Design and Implementation of a Dual-Mode Ultrasound Array Driver

A THESIS
SUBMITTED TO THE FACULTY OF THE GRADUATE SCHOOL
OF THE UNIVERSITY OF MINNESOTA
BY

Andrew Jacob Casper

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
Master of Science

Emad Ebbini

November, 2011
Acknowledgements

My parents deserve top billing in any acknowledgements I give, thesis or otherwise. They are just fantastic people who somehow got stuck with me as a son, but they’ve made the best of it and I appreciate that.

This thesis wouldn’t have been possible without my advisor, Emad Ebbini. I’m very grateful for the chance to work in such a dynamic and innovative lab. It’s a testament to the environment he’s created in the lab, that the excitement I felt after my first day still hasn’t gone away.

There have been many of my fellow students who have had a positive impact on my research and experience here, but three deserve special mention: Casey Murray, John Ballard and Dalong Liu.

Casey has been a good friend from my first year as an EE undergrad to now. We’ve had hours of meaningful, thought provoking discussions, and many more hours of not so meaningful discussions, all of it much appreciated.

John is the type of guy who, no matter what the situation, manages to have a genuine and positive impact on everyone involved, that was certainly the case with me.

I’ve really enjoyed working with Dalong. He is a top rate engineer who constantly surprises me with the creativity and professionalism in his work.
Dedication

For my grandfather
Abstract

Ultrasound has a long history as an important medical diagnostic tool. Its non-ionizing nature and relative low cost has enabled this technology to gain widespread acceptance and use in hospitals worldwide. It’s currently used to create images of many internal body structures allowing for rapid assessment by a physician. While it is in the diagnostic imaging context that ultrasound is most commonly used, it is however, not the only medical use of ultrasound. Ultrasound is also capable of non-invasively targeting organs and tissue for therapeutic benefits. These benefits can range from non-invasive drug delivery to tissue cauterization. Recent advances in piezocomposite transducer technology have allowed for a new generation of array transducers that are capable of both delivering ultrasound therapy, and imaging with the same device. These transducers, referred to as Dual-Mode Ultrasound Arrays (DMUAs), use the same elements for therapy and imaging, allowing for absolute registration between therapeutic and imaging coordinates. Therefore, realtime DMUA imaging provides unique form of feedback to the physician allowing her/him to identify and quantify the exposure to any obstacles in the path of the therapeutic beam. This feedback provides the basis for realtime resynthesis of the therapeutic beam to minimize the exposure to these obstacles while maximizing the exposure at the target.

The advantages of the DMUA approach to image-guided surgery can be realized only with drivers that fully integrate the imaging and therapy functions in a seamless manner. This thesis describes the design and implementation of two real-time DMUA drivers. The first system was an enhancement of a previous design that allowed for the basic features of the DMUA system to be demonstrated. The second was a new design that allowed for a wider range of operation and the implementation of microsequencer to precisely control imaging and therapy sequences.
Contents

Acknowledgements i

Dedication ii

Abstract iii

List of Tables vi

List of Figures vii

1 Introduction 1

1.1 Background ...................................................... 1

1.2 Dual Mode Ultrasound Array ................................... 2

1.2.1 Therapeutic Ultrasound .................................. 3

1.2.2 Ultrasound Imaging ....................................... 3

1.2.3 Optimal Synthesis of Array Field Patterns ............... 5

1.2.4 Digital Driver ............................................. 8

1.3 Conclusion and Summary ....................................... 10

2 System 11

2.1 Introduction .................................................... 11

2.2 Design 1: Bitmap Based Driver ............................... 12
# List of Tables

<table>
<thead>
<tr>
<th>Table</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1</td>
<td>Choices of Weighting Matrix</td>
<td>6</td>
</tr>
<tr>
<td>2.1</td>
<td>Spartan 3E 500k Resources</td>
<td>16</td>
</tr>
<tr>
<td>2.2</td>
<td>1 MHz, 3 cycle signal, delayed 750 ns</td>
<td>23</td>
</tr>
<tr>
<td>2.3</td>
<td>Virtex 5 SX50T Resources</td>
<td>26</td>
</tr>
<tr>
<td>2.4</td>
<td>Microsequencer Field Definitions</td>
<td>27</td>
</tr>
<tr>
<td>2.5</td>
<td>Termination Flag Definitions</td>
<td>28</td>
</tr>
<tr>
<td>2.6</td>
<td>Array Definition File</td>
<td>33</td>
</tr>
</tbody>
</table>
List of Figures

1.1 Simulated Patterns for a 1 MHz, 64 Element DMUA . . . . . . . . . . . 7
1.2 Relationship between Duty Cycle and Driving Amplitude . . . . . . . . . . . 10
2.1 Spartan 3E Driver . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 13
2.2 Bitmap Based Array Driver . . . . . . . . . . . . . . . . . . . . . . . . . 15
2.3 Virtex 5 FPGA Evaluation Board . . . . . . . . . . . . . . . . . . . . . . 18
2.4 State Machine Driver . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 19
2.5 Adjustable Parameters . . . . . . . . . . . . . . . . . . . . . . . . . . . . 21
2.6 Digital Driver State Machine . . . . . . . . . . . . . . . . . . . . . . . . 22
2.7 State Machine Based Array Driver . . . . . . . . . . . . . . . . . . . . . 26
2.8 Therapy Pulse with interleaved STF pulse . . . . . . . . . . . . . . . . . . 30
2.9 Ramp Up and Down with Sinusoidal Modulation . . . . . . . . . . . . . . 31
2.10 3D model of 1 MHz DMUA . . . . . . . . . . . . . . . . . . . . . . . . . 34
2.11 Close up of Dicing on the 1 MHz DMUA . . . . . . . . . . . . . . . . . . 34
2.12 3D model of 3.5 MHz, Fenestrated DMUA . . . . . . . . . . . . . . . . . . 35
3.1 Temperature Control Setup . . . . . . . . . . . . . . . . . . . . . . . . . 41
3.2 Power Reallocation for Double Focus . . . . . . . . . . . . . . . . . . . . . 43
3.3 With and Without Dynamic Power Reallocation . . . . . . . . . . . . . . . 44
3.4 Independent Temperature Control . . . . . . . . . . . . . . . . . . . . . . 45
3.5 Temperature Map of Phantom during Experiment . . . . . . . . . . . . . . 45
3.6 In-Vitro Porcine Kidney Temperature Control . . . . . . . . . . . . . . . . 47
<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.7</td>
<td>Porcine Kidney Tissue Damage</td>
<td>48</td>
</tr>
<tr>
<td>3.8</td>
<td>Synthetic Aperture Image of Pre-Therapy Setup</td>
<td>49</td>
</tr>
<tr>
<td>3.9</td>
<td>Single Focal Point with Simulation</td>
<td>50</td>
</tr>
<tr>
<td>3.10</td>
<td>Double Focal Point with Simulation</td>
<td>51</td>
</tr>
<tr>
<td>3.11</td>
<td>Therapy and Imaging Sequence</td>
<td>51</td>
</tr>
<tr>
<td>3.12</td>
<td>Location of Focus During Therapy</td>
<td>52</td>
</tr>
<tr>
<td>3.13</td>
<td>Single Transmit Focus Image 1.5 Seconds into Therapy</td>
<td>53</td>
</tr>
<tr>
<td>3.14</td>
<td>Single Transmit Focus Image Post Boiling</td>
<td>54</td>
</tr>
<tr>
<td>3.15</td>
<td>Synthetic Aperture Image Post Therapy</td>
<td>55</td>
</tr>
</tbody>
</table>
Chapter 1

Introduction

1.1 Background

Ultrasound is a broad term encompassing all acoustics above the limits of human hearing. Its uses are too numerous to list, being found everywhere from a bat looking for its next meal, to an expectant mother seeing her child for the first time. Given ultrasound’s ubiquitous nature, it’s no surprise that it can trace its theoretical roots to such physics and mathematical heavyweights as Euler, Poisson, Rayleigh, and Curie, among many others. The mathematics describing ultrasound were laid down hundreds of years ago, yet it wasn’t until the first World War that ultrasound was put to practical use. Motivated by the sinking of the Titanic and the rising of submarine warfare, the French scientist Paul Langevin used the recent discovery of the piezoelectric effect to create a rudimentary sonar system capable of detecting underwater targets [2].

