

# Beamforming and MIMO Digital Radio Baseband and Testbed for Next Generation Wireless System

*Zhen Yuan*



Electrical Engineering and Computer Sciences  
University of California at Berkeley

Technical Report No. UCB/EECS-2016-93

<http://www.eecs.berkeley.edu/Pubs/TechRpts/2016/EECS-2016-93.html>

May 13, 2016

Copyright © 2016, by the author(s).  
All rights reserved.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission.

### Acknowledgement

I take this opportunity to express my thanks to my advisors, professors Elad Alon, Borivoje Nikolic, and Vladimir Stojanovic for their advice and supervision. I am also grateful to receive help from Ph.D. students at BWRC, Antonio Puglielli, Paul Rigge, Angie Wang, and Christopher Yarp. All of them guided me through the process of the whole project, and made this capstone project meaningful and impressive to me.

I also say thanks to my teammates – Niral Milan Sheth and Yiduo Xu for their collaboration and contribution to this project, leading to the successful implementation of this project.

University of California, Berkeley College of Engineering

**MASTER OF ENGINEERING - SPRING 2016**

**Electrical Engineering and Computer Science**

**Physical Electronics and Integrated Circuits**

**Beamforming and MIMO Digital Radio Baseband and Testbed for Next  
Generation Wireless System**

**Zhen Yuan**

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: [Borivoje Nikolic/EECS](#)

# Table of Contents

INDIVIDUAL PAPERS.....	1
1. Introduction.....	1
2. Background Knowledge.....	2
3. Beamforming Matrix Hardware Implementation.....	5
4. Simulation Results.....	13
5. Conculsion.....	13
TEAM PAPERS.....	15
1. Introduction.....	15
2. Trends and IP Strategy.....	15
3. Industry Analysis.....	17
4. Market Strategy.....	19
Reference.....	21

# List of Figures

Figure1 Work breakdown structure.....	2
Figure2 Beamforming process with phase shift $360^\circ$ .....	3
Figure3 $2 \times 2$ MIMO structure.....	3
Fiuure4 High level design of signal recovery.....	4
Figure5 Hardware structure for beamforming matrix block.....	6
Figure6 Conjugate beamforming implementation.....	7
Figure7 Simple multiplication.....	8
Figure8 Top-level hardware implementation for matrix multiplication.....	8
Figure9 Normalization of result.....	10
Figure10 Inverse implementation example.....	11
Figure11 Matrix transformation.....	11
Figure12 Overall hardware structure.....	12

# INDIVIDUAL PAPERS

## 1. Introduction

Wireless communication has been an essential technology in our daily lives. With the help of wireless communication, information can be rapidly and accurately transmitted from phone to phone, and server to server. Currently, more and more companies are working on developing communication systems with high speed and low noise. In this process, different communication standards emerge, each requiring its own hardware structure. The lack of a consistent hardware structure among different companies impede the transformation from current generation of communication to the next one. If there is a general design frame for the communication system, developers will be able to improve current design method from the frame rather than the scratch. Our project aims to help design such a hardware generator for a Software-Defined Radio (SDR) system. The hardware generator will be created by Chisel, an open-source hardware construction language developed at UC Berkeley. Chisel will contribute to develop an open-source and generalized wireless communication system. With the help of open source, developers will share their codes and access to others' codes for free, making it easier to learn from others and improve the communication system design. A generalized system provides developers with a flexible design, meaning that developers have ability to control the parameters of the communication system, and find the optimized design space.

Belonging to a research program at Berkeley Wireless Research Center (BWRC), our capstone project focuses on the construction of beamforming and Multiple-Input and Multiple-Output (MIMO) blocks for digital radio baseband and testbed for next generation wireless system. We implemented these blocks in hardware, and we tested the functionality of the blocks based on Register-Transfer Level (RTL). Our blocks are divided into multiple pieces, including CORDIC, matrix multiplication unit, and beamforming matrix block. These pieces are

integrated together in a higher level to function as a whole system. There are three graduate students in our team. Yiduo (Eva) Xu worked on CORDIC design and test, and provided it for BWRC to integrate with Carrier Frequency Offset (CFO) block. Niral Sheth focused on the implementation of matrix multiplication unit, and the integration of different blocks. I contributed to the construction of beamforming matrix block in hardware. Our work breakdown structure is shown below.

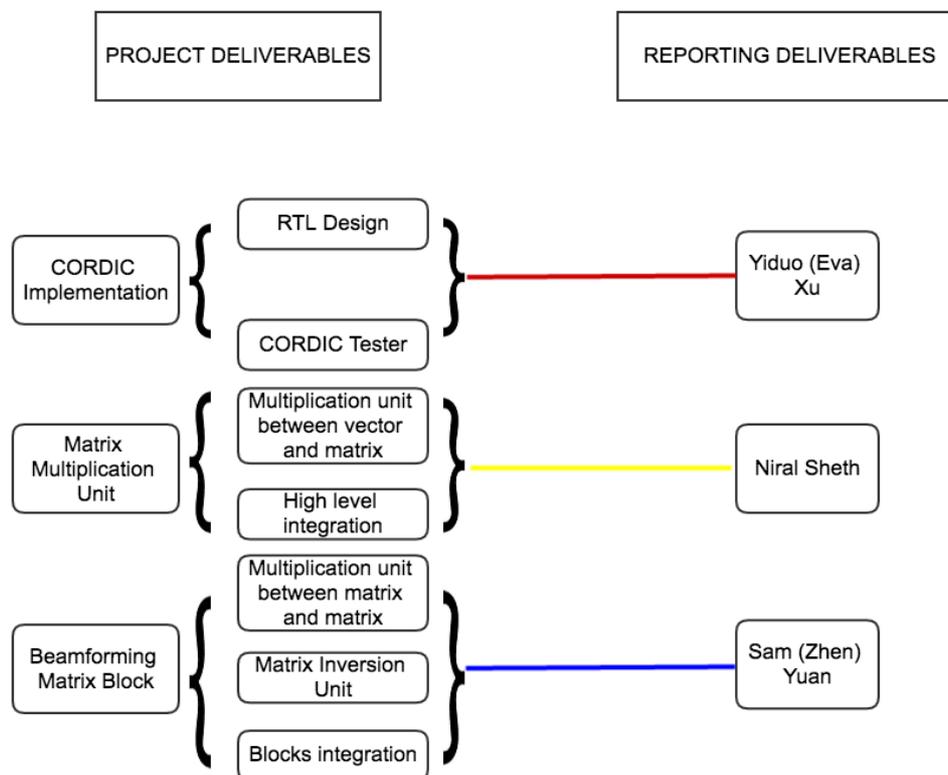


Figure 1 Work breakdown structure.

