclusions of the different chapters and reflects on the research questions posed in the first chapter. The recommendations are listed in Section 7.2 and suggest various actions and paths to be taken for future development and research.

7.1 Conclusions

The thesis project started off with the proposition of using 3D NAND technology to monitor space radiation. Literature showed promising data, but no methodology was presented to repeat any measurements to further investigate the topic. In order to advance towards the goal of a 3D NAND memory radiation monitor, the following research questions were proposed:

1. **What methodology could be used to make threshold voltage measurements in a commercially available 3D NAND memory?**
2. **Can a sensitive volume within the memory be defined and quantified? Which and at what frequency would particle events take place in space?**
3. **What hardware and software is necessary to make such measurements?**

An answer was provided to each of these questions. The thesis was finalized with threshold voltage measurements from an actual 3D NAND memory, using a custom built test setup. The measurements reveal a first level of capabilities and challenges when used for radiation monitoring. The test setup provides the in house capability to advance with the study of the radiation effects on NAND memories. This will enable further development and research.

7.1.1 Methodology to measure the threshold voltage of a memory cell

The detection of a particle in a flash memory is based on a change in threshold voltage. A methodology is proposed with which the threshold voltage of individual memory cells can be determined.

The method leverages the traditional way in which the memory translates the threshold voltage of a cell into a binary sequence of data, used for data storage. By changing the boundary separating two logic states, a bit flip can be introduced. This boundary, referred to as *reference level* in this work can be given an offset voltage using the READ OFFSET function. A visualization of this method is given in Fig. 7.1.

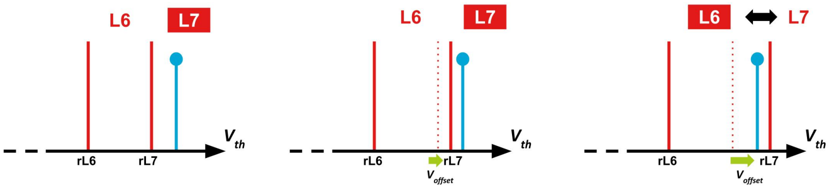


Figure 7.1: A memory cell is originally programmed to *L7*. The blue marker indicates the threshold voltage of this cell, which is desired to measure. By increasing the threshold voltage of reference level *rL7*, a bit flip will be introduced.

The practical implementation is based on rereading the same cell with each time modifying the voltage offset. A data analysis process will identify the bit flip for each cell and report at which offset voltage this took place. This will be a relative measurement. It is determined that a voltage threshold shift measurement could be made with a minimum theoretical error of 5.3 mV.

7.1.2 Sensitive volumes and SPENVIS simulations

Memory cells are addressed by column, page and block. An investigation estimates the corresponding physical sizes, based on die and SEM images of the device and literature on 3D NAND architectures. The individual cell size is concluded to measure 198 nm in diameter, with a height of 52.6 nm. The physical architecture of a page is based on a staggered pillar and bitline configuration: a single page is four rows wide, effectively measuring 7.36 mm in length. A memory block measures 7360 μm (depth) x 3.37 μm (height) x 9.5 μm (width).

A SPENVIS radiation simulation was ran with the detector based on the measurements of a single memory block. The final results of the simulation indicate how many particle detections are possible, in function of different Earth orbits and the minimal particle energy needed for a detection to happen. At this point it is unclear what the minimum LET(Si) is that will be able to be measured. Galactic cosmic ray particles are likely to be encountered less than once in ten days. If protons can be registered, the fluence will be dependent on the orbit and may range from 10^4 to 10^9 events per day (ISS and MEO).

7.1.3 Test setup development

A test setup was developed with which the proposed measuring methodology was validated. A user is able to execute memory instructions such as writing and reading data to/from a specified memory address. Custom commands can be added for future extensions.

The requirements for the test setup development were based on the ONFI specification. The test setup features a ZedBoard FPGA to run a memory controller, a custom designed PCB with a socket in which the memory is placed and a computer from which the instructions are requested. Python scripts send instructions to the FPGA over a UART communication protocol. The memory controller on the FPGA is based on an open source ONFI controller, generating the control and data signals according to the ONFI standard.

