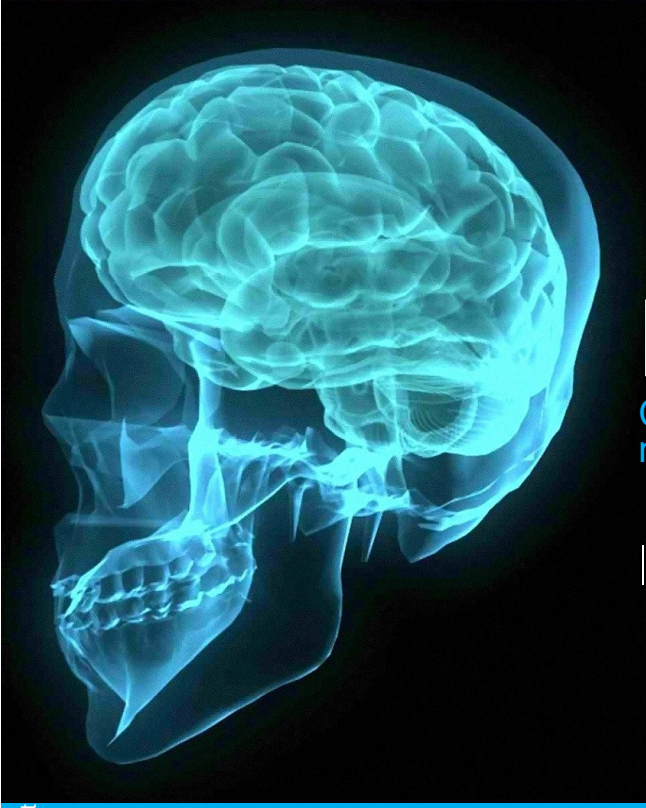


Neuro-Stent

CMOS driving channel for ultrasonic neurovascular stent

Ishaan Ghosh

Technische Universiteit Delft

TU Delft

Neuro-Stent

CMOS driving channel for ultrasonic neurovascular stent

by

Ishaan Ghosh

to obtain the degree of Master of Science
at the Delft University of Technology,
to be defended publicly on Wednesday August 25, 2021 at 9:00 AM.

Student number:	5023939	
Project duration:	September 1, 2020 – August 25, 2021	
Thesis committee:	Prof. dr. ir. W.A Serdijn,	TU Delft
	Dr. T. Costa,	TU Delft, supervisor
	Dr. ir. M.A.P Pertijs,	TU Delft

TU Delft

If you're going to try, go all the way
Otherwise, don't even start.

Charles Bukowski

ABSTRACT

Ultrasound stimulation is slowly emerging as an effective treatment to alleviate multiple mental illnesses. The advantage of ultrasound stimulation stems from the ability to stimulate regions that are spatially far off from the stimulating device without sacrificing spacial resolution. This directly implies the ability to devise minimally invasive stimulators that can employ beamforming to focus and steer ultrasound waves to different points in the brain. This is in stark contrast to electrical stimulation techniques that rely on invasive methods of inserting electrodes very close to the site of stimulation to obtain the desired resolution and selectivity. This work involves the preliminary system design of a 1-D IC array for beamforming. This novel low area and power IC can be integrated with a neuro-vascular stent and using minimal surgery, inserted in a blood vessel. Once placed, it can independantly send out focussed ultrasound waves without losing any resolution due to the skull low-pass characteristics. The final IC channel devised consumes an area of 143*52 μm^2 and consists of the beamforming circuitry, level shifter, the high voltage driver and the digital control block. The chip was fabricated using TSMC 180 nm BCD technology and measured using a PCB and FPGA (for the control signals). The final measured results match the expected results from post layout simulations.

ACKNOWLEDGEMENT

The past one year has been a whirlwind to say the least. With countless failures and setbacks, coupled with a raging pandemic, the year has been a tough test in resilience. At the end, as I come out satisfied with the person I have become, it is a must for me to acknowledge what kept me going in the darkest of times. Firstly the dream that my work could one day help mankind in a small way kept me motivated to keep pushing through the pain. I strongly feel ultrasound stimulation could one day replace alot of treatment methods available today and pave the way for a new class of bioelectronic medicine.

It is also my duty to thank the people to whom I owe everything. Firstly, credit goes to my supervisor Tiago for always patiently clarifying all my some what mundane queries about every tiny issue in the project. Tiago always kept a cheerful atmosphere, even during the toughest phases and his structured approach to organising my thesis had a profound impact in ensuring a smooth workflow. Secondly I would like to thank Hassan for always being there to help me out in every single step of the way. From setting up Cadence to teaching me about device layout, I appreciate the many hours Hassan has spent on patiently trying to help me along the way. Thirdly, the group members of the ultrasound group were a source of great learning for me. The group meetings were a place for me to learn about so many different aspects to a project and I personally appreciate the help that Gandhi, Ignas, Berend, Christiaan, Francesc and Xinyu sent my way. A special thanks is also due for Paul and Massimo for the constant help and support in conducting the very eye opening Honours Project as part of the Honours Programme with the ECTM group.

A big thanks to the constant support and faith my family placed in me through the past two years. I could not have done it without my 24/7 family support system. I would also like to thank my friends both in India and here in Delft for being there to cheer me up when things weren't going well for me. I usually don't appreciate my brethren enough so take this as a formal acknowledgement of your efforts. Lastly I should also thank myself for going through all that I have, as I'm pretty sure I couldn't have done it without me.

Ishaan Ghosh
Delft, August 2021

CONTENTS

1

INTRODUCTION

This thesis is an attempt to design a novel integrated circuit that can be used as part of a one-dimensional array for brain stimulation using ultrasound beamforming. The major physical constraints involved making the chip using very low area while also consuming low power. Due to the multidisciplinary nature of the work, multiple aspects of the design process are considered one by one, in the subsequent sections. The current chapter shall be a brief introduction to the current brain stimulation techniques and their inherent drawbacks. Possible solutions are then discussed to address the aforementioned drawbacks and finally, our proposed system is explained.

1.1. BRAIN STIMULATION

TREATING mental illnesses have been a source of interest for a long duration of time. Historic records from 7000 years ago show primitive attempts at brain stimulation to cure mental issues ranging from simple headaches to more severe illnesses. As technology advanced, research into the brain anatomy started getting more refined. The knowledge of mental disorders and their potential causes became a subject of great interest. A unanimous conclusion that was drawn over the years, was the fact that mental illnesses stemmed from dysfunctional neuronal activity in certain specific regions of the brain. Researchers could also extrapolate and go on to say that by manipulating the neuronal activity in those particular regions, mental disorders could be suppressed and even potentially treated. This line of thought led to a surge in brain research and potential ways to stimulate different areas of the brain were looked into in-depth. Over the past 50 years, a few methods of brain stimulation came to prominence due to their proven ability in alleviating the symptoms of mental disorders.

1.2. EXISTING STIMULATION METHODS

Currently, a handful of brain stimulation techniques are medically approved for treatment purposes. These methods are briefly discussed in the following section.

1

1.2.1. DEEP BRAIN STIMULATION

Electric deep brain stimulation (EDBS) has been used to alleviate the symptoms of many brain diseases[1]. DBS has been used as an effective treatment strategy for a number of years and has been clinically approved for Parkinson's [2], essential tremor [3], dystonia[4], obsessive-compulsive disorder[5], and epilepsy[6]. In recent times, it is also being studied as a potential treatment solution for other diseases, such as depression[7], multiple sclerosis[8], Huntington's [9] and Tourette's [10].

EDBS devices are usually composed of electrode leads and a stimulator. The electrode leads are usually thin and long needle-like structures that are inserted deep into the brain. The stimulator generates current or voltage pulses with a patient-specific dosage (amplitude and frequency). The stimulator is also often implanted inside the patient's body to minimize the chances of infection. Fig. 1 shows the electrode leads inserted deep into the brain for treatment.

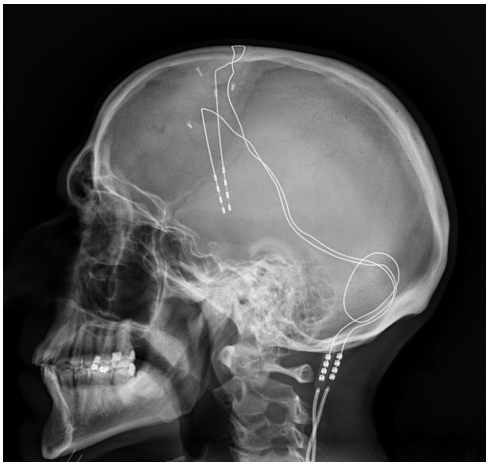


Figure 1: X-ray image of DBS device

By carefully choosing the electrical dosage needed, the ionic activity of faulty neurons at the stimulation site is altered which results in a substantial mitigation of symptoms and subsequent improvement in the quality of life for the patients.

One major advantage of DBS lies in the fact that the stimulation only occurs at the electrode location, thus making it a highly selective treatment approach. This is quite advantageous as there is no effect on neurons located outside the electrode contact region. This is in stark contrast to pharmaceutical treatment options where the drug is delivered throughout the body using the circulatory system and will undoubtedly have unwanted side effects.

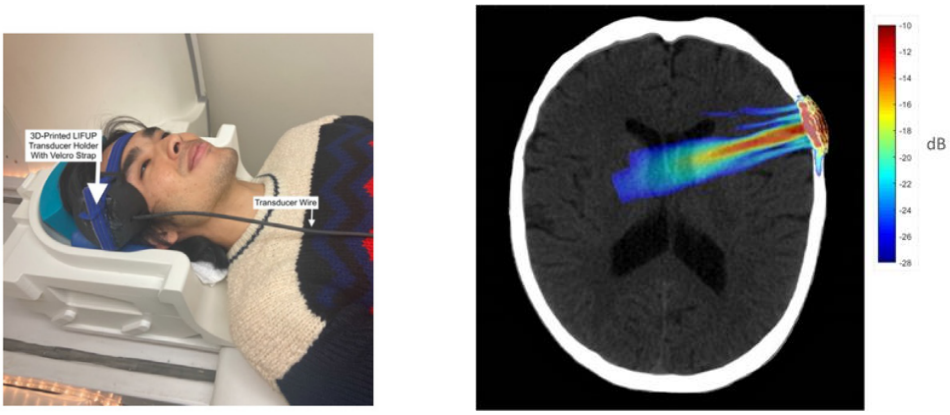
However, EDBS has some major disadvantages to address. The biggest issue arises from its major strength: stimulation only occurs at the electrode location, which means that proper insertion is extremely important to be very close to the region that needs to be stimulated. Neurons outside the electrode active area cannot be targeted easily without re-insertion, which is a risky procedure and includes chances of infection. DBS also suffers from being a very invasive procedure requiring anesthesia and also has risks

of lead fracture and stimulator malfunction. Only 10-15 % of patients are approved for DBS treatment in the first place [2], implying the need for a safer and more robust form of treatment.

1.2.2. TRANSCRANIAL LOW INTENSITY FOCUSSED ULTRASOUND STIMULATION

In order to resolve the drawbacks of DBS, ultrasound as a tool for brain stimulation was looked into. Research showed that ultrasound could be used to excite or inhibit the neurons in the brain. The energy possessed in ultrasound waves was capable of modulating the neuronal activity in the target neurons. This approach has been demonstrated (*in vivo*) in humans to stimulate the somatosensory, visual, primary motor cortices, and thalamus[11].

Transcranial stimulation already existed in the form of Transcranial electrical stimulation and Transcranial magnetic stimulation, however, using ultrasound provided better spatial resolution and depth of penetration than using electrical/magnetic stimulation. Transcranial ultrasound stimulation meant placing an ultrasound generator on the surface of the skull as shown in fig. 2 [12]



(a) Device Placement

(b) ultrasound focussing at the stimulation site

Figure 2: TLIFU stimulation

This had the big advantage of not requiring any invasive surgery, making it much simpler to implement and had little to no associated risks.

However, Transcranial Low Intensity Ultrasound Stimulation (tLIFU) however suffers from one major drawback. just as in the case of EDBS, the major drawback of TLIFU comes from its major strength: its non-invasiveness requires the ultrasound emitter to face a bone barrier before reaching the brain tissue. The skull exhibits a low pass characteristic wherein higher frequency waves are absorbed strongly as compared to low frequency wavelengths. This restricts the usage of high-frequency ultrasound and instead forces clinicians to use ultrasound in the hundred-kilohertz range. This leads to a very poor spatial resolution at the site of stimulation usually in the millimetre range for lateral resolution and centimetre range for axial resolution [13]). This falls short of DBS

treatment where the spatial resolution is as low as tens of micrometres[14]. So, despite overcoming some of the major drawbacks, TLIFU lacks high resolving power due to the trans-cranial nature of stimulation.

1.2.3. NEUROVASCULAR STENT BASED ELECTRODE STIMULATION

Another upcoming paradigm of stimulation tries to utilize the advantages of the prior two stimulation methods. In this method, a neurovascular stent is inserted into a blood vessel running through the brain. The stent is integrated with recording and stimulating electrodes that can affect the neurons in the vicinity of it. This method is minimally invasive as stent insertion has been carried out for many years[15] and is a simpler procedure dealing with inserting a catheter-guided stent through a blood vessel in the groin and reaching the target blood vessel in the brain. This is a fairly low-risk operation and has been used extensively for treating brain aneurysms and intracranial stenosis [16]. This particular stimulation technique was first introduced by the company Stentrode™ for brain EEG recording[17] and subsequently for stimulation as well[18]. It went on to be medically approved for human use and is currently being researched in-depth for other potential applications[19]. As visible in fig. 3, the stent is attached to a catheter (green arrow) has small 750 μm electrode discs for stimulation and recording (yellow arrow). [20].

Figure 3: Neurovascular Stent showing the catheter and electrode discs

This process being minimally invasive, may allow for easier acceptability for a larger patient base as compared to deep brain stimulation for which only 3-5% of patients are ever eligible for implantation. However, like DBS, the electrode location is crucial for proper treatment. Only the neurons at the vicinity of the electrodes can be recorded and stimulated. There is very little opportunity to re-insert the stent, in case the correct stimulation region is not in the vicinity of the electrode discs. The blood vessel diameter is also a constraint. The stent cannot be integrated into narrow blood vessels and clinicians usually use thicker vessels like the Superior Sagittal Sinus (SSS) and Central Sulcan Vein (CSV) that have diameters greater than 2.5 mm at certain places[21]. Since the depth of penetration of electrode based stimulation is nearly non-existent, only regions of the brain close to these wider blood vessels can be targeted. This is a severe drawback of electrode-based neurovascular stents that needs to be addressed.

1.3. PROPOSED SOLUTION

From the above described methods on brain stimulation, we see that there there are some special requirements for an effective neuro-stimulator. These can be described as

follows:

- Minimally Invasive in nature
- High spatial resolution
- Ability to stimulate different regions at different depths.

The DBS system offers very high spatial resolution, but comes with no freedom in stimulation location while also being a very invasive procedure. tLIFUS allows for a non invasive procedure and also allows stimulating different sites, however it comes with a much lower spatial resolution. The Strentrode neurovascular stent approach allows for a minimally invasive procedure with a high spatial resolution but does not allow stimulating at different depths/ angles. The three methods are summarised in the table below:

	Invasiveness	Resolution	Variable stimulation site
Deep Brain Stimulation	×	✓	×
Transcranial LIFU	✓	×	✓
Neurovascular Electrode Stent	✓	✓	×

Table 1: Trade-off table

1.3.1. NEUROVASCULAR LOW INTENSITY FOCUSSED ULTRASOUND STIMULATION

From the table, we can say that each method has certain advantages while also having some limitations. We can utilize the best qualities of each of the above methods and create a new stimulation modality that overcomes the individual drawbacks of the systems mentioned above.

We thus propose a new form of brain stimulation that uses a neurovascular stent for insertion, while also stimulating using ultrasound waves. This 'Neuro-Stent' would be minimally invasive and allow very close proximity to stimulation sites. By using ultrasound instead of electrodes, we also ensure variable stimulation sites to be possible. More importantly, since the stent is inserted into a blood vessel in the brain, it bypasses the skull low pass issue. This directly permits the usage of high-frequency ultrasound, (within practical limits) and thus allows for higher spatial resolution. Thus the neurovascular Low Intensity Focused Ultrasound (nLIFU) stimulation modality overcomes the drawbacks of the prior methods, while preserving all of their advantages.

1.3.2. DESIGN PLAN

In order to implement such a modality. there are multiple things to consider, from an integration point of view. Similar to the Stentrode design of fig. 3, the proposed design involves a 1-D array of channels that will be implanted along the wall of the stent. The 1-D array will then be able to generate focused ultrasound beams into the area surrounding the blood vessel. This principle is illustrated in fig. 4.
To realise this structure, the array comprising of individual channels is first looked into.

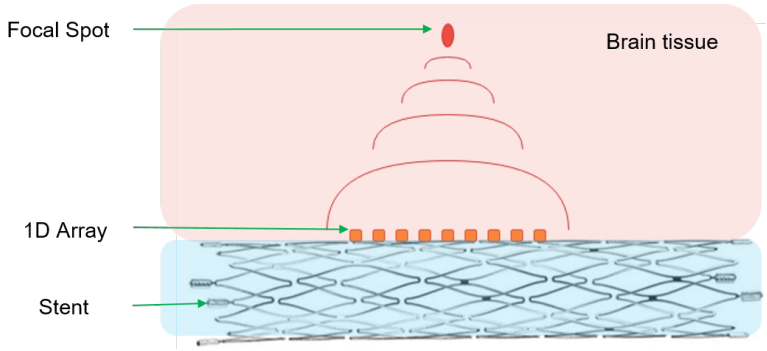


Figure 4: Neurovascular Low Intensity Focused Ultrasound

Each individual channel comprises of the CMOS IC that is used to generate the required phase-shifted pulses. The piezoelectric material is also integrated directly on top of the aforementioned IC. This reduces interference and parasitic capacitances from long wires while also saving considerable area. The integrated channel is then placed on a flexible substrate. The flexible substrate will also need to support the metal lines to power the channels and send the appropriate control signals. The flexible substrate along with the array is then attached to the stent frame using some form of adhesive. This is shown in fig 5 (side-view).

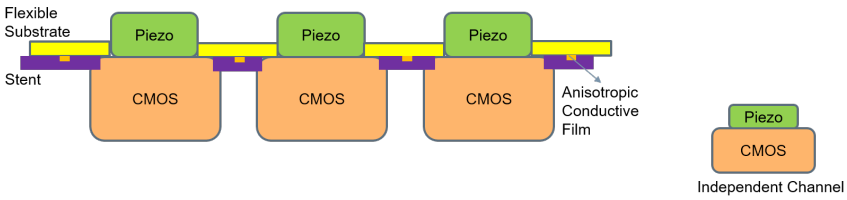


Figure 5: Expected device design

From the design, there are certain major constraints that need to be considered.

1. The blood vessel where the stent is placed has a limited diameter. Even the most geometrically appropriate vessels in the human brain like the SSS, has a diameter in the range of 2-3 mm[21]. This directly places a constraint in terms of the length of the CMOS chips, that form the structural and functional building block of the channel.

2. Since the array is being implanted into the brain for long-term usage, there must be no toxicity or harm to the surrounding tissue. This puts a constraint on the materials chosen for the channel. It also means the final device will need to have some form of encapsulation with an epoxy for example.

3. The channel will be mounted on a flexible substrate. This flexible substrate will also have multiple metal interconnects running through it, to supply the neces-

sary signals to each channel. The more interconnects there are, the more crowded and harder it is to route them. The chances of accidental breaks in the metal interconnects also increase. Thus it is quite imperative to ensure that the minimum number of lines are used for each channel.

The main focus of the subsequent work will be on designing the CMOS circuit aspect of the individual channel, while also keeping in mind, the general device constraints that need to be satisfied.

2

ULTRASOUND BEAMFORMING

In recent times, ultrasound is being used, more and more frequently for neuronal stimulation, as compared to the traditional applications in non-destructive imaging and tissue ablation. To stimulate the neurons, the ultrasound waves need to be focussed and steered to the target region and generate enough pressure for the neuron ion channels to open or close. This is done using ultrasound beamforming which is the main aspect to be covered in the subsequent section. The mechanisms of ultrasound generation and some major considerations in beamforming are looked into in depth.

2.1. ULTRASOUND GENERATION

ULTRASOUND is usually generated by sending a time-varying voltage to a piezoelectric transducer. On being fed with a varying voltage, the piezoelectric material starts vibrating (ideally at it's resonant frequency), and this vibration generates ultrasound waves that propagate into the medium.

2.1.1. WAVE PROPAGATION

Ultrasound refers to sound waves that are above the audible range of hearing, thus extending from 20 kHz to several gigahertz. Like all traveling waves, ultrasound waves propagate as spherical waves and can be mathematically expressed by the following equation:

$$y(x,t) = A * sin(kx - wt + \phi) \tag{2.1}$$

A refers to the maximum wave amplitude
k refers to the angular wave number
w refers to the angular frequency.
This equation is a compact description of the wave in terms of its horizontal position x and time t. ϕ refers to the phase offset of the wave and is one of the most important parameters for the subsequent discussion.

When there are multiple waves being generated simultaneously from different sources, they experience constructive and destructive interference from the other wavelets at a

given location. Constructive interference has a summing effect on the individual wave amplitudes and contributes to a maxima at that point while destructive interference tends to cancel each other out, leading to a minima. By controlling the location of constructive interference, we are able to create a region of high amplitude or high sound pressure. This region is then referred to as a focal spot and is the central theme of ultrasound beamforming.

2.1.2. PHASED ARRAYS

Ultrasound can be generated from piezoelectric transducers and naturally form a beam in the direction perpendicular to the plane of the transducer as shown below [22]:

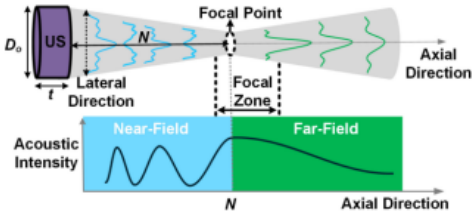


Figure 6: Natural Focus

The beam tends to focus at a point referred to as the natural focus. The depth and width of this focal spot depend on the geometric properties of the piezoelectric transducer source and is given by the equation as follows:

$$N_f = \frac{kL^2}{4\lambda} \tag{2.2}$$

Where k is the aspect ratio constant
L is the length/aperture of the element
λ is the wavelength of the ultrasound generated

2.2. PHASED ARRAY BEAMFORMING

By using a single element, we can get converged beams of ultrasound at the natural focus and by varying the aperture and frequency of propagation, we can generate the desired sound pressure level at the focal spot. However, one major drawback of this approach is the fact that the natural focus position cannot be altered during operation as it only depends on the device geometry and frequency of operation. There is no scope of changing the location of the focus apart from manually moving the transducer itself. This is bound to lead to complicated motor stages and a loss in spatial resolution. To resolve this issue, phased arrays have been used as a replacement for single-element transducers.

A phased array comprises of multiple transducers placed one after the other in a one-dimensional or two-dimensional array. Each transducer generates ultrasound waves independant of each other. However, unlike a single transducer, we are able to utilise the phase aspect of wave propagation to change the focal spot location. By setting appropriate phase delays to individual elements of the array, we can make the wavefronts arrive

simultaneously to a single point, thus generating constructive interference and subsequently creating a high intensity focal spot. This spot can be now moved to any position we want. within physical limits. The idea behind phased arrays is elaborated in fig 7

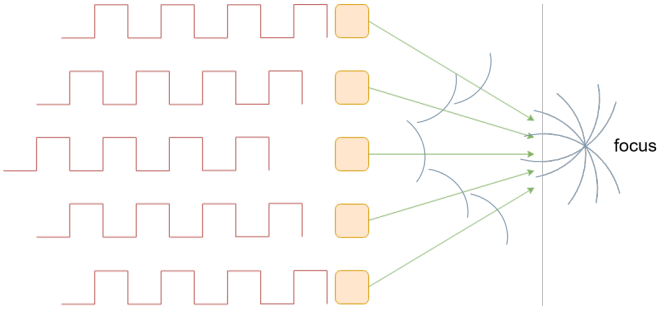


Figure 7: Phased Array Concept

2.2.1. WAVE PROPAGATION EQUATIONS

To utilize this powerful property of phased arrays, we need to estimate the exact phase delays that need to be provided to the voltage signals that subsequently excite the piezoelectric transducer. This is done by making use of the basic wave propagation equations. To illustrate the idea behind the equation derivations, we use a simple 1-D array of transducers as shown in figure 8.

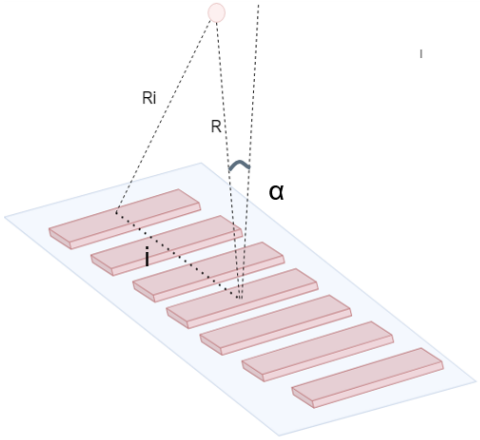


Figure 8: 1D Array model

Here we see an array of 7 elements with the focal spot at an angle α away from the midline that runs through the central element of the array and is perpendicular to the array

plane. Since all waves obey the distance speed relation, for multiple wavefronts to arrive at the focal spot simultaneously, there must be a difference in time of flight. The distance to the focal spot (R) varies for each element while the velocity of all the waves are the same in the medium. This implies that there must be a difference in the time of flight introduced to create constructive interference at the focal spot. This is done by introducing small time delays Δt. This is shown in the following equations:

$$R_i = c\,t_1$$

$$R = c\,t_0$$

Thus

$$\Delta t = \frac{R - R_i}{c} \tag{2.3}$$

Where R_i is the distance from the i^{th} element to the focal spot
R is the distance of the central element to the focal spot
Δt is the time delay that needs to be provided between the i^{th} wavefront and the mid-element wavefront.
This equation can be written in terms of the angle α by representing the distances in terms of trigonometric functions as shown below:

$$\Delta t = \frac{R - R_i}{c} = \frac{R}{c}[1 - \frac{R_i}{R}] = \frac{R}{c}[1 - \sqrt{(sin\alpha - \frac{i}{R})^2 + cos\alpha^2}] \tag{2.4}$$

Where i is the distance of the i^{th} element from the central element.