## 2. Background Knowledge

Beamforming is a technique that forces signals to transmit in a certain direction. Generally, as the signal transmits in the medium, the signal strength will get weaker and weaker. First, the signal strength may get too weak to be recognized by the receiver. Second, the noise signal in the medium may get comparable to the weakened signal, and disrupt signal reception. Beamforming takes advantage of the interference and gets rid of this problem. When two

signals with a phase shift but same frequency interfere with each other, there will form constructive points and destructive points.

Constructive points are formed when the peaks of two signals encounter each other, and the signal gets strengthened in this case. If we connect all the constructive points and draw a curve, we will find it is exactly the signal transmission path. The figure on the right illustrates this process. The two antennas are sending signals with a phase shift

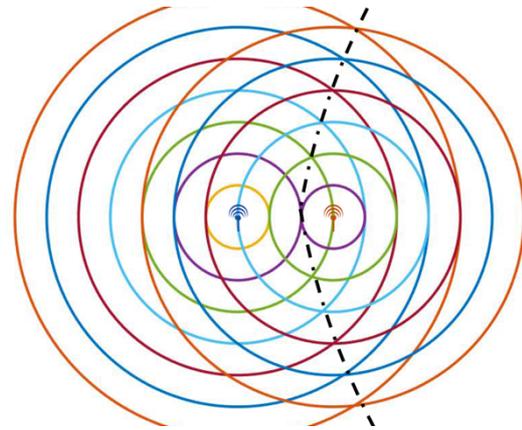


Figure 2 Beamforming process with phase shift  $360^\circ$ .

$360^\circ$  between each other. The constructive points are the intersections of curves with the same color. The black dash line indicates beamforming path. Therefore, the signal strength will get increased and not get weakened when arriving at receiver, and thus ensure the correctness of receiving process.

Future wireless communication system will include features like high data rates and high signal quality (Veena et al. 2011). MIMO systems exactly fulfill these functions by utilizing both temporal and spatial domains. The figure below illustrates a simple MIMO structure. There are

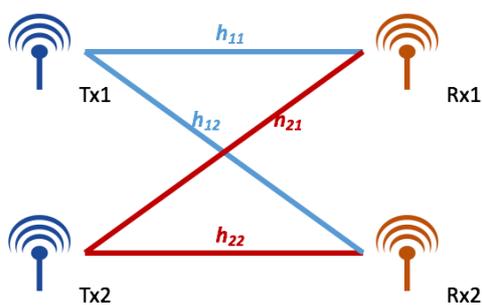


Figure 3  $2 \times 2$  MIMO structure.

two antennas at the transmitter and two at the receiver.

Both transmitter 1 and transmitter 2 can send signals to either receiver 1 or receiver 2. Therefore, this

MIMO system can work in parallel channels rather than a single channel. The channel environment is

denoted by a matrix  $H$ . Generally, if we have  $K$

antennas at transmitter and  $M$  antennas at receiver, the environment matrix size should be  $M \times$

$K$ . Here, we need to set  $M$  greater or equal to  $K$ , thus ensuring that receiver can successfully

decode the signals. Now suppose we have a signal with high data rates. If we use traditional transmission method, it will take a long time and much space to transmit it. MIMO systems help solve this problem. At the transmitter, the high data rate signal can be first divided into multiple low data rate signals, that is, cut a signal into multiple pieces. Then, these low data rate signals will transmit from transmitter, pass through the environment, and finally come to receiver (Powell 2014). The antennas at receiver will analyze the received signals as well as the environment, and then recover the high data rate signal, “by taking advantage of the spatial diversity resulting from spatially separated antennas” (Karkooti et al. 2005). In this situation, multiple pieces of signals are transmitted through the environment at the same time, saving time and energy.

We need to implement several blocks to recover the transmitted signal from receiver, including channel estimation, matrix multiplication unit, and beamforming matrix determination. The other capstone team *Digital Radio Baseband and Testbed for Next Generation Wireless System* is working on channel estimation. The figure below shows the high level structure of signal recovering process. In the first phase, channel estimation block determines the environment matrix  $H$ , and then output it to beamforming matrix block. The beamforming matrix block analyzes received signal and environment matrix  $H$ , and then produce a beamforming matrix

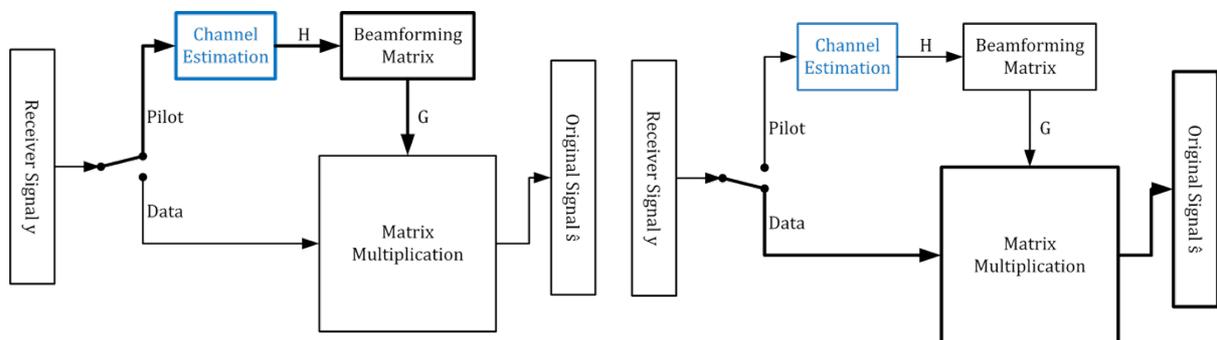


Figure 4 High level design of signal recovery. (Left: phase 1 Right: phase 2)

$G$  and transmit it to the matrix multiplication block. In the second phase, data signals as well as beamforming matrix will flow into matrix multiplication unit, and then recover the signal. Readers can go to *Digital Radio Baseband* team's paper to find more information about channel estimation implementation. Niral talks about matrix multiplication block and the integration of different blocks in his paper. Later on, I will discuss the specific implementation of the beamforming matrix unit.

