Next Generation Memory Interfaces - Deserializer

Miron Veryanskiy
Kyle Dillon
Kalika Saxena
Sinan Liu
Chenyang Xu
Elad Alon
Vladimir Stojanovic

Electrical Engineering and Computer Sciences
University of California at Berkeley

Technical Report No. UCB/EECS-2015-111
http://www.eecs.berkeley.edu/Pubs/TechRpts/2015/EECS-2015-111.html

May 14, 2015

Acknowledgement

Thank you dearly Elad Alon and Vladimir Stojanovic. It has been nothing short of an absolutely incredible journey working on this capstone under both of your supervision.
I want to thank my team, Kyle Dillon, Sinan Liu, Kalika Saxena, Chenyang Xu for being right there in the ranks along my side.
I want to thank all of my friends and family who believed in my abilities, and encouraged me to pursue my ambitions. You saw in more in me than I can comfortably accept credit for.
Finally, I would like to thank UC Berkeley and its extraordinary community. The experiences and lessons that I learned during my time at Cal will forever be with me. I will carry these lessons with me far beyond

the boundaries of your gates.

University of California, Berkeley College of Engineering

MASTER OF ENGINEERING - SPRING 2015

Electrical Engineering & Computer Science

Integrated Circuits & Physical Electronics

NEXT GENERATION MEMORY INTERFACES - DESERIALIZER

MIRON VERYANSKIY

This **Masters Project Paper** fulfills the Master of Engineering degree requirement.

Approved by:

1. Capstone Project Advisor:

Signature: _____ Date _____

Print Name/Department: ELAD ALON / EECS

2. Faculty Committee Member #2:

Signature: _____ Date _____

Print Name/Department: VLADIMIR STOJANOVIC / EECS

Capstone Final Report

Project: Next Generation Memory Interfaces

Advisors: Elad Alon, Vladimir Stojanovic

Team Members: Kyle Dillon, Sinan Liu, Kalika Saxena, Miron Veryanskiy, Chenyang Xu

Report Author: Miron Veryanskiy

Abstract

This Capstone project aims to develop a novel memory controller to deliver a high-bandwidth interface for the DDR4 memory standard. DDR4 is the current cutting edge memory standard developed by JEDEC. The high-bandwidth interface is used as a communication link between a memory controller operating at 400MHz and a DDR4 SDRAM. Our team developed a physical interface that can transmit 3.2Gbps of data using only one transmission line. The design consisted of five major sub-modules: 8 to 1 Serializer, Transmitter, Receiver, 2 to 8 Deserializer, and Clock-Generating circuits. This paper discusses the design process, as well as the final results, of the completed 3.2Gbps 2 to 8 Deserializer module. The Deserializer discussed in this paper takes two data-line inputs operating at 1.6Gbps each, and deserializes them onto eight data-lines operating at 400Mbps each.

Capstone Final Report

Project: Next Generation Memory Interfaces

Advisors: Elad Alon, Vladimir Stojanovic

Team Members: Kyle Dillon, Sinan Liu, Kalika Saxena, Miron Veryanskiy, Chenyang Xu

Report Author: Miron Veryanskiy

Date: May 15, 2015

Table of Context

Section 1 – Project Context and Introduction

Our project, ‘Next Generation Memory Interfaces’ aims to develop a novel method for memory controllers to utilize a high-bandwidth interface under the latest memory standard i.e. DDR4 SDRAM (Double Data Rate 4th generation, Synchronous Dynamic Random Access Memory). The DDR4 standard allows for memory to be accessed at twice the data rate of its preceding standard, DDR3 while simultaneously reducing the total power consumption and increasing the memory density. As memory devices eternally seek to be faster, denser and extremely low power consuming systems, our interface will get us another step further in this quest.

In September 2012, JEDEC Solid State Technology Association, the organization that defines standards for the semiconductor industry, released the JESD79-4 for DDR4 SDRAM. With the release of this standard, companies such as Samsung, Micron Technology, Agilent Technologies and several others are developing SDRAMs compliant with this standard. The intent of our project is not to compete with the industry, but to explore the research and development opportunity presented by the new standard. To achieve the higher speed and low power requirements of DDR4, several changes are required at the architecture and circuit level. We aim to implement the strategies required to design a functional interface that meets the specifications of the DDR4 standard.

SDRAM is one of the commonly used types of memory in computing systems. It is a volatile memory that requires periodic refreshing to store the data. Owing to the speed and structural simplicity, DRAMs are often used as the main memory in personal computers and workstations. DDR is a class of memory that transfers data twice as fast as a SDR (Single Data Rate) memory since the transmission occurs on both positive and negative edge of the clock.

DDR4 is the fourth generation of DDR SDRAM which operates at a supply voltage of 1.2 V with data rates up to 3.2 Mbps – twice that of the preceding DDR3 standard.

Section 2 – Industry Market & Trends Analysis

In this section we will investigate the current semiconductor industry, what technologies exist and how our technology fits, and elaborate on the current competitive landscape of the market. We will also establish our possible clients, stakeholders, and our go-to-market strategy. Finally, we will evaluate the current social, technological, and economic trends and how these forces affect the industry.

Integrated circuits are an important sector in the semiconductor industry. The semiconductor industry is known to be highly competitive in nature, and the trend has been increasing over the years (Ulama 2014:19). Product life cycles are short, as more technologically advanced products replace older ones. Adoption of products is majorly affected by performance and reliability. The notable companies in the memory technology industry are Intel Corporation and Samsung Electronics with 7.6% and 5.3% of the global semiconductor and electronics parts manufacturing market, respectively (IBISWorld 2015: 27). Broadcom Corporation, Texas Instruments Inc., Advanced Micro Devices (AMD) Inc., Micron Technology Inc. are a few of the other major companies that compete in this industry (Ulama 2014:19). The major companies in this industry are all fairly large and well established, and compete over products and technologies. High demand for products and extremely low pricing intensify the competition in the industry. This poses a significant barrier to entry for new and smaller companies leading to

only several companies currently building DDR4 memory chips and controllers. In fact, out of the large integrated circuits companies listed above, Micron and Samsung are the only two that develop DRAM technology.

As we strive to make a smaller, faster, and more efficient memory interface, we have to compete with the products, research and development efforts of competing companies. Our competitive landscape does not merely include semiconductor companies, but also technologies that have similar features and functions when compared against our project. Existing memory technologies, such as DDR3, 3D stacked (3DS) - DDR3, and GDDR4, compete with DDR4 on various parameters such as cost, speed, and use-cases. While DDR4 is faster than previous memory generations, the higher cost of the new chip technology would make the cheaper DDR3 technology a strong competitor.

Emphasis is placed on the significant performance improvements that DDR4 presents over DDR3 technology. The following table shows a brief comparison of the key features between two technologies.

Table1. Comparison between DDR3 and DDR4 [1]:

	DDR3	DDR4
Power supply voltage	1.5V	1.2 V
Speed	1.6~2.1 Gbps	1.6~3.2 Gbps
Density	8GB(max)	16GB(max)
Price	$100 avg	$200 avg.

The first comparison is in regards to power efficiency, not only does DDR4 have a lower supply voltage, but it also implements a new algorithm to control its energy consumption by

entering its “standby” mode more frequently and precisely than DDR3. The improvements lead to better performance in both the power consumption, and operating temperature.

Furthermore, one of the most essential features, memory speed, has been improved significantly in DDR4. The analogy between the memory speed and highway traffic speed is very descriptive. The speed of the memory is the amount of data that can be transferred in a certain period of time. There are two factors determining the speed, which are interface lane width (the number of parallel wires on the interface) and the frequency of the memory’s operation. Considering the analogy, the bandwidth is the quantity of lanes on a highway, and frequency is the travel speed of its vehicles. Within a fixed time period, having more lines and a faster speeds will allow for more vehicles to travel. Similarly, having an improved working frequency, along with an enlarged bandwidth, DDR4 achieves a data transmission speed that is approximately 1.5 times faster than DDR3, as Table 1 indicates. The increase of the speed is benefited from the revolutionary bank-group management technology.

Another differentiating factor is the density, or the physical space of a single memory chip. Advancements in the chip’s encapsulation provide DDR4 with a 50% density increase, with regard to maximum space. With a larger storage space, DDR4 is able to process more information simultaneously. However, similar to every emerging new technology, the current price of DDR4 memory is 30% to 50% more expensive than DDR3, which can achieve similar functionality at lower speeds. With a large-scale adaptation for DDR4 memory, and hardware compatibility of its peripheral devices, the manufacture price would quickly become more affordable in the future.

GDDR3 and GDDR5, which stand for Graphics Double Data Rate 3 and 5 respectively, are a kind of memory specifically designed for processing graphics. Despite the similarity in

terms of the name, the graphic memory is named one generation ahead regular memory. This means that the core technology of GDDR3 is essentially an upgraded version based on DDR2 technology, rather than DDR3. The graphic memory is designed to have lower energy consumption, and an optimized performance when dealing with graphical-data processing. Since the application area of these two kinds memory differentiates amongst each other, they do not compete directly. The Graphic DDR is typically developed based on the previous generation of DDR memory technology, with improvements on speed and application-specific functional modifications.

The main markets for our project include traditional memory devices and consumer electronics, which are currently in a prosperous state. The global-scope next generation memory technologies market was worth $207.8 million in 2012, and is projected to be worth $2,837.0 million by 2019, growing at a 46.1% average growth rate from 2013 to 2019 (Transparency Market Research, 2014). The report divides the overall market for next generation memory technologies on the basis of certain parameters, namely, interface type, application, and geography. On the basis of interface, the market for next generation memory technologies can be categorized into SATA, SAS, DDR, and PCIe and I2C (Transparency Market Research, 2014). The main applications of next generation memory technologies include embedded MCU and smart card, mobile phones, mass storage, cache memory, enterprise storage, and automotive.

Geographically, the global next generation memory technologies' markets can be divided into four major regions - North America, Europe, Asia-Pacific, and the rest of the world. This industry is always looking for ways to decrease power consumption, increase density, and develop clever architectures that promote efficient and faster computing. The new generation

memory technologies market has gained significant momentum in recent years due to growing demand for faster, highly scalable, and cost-effective memory solutions.