7.1.4 Threshold voltage measurements

Threshold voltage measurements were taken using the latest Micron 64-layer 3D NAND memory technology. The chips were acquired through purchasing a commercially available SSD and unsoldering the packages.

Results were taken and analyzed on block, page and individual memory cell level. The distribution of the cell threshold voltages for a full page showed the expected Gaussian distribution. Interesting observations were made when investigating the measurements for single cells. It was found that the cells do not always transition immediately from one logic state to another, a flickering behavior is seen. Additionally, a different threshold voltage is measured when the same memory cells is read a second or a third time.

The availability of actual data allows for a better interpretation of the results published in the one paper on radiation effects in 3D NAND memory. Voltage threshold shift data due to irradiation was only visually represented, not quantified. The phenomena of flickering and reading different voltages between cycles need to be further studied to be better understood as they will introduce a measurement error.

7.2 Recommendations

This final section of the report lists a variety of recommendations for the continuation of this project. A basic proof of concept was shown to be working, but many questions remain and different solutions are encouraged to be developed.

7.2.1 Internal hardware layout

The relationship between the addressed memory cells and the package should be further investigated.

The first step could be the investigation of the dies within the memory package. Are multiple dies stacked within the molded plastic? How are the pages and block oriented in comparison to the package? Depending on this orientation, particles will cross a single or multiple memory blocks during testing with a specific angle of irradiation. It is proposed to use nitric acid to remove the packaging. This could be done at the faculty of aerospace engineering, the right person to contact for this would be Johan Bijleveld (J.C.Bijleveld@TUDelft.nl).

Secondly, it is recommended gain further insight in the relative or exact physical memory cell locations in the 3D architecture. The ideal outcome would be a mapping between the virtual memory addresses and the physical location of each cell within the die. Through contact with Jim Handy (`objective-analysis.com`, `thessdguy.com` and `thememoryguy.com`), the following persons were advised to approach: Venkat Vasudevan from Intel, Stephen Lazuardi at Micron and Niel Mielke at Intel (former cell reliability fellow, retired). TechInsights has written a report on the Micron B16A memory, likely to contain a detailed breakdown and analysis on the found architecture[1]. Contact was made with Mike Sharp (msharp@techinsights.com), but the cost for the report was quoted at US$10.000. A discount when used for academic purposes was on the table. It might be possible to get in touch again when radiation test data is available, to start a possible data exchange benefiting both parties.

7.2.2 Environmental effects on the voltage threshold measurement

It was shown in Section 6.3.3 that rereading a memory cell resulted in different threshold voltage measurements. It will be interesting to study the effects of certain environmental effects on the behavior of the memory. A first obvious investigation would be to quantify a possible relationship between the measured threshold voltage and the temperature of the memory cell. Tests could be conducted in the vacuum oven available in the clean room on the 8th floor at the faculty of aerospace engineering, which could be heated to about 200°C. The internal chip temperature can be measured using Feature Address E7h (Temperature Sensor Readout). As an additional note on the temperature effects, the data sheet indicates an operation temperature between 0 and 70 °C for commercial use. It would be interested to check the lower and upper limits, which would result in a payload requirements for a satellite mission.

7.2.3 System architecture

The current system architecture is based on the use of an FPGA, which generates the correct signals to control the memory. It also transfers measurements from the memory on to an external computer for data processing. The current configuration with UART communication has proven to be quite slow. When using the

[1] Intel/Micron 29F01T2ANCTH2 64-Layer 3D NAND Flash Memory Circuit Analysis - http://www.techinsights.com/reports-and-subscriptions/open-market-reports/Report-Profile/?ReportKey=CAR-1804-903

memory as an actual sensor, the readout should be able to happen faster. It is recommended to make a tradeoff between the use of an FPGA or a (dedicated) microcontroller. The question could be asked if the payload should have its own computer, or if it should be run off the main on board computer. When selecting an FPGA or microcontroller, it is advised to find one that can operate at 1.8 or even 1.2V to communicate directly to the memory. If this is not the case, it would be ideal to use a bi-directional voltage shifter. When using a dedicated computer, one should investigate the possibility to do more or less real time data processing while each page is being read off the memory. Another path to look into is whether a part of the 3D NAND memory can still be used to store (the test) data. The payload could run completely independent, from which only interesting data results are pushed to the main on board computer.