### 3. Beamforming Matrix Hardware Implementation

The beamforming matrix can be determined with three methods, depending on the Signal to Noise Ratio (SNR) of the system (Puglielli et al. 2015). The first method is named conjugate beamforming. It is applied when SNR is very low. The beamforming matrix  $G$  can be found from environment matrix  $H$  by just matrix transpose and conjugation (or Hermitian matrix). The formula below provides a mathematical representation of this method.

$$G = H^H$$

The second method is zero forcing, applicable for conditions when SNR is very high. This method is more difficult to implement because of more matrix operations. The mathematical representation is shown below.

$$G = (H^H H)^{-1} H^H$$

Compared with conjugate beamforming method, two more matrix multiplications and one more matrix inversion are required. There is a special case for method zero forcing. When the numbers of antennas at transmitter and receiver are equal, the environment matrix  $H$  will be square. For square matrix  $A$ , we have  $A^{-1} = (A^H A)^{-1} A^H$ . This means, in this case, a single matrix inversion is enough, speeding up the beamforming matrix determination. The general method MMSE is used to determine beamforming matrix when SNR is between upper bound and lower bound. Noise factors are included in this method. Compared with zero forcing

method, the only difference lies in the consideration of noise variance  $\sigma^2$ , and one more matrix addition is necessary.

$$G = (H^H H + \sigma^2 I)^{-1} H^H$$

SNR can be encoded into a two-bit number. If SNR is low, it is represented by 00. If SNR is high, it is represented by 10. Otherwise, it is denoted as 01.

All the data are represented in the complex form in our implementation. Both real and imaginary parts consist of  $N$  bits of data. This means, for each entry of data, we have  $2N$  bits. The structure was constructed in Chisel DSP environment. In the top level of the design, the number of antennas at transmitter and receiver was defined. In this report, the number of antennas at transmitter is denoted as  $K$ , and that at receiver is denoted as  $M$ . In the beamforming matrix block, environment matrix  $H$ , SNR, noise variances and a valid signal from upstream were defined as inputs, and beamforming matrix  $G$ , a valid signal to down stream, and a ready signal to up stream were generated as outputs.  $H$  is an  $M \times K$  matrix provided by channel estimation block, which was represented by a vector with length  $M \times K$  in Chisel. SNR, noise variance and valid signal originated from previous block calculation results.  $G$  is an  $K \times M$  matrix represented by another vector with length  $K \times M$ , transmitting to matrix multiplication unit. The hardware structure for beamforming matrix can be described in the figure below.

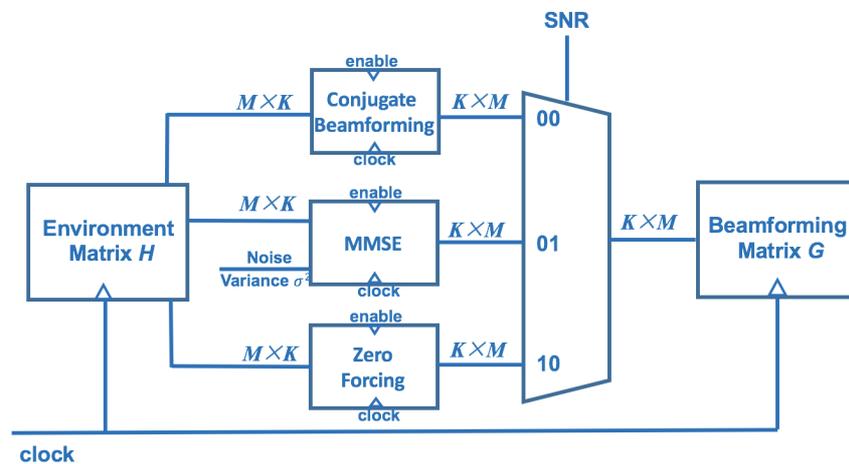


Figure 5 Hardware structure for beamforming matrix block.

Conjugate beamforming was the easiest method to implement. The block accepted environment matrix  $H$  and assigned the conjugate of complex data to corresponding entries in the register which stores the results. The following diagram illustrates this process using a  $3 \times 2$   $H$  matrix. Take the third entry ( $a_{21}$ ) of  $H$  for example. Data  $1+2i$  is first conjugate to  $1-2i$ , then mapped to the second entry of the beamforming register, and stored here for later usage. This

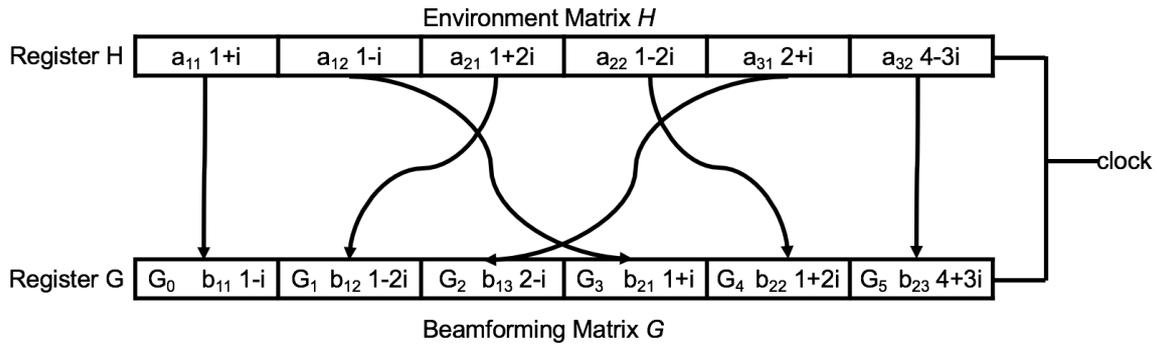


Figure 6 Conjugate beamforming implementation.

process involves only transpose and conjugate and requires no complicated matrix operation. This method is completed in one cycle and saves most power and space.

