FLASHRAD: A Reliable 3D Rad Hard Flash Memory Cube Utilizing COTS for Space

Da Eun Shim
Georgia Institute of Technology
266 Ferst Drive
Atlanta, GA 30332
daeun@gatech.edu

Amanvir Singh Sidana
Georgia Institute of Technology
266 Ferst Drive
Atlanta, GA 30332
amanvir.sidana@gatech.edu

Jim S. Yamaguchi
Irvine Sensors Corporation
3001 Red Hill Avenue, B3-108
Costa Mesa, CA 92626
jyamaguchi@irvine-sensors.com

Christian Krutzik
Irvine Sensors Corporation
3001 Red Hill Avenue, B3-108
Costa Mesa, CA 92626
ckrutzik@irvine-sensors.com

Dan Nakamura
JPL-California Institute of Technology
4800 Oak Grove Drive
Pasadena, CA 91109
Daniel.I.Nakamura@jpl.nasa.gov

Sung Kyu Lim
Georgia Institute of Technology
266 Ferst Drive
Atlanta, GA 30332
limsk@ece.gatech.edu

Abstract—**Given the rapid rate of growth and scope for space missions, improving computing capabilities of onboard spacecraft and memory systems is vital for future space missions. Currently, many space missions are limited by memory capacity as there is not enough onboard memory to store the copious amounts of data obtained in a single space mission. Onboard memory systems must also be capable of providing the necessary operational robustness and fault tolerance based on system and mission requirements. Some mission requirements may include data-intensive operations such as terrain navigation, hazard detection and avoidance, autonomous planning, and onboard science data processing. Thus, space missions require onboard memory that has high bandwidth, high capacity, and high reliability to securely store recorded data when exposed to space radiation. To meet the aforementioned necessities, various designs of memory cubes that make use of horizontal integration of memory dies have been proposed. However, these methods require high design effort and a long lead time. Alternatively, a loaf-of-bread (LOB) design using vertical integration of DDR3 SDRAM dies has recently been proposed. Straightforward access to each individual die from the memory controller is possible in the LOB design due to the vertical integration of the dies which allows Commercial-Off-The-Shelf (COTS) dies to be used, reducing cost and lead time for designers.**

In this paper, we propose a method to effectively increase data storage for onboard memory while reducing the cost and effort that goes into design by presenting a 3D memory cube design utilizing 24 COTS NAND flash dies in a LOB configuration. The design includes various features that increase the data storage available while considering hazards specifically in space environments such as errors from single event effects (SEE) or single event functional interrupt (SEFI) events. Currently, the preliminary RTL code is ready with support for NAND Flash commands, error-correcting codes (ECC), and scrubbing. Features such as wear leveling, bad block management, data scrambling and a serial rapid IO (SRIO) interface to further mitigate errors due to radiation effects in the space environment will be incorporated in the future. The functionality of the memory controller has been verified via simulation of the RTL code. Further validation and testing using a FPGA board are also underway to verify the design at this stage. Therefore the proposed design addresses the need for increased memory storage while also allowing COTS dies to be used. This paves way for reduced design efforts as well as the incorporation of state-of-the-art memory dies in space missions.

TABLE OF CONTENTS

1. INTRODUCTION 1
2. DESIGN OF MEMORY CUBE 2
3. DESIGN OF MEMORY CONTROLLER 3
4. TEST SETUP 5
5. CONCLUSION 6
ACKNOWLEDGMENTS 7
REFERENCES 7
BIOGRAPHY 7

978-1-5386-6854-2/19/$31.00 ©2019 IEEE

1. INTRODUCTION

The concept of space exploration has been at the forefront of human excellence and development for decades. The current era shows a rapid increase of space missions as well as a rapid growth in scope for each space mission. At this time, the computing capabilities of onboard spacecraft are becoming a major limiting factor for accomplishing many classes of future missions. Although there are efforts to improve spacecraft central processing units (CPUs), attempts to address the limitations of current onboard memory systems are rarely made. With data-intensive operations such as terrain navigation and onboard science data processing, memory devices with high-bandwidth, high-capacity, and high-reliability is required to maximize the mission data storage. Following the stacking methodologies of [1] [2] [3], Agnesina et al [4] came up with a 3D DRAM cube with dies integrated in a loaf-of-bread (LOB) configuration in 2018. This contributes to the active memory portion of the high performance space computer (HPSC). However, there has not been an equivalent effort to improve the storage memory of the HPSC. In this paper, we will focus on enhancing the storage space for onboard systems using Flash memory.

