INTEGRATED CIRCUIT DESIGN FOR ANNULAR CAPACITIVE MICROMACHINED ULTRASONIC TRANSDUCER

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
MIDDLE EAST TECHNICAL UNIVERSITY

YUSUF KUTLUHAN KILIÇ

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF MASTER OF SCIENCE
IN
ELECTRICAL AND ELECTRONICS ENGINEERING

AUGUST 2015

Approval of the Thesis:

INTEGRATED CIRCUIT DESIGN FOR ANNULAR CAPACITIVE MICROMACHINED ULTRASONIC TRANSDUCER

Submitted by **YUSUF KUTLUHAN KILIÇ** in partial fulfillment of the requirements for the degree of **Master of Science in Electrical and Electronics Engineering Department, Middle East Technical University** by

Prof. Dr. Gülbin Dural Ünver _____
Dean, Graduate School of **Natural and Applied Sciences**

Prof. Dr. Gönül Turhan Sayan _____
Head of Department, **Electrical and Electronics Engineering**

Assoc. Prof. Dr. Barış Bayram _____
Supervisor, **Electrical and Electronics Eng. Dept., METU**

Examining Committee Members:

Prof. Dr. Cengiz Beşikçi _____
Electrical and Electronics Engineering Dept., METU

Assoc. Prof. Dr. Barış Bayram _____
Electrical and Electronics Engineering Dept., METU

Prof. Dr. Nevzat Güneri Gençer _____
Electrical and Electronics Engineering Dept., METU

Assist. Prof. Dr. Serdar Kocaman _____
Electrical and Electronics Engineering Dept., METU

Dr. Oğuz Altun _____
Aselsan AŞ

Date**:** **11.08.2015**

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Last name: Yusuf Kutluhan Kılıç

Signature:

ABSTRACT

INTEGRATED CIRCUIT DESIGN FOR ANNULAR CAPACITIVE MICROMACHINED ULTRASONIC TRANSDUCER

Kılıç, Yusuf Kutluhan

M. S., Department of Electrical and Electronics Engineering

Supervisor: Assoc. Prof. Dr. Barış Bayram

August 2015, 151 pages

The conventional ultrasound systems consist of many separated parts, and the bulky cables for connections result in a problem to be resolved. The systems with integrated interface electronics preferable since the removal of bulky cables resolves the signal degradation problems.

In this thesis, a flexible and high drive capability digitally controlled beamformer is designed. The beamformer is designed to be integrated with annular capacitive micromachined ultrasonic transducer (CMUT) arrays. Although the beamformer designed in this thesis is intended to be used with therapeutic ultrasound systems, it may be used as transmitter block of an ultrasound imaging system. The aim of the beamformer is to focus the acoustic waves generated by the CMUT elements

into a focal point to increase the energy at that point. The application of increasing this energy by focusing the acoustic waves is called high intensity focused ultrasound (HIFU).

The system is designed in high voltage 0.18 μm CMOS process which offers 45 V at the output. The system consists of power conservative 2x1 demultiplexers (8.8 μW), oneshot circuits that generate single pulses with adjustable pulse widths (30 μW), digitally controlled oscillators that generates pulse trains at variable frequencies (246.3 μW), frequency down converters (203.5 μW), 8x1 and 4x1 multiplexers (5.64 μW and 94 nW), medium voltage pulsers (313 μW), and high voltage pulsers (148.5 mW). The oneshot circuit generates output pulses with 2.5 ns resolution. The duration of the output pulses can be any value that is multiple of 2.5 ns. The digitally controlled oscillator generates pulse trains with frequencies in the range of 20 to 312 MHz. The high voltage pulser circuit can drive loads up to 10 pF. The figure of merit which is defined as (delay/(node length x voltage)) characterizes the performance of pulsers and it is 0.49 for the high voltage pulser.

In this work, 4 different beamformers are designed for 1x4, 4x1 and 4x4 annular CMUT arrays, and a test cell composed of all designed blocks is also laid out for economical testing purpose. The total chip area required for the test cell, 1x4, 4x1 and 4x4 cells are 6.68 mm2, 1.94 mm2, 4.07 mm2 and 30.73 mm2 respectively.

Keywords: CMOS, CMUT, Ultrasound, ASIC, therapeutic, VLSI, high voltage

ÖZ

DİSK ŞEKİLLİ KAPASİTİF MİKROÜRETİLMİŞ ULTRASONİK ÇEVİRGEÇLER İÇİN ENTEGRE DEVRE TASARIMI

Kılıç, Yusuf Kutluhan

Yüksek Lisans, Elektrik ve Elektronik Mühendisliği Bölümü

Tez Yöneticisi: Assoc. Prof. Dr. Barış Bayram

Ağustos 2015, 151 sayfa

Konvansiyonel ultrason sistemleri yarık parçalardan ve bir soruna dönüşen hantal kablolardan oluşmakradır. Entegre arayüz elektroniği olan sistemler hantal kabloların kaldırılması ile sinyal bozulmasını çözdüğü için tercih edilir sistemlerdir.

Bu tezde, esnek ve yüksek sürme kabiliyetli sayısal kontrol edilebilen hüzmeleyici entegre devre tasarlanmıştır. Hüzmeleyici dairesel kapasitif mikroüretilmiş ultrasonic çevirgeçler (CMUT) ile entegre edilmek üzere tasarlanmıştır. Bu tezde tasarlanan hüzmeleyici terapatik sistemler için tasarlanmış olsada, ultrasonik görüntüleme sistemlerinin göndermeç ksımı olarak da kullanılabilir. Bu hüzmeleyicinin amacı belirli bir odak noktasındaki hacmin enerjisini artırmak için CMUT elemanları tarafından üretilen akustik dalgaları bu odak noktasına odaklamaktır. Belirli bir noktanın akustik dalgaları o noktaya odaklayarak enerjisinin artırılması uygulaması yüksek yoğunlukta odaklanmış ultrason (HIFU) olarak adlandırılır.

Hüzmeleyici çıkışta 45 volt sunan yüksek voltaj 0.18 mikron CMOS işleminde tasarlanmıştır. Bu devre güç tasarrufu sağlayan ters çoklayıcılar (8.8 μW), ayarlanabilir darbe genişliği ile darbe üreten tek darbe devreleri (30 μW), değişken frekanslarda darbe katarı üretebilen ve sayısal kontrol edilen osilatörler (246.3 μW), frekans aşağı çeviricileri (203.5 μW), 8x1 ve 4x1 çoklayıcılar (5.64 μW and 94 nW), orta gerilim (313 μW) ve yüksek gerilim (148.5 mW) darbe üreteçlerinden oluşmaktadır. Tek darbe devresi 2.5 ns çözünürlükte sinyal üretebilir. Darbelerin uzunluğu 2.5 ns katı olan herhangi bir değer olabilir. Darbe katarı üreten devre 20 ila 312 MHz frekans aralığında katarlar üretebilir. Yüksek gerilim darbe üreteci 10 pF seviyesinde yükleri sürebilmektedir. Darbe yükselteçlerinin verim karakterini ortaya koyan ve (gecikme/(düğüm uzunluğu x gerilim)) olarak tanımlanan liyakat figürü bu devre için 0.49 olarak hesaplanmıştır.

Bu çalışmada, ekonomik test amaçlı ve bütün blokları içeren bir test hücresi ve 1x4, 4x1 ve 4x4 dairesel CMUT dizileri için olmak üzere 4 adet hücre tasarlanmıştır. Test hücresi, 1x4, 4x1 ve 4x4 CMUT için tasarlanan hücreler sırasıyla 6.68 mm2, 1.94 mm2, 4.07 mm2 ve 30.73 mm2 alana oturtulmuşlardır.

Anahtar Kelimeler: CMOS, CMUT, Ultrason, ASIC, terapatik, VLSI, yüksek voltaj

To my parents, family and lovely one

ACKNOWLEDGEMENTS

Hereby, I would like to give my best thanks to my thesis Supervisor, Assoc. Prof. Dr. Barış Bayram for all his support, guidance and understanding during this thesis work.

I would like to thank to my jury members for their understanding attitudes and contributions to thesis work.

I would also express my gratitude to the friends in ULTRAMEMS Research Group for their kind contributions.

In the end, I would like to give my special thanks to my parents, grandparents, family, big brother and fiancee who helped and supported me during my studies in METU.

TABLE OF CONTENTS

ABSTRACT v
ÖZ vii
ACKNOWLEDGEMENTS x
TABLE OF CONTENTS xi
LIST OF FIGURES xiv
LIST OF TABLES xviii
LIST OF ABBREVIATIONS xix
1.INTRODUCTION 1
1.1 Ultrasound Theory 3
1.2 Therapeutic Ultrasound 4
1.3 Micromachined Ultrasonic Transducers (MUTs) 6
1.3.1 Capacitive Micromachined Ultrasonic Transducers (CMUTs) 6
1.4 Front-end Electronics for MUTs 9
1.4.1 Electronics of CMUTs 10
1.5 Objectives of the Thesis 13
1.6 Outline of the Thesis 15
2. RECEIVER ELECTRONICS FOR CAPACITIVE MICROMACHINED ULTRASONIC TRANSDUCERS (CMUTs) 17
2.1 Receiver System Overview 18
2.2 Transmitter/Receiver Switch 20
2.3 Low-Noise Amplifier 21
2.3.1 TIAs with Resistive Feedback 22
2.3.2 TIAs with Capacitive Feedback 24
2.4 Buffer 25
2.5 Integration of CMUT and Front-end Electronics 28
2.6 Summary of the Chapter 29

3. THE DESIGN OF FRONT-END ELECTRONICS FOR ANNULAR CMUT ARRAYS IN HV 0.18 μm CMOS TECHNOLOGY 31
3.1 The Overview of the Beamformer ASIC 33
3.2 Architecture of the Beamformer ASIC 33
3.3 FPGA Algorithm 36
3.4 Two Input Demultiplexer 37
3.5 Circuits Composed of Delay Elements 40
3.5.1 Oneshot Circuit 43
3.5.2 Digitally Controlled Oscillator 48
3.6 Frequency Down Converter 55
3.7 Multiplexers 58
3.8 Buffer 60
3.9 Medium Voltage Pulser 61
3.10 High Voltage Pulser 64
3.11 CMUT Element DC Biasing 69
3.12 Summary of the Chapter 70
4. DESIGN AND PERFORMANCE OF THE FRONT-END ELECTRONICS FOR CMUTs 73
4.1 Overview of Layout 74
4.2 Layout of the Low Voltage Block 76
4.2.1 Demultiplexer 77
4.2.2 Oneshot Circuit 78
4.2.3 Digitally Controlled Oscillator 78
4.2.4 Frequency Down Conversion Circuit 81
4.2.5 Multiplexers 81
4.2.6 Buffer 83
4.2.7 Medium Voltage (MV) Pulser 84
4.3 Low Voltage Block (Driver) Assembly 85
4.4 High Voltage Pulser 86
4.5 Bond Pads 89
4.6 I/O and Supply Structures 91
4.7 Test Cell of the Beamformer 93
4.8 Beamformer ASIC for 4x1 Element Annular CMUT Array 95

4.9 Beamformer ASIC for 1x4 Annular CMUT Array 96
4.10 Beamformer ASIC for 4x4 Annular CMUT Array 98
4.11 Simulation Results of the Layouts 99
4.12 Summary of the Chapter 104
5. FPGA SETTING ALGORITHM FOR ANNULAR CMUT ARRAYS .. 107
5.1 Theory of the Program 107
5.2 Program Flow 108
5.3 Summary of the Chapter 113
6. CONCLUSIONS AND FUTURE WORKS 115
REFERENCES 121
APPENDIX A 128
MATLAB PROGRAM DATA 128
FPGA PROGRAM DATA 143

LIST OF FIGURES

FIGURES

Figure 1.1 Spectrum of the acoustic wave..... 3
Figure1.2 Theory of Therapeutic Ultrasound..... 5
Figure1.3 Coagulation necrosis..... 5
Figure 1.4 Cross-sectional view of CMUT fabrication steps..... 8
Figure 1.5 CMUT under DC bias..... 9
Figure 1.6 Simplified block diagram of interface electronics..... 10
Figure 1.7 An interface for CMUT..... 11
Figure 1.8 Integration of CMUTs and ASICs..... 12
Figure 1.9 Flip-chip bonding sample..... 13
Figure 2.1 System level overview..... 18
Figure 2.2 System level overview..... 19
Figure 2.3 CMUT Model..... 19
Figure 2.4 The T/R switch..... 20
Figure 2.5 DMOS switch..... 21
Figure 2.6 The differential pair based TIA with resistive feedback..... 22
Figure 2.7 TIA with common-source followed by source-follower..... 23
Figure 2.8 Cascode amplifier in resistive feedback TIA..... 24
Figure 2.9 Capacitive feedback TIA based on common-source followed by source-follower..... 24
Figure 2.10 Capacitive feedback TIA based on differential pair..... 25

Figure 2.11 Output Buffer (b) 26
Figure 2.12 Variable gain amplifier 26
Figure 3.1 Overview of the beamformer ASIC 33
Figure 3.2 The Architecture of the system 34
Figure 3.3 Block diagram of the ASIC 35
Figure 3.4 Inverter in the demultiplexer 38
Figure 3.5 NAND gate in the multiplexer 38
Figure 3.6 Configuration of the demux 39
Figure 3.7 Simulation results of the demultiplexer 39
Figure 3.8 Delay element model 41
Figure 3.9 A simplified version of the delay element 42
Figure 3.10 The configuration of oneshot circuit 44
Figure 3.11 Oneshot outputs when input is 10 ns 45
Figure 3.12 Oneshot outputs when input is 40 ns 45
Figure 3.13 Oneshot simulation when input is 40 ns and input vector is 100 46
Figure 3.14 Supply variations of oneshot 47
Figure 3.15 Temperature variations of oneshot 47
Figure 3.16 Block diagram of the DCO 48
Figure 3.17 Simulation results of the enable circuit 49
Figure 3.18 Schematic of the DCDE 50
Figure 3.19 Supply variation simulation of DCDE 51
Figure 3.20 Simulation result of DCDE with input vector 00011 53
Figure 3.21 Input vector vs frequency chart of DCO 53
Figure 3.22 Input vector vs pulse train period of DCO 54
Figure 3.23 Linear curve fitted to DCO frequency chart 54
Figure 3.24 D flip-flop 55
Figure 3.25 3-input NAND gate 56
Figure 3.26 Block diagram of the FDC 56
Figure 3.27 Simulation result of FDC with first 4 outputs 57
Figure 3.28 Simulation result of FDC with last 4 outputs 58
Figure 3.29 Transmission gate 58
Figure 3.30 4-input multiplexer 59

Figure 3.31 Simulation results of 4-input multiplexer 59
Figure 3.32 Schematic of the buffer 60
Figure 3.33 Block diagram of the MV pulser 61
Figure 3.34 Simulation results of the medium voltage pulser 62
Figure 3.35 Temperature variation simulation of the MV pulser 63
Figure 3.36 Supply variation simulation of the MV pulser 63
Figure 3.37 HV Pulser configuration 65
Figure 3.38 Cross-sectional view of HV transistors 66
Figure 3.39 Simulation results of the HV pulser (10 MHz) 67
Figure 3.40 Temperature sweep simulation of the HV pulser 67
Figure 3.41 Simulation results of HV pulser with 5.1 pF load (10 MHz) 68
Figure 3.42 Simulation results of HV pulser with 10 pF load (10 MHz) 68
Figure 3.43 DC biasing scheme of CMUT elements 69
Figure 4.1 4 element annular CMUT array 74
Figure 4.2 4x4 element annular CMUT array 75
Figure 4.3 Overview of the IC 76
Figure 4.4 Demultiplexer layout 77
Figure 4.5 Oneshot circuit layout 78
Figure 4.6 Enable of DCO 79
Figure 4.7 Digitally controlled delay element layout 80
Figure 4.8 Digitally controlled oscillator layout 80
Figure 4.9 Frequency down conversion circuit layout 81
Figure 4.10 4-input multiplexer layout 82
Figure 4.11 8-input multiplexer layout 82
Figure 4.12 Buffer layout 83
Figure 4.13 MV Pulser layout 84
Figure 4.14 Driver layout 85
Figure 4.15 Driver layout without layer definitions 85
Figure 4.16 HVNMOS (left) and HVPMOS (right) 87
Figure 4.17 HV Pulser layout 88
Figure 4.18 Pad layout 89
Figure 4.19 250µm x 250µm pads for 4 element CMUT array 90

Figure 4.20 Pad orientation of 4x4 element CMUT array 90
Figure 4.21 Typical I/O rail..... 91
Figure 4.22 HV ESD supply layout 92
Figure 4.23 Output cell diagram 92
Figure 4.24 Input Cell diagram 92
Figure 4.25 Layout of the test cell..... 94
Figure 4.26 Layout of the ASIC for 4 element array 95
Figure 4.27 Connections between LV, HV and pads..... 96
Figure 4.28 Layout of the ASIC for 1x4 array..... 97
Figure 4.29 Layout of the ASIC for 4x4 array..... 98
Figure 4.30 Connection of pad and HV pulser..... 99
Figure 4.31 Simulation result of oneshot layout 100
Figure 4.32 Simulation result of the DCO layout 101
Figure 4.33 Simulation result of the MV Pulser 101
Figure 4.34 Simulation result of HV Pulser..... 102
Figure 4.35 Top level simulation results..... 102
Figure 4.36 Top level simulation results..... 103
Figure 5.1 CMUT phasing program..... 108
Figure 5.2 FPGA settings window without any choice..... 108
Figure 5.3 FPGA settings window with oneshot choice 109
Figure 5.4 FPGA settings window with DCO choice 109
Figure 5.5 Oneshot settings window..... 110
Figure 5.6 DCO settings window..... 110
Figure 5.7 How to select CMUT array type..... 111
Figure 5.8 FPGA and RS-232 Configuration..... 112
Figure 5.9 16 clock cycle latency..... 112
Figure 5.10 20 clock cycle latency..... 113
Figure 5.11 MAX-232 circuit configuration..... 113

LIST OF TABLES

TABLES

Table 2.1 Receiver Specifications……....27

Table 2.2 Summary of the Integration Methods……...28

Table 3.1 Comparison of similar oneshot circuits……46

Table 3.2 Delay values of DCDE for some input vector combinations……52

Table3.3 Transistor sizes of buffer inverters…….60

Table 3.4 Transistor sizes of MV pulser inverters……62

Table 3.5 Comparison of similar HV Pulsers……...69

Table 4.1 Cell sizes and densities……...104

Table 5.1 Device List for RS-232……...111

LIST OF ABBREVIATIONS

ASIC	Application Specific Integrated Circuit
CMUT	Capacitive Micromachined Ultrasonic Transducer
DCO	Digitally Controlled Oscillator
DEMUX	Demultiplexer
FDC	Frequency Down Converter
HIFU	High Intensity Focused Ultrasound
HV	High Voltage
IC	Integrated Circuit
I/O	Input/Output
LV	Low Voltage
MUT	Micromachined Ultrasonic Transducer
MUX	Multiplexer
MV	Medium Voltage
PMUT	Piezoelectric Micromachined Ultrasonic Transducer
FPGA	Field Programmable Gate Array

CHAPTER 1

1. INTRODUCTION

The ultrasound systems and technologies have been improved and become easy-to-use for a variety of fields by the help of researches and developments in the industry. The systems relying on ultrasound technologies are utilized for detecting objects and measuring distances in general, such as car parking systems, liquid level measurements and medical imaging systems. In addition to these application fields, nondestructive testing, cleaning, mixing and accelerating chemical processes are other application areas. During these developments in ultrasound technologies, microfabrication processes have been also growing. The developments in the Micro Electromechanical Systems (MEMS) and Integrated Circuit (IC) fabrication processes allows the designer to use combinations of instances of those technologies in a variety of fields such as medical applications that ease our life [1]. The instances of MEMS technology such as sensors, transducers and actuators are combined with Application Specific Integrated Circuits (ASIC) such as beam-formers, transceivers and drivers to be used in the application stated above. Especially the medical imaging and medical therapy systems utilize those combinations. By having high quality microfabricated devices, sonography systems, 3D imaging systems and therapeutic systems can be designed.

The medical imaging systems consist of transceiver system, signal processing unit and a display basically [1]. In the transceiver part, the signals that are to be sent to target object are formed in the transmitter block and the returning signals are detected or received in the receiver block. In ultrasound systems, the transmitter of the system generates the ultrasound signal and sends it to the target

while the receiver of the systems detects the acoustic signals reflected from the material surfaces. These reflected waves are called as echoes. By processing received echoes, 2D or 3D images are formed depending on the system. The ultrasound transducer that generates and receives the acoustic wave converts that electrical energy into mechanical energy and vice versa. The difference between the ultrasound imaging systems and ultrasound therapeutic systems are the level of transmit signals. In therapeutic systems, the transducers are excited with high potential signals such as 45 V or 90 V. These high potential signals are harmless while passing through tissues between the transducer and the target. The aim of those systems is to focus that high potential acoustic waves into a point and quickly increase the mechanical energy at that point. Those systems that focus high potential ultrasound waves at a point are called as High Intensity Focused Ultrasound (HIFU) systems. Recently, HIFU systems using PZT-8 and PZT-4 (piezoelectric materials in transducers) in the 1-4 MHz range are used in therapeutic applications [1].

In today's world, many people have experienced an ultrasound imaging session at least once and know how the system functions up to a certain degree. Moreover, there are number of researches and detailed information about ultrasound imaging systems and the consisting parts due to the common usage and applications of the system. However, there are not many researches on design or implementation of therapeutic ultrasound. A few research groups researches and contributes to the design and fabrication of HIFU beam-formers [9].

The aim of this thesis is to design a digitally controllable and high drive capability beam-former ASIC for Capacitive Micromachined Ultrasonic Transducers (CMUTs) to be used in therapeutic ultrasound applications. The aim of this chapter is to give a brief introduction on ultrasound theory, therapeutic application fundamentals, ultrasound transducers and corresponding beam-formers. Section 1.1 gives the summary of ultrasound theory, Section 1.2 describes the fundamentals of therapeutic usage by summarizing the focusing theory. Section 1.3 gives an introduction on Micromachined Ultrasonic Transducers (MUTs) by describing design procedures and functioning of Capacitive Micromachined Ultrasonic Transducers (CMUTs) in subsection 1.3.1.

Section 1.4 describes the fundamentals of beam-formers of transducers by declaring design issues and overviews in subsections 1.4.1 and 1.4.2 for PMUTs and CMUTs, respectively. Section 1.5 describes the aim of the thesis by listing issues that should be considered and achieved throughout the thesis work. Finally, Section 1.6 concludes this chapter by summarizing the outline of the thesis.

1.1 Ultrasound Theory

The sound is an energy of the wave form same as the light. The difference between the light waves and sound waves is that the light do not necessarily need any medium to propagate, it can propagate in vacuum also. However, the sound waves need an elastic medium such as solids, liquids or gases in order to continue travelling. A similarity between light waves and acoustic waves (sound waves) is that both of them have a propagation frequency. Depending on that propagation frequency, the sensors on systems or creatures can sense the acoustic waves. Figure 1.1 illustrates the spectrum of acoustic waves, which mainly divided into three sections as infrasound region, audible region and ultrasound region [3].

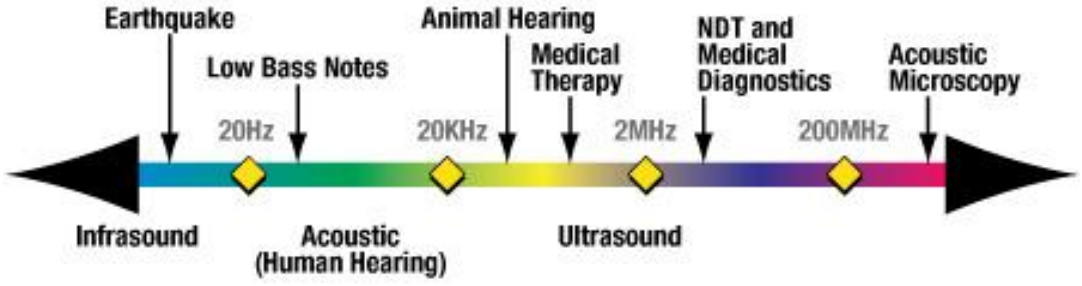

Figure 1.1 Spectrum of the acoustic wave [3]

In general, the acoustic waves in the range of 1-100 MHz are preferred for medical application. Different frequencies are preferred for different applications. For instance, the frequencies higher than the medical usage range are used for non-destructive testing applications. Besides, acoustic waves carry energy like the light waves. Energy of an acoustic wave depends on several factors such as

the transducer generating that wave, firing energy of the transducer and the medium that the wave travels inside. Acoustic waves are reflected by each medium surface while trying to pass inside and these reflections are called as echoes. These echoes are used in imaging system.

1.2 Therapeutic Ultrasound

The ultrasound systems has broad usage area in medicine for diagnostic and assistance reasons for a long time. By the help of ultrasound imaging systems, diagnostics of diseases and assistance to other invasive or non-invasive systems are possible. However, the use of ultrasound for therapeutic reasons is being discussed and investigated nowadays. The possibility of using high intensity focused ultrasound in order to treat some neurological disorders is first investigated by Lindstrom [5] and Fry et al. [6] at 1954. During these investigations, Fry et al. [6] noticed that a magnifying glass can be used to focus ultrasound waves similar to the light focusing. The ultrasound was again investigated in 1970s for treating tumors at lower intensities [7]. The idea was to increase the temperature of the target tissue up to 43ºC and keep that part of tissue for a time. However, measuring the temperature with non-invasive systems at focal point was not possible for those times. As a result, the idea of destruction of tumors by heating them up was postponed. In addition to those ideas, in 1984, the idea of kidney stone treatment by ultrasound was approved [6]. In 1990s, a rediscovery of ultrasound usage for tumor treatment occurred by the help of MR thermometry this time [8]. Following those ideas, more clinical experiments and investigations are done in last decades.

The difference between diagnostic and HIFU ultrasound is the intensity of the signal. HIFU has higher intensity wave accumulation in the focal point. The effect of high acoustic intensities in the focal point (part of the tissue) is the rapid heat generation owing to acoustic energy absorption by incident waves [8]. Figure1.2 illustrates the usage and theory of therapeutic ultrasound.

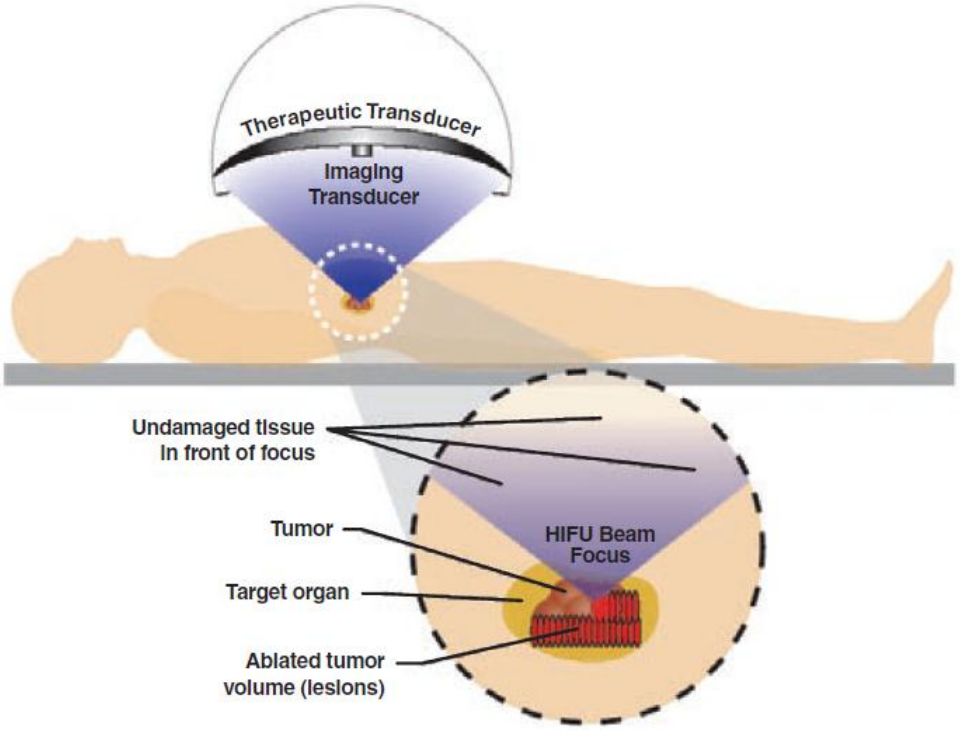

Figure1.2 Theory of Therapeutic Ultrasound [8]

Figure 1.3 shows the result of rapid heating in tissue as a result of incident waves.

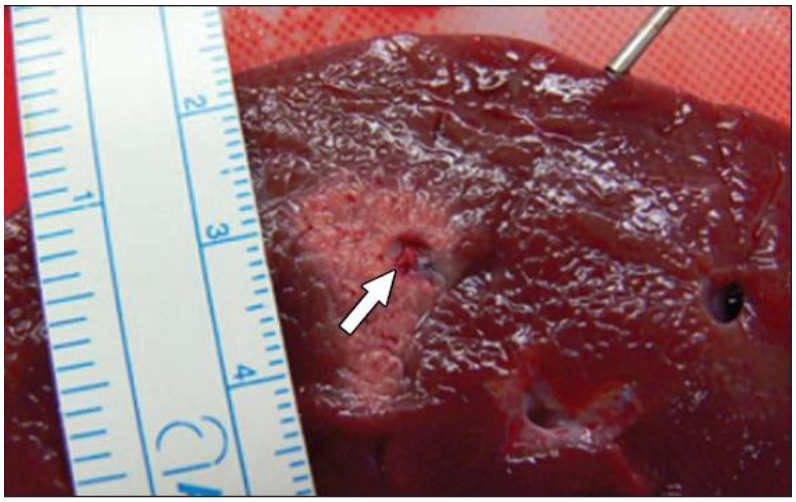

Figure1.3 Coagulation necrosis [8]

The result of this rapid heating event is called as coagulation necrosis. In modern systems, the waves are focus into a point by the help of beam-formers instead of magnifying glasses.

1.3 Micromachined Ultrasonic Transducers (MUTs)

A MUT element is a transducer which converts mechanical energy into electrical energy or vice versa. The mechanical energy is in the form of acoustic wave. Generally, MUTs have one fixed plate and one flexible plate which is moving according to the applied signal or wave. Basically, this flexible plate is the source of conversion from acoustic wave to electrical signal or vice versa. However, not all type of signals generate corresponding waves. For instance, a DC signal can attract the flexible plate of the transducer, but it cannot cause a oscillatory motion of the plate that generated the acoustic wave. Briefly, under a certain DC bias, the transducer can convert an AC signal into an acoustic wave, or an acoustic wave into an AC signal.

The ultrasound systems utilize many transducer elements to form the required signal. As a result, the probes are formed of many MUTs that are called as arrays or phased arrays. Although a probe has a significant size, a MUT element is at microscopic sizes. The fabrication processes limits the reduction in size of MUTs used in medical applications [9]. The performance of the transducer in terms of bandwidth and sensitivity is determined by the fabrication process, since the sizes and specifications are affected by the process. Due to the fact that each transducer element has its own bandwidth and sensitivity, the application field is also affected by the process. For instance, the transducers used in medical imaging and non-destructive testing have different specifications and fabrication process. In following subsections, the Piezoelectric Micromachined Ultrasonic Transducers (PMUTs) and Capacitive Micromachined Ultrasonic Transducers (CMUTs) are briefly described.

1.3.1 Capacitive Micromachined Ultrasonic Transducers (CMUTs)

The systems based on PMUTs have bulky connections and bandwidth limitation that should be resolved. In 1990s, a new type of transducer was announced to overcome those problems meet the challenging demands of the industry [10]. Although traditional piezoelectric transducers are commonly used as ultrasound probes, CMUTs have demonstrated some advantages such as easy fabrication, compact integration with the interface electronics, better bandwidth and better

high power performance [11]. Another advantage of CMUTs over piezoelectric counterparts is that CMUTs can be designed with a conductive silicon membrane as a top electrode which eliminates the need of metal layer on the surface of the cell [11]. In addition, the heating problem in piezoelectric transducers limits the usage of those type of transducers in high power applications, especially for therapeutic applications. Due to the fact that the silicon itself is highly thermally conductive material, CMUTs reduces the self-heating problem experienced in piezoelectric counterparts [12].

A CMUT is vacuum gap capacitor which has one fixed plate and one flexible plate that bends under DC bias application [13]. The electrostatic force causes the flexible plate to deflect toward the fixed one, as a result, the smaller gap causes a higher electric field that increases efficiency of the transducer [13]. In addition to DC bias, an AC signal is applied to the other plate (other electrode) of the capacitor to force the flexible plate to begin oscillation. This oscillatory movement results in generation of the acoustic wave. Similarly, the incident acoustic waves (ultrasound) causes the flexible plate to begin oscillation and that movement of the plate is sensed as capacitance change [13]. Moreover, the collapse voltage is the one of the most important parameters of a CMUT. Collapse voltage is the voltage level that the mechanical restoring force of flexible plate (in general membrane) can no more balances the electrostatic attraction force caused by DC bias [14]. Until this critical DC bias level, the membrane keeps its flexibility, however, any DC bias higher than that level causes membrane to move further toward the fixed plate (substrate) and collapse. The fabrication steps and details of the CMUT shown in Figure 1.4 are described in [15] and the figure shows the feasibility of construction of CMUT and CMOS on the same wafer. The typical value of the plate dimension is in the order of tens of micrometers and the distance between plates (gap) is in the order of tens or hundreds of nanometers [16]. Similar to the piezoelectric counterparts, those dimensions are effective on the parameters of the CMUT such as bandwidth, sensitivity and others.

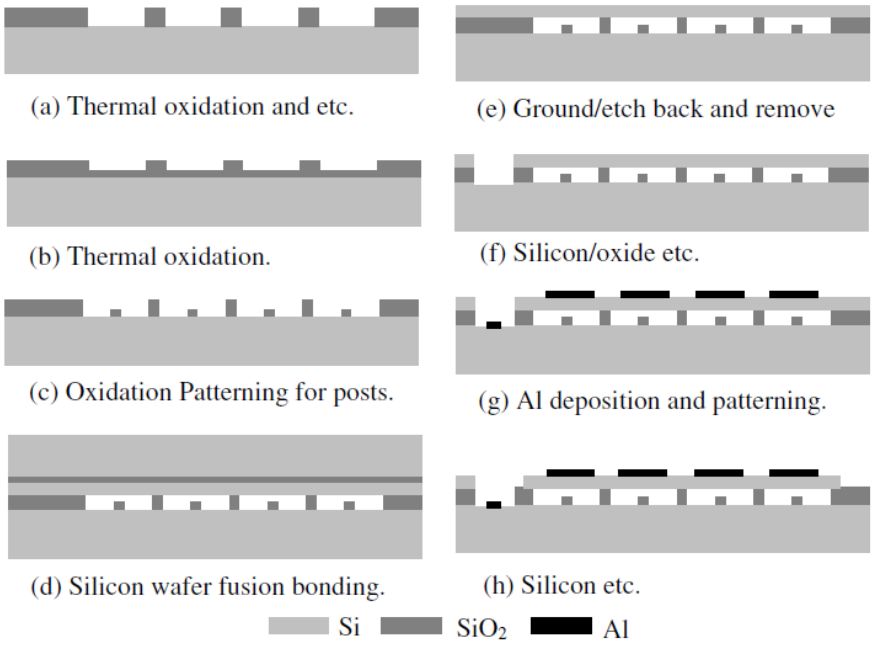

Figure 1.4 Cross-sectional view of CMUT fabrication steps [15]

The cross-sectional view of a CMUT that has its flexible plate attracted towards the substrate to the operation point is shown in Figure 1.5. As can be seen through figures, the connection with CMUTs may differ from design to design. The CMUT in [17] does not have aluminum contacts but the layer itself is highly conductive.

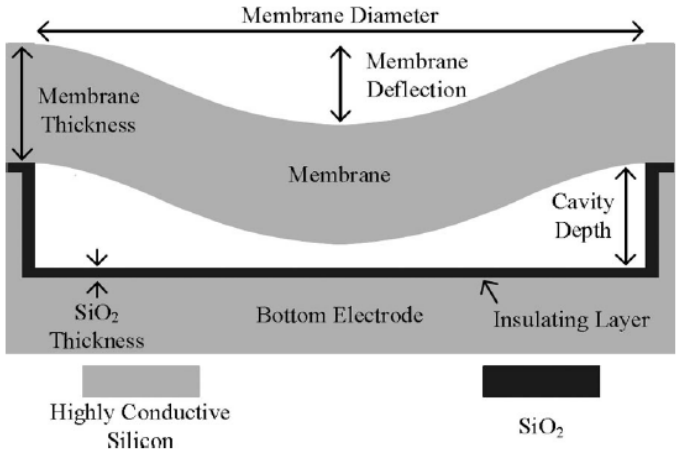

Figure 1.5 CMUT under DC bias [17]

There are several methods to calculate the collapse voltage of the CMUT, but the most accurate one is the Finite Element Method (FEM) [18-19]. Besides, CMUTs can be operated in different mode such as conventional, collapsed and collapsed-snapback mode [20].

There are several processes for microfabrication of the CMUT elements. The most popular ones are the sacrificial release and wafer bonding process. In sacrificial process, a sacrificial layer is deposited at the position of gap and it is removed by selective etchants after the surrounding layers are formed. As a result, the gap is formed. In wafer bonding process, the CMUT is formed by the combination of two wafers one on top of the other one. The fabrication processes are detailed and described in [21].

1.4 Front-end Electronics for MUTs

In order to drive the ultrasound transducer elements in various application, design of interface electronics is a must. For instance, transmitter and receiver blocks are necessary for a medical imaging system. Moreover, a beam-former that focuses waves in to a focal point is necessary for a therapeutic ultrasound system. As many of the researchers know, majority of the ultrasound systems used in today's world is composed of several items such as transducer probe, signal processing unit. However, the transducer probe is not the source of transmit signals and it does not process the received echoes. Instead, the signal is transmitted to probe and received by the probe via bulky cables. As a result of the utilization of those bulky cables, the signal to be transmitted or received is attenuated due to the parasitic capacitance introduced by that bulky cables. This problem increases the system design complexity and the way to resolve this problem is to place the front-end electronics very close to the transducer [22]. There are several ways to design front-end electronics very close to the transducers and those will be discussed in following sections. Figure 1.6 illustrates a simple interface electronics block.

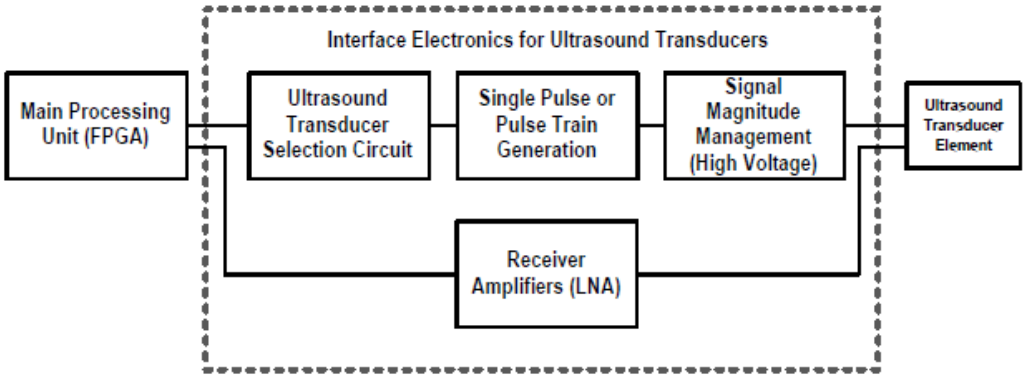

Figure 1.6 Simplified block diagram of interface electronics [2]

The receiver block in Figure 1.6 is necessary for the interface designed for an imaging system, but not required by a beam-former designed for therapeutic applications. The selection circuit contains some logic circuits that fires MUT elements with the corresponding firing times. However, the transducer selection algorithm is realized in FPGA that drive the beam-former in this thesis. The single pulse circuit generated variable width pulses and pulse train generator is an oscillator that generates continuous waves. Signal magnitude management block level shifts the low voltage signals to the required high voltage level. In imaging systems, the required receiver block is a transimpedance amplifier that gets the current as input and gives an amplified voltage output, generally. The processing unit is an FPGA in general, but any other high capability processor might be used. The connection with FPGA is done through the I/O connection blocks. The connection between the electronics and the transducer can be done by flip-chip bonding pads or wire bonding.

1.4.1 Electronics of CMUTs

As discussed in above sections, the ultrasound transducers needs interface electronics for various applications. The theory of interface electronics for CMUTs is similar to the PMUTs. Depending on the application, CMUTs also require receive blocks or focusing blocks, but the signal generation and shifting the level of the output signal to the high voltages are the common requirements. For the receive block, transimpedance amplifiers are preferred generally. The

incident acoustic waves change the capacitance of the CMUT element as stated earlier. This capacitance change results in a current change at the output. An amplifier which converts current into voltage while amplifying signal is required, as a result, the transimpedance amplifier is the best fitting option. In order to focus acoustic waves into a point, several ways can be preferred. Generally, combination of a shift register and a comparator is utilized. The processing unit loads a comparison value to comparator and the required latency value to shift register. However, especially for small arrays having less elements, direct feeding from processing unit is an alternative way. For the signal generation, the researchers designed an oneshot circuit that generates pulses with variable duration for their system in therapeutic applications. However, for volumetric and 3D imaging, a continuous wave generator is also necessary. Figure 1.7 illustrates a typical interface for a CMUT.

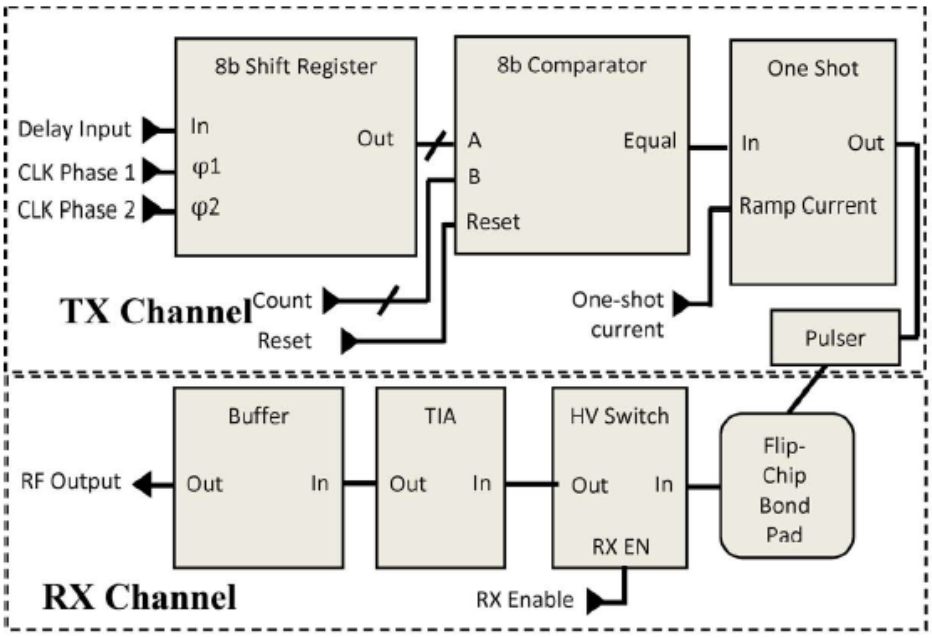

Figure 1.7 An interface for CMUT [22]

There are two main ways to integrate interface electronics with the CMUTs which are monolithic and multichip integration methods [21]. The monolithic integration method can be divided into three groups: Co-processing, post-processing by low temperature surface micromachining and post-processing by

low temperature direct wafer bonding [21]. The multichip integration method is also divided into two groups: chip-to-chip bonding and bonding on intermediate substrates [21]. The detailed information about the advantages, constraints and specifications can be found in [21]. Figure 1.8 illustrates an instance of integration of CMUTs and interface electronics. In the left side of the figure, there is an interposer which establishes the communication between 32x32 CMUT array and four 16x16 ASICs. The right side of the figure shows the cross-sectional view of the assembly. As can be seen through the figure, there is no connection cable and no additional parasitic capacitance attenuation is experienced. Besides, this type of configuration decreases the size of the system elements.