When coming to methods zero forcing and MMSE, additional matrix multiplications and inversions were a challenge for implementation. The beamforming and MIMO blocks were designed to realize parameterization rather than fixed design. Therefore, the hardware structure must be able to retain its efficiency when parameters change. Another truth is that, one beamforming matrix  $G$  can be used for multiple matrix multiplications after output, meaning that  $G$  is updated much slower. Therefore, there will not be much worry about working frequency of the calculation of  $G$ .

Based on previous analysis, matrix multiplication was implemented using iterative structure rather than pipelined structure. First, an  $K \times M$  matrix was divided into  $K$   $1 \times M$  vectors. Then

multiplication was performed between vectors and matrix  $K$  times. The corresponding entries were multiplied together, and then added to last multiplication result. The figure below calculates a single entry between  $1 \times 3$  vector and  $3 \times 1$  matrix. A scala map function was employed to realize this process. Each single cycle, it outputted a vector with length  $K$ , and stored the result to another register for later usage. The subsequent outputs were then appended after previous results. In this case, it requires  $K$  cycles to obtain final multiplication results. In each

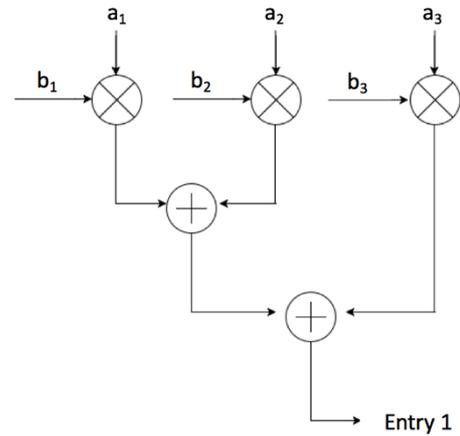


Figure 7 Simple multiplication.

cycle, there performs complex number multiplication  $K \times M$  times and complex number addition  $K \times (M-1)$  times. Since addition is much easier to implement in ASIC design than multiplication, the power and time latency contributed by addition can be ignored. Therefore, the main source for delay and power in each cycle came from multiplication. The figure below describes the hardware structure for  $H^H H$  in my matrix multiplication unit. The multiplication unit reads data from beamforming matrix and performs calculations. In each cycle, the results

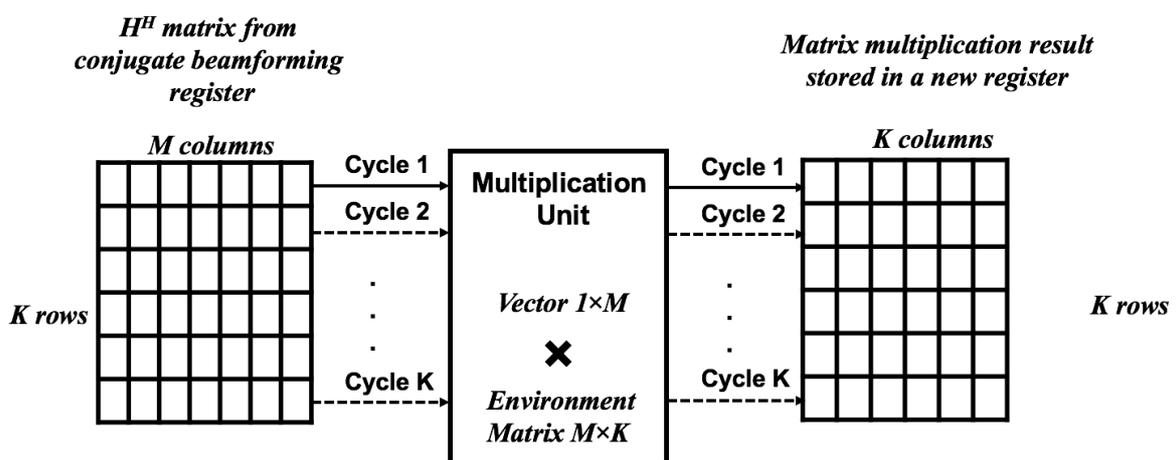


Figure 8 Top-level hardware implementation for matrix multiplication.

are transmitted to a wire and then stored in a new register. Note that, in the beamforming matrix block, there are two matrix multiplication operations for zero forcing and MMSE. The sizes of the matrix in these two multiplications are different. Therefore, two multiplication modules in the final hardware structure were instantiated. The other multiplication module in this block is inverse matrix with size  $K \times K$  times matrix  $H^H$  with size  $K \times M$ . This module also requires  $K$  cycles to obtain final multiplication result.

The most difficult part to implement is matrix inversion, because most methods include division. Division operation is very complicated to implement in hardware. The size of the matrix is decided by the number of antennas at transmitter and receiver. Generally, the size is smaller than  $4 \times 4$ , and we don't need to worry about the scalability of the algorithm (Eilert et al. 2007). However, our hardware structure needs to be parameterizable. This requires a lot of divisions during calculation. A single division takes several decades of cycles. If there is no consideration of scalability, the area and power will be serious issues when the matrix grows big. The division was finally eliminated in the implementation. Instead, a multiplication with coefficients was executed whenever there was a need for normalization (division) and then passed the coefficient downstream. There might be other blocks in the whole system requiring divisions. These blocks could avoid division using the same idea, and multiply their coefficients with my coefficient. Therefore, in the last stage, the division was implemented only once and the normalized result was obtained. The figure below helps clarify the idea. The inverse matrix block generates a coefficient  $A$ , and the beamforming matrix block passes this value to downstream. In the following blocks, there may be other coefficients generated due to avoidance of division. All the coefficients are multiplied together as  $\alpha$  and divided in the last step to obtain normalized results. This greatly decreases the number of divisions, and saves

time and power consumption. The entries of environment matrix were usually complex decimals smaller than 1. Therefore, the iterative multiplications caused the results to be a very small number with decades of bits. Scaling up and bit truncation were adopted here to save power and area, but this resulted in inaccuracy. Also, for big matrix, the huge number of multiplications might not be more efficient than limited number of divisions.