With planar NAND flash slowly reaching its limit in terms of capacity due to challenges in further scaling, industry is looking into alternative approaches to increase the capacity of flash memory such as 3D integration of memory devices. Several companies have been investing in research for 3D NAND structures in order to obtain higher memory density. Starting with the Samsung 3D V-NAND in 2012, other 3D NAND Flash products from companies such as Toshiba, SK Hynix and Micron continued to arrive in the commercial market one after another. Smaller transistors have lower tolerances to Single Event Effects (SEE) or Single Event Functional Interrupts (SEFIs) and thus are less reliable in a radiation environment. Due to this reason, many electronic devices for space applications are 1 to 2 generations behind state-of-the-art technology.

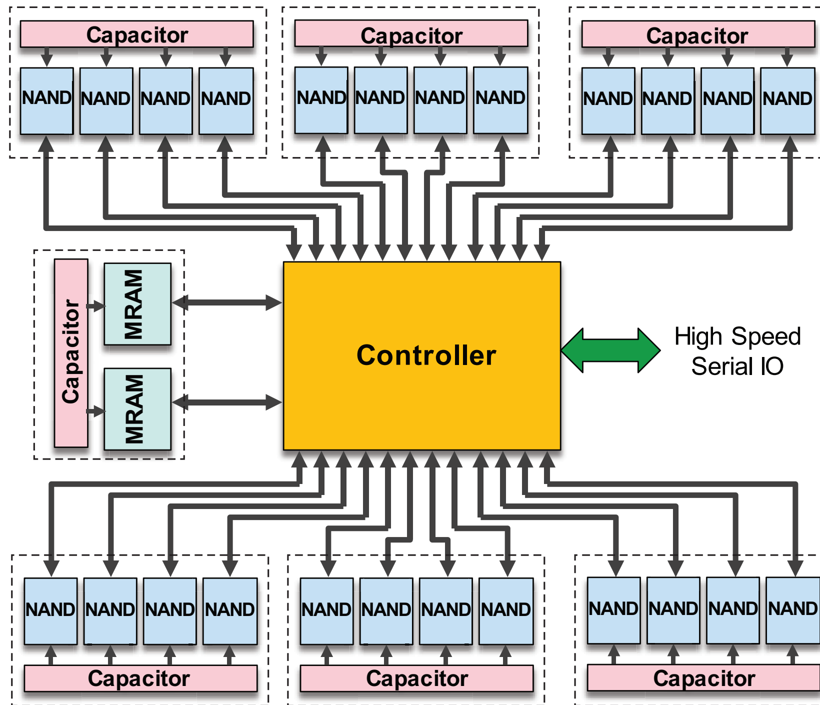


Figure 1. Cube Configuration Diagram

In addition, due to the fact that there is considerably higher demand and use in the commercial market, the main target of these memory companies is for commercial use. Thus, the space community cannot avoid having to custom build their own onboard memory systems. Despite the situation, there is an ongoing desire to use Commercial-Off-The-Shelf (COTS) devices in the space community due to cost reduction and lead time reduction as well as increase in hardware capabilities within smaller volume constraints. Another benefit of using COTS devices would be the availability of cutting edge technology.

In 2014, 3D-Plus released the radiation tolerant intelligent memory stack (RTIMS) Flash, which would be one of the first efforts to stack NAND Flash dies with a microprocessor based memory controller for use in space. Although it provides a space-ready flash cube with protection against radiation, the memory capacity is not that high as it offers up to only 24 Gb of memory. With end cap shields and various error correction methods, the memory capacity can be increased greatly. In this paper, we present a 3D NAND flash cube design which utilizes 24 COTS dies in a LOB configuration which effectively increases data storage for onboard memory while reducing cost and design effort.

2. Design of Memory Cube

In the presented design, 24 identical 32Gb NAND Flash dies is stacked together with a single memory controller connected to each die separately in order to obtain a total of 768 Gb of memory space. Each flash die has an 8-bit interface for input/output. The configuration of the design can be seen in Figure 1 where 24 Flash dies are connected to the controller and each of the four dies share a capacitor layer. The MRAMs shown in the figure will be used for caching and Flash Translation Layer (FTL) which will be further explained in section 3.

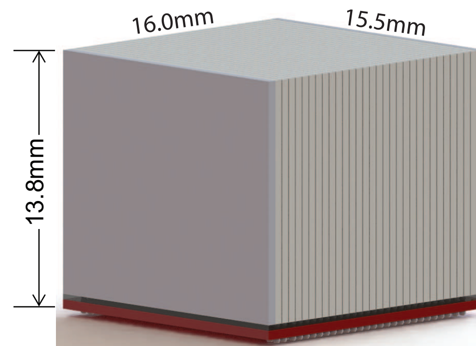


Figure 2. Our Memory Cube

Stacking Method

In order to maximize inter-connectivity and thermal performance, the loaf-of-bread (LOB) configuration first proposed by Agnesina et al [4] and which was originally used for a DRAM cube has been used for the Flash memory cube. With 24 NAND Flash dies vertically stacked together, the approximate dimensions of the cube are 13.8mm by 16.00mm by 15.5mm with increased memory space compared to the RTIMS from 3D Plus. Figure 2 shows the Flash cube with the approximate dimensions.