Appreciation for the necessity of our effort follows from understanding the dynamics of the industry which our product seeks to enter. After understanding the landscape within which we stand, we have good reason to believe that our project is valuable to our stakeholders. We reason that our stakeholders should be more interested in receiving a completed deliverable from us over any other, equally qualified, external competitor.

Our first differentiating quality is that we offer to provide “non-contracted” work. Contracted work is any work commissioned by one party to be executed by another party. To begin such work, both parties must agree on the terms defined within the agreement document prior to the work’s commencement. The agreement is realized through means of a binding contract that both parties agree to enact. Once the contract is created, it typically cannot be altered or modified, unless the consent of all parties is evident. This could place the requesting party into a stiff situation if it discovers that its priorities have changed mid-way through a contract.

Upon the project’s completion, the completed work is commonly handed off “as-is.” This means that no additional support is to be provided in the future (unless explicitly negotiated upon within the original contract). Any additional requested support or modification requires for a new contract to be written. Not only is this financially inconvenient, but it can also be logistically inconvenient for the recipient. Without support, the deliverable is handed off with a decreased utility. The recipient of the deliverable is stuck with using the deliverable solely within its original scope.

Our stakeholder, Berkeley Wireless Research Center (BWRC), benefits from ownership over the development process. A common clause added to most contractual work instills a limit on interim design modification requests. This clause exists to prevent the requesting party from overexerting the contracted party without compensation. Internal control over the developmental process allows for precise design-source malleability during development, and full exposure of the design files. Design-source malleability allows for the BWRC professors to more closely guide our direction through the project's development. It allows for them to change the path that we follow if new interests arise. There is no contractual overhead to worry about in this scenario.

BWRC benefits from retaining access, and owning, the source code and designs. The design-source exposure enables BWRC to question every aspect of the implementation until they understand it completely. With contracted work, this information is typically unavailable to the requesting party due to trade secrets being used in a design. Owning the source enables BWRC to have permanent design-source access. Long-term source access enables cost-effective and effort-effective technology adaptation into any future BWRC projects. Along with adaptation, owning the source creates the opportunity for growing in-house expertise at BWRC through education.

The benefits mentioned above align very closely with our stakeholder's interests. The stakeholder, being BWRC, is interested in three main attributes from the project. First, BWRC wants a fully customizable deliverable due to unpredictable future demands. Second, BWRC wants the freedom to optimize the design for unique implementations that would require the modification of the source on a per-use basis. Third, BWRC wants to avoid the financial, temporal, and contractual overheads associated with third-party work. Our project delivers on all

three attributes. By choosing to complete this project through our team, rather than a team of contractors, BWRC satisfies its internal interests.

Our team anticipates BWRC's decision to work with us as opposed to larger suppliers. The current semiconductor marketplace is saturated with both customers and suppliers. As Ulama describes (Ulama 2014:28), "Established operators in this industry have been able to develop solid relationships with customers, and it can be extremely difficult for new companies to gain contracts with customers when existing semiconductor manufacturing operators have built reputations over a long period." To exemplify the significance and the weight carried by the previous statement, note that the Semiconductor and Circuit Manufacturing industry is one of the largest exporting industries in the United States (Ulama 2014:5). It indirectly provides jobs to 250,000 Americans, is currently valued at $79.5 billion, and has grown at an annual rate of 4.8% (Ulama 2014:5).

The current players, both customers and producers, are very well established, and very tightly connected. Penetrating into the customer base that the massive producers currently support is near impossible for a small team like ours due to lack of reputation. Aside from penetrating, the customers in this segment of the market are a significantly strong force due to two reasons: 1. The intrinsic competitiveness of the current suppliers, and 2. "The electronics marketplace is continually under pressure to improve product functionality, decrease size, increase speed, and decrease cost." (IBISWorld Global Semiconductor & Electronic Parts 2015:33)

Our team has set our target in a completely different direction. Instead of focusing on the massive customers, who are already served very competitively, we direct our focus at an interestingly under-served segment in this market space. In part, our choice of direction is due to

the methods through which our Capstone project was decided upon. The decision process confined the scope of the project to target academic goals and provide solutions for academic institutions. Thus, our customer space currently only encompasses the Berkeley Wireless Research Center, but is functionally able to serve any academic or small-scale organizations.

As we currently stand, with one effective customer in our sights, we are subjecting ourselves to a very strong customer market force. This is an undesirable outcome due to the limited size of the space, which we choose to enter, but success in this space will send positive signals at other research institutions. We would be able to expand to encompass more academic institutions because they would prefer to acquire the product through us. Our effective results are comparable to their current methods of operation, but with the benefit of reduced fixed-cost expenditures – which arise when placing orders with large design and manufacture firms.

A majority of the market belongs to well-established companies. These include Micron (IBISWorld 2015: 27) and Texas Instruments (IBISWorld 2015: 30). The limiting factors that lead to this situation include “access to latest technology and intellectual property, the level of investment…, access to skilled employees, and the dominance of existing players” (IBISWorld 2015: 25). In the memory industry, the companies compete over a very specific set of criteria including price, performance, features and power consumption, all of which are highly measurable and quantifiable metrics (IBISWorld 2015: 24).

If the dimensions of competition between companies in a given industry converge, then the companies are left to compete solely on price (Porter 2008: 12). In the integrated circuit market, the industry has converged heavily on these metrics of performance, features and power consumption, which has resulted in fierce price competition. Because “economies of scale can be significant in this industry” (IBISWorld 2015: 25), new entrants must manufacture large volumes

to stand a chance against the bigger companies. This requires up-front capital that many smaller new entrants do not have available. Entering the market attempting to compete on these highly competitive dimensions would result in “zero sum competition” (Porter 2008: 13), and would not be a viable business strategy.

When instead of converging on the same dimensions, companies target different segments of the customer base, the result can be “positive sum” competition: competition that increases the profitability of all companies (Porter 2008: 13). We plan to employ this strategy with our DDR4 memory interface as the needs of a research institution like BWRC are different from the typical semiconductor customer. BWRC focuses on the novelty of the technology, not on the economies of scale. They require the interface to be designed custom to the application of interest and not the general industry standard.

Although the memory technology industry is highly competitive, growing, and difficult to penetrate, the market is growing fast due to this a demand for consumer electronics, an industry which is expected to grow 5.3% annually to nearly $300 billion dollars by 2019 (IBISWorld 2014: 4). This high demand and new market bring some space for new companies to enter and grow. These new entrants usually emerge during the transition between the technological revolutions and each one has its own specialty.

From 2002 to 2013, the DDR memory industry has undergone 4 significant technological transitions, all of which are aiming at improving in three performance aspects and achieving a denser data processing capability. Products with “high levels of performance, reliability, quality, and low levels of power consumption” (Bach,2014:6) can gain an apparent advance in the competition of memory design industry. Being the three largest manufactures of memory chip and developer of DDR memory technology, Samsung, Crucial (Micron), and Hynix have already

invested millions of dollars in their R&D sector to develop the new generation DDR4 memory interface in order to reinforce their dominant market share.

Given such a giant market, other major memory designers, such as Kingston, keep ramping up their pace to meet the memory controller design requirements for the recent DDR3 to DDR4 transition. Besides the companies who are already in the market, there are significant number of new entrants, trying to seize this opportunity. “The latest Census data indicates that 64.1% of operators in this industry have fewer than 20 employees” (Ulama 2014:25). The development strategies of new entrants are highly focused on niche features. New entrants follow a model of “specializing in a small number of product lines to serve niche markets” (Ulama 2014:25) in order to avoid a direct competition with large companies.

Table 2. Representative new entrants in DDR4 memory development

Company Name	Specialized market/feature
Century Micro INC.	Small physical size & low energy consumption
Montage Technology	Fast operating speed & low energy consumption
G.SKILL	Enhanced gaming performance

Three unique, representative companies are provided to conduct the analysis of the new entrant. The table 1 above shows a brief comparison of three distinguished new entrants’ key product features. It indicates each new entrant is trying to gain its market share by specializing its product from the three technical aspects mentioned in the previous paragraph.

The Japanese based company Century has just halved the physical size of DDR4 memory in their most recent product at the year of 2014. The China-based Montage Tech is more focusing on developing fast speed and lower power rate DDR4 memory for large scale server

use. “Less power draw means less heat and longer battery life”, which indicates “the servers are expected to be the biggest beneficiaries of the jump to DDR4” (Andy, 2014:6). Meanwhile, G.SKILL put majority of its resources into developing DDR4 memory controller with improved gaming performance. These companies are increasingly securing their niche markets by making breakthroughs in design of the memory controllers and interfaces, even while the major developers are still dominating the memory chip manufacturing area.

Big companies enjoy economies of scale, making it difficult to compete with them in manufacturing the integrated circuits (ICs). Based on the analysis of the new entrants, in order to build immunity for our design, we plan to segment the market to research institutes like BWRC. Their needs are different from most, and provide an opportunity for us to develop a product that satisfies these needs better than the competition. Since the design of our project is specifically for BWRC internal research use, there will be no direct competition and obvious threat from these new entrants either.

The threat from other technologies is weaker, as our DDR4 interface is more advanced than existing DDR3/GDDR5 interfaces. Therefore, we focus on developing the intellectual property and targeting the specific needs of the academic communities. This specific category of consumers require more customizable, and open, circuit designs at a lower volume – a need that is unmet by the larger companies that package their circuits in black boxes, manufacture in high volume, and allow little to no customization. By segmenting the market based on unmet needs, and our abilities to satisfy them, we hope to entrench our position as a profitable part of the semiconductor industry.

From the perspective of semiconductor circuit design, it is a complicated process to design a controller and its interface, and integrate it with the memory chip. Therefore, our

technology suppliers include both software side and hardware suppliers. Software suppliers are those who provide coding languages, design platforms, and simulation tools. Hardware suppliers are those who provide electrical specifications, datasheets, and fabrication characteristics relating to memory chips.

Software suppliers mainly provide programming language support. Verilog is the main hardware description language (HDL) that we are using to model digital electronic systems. Cadence, a company that provides electronic design automation (EDA) software, covers many language design platforms, including Verilog, and provides EDA tools for full custom design of integrated circuits. As an all-in-one suite, Cadence is our main software supplier.