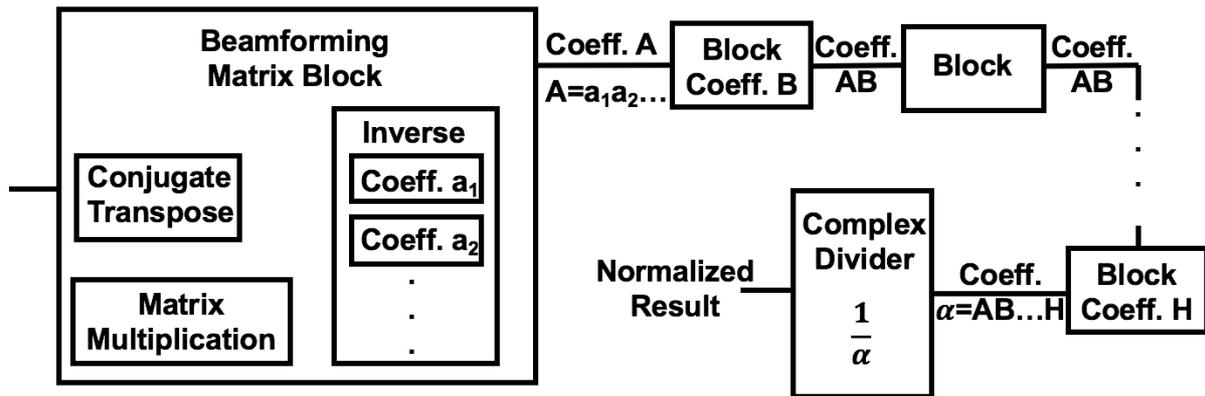


Figure 9 Normalization of the result.

Theoretically, there are multiple methods to implement matrix inversion. Traditional analytical method includes the calculation of adjoint matrix and determinate. This method is efficient when the size of the matrix is small, like  $2 \times 2$ ,  $3 \times 3$  (Rao 2015). However, our MIMO system design is parameterizable, meaning we have to ensure calculation efficiency when matrix grows. After literature review, Gauss-Jordan (GJ) elimination method was finally adopted, because of its easier implementation and higher efficiency.

GJ elimination only performs row addition and subtraction, and complex number multiplication (division is not included here because the result is normalized in the last step). Other matrix inverse operations like QR decomposition and Cholesky require extra complicated operations such as multiplication, square root, sine and cosine, and therefore require more space (Arias-García et al. 2011). Suppose the matrix being inverted is  $A$  with size

$K \times K$ . The steps of GJ elimination are listed below. First, an identity matrix  $I$  with size  $K \times K$  was appended after  $A$ , and formed a new matrix  $[A \ I]$  with size  $K \times 2K$  in one clock cycle. When SNR was within lower bound and upper bound (MMSE), noise variance was added to the elements. Second, row operations were performed and the matrix was transformed into the form  $[\alpha I \ \alpha A^{-1}]$ . Here,  $\alpha$  was the coefficient passed downstream, and  $\alpha A^{-1}$  was the inverse matrix of  $A$  before normalization. During row operations, first row  $R_1$  was designated as the reference, and the first element in first row was denoted as pivot  $p_1$ . Assume the first element of second row was  $p_{21}$ , the calculation  $p_1 R_2 - p_{21} R_1$  was performed to reduce pivot of second row to zero. In each single cycle, only one row operation was conducted due to Chisel's syntax. After  $K-1$  cycles, all the pivots from  $R_2$  to  $R_K$  were changed to zero. From the  $K$ -th cycle, repeated row operation from  $R_2$  to  $R_K$  were executed. After  $O(K^2)$  cycles, a matrix with its bottom left part all zeroes was formed. A  $3 \times 3$  matrix example helps explain this process. The red box indicates current reference row. In each single cycle, there are at most  $4K$  multiplications and  $2K$  subtractions. After obtaining an up-triangle matrix,

$$\begin{array}{c}
 \left[ \begin{array}{cccccc} \color{red}{1} & 0 & 2 & 1 & 0 & 0 \\ 3 & 1 & 0 & 0 & 1 & 0 \\ 2 & 1-i & 3 & 0 & 0 & 1 \end{array} \right] \xrightarrow{\text{cycle1}} \left[ \begin{array}{cccccc} \color{red}{1} & 0 & 2 & 1 & 0 & 0 \\ 0 & 1 & -6 & -3 & 1 & 0 \\ 2 & 1-i & 3 & 0 & 0 & 1 \end{array} \right] \xrightarrow{\text{cycle2}} \left[ \begin{array}{cccccc} \color{red}{1} & 0 & 2 & 1 & 0 & 0 \\ 0 & 1 & -6 & -3 & 1 & 0 \\ 0 & 1-i & -1 & -2 & 0 & 1 \end{array} \right] \\
 \\
 \xrightarrow{\text{cycle3}} \left[ \begin{array}{cccccc} 1 & 0 & 2 & 1 & 0 & 0 \\ \color{red}{0} & 1 & -6 & -3 & 1 & 0 \\ 0 & 0 & 5-6i & 1-3i & -1+i & 1 \end{array} \right]
 \end{array}$$

Figure 10 Inverse implementation example.

Then, the above matrix was transformed to a down-triangle matrix as shown below. This new matrix shared the same hardware structure designed above to perform calculation and avoided

$$\left[ \begin{array}{cccccc} 1 & 0 & 2 & 1 & 0 & 0 \\ 0 & 1 & -6 & -3 & 1 & 0 \\ 0 & 0 & 5-6i & 1-3i & -1+i & 1 \end{array} \right] \rightarrow \left[ \begin{array}{cccccc} 5-6i & 0 & 0 & 1-3i & -1+i & 0 \\ -6 & 1 & 0 & -3 & 1 & 0 \\ 2 & 0 & 1 & 1 & 0 & 0 \end{array} \right]$$

Figure 11 Matrix transformation.

redundant hardware reconstruction. The transformation was performed in one cycle. Again, after  $O(K^2)$  cycles, the left bottom part of the new matrix was reduced to zeroes. Totally, the implementation of GJ elimination only required two registers with size  $K \times 2K$  to store data, and other calculations were performed through wires, saving much space.