With the die in LOB configuration, straightforward access to individual dies within the cube for power management and increased bandwidth capabilities can be achieved. It also minimizes electrical parasitics which helps reduce IO drive requirements and improves signal integrity.

NAND flash dies from multiple manufacturers have been investigated for acceptable radiation tolerance. Micron dies have been selected to be used for implementation of our cube

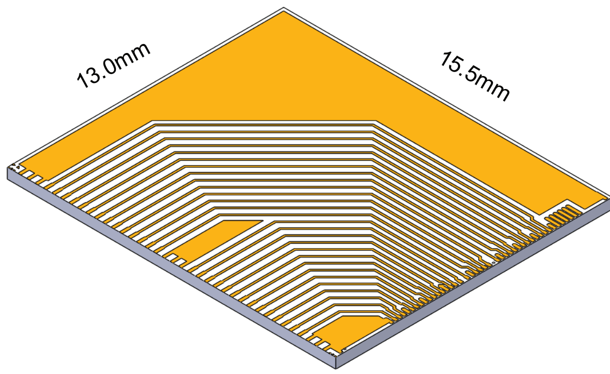


Figure 3. Reroute Distribution Layer

and thus Micron die parameters will be used for comparison against the 3D Plus' RTIMS. The comparison of the cube's estimated radiation tolerance to the 3D plus RTIMS' radiation tolerance can be found in Table 1. Due to difficulties in custom radiation testing and lack of radiation testing information from the manufacturers, the radiation tolerance data for Micron dies from [5] has been used.

Reroute Distribution Layer

With the cube being in the LOB configuration, it is necessary to connect the interconnects of each die to the memory controller. In order to bring the necessary interconnections from the die I/O to the bottom edge of the die, a Reroute Distribution Layer (RDL) is applied to each die. These dies are then stacked together to form the cube structure. Figure 3 shows the designed RDL for this cube with the dimensions being 13.0mm by 15.5mm. With a multi-layer RDL design, both the impedance control of the design circuit and the power distribution of the die are improved. The face of the cube where the lead extensions go to the edge will be processed with an isolation dielectric to allow for formation of interconnect pads that will subsequently be used as the I/O for the cube.

End Cap Shields

With the LOB stacking method, end caps can be added to each side of the stack for additional radiation shielding as well. The end cap adds very little additional size and weight for a stacked structure compared with a 2D layout of individual packages. Figure 4 shows the design of the end cap using both high-z materials to protect the cube against high energy photons such as X-rays and γ-rays and low-z materials to absorb secondary radiation from highly penetrative γ-rays. Although the end caps do not provide perfect protection, they do provide a significant reduction in the overall radiation cross-section of the device. A design tradeoff can also be made between the shielding effectiveness and the individual die thickness and number of layers.

3. Design of Memory Controller

For the memory controller of the Flash cube, a processor based implementation with firmware based control will be implemented. Figure 5 shows the block diagram of the whole system including both the processor subsystem and the hardware data path. The hardware data path consists of all Flash controller functions required to decouple the Flash management from the host system. The interface to the host system will consist of a high-speed serial IO in the form of daisy-chainable Serial Rapid IO (SRIO). The host will access the NAND Flash array using the SRIO interfaces as a linear-addressable memory space with no concerns of typical Flash management as the 3D memory module will handle all management functions. This will provide a simple to use interface from the host's perspective.

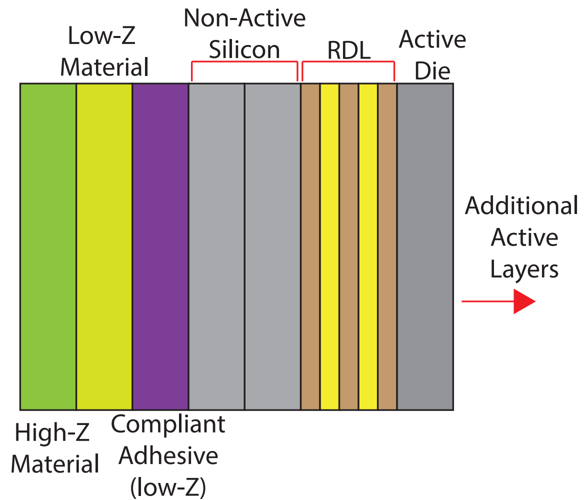


Figure 4. Radiation Shield Layering for the 3D Memory Cube Ends

The NAND controller and the ECC has been implemented and is marked in Figure 5. The block diagram is color-coded depending on whether the module is implemented via hardware, software, or a mixture of the two.