Using this equation, we can theoretically beamform ultrasound waves to any position in the given plane by knowing the distance R from the central element, and the steering angle α. However, as we shall see in the subsequent section, beamforming using phased arrays has some unwanted effect associated with it and needs to be taken into consideration while sizing the array dimensions.

2.3. ARRAY ELEMENT SIZING

While beamforming is a very powerful and convenient method to add a lot of flexibility to an ultrasound stimulator, by using phased arrays we need to take into consideration some factors that are important in such applications.

2.3.1. MAIN LOBE WIDTH OPTIMISATION

When a focal spot is generated, the normalised radiation pattern generates a main lobe along with multiple sidelobes in the region around it [23], as shown in figure 9: The beam width is usually defined as the distance between the points on the main lobe where the power has dropped by $\sqrt{2}$ times its maximum value (-3 dB), also known as the Full Width at Half Maximum (FWHM). Since we want a very high selectivity in terms of the neurons being excited, the FWHM should be as small as possible.

It has been seen that the beam diameter strongly depends on the array geometry [24], namely the total number of elements N, the inter-element spacing d, and the wavelength λ. Usually, the beam diameter is related to a sharpness index q where a lower sharpness

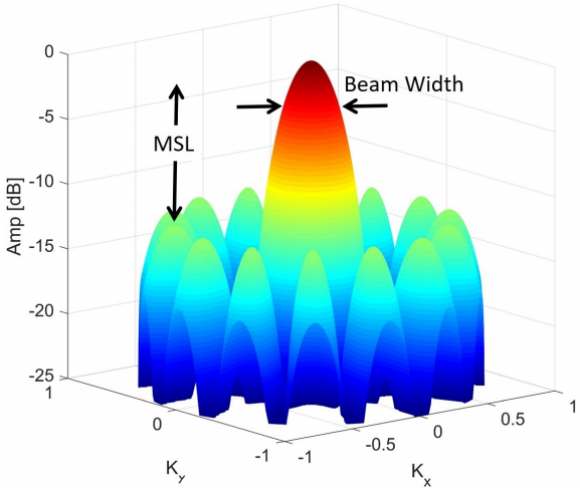
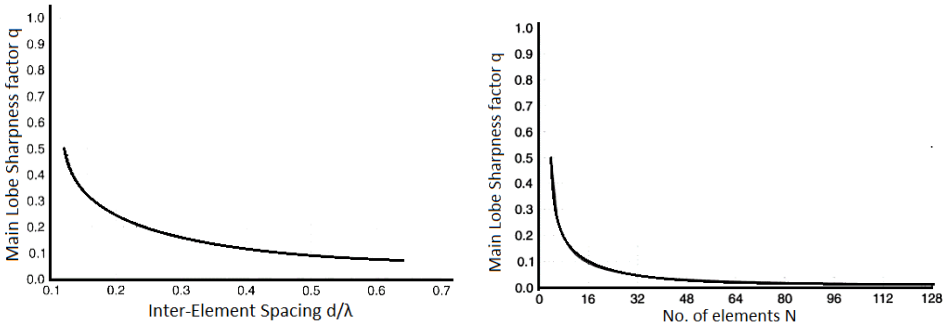


Figure 9: Main lobe with corresponding side lobes

index means a smaller beam diameter. From the following plots of fig 10, we see that the sharpness factor reduces as $\frac{d}{\lambda}$ increases and also rapidly falls off as N increases till roughly 32 elements, after which it stays roughly the same [24].



(a) d, λ relation

(b) N relation

Figure 10: Beam Diameter geometry relation adapted from [24]

2.3.2. SIDELOBE SUPPRESSION

In beamforming applications, sidelobes can be considered as unwanted effects that do not contribute to focal spot stimulation, and are instead a source of power loss of the system. It is thus necessary to reduce the effect of sidelobes.
It can be seen that the side lobe amplitude falls sharply as the number of elements increases [25]. This can be observed in fig 11

It is thus more advantageous to increase the number of elements to reduce the effect of sidelobes.

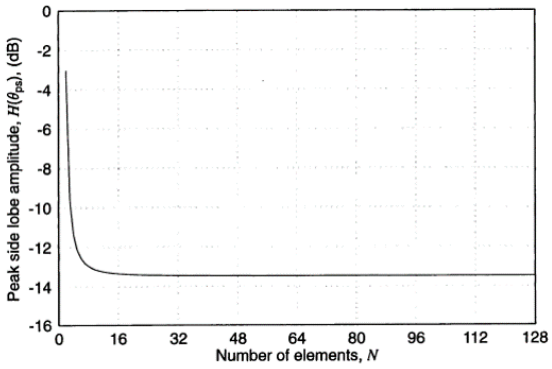


Figure 11: Side lobe suppression

2.3.3. GRATING LOBE ELIMINATION

When using phased arrays, there are many advantages over single-piece ultrasound transducers. However, one major inherent issue of phased arrays that arises, is the generation of grating lobes. Grating lobes are secondary main lobes or very strong side lobes that are similar in amplitude to the main lobe and can be seen distributed around the phase map [24]. These are especially troublesome for stimulation as it creates an unwanted focal spot away of the desired focal spot. This would lead to undesired stimulation of brain regions that could have unpredictable consequences. The distribution of the lobes is shown explicitly in fig 12

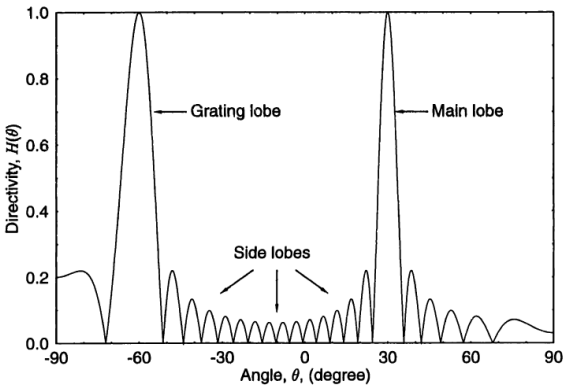


Figure 12: Grating lobes

Grating lobes are a consequence of individual array elements being spaced out too far from each other. By reducing the inter-element spacing 'd', the grating lobes can be minimized up to practical feasibility limits. The grating lobes can be eliminated by keeping the spacing d less than an upper limit d_{max} . From fig. 13, We see that grating lobes can be removed even for high steering angles by always keeping $\frac{d_{max}}{\lambda} \leq 0.5$. This means

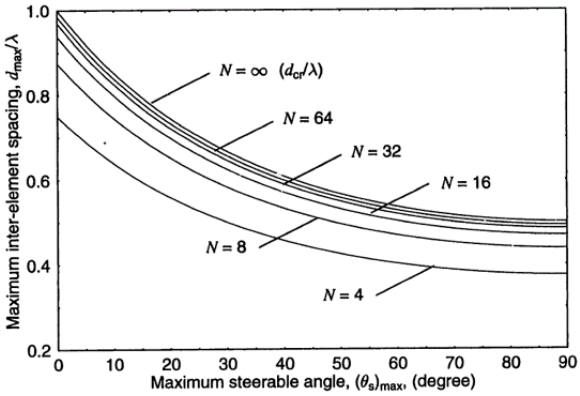


Figure 13: Grating lobe elimination

that keeping $d \leq \frac{\lambda}{2}$ would lead to the elimination of grating lobes for high steering angles θ and any number of total elements N.

Thus from analyzing the above three constraints, we can conclude that:

- To reduce beam diameter, we need higher inter-element spacing $\frac{d}{\lambda}$ and a large number of elements N.
- To suppress sidelobes, we need a larger number of elements N, at least more than 16.
- To eliminate grating lobes, we need the inter-element spacing to be low, with $d \leq \frac{\lambda}{2}$

3

TRANSMIT BEAMFORMING CIRCUIT ARCHITECTURES

In recent years, there has been a rising interest in ultrasound transmit beamforming for stimulation and imaging. This has given rise to several commonly used circuit architectures that can generate multiple phase delays to beamform and generate a focal spot. Some of the most common architectures are discussed in the chapter.

3.1. DELAY LOCKED LOOP

THESE are one of the most commonly used blocks to generate very precise phase delays. Most ultrasound stimulators usually have the beamforming core comprising of a DLL. Often they are used in tandem with phase interpolators in a coarse-fine phase delay fashion. The DLL uses a negative feedback loop topology to ensure minimal variation from the expected delays as illustrated in fig 14.

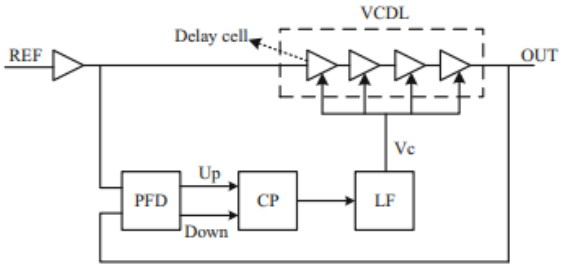
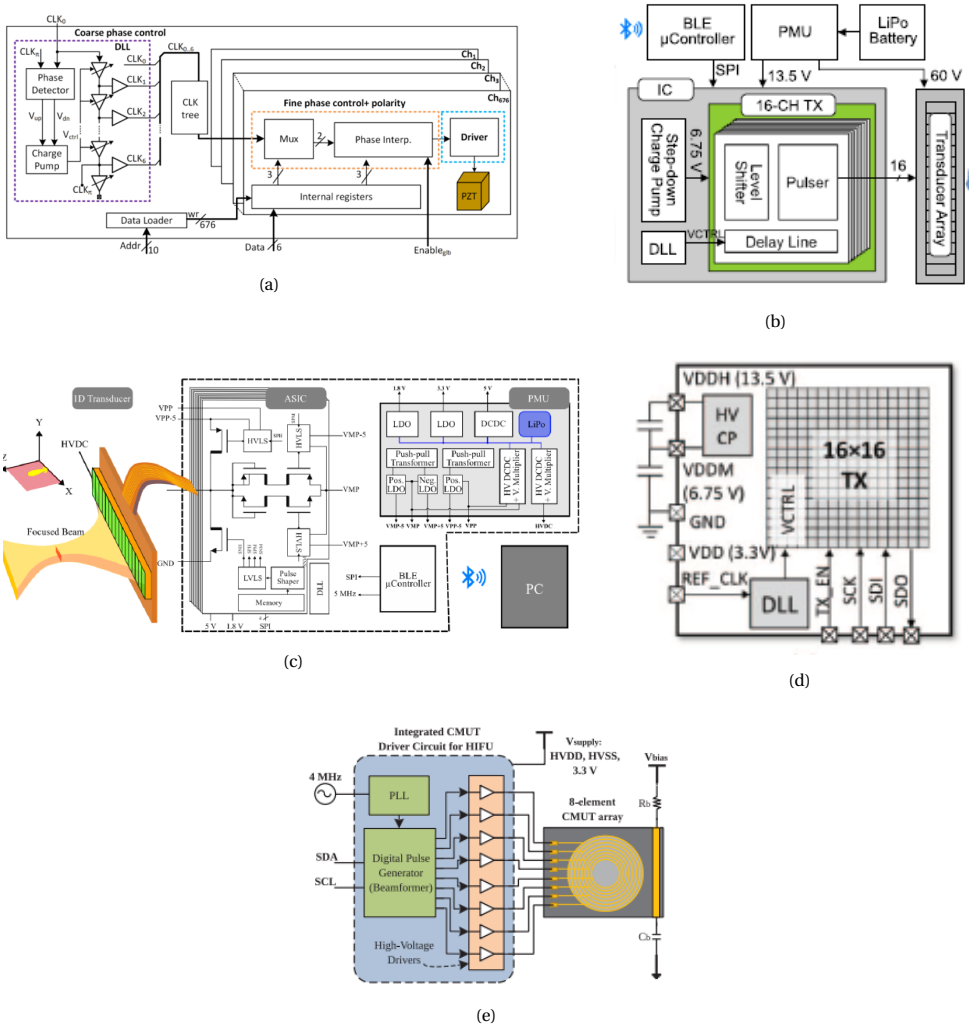


Figure 14: DLL Block Diagram

It consists of a phase detector(PD) to compare the reference clock with a 360° phase-shifted clock from the feedback loop. This is followed by a charge pump (CP) which is used to convert the phase difference of the prior clocks into a voltage signal. Finally the

voltage signal is sent to a voltage-controlled delay line (VCDL) wherein the delay of the elements is adjusted. The final output of the VCDL is then fed back to the phase detector. By this, very precise phases can be generated by tapping different points of the VCDL. DLLs are one of the best choices for generating stable process invariant delays and are used in most ultrasound beamformers as shown below in fig 15.



(a) (b) (c) (d) (e)

Figure 15: (a)DLL used as coarse phase generator [26] (b)DLL used for 1D array beamforming [27] (c)DLL used off-chip for beamforming [28] (d)DLL used for 2D array beamforming [29] (e)PLL used for beamforming [30]

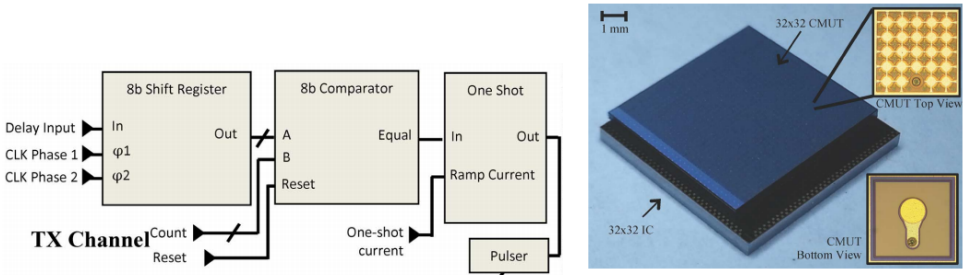
In fig 15a we see the DLL being used in a coarse-fine delay arrangement along with a phase interpolator. The DLL provides 6 coarse delays, while the interpolator generates the phases between subsequent coarse delays using a phase mixing strategy. In fig 15b

the DLL is used to directly drive a level shifter that interfaces with the pulser and finally the 16 channel array. In fig 15c, we see the DLL again being used to drive a pulser with a pulse shaping functionality. It should be noted that the DLL is not kept on the chip but is kept outside the channels. The delays generated from the DLL are then sent to each of the channels for further processing. Fig 15d uses a DLL for a 16*16 2D array while Fig 15e uses a PLL to drive an 8 channel annular CMUT array. These DLLs are then usually followed by a level shifter and/or a driver circuit that then interfaces directly with the piezoelectric material, in order to generate the ultrasound waves.

However, despite their popularity, they have two severe limitations. Firstly, they consume a large area, usually in the 0.1-0.5 mm^2 range. This is due to the long delay line and the large integrating capacitor at the charge pump output. The second is the fact that DLLs are power-hungry blocks owing the large number of components and usually consumes power in the hundreds of microwatts to even a few milliwatts. These make them hard to implement in area/power constrained problems.

3.2. SHIFT REGISTER WITH CLOCKED COUNTER

In this beamforming approach, a digital address-based delay formulation is used. Before transmitting, the shift register is loaded with a particular delay value. Once all the channels are loaded with the correct delays, an external counter is enabled and starts counting from 1 till the maximum possible delay [31]. When the counter value matches the delay stored in the shift register, the comparator would output a high. The comparator uses dynamic logic and the output high triggers the one-shot circuitry, which then drives the high voltage pulser. The system is made to generate unipolar pulses, but by adding some elements, it can be used for continuous-wave generation. Therefore, by storing appropriate delay values in each shift register, one can steer the ultrasound beam in space as shown in fig 16.



(a) Circuit Elements

(b) Low form factor

Figure 16: Shift Register Based Approach

This approach allows for much smaller area consumption. The IC can be directly integrated with CMUT transducers [32] using flip-chip bonding, leading to low form factors of 250 μm * 250 μm [33]. However, to operate the counter, the triggering clock must be in the order of the resolution of the system (the LSB delay). The same approach has been used in Ghovanloos group which needed a 200 MHz clock to generate the corresponding 5ns resolution [34] illustrated in figure 17.

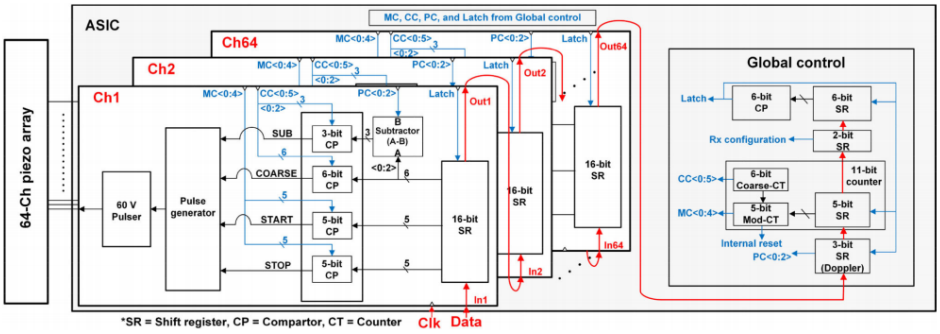


Figure 17: Similar architecture using counter-comparator based approach

For small LSBs, the clock frequency can easily reach hundreds of megahertz which leads to a proportional increase in power consumption. To use the given circuit in continuous wave mode, there are additional blocks that need to be incorporated that lead to an increase in power and area. One possible modification is as shown below in figure 18

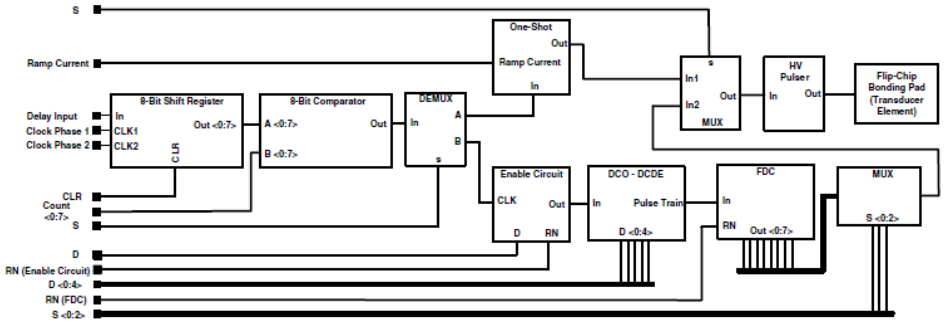


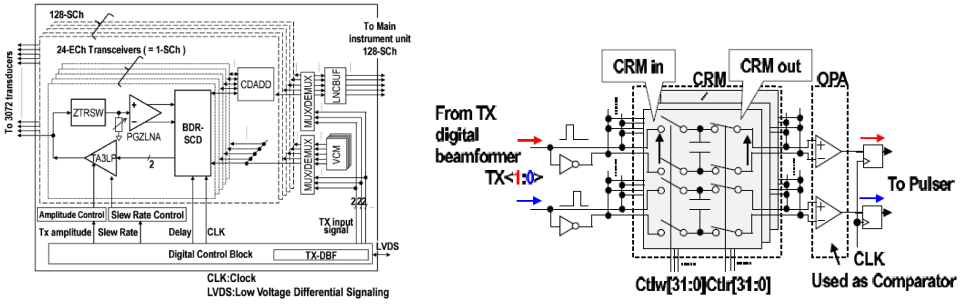
Figure 18: Modification for continuous stimulation

As can be seen there are several additional components in fig. 18 as compared to 16a, which makes it hard to integrate for low-area applications [35].

3.3. SAMPLE AND HOLD CIRCUITS

This method is particularly useful for imaging applications wherein ultrasound beamforming is desired for receiver beamforming. However, it is also often employed for transmit beamforming [36] as shown in fig 19.

In this method, a capacitor bank is used along with a sampling switch, and readout switch to achieve the necessary delay. The idea behind the sample and hold delay method is illustrated in fig. 20 and is used in the Pertijs Group for sub-array receiver beamforming [37]. In their work, the transmit beamforming is done in the digital domain using the counter-comparator approach mentioned in the prior section [38]



(a) System level implementation

(b) TX beamformer using sample and hold capacitors

Figure 19: Sample and Hold Based Beamforming Approach

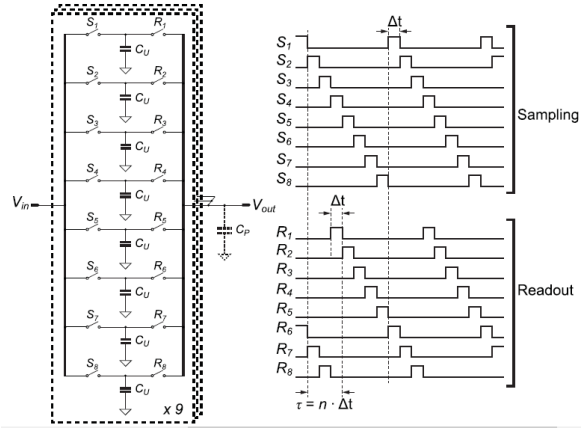


Figure 20: Sample and Hold Concept

The sample and hold technique is an effective low power (due to passive capacitor components) method to delay and sum signals. The delay generated from such a sample and hold circuit can be written in terms of the pulse width of the switching pulse[39]:

$$\tau = k * \Delta t$$

Where k is an integer from 1 to N-1
Where N is the total number of capacitors in the bank
and Δt is the pulse width for the switches.
From this, we can say that the delay is bounded by :

$$\Delta t < \tau < (N-1) * \Delta t$$

This can potentially lead to two issues. Firstly, to generate a wide range of delays, it directly implies a need for a large number of capacitors (sometimes in the order of 8 [37] to 32 [36]) leading to a large area being lost for other active components. Secondly, the

signal needs time to settle on the capacitors which might be an issue at higher frequencies of operation. An additional point to note is that high-frequency switching might incur additional power losses.

4

System Level Design

Based on the background information presented in the prior chapters, we are now in a position to analyze the multiple facets of the problem and arrive at a system level design. We first talk about the channel from an ultrasound generation point of view, following which, we delve into the beamforming circuit aspect.

4.1. Device Specifications

To start defining the device specifications, we start with the ultrasound generation aspect. The most fundamental consideration is the ultrasound frequency to use. It is a well known fact that there exists a trade-off between spatial resolution and penetration depth which are both dependant on the ultrasound frequency. If we choose a higher frequency, we achieve a higher spatial resolution, but we lose out on the ability to reach deeper regions and vice versa. Most ultrasound applications use a frequency range from 1-5 MHz and often even in the hundreds of kilohertz range. To keep a high spatial resolution for accurate stimulation, we keep the ultrasound frequency at 10 MHz. This allows us to keep a level of spatial resolution that is comparable to DBS, however we lose out on penetration depth as compared to lower frequencies of operation. However, since we use the stent in a blood vessel chosen close to the site of stimulation, we may not require very high depths of penetration, as the site is relatively close, due to the flexibility in placing the stent.

4.1.1. Channel Geometry

The array is composed of individual channels, and as mentioned in chapter 1, there are various constraints to consider while designing it. In this section, we attempt to delve deeper into the geometric constraints.

1. The first major constraint is the limited length of the channel as it is restricted by the blood vessel diameter. If the channel is too long, it will not lay flat on the blood vessel circumference, and instead will experience a great deal of stress from the

stent curvature. To minimize this, the length of the individual channel is restricted to a maximum of 250 μm, or roughly $\frac{1}{10}^{th}$ the diameter of a major blood vessel.