These early sonar experiments provided the first demonstration of not only ultrasound’s utility, but also its ability to affect the biology of living creatures when fish in close proximity of the ultrasound transducer were killed. Early research by scientist Robert Wood and scientific philanthropist, Alfred Lee Loomis confirmed ultrasound’s destructive ability by showing it capable of heating and destroying biological tissues
It wasn’t until the 1940’s and 50’s that ultrasound found its first medical diagnostic use by measuring tissue properties. Single element transducers were used to measure the thickness of tissue layers along with the echogenicity (i.e. how much backscatter is produced as ultrasound passes through a medium) of cancerous and non-cancerous tissue. The first ultrasound scanner capable of producing ultrasound images was created at the University of Minnesota, by John Wild and John Reid. This scanner was the precursor to the diagnostic scanners used in hospitals today [18].

Both ultrasound imaging and therapy have come a long way since the 1950’s but it wasn’t until recently that the technology existed to incorporate both of these modes of operation into a single device [6]. This device, referred to as a Dual Mode Ultrasound Array (DMUA), is described throughout the rest of this chapter and thesis.

1.2 Dual Mode Ultrasound Array

The DMUA is a unique device that can only really exist in a field such as ultrasound, it is a therapy device that also serves as an imaging probe. The DMUA uses advances in piezoelectric materials to create an array of elements that are efficient enough to deliver high levels of acoustic power, but also wideband enough to be used for imaging. Unlike other devices which use one ultrasound transducer to deliver therapy and a second to image the medium, the DMUA uses the same elements to do both. Using the same elements offers the ultimate form of registration. There are no transformations that need to be applied to bring the imaging coordinates to the therapy coordinates, with the DMUA they are one and the same. This registration is critically important for the DMUA since the multi-element configuration allows for very fine control over where the ultrasound is deposited [1].
1.2.1 Therapeutic Ultrasound

Ultrasound is attractive in a medical setting because it uses a non-ionizing radiation to propagate energy into the body. The biological effects of this energy propagation depend heavily on the intensity of the ultrasound waves. The intensity of ultrasound used in diagnostic imaging varies based on the target. For fetal ultrasound, the FDA allows intensities up to $94 \text{ mW/cm}^2$. When targeting the heart, the FDA allows intensities of up to $430 \text{ mW/cm}^2$. These levels of ultrasound minimize the possible biological effects of the ultrasound waves while still allowing for the generation of useful images.

In contrast to diagnostic imaging, ultrasound therapies intend to cause biological effects and tend to use higher intensity waves. The intensities used range many orders of magnitude depending on the desired effect. On the low end, intensities of just a few $\text{W/cm}^2$ are capable of raising the temperature of tissue a few degrees over the course of many minutes to hours. Intensities on the order of $1000 \text{ W/cm}^2$ can cause tissue damage in the span of a few seconds. On the very high end, intensities over $20,000 \text{ W/cm}^2$ can cause tissue boiling in just milliseconds. The DMUAs described in this thesis are capable of generating intensities of a few thousand $\text{W/cm}^2$ resulting in exposure times on the order of a second.

1.2.2 Ultrasound Imaging

Traditional diagnostic ultrasound imaging systems rely on a large number of closely spaced elements and conventional beamforming to form an image. Images of the acoustic medium directly in front of the transducer can be built up line by line by grouping elements together and transmitting a wave into the medium. Acoustic impedance differences at interfaces within the medium directly in front of the transmit elements will generate echoes which can be received by the same elements. This process can be repeated across the whole transducer to create an ultrasound image.

---

1 For comparison, the human ear is capable of hearing intensity levels as low as $10^{-14} \text{ W/cm}^2$. 
There are two primary reasons why traditional beamforming techniques are not well-suited for use in generating DMUA images. First, the elements of a DMUA typically lie on a concave surface to help focus the energy efficiently during therapy. This would severely limit the field of view if a traditional beamforming method was employed. Second, the elements of a DMUA are by necessity much larger than those used on a diagnostic scanner due to the amount of energy an element on a DMUA must convey into the field. This not only creates a coarsely sampled array, it also limits the total number of elements.

In spite of these limitations, there are imaging methods available to the DMUA. The two imaging techniques most commonly used by the DMUA are Synthetic Aperture (SA) imaging, and Single Transmit Focus (STF) imaging. While this thesis is not intended to be an imaging paper, these imaging methods create unique needs on the driver which merit some discussion.

SA imaging is the highest quality form of delay-and-sum beamforming available on any imaging array. SA imaging involves transmitting on one element at a time while receiving on all elements. This process can be repeated with different elements to synthetically create a transmit aperture [14]. In contrast to traditional diagnostic imaging, SA imaging is able to focus at all locations both in transmit and receive modes. SA imaging requires a driver that can quickly switch which element is transmitting a pulse. SA imaging assumes the imaging medium is time-invariant throughout imaging process. In a dynamic medium, if the time required to switch between element transmissions is long, this assumption will no longer hold true leading to degraded image quality [23].

STF imaging, while not unique to DMUAs, is a very useful mode of imaging before, during, and after delivering therapeutic HIFU shots. STF imaging involves using all elements to pulse a truncated version of the therapy wave. The imaging and therapy device share the same elements, so the acoustic medium seen by both is identical. Constructing an imaging pulse from the therapy pulse, allows the backscattered imaging echoes to be used as a map of energy deposition during therapy. STF imaging requires
the driver to be able to load and play a few cycles of a therapy driving pattern with
great speed. Often these STF imaging pulses are interleaved with therapy, so any delay
in loading an STF imaging pulse can severely compromise the overall therapy.

1.2.3 Optimal Synthesis of Array Field Patterns

Like any phased array, the DMUA can be steered by controlling the phase and amplitude
of the signals on the elements.\[5\] The pressure at discrete points in the field due to a
chosen phase and amplitude pattern can be calculated by (1.1).

\[ p = Hu \] \hspace{1cm} (1.1)

For an N element array, \( u \) is a \( N \times 1 \) vector which contains the complex representation
of the driving pattern for each element. The \( L \times 1 \) vector, \( p \), represents the pressures
observed at the \( L \) observation points. The elements of the matrix \( H \) are the directivities
of the array elements at the observation points calculated from the Green’s functions
of the elements, i.e. \( H(i,j) \) represents the Green’s function of element \( j \) evaluated at
location \( i \). When the array is radiating into a half space, the elements of \( H \) could
be easily obtained using the Rayleigh-Sommerfeld integral \[12\]. For inhomogeneous
media, they could be computed or directly measured. For the purposes of this thesis
work, the homogeneous half-space Green’s function was assumed and the entries of \( H \)
were evaluated using a method similar to the rectangular radiator method described by
Ocheltree and Frizzel \[19\].

In therapeutic ultrasound, the inverse problem of determining a driving pattern from
specifying the pressure at certain locations in the field is of great interest. A quick look
at the dimensions of the matrices in (1.1) suggests that if the number of target locations
is less than the number of elements, there exists an infinite number of driving patterns
capable of creating the specified pressures. In this case it is more informative to ask
not, can a driving pattern be synthesized, but which driving pattern should be used.
The work presented here used a Lagrangian multipliers approach to solve for a driving
pattern that both satisfied the forward problem, but also minimized some weighted norm \(1.2\) shows the general form of the inverse solution \([4, 20]\).

\[
\bar{u} = W^{-1}H'(HW^{-1}H')^{-1}p
\]

The choice of \(W\) does not affect the pressures at the targeted locations, but it does have a large influence on the spatial distribution of the acoustic field. Table 1.1 details the three main choices of \(W\) and their influence on the synthesized field.

Table 1.1: Choices of Weighting Matrix

<table>
<thead>
<tr>
<th>(W)</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Identity Matrix</td>
<td>Minimizes the total energy in the driving pattern.</td>
</tr>
<tr>
<td>(H_e' H_e)</td>
<td>(H_e) represents a propagation matrix from the array to certain control points. This weighting synthesizes a driving pattern which minimizes the energy at these control points</td>
</tr>
<tr>
<td>(\text{diag}(W_i) = W_{i-1}\text{diag}(</td>
<td>u_{i-1}</td>
</tr>
</tbody>
</table>

The weighting choices presented in Table 1.1 offer great latitude in the synthesis of a driving pattern. In addition to the applicability of the method to conventional focusing and steering of single-focus patterns, it allows for the synthesis of multiple-focus patterns with constraints at the foci and auxiliary optimization criteria at other critical points in the field. This is illustrated by the simulation result shown in Figure 1.1. Using
the second option in Table 1.1, one can design a pattern that directs energy away from critical structures (i.e. ribs, nerves, etc...), while maintaining the desired intensity levels at the target(s). Figure 1.1 shows two simulated patterns which target two focal locations. The left image shows an instance where the identity matrix was used to synthesize the field. The right image shows a case were energy was directed away from a region prefocally while still targeting two separate focal points.