Basic Functions

Basic read, write, erase, and reset functions has been implemented and tested. The test results of the simulated code can be found in Section 4. The interface will be to each die individually, so all the dies may be accessed at the same time, i.e. 192bit bandwidth, but not with the same set of functions. This way, each byte channel can have a different instruction. By accessing the dies at the same time, the usage of the dies can be distributed and this will also be good for wear leveling.

Error Correcting Code

In general, NAND Flash requires more control and error mitigation compared to the conventional random access memories (RAM) due to their physical properties including but not limited to bad blocks, limited program/erase cycles, having different sizes for read/write and erase. Since the memory cube is designed for use in space, it is more vital to have as much Error Correcting Code (ECC) done as possible in order to deal with the additional radiation effects on top of NAND Flash's inherent issues. For performance reasons, direct implementation of ECC in hardware is required. NAND Flash devices inherently have spare space in them which is typically used for ECC, wear leveling, and other error correcting modules. This spare space in each Flash die will mainly be used for ECC separately allowing fully parallel operation within the stack. Hamming code is not sufficient enough for a space application. Thus, in this memory cube, Bose-Chaudhuri-Hocquenghem code (BCH) is used.

Table 1. Comparison table of FLASHRAD to RTIMS

Parameters	3D Plus RTIMS	Our Cube : FLASHRAD
TID	>50krad (Si)	>90 krad (Si)
SEU	Immune by design	Immune by design
SEFI	Immune by design	Immune by design
Capacity	48Gb (8Gb with TMR protection, 16Gb with EDAC memory protection, 24Gb with no additional memory protection)	768Gb
Dimensions	31mm*28mm*11.2mm	13.8mm*16.00mm*15.5mm

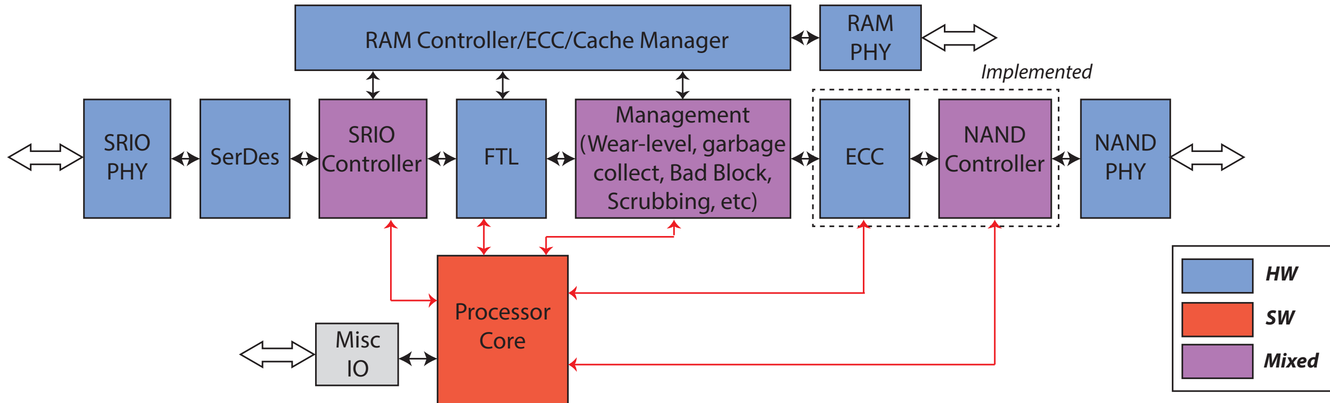


Figure 5. Flash Controller Block Diagram

Scrubbing

In addition to the basic ECC, scrubbing for the cube is used as a background task which will re-write a whole block if read errors exceed a certain threshold. The scrub rate and process will be fully programmable and the scrub algorithm will utilize the wear-leveling meta-data in order to process new address locations if block relocation is required. The baseline algorithm will be to use a combination of program/erase count and read errors to determine scrubbing action. If read errors exceed a certain threshold, the block will be re-written or relocated depending on the program/erase count. The processor will also be able to perform data movement from Flash to RAM which allows it to temporarily store the block to be relocated if required.

Bad Block Management

NAND Flash has bad blocks issues, but factory bad blocks are usually mapped out during production and do not become a big issue as they are isolated from the valid blocks. However, NAND Flash devices can also develop grown bad blocks during their lifetime that need to be mapped out of usable space when encountered. In order to resolve this issue, the controller will monitor the write status of every write as well as the maximum bit errors corrected during a read. These indicators will be used to determine if a block is ready to be retired. This function is performed on the top level directly in hardware where a faulty write will trigger a new write using the next available free block address while the failing address is stored in a small FIFO. This also requires additional logic that ties back to the write RAM buffer so that new data is not processed into the buffer until a successful write. The block address is also temporarily removed from the free list. Similarly, on each read the ECC will determine the read errors. If the errors exceed a threshold, the block address is entered into the FIFO as well with appropriate bit marker allowing various cases to be distinguished.