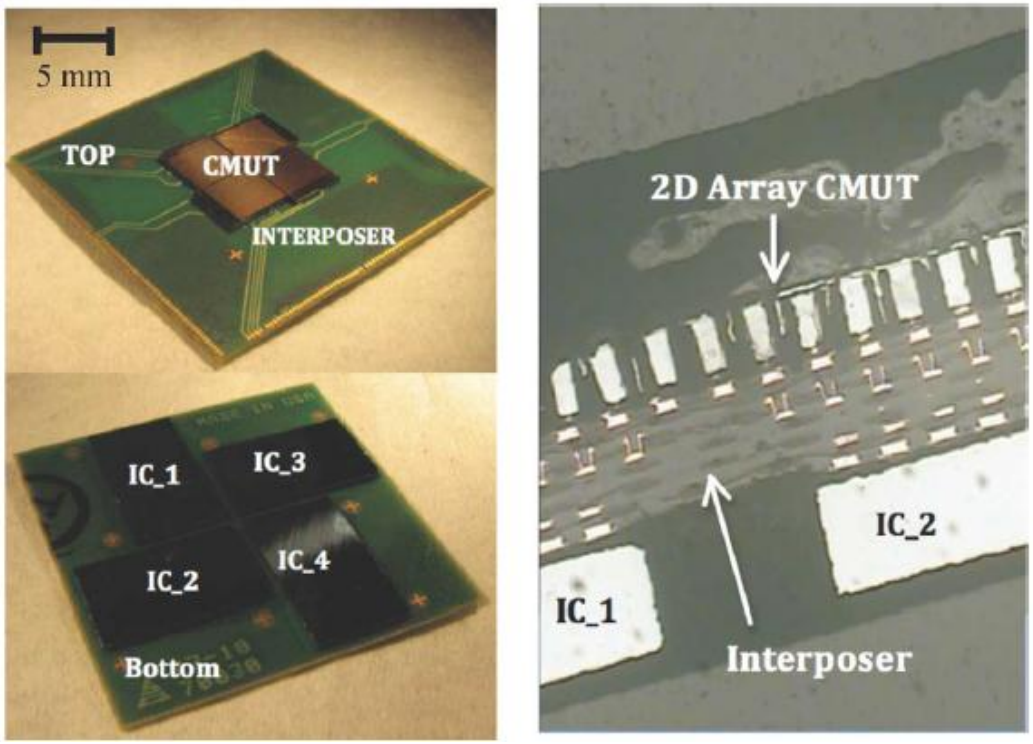

Figure 1.8 Integration of CMUTs and ASICs [22]

In Figure 1.9, a sample view of flip-chip bonding method is illustrated.

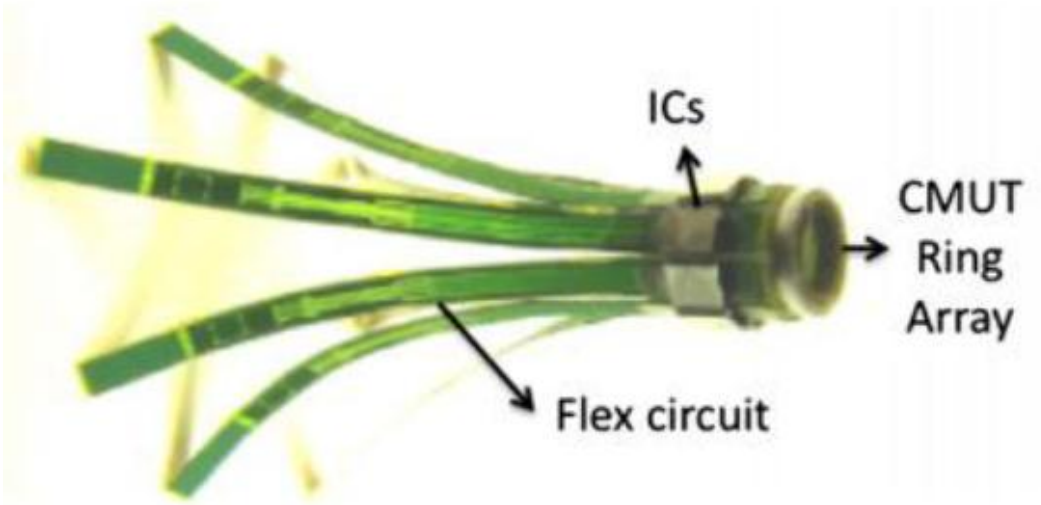

Figure 1.9 Flip-chip bonding sample [21]

In [45], the integration methods are discussed in detail in terms of array structures. As can be understood from the previous statements, CMUT arrays can be constructed as 1-D and 2-D arrays. In case of 1-D arrays, the integration with the electronics can be easily provided. In this case, wire bonding technique can be employed due to its advantages over other techniques such as non-limiting the CMUT fabrication process. However, in case 2-D arrays are preferred, the integration with the wire bonding gets harder. The flip-chip bonding technique is more suitable in this case. As a result, the front-end electronics are also designed according to the array structures.

1.5 Objectives of the Thesis

The aim of this thesis is to design a flexible, digitally controllable and high drive capability beam-former application specific integrated circuit (ASIC) to be flip-chip bonded or wire bonded to CMUT arrays in ultrasound therapeutic applications. The designed beam-former circuit can be used as the transmitter circuit for CMUTs utilized in real-time 3D imaging applications by the help of additional digitally controlled oscillator. Besides, the effort here is also to develop a system which exhibits better performance and cost parameters than the

circuit designed in [4]. Following is a list of objectives that are to be achieved throughout this research:

1. Research and study on the therapeutic ultrasound theory, understand how therapeutic systems work and characterization of the required beam-former topology for a therapeutic system.
2. Outline the system level overview of the ASIC to be designed in this thesis by naming the subcircuits involved in the ASIC in the overview
3. Design and theory of a CMUT selection algorithm to adjust the delay of each CMUT element in the system (set the phase), briefly, set the phases of each element in a phased array configuration. The phases (delays) are set according to the arrival time to the focal point of the signal from each individual element. The algorithm is designed in the processing unit (FPGA).
4. Design of a pulse generator that pulse duration (pulse width) of the generator output can be adjusted digitally from FPGA. The circuit is called as oneshot, which generates outputs in the range 2.5-10 ns. FPGA sends pulses to that circuit as 10 ns or integer multiples of 10 ns. The circuit manipulates first 10 ns region of the incoming pulse only. As a result, generation of high resolution pulses with infinitely large setting range is possible.
5. Design of a digitally controlled oscillator to generate continuous waves with a desired frequency set from FPGA. The output frequency range is large for a typical CMUT, but an additional frequency down circuit is designed to convert down high frequency signals into 1-10 MHz range. The output of the frequency down conversion circuit is selected by a 8-input multiplexer.
6. Design of a medium voltage level shifter circuit that is used to increase the core voltage level (1.8 volts) signals to medium voltage level (3.3 volts) signal in order to be supplied to high voltage pulse circuit. This circuit is designed due to the fact that threshold voltages of high voltage transistors in HV 0.18 μm CMOS technology are almost equal or slightly higher than the core voltage of 1.8 volts.

7. Design of a high voltage level shifter circuit that provides the highest possible voltage of the process to the output which is 45 volts. The configurations with DMOS transistors and high voltage extension modules are possible, but the best current supplying transistors under same bias are asymmetrical high voltage extension transistors. Besides, the designer should check whether a module is suitable for multi-project wafer (MPW) runs or not if the circuit is to be fabricated in a MPW run.
8. Design of an ASIC for ultrasound therapeutic systems to be integrated with 4 element and 4x4 element CMUT ring arrays in HV 0.18 μm CMOS Process.
9. Design of a test cell that includes all required blocks to drive a CMUT element and a capacitor having the same capacitance with a single CMUT element. In test cell case, there are additional I/O ports to measure the output of each subcircuit to understand whether it's working properly or not. The results of this test cell warns the designer to correct the mistakes made during design before fabrication of real therapeutic ASICs.
10. Preparation of a MATLAB program that computes the required digital control bits under certain configurations. The program computes the required phases (clock latencies) of each element to focus signals into the desired focal point by addressing the coordinates of focal point. Besides, the user enters the desired pulse width or pulse frequency and the program computes the corresponding digital bits.

1.6 Outline of the Thesis

The thesis consists of 6 chapters including this chapter which introduces the theory of the systems consisting this thesis.

The chapter 2 describes the theory and working principles of required subcircuits of beam-former to design the desired ASIC for therapeutic applications. Available circuit topologies are presented in this chapter.

Chapter 3 describes the design, schematic representation and simulation results of the circuits of the ASIC. There are LV (low voltage), MV (medium voltage) and

HV (high voltage) designs in the ASIC. Special considerations for each design are also included.

Chapter 4 illustrates the layout organization of the designed circuits and top level cells to be integrated with the CMUT arrays. The cells designed for 4 element CMUT array, 4x4 element CMUT array and test cell are presented.

Chapter 5 introduces the MATLAB program prepared to be used with the circuits in this thesis. The program is prepared for 4x4 element CMUT array ASIC, but adaptation to any other ASIC type is not a difficult process.

Chapter 6 is the conclusion of this thesis. The results are discussed and the achievements during this research are stated.

CHAPTER 2

2. RECEIVER ELECTRONICS FOR CAPACITIVE MICROMACHINED ULTRASONIC TRANSDUCERS (CMUTs)

The interface electronics of the ultrasound systems is designed to generate signals to be transmitted by the transducers to form the required acoustic waves and receive the echoes reflected back and amplify those echoes to a certain signal level which can be sensed by the processing unit. The signal generation path (transmitter) of the front-end electronics is discussed and described in the next chapter as a part of the system design. The receiver path of the electronics is described in this chapter as a literature review.

In order to acquire better images and performances from the ultrasound systems, the designers have been working on better arrays and front-end electronics. The way to design such high performance systems, the transmitted signal quality and the SNR of the received signals must be as high as possible along with the better performance arrays. Besides, the front-end electronics must be placed as close as possible to keep signal quality high in both transmit and receive modes, because this method eliminated the need for bulky interconnection cables which introduce additional parasitic capacitances.

The main aim of the receiver path of the electronics is to amplify the low-level noisy echo signals with eliminating the noise component and feed it to the digital

signal processing platform [23], [24]. In order to supply such a signal to the output, the systems requires low-noise blocks to keep the SNR high.

This chapter contains analysis and literature review of the main required building blocks of the receiver path by presenting instances and describing the operation principles of the blocks. Section 2.1 explains the system level overview by introducing the blocks used in this system. Section 2.2 is the description of transmit/receive switch. Section 2.3 introduces the low-noise amplifier types and examples. Section 2.4 introduces the buffers. Section 2.5 describes the integration methods of CMUTs and front-end electronics. Section 2.6 summarizes this chapter.

2.1 Receiver System Overview

As there are lots of different topologies present in the literature, the most of them include high voltage switches (transmitter/receiver switch), low-noise amplifier and output buffer as the common subcircuits. In some of the designs, the digital processing is totally left to the processor side, on the other hand, some of them includes low-pass filters, analog-to-digital convertors, FIFOs etc. on the chip.

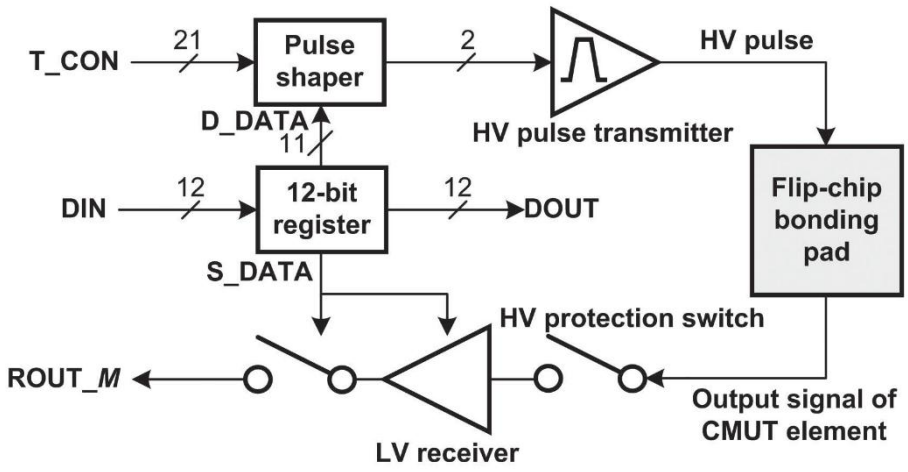

Figure 2.1 System level overview [26]

Jung et al. described their system level overview as shown in Figure 2.1. The receiver system includes the switch to isolate the transmitter part from the receiver part and the low-voltage receiver to amplify the incoming echo signal.

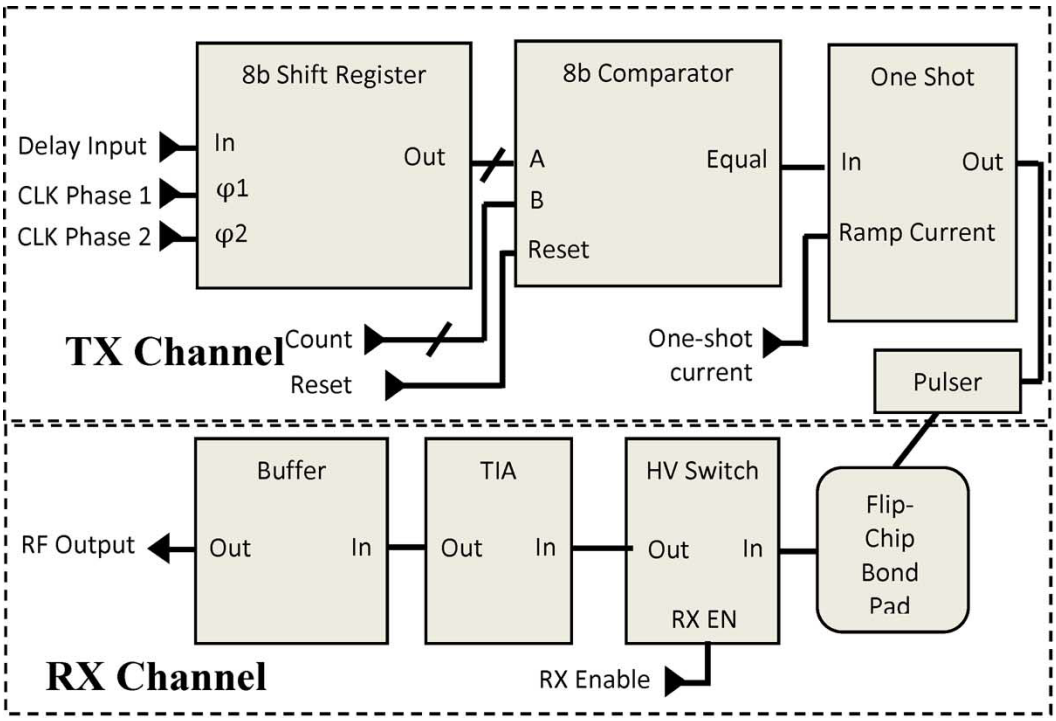

Figure 2.2 System level overview [32]

Bhuyan et al. described their system level overview as shown in Figure 2.2. Their system also includes the switch and the low-noise amplifier (a transimpedance amplifier noted as TIA). At the end, the system also has a buffer to drive the higher loads. In the proposed system of [41], a variable gain amplifier follows the low-noise amplifier and it also low-pass filter and the ADC on the chip. However, all of the system mainly has the switch, amplifier and the buffer stage.

During the design stage of those blocks, a simplified model of the CMUT is also adopted. Wygant et al. described the model they adopted for their research in [33].

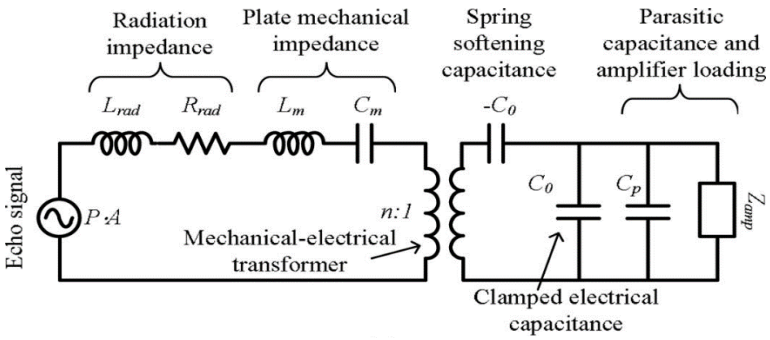

Figure 2.3 CMUT Model [33]

Excluding the parasitic capacitance and amplifier loading and radiation impedance variables, the remaining variables depend on the transducer size [33]. At the resonance frequency of the transducer, the plate mechanical impedance and spring softening capacitance parameters can be ignored and the model can be simplified [33].

2.2 Transmitter/Receiver Switch

As the designed front-end electronics gets bonded to the transducer through flip-chip bonding pads, there must be a block isolating the high voltage transmitter and the low-noise receiver. Although wire-bonding technique is generally used for 1D arrays, the transmitter and the receiver of such systems also connect to the transducer through the same pad. Generally, a high voltage transistor of a high voltage process (like 0.18 μm HV CMOS) or the DMOS transistors of high voltage processes are preferred as the high voltage switch in the receiver block. In the articles [26], [27], [28], [29], high voltage transistors are used to isolate the two signal paths.

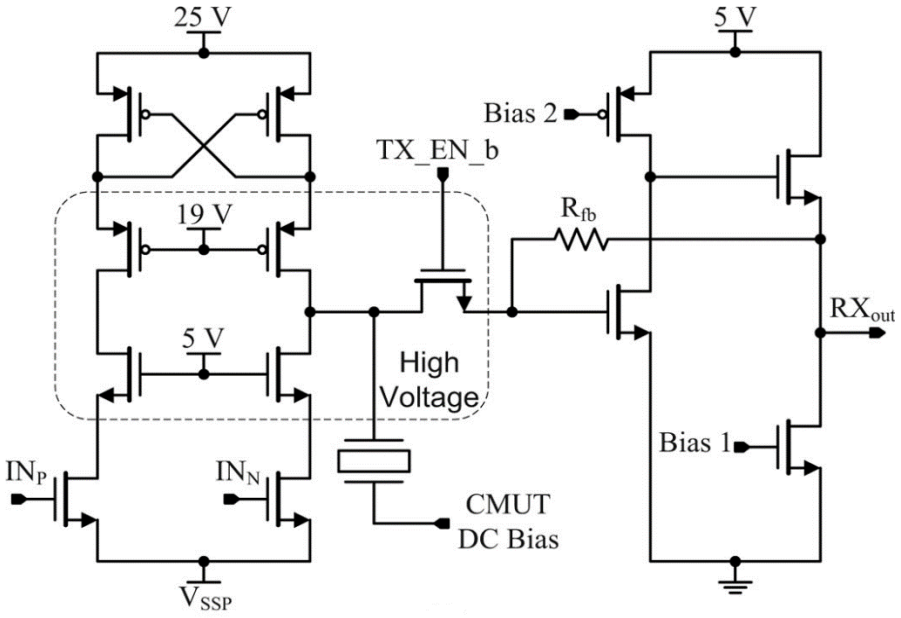

Figure 2.4 The T/R switch [29]

As shown in the Figure 2.4, the T/R switch isolates the high voltage pulser of the transmitter and the low-noise amplifier of the receiver [29]. The operation is controlled with a logic signal in this application.

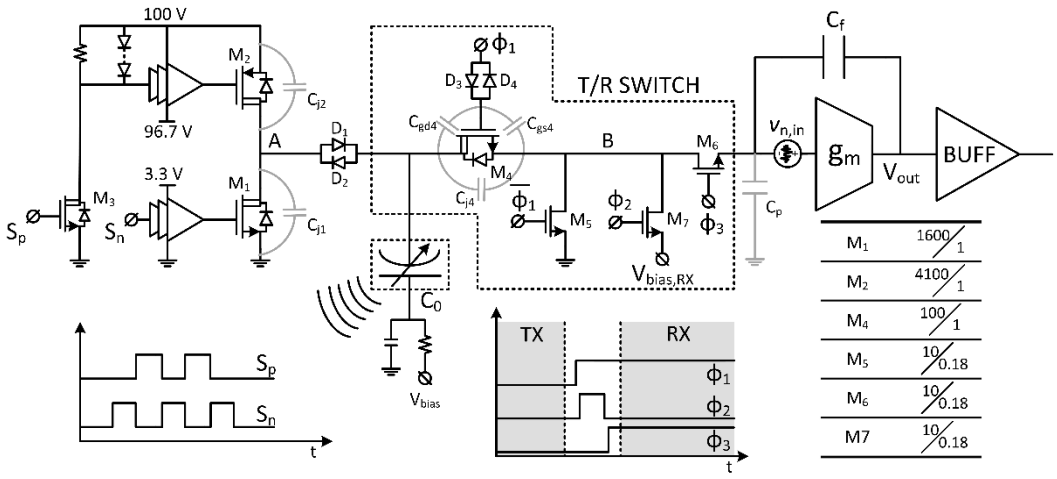

Figure 2.5 DMOS switch [40]

Sautto et al. used DMOS transistor network to implement the T/R switch block [40]. According to Sautto et al., the T/R switch network should provide low on-resistance to limit degradation of SNR owing to thermal noise and isolation to protect the low-noise amplifier. In this design, DMOS M4 having large width is employed to withstand the high voltage transmit pulses and DMOS M5 is employed to pull-down the node B to ground to guarantee the sufficient protection [40].

2.3 Low-Noise Amplifier

Due to the fact that the capacitive micromachined ultrasonic transducers converts the incoming echoes (acoustic signals) into current signals, the amplifier to be designed should be in the form of current-to-voltage converter type. As the amplifier converts the current signal into voltage signal, the following blocks processes the signal up to the signal processing unit. In order to meet the above mentioned requirement, the researches in the literature employs transimpedance amplifiers (TIAs). Basically, the transimpedance amplifier is an opamp having the input as current and generating a voltage output on the load.

Although an opamp may be composed of different types of amplifiers such as common-source amplifier, differential amplifier, source-follower etc., the transimpedance amplifiers are divided into two main groups as TIAs with resistive feedback and TIAs with capacitive feedback. These two groups of TIAs

may employ the above stated amplifiers as the opamp with the relevant feedback type. Section 2.3.1 introduces the TIAs with resistive feedback and Section 2.3.2 introduces the TIAs with capacitive feedback.

2.3.1 TIAs with Resistive Feedback

During the design of the TIA, the designer must take care of the specifications according to the transducer element and the application area. A few of the constraints that the designer should think over are gain, bandwidth, noise figure, power consumption, voltage swing, SNR etc.

In case the TIA with resistive feedback is chosen, setting the feedback resistance as high as possible decreases the input-referred current noise of the TIA [26]. This improvement may reduce the noise requirement of the next stages.

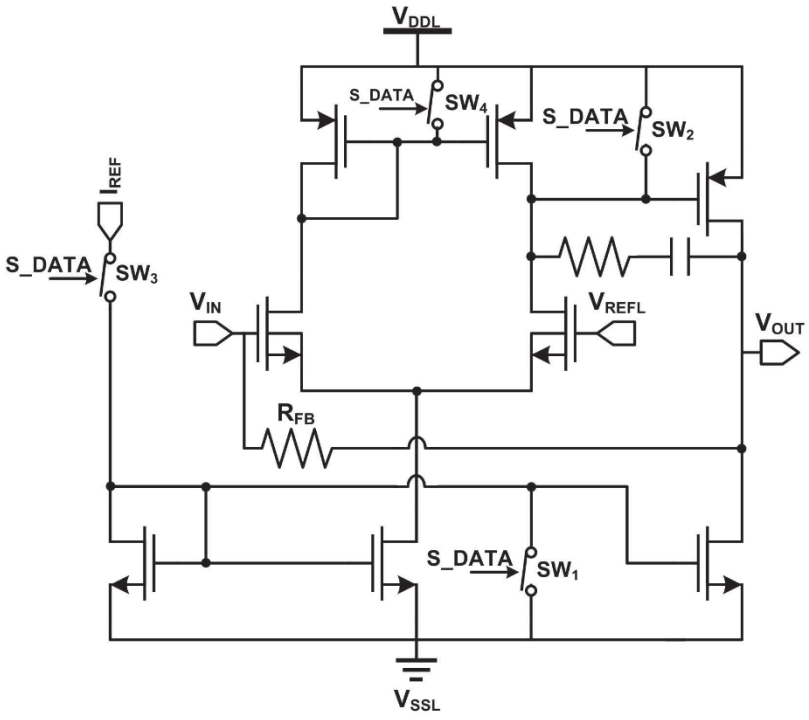

Figure 2.6 The differential pair based TIA with resistive feedback [26]

The resistive feedback TIA used in [26] is presented in the above figure. The resistance may be chosen as a diode connected MOS transistor to ease the design. The designs in [29], [33] employs physical resistors similar to the above topology, on the other, the designs of [36], [38] employs diode connected loads as the feedback resistance. By the way, the above illustrated topology is an instance

of the differential pair based TIA. According to [38], a differential pair provides better voltage swing compared to cascode structures.

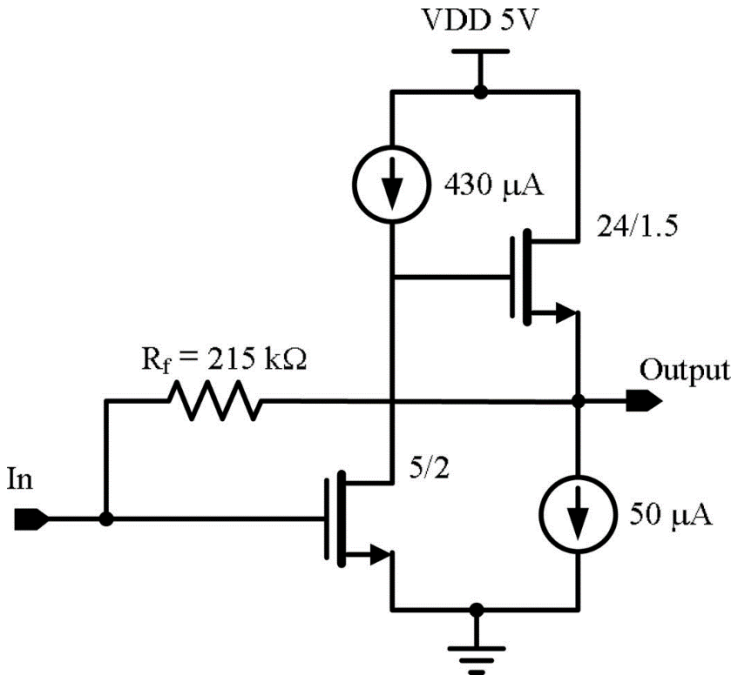

Figure 2.7 TIA with common-source followed by source-follower [33]

The above figure from [33] represents the resistive feedback TIA based on the amplifier stage having a common-source and source-follower. Same topology is preferred in [29], [32], [36], [41] as the low-noise amplifier stage. The amplifier stage of [29] is presented in Figure 2.4. More complex designs based on this topology is possible and in [41], Huang et al. presents a topology having the biasing circuitry with above topology. Huang et al. also states that TIA has a low input impedance compared to the impedance at the CMUT side which maximizes the received input current amount.

In addition to above stated topologies, a cascode common-source amplifier topology is offered in [36], which improves compactness according to Singh et al. A diode connected PMOS transistor is employed as the feedback resistance noted as MX1 in Figure 2.8.

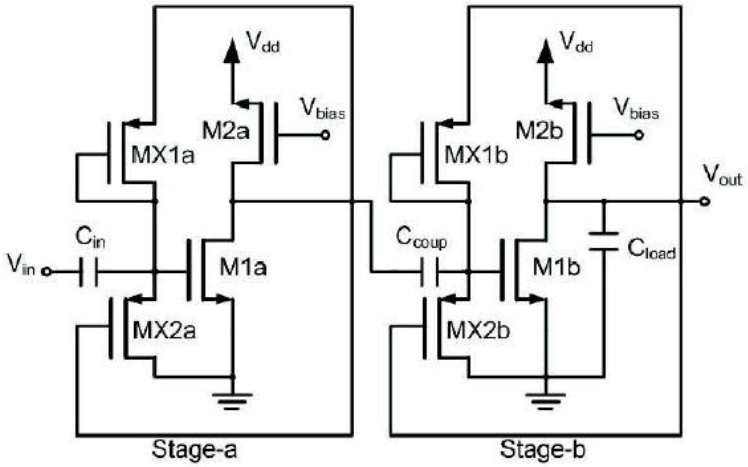

Figure 2.8 Cascode amplifier in resistive feedback TIA [36]

2.3.2 TIAs with Capacitive Feedback

Having above stated benefits, the TIA with resistive feedback fails when the application requires high operating frequencies. In this case, using capacitive feedback TIAs eliminates the sensitivity-bandwidth trade-off of the resistive feedback TIA [37]. In addition, omitting the noise generating resistor in the circuit improves the low-noise detection [30].

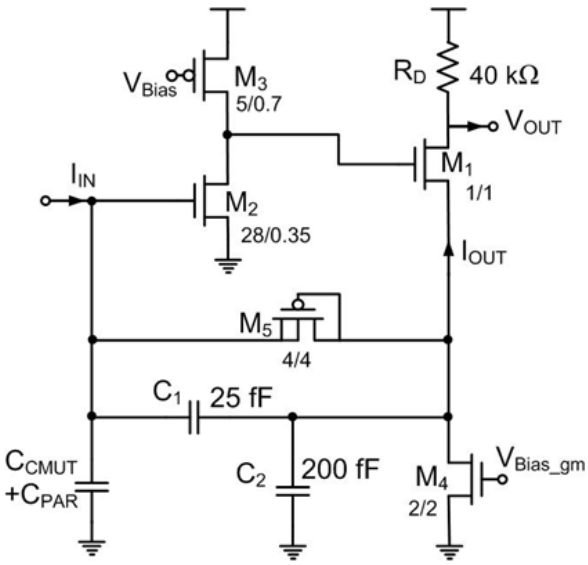

Figure 2.9 Capacitive feedback TIA based on common-source followed by source-follower [30]

The above figure illustrates the topology preferred in [30]. The topology employs common-source followed by a source-follower. The transistor M5 which is diode connected provides the dc path to input [30].

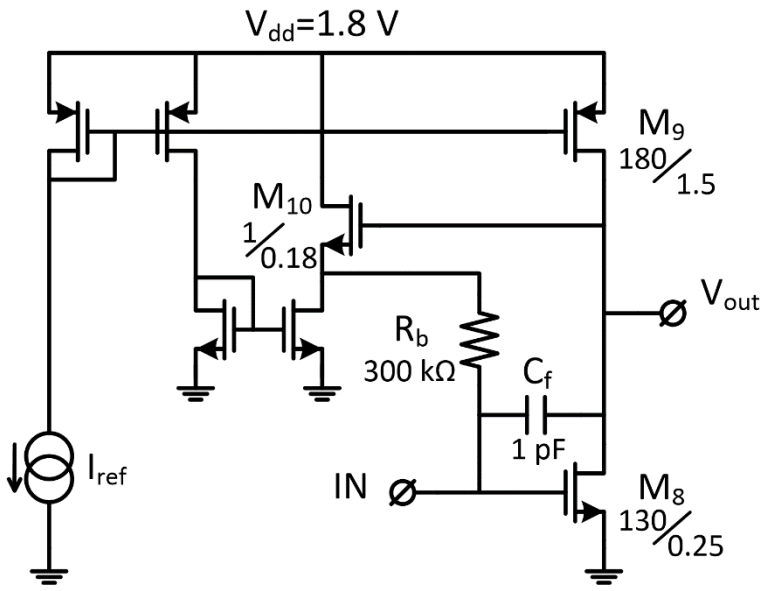

Figure 2.10 Capacitive feedback TIA based on differential pair [30]

R_b is the bias resistor to set the gate bias of the M8 transistor. According to Sautto et al., the capacitive feedback in the above differential topology provides better frequency response and SNR compared to resistive feedback counterparts.

Similar capacitive feedback topologies are employed in [24], [35], [37], and [39] too, but Peng et al. includes charge adaptation circuit in parallel to their feedback capacitor in [37]. The charge adaptation circuit detects the changes in the capacitance of the CMUT and avoids the sensitivity-bandwidth trade-off [37].

In addition to those configurations, in [23] and [34], the experimental work to test the transducers is conducted using the receiver electronics separately. The philosophy behind the working principles are the same, however, they used commercial-of-the-shelf opamps.

2.4 Buffer

Before supplying the amplified echo signal to the processing unit, a buffer stage is generally employed to set the signal voltage swing according to the following

circuit, to drive the output loads and to set the required the DC level. In general source-follower stages with unity gain is preferred and Figure 2.11 shows the one employed in [24].

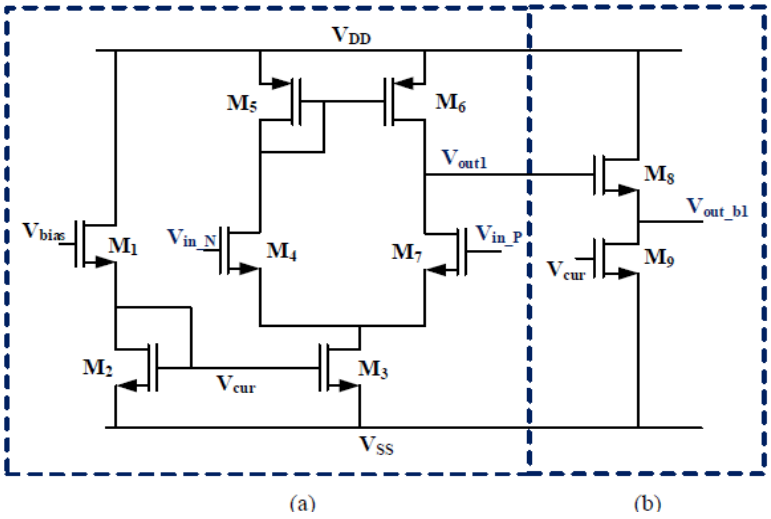

Figure 2.11 Output Buffer (b) [24]

In addition to the conventional buffer stages, a variable gain amplifier (VGA) is used as the buffer stage in [25]. The underlying reason of the utilization of such an amplifier as the buffer stage is that the range of the signal supplied to the processing unit may be out of limits. In this case, the VGA adjust the signal range according to the following stage. Besides, the VGA may compensate the signal attenuation [25]. The schematic view of the VGA used in [25] is shown in Figure 2.12.

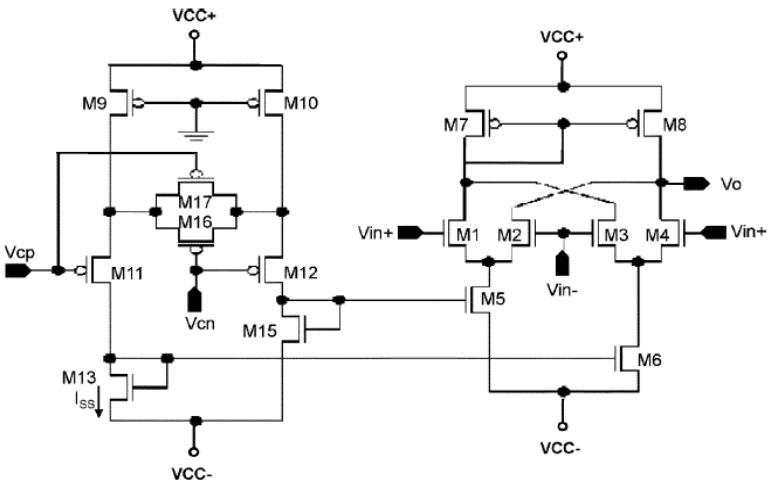

Figure 2.12 Variable gain amplifier [25]

In addition to the above described stages, the following is a table of specifications of the receivers from different references.

Table 2.1 Receiver Specifications

	Bandwidth	SNR	Gain	Tech.
[23]	0.1-5MHz	61.1±4.9	20dB	-
[24]	up to 10MHz	-	100-120dBΩ	0.35μm HV CMOS
[25]	25-75MHz	48dB	20dB	0.35μm CMOS
[26]	10MHz	Input noise 0.5 mPa/$\sqrt{\text{Hz}}$	85dBΩ	0.35μm CMOS
[27]	2-10MHz	Input noise 38 nV/$\sqrt{\text{Hz}}$	100±0.6dB	-
[28]	10MHz	-	-	-
[29]	-	-	-	-
[30]	20MHz	Input noise 90 fA/$\sqrt{\text{Hz}}$	3 MΩ	0.35μm CMOS
[31]	20 MHz	Input noise 90 fA/$\sqrt{\text{Hz}}$	3 MΩ	-
[32]	25MHz	NF 4.5dB	215 kΩ	0.25μm HV CMOS
[33]	10MHz	Input noise 0.9 mPa/$\sqrt{\text{Hz}}$	430 kΩ	-
[34]	2-4MHz	Input noise 12 nV/$\sqrt{\text{Hz}}$	-	1.5μm N-well
[35]	10MHz	NF 2dB	60.15dB	0.35 μm HV CMOS
[36]	751MHz	35.2dB	31.4dB	90nm CMOS
[37]	-	16.65dB	-	0.5μm CMOS
[38]	1-15MHz	24.3 μV rms	25.5dB	0.18 μm CMOS
[39]	-	-	-	-
[40]	40MHz	133 μV rms	14.2dB	BCD-SOI
[41]	17.5-52.5MHz	Input noise 27.5 pA/$\sqrt{\text{Hz}}$	61 dBΩ	0.18 μm 30V BCD

2.5 Integration of CMUT and Front-end Electronics

In order to eliminate the bulky connection cables and to increase the signal-to-noise ratio and signal integrity, the ultrasound systems today have probes having integrated transducers and electronics. In test systems having lower number of array elements, the wire bonding technique seems feasible in terms of application and cost, however, for the professional systems having large number of array elements (especially for 2D arrays), it is not practical to integrate the transducer and the electronics by wire bonding or through bulky cables. In [28], Khuri-Yakub et al. describes the integration techniques to overcome this problem.

Table 2.2 Summary of the Integration Methods [28]

<table>
<tr><th>Monolithic</th><th>Co-processing</th><th>Post-processing</th></tr>
<tr><td rowspan="2">Multichip</td><td rowspan="2">Chip-to-chip bonding</td><td>Bonding on a flexible intermediate substrate</td></tr>
<tr><td>Bonding on a rigid intermediate substrate</td></tr>
</table>

The monolithic integration means the transducer array and the front-end electronics are fabricated on the same substrate, whereas the multichip integration means the transducer array and the front-end electronics are fabricated on separate substrates and the substrates are bonded afterwards.

In monolithic integration, the CMUTs and the electronics can be fabricated at the same time (co-processing) or the CMUT array is realized on a finished electronics wafer (post-processing) [28]. However, this method has some drawbacks such as processing techniques limitation for CMUTs due to temperature limitations.

In multichip integration method, the CMUTs and the electronics are processed independently and bonded by vias [28]. These two separate wafer can be bonded directly with electronics substrate having larger area than the CMUT substrate, or

can be bonded on an intermediate substrate which allows independent substrate sizes [28]. The intermediate substrate can be flexible or rigid and the vias can be enabled by special techniques such as deep reactive ion etching (DRIE) [28]. The circuit designed in the following chapters of this thesis is to be bonded to an annular CMUT array through bonding pads without any intermediate substrate. The bonding pads are filled with solder bumps and then heated up to get integrated with each other. The 1D arrays with small element numbers do not necessarily requires those methods.

2.6 Summary of the Chapter

In this chapter, the main blocks required to design the receiver electronics are introduced and described. The circuits described in this chapter are the well-known circuit topologies and commonly used in the receiver electronics part of a CMUT-based ultrasound system. The researches cited in this chapter includes works with different fabrication processes. Besides, the CMUT arrays under development or use in the works cited here are generally rectangular arrays, however, the design in this thesis is done for the annular CMUTs. The integration methods described here is the common methods used to combine the array and electronics, and the design in this thesis includes ASICs to be bonded both by wire bonding technique and chip-to-chip integration technique.

CHAPTER 3

3. THE DESIGN OF FRONT-END ELECTRONICS FOR ANNULAR CMUT ARRAYS IN HV 0.18 μm CMOS TECHNOLOGY

The architecture, operation principles and simulation results of the circuits designed in the ultrasound therapeutic application specific integrated circuit (ASIC) are described in this chapter. The high voltage (HV) 0.18 μm CMOS technology is used during design which is an available and common process. The core voltage of the technology is 1.8 volts whereas 3.3 volts is called medium voltage and 45 volts is called as high voltage. In this chapter, in addition to the presented low voltage and high voltage circuits in previous chapter, combinational logic circuits and other low voltage digital blocks are presented in detail. The temperature variations and supply variations of some blocks are reported in addition to comparisons with some similar circuits in various technologies.

As stated in previous chapters, there are a lot of different circuits in the interface circuits for different applications. It is also possible to choose different topologies with different circuits for the same functioning ICs. Similar to its counterparts, the designed ASIC in this thesis is composed of low voltage (LV) and high voltage (HV) parts. In addition to those blocks, the ASIC includes a medium voltage block (MV) in order to bias the high voltage block properly. This requirement is due to the fact that high voltage transistors in HV 0.18 μm CMOS

process have a gate threshold almost in range 1.6 to 1.8 volts. As a result, proper functioning of high voltage blocks require a medium voltage (MV) block. Medium voltage blocks can be supplied by 3.3 volts or 6 volts according to design. 3.3 volts is used as medium voltage in this design, since this level of supply can be drawn from FPGA too. The low voltage (LV) part of the ASIC includes demultiplexer, oneshot circuit, digitally controlled oscillator (DCO), frequency down conversion (FDC) circuit, a multiplexer with 8 inputs, another multiplexer with 4 inputs and a buffer at the end. Following the LV part, a medium voltage level shifter is designed that levels up the core voltage (1.8 volts) to medium level voltage (3.3 volts). Finally, high voltage level shifter gets the MV input and levels up it to 45 volts and drives the CMUT element as the load. Section 3.1 illustrated a general over of the ASIC. Section 3.2 presents the architecture and block diagram of the ASIC. In section 3.3, the algorithm implemented in the processing unit (FPGA) side is explained. In section 3.4 the demultiplexer block is explained in detail by representation of the schematic and simulation results. Section 3.5 describes the fundamentals of the delay elements and oneshot circuit and digitally controlled oscillator is represented in detail in subsections. Section 3.6 explains the design and operation principle of the frequency down conversion (FDC) circuit. Following FDC, the multiplexers with 8 inputs and 4 inputs is explained in Section 3.7. Section 3.8 gives s brief info about well-known buffer structure. Section 3.9 clarifies the medium voltage (MV) level shifter and Section 3.10 clarifies the high voltage (HV) level shifter. Then, DC biasing of CMUT array elements is described in Section 3.11. Finally, Section 3.12 concludes this chapter with a discussion and summary.

3.1 The Overview of the Beamformer ASIC

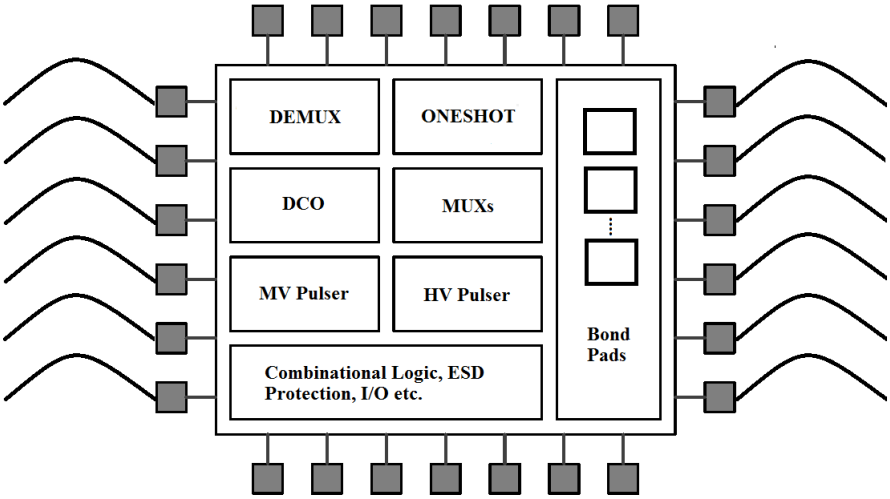

Figure 3.1 Overview of the beamformer ASIC

The designed beamformer ASIC includes low voltage (LV), medium voltage (MV) and high voltage (HV) blocks implemented in the same chip. Figure 3.1 illustrates the general overview of the ASIC by presenting the main blocks and connections with the outer world. The data connection will be implemented by wire bonding process through I/O pads and signal connection with the CMUT element will be implemented through the flip-chip bonding pads. The figure presents a top level overview and flow chart and details will be given in the following sections.