The final weighting choice in Table 1.1 provides a method for compressing the dynamic range of the driving amplitudes. There are two main reasons one may wish to have a driving amplitude with limited dynamic range. One, there are driving systems that are able to vary only the phase of the driving signals. For such systems, any driving pattern which does not have uniform weighting would be unrealizable. Two, a uniform driving pattern allows the array to output higher levels of power. Imagine two driving patterns which both create the desired focal pressures, but one pattern has a single
element at a normalized amplitude of 0.5 and the others at 0.1, and the other pattern
has a uniform amplitude of 0.125. If the maximum amplitude the array can sustain is
1.0, then the pressures at the focal points for the first pattern can only be double before
the driving pattern would saturate the array, whereas the second pattern could achieve
focal pressures eight times higher before saturating the array. For both these reasons,
it’s nice to at least have the option of compressing the dynamic range of the driving
pattern.

1.2.4 Digital Driver

Each element of $\bar{u}$ can be represented by a phase and amplitude, this requires a driver
be able to control the phase and amplitude of a sinusoidal wave on individual channels.
A digital driver accomplishes this by modulating the width and phase of a square wave
and then filtering this wave to generate a sinusoid. The phase of a channel can be
adjusted by delaying or advancing the rising edge of the square wave. The amplitude
of the signal can be adjusted by varying the duty cycle of the signal.

The relationship between the amplitude and the duty cycle of the square wave
deserves a brief discussion. To properly relate the duty cycle to the resulting signal
amplitude, one must look at the Fourier Series of a square wave with variable duty
cycle to determine the signal level in the frequency of interest.

One period of a square wave with a variable duty cycle can be defined as:

$$f(t) = \begin{cases} 
0 : & -\frac{T}{2} < t < -\frac{aT}{2} \\
1 : & -\frac{aT}{2} < t < \frac{aT}{2} \\
0 : & \frac{aT}{2} < t < \frac{T}{2} 
\end{cases}$$

Using this signal to calculate the Fourier Series components leads to a very straightforward integral.
\[ A_k = \frac{2}{T} \int_{-\frac{\alpha T}{2}}^{\frac{\alpha T}{2}} \cos \left( \frac{2\pi kt}{T} \right) \, dt \]  
(1.3)

\[ B_k = \frac{2}{T} \int_{-\frac{\alpha T}{2}}^{\frac{\alpha T}{2}} \sin \left( \frac{2\pi kt}{T} \right) \, dt \]  
(1.4)

The square wave, as defined, is an even function which allows us to ignore the computation of the \( B_k \) terms as they will all be zero.

Carrying out the integration for the \( A_k \) terms gives the following:

\[ A_k = \frac{2}{\pi k} \sin(\alpha \pi k) \]  
(1.6)

Typically, in a digital driver, the square wave is driven at the desired frequency of excitation, so it is the \( k = 1 \) term that is of interest. Plugging \( k = 1 \) into (1.6) gives the relationship between the driving amplitude and the duty cycle as: \( B_1 = \frac{2}{\pi} \sin(\alpha \pi) \). It’s generally more useful to use the inverse of this function to find the duty cycle from a desired amplitude. It’s also desirable to have 1.0 correspond to the maximum amplitude level. Using this normalization, the inverse is easily obtained as \( \frac{1}{\pi} \arcsin \left( \frac{C B_1}{2} \right) \) where \( C \) is equal to \( \frac{2}{\pi} \sin(0.5\pi) (\approx 0.63661977) \). Figure 1.2 shows the relationship between the driving amplitude and the required duty cycle.
Figure 1.2: Relationship between Duty Cycle and Driving Amplitude

This mapping establishes the ability of a digital driver to control, via a variable duty cycle, the amplitude of a sinusoidal signal.

1.3 Conclusion and Summary

Ultrasound is an ever evolving field, driven forward both by new technologies and new applications. Today, as before, advances in piezo-composite materials and computing hardware is allowing a new form of ultrasound therapy to take hold. The DMUA is a device which can combine imaging and therapy into a single device to deliver safe, reliable, and low cost treatments. This thesis describes the design and operation of a system to implement a realtime DMUA driver.
Chapter 2

System

2.1 Introduction

This chapter describes the design of the hardware and software necessary for a real-time DMUA driver. Real-time operation of the diver is key to demonstrating the system’s capabilities in the pre-clinical in vivo experiments currently being planned. It is also important for laboratory experiments, with ex vivo tissues and tissue-mimicking phantoms, designed to demonstrate the real-time feedback control of tissue exposure to therapeutic HIFU beams. This is due to the nature of HIFU exposure where single HIFU shots are typically between 1 - 10 seconds in duration. A real-time feedback control algorithm for these short exposures requires a driver capable of switching array patterns within milliseconds. A system that requires minutes for each update severely limits the type of experiments that can be run. This was the case with the driver that was used previously for validating many of the core DMUA concepts.

In addition to the obvious motivation of developing the ability to track a moving target, a real-time driver is necessary for coping with the fast tissue changes in response to HIFU exposure. These tissue dynamics range from cavitation to tissue boiling over short durations (from tens of microseconds to hundreds of milliseconds). In addition, we
have previously proposed an image-based refocusing algorithm for minimizing the HIFU exposure to strongly reflecting targets in the path of the HIFU beam \[1\]. In practice, such an algorithm may be implemented adaptively where a number of test beams are used before deciding on the optimal pattern. This requires a high-speed switching of test patterns to arrive at the optimal in realistically short time (order of milliseconds.)

2.2 Design 1: Bitmap Based Driver

The general layout of the following descriptions will be to first discuss the FPGA platform chosen for the driver, followed by the communication protocol chosen to transfer data from the CPU to the FPGA, and finally, the actual implementation. It’s hoped that this flow will help to mimic the design process, where ideas are often constrained by the realities of hardware.

2.2.1 FPGA

The FPGA chosen for this driver was the Xilinx Spartan 3E, 500K. This choice was motivated by previous experimentation with evaluation boards containing this FPGA. Figure 2.1 shows a picture of the assembled Spartan 3E driver board. The FPGA is outlined with the purple box. Some of the necessary supporting hardware can be seen around the FPGA, there is a 50 MHz clock and JTAG programmer to the left, and a programmable ROM (PROM) chip to the right.
IO for the board can be seen outlined in red and blue. The blue outline along the top of the board is the driver output. These ribbon cable connectors allowed the signal to mate with a switching driver constructed by many students over many years, but most recently worked on by Dr. Hanwoo Lee. The IO outlined in red were primarily digital control signals. The outline along the bottom was used for trigger in/out and silence in/out signals. The column of connectors along the left side of the board was used to distribute clock signals from the driver to a high speed data acquisition unit. Without a common clock between the ultrasonic driver and the data acquisition unit, the per acquisition jitter can become unacceptably high, degrading the overall image.

2.2.2 USB 2.0 Communications

The communications link is critically important to the system’s overall intended use as a research platform. It must quickly, and without errors, transfer data to and from the CPU and FPGA. Modern computers have many data transfer options available (serial,
parallel, USB, PCI, PCIexpress, Ethernet, etc...), with varying speeds and protocol complexity. The best protocol would be fast and easy to interface with an FPGA. Serial and parallel protocols were ruled out as too slow, while PCIexpress and Ethernet were ruled out as too complex for the chosen FPGA. The optimal solution for this board was found in a USB 2.0 module made by Bitwise Systems, outlined in black in Figure 2.1.

This module abstracted away much of the USB protocol details, presenting an 8-bit address bus and an 8-bit data bus to the FPGA. In addition, a software library was provided that could be interfaced with C++ code allowing for easy integration with the overall research platform. These modules provided a nice balance between transfer speed (USB 2.0 has an ideal maximum throughput of 54 MB/s) and complexity. The typical transfer speed achieved was significantly below the 54 MB/s theoretical limit, more around 30 MB/s and was dependent upon the loading of the bus (i.e. if the driver was connected the same hub as an external hard drive transferring data, the achieved data rate was much slower).

2.2.3 Bitmap Based Driver

In a bitmap based driver, each channel is represented as a series of bits arranged sequentially. To play a signal, this series of bits is clocked out bit by bit, at a frequency well above the intended driving frequency. To play more than one pulse, this bitstream can be cycled back on itself over and again. An entire array driving pattern can be represented as an arrangement of these one bit sequences into a 2D bitmap.

Figure 2.2 shows a Spartan 3E implementation of this driver topology. The design sets aside two, 64x440 bit buffers to store driving patterns. These driving patterns can be of any length below 440 bits. Data is clocked out of each buffer at 200 MHz, allowing for 100 amplitude levels and a phase resolution of 1.8 degrees. USB 2.0 is used to download data and control commands to the FPGA. The speed of USB 2.0 allows for more than 1000 bitmap updates per second.
To help facilitate basic imaging tasks, an SA imaging unit was created to free the buffers from having to perform this chore. Without such a unit, an entire bitmap needs to be downloaded for a single channel excitation.