2. The next important parameter to look into is the width of each channel. Drawing from the inferences drawn from chapter 2, we notice that the inter-element spacing between channels of a phased array plays a vital role in the beamforming properties of the array. According to the discussion on grating lobes, we know that:

$$d \leq \frac{\lambda}{2}$$

Since we have chosen an ultrasound frequency of 10 MHz, this implies a λ of 150μm, or a $\frac{\lambda}{2}$ of 75μm. This means that the inter element spacing must be at the most equal to 75μm. Since we want as much area as possible to accommodate the beamforming circuitry, we set the inter-element spacing or pitch as 75 μm. One important consideration is the fact that the array must be physically pieced together from the individual channels. This poses a fabrication challenge as the width of the channel is closely related to the inter-element spacing from the basic relation:

$$w = d - k$$

Where d is the inter-element spacing (set at 75 μm)
k is the kerf or edge to edge separation between the channels
and w is the width of each channel
Since d is fixed, if we increase w, the kerf must decrease accordingly. This makes is quite challenging for the dicing saw to accurately cut such narrow kerfs. Keeping this in mind, we set the dicing saw line width or the kerf in this case to be equal to 25 μm.
That leaves 50 μm for the width of each channel. The final channel geometry can be visually realised from figure 21

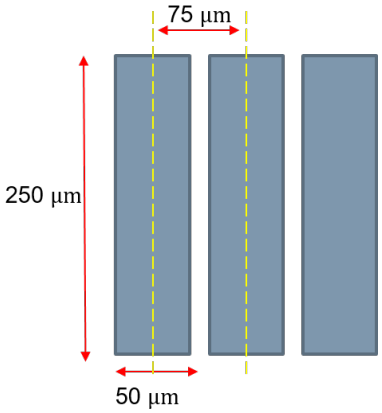


Figure 21: Final Device Dimensions

4.2. BIT RESOLUTION

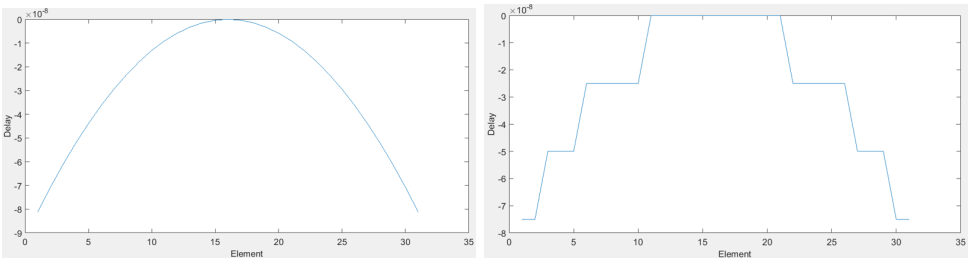
The beamforming circuitry is expected to generate delayed clock signals that cover the entire phase map. Since the delays generated are discrete in nature, there is a need to define the minimum possible delay that the system can generate. The entire delay spectrum can thus be represented as:

$$\Delta t = n * t_{min}$$

Where Δt is an arbitrary delay
Δt is the minimum delay and
n is an integer.
This also shows that all the delays have a linear relationship with respect to the integer 'n' which corresponds to the digital code for that particular 'n' value and should ideally form a straight line if plotted. This can be used as a good test for linearity later on.

4.2.1. QUANTISATION

From section 2.2.1 on wave equations, we notice that each element gets a delay that is continuous in nature and not discretized. From the previous section however, we notice that the system can only generate delays in multiples of t_{min}. Thus the continuous delays must be converted to a discrete form in a process known as quantisation. The results of the quantisation process are shown in fig 22 for 2 bits of resolution as an example. Here the Y-axis corresponds to the delays that the system can generate, and the X-axis



(a) Before Quantisation

(b) After 2 bit quantisation

Figure 22: Quantisation for 2 bits of resolution

corresponds to the different channels that are being assigned the given delays. This is an example of a beam-focussing delay profile, along the mid-line of the array. This process of quantisation directly provides us with the bit resolution required by the system. In the following section, we discuss the procedure followed to obtain the required bit resolution.

BIT RESOLUTION

First, we discuss the iterative procedure implemented to decide upon the correct number of bits required. We start with a MATLAB script that implements the wave equations discussed in section 2.2.1. The program is made specifically for a 1D array where the inputs given are the number of elements N, the ultrasound frequency used, the inter-element spacing d, the bit resolution desired, and the steering angle α. Once the delays

are generated for the appropriate channels, they are then quantised by rounding them off to the nearest discrete delay point. The final quantised delays are then converted to the corresponding phase offsets by the relation:

$$\Delta\phi = \frac{2 * \pi * \Delta t}{T}$$

Once we have the quantised phase offsets, we can move into the COMSOL model to test how well the beamforming properties are. This process is then iteratively repeated for a different number of bits to obtain the optimal resolution. The figure 23 below describes the process implemented to decide upon the bit resolution.

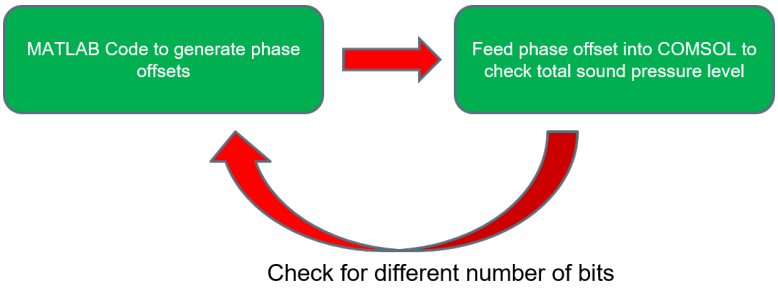


Figure 23: Process to obtain optimal bit resolution

4.2.2. COMSOL SIMULATIONS

COMSOL MODEL

To observe the beamforming profile for each of the phase offsets generated from the MATLAB code, we first need to design the COMSOL environment. To ease the computational load, we remove unnecessary complexities in the model and use the simplest possible environment to observe the beam profile with reasonable accuracy.
Since we are doing a mechanical simulation, we only consider the piezoelectric transducer in each channel. We assume that each channel is made of a piezoelectric material with the dimensions decided in the prior section. For the simulations, we assume the piezoelectric material to be PZT-5H as it is one of the most widely used materials for such applications and is easier to integrate monolithically on silicon as compared to PMN-PT. We use a 2-D simulation environment on COMSOL and use 32 channels. One side of the channel is connected to the ground potential while the other side is given a varying voltage according to the phase offset. The environment around the channel is given properties similar to that of brain tissue, in terms of density and speed of sound propagation. An extra addition to the model incorporates a Perfectly Matched Layer(PML) around the boundary of the model, to perfectly absorb stray wavelets and thus prevent unexpected reflections from the boundary of the model. Finally, a multiphysics coupling is made to account for the interactions between the solid mechanics domain and the piezoelectric domain. The basic model used in all subsequent simulations is shown in figure 24

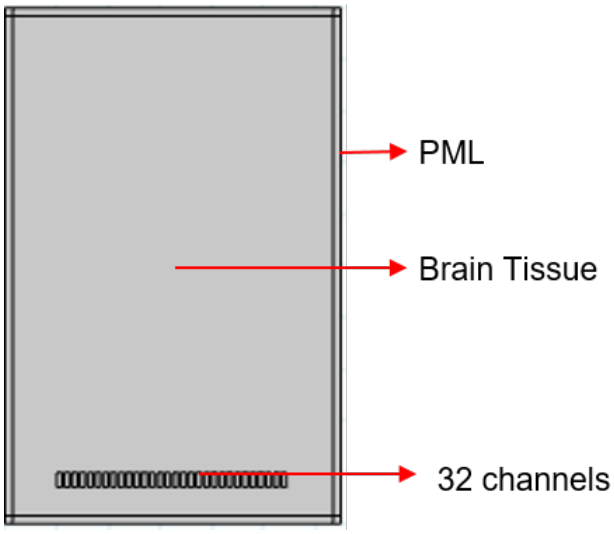


Figure 24: COMSOL Model

The following sub-sections will go deeper into the COMSOL simulations and dissect the beam profile, in terms of the desired characteristics.

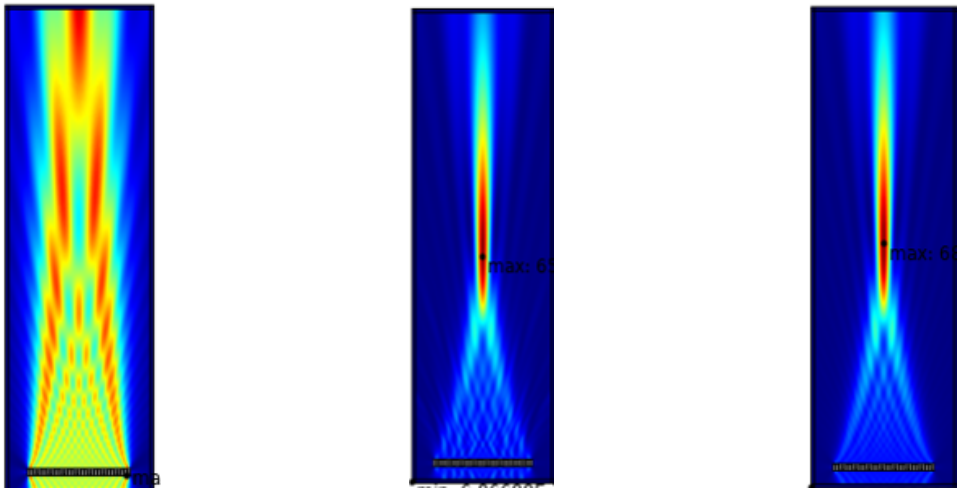
BEAM-FOCUSSING

The primary goal is to see how well the ultrasound waves get focussed to a point. We thus start with trying to create a focal spot along the mid-line of the array. The natural focus from Eq. 2.2 comes to be at around 12 mm. We set the parameters in the MATLAB code keeping the desired focal spot to be at 5 mm from the array. In doing so, we get the quantised phase offsets that are then inserted into the COMSOL model for voltage excitation of the piezoelectric material. On performing the simulation, we obtain intensity and sound pressure level plots that clearly show a focal spot forming at a distance of 5mm. This is also a good sign, showing the agreement with theoretical results across two different softwares: MATLAB and COMSOL. The next step, according to the plan outlined in fig 23 was to iterate for different bits of resolution. In doing so, we see the results in fig 25.

From the figure, we can see that 1 bit of resolution (only 2 possible delay values) is too low, and there is no proper focusing happening. At the same time, it is quite interesting to notice that at first glance, the beam profile for fig 25b and fig 25c are quite similar to each other. This point will be addressed further in subsequent sections.

BEAM-STEERING

Now that we see the ability of the system to focus ultrasound beams to a point at a distance, it is also very important to ensure that the ultrasound focal spot can be steered over a wide range of angles. This would be one of the major advantages of the nLIFU system, over the other conventional systems. Since our array geometry has been sized

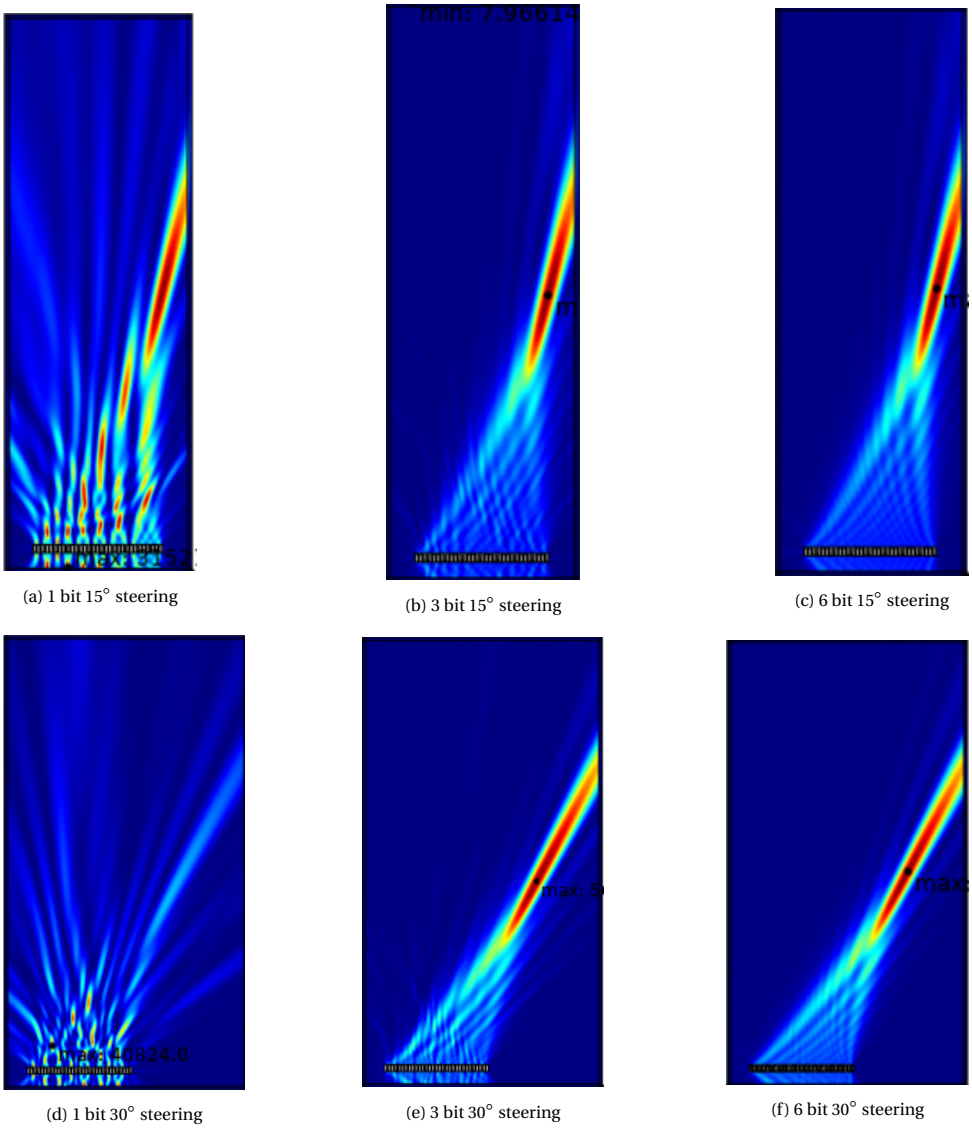


(a) 1 bit resolution (b) 3 bit resolution (c) 6 bit resolution

Figure 25: Varying bits of resolution

to prevent grating lobes for extreme steering angles going from $-90° < \alpha < +90°$, we do not expect to see any grating lobes while steering. This is further supported by the beam profiles when we implement 30° and 60° steering, as shown in fig 26.

From figures 26b, 26c, 26e and 26f, we can again observe that the beam profiles obtained for 3 bits of resolution is very similar to 6 bits of resolution. This points towards a possibility that we don't get much marginal utility when going from 3 bits to 6 bits and above.



(a) 1 bit 15° steering

(b) 3 bit 15° steering

(c) 6 bit 15° steering

(d) 1 bit 30° steering

(e) 3 bit 30° steering

(f) 6 bit 30° steering

Figure 26: Beam Profile for different bit resolutions and steering angles

FOCAL PRESSURE

In the previous sections, we see the beam-focussing and steering ability of the 1-D array. Next, we try to estimate the maximum focal pressure generated. To do this, we go back to our beam-focussing simulation setup and then simply observe the sound pressure level at the expected focal spot of 5 mm. We again repeat the process for different numbers of bits and plot them together in fig. 27

From here we see that the focal pressure is lowest at 1 bit of resolution, but rapidly

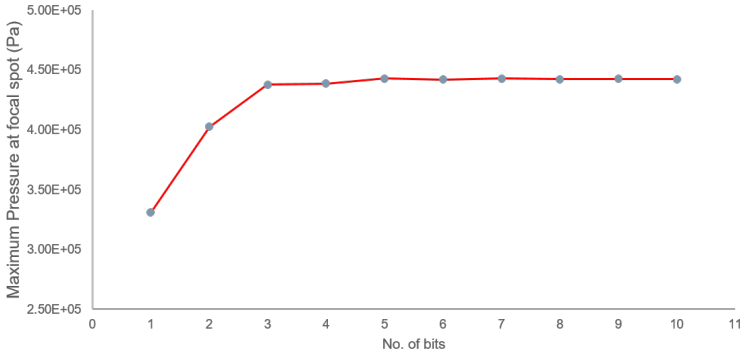


Figure 27: Sound Pressure Level vs No. of Bits

rises for 2 bits and 3 bits of resolution. However, for 3 bits of resolution onwards, there is no extra increase in pressure and the marginal utility of adding on extra bits is negligible. This is a very crucial observation as every additional bit needed greatly increases the complexity of the electronics, sometimes even exponentially [40]. We can say that 3 bits is enough resolution from a beam focusing, beam steering and focal pressure point of view, however, there are some additional constraints to look into.

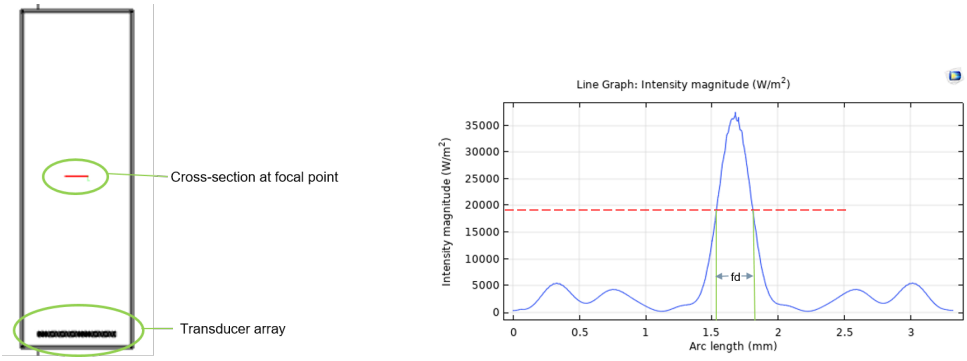
Beam Diameter

As mentioned in section 4.1, it is desirable to have the beam diameter as low as possible. We have chosen the device geometry appropriately to ensure optimal beam diameter (BD), however, it is also necessary to check if the beam diameter varies with the bit resolution. The beam diameter for a phased array is given by the equation

$$BD(-3dB) = \frac{1.02 * F * c}{f * D} \tag{4.1}$$

Where F is the focal length
c is the speed of sound in the medium
f is the frequency of ultrasound used and
D is the effective aperture

We thus expect the beam diameter to be independent of bit resolution, as it only depends on the physical characteristics of the array, and not the electrical back-end.
To calculate the beam diameter, we set the focal spot at 5 mm and take a 1 dimensional spacial cross-section of the intensity profile at that point as shown in fig. 28a.As seen in Fig. 28b From the intensity vs distance plot, we then calculate the -3 dB points with respect to the maximum central intensity and observe the corresponding diameter on the X-axis.
We then follow the outline of fig.23 and repeat the process for different number of bits. In doing so, we get the following intensity profiles that have been plotted together on fig. 29



(a) Setup to calculate beam diameter

(b) Spacial Intensity Profile

Figure 28: Beam Diameter Calculation

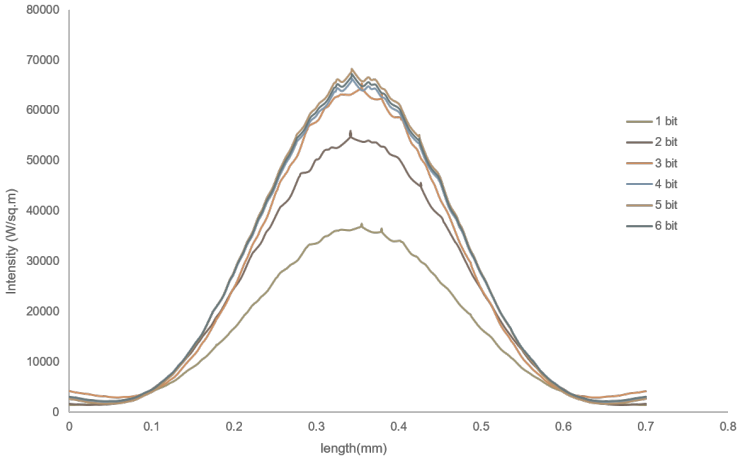


Figure 29: Beam Diameter for different Bit resolutions

On finding the -3 dB points and calculating the beam diameter, we see that the overall diameter does not change for different bit resolutions which is what we theoretically expected as well. The beam diameter hardly changes for different bit resolutions as shown in the table below :

No.of Bits	Beam Diameter (μm)
1	275
2	280
3	252
4	269.5
5	269
6	270

UNWANTED FOCAL SPOTS

Sometimes, there are unwanted focal spots generated during beam-focussing. Some are a result of grating lobes and others are due to constructive interference points occurring outside the focal spot. The grating lobes have been effectively removed for all steering angles through the appropriate array dimension sizing. The other unwanted focal points are resolution dependant. As can be seen from fig. 25 for 1 bit, there are multiple high-intensity focal spots apart from the primary focal spot. This reduces drastically for 3 bits and is almost the same for 6 bits of resolution. The difference between 3 and 6 bits is almost negligible and is unlikely to be a cause of unwanted stimulation at sites apart from the focal spot. This can only be confirmed by actual experimentation, but for now, we can safely say that 3 bits of resolution is sufficient to get the desired stimulation.

4.3. DEVICE SPECIFICATIONS

Based on the past sections, we can now define a system level device specification list. This includes all the major requirements that the system must meet to function as desired. The table below is the complete list of specifications obtained from the prior sections:

Parameter	Value
Operating Frequency	10 MHz
Bit Resolution	3
No. Of Elements	32
Minimum Delay	12.5 ns
Maximum Delay	87.5 ns
Pitch	75 μm
Channel Length	250 μm
Channel Width	50 μm
Element Spacing	25 μm
Active Area	0.0125 mm^2

5

DRIVING CHANNEL ARCHITECTURE

From the last few chapters, we developed a solid understanding of how the system is expected to perform. We also came to a final list of specifications that the system must meet at the end of the design process. In this chapter, we dive deeper into the beamformer design and try to develop a driving channel architecture from scratch. We start by defining our system in terms of the major blocks and at the end, we arrive at a possible driving channel architecture that can be applied for brain stimulation.

5.1. SYSTEM BLOCKS

As has been covered in the prior chapters, our system requires three building blocks that create the complete stimulating channel. As shown in fig. 30, the first block is the beamforming unit that generates the correct phase required by the channel in question. The beamformer unit is in the 1V domain to reduce power consumption. The second block is the level shifter that acts as an interface between the beamformer and the subsequent block. The final block is the high voltage driver that generates the required high voltage signals that directly reach the piezoelectric material and generates the ultrasound waves. The level shifter is often a crucial block because the 1V signals from the beamforming block are not capable of driving the much bigger MOSFETs in the driver block.

5.2. TRANSMIT BEAMFORMING TOPOLOGIES

From Chapter 3, we saw various beamforming topologies. The beamformer is going to be at the core of our system, thus special care must be taken to ensure that the constraints are taken into account. We saw that the DLL approach has area requirements in the range of 0.1 mm^2, however, according to our device specifications, we are allowed an area of 0.012 mm^2, which is about $\frac{1}{10}^{th}$ of the area occupied by the DLL, and we haven't even considered the area occupied by other major blocks in the system, like the pulser which is also expected to consume a large area. The counter-comparator method is efficient, however, it requires a fast clock depending on the counter frequency required.

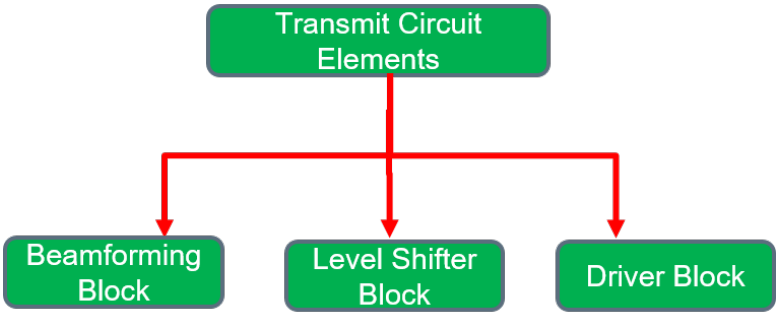


Figure 30: System Blocks

This can be hard to generate on-chip, and also incurs larger switching losses. Continuous wave generation also requires additional blocks that consume extra area. The sample and hold method requires a large capacitor bank and is area inefficient. To mitigate these issues, we try to use first principles and create a design topology from scratch.

5.2.1. DESIGN METHODOLOGY

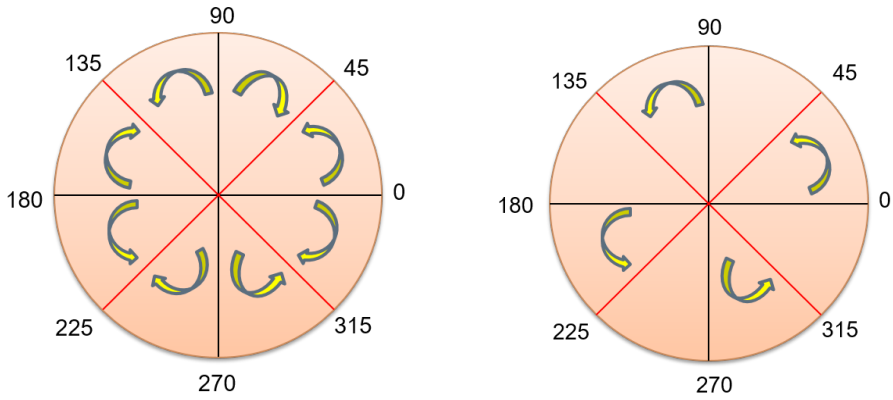
To start the process, we must keep in mind the fact that the two biggest constraints presented to us are the area and power constraints. To create the beamformer, we need to use low-area blocks that also consume a low power. This eliminates all the prior architectures and forces us to think of a novel approach. To ease the phase generation requirements of the beamformer, we choose to use a coarse delay + Fine delay structure. This means that we could use a coarse delay block to generate 2 bits of phase resolution, and then use a fine delay block to generate the additional 1 bit of resolution.
Since 2 bits of resolution corresponds to the phases $0°, 90°, 180°$, and $270°$, we need to generate quadrature signals. The fine delay block can then use the quadrature signals generated from the coarse block, to implement the third bit of resolution.
Once the quadrature signals are available, the 3rd bit can be intuitively generated from them, in two different ways:

- Interpolate between subsequent quadrature signals.
- Delay each quadrature signal by the minimum delay.

This is illustrated on the phase maps in fig. 31. These two ideas form the basis for the final novel transmit architecture talked about in the subsequent sections.

5.3. COMPLETE STIMULATION SYSTEM

Based on this idea, we can come up with two new ways of creating the complete system. This is the first time these approaches are being used for a low area phase generator. Both the methods discussed below have a significant advantage over conventional ultrasound transmit architectures, in terms of power and area consumption.