Without developing a new test setup, it would be interesting to further improve the current test setup by working on the timing bottlenecks described in Section 5.3. 66% of one read cycle is spent saving the data in Python and 23% to transfer the data over the UART protocol, while the actual read operation is only 35 ms or 5% of the overall time. The best option would be to extend the current VHDL code with a data processing unit on the FPGA. Otherwise, investigate the implementation of a faster communication protocol and change from Python to a language capable of storing the received data faster.

7.2.4 Investigation of different 3D NAND technology

The Micron technology was chosen because of the availability of the READ OFFSET function. This was found in the data sheet which was available for download upon registering on their website. Micron uses the floating gate technology, while others based their 3D architecture on charge trap flash cells. It is advised to contact the other suppliers (Samsung, SK Hynix, Toshiba/WD), request data sheets and check if a similar READ OFFSET function is available. One might have to go through their business portal or contact local sales representatives.

7.2.5 Sourcing rawNAND flash memories

It was proven that unsoldering BGA memory packages from a commercially bought SSD was a feasible option to obtain working rawNAND 3D NAND memories. We have relied on the NFI for the unsoldering process. It is not clear if their staff/facilities will always be available for this purpose. It is recommended to investigate other options within the university where the unsoldering can take place. Reballing the chips will not necessarily be needed when using pinned sockets.

Bibliography

[1] M Bagatin, S Gerardin, A Paccagnella, S Beltrami, E Camerlenghi, M Bertuccio, A Costantino, A Zadeh, G Santin, and E Daly. Effects of Heavy-Ion Irradiation in Vertical 3D NAND Flash Memories. 2017.

[2] Marta Bagatin, Simone Gerardin, Alessandro Paccagnella, Veronique Ferlet-Cavrois, James R. Schwank, Marty R. Shaneyfelt, and Angelo Visconti. Proton-induced upsets in SLC and MLC nand flash memories. *IEEE Transactions on Nuclear Science*, 60(6):4130–4135, 2013. ISSN 00189499. doi: 10.1109/TNS.2013.2290033.

[3] L. A. D. Bathen and N. D. Dutt. 3D-FlashMap: A Physical-location-Aware Block Mapping Strategy for 3D NAND Flash Memory. *Proc. Design, Automation & Test in Europe (DATE)*, pages 1307–1312, 2012. ISSN 1530-1591. doi: 10.1109/DATE.2012.6176694. URL `http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6176694`.

[4] G. Cellere, A. Paccagnella, A. Visconti, M. Bonanomi, and A. Candelori. Transient conductive path induced by a Single ion in 10 nm SiO2 Layers. *IEEE Transactions on Nuclear Science*, 51(6):3304–3311, 2004. ISSN 0018-9499. doi: 10.1109/TNS.2004.839146.

[5] G. Cellere, A. Paccagnella, A. Visconti, and M. Bonanomi. Subpicosecond conduction through thin SiO2 layers triggered by heavy ions. *Journal of Applied Physics*, 99(7), 2006. ISSN 00218979. doi: 10.1063/1.2181313.

[6] G. Cellere, A. Paccagnella, A. Visconti, S. Beltrami, J. Schwank, M. Shaneyfelt, D. Lambert, P. Paillet, V. Ferlet-Cavrois, J. Baggio, R. Harboe-Sorensen, E. Blackmore, A. Virtanen, and P. Fuochi. Direct evidence of secondary recoiled nuclei from high energy protons. *IEEE Transactions on Nuclear Science*, 55 (6):2904–2913, 2008. ISSN 00189499. doi: 10.1109/TNS.2008.2007799.

[7] Giorgio Cellere, Paolo Pellati, Andrea Chimenton, Jeff Wyss, Alberto Modelli, Luca Larcher, and Alessandro Paccagnella. Radiation effects on floating-gate memory cells. *IEEE Transactions on Nuclear Science*, 48(6 I):2222–2228, 2001. ISSN 00189499. doi: 10.1109/23.983199.

[8] Dakai Chen, Edward Wilcox, Raymond Ladbury, Christina Seidleck, Hak Kim, Anthony Phan, and Kenneth LaBel. Heavy Ion and Proton-induced Single Event Upset Characteristics of a 3D NAND Flash Memory. *IEEE Transactions on Nuclear Science*, 65(1):19–26, 2017. ISSN 00189499. doi: 10.1109/TNS.2017.2764852.