![Spartan 3E BRAM Based Driver](image)

Figure 2.2: Bitmap Based Array Driver

The two bitmap buffers are used to allow for glitchless updating of driving patterns. The driver can output one buffer, while the other is being loaded with a new pattern. This allows focal points to be swept in space without needing to stop therapy while a new pattern is loaded.

These two buffers can also be used to store separate therapy patterns and STF imaging patterns. It is often desirable to use two different amplitudes for the imaging and therapy patterns. This driver allows both pulses to be loaded into memory and interleaved quickly during an experiment.

Another key component of this driver is the silence signal. This is a signal that
allows for external hardware to silence the therapy beam. The ability to silence a beam is often useful when integrating diagnostic imaging devices with the DMUA. The therapy beam from the DMUA is so powerful that, even though it may not operate at the same frequency, it can cause significant interference with a diagnostic system.

To minimize the effect of the silencing on the overall treatment plan, an exposure timer was created that tracks the amount of time the DMUA is actually playing a signal vs. being silenced. This timer allows the user, for example, to specify a 4 second exposure without concern over the amount of time the beam is silenced. If there is no silencing, the treatment will cease in 4 seconds as directed. If a diagnostic probe silences the therapy beam for 10 ms every 20 ms, then the treatment will proceed for a total of 8 seconds. While this exposure timer doesn’t guarantee all average power levels to be equal, it at least assures that the intended amount of energy was delivered. It also allows for retrospective analysis of what occurred during the therapy.

Table 2.1 shows the resources consumed and those available with the Spartan 3E 500k FPGA. The available block RAM is quite large for a 64 channel driver. One thing to note about the block RAM is that it is arranged in 18K blocks. This means that an application that would wish to store a single byte in BRAM would be forced to consume an entire 18K block. This can lead to situations where the actual memory used is significantly more than would be predicted based on the asked for memory alone.

<table>
<thead>
<tr>
<th>Table 2.1: Spartan 3E 500k Resources</th>
</tr>
</thead>
<tbody>
<tr>
<td>Slices</td>
</tr>
<tr>
<td>Block RAM</td>
</tr>
<tr>
<td>DCMs</td>
</tr>
<tr>
<td>User IO Pins</td>
</tr>
<tr>
<td>Multipliers</td>
</tr>
</tbody>
</table>
The resource consumption detailed in Table 2.1 make it clear that this driver is essentially glue logic surrounding some high speed memory. By far, the most utilized resource is the BRAM. The rest of the FPGA serves as a conduit to guide data from the USB port to the BRAM, and from the BRAM to the channel outputs.

There are two main limitations with this current implementation, phase unwrapping and memory usage. Phase unwrapping refers to removing the $2\pi$ modulo wraps in a phase profile. When pulsing an STF waveform it is generally advisable to unwrap the phase. This helps to ensure that all wavefronts arrive at the focal location at the same time. With a bitmap based driver, it is not possible to both unwrap the phase, and cycle through the bitmap. This requires that multiple cycles be stored in the bitmap, and the bitmap be cycled through only once, which can lead to excessively long bitmaps. The second problem is that the bitmap uses a large chunk of memory to define a wave that can be described by a few sets of parameters (i.e. frequency, phase, amplitude, etc...). In addition, the total memory required is dependent on the value of the parameters used. A 1 MHz wave requires twice as much memory as a 2 MHz wave. It would be nice to simply store the waveform parameters and dynamically generate the square wave based on these parameters. These two limitations led to the development of a state machine based driver.

### 2.3 Design 2: State Machine Based Driver

The just described system worked well both as a demonstration platform to showcase the DMUA concept, as well as a research platform for experimentation. The system was used for months, serving as the DMUA driver during phantom, *in vitro*, and *in vivo* experiments. It’s heavy usage is a testament to its utility. Motivated by the limitations of the Spartan 3E driver and the bitmap architecture, a new driver was designed and implemented. This section describes that driver.
2.3.1 FPGA

The brain of this new system was a Virtex-5 SX50T development board, shown in Figure 2.3. This is a $1400 board available from electronics supplier AVNET. This board provides all the supporting circuitry for the Virtex 5 SX50T (red rectangle), along with a number of different external ICs that enable 2 Gb Ethernet connections (blue rectangle), up to 8 lanes of PCIexpress, and 2, 10 Gb Ethernet ports. This board also provides 2 high-speed, 120 pin connectors (black rectangle). These high-speed connectors allow a baseboard to mount directly on top of the FPGA.

![Figure 2.3: Virtex 5 FPGA Evaluation Board](image)

Figure 2.3 shows the complete board with the baseboard in place, and a blue ribbon cable connected to one of the outputs. In this revision of the driver, the decision was made to use the shown blue ribbon cable instead of the ribbon cable of the previous revision. This blue ribbon cable consists of 40, micro-coaxial cables attached to a polarized header on each end. Not only does the cable help the overall signal integrity of the driver, but the polarized header prevents the user from mis-connecting cables. Each cable was used to carry 32 channels from the driver to the amplifier.
2.3.2 Gigabit Ethernet Communications

This revised system uses a Gigabit Ethernet communication link to transfer data between the CPU and FPGA. As its name suggests, Gigabit Ethernet has a maximum transfer speed of 1 billion bits/second. Not only is this link high speed, it also has very low latency. In practice, packets can be sent from the FPGA to the CPU, processed, and sent back down to the FPGA within 40 us.

Another major benefit of using Gb Ethernet is the ability of the FPGA to send data packets to the CPU unprompted. In this way the CPU does not need to poll the FPGA
to determine if there is an event that needs to be taken care of. Instead, the FPGA can
transmit a packet which can alert the CPU that something just occurred or that some
action needs to be taken.

Much of the complexity associated with the Gb Ethernet protocol can be abstracted
away through the choice of FPGA and development board. The chosen Virtex-5 de-
development board contains two PHY chips that implement the physical link layer of the
Ethernet Protocol. The Virtex-5 SX50T contains 4 Ethernet MACs which implement
the data link layer of the Ethernet Protocol. This frees the developer from dealing
with the lower layers of the communication scheme, but still requires a transport layer
implementation.

The first step in developing the transport layer was the creation of an IPv4 packet
engine to encapsulate the data. IP is a fairly lightweight protocol that requires the
generation of a 20 byte header. Most of the entries are static from transmission to
transmission (i.e. destination IP address, sender IP address, etc...), the one entry which
is not static is the header. The header requires the calculation of a checksum which is
dependent on the length of the packet, so the header, with the correct checksum, must
be generated for each new packet.

UDP was chosen to encapsulate the data inside an IP packet. This protocol has no
mechanism for detecting transmission errors, but is able to transfer data with very little
overhead. A UDP packet can be generated in software of the host computer and sent
down to the FPGA, where each layer of the protocol is stripped away until the raw data
can be recovered. In practice, this communication scheme was able to achieve well over
100 MB/s sustained data transfer speeds with no dropped frames.

2.3.3 State Machine Concept

Unlike arbitrary waveform generators which can play multi-frequency waveforms, a dig-
ital driver is constrained to a mono-frequency excitation. As such, the wave it plays
can be described by the set of parameters listed in Figure 2.5. These parameters can be
used to construct an appropriate driving waveform on the fly, without the need to create and store a bitmap. In order to generate this waveform, a state machine was developed that could load stored parameters and create the driving waveform. Figure 2.6 shows a flowchart describing the operation of the state machine.

Figure 2.5: Adjustable Parameters
At its heart, this state machine is a series of four counters and comparators. One counter is used to delay the start of the sequence to allow for phase unwrapping. Another two counters are used within a single cycle to control the pulse width modulation and the period of the wave, and a final counter is used to count the total number of cycles played. This state machine based approach allows for a lightweight driver to be used on each channel to control the desired waveform with great flexibility. The memory
requirements of this approach are independent of the parameters chosen, a 1 MHz wave requires no more memory than 2 MHz wave.

One problem that still remains is that of amplitude and phase quantization. The number of representable amplitude and phase levels is dictated by the overall clock frequency. A 200 MHz clock was used to run the overall state machine. This speed struck a good balance between driving performance and the timing requirements of the FPGA. Any additional increase in driving frequency risked introducing timing errors and making the overall performance of the system unstable.