(a) Interpolation based fine phase block

(b) Delay based fine phase block

Figure 31: 3 Bit Generation

5.3.1. INTERPOLATOR BASED SYSTEM

As can be seen in fig 32, the quadrature generator generates the four 90° shifted phases. Following this, we pass them through buffers and then use multiplexers to choose two adjacent quadrature signals to interpolate between. Once the quadratures are chosen, the interpolator receives them and generates the phase between them thus covering all 3 bits of resolution (8 phases). Following this, the 3 phases are passed through another multiplexer to choose the final desired phase. This is then passed directly to the level shifter and high voltage driver that directly interfaces with the ultrasound transducer. The bits required to program the multiplexers are sent through a shift register-based control block which also needs to be placed inside the channel. This novel system level design uses a quadrature generator and phase interpolator as the main elements and using simple auxiliary circuits for the additional tasks.Since the overall architecture is simple, by making the correct design choices for the individual blocks, we are able in a position to get low area and power from the system.

5.3.2. DELAY BASED SYSTEM

In this system, we create an absolute delay to generate the required fine phases. Similar to the prior method, the quadrature generator creates the four quadrature phases that are then passed to a buffer and then to a multiplexer. The multiplexer chooses one of the phases to delay. This chosen phase is then passed to the delay block that provides the necessary 45° shift. Finally, a second multiplexer chooses between the two phases and sends it to the level shifter and subsequently, the driver. This process is illustrated in fig 33. Like the previous system design, this novel approach also has a simple design allowing it to be low in area and power.

The final system to shortlist will depend on how the fundamental interpolator block/ delay block function with respect to the given constraints. The circuit-level aspects of these two systems will be covered in the subsequent section.

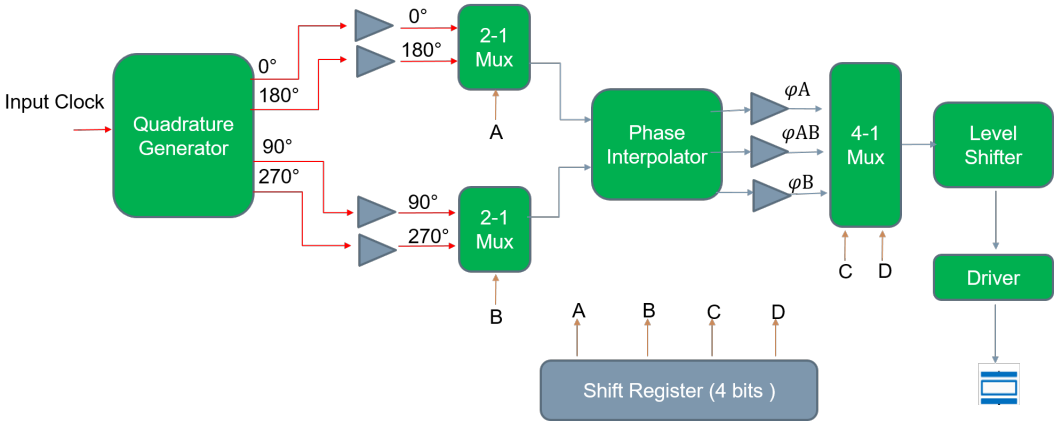


Figure 32: Possible interpolator Based System

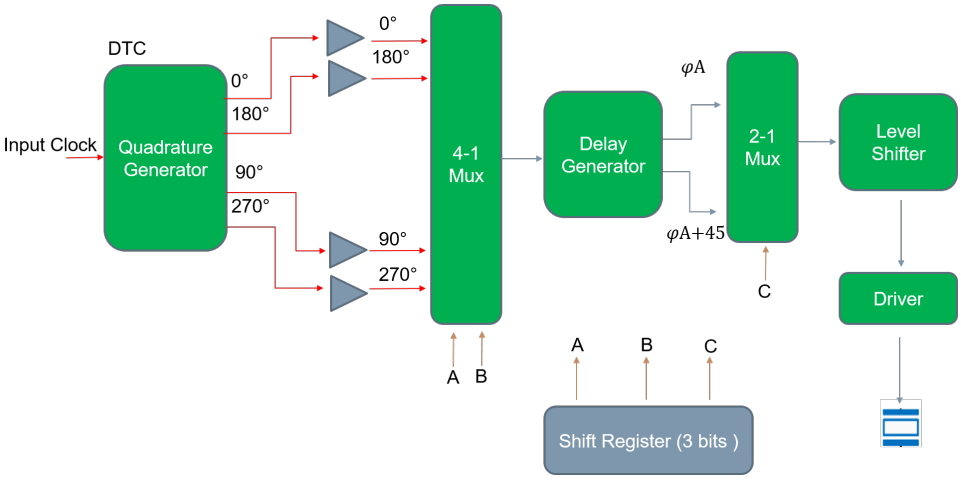


Figure 33: Possible Delay Based System

6

CIRCUIT DESIGN

In this chapter, we shall explore the fundamental CMOS circuitry of the channel that makes up the blocks discussed in the prior chapter. This chapter explores the circuit design aspects and the various trade-offs made therein. At the end of the chapter, we arrive at a final system that meets the desired specifications and constraints mentioned in the initial chapters.

6.1. BEAMFORMING CIRCUITRY

THEThe core of the chip is the beamformer that generates the timed delay signals to each channel. In order to reduce the power consumption, we decide to operate the beamformer block at a supply voltage of 1V.The subsequent sections talk about the sub-blocks that constitute the beamformer.

6.1.1. QUADRATURE GENERATOR

On recalling the system architecture diagram discussed in chapter 5, we see that the first major block that forms the core of the entire stimulation channel is the quadrature generator. The quadrature generator must create very stable and process invariant quadrature signals as the fine phase generator block uses these signals to generate the complete phase map of 8 phases. This has to be done while consuming very low power and low area.

One of the most common and effective ways of generating stable phases is by utilizing the DLL. However, as mentioned below, a single DLL is likely to occupy nearly the entire channel area, leaving no room for the level shifter and the high voltage driver. This led to the need to search for novel alternatives that are capable of generating quadrature signals while satisfying the constraints.

In the RF domain, there is often a common need to generate 0° and 90° phases in I/Q generators. Drawing inspiration from these generators we arrived at two low area and low power digital blocks that are capable of generating such signals. Since we are dealing in the digital domain, it makes things very convenient to use digital blocks. The method

of using quadrature generators to generate the 90° phase-shifted signals has not been used before for ultrasound transmit architectures and is thus a new way of seeing phase generation for low area and power systems. Here, two possible methods of generating the quadrature signals are discussed. The two digital blocks to use as quadrature generators are similar in a number of ways but have subtle differences. The two approaches A and B are discussed in the following section below.

6.1.2. APPROACH A

The first method involves using two D Flip Flops to generate the quadrature signals by sending inverted clocks to each of them and recording the output from their Q output terminal. Another advantage of using D Flip Flops is the fact that they possess both Q and $\bar{Q}$ implying that the two flip flops can together generate all the four quadrature phases 0°, 90°, 180° and 270°. This is shown in fig.34 below: In this method, the $\bar{Q}$ output

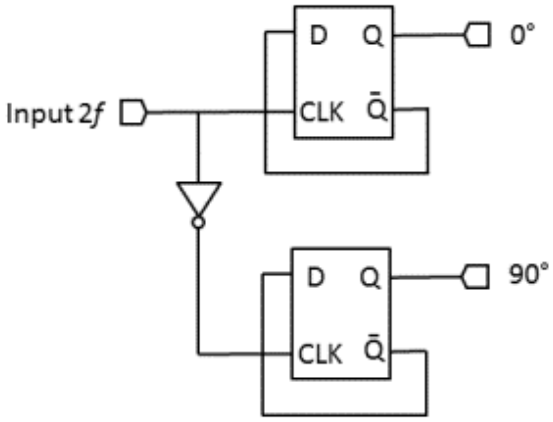


Figure 34: Quadrature Generator A

is fed back to its corresponding D input. The system also acts as a frequency divider so a frequency double the required output frequency must be applied to the input.

6.1.3. APPROACH B

The alternate method also involves two simple D Flops but in this approach, the flip flops do not operate independently but are instead in a cascaded structure as shown in fig 35.

Here, the $\bar{Q}$ of one flip flop feeds back into the D input of the other flip flop. Also, both clocks receive the same clock trigger. Another point to note, is the fact that this architecture also acts as a frequency divider, and the input frequency must be four times the expected output frequency which is 10 MHz according to our specifications.

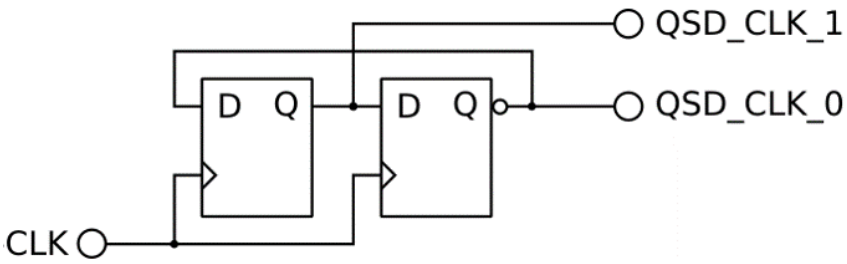


Figure 35: Quadrature Generator B

6.1.4. D FLIP FLOP

From the prior sections, we notice that for digital quadrature generators, D Flip Flops can be used as the basic building block. It is thus necessary to choose the most appropriate D Flip Flop circuit structure to optimize the performance of the flip flop and in the process, improve the functioning of the quadrature generator as a whole.

TSPC D FLIP FLOP

The True Single Phase Clock (TSPC) D Flip Flop is a very commonly used D Flip Flop in many digital applications. The flip flop employs dynamic logic that allows it to be much faster (reduced Clk to Q delay), while also requiring a lesser number of devices to implement it [41]. The most simple TSPC flip flop is shown in fig 36.

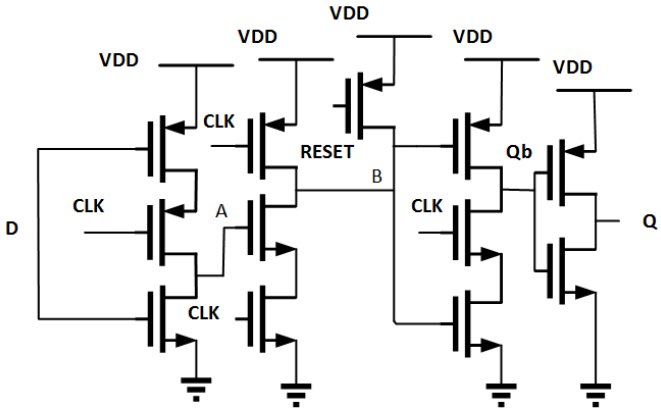


Figure 36: TSPC Flip Flop

The TSPC flip flop seems quite advantageous due to its low component count, however there are some serious issues to address:

1. The flip flops use dynamic logic and from fig 36 we notice that nodes A and B are floating nodes. They are thus susceptible to race conditions and proper MOS sizing is extremely important to ensure they work correctly. Some MOSFETs have

to be sized larger to increase their drive strength, which cancels out the advantage of a lesser number of components, from an area perspective.

2. At certain process corners, the PMOS and NMOS V_{th} changes leading to a big change in the race conditions for the floating node. This means that the pull-up / pull-down contention might go against expectation, leading to a faulty output.

3. The TSPC flip flops are cascoded structures of three MOSFETS. This makes is harder to scale down to lower voltages while ensuring correct operation across corners, and taking care of the race conditions.

These factors combined make the TSPC flip flop much harder to implement in our system.

TRADITIONAL D FLIP FLOP

The traditional flip flop uses static logic for its operation. There are multiple ways to implement the flip flop with the Transmission Gate Flip Flop (TGFF) [42] being a popular option as shown in fig. 37. In this flip flop, transmission gates are used to control the flow of data through the master and slave units. The positive feedback loops ensure fast settling and leaves no floating conditions like the TSPC variant. In this method, we do

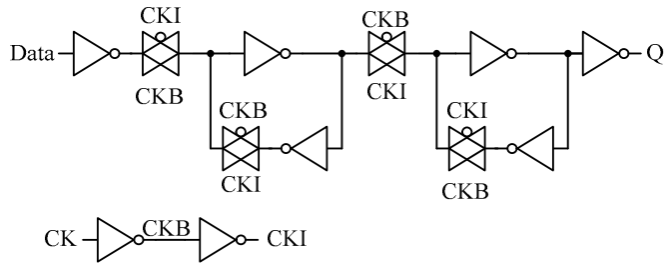


Figure 37: TGFF Flip Flop

not suffer from the drawbacks of the TSPC type flip flop. We use inverters as the basic building blocks and they can withstand supply voltages even below 1V. They also do not suffer from race conditions and can be scaled to very small MOS sizes without any loss of performance across the different corners. However, the only drawback is the fact that it requires more components (minimum 18 MOSFETs as compared to 12 in the TSPC variant), however, since they can be scaled down to very small dimensions, the increase in area is offset by this and there is not much area benefit from the TSPC variant. This makes the TGFF design more favorable for our application.

6.1.5. APPROACH A VS APPROACH B

Now that we have defined our basic building blocks i.e. the D flip flops, we are in a position to compare the two different quadrature generators to decide the optimal variant. In the first method A shown in fig. 34, we see that the two flip flops operate independently of each other to generate 0°, 90° and 180°, 270° respectively. There is also an

additional inverter required to invert the clock that enters the second flip flop. This additional inverter can generate some small delay which can generate a skew in the clock outputs. The 4 phases would therefore not be in perfect quadrature to each other. In addition to this, the initial conditions on the flip flops have a big effect on the output clock phases. If the outputs are not set properly, there is a chance that the flip flops generate output phases of 0°, 270°, and 90°, 180° which is not the correct order. Since the system is defined based on the correct phase outputs from the flip flops, it is very important to consider the initial conditions. Having a reset input to set the output to $Q = 0$ and $\bar{Q} = 1$ is thus a compulsory requirement.
In approach B, the flip flops are connected in a sort of feedback mechanism. The output of one flip flop becomes the input of the other. This makes the system much more accurate and minimises the variation in phases. The reset input is still necessary for this system as well, however the same clock is fed to both flip flops and there is no additional inverter, thus leading to even lesser variation in the output phases.

APPROACH B SIMULATIONS

Using approach B, we created the circuit design using the TGFF D flip-flops on Cadence. The output waveforms depicting the four quadrature outputs are shown in fig. 38. We see

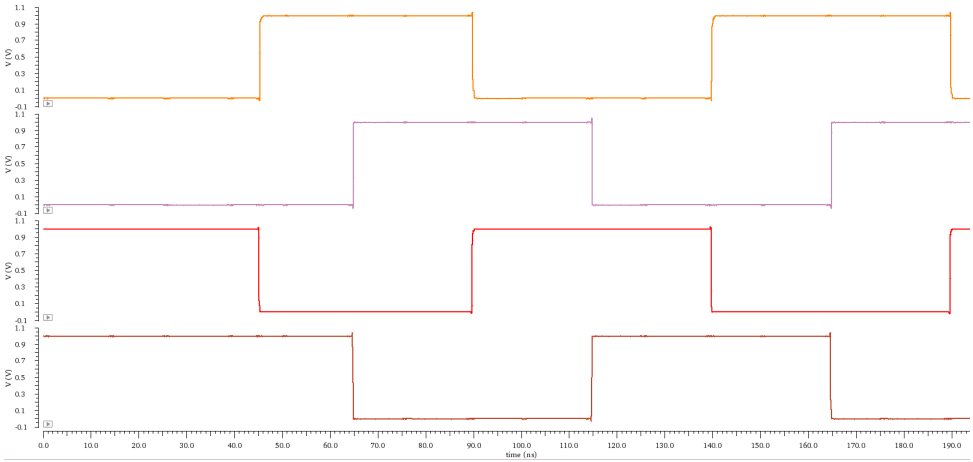


Figure 38: Quadrature Generator Output

the desired 10 MHz, 50% duty cycle quadrature signals generated from the quadrature generator.

PROCESS VARIATIONS

It is also essential to check the process variations as the entire system depends on how well the quadrature generator can produce the desired 25 ns spaced signals. Checking process corners is a good way to see how robust the quadrature generator will be. From fig. 39, we see that the process variations are very small compared to the LSB of the system which is 12.5 ns. The observed edge to edge variation for the extreme corners

is less than 1 ns (647 ps), showing that this method of generating phases is quite robust and can be compared to the stable output of a DLL.

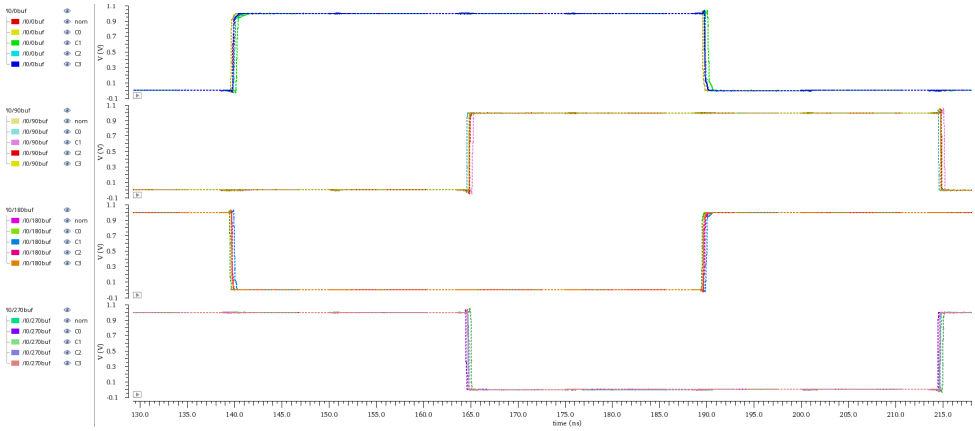


Figure 39: Quadrature Generator Process Variations

POWER CONSUMPTION

It is also interesting to note that the average power consumption of the quadrature generator is 0.892 μW which is several orders smaller than the power consumption we can expect from a DLL based quadrature generator (past work on DLL based ultrasound generator showed a power consumption of 0.18 mW for a 1.8V supply [43]).

6.2. MULTIPLEXER

The multiplexers need to be designed using a minimal number of elements and using very small components. There are several different approaches used to create a multiplexer with the most basic being the one built using logic gates, based on the truth table. This is shown in fig. 40: However, this multiplexer requires a very large number of MOS-

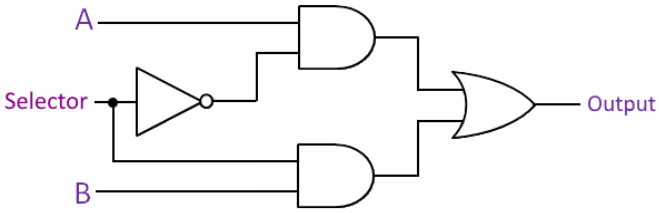


Figure 40: Logic Gate Based 2*1 Multiplexer

FETs as each logic gate itself comprises of multiple MOSFETs (4 MOSFETs in 1 NOR gate and 4 MOSFETs in 1 NAND gate). This makes a simple 2*1 multiplexer quite big and power hungry. A 4*1 multiplexer would be even larger in area. A more elegant approach

is to use complementary pass transistor logic to build the multiplexer. In this architecture, the signal flow is controlled using the gate of the MOSFET as an enable switch. Fig. 41 below shows the implementation of a simple 2*1 mux using transmission gates.

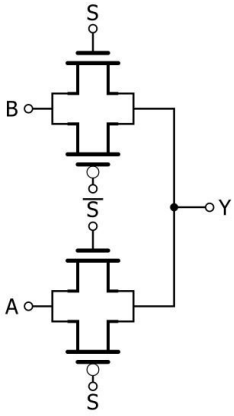


Figure 41: Complementary Pass Transistor Logic Based 2*1 Multiplexer

As can easily be observed, this approach requires much lesser components and can easily be sized down to very small component sizes without any loss in functionality, making it ideal for our application.

6.3. FINE DELAY BLOCK

Following the level block diagram of fig. 32, we can see that the next major block is the fine delay block. As was talked about in the previous chapter, we came down to two feasible methods of creating the desired phases: interpolation and delay. These two methods are now discussed in the subsequent sections below.

6.3.1. PHASE INTERPOLATION

The first method that could be used is based on the theory of phase mixing. Phase interpolators are usually very compact structures that consume very low areas. They are also not power hungry in general, thus making them a good candidate for our application. The phase mixing in interpolators are done using the formula:

$$\Phi_i = a * \Phi_a + b * \Phi_b \tag{6.1}$$

Where a,b are the weights that can be varied to steer the interpolated phase between Φ_a and Φ_b.

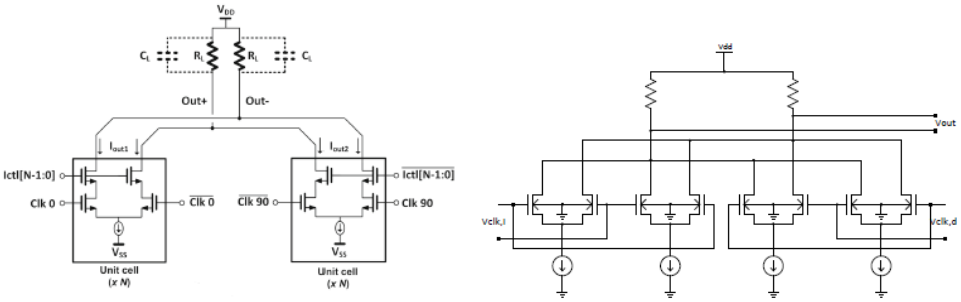
TYPES OF PHASE INTERPOLATORS

Phase interpolators can be classified into two major forms:

- Analog Phase Interpolators

- Digital Phase Interpolators

Analog phase interpolators usually have a cascoded MOS structure with current source tails as shown in figure 63a. From equation 6.1, we can see that the weighing factors a,b must be varied to change the phase [44]. This is done by placing multiple MOSFETs in parallel. By switching different numbers of them On or Off, we can steer the phase between the two extremes [45].



(a) Setup to calculate beam diameter

(b) Spacial Intensity Profile

Figure 42: Analog Phase Interpolator

The major drawback of analog phase interpolators is the fact that the current sources are a source of static power dissipation that can be an issue for long-term usage. The cascoded structure also becomes an issue in terms of maximum output voltage swing and cannot be used for low supply voltages.

Digital Interpolators on the other hand often use inverter based interpolation structures. These are not just low area, but also consume very low power (almost negligible static power dissipation). Figure 43 shows a 2-bit digital phase interpolators using current starved inverters to achieve the phase mixing [46]

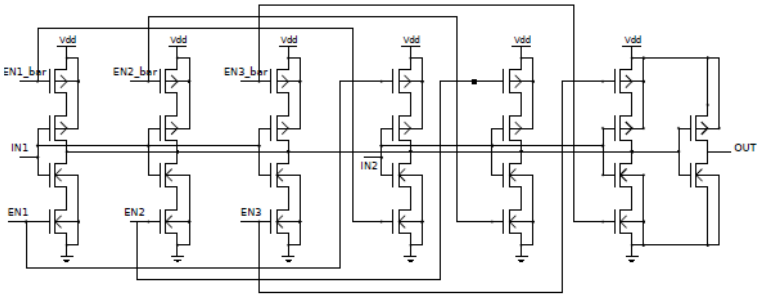


Figure 43: Digital Phase Interpolator

As mentioned before, we need the quadrature phases to be interpolated or 'mixed equally to generate the required one extra bit of resolution. This means we ideally require

a 1-bit digital phase interpolator. The 2-bit phase interpolator of fig. 43 can be simplified and used as a 1-bit inverter based interpolator [47] as shown in figure 44.

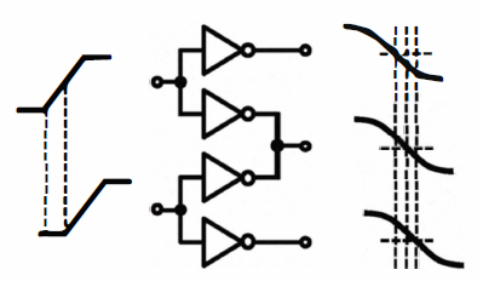


Figure 44: Digital Phase Interpolator

This 1-bit interpolator seems to be the perfect match for our fine phase block. It satisfies the requirement of being very small and can simultaneously be used at very low supply voltages of even 0.5V [47]. The number of blocks required increases by the order of $2^N - 1$ [48] but since we only require 1 bit of resolution, this is not an issue. Since it is based on inverters, the power consumption is also quite low as the static power losses are almost negligible.

DRAWBACKS

Despite the apparent perfect fit, there are some major issues to address for the phase interpolator based approach. Firstly, the phase interpolator is always used to generate very fine phase steps (in the order of 7.5° [26] to 11.25° [49]) while also being used in high frequency clock applications (frequency > 100 MHz). In our case, we are using a relatively slow clock with a time period of 100 ns. Thus the quadrature outputs are already spaced 25 ns apart which is a significant duration. The following points are some major issues that arise because of this.