Now, a matrix with all the elements zeroes except those on the diagonal was generated. Then, the results were normalized by multiplying each row with the values on the diagonal of other rows. This whole process took  $O(K^2)$  cycles. Finally, a matrix in the form  $[\alpha I \ \alpha A^{-1}]$  was produced. The matrix  $\alpha A^{-1}$  and coefficient  $\alpha$  were passed downstream. The last stage is the multiplication between matrix  $\alpha A^{-1}$  and  $H^H$ . The overall hardware structure of the beamforming matrix unit is shown below.

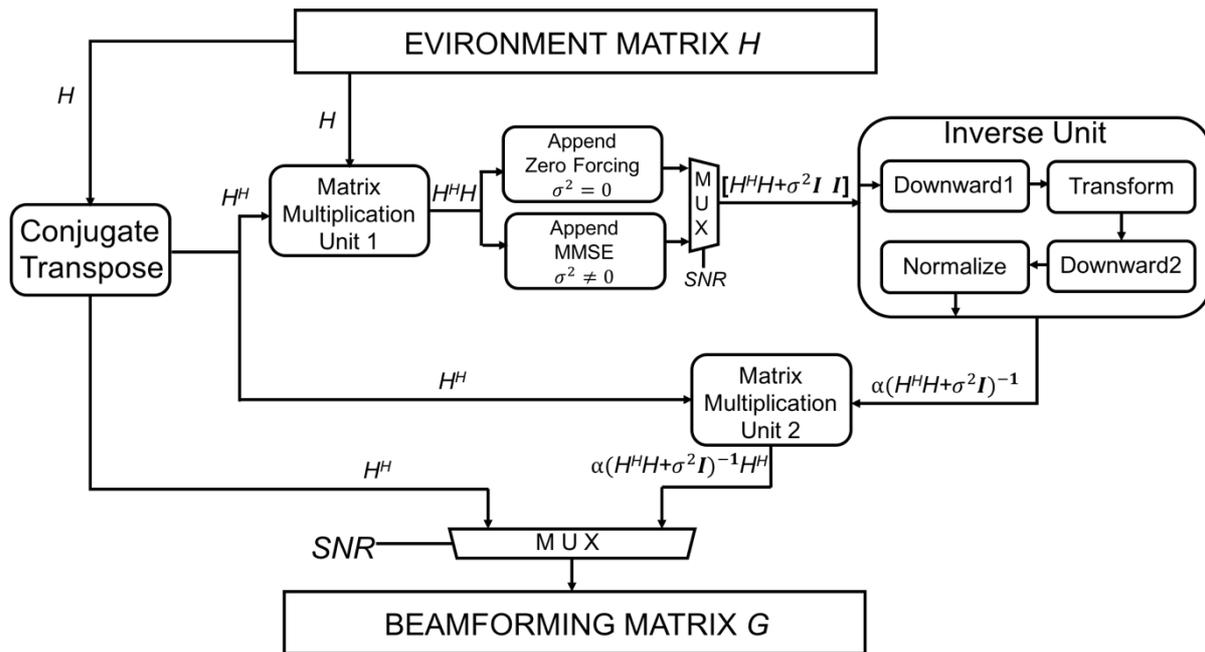


Figure 12 Overall hardware structure.

## 4. Simulation Results

Matrix operations can be easily verified by MATLAB. Up to now, the channel estimation block is still implemented by the other team. Therefore, environment matrix  $H$  was generated randomly by Rayleigh distribution in MATLAB as an emulation. Also, in MATLAB, a noise variance was generated for method MMSE. The same procedure was realized in MATLAB as that in ChiselDSP from multiplication to normalization step by step. By implementing three methods of beamforming matrix, corresponding results were found. In the Chisel tester, the functionality for a single beamforming matrix calculation using three different SNR values was individually tested. The simulation result showed that all conjugate beamforming, zero forcing, and MMSE functioned well. Then, the environment matrix  $H$  and SNR were updated when upstream told me the data were available and beamforming matrix block had already finished current calculation. The simulation result turned out that my beamforming matrix block successfully switched beamforming matrix calculation methods when SNR updated, and produced correct beamforming matrix  $G$ .

## 5. Conclusion

Beamforming matrix block is successfully in Chisel, including matrix transpose and conjugate, matrix multiplication unit, and matrix inversion. The functionality is verified by comparing simulation results with MATLAB results. Currently, the hardware structure of each unit is constructed by referring to Ph.D. students at BWRC and related papers. However, there is not enough information about parameterizable MIMO design. When the design specifications, like number of antennas, change, the hardware structure should change as well to a more efficient design. My implementation realizes generalization, but it may not explore tradeoff between space and time on hardware thoroughly. From the analysis above, the number of multiplications in each cycle is roughly balanced. However, Chisel's syntax restricts further

optimization. For example, the normalization process has to be finished in  $O(K^2)$  cycles, causing great inefficiency. Another issue is that, when the number of antennas is very small or very big, there will be some specific implementations increasing efficiency.

Currently, both Chisel and ChiselDSP are under development. Therefore, there sometimes occurs bugs like environment incompatibility, which requires developers to fix and improve. My generalized beamforming matrix block hardware structure can be improved. For future students, they can try to reorganize my structure, balance operation periods between registers and registers, and make it work under a higher frequency. Another issue is that, my block updates beamforming matrix much slower than Niral's matrix multiplication block, but we are using the same clock signal in the MIMO design. Future students can apply multiple clock domains and improve system performance. The last thing that has not been resolved is the determination of SNR. Since this value theoretically should be updated based on signal and noise strengths of the whole system in real time, we are only able to determine this value after block integration. For current design, I switch SNR value manually. Future students should determine SNR based on the performance of the integrated system.