In addition to the hardware path, the processor core will receive an interrupt from the FIFO alerting it that there is a potential bad block. For the case of a write error, after the block has already been moved to a new location, the processor will erase the potential bad block, write a test pattern, and recheck status in the background. If the block passes, then it will be erased and returned to the free list. If it fails the second time, the block is marked as bad and added to the bad block table. Note that each block is marked bad in addition to storing a bad block table in the Flash array which provides redundancy. In the case of read error count, the processor may choose to reread the address and monitor the errors. If a potential error condition is ascertained, the processor will trigger the scrubbing module to move the block. The threshold set to the ECC module would be more conservative allowing the processor to make a final decision. The processor may then execute additional write/read of a test pattern to determine if the block should be retired. The processor code can be tuned to be more or less aggressive for bad block determination as desired.

Wear Leveling

In order to implement wear leveling, the Flash controller will be designed to store both the meta-data of total writes in the first page of each block and a SRAM based lookup table containing free blocks. The free block table will be maintained as two tables - one that holds the full free block addresses and a smaller, more optimized, partial table that hold the blocks addresses with the lowest Program/Erase count. The partial table will be maintained by the processor and will be implemented as a write-addressable FIFO in hardware in the background by scanning the Flash array for the lowest Program/Erase cycle blocks. As the processor scans the array, it will add blocks to the the free block table whenever a block

has less Program/Erase cycles than the lowest Program/Erase cycle block in the table. The processor will update the table during garbage collection as well as scrubbing as needed. The partial free block table will be stored in volatile memory as it can be generated on each power-up.

In addition to the partial free block table, the full free block table is generated from the FTL (by the logical block table) as it naturally contains all available used/free blocks. However, as the FTL is addressed by logical blocks, it requires a full scan of the logical block table to generate. This will be performed at power-up by the processor. Under peak, sustained write conditions, it is possible for the partial free block table to empty in which case free blocks will be selected by the FTL in a round-robin fashion. This will incur only minimal write imbalance and can be throttled by processor if required.

SRIO Interface

Serial Rapid IO protocol will be used for the interface protocol and will be designed to support version 3.1 specifications at a minimum which added space compliance support. The in/out ports will be fully buffered and the in-out path will be internally processed by the controller allowing for advanced features. The SRIO will operate at multi gigabit speeds which will unfortunately add additional latency to each module in the daisy-chain. However, the delay can become transparent to each module in the daisy-chain with proper processing methods and since the daisy-chain works in a pipeline mode, the length of the delay should not affect the throughput. The SRIO interfaces to the wide-word interface of the 3D stack through a serializer-deserializer (SERDES) block which takes the SRIO data at Gbps rates and reduces it to a slower clock domain for processing. Since the cube includes multiple layers, the wide-word data in the lower clock domain is easily consumed by the Flash devices.

Using SRIO as the Flash IO along with daisy chain support allows the Flash controller to be designed with support for host-independent, n-modular redundancy (NMR) modes when required for increased reliability operation. Additionally, for cases where only certain data is system critical, the redundancy can be implemented within a subset of modules in a chain as shown in Figure 6 which shows a Triple-Modular Redundancy (TMR) technique. The redundancy can be supported in a transparent manner such that the host processor requires no additional processing or decoding. This means that a redundant configuration of Flash modules is accessed in the same manner as a single device as far as the host processor is concerned. The Flash controller would take the responsibility of passing along write operations to replicate the data among grouped modules, issuing reads to the redundant modules, and performing the voting. The device closest to the host processor in the chain is referred to as the ”edge” module as shown in Figure 6 and performs as the additional processor in terms of replicating and addressing additional modules, off-loading the host processor from any special duties; the ”redundant” modules are those that contain redundant data copies along with the edge module.

Flash Translation Layer

The Flash Translation Layer (FTL) will perform the logical to physical block mapping. This allows the external interface to have consistent block addressable interfaces without having to know the internal remapping operations. FTL needs fast access to the logical block table as address translation will be required for every read and write operation. A separate RAM device stores the logical block table during operation to provide fast access. The RAM device is part of the stack and will be transparent to the users.

To reduce controller complexity, the logical block table is designed as a 4-byte wide lookup table on a 16 sector (8kB) granularity. Basic operation is shown in Figure 7 and will be implemented in hardware. The block table stores the physical block number such that the upper bits of the sector address can be used to directly address the RAM memory to obtain the corresponding physical block. Within the physical block, the sectors will be distributed sequentially and will therefore be addressed without further translation.