Hardware suppliers provide descriptive information about the memory chip technology. Each generation of memory chips has new fabrication breakthrough. Thus, during our controller interface design, the latest information about memory chips is critical, such as voltage supply of the chips and the memory bank structure. Our hardware suppliers, such as Micron Technology, Intel Corp., and Samsung, are big semiconductor companies in this industry. In Semiconductor & Circuit Manufacturing in the US Industry Report, Intel Corp. and Samsung have 18% and 13.8% market share in 2014 (Ulama 2014:4). Although they seem like our competitors from the sales end, they also have the best research departments and technical experts in the chip fabrication domain. Samsung competes in the Semiconductor and Circuit Manufacturing industry via its fabrication and research and development facilities in the United States (Ulama 2014:4). They release academic papers and datasheet of their latest research results about DDR4 memory chip. In accordance with the information provided by these large semiconductor-manufacturing companies, we define the interface and design our memory interface.

Powerful suppliers capture more of the value for themselves by charging higher prices, limiting quality or services, or shifting costs to industry participants. As mentioned above, Intel Corp. and Samsung are both suppliers and competitors for us. If they limit our access to their latest technology about DDR4 memory chip, it will be hard for us to compete with them. However, the good news is that the DDR4 memory specification is becoming a standard, so we will be less dependent on them.

There are certain aspects that we can focus on to succeed in this capital-intensive, and research-intensive, memory design industry. New companies are trying to explore the market by boosting their expertise in faster-speed designs, smaller dimension layouts, and highly customized application-specific designs. With increasing maturity of the DDR4 technology, the competition is becoming fiercer. This increased competition will largely benefit the semiconductor industry's evolution speed, as well as provide customers with cheaper and higher efficiency devices. Our project will not only encourage further development from competing companies and research groups, but also benefit BWRC's exploration of the utilization of DDR4's capabilities.

Section 3 – Intellectual Property Strategy

The PHY interface provides us a good scope for creating a patentable Intellectual Property (IP). The physical layer has been split into 5 major parts - Serializer, Transmitter, Receiver, Deserializer and Timing Circuits. Each of these allow for novel implementations and innovations in circuit design. As we are working at the cutting edge of technology, we would have to adopt ingenious techniques to meet the specifications for high data rates of DDR4. One or more of these

implementations can provide us patentable IP. This paper will discusses why this technology may be patentable, the advantages and disadvantages of seeking a patent, the current state of the semiconductor IP space, and the risks associated with not seeking a patent.

In the context of IP, creative designs and creative solutions fall cleanly under the category of patentable assets. In essence, the purpose for securing IP is to declare discernible ownership over a design or utility (USPTO, 2013). As an independent entity, we can draw benefits from securing patents and owning patents. The benefits we pose to secure range from monetary compensation to strategic industrial presence.

From a monetary perspective, owning patents allows our team to claim ownership to a recognizable asset. After incorporating our team as a legal entity, a patent opens us to the opportunity of being acquired. The proceeds from an acquisition could be used to finance additional ventures, which our team currently does not have the financial freedom to pursue.

A secondary monetization strategy that patent ownership affords us, is the option to license our technology to independent entities who wish to avoid committing R&D expenses for the purpose of developing said technology independently. Aside from the legal expense that we would need to undertake, the licensing option is financially robust.

The third and final benefit is an unquantifiable benefit. The third benefit arises from establishing a reputation as an entity. Acquiring a patent will demonstrate that we, as a team, know how to drive concepts into patentable ideas, and patentable ideas into awarded patents. Successfully acquiring a patent will demonstrate to that we are capable as a team, and will instill external confidence into our capabilities. This reputation will position us to open new leads amongst skeptical and risk averse customers.

The disadvantage of applying patent is obvious: it burns money. Filing a patent is not as simple as people imagine. Normally attorney fee becomes a big piece of the cost. Determined by the type of invention, the attorney fees are range from $5000 to more than $15,000(Quinn, 2011). Adding the government filing fee and all kinds of application fees, the total cost of preparing and filing a patent may exceed ten or twenty thousand dollars. In addition, the maintenance fees would be another big part of the cost. Depends on how many years the owner wants to keep, the maintenance fees float from $490 for small entities and $980 for large entities due at 3.5 years to $2055 for small entities and $4110 for large entities due at 11.5 years (Stim, 2012).

Considering that this IP would be used only for research or instructional purposes, it would be directly non-profitable. Applying for a patent would subject the applicant to financial burdens. It would not be valuable for individuals at BWRC to apply for a patent with this IP. However, if the IP owner switches from a group of individuals to the University, the decision would be different. The university has the budget, and interest, to cover the expenses. The patent would add to the university's reputation, which is far more important than profit due to it existing in the public domain. It would be valuable to apply a patent for this IP if the applicant is a university. (Note, BWRC happens to exists entirely in the public domain and does not patent its work).

Unfortunately, the semiconductor IP market can be difficult for smaller entrants like us. The rate of patent enforcement by larger corporations has not increased over the past few decades (Hall 2007, 5). However, in attempts to increase market share and presence, they have increased the number of patents they file. In the 1980's, the median number of patents filed by an employee was less than one, whereas during the turn of the century it was near eight (Hall 2007, 10). While larger corporations have a broad and ever expanding portfolio, smaller firms focus on particular market segments in attempts to perfect and own this portion of the total revenue stream.

Unfortunately for these smaller firms, this means that if and when larger corporations expand into their territory, they have no choice to defend what little they have. It is for this reason that smaller firms tend to more be more aggressive in enforcing their patents (Hall 2007, 3). Thus, it can be expected that we would have to actively enforce our patent. If our patent (or patents) focused solely on DDR4 memory control and interfacing, then we would have no choice but to defend the few eggs in our basket.

The risks associated with not patenting the design are significant. Since the integrated circuit design is based on following certain physical requirement and universal specifications, hundreds of similar design and product can be invented in the short time of period based on a same standard. In DDR4 memory design particularly, JEDEC standard is the critical specifications that everyone need to comply. There is high possibility that other individuals or companies will come up with very similar or even the same design. As Gene indicated in his article, engineers who are working on solving a certain problem "are likely to find solutions that are similar" (Gene 2009:8). If a similar design is first patented by other entities, the potential financial loss is irreparable and a great amount design effort would be wasted. Furthermore, without patent the design appropriately, competitors and free-riders can easily take advantage of the design or embedded our inventions into their products without any recognition of our work and having any consequence. Besides these two factors, without right patenting, it is almost impossible to conduct technology transferring or licensing. And this would greatly impede the process of commercialization of the invention or designs.

Therefore, there are a great number of critical risks involving in not patenting the design. Our memory controller interface design should be patented when its major functions and specification are met. The management of the patent can be done when creating a patent portfolio.

By using management software, or having regular reviews, updates, recategorizations, and rebalancing of the patent portfolio, quality management could be achieved.

Trade secret is one type of intellectual property with unlimited time of protection. It can be a method to protect our technology, but it is not optimal. The DDR technology evolves every three years averagely, an unlimited protection time is unnecessary. Giving the fact that circuit design industry is highly standardized, and reverse engineering methods are quite mature, it would be difficult to protect the design with under solely a trade secret. Not patenting and relying solely on trade secret subjects us to potential commercial espionage and high cost of protection. Moreover, the trade secret cannot prevent the similar, or identical, products from being fabricated. Due the nature of the circuit design industry, trade secret won't play a significant role in limiting other similar designs. Patents exist to "protects your rights regardless of what anyone subsequently develops" (Shane 2007:8). Therefore, original technological inventions such as circuit designs would best be protected under patents.

Ultimately, deciding whether to seek or not to seek a patent for our design depends on the novelty of the final product. If we discover and implement a new physical layer architecture that provides performance, costs, and/or feature improvements over the competition, then the patent's value overcomes the cost associated with filing it. If the final outcome is unique, but provides only marginal benefits compared to the competition, then there will be no benefit in filing the patent.

Section 4 – Technical Contributions

Section 4.1 – Project Overview and Context

Our capstone team has worked to create a Physical Interface Layer (PHY) between a double data rate, fourth generation, synchronous dynamic random-access memory controller (DDR4 SDRAM controller) and the DDR4 SDRAM module that it controls. Due to the mixed-signal nature of a PHY, the task of creating the PHY was modularized into smaller actionable modules. The modularization is based on the major components that need to be structured as individual component blocks. We modularized the functional PHY model into five modules: A Digital Serializer block, an Analog Transmitter Block, an Analog Receiver Block, a Digital Deserializer Block, and a Phase Shifting Clock Generating block. We distributed the modularized blocks amongst group members, and thus established the division of labor. Even with the distributed division of labor, the design of individual components is not a distinctly individual role. The modules themselves have interface specifications that they must adhere to, or suffer the consequence of not synergizing successfully.

My explicit contribution to the DDR4 PHY is its Deserializer module. The Deserializer's purpose is to convert a single high-speed data transmitting channel into multiple lower-speed data transmitting channels. The necessity for a Deserializer arises when multiple transmission signal lines are encoded onto a single transmission line operating at a faster bitstream transmission rate. The faster bitstream utilizes fewer physical metal lines in order to improve bandwidth density. Serializing data onto one channel comes at the cost of higher data clock rates, but provides the benefit of fewer fabricated interconnects. My synthesized Deserializer allows the DDR4 SDRAM controller to "decode" the originally encoded transmission signals. The necessity for decoding the transmission signals arises from having to meet the controller's input

interface specifications. The controller has a predefined operating frequency and a predefined quantity of input channels. Our PHY is designed for a controller that will operate at 400MHz (400 Mbps) and require eight parallel data lines. The data is deserialized from one channel operating at 3.2Gbps to eight channels operating at 400Mbps in order to fit the controller's expected input interface. After the deserialization, the data stream will be in compliance with the controller's intrinsic architecture.