1. Slew Rate: One of the biggest issues of using a phase interpolator-based architecture is the dependency of linearity on the slew rate and delay interval between the two signals being interpolated. It has been seen that the ratio between the rise time and the delay between the interpolated signals plays a crucial role in determining the overall accuracy of the interpolation[47]. It has been found that the ratio $\frac{t_{rise}}{t_{delay}}$ should be from 1-10 in order to get accurate interpolation. Fig. 45 below shows the ratio of $\frac{output\,delay}{input\,delay}$ which should ideally be 0.5 for the 1 bit interpolator and plots it with respect to $\frac{t_{rise}}{t_{delay}}$. As can be seen, the output tends to rest at the desired 0.5 value, when the $\frac{t_{rise}}{t_{delay}}$ is above 1. This poses a problem for us, as in our case, the quadrature signals are spaced 25 ns away from each other. Thus t_{delay} is 25 ns in this case. In order to get accurate phase interpolation, we need the rise time to be at least 25 ns. For a symmetric waveform, the fall time should also be 25 ns. However, our input signals have a time period of 100 ns. This makes our input signals trapezoidal in shape with a high time of only 25 ns. The quadrature generator creates quadrature signals that are square waves with rise times around 1-2 ns,

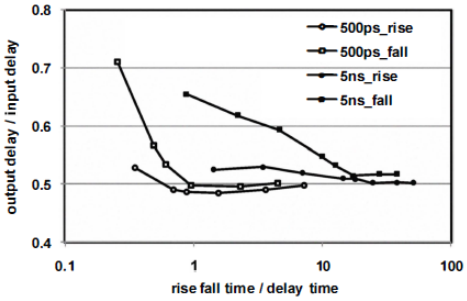


Figure 45: Rise time and input delay dependency on phase interpolation

and duty cycles of 50%. Thus, there is going to be a slew rate controller required in order to create the correct slew rate for accurate interpolation as illustrated in fig. 46. To verify this, the phase interpolator was implemented on Cadence and tested

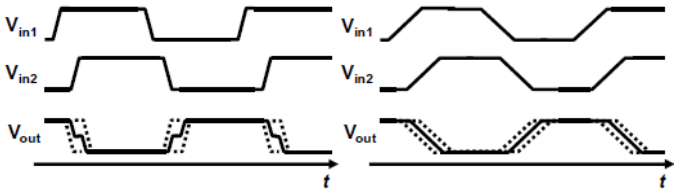
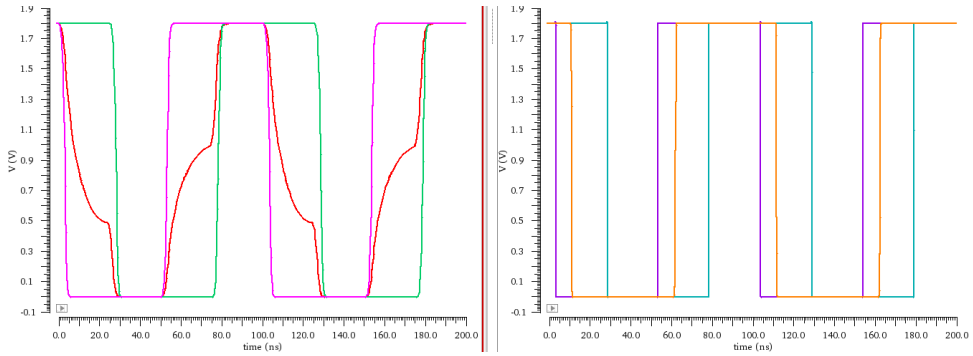


Figure 46: Slew rate control

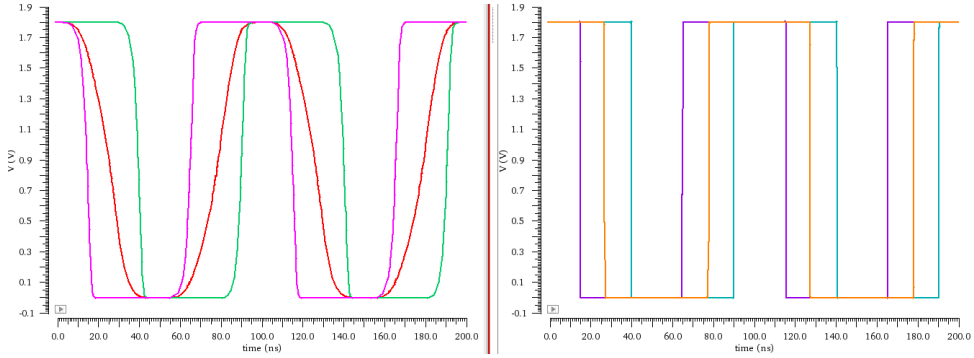
with a rise time of 5 ns, and 25 ns respectively, for a fixed input delay of 25 ns. The results shown in fig. 51 match the expected plots of fig. 46 .

The left subplots show the output of the interpolation, along with inverted input signals as shown in fig.44. The right subplots show the final outputs from the buffer. It can clearly be seen that the interpolation is very close to ideal (midway between the quadrature signals) for the 25 ns rise time case, while in the 5 ns case, the interpolation itself is very non uniform owing to the large delay of 25 ns between the quadrature inputs to the phase interpolator.

2. Output Capacitor: A phase interpolator requires an output capacitor bank to perform the interpolation correctly[50]. On layout, capacitances occupy a significant area, and even stacked MIM capacitors have a relatively low unit capacitance (fF/mm^2) making it hard to save area for other critical components. It has been observed that the requirement for a large $\frac{t_{rise}}{t_{delay}}$ can be relaxed by using a large capacitor bank at the output of the interpolation node[51], as shown in fig. 48 In order to ensure all the outputs are affected equally, the same capacitor must be placed at the other outputs as well, as shown in fig 43. This implies that the phase interpolator block requires a total of 3 output capacitors corresponding to the three outputs. In our case, we notice that in order to obtain sufficient accuracy



(a) Interpolation for 5 ns rise time



(b) Interpolation for 25 ns rise time

Figure 47: 5 ns input rise time vs 25 ns input rise time

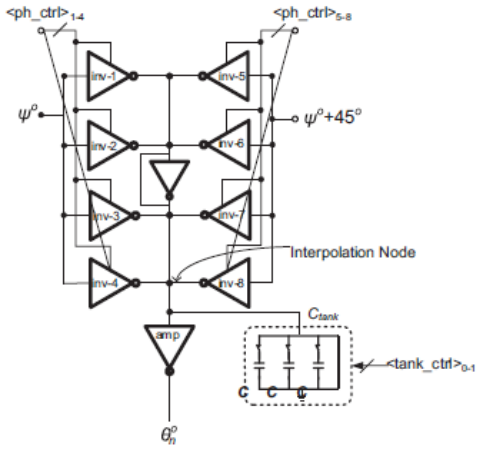


Figure 48: Capacitor Bank at interpolation node

in interpolation, for a small rise time of 5 ns, we require three capacitors of 500 fF each, which leads to a significant area consumption.

3. Process Variations: Process variations are quite important in this application as the phase delays generated should be spaced out at appropriate intervals. If the fine phase delay block generates delays that are not exactly between the quadrature phases at different corners, our overall system linearity will be affected. Our ideal delay expected from the interpolator output is $\frac{1}{2} * 25 = 12.5\,ns$. We can see how much this varies for different corners. The results of the corner analysis is shown below in fig. 49: The corresponding duty cycle and delays are mentioned in the

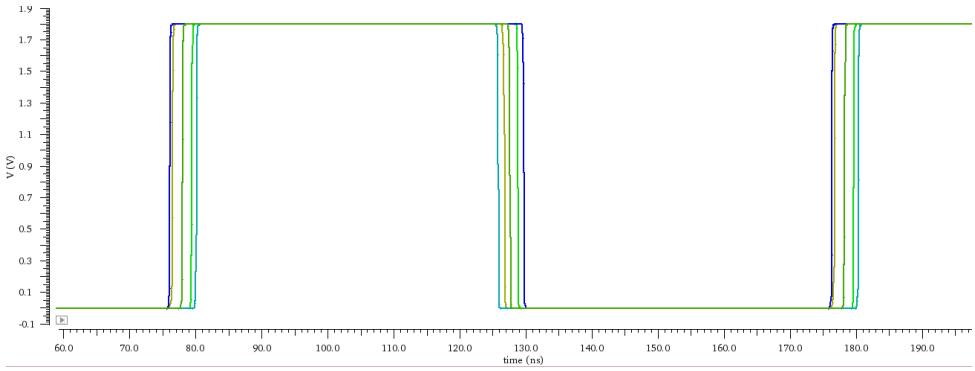


Figure 49: Effect of process variation

table below:

Process	TT	FF	SS	FS	SF
Duty Cycle (%)	50.06	50.09	50.65	46.62	53.71
Delay (ns)	12.65	12.09	12.68	13.75	11.61

Table 2: Process variations for phase interpolator

6.3.2. DELAY GENERATION

From the previous method, we see that despite being a very appropriate solution on the surface, many disadvantages cannot be easily resolved. This is why we extend our search to use a delay generation method for this particular application. Such an approach has not been used before for such ultrasound transmit architectures. It has a big advantage of consuming very low area due to the simplicity in its design. Based on fig. 31b, we see that there is a need to produce a stable delay of 12.5 ns, which corresponds to the LSB of the 3 bit system. This delay can be generated in a number of ways, however care must be taken to ensure the constraints are satisfied. Some of the most important being area, power and process variations.
One simple approach is using a simple inverter based delay like the method used in fig. 50 where a DTC is implemented to generate fine delays based off an input clock and capacitor banks [52].

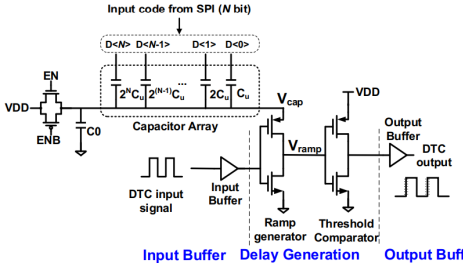
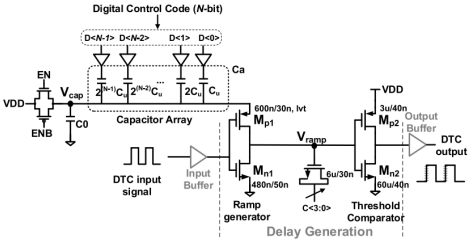
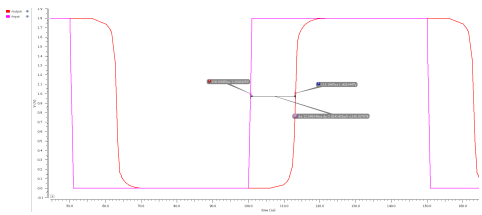


Figure 50: Fine delay generation using inverter based DTC

In this method, the power consumption is very low, with a high delay resolution, however the LSB is very small at 103 fs [53]. In our case, the delay required (12.5 ns) is quite high and requires sizing the length of the single inverter considerably in order to slow the inverter enough. These large MOSFETs with big lengths develop large self capacitances that do not fare well under different process corners and also consumes significant area. Using a chain of inverters is a better approach, as it is possible to achieve the 12.5 ns delay without any unreasonable device sizing. The disadvantage of this lies in the fact that process variations get added up for each extra stage used, and still doesn't solve the issue of area consumption. One solution is to use a capacitor at the output of the inverter stage, as implemented in fig. 51a. Using this idea, we are able to generate 12.5 ns of delay as shown in fig. 51b. However, at the ramp node, we need a capacitor of at least 1pF. This is not feasible as it consumes a very large area.



(a) Capacitor Bank at node V_{ramp} [52]



(b) Generating 12.5 ns delay using single stage inverter

Figure 51: Delay Generation using single stage inverter

6.3.3. CURRENT STARVED INVERTER

One possible solution is to starve the current flowing through the inverter. A current starved inverter allows us to use a single stage inverter, while also allowing very slow changing outputs. These outputs can then be passed through another fast inverter stage that acts as a threshold comparator i.e it changes the output level based on the threshold crossing decided by the sizing of the individual MOSFETs in the inverter. This is similar to what is used in fig. 48 and 51a. The current starved inverter has two core MOSFETs that act as the switches. The outer two MOSFETs act as current sources that source or

sink current from the output node capacitor, depending on which inner MOSFET is active. The idea behind a current starved inverter is shown in fig. 52 below: By fixing the

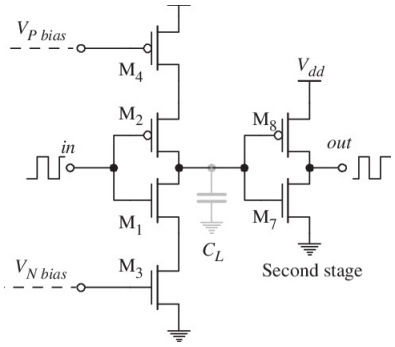


Figure 52: Current Starved Inverter

current flowing through the inverter, we can directly modify the slew rate, and thereby change the delay precisely. The slew rate output can be written as:

$$SR = \frac{dV}{dt} = \frac{I}{C}$$

By changing the bias current, or the capacitance, we are able to modify the slew rate of the circuit.

6.4. PROPOSED ARCHITECTURE

As mentioned in the preceding sections, having a current starved inverter could lead to a feasible solution in terms of area, power consumption, and process variations. As can be seen from fig. 52, the current starved inverter requires additional biasing circuitry in order to correctly control the outer MOSFETs in the current starved inverter (i.e V_{Nbias} and V_{Pbias}. There are multiple ways of biasing the MOSFETs however we must try to use the least number of components. One of the most common ways in which this is done is by using a simple bias circuit to set the correct bias point. We often see bias circuit implementations using a voltage controlled device in ring oscillators[54] like the one shown in fig. 53.

This method works well when we require different delays, depending on the V_{cntrl} we provide. The control voltage must also be generated externally for use in the bias block. However in our case, we require a constant delay and since our channel is autonomous, we require the control voltage to be generated locally (to reduce total cable count), which implies an additional block to generate this control voltage, leading to an additional area consumption. It is thus a better idea to look into a different bias block that is 'self biasing' in nature.

6.4.1. β MULTIPLIER

The beta multiplier is one of the most well studied and used self biasing blocks. It does not require any additional stage to bias it and is able to generate bias currents based on

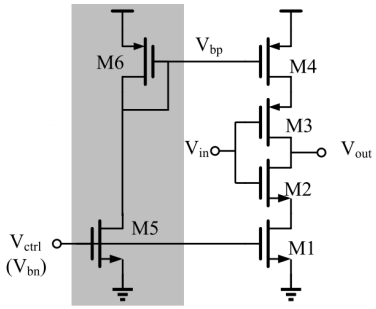


Figure 53: Simple Biasing Block

the MOSFET sizes and control resistor chosen [55]. The simplest beta multiplier is shown in fig. 54. By sizing the resistor, we are capable of changing the current flowing through

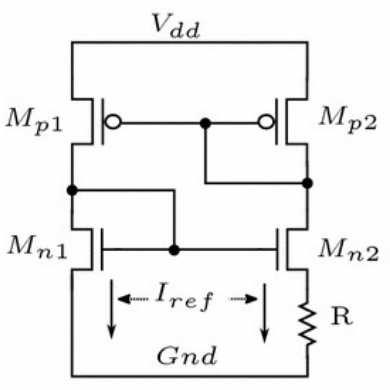


Figure 54: Basic Beta Multiplier

the bias branch and are thus able to reduce the current consumption considerably.This is because the current is given by:

$$I_{ref} = \frac{2}{R^2 \mu_n C_{ox} S_{n1}} (1 - \sqrt{\frac{1}{N}})^2$$

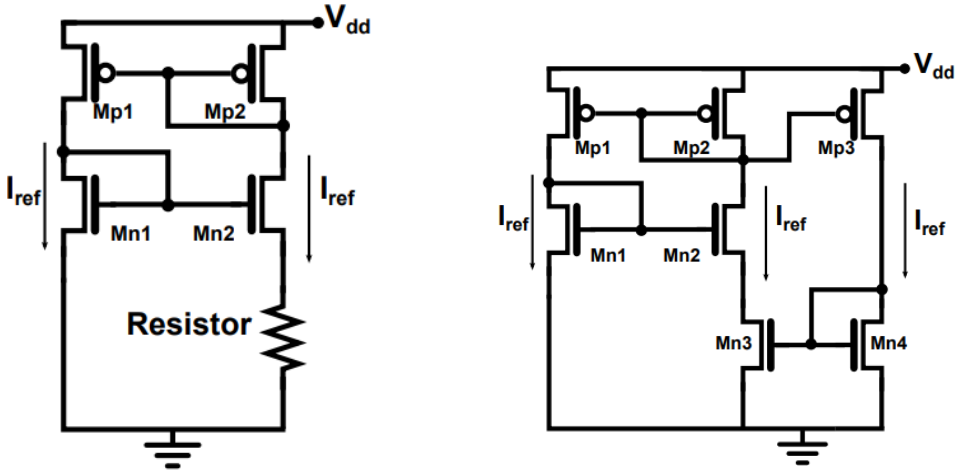
Where $S_{n1} = \frac{W_1}{L_1}$ and N is the multiplying factor ($W_{n2} = N * W_{n1}$). The N factor is kept as 4 in this case. From the equation, we see that by increasing R, we are able to lower the bias current and save power, however it comes with an increase in area consumption.

BETA MULTIPLIER OPTIMISATION

1. Startup Circuit: The beta multiplier, being a self biasing circuit, can be made to operate in two states: the 'zero current' state where all the currents in it are zero and the MOSFETs are off, or the stable operating region, where the current bias currents are flowing and all the MOSFETs are on. In order to ensure that the device

operates in the correct state, we also require a start up circuit, to leak some current and push the system away from the 'zero state'.

2. Pseudo-Resistor: The simple beta multiplier is capable of generating the desired bias conditions, however to reduce the bias currents, we need to use large values for the resistor. This is a problem in terms of area consumption since having a large resistor consumes a significant area, in spite of using serpentine structures and other layout techniques. Resistors are also known to be susceptible to large process variations during fabrication. A large enough variation could change the bias currents that would eventually lead to an incorrect delay generation. Instead, it is possible to use a resistorless beta multiplier [56], by implementing a pseudo resistor, using a MOSFET in the triode region as shown in fig. 60



(a) Beta multiplier with resistor

(b) Beta multiplier without resistor

Figure 55: Resistor-less beta multiplier [57]

FINAL DELAY GENERATOR

Based on the idea of a current starved inverter and using the optimizations/ modifications made above, the complete system is implemented as shown in fig. 56

RESULTS

When we simulate the delay generator circuit shown above on Cadence, we can tune the bias current, and the MOSFET sizings to generate the exact delay of 12.5 ns that we desire. After tuning the circuit parameters, we can arrive at the result shown below in fig. 57:

From here, we see the correct delay being generated with a value of 12.49 ns and a duty cycle of 51%. In addition, since we are using a very low bias current and also small-sized elements, the power and area consumption are relatively low.
It is also important to see how much the delay varies with respect to process variations as

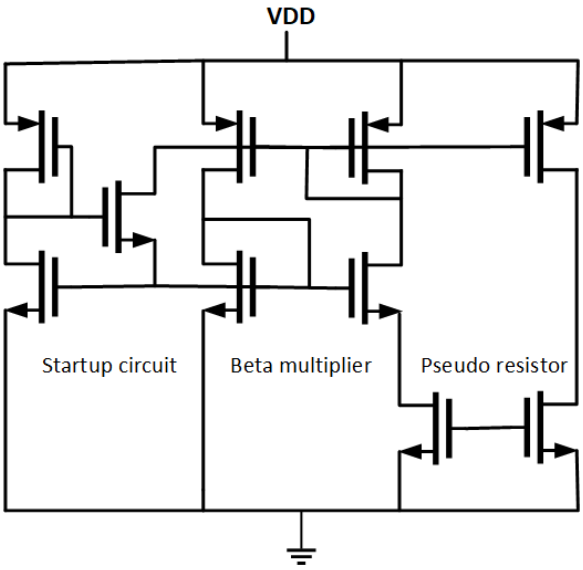


Figure 56: Final Beta Multiplier

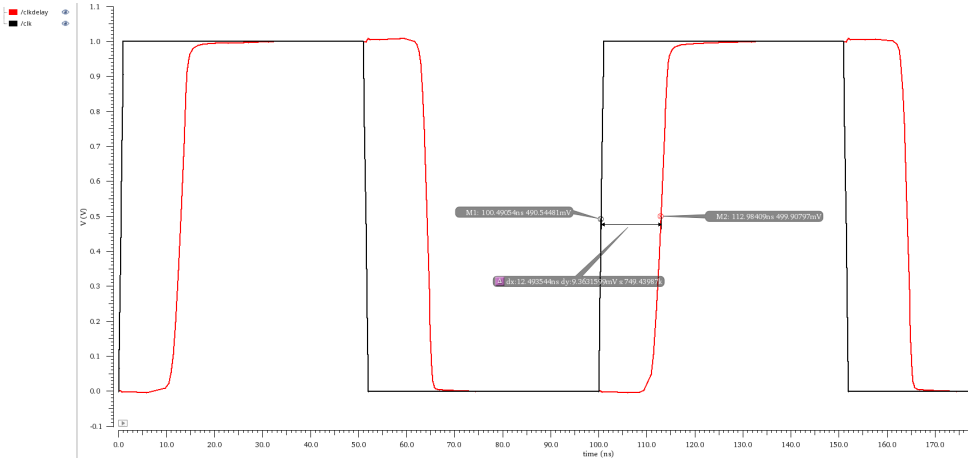


Figure 57: Delay Generator Output

this would directly affect the linearity of the system. The results of the process variation test are shown below in fig. 58. We see that the delay variation is quite small with an extreme deviation from ideal of around 2.5 ns. Compared to the LSB delay of the system at 12.5 ns, a 2 ns delay is quite small in comparison. The exact delay and duty cycle variations in each of the process corners is mentioned in the following table: Thus this method is able to generate the stable 1 LSB delay while consuming very low area and power.

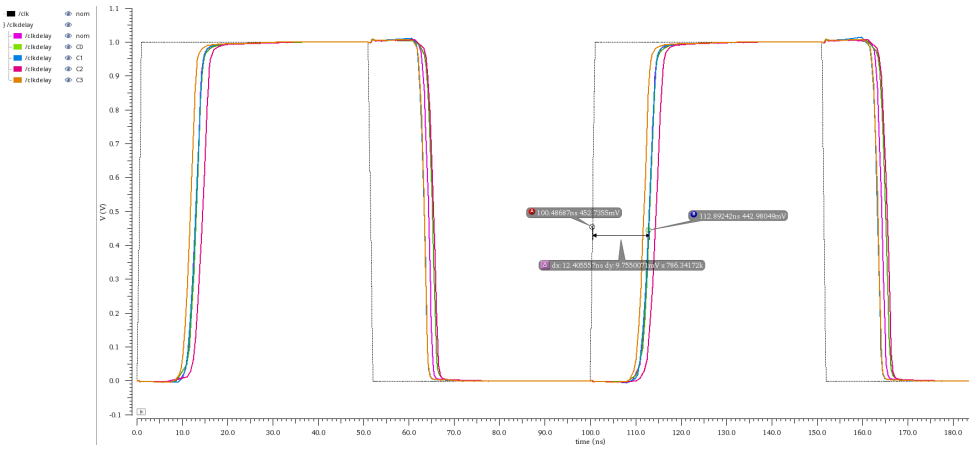


Figure 58: Delay Generator Output Variation with process corners

Process	TT	FF	SS	FS	SF
Duty Cycle (%)	51.23	51.95	50.33	50.89	51.48
Delay (ns)	12.48	12.59	12.48	13.99	11.31

Table 3: Process variations for delay generator

6.5. LEVEL SHIFTER

The next major block to design is the level shifter. Since we are in the 1V domain for all the beamforming, it becomes a challenge to drive the high voltage devices in the driver stage. The 1V signals are unable to provide enough drive strength, leading to partial or sometimes no turning on of the high voltage MOSFETs. This is undesirable and must be addressed. To resolve this issue, a level shifter can be used effectively to interface between the 1V domain, and the high voltage domain. The block diagram below illustrates the need for the level shifter.

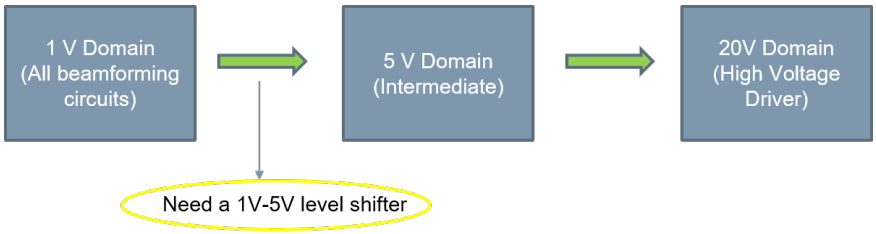
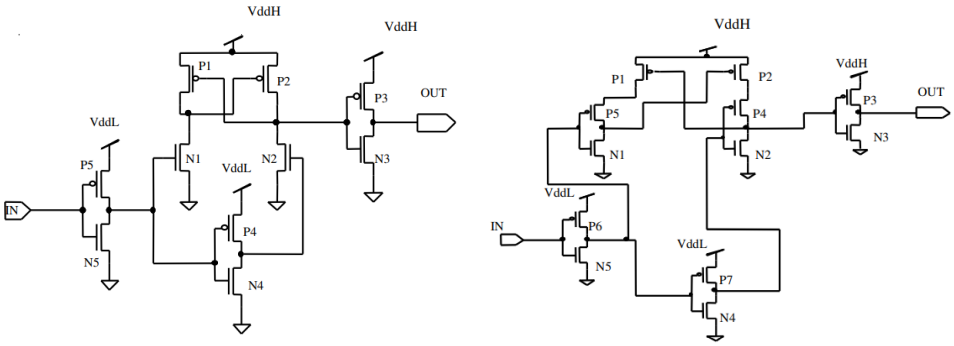


Figure 59: Level Shifter Block

6.5.1. LEVEL SHIFTER TOPOLOGIES

There are multiple ways to implement a level shifter, but as before we have to ensure that the area and power consumption are least, while also being resistant to process varia-

tions. The most simple form of the level shifter is the conventional cross bar level shifter as shown in fig. 60a



(a) Conventional Level Shifter

(b) Contention Mitigated Level Shifter

Figure 60: Level shifter topologies

In this particular level shifter, the input inverters operate at the lower voltage level and are used to turn on the other MOSFETs that are connected in a cross-linked fashion [58]. The operation can be understood as follows- From fig. 60a, we see that when N1 receives a high voltage at its gate, it turns on, while N2 turns off. By N1 turning on, the MOSFET P2 receives a low voltage at its gate, which switches it on. This further allows a high voltage to appear at the gate of P1 which turns it off. This kind of positive feedback arrangement allows very low short circuit currents to flow through the NMOS PMOS pairs, thus saving power.