3.2 Architecture of the Beamformer ASIC

In this section, architecture and the operation principles of the designed ASIC is described in functioning point of view. The architecture and the flow chart of the system is shown in Figure 3.2

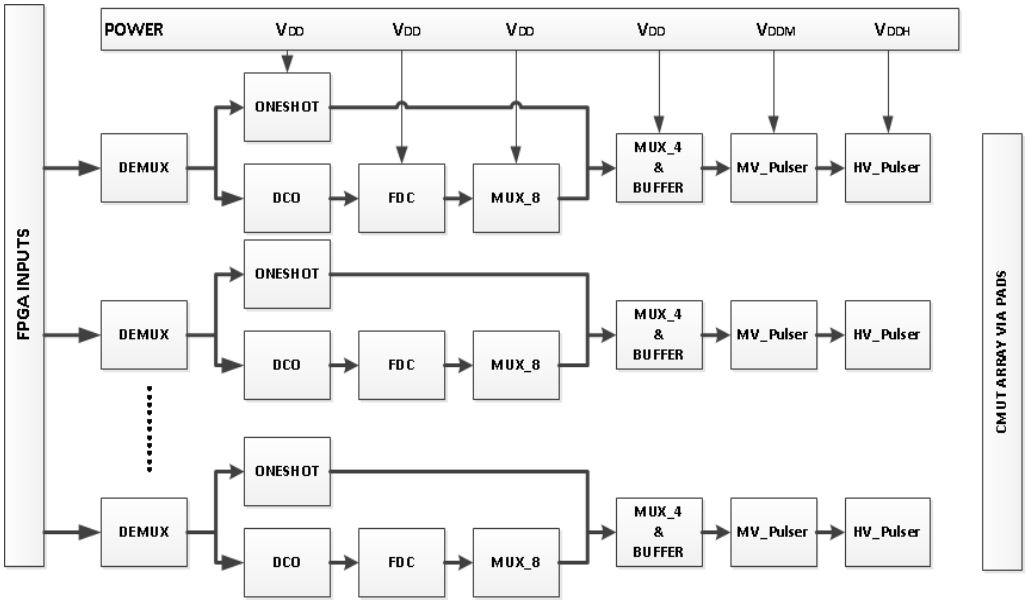

Figure 3.2 The Architecture of the system

As can be seen in the figure, the ASIC is composed of rows that function in the same manner and feeds a CMUT array element. The number of rows depends on the number of elements in the CMUT array. In this thesis, three different ASICs are designed for 4x4 element ring array, 4 element ring array and a test cell that is designed to test all the blocks in a row with an in-chip capacitor modeling the CMUT element. As a result, chips have 16, 4 or 1 elements depending on the configuration. Each row begins with demultiplexer circuit that is fed through FPGA. As the input trigger is fed by FPGA, the demultiplexer circuit feeds that input signal to oneshot circuit or to the enable circuit of the DCO depending on the select bit set by the user. If the oneshot is selected, the oneshot adjusts the signal depending on the control bits set by the user and feeds the resulting signal to the 4-input multiplexer. In the other case, the enable circuit of the DCO enables the oscillator and a pulse train with a frequency depending on the control bits is fed to frequency down conversion (FDC) circuit. Frequency down conversion circuit provides 8 outputs as f_0, f_1, f_2, f_3, f_4, f_5, f_6 and f_7 that are the divisions of the main frequency set by DCO by 2, 4, 8, 16, 32, 64, 128 and 256 respectively. The following 8-input multiplexer chooses one of the frequencies fed by FDC depending on the 3 select bits set by the user and feeds it to the 4-

input multiplexer. 4-input multiplexer has oneshot output, 8-input multiplexer output, a direct FPGA bit and a connection to ground as the inputs and feeds one of them to the buffer according to 2 select bits set by the user. Following 4-input multiplexer, buffer gets the output signal and buffers it to prevent any distortions and to have a sharp signal at the input of MV stage.

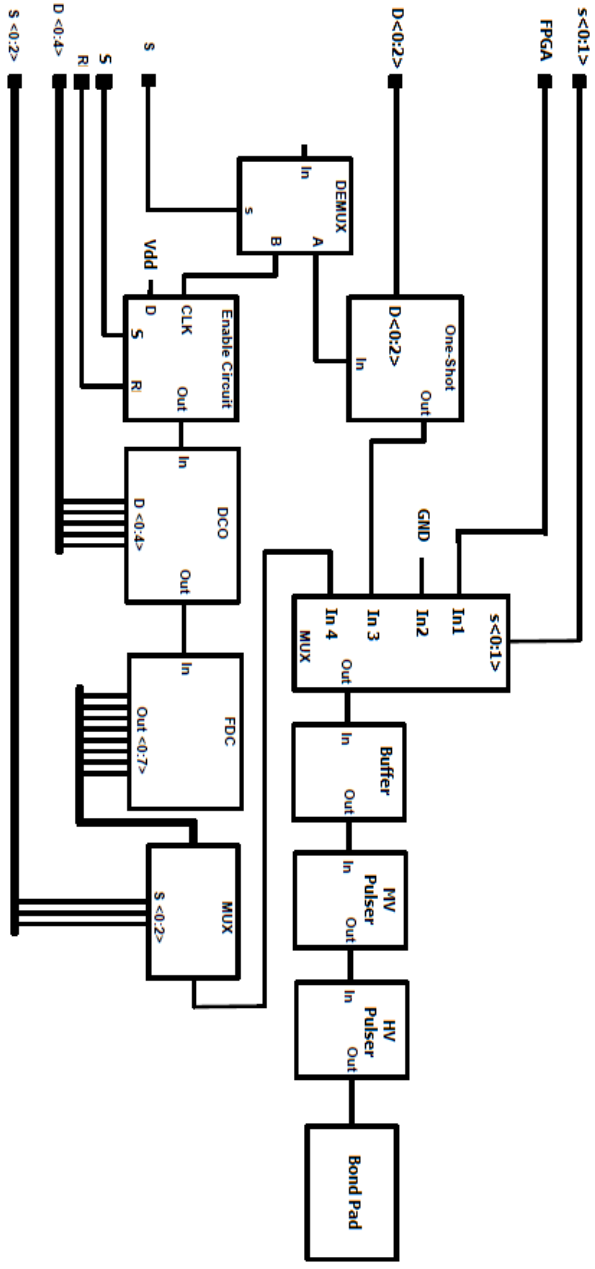

Figure 3.3 Block diagram of the ASIC

The medium voltage (MV) level shifter amplifies the signal at the core voltage level to medium voltage (3.3 volts) level. This medium voltage signal is then fed to the input of the high voltage pulser. The high voltage (HV) pulser levels up the medium voltage signal to high voltage likewise the MV level shifter and feeds the signal to the load which is the CMUT array element.

The user can excite the CMUT elements with two types of signals as single pulses with adjustable pulse width and pulse trains with certain frequencies. The pulse trains as the excitation signals are very suitable for volumetric and color Doppler imaging applications [43]. As stated in previous chapters, the designed ASIC can be used as a transceiver for imaging systems with an addition of receiver LNA block such as a transimpedance amplifier (TIA). The underlying reason of why these two signal types are generated with variable properties is that each CMUT array has its unique properties and should be excited according to those specifications. Due to this uniqueness, the ASIC should provide suitable signals for different CMUT arrays to be a suitable for integration with number of arrays as much as possible. Why 45 volts are chosen as the high voltage is that the technology used in this design offers several voltages up to 45 volts. As a result, the highest possible voltage is used in this design.

3.3 FPGA Algorithm

In the FPGA side, the control of the settings bits and phasing operations are realized in conjunction with the MATLAB program. As the user defines the signal types and shapes in the MATLAB program and sets the focal point to be fired, the program computes the phasing algorithm and sets the required control bits. Then, the program sends these values to FPGA to be fed to ASIC. FPGA sets the control bits of the beamformer ASIC rapidly and starts counting down to excite each row according to firing times coming from MATLAB program. The MATLAB program computes the phasing of the CMUT elements according to the distances from focal point and clock speed used in the FPGA side. It computes the travelling time of the wave generated by each CMUT element and divides that computed time into one clock duration of FPGA. The result gives how many cycles FPGA needs to wait for firing. For instance, assuming that it

takes 10 clock cycles for the wave generated by the first 8 elements of CMUT array and 20 cycles for the remaining elements, FPGA waits 10 cycles and feeds the demultiplexers of the first 8 element and 10 more cycles for the remaining elements. Due to the fact that all blocks in the rows for different CMUT elements are totally same, the excitation time of demultiplexer determines the excitation time of the corresponding CMUT element. The delay introduced by each row is the same and each row generates the same signal under same control bit settings.

By using this algorithm, the ASIC does not require any combinational logic or specific circuit to phase all CMUT elements. The advantage of this algorithm over other types are reduction of silicon area and reduction in power consumption by ASIC basically. However, the algorithm is suitable for arrays having small number of elements due to the individual connections of each row from FPGA. In case of shift registers are used for phasing, higher number of elements can be driven yet it depends on the configuration and the shift register itself. For instance, and 8 bit shift register may generate 256 clock cycles of delay, but there may be some cases that requires more delays. As a result, the phasing topology should be designed depending on the application area and array structure.

The FPGA side also supplies the core voltage and the medium voltage of the beamformer ASIC. By the help of easy setting of the FPGA I/O banks, 1.8 volts can be set for control bits by setting an I/O bank of FPGA to 1.8volts level. Supply ports are predefined and can be found from the manual of the FPGA.

3.4 Two Input Demultiplexer

In this beamformer ASIC, the demultiplexer is used to feed the input signal coming from FPGA to either oneshot circuit that generates single pulses with variable widths or digitally controlled oscillator that generates pulse train with various frequencies. The demultiplexer is a generic circuit that has 2 NAND gates and 3 INVERTERS. Figure 3.4 illustrates the typical inverter used in the demultiplexer. The standard sizes of 2/0.18 μm is preferred for both NMOS and PMOS.

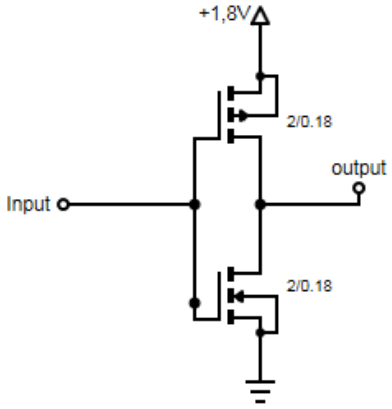

Figure 3.4 Inverter in the demultiplexer

The NAND gate is also the generic design. The transistors have the same sizes with the ones used in the inverter. The topology is shown in Figure 3.5.

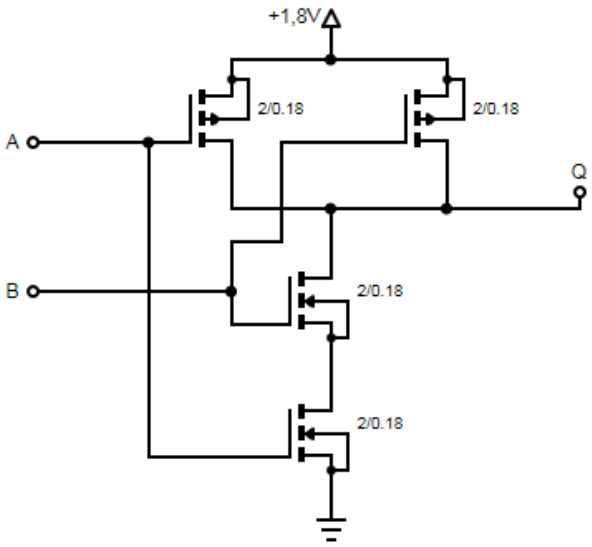

Figure 3.5 NAND gate in the multiplexer

The demultiplexer is composed of two NAND gates that each one feeds one of the signal generator circuits. The select bit of the circuit is fed one of the NAND gates to directly and the other after inverted by an inverter. The outputs of the

NAND gates are inverted by two different inverters before stepping into the following circuit. Figure 3.6 shows the configuration of the demultiplexer.

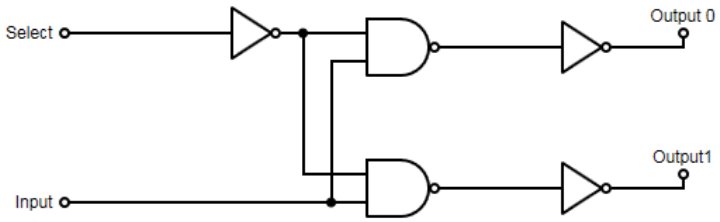

Figure 3.6 Configuration of the demux

The width-length ratio of the transistors employed herein is enough to charge or discharge the load rapidly. In order to make sure that the circuit operates properly, a spice simulation is done and Figure 3.7 shows the simulation results.

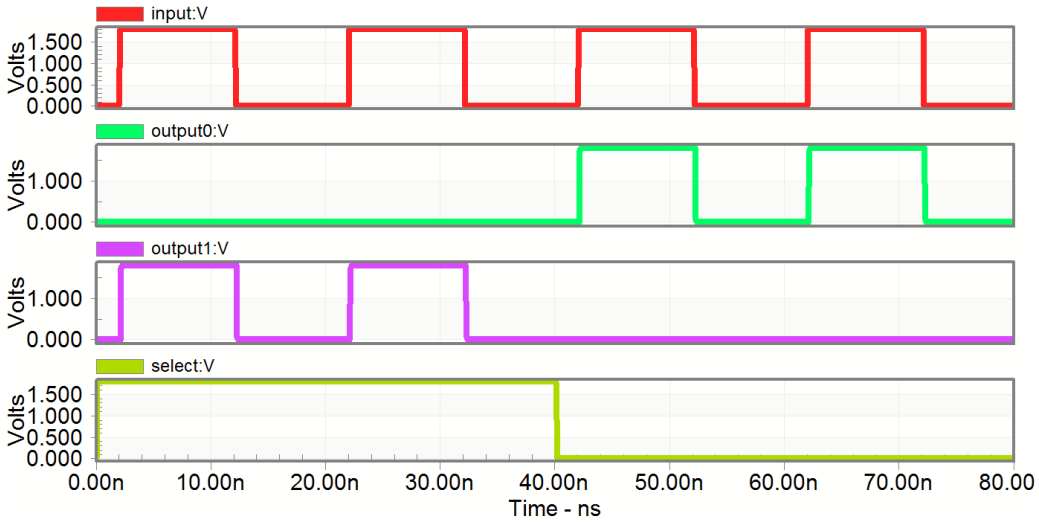

Figure 3.7 Simulation results of the demultiplexer

The average power consumption of the circuit is 8.8 µW. As can be seen through the graph in figure, the first signal is the input and the second one is the select bit. According to select bit, the input is transferred to the corresponding output.

3.5 Circuits Composed of Delay Elements

In today's world, the demand to oscillators and delay elements are at an amazing degree owing to the need of them in almost all circuits in digital applications. Many of the digital circuits contains those elements in various types. There are RLC oscillators that are widely used in the industry, however, implementation of those type of oscillators are challenging and not cost effective in many cases although those are more accurate. Due to that problems, the designer prefer digital oscillators such as VCOs and DCOs since those are cost effective and easy to design and implement. As stated in previous chapters, there are several ways to design an oscillator and manipulation of charge/discharge current and load capacitance is the most popular ones among digital blocks. The variable delay elements are widely used in digital locked loops (DLLs) [47], phase locked loops (PLLs) [48], [49], processing units, digitally controlled oscillators (DCOs) [50], [51] and memory devices [52], [53]. The delay elements can be controlled by digital or analog inputs. However, in digital circuits, the analog control is not the preferred way of control. The designers generally prefer control over manipulation of logic bits by setting them high or low. In this manner, digitally controlled current sources and digitally controlled capacitive loads can be designed. However, digital control means that the user have discrete control levels instead of having full resolution control over the variable to be manipulated. For instance, in case of digitally controlled current source, the current can be set to predefined levels and cannot be changed in a continuous manner. Similarly, in case of load capacitance control, the load can be set to predefined levels and charge or discharge time changes in discrete levels. In this case, the delay can be changed by discrete predefined packets, meaning 5ns per increase for instance. Besides, the current control may be more suitable for power budget limited applications since increasing the load capacitance may result in increase of average power consumption due to higher charge and discharge time with the same current. Capacitance manipulation in order to design a delay element is described in [54] with examples.

In the beamformer ASIC designed in this thesis, the delay element in the Figure 3.8 is taken as the model and a similar circuit to the one in figure is used for DCO whereas a modified version of that is preferred in oneshot circuit.

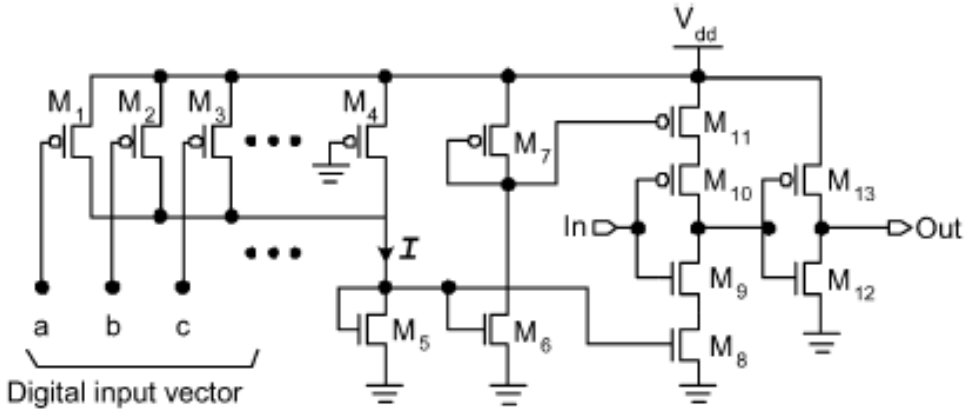

Figure 3.8 Delay element model [44]

As can be seen through the configuration in the Figure 3.8, the digital input vector determines the state of the transistors that generate the required current to be mirrored. The transistors M5, M6, M7, M8 and M11 are responsible of mirroring the current generated by M1 to M4 in this configuration. The number of transistors controlled by the digital input vector can be increased or decreased, it depends on the design specifications and the designer. Those transistors can be called as current source transistors. The current generated herein is used as both charge and discharge current by the help of current mirrors. If the designer needs only discharge or charge current manipulation, the upper or lower mirror transistors may be omitted. For instance, if the designer do not include M7 and M11 in this configuration, then the user can manipulate only the discharge current and it will result in a shortened pulse at the output. In the other case where the user manipulates the charge current, it will result in an enlarged pulse at the output. In the oneshot circuit designed for beamformer ASIC, a similar topology without the charge manipulation is preferred. In the DCO used in this ASIC, same topology with 5 controlled transistors in current source stage is employed as digitally controlled delay element (DCDE).

To have an insight on how the system works and to understand the theory and math behind the circuit, a mathematical model is as follows [55].

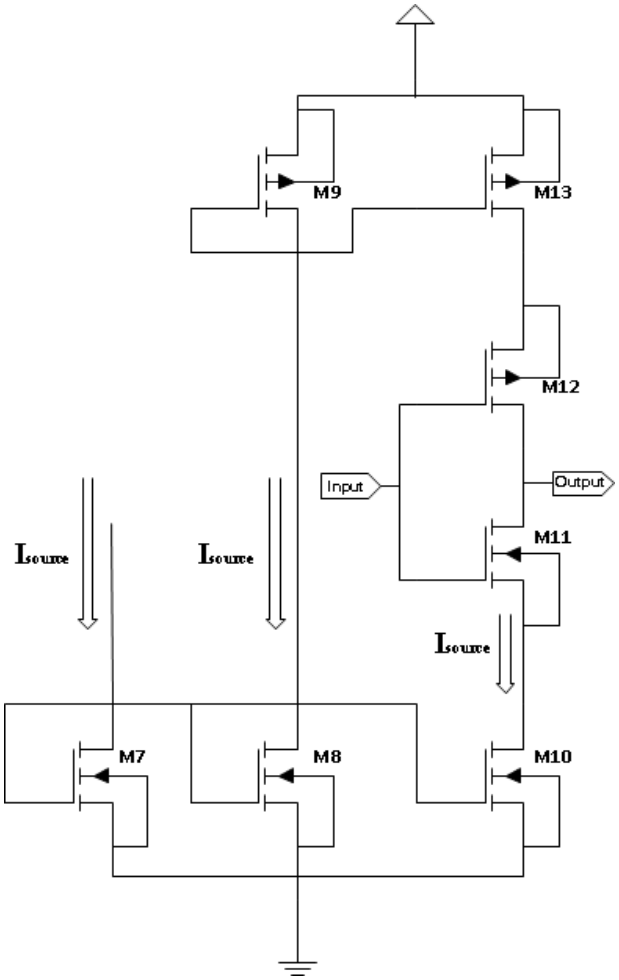

Figure 3.9 A simplified version of the delay element [2]

At the beginning, it can be assumed that the current source provides a current of I and the transistors are in saturation. The drain current of M10 in Figure 3.9 is as follows

$$I_{ds-M10} = \frac{1}{2}\,\mu_n C_{ox}\left(\frac{W}{L}\right)_{10}[V_G - V_{TH10}](1 + \lambda_{10}V_{DS10}) \qquad (3.1)$$

As can be understood from the equation, it is assumed that the M10 exhibits a velocity saturated behavior due to its very small gate length. Assuming that the voltage drop across the M11 transistor is negligible, then

$$V_{DS10} \cong V_{OUT} \qquad (3.2)$$

is obtained. Then, by applying the general current equation of a capacitor, a differential equation of

$$I_{ds-M10} = -C_L \frac{dV_{OUT}}{dt} \quad (3.3)$$

is obtained. By equating the equations 3.1 and 3.3, the following can be obtained.

$$\frac{1}{2}\mu_n C_{ox}\left(\frac{W}{L}\right)_{10}[V_G - V_{TH10}](1 + \lambda_{10}V_{DS10}) = -C_L \frac{dV_{OUT}}{dt} \quad (3.4)$$

Having the initial condition in 3.5 and the equation in 3.6

$$V_{OUT} = V_{DD} \quad @ \quad t = 0 \quad (3.5)$$

$$V_{OUT} = \left(V_{DD} + \frac{1}{\lambda_{10}}\right)e^{\frac{-t}{\tau}} - \frac{1}{\lambda_{10}} \quad (3.6)$$

and knowing that the C_L is the load capacitance, the solution of the 3.4 is as follows

$$\tau = \frac{C_L}{\lambda_{10}} \times \frac{1}{\frac{1}{2}\mu_n C_{ox}\left(\frac{W}{L}\right)_{10}[V_G - V_{TH10}]} \quad (3.7)$$

By having those relations and assuming the equation in 3.8 is true, the delay can be found using the equation in 3.9.

$$V_{OUT} = \frac{V_{DD}}{2} \quad @ \quad t = t_{delay} \quad (3.8)$$

$$t_{delay} = \tau \times \ln\frac{1 + \lambda_{10}V_{DD}}{1 + \lambda_{10}\frac{V_{DD}}{2}} \quad (3.9)$$

The important thing here is that the designers must be careful about whether the M10 transistor is velocity saturated or not in their designs. The designer should check the fabrication technology manuals to see if the transistor with the designed W/L ratio is accepted as velocity saturated or not. If not, the 3.1 should be changed with the relevant saturation drain current equation.

3.5.1 Oneshot Circuit

The single pulse with variable pulse durations is generated with a circuit configuration similar to the presented in Figure 3.8. The circuit employs a current source that has 3 controlled and 1 on-state PMOS transistors. The gate of the on-

state transistor is tied to ground. This transistor is always on in the saturation region and supplies current in case other controlled transistors are not on. The digital input vector controls the state of the transistors upon the user's choice. The configuration is shown in Figure 3.10.

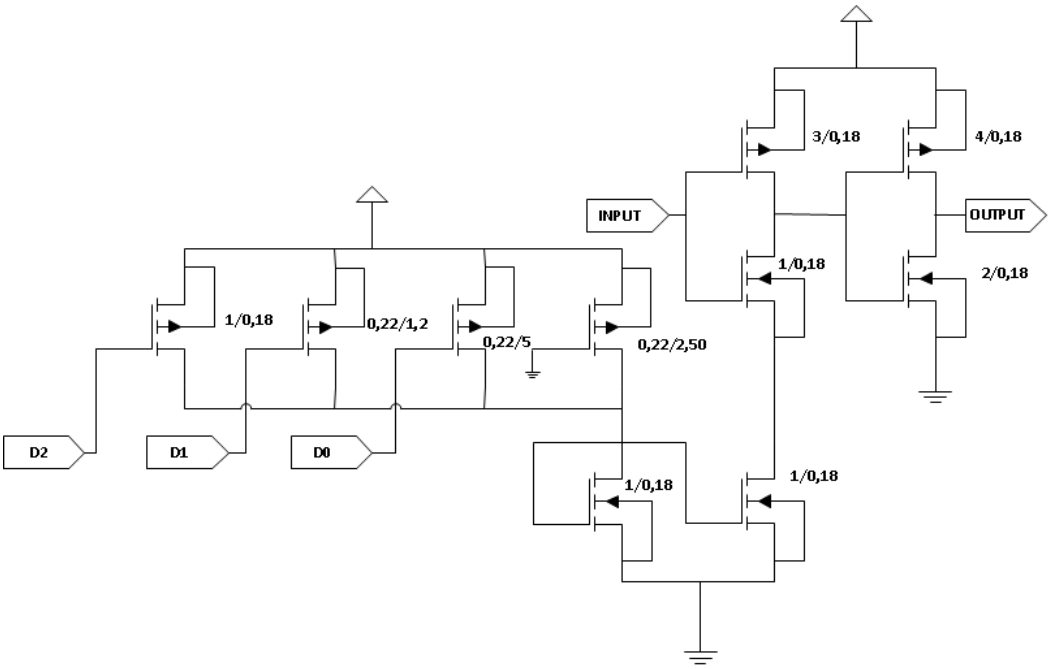

Figure 3.10 The configuration of oneshot circuit

By the help of current mirror, the discharge of the first inverter is controlled. Since each individual transducer array has its own unique properties like working frequency bandwidth, the circuit must generate suitable pulses to those arrays. For example, pulse width minimum is around 8-10 ns for 40 MHz transducers and pulse width maximum is around 100 ns for 5 MHz transducers [2]. In this topology, the circuit manipulates the 10 ns long signal. The signal provided from the FPGA is 10 ns or its multiples long. The circuit manipulates the very first 10 ns and do not pass the remaining part. For instance, the circuit outputs a 5 ns pulse under a certain digital input vector configuration if the input is 10 ns. Having that specific input vector configuration same, if the FPGA sends a 50 ns pulse, then the circuit outputs a 45 ns long pulse. In this way, circuit spans the full range of the available pulse widths with a high resolution. None of the designs in the literature have that much resolution with a basic circuit like the one

used in this design. The corresponding transistor sizes are illustrated in the Figure 3.10 each labeled near the corresponding transistor.

The available outputs in case the input is 10 ns is shown in the Figure 3.11 with a graph. The graph shows the outputs according to the 3 bit digital input vector combinations.

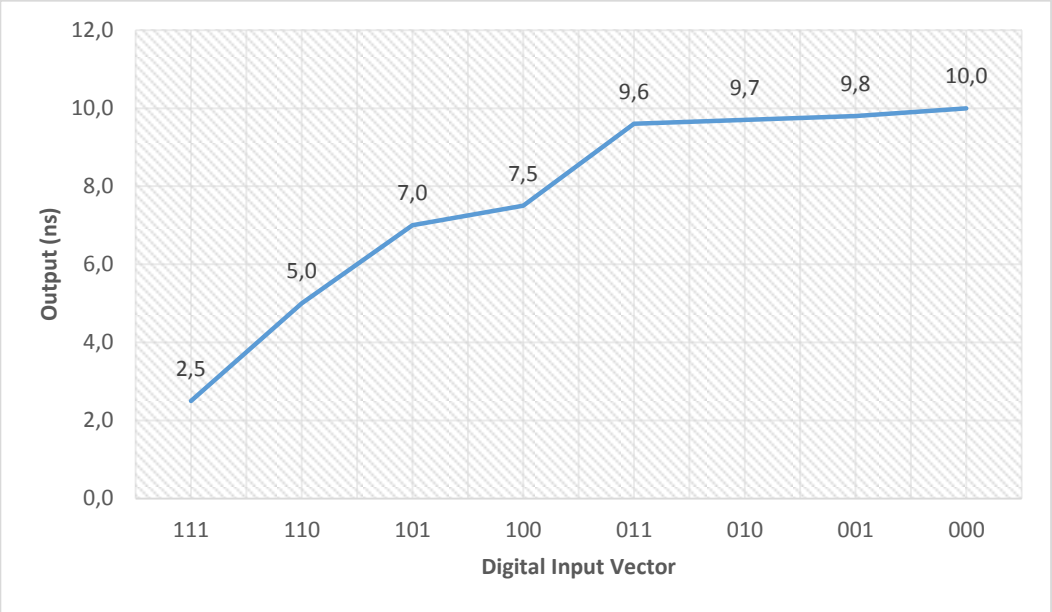

Figure 3.11 Oneshot outputs when input is 10 ns

In case the input is 40 ns long, the results are shown in Figure 3.12

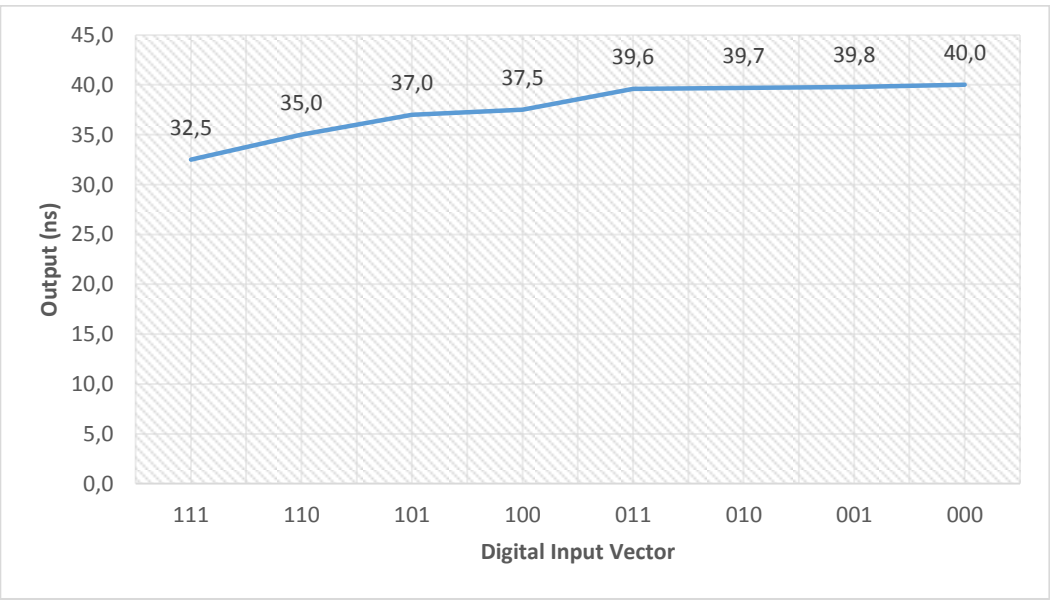

Figure 3.12 Oneshot outputs when input is 40 ns

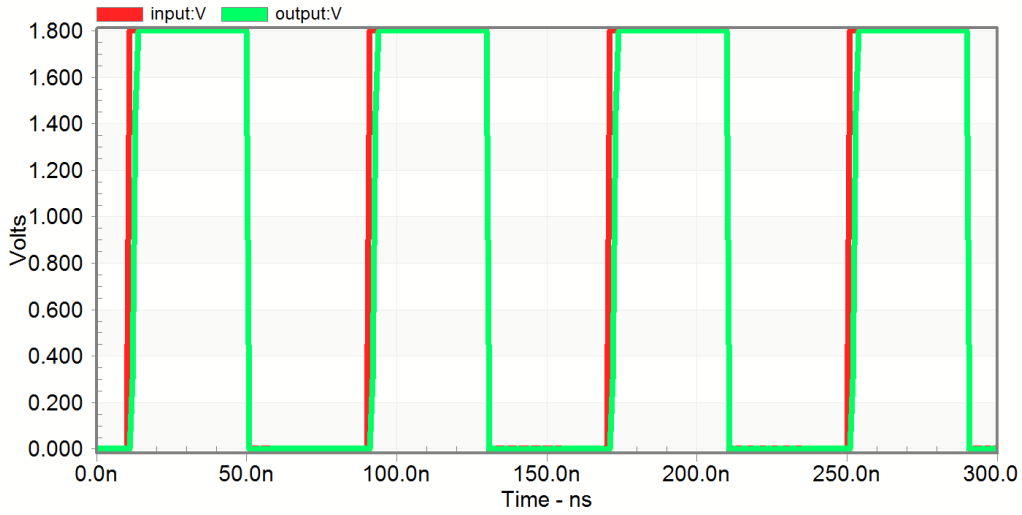

Figure 3.13 Oneshot simulation when input is 40 ns and input vector is 100

The average of the current drawn from the supply by the circuit is 141.8 µA and 16.2 µA for 000 and 111 input vectors respectively. Since the circuit is generally set to draw less current, it can be said that the average consumption is about 30 µW in this circuit. It is very low compared to the counterparts used in similar ASICs.

Table 3.1 Comparison of similar oneshot circuits

	Circuit in 2009 [42]	Circuit in 2013 [4]	Proposed circuit
I_{SET} source	M1 and M2 configuration	M0	Digitally controlled current source
Capacitor (pF)	1.5	2	-
I_{SET} vs Pulse	I_{SET} between 1-2 mA, Pulse between 70-150 ns	I_{SET} between 70-77.5 µA, Pulse between 20-650 ns	I_{SET} between 2.5-137.5 µA, Pulse between 2.5ns to any extend
Technology (µm)	1.5	0.35	0.18
Control	External current source	External controlled current source	External controlled current source

Table 3.1 presents a comparison table with similar circuits. As can be understood from the table, the circuit is really a state-of-the-art design in terms of power consumption and output resolution.

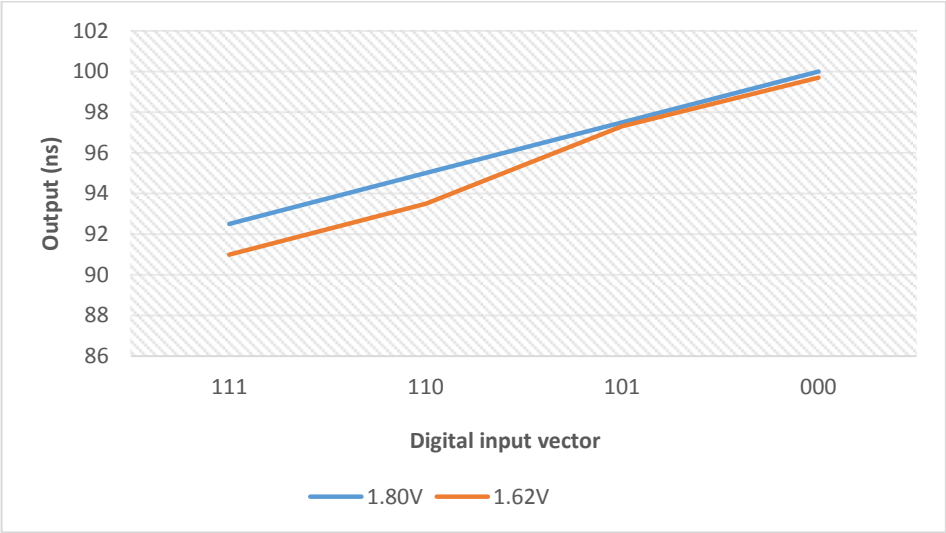

Figure 3.14 Supply variations of oneshot

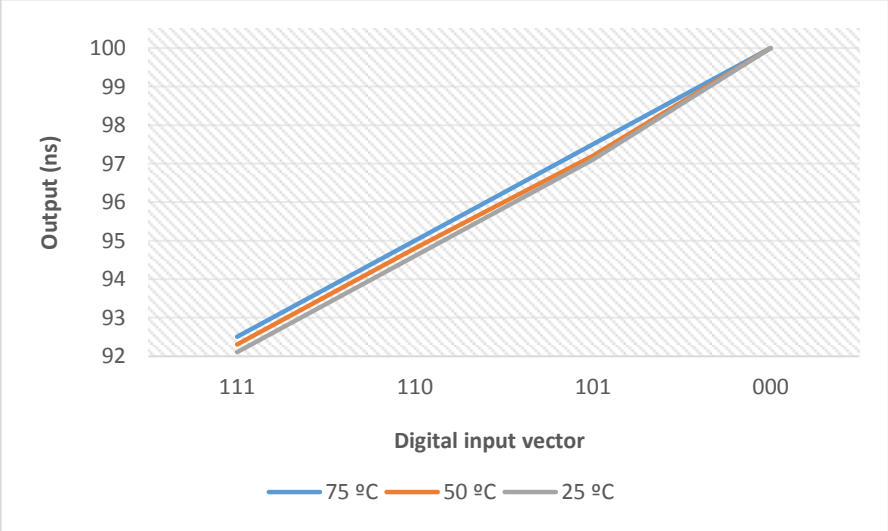

Figure 3.15 Temperature variations of oneshot

The variations of supply and temperature is also simulated for 1.62 volts of supply (-10% V_{DD}) and 25 ºC, 50 ºC and 75 ºC. The simulated values are 2.5, 5,

7.5 and 10 ns values. The graphs are prepared assuming an average pulse length of 100 ns is used. The Figure 3.14 presents the supply variations and the Figure 3.15 shows the temperature variations. The supply variations are less than 2%, 2%, 0.5% and 0.4% for 2.5, 5, 7.5 and 10 ns options respectively. The temperature variations are less than 1% in all cases. For 10 ns option, there is no variation even in 75 ºC condition.

3.5.2 Digitally Controlled Oscillator

The DCO designed herein includes an enable circuit, a 2-input NAND gate similar to the one used in demultiplexer, 3 digitally controlled delay elements (DCDEs) connected in series, a inverter in between the first DCDE and the NAND gate and 4 different inverter connected in series as the buffer stage. The block diagram of the designed circuit is shown in Figure 3.16.

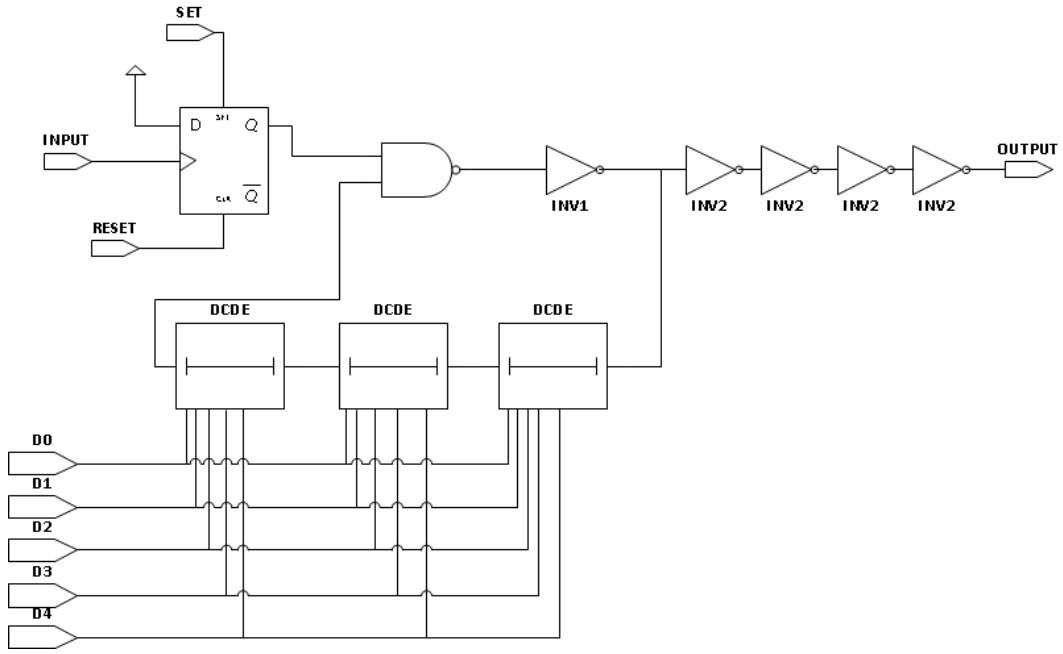

Figure 3.16 Block diagram of the DCO

The theory and mathematics behind the DCDEs used in DCO is defined in the beginning of this section. The enable circuit of the DCO is D flip-flop and the detailed explanation of this block will be given in the section that introduces

frequency down conversion circuit. Only the simulation results of the enable circuit will be represented in this chapter. The INV1 is an inverter same as the one used in demultiplexer with PMOS having 2/0.18 ratio and NMOS having 1/0.18 ratio as W/L ratio. The INV2 is an inverter with sizes 2/0.18 and 0.5/0.18 as PMOS and NMOS sizes.

The enable circuit gets its input from the demultiplexer if the user selects DCO as the signal source. The output of the demultiplexer is connected to clock input of the flip-flop and the D input of the block is tied to V_{DD}. As a low-to-high transition happens at the corresponding output of the demultiplexer, the Q output becomes high too. The QN (not Q) output of the flip-flop is not used in this case since there is no need for it. The set and reset bits of the circuit is directly controlled from the FPGA by the user. The Figure 3.17 shows the simulation results of the enable circuit.

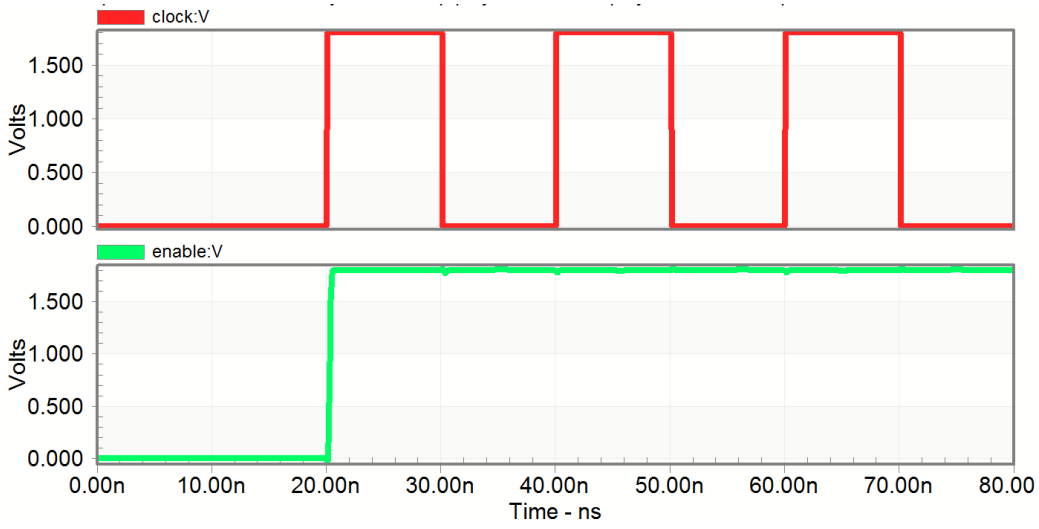

Figure 3.17 Simulation results of the enable circuit.