# **TEAM PAPERS**

## **1. Introduction**

With the recent surge in wireless communications, the radio industry has seen an increase in the number of different communication standards, each requiring its own specific hardware and processing. Our project intends to address the need for radio interoperability with these various standards through the development of hardware generators for a Software-Defined Radio (SDR) system. These hardware generators will be created using Chisel (Bachrach 2012), a hardware construction language. When given a set of parameters or constraints, the hardware generators will output automated circuit designs for the given application, thereby accelerating the hardware design process and introducing a new methodology for multi-standard support. In this paper, we discuss topics relevant to bringing our project to market. These topics are divided into three sections: 1) the project's Intellectual Property (IP) approach, 2) the project's industry analysis, and 3) the project's market segment.

## **2. Trends and IP Strategy**

With recent advances in semiconductor technology, the Integrated-Circuit (IC) industry has experienced rapid growth over the past few decades (Ulama 2015:6-9). However, the industry is now starting to stagnate due to the increasing complexity required in designing chips to provide competitive functionalities within demanding constraints (Sangiovanni-Vincentelli 2007: 467-68). In particular, new opportunities rising in the consumer electronics and Internet of Thing (IoT) domains have made time-to-market the primary concern for IC companies due to first-mover advantages (Smith 2014). With the demand for shorter design cycles and higher volumes of functionality to be incorporated into designs, IC developers are facing costly project

delays because changes in project requirements often necessitate large loop iterations due to the sequential nature of current industry design methodologies (Sperling 2014).

In addition to the design flow challenges, IC developers are facing problems with the role of IP in the semiconductor industry. Given the increasing complexity of chips, it is too costly and slow to develop all the functionalities from scratch. Hence, IC designers rely on licensing reusable system building blocks from an external party, known as IP blocks (Tamme, et al. 2013: 221). While these IP blocks can accelerate design cycles, the primary issue arises during system integration and verification. When incorporating a supplier's IP block into the system, no guarantee exists that the IP block will interact with other system components to provide correct functionality. Since these IP blocks are "black boxes", verification and modifications to the IP block to meet the developer's need become difficult, thereby creating delays and long design cycles.

Our project intends to address these problems in the wireless IC domain as it aims to implement a new design paradigm based on Agile and platform-based schemes. The development of flexible hardware generators achieves this by facilitating initial chip design to be independent of specific processes or hardware implementations such as IP blocks. By raising the level of design abstraction towards the desired functionality rather than a specific implementation, large loop iterations can be avoided since system components can dynamically change with requirements.

In bringing this project to market, our IP approach must maximize the project's impact on the wireless IC domain. A patent approach is not suitable for a few reasons. First, the project is part of ongoing research at the Berkeley Wireless Research Center, which follows a non-patent

policy to encourage innovation. Second, the hardware generator design flow is based on Chisel, an open source language for creating circuit generators. Instead of obtaining a patent, we will be taking an open-source IP strategy to bring this technology to market. The primary motivation for this approach comes from the project's holistic goal of reshaping wireless IC design flows towards an Agile scheme to shorten design cycles and revive the growth of the IC and semiconductor industries. Taking an approach to protect the IP of this technology would only result in inhibited adoption of the new methodologies and limited growth of this new platform.

### **3. Industry Analysis**

Within the broader wireless industry, our capstone project targets two specific technologies: Wi-Fi and cellular data. These two industry sectors were chosen as they contain common characteristics and challenges that our project addresses.

The first common characteristic of Wi-Fi and cellular data network is that they are both widely used. Wi-Fi is becoming the standard Internet access method in various environments such as households, offices and public places (Henry 2002). Cellular data service is also reaching more and more people with the rapid development of the smartphone industry. As we are building a completely new platform for the wireless industry, choosing Wi-Fi and cellular data will allow us to maximize the number of potential developers who will benefit from the adoption of our hardware generators.

Secondly, both Wi-Fi and cellular data have development patterns consisting of rapid generation iterations and continuous improvement potential. Since the introduction of first generation Wi-Fi in 1997, it has evolved to fifth generation within 15 years (Nagarajan 2012).

Cellular data networks exhibit the same pattern, as the fifth generation is expected to be commercialized in the near future. These trends incentivize our design of flexible and parameterizable generators to reduce application redesign costs resulting from generation transitions.

The steep development curve and considerable future potential of Wi-Fi and cellular data networks have brought great challenges to the hardware design process. In the past, it would take engineers many years to design a series of new devices from scratch for each generation of wireless technology. This has delayed the new technology from reaching potential customers before the next generation emerges. In fact, some generations of the technology have suffered from a lack of supporting devices (Ferro 2005). Our project aims to ease this transition process by providing a flexible and generalized design framework. Our generators will consider the key factors that change between generations of technologies and will make them into parameters. Different hardware designs can then be produced by the generators, thereby reducing the development time for new device design and old device upgrade.

With the understanding of our industry above, we analyzed the five market forces (Porter 2008) on the cellular data industry to determine the profitability of entering the market. To be more specific, we are considering the market from the perspective of a hardware company that sells signal processing chips for smart phones.

First, the threat of new entrants would be weak. This is because the cellular data network industry greatly relies on technology, which makes it difficult to enter without substantial expertise of this area. New entrants would also struggle with the lack of credibility, which is essential for selling products to the customers in this industry. This leads to our second force,

the bargaining power of buyers. The buyers of our signal processing chips would be major mobile phone companies like Apple and Samsung. The size of these companies indicates their strong bargaining power, because they could compare the reliability, price, and performance of our product with many other alternative offers. The third force, threat of substitutes, is weak according to our analysis. Even though people can use Wi-Fi to connect to the Internet with their smartphones, the cellular data connection is an indispensable feature for any smartphone nowadays. Thus, there is almost no substitute technology. Fourth, the bargaining power of suppliers is also weak. The fabrication process for integrated circuit chips is standardized and many fabrication factories exist, thus allowing control of supplier costs. Lastly, the rivalry among existing competitors would be strong and feature-based. With the rapid development of wireless technology, the chip company that develops the first next-generation chip would obtain the biggest share of the market. Before other companies can catch up, enter the market and bring down the price, the industry might have already moved into the next generation.