Section 4.2 – Knowledge Domain Review

The first area of interest that our team explored was the overall technological foundation behind the DDR class of SDRAMs. Original SDRAM is based on non-synchronous dynamic random access memoery (DRAM) technology. The DRAM is a low-cost memory storage technology that utilizes a transistor and a capacitor to store a single bit of information. The term "dynamic" signifies the use of a capacitor that does not permanently store charge, but rather stores charge that leaks out parasitically over a period of time. Due to the leakage, DRAM cells need to have their data refreshed periodically in order to avoid data corruption. The alternative to DRAM is Static Random Access Memory (SRAM) which utilizes bistable latching circuitry to keep data valid without refreshing, but it is more complex and more expensive to make. In the context of commercially available memory modules, the standard is to use Synchronous Dynamic Random Access Memory (SDRAM) modules. The difference between a DRAM module and a SDRAM module is that the SDRAM modules are synchronized with the system bus.

Data is retrieved from DRAM and SDRAM cells destructively – this means that the data itself is corrupted during retrieval, and needs to be re-written after being read. Data cells within

SDRAM chips are structurally abstracted into logical structures referred to as Memory Banks, Bank Groups, Memory Ranks, and Memory Pages.

Memory Banks are an assemblage of storage units that are organized in rows and columns. The quantity of bits in a typical unit is 8, which leads each unit to contain exactly 1 byte of information. Accessing the data cells occurs by a request containing the coordinate of the row and column. Due to the functional structure of the SDRAM, any data access "opens" a full row across all columns within a SDRAM Bank. The act of "opening" simply means that the data is read from that row of units and will need to be re-written later in order to preserve the read data's integrity.

Memory Ranks are a higher abstraction level that encompasses multiple DRAM chips. A group of DRAM chips form a memory rank when they are all enabled by one chip-select signal value. This allows multiple DRAM chips to be accessed simultaneously using only one bank & row & column tuple.

Combining the concepts of multiple memory banks being opened across the DRAM chips within one memory rank leads the definition of a "Memory Page." A Memory Page is

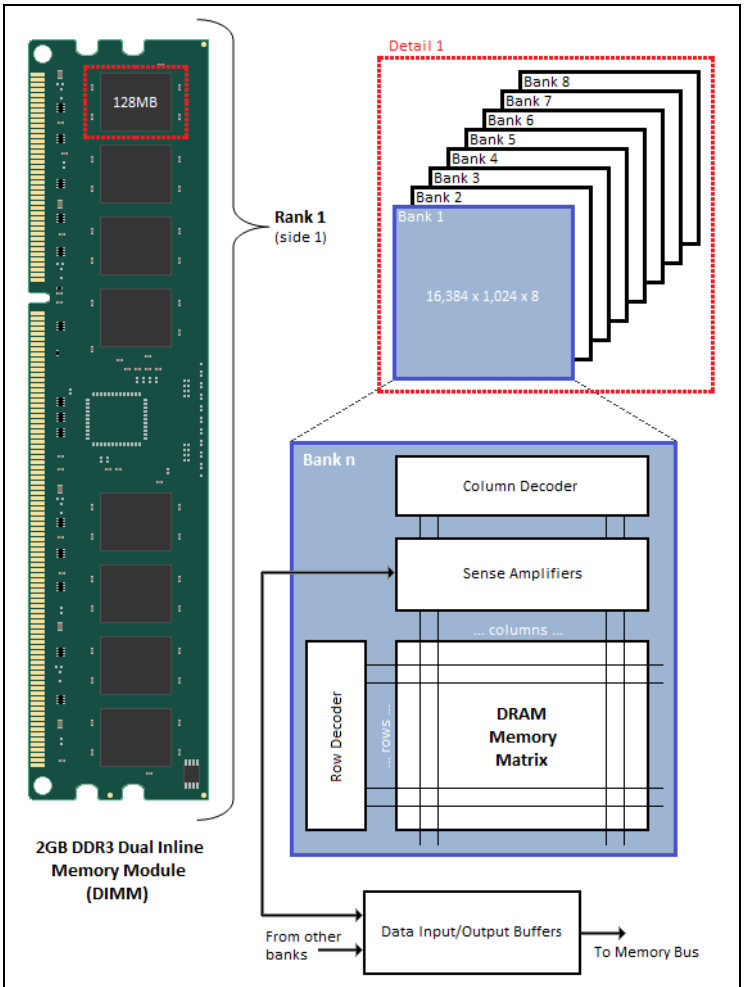


Figure 1 – DRAM Logical Abstractions (2010 Gill)

one row opened within a given bank index across the whole memory rank. This is the full row of data that becomes accessible when a bank & row & column request is sent by a data controller. The quantity of bits in a full page is referred to as the prefetch, or the total number of bits that are placed on a memory-internal Input/Output (I/O) buffer every clock cycle. See Figure 1 for a visual reference of the described abstractions.

An abstraction called “Bank Group” is introduced in DDR4 memory. It creates discrete groupings of banks within one rank. Each bank group will produce one page of data at a time. Requesting data from alternating bank groups allows for quicker performance due to a clever delay avoidance. During the time that the first bank group is closing its page by writing all of the data back to the DRAM cells, the second bank group has time to prepare its page and place it on the I/O buffer. This allows the delays associated with each open operation to be functionally hidden from the perspective of the controller’s request (as long as requests query alternating bank groups). By hiding the delays using this clever alternate-access approach, the controller can request data from the rank far more rapidly when compared against DDR3.

With a general description of the abstractions around a typical DRAM module in place, the concept of Double Data Rate (DDR) can be explored. As the name suggests, DDR SDRAM allows for the controller to retrieve twice as much data within a fixed amount of time. An interesting constraint to keep in mind is the limited frequency at which an SDRAM module can operate internally. The SDRAM module is limited by the fundamental delay associated with reading and writing to one DRAM cell. Thus, the SDRAM module has an upper limit to its operating frequency. DDR boasts dual clock-edge data access – or twice as much data per clock period. It accomplishes this by doubling the length of the prefetch. During the rising clock edge

the first half of the prefetch is received by the controller, and during the falling clock edge the second half of the prefetch is received by the controller. Although the controller experiences a doubled data rate, the cell-access granularity is not increased. What this implies is that the data stored within one full prefetch should all be desired data, if double data rate is to be realized. The lack of granularity implies another limitation – accessing individual bytes of information from SDRAM memory is a temporally irresponsible operation.

Since the advent of DDR, successive generations were released as DDR2, DDR3, and DDR4. Each successive generation boasts a data rate twice that of the generation before. Behind the generationally improving data rate claims, the minimum DRAM access delay is still in effect. With the DRAM cell-access frequency limited by the intrinsic DRAM access delay, the data rate claims are realized solely through an increased prefetch length, as seen with DDR, DDR2, and DDR3. DDR4 on the other hand has the same prefetch length as DDR3, but it employs the previously mentioned bank group abstraction which allows for the doubled data rate over DDR3. In DDR memory, the controller-access clock speed is coupled to the size of the prefetch. As the memory's claimed access speed increases, the size of the prefetch increases (except in DDR4 where bank groups are utilized). One theme that remains consistent throughout the generational improvements of the DDR memory releases is that the memory's claim to increased speed is not a breakthrough in technology, but rather a more comprehensive approach to data-cell access.

For a large portion of our capstone's work, our team focused on the DDR4 JEDEC Specification that outlined the memory module's characteristics. Our original purpose was to develop a memory controller for DDR4 SDRAM modules. To better understand the functional capabilities that a controller needs to possess, I pursued the goal of exploring DDR4 modes of operation.

Section 4.2.1 – DDR4 Modes of Operation

I will use this portion of my report to delve into the details pertaining to the various Modes of Operation outlined in the DDR4 JEDEC Specification.

The modes discussed below are enabled using Mode Registers residing on the DDR4 SDRAM units. The Mode Registers' values are defined when the controller boots up the DDR4 SDRAM. There are seven total registers, each register 8 bits in size. A given mode can be defined by 1 or more bits anywhere within the 7 mode registers. To enter Mode Register Programming mode, the MRS command is sent from the controller to the SDRAM unit.

The first mode of interest is the Test Mode. This mode is typically reserved for the SDRAM manufacturers for the sole purpose of testing the produced unit after its fabrication. This mode is enabled in Mode Register 0 (MR0) on bit 7 (MR0[7]). Enabling this bit in the mode registers will place the SDRAM into test mode. The end-user of the SDRAM unit will typically never interface with this mode.

Target Row Refresh Mode is an interesting mode that addresses some of the parasitic flaws of the DRAM architecture. There is a concept within SDRAM access referred to as "Row Hammering." Row Hammering occurs when any give row is repeatedly opened by the controller. Although this does not immediately raise any red flags, it does have an effect on the neighboring rows. Due to capacitive coupling between adjacent rows, heavy access on one row can degrade the data integrity in the neighboring rows. Mitigating this phenomena involves adding two counters - the Maximum Activate Count (MAC) and the Maximum Activate Window (MAW). The MAC keeps tracks of the activations of neighboring aggressive rows. The MAW keeps track of a given row's activation. If either of these counters surpass a minimum threshold value

(defined by the manufacturer depending on their technology) an explicit refresh is performed on the suffering rows.

Memory refreshes occur periodically over the whole memory rank regardless of the row-hammering effect. Due to the large page sizes in high-density memory designs, the refresh period itself (reading the data and writing it back) can impose significant delays on the memory's operation. The delays arise from the stall (delay) that the memory undergoes while a row is being refreshed. This stall is referred to as "Refresh Lockout." To mitigate the length of the introduced Refresh Lockout by the full-page refresh, the refresh can only target half of a page on one clock cycle, perform a data access on another cycle, and perform the rest of the refresh on another cycle. The mode in the DDR4 Specification that allows for this "smart" refresh is called the Fine Granularity Refresh Mode. It is enabled in MR3[8:6]. Fine granularity Refresh Mode reduces refresh cycle time (tRFC) as well as the average refresh interval (tREFI). During the Fine Granularity Refresh, the whole rank is still locked, but for a shorter period of continuous time.

For the purpose of saving energy, DDR4 introduced a "Gear-Down" Mode. The purpose of this mode is to power down non-critical parts of the SDRAM while the SDRAM is either in the initialization sequence or the self-refresh entrance sequence. The mode is exited only after the self-refresh exit. The mode lowers the Command and Address clock rate while increasing the DQ rate.