The output of the enable circuit goes high as soon as a low-to-high transition is occurred at its clock input. In case the input signal is not a pulse train but a low-to-high transition only, the output will not change. Briefly, since the D input of the flip-flop is tied to supply, the only requirement to have high at the output is a low-to-high transition. As the output of the enable circuit changes to high, the NAND gate following it is freed, meaning that the output of the NAND gate

depends on the other input coming from the DCDE. Shortly, whenever the enable circuit gives logic 1 to NAND, it starts to behave as an inverter. In this case, the loop becomes an oscillator composed of 5 inverters. To have oscillation, odd number of inverters should be connected as a loop, otherwise (even number of inverters) the loop stops oscillatory behavior. In addition, it is obvious that the delay introduced by the inverter and NAND gate is fixed, yet the DCDEs introduce delay depending on the digital input vector. How DCDE changes the delay it introduces is explained briefly in previous parts. In [55] the design steps of such a circuit is clarified in detail. The step by step design flow is presented. First of all, the schematic of the designed digitally controlled delay element is presented in Figure 3.18.

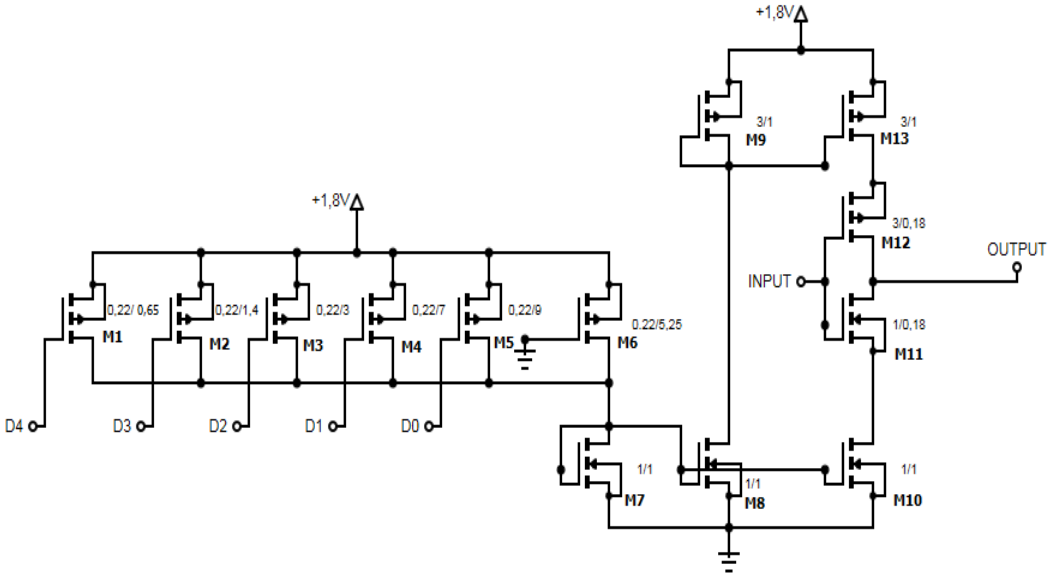

Figure 3.18 Schematic of the DCDE

The first step is to determine the sizes (W/L) of the M11 and M12 since those transistors are the load of preceding inverter. If a DCDE is connected to other in series, these 2 transistors form the load to be charged or discharged and as a result, the delay with a specific current value is determined. The intention of the designer in this configuration is to set charge and discharge currents digitally using current mirrors. In order to achieve this goal, M11 and M12 transistors should be designed much bigger than the M10 and M13 transistors. Then, the

designer should set the W/L ratios of M7, M8 and M10 according to the desired current copying factor. If the designer intends to have the same current as the generated current at the output of the digitally controlled source, the W/L ratios of these 3 transistors must be the same. By changing the ratio, designer can copy a current which is the multiple of generated one by a constant determined by W/L ratios. The same facts are valid for M9 and M13 transistors forming the current mirror for charge phase. Then, the number of transistors at the current source stage is determined according to how many discrete steps that the designer wants. In the designed DCDE, there are 5 controlled and 1 fixed transistor which supplies 32 different delay values and at the end 32 different oscillation frequencies. The key point here is to adjust M6 according to the longest delay that the designer requires to have. In case all 5 controlled transistor are in off state, the current is generated through M6 only and it is the smallest possible current generated by the current source. In order to adjust other transistors, the designer needs to assume parallel connection of that 6 transistor as a single transistor and determine the size of this single transistor depending on the shortest delay value to be required. Then, designer can determine settings of other transistors according to the frequencies that user needs. In Figure 3.19, the variations of delays generated is shown under -10% supply variation. The variations are less than 5%.

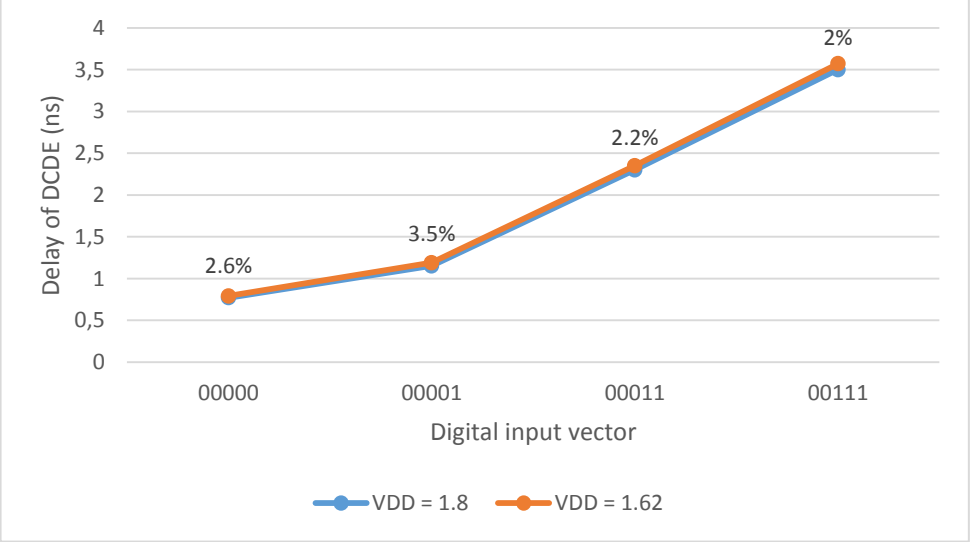

Figure 3.19 Supply variation simulation of DCDE

The temperature dependency of the DCDE is also simulated so as to test circuit under high temperature working conditions. The nominal simulation temperature is 25 ºC and the simulated high temperature values are 50 ºC and 75 ºC. Table 3.2 lists the resulting delay values of the DCDE under some certain digital input vector combinations. The simulation is realized by adding a linear temperature sweep to standard spice simulation. The delay values presented in the Table 3.2 is obtained by connecting DCDEs in series and the measured values of delays are divided in to number of DCDEs used. This way the introduced by one of the DCDEs is obtained.

Table 3.2 Delay values of DCDE for some input vector combinations

Temperature (ºC) / **Input Vector**	**25 ºC**	**50 ºC**	**75 ºC**
00000	0.77	0.79	0.8
00001	1.15	1.17	1.18
00011	2.3	2.32	2.35
00111	3.5	3.52	3.57

The percentage changes of the delay values in the table is not more than 3.5% for any of the combinations under any temperature value. The delay values represented here is in nanoseconds.

As the blocks in the DCO is explained and tested, the complete block can be tested to understand whether it is functioning properly or not. In the simulation phase of the DCO, the excitation starts from the enable circuit and the output pulse train is obtained at the end of the buffer stage realized by 4 inverters in series.

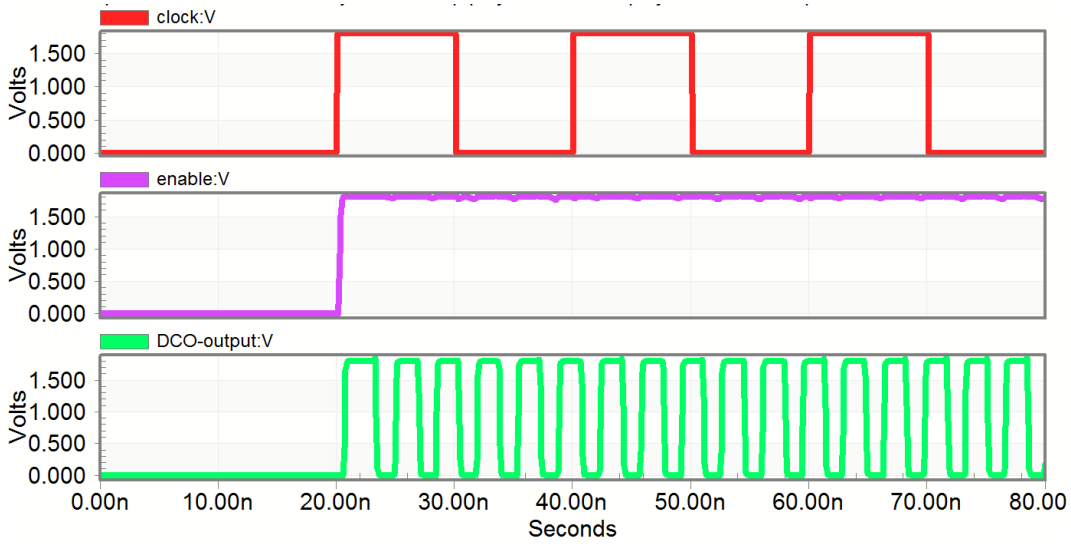

Figure 3.20 Simulation result of DCDE with input vector 00011

In Figure 3.20, as the first rising edge is appeared at clock port of the enable circuit, it enables the DCO to oscillate. The period is 3,48 ns under 00011 input vector combination which corresponds to 287 MHz oscillation frequency. The resulting pulse train frequency and pulse train period graphs of the DCO is represented in Figure 3.20 and 3.21 respectively.

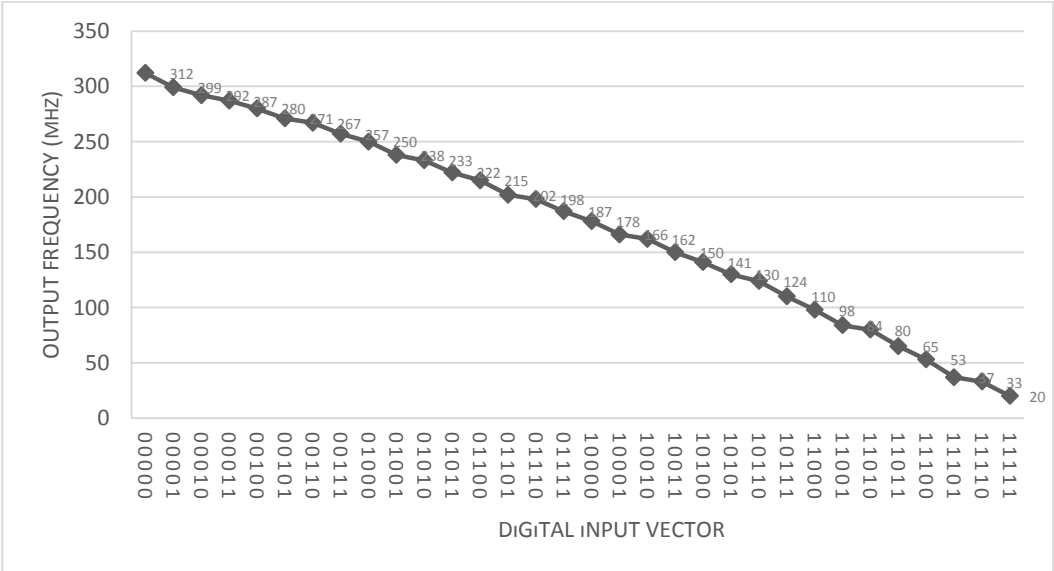

Figure 3.21 Input vector vs frequency chart of DCO

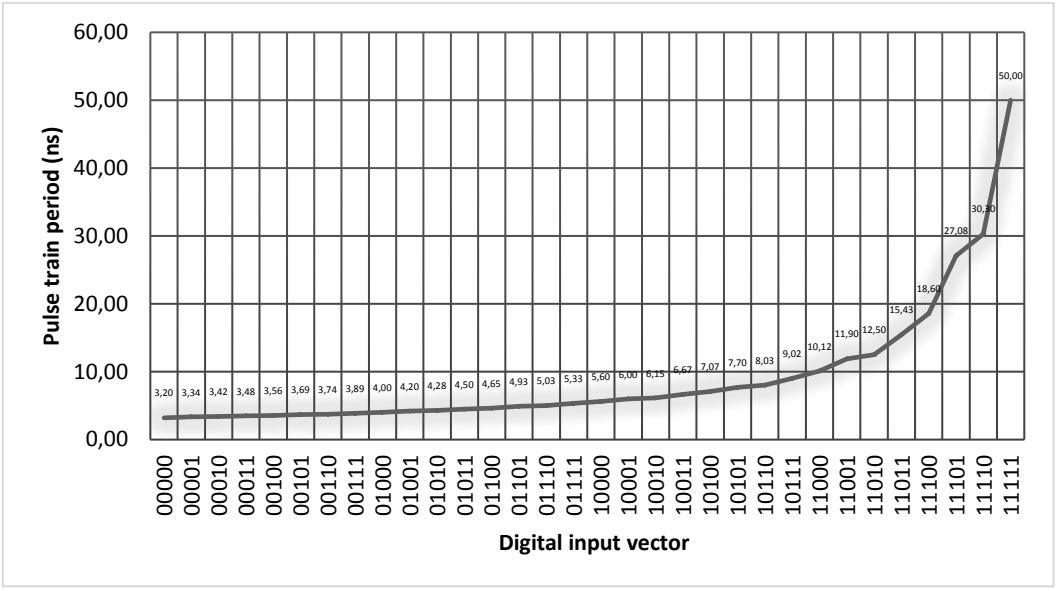

Figure 3.22 Input vector vs pulse train period of DCO

From Figure 3.21, the monotonic behavior of the DCO can be observed. If a linear curve is fitted to the graph presented, the graph in Figure 3.22 is obtained.

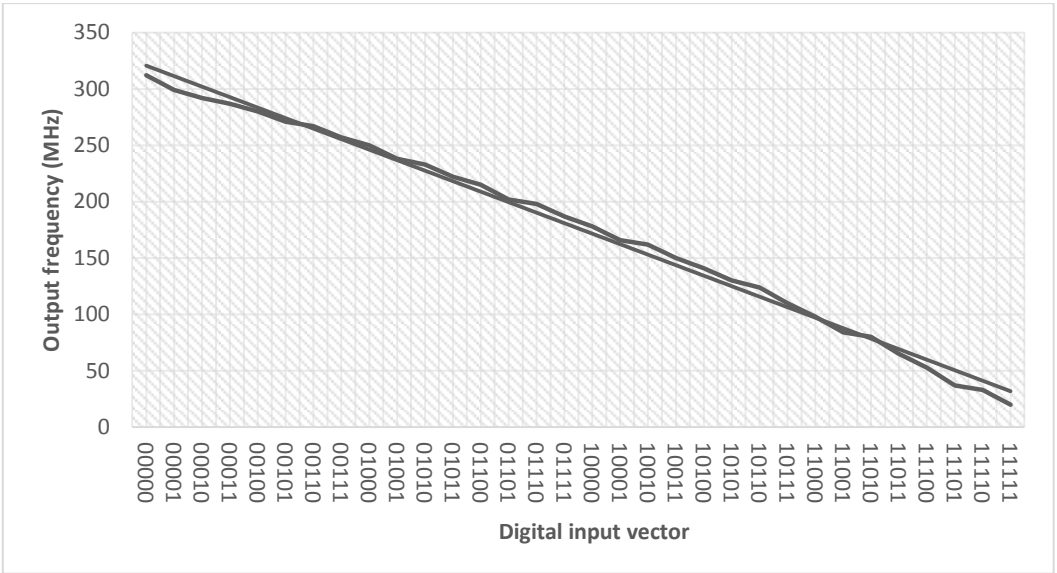

Figure 3.23 Linear curve fitted to DCO frequency chart

By the help of this monotonic behavior, the user can set the DCO to desired frequency more accurately. In some designs, the nonlinear behavior of the

oscillator prevents designers to design a system compatible with a wide range of CMUT arrays.

3.6 Frequency Down Converter

A frequency down converter fundamentally divides the input frequency by 2^n where n depends on the number of stages. As stated in previous section, D flip-flops are used as fundamental elements in this design. Typically, a flip-flop is device that keeps its output state until a signal transition happens on its clock input ports. If a transition happens, the device sets the input signal as its output state. However, if a transition does not occur, the output does not change even though the user changes the input states. Depending on the design, the device is a rising or falling edge detector. The following figure presents the topology used for D flip-flop design.

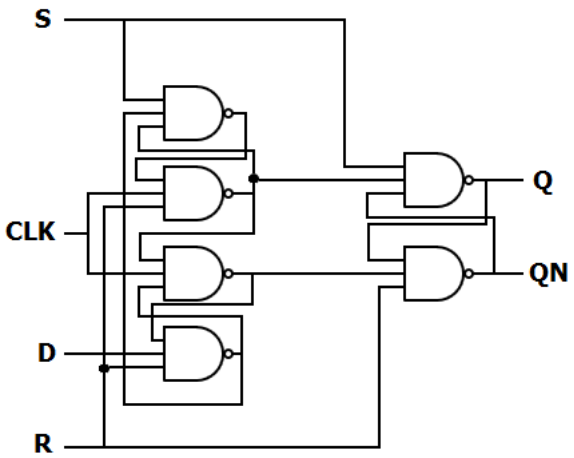

Figure 3.24 D flip-flop

As shown in Figure 3.24, this is asynchronous set/reset type flip-flop, which means that the user can reset or set the output of the device without waiting for a transition at the CLK pin of the device. The NAND gates employed in the design are 3-input NAND gates which are similar to those used in previous circuits. The following figure shows the 3-input NAND gate configuration.

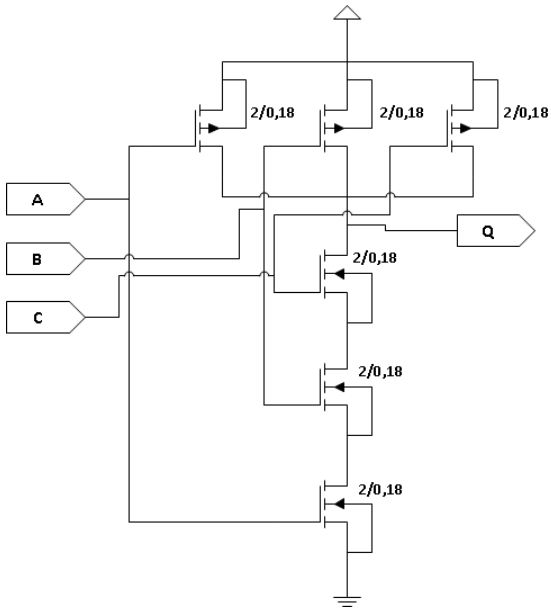

Figure 3.25 3-input NAND gate

As the main blocks of the system is defined, the FDC can be created using these items. The FDC is composed of 8 D flip-flop connected in series. The QN (not Q) pin and D pin of each flip-flop is connected and each frequency division is taken from the Q pin of the flip-flop. Figure 3.26 shows the block diagram of the frequency down converter (FDC). The Set and Reset pins are connected together even though not shown in block diagram.

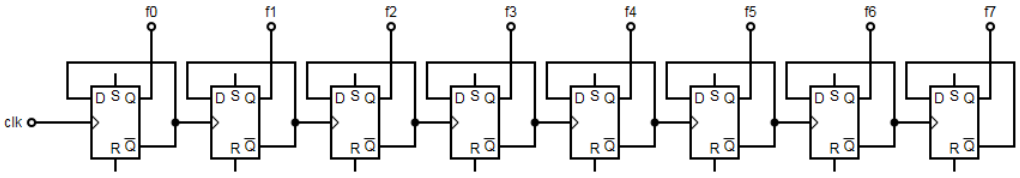

Figure 3.26 Block diagram of the FDC

The FDC is simulated by applying a 250 MHz input signal to the clock pin shown in the block diagram. As the first D flip-flop starts oscillating with the application

of clock transitions, the output pin of it starts to change its state from rising edge to rising edge of the input signal. This will result in an oscillation with frequency of half of the input signal frequency. It goes on like this through the flip-flop chain in the circuit. Each flip-flop divides the input signal into 2 and as a result, the frequency at each stage is represented as the division by 2^n where n represents the order of the stage. If the frequency is taken from the last flip-flop, n is taken as 8 in this case. A spice simulation of the frequency down conversion circuit is realized with the input signal stated above. The simulation results are shown in Figure 3.27 and Figure 3.28 representing the first 4 outputs and last 4 outputs respectively. Each graph represents the fundamental signal at the first row. As can be seen through the graphs, each cycle works properly and divides the frequency by 2. The period of the fundamental frequency is 4 ns, whereas the period of the signal taken at the last flip-flop is 2^8 = 256 times the period of the fundamental signal, which is 1024 ns. The fundamental signal in the second graph looks distorted because it is too fast compared to other signals in the graph, meaning there is no problem in signal

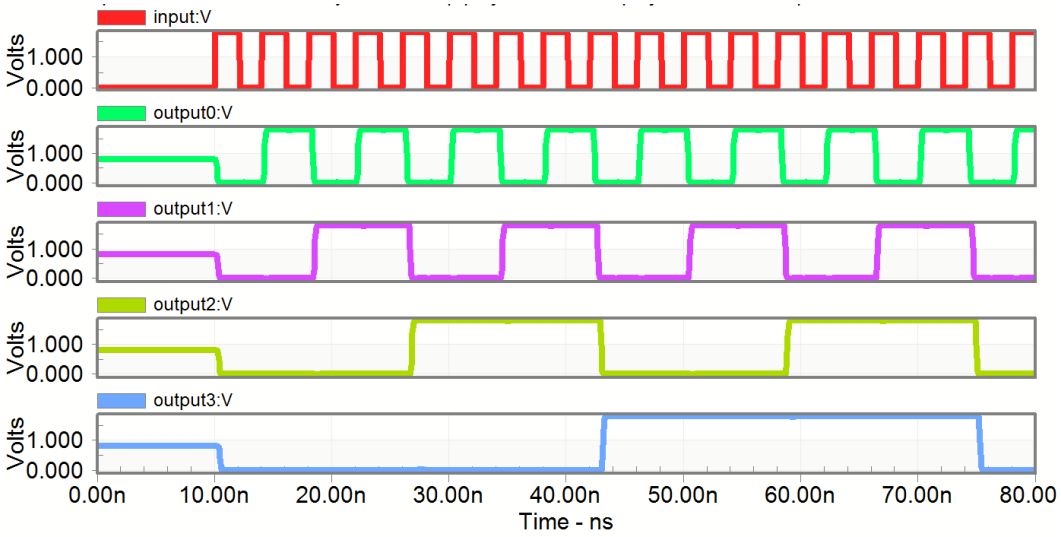

Figure 3.27 Simulation result of FDC with first 4 outputs

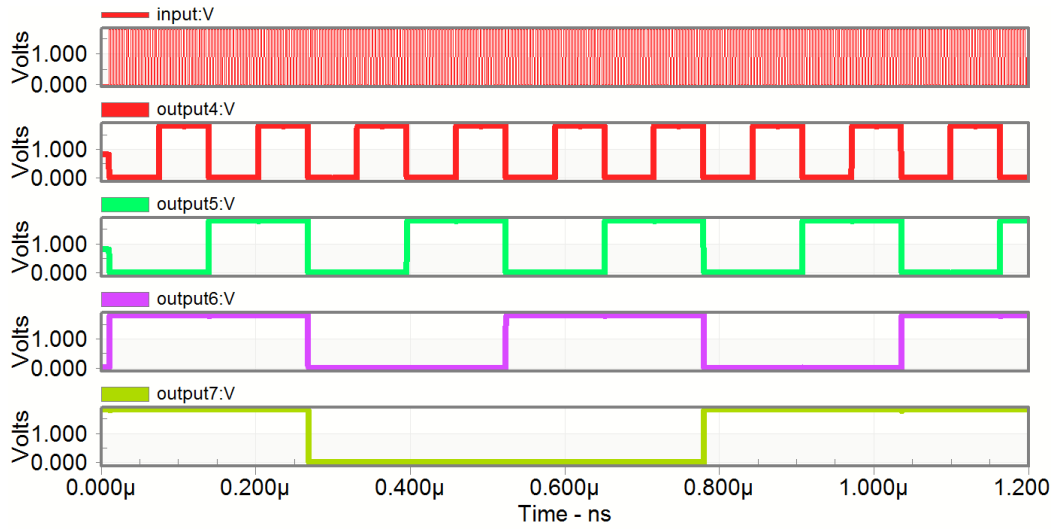

Figure 3.28 Simulation result of FDC with last 4 outputs

3.7 Multiplexers

The multiplexer is a block that helps the designer to select a signal among many of them by logic functioning. By the help of select bits each multiplexer has, the user selects an input to be passed to output. In the multiplexers used in the designed ASIC, the transmission gate structure in the Figure 3.29 is employed to form the multiplexers. Both of the multiplexer are designed in the same manner, the only difference is the number of transmission gate they have.

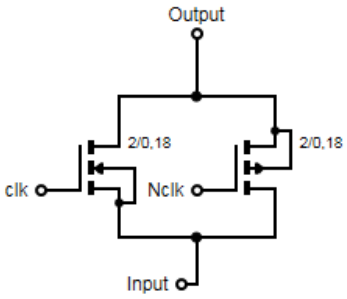

Figure 3.29 Transmission gate

The transmission gate (t-gate) shown in figure consists of 1 NMOS and 1PMOS transistor that are connected. Due to lack of power connection, the source or drain junctions of the devices are not determined yet. The responsibility of the structure

is to pass the input to the output when a clock (clk) signal at logic high is applied to MOS gate. The Nclk pin is the inverter version of the clk pin. The multiplexer topologies have inverters same as the one used in demultiplexer, to inverter the clk signal into Nclk. The topology of the 4-input multiplexer is shown in Figure 3.30.

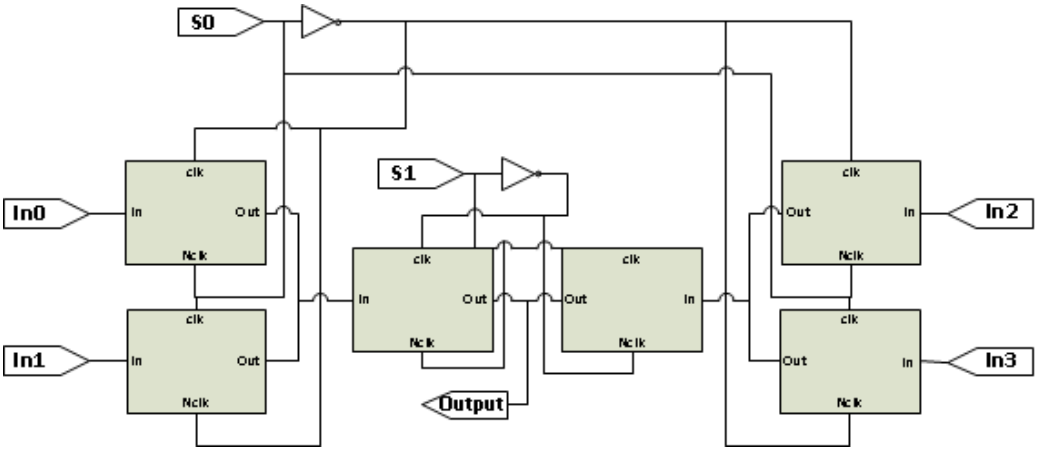

Figure 3.30 4-input multiplexer

As shown in the figure, the multiplexer employs 2 inverters to invert the 2 select bits S0 and S1. Those select bits and their inverted versions are applied to clk or Nclk pins of the transmission gates. According to the combination of these select bits, the circuit passes the relative input to output port. The simulation results of the circuit is shown in Figure 3.31.

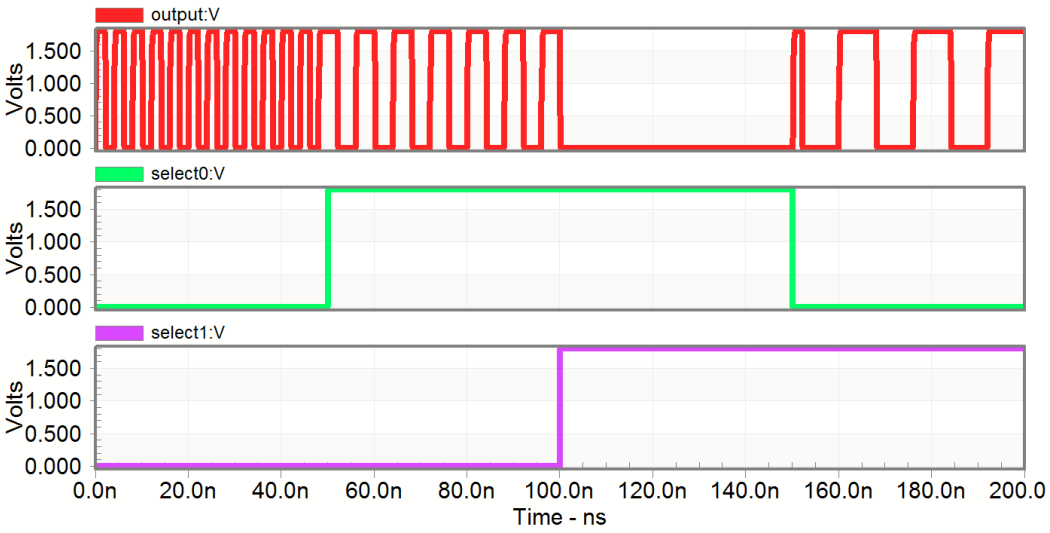

Figure 3.31 Simulation results of 4-input multiplexer

The signal at the first row is S0 and the second signal is S1. The fastest signal is applied to In0 and the signal having half frequency of first is applied to In1. No signal is applied to In2 and a signal having half frequency of second signal is applied to In3. When both select bits are 0, the In0 is passed to output. Similarly, when both select signal are high, In3 is passed to output. It proves that the design is functioning properly. The 8-input multiplexer is designed in the same fashion and simulated also in the same manner.

3.8 Buffer

The buffer is used to get rid of any distortions on the input signal and also to drive higher loads than a standard logic inverter. The buffer is designed to drive the gate capacitance of the high voltage pulser at the beginning, then it is used to drive the input capacitance of the medium voltage pulser circuit. It has 4 inverters connected in series each having different sized transistors. Table 3.3 shows the W/L ratios of the transistors of inverters.

Table3.3 Transistor sizes of buffer inverters

	NMOS (µm)	PMOS (µm)
Inverter1	1/0.18	1/0.18
Inverter2	2/0.18	2/0.18
Inverter3	4/0.18	4/0.18
Inverter4	10/0.18	30/0.18

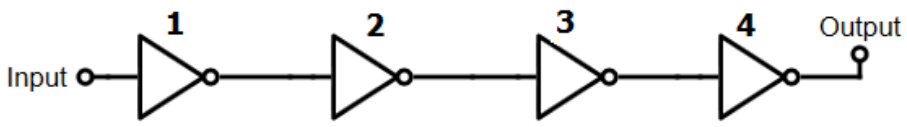

Figure 3.32 Schematic of the buffer

As can be seen through the table, the last inverter have big transistors to drive higher loads. It is not a must to use that buffer, however, loading the next stage with a transmission gate might be dangerous in some cases, especially when the next stage has different type transistor like MV and HV transistors.

3.9 Medium Voltage Pulser

The need for the medium voltage pulser in this beamformer ASIC is occurred due to the high threshold voltages of high voltage transistor. It is almost impossible to bias them with the core voltage of 1.8 volts. While driving high capacitive loads like 2.5 pF at high voltage stage, the designer needs high bias voltage so as to charge and discharge the load at high voltage level in suitable rise and fall times. There are different option to choose in HV 0.18 µm CMOS process such as 3.3 volts, 6 volts and 15 volts. However, the problem is the power budget of the project. Due to the fact that the high voltage or medium voltage supplies are limited, design is limited to 3.3 volts since it is possible to get that much voltage through the processing unit FPGA. In case 6 or 15 volts are used, the designer needs an additional power supply and additional HV connection and ESD protection structure in the layout.

As the medium voltage level shifter, a buffer including 2 inverters are used since biasing MV transistors at 3.3 volts is not that challenging using core voltage. Figure 3.33 shows the block diagram of the MV pulser.

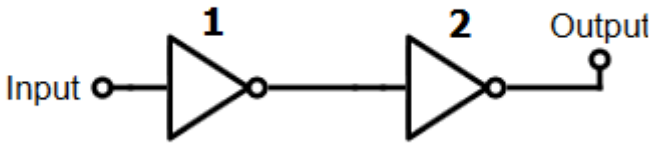

Figure 3.33 Block diagram of the MV pulser

Table 3.4 Transistor sizes of MV pulser inverters

	NMOS (µm)	PMOS (µm)
Inverter1	2/0.35	2/0.3
Inverter2	10/0.35	10/0.3

As stated in table 3.4, the sizes of the transistors are set according to the output load, which is the input transistor of HV stage. The inverter 2 is designed to load the output and the inverter 1 is designed to correctly bias the inverter 2. In short, the first inverter levels up the core voltage signal to 3.3 volts and second inverter drives the output load. There is no need for a special topology to level shift the core voltage signal to 3.3 volts. The following figure shows the simulation results of the medium voltage pulser circuit. The input transistor of the high voltage stage is set as the load of the simulation configuration. The delay introduced by the circuit is 0.6 ns when increasing the signal to 3.3 volts.

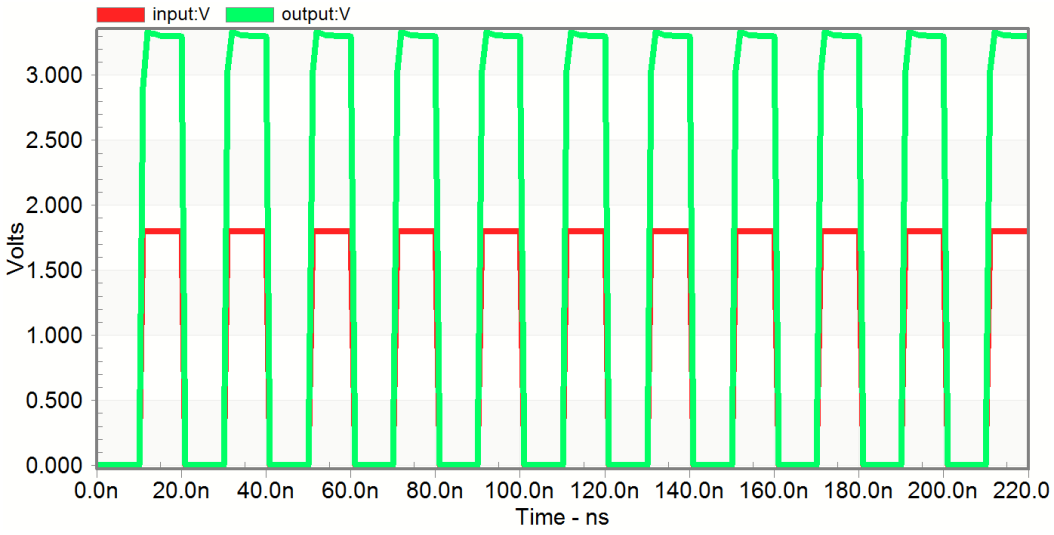

Figure 3.34 Simulation results of the medium voltage pulser

The rise time is 0.5 ns and the fall time is 0.3 ns. The circuit dissipates almost 5 mW of power. The figure of merit (D= delay/ (L*V)) of the circuit is D= 1.01 and it is a good result compared the many of the similar circuits.

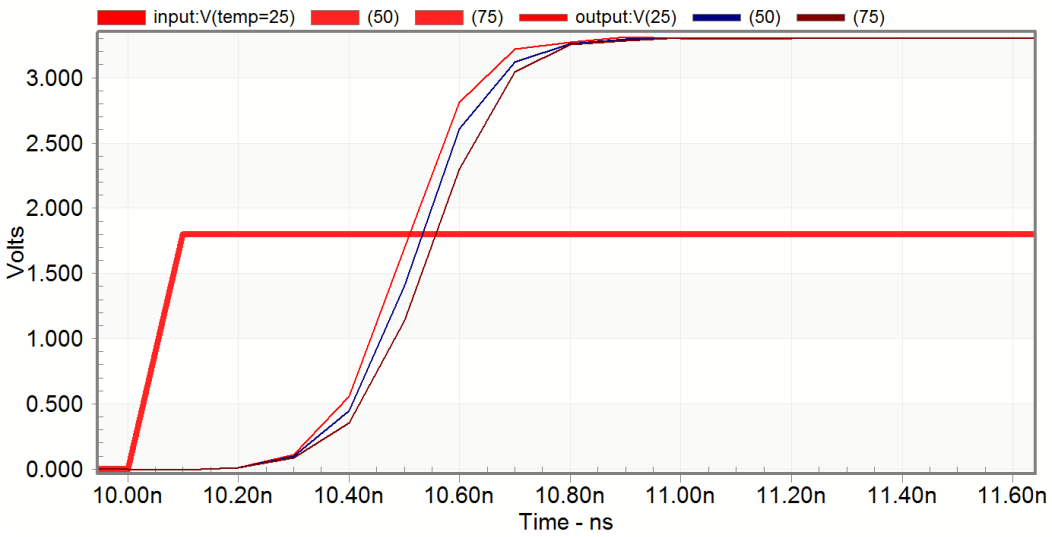

Figure 3.35 Temperature variation simulation of the MV pulser

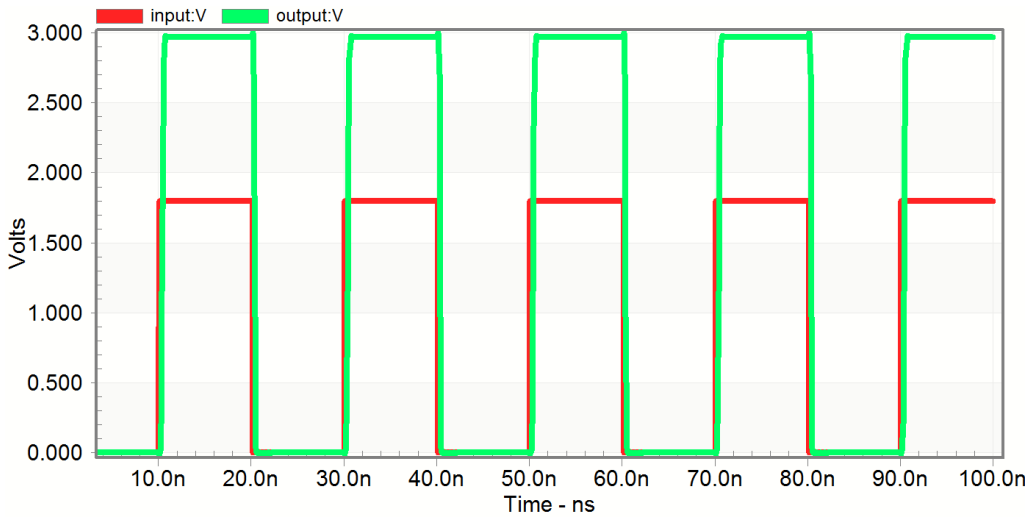

Figure 3.36 Supply variation simulation of the MV pulser

As can be seen through the Figure 3.35, the circuit is not affected by the temperature changes. The delay, rise/fall time of the circuit are almost the same with the normal condition. The simulation is done in 25 ºC, 50 ºC and 75 ºC. Figure 3.36 shows the supply variation simulation when supply is at 2.97 Volts. The delay introduced is not changed, however rise time increases to 0.8 ns and fall time increases to 0.8 ns too.

3.10 High Voltage Pulser

Up to this section, the LV and MV blocks of the beamformer ASIC is discussed in detail. What the LV and MV circuits do is that those circuit generates signals to be fed to high voltage pulser that is explained in this section. The critical point in this application is to have suitable high voltage signals at end of the beamformer chain. As a result, the proper functioning of the HV circuit is very crucial for the LV and MV stage too.

As the CMOS technologies offer HV extensions with LV parts, the designers easily create ASIC to be used in HV application with these technologies. The technology that is preferred in this project offers a few high voltage transistors. The first option is to use high voltage extension modules (NHVE and PHVE) to have high voltage transistors. The NHVE and PHVE modules offer symmetric and asymmetric high voltage NMOS (HVNMOS) and PMOS (HVPMOS) respectively. The second option is to use DMOS (double diffused MOS) module that offers DMOS transistors. However, in order to choose suitable modules for the project, DMOS is not preferred in most of the cases due to the problem with the MPW (multi-project wafer) runs. Among the high voltage extension transistors, the asymmetric transistors are far more suitable and beneficial transistors than the symmetric ones. The threshold voltages of asymmetric transistors are a little bit lower than the threshold voltages of symmetric ones. On the other hand, the drain-source currents defined in the manual of the process is almost 3 times the drain-source currents of the symmetric ones. In addition to that, the I_{ds} of asymmetric transistor is also higher than the I_{ds} of the DMOS one. Briefly, the asymmetric transistor in the NHVE and PHVE modules are selected as the high voltage transistors.

Having discussed the technology offerings, the crucial points in designing high voltage circuits can be explained. First of all, due to the fact that the junction capacitances of a high voltage transistor is really high compared to core transistors, they may have higher switching delay [56], [57]. Due to this drawback of high voltage transistors, those device should not be excited with very high frequency signals. In addition, the power consumption of those devices

may be in a dramatically high degree in comparison with the core devices due to their huge gate area [58]. This is the underlying reason of why the researchers of similar circuits or applications do not share power consumption data of their designs. One more drawback of these device is that to level up the signals to a high voltage level, those devices engage really huge silicon area [56], [57] that results in the reduction in cost effectiveness in some cases. Lastly, those devices may cause a heating problem in the circuit since they draw high currents from high voltage supplies.

There are different topologies used to design such circuits, however, many of them include a typical inverter with diode connected loads. The source follower configuration is also available, however, the low voltage biasing of discharge NMOS transistor at the output stage may cause serious problems such as high fall time at the output or system may not discharge in case of huge loads. The topology that is employed in the design of the beamformer ASIC is shown in Figure 3.37.

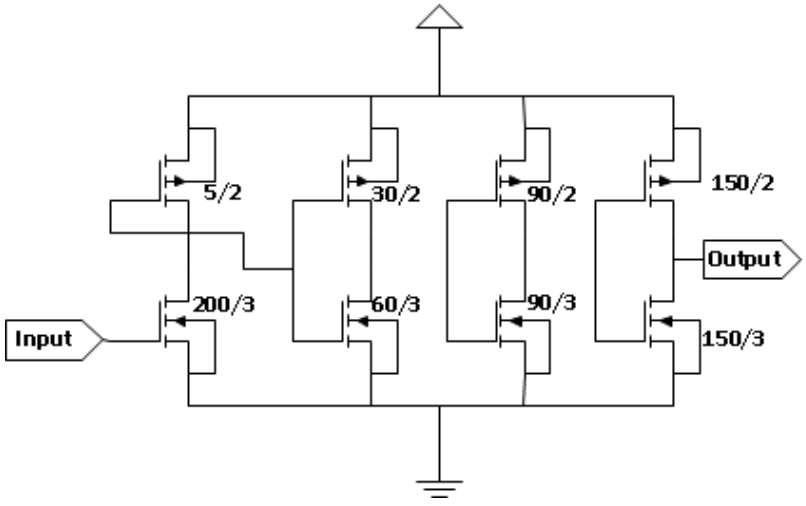

Figure 3.37 HV Pulser configuration

As shown in the configuration, the first step is the level shifting step composed of an HVNMOS and diode connected HVPMOS. The size of the HVNMOS is really huge due to low biasing voltage at the beginning. The following steps are the buffer steps that corrects the signal range and shape in addition to increasing

drive capability. The transistors used in the design is called as “nhv” and “phv” in the technology manual and Figure 3.38 shows the cross-sectional view of both transistors.