[9] Digilent. ZedBoard Hardware User's Guide. Technical Report January, 2014.

[10] S Gerardin, M Bagatin, A Paccagnella, K Grürmann, F Gliem, T R Oldham, F Irom, and D N Nguyen. Radiation Effects in Flash Memories. *Ieee Transactions on Nuclear Science*, 60(3):1953–1969, 2013. ISSN 0018-9499. doi: 10.1109/TNS.2013.2254497.

[11] Kai Grümann, Dietmar Walter, Martin Herrmann, Fritz Gliem, and Heikki Kettunen. SEU and MBU Angular Dependence of Samsung and Micron 8-Gbit SLC NAND-Flash Memories under Heavy-Ion Irradiation. pages 6–10, 2011.

[12] Glenn F. Knoll. *Radiation Detection and Measurement.* Fourth edi edition, 2010.

[13] L. Larcher, G. Cellere, A. Paccagnella, A. Chimenton, A. Candelori, and A. Modelli. Data Retention after Heavy Ion Exposure of Floating Gate Memories: Analysis and Simulation. *IEEE Transactions on Nuclear Science*, 50(6 I):2176–2183, 2003. ISSN 00189499. doi: 10.1109/TNS.2003.821598.

[14] Rino Micheloni. *3D flash memories*. 2016. ISBN 9789401775120. doi: 10.1007/978-94-017-7512-0.

[15] Rino Micheloni, Luca Crippa, and Alessia Marelli. *Inside NAND flash memories*. Springer Science + Business Media, 2010. ISBN 978-90-481-9430-8. doi: 10.1007/978-90-481-9431-5. URL `http://link.springer.com/10.1007/978-90-481-9431-5`.

[16] Micron. B17A FortisFlash. 2018.

[17] ONFI. Open NAND Flash Interface Specification 4.0. *Flash*, 2014.

[18] Krishna Parat and Chuck Dennison. A floating gate based 3D NAND technology with CMOS under array. *Technical Digest - International Electron Devices Meeting, IEDM*, 2016-Febru:3.3.1–3.3.4, 2015. ISSN 01631918. doi: 10.1109/IEDM.2015.7409618.

[19] Ki Tae Park, Sangwan Nam, Daehan Kim, Pansuk Kwak, Doosub Lee, Yoon He Choi, Myung Hoon Choi, Dong Hun Kwak, Doo Hyun Kim, Min Su Kim, Hyun Wook Park, Sang Won Shim, Kyung Min Kang, Sang Won Park, Kangbin Lee, Hyun Jun Yoon, Kuihan Ko, Dong Kyo Shim, Yang Lo Ahn, Jinho Ryu, Donghyun Kim, Kyunghwa Yun, Joonsoo Kwon, Seunghoon Shin, Dae Seok Byeon, Kihwan Choi, Jin Man Han, Kye Hyun Kyung, Jeong Hyuk Choi, and Kinam Kim. Three-dimensional 128 Gb MLC vertical nand flash memory with 24-WL stacked layers and 50 MB/s high-speed programming. *IEEE Journal of Solid-State Circuits*, 50(1):204–213, 2015. ISSN 00189200. doi: 10.1109/JSSC.2014.2352293.

[20] Sergei Skorobogatov. The bumpy road towards iPhone 5C NAND mirroring. pages 1–10, 2016.

[21] E.G. Stassinopoulos and J.P. Raymond. The space radiation environment for electronics. *Proceedings of the IEEE*, 76(11):1423–1442, 1988. ISSN 00189219. doi: 10.1109/5.90113. URL `http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=90113`.