The logical block table will be stored into Flash at various checkpoints during operation to reduce impact of potential unexpected power loss or other radiation effects. Redundant copies is stored across multiple dies to mitigate radiation effects and Flash block failures. For further protection, the logical block address will be stored as part of the meta-data for each page. This allows reconstruction of the block table if required. Furthermore, it provides a mechanism for the Flash controller to verify that the proper block has been addressed to reduce potential radiation induced errors on the control logic. As multiple pages are stored per block, there is an inherent level of redundancy that further protects block mapping info.

Garbage Collection

The garbage collection will be performed as a background process. Primary garbage collection occurs via a FIFO interface from the management module and NAND controller module that keeps track when Flash blocks are relocated. The processor will interrogate the FIFO and erase those blocks in idle periods. Furthermore, the processor can scan the logical block table and Flash array to determine any erasable stale blocks that may be a result of incomplete collection due to power-down or radiation effects.

4. TEST SETUP

Verification of Code

Preliminary verification of the RTL code for the memory controller using the Micron Verilog model has been done with Vivado. The simulations show that the basic Flash functions work as expected. An example of the results is shown in Figure 9 where a write program is in operation.

FPGA Prototyping

The processor subsystem will make extensive use of the DesignStart Cortex-M0/3 processor available from ARM which provides configurable RTL for the core processor and application examples for peripheral connectivity and development. It also provides necessary tools to design and test on a simulator and then proceed with hardware prototyping using an FPGA. The first part of the RTL coding will use the MPS2+ platform as it provides ready-to-use processor evaluation designs.

After updating and testing the core configuration and peripheral connectivity on the MPS2+ platform, the design will be migrated to the selected FPGA (Xilinx FPGA) used for the 3D module Prototype system. The MPS2+ will provide a valuable reference for test and debug throughout the development. However, the MPS2+ board does not have sufficient FPGA resources to implement the necessary Flash connectivity and SRIO which is why a custom board must be

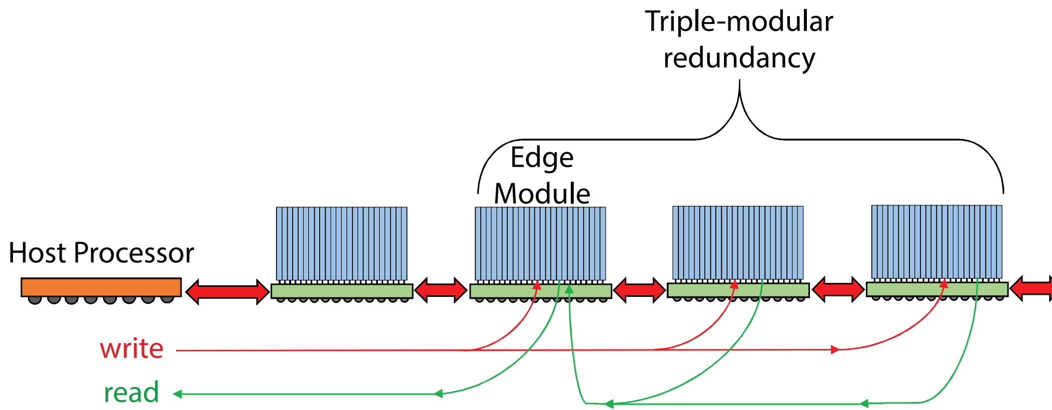


Figure 6. N-Modular Redundancy Operational Diagram (TMR)

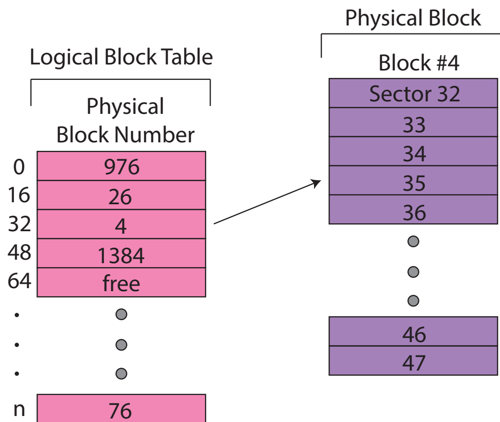


Figure 7. Logical Block Table Structure

developed to interface with the 3D stack.

The processor subsystem will be then further developed using a combination of the DesignStart Processor IP and custom logic circuitry. Sample projects and RTL are provided by ARM for this board and can be used to develop and debug an initial processor subsystem. Only the skeleton connectivity will be evaluated at this time. Development of the necessary peripherals including the internal bus structure and definition of the peripheral memory map will also be developed. The DesignStart includes integration tests to validate the integration in a structured way once the overall subsystem is designed.

The processor subsystem development will also include the RTL for necessary test and debug features such as trace and JTAG ports. Programming of the processor core will be performed during this task to validate end-to-end operation. This will include setup and execution of the ARM toolchain (compiler, assembler, linker, etc). The ARM Mbed cloud program may be used for initial development as it provides evaluation designs for the MPS2+. Eventually, the gcc suite of tools is preferred as it is open source and can establish a baseline development path that can be easily utilized by potential customers of the 3D module.