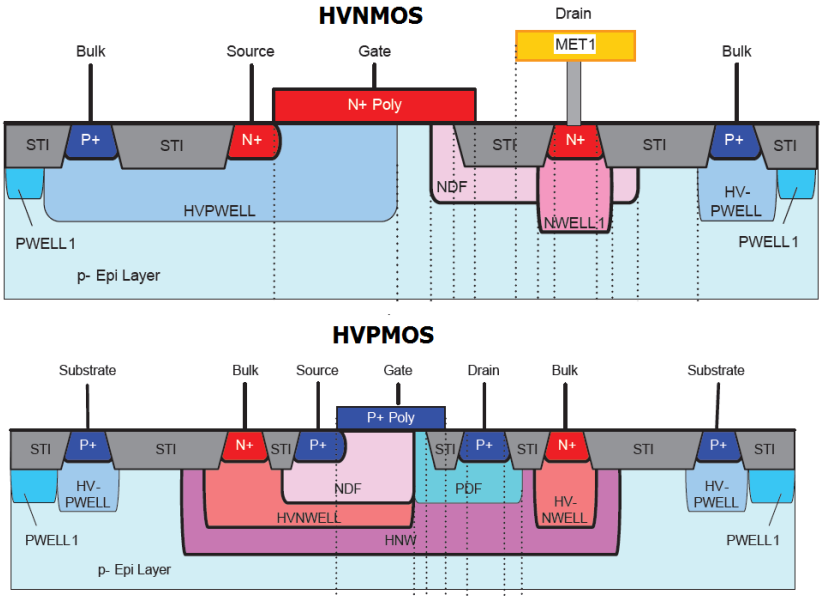

Figure 3.38 Cross-sectional view of HV transistors

The high voltage gate oxide layer is not shown in this figure, since the figure is a basic illustration. The advantage of this configuration is that the first stage of the circuit bias both HVPMOS and HVNMOS in the next step in contrast with generic topologies. In generic topologies, the inverter with the load is used to bias only the HVPMOS transistor. By using this hint, the sizes of NMOS transistors are reduced.

Following the introduction, next step is the simulation results part. Most of the designers simulate their high voltage pulser circuits with a 2.5pF load so as to realize a CMUT element, however, there are some transducers having higher capacitance values. Although the circuit is designed to drive a 5.1 pF load, the simulation results represent the configuration with 2.5 pF load so as to have a comparable result. The other simulation results follow the comparison step. The Figure 3.39 illustrates the simulation results with 2.5 pF load capacitance with the signal of core level and medium voltage level.

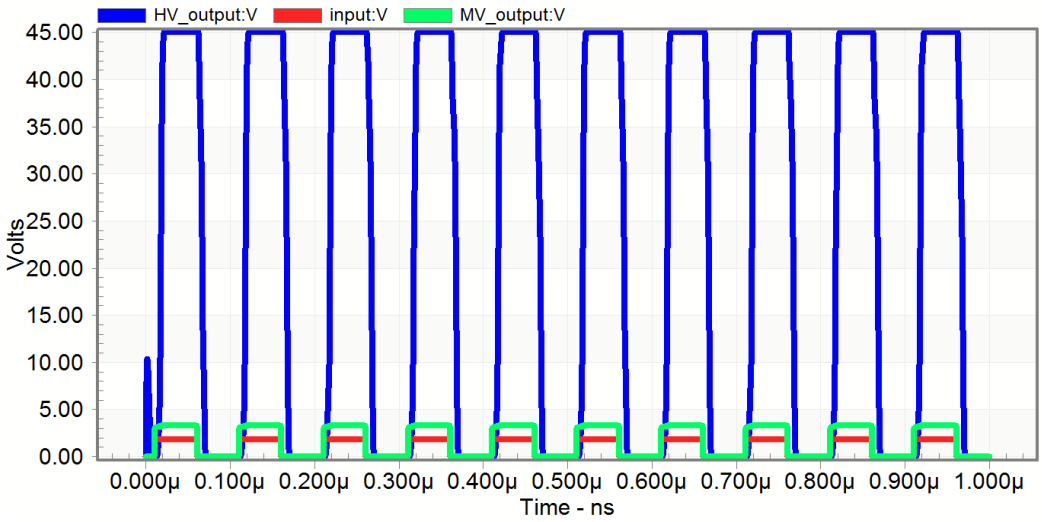

Figure 3.39 Simulation results of the HV pulser (10 MHz)

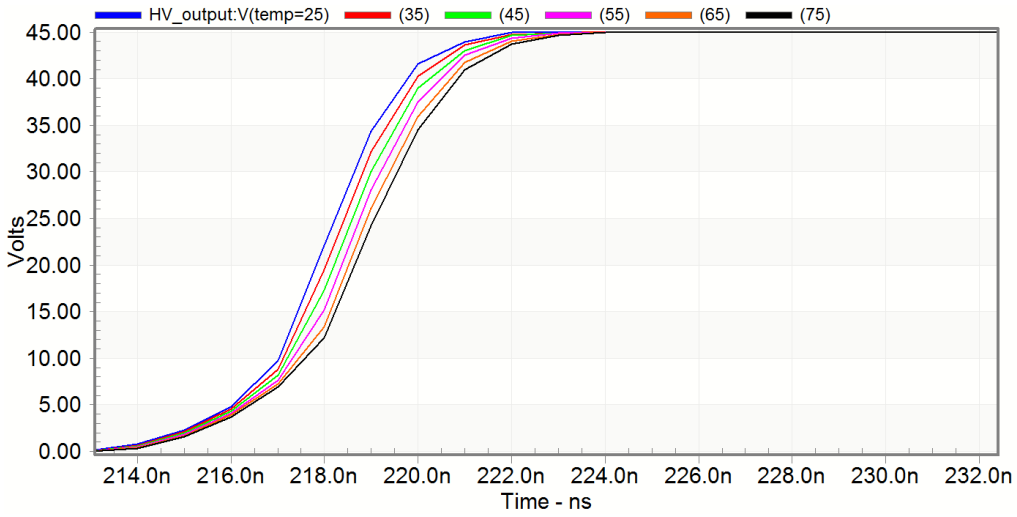

Figure 3.40 Temperature sweep simulation of the HV pulser

The simulation results proves that the circuit can easily drive a load of 2.5 pF. The delay of the circuit 4 ns, the rise time is 4.2 ns and the fall time is 5.3 ns. The resulting figure of merit is 0.49 which is really a sensitive value among its counterparts. The power dissipation in this configuration is 148.5 mW. For the temperature variations, a linear sweep simulation is done and shown in Figure 3.40. As shown in the figure, the temperature variations almost do not affect the circuit performance. The delay is at 4 ns for 25 ºC and 4.16 ns for 75 ºC which results in a 4% change which is acceptable.

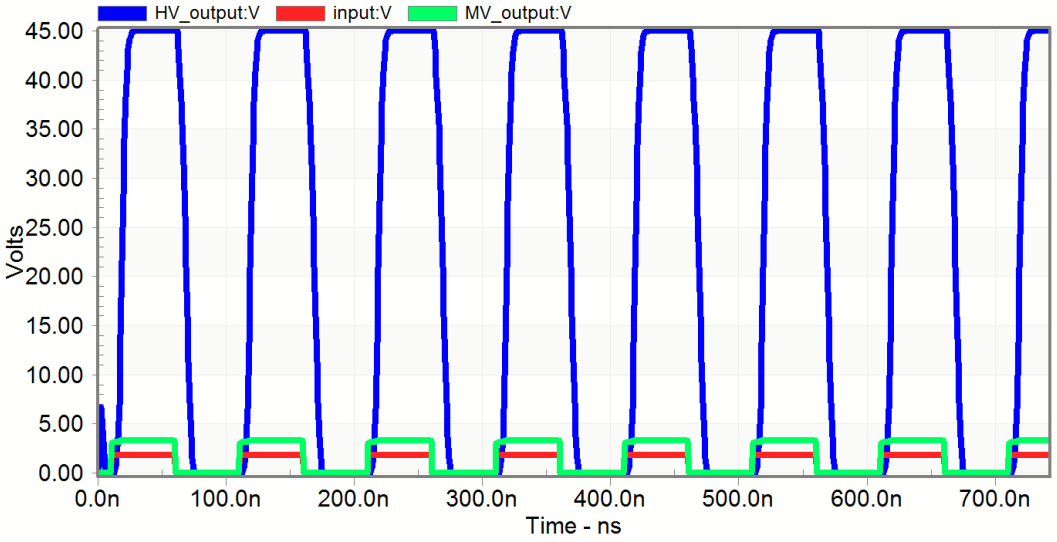

Figure 3.41 Simulation results of HV pulser with 5.1 pF load (10 MHz)

The Figure 3.41 illustrates the simulation results of the circuit while driving a load of 5.1 pF. The delay is again 4 ns, rise time is 7.8 ns and fall time is 7.8 ns too. The figure of merit does not change, 0.49.

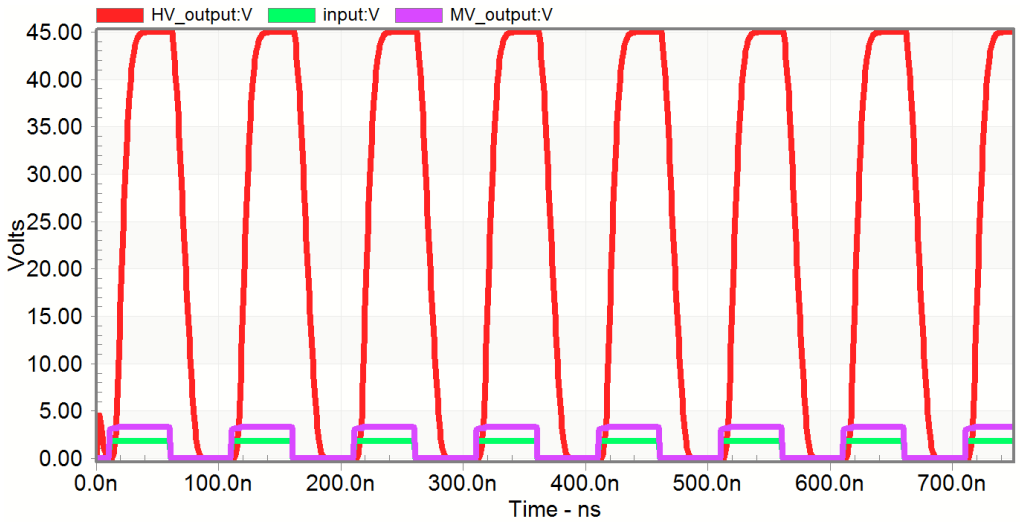

Figure 3.42 Simulation results of HV pulser with 10 pF load (10 MHz)

As Figure 3.42 shows, when the load is 10 pF, the delay is 5 ns, rise and fall times are 13.3 and 17 ns. The figure of merit is 0.62 this time. The frequency of the signal applied to the circuits is 5 MHz in all simulation cases.

Table 3.5 Comparison of similar HV Pulsers

	Prior Work	Year	Tech Type	Tech Node L(µm)	Delay D (ns)	Voltage V (V)	Figure of Merit
1	Buyle et al. [59]	2008	HV CMOS	0.35	2.5	25	0.29
2	Moghe et al. [46]	2010	HV CMOS	0.35	2.4	10	0.69
3	Maadi [2]	2013	HV CMOS	0.35	8	45	0.51
4	Maadi [2]	2013	HV CMOS	0.35	5.8	15	1.08
5	Present work	2015	HV CMOS	0.18	0.6	3.3	1.01
6	Present work	2015	HV CMOS	0.18	4	45	0.49

The table 3.5 is comparison table that shows the results of each design by different researchers. The designed level shifters are really functional compared to their counterparts.

3.11 CMUT Element DC Biasing

The DC bias requirement of CMUT arrays is discussed in previous chapters. In order to operate the CMUT array properly, the flexible plate of the transducer element must be forced to move to the operation point by the application of a DC bias. The DC bias scheme of the CMUT elements is shown in Figure 3.43 with the required elements.

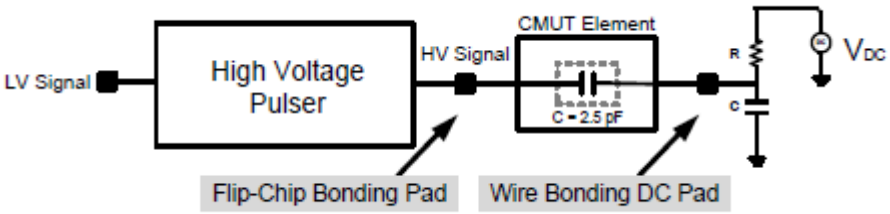

Figure 3.43 DC biasing scheme of CMUT elements [2]

The DC biasing of the transducers elements are realized through specially designed pads in this design. Those pads are connected to top plates of the

CMUT elements which are common to all elements in the array. The AC signal supply is fed to other plate of the CMUT elements. The resistor and capacitor shown in the figure is in the order of several hundred kilo ohms and hundreds of nanofarads respectively [2]. The resistor is used for short circuit protection reasons and the capacitor is used to filter any noise contributed by the DC supply [2]. Besides, the capacitor behaves like open circuit to DC signal and provide an AC ground to the pad. In this thesis, the pads designed for DC biasing reasons will be introduced in the next chapter.

3.12 Summary of the Chapter

Throughout this chapter, the design procedure of the required blocks are introduced in detail. The schematics, simulations results and some comparisons are given in the related sections. As described in the chapter, the beamformer can be divided into 3 main groups as low voltage part, medium voltage part and the high voltage part. The design of each main block has been explained in detail in this chapter. The underlying reason of why those circuits are employed and why the topologies used are chosen is described. The operational flow of the beamformer is also introduced. The operational principles of all circuits are given and the functioning test as simulations are done. According to the simulation results presented in this chapter, all blocks are ready for layout organization of the beamformer ASIC.

During the design phase, the generation of required signals, power dissipation of the generator or combinational logic circuits and the high voltage feeding are the main criteria that the design tries to meet. The first goal is to design the LV block of the beamformer to generate required signals and logics while not exceeding the power consumption budget. Actually, there is not any predefined power budget, however, since the power is drawn from the FPGA side, the design should not try to get a current more than the FPGA can supply. This is the first goal that is achieved. Secondly, the medium voltage is also taken from FPGA, but the problem here is to design a circuit that dissipates low power while driving the junction capacitances of a high voltage transistor. Fortunately, there is one MV circuit and the goals are easily achieved. The biggest problem during the design

was the high voltage block. The technology used herein (HV 0.18μm CMOS) offers a few high voltage transistors and it is really challenging to meet the driving requirements with those transistors. However, by using big devices and designing the high voltage pulser in problem-solving fashion, the high voltage requirements are also met.

In addition, the beamformer is designed to interface many different type of CMUTs by the help of high load driving capability, low power consumption, digital control from a processor and compatible topology for both imaging and therapeutic applications.

The next chapter introduces the layout design of the circuits described in this chapter, and interconnection techniques used to interface CMUT array and FPGA are also discussed. The CMUT array to be integrated with ASIC designed herein is also introduced.

CHAPTER 4

4. DESIGN AND PERFORMANCE OF THE FRONT-END ELECTRONICS FOR CMUTs

This chapter is the definition of the chip architecture in terms of the physical layers used to fabricate those circuits. The HV 0.18μm CMOS technology is used as the design and fabrication technology. The circuits described in previous chapters are illustrated with their physical implementations in this chapter. The importance of this technique to create a beamformer application specific integrated circuit is because of the easy integration with CMUT transducer arrays as stated. In this chapter, the CMUT arrays to be integrated with described and designed beamformer ASIC is also presented. Although the designed IC here is designed according to 2 specific CMUT arrays, the circuit of low voltage, medium voltage and high voltage can be employed in a different configuration for different CMUTs. Due to the fact that the IC is designed as a flexible beamform generator, it can also be employed in transceiver IC for imaging applications.

All the design techniques, measurements, cell sizes and other information about the layout implementation is presented in this chapter. The I/O and supply connection techniques and the circuits forming those blocks are also described. Since those circuits are predefined and taken from the libraries of the technology, they are not described in previous chapters. The bonding mechanisms are also presented in this chapter. Two beamform generators for two similar CMUT arrays and a test cell will be prepared at the end of this chapter.

Firstly, section 4.1 is the description of the layout overview and presentation of the CMUTs to be integrated. Section 4.2 continues with explanation of low voltage part by illustrating the layouts of the circuits contained in LV block. This section also includes the explanation of the MV pulser circuit. Section 4.3 is the presentation of the assembly of LV blocks. Section 4.4 describes the high voltage pulser circuit layout. Section 4.5 presents the bond pad structures used in this design and section 4.6 clarifies the I/O topologies used in the beamformer ASICs. This section includes supply connections and their layout too. Section 4.7 is the presentation of beamformer ASIC for 4x4 element array and section 4.8 is the presentation of the ASIC for 4 element array. Section 4.9 presents the single row beamformer test cell that is designed to test the employed topologies. Section 4.10 is the summary and discussion of this chapter.

4.1 Overview of Layout

The aim of this thesis is to design flexible beamformer application specific integrated circuit for ultrasound therapeutic applications to be integrated with the annular CMUT arrays presented in [60]. There are two types of annular CMUT arrays in [60]: a 4x4 annular CMUT with each ring divided into 4 elements and a 4 element annular CMUT array composed of 4 rings each inside the other. The configuration of CMUTs are shown in Figure 4.1 and Figure 4.2.

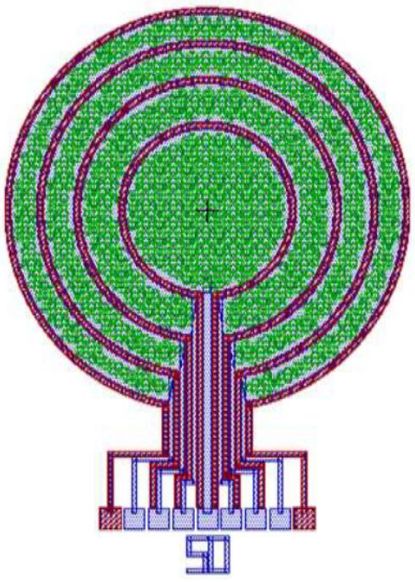

Figure 4.1 4 element annular CMUT array [60]

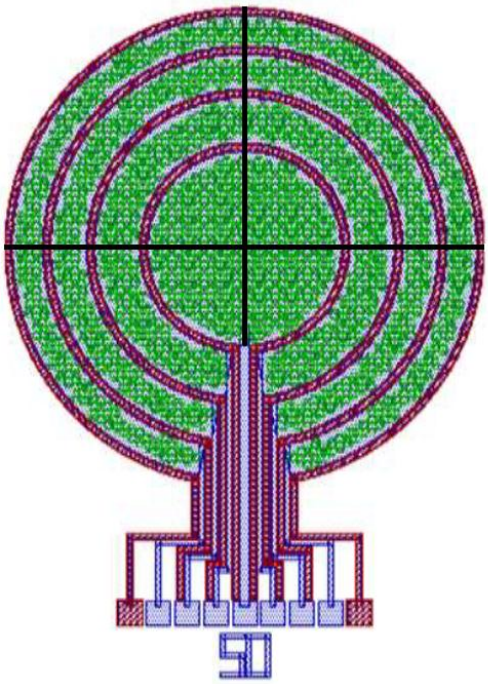

Figure 4.2 4x4 element annular CMUT array [60]

The beamformer ASICs are designed to match the CMUT arrays shown in Figure 4.1 and 4.2. Actually, the 4x4 array has its connections in different places, but the figure is a replica of the 4 element array to give an understanding of the mechanism. As stated in the previous chapters, the high voltage pulser circuit is designed to drive the elements of those CMUT arrays. According to the sizes of plates depicted in [60], a CMUT element can be approximately modeled as 5.1 pF capacitor. The microfabrication procedures, the design techniques and theory of the operation can be found in [0].

The low voltage, medium voltage and high voltage blocks are placed according to the bonding mechanisms with the presented CMUTs. The 4 element CMUT array has its pads under array itself and the 4x4 array has bonding mechanism under the element itself, as a result, the bonding pads are linear in 4 element orientation and in a cross-shaped orientation in 16 element array. As stated previously, the DC biasing pads are common to all elements.

The LPMOS main module is used for core transistors and MV transistor whereas PHVE and NHVE modules are used for HV block. In the test cell, a capacitor is realized with using MET5 module because it's a sandwich capacitor that can bear 45 Volts. Metal 1 to metal 4 is used as the interconnects of the chip. The bonding pads include metal 1 to metal 3 layer and top metal and thick metal layers. As a result, METTP and METTHK modules are also employed. The other metal layers in between are not used in bond pads.

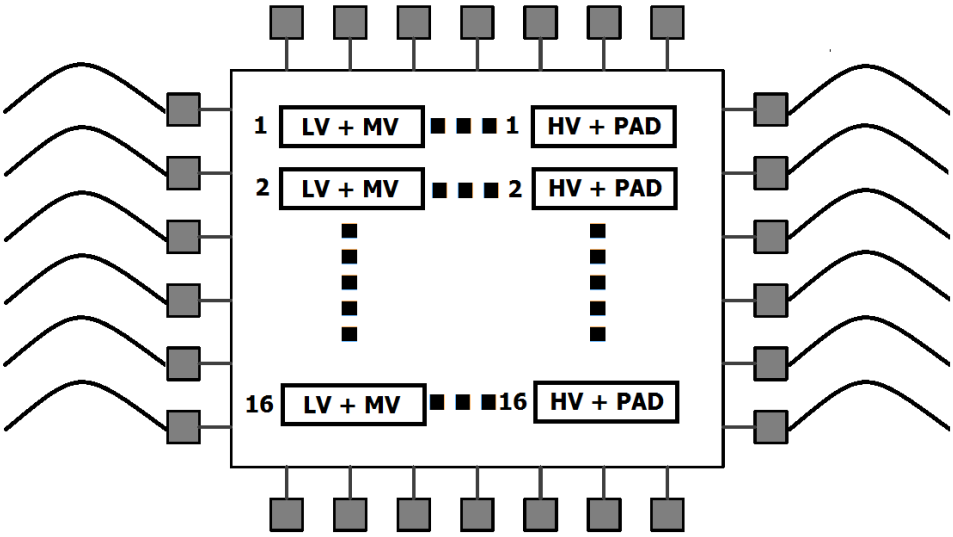

Figure 4.3 Overview of the IC

Figure 4.3 illustrates the general overview of the ASICs designed. The LV and MV blocks are placed in one side and the HV block and the Pads are placed on the other side in general. The low voltage and medium voltage connections are done through I/O pads, however, the high voltage connection is realized with a special cell containing standard thick top metal pad and high voltage ESD (electrostatic discharge) protection circuits.

4.2 Layout of the Low Voltage Block

In this section, the layouts and organizations of the driver circuits that are the demultiplexer, oneshot circuit, digitally controlled oscillator, frequency down

conversion circuit, multiplexers, buffer and a medium voltage level shifter are illustrated. The interconnections and design procedure is also explained.

4.2.1 Demultiplexer

As stated previously, the beamformer starts functioning from the demultiplexer stage. The input signal from the FPGA is fed to demultiplexer and this circuit feeds the following two circuits depending on its select bit.

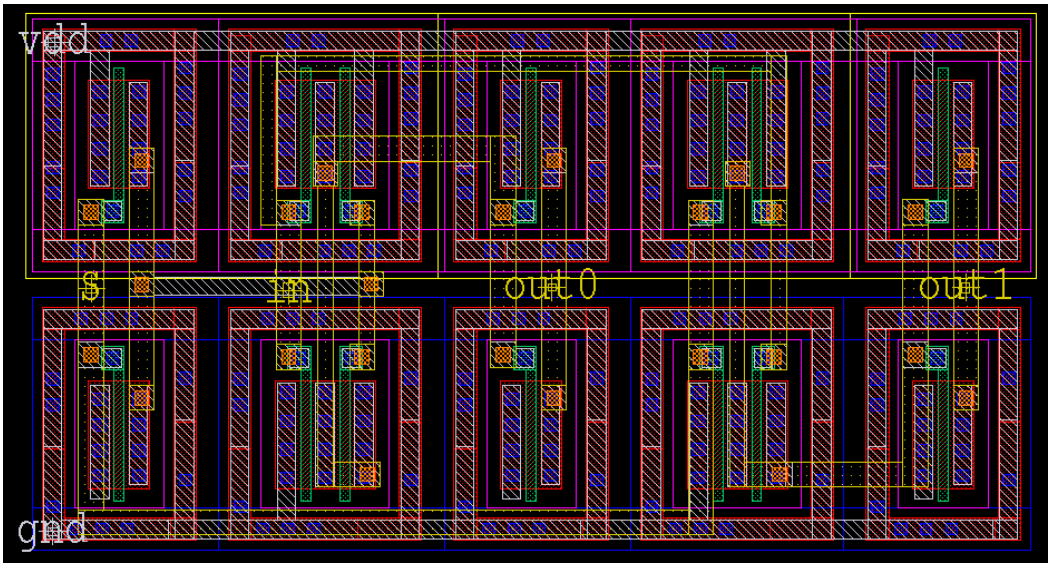

Figure 4.4 Demultiplexer layout

Figure 4.4 shows the layout organization of the demultiplexer. The first two transistors are the inverter transistors for the select bit (s) and the third and fifth two transistors are the inverter transistor for the outputs. The wider transistors are the three parallel connected PMOS and three series connected NMOS transistors forming the 3-input NAND gate. The area of the circuit is 181.55 μm^2 (17.84 µm x 9.96 µm) and it has a density of 97.26%. The interconnections are done using metal1 and metal2 layers. The DRC, Extraction and LVS procedures are done without any problem.

4.2.2 Oneshot Circuit

The oneshot circuit is one of the fundamental circuits of the beamformer ASIC that generates single pulses with adjustable widths. The input of this circuit is fed by demultiplexer and it sends its output to 4-input multiplexer.

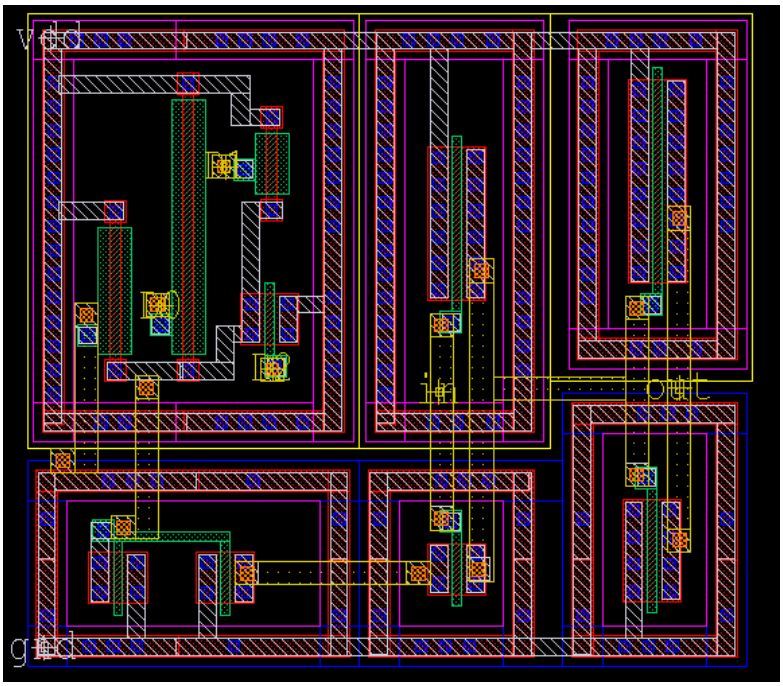

Figure 4.5 Oneshot circuit layout

Figure 4.5 shows the layout of the circuit. The top left most 3 transistors are the controlled current source transistors and the other transistor near them is the fixed transistor of the current source. The current mirror transistors are shown at the bottom and the other transistors are the part of inverters. The area of the circuit is 178.47 μm^2 (14.22 µm x 12.78 µm) and it has a density of 98.17%. The interconnections are done using metal1 and metal2 layers. The DRC, Extraction and LVS procedures are done without any problem.

4.2.3 Digitally Controlled Oscillator

The digitally controlled oscillator is the other fundamental circuit of the beamformer ASIC which includes an enable circuit (D flip-flop), 3 digitally

controlled delay elements and other combinational logics. It is also fed from demultiplexer and sends its output to frequency down converter.

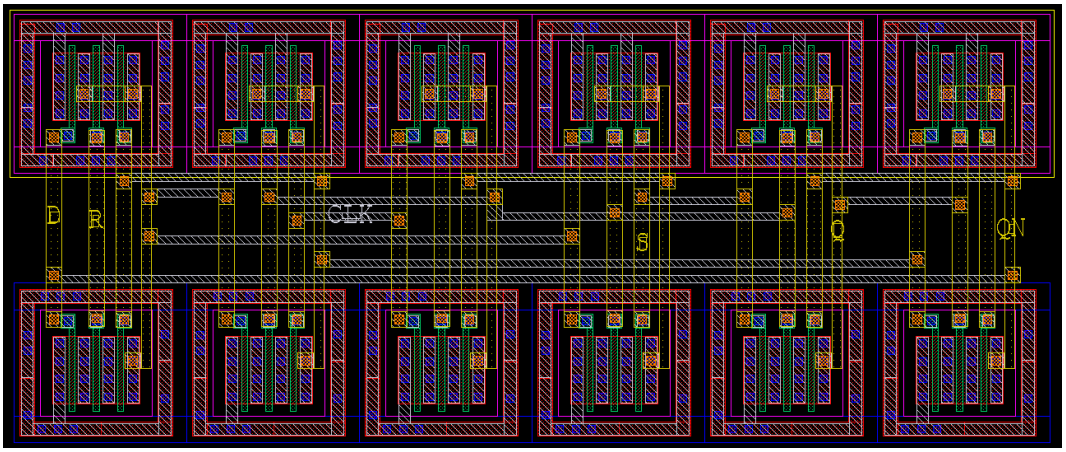

Figure 4.6 Enable of DCO

Figure 4.6 shows the enable circuit of the DCO. It is a D flip-flop formed by six 3-input NAND gates which are obvious in the figure. The area of the circuit is 340.06 μm^2 (30.72 µm x 12.7 µm) and it has a density of 87.16%. The interconnections are done using metal1 and metal2 layers. The DRC, Extraction and LVS procedures are done without any problem.

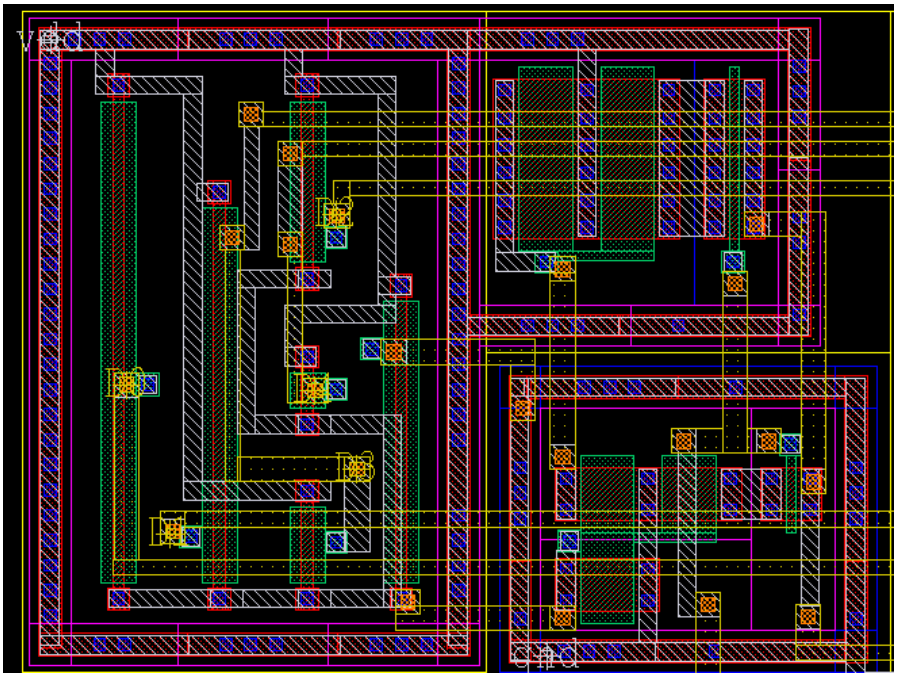

Figure 4.7 Digitally controlled delay element layout

Figure 4.7 illustrates the DCDE of the DCO. The left most 6 transistors are the current source transistors. The right most two transistors form the inverter of the DCDE. The remaining transistors are the current mirror transistors. The area of the circuit is 187.71 μm^2 (16.01 µm x 12.7 µm) and it has a density of 92.29%. The interconnections are done using metal1 and metal2 layers. The DRC, Extraction and LVS procedures are done without any problem.

In the DCO designed for the beamformer ASIC, there are 3 digitally controlled delay elements, 1 2-input NAND gate and 1 inverter as the loop elements. The loop is followed by a buffer stage formed by 4 inverters. The enable circuit of the DCO is accepted as an external circuit, however, it is a part of the circuit.

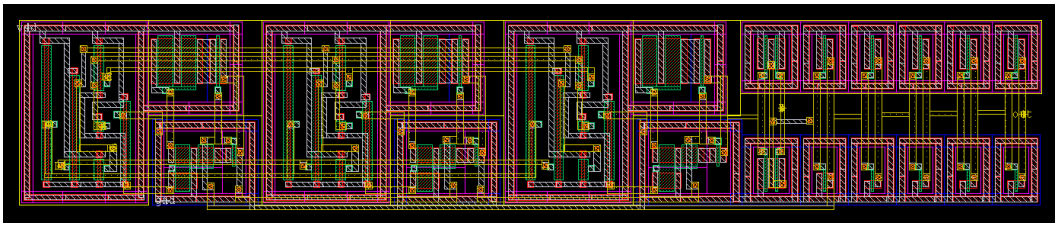

Figure 4.8 Digitally controlled oscillator layout

Figure 4.8 shows the layout of the DCO. The 3 DCDEs are placed in the left side and the 2-input NAND gate and the buffer stage is placed in the right side. The area of the circuit is 809.9 μm^2 (68.58 μm x 12.7 μm) and it has a density of 92.98%. The interconnections are done using metal1 and metal2 layers. The DRC, Extraction and LVS procedures are done without any problem.

4.2.4 Frequency Down Conversion Circuit

The frequency down conversion circuit is fed from DCO and sends its outputs to the 8-input multiplexer.

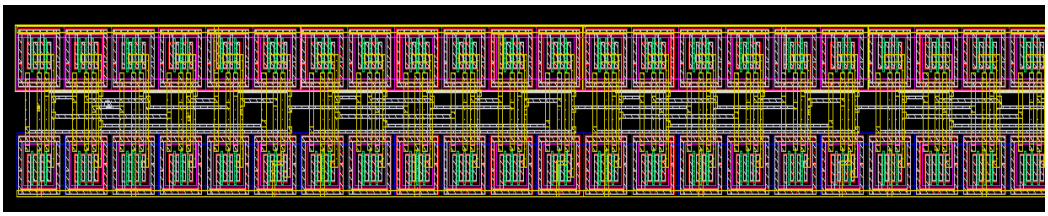

Figure 4.9 Frequency down conversion circuit layout

The circuit is formed of 8 D flip-flops connected in series and the layout of the D flip-flop is shown in Figure 4.6. The yellow parts in the figure is because of metal2 interconnection and density of that connection make the figure complex. The area of the circuit is 2723.1 μm^2 (244.08 μm x 12.7 μm) and it has a density of 87.84%. The interconnections are done using metal1 and metal2 layers. The DRC, Extraction and LVS procedures are done without any problem.

4.2.5 Multiplexers

There are two multiplexers in this design and one of them have 8-inputs and the other have 4-inputs.

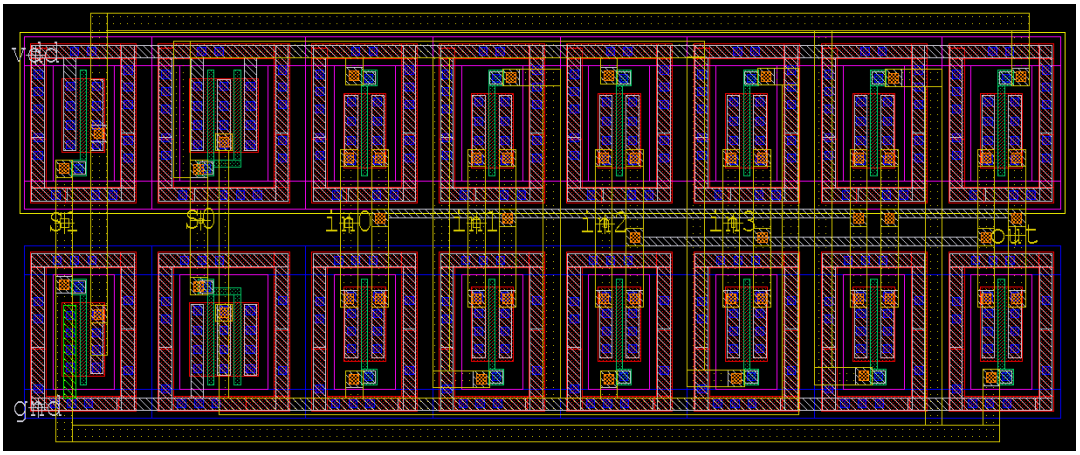

Figure 4.10 4-input multiplexer layout

First, the 4-input multiplexer is shown in Figure 4.10. The inverters of the select bits are placed in the left side and the transmission gates are placed in the left side. The area of the circuit is 306.21 μm^2 (28.4 µm x 11.62 µm) and it has a density of 92.79%. The interconnections are done using metal1 and metal2 layers. The DRC, Extraction and LVS procedures are done without any problem.

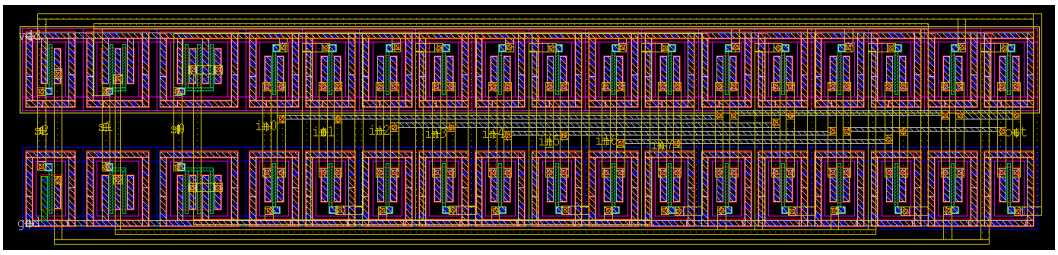

Figure 4.11 8-input multiplexer layout

Similar to the 4-input one, 8-input multiplexer is shown in Figure 4.11. The inverters of the select bits are placed in the left side and the transmission gates are placed in the left side. The area of the circuit is 664.26 μm^2 (58.16 µm x 13.08 µm) and it has a density of 87.28%. The interconnections are done using metal1 and metal2 layers. The DRC, Extraction and LVS procedures are done without any problem.

As stated in previous chapters, one of the inputs of the 4-input multiplexer is tied to ground in the final version of the low voltage driver block in order not to consume energy through the level shifters in the idle mode. One of the other inputs is directly fed from the FPGA side for the user specific applications. The other two inputs are connected to oneshot and digitally controlled oscillator circuits as states. In addition, the above illustrated 8-input multiplexer gets all of its inputs from the frequency down conversion circuit.

4.2.6 Buffer

The buffer of the low voltage driver block is designed to drive higher loads at the core voltage level. The circuit gets its input data from the 4-input multiplexer and feeds the input of the medium voltage level shifter circuit. The buffer is designed to drive higher loads than the input capacitance of the medium voltage pulser in order to be suitable with the applications where 6 volts or 15 volts are used as the medium voltage. Briefly, the buffer is capable of driving junction capacitance of 6V and 15V transistors, especially in the technology used herein.

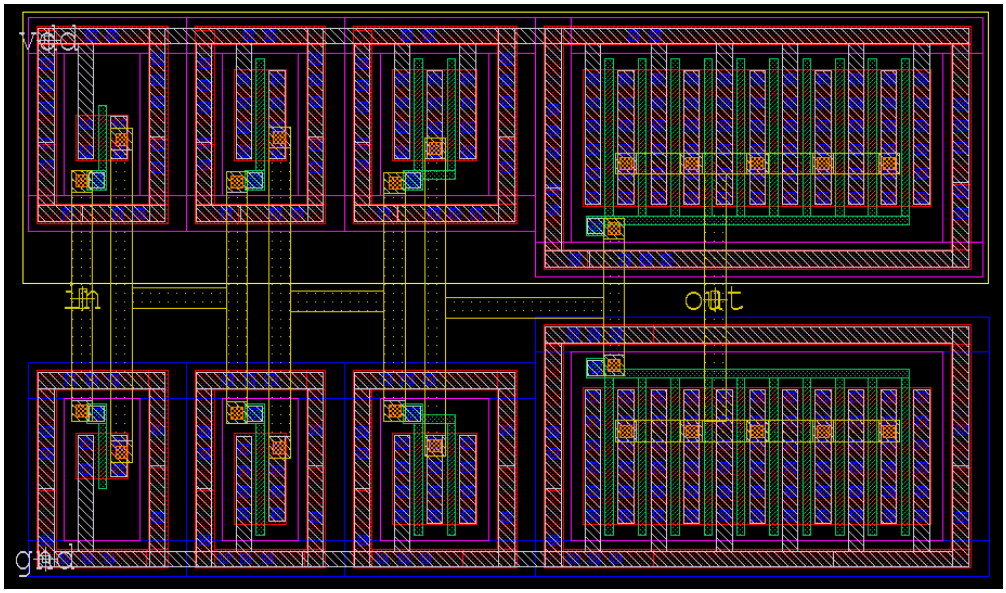

Figure 4.12 Buffer layout

The buffer consists of 4 inverters with different transistor sizes. The area of the circuit is 242.33 μm^2 (21.16 μm x 12.33 μm) and it has a density of 92.84%. The interconnections are done using metal1 and metal2 layers. The DRC, Extraction and LVS procedures are done without any problem.

4.2.7 Medium Voltage (MV) Pulser

The medium voltage pulser is the last stage in the driver part of the beamformer ASIC. The topology is a simple buffer consisting of two inverters. The input voltage level is 1.8 volts (core voltage) and the output voltage level is 3.3 volts (medium voltage). Actually, the 3.3 volts is called as low voltage too, however, this level is called as medium voltage in the HV 0.18μm CMOS technology.

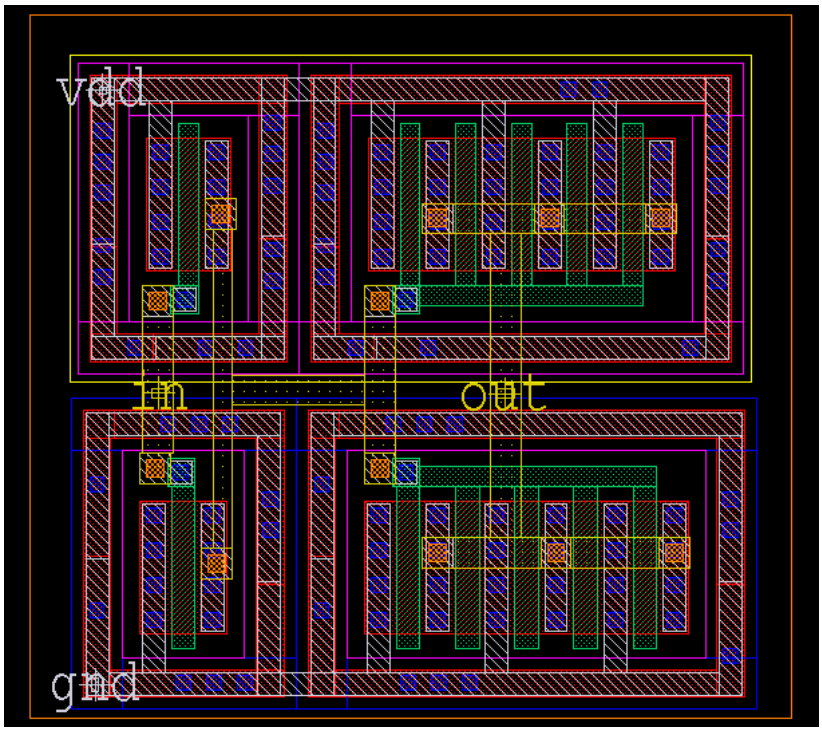

Figure 4.13 MV Pulser layout

As can be seen from Figure 4.13, the circuit contain an additional MV layer as a difference from the other buffers. The description of the MV layer in the technology manual is 3.3 volts gate oxide. The area of the circuit is 121.57 μm^2 (11.48 μm x 10.59 μm) and it has a density of 100%. The interconnections are

done using metal1 and metal2 layers. The DRC, Extraction and LVS procedures are done without any problem.