Errors in the DRAM structure can occur after the DRAM is shipped and implemented. The JEDEC specification anticipated this and introduced a mitigation mode known as Post Package Repair Mode (PPRM). This mode is optional for implementation, but very useful when incredibly low error tolerances are desired. The PPRM allows to repair one complete row in a bank-group. Figure 2 shows a graphical representation of this process as well as the distribution

of repairable fault occurrences. The process is irrevocable after it is completed. The process can be executed during runtime without the necessity to power-cycle the SDRAM module.

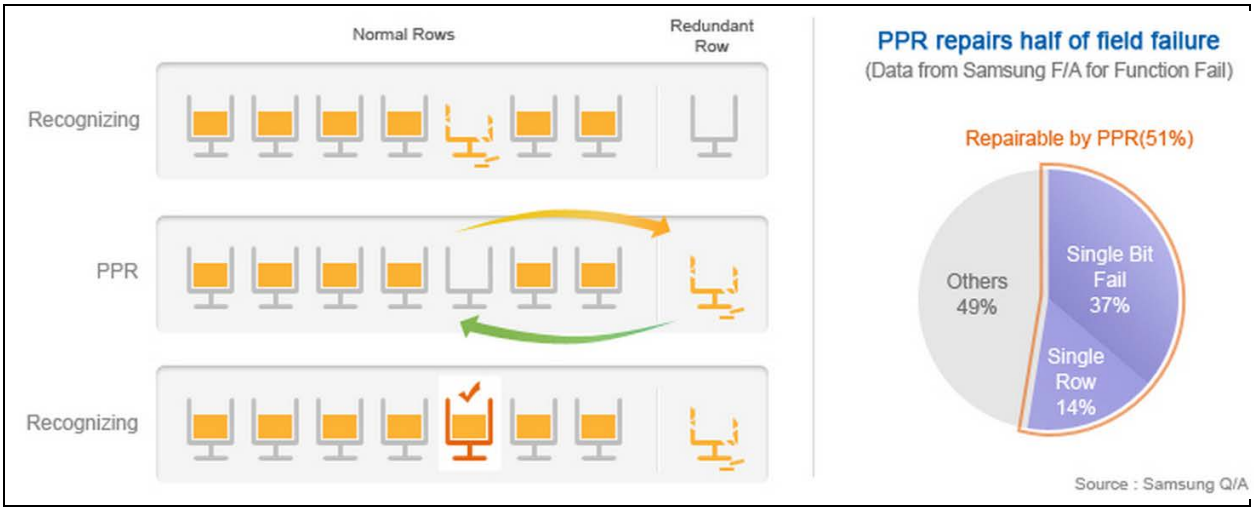


Figure 2 – PPR Visualization (2014 Samsung)

Another solution to mitigating DRAM cell flaws is called Soft Post Package Repair Mode (SPPRM). This mode is also optional to implement, but just as useful as the PPRM. It provides the option for temporary repair a complete row in a bank group. The operation is quicker than the PPRM, but it is not permanent. It can be reversed, and it gets reset when the DRAM power cycles. It can be made permanent via PPRM.

One of the main spec design focuses that JEDEC placed on DDR4 is the requirement for energy efficiency. In the same spirit as Gear-Down Mode, two additional modes were created to facilitate this goal. First, the Maximum Power Savings Mode (MPS Mode), and secondly the Per DRAM Addressability Mode (PDA Mode). MPS Mode is the lowest power mode that the SDRAM can enter. No data is retained and most external commands are ignored. It is essentially a “Hibernation” mode that is used to park the DRAM at low power. Recovery from MPS Mode is achieved by a call from the PDA Mode. PDA Mode allows for the controller to selectively

direct instructions at a single, specific, SDRAM on a rank. This mode allows for customized On Device Termination (ODT) as well as a customized reference voltage (V_{REF}) for every SDRAM.

Another method for minimizing power consumption is artificially throttling the speed at which the SDRAM stores and reads information. This is accomplished by using a lower clock rate and waiting (relatively) long periods of times between data commands. At low speeds such as these, the phase differences due to Process, Voltage, and Timing (PVT) variations are insignificant. Being that the phase differences are insignificant, the Delay Lock Loop (DLL), which helps control the phase of the signals, becomes irrelevant. Turning off the DLL altogether allows for additional power savings while operating at lower frequencies.

The transfer characteristics of transistors can vary with the environmental conditions within which they operate. A Temperature Controlled Refresh Mode (TCR Mode) was created in the specification to account explicitly for the temperature's effect on the delay. The safe route is to assume long delays, but this would sacrifice component performance during good environmental conditions. The TCR Mode allows the device to skip the refresh command when the environmental conditions allows for it to be skipped. This reduces the quantity of Refresh Lockouts. The external refresh rate MUST remain correct, even with TCR enabled. Only the automatic internal refresh rate is affected.

Low-Power Auto Self Refresh Mode (LPASR Mode) is the automatic refresh mode that sets the refresh frequency depending on the device's ambient temperature. The alternative to LPASR Mode is the Manual Self Refresh Mode (MSR Mode) where the refresh frequency is permanently decided by the controller. It places the burden of tracking temperature conditions onto the controller.

The Power-Down mode is the mode that the SDRAM can enter when memory is not being used. Command, Address, and Data Receivers are disabled. Data Drivers, and some Generators are disabled. This mode is enacted by driving the clock enable signal (CKE) low and holding it low for the duration of the power-down. Exit from the Power-Down mode occurs when the CKE is brought back high. The delay (tXP) must pass before the SDRAM is ready to receive its next command.

Another Power-Down mode is ODT (On-Device Termination) Input Buffer Disable Mode (ODTIBD Mode). When this mode is enacted, this prevents the device from providing termination during power-down (For power saving purposes). The ODT Mode, when enabled, can be in two modes – Synchronous and Asynchronous. The synchronous mode is enacted whenever the DLL is on and locked in phase. The asynchronous ODT mode is used when DLL is off and not used.

Section 4.2.2 – Knowledge Domain Review of the Deserializer

Upon completion of the first semester, our team pivoted and recalibrated our focus onto the physical interface link that resides between a memory controller and a DDR4 SDRAM module. In collaboration with the Capstone thesis papers written by other members of my Capstone Team on the Serializer (Dillon 2015), Transmitter (Xu 2015), Receiver (Saxena 2015), Timing (Liu 2015), my focus on the Deserializer will complete the full PHY interface's design.

A Deserializer is one of the two portions of a unit referred to as a Serializer-Deserializer (SerDes). SerDes are typically used for high-speed communications where the quantity of physically-available pin space is finite and limited. When a design has inputs and outputs on the order of thousands, providing a single macro-sized pin for each I/O becomes impractical. The required pin densities for reasonably sized chips are not feasible (Stauffer et al. 2008). What a SerDes combination accomplishes is squeezing multiple parallel data streams into a single serialized data stream. Figure 3 demonstrates SerDes units converting parallel data into serial data on a transmission channel, and back into parallel data at the receiver. Serializing data allows for an exponential reduction in the quantity of transmission lines when interconnecting individual components.

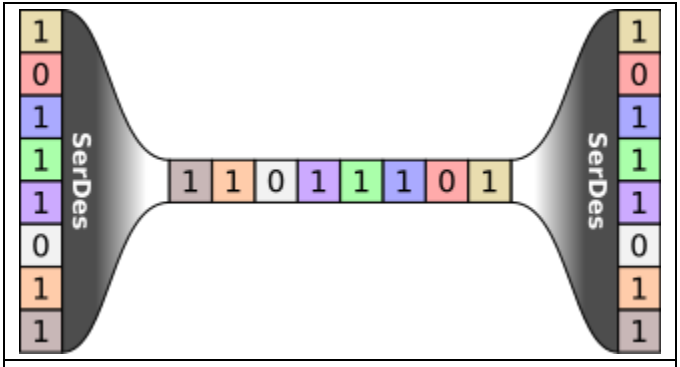


Figure 3 – SerDes Utilization (2014 Surrel)

Section 4.3 – Methods and Materials

The first goal that our team tackled was to understand the standardized specifications defined by the JEDEC Solid State Technology Association (formerly referred to as the Joint Electron Device Engineering Council, or simply JEDEC). The two specifications in question are the 2003 JESD79C document (JEDEC, 2003) and the 2012 JESD79-4 DDR4 Specification (JEDEC, 2012). The 2003 JESD79C document standardized the double data rate SDRAM structure. The 2012 JESD79-4 extrapolated from the DDR, DDR2, and DDR3 documents, and defined new DRAM features and functional-level operating schemes. As a team that is new to all DDR standards, let alone DDR4, we had to spend a notable amount of time familiarizing ourselves with the underlying technologies behind DDR4. This posed a barrier to entry for us.

The entry, in this context, is the actual initiation of the design and development process. The first deliverable was thus a thorough presentation on the technology. After the documents were split up amongst our members, I was responsible for digesting all chapters in the 2012 JESD79-4 document that related to “Modes of Operation.” The modes of operation that I covered are: Test Mode, Target Row Refresh Mode, Fine Granularity Refresh Mode, Gear-Down Mode, Post Package Repair (Optional) Mode, Soft Post Package Repair (Optional) Mode, Per DRAM Addressability Mode, CA Parity Persistent Error Mode, CA Parity Latency Mode, DLL-Off Mode, Temperature Refresh Mode, Low-Power Auto Self Refresh Mode, Power-Down Mode, Programmable Preamble Modes, Synchronous ODT Mode, and finally Asynchronous ODT Mode.

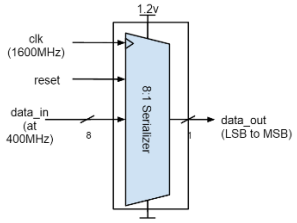
Upon our team’s completion of the JEDEC spec dissection, we set our sights upon Micron’s DDR4 SDRAM MT40A datasheet. This datasheet showed the application of the JEDEC DDR4 specification onto a designed and commercially available DDR4 SDRAM module. Due to the fact that JEDEC expects its readers to have prior exposure to SDRAM, DDR, DDR2, and DDR3, its documentation is not fully exhaustive, and it builds off of definitions created in the previous DDR standards. This made the process of entering a new technical domain more difficult. It was helpful to explore Micron’s DDR4 SDRAM documentation because it brought clarity to information that was vague in nature within the JEDEC specification.