[22] Tomoharu Tanaka, Mark Helm, Tommaso Vali, Ramin Ghodsi, Koichi Kawai, Jae Kwan Park, Shigekazu Yamada, Feng Pan, Yuichi Einaga, Ali Ghalam, Toru Tanzawa, Jason Guo, Takaaki Ichikawa, Erwin Yu, Satoru Tamada, Tetsuji Manabe, Jiro Kishimoto, Yoko Oikawa, Yasuhiro Takashima, Hidehiko Kuge, Midori Morooka, Ali Mohammadzadeh, Jong Kang, Jeff Tsai, Emanuele Sirizotti, Eric Lee, Luyen Vu, Yuxing Liu, Hoon Choi, Kwonsu Cheon, Daesik Song, Daniel Shin, Jung Hee Yun, Michele Piccardi, Kim Fung Chan, Yogesh Luthra, Dheeraj Srinivasan, Srinivasarao Deshmukh, Kalyan Kavalipurapu, Dan Nguyen, Girolamo Gallo, Sumant Ramprasad, Michelle Luo, Qiang Tang, Michele Incarnati, Agostino MacErola, Luigi Pilolli, Luca De Santis, Massimo Rossini, Violante Moschiano, Giovanni Santin, Bernardino Tronca, Hyunseok Lee, Vipul Patel, Ted Pekny, Aaron Yip, Naveen Prabhu, Purval Sule, Trupti Bemalkhedkar, Kiranmayee Upadhyayula, and Camila Jaramillo. A 768Gb 3b/cell 3D-floating-gate NAND flash memory. *Digest of Technical Papers - IEEE International Solid-State Circuits Conference*, 59:142–144, 2016. ISSN 01936530. doi: 10.1109/ISSCC.2016.7417947.

[23] TechInsights. Intel/Micron 64L 3D NAND Analysis. URL `www.techinsights.com/blog{_}page-2017.aspx?pageid=27917296266`.

Appendix A

PCB design

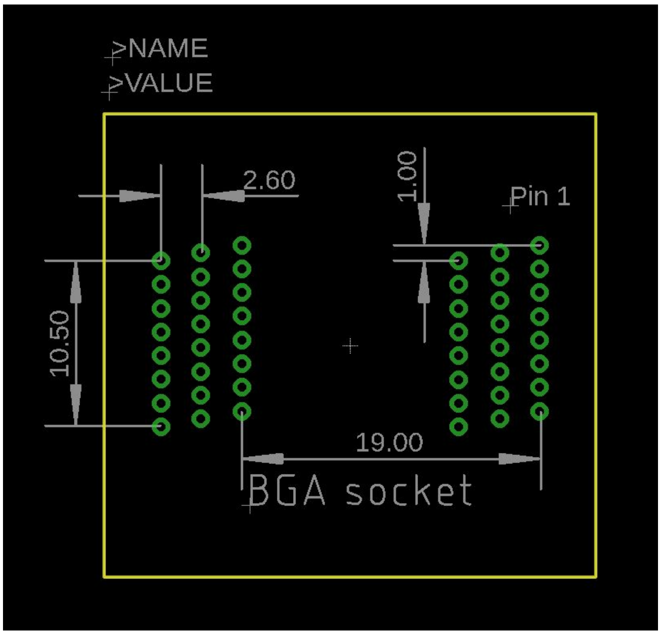


Figure A.1: A custom component was created for the purchased BGA socket. The image indicates the dimensions that were used for the holes one the PCB to accomodate the socket pin layout.