To be useful on an FPGA, the parameters listed in Figure 2.5 must be converted to a digital representation. The format used to store these parameters has a large impact on both the abilities of the driver and the amount of data required to represent a single driving pattern. A naive approach would be to use a floating point representation for these values. This would have the advantage of making the software side programming very easy (floats are very straightforward in C) but would make the FPGA side programming a nightmare. In addition, it is not necessary to use floating point precision for this representation. The state machine described in Figure 2.6 is synchronous to the driving clock, this allows us to represent the signal parameters in terms of clock cycles. As illustration, the description of a 1 MHz, 50% duty cycle, 3 cycle signal, delayed 750 ns with a 200 MHz driving clock would be:

<table>
<thead>
<tr>
<th>Description</th>
<th>Cycles</th>
</tr>
</thead>
<tbody>
<tr>
<td>Delay</td>
<td>150</td>
</tr>
<tr>
<td>Duty Cycle</td>
<td>100</td>
</tr>
<tr>
<td>Period</td>
<td>200</td>
</tr>
<tr>
<td>Duration</td>
<td>3</td>
</tr>
</tbody>
</table>

Table 2.2: 1 MHz, 3 cycle signal, delayed 750 ns

Representing the parameters in Figure 2.5 as an integer number of clock cycles is still not the end of the story. These numbers still must be represented by a certain
number of bits. In order to get some idea on the appropriate number of bits, the range of the values in Figure 2.5 must be considered.

In order to give this driver maximum flexibility over its driving frequency, 10 bits were used to represent the period and duty cycle parameters. At 200 MHz clock frequency the driver is capable of representing frequencies down to 200 KHz. This also gives the driver great flexibility in future choice of driving frequency. 8 bits could have been used to represent the period and duty cycle parameters of a 1 MHz wave with a 200 MHz driving clock, however, this design choice would have limited future driving clocks to under 256 MHz at a 1 MHz excitation. To push beyond this barrier, the state machine driver would need to be regenerated with an increased bit width.

As mentioned earlier, the purpose of the delay is to allow for individual element excitations to be delayed, such that the created wavefront converges at a single point in space simultaneously. To gain some understanding on the needed delays, we must look at reasonable time of flight differences for different focal points. The largest array currently used in this lab is a 1 MHz, 64 element, concave array with a geometric focus at 100 mm axially. When focused at the geometric focus, there is no required delays on the elements. As the steered focus begins to move away from the geometric focus, the array must begin to add delays to account for time of flight differences between the elements. To give the driver ability to target any conceivable points away from the geometric focus, 15 bits were used to represent the delay parameter. At 200 MHz, this allows for just over 163 us difference in time of flight. At a typical speed of sound in water, this corresponds to over 24 cm in path length difference between the elements, far beyond what is needed.

The final parameter that must be represented is the overall duration. Unlike the other parameters, this parameter is not represented in clock cycles, instead it is represented in overall wave cycles. The range of ultrasound therapy durations is quite large. Some high power ablations last only milliseconds, while other lower power hyperthermia treatments can last seconds to minutes. In spite of this large dynamic range, only 16
bits were allocated to represent the total duration. At first this may seem like far too few bits required to represent most therapies, as 16 bits would only represent around 65 ms worth of therapy for a 1 MHz signal. As will be explained later, this driving implementation incorporated a microsequencer. This microsequencer allowed for the same pattern to be recycled many times, so the overall duration of treatment could be extended far beyond the limit imposed by the 16 bit duration parameter.

2.3.4 Implementation

In order to be effective as a total array driver, each channel must be allocated its own state machine driver. This requires a large level of parallelism which makes an FPGA the perfect platform for this sort of implementation. Not only are counters and comparators readily synthesized in an FPGA, but modules can be created in parallel with ease. As seen in Table 2.3, the Virtex 5 SX50T had more than enough resources to implement this sort of driver.

Figure 2.7 shows a system level diagram of the implemented driver. In many ways, the flow of data in this driver is the same as the bitmap driver. Data is taken from the communications port and stored, and then at the appropriate time, the data is unloaded and played out.
The distribution of the resource utilization is significantly different in this driver, than in the previous case. Where as before the most consumed resource was the BRAM, in this case the most consumed resource is the FPGA logic. Part of this increased logic consumption is due to the additional logic needed to implement the Gb Ethernet connection, the rest of the logic is consumed by the state machine driver and the microsequencer.

Table 2.3: Virtex 5 SX50T Resources

<table>
<thead>
<tr>
<th>Resource</th>
<th>Allocation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Slices</td>
<td>6,046/8,160</td>
</tr>
<tr>
<td>Block RAM</td>
<td>70/132 blocks</td>
</tr>
<tr>
<td>DCMs</td>
<td>2/12</td>
</tr>
<tr>
<td>User IO Pins</td>
<td>103/480</td>
</tr>
<tr>
<td>Multipliers</td>
<td>0/288 (25 x 18 bits)</td>
</tr>
</tbody>
</table>

Using the bit lengths described above, each channel of a driving pattern could be
represented in 51 bits, and a whole 64 channel driving pattern could be represented in 3264 bits. Even accounting for latency and overhead, with the Gb Ethernet, frame rates of over 10,000 updates/second can easily be achieved.

2.3.5 Microsequencer

The implementation of the state machine driver alone on this FPGA left the device underutilized. This excess computing capacity allowed for the implementation of a microsequencer. A microsequencer is a piece of logic that allows for the precise sequencing of driving patterns to implement a therapeutic treatment plan. It consists of two parts, a library of driving patterns and a control structure to access and retrieve these patterns.

The library implemented in this microsequencer can hold up to 512 unique waveforms. Each waveform is stored in BRAM and identified by its address. These driving patterns can be stored before therapy and retrieved as needed, or, they can be updated during therapy to affect the course of treatment.

The control structure is charged with retrieving the proper driving pattern at the appropriate time and presenting this pattern to the state machine engine to be synthesized into the needed square wave. This is accomplished through a programmable playlist that stores five separate parameters as listed in Table 2.3.5. In addition to the parameters in Table 2.3.5, there is a 16 bit repetition limit that determines how many times to cycle through the downloaded playlist. Recycling through a playlist allows for efficient use of the waveform library.

<table>
<thead>
<tr>
<th>Address (8 bits)</th>
<th>Duration in us (16 bits)</th>
<th>Silence (1 bit)</th>
<th>Trigger (1 bit)</th>
<th>Termination Flags (2 bits)</th>
</tr>
</thead>
</table>

Table 2.4: Microsequencer Field Definitions

The address field stores the address of the requested driving pattern. The duration field specifies how long to wait until moving to the next driving pattern. This field
is always as long, or longer, than the duration of the requested driving pattern. This parameter allows for a silencing period to be inserted into therapy plans.

The silence bit signals if the requested pattern is allowed to be silenced. Not all driving patterns need to be silenced when the silencing signal is applied. For instance, imaging pulses do not create significant levels of interference with external diagnostic probes and therefore do not need to be silenced. This bit allows the driver to dynamically configure how the silencing signal is interpreted.

The trigger bit signals whether the driver should emit a trigger when it begins to play the requested pattern. Not all driving patterns need to emit a trigger when they commence playing, and in fact, can cause problems if they do. Often times therapy bursts are interleaved with imaging bursts, the driver must trigger the data collection system when the imaging pulse is begun, but it should not trigger the data collection system when the therapy burst starts. This bit allows for the trigger to be outputted selectively.

The final parameters stored are the termination flags. Table 2.5 describes how the different termination flags affect the behavior of the microsequencer.

<table>
<thead>
<tr>
<th>Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>00</td>
<td>After the current pattern finishes playing, move on to the next pattern in the playlist.</td>
</tr>
<tr>
<td>01</td>
<td>If the playlist has been cycled through the requested number of times, stop, otherwise, cycle through again.</td>
</tr>
<tr>
<td>10,11</td>
<td>Stop after this pattern.</td>
</tr>
</tbody>
</table>

Table 2.5: Termination Flag Definitions

The microsequencer can hold up to 512 slots allowing for the implementation of even the most complicated therapy plans.

Both the driving pattern library and the playlist are double buffered, allowing for
the user to update their parameters during therapy. Combining this ability, with the very low latency Gb Ethernet connection provides for a very agile controller capable of not only delivering therapy, but also responding to inputs during therapy.

2.4 Software

The final piece of the DMUA driving system is the software necessary to generate and download the desired driving patterns. The software implementation is broken into two parts, therapy definition, and waveform synthesis. The therapy definition involves specifying points in space, durations, and pressure amplitudes of a focal locations for targeting. The waveform synthesis takes these specified points and constructs the driving pattern, and playlist parameters for download to the FPGA. Splitting these two functions into two separate software packages, allows the waveform synthesis to be used repeatedly with different GUI front ends.