The third deliverable that we approached was a presentation based on a research paper titled “A 1.2 V 30 nm 3.2 Gb/s/pin 4 Gb DDR4 SDRAM with Dual-Error Detection and PVT-Tolerant Data-Fetch Scheme” (Sohn et al. 2013). This paper discusses a non-commercial implementation of the DDR4 SDRAM standard. I delved into the bank structure that the paper

employed. This paper went a step further than the Micron documentation because it revealed functional implementation-level circuitry that a company such as Micron would deem proprietary. Micron only revealed physical operating specifications such as the pin layouts and environmental operating conditions. The deliverable drawn from this paper was to create a presentation and explain the findings that the paper disclosed.

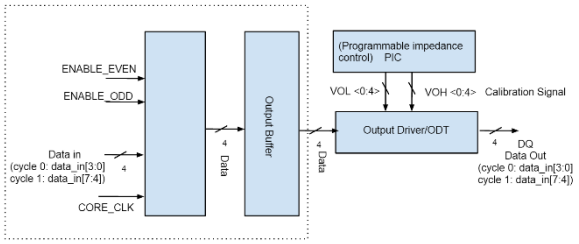
Section 4.3.1 Deserializer Development

The fourth and final goal for our Capstone is the realization and development of a DDR4 compliant PHY. This portion of the Capstone is when our team finally took to the tools and began designing blocks for use in the DDR4 Controller Interface. We targeted our PHY design for a controller that operated at 400MHz and used eight parallel lines to process a total of 3.2 Gbps. Following the controller's specifications, we needed to agree upon the required intra-PHY interface. The intra-PHY interface is the interface that outlines the connections between our modularized PHY blocks and produces the complete PHY interface. Figure 4 visualizes the interface specification that we outlined for our intra-PHY interface. The flow of data follows the sub-figures' notations; The Serializer's data_in comes from the controller and feeds data_out into the Transmitter's Data_in, The Transmitter's DQ outputs to a transmission line, The Receiver's data_in reads from a transmission line and outputs data_rec to the Deserializer, and the Deserializer's data_deser_out and data_valid go to the controller.



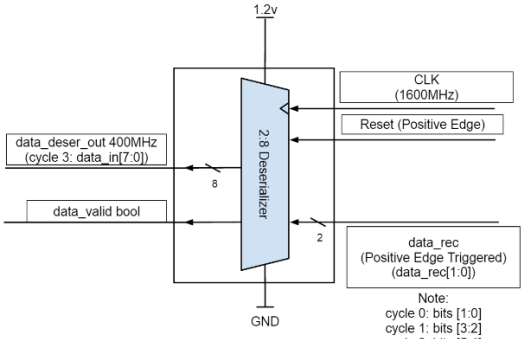
Serializer

Figure 4.1



Transmitter

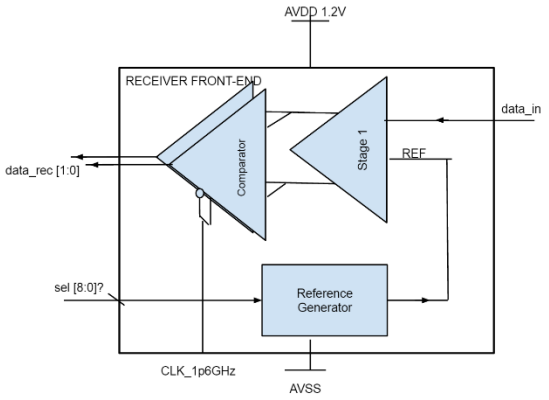
Figure 4.2



Deserializer

(Updated 2015-03-14)

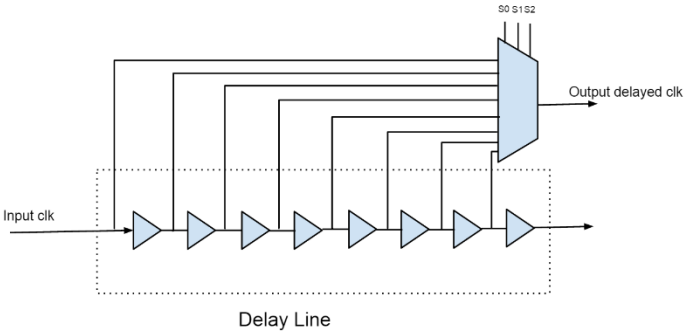
Figure 4.4



Receiver

Updated : 2015-03-08

Figure 4.3



Clock Generator

Figure 4.5

Figure 4 – The intra-PHY interface specification

The next step was to model our respective modules in either Verilog (Digital) or Verilog-A (Analog) and create a test bench to confirm our Verilog descriptions perform to standard. I modeled the Deserializer in Verilog due to it being a digital circuit. Figure 5 shows the resulting

Verilog implementation of the Deserializer. The block's outputs are data_valid and data_out. The data_out pins represent the deserialized output.

The data_valid keeps track of the accuracy on the Deserializer's outputs and raises a positive edge as soon as the deserialized data on the output lines is valid. The data_valid pin drops low after a 1.6GHz clock period, but the deserialized data remains valid on the device's output for a complete 400MHz clock cycle. The inputCounter variable is used to demultiplex the two input data signals onto eight output data lines. The two input signals are expected to arrive on the positive clock edge of a 1.6GHz clock. If the device's inputs are continuously provided data without interruptions, the inputCounter will overflow and continue to demultiplex the data correctly. If the Deserializer's input data has interruptions, the Deserializer needs to be reset-cycled. Reset-cycling occurs when the Deserializer sees a positive edge reset signal. As soon as the reset signal returns low, the Deserializer will deserialize new data on the next possible positive clock edge.

```
always@(posedge clk_1600, posedge reset)
begin
        if(reset) begin
                data_valid <= 1'b0;
                inputCounter  <= 2'b00;
        end
        else begin
                data_valid <= 1'b0;
                case(inputCounter)
                        2'b00: begin
                                data_out_1600mhz[0] <= data_rec[0];
                                data_out_1600mhz[1] <= data_rec[1];
                                end
                        2'b01: begin
                                data_out_1600mhz[2] <= data_rec[0];
                                data_out_1600mhz[3] <= data_rec[1];
                                end
                        2'b10: begin
                                data_out_1600mhz[4] <= data_rec[0];
                                data_out_1600mhz[5] <= data_rec[1];
                                end
                        2'b11: begin
                                data_out_1600mhz[6] <= data_rec[0];
                                data_out_1600mhz[7] <= data_rec[1];
                                //produce 400MHz output
                                data_out[7:6] <= data_rec[1:0];
                                data_out[5:0] <= data_out_1600mhz[5:0];
                                data_valid <= 1'b1;
                                end
                endcase
                inputCounter <= inputCounter + 1;
                //inputCounter overflows (resets) on its own as long
                //as data is coming in without interruption.
        end
end
```

Figure 5 – Verilog snippet outlining the state transitions of the Deserializer. The final outputs from the block are data_out as well as data_valid.

```
// Instantiate modules to test
reg reset_reg;
reg [1:0] data_rec_reg;
wire [7:0] data_out_wires;
wire data_valid_wire;
// This is the DUT (Device Under Test)
deserializer deserializer_0 (
        .clk_1600(clk_1600),
        .reset(reset_reg),
        .data_rec(data_rec_reg),
        .data_out(data_out_wires),
        .data_valid(data_valid_wire)
);

//Execute the test vectors
reg [7:0] test_index; //Track which test vector is being tested
reg [7:0] data_index; //Track position of data within said test vector
initial
begin
        $vcdpluson;
        test_index = 0;
        data_index = 0;
        assign data_rec_reg = {src_bits[test_index][data_index+1], src_bits[test_index][data_index]};
        //Strobe Reset on Negative clock edge to prepare testing the vectors
        repeat(5) @(negedge clk_1600) reset_reg = 0;
        repeat(5) @(negedge clk_1600) reset_reg = 1;
        @(negedge clk_1600) reset_reg = 0;
        //Test each vector and display results
        for(test_index = 0; test_index < 8; test_index = test_index + 1)
        begin
                for(data_index = 0; data_index < 8; data_index = data_index + 2)
                begin
                        @(posedge clk_1600);//allow a positive edge to bang
                        #(0.001);
                        // ^ This delay is to account for the testbenches delta-cycle
                        #(`CLOCK_PERIOD/2) // This delay is necessary for the PAR simulator.
                        if(data_valid_wire)
                        begin
                                if(data_out_wires == src_bits[test_index])
                                        $display("==== PASSED test %d", test_index);
                                else
                                        $display("==== FAILED test %d,\tTest Vector : %b,\t Output :
%b", test_index, src_bits[test_index], data_out_wires);
                        end
                end
        end
        $vcdplusoff;
        $finish;
end
```

Figure 6 – The test bench's test procedure

To test the functionality of the device, I prepared a test bench in Verilog (Figure 6). The test bench tests eight data vectors consecutively against my Deserializer. The data vectors are assorted combinations of input patterns (Figure 7). Referring back to Figure 6, the test bench first reset-cycles my device by creating a positive edge, holding reset positive for 5 periods of

1.6GHz, and then settings reset LOW. On the immediately following positive clock edge, the test bench begins to place the test vectors on the data lines. The outputs of the Deserializer are compared against the test vectors when data_valid goes HIGH, and the results are displayed in the console.

```
//Test Data Vectors
wire [7:0] src_bits [7:0];
assign src_bits[0] = 8'hAB;
assign src_bits[1] = 8'h12;
assign src_bits[2] = 8'hFF;
assign src_bits[3] = 8'h00;
assign src_bits[4] = 8'h0F;
assign src_bits[5] = 8'hF0;
assign src_bits[6] = 8'hAA;
assign src_bits[7] = 8'h1F;
```

Figure 7 – The test data vectors that the test bench uses

After the RTL descriptions passed the test bench simulation, I synthesized my Verilog model and re-simulated the test bench. The synthesized Deserializer successfully completed the test bench. I then ran place-and-route on the synthesized Deserializer and again tested it against my test bench.