4.3 Low Voltage Block (Driver) Assembly

As the low voltage and medium voltage blocks are explained, the assembly of the subcircuits is possible and defined in this section. As a recall, the driver consists of demultiplexer, oneshot, DCO, FDC, multiplexers, buffer and the MV pulser.

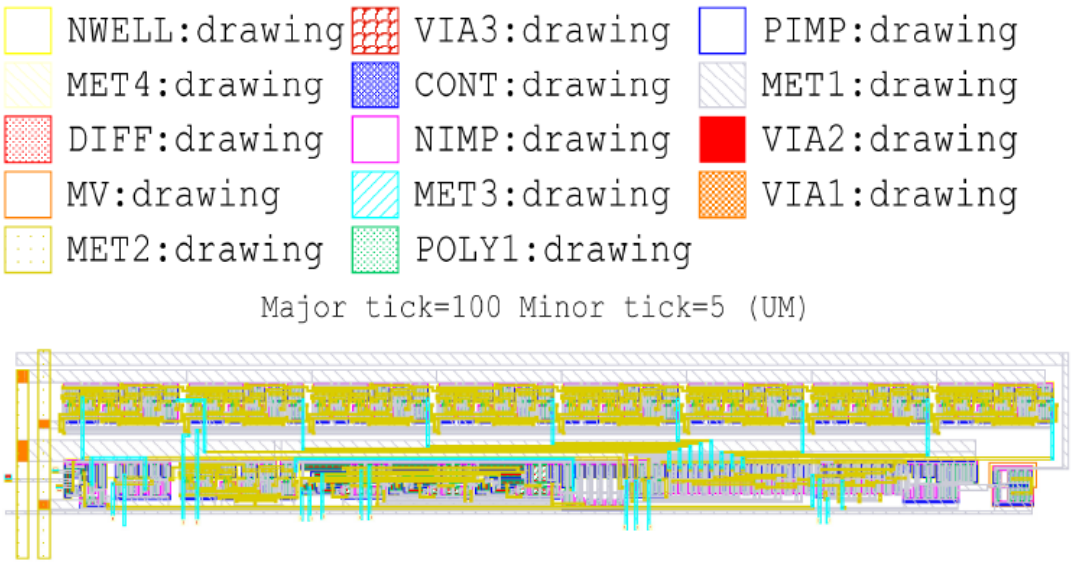

Figure 4.14 Driver layout

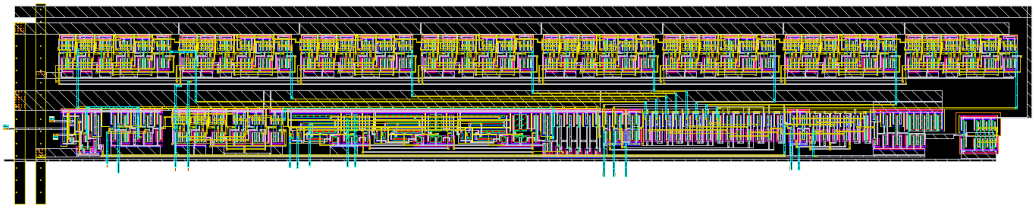

Figure 4.15 Driver layout without layer definitions

The architecture of the driver and the layers employed can be seen from Figure 4.14 and Figure 4.15. The upper row is the frequency down conversion circuit only. The lower row includes oneshot circuit, demultiplexer, enable of DCO, DCO itself, 8-input multiplexer, 4-input multiplexer, buffer and the MV pulser from left to right. The internal connections of the subcircuits are done using metal1 and metal2. For the connections between circuits, metal3 and metal2 wherever possible is used. The metal 4 layer is not used as interconnect in driver part, yet the connection between drivers are done using metal4. As a result, metal4 connections are prepared before the top cell step. This is the underlying reason of metal4 usage depicted in Figure 4.14. The area of the driver is 5259.28 μm^2 (211.16 µm x 41.16 µm) and it has a density of 60.51%. The interconnections are done using metal1 and metal2 layers. The DRC, Extraction and LVS procedures are done without any problem. The area of the driver is actually a little bit less than the stated value because of the power rail and interconnect area that is included in the stated value. The density of the driver is also higher actually. The LV driver can be supplied easily by preparing common rails for common inputs and drawing proper connections.

4.4 High Voltage Pulser

The high voltage extension module transistors are introduced in the previous chapter with a cross-sectional view. A sample view of the implemented asymmetric transistors is shown in Figure 4.16.

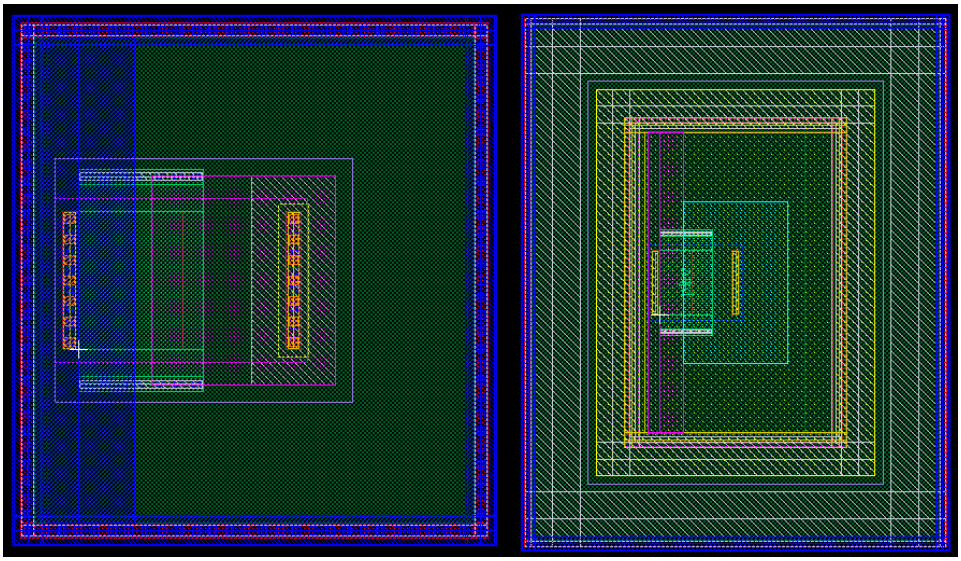

Figure 4.16 HVNMOS (left) and HVPMOS (right)

The transistors in Figure 4.16 are the predefined ones in the library with smallest possible values and not scaled in the same manner. The actual size of smallest HVPMOS is almost 2 times the smallest HVNMOS. The cross-sectional view provided previously shows the orientation of layers. The guard ring used in the design is a must for both transistors by the rule set of the technology. By using combination of those transistors and their multiple gate derivatives, the circuit in Figure 4.17 is designed.

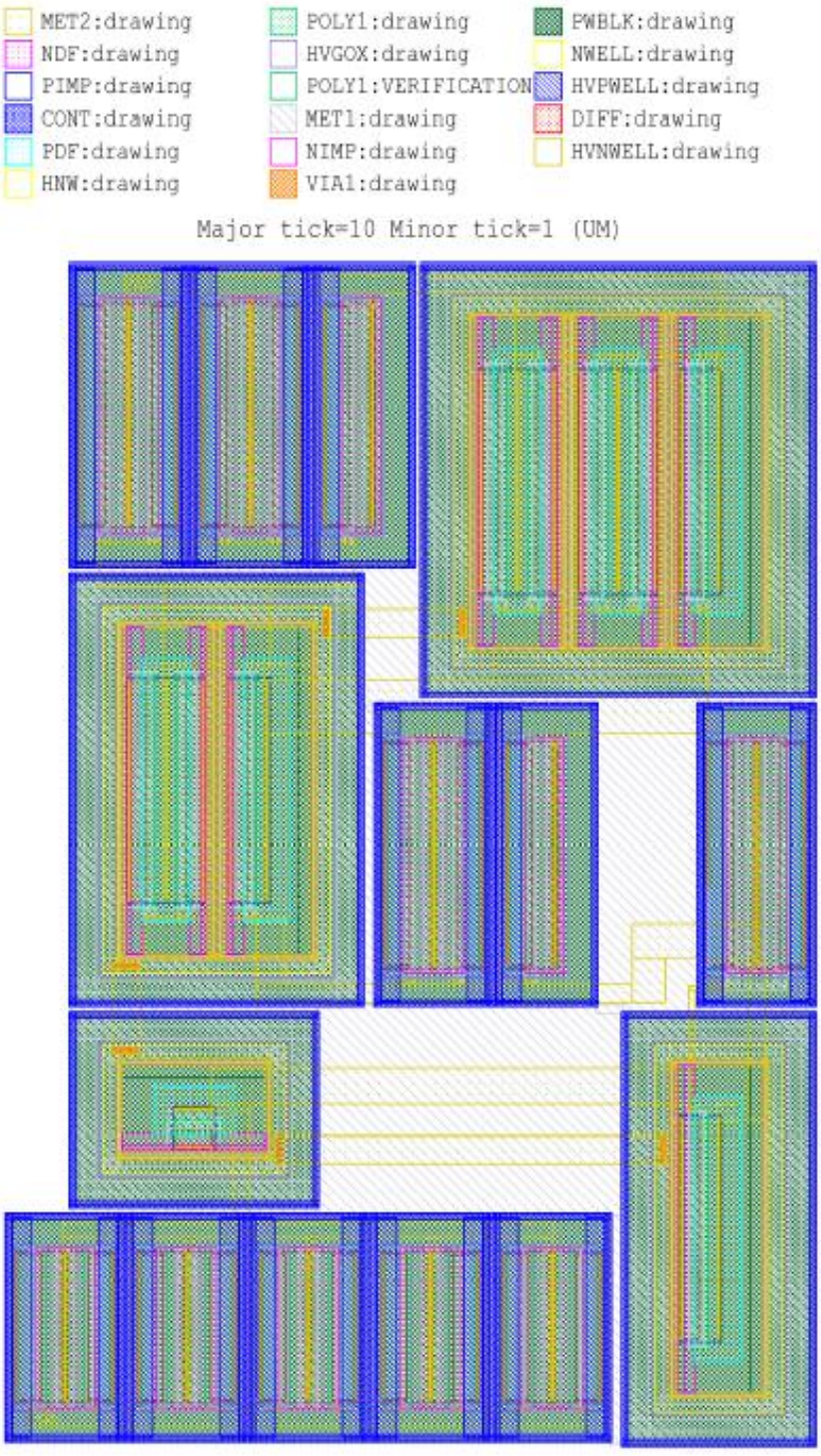

Figure 4.17 HV Pulser layout

The transistors used in this circuit is designed separately, they are not designed by the t-cell generator or any other automatic device generator tool. Automatic scaling is not possible for any high voltage transistor in this technology yet. The

area of the HV pulser is 16654.9 μm^2 (110.73 µm x 161 µm) and it has a density of 93.42%. The interconnections are done using metal1 and metal2 layers. The DRC, Extraction and LVS procedures are done without any problem. The area used in these transistors describes why their junction capacitances are really high.

4.5 Bond Pads

As stated in previous chapters, in order to interface the beamformer ASIC that is designed here, the chip requires interconnection pads. The pads are designed for two reason: flip-chip bonding and wire bonding. The pads designed here have 3 different sizes all having same square shape. The pads used in I/O connections and HV supply connections are taken from the standard library which are predefined. All of the pads used in this design has 5 layers connected with relevant vias: metal1, metal2, metal3, top metal and thick top metal. The reason of why metal4 or metal5 is not used is that each additional layer introduces more parasitic capacitance and especially for the pads driven by HV pulser, this may cause loading harder. Even though this parasitic capacitance is very low compared to load capacitance, there is no need to use them.

Figure 4.18 Pad layout

As stated in the beginning of the section, 3 different bond pads are designed with same shape. The first one is 80µm x 80µm, the second one is 200µm x 200µm and the last one is 250µm x 250µm. The first two is employed in the ASIC for

4x4 annular array and the third type is used for 4 element annular array. The definition “USERPAD” on the pad is a must enforced by the process and described in the process manual. The designer must define the type of pad according to usage of it. For instance, if the user needs that pad for measurements, “PROBEPAD” should be used as definition. Otherwise, the designed pad cannot pass the DRC check.

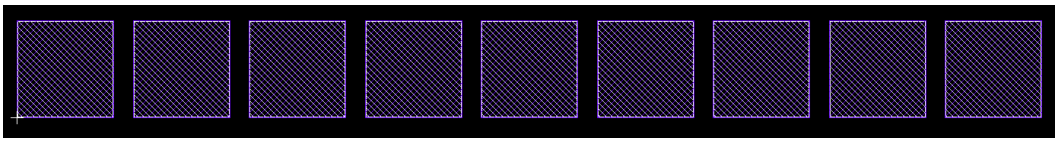

Figure 4.19 250µm x 250µm pads for 4 element CMUT array

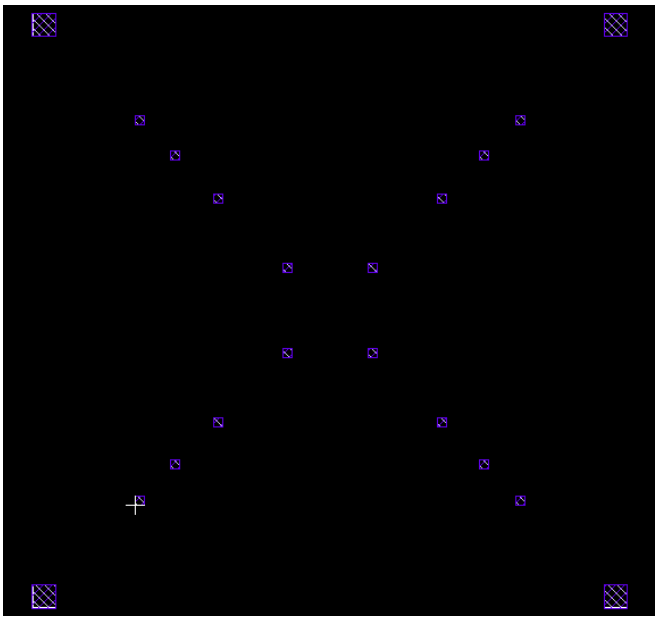

Figure 4.20 Pad orientation of 4x4 element CMUT array

Those pads in the Figure 4.19 and 4.20 have also definitions on them, but not visible in the images.

4.6 I/O and Supply Structures

In all three ASICs designed here, the non-inverting input buffer type input cells are employed. As the output cells employed in the test ASIC, the tri-state output buffer is employed. Each cell has 2 ground connections, a 3.3V supply connection and a 1.8V supply connection. By the help of those connection, the supply and ground rails of the I/O structure is realized and connected. The corner cells used in this design includes ESD structures, as a result, there is no need for the clamp cells.

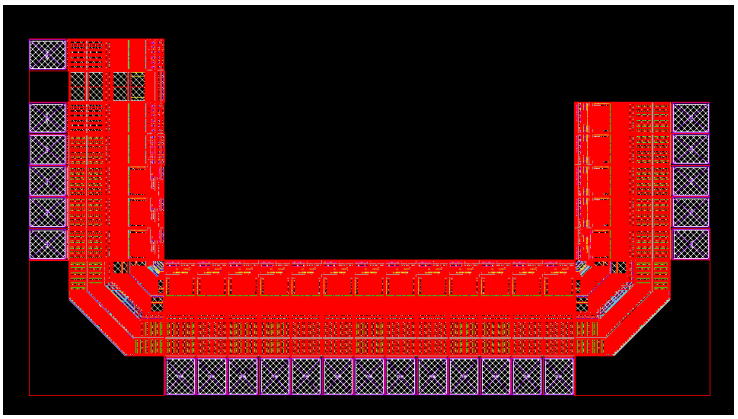

Figure 4.21 Typical I/O rail

Figure 4.21 illustrates a typical I/O rail that includes all the cells stated above. Those are the pad limited type cells and all of them have the same size of 257µm x 60µm. The only different one in the figure is the filler cell. Typically, there are 3 bars in the supply rail of a chip, the GND R rail, GND O rail and VDD rail. All the cells depicted in the figure shares the rails with proper connections to neighboring cells. The size and number of the filler cells used can be different, it depends on the design and the wire bonding capability. For the chips with large areas like the case in the designs herein, the pad limited cells are suggested. The other type, core limited cells are suggested for small sized designs.

For the high voltage supply connections, instead of I/O library of the technology, the HV ESD library is preferred, since each supply connection requires its own

ESD protection and I/O library does not have a ESD structure to protect the chip from electrostatic discharges from the 45 V supply. Non-isolated latch-up robust HV supply/IO ESD protection cell is employed to connect chip to the HV supply.

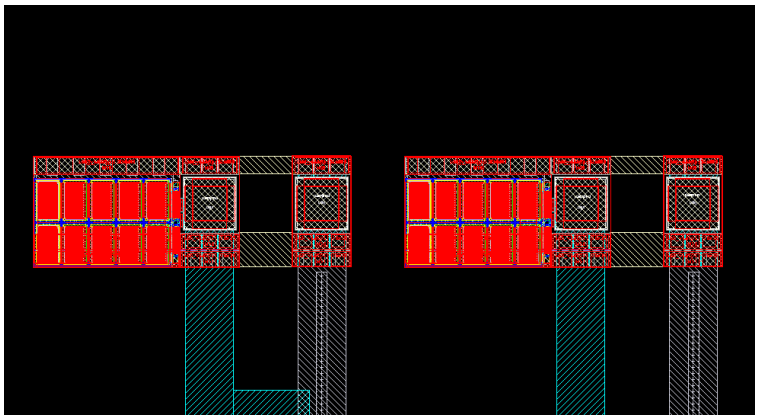

Figure 4.22 HV ESD supply layout

The circuitry on left side of the connection depicted in Figure 4.22 is the ESD protection circuitry.

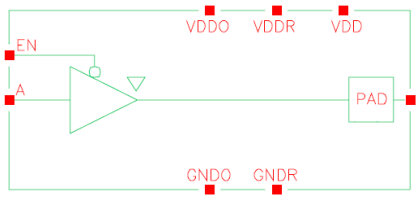

Figure 4.23 Output cell diagram

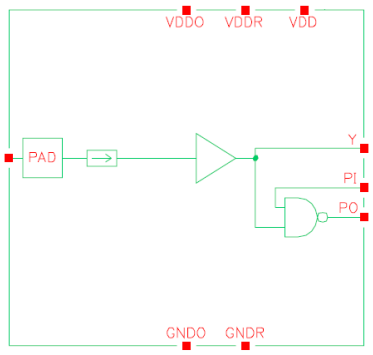

Figure 4.24 Input Cell diagram

Figures 4.23 and 4.24 shows the block diagram of the employed input and output cells. It is not necessary to PI and PO pins of the input cell, however, the EN pin of the output cell should be connected to ground for proper functioning. The logic tables of the cells used herein can be found on the process I/O manual.

4.7 Test Cell of the Beamformer

The difference between the other ASICs and the test cell is that the test cell is not designed to be integrated with a CMUT element or array, rather it is designed to test all subcircuits employed in this design. Before the fabrication phase of the designed ASICs, the designer should make sure that the design meets the requirements and specifications. In order to prove that the ASIC is properly functioning, a test cell should be designed. In this cell, there is one beamforming cell driving a capacitor-on-chip which is a model of CMUT element. The capacitor designed in this cell is 5.17 pF sandwich model capacitor having 5 metal layer and capacitor layer between them. There are also output pins which provide outputs of each subcircuit to the outer world. Although the cell can be designed in very smaller sizes, it is designed larger since the lowest possible cost at an MPW (multi project wafer) run is calculated for 10 mm^2 area and is fixed even though the designer have a smaller design.

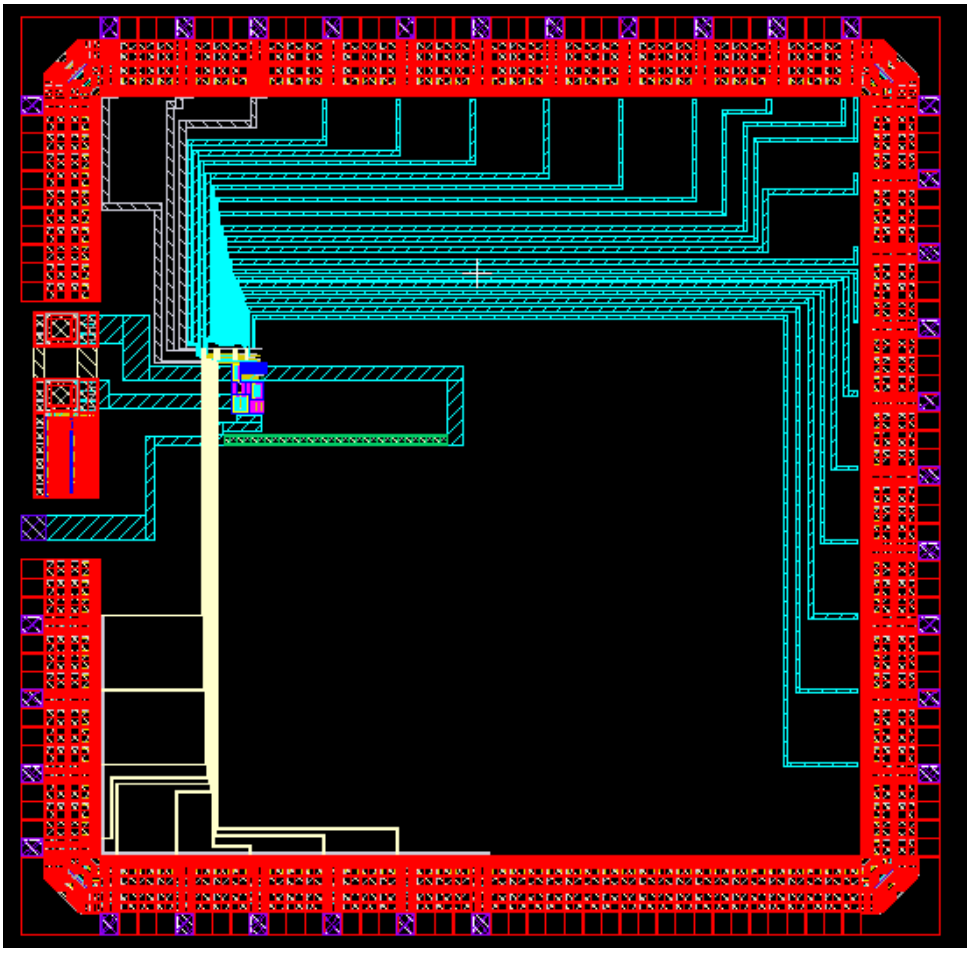

Figure 4.25 Layout of the test cell

As shown in the Figure 4.25, the capacitor is placed under the HV pulser. The capacitor is created by t-cell generator automatically. The data and HV supply connections are done using metal3 layer and the LV supply is connected using metal1. However, the outputs are sent over metal4 lines to the I/O pins. There are more filler cells in between the I/O pinto ease the wire bonding phase in this design. The bottom left side is the output ports and the upper left most side is for supply connections. The other connections on the I/O ring is for data connections. The bond pad at he left side of the chip is to measure the voltage on the capacitor. The area of the test cell is 6.68 mm^2 (2717 µm x 2460 µm) and it has a density of 14.20%. The DRC, Extraction and LVS procedures are done without any problem.

4.8 Beamformer ASIC for 4x1 Element Annular CMUT Array

In the ASIC designed for the 4 element annular CMUT array, there are 4 beamforming circuits instead of 16. The bonding scheme is also different for this case. The 250µm x 250µm bonding pad is used in this configuration. The connection between the beamformer ASIC and the CMUT array will be done by wire bonding this time.

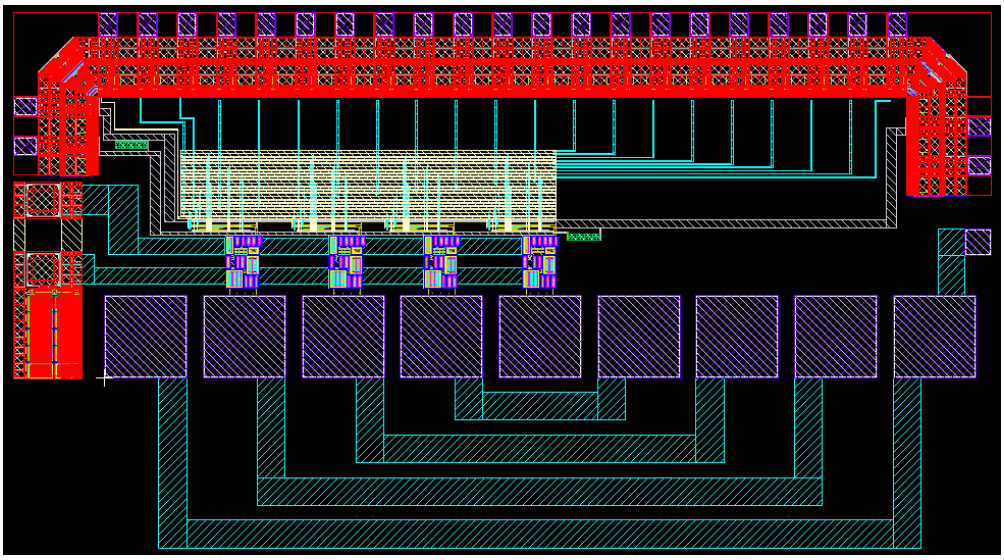

Figure 4.26 Layout of the ASIC for 4 element array

In this configuration, metal4 lines are drawn on top of the driver cells to be used as common data bars. The drivers get the data bits through these bars. The connection between the bars and cells and between the I/O cells and bars are done using metal3. Metal3 is used again for the delivery of the high voltage and metal1 is used for ground and low voltage delivery.

The right most pad is connected to left most pad since they are connected in the CMUT too and the other pads copies that connection until the one at the middle. The pad at the middle of row is the connection pad of the element in the middle of the annular ring, so it has only one connection. The additional 80µm x 80µm pad located on the left side of the chip is used for DC bias connection with outer

world. The distance between the metal3 lines connecting the pads to each other is wide enough to obey the warning of the process stated in the last section.

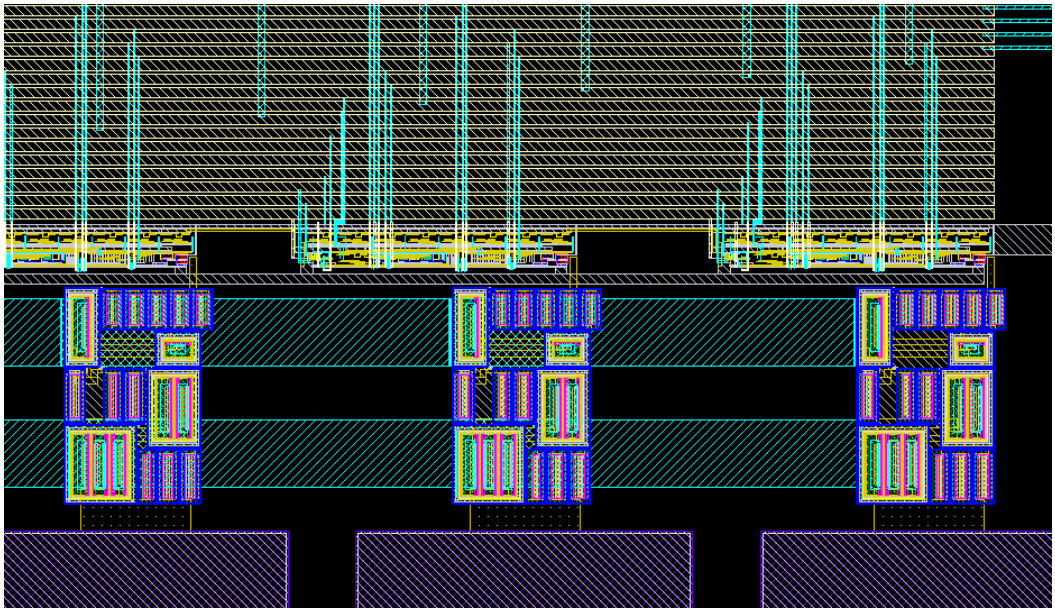

Figure 4.27 Connections between LV, HV and pads

Figure 4.27 shows the connections in the circuit which are similar to the previous configuration. The area of the ASIC is 4.07 mm^2 (2973 µm x 1371.22 µm) and it has a density of 28.84%. The DRC, Extraction and LVS procedures are done without any problem.

4.9 Beamformer ASIC for 1x4 Annular CMUT Array

In the beamformer for the disk CMUT array which is divided into 4 sectors instead of four sequential rings, the number of the driver circuits are the same and the number of the I/O structures are also the same. The major difference exists due to the alignment of the transducer elements. In this case, only one 250µm x 250µm bonding pad for each transducer element is used. In addition to pads for transducer elements, a 80µm x 80µm bonding pad at the middle of the signal pads is located for the sake of DC bias application. This pad carries the DC bias applied to the fixed plate of the CMUT element.

The signal interconnections are realized by metal4 lines and the supplies are connected to the driver part by metal 2 lines. For the high voltage connection, wide metal 3 lines are preferred to avoid from self-heating. The area of the ASIC is 1.94 mm^2 (1530 µm x 1277 µm) and it has a density of 20.52%. The DRC, Extraction and LVS procedures are done without any problem.

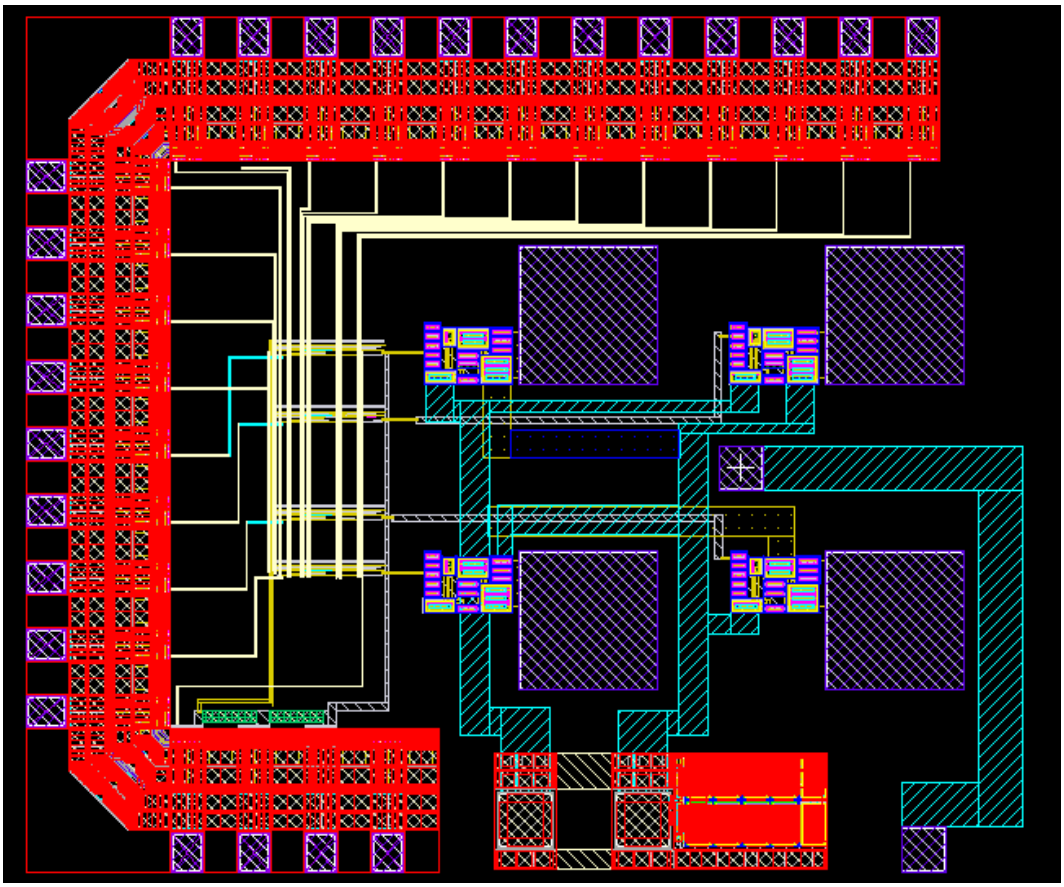

Figure 4.28 Layout of the ASIC for 1x4 array

4.10 Beamformer ASIC for 4x4 Annular CMUT Array

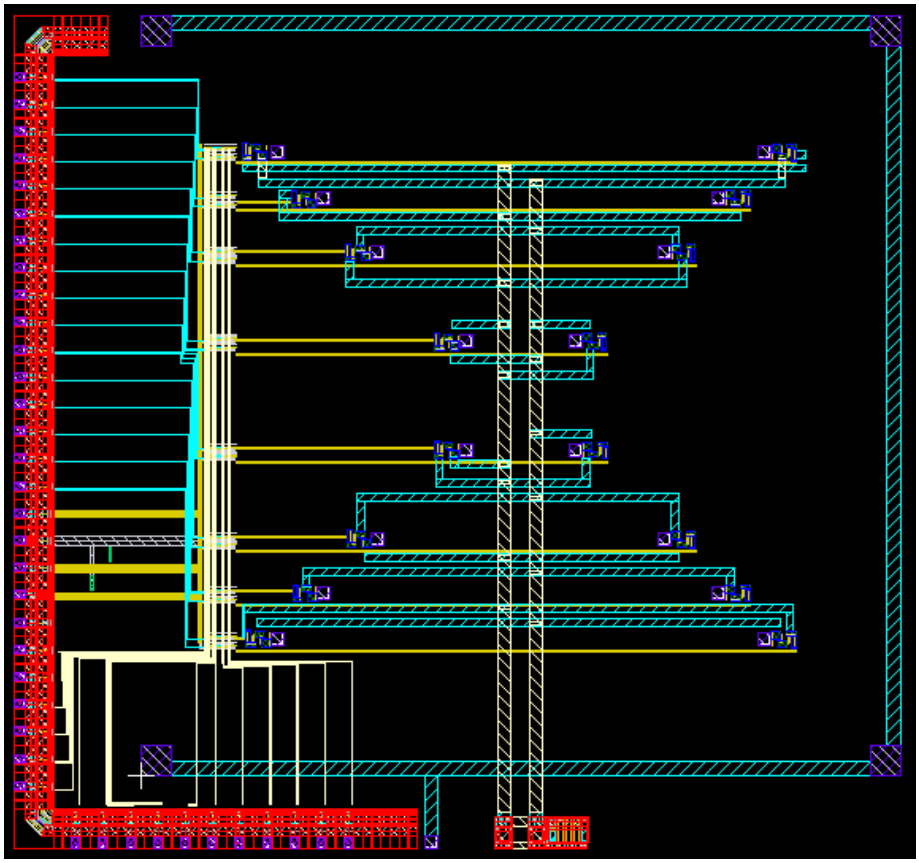

Figure 4.29 Layout of the ASIC for 4x4 array

The beamformer ASIC designed for the 4x4 annular CMUT array has its own pad structure, driver configuration and HV pulser placement as depicted in Figure 4.28. The outer metal ring in the figure is the connection of DC biasing pads of the CMUT array. The pads located in a cross shape is the flip-chip bonding pads of the CMUT. The HV pulser circuits are placed near the bonding pads (as close as possible). The high voltage is transferred to the HV pulsers by using metal3 and metal4 interconnections. The metal4 line at the middle of the circuit carries the high voltage and the connection between this line and the HV pulsers are realized using metal3 lines. The inputs from FPGA to the demultiplexers of each row is fed from the top left side of the circuit via metal3 lines. The common data inputs are fed to the drivers from the bottom and bottom left I/O blocks via metal4 lines. The LV and MV connections are done through standard I/O pads. There is 2 connection for 1.8 volts supply and 1 connection for 3.3 volts supply,

since the current drawn from the 1.8 V supply is more than 3.3 V supply. Besides, it protects the circuit to exceed the maximum current limit defined in I/O manual of the process for an I/O pad. The ground connection is done via metal1 and supply connections are done via metal2. The signals generated at the driver part of the ASIC is transmitted to the HV part via metal2. The distance between lines having voltage difference more than 25 volts are wider than 45 µm or more in order to obey the warnings in the design manual.

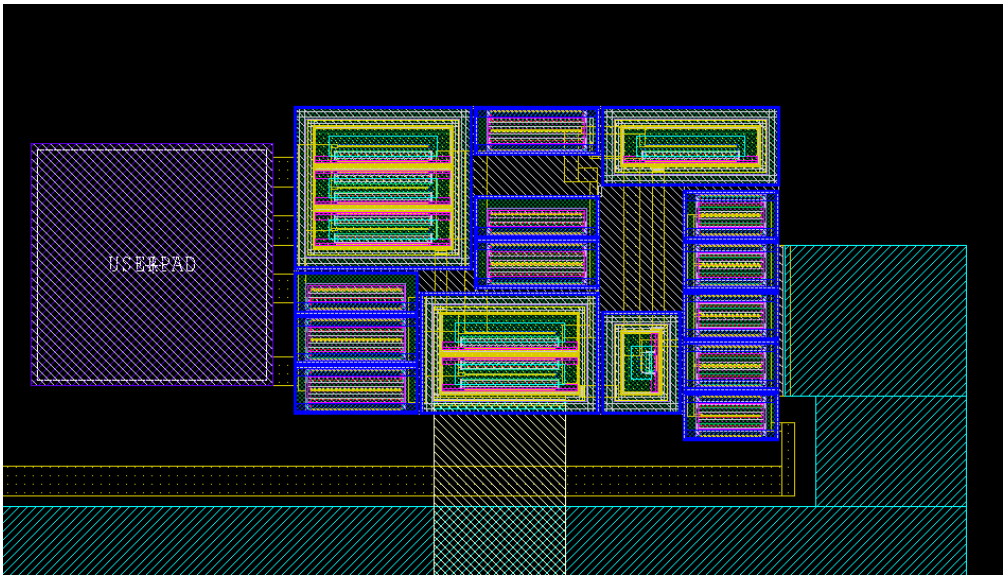

Figure 4.30 Connection of pad and HV pulser

As depicted in Figure 4.29, the connection between the pad and the high voltage pulser is done via metal2. The HV pulser loading line and the line carrying the driver signal is separated from each other by a distance more than 45 µm. The area of the ASIC is 30.73 mm^2 (5593.93 µm x 5494 µm) and it has a density of 16.21%. The DRC, Extraction and LVS procedures are done without any problem.

4.11 Simulation Results of the Layouts

As all of the layout designs include some devices that are not modeled in the schematic design stage such as parasitic capacitances and parasitic diodes, the

designer needs to simulate the behavior of the layouts. The parasitics may lead to minor changes or critical changes according to some factors such as their values, operation frequency of the system, sizes of transistors etc. There are also some precautions that are taken by the designer to get rid of problems which are not foreseen in the simulation steps. For instance, addins capacitors between the supply and ground to prevent jitters and supply variations, drawing guard rings to isolate the transistor etc. are some examples. In this step, the simulation results of the drawn circuits are presented and, although the changes are really small, they are stated. Due to the fact that combinational logic circuits do not have performance degradations, their simulation result is presented in the top level simulation chart.

The fisrt circuit that is simulated is the oneshot circuit. In the layout simulation case, the digital control bits are set as “110”, which is intended to result in a 5ns pulse. In the simulation result, the generated pulse is 5.1 ns, which means there is 2% performance degradation. According to simulations, the rise time of the signal is 0.8 ns which is same as the schematic-based result.

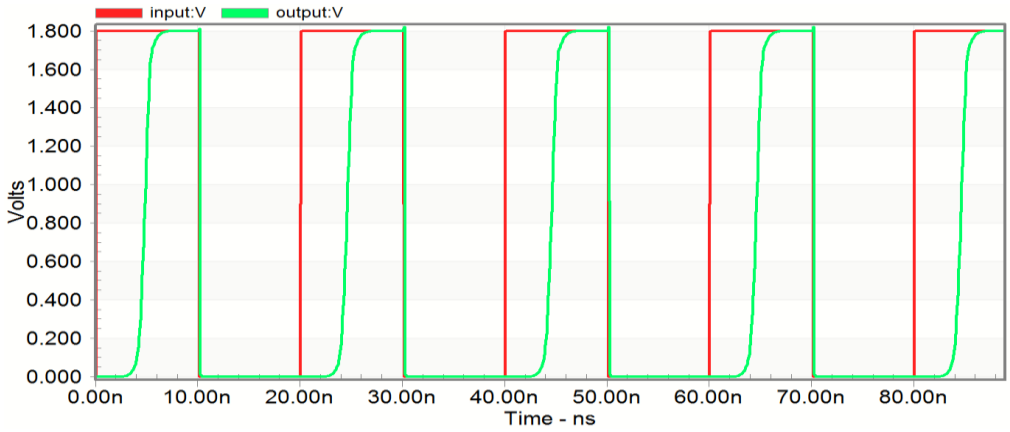

Figure 4.31 Simulation result of oneshot layout

For the digitally controlled oscillator circuit, a high frequency generating set of control bits are chosen to test under critical conditions. The digital control bits are set as “00011” which results in 287,3 MHz output frequency. The period of the frequency is 3,48 ns and it is the result of schematic-based simulation. The layout-extracted based simulation result shows that the period is almost 3,46 ns which results in 289 MHz output frequency. The change of the performance is

less 1% and does not affect the performance of the overall system, because the generated frequency will be divided by the FDC circuit.

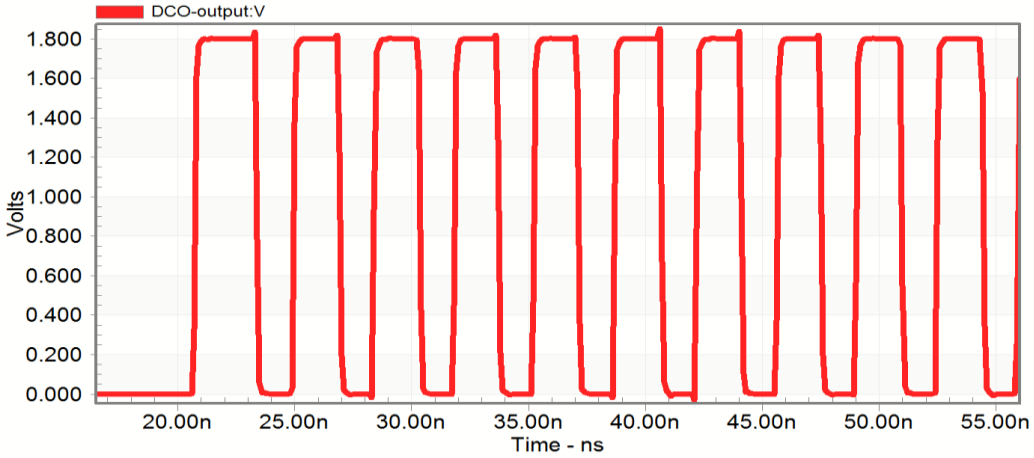

Figure 4.32 Simulation result of the DCO layout

The rise time of the signals in both the schematic-based and the layout-extracted is 0,01 ns which are the same. There is no difference on the rise time of the signals.

For the medium voltage pulser which levels up the 1.8 V signal to 3.3 V signal, the delay introduced by the circuit is not changed. The rise time of the output signal is changed from 0.5 ns to 0. 52 ns which means 4% performance degradation. However, this amount of performance degradation is not problematic due to the fact that the system will work at most at 10 MHz frequency. For the power consumption and fall time parameters, the system performs at the same level.