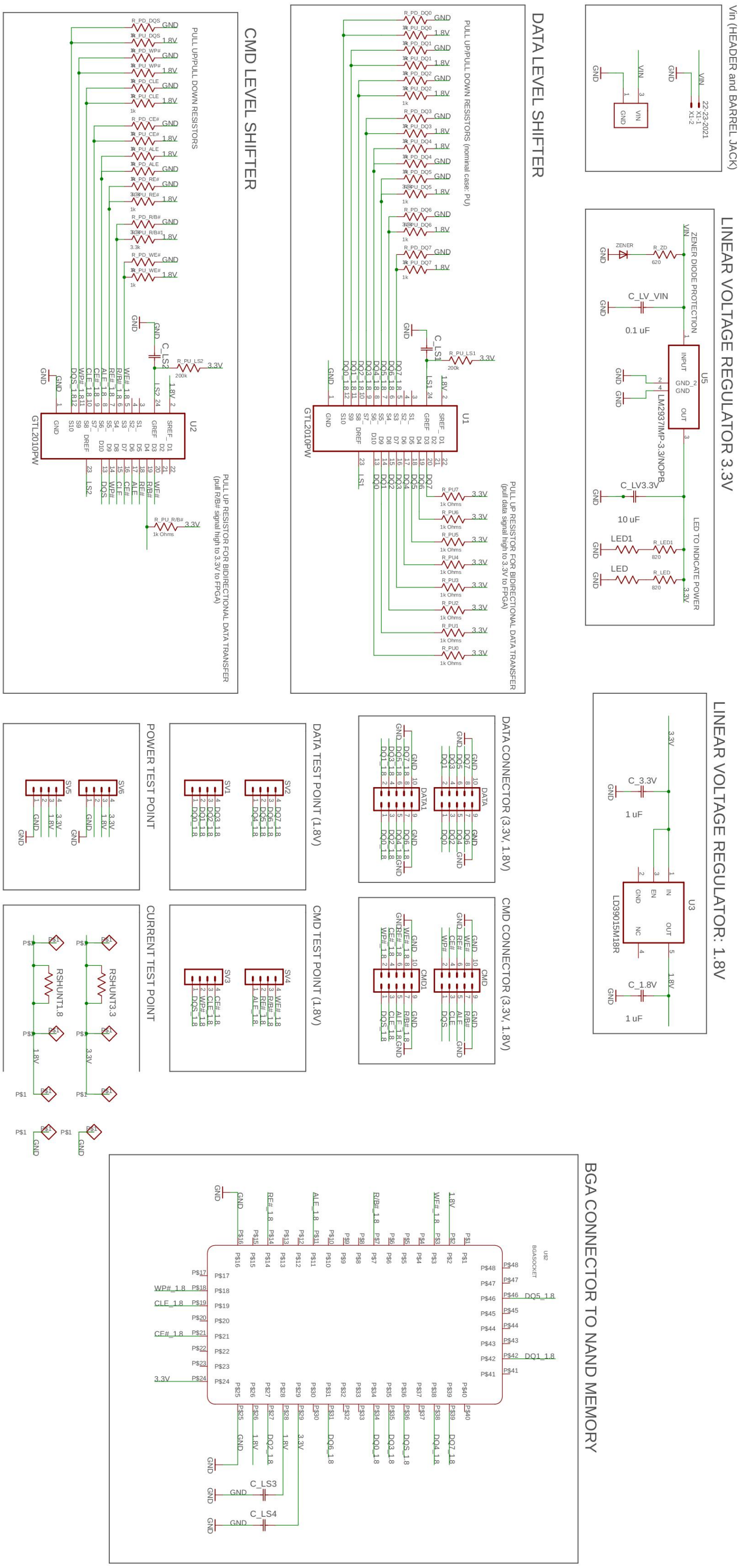
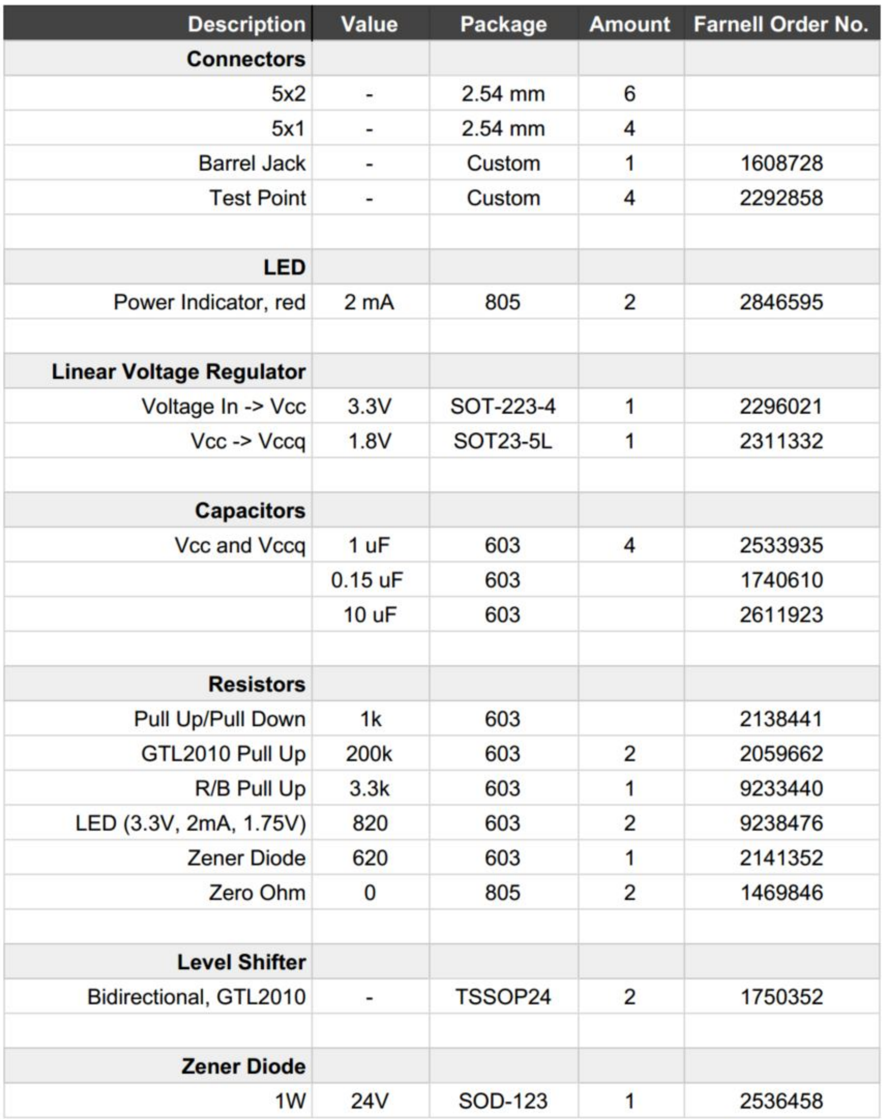