The following sections will first describe one therapy definition program commonly used to synthesize patterns, and then describe the core waveform synthesis software. There were actually a number of therapy definition programs developed, some that used images created by the DMUA to help guide the placement of focal spots. These software packages will be further described in the results section.

2.4.1 Therapy Shot Planning

A basic shot planning interface was developed to help make effective use of the microsequencer library of waveforms. This interface allowed for the arrangement of both therapy and imaging pulses, together with amplitude modulation of the therapy bursts.

Figure 2.8 shows a screen capture of a therapy burst interleaved with an STF imaging pulse. The STF pulse parameters are shown on the screen, focused at 4 mm laterally and 100 mm axially, a 2 cycle excitation, and a dwell time of 300 $\mu$s. Along the bottom of the screen can be seen a graphical representation of the therapy. The STF pulse
is preceded by a 500 $\mu$s therapy burst with a 700 $\mu$s dwell time, focused at the same location as the imaging pulse.

Figure 2.8: Therapy Pulse with interleaved STF pulse

To help standardize the development of a standalone therapy planner, a text based therapy definition file format was developed. The intent of this file was to provide a standardized method of specifying therapy and imaging beams that could then be used by the waveform synthesis program to generate an appropriate driving pattern. In addition, by making the format of this file freely available, anyone who wishes to test out an imaging or therapy idea on the array can easily make a therapy definition file which can be loaded and played on the array.

As an example, the contents of the therapy definition file for Figure 2.8 are shown below.
31

2,4,0,100,1,500,700,0,0
0,2,4,0,100,1,300,1,0

These parameters are stored as text and can be read and edited in any text viewer.

The treatment planning software can be used to create more complex waveforms as well. Figure 2.9 shows an example of a therapeutic profile that modulates the therapy beam with a ramp up to a 10 Hz sinusoidal modulation, followed by a ramp back down. These modulations are implemented by sampling desired modulation and storing these samples in the waveform library of the microsequencer. The plot along the bottom shows the approximated staircase implementation of the desired modulation. The sinusoidal modulation shown in Figure 2.9 was sampled at 200 Hz and consumes only 152 of the available 512 memory slots.

Figure 2.9: Ramp Up and Down with Sinusoidal Modulation
The following shows a brief snippet of the therapy definition file for figure 2.9.

```plaintext
... 2,0,0,100,0.500001,4980,5000,0,1 2,0,0,100,0.407295,4980,5000,0,1 2,0,0,100,0.323665,4980,5000,0,1 2,0,0,100,0.257295,4980,5000,0,1 2,0,0,100,0.214683,4980,5000,0,1 2,0,0,100,0.2,4980,5000,0,1 2,0,0,100,0.214683,4980,5000,0,1 2,0,0,100,0.257295,4980,5000,0,1 ...
```

### 2.4.2 Waveform Synthesis

The implementation details of the previous sections have all assumed that a driving pattern exists, and focused on turning the driving pattern into a reality. This section details the software necessary to carry out the calculations in (1.2).

#### Array Definition

The first step to creating a driving pattern is to fully understand the geometry of the array and its elements. For each array, a Matlab file was created that holds the following set of variables
<table>
<thead>
<tr>
<th>Variable Name</th>
<th>Variable Dimensions</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>fullArray</td>
<td>$3 \times \text{numDicing} \times \text{numElements}$</td>
<td>Dices each array element (numElements total elements) into numDicing square sections with the $i$th dicing of the $j$th element centered at $\text{fullArray}(,:,i,j)$</td>
</tr>
<tr>
<td>geoCenter</td>
<td>$3 \times 1$</td>
<td>Specifies the geometric focal point of the array</td>
</tr>
<tr>
<td>normCenter</td>
<td>$3 \times \text{numElements}$</td>
<td>This variable describes where the center of each element is located. Primarily useful in Synthetic Aperture and Single Transmit Focus beamforming</td>
</tr>
<tr>
<td>normDir</td>
<td>$3 \times \text{numElements}$</td>
<td>Specifies the direction normal to the surface of the element, at the locations specified in $\text{normCenter}$</td>
</tr>
</tbody>
</table>

Table 2.6: Array Definition File

All dimensions used in these files are in Meters using an $XYZ$, right hand coordinate system. In a typical array used, the elements are spaced along the $x$-axis, with their elevation dimension extending along the $y$-axis, and the acoustic energy propagating along the $z$-axis. Figure 2.10 shows the diced, 3D model of a 1 MHz, 64 element DMUA with each element shown in a different color.
Figure 2.10: 3D model of 1 MHz DMUA

Figure 2.11: Close up of Dicing on the 1 MHZ DMUA

Figure 2.12 shows the dicing for a 3.5 MHz, 32 element DMUA that has a hole through the
center to allow for the insertion of a diagnostic imaging probe.

Figure 2.12: 3D model of 3.5 MHz, Fenestrated DMUA

The Propagation Operator, $H$

The propagation operator used in (1.1) defines the basis for the synthesis problem. Its elements are formed from the array element directivity values at the individual control points in the field. Specifically, the element $H(m, n)$ is the directivity of the $n$th element at the $m$th control point. Obtaining the elements of the propagation operator, in the author’s opinion, is the most critical part of synthesizing a driving pattern. Once this matrix is populated, the machinery of linear algebra can take over to create the optimal driving pattern. If this matrix is populated incorrectly, however, the synthesized driving pattern will not create the desired acoustic field.

The elements of the propagation operator can be measured or computed. Measurements could be achieved by the placement of a hydrophone at the desired location and recording the complex pressure values produced by each element, when driven by unit-amplitude, zero-phase monochromatic signal. In practice, the $H$ matrix can then be
populated directly by measuring the amplitude and phase of the received signals, e.g. using quadrature detection. Unfortunately, it is often impractical to embed hydrophones in a target medium. Time-reversal methods have been suggested for obtaining refocusing directivity values [21][9], but these methods require strong point scatterers. Several research groups (e.g. Fink, Trahey, Ebbini and co workers) are working on alternative techniques that do not require conspicuous point reflectors at the control point locations.

The current method of populating $H$ uses the array geometry to numerically evaluate the Rayleigh-Sommerfeld solution to the Huygens-Fresnel principle, (2.1), across the face of each element [12].

$$H(m, n) = \frac{1}{j\lambda} \int \int \frac{\exp(jk|\vec{r}_m - \vec{r}_n|)}{|\vec{r}_m - \vec{r}_n|} \cos(\vec{r}_m - \vec{r}_n, \vec{E}_n) \, ds$$  \hspace{1cm} (2.1)

where $\lambda$ is the wavelength, $r_m$ is the position vector of field point, $r_n$ is the position vector of element $n$, and $\vec{E}_n$ is the surface normal vector of element $n$. The integration is performed over the active area of the element assuming a baffled source.

There are a number of steps taken to simplify the evaluation of this integral. First, the integral across the entire element is broken up into a summation of integrals, where each integral now represents the Rayleigh-Sommerfeld diffraction integral across a small section of the element, $^1$

$$H(m, n) = \frac{1}{j\lambda} \sum_i \int \int_{Dicing} \frac{\exp(jk|\vec{r}_m - \vec{r}_i|)}{|\vec{r}_m - \vec{r}_i|} \cos(\vec{r}_m - \vec{r}_n, \vec{E}_n) \, ds$$  \hspace{1cm} (2.2)

If we choose the dicing to be sufficiently small, such that the field point is in the far field of the diced element (i.e. $F_l >> kW$, where $F_l$ is the focal length of the array, $k$ is the wavenumber, and $W$ is the width of the dicing), we can replace the interior integral by the Fraunhofer diffraction pattern. For a square element the phase can be represented as $\exp(jk|\vec{r}_m - \vec{r}_i|)$ and the amplitude as $\frac{sinc(\frac{x}{p_m})sinc(\frac{Wy}{p_r})}{|\vec{r}_m - \vec{r}_i|}$ where $x$ and

---

$^1$ One thing to keep in mind throughout this analysis is that the elements for most DMUAs lie on a concave sphere centered on the geometric focus, this ensures that the surface of all the dicings are tangent to a sphere of radius $F_l$ centered at the geometric focus of the array.
are the distances away from the geometric focus on a plane orthogonal to the line connecting the dicing with the geometric focus. If we place one addition constraint, $\frac{\lambda}{r_i} > 15$ and $\frac{\lambda}{r_l} > 15$, we can ensure that the sinc-sinc term remains within 2% of 1 and can safely be ignored.

As an example of the limitations this places on the synthesis process we can examine the 1 MHz DMUA detailed in Figure 2.10. For this array the focal length is 100 mm and the chosen dicing width is $\frac{\lambda}{8}$, this limits the $x, y$ excursions to around 50 mm about the geometric focus. Therefore, it is not a significant limitation in the context of imaging and therapy with this DMUA.