In the final stage, I brought the place-and-route output into a block within the Cadence Virtuoso environment (Figure 8) for the purpose of integrating it with the remaining teams' PHY

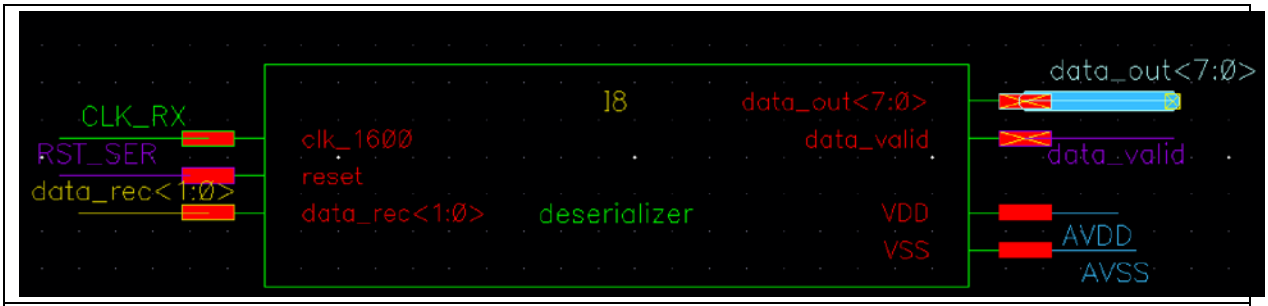


Figure 8 – The Deserializer imported into virtuoso

blocks. At this point the team collectively generated a virtuoso test bench and we ran it against our complete PHY interface.

Section 4.4 – Results and Discussion

By having all of our blocks imported into Cadence Virtuoso, we could simulate a stream of data that propagates through all of our modules. We ran a simulated data stream. I analyzed the signals that interface with the Deserializer. The inputs were properly provided by the

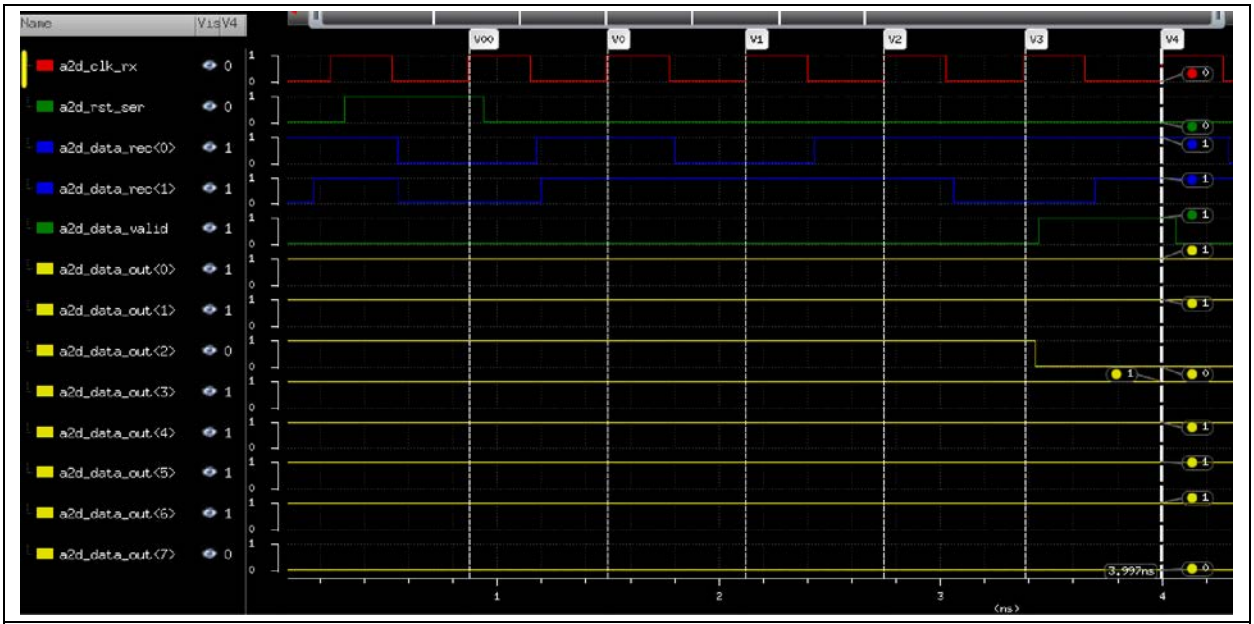


Figure 9.1 – First frame of deserialized data. Data begins coming in at time=V0

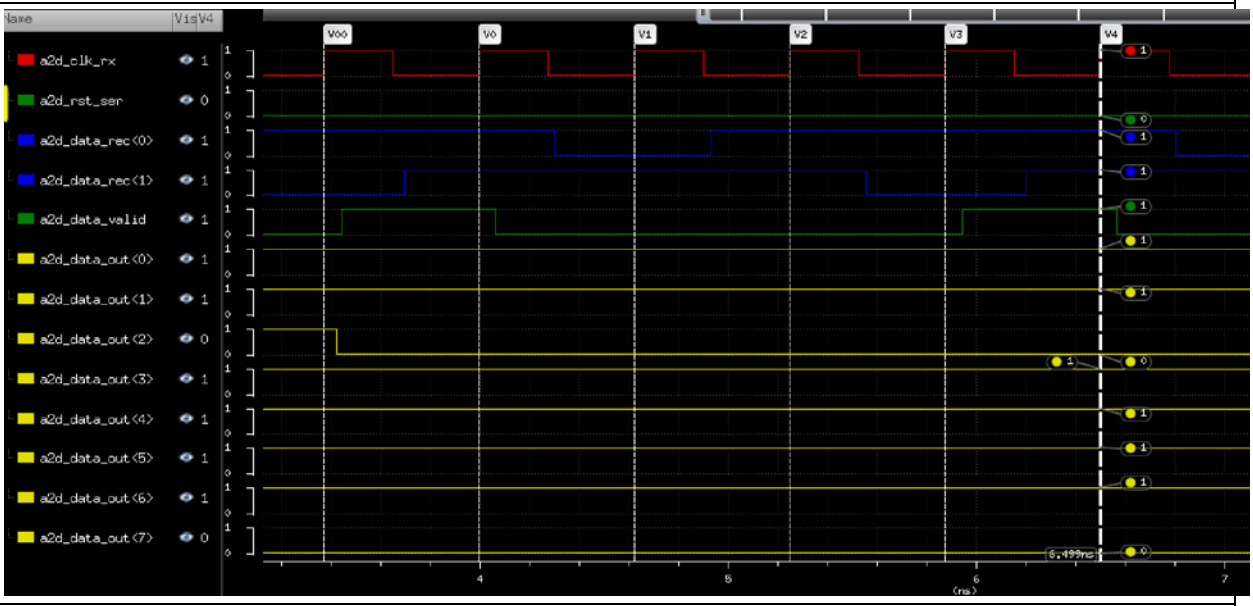


Figure 9.2 – Second frame of deserialized data. Data begins coming in at time=V0. Note that reset is never triggered.

Receiver block to the Deserializer's inputs. They were aligned with the positive edge of the clock. A reset signal was simulated in order to reset-cycle the Deserializer. After observing the produced traces, we can conclude that the data is in fact deserialized properly (Figure 9), and that the Deserializer is successfully implemented to fit the required specifications. The resulting traces were produced when analyzing two incoming data frames of eight bits each. Both frames were deserialized using the Deserializer. When looking at Figure 9, the time value at V00 is one full clock cycle before the first bit of data is placed on the inputs of the Deserializer. The data arrives at 1.6Gbps/line at times V0, V1, V2, and V3. At time V4 the data_valid bit shows HIGH and the value of the data on the output lines is valid. Notice that the reset signal is only triggered for the first frame of data, and it remains low for the remainder of the test. This is due to the data being sent to the Deserializer consecutively without any interruptions. The interface performs compliant to all of the test vectors within the test bench, our design proves to be functionally sound.

Section 5 – Concluding Reflections

Over the course of the capstone, our team had to make many adjustments as we tuned our focus more and more precisely toward what we have achieved today. We began with a very broad scope that permitted for us to explore an ample variety of directions toward which we could take our projects. We originally focused our efforts on the prospect of designing a controller for DDR4 memory. Throughout our ramp-up period, where we immersed ourselves in a thorough literature review, we found that a DDR4 memory controller had recently been created

at BWRC. In an effort to support the existing controller, we honed in on the exact portion of a memory interface that we wished to create – The Physical Interface Layer (PHY) between this controller and the DDR4 SDRAM. Through the journey of the process, I learned more about DDR memory and their control systems than I could have imagined. Being that it was my first time working on creating hardware, I found the opportunity to partake in a hardware development cycle to be an incredibly rewarding experience. Most importantly, it showed me how much more there is out in the field that I can learn and master. During the planning phase, I learned the value of experience. As we scoped out our project at the start of the Fall semester, we made naïve blunders that only experience could have caught. We originally overlooked loose ends such as acquiring design tool proficiency, collaborating on mixed signal design simulation, and clear definitions of intra-PHY component interfaces. Looking back, I recall our advisors stressing these exact issues ahead of time, but our lack of experience, as a novice hardware design group, prevented us from immediately appreciating the significance of the suggestions. We inevitably introduced delays to our design process. Fortunately, that is how growth works. I know that I learned an immense amount of knowledge about the hardware design domain.

Future development of our project can extend in a few directions. First, and foremost, our design can be extended to use actual foundries' technologies. This would affect the electrical characteristics of the devices and would need to be explored further to ensure compatibility. Second, the combination of the PHY Interface between the controller and the memory will have unexpected interface incongruencies. This would require the controller, or our PHY, to be tweaked to fit the existing interface definitions. Additional exploration of the Memory Physical Interface technology space can made in the space of error-detecting and error-correcting data transmission systems.

From an industry-scoped perspective, we also grew the ability to perform strategic analysis based on real-world market trends. We were able to take the analysis and produce actionable milestones to help bring us toward the goal of executing creative market-facing strategy. I found the overall Capstone experience to be a tremendous growth opportunity, and I am thankful to have had the opportunity to be a part of it.