Similar to the Flash controller module RTL, a COTS FPGA development board, such as a Xilinx Kintex-7 KC705, will be used for hardware testing while the final FPGA based 3D module prototype board is being designed. The COTS FPGA board will be selected to match the FPGA used in the prototype board so it will be a straightforward migration.

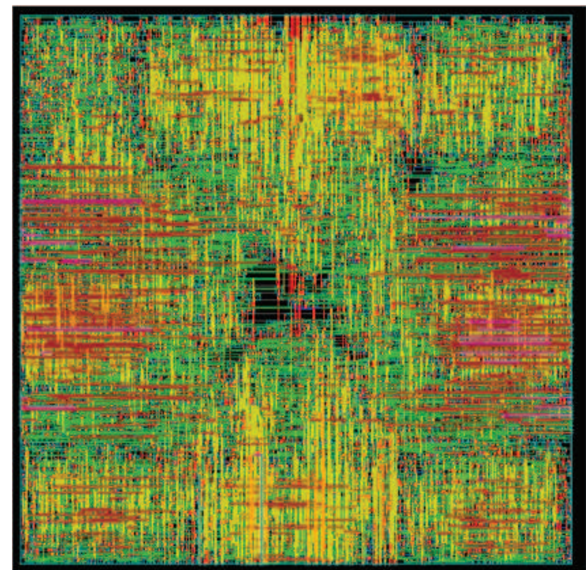


Figure 8. Synthesized Memory Controller RTL

Memory Controller RTL Synthesis

The RTL for the memory controller was synthesized using the NANGATE 45nm library. The fastest clock that could be achieved was 1.5ns (666.67MHz). Innovus was used to create the layout and run place and route (PnR) while Primetime was used to do static power analysis. Figure 8 shows the resulting layout and Table 2 shows the parameters for the controller when it is running on fastest clock frequency of 666.67MHz. It shows that the resulting design of the memory controller only consumes a total of $6.706mW$ of power.

5. CONCLUSION

This paper provides a method to effectively increase memory capacity using COTS dies to decrease cost and design effort. Although further efforts will be necessary to complete and make use of the Flashrad 3D NAND Flash memory cube, preliminary results of RTL simulations and ASIC design have shown promising results that indicate that the memory cube will succeed to support future space missions.

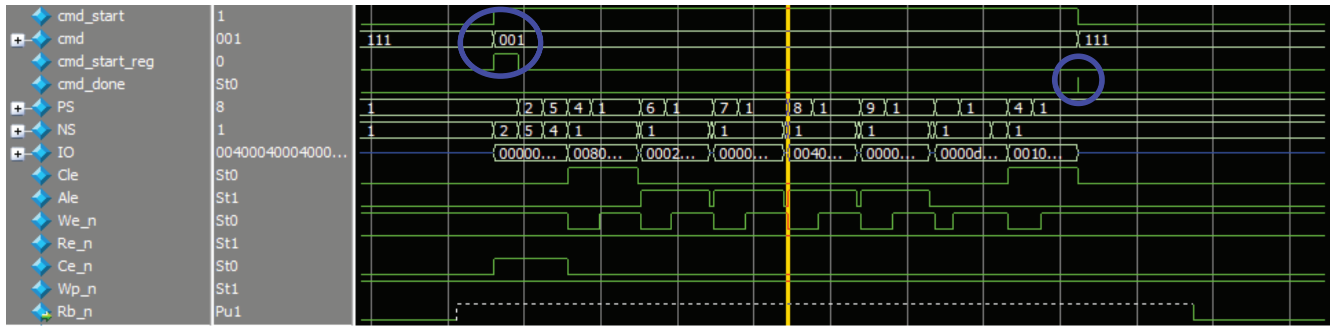


Figure 9. Simulated waveform for write page operation

Table 2. Controller Parameters at Fastest Clock Frequency

Parameters	Flash Controller Values
Cell Count	6520
FFs Count	851
Wirelength	152967.6050μm
Wire Cap	8.465pF
Pin Cap	14.442pF
Switching Power	3.3031mW
Internal Power	3.59mW
Leakage Power	81.4μW
Total Power	6.706mW

ACKNOWLEDGMENTS

This research is funded by the NASA SBIR Grant under the contract number NNX17CP47P. A portion of the research was carried out at the Jet Propulsion Laboratory, California Institute of Technology, under a contract with the National Aeronautics and Space Administration.

REFERENCES

[1] C. L. Bertin, D. J. Perlman, and S. N. Shanken, “Evaluation of a 3d memory cube system,” in *Proceedings of IEEE 43rd Electronic Components and Technology Conference (ECTC)*, 1993.