Utilizing the approximations just detailed, $H(m, n)$ can be calculated by

$$H(m, n) = \frac{1}{j\lambda} \sum_{i}^{numDicing} \frac{\exp(jk|\vec{r}_m - \vec{r}_i|)}{|\vec{r}_m - \vec{r}_i|} \, da$$  (2.3)

where $da$ is the area of the dicing.

To fully populate $H$, this calculation must be done for each element and for each focal spot. This is a laborious calculation but needs to be done only once for each focal point.

**Linear Algebra Routines**

Once the matrices are fully populated, the only remaining task is to carry out the linear algebra operations in (1.2). At first glance, there appear to be two linear algebra functions that need to be implemented, matrix-matrix multiplications and matrix inversion. Nevertheless, care must be taken in the matrix inversion since some of these matrices (specifically case 2 of the weighting matrix) are not full rank matrices.

In the case of inverting a matrix that is not full rank, a pseudo-inverse is employed. This inverse method uses a Singular Value Decomposition (SVD) to invert the matrices, excluding singular values which fall below a chosen threshold. In (1.2), all pseudo-inverse operations are performed on hermitian symmetric, positive semi-definite matrices, which
allows us to replace the SVD operation with an Eigenvalue Decomposition (EVD). Intel’s Math Kernel Library (MKL) was used for the linear algebra operations. MKL provides access to both Basic Linear Algebra Subroutines (BLAS) and Linear Algebra Package (LAPACK). BLAS is used to perform the matrix-matrix multiplies and the LAPACK routines are used to perform the EVD.

2.5 Summary and Conclusions

A digital DMUA driver is a multifaceted tool. There is a hardware component that must be able to create a specified wave with high fidelity, and there is a software component that must be capable of synthesizing a driving pattern.

This section explored two hardware architectures for creating the digital square wave. The first approach was a bitmap based approach that stored a digital version of the wave in memory and clocked this out at a high rate to form the needed signal. The second architecture was a state machine based approach that parameterized the waveform and used a state machine to dynamically synthesize the signal.

This section also looked at the software necessary to carry out the driving pattern synthesis. A standardized targeting and imaging definition file format was developed to help abstract away some of the refocusing details. A core refocusing software package was developed that allows for the accurate numerical evaluation of the Huygens-Fresnel integral to create and process the needed matrices.

These two components, the hardware and software, came together to create a robust DMUA driver that is capable of carrying out a wide range of experiments. The following section details two of these experiments.
Chapter 3

Results

3.1 Introduction

The realtime DMUA drivers have allowed for a large variety of experiments to take place. There have been in-vivo rat experiments, acoustic-radiation force imaging, cavitation-inception, and shearwave generation to name just a few. The following will detail two of these experiments. The first is a temperature control experiment where the DMUA was used in conjunction with a diagnostic probe to monitor and control the temperature at multiple points. The second is a lesion formation experiment where the DMUA was used to generate thermal lesions in in-vitro tissue.

3.2 Temperature Control

The system detailed above provides a flexible platform for many types of therapies, the ability to electronically steer multiple focal points allows for simultaneous treatment of large regions, and the ability to quickly update the driving patterns opens the door for real-time control. One nice application of these two features is in hyperthermia. Hyperthermia treatments require raising the temperature of a specified region a certain
number of degrees over a certain period of time, i.e. the region must follow a specified temperature profile. In order to ensure the region is truly following the required temperature profile, there must be some form of monitoring and feedback. Ideally this monitoring would be done fast, repeatedly, and non-invasively.

There are two forms of temperature monitoring capable of this sort of feedback, MRI based and ultrasound based. MRI temperature imaging is able to use a conventional scanner to track tissue property changes with temperature allowing for some control over the hyperthermia treatment. The main limitations, besides requiring an MRI, of MRI based temperature imaging are the low resolution and slow acquisition time of the images. The temperature resolution in space is dictated by the strength of the magnet and the scanning sequence used. For most setups, the spatial resolution is on the order of millimeters while the update rate is on the order of seconds. [10]

Ultrasound based temperature imaging monitors temperature by tracking heat induced speed of sound changes in the speckle pattern. [22] Liu and Ebbini have recently demonstrated a real-time system that is capable of producing temperature maps with sub-millimeter resolution at frame rates over 100 updates per second. [13] The fine temporal and spatial resolution provided by ultrasound based temperature monitoring makes it an ideal tool to monitor hyperthermia temperatures.

The following describes a multi-point temperature control experiment conducted in collaboration with Dalong Liu. The intent of this work was to demonstrate the ability of a DMUA to generate and control heat at multiple focal points with temperature feedback provided via ultrasound temperature imaging. The DMUA system allowed for accurate placement of focal points and rapid control of their temperature rise.

### 3.2.1 Setup

Figure [3.1](#) shows the setup used in this experiment. A tissue mimicking phantom was placed at the geometric focus of a 1 MHz, 64 element DMUA. A diagnostic transducer was placed orthogonal to the imaging plane of the DMUA. The imaging plane of the
diagnostic transducer intercepted the imaging plane of the DMUA at the geometric focus. All focal points were placed at the intersection of these two imaging planes allowing for registration between the two coordinate systems.

The DMUA is able to control the temperature at a focal point by varying the pressure at that location. For this experiment, the relationship between the pressure at the focal points and the monitored temperature rise was controlled via a PID controller. Each focal point was given its own controller, using a simplification that the temperature at one focal point does not influence the temperature at any others. As a general rule, this assumption is too simplistic. Two closely spaced focal points will interact if given enough time. In this experiment, however, the focal point separation was large enough, and the time duration was short enough, that this interaction was minimal and could safely be ignored.

One aspect of operation that could not be ignored was the finite acoustic power
output of the DMUA. The finite power of the DMUA means that some controller outputs ask for pressures that are unrealizable and must be clipped to achievable pressure levels. The levels at which these values are clipped can have a dramatic impact on the overall ability of the system to accurately control the temperatures.

Consider the following scenario, where a DMUA is attempting to control the temperature at two focal points simultaneously and the controller outputs have been normalized such that a value of 1.0 at each focal point corresponds to the array being driven at maximum power. If both controllers outputs are below 1.0 there is no conflict and these outputs can be realized. If both controllers ask for values over 1.0 there is clearly a conflict, and both outputs are clipped to 1.0. The interesting case is what happens when one controller output falls below 1.0 and the other rises above. The simplest method of handling this case would be to clip the high controller to 1.0 ensuring the total power output remained at achievable levels. This is, however, an inefficient solution since this results in the array outputting less than the maximal amount of power. Instead, we should find a clipping value above 1.0 that results in the array still delivering the maximum amount of power. In this way, power can be shared between focal points. For this experiment, each focal point was given an equal share of the power (i.e. for 2 focal points each focus was allocated 50% of the power, for 3 focal points each focus was given 33%, etc...), if the controller for a focal point requested less power than was allocated to it, the excess power was freed and distributed to the other foci. This is important since a treatment volume of any reasonable size is likely to be have varying degrees of perfusion and absorption causing different heating rates for different parts of the treatment volume. By dynamically shifting power to the spots that need it, the treatment volume can be heated both uniformly and as fast as possible.
Figure 3.2: Power Reallocation for Double Focus

Figure 3.2 shows how power is reallocated for a double focus pattern. The diagonal hatched region shows the area of operation in which both focal points can have their controller outputs satisfied. The vertical hatched region shows the expanded area of operation made available by power sharing. Points falling in this region are realizable due to the power shared by the first focus. The horizontally hatched region shows the area of operation made available by the power shared by the second focus. This procedure can be extended to an arbitrary number of foci.

3.2.2 Results

Figure 3.3 demonstrates how large an impact dynamic power reallocation can have on the achieved temperature profile. The green and blue dashed lines show the intended temperature profile while the solid and dot-dashed lines show the measured temperature with and without dynamic power reallocation respectively. With power reallocation enabled the control point with a higher temperature is able to quickly reach its set point once the other focal point has reached its set point. When power reallocation is
not enabled, the two focal points behave independently and the total time required for both points to reach their set points is increased.