CMD LEVEL SHIFTER
PULL UP/PULL DOWN RESISTORS
PULL UP RESISTOR FOR BIDIRECTIONAL DATA TRANSFER (pull R/B# signal high to 3.3V to FPGA)
DATA LEVEL SHIFTER
PULL UP/PULL DOWN RESISTORS (nominal case: PU)
PULL UP RESISTOR FOR BIDIRECTIONAL DATA TRANSFER (pull data signal high to 3.3V to FPGA)
Vin (HEADER and BARREL JACK)
LINEAR VOLTAGE REGULATOR 3.3V
ZENER DIODE PROTECTION
LED TO INDCATE POWER 3.3V
LM2937IMP-3.3/NOPB
LINEAR VOLTAGE REGULATOR: 1.8V
LD39015M18R
POWER TEST POINT
DATA TEST POINT (1.8V)
DATA CONNECTOR (3.3V, 1.8V)
CURRENT TEST POINT
CMD TEST POINT (1.8V)
CMD CONNECTOR (3.3V, 1.8V)
BGA CONNECTOR TO NAND MEMORY
GTL2010PW
U1
U2
U3
U5
RSHUNT1.8
RSHUNT3.3

Figure A.2: EAGLE schematic of the PCB to interface the 3D NAND memory with the ZedBoard.



Description	Value	Package	Amount	Farnell Order No.
Connectors				
5x2	-	2.54 mm	6	
5x1	-	2.54 mm	4	
Barrel Jack	-	Custom	1	1608728
Test Point	-	Custom	4	2292858
LED				
Power Indicator, red	2 mA	805	2	2846595
Linear Voltage Regulator				
Voltage In -> Vcc	3.3V	SOT-223-4	1	2296021
Vcc -> Vccq	1.8V	SOT23-5L	1	2311332
Capacitors				
Vcc and Vccq	1 uF	603	4	2533935
	0.15 uF	603		1740610
	10 uF	603		2611923
Resistors				
Pull Up/Pull Down	1k	603		2138441
GTL2010 Pull Up	200k	603	2	2059662
R/B Pull Up	3.3k	603	1	9233440
LED (3.3V, 2mA, 1.75V)	820	603	2	9238476
Zener Diode	620	603	1	2141352
Zero Ohm	0	805	2	1469846
Level Shifter				
Bidirectional, GTL2010	-	TSSOP24	2	1750352
Zener Diode				
1W	24V	SOD-123	1	2536458

Figure A.3: Bill of material to populate the PCB presented in Section 4.3.3, including Farnell order numbers.