As a whole, the three weak forces and a strong feature-based rivalry suggest promising profitability in this industry. Since our project would serve as a platform for this industry's developers, these results are great motivations for us.

## **4. Market Strategy**

According to the end-user industries, the SDR market is mainly subdivided into telecommunication, defense and public safety (Saha 2015). Considering SDR and Chisel, we focus our market segment on the telecommunication industries for a few reasons. First, our platform will be open-source, which heavily relies on a substantial contributor base. For commercialized industries like telecommunication, there are many engineers contributing to

the open source community. However, for other industries such as conventional defense and public safety, the aim of the communication system design is confidentiality and reliability rather than commercialization, so it is difficult to work on open source code. Second, the telecommunication industry has a big group of customers, so there will be extensive user feedback regarding the products which utilize our platform. Last, the competition among telecommunication industries is stronger than that in other industries. In order to obtain a competitive advantage in this market, companies are in great need of higher product quality and shorter design cycles, which can be achieved by using our hardware generators.

Before going to market, the users of our hardware generators must be defined. The two main categories of users that benefit from our project are university researchers and industry engineers. They are responsible for developing code, verifying it, and improving their design. University researchers can take advantage of the generators when designing new frameworks for the communication system. On the other hand, industry engineers can more effectively keep their designs up to date by using our generators, making it easier to go to market.

## References

- Arias-García, Janier, Ricardo Pezzuol Jacobi, Carlos H. Llanos, and Mauricio Ayala-Rincón  
2011 A suitable FPGA implementation of floating-point matrix inversion based on Gauss-Jordan elimination. In Programmable Logic (SPL), 2011 VII Southern Conference on, pp. 263-268.
- Bachrach, J., Vo, H., Richards, B., Lee, Y., Waterman, A., Avižienis, R., ... and Asanović, K.  
2012 Chisel: constructing hardware in a scala embedded language. Proceedings of the 49th Annual Design Automation Conference, ACM, 1216-1225.
- Berder, Emanuele De Carolis-Quentin, and Persefoni Kyritsi-Haibo Wang  
2008 MIMO/Beamforming for Multicast services in 4G scenario. Diss. Master Thesis, Aalborg University.
- Eilert, Johan, Di Wu, and Dake Liu  
2007 Efficient complex matrix inversion for MIMO software defined radio. Circuits and Systems, 2007. ISCAS 2007. IEEE International Symposium on.
- Ferro, E. and Potorti, F.  
2005 Bluetooth and Wi-Fi wireless protocols: a survey and a comparison. Wireless Communications, IEEE, 12(1):12-26.
- Henry, P. S., and Luo, H.  
2002 WiFi: what's next?. Communications Magazine, IEEE, 40(12):66-72.

Karkooti, Marjan, Joseph R. Cavallaro, and Chris Dick

2005 FPGA implementation of matrix inversion using QRD-RLS algorithm.

Proceedings of the 39th Asilomar Conference on Signals, Systems and Computers.

LaRoche, Isabelle, and Sebastien Roy

2006 An efficient regular matrix inversion circuit architecture for MIMO processing.

Circuits and Systems, 2006. ISCAS 2006. Proceedings. 2006 IEEE International Symposium on.

Nagarajan Vijay

2012 5G WiFi: Introducing a Wi-Fi Powerful Enough to Handle Next-Gen Devices and Demands.

<http://www.broadcom.com/blog/wireless-technology/5g-wifi-introducing-a-wi-fipowerful-enough-to-handle-next-gen-devices-and-demands/> , accessed October 18, 2015.

Porter, M. E.

2008 The five competitive forces that shape strategy. Harvard business review, 86(1):78-93.

Powell, Chuck

2014 Technical Analysis: Beamforming vs. MIMO Antennas

Puglielli, Antonio, Andrew Townley, Greg LaCaille, Vladimir Milovanovic, Pengpeng Lu, Konstantin Trotskovsky, Amy Whitcombe et al.

2015 Design of Energy-and Cost-Efficient Massive MIMO Arrays.

Rao, P. Venkata, and K. R. K. Sastry

2015 Implementation of Complex Matrix Inversion using Gauss-Jordan Elimination Method in Verilog. International Journal of Computer Applications 122.3.

Saha, Sudip

2015 FMI: Software Defined Radio (SDR) Market Analysis, Segments, Growth and Value Chain 2014-2020.

<https://www.newswire.com/press-release/fmi-software-defined-radio-sdr-market-analysis-segments-growth> , accessed February 27, 2016.

Sangiovanni-Vincentelli, Alberto

2007 Quo vadis, SLD? Reasoning about the trends and challenges of system level design. Proc. IEEE, 95(3):467-506.

Singh, Chitranjan K., Sushma Honnavara Prasad, and Poras T. Balsara

2007 VLSI Architecture for Matrix Inversion using Modified Gram-Schmidt based QR Decomposition. VLSI Design.

Smith, Randy

2014 Is IC Design Methodology At The Breaking Point? Semiconductor Engineering. <http://semiengineering.com/is-ic-design-methodology-at-the-breaking-point/>, accessed February 7, 2016.

Sperling, Ed

2014 Time To Market Concerns Worsen. Semiconductor Engineering.

<http://semiengineering.com/time-to-market-concerns-worsen/>, accessed February 7, 2016.

Tamme, S., S. Schott, D. Gunes, J. Wallace, R. Boadway, F. Razavi, and M. Pépin

2013 Trends And Opportunities In Semiconductor Licensing. les Nouvelles : 216-228.

Ulama, Darryle

2015 IBISWorld Industry Report 33441a Semiconductor & Circuit Manufacturing in the US. <http://www.ibis.com>, accessed October 18, 2015.

Veena, M. B., and MN Shanmukha Swamy.

2011 Implementation of Channel Estimation and Modulation Technique for MIMO System. International Journal of Wireless & Mobile Networks 3.2.