![Graph showing temperature rise over time](image)

Figure 3.3: With and Without Dynamic Power Reallocation

The desired temperature profiles are not limited to simple patterns as shown in Figure 3.3. Figure 3.4 shows an instance in which the temperatures at each focal point were controlled independently. Figure 3.5 shows a screen shot from the thermography research platform developed by Dalong Liu. The color overlay represents the calculated temperature map at a single instance during the experiment. Very small temperature rise values have been made transparent to allow the user to see the diagnostic B-Mode image of the tissue mimicking phantom. For reference, the DMUA would be oriented out of the page with its imaging plane perpendicular to the page.
Figure 3.4: Independent Temperature Control

Figure 3.5: Temperature Map of Phantom during Experiment

The final temperature control experiment worth sharing is a single focus experiment
that was done in a in-vitro porcine kidney. The setup was the same as the previous experiments except the tissue mimicking phantom was replaced with the porcine kidney. Figure 3.6 shows both a screen shot of the measured temperature field along with actual and controlled temperature profile at the control point.
Figure 3.6: In-Vitro Porcine Kidney Temperature Control
3.3 Lesion Formation

The DMUA is capable of raising the temperature of tissue beyond the few degrees shown in the hyperthermia control experiments above. When run at high power levels, the DMUA can deposit enough energy to cause irreversible tissue damage. Figure 3.7 shows the type of tissue damage the DMUA is capable of creating. In this case, a matrix of 3x3 lesions were created with a spacing of 1 cm between each shot (the lesions are the discolored circular regions). For orientation, the lesions created here were placed in the XY plane with the DMUA oriented out of the screen in this case.

![Figure 3.7: Porcine Kidney Tissue Damage](image)

The ability to create lesions is not unique to the DMUA, any transducer of appropriate power levels can do this. What is unique to the DMUA is the ability to provide feedback during therapy. This sections describes an example of a typical lesion formation experiment, where the therapy is monitored before, during and after lesion formation. In addition to the DMUA monitoring, this experiment incorporated a thermocouple near the focus to monitor temperature rise. This is the type of experiment
made possible by the realtime drivers detailed above.

3.3.1 Setup

The tissue used in this experiment was freshly excised porcine liver. This tissue was held with a rubber holder and placed in a degassed water bath. A 3D positioning stage was used to place the piece of liver at the geometric focus of the array. SA imaging on the DMUA was used to aid in positioning the thermocouple just off the focus. Thermocouple placement is typically a very difficult process, the focal spots of most transducers are very small, so even a positioning error of a millimeter can cause wildly different temperature readings. The DMUA eliminates this problem both with the ability to image where the thermocouple is placed in the field, but also with the ability to steer the focus to the needed position.

![Figure 3.8: Synthetic Aperture Image of Pre-Therapy Setup](image)
Figure 3.8 shows an SA image of the tissue setup up. The red arrow points to the thermocouple while the blue arrow points to the surface of the tissue. The two bright spots between the thermocouple and the surface of the tissue are blood vessels.

Figure 3.9: Single Focal Point with Simulation

The realtime refocusing software and the drivers allow the user to drag and drop focal spots on the provided DMUA image. Figure 3.9 shows an example of a single focal point with the simulated therapy pattern overlaid on the image. The static image is unable to convey the ability of the user to dynamically move this focal point with a click of the mouse. With each movement, the driving patterns are recalculated and downloaded to the driver.

The user is not limited to a single focal point. Multiple focal points can be added with ease. Figure 3.10 shows a double focal point example.
3.3.2 Results

As described above, the DMUA driver allows for precise interleaving of therapy with imaging. Figure 3.11 shows the interleaving sequenced used for this experiment. Before the experiment, SA images were taken to get a high resolution image of the medium. During therapy, STF imaging pulses were interleaved with the therapy pulses to allow for near instant analysis of each short therapy burst. This monitoring allows for the system to monitor if something has gone wrong (e.g. the patient moved, damage was created at an unintended location, etc...) and stop therapy if the need arises.
For this experiment, the focus was placed 100mm axially and 0mm laterally in front of the array. The thermocouple was positioned slightly proximal and to the side of the focus to minimize its effect on the treatment but still allow it to record the temperature on the periphery of the focal zone. Figure 3.12 shows the location of the focus, red dot, with relation to the thermocouple.

Figure 3.12: Location of Focus During Therapy

Therapy commenced after 4 seconds of baseline readings. Figure 3.13 shows an STF image taken 1.5 seconds into therapy. The thermocouple readings are displayed below the STF image and indicate a nice exponential temperature rise at this point in the experiment.
Figure 3.13: Single Transmit Focus Image 1.5 Seconds into Therapy

Approximately 3 seconds into therapy there is a dramatic increase in the heating rate and an associated change in the echogenicity at the focus of the STF image. Figure 3.14 shows the dramatic change in the image at the focus along with the measured temperature rise. This sort of echogenicity change is commonly seen when the tissue at the focus begins boiling. This is not to say that all tissue damage results in an echogenic change. It has been observed that even without echogenic changes in the image, tissue damage can still occur. Work is currently underway to develop methods that can detect...
tissue damage without waiting until there is an echogenic change in the tissue.

It can't be observed on these still images, but the STF image time series show the echogenic region grow towards the transducer with time. This has the effect of shielding the thermocouple from additional energy. This can be seen in Figure 3.14, where after the tissue reaches boiling, the temperature actually begins to decrease even though therapy is still underway.

![Figure 3.14: Single Transmit Focus Image Post Boiling](image)

Immediately after therapy is finished, the driver switched back to SA imaging to
assess the damage. Figure 3.15 shows the SA image captured 1 second after therapy was stopped. This soon after the formation of the lesion, the echogenic spot created is quite bright. The implementation of this realtime driver has led to the ability to watch the echogenicity of this spot fade with time. This was something that hadn’t been observed with the previous driver which required significant amount of time to collect and process images.

Figure 3.15: Synthetic Aperture Image Post Therapy
DMUAs have the potential to create a new paradigm in ultrasound therapies. They are able to leverage the benefits of both imaging and therapy to provide a safe platform for delivering ultrasound therapies. This thesis described the design and implementation of one part of an overall DMUA system: the driver.

The task of the driver is to create a waveform for each element of the array that will focus the energy in a desired manner. The method chosen in this thesis was that of a digital driver. A digital driver creates a time shifted, pulse width modulated square wave that can be filtered to produce a sinusoid with the desired phase and amplitude. Two separate techniques were investigated for forming the required square wave, a bitmap based and a state machine based approach. The bitmap based approach stored a complete image of the wave in memory and synthesized the wave by recalling this pattern from memory. The state machine approach parameterized the wave and used a finite state machine to dynamically synthesize the wave. The state machine approach allowed for decreased data storage and the generation of more flexible phase unwrapped waveforms. The decreased storage requirements of the state machine approached permitted the creation of an onboard microsequencer. The microsequencer stored hundreds of
unique driving patterns and orchestrated their presentation to the array with microsecond timing, allowing for the synthesis of complicated, amplitude modulated waveforms.

As a complement to the hardware, a software component was also developed to quickly synthesize and download new driving patterns. The software consisted of two separate parts, a text based targeting scheme and a low level synthesis package. The text based targeting scheme uses human readable, text based descriptions of a therapy profile. This allows for the definition of simple, single focus targets, or more complicated multiple focus patterns with time-varying amplitude modulation. This text based description was used as the input to the second part of the software that actually created an array driving pattern. In the synthesis package, an accurate model of the DMUA was used to calculate the half-space element directivity values as entries of the propagation operator matrix. A linear algebra package was used for solving the ensuing optimization problem(s) based on the definition of the control points and critical points in the target region.

The real-time implementation of the resynthesis approach was enabled by the use of high performance computing and advanced algebra packages that have become more widely available in recent years. While the implementation described in this thesis, to the best of our knowledge, is one of the first of its kind, these tools will become more widely used by most research groups in both academia and industry. This will undoubtedly propel the research and development of new therapy and imaging schemes in the future. In fact, the combination of the large-aperture arrays typically used as DMUAs and more accurate, computationally-based beamforming will enable more reconstructive ultrasound imaging going forward. This will allow for new array design approaches that employ larger, more directive array elements than is currently mandated by conventional beamforming methods.

Finally, the thesis research work described herein was and continues to be much more than just a hardware and software development. The seamless transitions between imaging and therapy enabled by the developments described in this thesis will open the door for fundamental understanding of the dynamics of lesion formation in IgHIFU
(Image-guided HIFU). Numerous researchers speculated on the exact nature of tissue damage due to HIFU shots, especially the degree of involvement of cavitation. The ability to continuously monitor the echo data from the exact treatment spot, enabled by the inherent registration between the imaging and therapy coordinates, is a key feature of DMUA technology. However, one can state with confidence that the real-time aspect of data acquisition and field control of DMUAs is the feature that will finally bring the full benefits of DMUAs in providing safe and efficacious HIFU treatments, even in complex mobile organs like the liver. This will provide options to patient groups currently ineligible for resectional surgery with few other options available.
References