A modification of the conventional level shifter is the contention mitigated level shifter shown in fig. 60b. This particular topology is known to further reduce the short circuit power, by inserting an inverter stage between the input and cross bar devices[59]. The conventional level shifter of fig. 60a suffers from contentions between the pull-up and pull-down transistors. This contention leads to larger power losses and large device sizings in order to handle the contention. The smaller the supply voltage, the more severe the issue becomes. In our case, a 1V supply is a big concern in terms of contention. The insertion of the quasi-inverter allows the cross-bar nodes to settle faster leading to lower power consumption due to a crowbar current reduction. This makes it a better choice than the conventional level shifter.

DRAWBACKS

Both the methods mentioned above are suitable to act as level shifters for our application. However, there is an issue here, as the levels we need to shift between are 1V and 5V. Normal 1.8V MOSFETs are unable to withstand V_{ds} values of 5V. There is thus a need to use special 5V MOSFETs that are also available in the TSMC 180nm BCD technology that we are using. These MOSFETs are capable of withstanding 5V differences across them and are thus the perfect MOSFETs to use in the crow bar structure that will frequently see 5V differences being generated. Unfortunately, the 5V MOSFETs are significantly larger than the 1.8V devices but is a requirement for the system to function properly. We

must ensure that only the minimum number of such devices are used, and only when required.

A much bigger problem is the fact that these 5V MOSFETs have a V_{th} that is around 800 mV. This becomes a big issue as the maximum voltage that the main MOSFETs of the level shifter see is only 1V since our beamforming block works at that voltage. This implies that the maximum overdrive voltage available at the 5V MOSFETs is only 200 mV. This is quite low and in the contention, the 5V MOSFETs that receive this overdrive voltage would always lose, and in the process, not level shift the 1V pulses accurately. For this, those MOSFETs have to be sized very large to compensate for the low overdrive, and be able to contend with the pull up branch. Through simulations, it was observed that a MOSFET width of around 70 μm was needed in order to successfully contend with the pull-up branch. This kind of device sizing is impractical in this application where area constraint is such a big issue.

6.5.2. FINAL LEVEL SHIFTER TOPOLOGY

The prior section showed how it is difficult for the general 5V MOSFETs to be used with such low supply voltages. An intuitive solution is to use 5V low V_{th} MOSFETs instead, as they are capable of handling 5V levels while simultaneously having a low V_{th} level. This is a very good solution, as such devices are available in the technology node being used. However, a low v_{th} 5V MOSFET occupies an even larger area than the nominal V_{th} 5V MOSFET. Once again, this is a necessary requirement in order to deal with the 1V supply voltage issue. Thus we must ensure that the absolute minimum number of such devices are used. A topology that uses the effectiveness of the low V_{th} MOSFETs is shown in fig. 62 [60]. In this figure it can be seen that the topology is very similar to the contention mitigated structure mentioned above with the exception that this particular topology has an additional diode-connected pull-up MOSFET in each branch, and also possesses the low V_{th} MOSFETs talked about in the prior section.

In this structure, the diode-connected MOSFETs are used to weaken the pull-up strength of the PMOSs and thereby allow for smaller NMOS sizings. However since we are using such a diode-connected device, there is a voltage drop of one threshold across it. In order to restore the output swing to 5V, we thus need one additional inverter stage which leads to an increase in the area requirement of the overall design structure. It is thus a worthwhile experiment to check whether this modified contention mitigated structure of fig. 62 fares better than the original contention mitigated structure shown in fig. 60b with low V_{th} MOSFETs instead of the standard ones.

COMPARISON

Firstly, both the structures were designed on Cadence and low V_{th} 5V transistors were used (As we know that the contention mitigated structure would require device widths greater than 70μm and is not a fair comparison).
To make the comparison fairer, the same MOSFET sizings are used for both structures. The simulations are run across different process corners (as slow and fast MOS devices would lead to a change in the contention strength of the pull-down and pull-up paths). The figure below shows the resulting outputs of the level shifters being compared.

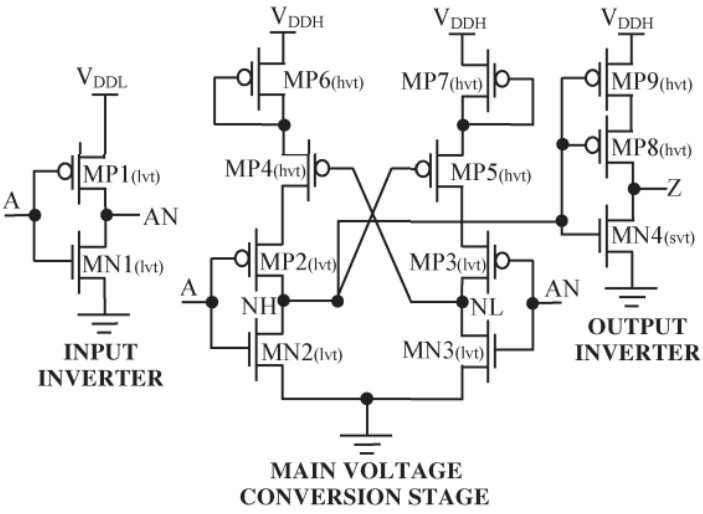


Figure 61: Modified Contention Mitigated Level Shifter

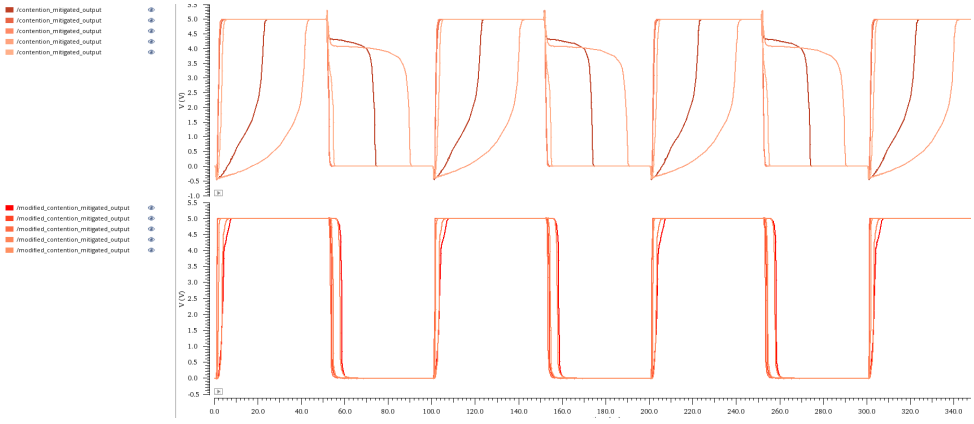


Figure 62: Output waveforms for different process corners

We keep the same NMOS and PMOS sizings for both setups ($W_n = 3\mu m$, $W_p = 1.2\mu m$) and see that the modified contention mitigated structure does much better at the extreme process corners. The basic contention mitigated structure is not able to pull up/ pull down at some process corners which makes it unsuitable.

Another point to note is the fact that delay in the level shifter is not an issue for us. This is because all the outputs of the beamforming blocks from all the channels need to pass through the level shifter. Thus any delay happening in one channel would ideally happen in the other channels as well. This implies that the linearity would not be affected, as the delay offsets of all the channels would nullify each other. The rough

power consumption and device component of both topologies are mentioned in the table below: From this, we see that the power consumption of both is almost the same,

	Contention Mitigated LS	Modified Contention Mitigated LS
Power Consumption	11.9 μW	12.25 μW
Number of Transistors	6	10

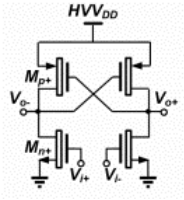
the modified contention mitigated LS has a few extra devices, however, as was shown in fig. 62, the size of these devices is smaller at $W_n = 3\mu m$, $W_p = 1.2\mu m$. The contention mitigated LS is not able to function effectively at these small sizes for different process corners. Thus we don't lose much in terms of area and power, while still level shifting effectively at different process corners. The Modified Contention Mitigated Level Shifter is thus the most effective level shifter for this application.

6.6. HIGH VOLTAGE DRIVER

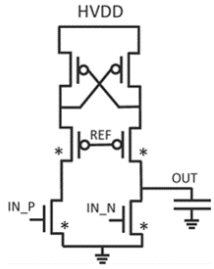
The final block of the complete system is the high voltage driver that directly interfaces with the ultrasound transducer and generates the high pressure ultrasound waves. It is intuitive to realize that the higher the voltage swing at the output node of the level shifter, the higher the amplitude of the ultrasound wave and consequently higher is the pressure at the focal spot. We need a high enough focal pressure of around 1 MPa, to induce neural activation. Based on the initial COMSOL simulations, we expect that a 20V output voltage swing will be sufficient to generate enough pressure at the focal spot. We thus now look into possible high voltage drivers that are capable of handling this voltage level while keeping in mind the constraints we have to adhere to.

6.6.1. DRIVER TOPOLOGIES

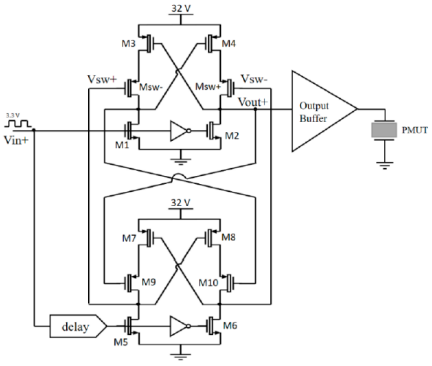
Needing to drive ultrasound transducers is a common issue that has been addressed numerous times in the past. It is a requirement for both imaging and stimulation systems and is a common issue to deal with it. There are a number of ways to design a high voltage driver. Most systems try to reduce the power consumption of the driver since any short circuit current will lead to very high instantaneous power consumption, owing to the high supply voltages being used. The most intuitive way to resolve this is to imagine the driver as another level shifter stage that converts 5V waveforms to 20V or higher waveforms. On thinking of it this way, we can once again employ the level shifter topologies talked about in subsection 6.5.2. This is how many groups have approached the issue. The crow-bar structure is a common theme in a number of works as shown below in fig. 63. These methods of designing a high voltage driver are effective however, they need at least 4 high voltage MOSFETs. The major issue is the fact that these MOSFETs experience a voltage difference of 20V across them and need to be fabricated using special wells and extra isolation. This makes the device very large in comparison to simple 1.8V MOSFETs. A comparison of the MOSFETs available in the TSMC technology library is shown below in fig. 64: All the NMOSs have the same W, L of $5\mu m$ and $1.6\mu m$ respectively. As can be seen, the high voltage MOSFETs, though capable of withstanding very large voltages across their drain-source junction, consume a very large area in gen-



(a) High Voltage Driver A[27]



(b) High Voltage Driver B[61]



(c) High Voltage Driver C [62]

Figure 63: High Voltage Driver Topologies

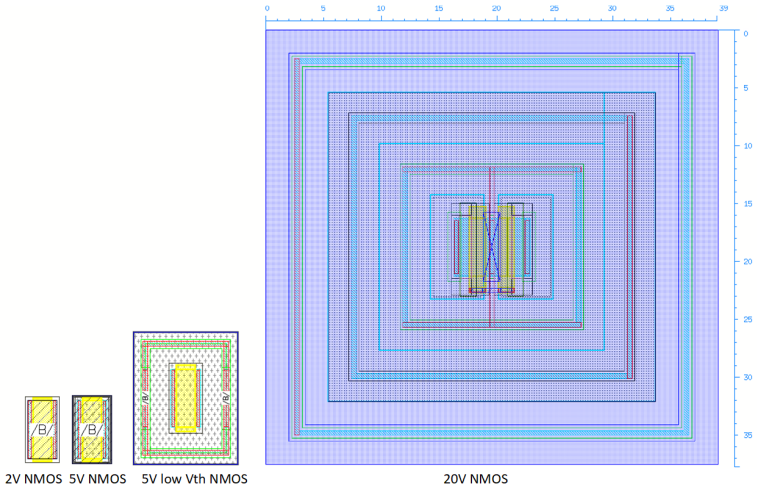


Figure 64: MOSFET size comparison

eral. This is a big issue in very tight area constraints like in our application. It is thus a necessity to use the minimal number of such high voltage devices in order to save area. As shown in the figure, despite the NMOS having relatively small device dimensions, occupies a significant area of roughly $37\mu m$*$37\mu m$. This number would be much higher as the device dimensions are expected to be considerably higher to ensure enough drive strength for the load transducer. Four high voltage MOSFETs would then become hard to implement in the area at hand. It is thus necessary to implement a driver with a lesser number of such devices.

6.6.2. PROPOSED HIGH VOLTAGE DRIVER

Since most of the level shifter-based drivers need a minimum of 4 HV MOSFETs, it makes sense to search for alternatives. One such alternative is using a simple inverter stage to drive the transducer. This would mean needing only two HV MOSFETs thus saving twice the area as the previous level shifter approach. However, the HV MOSFETs require a 0-5V V_{gs}. Our 5V level shifter provides us with this required voltage range, however the HV PMOS accordingly requires a voltage of 15V-20V at it's gate. The input to the inverter stage only goes from 0-5V. There is thus a need to translate the voltage level from 0-5V to 15-20V to activate the HV PMOS. This is done by using a high pass AC coupler that translates the 0-5V voltage pulse, to a 15-20V pulse, [63] shown in fig. 65.

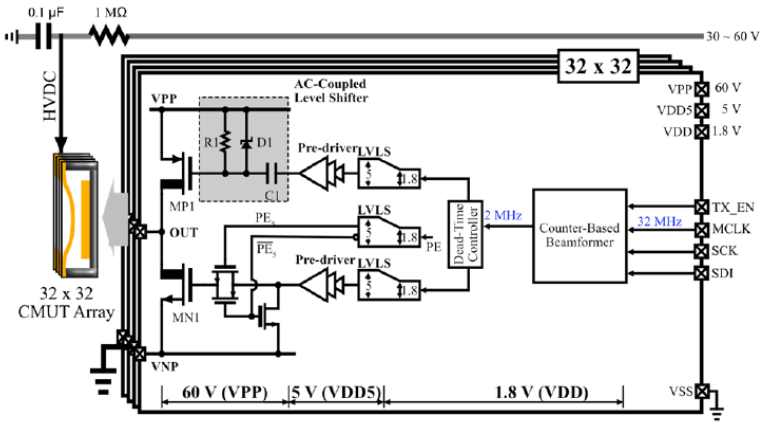


Figure 65: HV Driver with high pass filter

From the figure, we can see the greyed-out region encompasses the high pass AC coupler that allows the correct voltages to be applied to the HV PMOS in the inverter structure. This topology is quite well suited for our application as it only requires two HV MOS devices as compared to four in the previous level shifter-based topology.

The high pass filter cutoff frequency is an important factor in deciding how well the driver behaves. The cutoff frequency is defined by

$$f_{-3dB} = \frac{1}{2\pi RC}$$

. Since our input is a square wave of 10 MHz, with many frequency components, we want to preserve as much as possible, and this can be done by setting the cutoff frequency of the high pass filter as low as possible. This is done by increasing R and/or C, however, this is not ideal as we start consuming more area. We try to set the cut-off frequency of the filter at 1 MHz which is a decade lower than the fundamental frequency and should prevent excessive loss of frequency components. The driver requires a resistor, capacitor and a diode.In order to save area, the resistor is made using a serpentine structure. A custom interdigitated capacitor is made using metal layers 4,5 and 6, and placed above the resistor and diode, to make the system more compact. The diode, is implemented using a HV 5V diode available in the BCD HV library. The layout of the driver looks as shown in fig. 66

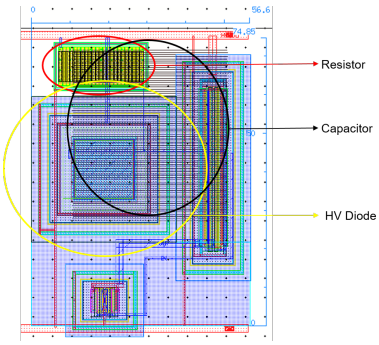


Figure 66: Driver Layout

6.6.3. PIEZOELECTRIC TRANSDUCER MODELLING

Now that we have fixed the driver topology, we need to properly size the MOSFETs so that the stage can properly drive the piezoelectric transducer. In order to do so, we must model the transducer in the electrical domain, to estimate the kind of loading it poses for the driver. This kind of modeling is achieved using the BVD (Butterworth Van Dyke) model for piezoelectric materials[64]. According to this model, the piezoelectric material can be modeled by a capacitor in parallel with a series connection of a resistor, capacitor and inductor as shown in fig. 67 It is thus necessary to estimate the values of the different

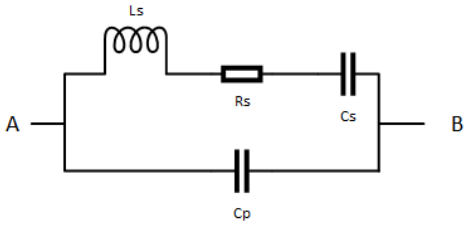


Figure 67: BVD model of piezoelectric transducer

parameters like C_p, C_s, R_s and L_s. To get the values, we first look into the parameters of

the piezoelectric material we are using. To have a starting point, we decide to work with PZT-5H. due to the extensive work done on it, in terms of piezoelectric transducers, and also due to the ease in implementing it within fabrication processes. The PZT-5H that we have ordered has the following important characteristics shown below:

Parameter	Value
t	270 μm
e_{33}	6
k_{31}	0.44
k_3^T	3800
k_{33}	0.75
ρ	7800 kg/m^3
Q	32
E	$5 * 10^{10}$ N/m^2

We shall now use some relations in order to estimate the BVD parameters. The PZT-5H ordered has a resonant frequency of 8.2 MHz which is close to the frequency we want to operate the device at. Some important parameters can be deduced from the following equations[65]:

$$Q = \frac{1}{R_s}\sqrt{\frac{L_s}{C_s}}$$

$$C_s = \frac{8lwd_{31}^2}{\pi^2 t s_{11}^E}$$

$$L_s = \frac{\rho l t}{8w}(\frac{s_{11}^E}{d_{31}})^2$$

$$R_s = \frac{|Z|_{min}}{\sqrt{1-\omega_s^2 C_p^2 |Z|_{min}^2}}$$

$$C_p = \frac{lw}{t}\epsilon_{33}^T(1-k_{31}^2)$$

Using these relations, and the parameter values obtained from the table above, we are able to compute the values of the model. By replacing the parameter values, it can be shown that L_s=3.5 mH, C_s=0.23 pF, R_s=3.8 kΩ and C_p=1.25 pF (Based on the estimated device area of 250 μm*50 μm. These values can now be replaced in the BVD model to estimate the loading. It is interesting to note that since we shall be using the piezoelectric material close to its resonance frequency, the net impedance seen will be R_s parallel to the impedance offered by C_p at that frequency. This is because at resonance the L_s and C_s impedance components tend to cancel each other out. This simplifies our BVD model at resonance, making it easier to understand the loading effect of the PZT-5H.

6.6.4. RESULTS

From the parameter values obtained, we can now simulate the device on Cadence to see if the driver can drive the piezoelectric transducer with the estimated parameter values. On doing so, we get the following fig. 68:

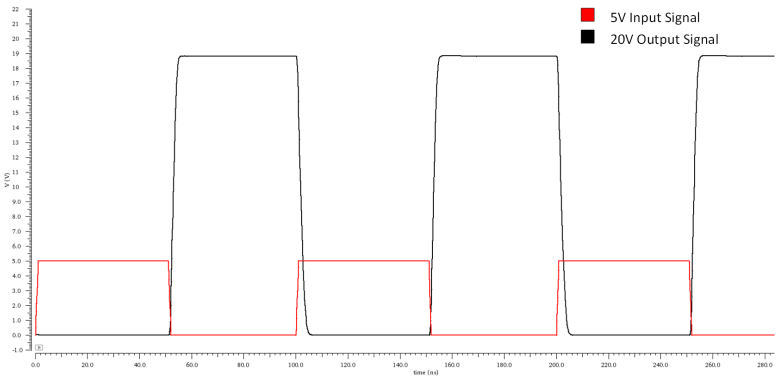


Figure 68: Driver Output with transducer load

6.7. CONTROL LOGIC

The final block in the system is the digital control unit that generates the 3-bit digital code required by the system multiplexers. Since we always want to minimize the cable count in this application, using a parallel data input is not ideal. Instead, a serial input is better, which can then be moved along a shift register, and read in a parallel manner (Serial In Parallel Out).

We also need to use the system to stimulate regions of the brain for time periods in the millisecond to seconds range. To do so, the output codes of the shift registers must be available for long intervals of time and should not be shifted out of the register after 3 clock cycles. This can be done by using a multiplexer as shown in fig. 69: When the en-

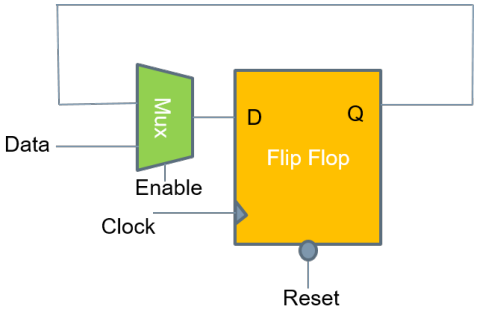


Figure 69: D Flip Flop with enable for continuous code generation

able signal is inactive, the data input is directly fed into the D flip flop via the multiplexer. When the enable signal is active, the output of the flip flop is directly fed back to the input via the multiplexer, thus preserving the output until the reset is active, or the enable signal goes inactive. As an example, to generate the 001 code, we can set the timing on the data input and enable signals to create the desired output from the shift register as shown in fig. 70. The shift register is implemented using a stack of three D Flip Flops like the ones used in the quadrature generator talked about in section 6.1.5.

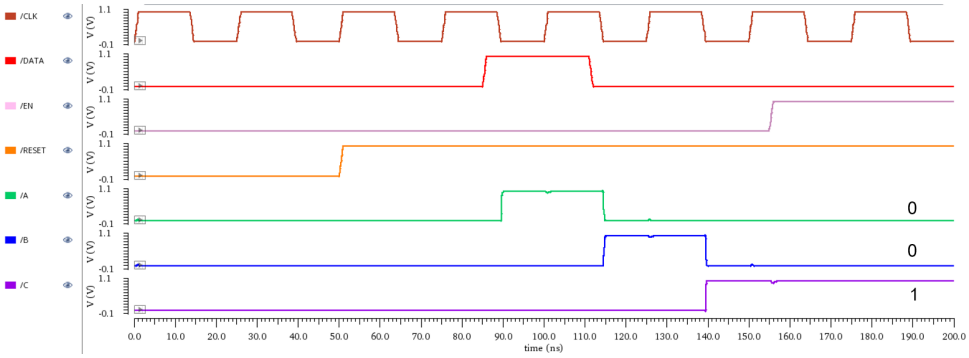


Figure 70: Generating the digital codes

7

RESULTS

This section attempts to implement the entire system described in the previous chapter and observe the results generated from it. Firstly the whole system is converted to the actual device layout and from there post layout simulations are done to see how the system behaves in terms of area, linearity, process variations and power consumption.

7.1. POST-LAYOUT SIMULATIONS

7.1.1. LAYOUT

To implement the device talked about in the prior section, first, the layout of the system is planned, exactly how we want the device to appear post fabrication. As mentioned in section 4.3, the device should be fit into the area of 250 μm* 50 μm. During each block implementation in layout, care must be taken to ensure this constraint is met. We also need to keep in mind the fact that the high voltage devices need to be kept isolated from the low voltage devices. Due to the special BCD technology being used, there are a number of extra design rules that have to be satisfied when using high voltage devices like DMOSs and HV diodes. Most of these design rules incorporate large separation distances between HV components, making it tricky to ensure a tight form factor. We thus start with floor-planning the layout, by keeping the low voltage components at the base of the structure and then stacking up the blocks one at a time. In doing so, we can arrive at the layout shown in fig. 71. From this layout, we see that we are occupying a total area of roughly 143 μm * 52 μm, which is much smaller than the initial intended area requirement.

7.1.2. OUTPUT WAVEFORMS

Now that we see that our channel is able to meet the area constraint, we are now in a position to check the post layout simulations to see if the channel performs as expected. As a start, we do a post-layout simulation for one arbitrary code that corresponds to a certain delay. This simulation is shown in fig. 72. The figure shows the intermediate nodes along with the final driver output. The intermediate 1V beamformer signal, along

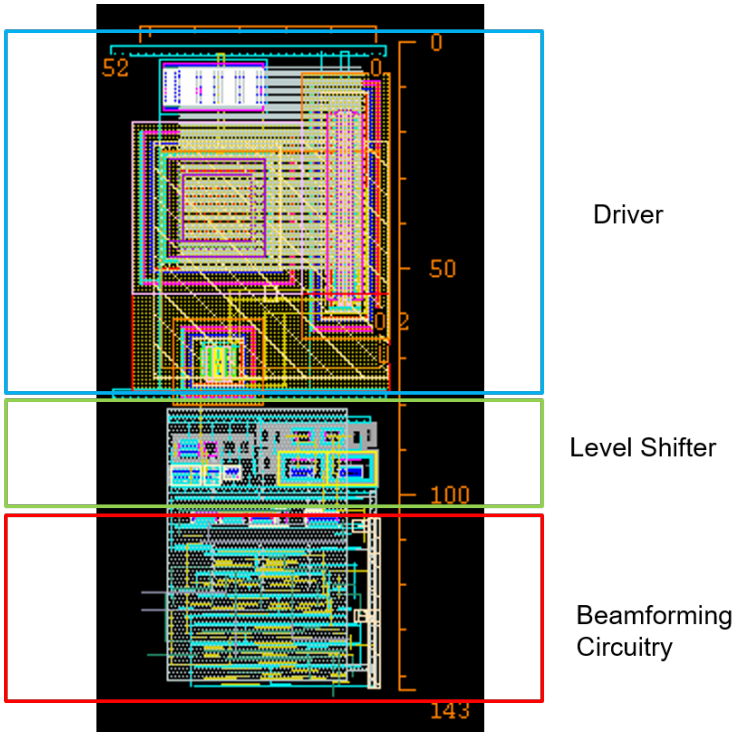


Figure 71: Full Layout

with the 5V level shifted outputs are shown.