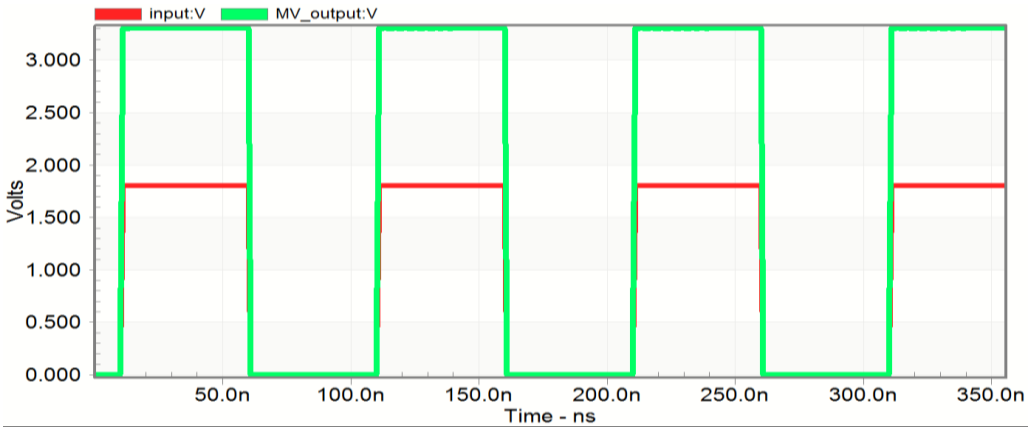

Figure 4.33 Simulation result of the MV Pulser

The high voltage pulser which which will drive the capacitive load of the transducer is also simulated with the layout-extracted netlist. The delay of the circuit noted as 4 ns in the schematic simulation results. For the layout simulation, it is 4,1 ns, which results in a 2,5% performance degradation. The rise time is changed from 4,2 ns to 4,3 which means 2,5% performance degradation. Lastly, the fall time is changed from 5,3 ns to 5,4 which is 2% degradation.

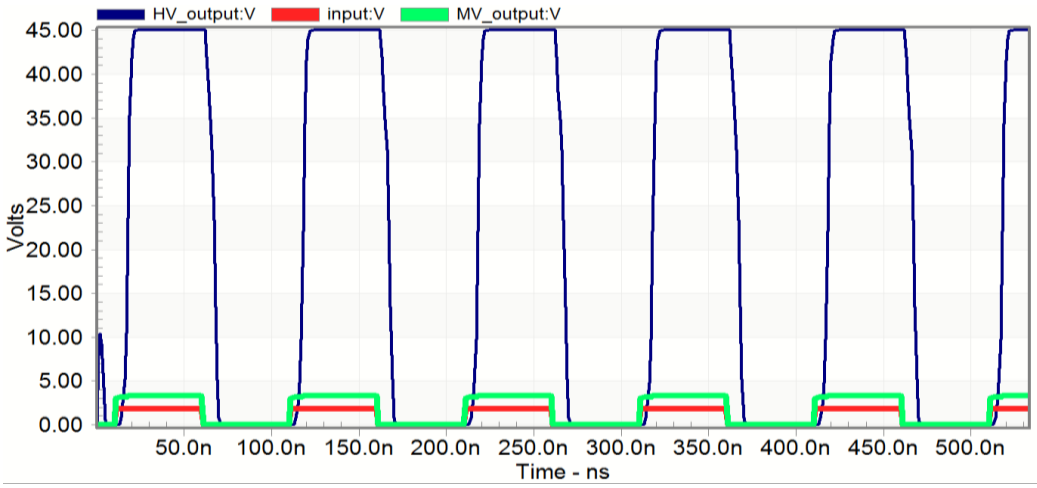

Figure 4.34 Simulation result of HV Pulser

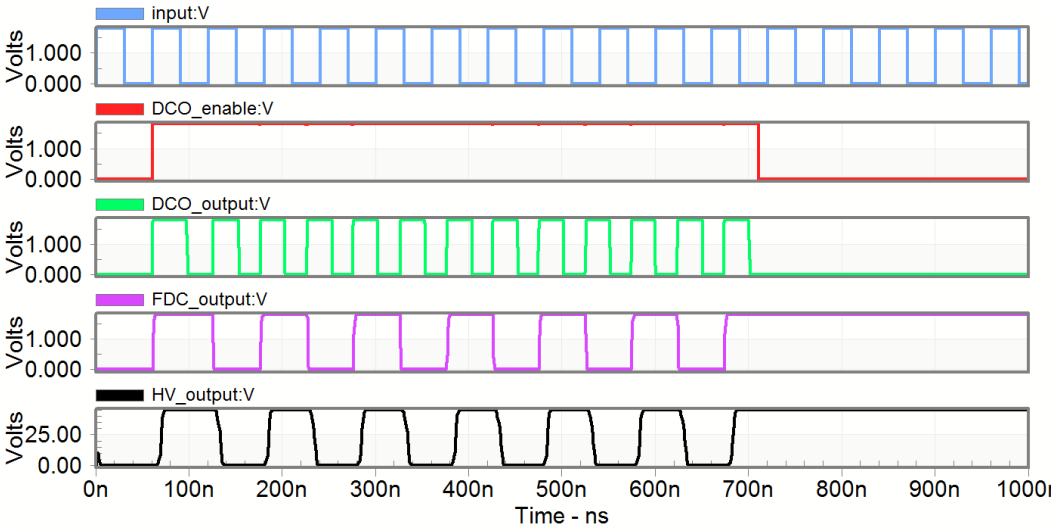

Figure 4.35 Top level simulation results

The top first line in Figure 4.35 represents the high voltage puser output. The bottom two lines are the supplied signal from FPGA and the output of the DCO. The two line in between those groups represent the enable signal and input of the enable circuit.

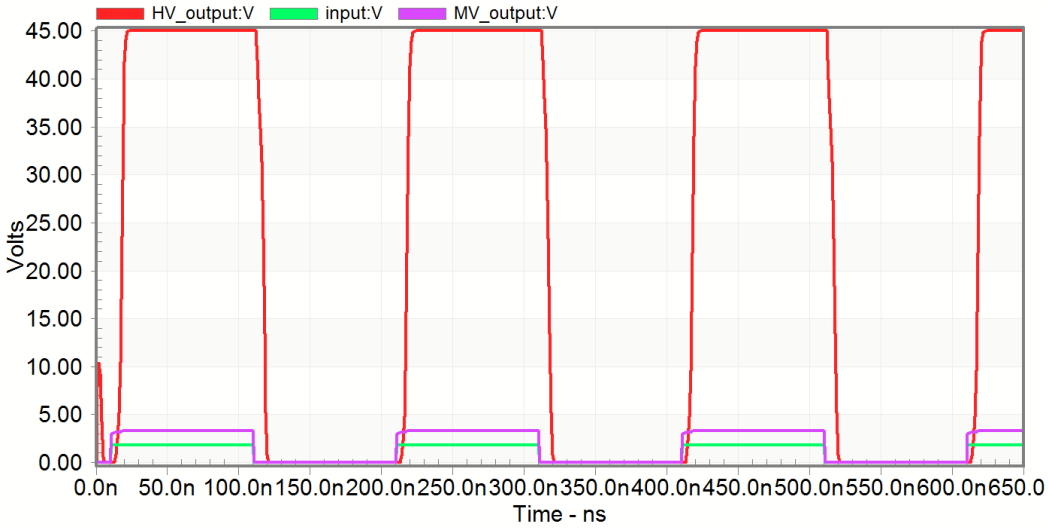

Figure 4.36 Top level simulation results

In Figure 4.36, the high voltage and medium voltage level shifter outputs are shown as higher amplitude signals. The outputs of the FDC, MUX8, MUX4 and prebuffer is located on top of each other in the simulation results, which means that there is no performance degradation on combinatial logic circuits.

4.12 Summary of the Chapter

Table 4.1 Cell sizes and densities

CELL	SIZE	DENSITY	AREA
DEMUX	18,74 µm x9,96 µm	97,26 %	181,55 μm^2
ONESHOT	14,22 µm x 12,78 µm	98,17 %	178,47 μm^2
DCO	68,58 µm x 12,7 µm	92,98 %	809,9 μm^2
FDC	244.08 µm x 12.7 µm	87,84 %	2723,1 μm^2
MUX_4	28.4 µm x 11.62 µm	92,79 %	306,21 μm^2
MUX_8	58.16 µm x 13.08 µm	87,28 %	664,26 μm^2
BUFFER	21.16 µm x 12.33 µm	92,84%	242,33 μm^2
MV_PULSER	11.48 µm x 10.59 µm	100 %	121,57 μm^2
LV_DRIVER	211,16 µm x 41,16 µm	60,51 %	5259,28 μm^2
HV_PULSER	110,73 µm x 161 µm	93,42 %	16654,9 μm^2
TEST_CELL	2717 µm x 2460 µm	14,20 %	6,68 mm^2
1D_CELL(4x1)	2973 µm x 1371,22 µm	28,84 %	4,07 mm^2
1D_CELL(1x4)	1530 µm x 1277 µm	20,52%	1,94 mm^2
2D_CELL	5593,930 µm x 5494 µm	16,21 %	30,73 mm^2

In this chapter, the layout design of 3 beamformer ASICs are explained with their placement and routing figures. Table 4.1 shows the size and density values of the cells. Each beamformer is composed of low voltage blocks including medium voltage pulser, high voltage blocks, I/O and supply cells and bonding pads. The designed test cell includes a capacitor modeling the CMUT element in addition to above stated parts. In the ASIC designed for 4x4 annular CMUT array (2D), there are 16 beamforming rows and in the ASIC designed for 4 element annular CMUT array (1D), there are 4 beamforming rows. The test cell employs only one beamforming row. Since there are many types of I/O structures, only the employed ones are described in this chapter. The only predefined blocks included

in the designs are the I/O blocks including HV supply structures. During the cell placement and routing phase, the warnings are cared seriously to prevent any malfunctioning after the fabrication, since there are some points to be carefully analyzed in this process which are not contained in the rule set of the process. For instance, the manual of the process declares that the designer shall not draw metal lines too closely if the potential difference between the lines are higher than 25 volts. Such warnings may not be checked during DRC phase, but should be taken into account during design. In the next chapter, the MATLAB program that computes the relevant data and sends them to FPGA is introduced.

CHAPTER 5

5. FPGA SETTING ALGORITHM FOR ANNULAR CMUT ARRAYS

The MATLAB program prepared to work with the FPGA to phase the CMUT array elements is presented in this chapter. The program is prepared for the beamformer ASIC to be integrated with the 4x4 annular CMUT array. However, the program can be modified to be compatible with other beamformer types by a few modifications. It is a complete program that computes the relevant latency values and required data bits. The theory of the program will be presented in Section 5.1. Section 5.2 presents the program step by step. Section 5.3 is the summary of this chapter.

5.1 Theory of the Program

As the program knows the location of the annular CMUT array (assumed to be at the origin of Cartesian coordinates), upon the entrance of the focal point coordinates, the program can compute the distances between each element and the focal point by using the wave speed defined. The first stage of the program is designed to calculate the distances, travel times of waves and priorities. Following this stage, next stage is the FPGA setting stage. The user selects which type signal to be used firstly and set the relevant data to that specific circuit. Upon the choice of the user, a “DCO Settings” or “Oneshot Settings” button becomes visible at the bottom left side of the FPGA setting page. As the user goes to the page of relevant circuit and sets it, user goes back to the FPGA

settings program and sets the FPGA. That is the theory behind the program and really easy to follow.

5.2 Program Flow

Following figures represent the user interface windows of the program and each window is illustrated in the relevant sequence. The first figure is the first window and the second, third and fourth is the figure of second window. The windows of oneshot settings and DCO settings follows others.

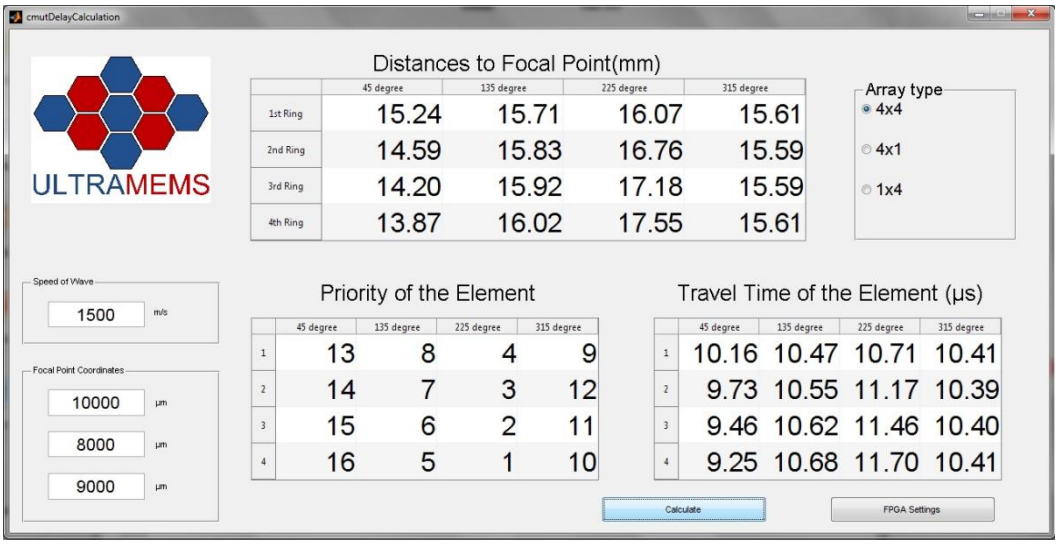

Figure 5.1 CMUT phasing program

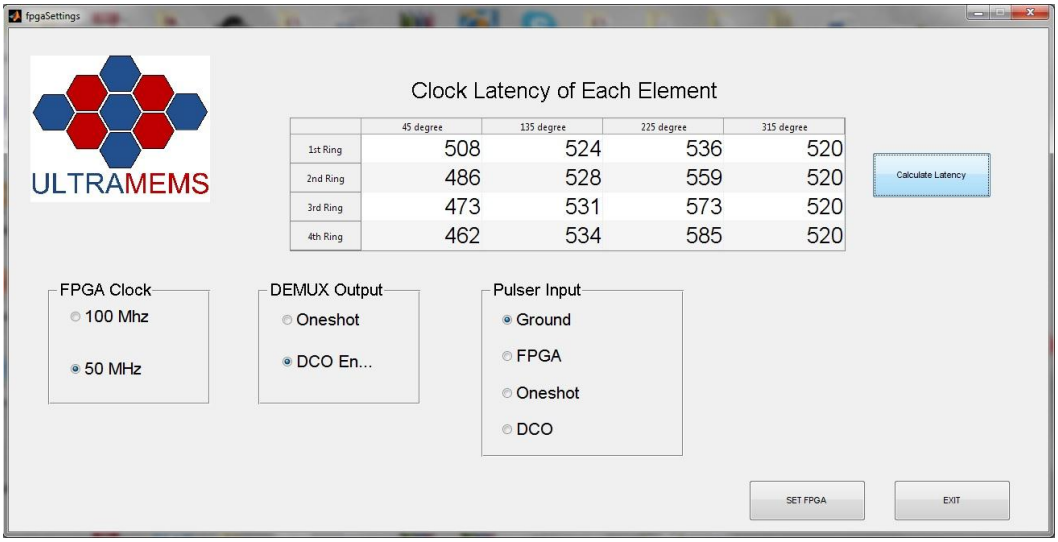

Figure 5.2 FPGA settings window without any choice

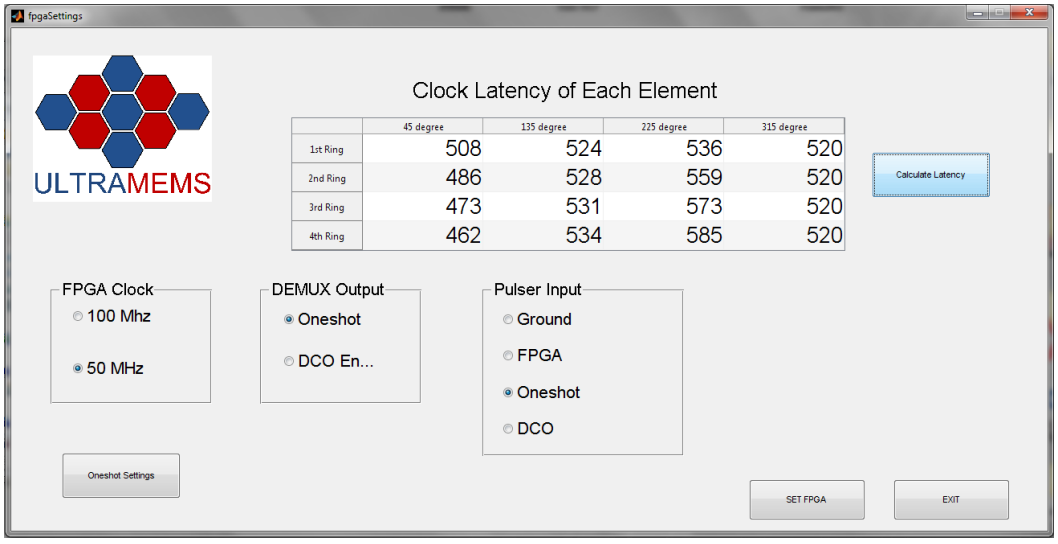

Figure 5.3 FPGA settings window with oneshot choice

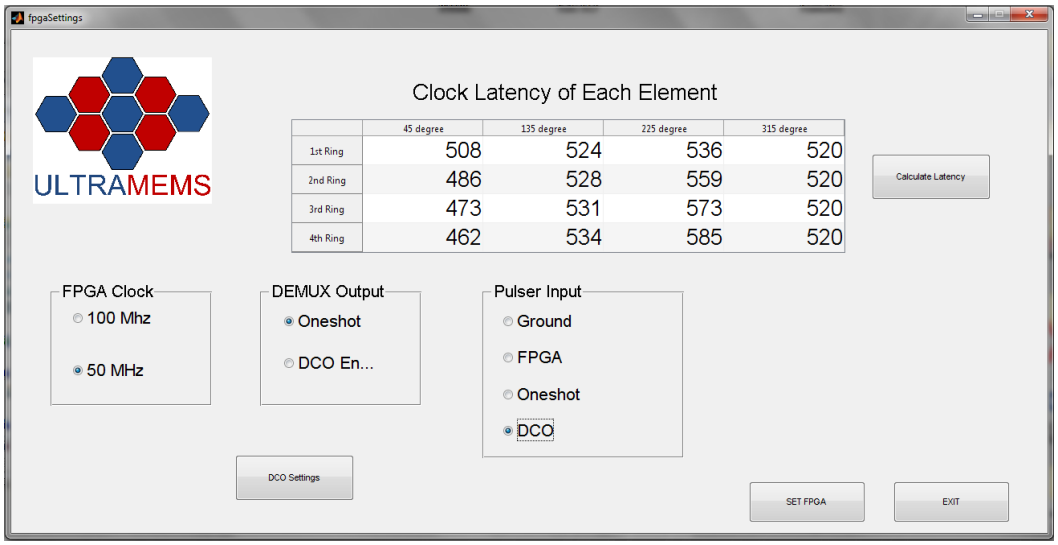

Figure 5.4 FPGA settings window with DCO choice

As can be seen from the above figures, the buttons become visible upon the selection on the “preamplifier input” section which is actually the settings of the select bits of 4-input multiplexer. The clock latency calculated here depends on the FPGA clock speed. As a result, the left most bar is to select the clock speed of the FPGA.

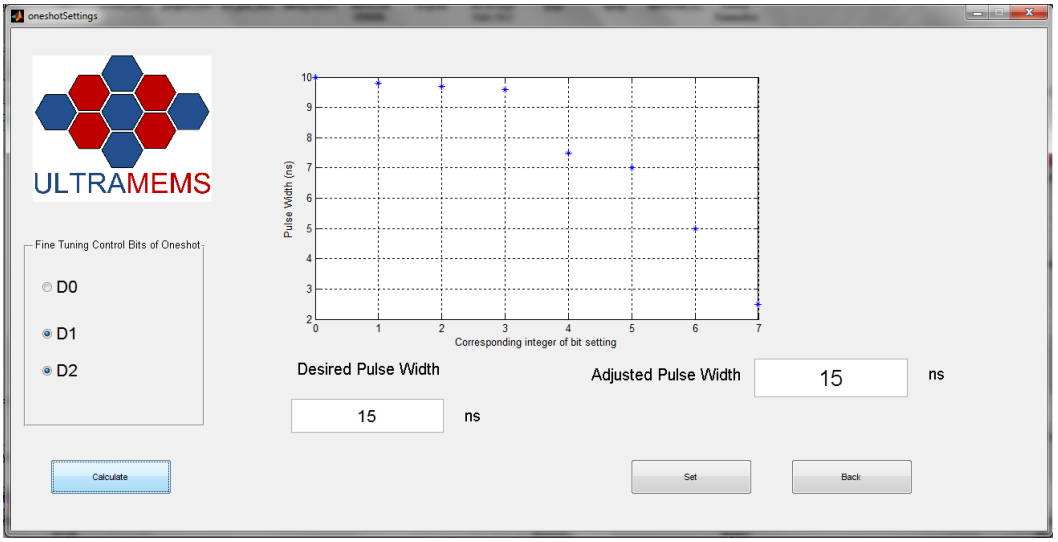

Figure 5.5 Oneshot settings window

As can be seen from Figure 5.5 and 5.6, the user enters the desired signal length or frequency, then the program adjust the signal that are the closest possible ones. The graphs in figures represent the possible outputs of the circuit.

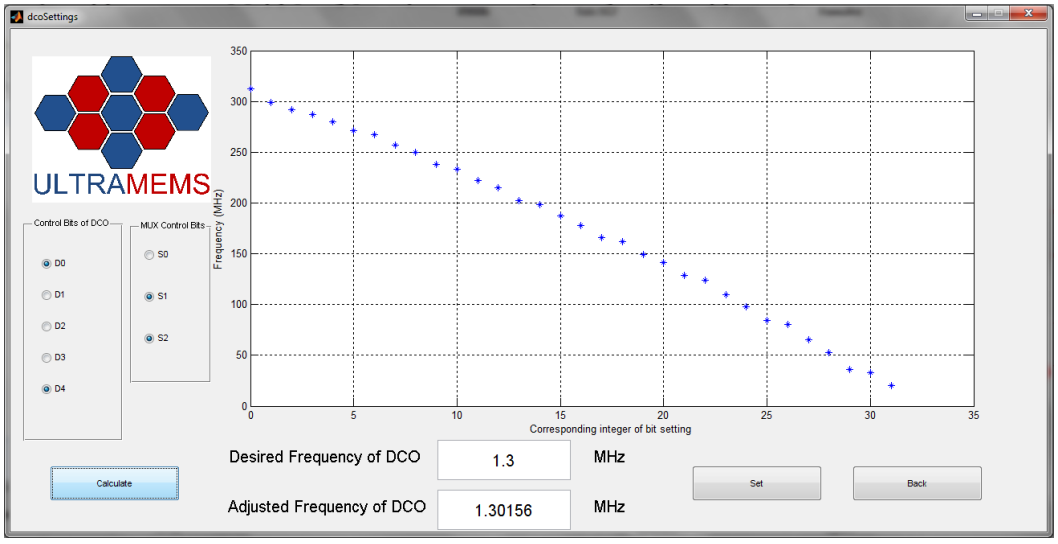

Figure 5.6 DCO settings window

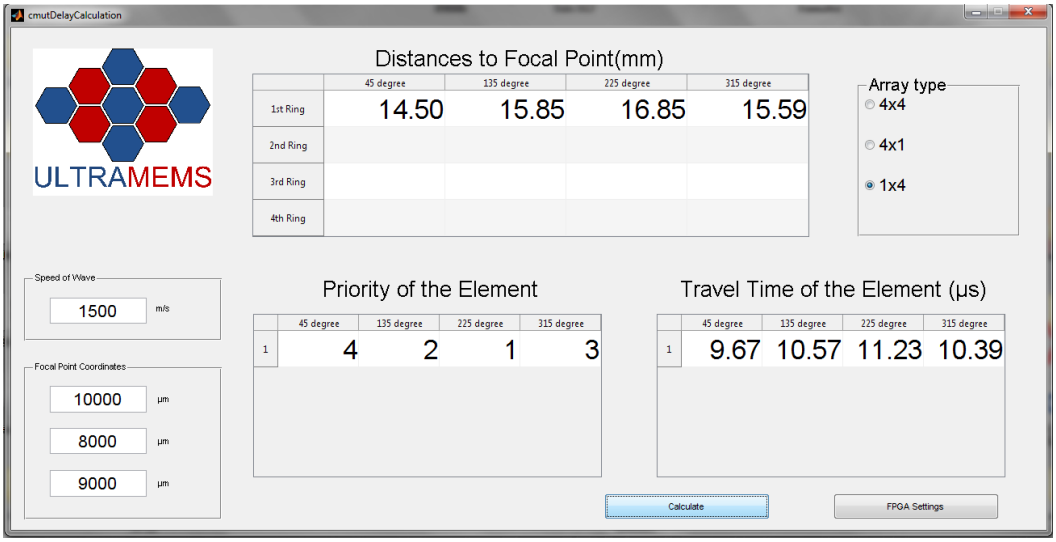

Figure 5.7 How to select CMUT array type

As can be seen through Figure 5.7, the array type can be selected from the table at the top right side of the window. First option is the 2D (4x4) array and the second one is the 1D 4X1 element option and the third one is the 1D 1x4 element option.

At the end of the calculation process, if the user pushes the “SET FPGA” button, the program send the calculated data to the FPGA. In order to test the functioning of the program and FPGA algorithm, a Digilent Basys2 evaluation board is used. Due to the fact that this evaluation board has no on board connection port to the computer, a serial communication (RS-232) platform is prepared. The devices used to prepare the platform is listed in Table 5.1.

Table 5.1 Device List for RS-232

Quantity	Device
1	Digilent Basys 2 FPGA
1	S-Link USB-to-Serial Convertor
1	TI MAX232
4	1µF Capacitor
1	5V DC Adapter

The configuration using the above listed items is illustrated in Figure 5.8.

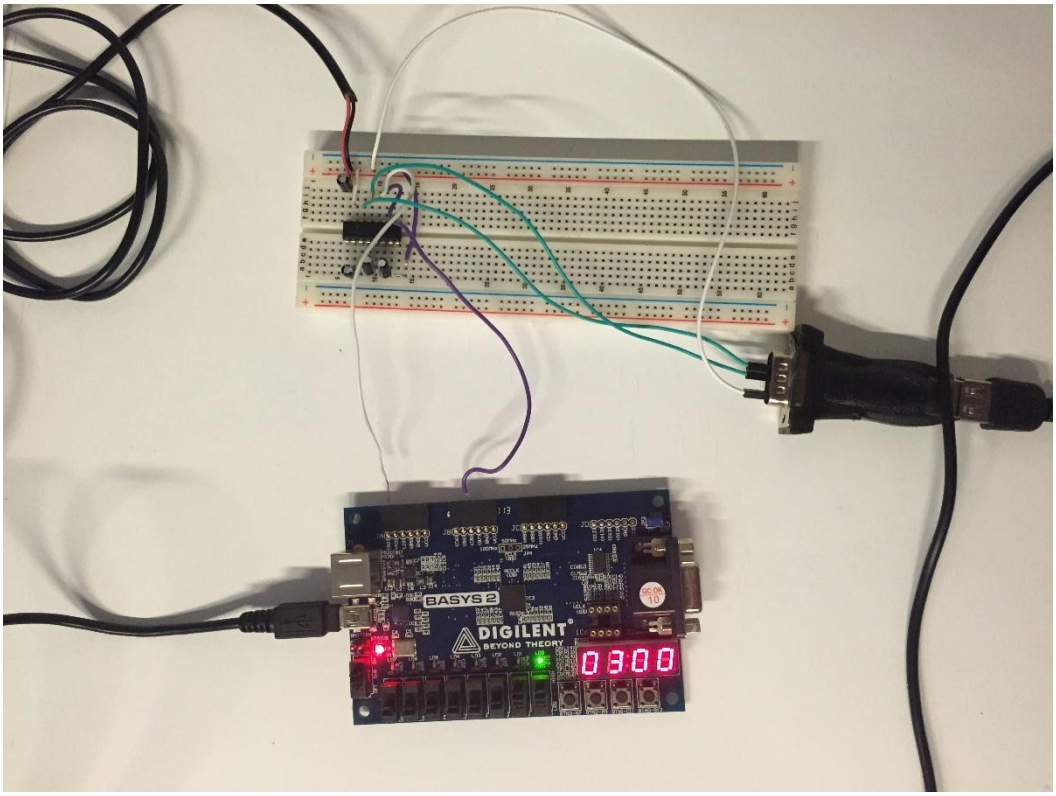

Figure 5.8 FPGA and RS-232 Configuration

The program computes the latency values and sends them to FPGA to trigger signals according to that list. In Figure 5.2, there are 16 clock cycles between 1st Ring 45 degree and 135 degree, and 20 clock cycles between 1st Ring 45 degree and 2nd Ring 135 degree. The two different cases are measured with oscilloscope and the data obtained by oscilloscope is plotted in MATLAB.

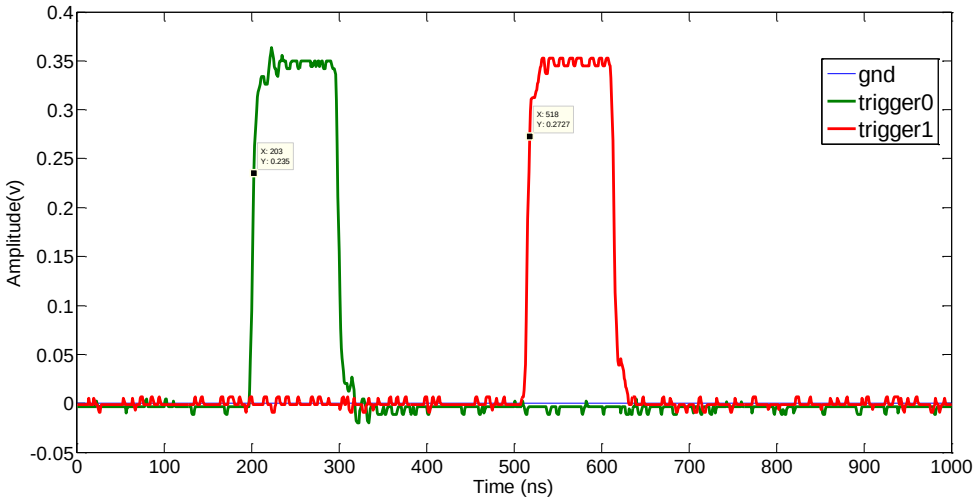

Figure 5.9 16 clock cycle latency

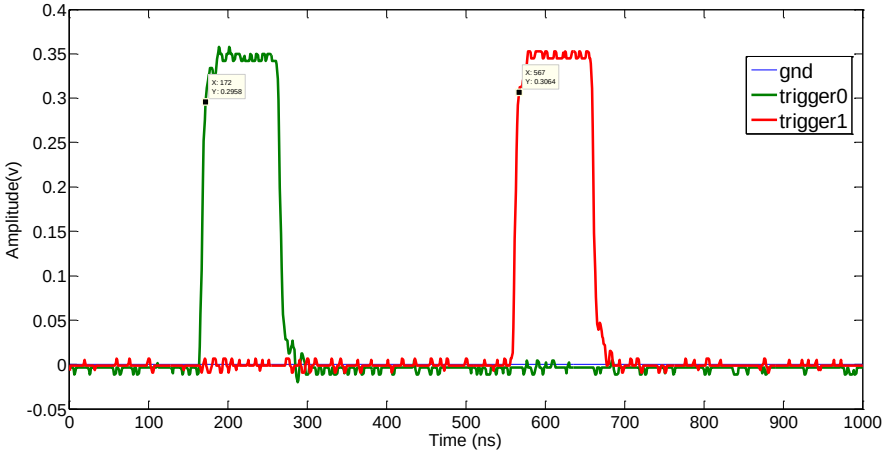

Figure 5.10 20 clock cycle latency

As can be seen from the figures above, FPGA sets the trigger times correctly with a deviation of 3-4 ns maximum. The FPGA algorithm is also presented in the Appendix. The circuit configuration is supplied in [62] and presented in Figure 5.10.

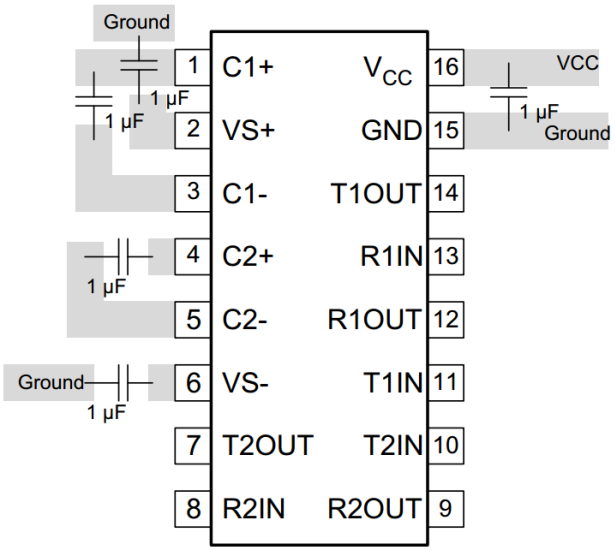

Figure 5.11 MAX-232 circuit configuration [62]

5.3 Summary of the Chapter

The theory behind the algorithm and the program flow is described in this section.

The program basically computes the traveling time of the waves by calculating the distance to focal point and dividing them into defined speed. Then, by

dividing the travelling time calculated herein into the period of the clock of FPGA, the program finds how many clock cycles it needs to phase the CMUT element. The figures illustrates the user interfaces of the designed program. Since the operational principles are same for the two defined type CMUTs, the program can be used for the 4 element annular CMUT array by modifying the number of elements in the program and their locations easily. The locations of the transducer elements are predefined in the program. Upon the application or configuration, the location of the transducer elements may be optional and set by the user. How to use the program is described in this chapter in detail. There are some fundamental information in previous chapters too. The sources codes of the program is presented in the Appendix.

CHAPTER 6

6. CONCLUSIONS AND FUTURE WORKS

Throughout this thesis work, a flexible, high drive capability and digitally controllable beamformers for capacitive micromachined ultrasonic transducers to be used in medical applications have been designed in high voltage 0.18μm CMOS process and introduced. The underlying reason of designing the topology in this thesis is to excite capacitive micromachined ultrasonic transducers (CMUTs) with high voltage (45V) signal to generate 3D beamforms to be used in therapeutic ultrasound applications. In addition to therapeutic applications, designed beamformer ASIC can be employed as a transmit beamformer of an ASIC used in ultrasound imaging applications. An addition of receiver topology such as a transimpedance amplifier converts this ASIC into a transceiver topology to be used especially in 3D ultrasound imaging applications. The designed ASICs are intended to be integrated with the diamond based annular capacitive micromachined ultrasonic transducer (CMUT) arrays presented in [60] in ULTRAMEMS research group. Besides, a brief introduction to therapeutic ultrasound applications is provided in the first chapters of the thesis. The resulting action of the therapeutic application is called as coagulation necrosis or ablation of cells in some different researches, however, the work done can be called as the death of cells due to high energy absorption. The final work can be modified in terms of accommodation of different cells in the ASIC layout to be integrated with various CMUT arrays.

The results and achievements of the thesis work is as follows:

1. In order to give an understanding to the reader, a literature research has been made and presented in the thesis. A detailed analysis on the required circuits and their designs has been made so as to design a flexible and high capability beamformer ASIC to meet the requirements of an ultrasound transmit driver topology. In addition to the transmitter part analysis, a literature review on receiver electronics is included in the thesis.
2. A new phasing algorithm is proposed with the control of a high capability processing unit. The conventional phasing configurations employs logic blocks and switches whereas the phasing is realized through the algorithms on FPGA supplied by a MATLAB program in this thesis. This new algorithm reduces the number of circuits required in the beamformer. For instance, in [2], [32] and [33], there are shift registers and comparators in addition to the processing unit like FPGA. However, this design employs only FPGA and saves both area and power.
3. Smart designs with their advantages over conventional circuits and new blocks are proposed to generate the required single pulses with variable widths or pulse trains with variable frequencies in a wider range to be suitable with as many CMUT arrays as possible. Three different voltage levels are used in the design. 1.8 volts (core voltage, LV), 3.3 volts (MV) and 45 volts (HV) are used.
4. Three different beamformer ASICs and a test cell each including the demultiplexer, oneshot circuit, digitally controlled oscillator, frequency down converter, multiplexers, buffers and MV and HV pulser are designed using HV 0.18µm CMOS process. In addition to those circuits, the test cell has a capacitor modeling the CMUT element. Due to the challenges of the process, a medium voltage pulser is designed to drive the high voltage circuitry. Standard LPMOS module of the process is used to design the LV and MV blocks whereas the PHVE and NHVE modules are used to design the HV pulser. In addition to the ESD protection cells of low voltages, a high voltage ESD protection cell is also employed which is missing in many of the similar designs. The only high voltage

values of the process are 35 and 45 volts. As a result, only 45 V is used as the high voltage of the circuit.

5. As a part of the design, a temperature analysis has been made on the interconnects by the help of the information and calculation in the [61]. The width of the metal wires are carefully designed especially for the high voltage blocks. In order to prevent wire self-heating problems, the considerations in [61] are taken into account. Electromigration considerations are also taken into account to design a reliable ASIC.
6. The designed circuits presents better performances compared to the ones in [4]. The total employed silicon area and the power consumption is decreased whereas the circuit drives higher capacitive loads and provides higher resolutions in signal generation ranges. The generator circuits provide more linear responses and the timing of the transducer elements are realized directly from FPGA side. Additional high voltage ESD protection cells are added which lacks in the former designs.

A similar work has been done by Mohammad Maadi in [2]. The differences and improvements are as follows:

1. The phasing algorithm is totally different than the one in [2]. This design utilizes only the FPGA to trigger the transducer elements according to the data from MATLAB program.
2. The single pulse generator circuit is a new design which is different than any other oneshot circuit in the literature. Although other designs basically rely on the charging/discharging an additional capacitor, the circuit in this design changes the discharge current only and does not include any additional capacitor than the gate capacitance of the following transistors.
3. The pulse train generator circuit (DCO) is similar but the performance is totally linear in this design. The problems of nonlinearities of [2] does not occur in this design and the power consumption is lower. The design is based on the proposal of [44].
4. The system is designed in 0.18 μm HV CMOS process whereas the work in [2] is designed in 0.35 μm HV CMOS. As a result of this difference,

the core voltages of the systems are different. The core voltage of 0.18 μm process is 1.8 Volts whereas the core voltage of 0.35 μm is 3.3 Volts.

5. As the core voltage of the process used in this design is lower, an additional buffer and level shifter circuit is required to supply a signal to high voltage pulser to be sensed. In order to overcome this problem, a 1.8 V to 3.3V level shifter is designed.
6. A high voltage level shifter with a totally different topology is designed to drive loads up to 10pF while the design in [2] can drive a load of 2.5pF at most. While driving higher loads, this design consumes less energy than the one in [2].
7. The layout designs in this thesis employs additional capacitor to get rid of jitters and supply variations. Besides, HV ESD protection blocks are included in this design. However, the design in [2] does not include these two items.
8. The ASICs in this thesis is designed for annular CMUTs whereas the design in [2] is designed for rectangular CMUTs. As a result of this difference, 3 different ASICs are desgined in this thesis.
9. The power consumption in the LV side in this design is almost a quarter of the consumed in [2]. For the high voltage side, the consumption is almost same while driving much higher loads.
10. The FPGA algorithm is presented and experimental work is done in this thesis which is another difference.
11. The MATLAB program is more user friendly, which means the user is not intended to try and find the frequency or pulse width of the signals. Instead, the user enters the desired value and the program calculates closest possible value in the program in this thesis.

The future works intended to be done is as follows:

1. Due to the fact that fabrication procedure of such circuits cost too much, an application will be sent to TUBITAK for funding of the fabrication process. In case the ASIC is fabricated, it will first be integrated with the 4 element annular array.

2. The realization of wire bonding process and techniques will be learned not theoretically but practically to be prepared for the integration.
3. An additional comprehensive research will be conducted on the receiver circuits used in the imaging applications to further develop the designed ASIC.
4. A thermal analysis of high voltage blocks will be done using a thermal analysis and simulation tool which analyzes especially the high voltage cells and figures out the heating problems.
5. A research on thermal imaging systems similar to thermal MR imaging will be conducted to figure out the best fitting assistive imaging system to be used with therapeutic ultrasound.

REFERENCES

[1] E. S. Ebbini, R. Seip, P. Vanbaren, O. Haddadin, C. Simon, and Y. Y. Botros, *Therapeutic ultrasound*, John Wiley & Sons, Inc, 1999.

[2] Mohammad Maadi, "Integrated Circuit Design For Flip-Chip Bonded Capacitive Micromachined Ultrasonic Transducers", *M.Sc. Thesis*, Middle East Technical University, 2013

[3] Olympus Corporation, http://www.olympus-ims.com/en/ndt-tutorials/flaw-detection/ultrasound/

[4] M. Maadi, B. Bayram, "Custom integrated circuit design for ultrasonic therapeutic CMUT array", *Microsystems Technologies*, vol. 21, pp. 875-891, 2014

[5] P.A. Lindstrom, "Prefrontal ultrasonic irradiation: a substitute for lobotomy", *Arch Neurol Psych*, vol. 72, pp.399‑425, 1954

[6] W. Fry, J. Barnard, F. Fry, R. Krumins, J. Brennan, "Ultrasonic lesions in the mammalian central nervous system with ultrasound", *Science*, vol. 122, pp.517‑518, 1955

[7] K. Hynynen, B.A. Lulu, "Hyperthermia in cancer treatment", *Invest Radiol 1990*; vol. 25, pp.824‑834, 1990

[8] T.J. Dubinsky, C. Cuevas, M.K. Dighe, O. Kolokythas , J.H. Hwang, "High-Intensity Focused Ultrasound: Current Potential and Oncologic Applications", *American Journal of Roentgenology*, vol. 190, pp. 191-199, 2008

[9] B. T. Khuri-Yakub, and Ö. Oralkan, "Capacitive Micromachined Ultrasonic Transducers for Medical Imaging and Therapy", *Journal of Micromechanics and Microengineering*, vol. 21, no.5, 054004, 2011.

[10] M. I. Haller, and B. T. Khuri-Yakub, "A Surface Micromachined Electrostatic Ultrasonic Air Transducer," *IEEE Proce. Ultrason. Symp*., vol. 2, pp 1241–1244, Oct. 1994- Nov. 1994.

[11] S.H. Wong, M. Kupnik, K, Butts-Pauly, B.T. Khuri-Yakub, "Advantages of Capacitive Micromachined Ultrasonic Transducers (CMUTs) for High Intensity Focused Ultrasound", *IEEE Ultrasonics Symposium*, pp. 1313-1316, 2007

[12] Y. Chen, Y. Wen and P. Lai, "Loss Mechanisms in Piezoelectric Transducer and Its Response to Stress", *Proceedings of the IEEE InformationAcquisition*, pp.213-219, 2004

[13] I.O. Wygant, D.T. Yeh, X. Zhuang, S. Vaithilingam, A. Nikoozadeh, O. Oralkan, A.S. Ergun, G.G. Yaralioglu and B.T. Khuri-Yakub, "Integrated Ultrasound Imaging Systems Based on Capacitive Micromachined Ultrasonic Transducer Arrays", *IEEE Sensors 2005*, pp. 704-707, 2005

[14] Mehmet Cezar, "Development and Microfabrication of Capacitive Micromachined Ultrasound Transducers With diamond Membranes", *M.Sc. Thesis*, Middle East Technical University, 2011

[15] Y. Huang, E.O. Haeggstrom, X. Zhuang , A.S. Ergun, C.H. Cheng, B.T. Khuri-Yakub, "Capacitive micromachined ultrasonic transducers (CMUTs) with isolation posts", *IEEE Ultrasonics Symposium 2004,* vol. 3, pp. 2223 - 2226, 2004

[16] D. R. Chou, "Piezoelectric Micrimachined Ultrasound Transducer for Medical Imaging", *Ph.D. Thesis*, Duke University, 2011.