BIBLIOGRAPHY

Allan, Graham
2015, DDR4 Bank Groups in Embedded Applications, http://www.synopsys.com/Company/Publications/DWTB/Pages/dwtb-ddr4-bank-groups-2013Q2.aspx, accessed March 13th, 2015

Altera
2013 December, DDR2, DDR3, and DDR4 SDRAM Board Design Guidelines, https://www.altera.com/en_US/pdfs/literature/hb/external-memory/emi_plan_board_ddr2.pdf, accessed March 14th, 2015

Bach, Matt
Aug. 2014, Tech Primer: DDR4 RAM, http://www.pugetsystems.com/labs/articles/Tech-Primer-DDR4-RAM-589/ , accessed March 1st ,2015.

Feng, J., et al.
2013, System level signal and power integrity analysis for 3200 Mbps DDR4 interface. Electronic Components and Technology Conference (ECTC), 63rd IEEE 1081:1086., accessed February 16, 2015.

Gill, Rajinder
2010, August, Everything You Always Wanted to Know About SDRAM (Memory): But Were Afraid to Ask, http://www.anandtech.com/show/3851/everything-you-always-wanted-to-know-about-sdram-memory-but-were-afraid-to-ask/2, accessed March 13th, 2015

Hall, Bronwyn
January 2007 An Empirical Analysis of Patent Litigation in the Semiconductor Industry, accessed March 1st, 2015.

Hulkower, Billy
August 2013 2013 Laptops and Desktop Computers - US. http://academic.mintel.com/display/637710/, accessed March 1st ,2015.

IBISWorld
2015 IBISWorld Industry Report Global Semiconductor & Electronic Parts. http://clients1.ibisworld.com/reports/gl/industry/default.aspx?entid=960, accessed March 1st ,2015.

JEDEC Solid State Technology Association
2003, March, Standard: DDR4 SDRAM (JESD79C) Double Data Rate (DDR) SDRAM Specification

JEDEC Solid State Technology Association
2012, September, Standard: DDR4 SDRAM (JESD79-4) DDR4 SDRAM

Micron
2014, DDR4 SDRAM MT40A Datasheet

Panek, Tracey
2015, Samsung Electronics Co., Ltd. http://subscriber.hoovers.com/H/company360/overview.html?companyId=55277000000000, accessed March 1st ,2015.

Patrizio, Andy
24 Jun. 2014,All about DDR4, the next-gen memory coming soon for PCs and mobile devices. PCWorld, http://www.pcworld.com/article/2365823/next-gen-memory-is-coming-fast-here-s-what-you-need-to-know-about-ddr4.html,, accessed March 1st ,2015.

Poole, Ian
SDRAM Memory Basics & Tutorial, http://www.radio-electronics.com/info/data/semicond/memory/sdram-memory-basics-tutorial.php, accessed March 13th, 2015

Porter, Michael E., and Victor E. Millar.
1985, How Information Gives You Competitive Advantage, accessed March 1st ,2015.

Porter, Michael
2008, The Five Competitive Forces That Shape Strategy, accessed February 16th, 2015.

Quinn, Gene

October 30 2009 The Risk of Not Immediately Filing a Patent Application, accessed March 1st, 2015.

Quinn, Gene
January 2011, The cost of obtaining a patent in the US, accessed February 28th, 2015

Samsung
Samsung DDR4 SDRAM, http://www.samsung.com/global/business/semiconductor/product/computing-dram/overview, accessed March 16th, 2015

Shane, Olafson
April 2007 Patents vs. Trade Secrets-Giving Your Business the Competitive Edge, accessed March 1st, 2015.

Sohn, K., et al.
2013, A 1.2 V 30 nm 3.2 Gb/s/pin 4 Gb DDR4 SDRAM with dual-error detection and PVT-tolerant data-fetch scheme. IEEE Journal of Solid State Circuits 48(1):168-177. , accessed February 16, 2015.

Stauffer et al,
2009, High Speed SerDes Devices and Applications, Springer Science & Business Media

Stim, Rich
March 2011, The usual costs of filing a patent application, accessed February 28th, 2015

Surrel Grégoire
2014, October, File:SerDes (Serializer - Deserializer).svg http://commons.wikimedia.org/wiki/File:SerDes_(Serializer_-_Deserializer).svg, accessed March 13th, 2015

Transparency Market Research
May 2014, Next Generation Memory Technologies Market (Non-volatile and Volatile) - Global Industry Analysis, Market Size, Growth, Trends and Forecast 2013 - 2019, accessed October 14th, 2014

Ulama, Darryle
2014 IBISWorld Industry Report 33441a: Semiconductor & Circuit Manufacturing in the US. http://www.ibis.com, accessed February 16, 2015.

U.S. Patent and Trademark Office
October 2013 Description of Patent Types, http://www.uspto.gov/web/offices/ac/ido/oeip/taf/patdesc.htm, accessed March 1st, 2015

World Intellectual Property Organization

May 26 1989 Treaty on Intellectual Property in Respect of Integrated Circuits, http://www.wipo.int/treaties/en/text.jsp?file_id=295136, accessed March 1st, 2015

Appendix

Appendix A – Deserializer Verilog

```
// deserializer.v

// Author : Miron Veryanskiy
// Revision : 0.2.1
// Revision Date : 2015-04-15 - Cleaning up comments.

// Input :
//	Two channels clocked at 1.6GHz each
//	One 1.6GHz clk
//	One positive-edge reset
//		data can start coming in on the posedge of the following clock cycle
// Output :
//	Eight 400 MHz channels

module deserializer (
	input clk_1600,
	input reset,
	input [1:0] data_rec,
	output reg [7:0] data_out,//Output frame. Valid for full 400MHz cycle
	output reg data_valid
	);

	reg [1:0] inputCounter;
	reg [7:0] data_out_1600mhz;//Builds the frame to put on the output buffer.

//	ASSUMING : I RECEIVE DATA ON 2 LINES ON POS EDGE AT 1.6GHz
//	ASSUMING : Kalika sends me data that is positive-edge aligned
//	ASSUMING : Data chunks are coming in from LSB to MSB
//	NOTE : Deserialized data is valid for ONE full 400Mhz cycle after
```

```
//      data_valid is high.
        always@(posedge clk_1600, posedge reset)
        begin
                if(reset) begin
                        data_valid <= 1'b0;
                        inputCounter  <= 2'b00;
                end
                else begin
                        data_valid <= 1'b0;
                        case(inputCounter)
                                2'b00: begin
                                        data_out_1600mhz[0] <= data_rec[0];
                                        data_out_1600mhz[1] <= data_rec[1];
                                        end
                                2'b01: begin
                                        data_out_1600mhz[2] <= data_rec[0];
                                        data_out_1600mhz[3] <= data_rec[1];
                                        end
                                2'b10: begin
                                        data_out_1600mhz[4] <= data_rec[0];
                                        data_out_1600mhz[5] <= data_rec[1];
                                        end
                                2'b11: begin
                                        data_out_1600mhz[6] <= data_rec[0];
                                        data_out_1600mhz[7] <= data_rec[1];
                                        //produce 400MHz output
                                        data_out[7:6] <= data_rec[1:0];
                                        data_out[5:0] <= data_out_1600mhz[5:0];
                                        data_valid <= 1'b1;
                                        end
                        endcase
                        inputCounter <= inputCounter + 1;
                        //inputCounter overflows (resets) on its own as long
                        //as data is coming in without interruption.
                end
        end
endmodule
```

Appendix B – Test Bench Verilog

```
//Filename ; tb_deserializer.v
```

```
//Description : Testbench to test deserializer.v
//Author : Miron Veryanskiy
//Revision : 0.1.2
//Revision Date : April 15, 2015 - Cleaned up comments for report


`timescale 1 ns / 1 ps //Time-units measured in 1 ns with 1ps resolution (0.001 ns resolution)
module tb_deserializer;

	//Test Data Vectors
	wire [7:0] src_bits [7:0]; //2D array-like structure of wires http://goo.gl/zDakcmi
	assign src_bits[0] = 8'hAB;
	assign src_bits[1] = 8'h12;//src_bits[0][1] will return the 2nd lsb from 8'h12 = '1'
	assign src_bits[2] = 8'hFF;
	assign src_bits[3] = 8'h00;
	assign src_bits[4] = 8'h0F;
	assign src_bits[5] = 8'hF0;
	assign src_bits[6] = 8'hAA;
	assign src_bits[7] = 8'h1F;

	//Clock Setup - 0.625ns Period  = 1.6Ghz CLK
	reg clk_1600 = 0;//First Clock
	reg clk_1600_2 = 0;//First Clock phase shifted 90 degrees - Unused in the tb.
	always begin
		#(`CLOCK_PERIOD/2) clk_1600 = ~clk_1600; //# introduces delay prior to
execution
	end

// Instantiate modules to test
reg reset_reg;
reg [1:0] data_rec_reg;
wire [7:0] data_out_wires;
wire data_valid_wire;
// This is the DUT (Device Under Test)
deserializer deserializer_0 (
	.clk_1600(clk_1600),
	.reset(reset_reg),
	.data_rec(data_rec_reg),
	.data_out(data_out_wires),
	.data_valid(data_valid_wire)
);

//Execute the test vectors
reg [7:0] test_index; //Track which test vector is being tested
```

```
reg [7:0] data_index; //Track position of data within said test vector
initial
begin
	$vcdpluson;
	test_index = 0;
	data_index = 0;
	assign data_rec_reg = {src_bits[test_index][data_index+1],
src_bits[test_index][data_index]};
	//Strobe Reset on Negative clock edge to prepare testing the vectors
	repeat(5) @(negedge clk_1600) reset_reg = 0;
	repeat(5) @(negedge clk_1600) reset_reg = 1;
	@(negedge clk_1600) reset_reg = 0;
	//Test each vector and display results
	for(test_index = 0; test_index < 8; test_index = test_index + 1)
	begin
		for(data_index = 0; data_index < 8; data_index = data_index + 2)
		begin
			@(posedge clk_1600);//allow a positive edge to bang
			#(0.001);
			// ^ This delay is to account for the testbenches delta-cycle
			#(`CLOCK_PERIOD/2) // This delay is necessary for the PAR simulator.
			if(data_valid_wire)
			begin
				if(data_out_wires == src_bits[test_index])
					$display("==== PASSED test %d", test_index);
				else
					$display("==== FAILED test %d,\tTest Vector : %b,\t Output
: %b", test_index, src_bits[test_index], data_out_wires);
			end
		end
	end
	$vcdplusoff;
	$finish;
end

endmodule
```