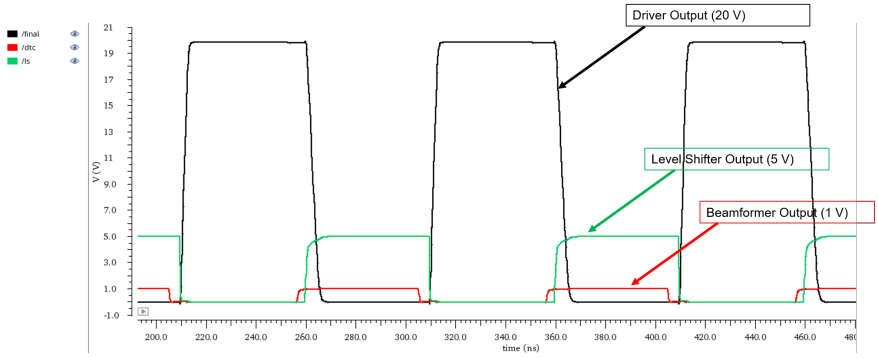


Figure 72: Final output with intermediate nodes

7.1.3. LINEARITY

Linearity is one of the most crucial aspects of the system. This is because when implemented in the full array, we need all the delays to be generated according to the digital

code provided. A one-step increase in the digital code should correspond to one LSB (12.5 ns) increase in the delay generated by that particular channel. In our study, all the delays are relative to each other, and to set a reference for comparison, all the delays generated, are computed relative to the zero delay case corresponding to code 000.
The fact that we need to ensure the relative delays are linear makes it more robust from a design perspective as any collective inter-channel process variation happening to all the channels will have no effect on the final linearity (however intra-channel process variations are still very important to consider). The final output waveforms are plotted for all the digital codes one after the other, and the delays with respect to code 000 are computed and plotted. The output of all 8 digital codes for a single pulse (for ease of readability) are plotted together in the following figure fig. 73: From this figure, we see

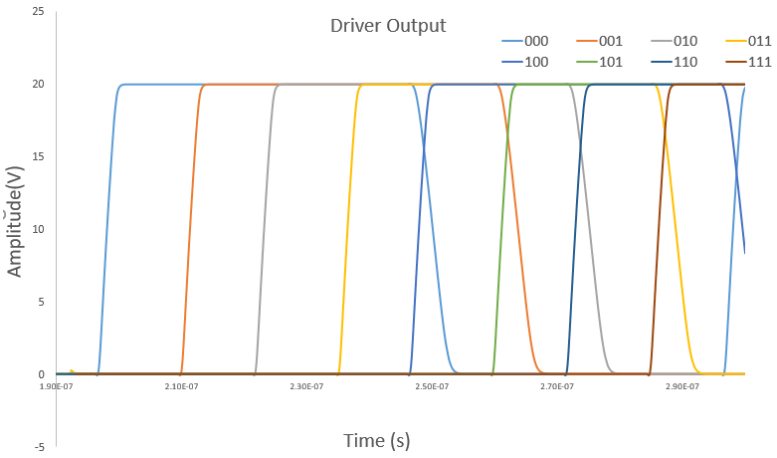


Figure 73: All digital code outputs

that the outputs are spaced out quite accurately. A better estimate of the linearity is visible from the following fig. 74 where the relative delays for each code are plotted for all the 8 digital codes and across all possible process corners. From this, we see a very good linearity from the system that is capable of generating a minimum delay of 12.5 ns, up to 87.5 ns in steps of 12.5 ns. The system is also very resistant to process corners. An interesting observation is the fact that the quadrature delays coming directly from the quadrature generator of the beamforming block, are very consistent across the corners, while the delays generated from the delay generator block show a slight deviation from ideal, across corners. The worst-case deviation from ideal, across corners is 2.6 ns and is considerably smaller than the system LSB of 12.5 ns, making the system very robust and linear across the process corners.

7.1.4. MONTE CARLO SIMULATIONS

Another important and realistic test is the Monte Carlo simulation that gives a statistical approach using random variations, to see how much the system varies from ideal. We use the code 001, to check how much the delay and duty cycle vary under these statistical random variations. The results of the monte-carlo test for 1000 samples are shown below

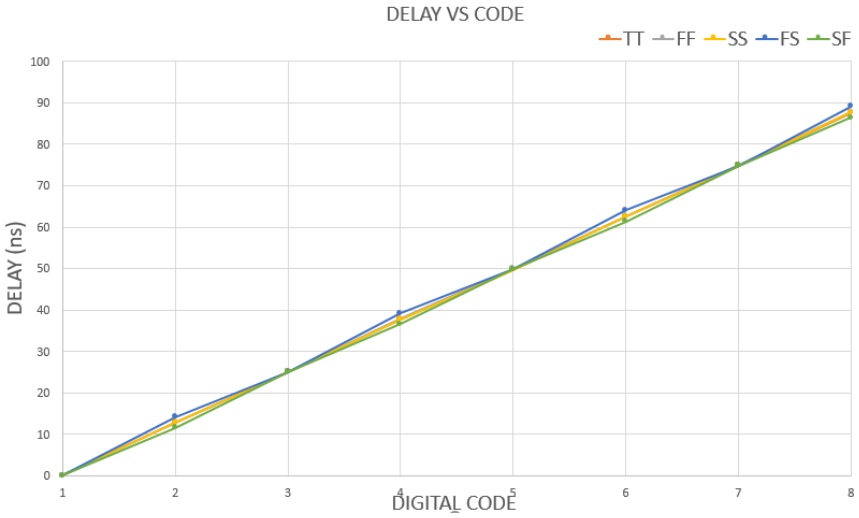


Figure 74: Linearity Plot

in fig. 75 From this, we see the LSB delay centred around 12.7 ns with a standard deviation of only 491 ps. The duty cyle is also centred around 50%, with a standard deviation of 0.3%. This adds to the claim of the system being stable and robust for this particular application.

7

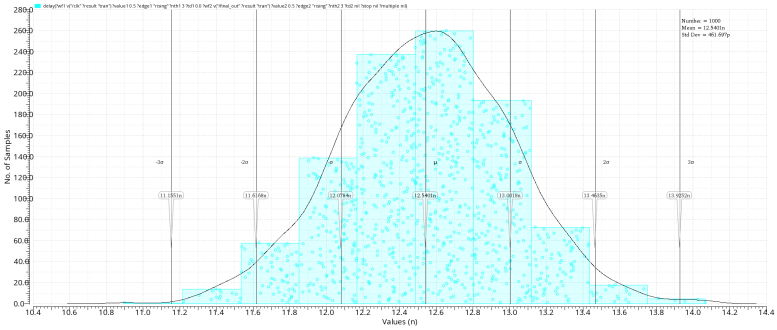
7.1.5. POWER CONSUMPTION

It would also be interesting to observe the individual power consumptions from all the blocks to estimate the possible sources of large power loss and attempt to mitigate them in the future. The power consumptions of the major blocks are shown in the table below:

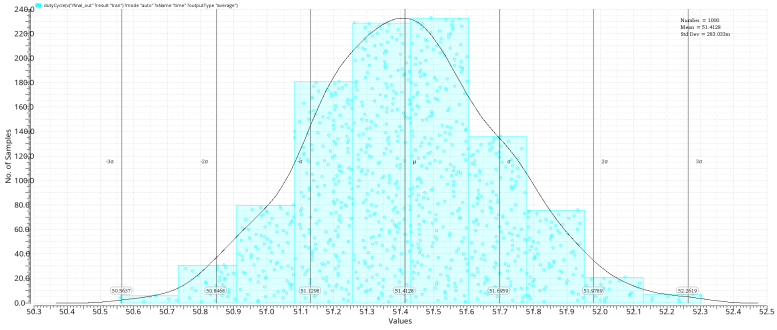
Block	Average Power Consumption
Beamformer	20 μW
Level Shifter	220 μW
Driver	12 mW

From this, we see that the driver intuitively consumes the highest power. By keeping our beamforming block at a low supply voltage of 1V, we hav reduced the power consumption from it considerably. The level shifter includes 5V buffers and has a relatively low power consumption as well. Further, it is interesting to note the power consumption from the sub-blocks of the beamforming block. This is shown in the pie chart of fig. 76

It is interesting to note that the core of the beamformer i.e. the quadrature generator consumes a minimal power of 0.9 μW, while the fine delay generator consumes the maximum power of 16 μW owing to the bias block i.e. the beta multiplier. The current passing through the beta multiplier can be lowered even further by increasing the resistor, thereby reducing the power consumption.



(a) Delay Monte Carlo Simulations



(b) Duty Cycle Monte Carlo Simulations

Figure 75: Monte Carlo Simulations

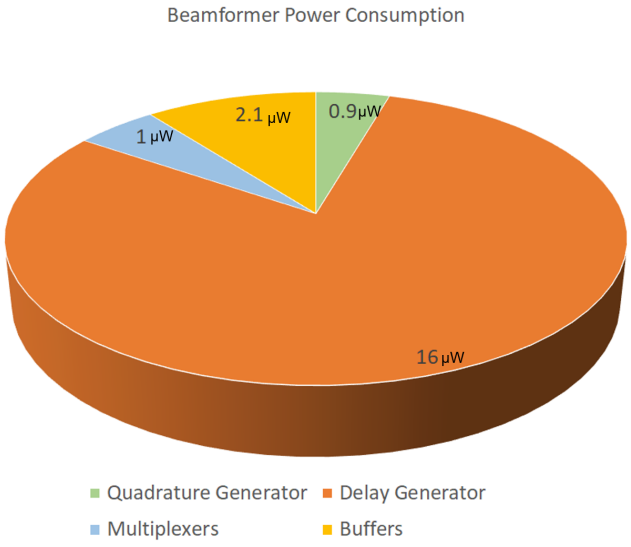


Figure 76: Beamformer sub-block power consumption

8

MEASUREMENT

In order to evaluate the system design, we are left with testing the device and verifying whether it matches the desired performance. Once the chips arrived from TSMC, a PCB board was designed to facilitate the chip requirements like stable power and the input/ output points. An FPGA was also programmed to send the correct inputs to the chip. The whole measurement process has been discussed in this chapter.

8.1. CHIP FLOORPLANNING

THE chip was surrounded by the I/O pads with ESD protection circuits inbuilt in them. Since a number of intermediate nodes of the chip are tapped in order to test the functioning, care must be taken to ensure enough drive strength is provided. The core chip circuitry is not capable of driving large loads, thus extra output buffers were designed around the nodes to be tapped. Decoupling capacitors were also added to ensure a stable power supply reaches the chip. The overall floorplan is shown in fig. 77 The chip obtained from the foundry is made according to this floorplan and is wirebonded according to the 28-DIP package being used. The chip micrograph is shown in fig. 78

8.2. PCB DESIGN

After the chip was sent for fabrication, we build a test PCB in order to ensure the correct signals are being sent to the chip, while also supplying the correct and stable power supplies. The outputs of the chip must also be tapped from the critical nets. The chip requires stable power and the outputs of the chip are not strong enough to drive large capacitative loads of the PCB and oscilloscope lines. Thus we need output buffers as well. The output of the FPGA also needs to be shifted from the 3.3 V domain to the 1V domain that can be sent to the chip. There is also thus, a requirement for a level shifter. We start with a setup incorporating all the necessary components in order to test the chip. Fig. 79 shows the setup in mind. The FPGA is powered by a USB connector to a workstation. The buffer is driven by 3.3V (as the FPGA outputs are at 3.3V) thus an LDO is used to convert the 5V VDD to a stable 3.3V. This 3.3V is then fed to another LDO to generate the 1V

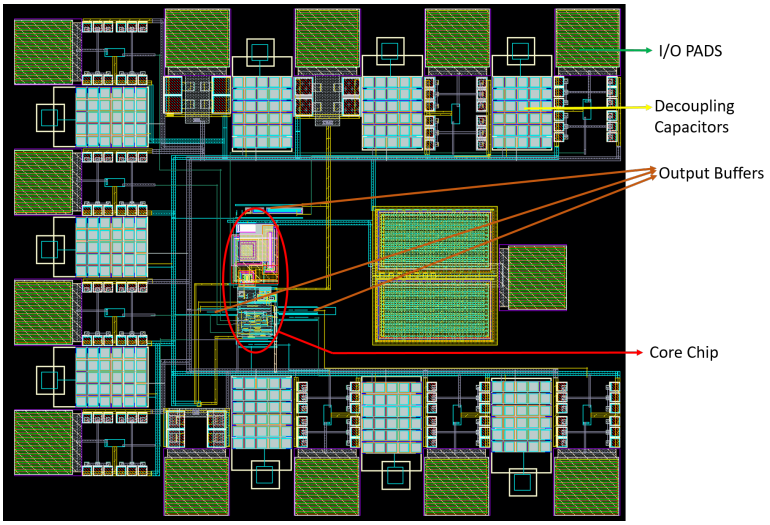


Figure 77: Complete Chip Floorplan

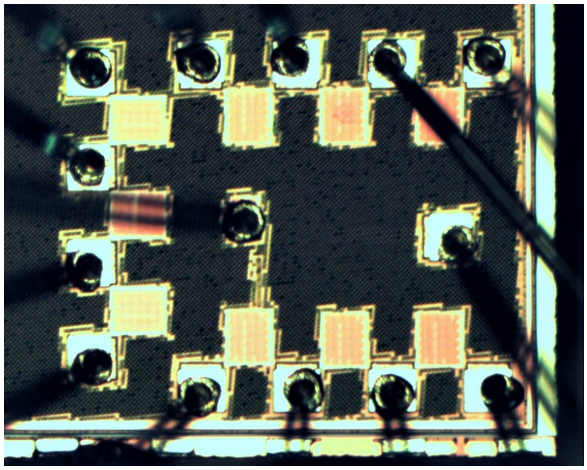


Figure 78: Chip Micrograph

supply. The power supplies of 5V and 20V are obtained directly from the external supply and then passed through LDOs as well. The level shifter converts the input signal voltage level from 3.3V (FPGA voltage level) to 1V (IC voltage level). There are also test points set up to directly read the output signals and intermediate nodes. The output buffers are also set up to ensure enough drive strength. The final output is ideally meant to be probed from the top of the IC (A pad has been made for this on top of the IC using metal 6). However, a probe with low enough capacitance and tolerance of high voltages is not available at the moment, thus we chose to wirebond the pad to the IC socket as a backup. However, the buffer for this output needs to have a low enough capacitance (since the

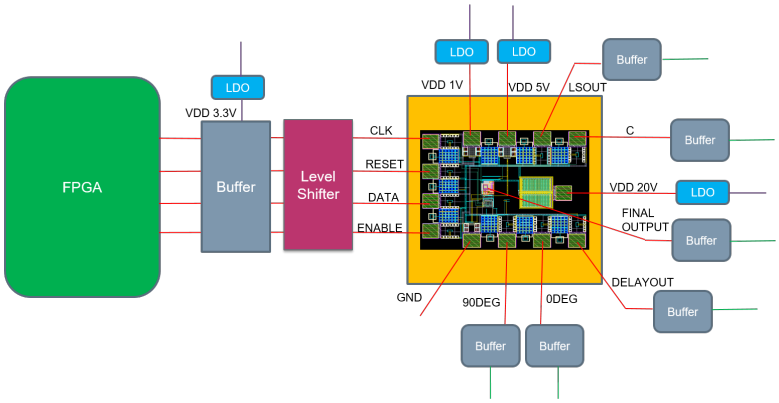


Figure 79: PCB Setup

HV driver was designed to drive a PZT load of 1.25 pF capacitance). The correct buffer was chosen with a low input capacitance of 1.5 pF.

8.3. FPGA PROGRAMMING

The FPGA is used to generate the correct sequence of inputs in order to generate the correct phase relationships in the digital control unit present in the chip. Since we use a serial input for the shift register present in the chip, special care must be taken to ensure the properly timed data and enable and reset signals are sent to the shift register as they must all be synchronised with the input clock being generated by the FPGA. This is achieved using the FPGA Cmod S7 with a Xilinx Spartan S7 core. The FPGA was programmed with the Vivado software using Verilog.

8.3.1. FINITE STATE MACHINE

In order to implement the timed signals for all 8 output signals, a Finite State Machine (FSM) was used to generate the signals. The push buttons on the FPGA were used to signal a switch between states. Fig. 80 shows the functioning of the state machine and the corresponding movement between subsequent states.

From the figure, we can see that the FSM cycles between all 8 states one after the other. The push button A acts as a trigger to move between one state to the subsequent state. The push button B allows the system to switch to an idle state where the whole system is reset. Since we do not need to cycle back to a previous state in order to check the working of the chip, we have only uni-directional control going from state 1 to state 8.

An example of the timed signals simulated on Vivado, for the code 001 or State 2 of the FSM is shown in fig. 81. From fig. 81a, we see from cadence simulations that the reset, data and enable inputs have to be timed properly with respect to the 40 MHz clock, in order to generate the ABC output bits of 001 respectively. The same required synchronous signals can be generated in our finite state machine, as shown in fig. 81b

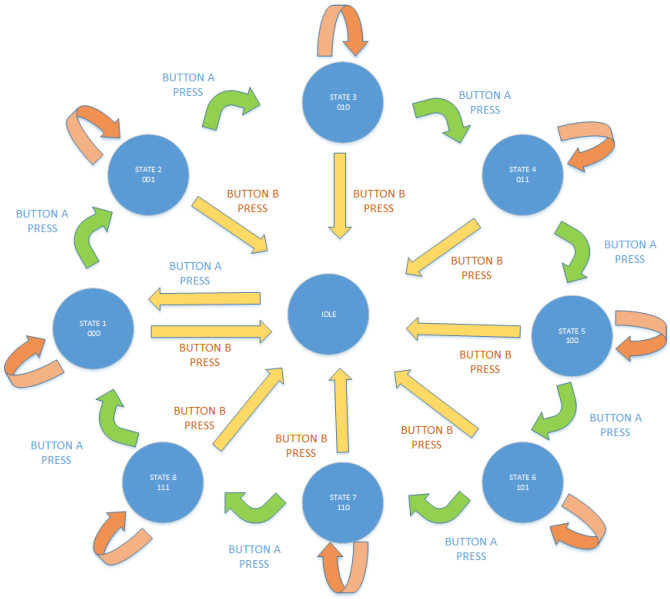
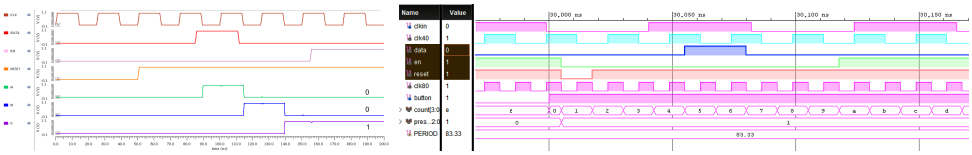


Figure 80: Finite State Machine



(a) Cadence Simulations for 001 code

(b) Vivado Simulations for 001 Code

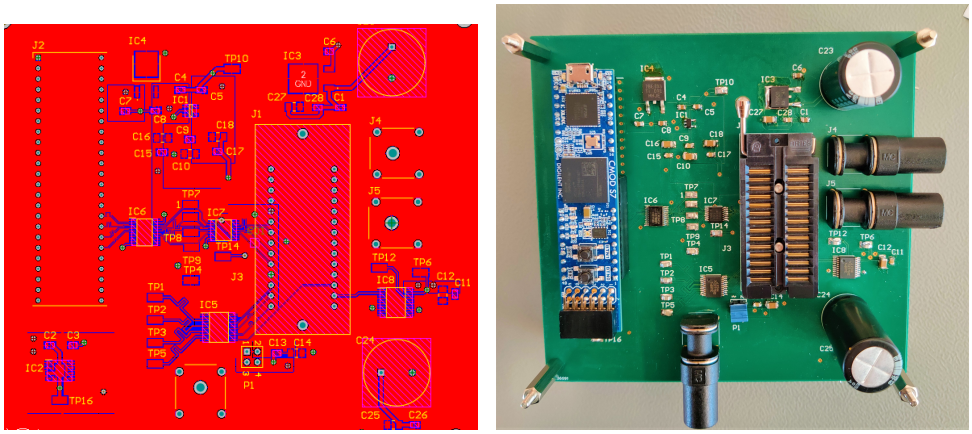
Figure 81: Digital Code Generation

8

with the waveforms for the reset, data and enable highlighted and matching the cadence waveforms. The finite state machine uses the input clock of the cmod S7 FPGA at 12 MHz and generates two clocks of 40 MHz and 80 MHz from it. The 40 MHz clock is the final clock that enters the chip and all signals are synchronised to it. The 80 MHz clock is used as the input to a counter that is used to position the rising and falling edges of reset, data and enable. As seen in fig. 81b, the button input is used to change states (here seen state switching from 0 to 1 at the rising edge of the button).
The same process is then repeated for the other digital codes. The required timed signals for all the digital codes are observed on Cadence and then translated to Vivado using Verilog.

8.4. COMPLETE SETUP

The PCB was designed on Altium using a 4 layer board and the following fig. 82 shows the final PCB after placing all the components.



(a) PCB Layout on Altium

(b) Final PCB after soldering

Figure 82: Test PCB

8.5. MEASUREMENT RESULTS

Once the chip is placed on the Zero Insertion Force (ZIF) IC socket, we power up the board and use test points made on the PCB to check the desired output. The major test points that are checked are:

1. Quadrature Signals (0° and 90°)
2. Level Shifter Output
3. Final Driver Output

8.5.1. QUADRATURE OUTPUTS

The first thing to check is the quadrature output signals generated from the chip. The 0° and 90° output waveforms are shown below in fig. 83:

8.5.2. LEVEL SHIFTER OUTPUT

The level shifter is required to convert the beamformer output to the required 5V level to interface with the high voltage driver as mentioned in prior sections. The output of the level shifter along with its corresponding input are shown in fig. 84

8.5.3. FINAL DRIVER OUTPUT

A third important test point was kept at the final output of the channel. As mentioned before, due to a lack of suitable probes, the output was wirebonded to a pad and was then sent through a 3.3 V buffer to see if an output signal is obtained (ideally the output stage would be powered by a 20V signal, however to test functionality, we pass it through a low input capacitance buffer that has a maximum working voltage of 3.3V). The output of the final output driver stage is shown in fig. 85 below. We see the driver sending out a train of pulses implying that the underlying driver circuitry is working as intended.

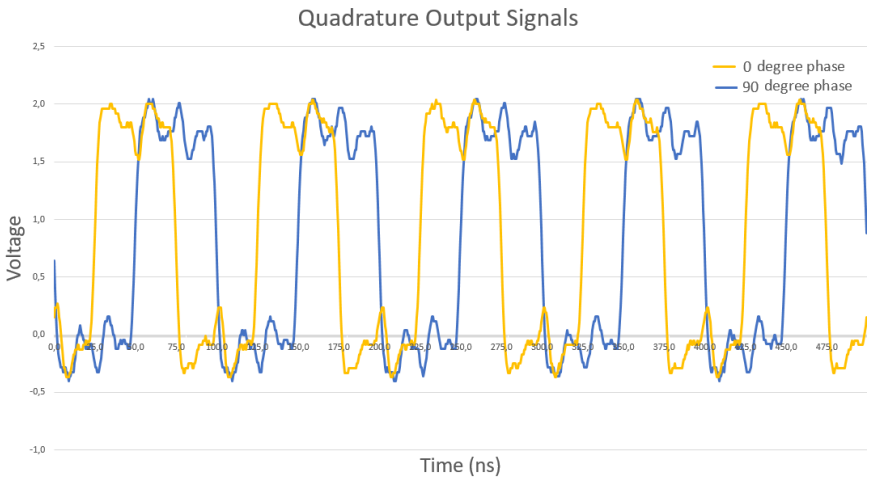


Figure 83: Measured Quadrature outputs

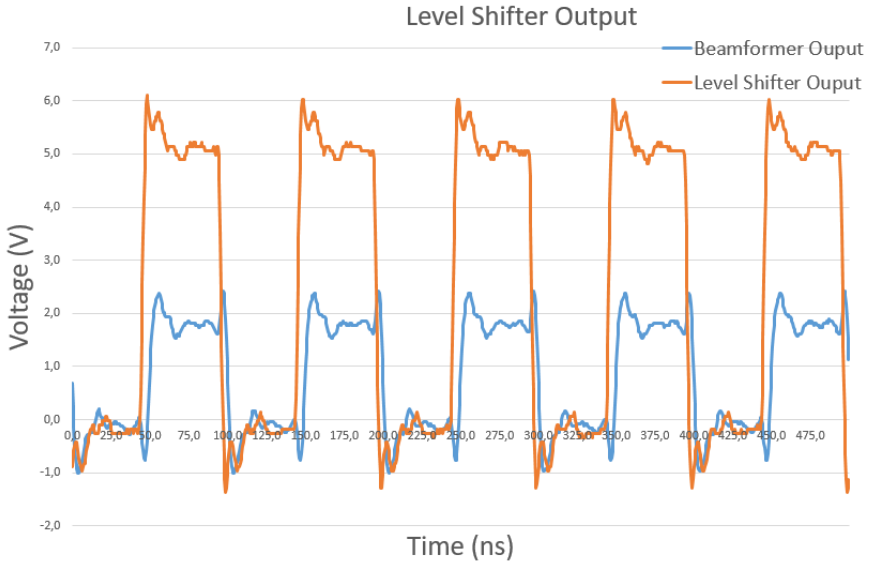


Figure 84: Measured level shifter output

As expected, since the driver is an inverter stage, the final output is an inverted version of the beamformer output, while the level shifter outputs a non inverted signal. This does not matter eventually, since all the channels will have the same inversion resulting in no additional phase difference between channel outputs.