[17] I.O. Wygant, M. Kupnik, J.C. Windsor, W.M. Wright, M.S. Wochner, G.G. Yaralioglu, M.F. Hamilton, B.T. Khuri-Yakub, "50 kHz Capacitive Micromachined Ultrasonic Transducers for Generation of Highly Directional Sound with Parametric Arrays", *IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control,* vol. 56, pp. 193-203, 2009

[18] G. G. Yaralioglu, A. S. Ergun, and B. T. Khuri-Yakub, "Finite-element Analysis of Capacitive Micromachined Ultrasonic Transducers," *IEEE Trans. Ultra. Ferroelectr. Freq. Contr.*, vol. 52, no. 12, pp. 2185–98, Dec. 2005.

[19] Y. Roh, and B. T. Khuri-Yakub, "Finite Element Analysis of Underwater Micromachined Ultrasonic Transducers," *IEEE Trans.*

[20] B. Bayram, Ö. Oralkan, A. S. Ergun, E. Hæggstrom, G. G. Yaralioglu, and B. T. Khuri-Yakub, "Capacitive Micromachined Ultrasonic Transducer Design for High Power Transmission," *IEEE Trans. Ultra. Ferroelectr. Freq. Contr.*, vol. 52, no. 2, pp. 326-339, Feb. 2005.

[21] B. T. Khuri-Yakub, and Ö. Oralkan, "Capacitive Micromachined Ultrasonic Transducers for Medical Imaging and Therapy," *Journal of micromechanics and microengineering*, vol. 21, no.5, 054004, 2011.

[22] A. Bhuyan, J. W. Choe, B. C. Lee, I. O. Wygant, A. Nikoozadeh, O. Oralkan and B. T. Khuri-Yakub, "Integrated Circuits for Volumetric Ultrasound Imaging With 2-D CMUT Arrays", *IEEE Transactions on Biomedical Circuits and Systems*, vol. 7, pp. 796-804, 2013

[23] M. A. O'Reilly, K. Hynynen, "A PVDF Receiver for Ultrasound Monitoring of Transcrainal Focused Ultrasound Therapy", *IEEE Transactions on Biomedical Engineering*, Vol. 57, pp. 2286-2294, 2010.

[24] P. Behnamfar, R. Molavi, S. Mirabbasi, "Receiver Design for CMUT-Based Super-Resoltioun Ultrasound Imaging", *IEEE International Symposium on Circuits and Systems 2014*, pp. 878-881, 2014

[25] I. Kim, H. Kim, F. Griggio, R. L. Tutwiler, T. N. Jackson, S. Trolier-Mckinstry, K. Choi, "CMOS Ultrasound Transceiver Chip for High-Resolution Ultrasonic Imaging Systems", *IEEE Transactions on Biomedical Circuits and Systems*, Vol. 3, pp. 293-303, 2009.

[26] S. Jung, J. Song, O. Kwon, "Three-Side Buttable Integrated Ultrasound Chip with a 16 x 16 Reconfigurable Transceiver and Capacitive Micromachined Ultrasonic Transducer Array for 3-D Ultrasound Imaging Systems", *IEEE Transactions on Electron Devices,* Vol. 60, pp. 3562-3569, 2013.

[27] B. Kim, Y. Kim, S. Lee, K. Cho, J. Song, "Design and Test of A Fully Controllable 64x128 2-D CMUT Array Integrated with Reconfigurable Frontend ASICs for Volumetric Ultrasound Imaging", *IEEE International Ultrasonics Symposium 2012*, pp. 77-80, 2012

[28] B.T. Khuri-Yakub, O. Oralkan, A. Nikoozadeh, I.O Wygant, S. Zhuang, M. Gencel, J.W. Choe, D.N. Stephens, A. de la Rama, P. Chen, F. Lin, A. Dentinger, D. Wildes, K. Thomenius, K. Shivkumar, A. Mahajan, C.H. Seo, M. O'Donnell, U. Truong, D.J. Sahn, "Miniaturized Ultrasound Imaging Probes Enabled by CMUT Arrays with Integrated Frontend Electronic Circuits", *32nd Annual International Conference of the IEEE EMBS*, pp. 5987-5990, 2010

[29] A. Nikoozadeh, I.O Wygant, D.S. Lin, O. Oralkan, A.S. Ergun, D.N. Stephens, K. Thomenius, A. Dentinger, D. Wildes, G. Akopyan, K. Shivkumar, A. Mahajan, D.J. Sahn, B.T. Khuri-Yakub, "Forward-Looking Intracardiac Ultrasound Imaging Using a 1-D CMUT Array Integrated With Custom Front-End Electronics", *IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control,* vol. 55, no. 12, December 2008

[30] G. Gurun, P. Hasler, F.L. Degertekin, "A 1.5-mm Diameter Single-Chip CMOS Front-End System with Transmit-Receive Capability for CMUT-on-CMOS Forward-Looking IVUS", *IEEE International Ultrasonics Symposium 2011,* pp. 478-481, 2011

[31] G. Gurun, M. Hochman, P. Hasler, F.L. Degertekin, "Thermal‐Mechanical-Noise-Based CMUT Characterization and Sensing", *IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control,* vol. 59, no. 6, June 2012

[32] A. Bhuyan, J.W. Choe, B.C. Lee, I.O. Wygant, A. Nikoozadeh, O. Oralkan, B.T. Khuri-Yakub, "Integrated Circuits for Volumetric Ultrasound Imaging With 2-D CMUT Arrays", *IEEE Transactions on Biomedical Circuits and Systems*, Vol. 7, No. 6, December 2013

[33] I.O. Wygant, N.S. Jamal, H.J. Lee, A. Nikoozadeh, O. Oralkan, M. Karaman, B.T. Khuri-Yakub, "An Integrated Circuit With Transmit Beamforming Flip-Chip Bonded to a 2-D CMUT Array for 3-D Ultrasound Imaging", *IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control*, vol. 56, no. 10, October 2009

[34] D.F. Lemmerhirt, X. Cheng, O. D. Kripfgans, M. Zhang and J. B. Fowlkes, "A Fully- Populated 32x32 CMUT-in-CMOS Array", *IEEE International Ultrasonics Symposium 2010,* pp. 559-562, 2011

[35] L. R. Cenkeramaddi, A. Bozkurt, F. Y. Yamaner and T. Ytterdal, "A Low Noise Capacitive Feedback Analog Front-end for CMUTs in Intra Vascular Ultrasound Imaging", *IEEE International Ultrasonics Symposium 2007,* pp. 2143-2146, 2007

[36] T. Singh, T. Soether and T. Ytterdal, "Common Source Amplifier with Feedback Biasing in 90nm CMOS", *Ph. D. Research in Microelectronics and Electronics 2006*,

[37] S. Peng, M. S. Qureshi, A. Basu, R. O. Guldiken, F. L. Degertekin and P. E. Hasler, "Floating-gate Based CMUT Sensing Circuit Using Capacitive Feedback Charge Amplifier", *IEEE International Ultrasonics Symposium 2006,* pp. 2425-2428, 2006

[38] S. Bose and P. Mandal, "A Fully Differential Amplifier with CMOS Feedback Biasing for sensing CMUT signals", *2014 IEEE Students' Technology Symposium (TechSym),* pp. 159-163, 2014

[39] S. Berg, T. Ytterdal and A. Rønnekleiv, "Co-optimization of CMUT and receive amplifiers to suppress effects of neighbor coupling between CMUT elements", *IEEE International Ultrasonics Symposium 2008,* pp. 2103-2106, 2008

[40] M.Sautto, D.Leone, A.Savoia, D.Ghisu, F.Quaglia, G.Caliano and A.Mazzanti, "A CMUT Transceiver Front-End with 100-V TX Driver and 1-mW Low-Noise Capacitive Feedback RX Amplifier in BCD-SOI Technology", *ESSCIRC 2014 - 40th European Solid State Circuits Conference (ESSCIRC),* pp. 407-410, September 2014

[41] Xiwei Huang, Jia Hao Cheong, Hyouk-Kyu Cha, Hongbin Yu, Minkyu Je, and Hao Yu, "A High-frequency Transimpedance Amplifier for CMOS Integrated 2D CMUT Array towards 3D Ultrasound Imaging", *35th Annual International Conference of the IEEE EMBS*, Osaka, Japan, 3 - 7 July, 2013

[42] I. O. Wygant, N. S. Jamal, H. J. Lee, A. Nikoozadeh, Ö. Oralkan, M. Karaman, and Butrus T. Khuri-Yakub, "An Integrated Circuit with Transmit Beam forming Flip-Chip Bonded to a 2-D CMUT Array for 3-D Ultrasound Imaging," *IEEE Trans. Ultrason. Ferroelectr. Freq. Control.*, vol. 56, no. 10, pp. 2145– 2156, Oct. 2009.

[43] G. I. Athanasopoulos, S. J. Carey, and J. V. Hatfiel, "Circuit Design and Simulation of a Transmit Beamforming ASIC for High-Frequency Ultrasonic Imaging Systems," *IEEE Trans. on Ult., Ferro Elec. and Freq. con.,* vol. 58, no. 7, pp. 1320–1331, Jul. 2011.

[44] M. Maymandi-Nejad and M. Sachdev, "A Monotonic Digitally Controlled Delay Element", *IEEE Journal of Solid-State Circuits,* vol. 50, no. 11, pp. 2212–2219, 2005.

[45] Der-Song Lin, "Interface Engineering of Capacitive Micromachined Ultrasonic Transducers for Medical Applications", *Ph.D. Thesis,* Stanford University, 2011.

[46] Y. Moghe, T. Lehmann, and T. Piessens, "Nanosecond Delay Floating High Voltage Level Shifters in a 0.35 um HV-CMOS Technology," *IEEE J. Solid-State Circuits*, vol. 46, no. 2, pp. 485-497, Feb. 2011.

[47] S. Eto, H. Akita, K. Isobe, K. Tsuchida, H. Toda, and T. Seki, "A 333MHz, 20mW, 18ps Resolution Digital DLL Using Current Controlled Delay with Parallel Variable Resistor DAC (PVR-DAC)," *IEEE 2nd Asia Pacific Conf. on ASIC*, , pp. 349–350, 2000.

[48] M. G. Johnson, E. L. Hudson, and H. Kopka, "A Variable Delay Line PLL for CPU-Coprocessor Synchronization," IEEE J. Solid-State Circuits, vol. 23, pp. 1218–1223, Oct. 1988.

[49] J. Dunning, J. Lundberg, and E. Nuckolls, "An All-Digital Phase Locked Loop with 50-Cycle Lock Time Suitable for High Performance Microprocessors," *IEEE J.Solid-Sate Circuits*, vol. 30, no. 4, pp. 412–422, Apr. 1995.

[50] J. S. Chiang and K. Y. Chen, "The Design of an All-Digital Phase Locked Loop with Small DCO Hardware and Fast Phase Lock," *IEEE Trans.Circuits Syst*. I, vol. 46, no. 7, pp. 945–950, July, 1999.
158

[51] M. Saint-Laurent and G. P. Muyshondt, "A Digitally Controlled Oscillator Constructed Using Adjustable Resistors," *IEEE in Proc. Southwest Symp. Mixed-Signal Design*, pp. 80–82, 2001.

[52] M. Saint-Laurent and M. Swaminathan, “A Digitally Adjustable Resistor for Path Delay Characterization in High Frequency Microprocessors,” *IEEE in Proc. Southwest Symp. Mixed-Signal Design*, pp. 61–64, 2001.

[53] H. Noda, M. Aoki, H. Tanaka, O. Nagashima, and H. Aoki, “An On-Chip Clock Adjusting Circuit with Sub 100-Ps Resolution for a High Speed DRAM Interface,” IEEE Trans. Circuits Syst.II, vol. 47, no. 8, pp. 771–775, Aug. 2000.

[54] F. Baronti, D. Lunardini, R. Roncella, and R. Saletti, “A Self Calibrating Delay-Locked Delay Line with Shunt Capacitor Circuit Scheme,” *IEEE J. Solid-State Circuits*, vol. 39, no. 2, pp. 384-387, Feb. 2004.

[55] M. Maymandi-Nejad, and M. Sachdev, “A Digitally Programmable Delay Element: Design and Analysis,” *IEEE trans. Very Large Scale Integration (VLSI) Systems*., vol. 11, no. 5, pp. 871-878, Oct. 2003.

[56] D. Pan, H. W. Li, and B. M. Wilamowski, “A low Voltage to High Voltage Level Shifter Circuit for MEMS Application,” *IEEE in Proc. 15th Biennial University/Government/Industry Microelectronics Symp*., pp. 128–131, July, 2003.

[57] B. Choi, “Enhancement of Current Driving Capability in Data Driver ICs for Plasma Display Panels,” *IEEE Trans. Consumer Electron*., vol. 55, no. 3, pp. 992–997, Aug. 2009.

[58] G. C. Moyer, “The Vernier Techniques for Precise Delay Generation and other Applications,” *Ph.D. dissertation*, Dept. of Electrical and Computer Engineering, Univ. of Raleigh, Durham, NC, 1996.

[59] J. Buyle, V. De Gezelle, B. Bakeroot, and J. Doutreloigne, “A New Type of Level-Shifter for n-type High Side Switches Used in High-Voltage Switching ADSL Line-Drivers,” *IEEE in Proc. Int. Conf. Electronics, Circuits and Systems.*, pp. 954–957, Aug. 2008.

[60] Parisa Sharif, “Design and Microfabrication of Annular Concentric Capacitive Micromachined Ultrasonic Transducer”, *M.Sc. Thesis,* Middle East Technical University, 2014

[61] Harry Veendrick , “Wire Self-heating in Supply Lines on Bulk-CMOS ICs”, *Solid-State Circuits Conference, 2002,* pp. 199-202, 2002

[62] Analog, Embedded Processing, Sem,conductor Company, Texas Instruments, TI.com

APPENDIX A

MATLAB PROGRAM DATA

```
function varargout = cmutDelayCalculation(varargin)
% CMUTDELAYCALCULATION M-file for cmutDelayCalculation.fig
%      CMUTDELAYCALCULATION, by itself, creates a new
CMUTDELAYCALCULATION or raises the existing
%      singleton*.
%
%      H = CMUTDELAYCALCULATION returns the handle to a new
CMUTDELAYCALCULATION or the handle to
%      the existing singleton*.
%
%
CMUTDELAYCALCULATION('CALLBACK',hObject,eventData,handles,...)
calls the local
%      function named CALLBACK in CMUTDELAYCALCULATION.M with the
given input arguments.
%
%      CMUTDELAYCALCULATION('Property','Value',...) creates a new
CMUTDELAYCALCULATION or raises the
%      existing singleton*.  Starting from the left, property
value pairs are
%      applied to the GUI before cmutDelayCalculation_OpeningFcn
gets called.  An
%      unrecognized property name or invalid value makes property
application
%      stop.  All inputs are passed to
cmutDelayCalculation_OpeningFcn via varargin.
%
%      *See GUI Options on GUIDE's Tools menu.  Choose "GUI
allows only one
%      instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help
cmutDelayCalculation

% Last Modified by GUIDE v2.5 01-Jan-2009 00:27:03

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
```

```
                   'gui_Singleton',  gui_Singleton, ...
                   'gui_OpeningFcn',
@cmutDelayCalculation_OpeningFcn, ...
                   'gui_OutputFcn',
@cmutDelayCalculation_OutputFcn, ...
                   'gui_LayoutFcn',  [] , ...
                   'gui_Callback',   []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT


% --- Executes just before cmutDelayCalculation is made visible.
function cmutDelayCalculation_OpeningFcn(hObject, eventdata,
handles, varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)
% varargin   command line arguments to cmutDelayCalculation (see
VARARGIN)

% Choose default command line output for cmutDelayCalculation
handles.output = hObject;
icon_1=imread('Ultramems.jpg');
axes(handles.icon_axes);
imshow(icon_1);
axis off

% Update handles structure
guidata(hObject, handles);

% UIWAIT makes cmutDelayCalculation wait for user response (see
UIRESUME)
% uiwait(handles.figure1);


% --- Outputs from this function are returned to the command
line.
function varargout = cmutDelayCalculation_OutputFcn(hObject,
eventdata, handles)
% varargout  cell array for returning output args (see
VARARGOUT);
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
```

```
varargout{1} = handles.output;



function x_axis_edit_Callback(hObject, eventdata, handles)
% hObject    handle to x_axis_edit (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of x_axis_edit as
text
%        str2double(get(hObject,'String')) returns contents of
x_axis_edit as a double


% --- Executes during object creation, after setting all
properties.
function x_axis_edit_CreateFcn(hObject, eventdata, handles)
% hObject    handle to x_axis_edit (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    empty - handles not created until after all
CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end



function y_axis_edit_Callback(hObject, eventdata, handles)
% hObject    handle to y_axis_edit (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of y_axis_edit as
text
%        str2double(get(hObject,'String')) returns contents of
y_axis_edit as a double


% --- Executes during object creation, after setting all
properties.
function y_axis_edit_CreateFcn(hObject, eventdata, handles)
% hObject    handle to y_axis_edit (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    empty - handles not created until after all
CreateFcns called

% Hint: edit controls usually have a white background on Windows.
```

```
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end



function z_axis_edit_Callback(hObject, eventdata, handles)

function z_axis_edit_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function speed_edit_Callback(hObject, eventdata, handles)
function speed_edit_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function calculate_button_Callback(hObject, eventdata, handles)
if (get(handles.twoDim_button,'value'))==1
    global A;
    A=cmutElement;
    A(1,1).xc=358.15;
    A(1,1).yc=358.15;
    A(1,1).zc=0;
    A(1,2).xc=-358.15;
    A(1,2).yc=358.15;
    A(1,2).zc=0;
    A(1,3).xc=-358.15;
    A(1,3).yc=-358.15;
    A(1,3).zc=0;
    A(1,4).xc=358.15;
    A(1,4).yc=-358.15;
    A(1,4).zc=0;
    A(2,1).xc=943.98;
    A(2,1).yc=943.98;
    A(2,1).zc=0;
    A(2,2).xc=-943.98;
    A(2,2).yc=943.98;
    A(2,2).zc=0;
    A(2,3).xc=-943.98;
    A(2,3).yc=-943.98;
    A(2,3).zc=0;
    A(2,4).xc=943.98;
    A(2,4).yc=-943.98;
    A(2,4).zc=0;
    A(3,1).xc=1301.78;
    A(3,1).yc=1301.78;
    A(3,1).zc=0;
    A(3,2).xc=-1301.78;
    A(3,2).yc=1301.78;
    A(3,2).zc=0;
    A(3,3).xc=-1301.78;
```

```
    A(3,3).yc=-1301.78;
    A(3,3).zc=0;
    A(3,4).xc=1301.78;
    A(3,4).yc=-1301.78;
    A(3,4).zc=0;
    A(4,1).xc=1606.9;
    A(4,1).yc=1606.9;
    A(4,1).zc=0;
    A(4,2).xc=-1606.9;
    A(4,2).yc=1606.9;
    A(4,2).zc=0;
    A(4,3).xc=-1606.9;
    A(4,3).yc=-1606.9;
    A(4,3).zc=0;
    A(4,4).xc=1606.9;
    A(4,4).yc=-1606.9;
    A(4,4).zc=0;
    fpxc=str2num(get(handles.x_axis_edit,'string'));
    fpyc=str2num(get(handles.y_axis_edit,'string'));
    fpzc=str2num(get(handles.z_axis_edit,'string'));
    dist=zeros(4,4);
    for i=1:1:4
        for j=1:1:4
            dist(i,j)=sqrt((A(i,j).xc - fpxc)^2 + (A(i,j).yc -
fpyc)^2 + (A(i,j).zc - fpzc)^2);
        end
    end
else
    global A;
    A=cmutElement;
    A(1,1).xc=506.5;
    A(1,1).yc=0;
    A(1,1).zc=0;
    A(1,2).xc=0;
    A(1,2).yc=0;
    A(1,2).zc=0;
    A(1,3).xc=0;
    A(1,3).yc=0;
    A(1,3).zc=0;
    A(1,4).xc=0;
    A(1,4).yc=0;
    A(1,4).zc=0;
    A(2,1).xc=1334.98;
    A(2,1).yc=0;
    A(2,1).zc=0;
    A(2,2).xc=0;
    A(2,2).yc=0;
    A(2,2).zc=0;
    A(2,3).xc=0;
    A(2,3).yc=0;
    A(2,3).zc=0;
    A(2,4).xc=0;
    A(2,4).yc=0;
    A(2,4).zc=0;
    A(3,1).xc=1840.99;
    A(3,1).yc=0;
    A(3,1).zc=0;
    A(3,2).xc=0;
    A(3,2).yc=0;
```

```
    A(3,2).zc=0;
    A(3,3).xc=0;
    A(3,3).yc=0;
    A(3,3).zc=0;
    A(3,4).xc=0;
    A(3,4).yc=0;
    A(3,4).zc=0;
    A(4,1).xc=2272.49;
    A(4,1).yc=0;
    A(4,1).zc=0;
    A(4,2).xc=0;
    A(4,2).yc=0;
    A(4,2).zc=0;
    A(4,3).xc=0;
    A(4,3).yc=0;
    A(4,3).zc=0;
    A(4,4).xc=0;
    A(4,4).yc=0;
    A(4,4).zc=0;
    fpxc=str2num(get(handles.x_axis_edit,'string'));
    fpyc=str2num(get(handles.y_axis_edit,'string'));
    fpzc=str2num(get(handles.z_axis_edit,'string'));
    dist=zeros(4,4);
    for i=1:1:4
        for j=1:1:4
            dist(i,j)=sqrt((A(i,j).xc - fpxc)^2 + (A(i,j).yc -
fpyc)^2 + (A(i,j).zc - fpzc)^2);
        end
    end
    dist(1,2)=0;
    dist(1,3)=0;
    dist(1,4)=0;
    dist(2,2)=0;
    dist(2,3)=0;
    dist(2,4)=0;
    dist(3,2)=0;
    dist(3,3)=0;
    dist(3,4)=0;
    dist(4,2)=0;
    dist(4,3)=0;
    dist(4,4)=0;
end
T=zeros(4,4);
spd=str2num(get(handles.speed_edit,'string'));
for i=1:1:4
    for j=1:1:4
        T(i,j)=dist(i,j)/spd;
    end
end
for i=1:1:4
    for j=1:1:4
        A(i,j).cl=T(i,j);
    end
end
dist_1=dist;
P=zeros(4,4);
dmax=max(max(dist_1));
a=1;
while dmax>0
```

```
    for i=1:1:4
        for j=1:1:4
            if dist_1(i,j)==dmax
                P(i,j)=a;
                dist_1(i,j)=0;
            end
        end
    end
    a=a+1;
    dmax=max(max(dist_1));
end
set(handles.priority_list,'data',P);
set(handles.clock_latency_list,'data',T);
set(handles.distances_list,'data',dist);
function fpga_button_Callback(hObject, eventdata, handles)
close cmutDelayCalculation;
h5=guihandles(fpgaSettings);
function distances_list_CellEditCallback(hObject, eventdata,
handles)
function distances_list_CellSelectionCallback(hObject, eventdata,
handles)
function twoDim_button_Callback(hObject, eventdata, handles)
function oneDim_button_Callback(hObject, eventdata, handles)

function varargout = fpgaSettings(varargin)
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                   'gui_Singleton',  gui_Singleton, ...
                   'gui_OpeningFcn', @fpgaSettings_OpeningFcn,
...
                   'gui_OutputFcn',  @fpgaSettings_OutputFcn, ...
                   'gui_LayoutFcn',  [] , ...
                   'gui_Callback',   []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
function fpgaSettings_OpeningFcn(hObject, eventdata, handles,
varargin)
handles.output = hObject;
icon_4=imread('Ultramems.jpg');
axes(handles.icon_axes);
imshow(icon_4);
axis off

guidata(hObject, handles);

function varargout = fpgaSettings_OutputFcn(hObject, eventdata,
handles)
varargout{1} = handles.output;
```

```
function oneshot_settings_button_Callback(hObject, eventdata,
handles)
close fpgaSettings;
h4=guihandles(oneshotSettings);

function dco_settings_button_Callback(hObject, eventdata,
handles)
close fpgaSettings;
h1=guihandles(dcoSettings);


function set_fpga_button_Callback(hObject, eventdata, handles)
global dco;
global oneshot;
global muxe;
fpga=fpgaElement;

fpga.dco_s0=dco.d0;
fpga.dco_s1=dco.d1;
fpga.dco_s2=dco.d2;
fpga.dco_s3=dco.d3;
fpga.dco_s4=dco.d4;

fpga.oneshot_s0=oneshot.d0;
fpga.oneshot_s1=oneshot.d1;
fpga.oneshot_s2=oneshot.d2;
fpga.oneshot_pulsew=oneshot.pulsew;

if (get(handles.oneshot_button,'Value'))==1
    fpga.demux_s=0;
elseif (get(handles.dco_button,'Value'))==1
    fpga.demux_s=1;
end


fpga.mux8_s0=muxe.s0;
fpga.mux8_s1=muxe.s1;
fpga.mux8_s2=muxe.s2;


if (get(handles.mux4_ground_button,'Value'))==1
    fpga.mux4_s0=0;
    fpga.mux4_s1=0;
elseif (get(handles.mux4_fpga_button,'Value'))==1
    fpga.mux4_s0=1;
    fpga.mux4_s1=0;
elseif (get(handles.mux4_oneshot_button,'Value'))==1
    fpga.mux4_s0=0;
    fpga.mux4_s1=1;
elseif (get(handles.mux4_dco_button,'Value'))==1
    fpga.mux4_s0=1;
    fpga.mux4_s1=1;
end
```

```
function exit_button_Callback(hObject, eventdata, handles)
close all;

function calculate_button_Callback(hObject, eventdata, handles)
    global A;
    if (get(handles.clock100_button,'Value'))==1
        c1=100;
    elseif (get(handles.clock50_button,'Value'))==1
        c1=50;
    else
        c1=1;
    end

    cycle=1/c1;
    L=zeros(4,4);

    for i=1:1:4
        for j=1:1:4
            L(i,j)=round((A(i,j).cl)/cycle);
        end
    end
    set(handles.latency_list,'data',L);


function mux8_s0_button_Callback(hObject, eventdata, handles)
function mux8_s1_button_Callback(hObject, eventdata, handles)
function mux8_s2_button_Callback(hObject, eventdata, handles)
function mux4_oneshot_button_Callback(hObject, eventdata,
handles)
    if (get(handles.mux4_oneshot_button,'value'))==1
        set(handles.oneshot_settings_button,'visible','on');
    else
        set(handles.oneshot_settings_button,'visible','off');
    end
    if (get(handles.mux4_dco_button,'value'))==1
        set(handles.dco_settings_button,'visible','on');
    else
        set(handles.dco_settings_button,'visible','off');
    end


function mux4_ground_button_Callback(hObject, eventdata, handles)
    if (get(handles.mux4_oneshot_button,'value'))==1
        set(handles.oneshot_settings_button,'visible','on');
    else
        set(handles.oneshot_settings_button,'visible','off');
    end
    if (get(handles.mux4_dco_button,'value'))==1
        set(handles.dco_settings_button,'visible','on');
    else
        set(handles.dco_settings_button,'visible','off');
    end

function mux4_fpga_button_Callback(hObject, eventdata, handles)
    if (get(handles.mux4_oneshot_button,'value'))==1
        set(handles.oneshot_settings_button,'visible','on');
    else
```

```
        set(handles.oneshot_settings_button,'visible','off');
    end
    if (get(handles.mux4_dco_button,'value'))==1
        set(handles.dco_settings_button,'visible','on');
    else
        set(handles.dco_settings_button,'visible','off');
    end


function mux4_dco_button_Callback(hObject, eventdata, handles)
    if (get(handles.mux4_oneshot_button,'value'))==1
        set(handles.oneshot_settings_button,'visible','on');
    else
        set(handles.oneshot_settings_button,'visible','off');
    end
    if (get(handles.mux4_dco_button,'value'))==1
        set(handles.dco_settings_button,'visible','on');
    else
        set(handles.dco_settings_button,'visible','off');
    end


function varargout = oneshotSettings(varargin)
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                   'gui_Singleton',  gui_Singleton, ...
                   'gui_OpeningFcn', @oneshotSettings_OpeningFcn,
...
                   'gui_OutputFcn',  @oneshotSettings_OutputFcn,
...
                   'gui_LayoutFcn',  [] , ...
                   'gui_Callback',   []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end


function oneshotSettings_OpeningFcn(hObject, eventdata, handles,
varargin)

handles.output = hObject;
icon_3=imread('Ultramems.jpg');
axes(handles.icon_axes);
imshow(icon_3);
x3=[10 9.8 9.7 9.6 7.5 7 5 2.5];
y2=[0 1 2 3 4 5 6 7];
axes(handles.plot_axes);
plot(y2,x3,'*');
grid on
xlabel('Corresponding integer of bit setting');
ylabel('Pulse Width (ns)');
```

```
guidata(hObject, handles);

function varargout = oneshotSettings_OutputFcn(hObject,
eventdata, handles)

varargout{1} = handles.output;



function width_edit_Callback(hObject, eventdata, handles)
function width_edit_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end


function calculate_button_Callback(hObject, eventdata, handles)
    vals=[0 9.8 9.7 9.6 7.5 7 5 2.5];
    var1=str2double(get(handles.coarse_edit,'String'));
    var2=var1/10;
    var5=floor(var2);
    var3=var1 - (var5*10);
    dz=[0 1 0 1 0 1 0 1];
    do=[0 0 1 1 0 0 1 1];
    dt=[0 0 0 0 1 1 1 1];
    tempval=[];
    for i=1:1:8
        tempval(i)= abs(vals(i)-var3);
    end
    for i=1:1:7
        if tempval(i) == tempval(i+1)
            tempval(i+1)=10;
        end
    end
    ind=find(tempval == min(tempval(:)));
    var4=var5*10+vals(ind);
    if ind == 0
        var4 = var4 +10;
    end
    set(handles.width_edit,'string',var4);
    set(handles.d0_button,'Value',dz(ind));
    set(handles.d1_button,'Value',do(ind));
    set(handles.d2_button,'Value',dt(ind));


function set_button_Callback(hObject, eventdata, handles)
global oneshot;
oneshot=oneshotElement;
    if (get(handles.d0_button,'Value'))==0
        oneshot.d0=0;
    elseif (get(handles.d0_button,'Value'))==1
        oneshot.d0;
    end
    if (get(handles.d1_button,'Value'))==0
        oneshot.d1=0;
```

```
    elseif (get(handles.d1_button,'Value'))==1
        oneshot.d1=1;
    end
    if (get(handles.d2_button,'Value'))==0
        oneshot.d2=0;
    elseif (get(handles.d2_button,'Value'))==1
        oneshot.d2=1;
    end

    oneshot.pulsew=str2num(get(handles.coarse_edit,'String'));
function back_button_Callback(hObject, eventdata, handles)
close;
h3=guihandles(fpgaSettings);
function d0_button_Callback(hObject, eventdata, handles)


function d1_button_Callback(hObject, eventdata, handles)
function d2_button_Callback(hObject, eventdata, handles)


function coarse_edit_Callback(hObject, eventdata, handles)
function coarse_edit_CreateFcn(hObject, eventdata, handles)

if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end



function varargout = dcoSettings(varargin)
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                   'gui_Singleton',  gui_Singleton, ...
                   'gui_OpeningFcn', @dcoSettings_OpeningFcn, ...
                   'gui_OutputFcn',  @dcoSettings_OutputFcn, ...
                   'gui_LayoutFcn',  [] , ...
                   'gui_Callback',   []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end

function dcoSettings_OpeningFcn(hObject, eventdata, handles,
varargin)
handles.output = hObject;
icon_2=imread('Ultramems.jpg');
axes(handles.icon_axes);
imshow(icon_2);
x1=[256 250 244 238 222 217 213 200 196 179 175 164 156 143 137
127 178 169 158 151 142 130 122 112 100 85 81 65 53 39 34 20];
```

```
y1=[0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23
24 25 26 27 28 29 30 31];
axes(handles.plot_axes);
plot(y1,x1,'*');
grid on
xlabel('Corresponding integer of bit setting');
ylabel('Frequency (MHz)');
guidata(hObject, handles);

function varargout = dcoSettings_OutputFcn(hObject, eventdata,
handles)
varargout{1} = handles.output;



function fdco_edit_Callback(hObject, eventdata, handles)
function fdco_edit_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end


function set_button_Callback(hObject, eventdata, handles)
    global dco;
    dco=dcoElement;
    global muxe;
    muxe=mux8Element;
    if (get(handles.d0_button,'Value'))==0
        dco.d0=0;
    elseif (get(handles.d0_button,'Value'))==1
        dco.d0;
    end
    if (get(handles.d1_button,'Value'))==0
        dco.d1=0;
    elseif (get(handles.d1_button,'Value'))==1
        dco.d1=1;
    end
    if (get(handles.d2_button,'Value'))==0
        dco.d2=0;
    elseif (get(handles.d2_button,'Value'))==1
        dco.d2=1;
    end
    if (get(handles.d3_button,'Value'))==0
        dco.d3=0;
    elseif (get(handles.d3_button,'Value'))==1
        dco.d3=1;
    end
    if (get(handles.d4_button,'Value'))==0
        dco.d4=0;
    elseif (get(handles.d4_button,'Value'))==1
        dco.d4=1;
    end
    if (get(handles.s00_button,'Value'))==0
        muxe.s0=0;
    elseif (get(handles.s00_button,'Value'))==1
        muxe.s0=1;
    end
```

```
    if (get(handles.s11_button,'Value'))==0
        muxe.s1=0;
    elseif (get(handles.s11_button,'Value'))==1
        muxe.s1=1;
    end
    if (get(handles.s22_button,'Value'))==0
        muxe.s2=0;
    elseif (get(handles.s22_button,'Value'))==1
        muxe.s2=1;
    end
function back_button_Callback(hObject, eventdata, handles)
    close;
    h2=guihandles(fpgaSettings);

function calculate_button_Callback(hObject, eventdata, handles)

    freqlist=[256 250 244 238 222 217 213 200 196 179 175 164 156
143 137 127 178 169 158 151 142 130 122 112 100 85 81 65 53 39 34
20];
    freqlist1=[];
    for i=1:1:8
        for j=1:1:32
            freqlist1(i,j)=freqlist(j)/(2^i);
        end
    end
    freq=str2double(get(handles.fdco_edit,'string'));
    tempfreq=[];
    for i=1:1:8
        for j=1:1:32
            tempfreq(i,j)=abs(freqlist1(i,j)-freq);
        end
    end
    [ind,ind2]=find(tempfreq==min(tempfreq(:)));
    set(handles.afdco_edit,'String',freqlist1(ind,ind2));

    b1=de2bi([ind-1],3);
    set(handles.s00_button,'value',b1(1));
    set(handles.s11_button,'value',b1(2));
    set(handles.s22_button,'value',b1(3));
    b2=de2bi([ind2-1],5);
    set(handles.d0_button,'value',b2(1));
    set(handles.d1_button,'value',b2(2));
    set(handles.d2_button,'value',b2(3));
    set(handles.d3_button,'value',b2(4));
    set(handles.d4_button,'value',b2(5));

function d0_button_Callback(hObject, eventdata, handles)
function d1_button_Callback(hObject, eventdata, handles)
function d2_button_Callback(hObject, eventdata, handles)
function d3_button_Callback(hObject, eventdata, handles)
function d4_button_Callback(hObject, eventdata, handles)

function afdco_edit_Callback(hObject, eventdata, handles)
function afdco_edit_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
```

```
function s00_button_Callback(hObject, eventdata, handles)
function s22_button_Callback(hObject, eventdata, handles)
function s11_button_Callback(hObject, eventdata, handles)

classdef cmutElement
    properties(GetAccess = 'public', SetAccess = 'public')
        xc
        yc
        zc
        dtfp
        pn
        cl
    end

end
classdef fpgaElement
    properties(GetAccess = 'public', SetAccess = 'public')
        demux_s
        oneshot_s0
        oneshot_s1
        oneshot_s2
        oneshot_pulsew
        dco_s0
        dco_s1
        dco_s2
        dco_s3
        dco_s4
        mux8_s0
        mux8_s1
        mux8_s2
        mux4_s0
        mux4_s1
    end

end
classdef dcoElement
    properties(GetAccess = 'public', SetAccess = 'public')
        d0
        d1
        d2
        d3
        d4
    end

end
classdef mux8Element
    properties(GetAccess = 'public', SetAccess = 'public')
        s0
        s1
        s2
    end

end
classdef oneshotElement
    properties(GetAccess = 'public', SetAccess = 'public')
        d0
        d1
        d2
```

```
        pulsew
    end
end
```

FPGA PROGRAM DATA

```
-- Company:
-- Engineer:
-- Create Date:    09:48:33 06/30/2015
-- Design Name:
-- Module Name:    top - Behavioral
-- Project Name:
-- Target Devices:
-- Tool versions:
-- Description:
-- Dependencies:
--
-- Revision:
-- Revision 0.01 - File Created
-- Additional Comments:
----------------------------------------------------------------------------------
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use ieee.numeric_std.all;
use ieee.std_logic_unsigned.all;
use IEEE.STD_LOGIC_ARITH.ALL;
-- Uncomment the following library declaration if using
-- arithmetic functions with Signed or Unsigned values
--use IEEE.NUMERIC_STD.ALL;
-- Uncomment the following library declaration if instantiating
-- any Xilinx primitives in this code.
```

```
--library UNISIM;
--use UNISIM.VComponents.all;
entity top is
   Port ( clk :                 in       STD_LOGIC;
          sw :                  in       STD_LOGIC_VECTOR (7 downto 0);
          rx :                  in              STD_LOGIC;
          setd :          out      STD_LOGIC;
          setd2 :         out      STD_LOGIC;
          setd3 :                  out      STD_LOGIC;
          setd4 :         out      STD_LOGIC;
                           txa:                out      STD_LOGIC;
                           txc:                out      STD_LOGIC;
                           flash:              out      STD_LOGIC;
          segment7 :      out      STD_LOGIC_VECTOR (6 downto 0));
end top;

architecture Behavioral of top is

          component baudgenerator
          Port (   clk : in  STD_LOGIC;
                                 baud : out  STD_LOGIC);
          end component;
          component bgen
          Port (   clk : in  STD_LOGIC;
                                 b16 : out  STD_LOGIC);
          end component;

          component seg
          Port (   clk : in  STD_LOGIC;
```

```
                        bcd : in  STD_LOGIC_VECTOR (3 downto 0);
                        segment7 : out  STD_LOGIC_VECTOR (6 downto
0));
        end component;
        component trx
        Port ( clk : in  STD_LOGIC;
                    en : in  STD_LOGIC;
                    data: in        std_logic_vector (6 downto 0);
       tx : out  STD_LOGIC);
        end component;

--      component receiver
--  Port ( rin : in  STD_LOGIC;
--         clk : in  STD_LOGIC;
--         rout : out  STD_LOGIC_VECTOR (7 downto 0));
--      end component;

        signal segment:     std_logic_vector (6 downto 0);
        signal baudrate:    std_logic;
        signal bcdtemp:     std_logic_vector (3 downto 0);
        signal txd:             std_logic;
        signal data:        std_logic_vector (6 downto 0);
        signal rout:        std_logic_vector (7 downto 0);
        signal en:              std_logic:='0';
        signal rxd:             std_logic;
        signal count1: integer range 0 to 3:=0;
        signal ledcount: integer range 0 to 9600:=0;
        signal st0:         std_logic:='1';
        signal st1:         std_logic:='1';
        signal st2:         std_logic:='1';
```

```
signal st3:             std_logic:='1';
signal tf:      std_logic:='0';
signal br16:    std_logic;

signal cbit:    std_logic:='0';
signal ebit:    std_logic:='0';
signal errorbit:        std_logic:='1';
signal datan:   integer range 0 to 7:=0;
signal stopb:   integer range 0 to 7:=0;
signal cnt:     integer range 0 to 2603:=0;
signal cnt1:    integer range 0 to 5207:=0;
signal cnt2:    integer range 0 to 5207:=0;
signal rtemp:           std_logic_vector (7 downto 0):="11111111";



begin

        I1:     baudgenerator       port map (clk, baudrate);
        I2:     seg                 port map    (baudrate,
bcdtemp, segment);
        I3:     trx                 port map    (br16, en,
data,txd);
        I4:     bgen                port map    (clk, br16);

--      I5:     receiver            port map    (rxd, br16,
rout);

        process (clk)
                begin
                        if (rising_edge(clk)) then
```

```
if(rxd='0' and cbit='0' and cnt<2603) then
        cnt<=cnt+1;
elsif(rxd='0' and cbit='0' and cnt=2603) then
        cnt<=0;
        cbit<='1';
elsif (rxd='1' and cbit='0' and cnt<2603) then
        cnt<=0;
end if;

if (cbit='1') then
        if( cnt1<5207) then
                cnt1<=cnt1+1;
        elsif (cnt1=5207 and datan<7) then
                cnt1<=0;
                rtemp(datan)<= rxd;
                datan<=datan+1;
        elsif (cnt1=5207 and datan=7) then
                cnt1<=0;
                rtemp(datan)<= rxd;
                datan<=0;
                ebit<='1';
                cbit<='0';
        end if;
end if;

if (ebit='1') then
        if (cnt2<5207) then
                cnt2<=cnt2+1;
```

```
        elsif (cnt2=5207 and stopb<7) then
                if (rxd='1') then
                        cnt2<=0;
                        stopb<=stopb+1;
                else
                        cnt2<=0;
                        stopb<=0;
                        errorbit<='1';
                        ebit<='0';
                end if;
        elsif (cnt2=5207 and stopb=7) then
                if (rxd='1') then
                        cnt2<=0;
                        stopb<=0;
                        ebit<='0';
                        errorbit<='0';
                else
                        cnt2<=0;
                        stopb<=0;
                        errorbit<='1';
                        ebit<='0';
                end if;
        end if;
end if;
if (errorbit='0') then
        rout<=rtemp;
else
        rout<=rout;
end if;
```

```
                end if;
end process;



process (baudrate)

        begin

                if (baudrate'event and baudrate='1') then

                        if(count1=0) then
                                st0 <='0';
                                st1 <='1';
                                st2 <='1';
                                st3 <='1';
                                bcdtemp<=sw(7 downto 4);
                        elsif (count1=1) then
                                st0 <='1';
                                st1 <='0';
                                st2 <='1';
                                st3 <='1';
                                bcdtemp<= sw(3 downto 0);
                        elsif (count1=2) then
                                st0 <='1';
                                st1 <='1';
                                st2 <='0';
                                st3 <='1';
```

```
                bcdtemp<= rout(7 downto 4);
            elsif (count1=3) then
                st0 <='1';
                st1 <='1';
                st2 <='1';
                st3 <='0';
                bcdtemp<= rout(3 downto 0);
                count1<=0;

            end if;
            count1<=count1+1;
        end if;
end process;

process (clk)
    begin
        if(clk'event and clk='1') then

            txa <= txd;
            en<= sw(7);

            data<= sw(6 downto 0);

        end if;

end process;

process(baudrate)
    begin
        txc<=rxd;
```

```
		end process;
--		process(clk)
--			begin
--				if (rtemp="00000000") then
--					rout<="11111111";
--				elsif (rtemp="11111111") then
--					rout<="11111111";
--				elsif (rtemp="00110001") then
--					rout<="00110001";
--				elsif (rtemp="00110010" ) then
--					rout<="00110010";
--				end if;
--
--		end process;



		process (br16)
			begin
				if(rising_edge(br16)) then
					if (ledcount=0) then
						if (rxd='0') then
							tf<='1';
							ledcount<=ledcount+1;
						else
							tf<='0';
						end if;
					elsif ((ledcount>0) and (ledcount<9600))
then
```

```
                    ledcount<=ledcount+1;
                elsif (ledcount=9600) then
                    ledcount<=0;
                end if;

            end if;
        end process;



        flash<=tf;
        setd <= st0;
        setd2 <= st1;
        setd3 <= st2;
        setd4 <= st3;
        rxd<=rx;

        segment7 <= segment;

end Behavioral;
```