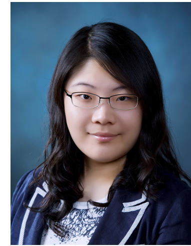
[2] R. S. J.C. Carson, “Electronic module comprising a stack of ic chips each interacting with an ic chip secured to the stack,” Patent, 1994.

[3] T. Go, “Method of fabricating electronic circuitry unit containing stacked ic layers having lead rerouting,” Patent, 1991.

[4] A. Agnesina, A. Sidana, J. Yamaguchi, C. Krutzik, J. Carson, J. Yang-Scharlotta, and S. K. Lim, “A novel 3d dram memory cube architecture for space applications,” in *Proceedings of the 55th Annual Design Automation Conference*, ser. DAC ’18. ACM, 2018.

[5] K. Grurmann, M. Herrmann, and R. Jyvaskyla. (2014) TN-IDA-RAD-14/14 Radiation Hard Memory (RHM) Comparison between NAND-Flash and DDR3 SDRAM Test Results.

BIOGRAPHY



Da Eun Shim *received her B.S. degree in general engineering from Harvey Mudd College, CA, USA in 2016. She is currently pursuing the Ph.D. degree with the School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA, USA. Her current research activities include 3D memory cubes and exploration of Monolithic 3D IC design limits and PPA, and 3D memory cubes.*

Amanvir Singh Sidana *received his masters degree in Electrical and Computer Engineering from Georgia Institute of Technology, Atlanta, GA, USA in 2017. He has worked as a research assistant at GTCAD lab on design of flash memory controllers under Prof. Sung Kyu Lim during his masters. He is currently working as a GPU RTL design engineer at Samsung Austin Semiconductor.*

Jim S. Yamaguchi *is VP of Irvine Sensors Corporations 3D Electronics and Mass Storage group. He received his B.A. in Chemistry in 1977 and his B.S. in Chemical Engineering from California State University, Long Beach. He joined Irvine Sensors Corporation in 1993. Mr. Yamaguchi has over 39 years of hands-on processing experience in the areas of 3D packaging, thin film deposition, electroplating, high density MCM fabrication and PWB fabrication. He has extensive R&D experience in fabrication technologies, process design and integration, facility operations and technology transfer. He is co-inventor on 14 patents and has co-authored 6 journal articles.*

Christian Krutzik *is a Senior Electrical Engineer at Irvine Sensors Corporation. He received his Masters Degree in Electrical and Computer Engineering from University of California, Irvine. Mr. Krutzik has over 18 years of experience involving electrical design and evaluation of 3D stacked modules and system miniaturization. Projects have included SSD design and development, compact 1TB USB SSD drives, NAND Flash characterization for secure erase products, FPGA development for LiDAR applications, miniaturized and wearable biotelemetry sensors, and acoustic processing/telemetry over RF using hearing-aid technology. Mr. Krutzik is currently involved with the design and development of a secure SSD product line and has experience with developing and debugging SATA, PCIe, and other high speed interfaces. Mr. Krutzik was also involved with a project to evaluate NAND flash devices for data remanence which included low level analysis of Flash devices and intricate knowledge of standard SSD processor capabilities. Further experience at Irvine Sensors includes stacked computer systems running Linux, various RF interfaces, high-density stacked memories, storage devices such as SSDs, FPGAs, embedded firmware, software, and DSP-based systems. Mr. Krutzik also has experience with multiple electrical simulation tools such as Hyperlynx, HSPICE, and other FEA tools. Software experience includes C, Python, Matlab, Javascript, PHP, HTML, assembly, and Labview on both Linux and Windows platforms.*

Daniel Nakamura *received his B.S. in engineering from Harvey Mudd College in 1987 and an M.S. in electrical engineering from the University of Southern California in 1993. From 1987 to 1996, he worked for Jet Propulsion Laboratory (JPL) as a design engineer for the Communication Systems group designing communication hardware. He left in 1996 to work for commercial companies designing IP telephony hardware, network processors, and server class network controllers. He returned to JPL in 2016 as a technical group supervisor for Flight Electronics and is fostering the next generation of flight hardware.*

Sung Kyu Lim *received the B.S., M.S., and Ph.D. degrees from UCLA in 1994, 1997, and 2000, respectively. He joined the School of Electrical and Computer Engineering, Georgia Institute of Technology in 2001, where he is currently Dan Fielder Endowed Chair Professor. His current research interests include modeling, architecture, and electronic design automation (EDA) for 3D ICs. His research on 3D IC reliability is featured as Research Highlight in the Communication of the ACM in 2014. Dr. Lim is a recipient of the National Science Foundation Faculty Early Career Development (CAREER) Award in 2006. He received the Best Paper Awards from the IEEE Asian Test Symposium (2012) and the IEEE International Interconnect Technology Conference (2014). He has been an Associate Editor of the IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems since 2013.*