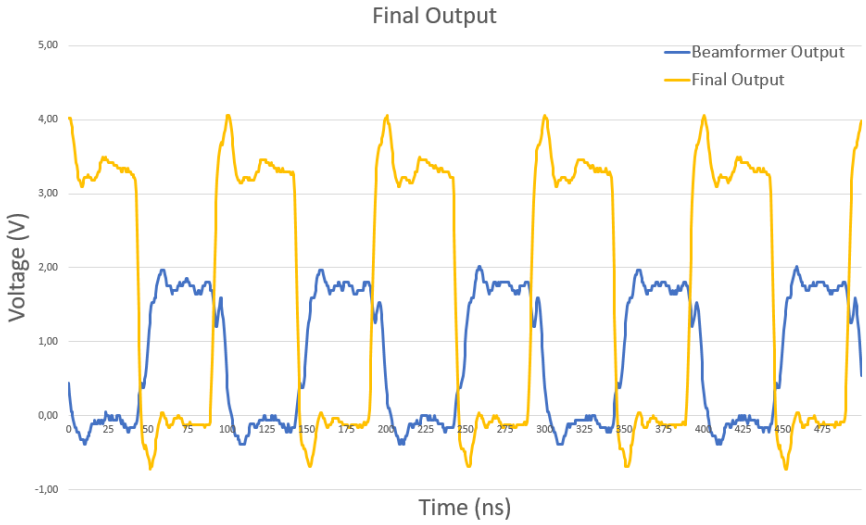


Figure 85: Measured Final output

8.5.4. DELAY GENERATOR

From the previous chapters, we know that a delay generator was employed to generate the third bit of resolution by delaying the quadrature signals and obtaining the extra 45° shift. The output of the delay generator was also tapped and the results are shown in fig. 86. This measured output can be directly compared with the expected simulated output shown in fig. 57.

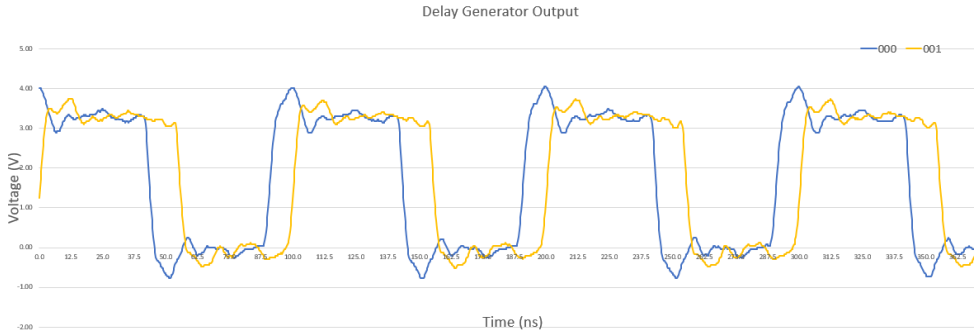


Figure 86: Measured Delay Generator Outputs

8.5.5. ALL PHASE GENERATION

From fig. 83 we see the quadrature generator working as expected, and from fig. 86, we see the fine delay block generating the required 45° phase offset. The final test done in the thesis, is to check the phase generations for all the different input digital codes. The results of the same are shown in the fig. 87 below. This can be compared to the post

layout simulation shown in fig. 73.

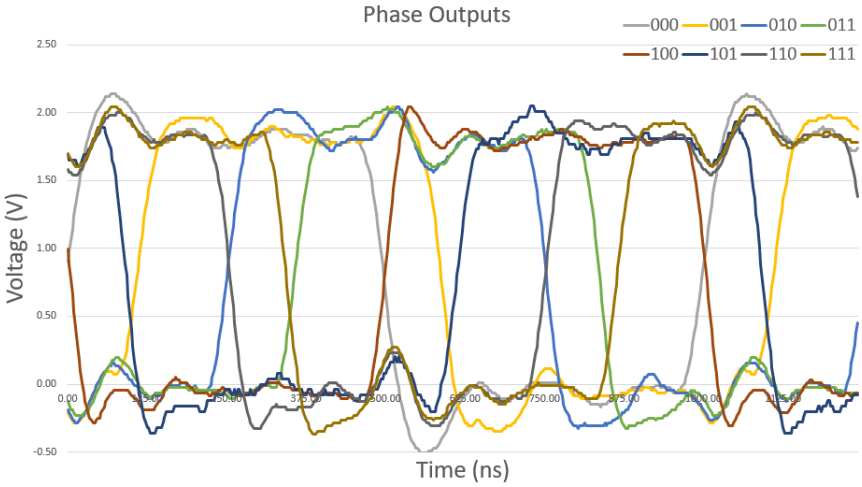


Figure 87: All phase outputs

8.5.6. LINEARITY

Now that we have the phases, we can check how closely they follow the expected results. In terms of linearity, the delay between outputs with respect to the 0° phase are calculated and plotted with respect to the corresponding digital codes.The linearity plot obtained is shown below in fig. 88. This can again be compared with the post layout simulation done prior in fig. 74,

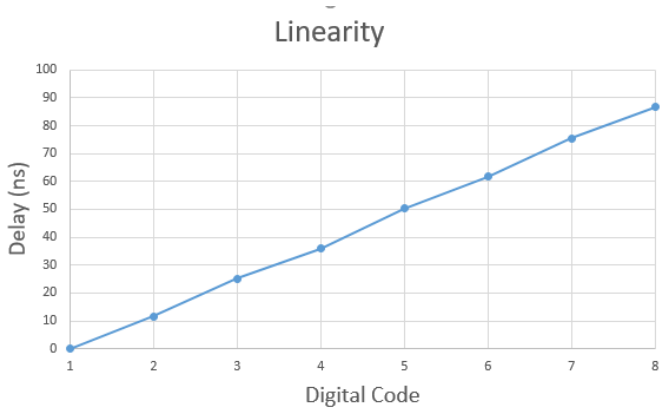


Figure 88: Measured Linearity

From the linearity plot, we see that the phases generated match the expected results quite well. The maximum deviation from ideal is around 2 ns which is much smaller that

the LSB of the system (12.5 ns) making it a robust system for phase generation.

By comparing the measured results with the expected post layout simulations, we see that the fabricated device behaves just as expected and is capable to be used in the future for low area transmit beamforming applications.

9

CONCLUSION

The work done has been described in detail starting from the problem statement and associated special constraints of the problem. Following this, a literature study was discussed and the current architectures were discussed. After this, a novel architecture that satisfied the given constraints was discussed. Finally, the performance of the designed chip was measured and tested.

ULTRASOUND beamforming is a new emerging platform for alleviating a number of mental disorders ranging from depression to Parkinsons disease. A new approach of an Ultrasonic Neurovascular Stent is discussed in the thesis. The associated issues and constraints of such a neuromodulation modality are discussed in detail. We started with MATLAB and COMSOL simulations to find the optimal number of bits required to obtain sufficient resolution for the 1-D array. We finally arrive at a novel transmit IC channel that occupies a very small area of 143 μm* 52μm and is capable of generating 8-phase shifted outputs corresponding to the 3 bits of resolution required for the system to beamform effectively. The novel beamformer design has been shown to be tolerant to process variations while also consuming a very low power, compared to other beamforming architectures used in the present scenario.

9.1. THESIS CONTRIBUTIONS

The thesis involved:

1. Going through the available literature on ultrasound transmit beamforming systems.

2. MATLAB and COMSOl simulations of a 1-D array of channels to obtain the desired bit resolution for sufficient beam steering and beam focussing.

3. Developing a novel IC channel architecture for low area and low power constraints of the neuro-stent.

4. CMOS level Schematic and verification on Cadence using corners and Monte Carlo simulations on post-layout design.

5. Taped out the chip and measured the preliminary performance. PCB design was done using Altium and the FPGA was programmed on Vivado in order to send the correct input signals.

9.2. FUTURE WORK

Since this is the first working prototype for the Neuro-Stent in mind, there are a number of issues that need to be addressed in the subsequent work:

1. The high voltage driver consumes a large amount of power. This is due to the fact that the HV MOSFETs were sized very large in order to drive the PZT transducer effectively. However, the PZT was modelled mathematically using numerical equations and is known to deviate from practical results. The actual load capacitance and resistance are expected to be much lower and higher respectively and this can only be verified using an impedance analyzer.

2. This thesis covers the complete channel architecture, however, the array is a culmination of such channels and requires some special considerations. For example, currently the channel has no hard-coded address specific to each channel. There is a need for such an address so that the serial data input can be configured to each channel separately. It is reassuring to know that we still have considerable area budget left on our chip ($143\mu m * 52\mu m$ consumed out of the budget of $250\mu m * 50\mu m$) and are fully capable of adding an address block to the system.

3. In order to complete the system, the IC has to be integrated with the piezoelectric transducer. This is a fabrication challenge that needs to be addressed separately.

This is an exciting new avenue of research that deals with a neurovascular stent with ultrasound transmit capability. With a working channel prototype having a low area and power consumption, capable of sufficient resolution, we are now in a position to explore multiple avenues in terms of driving voltages, total number of channels, extra calibration functionality, receiver sub-arrays for imaging etc. The possibilities are limitless and must be looked into in the future, to arrive at a Neuro-Stent capable of making human health better and raise the standard of life.

BIBLIOGRAPHY

[1] Andres M Lozano et al. “Deep brain stimulation: current challenges and future directions”. In: *Nature Reviews Neurology* 15.3 (2019), pp. 148–160.

[2] SJ Groiss et al. “Deep brain stimulation in Parkinson’s disease”. In: *Therapeutic advances in neurological disorders* 2.6 (2009), pp. 379–391.

[3] Günther Deuschl et al. “Treatment of patients with essential tremor”. In: *The Lancet Neurology* 10.2 (2011), pp. 148–161.

[4] René Reese and Jens Volkmann. “Deep brain stimulation for the dystonias: evidence, knowledge gaps, and practical considerations”. In: *Movement disorders clinical practice* 4.4 (2017), pp. 486–494.

[5] Candace Borders et al. “Deep brain stimulation for obsessive compulsive disorder: A review of results by anatomical target”. In: *Mental illness* 10.2 (2018).

[6] Nasser Zangiabadi et al. “Deep brain stimulation and drug-resistant epilepsy: a review of the literature”. In: *Frontiers in neurology* 10 (2019), p. 601.

[7] Sibylle Delaloye and Paul E Holtzheimer. “Deep brain stimulation in the treatment of depression”. In: *Dialogues in clinical neuroscience* 16.1 (2014), p. 83.

[8] Holly A Roy and Tipu Z Aziz. “Deep brain stimulation and multiple sclerosis: Therapeutic applications”. In: *Multiple sclerosis and related disorders* 3.4 (2014), pp. 431–439.

[9] Mayur Sharma and Milind Deogaonkar. “Deep brain stimulation in Huntington’s disease: assessment of potential targets”. In: *Journal of Clinical Neuroscience* 22.5 (2015), pp. 812–817.

[10] Rodolfo Savica et al. “Deep brain stimulation in tourette syndrome: a description of 3 patients with excellent outcome”. In: *Mayo Clinic Proceedings*. Vol. 87. 1. Elsevier. 2012, pp. 59–62.

[11] Anton Fomenko et al. “Low-intensity ultrasound neuromodulation: An overview of mechanisms and emerging human applications”. In: *Brain stimulation* 11.6 (2018), pp. 1209–1217.

[12] Bashar W Badran et al. “Sonication of the anterior thalamus with MRI-Guided transcranial focused ultrasound (tFUS) alters pain thresholds in healthy adults: A double-blind, sham-controlled study”. In: *Brain Stimulation* 13.6 (2020), pp. 1805–1812.

[13] William J Tyler, Shane W Lani, and Grace M Hwang. “Ultrasonic modulation of neural circuit activity”. In: *Current opinion in neurobiology* 50 (2018), pp. 222–231.

[14] Carolina Mora Lopez et al. “A Neural Probe with Up to 966 Electrodes and Up to 384 Configurable Channels in 0.13um SOI CMOS”. In: *IEEE transactions on biomedical circuits and systems* 11.3 (2017), pp. 510–522.

[15] Roberta K Sefcik et al. “The evolution of endovascular electroencephalography: historical perspective and future applications”. In: *Neurosurgical focus* 40.5 (2016), E7.

[16] Jill Novitzke. “A patient guide to brain stent placement”. In: *Journal of vascular and interventional neurology* 2.2 (2009), p. 177.

[17] Thomas J Oxley et al. “Minimally invasive endovascular stent-electrode array for high-fidelity, chronic recordings of cortical neural activity”. In: *Nature biotechnology* 34.3 (2016), pp. 320–327.

[18] NL Opie et al. “Focal stimulation of the sheep motor cortex with a chronically implanted minimally invasive electrode array mounted on an endovascular stent. Nat Biomed Eng 2018; 2: 907e14”. In: *Brain Stimulation* 13 (2020), pp. 401–402.

[19] Thomas J Oxley et al. “Motor neuroprosthesis implanted with neurointerventional surgery improves capacity for activities of daily living tasks in severe paralysis: first in-human experience”. In: *Journal of neurointerventional surgery* 13.2 (2021), pp. 102–108.

[20] Thomas J Oxley et al. “Minimally invasive endovascular stent-electrode array for high-fidelity, chronic recordings of cortical neural activity”. In: *Nature biotechnology* 34.3 (2016), pp. 320–327.

[21] Thomas J Oxley et al. “Motor neuroprosthesis implanted with neurointerventional surgery improves capacity for activities of daily living tasks in severe paralysis: first in-human experience”. In: *Journal of neurointerventional surgery* 13.2 (2021), pp. 102–108.

[22] Hesam Sadeghi Gougheri et al. “A comprehensive study of ultrasound transducer characteristics in microscopic ultrasound neuromodulation”. In: *IEEE transactions on biomedical circuits and systems* 13.5 (2019), pp. 835–847.

[23] Mathew Legg and Stuart Bradley. “Ultrasonic arrays for remote sensing of pasture biomass”. In: *Remote Sensing* 12.1 (2020), p. 111.

[24] Yijun Shi. “Modeling of acoustic waves for linear phased arrays”. PhD thesis. Massachusetts Institute of Technology, 1998.

[25] Shi-Chang Wooh and Yijun Shi. “Optimization of ultrasonic phased arrays”. In: *Review of progress in quantitative nondestructive evaluation*. Springer, 1998, pp. 883–890.

[26] Tiago Costa et al. “A CMOS 2D transmit beamformer with integrated PZT ultrasound transducers for neuromodulation”. In: *2019 IEEE Custom Integrated Circuits Conference (CICC)*. IEEE. 2019, pp. 1–4.

[27] Chunkyun Seok et al. “Towards an untethered ultrasound beamforming system for brain stimulation in behaving animals”. In: *2018 40th annual international conference of the IEEE engineering in medicine and biology society (EMBC)*. IEEE. 2018, pp. 1596–1599.

[28] Chunkyun Seok et al. “A Sub-Millimeter lateral resolution ultrasonic beamforming system for brain stimulation in behaving animals”. In: *2019 41st Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*. IEEE. 2019, pp. 6462–6465.

[29] Chunkyun Seok et al. “A front-end integrated circuit for a 2D capacitive micromachined ultrasound transducer (CMUT) array as a noninvasive neural stimulator”. In: *2017 IEEE International Ultrasonics Symposium (IUS)*. IEEE. 2017, pp. 1–4.

[30] Omid Farhanieh et al. “Integrated HIFU drive system on a chip for CMUT-based catheter ablation system”. In: *IEEE transactions on biomedical circuits and systems* 11.3 (2017), pp. 534–546.

[31] Anshuman Bhuyan et al. “Integrated circuits for volumetric ultrasound imaging with 2-D CMUT arrays”. In: *IEEE transactions on biomedical circuits and systems* 7.6 (2013), pp. 796–804.

[32] Ira O Wygant et al. “Integration of 2D CMUT arrays with front-end electronics for volumetric ultrasound imaging”. In: *IEEE transactions on ultrasonics, ferroelectrics, and frequency control* 55.2 (2008), pp. 327–342.

[33] Anshuman Bhuyan et al. “Integrated circuits for volumetric ultrasound imaging with 2-D CMUT arrays”. In: *IEEE transactions on biomedical circuits and systems* 7.6 (2013), pp. 796–804.

[34] Gwangrok Jung et al. “A reduced-wire ICE catheter ASIC with Tx beamforming and Rx time-division multiplexing”. In: *IEEE transactions on biomedical circuits and systems* 12.6 (2018), pp. 1246–1255.

[35] Mohammad Maadi and Baris Bayram. “Custom integrated circuit design for ultrasonic therapeutic CMUT array”. In: *Microsystem Technologies* 21.4 (2015), pp. 875–891.

[36] Yutaka Igarashi et al. “Single-chip 3072-element-channel transceiver/128-subarray-channel 2-D array IC with analog RX and all-digital TX beamformer for echocardiography”. In: *IEEE Journal of Solid-State Circuits* 54.9 (2019), pp. 2555–2567.

[37] Chao Chen et al. “A Front-End ASIC With Receive Sub-array Beamforming Integrated With a 32×32 PZT Matrix Transducer for 3-D Transesophageal Echocardiography”. In: *IEEE Journal of Solid-State Circuits* 52.4 (2017), pp. 994–1006.

[38] Mingliang Tan et al. “A 64-channel transmit beamformer with$\pm$30-V bipolar high-voltage pulsers for catheter-based ultrasound probes”. In: *IEEE Journal of Solid-State Circuits* 55.7 (2020), pp. 1796–1806.

[39] Zili Yu, Michiel AP Pertijs, and Gerard CM Meijer. “A programmable analog delay line for Micro-beamforming in a transesophageal ultrasound probe”. In: *2010 10th IEEE international conference on solid-state and integrated circuit technology*. IEEE. 2010, pp. 299–301.

[40] Satoshi Kumaki et al. “A 0.5 V 6-bit scalable phase interpolator”. In: *2010 IEEE Asia Pacific Conference on Circuits and Systems*. IEEE. 2010, pp. 1019–1022.

[41] Andrew Nicholson et al. “A 1.2 V 2-bit phase interpolator for 65nm CMOS”. In: *2012 IEEE International Symposium on Circuits and Systems (ISCAS)*. IEEE. 2012, pp. 2039–2042.

[42] Po-Yu Kuo et al. “A Novel Cross-Latch Shift Register Scheme for Low Power Applications”. In: *Applied Sciences* 11.1 (2021), p. 129.

[43] Xinyu Yang. “Ultrasound Transmitter for Non-invasive Vagus Nerve Stimulation”. In: (2021).

[44] Rainer Kreienkamp et al. “A 10-Gb/s CMOS clock and data recovery circuit with an analog phase interpolator”. In: *IEEE Journal of Solid-State Circuits* 40.3 (2005), pp. 736–743.

[45] George Souliotis et al. “Phase interpolator with improved linearity”. In: *Circuits, Systems, and Signal Processing* 35.2 (2016), pp. 367–383.

[46] Andrew Nicholson et al. “A 1.2 V 2-bit phase interpolator for 65nm CMOS”. In: *2012 IEEE International Symposium on Circuits and Systems (ISCAS)*. IEEE. 2012, pp. 2039–2042.

[47] Satoshi Kumaki et al. “A 0.5 V 6-bit scalable phase interpolator”. In: *2010 IEEE Asia Pacific Conference on Circuits and Systems*. IEEE. 2010, pp. 1019–1022.

[48] Bruno W Garlepp et al. “A portable digital DLL for high-speed CMOS interface circuits”. In: *IEEE Journal of solid-state circuits* 34.5 (1999), pp. 632–644.

[49] Andreas Tsimpos et al. “All digital phase interpolator”. In: *2015 10th International Conference on Design & Technology of Integrated Systems in Nanoscale Era (DTIS)*. IEEE. 2015, pp. 1–6.

[50] Andreas Tsimpos et al. “Multi-rate phase interpolator for high speed serial interfaces”. In: *Microelectronics Journal* 54 (2016), pp. 40–47.

[51] George Souliotis et al. “Phase interpolator with improved linearity”. In: *Circuits, Systems, and Signal Processing* 35.2 (2016), pp. 367–383.

[52] Peng Chen et al. “A 31-\ mu W, 148-fs Step, 9-bit Capacitor-DAC-Based Constant-Slope Digital-to-Time Converter in 28-nm CMOS”. In: *IEEE Journal of Solid-State Circuits* 54.11 (2019), pp. 3075–3085.

[53] Peng Chen et al. “A 15-μW, 103-fs step, 5-bit capacitor-DAC-based constant-slope digital-to-time converter in 28nm CMOS”. In: *2017 IEEE Asian Solid-State Circuits Conference (A-SSCC)*. IEEE. 2017, pp. 93–96.

[54] Chao Qun Liu, Yuan Cao, and Chip Hong Chang. “ACRO-PUF: A low-power, reliable and aging-resilient current starved inverter-based ring oscillator physical unclonable function”. In: *IEEE Transactions on Circuits and Systems I: Regular Papers* 64.12 (2017), pp. 3138–3149.

[55] Shailesh Singh Chouhan and Kari Halonen. “Ultra low power beta multiplier-based current reference circuit”. In: *Analog Integrated Circuits and Signal Processing* 93.3 (2017), pp. 523–529.

[56] Henri J Oguey and Daniel Aebischer. “CMOS current reference without resistance”. In: *IEEE Journal of Solid-State Circuits* 32.7 (1997), pp. 1132–1135.

[57] Shailesh Singh Chouhan and Kari Halonen. “A modified CMOS nano-power resistorless current reference circuit”. In: *2014 10th Conference on Ph. D. Research in Microelectronics and Electronics (PRIME)*. IEEE. 2014, pp. 1–4.

[58] Manoj Kumar, Sandeep K Arya, and Sujata Pandey. “Level shifter design for low power applications”. In: *arXiv preprint arXiv:1011.0507* (2010).

[59] Canh Q Tran, Hiroshi Kawaguchi, and Takayasu Sakurai. “Low-power high-speed level shifter design for block-level dynamic voltage scaling environment”. In: *2005 International Conference on Integrated Circuit Design and Technology, 2005. ICICDT 2005*. IEEE. 2005, pp. 229–232.

[60] Marco Lanuzza, Pasquale Corsonello, and Stefania Perri. “Low-power level shifter for multi-supply voltage designs”. In: *IEEE Transactions on Circuits and Systems II: Express Briefs* 59.12 (2012), pp. 922–926.

[61] Anshuman Bhuyan et al. “Integrated circuits for volumetric ultrasound imaging with 2-D CMUT arrays”. In: *IEEE transactions on biomedical circuits and systems* 7.6 (2013), pp. 796–804.

[62] Iván Zamora et al. “Miniaturized 0.13-μm CMOS front-end analog for AlN PMUT arrays”. In: *Sensors* 20.4 (2020), p. 1205.

[63] Chunkyun Seok et al. “A 2D Ultrasonic Transmit Phased Array Based on a 32x32 CMUT Array Flip-Chip Bonded to an ASIC for Neural Stimulation”. In: *2020 IEEE International Ultrasonics Symposium (IUS)*. IEEE. 2020, pp. 1–4.

[64] Wai Wong et al. “An integrated ultrasound transducer driver for HIFU applications”. In: *2013 26th IEEE Canadian Conference on Electrical and Computer Engineering (CCECE)*. IEEE. 2013, pp. 1–5.

[65] Guo Li et al. “Investigation of High-Power Properties of PIN-PMN-PT Relaxor-Based Ferroelectric Single Crystals and PZT-4 Piezoelectric Ceramics”. In: *IEEE transactions on ultrasonics, ferroelectrics, and frequency control* 67.8 (2020), pp. 1641–1646.

A

APPENDIX

A.1. JITTER

Jitter is another important aspect to consider for clocked systems. The edge to edge jitter talks about how much the clock edges deviate from ideal as shown in fig. 90. In our open

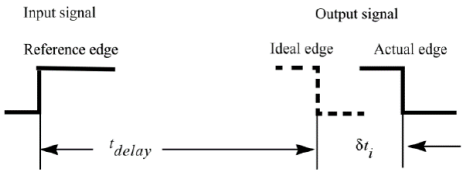


Figure 89: Edge to Edge Jitter

loop system, it is important to check the jitter performance of the system at 10 MHz. For this, the integrated phase noise is observed on cadence, and from the plot, we see that the corresponding edge to edge jitter is 2.59 ps. This is far smaller than the time period of the clock (100 ns) and can thus be neglected.

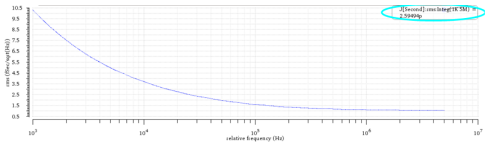


Figure 90: Jitter Performance